



# SUSE LINUX

PŘÍRUČKA SPRÁVCE SYSTÉMU

1. Vydání 2005

Copyright ©

Toto dílo je duševním vlastnictvím společností SuSE CR, s.r.o a Novell Inc. Je možné ho kopírovat jako celek nebo jeho části při dodržení povinnosti uvést na každé kopii toto upozornění o autorských právech.

Všechny programy, obrázky a informace uvedené v těchto materiálech jsou pečlivě kontrolovány, ale ani tak není možné zcela vyloučit výskyt případných chyb. Z tohoto důvodu nejsme s to nést žádné záruky jakéhokoliv druhu za případné vzniklé škody spojené s používáním této příručky. Autoři, překladatelé, ani SuSE CR, s.r.o., resp. SUSE Linux AG neposkytují žádné záruky a nenesou odpovědnost za případné škody vzniklé používáním těchto manuálů nebo programů zde uvedených uživatelům samotným nebo třetím stranám.

Všechny názvy produktů jsou bez záruky volného používání a může se jednat o registrované obchodní značky. SuSE CR, s.r.o. se obecně řídí informacemi výrobce. Jiné, zde uvedené, produkty mohou být obchodními značkami stávajících výrobců.

Poznámky a komentáře směrujte na adresu [feedback@suse.cz](mailto:feedback@suse.cz).

*Autoři:* Stefan Behlert, Frank Bodammer, Stefan Dirsch, Olaf Donjak, Roman Drahtmüller, Torsten Duwe, Thorsten Dubiel, Thomas Fehr, Stefan Fent, Werner Fink, Kurt Garloff, Carsten Groß, Joachim Gleißner, Andreas Grünbacher, Franz Hassels, Andreas Jaeger, Klaus Kämpf, Andi Kleen, Hubert Mantel, Lars Marowsky-Bree, Chris Mason, Johannes Meixner, Lars Müller, Matthias Nagorni, Anas Nashif, Siegfried Olschner, Peter Pöml, Thomas Renninger, Heiko Rommel, Marcus Schäfer, Nicolaus Schüler, Klaus Singvogel, Hendrik Vogelsang, Klaus G. Wagner, Rebecca Walter, Christian Zoz

*Překlad:* Klára Cihlářová, Jakub Friedl

*Odborná korektura:* Jörg Arndt, Antje Faber, Berthold Gunreben, Roland Haidl, Jana Jaeger, Lukáš Ocilka, Edith Parzefall, Ines Pozo, Thomas Rölz, Thomas Schraitle, Rebecca Walter

*Úprava:* Manuela Piotrowski, Thomas Schraitle

*Sazba:* DocBook-XML, L<sup>A</sup>T<sub>E</sub>X

# Předmluva

Nejdůležitější je najít požadované informace a hlavně, najít je rychle. Z tohoto důvodu jsme pro vás připravili tuto příručku obsahující základní přehled o systému, jeho nastavení některých nejdůležitějších a nejčastěji používaných aplikací.

Aby pro vás byla orientace co nejsnadnější, setřídili jsme jednotlivé kapitoly do modulů podle témat. SUSE LINUX *Příručka správce systému* se skládá z pěti hlavních částí:

**Instalace** Detaily o instalaci a nastavení např. LVM nebo RAIDového pole

**Systém** Nastavení zavaděče, X Window systému, tisku a mobilních zařízení

**Služby** Integrace heterogenních sítí, nastavení Apache, synchronizace souborů a bezpečnost.

**Správa** Informace o ACLs a důležitých monitorovacích nástrojích.

**Přílohy** Krátké přehledy všeho, co by se vám mohlo hodit

Digitální verze manuálů jsou po instalaci přístupné prostřednictvím systému nápovědy SUSE.

# Novinky v Příručce správce systému

V tomto seznamu najdete změny oproti předchozí verzi:

- Došlo ke změnám v částech věnovaných LVM dělení disku, viz část 3.7 na straně 95.
- Kapitola 8 na straně 163 byla rozšířena o popis modulu programu YaST. Obsahuje také pojednání o používání zástupných znaků (viz 8.3.1 na straně 172).
- V kapitole věnované souborovým systémům přibyl popis souborového systému Reiser4 (viz 20.2.4 na straně 349).
- Došlo ke kompletní změně částí věnovaných síťování. Najdete je od kapitoly 22 na straně 367 dále.
- Aktualizovaná kapitola o aplikaci SuSEfirewall2 nyní obsahuje také popis příslušného modulu programu YaST (viz 34.1.4 na straně 555).
- Kapitola 36 na straně 593 se rozrosta o popis několika dalších nástrojů.
- Změn se dočkal také V na straně 655.

# Nejdůležitější zdroje informací

Hlavním problémem jakéhokoliv uživatele je nalezení odpovědí na problémy. Zde jsou uvedeny některé z informačních zdrojů, které vám mohou pomoci:

- Systém nápovědy, který obsahuje SUSE LINUX s názvem *SUSE Help*. Spustit ho můžete např. pomocí menu v KDE nebo příkazem `susehelpcenter` z příkazové řádky
- Když používáte příkazovou řádku, pak používejte *manuálové stránky*, např.  
`man man`
- *Dokumentaci* k většině programů naleznete v adresáři `/usr/share/doc/název_balíku/`
- Používejte elektronickou verzi *tištěné dokumentace*. Velmi se hodí při vyhledávání klíčových slov
- Používejte internetové zdroje (`portal.suse.com` a vyhledávače, např. `http://www.google.com`)
- SUSE zdarma rozesílá emaily s bezpečnostními informacemi a tipy a triky týkajícími se systému SUSE LINUX. K odběru se můžete přihlásit zadáním své adresy na stránce `http://www.novell.com/company/subscribe/`.

## Instalační podpora

Po zakoupení plné verze máte možnost využívat bezplatnou instalační podporu. Podmínky instalační podpory v angličtině najdete na stránce

`http://www.novell.com/products/linuxprofessional/support/conditions.html`

Český překlad je dostupný na stránkách `http://www.novell.cz/suselinux`. Než se na instalační podporu obrátíte, nezapomeňte se zaregistrovat na SUSE portálu (`http://portal.suse.com`) a aktivovat své registrační číslo z přebalu CD.

# Typografické konvence

V této knize se používají následující typografické konvence:

- `/etc/passwd`: soubory nebo adresáře.
- `<Jmeno_uzivatele>`: položku `<Jmeno_uzivatele>` nahrad'te údajem platným ve svém systému.
- `PATH`: proměnné prostředí, zde `PATH`
- `ls`: příkazy.
- `--help`: volba nebo parametr.
- `user`: uživatel.
- `(Alt)`: klávesa.
- 'Soubor': tlačítka, položky nabídky atd.
- ► **x86, AMD64**  
Tento odstavec je platný pouze pro uvedenou platformu. ◀

# Poděkování

Na titulní stránce této knihy najdete seznam lidí, kteří se podíleli na tvorbě této knihy. Rádi bychom samozřejmě poděkovali všem, kdo se podíleli na vydání nové verzi SUSE LINUXu.

Samozřejmě děkujeme řadě vývojářů, kteří se podílejí na vývoji operačního systému Linux. Děkujeme jim za jejich skvělou práci - bez nich by naše distribuce nemohla existovat. Také děkujeme Franku Zappovi, Pawar a Sněhurce.

A poslední a zároveň největší dík patří panu Linusi Torvaldsovi!

Have a lot of fun!

Váš SUSE Team





# Obsah

<b>I</b>	<b>Instalace</b>	<b>1</b>
<b>1</b>	<b>Instalace pomocí programu YaST</b>	<b>3</b>
1.1	Spouštění instalačního programu . . . . .	4
1.1.1	Možnosti spuštění instalace . . . . .	4
1.1.2	Možné komplikace při startu z CD nebo DVD . . . . .	4
1.2	Úvodní obrazovka . . . . .	6
1.3	Výběr jazyka . . . . .	8
1.4	Typ instalace . . . . .	8
1.5	Návrh instalace . . . . .	9
1.5.1	Režim instalace . . . . .	10
1.5.2	Rozložení klávesnice . . . . .	10
1.5.3	Myš . . . . .	10
1.5.4	Rozdělování disku . . . . .	11
1.5.5	Software . . . . .	19
1.5.6	Konfigurace spouštění (instalace zavaděče) . . . . .	22
1.5.7	Časová pásma . . . . .	23
1.5.8	Jazyk . . . . .	24
1.5.9	Spuštění instalace . . . . .	24
1.6	Dokončení instalace . . . . .	24
1.6.1	Heslo uživatele root . . . . .	25

1.6.2	Konfigurace sítě . . . . .	26
1.6.3	Testování spojení do Internetu . . . . .	26
1.6.4	Aktualizace . . . . .	27
1.6.5	Ověřování uživatelů . . . . .	28
1.6.6	Konfigurace počítače jako NIS klienta . . . . .	29
1.6.7	Vytváření lokálních uživatelských účtů . . . . .	30
1.6.8	Čtení poznámek k verzi . . . . .	32
1.7	Konfigurace hardware . . . . .	32
1.8	Přihlašování v grafice . . . . .	33
<b>2</b>	<b>Konfigurace pomocí YaST</b>	<b>35</b>
2.1	Spuštění YaST . . . . .	36
2.2	Řídící středisko YaST . . . . .	37
2.3	Software . . . . .	37
2.3.1	Změnit instalační zdroj . . . . .	37
2.3.2	Aktualizace programů on-line . . . . .	38
2.3.3	Aktualizace systému . . . . .	42
2.3.4	Aktualizace programů z CD . . . . .	42
2.3.5	Správce programů . . . . .	43
2.4	Hardware . . . . .	45
2.4.1	CD-ROM mechaniky . . . . .	45
2.4.2	Informace o hardwaru . . . . .	45
2.4.3	Nastavení IDE DMA . . . . .	46
2.4.4	Joystick . . . . .	46
2.4.5	Zvolte model myši . . . . .	47
2.4.6	Skener . . . . .	47
2.4.7	Zvuk . . . . .	49
2.4.8	TV karta . . . . .	50
2.5	Síťová zařízení . . . . .	51
2.6	Síťové služby . . . . .	51
2.6.1	Agent přenosu pošty (MTA) . . . . .	52

2.6.2	NFS server a klient . . . . .	54
2.6.3	NIS server a klient . . . . .	54
2.6.4	NTP klient . . . . .	55
2.6.5	Síťové služby (inetd) . . . . .	55
2.6.6	DNS a jméno počítače . . . . .	55
2.6.7	Směrování . . . . .	55
2.6.8	Nastavení Samba serevru a klienta . . . . .	55
2.7	Bezpečnost a uživatelé . . . . .	56
2.7.1	Správce uživatelů . . . . .	56
2.7.2	Správce skupin . . . . .	56
2.7.3	Nastavení bezpečnosti . . . . .	56
2.8	Systém . . . . .	57
2.8.1	Záloha systému . . . . .	57
2.8.2	Obnova systému . . . . .	57
2.8.3	Vytvořit systémovou disketu . . . . .	58
2.8.4	Výběr časové zóny . . . . .	60
2.8.5	Výběr jazyka . . . . .	60
2.8.6	Výběr rozložení klávesnice . . . . .	60
2.8.7	Editor úrovní běhu . . . . .	62
2.8.8	Editor souborů /etc/sysconfig . . . . .	62
2.8.9	Dělení disku . . . . .	63
2.8.10	Správce profilů . . . . .	67
2.8.11	Rozdělování disku . . . . .	68
2.8.12	Konfigurace zavaděče . . . . .	72
2.9	Různé . . . . .	74
2.9.1	Dotaz na podporu . . . . .	74
2.9.2	Zobrazit startovací protokol (log) . . . . .	75
2.9.3	Zobrazit systémový protokol (log) . . . . .	75
2.9.4	Načíst CD s ovladačem od výrobce . . . . .	75
2.10	YaST v textovém režimu (ncurses) . . . . .	75
2.10.1	Navigace v modulech . . . . .	77
2.10.2	Omezení klávesových zkratek . . . . .	77
2.10.3	Spouštění jednotlivých modulů . . . . .	78
2.10.4	YOU modul . . . . .	78
2.11	Online update z příkazové řádky . . . . .	79

<b>3</b>	<b>Zvláštní instalační postupy</b>	<b>81</b>
3.1	Nastavení centrálního instalačního serveru . . . . .	82
3.1.1	Konfigurace pomocí YaST . . . . .	82
3.1.2	Instalace klienta . . . . .	84
3.2	Program linuxrc . . . . .	85
3.3	Instalace pomocí VNC . . . . .	87
3.3.1	Příprava pro instalaci pomocí VNC . . . . .	87
3.3.2	Klientské programy pro instalaci pomocí VNC . . . . .	88
3.4	Textová instalace pomocí YaST . . . . .	88
3.5	Tipy a triky . . . . .	90
3.5.1	Vytváření startovací diskety v operačním systému DOS . . .	90
3.5.2	Vytváření startovací diskety v operačním systému typu UNIX . . . . .	91
3.5.3	Zavádění systému z diskety (SYSLINUX) . . . . .	92
3.5.4	Použití CD2 pro zavádění systému . . . . .	93
3.5.5	Podporované CD mechaniky . . . . .	93
3.5.6	Instalace ze síťového zdroje . . . . .	93
3.6	Přiřazování trvalých souborů zařízení SCSI zařízením . . . . .	94
3.7	Konfigurace LVM . . . . .	95
3.7.1	Správce logických svazků . . . . .	95
3.7.2	Konfigurace LVM pomocí nástroje YaST . . . . .	97
3.8	Konfigurace softwarového RAIDu . . . . .	100
3.8.1	Běžné typy polí RAID . . . . .	101
3.8.2	Konfigurace softwarového RAIDu pomocí YaST . . . . .	103
3.8.3	Řešení problémů . . . . .	103
3.8.4	Další informace . . . . .	104

<b>4</b>	<b>Aktualizace systému a správa balíčků</b>	<b>107</b>
4.1	Aktualizace systému SUSE LINUX . . . . .	108
4.1.1	Přípravy . . . . .	108
4.1.2	Možné problémy . . . . .	108
4.1.3	Aktualizace pomocí YaST . . . . .	109
4.2	Od verze k verzi . . . . .	109
4.2.1	Změny z 8.1 na 8.2 . . . . .	110
4.2.2	Změny z 8.2 na 9.0 . . . . .	111
4.2.3	Změny z 9.0 na 9.1 . . . . .	111
4.2.4	Změny z 9.1 na 9.2 . . . . .	114
4.2.5	Změny z 9.2 na 9.3 . . . . .	118
4.3	RPM the Package Manager . . . . .	120
4.3.1	Ověření balíku . . . . .	121
4.3.2	Správa balíků -- instalace, aktualizace a smazání . . . . .	121
4.3.3	RPM a opravy . . . . .	122
4.3.4	Delta RPM balíčky . . . . .	124
4.3.5	Zadání dotazu . . . . .	125
4.3.6	Instalace a překlad zdrojových balíků . . . . .	127
4.3.7	Další nástroje pro práci s archivy a databází RPM . . . . .	129
<b>5</b>	<b>Oprava systému</b>	<b>131</b>
5.1	Automatická oprava . . . . .	132
5.2	Vlastní nastavení . . . . .	134
5.3	Expertní nástroje . . . . .	134
5.4	Záchranný systém SUSE . . . . .	135
5.4.1	Spouštění záchranného systému . . . . .	135
5.4.2	Práce v záchranném systému . . . . .	136

<b>II</b>	<b>Systém</b>	<b>139</b>
<b>6</b>	<b>32- a 64-bitové aplikace v 64-bitovém prostředí</b>	<b>141</b>
6.1	Podpora běhu aplikací . . . . .	142
6.2	Vývoj softwaru . . . . .	143
6.3	Kompilace softwaru pro jinou platformu . . . . .	143
6.4	Specifikace jádra . . . . .	144
<b>7</b>	<b>Startování</b>	<b>147</b>
7.1	Startovací proces v Linuxu . . . . .	148
7.1.1	initrd . . . . .	149
7.1.2	linuxrc . . . . .	150
7.1.3	Informace i initrd . . . . .	151
7.2	Program init . . . . .	151
7.3	Úrovně běhu . . . . .	151
7.4	Změna úrovně běhu . . . . .	153
7.5	Init skripty . . . . .	154
7.5.1	Vkládání skriptů . . . . .	156
7.6	Editor úrovní běhu . . . . .	158
7.7	SuSEconfig a /etc/sysconfig . . . . .	159
7.8	YaST sysconfig Editor . . . . .	161
<b>8</b>	<b>Startování systému a zavaděče</b>	<b>163</b>
8.1	Startování . . . . .	164
8.1.1	Startování DOSu a Windows 9x . . . . .	165
8.2	Výběr zavaděče . . . . .	165
8.3	Startování systému se zavaděčem GRUB . . . . .	166
8.3.1	Startovací menu . . . . .	166
8.3.2	Soubor device.map . . . . .	173
8.3.3	Soubor /etc/grub.conf . . . . .	173
8.3.4	GRUB shell . . . . .	174

8.3.5	Nastavení hesla pro zavádění . . . . .	174
8.4	Konfigurace zavaděče pomocí programu YaST . . . . .	176
8.4.1	Obrazovka nastavení zavaděče . . . . .	176
8.4.2	Volby nastavení zavaděče . . . . .	178
8.5	Odinstalace zavaděče LILO nebo GRUB . . . . .	179
8.5.1	Obnova MBR (DOS, Win9x/ME, OS/2) . . . . .	180
8.5.2	Obnova MBR v Windows XP . . . . .	180
8.5.3	Obnova MBR v Windows 2000 . . . . .	180
8.5.4	Zavedení systému Linux po obnovení MBR . . . . .	181
8.6	Vytvoření startovacího CD . . . . .	181
8.7	Grafická konzole SUSE . . . . .	183
8.8	Řešení problémů . . . . .	183
8.9	Další informace . . . . .	185
<b>9</b>	<b>Linuxové jádro</b>	<b>187</b>
9.1	Update jádra . . . . .	188
9.2	Zdrojové texty jádra . . . . .	189
9.3	Konfigurace jádra . . . . .	189
9.3.1	Konfigurace z příkazové řádky . . . . .	190
9.3.2	Konfigurace v textovém módu . . . . .	190
9.3.3	Konfigurace pod X Window . . . . .	190
9.4	Moduly jádra . . . . .	190
9.4.1	Detekce hardwaru příkazem hwinfo . . . . .	191
9.4.2	Práce s moduly . . . . .	191
9.4.3	Soubor /etc/modprobe.conf . . . . .	192
9.4.4	Kmodzavaděč modulů jádra . . . . .	192
9.5	Nastavení konfigurace jádra . . . . .	193
9.6	Překlad jádra . . . . .	193
9.7	Instalace jádra . . . . .	194
9.8	Úklid po překladu jádra . . . . .	194

<b>10 Speciální vlastnosti SUSE LINUXu</b>	<b>197</b>
10.1 Nápověda k některým zvláštním balíčkům . . . . .	198
10.1.1 Balíček bash a /etc/profile . . . . .	198
10.1.2 Balíček cron . . . . .	198
10.1.3 Soubory logů: logrotate a balíčky . . . . .	199
10.1.4 Manuálové stránky . . . . .	201
10.1.5 Příkaz ulimit . . . . .	201
10.1.6 Příkaz free . . . . .	202
10.1.7 Soubor /etc/resolv.conf . . . . .	203
10.1.8 Nastavení programu GNU Emacs . . . . .	203
10.1.9 Krátký úvod do editoru vi . . . . .	204
10.2 Virtuální konzole . . . . .	207
10.3 Mapování klávesnice . . . . .	207
10.4 Lokální přizpůsobení I18N and L10N . . . . .	208
10.4.1 Některé příklady . . . . .	210
10.4.2 Nastavení jazykové podpory . . . . .	211
<b>11 Systém X Window</b>	<b>213</b>
11.1 Nastavení X11 pomocí SaX2 . . . . .	214
11.1.1 Plocha . . . . .	215
11.1.2 Grafická karta . . . . .	216
11.1.3 Barevná hloubka a rozlišení . . . . .	217
11.1.4 Virtuální rozlišení . . . . .	218
11.1.5 3D Akcelarace . . . . .	218
11.1.6 Geometrie . . . . .	219
11.1.7 Multihead . . . . .	219
11.1.8 Vstupní zařízení . . . . .	220
11.1.9 AccessX . . . . .	221
11.1.10 Joystick . . . . .	222
11.2 Optimalizace systému X Window . . . . .	223
11.2.1 Sekce Screen . . . . .	225



11.2.2	Sekce Device . . . . .	227
11.2.3	Sekce Monitor a Modes . . . . .	228
11.3	Instalace a konfigurace fontů . . . . .	228
11.3.1	Systémy písem . . . . .	229
11.4	Konfigurace OpenGL – 3D . . . . .	234
11.4.1	Podpora hardware . . . . .	234
11.4.2	Ovladače OpenGL . . . . .	234
11.4.3	Diagnostický nástroj 3Ddiag . . . . .	235
11.4.4	Testování OpenGL . . . . .	235
11.4.5	Řešení problémů . . . . .	235
11.4.6	Instalační podpora . . . . .	236
11.4.7	Dodatečná online dokumentace . . . . .	236
<b>12</b>	<b>Obsluha tisku</b>	<b>237</b>
12.1	Příprava . . . . .	238
12.2	Práce tiskového systému . . . . .	239
12.3	Způsoby a protokoly pro připojení tiskáren . . . . .	240
12.4	Instalace softwaru . . . . .	240
12.5	Konfigurace tiskárny . . . . .	241
12.5.1	Lokální tiskárny . . . . .	241
12.5.2	Síťové tiskárny . . . . .	243
12.5.3	Konfigurace . . . . .	245
12.6	Nastavení aplikací . . . . .	246
12.6.1	Tisk z příkazové řádky . . . . .	246
12.6.2	Tisk z aplikací pomocí příkazů příkazové řádky . . . . .	246
12.6.3	Použití tiskového systému CUPS . . . . .	247
12.7	Zvláštní vlastnosti v systému SUSE LINUX . . . . .	247
12.7.1	CUPS server a firewall . . . . .	247
12.7.2	Administrátor webového frontendu CUPS . . . . .	248
12.7.3	Změny v tiskové službě CUPS (cupsd) . . . . .	248
12.7.4	PPD soubory v různých balíčcích . . . . .	250

12.8	Řešení problémů . . . . .	252
12.8.1	Tiskárny bez podpory standardního tiskového jazyka . . . .	252
12.8.2	Pro postscriptovou tiskárnu není k dispozici vhodný PPD soubor . . . . .	253
12.8.3	Paralelní porty . . . . .	253
12.8.4	Připojení síťových tiskáren . . . . .	254
12.8.5	Problém s tiskem bez chybového hlášení . . . . .	256
12.8.6	Nepřístupné fronty . . . . .	256
12.8.7	Rušení tiskových úloh . . . . .	256
12.8.8	Vadné tiskové úlohy a chyby v přenosu dat . . . . .	257
12.8.9	Hledání problémů v tiskovém systému CUPS . . . . .	258
12.8.10	Další informace . . . . .	258
<b>13</b>	<b>Mobilita v Linuxu</b>	<b>259</b>
13.1	Notebooky . . . . .	260
13.1.1	Zvláštní hardwarové vlastnosti notebooků . . . . .	260
13.1.2	Snížení spotřeby energie . . . . .	260
13.1.3	Změny nastavení systému . . . . .	261
13.1.4	Software . . . . .	262
13.1.5	Ochrana dat . . . . .	265
13.2	Mobilní hardware . . . . .	265
13.3	Mobilní telefony a kapesní počítače . . . . .	266
13.4	Další informace . . . . .	267
<b>14</b>	<b>Linux a notebooky</b>	<b>269</b>
14.1	Hardware . . . . .	270
14.2	Software . . . . .	270
14.2.1	Cardmanager . . . . .	270
14.3	Konfigurace . . . . .	271
14.3.1	Ethernet, bezdrát (wireless) a Token Ring . . . . .	271
14.3.2	ISDN . . . . .	271

14.3.3	Modem . . . . .	272
14.3.4	SCSI a IDE . . . . .	272
14.4	Problémové notebooky . . . . .	272
14.4.1	Základní systém PCMCIA nefunguje . . . . .	273
14.4.2	Karta PCMCIA nefunguje správně . . . . .	274
14.5	Další informace . . . . .	275
<b>15</b>	<b>Správa profilů . . . . .</b>	<b>277</b>
15.1	Základní terminologie . . . . .	278
15.2	Nastavení SCPM . . . . .	279
15.2.1	Spuštění SCPM a definice skupin zdrojů . . . . .	279
15.2.2	Vytváření a přepínání profilů . . . . .	280
15.2.3	Přepínání mezi profily . . . . .	281
15.2.4	Rozšířené nastavení . . . . .	281
15.3	Volba profilu při startu . . . . .	283
15.4	Problémy a jejich řešení . . . . .	283
15.4.1	Změna nastavení skupiny zdrojů . . . . .	284
15.5	Další informace . . . . .	284
<b>16</b>	<b>Správa napájení . . . . .</b>	<b>285</b>
16.1	Funkce šetření spotřeby . . . . .	286
16.2	APM . . . . .	287
16.3	ACPI . . . . .	288
16.3.1	ACPI v praxi . . . . .	289
16.3.2	Nastavení výkonu CPU . . . . .	291
16.3.3	Nástroje ACPI . . . . .	292
16.3.4	Možné problémy . . . . .	292
16.4	Zastavení disku . . . . .	294
16.5	Balík powersave . . . . .	295
16.5.1	Konfigurace powersave . . . . .	296
16.5.2	Konfigurace APM a ACPI . . . . .	298
16.5.3	Možné problémy . . . . .	301
16.6	Modul správy napájení programu YaST . . . . .	304

<b>17</b>	<b>Bezdrátová komunikace</b>	<b>309</b>
17.1	Bezdrátové sítě . . . . .	310
17.1.1	Hardware . . . . .	310
17.1.2	Funkce . . . . .	311
17.1.3	Nastavení pomocí programu YaST . . . . .	313
17.1.4	Dostupné programy . . . . .	315
17.1.5	Tipy a triky nastavení WLAN . . . . .	315
17.1.6	Možné problémy . . . . .	316
17.1.7	Další informace . . . . .	317
17.2	Bluetooth . . . . .	317
17.2.1	Základy . . . . .	318
17.2.2	Nastavení . . . . .	319
17.2.3	Systémové komponenty a programy pro práci s Bluetooth . . . . .	322
17.2.4	Grafické aplikace . . . . .	323
17.2.5	Příklady . . . . .	324
17.2.6	Řešení možných problémů . . . . .	325
17.2.7	Další informace . . . . .	327
17.3	IrDA Infrared Data Association . . . . .	327
17.3.1	Software . . . . .	327
17.3.2	Konfigurace . . . . .	328
17.3.3	Použití . . . . .	328
17.3.4	Možné potíže . . . . .	328
<b>18</b>	<b>Hotplug systém</b>	<b>331</b>
18.1	Zařízení a rozhraní . . . . .	332
18.2	Hotplug události . . . . .	333
18.3	Hotplug agenti . . . . .	334
18.3.1	Aktivace síťových rozhraní . . . . .	334
18.3.2	Aktivace zařízení pro ukládání dat . . . . .	335
18.4	Automatické nahrávání modulů . . . . .	335
18.5	Hotplug PCI zařízení . . . . .	336

18.6	Startovací skripty coldplug a hotplug . . . . .	337
18.7	Analýza chyb . . . . .	337
18.7.1	Log soubory . . . . .	337
18.7.2	Problémy při startu systému . . . . .	337
18.7.3	Zapisovač událostí . . . . .	338
<b>19</b>	<b>Dynamické uzly zařízení pomocí udev</b>	<b>339</b>
19.1	Tvorba pravidel . . . . .	340
19.2	Automatizace pomocí NAME a SYMLINK . . . . .	341
19.3	Regulární výrazy v klíších . . . . .	341
19.4	Výběr klíčů . . . . .	342
19.5	Konzistentní pojmenování zařízení pro hromadné uchovávání dat .	343
<b>20</b>	<b>Souborové systémy</b>	<b>345</b>
20.1	Termíny . . . . .	346
20.2	Hlavní souborové systémy Linuxu . . . . .	346
20.2.1	Ext2 . . . . .	347
20.2.2	Ext3 . . . . .	347
20.2.3	ReiserFS . . . . .	349
20.2.4	Reiser4 . . . . .	349
20.2.5	JFS . . . . .	350
20.2.6	XFS . . . . .	351
20.3	Některé další podporované souborové systémy . . . . .	352
20.4	Podpora souborů větších než 2 GB . . . . .	353
20.5	Další informace . . . . .	354
<b>21</b>	<b>Autentizace pomocí PAM</b>	<b>357</b>
21.1	Struktura PAM konfiguračního souboru . . . . .	358
21.2	Konfigurace PAM pro sshd . . . . .	360
21.3	Konfigurace PAM modulů . . . . .	362
21.3.1	pam_unix2.conf . . . . .	362
21.3.2	pam_env.conf . . . . .	363
21.3.3	pam_pwcheck.conf . . . . .	363
21.3.4	limits.conf . . . . .	364
21.4	Další informace . . . . .	364

<b>III Služby</b>	<b>365</b>
<b>22 Základy síťování</b>	<b>367</b>
22.1 IP adresy a směrování . . . . .	370
22.1.1 IP adresa . . . . .	370
22.1.2 Síťové masky a směrování . . . . .	371
22.2 IPv6 – Internet další generace . . . . .	373
22.2.1 Přednosti IPv6 . . . . .	374
22.2.2 Adresování v IPv6 . . . . .	375
22.2.3 IPv4 versus IPv6 – cestování mezi světy . . . . .	379
22.2.4 Konfigurace IPv6 . . . . .	380
22.2.5 Další informace . . . . .	380
22.3 Překlad jmen . . . . .	381
22.4 Konfigurace síťového připojení pomocí YaST . . . . .	382
22.4.1 Konfigurace síťové karty pomocí YaST . . . . .	382
22.4.2 Modem . . . . .	385
22.4.3 ISDN . . . . .	387
22.4.4 Kabelový modem . . . . .	389
22.4.5 DSL . . . . .	390
22.5 Manuální konfigurace sítě . . . . .	392
22.5.1 Konfigurační soubory . . . . .	395
22.5.2 Startovací skripty . . . . .	401
22.6 smpppd jako pomocník s vytáčeným připojením . . . . .	402
22.6.1 Konfigurace smpppd . . . . .	403
22.6.2 Programy kinternet, qinternet a cinternet a vzdálené použití	403
<b>23 SLP služby v síti</b>	<b>405</b>
23.1 Registrace vlastních služeb . . . . .	405
23.2 SLP frontendy v systému SUSE LINUX . . . . .	406
23.3 Aktivace SLP . . . . .	407
23.4 Další informace . . . . .	407

<b>24 DNS Domain Name System</b>	<b>409</b>
24.1 Konfigurace pomocí YaST	409
24.1.1 Průvodce konfigurací	409
24.1.2 Expertní nastavení	410
24.2 Spuštění nameserveru BIND	413
24.3 Konfigurační soubor /etc/named.conf	416
24.4 Nejdůležitější konfigurační volby v sekci options	418
24.5 Konfigurace v sekci logging	420
24.6 Struktura souboru odkazujícího na data pro zóny	420
24.7 Struktura souboru s daty pro zónu	421
24.8 Dynamická aktualizace údajů o zóně	425
24.9 Bezpečné transakce	425
24.10 DNSSEC	427
24.11 Další informace	427
<b>25 NIS Network Information Service</b>	<b>429</b>
25.1 Konfigurace NIS serveru	429
25.2 Konfigurace NIS klientů	432
<b>26 NFS sdílené souborové systémy</b>	<b>435</b>
26.1 Importování souborových systémů pomocí YaST2	436
26.2 Ruční import souborových systémů	436
26.3 Exportování souborových systémů pomocí YaST	437
26.4 Ruční export souborových systémů	438
<b>27 DHCP</b>	<b>441</b>
27.1 DHCP protokol	441
27.2 Konfigurace DHCP serveru pomocí nástroje YaST	442
27.3 DHCP softwarové balíčky	443
27.4 DHCP server dhcpd	444
27.4.1 Počítač s pevnou IP adresou	446
27.4.2 Zvláštnosti v systému SUSE LINUX	447
27.5 Další informace	448

<b>28 Synchronizace času pomocí ntp</b>	<b>449</b>
28.1 Nastavení ntp v síti . . . . .	450
28.2 Nastavení lokálních referenčních hodin . . . . .	451
28.3 Nastavení NTP klienta v programu YaST . . . . .	451
28.3.1 Rychlé nastavení NTP klienta . . . . .	451
28.3.2 Komplexní nastavení NTP klienta . . . . .	452
<b>29 LDAP adresářové služby</b>	<b>455</b>
29.1 LDAP versus NIS . . . . .	457
29.2 Struktura adresářového stromu LDAP . . . . .	458
29.3 Konfigurace LDAP serveru pomocí slapd.conf . . . . .	460
29.3.1 Globální nastavení v slapd.conf . . . . .	460
29.3.2 Nastavení specifická pro databázi v souboru slapd.conf . . . . .	464
29.3.3 Spuštění a zastavení serveru . . . . .	464
29.4 Správa dat v LDAP adresáři . . . . .	465
29.4.1 Vkládání dat do LDAP adresáře . . . . .	465
29.4.2 Úprava dat v LDAP adresáři . . . . .	467
29.4.3 Vyhledávání a čtení dat z LDAP adresáře . . . . .	468
29.4.4 Mazání dat z LDAP adresáře . . . . .	468
29.5 YaST LDAP klient . . . . .	468
29.5.1 Standardní procedura . . . . .	469
29.5.2 Konfigurace LDAP klienta . . . . .	470
29.5.3 Uživatelé a skupiny Konfigurace pomocí YaST . . . . .	474
29.6 Další informace . . . . .	474
<b>30 Webový server Apache</b>	<b>477</b>
30.1 Základy . . . . .	478
30.1.1 Webový server . . . . .	478
30.1.2 HTTP . . . . .	478
30.1.3 URL . . . . .	478
30.1.4 Automatický výstup výchozí stránky . . . . .	479



30.2	Nastavení HTTP serveru pomocí YaST . . . . .	479
30.3	Moduly Apache . . . . .	479
30.4	Vlákna (threads) . . . . .	480
30.5	Instalace . . . . .	481
30.5.1	Výběr balíků v programu YaST . . . . .	481
30.5.2	Aktivace Apache . . . . .	481
30.5.3	Moduly pro aktivní obsah . . . . .	481
30.5.4	Další doporučené balíky . . . . .	482
30.5.5	Instalace modulů pomocí apxs . . . . .	482
30.6	Nastavení . . . . .	482
30.6.1	Konfigurace pomocí skriptu SuSEconfig . . . . .	483
30.6.2	Ruční nastavení . . . . .	484
30.7	Používání Apache . . . . .	487
30.8	Aktivní obsah . . . . .	488
30.8.1	SSI . . . . .	488
30.8.2	CGI . . . . .	489
30.8.3	GET a POST . . . . .	489
30.8.4	Generování aktivního obsahu pomocí modulů . . . . .	489
30.8.5	mod_perl . . . . .	490
30.8.6	mod_php4 . . . . .	492
30.8.7	mod_python . . . . .	492
30.8.8	mod_ruby . . . . .	492
30.9	Virtuální servery . . . . .	492
30.9.1	Virtuální server založený na jménu . . . . .	493
30.9.2	Virtuální server založený na IP . . . . .	494
30.9.3	Vícenásobné instance Apache . . . . .	495
30.10	Bezpečnost . . . . .	495
30.10.1	Minimalizace rizika . . . . .	495
30.10.2	Přístupová práva . . . . .	496
30.10.3	Aktualizace . . . . .	496

30.11	Možné problémy . . . . .	497
30.12	Další dokumentace . . . . .	497
30.12.1	Apache . . . . .	497
30.12.2	CGI . . . . .	497
30.12.3	Bezpečnost . . . . .	498
30.12.4	Další zdroje . . . . .	498
<b>31</b>	<b>Synchronizace souborů</b>	<b>499</b>
31.1	Programy pro datovou synchronizaci . . . . .	500
31.1.1	Unison . . . . .	500
31.1.2	CVS . . . . .	500
31.1.3	subversion . . . . .	501
31.1.4	mailsync . . . . .	501
31.1.5	rsync . . . . .	501
31.2	Výběr vhodného programu . . . . .	502
31.2.1	Klient-Server vs. Peer-to-Peer . . . . .	502
31.2.2	Přenositelnost . . . . .	502
31.2.3	Interaktivní vs. automatický . . . . .	502
31.2.4	Konflikty: výskyt a řešení . . . . .	503
31.2.5	Výběr a vkládání souborů . . . . .	503
31.2.6	Historie . . . . .	503
31.2.7	Objem dat a požadavky na diskový prostor . . . . .	503
31.2.8	GUI . . . . .	504
31.2.9	Uživatelská přívětivost . . . . .	504
31.2.10	Bezpečnost . . . . .	504
31.2.11	Ochrana proti ztrátě dat . . . . .	504
31.3	Úvod do Unison . . . . .	505
31.3.1	Požadavky . . . . .	505
31.3.2	Používání Unison . . . . .	506
31.3.3	Další informace . . . . .	507
31.4	Úvod do programu CVS . . . . .	507

31.4.1	Konfigurace CVS serveru . . . . .	507
31.4.2	Používání CVS . . . . .	508
31.4.3	Další informace . . . . .	509
31.5	Úvod do Subversion . . . . .	510
31.5.1	Instalace Subversion serveru . . . . .	510
31.5.2	Použití a provoz . . . . .	511
31.5.3	Další informace . . . . .	512
31.6	Úvod do rsync . . . . .	513
31.6.1	Konfigurace a provoz . . . . .	513
31.6.2	Další informace . . . . .	514
31.7	Úvod do mailsync . . . . .	514
31.7.1	Konfigurace a použití . . . . .	515
31.7.2	Možné problémy . . . . .	517
31.7.3	Další informace . . . . .	517
<b>32</b>	<b>Samba</b>	<b>519</b>
32.1	Nastavení serveru . . . . .	520
32.1.1	Sekce [global] . . . . .	521
32.1.2	Sdílení . . . . .	522
32.1.3	Bezpečnostní úrovně . . . . .	523
32.2	Samba jako přihlašovací server . . . . .	524
32.3	Konfigurace Samba serveru pomocí programu YaST . . . . .	526
32.4	Nastavení klienta . . . . .	527
32.4.1	Nastavení Samba klienta pomocí YaST . . . . .	527
32.4.2	Windows 9x a ME . . . . .	527
32.5	Optimalizace . . . . .	528

<b>33 Proxy server Squid</b>	<b>531</b>
33.1 Informace o proxy-cache . . . . .	531
33.1.1 Squid a bezpečnost . . . . .	532
33.1.2 Vícenásobná cache . . . . .	532
33.1.3 Přechovávání objektů z Internetu . . . . .	533
33.2 Systémové požadavky . . . . .	533
33.2.1 Pevný disk . . . . .	533
33.2.2 Velikost diskové cache . . . . .	533
33.2.3 RAM . . . . .	534
33.2.4 CPU . . . . .	534
33.3 Spuštění squidů . . . . .	534
33.3.1 Příkazy pro spuštění squidů . . . . .	535
33.3.2 Lokální DNS server . . . . .	536
33.4 Konfigurační soubor /etc/squid/squid.conf . . . . .	537
33.4.1 Základní nastavení . . . . .	537
33.4.2 Volby pro kontrolu přístupu . . . . .	539
33.5 Konfigurace transparentní proxy . . . . .	542
33.5.1 Konfigurace jádra . . . . .	542
33.5.2 Možnosti konfigurace v /etc/squid/squid.conf . . . . .	542
33.5.3 Konfigurace firewallu pomocí SuSEfirewallu2 . . . . .	543
33.6 cachemgr.cgi . . . . .	544
33.6.1 Nastavení . . . . .	545
33.6.2 ACL cache manageru v /etc/squid/squid.conf . . . . .	545
33.6.3 Prohlížení statistik . . . . .	546
33.7 squidGuard . . . . .	546
33.8 Vytvoření protokolů programem Calamaris . . . . .	548
33.9 Další informace o Squidu . . . . .	548

## IV Správa

549

<b>34 Bezpečnost v Linuxu</b>	<b>551</b>
34.1 Firewall a maškaráda . . . . .	552
34.1.1 Filtrování paketů pomocí iptables . . . . .	552
34.1.2 Základy maškarády . . . . .	553
34.1.3 Základy firewallu . . . . .	554
34.1.4 SuSEfirewall2 . . . . .	555
34.1.5 Další informace . . . . .	559
34.2 SSH: bezpečná práce v síti . . . . .	559
34.2.1 Balíček OpenSSH . . . . .	560
34.2.2 Program ssh . . . . .	560
34.2.3 Bezpečné kopírování pomocí scp . . . . .	560
34.2.4 Bezpečný přenos souborů pomocí sftp . . . . .	561
34.2.5 SSH démon (sshd) – strana serveru . . . . .	561
34.2.6 Mechanismus ověřování pomocí SSH . . . . .	562
34.2.7 X server, ověřování a přeposílací mechanismy . . . . .	563
34.3 Šifrování diskových oddílů a souborů . . . . .	564
34.3.1 Vhodné nasazení . . . . .	564
34.3.2 Nastavení šifrovaného souborového systému pomocí YaST . . . . .	565
34.3.3 Šifrování obsahu vyměnitelného média . . . . .	566
34.4 Bezpečnost a soukromí . . . . .	567
34.4.1 Lokální a síťová bezpečnost . . . . .	567
34.4.2 Bezpečnostní tipy a triky . . . . .	575
34.4.3 Ústřední adresa pro hlášení bezpečnostních problémů . . . . .	577

<b>35 ACLs v Linuxu</b>	<b>581</b>
35.1 Výhody ACLs . . . . .	582
35.2 Definice . . . . .	582
35.3 Používání ACLs . . . . .	583
35.3.1 ACL položky a přístupové bity . . . . .	584
35.3.2 Adresář s ACL přístupem . . . . .	585
35.3.3 Adresář s výchozími ACL . . . . .	588
35.3.4 ACL kontrolní algoritmus . . . . .	591
35.4 Výhledy . . . . .	592
35.5 Další informace . . . . .	592
 <b>36 Nástroje monitorování systému</b>	 <b>593</b>
36.1 Seznam otevřených souborů: lsof . . . . .	595
36.2 Přístup uživatelů k souborům: fuser . . . . .	596
36.3 Vlastnosti souboru: stat . . . . .	597
36.4 USB zařízení: lsusb . . . . .	597
36.5 SCSI zařízení: scsiinfo . . . . .	598
36.6 Procesy: top . . . . .	599
36.7 Seznam procesů: ps . . . . .	599
36.8 Strom procesů: pstree . . . . .	602
36.9 Kdo co dělá: w . . . . .	603
36.10 Využití paměti: free . . . . .	603
36.11 Systémové hlášení jádra: dmesg . . . . .	604
36.12 Souborový systém a jeho využití: mount, df a du . . . . .	604
36.13 Souborový systém /proc . . . . .	605
36.14 vmstat, iostat a mpstat . . . . .	607
36.15 procinfo . . . . .	608
36.16 PCI zdroje: lspci . . . . .	609
36.17 Systémová volání běžícího programu: strace . . . . .	609
36.18 Volání knihoven běžícím příkazem: ltrace . . . . .	611
36.19 Zjištění vyžadovaných knihoven: ldd . . . . .	611
36.20 Dodatečné informace o ELF binárních souborech . . . . .	612
36.21 Meziprocsová komunikace: ipc . . . . .	612
36.22 Měření času: time . . . . .	613

<b>V Přílohy</b>	<b>615</b>
<b>A Dokumentace a zdroje informací</b>	<b>617</b>
<b>B SUSE LINUX FAQ</b>	<b>621</b>
<b>C Kontrola souborového systému</b>	<b>631</b>
<b>D GNU licence</b>	<b>647</b>
<b>Slovník pojmů</b>	<b>655</b>





**Část I**

**Instalace**



# Instalace pomocí programu YaST

Předcházející odstavec pokrýval rychlý instalační postup. Tato kapitola vám dá podrobnější informace o nastavení která můžete změnit použitím odpovídajících modulů z hlavního návrhu. Instalace je tak plně pod vaší kontrolou.

1.1	Spouštění instalačního programu . . . . .	4
1.2	Úvodní obrazovka . . . . .	6
1.3	Výběr jazyka . . . . .	8
1.4	Typ instalace . . . . .	8
1.5	Návrh instalace . . . . .	9
1.6	Dokončení instalace . . . . .	24
1.7	Konfigurace hardware . . . . .	32
1.8	Přihlašování v grafice . . . . .	33

## 1.1 Spouštění instalačního programu

Vlože první CD nebo DVD produktu SUSE LINUX do mechaniky. Potom restartujte počítač a spusťte instalační program z vloženého média.

### 1.1.1 Možnosti spuštění instalace

V případě problémů při spouštění instalace z CD nebo DVD můžete využít i jiný způsob spuštění instalace. Možnosti jsou popsány v tabulce 1.1 na této straně.

*Tabulka 1.1: Možnosti spuštění instalace*

Možnost	Popis
CD	Nejsnadnější způsob instalace. Tuto možnost lze využít, pokud má počítač lokální CD mechaniku podporovanou Linuxem.
Disketa	Obrazy pro vytvoření startovací diskety najdete na CD1 v adresáři <code>/boot/</code> . Ve stejném adresáři je také soubor <code>README</code> s postupem vytvoření.
PXE nebo BO- OTP	Tento způsob musí být podporován BIOSem vašeho počítače a případně firmwarem síťové karty. Na síti musí být instalační server. Úlohu instalačního serveru může převzít např. jiný počítač se systémem SUSE LINUX.
Pevný disk	SUSE LINUX může být nainstalován také z pevného disku. Překopírujte jádro ( <code>linux</code> ) a instalační systém ( <code>initrd</code> ) z adresáře <code>/boot/loader</code> z CD 1 na pevný disk a zavaděči zadejte příslušné údaje.

### 1.1.2 Možné komplikace při startu z CD nebo DVD

Problémy, na které narazíte při zavádění systému z CD nebo DVD, mohou mít mnoho příčin. Je možné, že CD-ROM mechanika není schopna načíst zaváděcí obraz disku za prvním CD. V takovém případě použijte CD2 k zavedení systému. Toto CD obsahuje standardní bootovací obraz 2.88MB diskety, který by měl

být načten i staršími mechanikami.

Další možnou příčinnou může být chybné nastavení sekvence pro zavádění systému v BIOSu (basic input output system). Informace o změně nastavení BIOS by měly být v dokumentaci k základní desce počítače, v obecné formě i v následujícím textu.

BIOS je softwarové vybavení, které zabezpečuje základní funkce počítače. Výrobci základních desek poskytují BIOS specifický pro jejich hardware.

Většinou lze do BIOSu vstoupit v určité fázi spouštění počítače. V průběhu inicializace provádí počítač množství diagnostických hardwarových testů. Jedním z nich je test paměti, indikovaný počítadlem. Ve chvíli kdy se objeví počítadlo, hledejte řádek, obvykle pod počítadlem paměti nebo ve spodní části obrazovky, vyzývající vás ke stisku klávesy pro vstup do BIOSu. V mnoha případech je touto klávesou (Del), (F1), (F2), nebo (Esc). Držte tuto klávesu dokud se neobjeví úvodní stránka BIOSu.

## Důležité

### Rozložení kláves v BIOSu

BIOS je obvykle limitován americkým rozložením klávesnice.

## Důležité

Pro změnu sekvence pro zavádění systému v AWARD BIOSu hledejte položku menu 'BIOS FEATURES SETUP'. Jiní výrobci mohou používat odlišná jména, například 'ADVANCED CMOS SETUP'. Poté, co tuto položku najdete, vyberte ji a potvrďte stiskem (Enter).

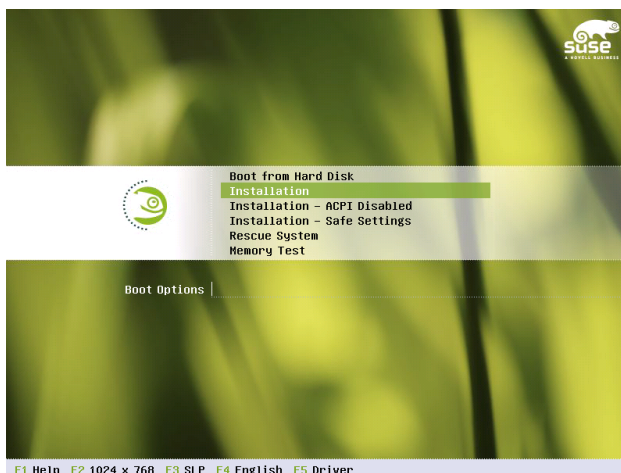
V obrazovce, která se otevře, hledejte menu s názvem 'BOOT SEQUENCE'. Sekvence je často nastavena na něco podobného jako C, A nebo A, C. V prvním případě se počítač nejprve pokusí použít harddisk (C) a poté disketovou jednotku (A) k zavedení systému. Měňte nastavení stiskem (Page up) nebo (Page down) dokud není zobrazená sekvence ve tvaru A, CDROM, C.

Opusťte BIOS stiskem (Esc). K uložení změn vyberte 'SAVE & EXIT SETUP' nebo stiskněte (F10). Pro potvrzení uložení stiskněte (Y).

Pokud máte SCSI CD-ROM mechaniku, změňte nastavení SCSI BIOSu. V případě adaptéru Adaptec otevřete nastavení stiskem (Ctrl)-(A). Poté vyberte 'Disk Utilities', kde se vám zobrazí připojené hardwarové komponenty. Poznamenejte si SCSI ID vaší CD-ROM mechaniky. Ukončete menu stiskem (Esc) a otevřete 'Configure Adapter Settings'. Pod 'Additional Options' vyberte 'Boot Device Options' a stiskněte (Enter). Zadejte SCSI ID vaší CD-ROM mechaniky který jste si poznamenali dříve a stiskněte znovu (Enter). Poté se dvakrát stiskem (Esc) vraťte do úvodní obrazovky SCSI BIOSu. Ukončete ho a potvrďte výběrem 'Yes' restart počítače.

## 1.2 Úvodní obrazovka

Úvodní obrazovka obsahuje několik položek menu, ze kterých můžete vybírat. 'Boot from Hard Disk' zavede systém už instalovaný na počítači (pokud již byla instalace provedena). Tato položka je vybrána jako výchozí, pro případ média zapomenutého v mechanice. Pro instalaci zvolte položku 'Installation' pomocí kurzorových kláves. Spustí se YaST a začne instalace.



Obrázek 1.1: Úvodní obrazovka

Položky menu úvodní obrazovky poskytují různé možnosti zavádění systému z CD-ROM - dá se vybírat z následujících voleb:

**Boot from Hard Disk** Zavede systém už instalovaný v počítači (který je normálně spuštěn při startu z pevného disku). Tato položka je vybrána jako výchozí.

**Installation** Standardní způsob instalace. Budou zapnuty všechny funkce moderního hardware.

**Installation ACPI Disabled** Selhání standardní instalace může být způsobeno vadnou podporou ACPI (Advanced Configuration and Power Interface).

V takovém případě použijte tuto volbu a proveďte instalaci bez podpory ACPI.

**Installation Safe Settings** Nastartuje počítač s vypnutým DMA (pro CD-ROM mechaniky) a s vypnutými subsystemy pro řízení spotřeby. Zkušební uživatelé a správci mohou také přidávat vlastní parametry do startovací řádky jádra.

**Rescue System** Pokud nemůžete z nějakého důvodu nastartovat vámi nainstalovaný Linux, můžete zavést systém z DVD nebo CD1 a vybrat tuto položku. Bude spuštěn *záchranný systém* minimalizovaná podoba Linuxu bez grafického rozhraní, která umožní správcům přistupovat k oddílům disku pro opravy a odstraňování chyb v instalovaném systému. Méně zkušení uživatelé mohou použít nástroje na opravu systému obsažené v programu YaST. Pro více informací hledejte v kapitole 5 na straně 131.

**Memory Test** Test paměti spočívá v opakovaných cyklech zápisu a čtení do paměti. Je prováděn v nekonečné smyčce, protože poškození paměti se většinou projevuje nahodile a pro jeho odhalení může být třeba mnoha nezávislých pokusů. Pokud máte podezření, že vaše RAM by mohla být poškozená, použijte tuto volbu a nechte test probíhat po dobu minimálně několika hodin. Pokud nebudou zjištěny žádné chyby ani po případně delší době, dá se předpokládat, že je paměť v pořádku. Test můžete ukončit restartem počítače.

Použijte funkční klávesy jak je popsáno v pruhu na spodní straně obrazovky ke změně dalších potřebných nastavení instalace.

- F1** Otevírá kontextovou nápovědu popis právě aktivní části úvodní obrazovky.
- F2** Vybírá různé grafické módy zobrazení pro instalaci. Mimo jiné obsahuje i volbu pro textový mód, který se používá zejména v případech kde grafická instalace způsobuje z nějakých důvodů problémy.
- F3** Pomůže vám vybrat mezi různými instalačními médii. Většinou je instalace prováděna z vložených instalačních disků, ale v některých případech je nutné použít jiný instalační zdroj, jako je FTP server nebo NFS adresář. SLP (service location protocol) umožňuje připojení k SLP serveru v lokální síti, který vrací informace o různých instalačních médiích, která jsou na serveru přístupná. .
- F4** Výběr jazyka pro instalaci.

- F5** Ve výchozím nastavení nejsou diagnostická hlášení linuxového jádra při startu systému zobrazována, je vidět jen souhrnný indikátor. Pro zobrazení těchto hlášení vyberte volbu 'Native'. Pro zobrazení všech dostupných informací při startu systému pak volbu 'Verbose'.
- F6** Pomocí této volby můžete specifikovat dodatečný disk s updaty ovladačů pro SUSE LINUX. Budete požádáni o jeho vložení v průběhu instalačního procesu.

Několik sekund pro startu instalace SUSE LINUX nahraje minimalizovaný linuxový systém nutný pro spuštění instalace. Objeví se řada hlášení, na jejichž konci se spustí instalační program YaST. Po několika dalších vteřinách by se měla objevit obrazovka grafického rozhraní instalace, která vás provede instalací.

Na tomto místě začíná vlastně instalace začíná a její průběh je řízen programem YaST. Všechny ovládací obrazovky YaST mají podobné rozvržení. Všechna tlačítka, vstupní pole a seznamy mohou být ovládány myší. Pokud se ukazatel myši nehýbe, nepodařilo se myš automaticky nastavit. V takovém případě použijte pro pohyb mezi ovládacími prvky klávesnici. Navigace pomocí klávesnice je popsána v části 2.10.1 na straně 77.

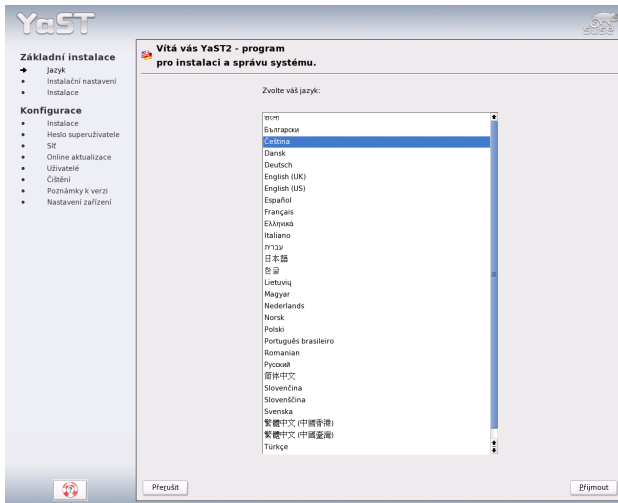
## 1.3 Výběr jazyka

Jak YaST, tak SUSE LINUX obecně mohou být nastaveny pro používání jazyka podle vašich potřeb. Jazyk zvolený v této fázi je pak použit jako výchozí pro rozložení klávesnice. Kromě toho používá YaST jazyková nastavení k vyplnění údajů o časovém pásmu a nastavení hodin v počítači. Pokud nemůžete použít myš, pohybujte se kurzorovými šipkami dokud nebude zvolen vámi požadovaný jazyk. Poté několikrát stiskněte **(Tab)** dokud nebude zvýrazněno tlačítko 'Další'. Stiskem klávesy **(Enter)** potvrdíte váš výběr jazyka.

## 1.4 Typ instalace

V tomto výběru můžete volit mezi položkami 'Nová instalace' a 'Aktualizace stávajícího systému'. Tato volba je samozřejmě použitelná jen pro předchozí instalace systémů SUSE LINUX. Dříve nainstalovaný systém také můžete spustit s pomocí volby 'Spustit nainstalovaný systém'. Pokud se systém nenastartuje z





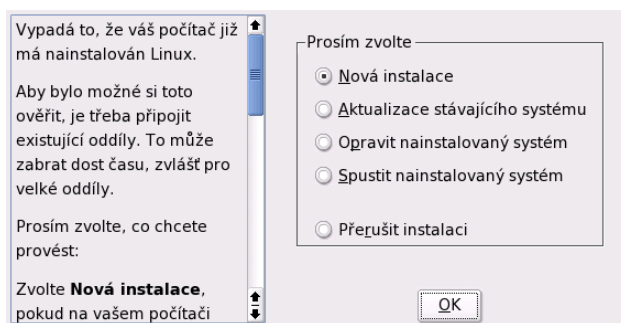
*Obrázek 1.2: Volba požadovaného jazyka*

důvodu poškození důležitých částí konfigurace, můžete se jej pokusit opravit pomocí volby ‘Opravit nainstalovaný systém’. Pokud na počítači nebyl dříve instalován SUSE LINUX, je jediná možná varianta provést instalaci novou. Pro pokračování klikněte na ‘OK’, viz obr. 1.3 na následující straně.

Následující text popisuje postup instalace nového systému. Detailní instrukce pro provádění aktualizace jsou uvedeny v části 2.3.3 na straně 42. Popis opravy systému můžete najít v kapitole 5 na straně 131.

## 1.5 Návrh instalace

Po úspěšné detekci hardware počítač se zobrazí návrh nastavení instalace (k náhledu na obr. 1.4 na straně 11), který obsahuje nějaké informace o hardware a nabízí množství instalačních a konfiguračních voleb. Po výběru některé z položek a její další konfiguraci v příslušných dialogových oknech se vždy navrátíte do okna návrhu nastavení instalace, které bude reflektovat vámi provedené změny. Jednotlivá nastavení jsou popsána v následujícím textu.



Obrázek 1.3: Výběr typu instalace

### 1.5.1 Režim instalace

V této části můžete změnit režim instalace, který jste nastavili v předchozím dialogu. Možnosti nastavení jsou popsány v sekci 1.4 na straně 8.

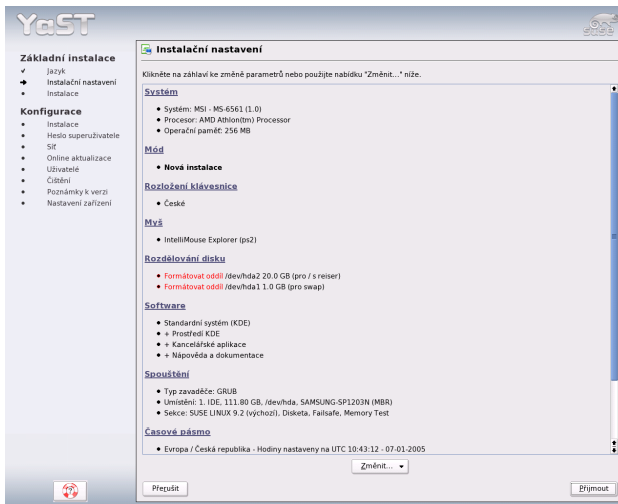
### 1.5.2 Rozložení klávesnice

Vyberte typ rozložení klávesnice. Výchozí nastavení koresponduje s nastavením jazyka. Po změně rozložení otestujte pozici písmen Y,Z a dalších speciálních znaků abyste se ujistili, že výběr byl správný. Až skončíte, použijte tlačítko 'Další' k návratu do okna návrhu nastavení.

### 1.5.3 Myš

Pokud YaST není schopen detekovat vaši myš automaticky, stiskněte několikrát klávesu (Tab) v okně návrhu dokud nebude vybrána položka 'Myš'. Potom použijte klávesu (Space) a otevřete tak okno s nabídkou typů myši. Výběrový dialog je ukázán na obr. 1.5 na straně 12.

Použijte klávesy ( ) a ( ) pro výběr typu myši. Pro více informací o ovladači a dalších podrobnostech nahlédněte do dokumentace zařízení. Poté, co vyberete typ myši, použijte (Alt)-(T) pro otestování zařízení na správnou funkčnost bez toho, aby byl výběr trvalý. Pokud se myš nechová jak jste očekávali, vyberte pomocí klávesnice jiný typ a otestujte jej. Klávesami (Tab) a (Enter) potvrdíte nakonec definitivní výběr, který už bude mít trvalou platnost.



Obrázek 1.4: Okno návrhu

## 1.5.4 Rozdělování disku

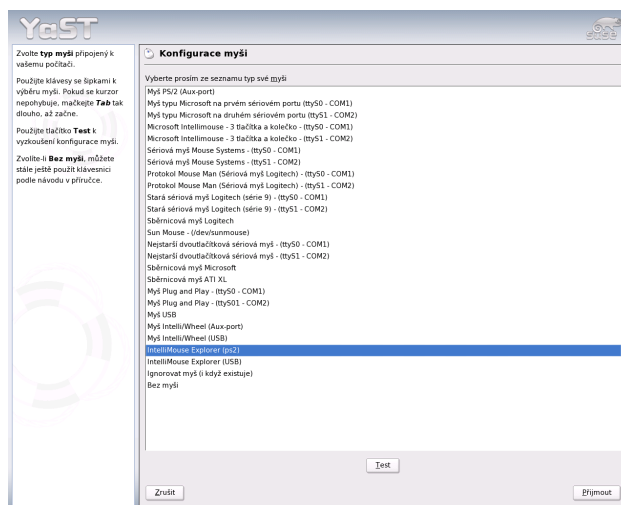
Ve většině případů vám YaST nabídne vyhovující schéma rozdělení disků, které můžete přijmout bez dalších změn. Můžete také použít YaST na přizpůsobení navrženého rozdělení. Následující text popisuje nutné kroky.

### Typy oddílů

Každý disk má tabulku rozdělení disku, kde je místo pro čtyři záznamy. Každý takový záznam znamená jeden primární nebo rozšířený oddíl. Rozšířený oddíl je však povolen pouze *jeden*.

Primární oddíl se skládá ze souvislého množství cylindrů (fyzických oblastí disku), které jsou přiřazeny danému operačnímu systému. Při použití pouze primárních oddílů byste byli omezeni na maximální počet čtyři, protože více oddílů nelze zapsat do tabulky rozdělení disku.

Z výše uvedeného důvodu se používají rozšířené oddíly. Jedná se také o souvislé oblasti fyzických disků, ale rozšířený oddíl může být dále rozdělován na *logické disky*. Logický disk nepotřebuje záznam v tabulce rozdělení disků. Jinými slovy rozšířený oddíl může obsahovat logické disky.



*Obrázek 1.5: Výběr typu myši*

Pokud potřebujete více než čtyři oddíly, vytvořte jeden z oddílů (čtvrtý nebo i dřívější) jako rozšířený. Tento oddíl by měl zabírat celý zbytek rozsahu cylindrů disku. Potom v něm můžete vytvořit jeden nebo více logických disků. Maximální počet takových oddílů je patnáct na SCSI discích a 63 (E)IDE discích.

Je víceméně jedno jaké oddíly jsou použity pro Linux. Primární oddíly a logické disky splní funkci stejně dobře.

### Potřebné místo na disku

YaST při standardní instalaci nabídne použitelné schéma rozdělení disku s dostatečným prostorem pro instalaci systému. Pokud chcete rozdělit disk podle svého, mějte na paměti následující doporučení která se týkají prostoru na disku.

**Minimální systém: 500MB** Instalace bez grafického rozhraní (X Window System), což znamená že na systému bude přístupná jen konzola. Z ostatních softwarových balíků je proveden jen základní výběr.

**Minimální grafický systém: 700MB** Tento výběr zahrnuje X Window System a další aplikace.

**Standardní systém: 2,5GB** Tento výběr zahrnuje nová pracovní prostředí jako KDE nebo GNOME a poskytuje dostatek prostoru pro instalaci rozsáhlých aplikací jako OpenOffice a Netscape nebo Mozilla.

V závislosti na volném místě a budoucím použití počítače rozložte instalaci na dostupné disky. Rozdělování disků by se mělo řídit těmito základními pravidly:

**Do 4GB:** Jeden oddíl pro swap a jeden pro kořenový souborový systém (/).  
V tomto případě bude kořenový souborový systém obsahovat i adresáře, které se někdy instalují na jiné oddíly.

**4GB a více:** Budete potřebovat odkládací oddíl, oddíl pro kořenový souborový systém (1GB), a jeden oddíl pokud možno pro každý z následujících adresářů: `/usr` (4GB nebo více), `/opt` (4GB nebo více) a `/var` (1GB). Zbytek volného místa můžete použít pro adresář `/home`.

Podle použitého hardware se také může vyplatit vytvořit speciální oddíl pro start systému (obsahující adresář `/boot`), který bude obsahovat soubory nutné pro start systému a Linuxové jádro. Tento oddíl by měl být na začátku pevného disku a měl by mít velikost alespoň 8MB nebo 1 cylindr. Platí pravidlo, že tento oddíl by měl být vytvořen vždy pokud ho YaST's nabídne v originálním návrhu instalace. Pokud si nejste jisti, bezpečnější je bootovací oddíl vytvořit.

Mějte na paměti, že některé (většinou komerční) programy instalují svá data do adresáře `/opt`. To může být důvodem k vytvoření zvláštního oddílu pro adresář `/opt` nebo k vytvoření dostatečně velkého kořenového souborového systému. KDE a GNOME jsou také instalovány do adresáře `/opt`.

## Tip

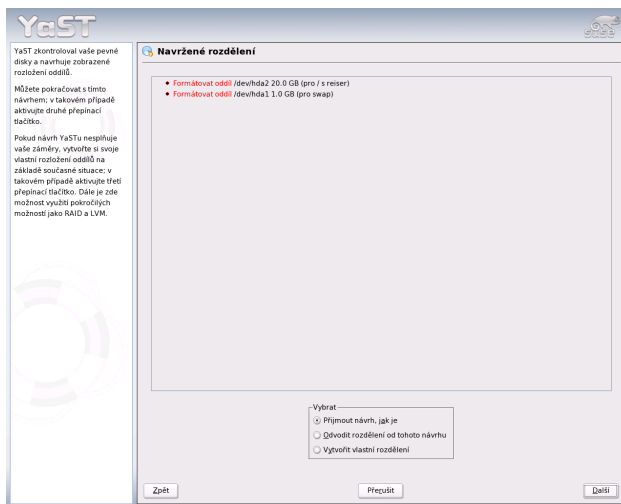
### Tipy pro dělení disku

Všechno bude pravděpodobně v pořádku pokud vaše rozdělení oddílů bude podobné návrhu, který vám předložil YaST. Obvykle se jedná o malý oddíl pro `/boot` na začátku disku (velký okolo 10MB, nebo 1cylindru na větších discích odkládací oddíl (mezi 256 a 500MB), a zbytek systému pro `/`.

## Tip

## Rozdělení disku pomocí programu YaST

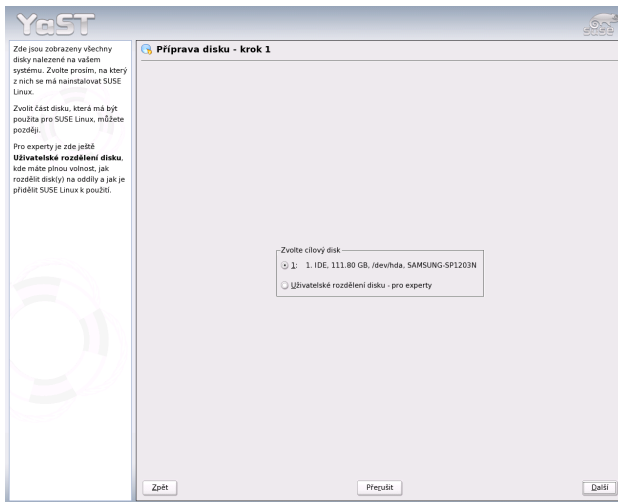
Když vyberete položku rozdělení disku v okně návrhu poprvé, YaST zobrazí okno s navrhovanými oddíly. Můžete je přijmout beze změny nebo provést úpravy před tím, než budete pokračovat. Také můžete nastavení celé zrušit a začít znovu od začátku.



*Obrázek 1.6: Úprava rozdělení disku*

Pokud nechcete v rozvržení oddílů nic měnit, vyberte ‘Přijmout návrh, jak je’. Pokud vyberete ‘Vytvořit vlastní rozdělení’, spustí se ‘Rozdělování disku pro experty’. Zde máte možnosti nastavit rozdělení disku velmi podrobně, průvodce je vysvětlen v části 2.8.11 na straně 68. Původní návrh rozdělení, který vytvořil YaST, bude použit jako základ pro další nastavení.

Když vyberete ‘Vytvořit vlastní rozdělení’, otevře se vám okno jak je ukázáno na obrázku 1.7 na této straně. Vyberte si jeden z existujících disků ve vašem počítači v seznamu a SUSE LINUX bude na tento disk nainstalován.



*Obrázek 1.7: Výběr pevného disku*

Dále je třeba stanovit jestli má být pro instalaci použit celý disk (‘Použít celý disk’) nebo jestli má být instalace provedena na jeden z již vytvořených oddílů. Pokud byl již na počítači instalován operační systém Windows a byl v něm použit souborový systém FAT nebo NTFS, můžete být dotázáni na smazání nebo zmenšení jeho oddílu. Před tím, než tak učiníte, přečte si sekci 1.5.4 na následující straně. Pokud je třeba, můžete už v této fázi instalace zvolit položku ‘Rozdělení disku pro experty’ a dále podrobněji rozdělit disk (viz 2.8.9 na straně 63).

## Varování

### Instalace, která používá celý disk

Když vyberete 'Použít celý disk', všechna data na zvoleném disku budou smazána a tím nenávratně ztracena v dalších krocích instalačního procesu.

## Varování

YaST v průběhu instalace kontroluje jestli je na cílovém disku dostatek místa pro všechny software vybraný v návrhu. Pokud ne, YaST automaticky odebere některé softwarové komponenty z instalace. Okno návrhu pak bude obsahovat upozornění. V případě, že na cílovém disku je dostatek místo pro uskutečnění instalace bude YaST prostě akceptovat vaše nastavení a provede podle něj rozdělení disku.

### Změna velikosti oddílů Windows

Pokud harddisk obsahuje oddíl se souborovým systémem Windows FAT nebo NTFS a vybrali jste tento oddíl jako cíl instalace, YaST vám nabídne smazání tohoto oddílu nebo zmenšení jeho velikosti. Tímto způsobem můžete nainstalovat SUSE LINUX i když v tom okamžiku nemáte na harddisku dostatek místa. Tato funkcionality je užitečná obzvláště pokud cílový harddisk obsahuje pouze jeden oddíl Windows, který zabírá celý disk. To se stává zejména na počítačích, do kterých byla Windows předinstalována. Pokud YaST zjistí že na vybraném harddisku není dost místa ale místo by mohlo být vytvořeno smazáním nebo zmenšením oddílu Windows, nabídne okno ve kterém si můžete vybrat jednu z těchto možností.

Pokud vyberete 'Smazat Windows kompletně', celý oddíl Windows bude označen ke smazání a volné místo bude použito pro instalaci systému SUSE LINUX.

## Varování

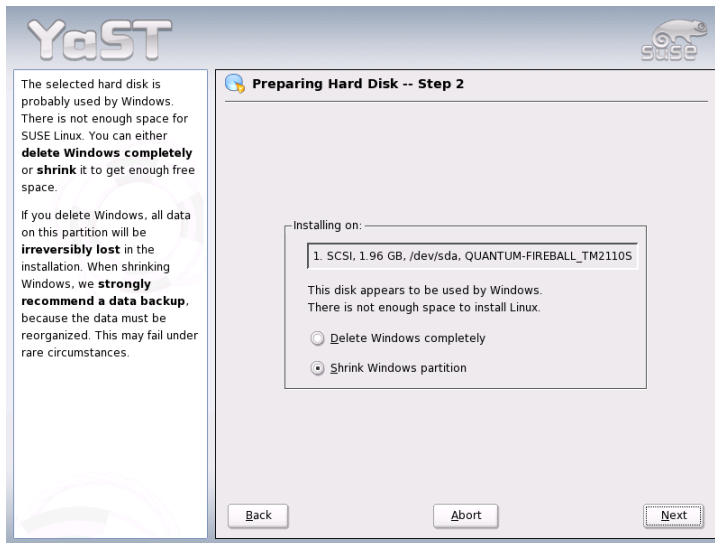
### Mazání Windows

Pokud smažete oddíl Windows, všechna data budou ztracena bez možnosti jejich obnovy jakmile začne formátování.

## Varování

Pokud chcete zmenšit oddíl Windows, přerušte instalaci a připravte oddíl z prostředí Windows. Pro oddíly se souborovým systémem FAT to není nutné, dojde však ke zrychlení procesu změny velikosti. Tento krok je však nezbytně nutný pro oddíly se souborovým systémem NTFS.





Obrázek 1.8: Možnosti pro oddíly Windows

**Souborový systém FAT** Ve Windows nejdříve spusťte `scandisk` abyste se ujistili že FAT neobsahuje ztracené fragmenty souborů a křížové odkazy. Poté spusťte aplikaci `defrag`, která přesune soubory na začátek oddílu. Tento krok zrychlí změnu velikosti souboru v Linuxu.

Pokud máte virtuální paměť ve Windows nastavenou tak, že používá souvislý odkládací soubor se stejnou minimální a maximální velikostí, další kroky zvažte. S tímto nastavením Windows může zmenšení harddisku způsobit rozdělení odkládacího souboru do mnoha malých částí rozptýlených po celé oblasti FAT. Také bude v průběhu změny velikosti přesunut celý odkládací soubor, což celý proces zpomalí. Je proto užitečné vypnout tuto optimalizaci Windows a znovu ji zapnout poté co bude změna velikosti dokončena.

**Souborový systém NTFS** Ve Windows spusťte aplikaci `scandisk` a `defrag` k přesunutí souborů na začátek harddisku. Oproti souborovému systému FAT *musíte* tyto kroky udělat než budete pokračovat. Jinak nemůže být velikost NTFS oddílu změněna.

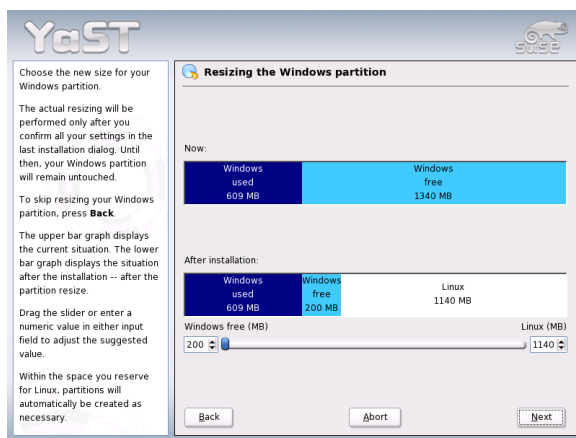
## Důležité

### Vypínání odkládacího souboru Windows

Pokud používáte váš systém s trvalým odkládacím souborem na NTFS oddílu, může se tento soubor nacházet na konci harddisku a zůstane tam bez ohledu na aplikacidefrag. Z toho důvodu pak nemusí být změna velikosti oddílu možná. V takovém případě dočasně deaktivujte odkládací soubor (virtuální paměť ve Windows). Poté co bude velikost oddílu změněna, znovu virtuální paměť nakonfigurujte.

## Důležité

Po těchto přípravných krocích se vraťte do nastavení oddílů v Linuxu a vyberte volbu 'Zmenšit windowsový oddíl'. Po rychlé kontrole oddílu otevře YaST okno s návrhem pro změnu velikosti oddílu Windows.



**Obrázek 1.9:** Změna velikosti oddílu Windows

První sloupec ukazuje kolik místa je v současnosti zabráno Windows a kolik je k dispozici. Druhý sloupec znázorňuje jak bude místo rozděleno po změně velikosti na základě návrhu systému YaST (obr.1.9 na této straně). Přijměte navrhovaná nastavení nebo použijte ovládací prvky ke změně velikosti oddílů (s určitými omezeními).

Pokud toto okno opustíte výběrem 'Další', nastavení budou uložena a vy se navrátíte do předchozího okna. Vlastní změna velikosti se odehraje později, před tím než budou oddíly naformátovány.

## Důležité

### Systém Windows instalovaný na oddílu NTFS

Windows ve verzích NT, 2000 a XP používají souborový systém NTFS jako výchozí volbu. Na rozdíl od systému FAT může být k NTFS systému v současnosti přistupováno z Linuxu pouze pro čtení. Proto můžete číst vaše Windows soubory z Linuxu, ale nemůžete je editovat. Pokud chcete přistupovat k datům vašich Windows i pro čtení a nepotřebujete souborový systém NTFS, nainstalujte Windows na souborový systém FAT32. V něm máte plný přístup k vašim datům ze systému SUSE LINUX.

## Důležité

## 1.5.5 Software

SUSE LINUX obsahuje množství softwarových komponent pro různé účely. Výběr jednotlivých softwarových balíčků by byl velmi komplikovaný, proto SUSE LINUX nabízí tři typy instalovaného systému s předdefinovaným výběrem software. V závislosti na volném místě na disku program YaST vybere jeden z nich a zobrazí vám jej v okně návrhu.

### Minimální systém (doporučen zejména pro zvláštní účely)

Tento výběr obsahuje jádro operačního systému s některými službami, ale bez grafického uživatelského rozhraní. Počítač může být ovládán jen pomocí ASCII terminálů (včetně lokální klávesnice a obrazovky). Minimální systém se používá zejména pro serverové instalace, kde se nepředpokládá přímá práce uživatelů.

### Minimální grafický systém (bez KDE)

Pokud nechcete, aby bylo do počítače nainstalováno grafické prostředí KDE nebo pokud nemáte na disku dostatek místa, vyberte tento typ instalace, která obsahuje rozhraní X Window System a základní grafické prostředí. Můžete použít většinu programů, které mají grafické rozhraní. Výběr nezahrnuje žádné kancelářské aplikace.

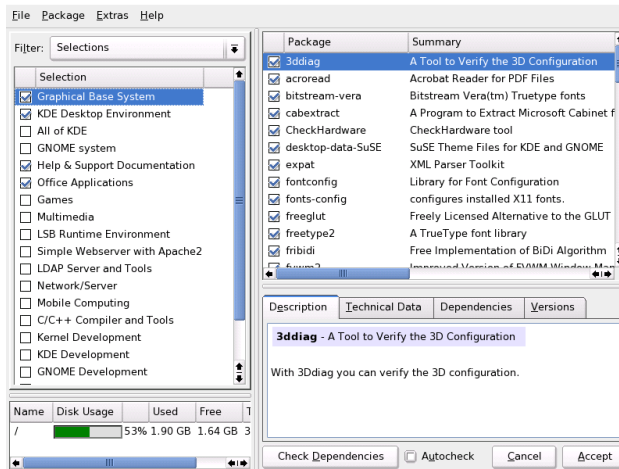
### Standardní systém (s GNOME a kancelářským balíkem)

Tento výběr je z hlediska množství instalovaného software největší. Obsahuje grafické prostředí GNOME s mnoha programy které toto prostředí obsahuje a navíc jsou instalovány kancelářské aplikace. Často se tento výběr používá pro standardní pracovní stanice. Pokud je to možné, YaST vybere tuto možnost automaticky.

### Standardní systém (s KDE a kancelářským balíkem)

Tento výběr je z hlediska množství instalovaného software největší. Obsahuje grafické prostředí KDE s mnoha programy které toto prostředí obsahuje a navíc jsou instalovány kancelářské aplikace. Často se tento výběr používá pro standardní pracovní stanice. Pokud je to možné, YaST vybere tuto možnost automaticky.

Klikněte na 'Výběr softwaru' v okně návrhu a otevře se vám okno ve kterém si můžete vybrat jeden z předdefinovaných výběrů. Pokud chcete spustit modul programu YaST pro správu instalovaného software (správce balíčků) a změnit obsah instalace vašeho počítače, klikněte na 'Detailní výběr'. Viz obr. 1.10 na této straně.



*Obrázek 1.10: Instalace a odinstalace programů s použitím správce balíčků programu YaST*

## Změna typu instalace

Pokud provedete instalaci standardního systému, většinou nemusíte přidávat nebo odebírat jednotlivé programové balíky. Předdefinované výběry jsou složeny tak, aby vyhověly většině vašich požadavků bez nutnosti dalších změn. Pokud je třeba změnit výběr instalovaného software, použijte správce balíků, který tuto činnost značně zjednodušuje. Nabízí několik filtrovacích kritérií, které vám pomůžou se zorientovat v množství softwarových komponent, které dohromady tvoří SUSE LINUX.

Výběr filtru je umístěn vlevo nahoře, pod menu. Po startu modulu je aktivní filtr 'Výběry'. Tento filtr třídí programové balíky podle účelu použití, jako třeba multimediální aplikace nebo kancelářský software. Všechny skupiny jsou zobrazeny pod políčkem výběru filtrovacího kritéria. Předvybrané jsou ty balíky, které jsou obsaženy v aktuálním typu instalovaného systému. Klikněte do příslušných políček a tak vyberte další nebo naopak deaktivujte instalaci dalších programových balíků, popřípadě celé jejich skupiny.

Pravá část okna zobrazuje tabulku s jednotlivými balíky, které jsou obsaženy v aktuálně vybraném typu instalace. První sloupec tabulky ukazuje status každého balíku. Pro instalaci jsou zejména důležité dva stavy: 'Instalovat' (políčko před jménem balíku je zaškrtnuto) a 'Neinstalovat' (políčko je prázdné). Pro aktivaci a deaktivaci jednotlivých balíků klikejte na políčko dokud se neobjeví vámi požadovaný status.

Kromě toho můžete pravým tlačítkem myši zobrazit kontextové menu, které obsahuje všechny možné stavy daného prvku. Většina z nich není ale pro instalaci důležitá. Pro více informací o tomto modulu si přečtěte detailní popis v části 2.3.5 na straně 43.

## Další filtry

Klikněte do pole výběru filtrů a uvidíte další možná filtrovací kritéria. Výběr podle položky 'Skupiny balíčků' můžete také s výhodou použít při instalaci. Tento filtr setřídí softwarové balíky podle jejich účelu do stromové struktury v levé části okna. Čím více rozbalíte jednotlivé větve stromu, k tím přesnějšímu výběru balíků se dostanete a tím méně balíků se vám ukáže v příslušném seznamu v levé části obrazovky.

Můžete také použít filtr 'Hledat' k nalezení specifického balíku podle jména nebo popisku. Použití hledání je detailně popsáno v části 2.3.5 na straně 43.

## **Závislosti a konflikty mezi softwarovými balíky**

Podobně jako jiné operační systémy má SUSE LINUX určitá omezení v tom, který software lze použít v kombinaci s jiným a který ne. Různé softwarové balíky musí být kompatibilní, jinak mezi nimi může nastat konflikt, který ovlivní celý instalovaný systém. Z tohoto důvodu budete upozorňováni na nevyřešené závislosti nebo konflikty mezi softwarovými balíky poté co vyberete nebo se pokusíte odstranit nějaký další softwarový balík. Pokud instalujete SUSE LINUX poprvé nebo upozorněním nerozumíte, přečtěte si nejprve část 2.3.5 na straně 43, která obsahuje podrobné informace o tom, jak pracovat se správcem balíků a také shrnutí celkové organizace software v Linuxu.

### **Varování**

Software předvybraný pro instalaci vychází z dlouhodobé zkušenosti a ve valné většině případů plně vyhoví téměř všem nováčkům a pokročilým domácím uživatelům. Víceméně není třeba měnit v této sekci žádná nastavení. Pokud ale chcete vybrat nebo naopak neinstalovat některé softwarové balíčky, měli byste si být vědomi možných budoucích následků. Zejména byste se měli řídit informacemi uvedenými ve varováních a být opatrní při neinstalování balíků, které jsou součástí základního systému.

### **Varování**

## **Ukončení výběru software**

Pokud jste spokojeni s výběrem software a všechny závislosti a konflikty jsou úspěšně vyřešeny, klikněte na 'Přijmout'. Všechny změny budou aktivovány a vy opustíte konfigurační modul. Pokud jste dané úpravy prováděli v již instalovaném systému, projeví se změny hned. Pokud se jedná o instalaci systému, změny se pouze zaznamenají a budou aplikovány později, v průběhu vlastní instalace.

## **1.5.6 Konfigurace spouštění (instalace zavaděče)**

V průběhu instalace vám YaST nabídne konfiguraci spouštění pro váš počítač. Ve většině případů můžete toto nastavení nechat beze změny. Pokud ale potřebujete vlastní nastavení, můžete upravit návrh tak jak je potřeba.

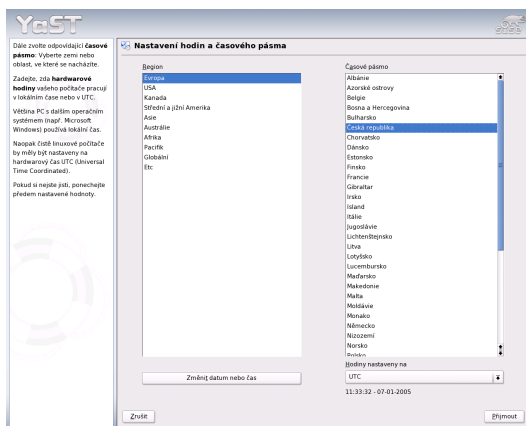
Jendou z možností je konfigurace spouštění počítače z diskety. Ačkoliv má tento způsob své nevýhody spočívající třeba v nutnosti použít disketu při každém

startu, nechává existující mechanismus spouštění počítače beze změn. V normálních případech byste tuto funkcionalitu neměli potřebovat, protože YaST může být konfigurován také pro start vašeho stávajícího operačního systému. Další variantou je změna umístění zaváděcích mechanismů na disku.

Pokud chcete změnit konfiguraci spouštění počítače pomocí programu YaST, vyberte v menu položku 'Spouštění' a otevře se vám okno ve kterém můžete nastavit každý detail mechanismu spouštění počítače. Pro více informací si můžete přečíst část 8.4 na straně 176. Úprava způsobu spouštění je určena pouze pro pokročilé uživatele.

### 1.5.7 Časová pásma

V tomto okně, které uvidíte na obrázku 1.11 na této straně, můžete vybrat mezi Místní čas a UTC v poli 'Hodiny nastaveny na'. Výběr závisí na tom, jak jsou nastaveny hardwarové hodiny v BIOSu vašeho počítače. Pokud jsou nastaveny na GMT, což koresponduje s časovým pásmem UTC, můžete nechat přechod z letního na zimní čas a zpět plně na systému SUSE LINUX



Obrázek 1.11: Výběr časového pásma

### 1.5.8 Jazyk

Jazykové nastavení jste již jednou zvolili na začátku instalace (v části 1.3 na straně 8). Zde můžete toto nastavení v případě potřeby ještě změnit. Pokud chcete, můžete ještě v sekci 'Detaily' nastavit jazyk pro uživatele `root`. Máte na výběr tři různé možnosti:

**ctype** Pro uživatele `root` bude použita hodnota proměnné `LC_CTYPE` v souboru `/etc/sysconfig/language`. To nastaví lokalizaci pro jazykově specifická volání funkcí.

**Ano** Uživatel `root` bude mít stejné nastavení jako ostatní uživatelé počítače.

**Ne** Jazyková nastavení pro uživatele `root` nebudou vůbec závislá na výběru jazyka.

Některým správcům systému nevyhovuje, když má uživatel `root` účet s podporou UTF-8. Podporu lze vypnout odškrtnutím 'Použít kódování UTF-8'.

Seznam níže obsahuje dialog pro výběr dalších podporovaných jazyků v systému. Po výběru dalšího jazyka či jazyků se při instalaci automaticky doinstalují všechny balíčky potřebné pro tento jazyk/y.

Klikněte na 'OK' pro ukončení konfigurace nebo 'Zrušit' k navrácení k původně navrženým hodnotám.

### 1.5.9 Spuštění instalace

Když budete spokojeni s nastavení instalace, klikněte v okně návrhu na tlačítko 'Další' a zahajte tak instalaci. Potvrďte tlačítkem 'Ano' v posledním varování. Instalace většinou trvá patnáct až třicet minut, v závislosti na rychlosti instalovaného počítače. Jakmile budou všechny softwarové balíky nainstalovány, YaST nainstaluje nový Linuxový systém, ve kterém již můžete zkonfigurovat váš hardware a nastavit základní služby.

## 1.6 Dokončení instalace

Pro ukončení instalace všech vybraných softwarových balíčků a základním nastavením zadejte heslo správce systému (uživatele `root`). Poté můžete nastavit typ vašeho připojení k internetu nebo provést aktualizaci systému. Pokud chcete, můžete nastavit server centralizující jména uživatelů v lokální síti. Posledním krokem je nastavení hardwarových zařízení připojených k počítači.



## 1.6.1 Heslo uživatele root

Root je jméno superuživatele, správce systému. Na rozdíl od normálních uživatelů, kteří mohou nebo nesmí přistupovat k různým částem systému, root má neomezenou působnost ve všech administrativních operacích: změnit konfiguraci systému, instalovat nové programy a nastavovat hardware. Pokud uživatelé zapomenou jejich hesla nebo mají jiné problémy s počítačem, root může pomoci. Účet uživatele root by měl být používán jen pro administraci systému, údržbu a opravy. Normální práce pod účtem uživatele root je značně riskantní: i malá chyba může vést k nevratným ztrátám v systémových souborech.

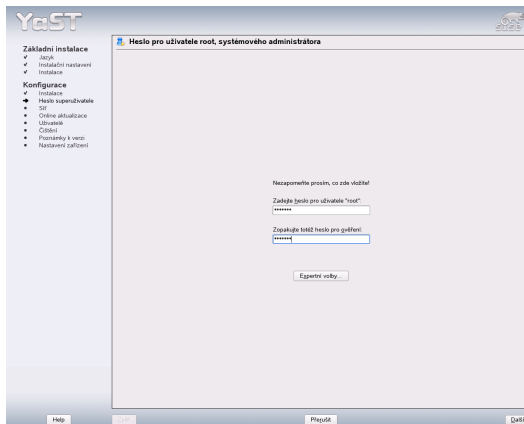
Pro účely kontroly a verifikace musíte zadat heslo uživatele root dvakrát (viz obr. 1.12 na této straně). Toto heslo byste neměli zapomenout. Heslo už nelze ze systému přechíst zpět.

### Varování

#### Uživatel root

Uživatel root má práva k jakýmkoliv změnám v systému. Pro provedení takových nastavení je vyžadováno jeho heslo. Bez něho nelze počítač spravovat.

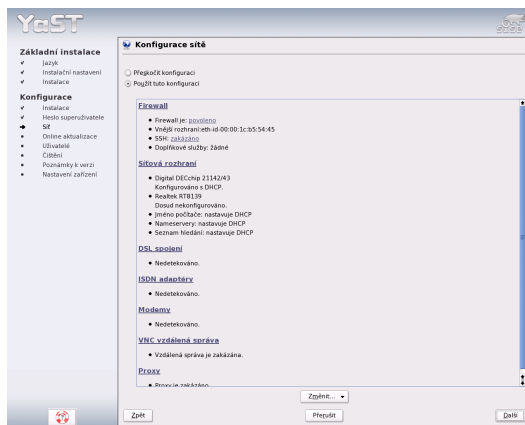
### Varování



Obrázek 1.12: Nastavování root hesla

## 1.6.2 Konfigurace sítě

Nyní můžete konfigurovat síťová zařízení pro lokální síť nebo připojení k Internetu jako síťové karty, modemy a ISDN nebo DSL hardware. Pokud máte síťová zařízení, je nejlepší nastavit je v této fázi instalace protože připojení k Internetu umožní programu YaST zkontrolovat dostupnost případných dalších aktualizací pro systém SUSE LINUX a nainstalovat je ještě v průběhu poslední fáze instalace.



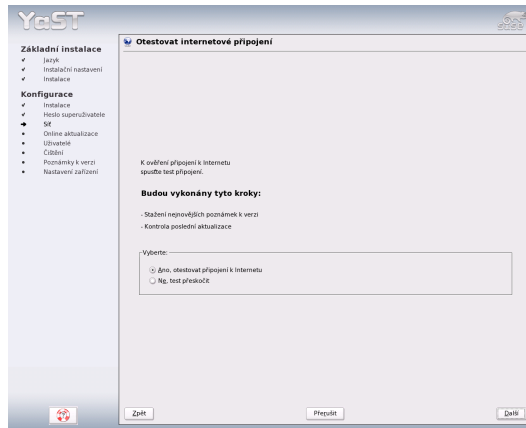
Obrázek 1.13: Konfigurace síťových zařízení

Zvolit můžete také 'Přeskočit nastavení sítě' a potvrdit tlačítkem 'Pokračovat'. Síťová zařízení můžete také konfigurovat až po dokončení instalace.

## 1.6.3 Testování spojení do Internetu

Pokud jste připojeni k Internetu, můžete funkčnost připojení otestovat. YaST vytvoří spojení se serverem SUSE a zkontroluje jestli jsou dostupné nějaké aktualizace pro vaši verzi systému SUSE LINUX. Pokud ano, mohou být zahrnuty do instalace. Také budou staženy nejnovější poznámky k instalované verzi. Můžete si je přečíst na konci instalace.

Pokud v tomto okamžiku nechcete spojení testovat, vyberte 'Přeskočit test' a 'Další'. Tento krok také vynechá stahování aktualizací a poznámek.



Obrázek 1.14: Test spojení do Internetu

## 1.6.4 Aktualizace

Pokud se YaST byl schopen připojit na jeden ze serverů SUSE, můžete ihned provést YaST online aktualizaci. Pokud jsou na serverech dostupné nějaké serrerverové balíky, budou staženy a instalovány s opravami chyb nebo bezpečnostních problémů.

### Důležité

#### Stahování aktualizací

Stahování aktualizací může chvíli trvat, v závislosti na rychlosti připojení k Internetu a velikosti stahovaných souborů.

### Důležité

Pro okamžité spuštění aktualizaci vyberte 'Spustit online aktualizaci' a klikněte na 'OK'. Otevře se okno YaST' online update se seznamem dostupných oprav (pokud jsou nějaké k dispozici), které mohou být vybrány a nahrány. O tomto procesu se můžete dočíst více v části 2.3.2 na straně 38. Aktualizaci můžete také provést kdykoliv po skončení instalace. Pokud ji nechcete provádět nyní, vyberte 'Přeskočit aktualizaci' a klikněte na 'OK'.

## 1.6.5 Ověřování uživatelů

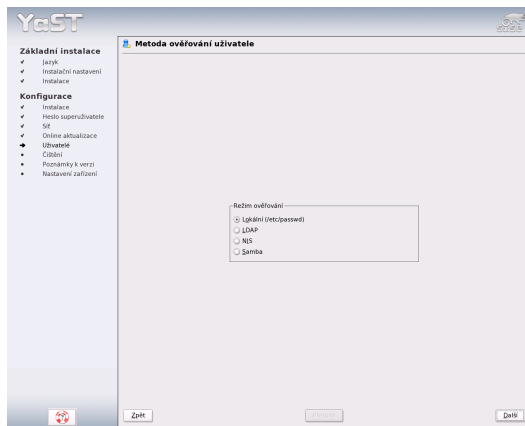
Pokud byl přístup k síti úspěšně nakonfigurován v předchozích krocích instalace, máte nyní další možnosti pro správu uživatelských účtů na vašem počítači.

**Správa lokálních uživatelů** Při použití této metody jsou uživatelé spravováni lokálně, na instalovaném počítači. Toto nastavení je typické pro samostatně používané pracovní stanice.

### Správa uživatelů s pomocí NIS nebo LDAP

Tato metoda je většinou používána v podnicích ke správě pracovních stanic na úrovni jednotlivých oddělení. Správa uživatelů pro celé oddělení je vykonávána na centrálním počítači nebo serveru. V tomto případě nejsou lokální účty třeba. Tato metoda může být také vybrána z důvodu nevhodnosti existence lokálních účtů jako takových.

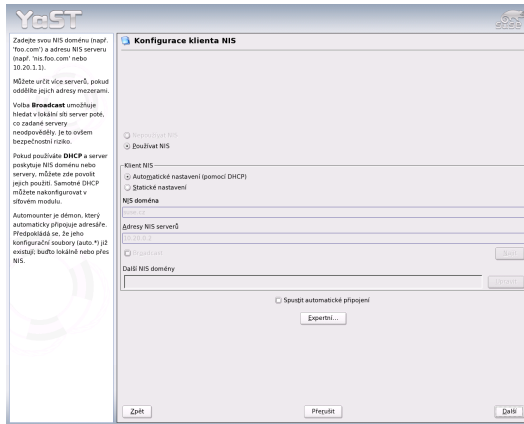
Pokud jsou splněny všechny předpoklady, YaST otevře okno ve kterém můžete vybrat metodu administrace uživatelů. Výběr můžete vidět na obrázku 1.15 na této straně. Pokud nedisponujete připojením k síti, vytvořte lokálního uživatele.



Obrázek 1.15: Ověřování uživatelů

## 1.6.6 Konfigurace počítače jako NIS klienta

Aby mohly být uživatelské účty spravovány pomocí NIS serveru, musíte nakonfigurovat počítač jako NIS klient. Síť, které je postavená na NIS, vyžaduje určité hlubší znalosti. Detaily NIS technologie jsou vysvětleny v manuálu *Průručka správce systému*. Následující text vysvětluje (poměrně jednoduché) nastavení klientské strany.



Obrázek 1.16: Konfigurace NIS klienta

V následujícím okně, které můžete vidět na obrázku 1.16 na této straně, nejprve vyberte jestli má počítač pevnou IP adresu nebo jestli je mu přidělována pomocí DHCP serveru. Pokud vyberete DHCP, nemůžete nastavit NIS doménu nebo adresu NIS serveru, protože tyto údaje by vám měly být také přiděleny DHCP serverem. Více informací o DHCP najdete v kapitole *DHCP* v *Průručce správce systému*. Pokud použijete statickou IP adresu, vyplňte NIS doménu a NIS server ručně.

Pro vyhledání NIS serverů v lokální síti zaškrtněte odpovídající volbu. Můžete také specifikovat více NIS domén a nastavit výchozí. Pro každou doménu vyberte 'Upravit' a nastavte několik adres serveru k zapnutí broadcast funkcionality oddělené pro jednotlivé domény.

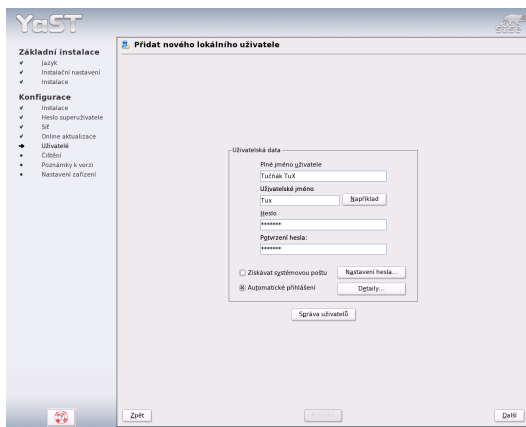
V expertním nastavení můžete použít 'Odpovídat pouze lokálnímu počítači' abyste zabránili jiným počítačům v síti zjistit jaký server používáte. Pokud aktivujete volbu 'Poškozený server', budou akceptovány i odpovědi od serverů na

nepovolených portech. Více informací o této problematice najdete v manuálových stránkách příkazu `ybind`.

## 1.6.7 Vytváření lokálních uživatelských účtů

Pokud se nerozhodnete k použití centrálního autentizačního serveru, musíte vytvořit lokální uživatele. Všechny údaje, které se k uživatelským účtům vztahují (jméno, uživatelské jméno, heslo atd.) budou uloženy a spravovány na instalovaném počítači.

Linux je operační systém, který umožňuje několika uživatelům pracovat ve stejném okamžiku na tomtéž počítači. Každý uživatel potřebuje k práci uživatelský účet, aby se mohl k počítači přihlásit. Osobní data daného uživatele nemohou být modifikována, prohlížena nebo jinak ovlivňována. Každý uživatel si může nastavit vlastní pracovní prostředí, které najde nedotčené při příštím přihlášení.



**Obrázek 1.17:** Zadávání uživatelského jména a hesla

Uživatelský účet můžete vytvořit s použitím dialogového okna ukázaného na obrázku 1.17 na této straně. Poté, co zadáte křestní jméno a příjmení, je nutné specifikovat uživatelské jméno (login). Klikněte na 'Například' a YaST vygeneruje uživatelské jméno automaticky.

Nakonec zadejte heslo pro zadávaného uživatele. Musíte ho zadat ještě jednou pro ujištění, že se nestala při zápisu žádná nechtěná chyba. Uživatelské jméno identifikuje uživatele a heslo zajišťuje jeho autenticitu.

## Varování

### Uživatelské jméno a heslo

Dobře si zvolené uživatelské jméno a heslo zapamatujte. Budete je potřebovat při každém přihlášení do systému.

## Varování

Aby heslo zaručovalo dostatečnou bezpečnost, mělo by být dlouhé mezi pěti a osmi znaky. Maximální délka hesla je 128 znaků. Pokud ale nejsou nahrány speciální bezpečnostní moduly, je pro kontrolu hesla používáno jen prvních osm znaků. Hesla jsou citlivá na velká a malá písmena a nejsou v nich povoleny akcentované znaky (například s čárkami a háčky). Různé speciální znaky z první poloviny ASCII tabulky a číslice jsou v heslech povoleny.

Pro lokální uživatele lze uplatnit dvě další volby:

**‘Získávat systémovou poštu’** Pokud zaškrtnete tuto volbu, počítač bude hlášky vygenerované systémovými službami posílat tomuto uživateli. Většinou jsou tyto výpisy zasílány pouze uživateli `root`, správci systému.

**‘Automatické přihlášení’** Tato volba je dostupná jen v případě, že je KDE nastaveno jako vaše výchozí prostředí. Zajišťuje automatické přihlášení uživatele k počítači po startu. Tento postup je výhodný zejména pokud je počítač používán jedním uživatelem.

## Varování

### Automatické přihlašování

Pokud je povoleno automatické přihlašování, systém nastartuje přímo do grafického rozhraní daného uživatele bez jakékoliv vyžádané autentizace. Pokud na počítači ukládáte důvěrné informace a k počítači mohou mít přístup i jiné osoby, *nezapínejte* tuto volbu.

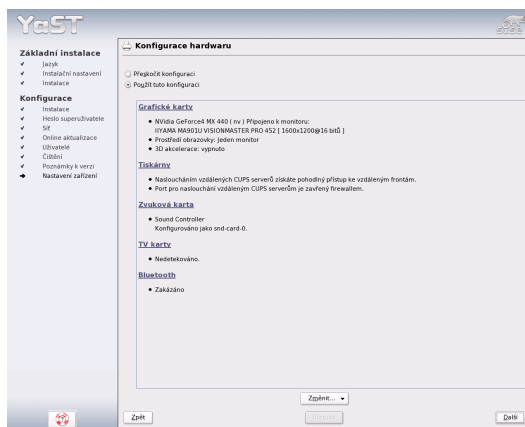
## Varování

## 1.6.8 Čtení poznámek k verzi

Po dokončení autentizace uživatelů YaST zobrazí poznámky k verzi. Přečtete si je, protože mohou obsahovat důležité a aktuální informace které nebyly k dispozici v době vytváření manuálů a příruček. Pokud jste instalovali balíky s aktualizacemi, bude vám k dispozici nejposlednější verze poznámek stažená ze serverů SUSE.

## 1.7 Konfigurace hardware

Na konci instalace YaST otevře okno ve kterém můžete nakonfigurovat grafickou kartu a jiná zařízení, jako jsou tiskárny a zvukové karty. Klikněte na daný komponent a spusťte tak jeho konfiguraci. Většinu součástí počítače bude YaST detekovat a konfigurovat automaticky.



Obrázek 1.18: Konfigurace hardware

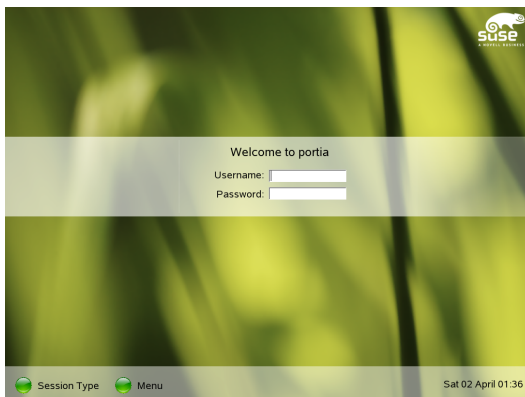
Můžete přeskočit konfiguraci dalších zařízení a provést ji až později v běžícím systému. Měli byste ale provést konfiguraci grafické karty. Ačkoliv jsou nastavení grafiky autodetekována programem YaST a měla by být přijatelně nastavena, většina uživatelů má velmi specifické preference pokud jde o rozlišení, barevnou hloubku a jiné parametry grafiky. Všechna tato nastavení můžete nastavit v sekci 'Grafické karty'. Konfigurace je podrobněji vysvětlena v části 11.1 na straně 214.

Poté, co program YaST zapíše data konfigurace, ukončete instalaci systému SUSE LINUX pomocí tlačítka 'Dokončit' v závěrečném okně.



## 1.8 Přihlašování v grafice

SUSE LINUX je nainstalován. Pokud jste zapnuli automatické přihlašování v modulu správy lokálních uživatelů, naskartuje bez přihlašování. Pokud ne, měli byste na vaší obrazovce vidět grafické přihlášení (viz obr. 1.19 na této straně). Zadejte přihlašovací jméno předem definovaného uživatele a heslo, systém vám pak umožní dále pracovat.



*Obrázek 1.19: Přihlašovací obrazovka*



# Konfigurace pomocí YaST

Tato kapitola je věnována konfiguraci vašeho systému. Konfiguraci zajišťuje YaST, se kterým jste již nainstalovali systém SUSE LINUX. Pomocí programu YaST nastavíte hardware, grafické rozhraní, přístup na Internet, zabezpečení. Použít ho můžete také pro správu uživatelů, instalaci software nebo aktualizaci systému. Po spuštění YaST budete mít v levé části okna záložky s jednotlivými oblastmi správy systému a v hlavním okně pak moduly pro nastavení jednotlivých komponent. YaST zapisuje u většiny modulů nastavení do textových konfiguračních souborů, které je možné v případě potřeby editovat i ručně.

2.1	Spuštění YaST . . . . .	36
2.2	Řídící středisko YaST . . . . .	37
2.3	Software . . . . .	37
2.4	Hardware . . . . .	45
2.5	Síťová zařízení . . . . .	51
2.6	Síťové služby . . . . .	51
2.7	Bezpečnost a uživatelé . . . . .	56
2.8	Systém . . . . .	57
2.9	Různé . . . . .	74
2.10	YaST v textovém režimu (ncurses) . . . . .	75
2.11	Online update z příkazové řádky . . . . .	79

## 2.1 Spuštění YaST

Program YaST funguje na bázi modulů, které použijete pro jednotlivé operace. Jedním z modulů nastavíte typ klávesnice, jiným síťové služby. V závislosti na své platformě a rozsahu instalace můžete spouštět jednotlivé moduly různými způsoby. Přehledný přístup ke všem modulům máte v Řídícím středisku YaST. V KDE nebo GNOME ho spustíte z menu 'SUSE' ('Systém' → 'YaST'). Následně budete vyzváni, abyste vložili heslo uživatele `root`.

### Tip

#### Nastavení jazyka

Jazyk programu YaST, změníte v Řídícím středisku v nabídce 'Systém' → 'Výběr jazyka'. Zvolte požadovaný jazyk, ukončete YaST a odhlašte se ze systému. Při dalším přihlášení a spuštění programu YaST bude již program komunikovat ve zvoleném jazyku.

### Tip

Některé platformy nepodporují přímé připojení zobrazovací jednotky (monitoru) a je nutné je spravovat vzdáleně. Pamatujte, že je v takovém případě potřeba povolit přístup vzdálenému uživateli `root` k vašemu X serveru. Např. příkaz `ssh -X root@<system_k_nastaveni>` povolí přístup všem uživatelům přihlášeným na lokálním počítači.

Následně použijte příkazy:

```
su -  
(zadejte heslo pro superuživatele)  
export DISPLAY=:0.0  
yast2
```

Po ukončení YaSTu použijte příkaz (jako uživatel `root`) `exit`, nebo stiskněte klávesovou zkratku `(Ctrl)-(D)` (v Xtermu) a následně zakažte ostatním uživatelům přístup k vašemu X serveru příkazem `xhost -`.

Další možností, pokud nechcete povolit přístup k vašemu displeji, je nechat `xhost` beze změny a přihlásit se jako uživatel `root` následujícím způsobem:

```
sux -  
(zadejte heslo pro superuživatele)  
yast2
```

Konfigurační nástroj YaST lze spouštět také v textovém režimu, jako uživatel `root`, příkazem `yast`.

## 2.2 Řídící středisko YaST

Po spuštění se zobrazí Řídící středisko. V levé části jsou uvedeny hlavní kategorie:

**Software** správa a instalace softwaru

**Hardware** správa, konfigurace a přidávání hardwaru

**Systém** nastavení zálohování, startování apod.

**Síťová zařízení** základní konfigurace sítě a připojení k Internetu

**Síťové služby** konfigurace pokročilých síťových služeb

**Bezpečnost a uživatelé** správa uživatelů a nastavení bezpečnosti

**Různé** zobrazí např. protokolové soubory

Po zvolení některé z kategorií se zobrazí jednotlivé moduly, které jsou k dispozici. Po spuštění modulu se zobrazí odpovídající dialogové okno, kde můžete provést požadované úpravy. Většinou se konfigurace provádí ve více po sobě jdoucích oknech. Po doplnění informací v prvním okně proto zvolte tlačítko 'Další' a přesunete se k dalšímu dialogu. Po provedení všech potřebných kroků, pak stačí kliknout na poslední dialog 'Konec', čímž uložíte provedené změny a veškerá nastavení uloží.

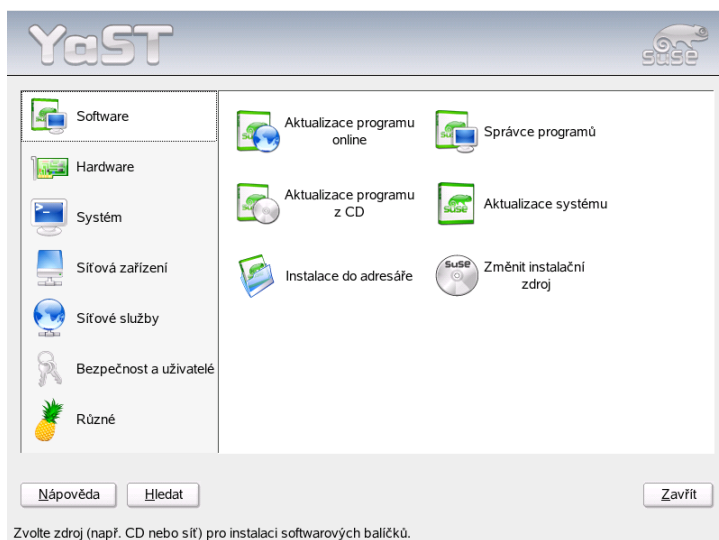
Víte-li přesně, s kterým modulem chcete pracovat, můžete ho přímo spustit příkazem `yast2 nazev_modulu`. Výpis všech modulů získáte příkazem `yast2 -l`.

## 2.3 Software

### 2.3.1 Změnit instalační zdroj

Instalační zdroj je médium, kde jsou k dispozici balíky distribuce SUSE LINUX. Většinou se instalace provádí z CD média, dále pak můžete instalovat prostřednictvím sítě nebo z pevného disku.

Po spuštění modulu se zobrazí seznam všech již dříve zadáných instalačních zdrojů. Pokud jste instalovali pouze z CD, na seznamu bude uvedeno pouze



*Obrázek 2.1: YaST: Řídící středisko*

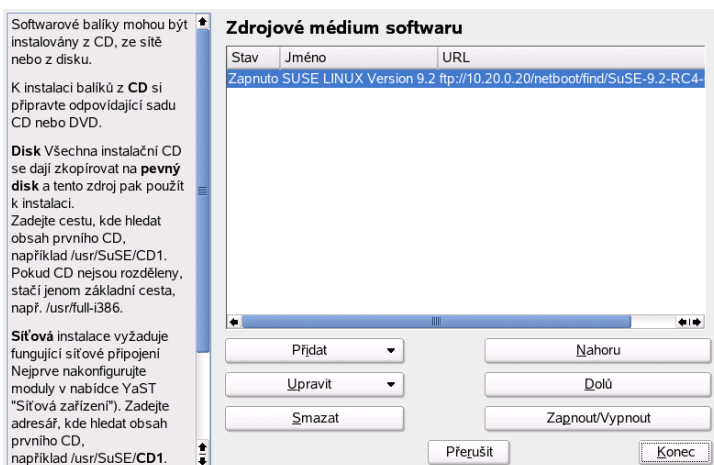
CD. Klikněte na ‘Přidat’ a zadejte další zdroj, odkud chcete instalovat balíky. Přidat můžete cestu k souborům na lokálním pevném disku, výměnná média (CD, DVD) nebo síťové zdroje (NFS, FTP, HTTP, Samba).

Během instalace nebo aktualizace používá YaST veškeré dostupné zdroje. Každá položka má tedy políčko, kde určíte, zda se má používat či ne. Pro změnu stavu použijete tlačítko ‘Zapnout/Vypnout’.

Po vypnutí modulu tlačítkem ‘Zavřít’ se uloží současné nastavení a moduly ‘Správce programů’ a ‘Aktualizace systému’ začnou používat nastavené zdroje.

### 2.3.2 Aktualizace programů on-line

Modul ‘Aktualizace programu on-line’ (YaST Online Update (YOU)) vám pomůže mít systém stále aktuální. Provádí jeho aktualizaci tak, že zkontroluje na vzdáleném SUSE ftp serveru (nebo jeho zrcadle) novější verze balíčků, které pak stáhne a nainstaluje na váš počítač. Samozřejmě až po potvrzení uživatelem. Kromě celých balíčků jsou na ftp serveru také záplaty, které opravují případné nedostatky v zabezpečení systému.



Obrázek 2.2: YaST: Instalační zdroj

Z jakého serveru se budou stahovat balíčky se zadává do položky 'Umístění'. Můžete zvolit v menu 'Zdroj pro instalaci' některý z předem nastavených serverů a jeho adresa URL se překopíruje do řádku 'Umístění'. Tuto adresu můžete následně editovat, nebo sem zapsat i váš vlastní lokální server, který tyto soubory obsahuje (například `file:/muj/adresar/`, `/muj/adresar/`, `ftp://muj.server/cesta/` atd.).

### Důležité

On-line aktualizace vyžaduje správně zkonfigurované internetové připojení, tj. nejdříve musíte nastavit modem nebo síťovou kartu.

### Důležité

Po zapnutí modulu je aktivní položka 'Ruční výběr novinek', která vám umožní rozhodnout se, zda konkrétní záplatu chcete instalovat či ne. K tomu abyste nainstalovali veškeré dostupné záplaty tuto položku vypněte. V závislosti na vašem připojení však může stahování dat probíhat relativně dlouho.

Další možností je aktualizovat váš systém automaticky. Klikněte na 'Konfigurovat plně automatickou aktualizaci...' a nastavte postup, jakým se bude systém sám aktualizovat. Tento proces je plně automatizovaný, takže se již dále nemusíte o

nic starat. Musíte samozřejmě zajistit, aby byl počítač v době, kdy aktualizuje balíčky, schopen se připojit na zadaný aktualizací server.

Pokud se rozhodnete provést interaktivní aktualizaci (implicitní volba), zaškrtněte 'Ruční výběr novinek' a poté na zvolte 'Další'. Zde můžete zakázat nebo povolit instalaci záplaty nebo aktualizované verze balíku. Nyní se spustí správce programů (popsaný v části 2.3.4 na straně 42, jenž má zapnutý filtr a zobrazuje pouze opravné záplaty. Ty aktualizace, jejichž instalace je žádoucí, jsou předem zvolené pro instalaci. Za běžných okolností byste měli schválit tento doporučený výběr.

Jakmile jste hotovi s výběrem aktualizací balíčků, klikněte na 'Přijmout'. Vybrané aktualizace se stáhnou a nainstalují. Jestliže během tohoto procesu nastane chyba, jste o tom informováni v okně. Je-li to nezbytné, přeskočte konkrétní chybový balíček. Některé záplaty mohou otevřít okno a informovat vás o detailech, žádat váš souhlas s instalací, nebo nabídnou možnost přeskočit instalaci této záplaty.

Zatímco se instalují aktualizace, můžete sledovat průběh v okně s protokolem. Po úspěšné instalaci ukončíte modul tlačítkem 'Zavřít'. Pokud nebudete aktualizovat další počítače, zaškrtněte položku 'Po aktualizaci odstranit zdroje balíčků' a po instalaci je YaST smaže. Nakonec se spustí SuSEconfig a upraví konfiguraci systému.

---

### Důležité

Někdy se může stát, že bude třeba provést aktualizaci dvakrát. Poprvé se aktualizuje samotná služba *YOU (YaST on-line Update)* a teprve po její aktualizaci a restartu modulu budou staženy ostatní záplaty.

---

**Důležité**

### Spouštění aktualizace z konzole

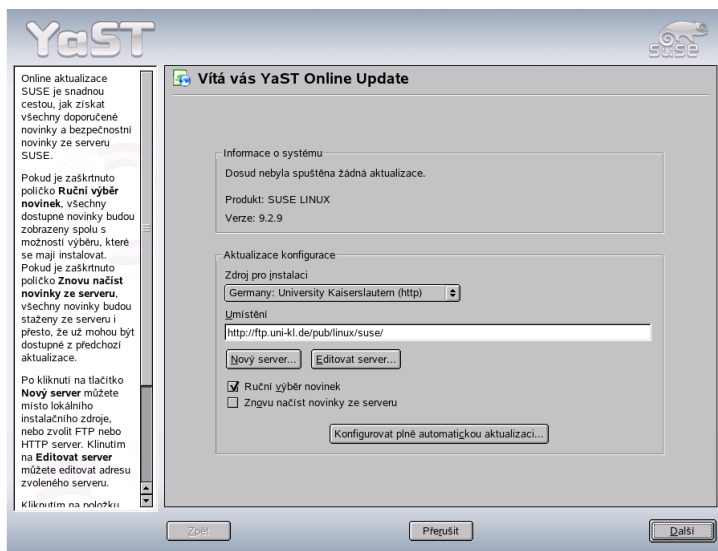
Modul 'Aktualizace programů online' můžete také ovládat z příkazové řádky. Program musíte spouštět jako uživatel `root`.

Po spuštění si program stáhne z prvního ftp serveru v seznamu, který je uložen v `/var/lib/YaST2/you/yourservers`, přehled dostupných oprav a opravné balíčky relevantních nainstalovaných aplikací. To docílíme příkazem `online_update`

Jestliže chceme stáhnout pouze některé opravy, můžeme programu upřesnit zadání pomocí parametrů *security*, *recommended*, *document*, a *optional*.

Parametr *security* zajistí, že se stáhnou opravy týkající se bezpečnosti, *recommended* jsou opravy doporučené SUSE, *document* zjistí informace o opravách





**Obrázek 2.3:** YaST Online aktualizace

a *optional* stáhne menší opravy. Informace o těchto opravách jsou uloženy v `/var/lib/YaST2/you/mnt/i386/update/X.Y`, kde *X.Y* znamená číslo verze systému SUSE LINUX.

K tomu, abyste si stáhli pouze bezpečnostní opravy, pak stačí napsat příkaz `yast2 online_update security`.

Pokud spustíte modul, standardně se uloží nový aktualizovaný seznam SUSE FTP serverů do `/var/lib/YaST2/you/yourservers`. Jestliže nechcete aby vám program přepisoval tento seznam, můžete tuto funkci vypnout v `/etc/sysconfig/onlineupdate`. Zde nastavte řádek `YAST2_LOADFTPSEVER=yes` na *no*.

Chcete-li balíčky pouze stáhnout a neinstalovat, spusťte program s parametrem: `online_update -g`

Tento proces je vhodný hlavně pro správce systémů. Přes noc si stáhnou veškeré opravné balíčky a ráno nainstalují ty, které potřebují.

### 2.3.3 Aktualizace systému

Tento modul vám umožní aktualizovat systém, tj. přejít na novější verzi distribuce.

#### Důležité

Pokud spouštíte aktualizaci za běhu systému, není možné aktualizovat *základní systém*. K tomu je třeba restartovat počítač a použít instalační CD nebo disketu, kde zvolíte aktualizaci systému. Základní systém není možné měnit za běhu stejně, jako si pod sebou nemůžete uříznout větev s tím, že si tam dáte jinou.

#### Důležité

#### Důležité informace o aktualizaci

Aktualizace systému je složitá procedura. Každý nainstalovaný balíček musí být programem YaST zkontrolován a YaST musí určit co je třeba učinit pro aktualizaci jednotlivých balíčků. YaST se snaží do této aktualizace zahrnout i změny nastavení, které provedl uživatel. Nicméně některá nastavení mohou být problémová a způsobit nekonzistenci mezi různými konfiguracemi systému. Týká se to i problému zpětné kompatibility některých programů, které mohou mít potíže s načtením konfiguračních souborů svých starších verzí. Některá nastavení proto musíte provést po aktualizaci systému znovu.

Čím starší verzi SUSE LINUX používáte anebo čím větší zásah do standardní konfigurace jste provedli, tím je větší pravděpodobnost, že narazíte na problémy. Předtím než začnete aktualizovat systém, proved'te zálohu vašeho stávajícího systému.

Tento postup se může hodit, pokud byste chtěli aktualizovat pouze pár aplikací. Při komplexnějších změnách se vyplatí provést aktualizaci restartováním počítače s vloženým CD nebo jiným zdrojem pro aktualizaci.

### 2.3.4 Aktualizace programů z CD

Před spuštěním modulu 'Aktualizace programů z CD' vložte do mechaniky CD se záplatami. Po načtení CD se otevře dialog 'Seznam dostupných novinek'. Zde jsou již předem zvoleny ty záplaty, které jsou relevantní pro váš systém, tj. máte nainstalovány programy, ke kterým se opravy vztahují. Samozřejmě máte možnost zvolit i další položky, případně neaktualizovat některé ze stávajících.

Protože dochází k sjednocování, spustí se vlastně ‘Aktualizace programu online’, kde je vybrán jako instalační zdroj CD.

### 2.3.5 Správce programů

Tento modul v záložce ‘Software’ umožňuje instalovat nebo odinstalovat balíčky s aplikacemi.

#### Důležité

**Balíčky** obsahují komprimované spustitelné soubory, knihovny a další data, která využívá daná aplikace. Jsou zabaleny dohromady tak, aby po nainstalování balíku bylo možné aplikaci ihned spustit. Balíček poznáte podle přípony `.rpm`.

#### Důležité

Některé balíky mohou také vyžadovat přítomnost jiných balíků, jsou na něm *závislé*. YaST vám při instalaci balíku oznámí, že je zde závislost na jiném balíku a zeptá se, zda si přejete nechat vyřešení závislostí na něm. Navíc se YaST stará také o kolidující balíky. Všechny informace o závislostech balíku a mnoho dalšího je uvedeno v hlavičce balíku.

Pokud instalujete z CD/DVD, vložte nejdříve instalační médium do mechaniky. Po spuštění se zobrazí okno s několika rámci. Velikost těchto rámců můžete změnit myší kliknutím na linky, které je oddělují. V následujícím textu bude popsán obsah těchto rámců.

#### Filtr

Vybírat všechny balíky instalace jeden po druhém může být velice pracné a zdoluhavé. Proto nabízí správce programů možnost použít filtry pro zjednodušení práce s balíky. Okno s filtrem je v levém horním rohu aplikace. Vybírat můžete z těchto filtrů:

**Výběry** Po spuštění je aktivní tento filtr. Seskupuje balíky s aplikacemi podle jejich účelu (*Multimédia, Kancelářské aplikace* atd.). Tyto výběry jsou vypsány v okně pod oknem filtru. V pravém okně můžeme vidět seznam balíčků zvoleného výběru. Vlevo od názvu výběru je políčko znázorňující stav - zaškrtnutý znamená nainstalovaný. Pokud chceme nainstalovat některý další výběr, zaškrtneme jej.

**Skupiny balíčků** Zde naleznete více technický přehled balíčků. Je vhodný pro zkušenější uživatele systému SUSE LINUX. Filtr uspořádá programové balíčky podle určení do stromové struktury (např. *Dokumentace, Vývoj, Hardware ...*). Čím více se vnoříte do struktury, tím zjemňujete výběr balíčků zobrazených vpravo.

Navíc můžete tímto filtrem zobrazit *všechny* balíčky uspořádané podle abecedy. To uděláte kliknutím na položku 'zzz Vše'. Protože SUSE LINUX obsahuje mnoho balíčků, může chvíli trvat než se zobrazí seznam programových balíčků.

**Hledat** Nejjednodušší cesta, jak nalézt konkrétní balíček. Hledat můžete podle jména, popisu, shrnutí, zda poskytuje konkrétní soubor, nebo zda ho vyžaduje. Zkušenější uživatelé mohou vyhledávat i pomocí expanzních znaků (tzv. wild cards) nebo regulárních výrazů.

---

#### Tip

Kdykoliv můžete prohledávat libovolný seznam. Stačí pouze myší kliknout do seznamu, a začít psát počáteční písmena názvu položky, kterou hledáte.

---

#### Tip

**Souhrn instalace** Zde si můžete prohlédnout seznam balíčků, které jste se rozhodli instalovat, aktualizovat nebo odstranit. Zobrazuje vlastně co se stane, pokud kliknete na 'Přijmout'. Pro změnu můžete použít zaškrťovací políčka vlevo od názvu balíčku. Podrobný popis, a vysvětlení jednotlivých ikon stavu balíčku, najdete v menu 'Nápověda', položka 'Symbols'.

Pokud jste hotovi s výběrem co nainstalovat/odinstalovat, tlačítkem 'Přijmout' spustíte instalaci balíků. V instalačním okně můžete sledovat průběh instalace. Po instalaci všech zvolených balíků je automaticky spuštěn *SuSEconfig*. Ten aktualizuje systémové a konfigurační soubory v závislosti na nainstalovaném softwaru. To si může vyžádat určitý čas (program často přistupuje k disku).

---

#### Varování

Při odstraňování balíků dbejte na doporučení programu YaST tak, abyste zachovali konzistenci operačního systému.

---

#### Varování

## 2.4 Hardware

Nejdříve musí být nový hardware zapojen do systému podle informací od výrobce. Připojte a zapněte odpovídající zařízení (např. tiskárnu) a spusťte modul (v našem příkladu modul *Tiskárna*. Pokud budete připojovat modem nebo jiné síťové zařízení, pak naleznete odpovídající moduly v kategorii *Síťová zařízení*.

Většina připojovaných zařízení je automaticky rozpoznána a provede se automatická konfigurace zařízení. Pokud YaST automaticky nerozpozná nové zařízení, pak máte možnost ho zvolit ze seznamu podporovaných zařízení, kde vyberete výrobce a název zařízení.

### Důležité

Pokud váš model není uveden v seznamu zařízení, pak můžete zkusit zvolit typově příbuzný model. To ale nemusí fungovat vždy, protože v některých případech i dvě podobná zařízení jedné typové řady nemusí instrukce systému interpretovat stejným způsobem.

### Důležité

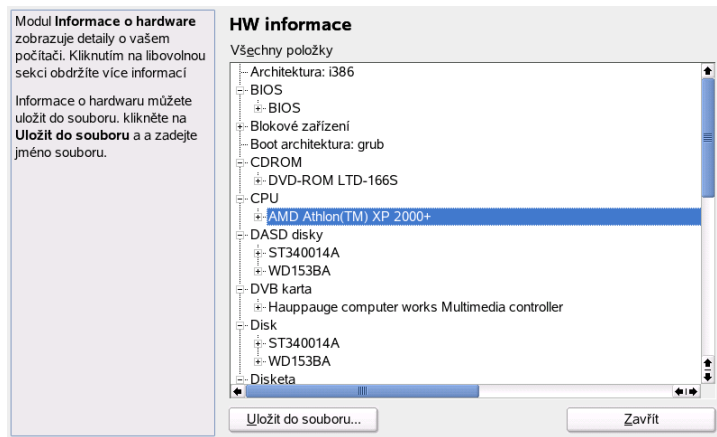
### 2.4.1 CD-ROM mechaniky

Během instalace systému jsou všechny nalezené mechaniky CD-ROM integrovány do systému. Je pro ně vytvořena položka v souboru `/etc/fstab` a podadresář v adresáři `/media`. Tento modul můžete použít pro přidání dalších mechanik do systému.

Po zapnutí modulu vypíše YaST seznam nalezených mechanik. Zaškrtněte novou mechaniku a klikněte na tlačítko 'Konec'. Nová CD-ROM mechanika byla právě integrována do systému.

### 2.4.2 Informace o hardwaru

YaST před konfigurací provádí automatické rozpoznání hardwaru. Informace o rozpoznaných zařízeních se pak zobrazí v tomto modulu. Ty se hodí především při kontaktování instalační podpory, kdy budete potřebovat informace o vašem hw vybavení. Výpis můžete uložit do textového souboru.



*Obrázek 2.4: Zobrazení informací o hardwaru*

### 2.4.3 Nastavení IDE DMA

Tento modul slouží pro aktivaci tzv. DMA režimu pro vaše IDE disky a CD/DVD mechaniky. Zapnutí režimu může výrazně zvýšit výkon při datových přenosech. Modul nijak neovlivní výkon SCSI zařízení.

Během instalace SUSE LINUX jádro automaticky aktivuje DMA u pevných disků, ale ne u CD mechanik. Zapnutí DMA pro všechny mechaniky totiž často způsobí potíže s CD. Můžete tedy zkusit, zda vám DMA s CD mechanikou bude fungovat. Pokud bude CD pracovat korektně, dojde k velkému nárůstu výkonu. Pokud narazíte na problémy, stačí u CD opět vypnout DMA.

#### Důležité

DMA (Direct Memory Access) znamená, že data jsou přenášena ze zařízení přímo do RAM bez zatěžování CPU.

#### Důležité

### 2.4.4 Joystick

Zde můžete nastavit joystick. Vyberte výrobce a model ze seznamu a pomocí položky 'Test' otestujte funkčnost. Protože se joystick obvykle připojuje přes zvuk-

kovou kartu, můžete tento modul spustit také z modulu pro nastavení zvukové karty.

## 2.4.5 Zvolte model myši

S tímto modulem YaST můžete nastavit a otestovat připojenou myš.

## 2.4.6 Skener

Pokud máte připojený a zapnutý skener, pak by měl být automaticky rozpoznán při startu tohoto modulu. Jestliže bude rozpoznán, zobrazí se dialog pro konfiguraci skeneru. Pokud nebude rozpoznáno žádné zařízení, pak budete pokračovat v ruční konfiguraci. Jako první krok musíte zvolit typ skeneru, tj. jak je k počítači připojen. Pokud používáte jiný než USB konektor a máte skener připojený k tomuto počítači, tak zvolte 'SCSI skener'.

Jako následující krok bude instalace standardního zařízení. Když bude instalace úspěšná, zobrazí se odpovídající hlášení. Nyní můžete otestovat skener. Vložte do skeneru stránku a klikněte na tlačítko 'Test'.

### Skener nebyl rozpoznán

Automaticky jsou rozpoznány pouze podporované skenery. Skener, který je připojen k jinému počítači v síti nebude rozpoznán. V tom případě nastupuje ruční konfigurace, kdy je třeba určit, zda se jedná o USB, SCSI nebo síťový skener.

**USB skener** zde je třeba uvést výrobce, resp. model skeneru. YaST se pak pokusí nahrát USB moduly. Pokud se jedná o novinku na trhu, může se stát, že modul nebude nahrán automaticky. V tom případě přejděte k dalšímu dialogu, kde budete moci ručně zvolit USB modul. Dále postupujte podle nápovědy v programu YaST.

**SCSI skener** uveďte název zařízení (např. `/dev/sg0`). SCSI skener nesmí být připojován nebo odpojován za běhu systému. Vždy je třeba systém nejdříve vypnout.

**Síťový skener** zadejte IP adresu, resp. název počítače.

Skenery jsou zařízení, která se rychle vyvíjí, proto se tomuto tématu věnujeme také na adrese <http://portal.suse.com/sdb/cz/index.html>, kde v české nebo anglické verzi naleznete aktuální informace a rady pro konfiguraci skeneru. Stačí pouze uvést klíčové slovo *skener*.

Podrobné informace o podporovaných skenerech naleznete také na <http://hardwaredb.suse.de> nebo <http://www.mustang.com/sane>.

## Varování

Při ručním výběru skeneru je třeba být velice opatrný. Výběrem špatného ovladače můžete hardware poškodit.

## Varování

## Řešení problémů

Pokud skener nebyl rozpoznán, pak to může mít následující příčiny:

- Skener není podporován. Konzultujte <http://cdb.suse.de/index.php?LANG=en>, kde je uveden seznam podporovaných skenerů
- Nemáte správně instalován SCSI řadič
- Špatně ukončená SCSI sběrnice terminátorem
- Existují problémy s přerušením u vašeho SCSI řadiče
- SCSI kabel překračuje přípustnou délku
- Skener má SCSI light řadič, který není v Linuxu podporován
- Skener je poškozený

## Varování

U SCSI skenerů nesmí být zařízení v žádném případě připojováno, resp. odpojováno za běhu systému. Nejdrívě je třeba systém vypnout.

## Varování

Další informace o skenování naleznete také v uživatelské příručce, v kapitole věnované programu Kooka.



## 2.4.7 Zvuk

### Konfigurace zvukové karty

YaST se při spuštění modulu pro konfiguraci zvukové karty pokusí automaticky rozpoznat její typ, resp. typy zvukových karet, protože SUSE LINUX podporuje i více zvukových karet v systému. V případě, že máte v systému více zvukových karet, pak nastavte jednu po druhé. Pokud typ vaší karty nebyl nalezen, pak zvolte 'Přidat zvukovou kartu' a přejdete do dialogu 'Manuální výběr zvukové karty', kde můžete vybrat ze seznamu podporovaných karet vaši.

Po výběru karty přejdete do 'Konfigurace zvukové karty'. Když zvolíte 'Rychlé automatické nastavení', pak již nebudete dotazováni a zvuková karta bude okamžitě zkonfigurována. Prostřednictvím 'Normální nastavení' máte možnost upravit v následujícím menu 'Hlasitost' a otestovat nastavení zvukové karty. Při výběru 'Detailnější instalace zvukových karet' přejdete do menu 'Expertní volby pro zvukovou kartu'. Zde můžete ručně upravovat všechny volby pro zvolenou kartu.

### Nastavení hlasitosti zvukové karty

V tomto dialogu můžete otestovat svou konfiguraci zvukové karty. Posuvníkem nastavíte hlasitost. Můžete začít tak na 10%, abyste se náhodou nepřipravili o sluch anebo reproduktory. Stiskem 'Test' pak zazní testovací znělka. Pokud nic neslyšíte, pak zkuste zvýšit hlasitost nebo zkontrolovat zapojení a napájení reproduktorů.

### Konfigurace zvuku

Pokud chcete odstranit konfiguraci, můžete tak učinit tlačítkem 'Odstranit'. Tím budou zakomentovány odpovídající položky v souboru `/etc/modprobe.conf`. Stiskem 'Volby' přejdete do menu **Expertní volby pro zvukovou kartu**. Zde pak můžete upravovat všechny dostupné parametry zvukové karty. Tlačítkem 'Hlasitost' spustíte dialog **Nastavení hlasitosti karty**, kde je možné nastavit hlasitost pro všechny vstupní i výstupní kanály zvukové karty. Pokud YaST nalezne v systému další zvukové karty, zobrazí se v seznamu, případně můžete zvukovou kartu 'Vybrat ze seznamu'.

Když vlastníte Creative Soundblaster Live nebo AWE, můžete volbou 'Instalovat soundfont' zkopírovat zvukové fonty z originálního ovladače (SF2 fonty na CD) na pevný disk. Ty pak budou uloženy do adresáře `/usr/share/sfbank/creative/`.

Pro přehrávání Midi souborů je třeba v dialogu **Konfigurace zvuku** zaškrtnout ‘Spustit sekvencer’. Tak budou nahrány potřebné zvukové moduly pro podporu sekvenceru.

Tlačítkem ‘Konec’ pak uložíte nastavené konfigurace pro jednotlivé karty. Nastavení hlasitosti se zapisuje do souboru `/etc/asound.state`.

### Konfigurovat zvukovou kartu

Pokud je v systému více zvukových karet, pak zvolte z pole ‘Seznam auto-detekovaných’ tu, kterou chcete právě nastavit. Tlačítkem ‘Další’ pak přejdete k dialogu **Konfigurace zvukové karty** (viz výše). Když karta není automaticky nalezena, pak zaškrtněte ‘Vybrat ze seznamu’ a skočíte do dialogu **Manuální výběr zvukové karty**.

### Manuální výběr zvukové karty

Pokud vaše karta není automaticky nalezena, zobrazí se seznam zvukových ovladačů a modelů zvukových karet, kde můžete zvolit odpovídající typ. V položce ‘Vše’ je kompletní přehled podporovaných zvukových karet. V případě potřeby se podívejte do dokumentace ke zvukové kartě, abyste zjistili informace o typu karty. Seznam karet, které ALSA podporuje je uveden na <http://www.alsa-project.org/goemon/>. Stiskem ‘Další’ přejdete do **Konfigurace zvukové karty**.

### Expertní nastavení s možností měnit volby

Zde je možné ručně upravovat všechny dostupné volby pro zvolenou kartu. U některých voleb je k dispozici pole ‘Možná hodnota’, kde jsou uvedeny doporučené hodnoty pro konfiguraci. Tyto přednastavené hodnoty upravujte pouze v případě, že jste si 100% jistí tím, co děláte. Pokud měníte hodnoty jednotlivých voleb, pak máte možnost zapisovat hodnoty v desítkové nebo šestnáctkové soustavě (při hexadecimálním zadávání je třeba psát `0x` před samotným číslem). Po uvedení hodnoty pak stisknete ‘Nastavit’. Stiskem ‘Obnovit vše’ budou **všechny** volby nastaveny na původní hodnotu.

## 2.4.8 TV karta

Po startu a inicializaci modulu YaST se zobrazí dialog **Nastavení TV a rádio karty**. Když je vaše karta rozpoznána automaticky, pak bude zobrazena jako první v seznamu. Klikněte na název TV karty a zvolte ‘Konfigurovat...’.

Ve spodní části dialogu jsou zobrazeny již zkonfigurované TV karty, jejichž parametry můžete upravit tlačítkem **Změnit...**

Pokud se systému nepodaří automaticky rozpoznat TV kartu, pak je třeba její výběr provést ručně. Označte položku 'Jiná (nedetekováno)' a tlačítkem 'Konfigurovat...' přejdete do dialogu **Ruční výběr TV karty**. V dialogu **Ruční výběr TV karty** zvolte nejdříve typ vaší TV karty ze seznamu. V případě potřeby pak můžete také 'Vybrat tuner' tak, abyste získali plnohodnotnou instalaci. Pokud si u výběru tuneru nejste jisti, pak zvolte 'Výchozí (detekováno)'. Když nebude možné naladit některé stanice, pak může být problém v tom, že se nepovedlo automatické rozpoznání typu tuneru nebo jste zvolili špatný typ.

V menu 'Expertní nastavení...' naleznete expertní konfiguraci. Zde můžete přímo zvolit jaderný modul, který bude použit jako ovladač pro vaši tv kartu a nastavit jeho parametry.

V dialogu **Zvuk TV a rádio karty** můžete využít již zkonfigurovanou zvukovou kartu pro zvukový výstup z TV karty. Většinou je spolu s TV kartou dodáván i krátký kabel, kterým můžete propojit zvukovou a TV kartu. Pokud je tato podmínka splněna, pak zvolte 'Ano' a zvolte ze seznamu zkonfigurovaných karet, resp přejděte do 'Nastavení zvukové karty...'. Některé TV karty mají přímo audio výstup, takže můžete připojit reproduktory bez další konfigurace zvukové karty. Existují ale i TV karty, které vůbec nepodporují zvukový výstup. Ty jsou určeny např. pro digitální kamery.

## 2.5 Síťová zařízení

Popis nastavení všech podporovaných typů síťových adaptérů v aplikaci YaST najdete v části 2.5 na této straně. Nastavení bezdrátové sítě je popsáno v kapitole 17 na straně 309.

## 2.6 Síťové služby

Tato záložka je určena pokročilým uživatelům a správcům sítí. Nastavování služeb vyžaduje hlubší znalosti správy systému a síťování. Je třeba si pečlivě prostudovat kapitolu 22 na straně 367 a poté se držte návodů v levé části jednotlivých modulů.

## Varování

Je třeba si uvědomit, že pro pokročilou správu není možné využít bezplatnou instalační podporu. Jsme vám samozřejmě schopni pomoci v rámci našich placených expertních služeb klientům.

## Varování

V této části je probráno pouze základní nastavení služeb. Více detailnějších informací o nastavení systému SUSE LINUX jako síťového serveru, najdete v pozdějších kapitolách této knihy.

### 2.6.1 Agent přenosu pošty (MTA)

V tomto modulu můžete nastavit poštovní služby běžící na vašem systému. Pro odeslání a příjem se používá program postfix nebo sendmail. Poštu lze odesílat i přes SMTP server vašeho ISP. Stahování pošty ze vzdálených účtů a její doručení lokálnímu uživateli pak můžete nastavit pomocí fetchmail.

Můžete také používat poštovní klientský program (např. KMail nebo Evolution) pro přístup k vaší poště pomocí POP3 a odesílání přes SMTP. V tomto případě nemusíte tento modul vůbec nastavovat a stačí když si nastavíte tyto klientské aplikace.

Pokud chcete nastavit poštovní systém, otevřete složku 'Síťové služby' a spusťte modul 'Agent přenosu pošty (MTA)'. Následně si YaST prohlédne váš systém a načte potřebné konfigurační soubory. Pak otevře dialog **Typ připojení**, kde můžete zvolit z následujících možností:

**'Permanentní'** připojení např. pevnou linkou nebo mikrovlnou k Internetu. Připojení k Internetu je trvalé (pokud nespadne) a není třeba se připojovat. Toto nastavení by měli zvolit také uživatelé v lokální síti nepoužívající pevnou linku, ale centrální *poštovní server* pro odesílání pošty

**'Vytáčená linka (modem)'** Toto nastavení asi bude používat většina uživatelů, kteří se připojují z domova bez lokální sítě, tedy pomocí modemu, ADSL, ISDN atd.

**'Žádné připojení'** bude aktivována podpora pro posílání pošty pouze mezi uživateli v rámci tohoto počítače

Další volbou v tomto dialogu je 'Povolit hledání virů (AMaViS)', což je antivirová ochrana. Po jejím zvolení bude automaticky nainstalován antivirový program, který bude kontrolovat příchozí i odchozí poštu. Ačkoliv 99% virů je vytvářeno pro operační systém Windows a základní filozofie Linuxu brání masivnějšímu šíření virů, může se antivirový program hodit v případě, že počítač slouží jako poštovní server a k němu se připojují počítače s Windows. Viry jsou pak odstraňovány již na serveru.

Další dialog bude závislý podle zvoleného typu připojení.

### Permanentní připojení

Zde je možné nastavit 'Server odchozí pošty', který se ale používá hlavně u vytáčených spojení. Zadejte zde SMTP server vašeho poskytovatele připojení. Stiskem 'Maškaráda' přejdete do dialogu **Maškaráda**. Nastavení maškarády se hodí především dvěma skupinám uživatelů. Pokud používáte jako svou doménu např. `mujpocitac.doma`, pak vám poštovní server může odmítnout spojení s tím, že takovou doménu nezná. Toto závisí také do značné míry na možnostech nastavení poštovního klienta, protože třeba KMail je s to provést toto nastavení sám. Druhým případem je ten, kdy se vypisuje i doména nižší úrovně, např. `jan.benda@pocitac03.suse.cz` a je třeba, aby odchozí pošta byla ve formátu `jan.benda@suse.cz`. Pro 'Domény určené k maškarádě' se používá jako oddělovač mezera. Další možností je nastavení **Ověřování**. Zde můžete nastavit přihlašovací údaje, které po vás případně při používání poštovního serveru žádá váš ISP.

Tímto je nastavena 'Odchozí pošta' a můžeme přistoupit k dialogu 'Příchozí pošta'. Pokud provozujete poštovní server, pak zaškrtněte 'Přijmout vzdálená SMTP spojení'. Navíc zde máte možnost nastavit stahování pošty ze vzdálených účtů. Dále můžete přesměrovat příchozí poštu pro superuživatele na jiný účet. Uživatelé jsou pak adresováni nejrozličnější systémové zprávy a hlášení. Další položkou je vyznačené pole 'Stahování'. Zde nastavíme vzdálené účty a v položce 'Protokol' způsob stahování z těchto účtů. Položka 'Alias...' se hodí především pro automaticky vytvářené účty spojené s užíváním určitého programu nebo služby. Tímto způsobem si tedy může správce systému přesměrovat systémovou poštu na svůj nerootovský účet. Zatímco aliasy přesměrovávají poštu podle části uvedené před zavináčem, 'Virtuální domény...' přesměrují poštu podle domény, tj. textu za zavináčem.

### Nastavení vytáčeného spojení

Při nastavování vytáčeného spojení jsou některé volby identické, jako u nastavení pro trvalé připojení. Doporučujeme proto prostudovat i výše uvedenou kapitolu.

V sekci 'Odchozí pošta' je nezbytně nutné zadat 'Server odchozí pošty', kde zadejte buď název vzdáleného serveru (např. `smtp.seznam.cz` nebo jeho IP adresu (v našem případě tedy `212.80.76.43`). Stejně jako u permanentního připojení lze nastavit maškarádu a ověřování, které jsou popsány výše.

Po nastavení odchozí pošty je možné přistoupit k nastavení příchozích zpráv. I zde je třeba uvést server, tentokrát však pro poštu, která vám přichází. Nejčastěji se používá protokol POP3 nebo IMAP, takže název serveru může být např. `pop3.seznam.cz`. Jako protokol je dobré nechat nastavenou hodnotu 'AUTO'. Pouze v případě, že máte problémy se stahováním pošty zde nastavte explicitně používaný protokol. Další položkou je 'Vzdálené uživatelské jméno' a 'Heslo', které budou použity pro přihlašování ke vzdálenému poštovnímu účtu. Když budete chtít povolit přístup přímo ke svému počítači, tj. vytvořit z něj poštovní server, tak zaškrtněte volbu 'Přijmout vzdálená SMTP spojení'. Uvědomte si ale, že v okamžiku, kdy budete mít zaškrtnutu tuto volbu a počítač nebude připojen k síti, budou se e-maily vracet odesílatelům s tím, že příjemce není dostupný. Jako poslední je nastavení 'Přesměrovat poštu uživatele root na' jiný účet. Což se hodí správci systému, který se nechce neustále přihlašovat jako root a kontrolovat příchozí poštu, což jsou většinou systémová hlášení.

## 2.6.2 NFS server a klient

NFS umožňuje nastavení souborového serveru, ke kterému mohou přistupovat všichni uživatelé ve vaší síti. V modulu 'NFS server' můžete počítač nastavit jako NFS server a zvolit adresáře, které se mají exportovat. Tyto exportované adresáře si pak budou moci připojit všichni uživatelé se správnými přístupovými právy. Podrobnější popis modulu a informace o NFS najdete v části 26 na straně 435.

## 2.6.3 NIS server a klient

Správa více systémů s lokálními uživateli (soubory `/etc/passwd` a `/etc/shadow`) je nepraktická a vyžaduje mnoho zásahů správců. Z toho důvodu je velmi výhodné všechna uživatelská data soustředit na jeden centrální server a z něj je distribuovat na jednotlivé klienty. Mimo NIS pro stejný účel můžete využít LDAP nebo Samba. Podrobnější informace o NIS a možnostech nastavení pomocí YaST najdete v části 25 na straně 429.

## 2.6.4 NTP klient

NTP (network time protocol) je protokol pro synchronizaci hardwarových hodin po síti. Podrobnější popis modulu a informace o NTP najdete v části 28 na straně 449.

## 2.6.5 Síťové služby (inetd)

Tento modul slouží pro nastavení přístupu k jednotlivým síťovým službám a je určen pro pokročilé uživatele. Můžete zde nastavit např. `telnet`, `talk`, `ftp` a další, které pak budou spouštěny přímo při startu systému. Když je povolíte -- umožníte vzdáleným uživatelům přístup k těmto službám. Pro každou službu máte také možnost nastavit různé parametry. Standardně je hlavní služba `xinetd`, která spouští ostatní služby, vypnuta.

### Varování

Znovu musíme upozornit, že se jedná o nástroj pro experty! Neprovádějte zde žádné změny, pokud si nejste jisti, co děláte!

### Varování

## 2.6.6 DNS a jméno počítače

Zde nastavíte jméno počítače a DNS. Podrobněji je problematika popsána v části 22 na straně 367 a kapitole 24 na straně 409.

## 2.6.7 Směrování

Směrování síťového provozu je důležitou vlastností Linuxových systémů. V části 22.1 na straně 370 najdete kompletní vysvětlení směrování v Linuxu.

## 2.6.8 Nastavení Samba serveru a klienta

V heterogenních sítích se často vedle sebe nacházejí systémy Linux a Windows. Samba mezi nimi zprostředkovává komunikaci. Informace o Sambě a nastavení serverů a klientů najdete v kapitole 32 na straně 519.

## 2.7 Bezpečnost a uživatelé

Základním rysem Linuxu je jeho víceuživatelské prostředí. Několik uživatelů může najednou nezávisle pracovat na jediném Linuxovém systému. Každý uživatel má svůj uživatelský účet a je identifikován podle jednoznačného přihlašovacího jména -- *login*. Uživatelé mají každý svůj vlastní domácí adresář, kam ukládají osobní data a individuální nastavení aplikací.

### 2.7.1 Správce uživatelů

Po spuštění modulu se otevře dialog ‘Správa uživatelů a jejich skupin’. Práce s tímto modulem je zcela intuitivní. Pomocí zaškrtnutí tlačítek v horní části, můžete zvolit zda chcete upravovat uživatele či skupiny. Pro odstranění uživatele stačí kliknout na uživatele a stisknout ‘Smazat’. Obdobným způsobem se mění nastavení uživatelů. Pokud máte na systému mnoho uživatelů, nebo jste připojeni na NIS server, můžete pomocí ‘Nastavit filtr’ přepínat mezi systémovými a lokálními uživateli. Užitečná je také možnost upravit výchozí nastavení pro nově založené uživatele. To provedeme výběrem ‘Výchozí nastavení pro nové uživatele’ z nabídky ‘Expertní volby...’. Zde můžeme nastavit výchozí příslušnost do skupiny, přihlašovací shell, kde bude domácí adresář, odkud se mají nahrát přednastavené konfigurační soubory atd.

### 2.7.2 Správce skupin

Tento modul vám výrazně usnadní správu skupin. Jedná se o identický dialog jako je ‘Správa uživatelů’, pouze je zde přednastavena ‘Správa skupin’. V okně jsou vypsány stávající skupiny, které můžete mazat nebo editovat, resp. vytvářet nové.

### 2.7.3 Nastavení bezpečnosti

V ‘Nastavení bezpečnosti’, které najdete v nabídce ‘Bezpečnost&uživatelé’, můžete zvolit jednu z následujících možností: úroveň1 pro samostatný počítač (přednastaveno). úroveň2 pro stanici v síti (přednastaveno). úroveň3 pro server v síti (přednastaveno). Pokud chcete jiné nastavení, použijte nabídku ‘Vlastní nastavení’.



V případě přednastavených úrovní jednu zvolte a aktivujte ji kliknutím na 'Dokončit'. V nabídce 'Podrobnosti' lze nastavit jednotlivé hodnoty. V případě volby 'Vlastní nastavení' přejděte do dalšího dialogu stisknutím tlačítka 'Další'.

## Firewall

Firewall slouží pro automatickou ochranu počítače před útoky z Internetu, resp. ostatní počítače nemohou navázat spojení s vaším počítačem. Zároveň je však povoleno navazování spojení z vašeho počítače k jiným stanicím. Není přitom třeba upravovat konfigurační soubory, vše je již připraveno. Musíte nastavit typ síťového rozhraní, tj. zda se připojujete prostřednictvím modemu, síťové karty nebo třeba ISDN. Tomu pak odpovídá ppp0, eth0 a ippp0. Pokud nebudete spokojeni s nastavením pomocí následujících dialogů, můžete nastavení ručně upravit v souboru `/etc/sysconfig/SuSEfirewall2`. YaST totiž ukládá nastavení firewallu do tohoto souboru, a odtud bere data pro nastavení samotného firewallu. Vaše ruční změny se tedy neztratí.

### Tip

#### Automatická aktivace firewallu

YaST automaticky spustí firewall s nastavením, které je přijatelné pro většinu sítí a počítačů. Modul nastavení firewallu pak nutně spouštět pouze v případě, že byste chtěli změnit nastavení nebo ho vypnout.

### Tip

## 2.8 Systém

### 2.8.1 Záloha systému

S pomocí tohoto modulu můžete vytvořit zálohu systému. Standardně se nevytváří záloha celého disku, ale pouze konfiguračních souborů, kritických oblastí disku a změn v instalovaných balíčcích. YaST prohledá systém a vytvoří zálohu souborů, které se změnily od posledního zálohování, nebo od nainstalování systému. Může uložit také tabulku rozdělení disků nebo MBR.

### 2.8.2 Obnova systému

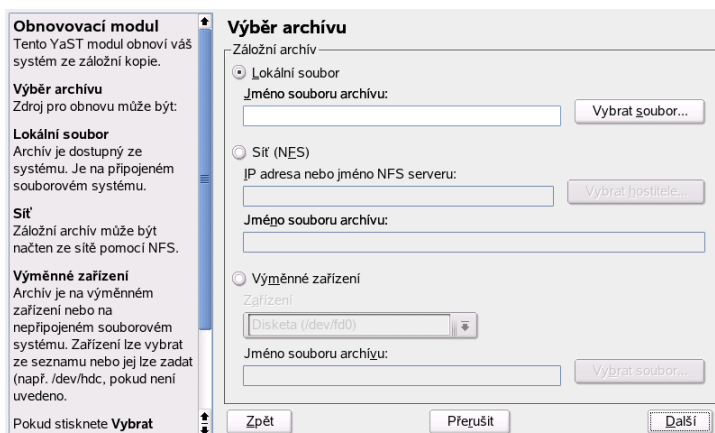
Při obnově systému ze zálohy se řiďte instrukcemi v nápovědě. Nejprve vyberte odkud se bude obnova provádět (pevný disk, cdrom...) a následně určete co se

bude obnovovat. Poté se objeví dva dialogy. Jeden pro odinstalování balíčků, které se do systému instalovaly od poslední zálohy. Druhý nainstaluje balíčky, které byly odinstalovány. Tyto úpravy by měly zaručit, že systém bude přesně v tom stavu, v jakém byl v průběhu vytvoření zálohy.

## Varování

Protože tento modul instaluje, maže a přepisuje mnoho souborů a balíčků, používejte ho pouze pokud již máte zkušenosti se zálohováním. Jinak můžete ztratit některá data.

## Varování

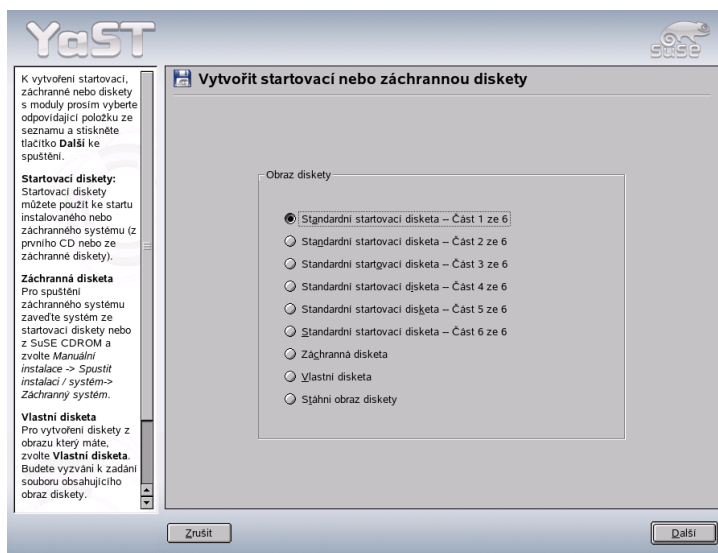


Obrázek 2.5: Úvodní okno modulu obnovy systému

### 2.8.3 Vytvořit systémovou disketu

Modul vytvoří různé zaváděcí diskety, které lze použít v případě potíží. Jednotlivé diskety se používají k následujícímu:

**‘Startovací disketa’** Tato nabídka slouží pro spuštění operačního systému (který je nainstalován na disku) nebo záchranného systému.



Obrázek 2.6: Vytvoření systémové diskety

**‘Záchranná disketa’** Disketa vytvořená pomocí této volby obsahuje záchranný systém, tj. speciální prostředí pro údržbu systému (jádro, základní systém a nástroje). Pokud tedy není možné spustit nainstalovaný systém ani prostřednictvím startovací diskety, pak se velice hodí.

Abyste se dostali do záchranného systému, zaveďte systém z běžné startovací diskety a zvolte ‘Manual Installation’, ‘Start Installation/System’, and ‘Rescue System’. Budete dotázáni na *rescue disk*. Jestliže váš systém využívá speciální zařízení (RAID, USB...) budete nejspíš potřebovat i diskety s moduly.

**‘Diskety s moduly’** Tato volba se hodí, pokud provádíte instalaci z médií umístěných někde v síti, nebo někde, kde není možné instalovat systém z prvního nebo druhého CD (máte starší typ CD mechaniky, SCSI mechaniku...). Jednotlivé diskety s ovladači obsahují moduly pro disky, řadiče, PCMCIA karty, starší CD jednotky a ovladače pro síťové karty.

**‘Vlastní disketa’** Tuto volbu použijte, pokud chcete na disketu zapsat existující obraz uložený na disku vašeho počítače.

**‘Stáhni obraz diskety’** Zde můžete zadat URL obrazu diskety a po zadání ověřovací dat jej stáhnout z Internetu.

Po zvolení typu vytvářené diskety a stisku ‘Další’ budete vyzváni ke vložení naformátované diskety do mechaniky. Následně pak bude vytvořena požadovaná disketa.

#### 2.8.4 Výběr časové zóny

Časovou zónu vybíráte většinou již při instalaci. Pokud jste se ale mezitím dostali do jiného časového pásma, např. používáte notebook, můžete průběžně upravovat časová pásma. Většinou stačí zvolit ze seznamu zemi, nebo přímo definovat časové pásmo podle GMT.

##### Důležité

Při driftování na ledové kře nezapomeňte kontrolovat nastavené časové pásmo.

##### Důležité

Linuxové počítače používají většinou nastavení systémového (hardwarového) času podle ‘GMT’, tj. *Greenwich Mean Time*, a při zobrazování k němu přičítají, nebo odečítají posuv časového pásma. Naproti tomu jiné operační systémy, např. Windows, dávají přednost hardwarovému nastavení hodin na místní čas.

#### 2.8.5 Výběr jazyka

Zde můžete nastavit, v jakém jazyku s vámi bude Linux komunikovat. Tato změna jazyka se projeví v celém systému, tedy i v KDE a konfiguračním nástroji YaST.

#### 2.8.6 Výběr rozložení klávesnice

### Důležité

V tomto modulu nastavíte klávesnici pouze pro textové prostředí. Jestliže používáte grafické rozhraní, nastavte rozložení klávesnice v modulu 'Grafická karta a monitor' v záložce 'Hardware'.

### Důležité

Po spuštění modulu se otevře dialog **Základní nastavení**. Standardně je nastavená klávesnice podle zvoleného jazyka. Pokud zvolíte rozložení kláves 'České', pak budete mít klasickou *qwertz* klávesnici, která je také přednastavena. *Qwerty* klávesnici využijí hlavně technicky zaměření uživatelé a programátoři. V poli 'Test klávesnice' můžete ihned vyzkoušet novou klávesovou mapu.

## 2.8.7 Editor úrovní běhu

V Linuxu se používají *úrovně běhu* *runlevel* pro odlišení různých stavů počítače. Existuje *runlevel*, kdy je spuštěn víceuživatelský režim. Na jiné úrovni jsou spuštěny i síťové služby a v další pak grafické prostředí. Pokud zlobí třeba síťové služby a není možná oprava za běhu, stačí pouze přejít na jiný, resp. nižší *runlevel*. Podrobné informace a technické pozadí najdete v části 7.3 na straně 151.

Po spuštění modulu se otevře okno **Editor úrovní běhu: výchozí úroveň běhu**. Standardně se zobrazí pouze 'Jednoduchý režim', kde můžete zvolit, která služba bude povole a která ne. Přepnete-li na 'Expertní režim', lze zvolit *runlevel*, do kterého bude počítač startovat. Přednastavená je úroveň běhu 5, tj. 'Plný víceuživatelský režim se sítí a xdm'. *xdm* je program pro grafické přihlášení. Začínající uživatelé by měli ponechat tuto úroveň běhu.

### Důležité

Nesprávným nastavením úrovní běhů můžete váš systém dostat do stavu, kdy bude nepoužitelný. Předtím než provedete změny, dobře uvažte, co děláte.

### Důležité

## 2.8.8 Editor souborů /etc/sysconfig

V distribuci SUSE LINUX je hlavním konfiguračním adresářem */etc/sysconfig*, kde se nastavují nejdůležitější parametry, které mají vliv na chování celého systému. Modul 'Editor souborů */etc/sysconfig*' pak slouží běžným uživatelům, kteří by chtěli upravit chování systému v pěkném grafickém prostředí.

Po spuštění modulu se zobrazí dialog, kde jsou tématicky řazeny proměnné k různým položkám. Tento modul je určen pokročilým uživatelům a správcům sítě, resp. systému.

### Varování

Neměňte hodnoty, pokud nevíte zcela přesně, co děláte. Mohli byste vážně poškodit váš systém.

### Varování

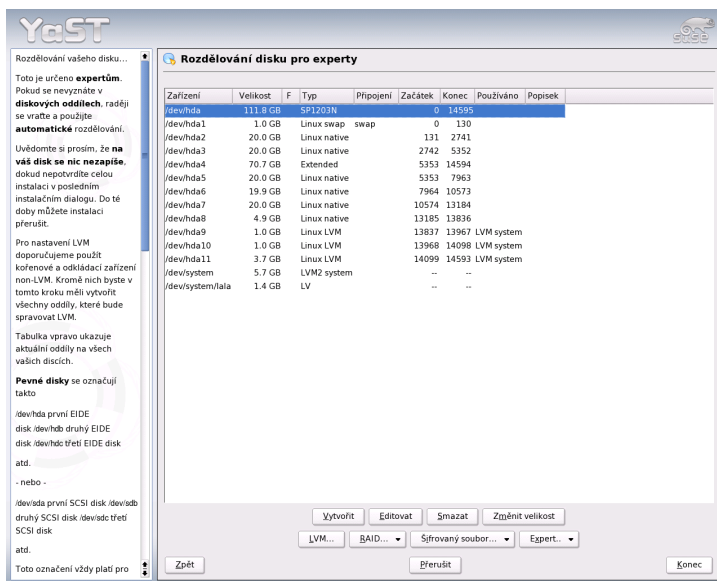
## 2.8.9 Dělení disku

Pomocí expertního dialogu (viz 2.7 na této straně) můžete ručně upravit oddíly na jednom nebo více pevných discích. Diskové oddíly lze přidávat, mazat a upravovat. Můžete také přejít do nastavení softwarového RAIDu a LVM.

### Varování

Ačkoliv je možné měnit oddíly v nainstalovaném systému, měli by to dělat pouze odborníci. V případě chyby hrozí ztráta dat. Pokud měníte oddíly na právě používaném disku, ihned po provedení změn systém restartujte. Bezpečnější než měnit oddíly za provozu je použití záchranného systému.

### Varování



Obrázek 2.7: Expertní režim dělení disku programu YaST

V seznamu expertního dialogu jsou zobrazeny všechny existující nebo navržené diskové oddíly na všech připojených pevných discích. Celý disk je označen jako

zařízení bez čísla např. `/dev/hda` nebo `/dev/sda`. Jednotlivé diskové oddíly jsou uvedeny jako části tohoto zařízení např. `/dev/hda1` nebo `/dev/sda1`. V seznamu jsou uvedeny také informace o velikosti, typu, souborovém systému a bodu připojení jednotlivých oddílů disku. Bod připojení říká, v jakém adresáři bude diskový oddíl přístupný v linuxovém systému.

Pokud spustíte expertní dialog během instalace, jsou uvedeny a zvoleny také volné diskové prostory. Volné místo pro Linux získáte uvolňováním jednotlivých diskových oddílů odspodu seznamu (od posledního oddílu k prvnímu). Například pokud máte tři diskové oddíly, nelze použít prostřední pro Linux a první a třetí ponechat volný pro jiné operační systémy.

### Vytváření diskových oddílů

Klikněte na tlačítko 'Vytvořit'. V případě, že v systému máte více disků, program se zeptá na cílový disk. Pak zadejte typ diskového oddílu (primární nebo rozšířený). Vytvořit můžete buď čtyři primární oddíly nebo tři primární oddíly a jeden rozšířený. Na rozšířeném diskovém oddíle můžete vytvářet další oddíly (viz 1.5.4 na straně 11).

Zvolte souborový systém a bod připojení. YaST vám pro každý vytvořený oddíl bod připojení nabídne. Podrobnější informace o jednotlivých parametrech diskového oddílu najdete v následující části. Změny aplikujete tlačítkem 'OK'. Nyní máte v tabulce zobrazen nově vytvořený oddíl. Kliknutím na 'Konec' budou změny přijaty a vy se vrátíte na stránku návrhu.

### Parametry diskových oddílů

Při vytváření nebo úpravě diskového oddílu můžete nastavit řadu různých parametrů. U nově vytvářených oddílů většinu parametrů nastaví YaST. Toto nastavení obvykle nepotřebuje žádné úpravy. Pokud chcete provést ruční nastavení, postupujte následujícím způsobem:

1. Zvolte diskový oddíl.
2. Stiskněte tlačítko 'Upravit' a v následujícím dialogu nastavte parametry:

**ID soub. systému** I v případě, že diskový oddíl nebude nyní formátovat, nezapomeňte mu přiřadit ID. Jen tak zajistíte, že bude vždy správně rozpoznán. Možné hodnoty jsou 'Linux', 'Linux swap', 'Linux LVM' a 'Linux RAID'. Podrobnější informace o LVM a RAIDu najdete v částech 3.7 na straně 95 a 3.8 na straně 100.



**Soub. systém** Chcete-li hned při instalaci diskový oddíl naformátovat, zvolte některý z dostupných systémů souborů: 'Ext2', 'Ext3', 'JFS', 'Reiser', 'XFS' nebo 'Swap'. Informace o souborových systémech najdete v kapitole 20 na straně 345.

Swap je zvláštní formát, který umožňuje diskový oddíl používat jako virtuální paměť. Každý systém by měl mít alespoň jeden oddíl swap o minimální velikosti 128MB. Jako výchozí souborový systém je nastaven ReiserFS. ReiserFS, JFS a Ext3 jsou žurnálovací souborové systémy. Jsou schopné se rychle vzpamatovat po pádu souborového systému, protože všechny operace jsou průběžně zaznamenávány v žurnálu. Navíc je ReiserFS velmi rychlý při práci s velkým množstvím malých souborů. Ext2 mezi žurnálovací souborové systémy nepatří, ale je stabilní a vhodný pro velmi malé diskové oddíly, protože nevyžaduje příliš mnoho diskového prostoru pro vlastní správu.

**Volby** Zde můžete nastavit volby zvoleného souborového systému. Dostupné volby jsou závislé na zvoleném souborovém systému.

**Krypt. souborový systém** Pokud tuto možnost zatrhnete, budou všechna data na zvoleném diskovém oddílu šifrovaná. Touto volbou můžete zvýšit bezpečnost svých dat, ale zároveň mírně zpomalíte rychlost systému. Více informací o šifrování souborového systému najdete v části 34.3 na straně 564.

**Volby fstab** Zde můžete zadat volby pro administrační soubor systémů souborů (/etc/fstab).

**Bod připojení** Zadejte adresář, do kterého se oddíl má připojovat. Můžete si vybrat některou z nabídek programu YaST nebo zadat vlastní adresář.

3. Oddíl aktivujete kliknutím na tlačítko 'OK'.

Pokud provádíte rozdělování manuálně, vytvořte odkládací (swap) oddíl o velikosti nejméně 256MB. Odkládací oddíl ulehčuje paměti od momentálně nepotřebných dat. Je tak uvolněna pro často používaná důležitá data.

## Expertní volby

'Expert' otevře nabídku s následujícími příkazy:

**Znovu načíst tabulku oddílů** Znovu načte tabulku oddílů. To je potřeba například po manuální úpravě oddílů v textové konzoli.

## Smazat tabulku oddílů a popis disků

Zcela přepíše starou tabulku oddílů. Může to pomoci například při problémech s nekonvenčními popisky disku. Tento příkaz vede ke ztrátě všech dat na disku.

## Další tipy pro dělení disků

Pokud dělení disku provádíte pomocí programu YaST a v systému se nachází další diskové oddíly, jsou tyto oddíly pro snadnější přístup také zaneseny do souboru `/etc/fstab`. Tento soubor obsahuje údaje o všech diskových oddílech a jejich vlastnostech (např. souborový systém, bod připojení a přístupová práva).

### *Příklad 2.1: Diskové oddíly v souboru fstab*

```
/dev/sda1    /data1      auto        noauto,user 0 0
/dev/sda5    /data2      auto        noauto,user 0 0
/dev/sda6    /data3      auto        noauto,user 0 0
```

Diskové oddíly mají bez ohledu na souborový systém nastavenou volbu `noauto` a `user`. Tím je umožněno, že si je může připojit každý uživatel systému. Z bezpečnostních důvodů YaST nepřidává volbu `exec`, která povoluje spouštění programů přímo ze zvoleného diskového oddílu. Pokud tuto volbu potřebujete, zadejte ji ručně. Ruční dodání této volby je nutné především v případě, že systém hlásí zprávy jako *bad interpreter* nebo *Permission denied*.

## Dělení disku a LVM

V expertním dělení můžete provést LVM konfiguraci kliknutím na 'LVM...' (viz 3.7 na straně 95). Pokud však již máte na počítači nastavenou funkční LVM konfiguraci, aktivuje se automaticky při prvním vstupu do LVM konfigurace během sezení. V takovém případě nelze měnit oddíly na žádném disku obsahujícím oddíl patřící do aktivní skupiny svazků, protože linuxové jádro nemůže znovu načíst tabulku oddílů z používaného disku. Pokud však máte funkční LVM konfiguraci, není nejspíš vůbec nutné fyzické oddíly měnit. Místo toho změňte logické svazky.

Na začátku fyzických svazků jsou na oddíl zapsány informace o svazku. Tak fyzický svazek *ví*, ke které skupině svazků patří. Pokud chcete oddíl zpřístupnit k jiným účelům než LVM, smažte tento začátek svazku. V případě skupiny `system` a fyzického svazku `/dev/sda2` to provedete příkazem `dd if=/dev/zero of=/dev/sda2 bs=512 count=1`.

## Varování

### Startovací systém souborů

Systém souborů používaný pro start systému (`/boot`) nesmí být uložen na logickém svazku LVM. Uložte ho na běžný fyzický oddíl.

## Varování

## 2.8.10 Správce profilů

Jsou situace, kde je nezbytné změnit systémovou konfiguraci. Pokud často provozujete svůj počítač v prostředích, kde potřebujete různá nastavení systému, možná by se vám hodilo uložit si tato nastavení a obnovit je později, kdykoliv je to potřeba. Toto je typická situace například pro uživatele notebooků, kteří pracují na různých místech. Také si lze představit stolní počítač, který chcete dočasně provozovat s jinou konfigurací. V takových případech byste rádi měli záložní mechanismus, který uloží současná systémová konfigurační data a uloží je do profilu. Tímto způsobem lze potom kdykoliv tuto konfiguraci obnovit.

SCPM (System Configuration Profile Management) je systém, který spravuje takovéto profily systémové konfigurace v Linuxu. Následující příklad je zamýšlen jako krátký přehled toho, k čemu se dá SCPM použít.

Předpokládejme, že máte notebook a chcete jej připojit ke své domácí i firemní síti a používat jej nezávisle, když jste na cestách. Toto obvykle vyžaduje nakonfigurovat systém tak, aby zapadl do různých sítí. Například potřebujete DHCP klienta v kanceláři a pevnou IP adresu doma. Dále máte třeba v kanceláři spuštěné služby jako `xntpd` nebo NIS klienta, ale doma pouze automounter, ale žádná z těchto služeb není potřeba pokud cestujete. Pro tyto případy vám SCPM pomůže zvládnout rozdílné konfigurace a jednoduše se mezi nimi přepínat.

SCPM toho ale umí daleko víc. Je velmi konfigurovatelný; zvládne skoro všechny možné scénáře, kdy je potřeba uložit a obnovit data v různých verzích. Dokonce jej lze použít pro spouštění skriptů v závislosti na profilech, mezi kterými je přepínáno. Více informací najdete v příslušných info stránkách.

### Omezení SCPM

SCPM je zamýšleno ke spravování systémových konfiguračních profilů. Není určeno pro správu uživatelských profilů, jako např. různá nastavení pracovního prostředí KDE.

## 2.8.11 Rozdělování disku

Historicky obsahuje každý disk tabulku oddílů (partition table) se čtyřmi řádky, z nichž každý ukazuje buď na primární oddíl, nebo na rozšířený oddíl, nebo na nic. V této tabulce (nikoli na celém disku) však smí být jen *jeden* řádek s rozšířeným oddílem.

*Primární oddíl* je souvislá sekvence cylindrů, přiřazená některému operačnímu systému. Kdyby se používaly pouze primární oddíly, dal by se disk rozdělit maximálně na čtyři oddíly -- víc by se do tabulky nevešlo.

Proto se později přešlo na koncepci *rozšířených oddílů*. Rozšířený oddíl je rovněž souvislou posloupností cylindrů, dá se však dále rozdělit na tzv. *logické oddíly*, které již nepotřebují žádnou další položku v tabulce diskových oddílů. Rozšířený diskový oddíl je tedy jakýsi obal na logické oddíly.

Potřebujete-li více než čtyři oddíly, musí být některý oddíl rozšířený a přidělíte mu celý zbytek diskového prostoru. V rozšířeném oddílu můžete vytvořit až 15 logických oddílů na SCSI disku a 63 logických oddílů na (E)IDE disku.

Linux zachází se všemi primárními či logickými oddíly rovnocenně, a může být instalován na kterýkoli z nich.

### Důležité

Jestliže měníte nastavení diskových oddílů, mě-li byste velice dobře vědět co děláte. Neodborná manipulace může způsobit ztrátu veškerých dat uložených na discích!

### Důležité

Pokud chcete upravovat velikosti diskových oddílů, doporučujeme, abyste měli alespoň základní znalosti o připojování unixových souborových systémů, vědět co je *bod připojení*, a také pečlivě rozlišovat primární, rozšířený a logické diskové oddíly.

Navíc je dobré si uvědomit, že neexistuje *jediná* zlatá cesta pro všechny -- optimální volba bude vždy silně individuální.

Nejprve je však nutno shromáždit základní údaje o vašem systému:

- Jakým způsobem budete počítač používat (např. jako souborový server, aplikační server, výpočetní server, pracovní stanice)?
- Kolik lidí na něm bude pracovat (současně přihlášených)?
- Kolik disků máte, jak jsou velké a jak jsou připojeny (přes EIDE, SCSI či jako RAID)?

## Velikost odkládacího oddílu

Často se dočtete, že by odkládací oddíl *swap* měl být zhruba dvakrát větší než velikost instalované paměti. Je to pozůstatek z dob, kdy 8 MB bylo považováno za hodně paměti.

I když mají nové aplikace větší a větší požadavky na paměť, obvykle by mělo stačit 128 MB virtuální paměti swap. Pokud však máte spuštěné KDE, netscapea emacs, a kompilujete jádro, moc volné paměti vám nezůstane. V současné době je pro běžného uživatele rozumné nastavit virtuální paměť na 256 MB.

Vždy byste měli mít nastaven odkládací oddíl a to i v případě, že máte v počítači více než 256 MB RAM. V tomto případě však pro nejnutnější práci obvykle stačí 64 MB swap oddíl. Při dnešních velikostech disku není vytvoření takového swapu žádný problém.

## Optimalizace

Omezujícím faktorem bývají většinou disky. K překonání tohoto úzkého hrdla můžete použít následující možnosti, které lze kombinovat:

- Rozdělte zatížení na více disků.
- Použijete optimalizovaný souborový systém, např. *ReiserFS*.
- Vybavte počítač větší paměti (min. 256 MB u souborového serveru).
- Nastavte u IDE disků DMA režim (viz 2.4.3 na straně 46).

## Paralelní využití více disků

K vysvětlení je potřeba si uvědomit, že celková doba pro přenos dat se skládá z následujících součástí:

1. doba, než požadavek na čtení či zápis dosáhne řadiče
2. doba, než řadič odešle požadavek disku
3. doba, než disk nastaví hlavu
4. doba, než se médium nastaví na hledaný sektor
5. doba pro vlastní přenos dat

První zpoždění je závislé na připojení sítě a je třeba jej řešit samostatně. Druhé zpoždění bývá zanedbatelné a záleží pouze na kvalitě řadiče. Třetí zpoždění je kritické a udává se v milisekundách. V porovnání s nanosekundovým přístupem k RAM se jedná o rozdíl až šest řádů. Čtvrté zpoždění závisí na otáčkách disku. Páté závisí kromě otáček disku ještě na počtu hlav a pozici dat na médiu (blíže ke středu či dále od něj).

Zlepšit se dá výkon u třetí položky. Zde mají výhodu SCSI řadiče a jejich inteligentní funkce `disconnect`. Ta způsobí při více diskových mechanikách na jednom SCSI řadiči, že ty disky, které v daném okamžiku nastavují hlavu a nepřenášejí data, se dočasně odpojí od sběrnice SCSI (pokud to umí). Sběrnice se tím uvolní pro ostatní disky, které mezitím data přenášejí. Jakmile se ukončí přenos nebo sníží zatížení (podle politiky řadiče) odpojený disk se zase připojí a je již připraven přenášet data.

Ve víceúlohovém, víceuživatelském operačním systému, jako je Linux, toho lze optimalizovat mnoho. Zkusíme například paralelizovat přístup k diskovým oddílům. Podívejme se na výpis z příkazu `df`.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda5	1.8G	1.6G	201M	89%	/
/dev/sda1	23M	3.9M	17M	18%	/boot
/dev/sdb1	2.9G	2.1G	677M	76%	/usr
/dev/sdc1	1.9G	958M	941M	51%	/usr/lib
shmfs	185M	0	184M	0%	/dev/shm

Co nám zde může přinést paralelizování? Dejme tomu, že uživatel `root` spustí v adresáři `/usr/src` příkaz:

```
tar xzf balik.tar.gz -C /usr/lib
```

Smyslem příkazu je rozbalit archiv `balik.tar.gz` do adresáře `/usr/lib/balik`. Na to zavolá příkazový interpret programy `tar` a `gzip`, které se nacházejí v adresáři `/bin` a tím i na prvním disku `/dev/sda`. Dále se bude číst soubor `balik.tar.gz` z adresáře `/usr/src` na druhém disku `/dev/sdb`. Jako poslední se budou extrahovat data a zapisovat do `/usr/lib` na třetím disku `/dev/sdc`.

Tím se rozdělí nastavování hlav, čtení z diskového bufferu a zápis do něj na tři nezávislá média a může být podle možnosti prováděno současně.

To je pouze jeden příklad z mnoha. Pro běžné systémy, jako je ten z uvedeného příkladu, platí pravidlo, že máme-li dva rovnocenné disky, rozdělíme mezi ně `/usr` a `/usr/lib`. Přitom by adresář `/usr/lib` měl mít rozsah zhruba 70% rozsahu `/usr`. Kořenový adresář `/` by se měl vzhledem k přístupu na něj při rozdělení na dva disky nacházet na stejném disku jako `/usr/lib`.

Od určitého počtu SCSI disků (4 až 5) bychom již měli pomýšlet na řešení pomocí softwarového diskového pole (RAID) nebo si raději přímo pořídit řadič RAID. Pak nám již nepoběží diskové operace kvaziparalelně, ale skutečně paralelně. Navíc v případě RAID5 jako vedlejší efekt dostaneme možnost úplné záchrany dat v případě výpadku některého z disků.

### Přístup k disku a velikost paměti

Již jsme uváděli, že pod Linuxem je velikost paměti důležitější než rychlost procesoru. Důvodem je schopnost Linuxu dynamicky vytvářet buffery pro disková data. Zde používá Linux různé triky jako *dopředné čtení* (předem si načítá sektory) a *opožděný zápis* (šetří si zápisy a provede je pak najednou). Opožděné zápisy jsou také důvodem, proč se nedá Linux bez řádného ukončení práce vypnout. Jak dopředné čtení, tak i opožděný zápis přispívají k tomu, že hlavní paměť neustále vypadá, jako by byla plně obsazena. Výsledkem je však výrazně vyšší rychlost Linuxového systému.

	total	used	free	shared	buffers	cached
Mem:	255	246	9	0	23	44
-/+ buffers/cache:		178	76			
Swap:	261	3	257			

Jak ukazuje výstup výše, přibližně 23 MB se právě nachází v bufferech. Cokoli se dá najít v bufferech, to je okamžitě dostupné pro nové čtení.

### Rozdělování diskových oddílů v modulu YaST

Pomocí modulu YaST pro konfiguraci diskových oddílů můžete existující diskové oddíly vytvářet, mazat, měnit velikost nebo upravovat. Můžete odsud také přejít do modulů pro práci s LVM a softwarovým RAIDem. Těmto modulům jsou věnovány samostatné sekce: 3.8 na straně 100 a 3.7 na straně 95.

V běžném případě jsou diskové oddíly vytvářeny během instalace. Pokud ale chcete, např. kvůli nedostatku místa, integrovat i druhý pevný disk, pak máte možnost ho přidat i ke stávajícímu linuxovému systému. Nejdříve je třeba tento disk rozdělit na jednotlivé diskové oddíly a vytvořit zde souborové systémy. Následně je možné diskové oddíly připojit a uvést v souboru `/etc/fstab`. Případně je ještě třeba překopírovat některá data, např. pokud chcete přemístit starý `/opt` diskový oddíl na nový pevný disk.

Pokud chcete provádět psí kusy s pevným diskem, se kterým právě pracujete, např. měnit množství nebo velikost jednotlivých diskových oddílů, je to v zásadě možné, ale je třeba být opatrný a po provedení změn restartovat počítač. Prozíravější je spustit systém z instalačních CD a následně provést změny na disku.

Ovládání je intuitivní a jednotlivé volby jsou podrobně vysvětleny v nápovědě na levé straně okna.

## 2.8.12 Konfigurace zavaděče

Tento modul velice zjednodušuje nastavení zavaděče systému. I tak byste ale neměli měnit konfiguraci tohoto programu bez znalosti celého konceptu zavádění systému. Přečtěte si proto nejdříve kapitolu 8 na straně 163.

Způsob startování počítače se vybírá většinou během instalace. Pokud tedy váš SUSE LINUX startuje tak jak má, není třeba zde nic měnit. Pokud jste ale dosud spouštěli systém z diskety a nyní chcete startovat z pevného disku, pak spusťte modul 'Konfigurace zavaděče'.

### Varování

V rámci přiblížení se ke standardům byl nahrazen zavaděč LILO za GRUB. Samozřejmě je LILO i nadále součástí SUSE LINUXu, takže je možné jej nainstalovat a používat.

### Varování

Po startu počítače je třeba spustit operační systém. Spuštění operačního systému má v systému SUSE LINUX na starost program GRUB. Po zapnutí se počítač aktivuje a zkontroluje hardware a spustí zavaděč. Zde si zvolíte, který operační systém chcete spustit a zavaděč se pak již postará o jeho spuštění.

### Důležité

Pokud máte nainstalováno více operačních systémů, můžete použít zavaděč z Linuxu i pro spuštění těchto systémů.

### Důležité

Po spuštění modulu 'Konfigurace zavaděče' se zobrazí dialog, kde bude zobrazena současná konfigurace. Můžete zde uložit nebo změnit konfiguraci zavaděče, resp. obnovit původní konfiguraci.

Jestliže chcete ke konfiguraci přidat některou volbu, klikněte na 'Přidat' a z nabídky vyberte požadovaný parametr a zadejte jeho hodnotu. Kliknutím na některou z položek a následně na tlačítko 'Upravit', lze změnit hodnotu parametru.



Podobným způsobem také můžete některé volby zcela odstranit. V pravé části je tlačítko 'Obnovit'. Jestliže na něj kliknete, zobrazí se seznam voleb. Ty mají tento význam:

**Navrhnout novou konfiguraci** SUSE LINUX vygeneruje nový návrh na konfiguraci zavaděče. Jestliže na dalších oddílech nalezne jiné operační systémy, umístí je do menu zavaděče. Pokud máte nainstalovánu i jinou nebo starší verzi Linuxu, lze zavádět tento Linux buď přímo, nebo spustit jeho zavaděč.

**Začít od nuly** Vytvoříte celou konfiguraci zavaděče sám.

**Znovu načíst konfiguraci z disku** Existující nastavení se opět načte.

**Navrhnout a sloučit s existujícími menu GRUB**

Pokud je na jiném oddíle nainstalován Linux, zahrne se jeho nabídka do vytvářeného menu. Tato volba není dostupná pokud používáte LILO.

Kliknutím na 'Upravit konfigurační soubory' můžete upravit nastavení přímo v konfiguračních souborech. Tvorba a modifikace těchto konfiguračních souborů je podrobně vysvětlena v kapitole 8 na straně 163.

## Konfigurační volby pro zavaděč

Pro začínající uživatele je určitě jednodušší nastavit proces zavádění z tohoto modulu. Stačí vybrat parametr myši, kliknout na 'Upravit', zadat hodnotu parametru a potvrdit změnu tlačítkem 'OK'. Jednotlivé volby se mohou u různých zavaděčů lišit. Následující sekce vysvětluje nejdůležitější parametry programu GRUB, který je standardním zavaděčem systému SUSE LINUX.

**Typ zavaděče** Zde můžete přepínat mezi programem GRUB a LILO. V následujícím dialogu pak zvolíte, jak se má změna provést. Lze převést konfiguraci GRUBu na konfiguraci pro LILO, ale tak se mohou ztratit některé volby, které v druhém programu neexistují. Můžete také vytvořit zcela novou konfiguraci.

**Umístění zavaděče** V této položce nastavíte, kam se má zavaděč uložit. Do MBR, zavaděcího sektoru zavaděcího oddílu, zavaděcího sektoru kořenového oddílu nebo na disketu. Zvolíte-li 'Ostatní' můžete zavaděč uložit na libovolné místo.

**Pořadí disků** Jestliže máte více disků, nastavte jejich pořadí podle BIOSu.

**Výchozí sekce** Standardně se, po uplynutí časové prodlevy, zavede ten operační systém, který je uveden v tomto políčku.

**Dostupné sekce** Zde musí být uvedené ty sekce, které má zavaděč zobrazit při startu.

**Aktivovat oddíl zavaděče** Nastaví ten oddíl, kam se ukládá zavaděč, jako aktivní.

**Nahradit kód v MBR** Pokud měníte umístění zavaděče, zvažte také, zda chcete přepsat MBR.

Z dalších voleb pak stojí za povšimnutí položka 'timeout', která v sekundách určuje, jak dlouho se má čekat na vstup od uživatele, než se zavede výchozí systém.

---

### Varování

Instalaci a konfiguraci zavaděče GRUB a LILO je věnována celá kapitola, kde jsou podrobně vysvětleny jednotlivé položky konfiguračního souboru a celé technické pozadí konfigurace. Viz 8 na straně 163.

---

Varování

## 2.9 Různé

V této části najdete moduly, které nebylo možné zařadit jinam. Pozornost věnujte především modulu 'Dotaz na podporu'.

### 2.9.1 Dotaz na podporu

Nákupem systému SUSE LINUX a jeho registrací získáváte nárok na bezplatnou instalační podporu. Bližší informace o kontaktních telefonních číslech, adrese a e-mailové adrese naleznete v příloze této příručky. Prostřednictvím tohoto modulu budete mít ulehčenu práci při vytváření požadavku pro instalační podporu a požadavek bude automaticky zaslán elektronickou poštou. Je však potřeba, abyste měl k dispozici registrační kód, který získáte registrací produktu. Registraci můžete provést na adrese <http://portal.suse.com/>. Při posílání dotazu je dobré zvolit 'Odeslat informace o hardwaru' i 'Odeslat informace o softwaru', protože tyto údaje mohou instalační podpoře výrazně pomoci a tím urychlit vyřešení vašich problémů.

**Tip**

Pokud vaše požadavky přesahují rámec instalační podpory, můžete se obrátit na oddělení služeb zákazníkům společnosti SUSE, kde vám rádi poskytneme placené expertní služby. Více informací získáte na webové stránce <http://www.suse.cz/cz/services/>.

**Tip**

## 2.9.2 Zobrazit startovací protokol (log)

Při startu systému se na obrazovku vypisují různá systémová hlášení. Začínajícímu uživateli signalizují především to, že počítač opravdu něco vykonává, ale později zjistíte, že obsahují množství zajímavých informací, které mohou být životně důležité v případě, že se objeví nějaká chyba v systému. Abyste se mohli případně později k těmto informacím vrátit a nahlédnout do výpisu při startu systému, stačí vám pouze spustit tento modul. Správci systému a příznivci textového režimu pak mohou nahlédnout přímo do protokolového souboru `/var/log/boot.msg`, kde jsou tyto informace uloženy, a odkud modul informace načítá.

## 2.9.3 Zobrazit systémový protokol (log)

Systémová hlášení nekončí pouze startem počítače. I poté jsou důležité informace o stavu systému ukládány do protokolového souboru -- logu. Tento soubor se podobně jako startovací protokol hodí pro odhalování příčin chyb. Protokolový soubor, ze kterého se informace čerpají je `/var/log/messages`.

## 2.9.4 Načíst CD s ovladačem od výrobce

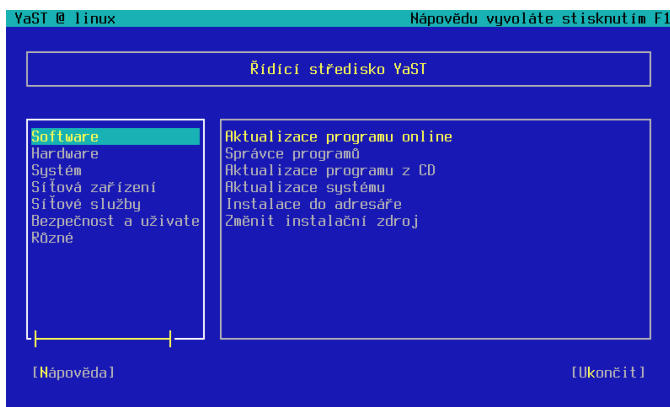
Pomocí tohoto modulu můžete automaticky instalovat linuxové ovladače pro SUSE LINUX. Pokud jste již SUSE LINUX nainstalovali, použijte tento modul pro doinstalování ovladačů od výrobce vašeho zařízení.

# 2.10 YaST v textovém režimu (ncurses)

Tato část je určena především pro správce systémů a pokročilé uživatele, kteří nechtějí spouštět X server a používají pouze textové nástroje. Najdete zde základní

informace o běhu programu YaST v textovém režimu (ncurses).

Při spuštění programu YaST v textovém režimu se nejdřív spustí Řídící středisko YaST viz 2.8 na této straně. Hlavní okno se skládá ze tří částí. V levé výrazně bíle orámované části najdete základní kategorie nabídky. Aktivní kategorie je označena barevným pozadím. V pravém okně, orámovaném tenkou bílou čarou, je seznam modulů spadajících do vybrané kategorie. Dole se nacházejí dvě tlačítka, 'Nápověda' a 'Ukončit'.



**Obrázek 2.8:** Hlavní okno programu YaST v textovém režimu

Po spuštění Řídícího střediska YaST je automaticky předvolena kategorie 'Software'. Mezi kategoriemi se můžete pohybovat pomocí kláves `↑` a `↓`. Do okna jednotlivých modulů se přepnete stisknutím klávesy `→`. Orámování okna s moduly se stane výraznější. Požadovaný modul vyberete pomocí kláves `→` a `↓`. Pozadí právě zvoleného modulu se podbarví.

Zvolený modul spustíte stisknutím klávesy `(Enter)`. Řada tlačítek obsahuje v názvu jedno zvýrazněné písmeno (standardně žluté). Tato písmena lze použít ke spuštění akce tlačítka pomocí klávesové zkratky. Místo přesunu na tlačítko pomocí klávesy `(Tab)` tak můžete řadu akcí spustit současným stisknutím kláves `(Alt)-(ŽlutéPísmeno)`. Řídící středisko YaST ukončíte stisknutím tlačítka 'Ukončit'.

## 2.10.1 Navigace v modulech

Následující popis používání modulů programu YaST předpokládá, že všechny funkční klávesy a klávesové kombinace s klávesou **(Alt)** fungují a nejsou obsazeny žádnou globální funkcí. O možných výjimkách si přečtete v části 2.10.2 na této straně.

**Navigace mezi tlačítky a výběry** K pohybu mezi tlačítky a jednotlivými seznamy s výběry použijte klávesu **(Tab)** a **(Alt)–(Tab)** nebo **(Shift)–(Tab)**.

**Navigace ve výběrech** Mezi jednotlivými položkami seznamu v aktivním rámu se pohybujte pomocí šipek (**(Up)** a **(Down)**). Pokud je některá položka větší než velikost okna, použijte pro horizontální posun doprava klávesu **(Shift)–(Right)** nebo doleva **(Shift)–(Left)**. Použít lze také klávesovou kombinaci **(Ctrl)–(E)** nebo **(Ctrl)–(A)**. Tuto kombinaci lze použít i v případě, že stisknutí kláves **(Up)** nebo **(Down)** vede ke změně aktivního rámu nebo seznamu.

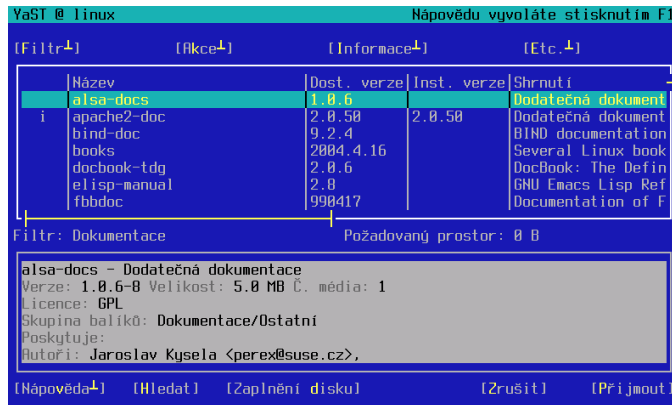
**Tlačítka a přepínače** Zaškrťovací přepínače (hranaté závorky) a skupinové přepínače (kulaté závorky) zvolíte pomocí mezerníku nebo klávesy **(Enter)**. Výběr je možný také současným stisknutím kombinace kláves **(Alt)–(ZlutePismo)**. V takovém případě není nutné potvrzení stisknutím klávesy **(Enter)**. Pokud se v nabídce pohybujete pomocí klávesy **(Tab)**, potvrďte výběr klávesou **(Enter)**.

**Funkční klávesy** Funkční klávesy **(F1)** až **(F12)** umožňují rychlý přístup k řadě tlačítek. Mapování je závislé na spuštěném modulu, takže různé moduly nabízí různá tlačítka (Podrobnosti, Vložit, Smazat...). Pro tlačítka 'OK', 'Další' a 'Ukončit' je používána klávesa **(F10)**. Náповědu, ve které najdete podrobnější informace o aktuálním mapování, vyvoláte stisknutím klávesy **(F1)**.

## 2.10.2 Omezení klávesových zkratk

Pokud správce oken používá globální kombinace s **(Alt)**, nemusí klávesové kombinace s **(Alt)** v programu YaST fungovat. Klávesy **(Alt)** nebo **(Shift)** mohou být také blokovány nastavením terminálu.

**Nahrazení **(Alt)** klávesou **(Esc)**** Klávesu **(Alt)** lze nahradit klávesou **(Esc)**. Například klávesová kombinace **(Esc)–(H)** (stisknuté postupně) nahrazuje **(Alt)–(H)** (stisknuté současně).



Obrázek 2.9: Modul instalace softwaru

Přechod k následujícímu nebo předchozímu je obvykle prováděn kombinací **Ctrl**–**F** a **Ctrl**–**B**.

Pokud jsou nefunkční klávesy **Alt** a **Shift**, použijte **Ctrl**–**F** (následující) a **Ctrl**–**B** (předchozí).

**Omezení funkčních kláves** Funkční klávesy jsou obvykle také využívány pro funkce. Řada z nich může být obsazena terminálem a tím pádem nedostupná v programu YaST. Klávesové kombinace s **Alt** i funkční klávesy by však vždy měly fungovat v čistě textové konzoli.

### 2.10.3 Spouštění jednotlivých modulů

Jednotlivé moduly programu YaST lze spouštět také samostatně. Stačí zadat příkaz `yast <jmeno_modulu>`. Síťový modul tak například spustíte příkazem `yast lan`. Seznam dostupných modulů získáte zadáním příkazu `yast -l` nebo `yast --list`.

### 2.10.4 YOU modul

Stejně jako všechny moduly lze také YaST online update (YOU) spouštět jednoduchým příkazem jako uživatel `root`:

```
yast online_update .url <url>
```

`yast online_update` spustí požadovaný modul. Volbou `.url` lze nastavit server (lokální nebo na Internetu), ze kterého YOU bude stahovat informace o opravách a samotné balíčky s opravami. Pokud žádný server nenastavíte, bude použito aktuální nastavení z YOU dialogu programu YaST. V případě, že chcete nastavit automatickou aktualizaci pomocí úlohy cronu, použijte nabídku 'Konfigurovat plně automatickou aktualizací'.

## 2.11 Online update z příkazové řádky

Automatický update systému můžete zajistit pomocí programu pro příkazovou řádku `online_update`. Můžete například nastavit server, ze kterého se budou aktualizace stahovat, jejich rozsah a také interval aktualizace. Pokud chcete, nemusíte nastavovat automatickou instalaci, ale pouze stažení oprav. Pak si můžete v klidu prostudovat popisy oprav a provést instalaci později.

Před použitím programu `online_update` nastavte cron, aby spouštěl následující příkaz:

```
online_update -u <URL> -g <type_specification>
```

`-u` určuje URL adresářového stromu, odkud se budou opravy stahovat. Podporované protokoly jsou `http`, `ftp`, `smb`, `nfs`, `cd`, `dvd` a `dir`. Volbou `-g` stáhnete opravy na svůj počítač, ale zakázete instalaci. Volitelně můžete zadat také typ oprav, které se mají stáhnout: `security` (bezpečnostní), `recommended` (doporučené) nebo `optional` (volitelné). Bez tohoto upřesnění `online_update` stáhne pouze všechny opravy označené jako `security` a `recommended`.

Stažené opravy lze nainstalovat najednou bez zadání typu. Program `online_update` ukládá všechny stažené opravy v adresáři `/var/lib/YaST2/you/mnt`. Stažené opravy nainstalujete příkazem:

```
online_update -u /var/lib/YaST2/you/mnt/ -i
```

Parametrem `-u` předáte přesné lokální URL oprav. Pomocí `-i` spustíte instalaci.

Pokud si chcete jednotlivé stažené opravy před instalací nejdříve prohlédnout, zadejte příkaz:

```
yast online_update .url /var/lib/YaST2/you/mnt/
```

Chování YOU (YaST Online Update) lze na příkazové řádce ovládat pomocí parametrů. Syntaxe je `online_update [parametry]`. Dostupné parametry a jejich význam jsou následující:

- u URL** URL adresářového stromu, ze kterého se budou stahovat opravy.
- g** Stažení oprav. Nic se nebude instalovat.
- i** Instalace již stažených oprav. Nic dalšího se nebude stahovat.
- k** Kontrola nových oprav.
- c** Zobrazení aktuálního nastavení. Nebude provedena žádná akce.
- p produkt** Produktu, pro který se budou stahovat opravy.
- v verze** Verze produktu, pro kterou se budou stahovat opravy.
- a architektura** Architektura, pro kterou se budou stahovat opravy.
- d** Běh nanečisto. Stažení oprav a simulace instalace (pouze pro testovací účely, systém zůstane nezměněn).
- n** Bez kontroly podpisu stažených souborů.
- s** Zobrazení dostupných oprav.
- v** Ladicí režim.
- D** Ladicí režim pro pokročilé uživatele a pro případ řešení problémů.

Další informace získáte zadáním příkazu `online_update-h`.



# Zvláštní instalační postupy

SUSE LINUX lze nainstalovat různými způsoby. Nabízejí se vám možnosti od pohodlné grafické instalace až po instalaci v textovém režimu s řadou ručních úprav. V následujícím oddíle najdete různé instalační postupy z různých instalačních zdrojů (CD-ROM, NFS). Také zde najdete informace o řešení možných potíží pro instalaci. V této kapitole najdete také informace o řešení problémů, které se objeví během instalace, a podrobný popis rozdělování disku.

3.1	Nastavení centrálního instalačního serveru . . . . .	82
3.2	Program linuxrc . . . . .	85
3.3	Instalace pomocí VNC . . . . .	87
3.4	Textová instalace pomocí YaST . . . . .	88
3.5	Tipy a triky . . . . .	90
3.6	Přiřazování trvalých souborů zařízení SCSI zařízením . .	94
3.7	Konfigurace LVM . . . . .	95
3.8	Konfigurace softwarového RAIDu . . . . .	100

## 3.1 Nastavení centrálního instalačního serveru

Místo ruční instalace z médií můžete použít síťovou instalaci z instalačního serveru. Tento typ je velmi užitečný především ve velkých sítích. YaST instalační server podporuje protokoly HTTP, FTP a NFS. S pomocí SLP (*service location protocols*) lze server rychle zpřístupnit všem klientům v síti, takže nebude nutné na každé stanici nastavovat instalační zdroj ručně.

### Tip

#### Informace o SLP

Podrobnější informaci o SLP v systému SUSE LINUX najdete v kapitole 23 na straně 405.

### Tip

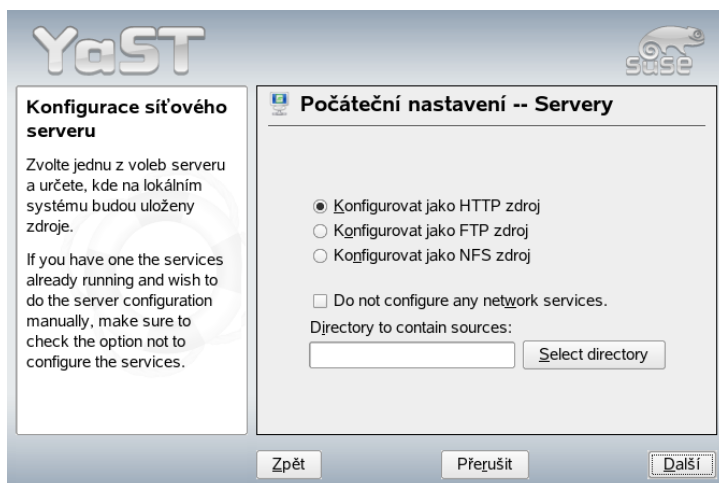
### 3.1.1 Konfigurace pomocí YaST

Zvolte 'Různé' → 'Instalační server'. Konfigurace nového instalačního serveru se skládá ze tří kroků:

**Volba typu serveru** YaST podporuje tři různé typy instalačních serverů: HTTP, FTP a NFS. Zvolte požadovaný typ. Výběrem docílíte automatické spuštění služeb spojených s instalačním serverem při každém startu systému. Pokud již některé ze služeb na vašem systému běží a vy ji chcete nastavit ručně, zrušte automatickou konfiguraci serveru odškrtnutím 'Do not configure any network services'. Ať už použijete automatické nastavení služeb nebo ne, musíte zadat adresář, ve kterém se budou nacházet instalační data viz 3.1 na následující straně.

#### Podrobnější nastavení instalačního serveru

tento krok je spojen s automatickým nastavením serveru. Pokud jste zvolili, aby se neprovádělo, bude přeskočen. Můžete zde nastavit alias kořenového adresáře FTP nebo HTTP serveru s instalačními daty. Instalační zdroj se bude později nacházet v `ftp://<Server-IP>/<Alias>/<Jmeno>` (FTP) nebo `http://<Server-IP>/<Alias>/<Jmeno>` (HTTP). Kde `<Jmeno>` je jméno vašeho instalačního zdroje, které nastavíte v následujícím kroku. V případě NFS v tomto kroku nastavíte zástupné znaky a volby



**Obrázek 3.1:** YaST instalační server: Výběr typu serveru

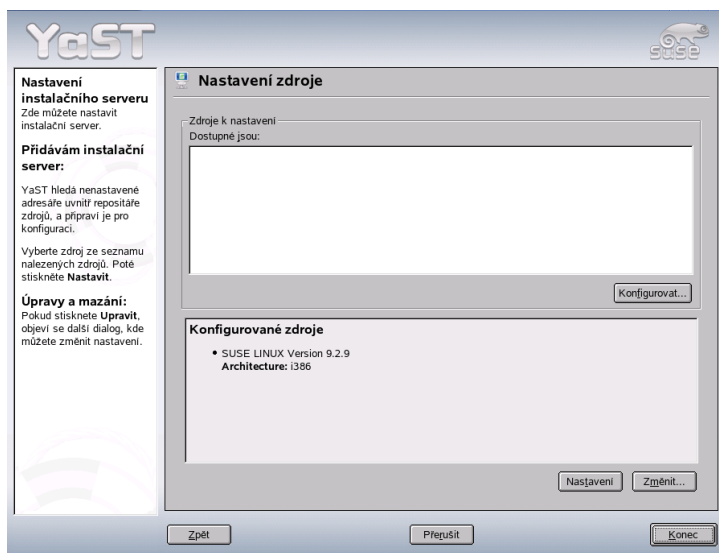
exports. NFS server bude dostupný na `nfs://<IP_Adresa>/<Jmeno>`. Podrobnosti o NFS a příkazu `exports` najdete v části 26.4 na straně 438.

**Konfigurace instalačního zdroje** Před překopírováním instalačních médií musíte nastavit jméno instalačního zdroje (obvykle zkratka jména produktu a verze). Místo překopírování obsahu instalačních médií systému SUSE LINUX můžete použít ISO obrazy. V takovém případě zatrhnete příslušnou volbu v nastavení zdroje a zadejte jejich umístění. V závislosti na produktu je možné, že pro kompletní instalaci budete potřebovat opravná (SP) nebo jiná dodatečná CD. V takovém případě aktivujte 'Prompt for Additional CDs', YaST vám pak nutnost zadání těchto CD automaticky připomene. Pokud chcete, aby byly služby vašeho instalačního serveru automaticky nabízené v síti, zatrhnete SLP.

**Nahrání instalačních dat** nejdůležitější krok v konfiguraci instalačního serveru je překopírování obsahu instalačních CD. Vkládejte média v pořadí požadovaném program YaST. Po dokončení kopírování se vraťte do přehledu zdrojů a stiskněte 'Finish'.

Váš instalační server je nyní nakonfigurován a připraven poskytovat služby. Pokud jste zvolili automatické nastavení, není nutné žádné další nastavení. V pří-

padě ručního nastavení nezapomeňte ručně spustit jednotlivé služby. Pokud chcete, aby váš instalační server sloužil pro více než jeden produkt, spusťte modul instalačního serveru programu YaST a zvolte v přehledu existujících instalačních zdrojů 'Configure' viz 3.2 na této straně. V následujícím dialogu nastavte nový instalační zdroj.



*Obrázek 3.2: YaST instalační server: Přehled instalačních zdrojů*

Instalační zdroj deaktivujete v přehledu instalačních zdrojů volbou 'Change'. V seznamu instalačních zdrojů zvolte ten, který si přejete změnit a klikněte na tlačítko 'Delete'. Dojde ke smazání všech služeb spojených s vybraným zdrojem. Samotná instalační data zůstanou netknutá. Zcela smazat je můžete ručně.

### 3.1.2 Instalace klienta

Po nastavení je instalační server dostupný pro všechny počítače v síti. Pokud je potřeba klienta nainstalovat, potřebujete pouze médium, kterým spustíte instalaci. Pak stačí zadat jméno serveru, ze kterého se provede instalace ve formátu `install=<URL>`. Informace o startovacích parametrech najdete v části 3.2 na následující straně.

V případě, že se vaše síťové rozhraní nastavuje automaticky např. přes DHCP, můžete nyní spustit instalaci. V opačném případě je nutné v `linuxrc` zadat parametr `HostIP`.

Mnohem jednoduší situaci máte, pokud jsou instalační zdroje dostupné přes SLP. Stiskněte klávesu `(F3)` a zvolte volbu `SLP`. Výběr potvrdíte stisknutím klávesy `(Enter)`. SLP zdroj můžete zadat také jako parametr `install=slp`. V obou případech se `linuxrc` pokusí v síti vyhledat instalační zdroj nabízený přes SLP.

Nyní v instalačním nabídce zvolte 'Installation' a spusťte instalaci stisknutím klávesy `(Enter)`. Zavede se instalační jádro a YaST zahájí instalaci. V případě, že instalační server nabízí instalační zdroje přes SLP, zvolte před spuštěním instalace instalační zdroj v `linuxrc`.

Zbytek instalace probíhá tak, jak je popsáno v kapitole 1 na straně 3. Podrobnější informace o SLP protokolu najdete v kapitole 23 na straně 405.

## 3.2 Program linuxrc

Každý počítač má zvláštní rutiny BIOSu, které při spuštění inicializují hardware. Při startu systému tyto rutiny zavedou také obraz, který obstará zbytek startovacího procesu. Obvykle je obsahem tohoto obrazu zavaděč, který uživateli nabídne možnost spuštění nainstalovaného operačního systému nebo instalaci systému nového. Při volbě instalace systému SUSE LINUX se zavede obraz, který obsahuje jádro a program `linuxrc`.

`linuxrc` je program, který analyzuje a inicializuje systém před spuštěním instalace. Nevyžaduje žádný zásah ze strany uživatele a po detekci hardwaru a zavedení potřebných modulů spouští instalační program YaST.

Využití `linuxrc` není omezeno jen na instalaci. Můžete jej použít také jako nástroj pro spuštění již nainstalovaného systému nebo jako nezávislý ramdiskzáchranný systém. Více informací o tomto způsobu použití najdete v části 5.4 na straně 135.

Pokud systém používá ramdisk (`initrd`), volá `linuxrc` také při zavádění modulů během startu systému. Tento skript je vytvářen dynamicky skriptem `/sbin/mkinitrd`. Jde o zcela odlišný proces a skript by neměl být zaměňován za program `linuxrc` volaný během instalace.

Když se Vám nepodaří spustit `linuxrc` v manuálovém módu, aplikace hledá info soubor na disketě nebo v `initrd` v adresáři `/info`. Následně `linuxrc` nahrává parametry kernelu. Základní hodnoty upravuje soubor `/linuxrc.config`. Doporučujeme implementovat změny v info souboru.

---

**Tip****linuxrc v ručním režimu**

program linuxrc je možné spustit v ručním režimu. Ruční režim spustíte zadáním startovacího parametru "manual=1".

---

**Tip**

Soubor info se skládá z klíčových slov a hodnot ve formátu *klíčové\_slovo: hodnota*. Tento pár klíč-hodnota můžete vložit do menu bootu, který je k dispozici z instalačního média a používá formát *key=value*. Seznam možných klíčů je v souboru `/usr/share/doc/packages/linuxrc/linuxrc.html`. To co následuje je seznam těch nejdůležitějších klíčů s příklady vložených hodnot:

**Install: URL (nfs, ftp, hd, etc.)** Umožňuje specifikovat instalační zdroj jako odkaz URL. Používané protokoly: `cd`, `hd`, `nfs`, `smb`, `ftp`, `http` and `tftp`. Syntaxe URL je stejná jako ta nejběžnější forma používaná webovými prohlížeči, např:

- `nfs://<server>/<directory>`
- `ftp://[user[:password]@]<server>/<directory>`

**Netdevice: <eth0>** Klíčové slovo `Netdevice`: umožňuje definovat rozhraní používané programem linuxrc v případě, že je na vzdáleném zdrojovém počítači k dispozici několik ethernetových rozhraní.

**HostIP: <10.10.0.2>** Toto umožňuje zvolit IP adresu vzdáleného serveru.

**Gateway: <10.10.0.128>** Toto umožňuje určit skrze kterou bránu je možné přistoupit k instalačnímu serveru. Hodí se to ve chvíli, kdy není zdrojový server ve stejné síti jako instalovaný počítač.

**Proxy: <10.10.0.1>** Klíčové slovo `Proxy`: umožňuje definovat proxy pro protokoly HTTP a FTP.

**ProxyPort: <3128>** Toto definuje port používaný proxy, v případě že nepoužíváte defaultní port.

**Textmode: <0|1>** Tímto klíčovým slovem startujete YaST v textovém módu.

**AutoYast: <ftp://autoyastfile>** `AutoYast` používáme, když potřebujeme automatickou instalaci. Hodnota musí být URL ukazující na instalační soubor `Autoyastu`.

**VNC:** `<0|1>` VNC parametr umožňuje kontrolovat instalační proces via VNC, což je zvlášť příjemné pro servery, které nemají grafickou konzoli. Když povolíte VNC, aktivujete tuto službu i na zdrojovém počítači. Podívejte se také na heslo `VNCPassword`.

**VNCPassword:** `<password>` Položka umožňuje nastavit heslo pro instalaci pomocí VNC a kontrolovat tak přístup k relaci.

**UseSSH:** `<0|1>` Tato položka umožňuje přistoupit k programu `linuxrc` pomocí protokolu SSH. Děje se tak při instalaci `YaSTem`, v jeho textovém módu.

**SSHPassword:** `<password>` Toto Vám umožní nastavit heslo pro administrátora `root` uživateli aplikace `linuxrc`.

**Insmo:** `<module parametry>` Umožňuje určit modul, který se má načíst spolu s jádrem systému a parametry, které potřebujete zadat. Parametry modulu musí být odděleny prázdnými znaky.

**AddSwap:** `<0|3|/dev/hda5>` Systém se nepokusí aktivovat swapový oddíl, když nastavíte hodnotu na 0. Když je nastavena kladná hodnota, bude příslušný oddíl aktivován a rozeznáván jako odkládací oddíl. Jinou variantou je napsat zde plné jméno zařízení daného oddílu.

## 3.3 Instalace pomocí VNC

VNC (*Virtual Network Computing*) je klient-server řešení, které umožňuje ovládat vzdálený X server pomocí jednoduchého klienta. Tento klient je dostupný pro řadu operačních systémů od různých verzí Microsoft Windows přes MacOS firmy Apple až po Linux.

Klientská aplikace VNC, `vncviewer`, je zodpovědná za zobrazení a ovládání grafického rozhraní programu `YaST` v průběhu instalačního procesu. Před startem instalovaného počítače s architekturou `(i386)` je třeba připravit vzdálený počítač tak, aby z něj bylo možné komunikovat s instalovaným počítačem pomocí sítě.

### 3.3.1 Příprava pro instalaci pomocí VNC

Abyste mohli provést instalaci pomocí VNC, je nutné předat jádru určité parametry před tím, než bude jádro spuštěno. Do příkazové řádky pro zavedení systému (boot prompt) zadejte následující text:

```
vnc=1 vncpassword=<Heslo> install=<Zdroj>
```

`vnc=1` způsobí start VNC serveru na instalovaném systému. `vncpassword` je heslo pro připojení, které bude použito později. Instalační zdroj (`install`) může být specifikován manuálně (zadejte protokol a URL zdrojového adresáře) nebo může obsahovat speciální instrukci `slp: /`. V takovém případě bude instalační zdroj automaticky specifikován SLP dotazem. Více informací o SLP technologii najdete v části 23 na straně 405.

### 3.3.2 Klientské programy pro instalaci pomocí VNC

Spojení na instalovaný počítač a jeho VNC server je zprostředkováno VNC klientem. Pokud použijete SUSE LINUX, je na tuto úlohu nejvhodnější aplikace `vn-cviewer`, která je součástí balíku `xorg-x11-Xvnc`. Pro spojení na instalovaný počítač z operačního systému Windows byste měli nainstalovat aplikaci `tightvnc`. Můžete ji najít na prvním CD SUSE LINUX, v adresáři `/dosutils/tightvnc`.

Spusťte klientský program VNC podle vašeho výběru. Jakmile budete dotázáni, zadejte IP adresu instalovaného systému a VNC heslo.

Jako alternativu můžete použít pro VNC spojení internetový prohlížeč s podporou Javy. Do políčka pro adresu zadejte následující adresu:

```
http://<IP_adresa_instalovaneho_stroje>:5801/
```

Jakmile bude spojení navázáno, YaST spustí instalaci a bude dále pokračovat.

## 3.4 Textová instalace pomocí YaST

Kromě instalace v grafickém módu může být SUSE LINUX instalován také pomocí textové verze programu YaST (konzolový mód). Všechny moduly YaST jsou dostupné také v textovém módu. Tato varianta je důležitá v případech, kdy grafické prostředí nepotřebujete (např. pro serverové instalace) nebo grafická karta není podporována systémem X Window. Textový režim je také vhodný pro zrakově postižené.

Nejprve je nutné nastavit pořadí médií pro zavádění systému v BIOS tak, aby bylo možné zavádět systém z CD-ROM mechaniky. Vložte DVD nebo CD1 do



mechaniky a restartujte systém. Po několika vteřinách se zobrazí úvodní obrazovka.

Použijte **↓** a **→** a vyberte 'Installation' v několika následujících vteřinách abyste zamezili automatickému stratu programu YaST. Pokud váš hardware vyžaduje nějaké zvláštní volby, zadejte je do `Boot Options`. Parametr `textmode=1` je použit ke spuštění programu YaST v textovém módu.

Použijte **F2** ('Video Mode') k nastavení rozlišení obrazovky pro instalaci. Pokud se dají očekávat problémy s vaší grafickou kartou, vyberte 'Text Mode'. Poté stiskněte **Enter**. Objeví se dialog se dialog 'Loading Linux kernel'. Jádro se zavede a spustí se `linuxrc`. Pokračujte v instalaci pomocí dalších menu.

### Tip

#### Instalace v Češtině

Pokud vám nevyhovuje instalační dialog v angličtině, můžete si nastavit češtinu pomocí klávesy **F4**.

### Tip

Různé problémy při startu systému mohou být obvykle odstraněny manipulací s parametry jádra. Pokud způsobuje problémy DMA, použijte nabídku 'Installation Safe Settings'.

Následující parametry jádra můžete použít pokud nastanou problémy se systémem řízení spotřeby ACPI (Advanced Configuration and Power Interface):

**acpi=off** Tento parametr vypíná kompletně ACPI subsystém na vašem počítači. Jeho použití je vhodné zejména v případech, kdy má počítač problémy s ACPI obecně nebo pokud máte pocit, že množství nespecifických problémů je způsobeno ACPI.

**acpi=oldboot** Vypíná ACPI pro všechny části systému kromě těch, které jsou důležité pro zavedení systému.

**acpi=force** Vždy zapíná ACPI, i v případě, že počítač má starší BIOS (vydaný před rokem 2000). Tento parametr zapne ACPI i potom, co je jádru zadán parametr `acpi=off`.

**pci=noacpi** Zakáže ACPI subsystému manipulaci s PCI IRQ routováním.

Pro více informací najdete v SDB databázi <https://portal.suse.com> po zadání klíčového slova *acpi*.

Pokud se při nahrávání jádra nebo v různých částech instalace objevují nahodilé chyby, vyberte po startu systému volbu 'Memory Test' v úvodním menu. Tato volba provede kontrolu paměti, na kterou má Linux poměrně velké nároky. V praxi to znamená, že paměť a její časování musí odpovídat všem standardům. Více informací je možné získat na adrese [http://portal.suse.com/sdb/en/2001/05/thallma\\_memtest86.html](http://portal.suse.com/sdb/en/2001/05/thallma_memtest86.html). Pokud je to možné, nechte běžet test paměti přes noc - jeho běh je poměrně dlouhý.

## 3.5 Tipy a triky

Především při instalaci na starší hardware se můžete setkat s problémy s instalací z CD nebo DVD. V takovém případě pro vás mám několik triků, jak tuto překážku překonat. Použít můžete buď instalační disketu, kterou si musíte nejdříve vytvořit, nebo spustit instalaci z druhého instalačního CD.

### 3.5.1 Vytváření startovací diskety v operačním systému DOS

Potřebujete naformátovanou 3.5" HD disketu a 3.5" mechaniku, ze které lze zavádět systém. Adresář `boot` na CD1 obsahuje několik obrazů disket. S patřičnou utilitou mohou být tyto obrazy nakopírovány na disketu. Takto připravená disketa se nazývá startovací disketa.

Obrazy disket také obsahují zavaděč systému `SYSLINUX` a program `linuxrc`. `SYSLINUX` umožňuje výběr jádra v průběhu zavádění systému a specifikaci parametrů potřebných pro použitý hardware. Aplikace `linuxrc` podporuje zavádění jaderných modulů pro váš hardware a řídí další instalační proces.

#### Vytváření startovací diskety s pomocí `rawrwritewin`

Ve Windows mohou být startovací diskety vytvořeny pomocí grafické utility `rawrwritewin` - tuto utilitu naleznete v adresáři `dosutils/rawrwritewin` na CD1.

Po startu vyberte soubor z obrazem diskety, soubory jsou uloženy v adresáři `boot` na CD1. Jako minimum budete potřebovat diskové obrazy *bootdisk* a *modules1*. Pro zobrazení těchto souborů v dialogu pro otevření nastavte typ souborů na *všechny soubory*. Po vybrání souboru vložte disketu do mechaniky a klikněte na *write*. Pro vytvoření více disket celý postup opakujte.

## Vytváření startovací diskety s pomocí rawrite

Utilita `rawrite.exe` pro DOS (CD1, adresář `dosutils/rawrite`) může být použita pro vytváření startovací diskety a diskety s moduly systému SUSE. Pro její použití potřebujete počítač s operačním systémem DOS (například FreeDOS) nebo Windows.

Ve Windows XP postupujte následujícím způsobem:

1. Vložte SUSE LINUX CD1.
2. Otevřete okno s příkazovou řádkou (ve start menu, vyberte 'Příslušenství' → 'Příkazová řádka').
3. Spusťte `rawrite.exe` se správnou specifikací cesty pro Vaší CD mechaniku. Následující příklad předpokládá, že jste v adresáři Windows na harddisku C: a vaše CD mechanika má označení D:.

```
d:\dosutils\rawrite\rawrite
```

4. Po startu budete dotázáni na zdroj a cíl souboru ke kopírování. Obraz startovací diskety je uložen v adresáři `boot` na CD1. Jméno souboru je `bootdisk`. Nezapomeňte zadat i cestu pro vaší CD mechaniku.

```
d:\dosutils\rawrite\rawrite
```

```
RaWrite 1.2 - Write disk file to raw floppy diskette
```

```
Enter source file name: d:\boot\bootdisk
```

```
Enter destination drive: a:
```

Jakmile zadáte cílovou mechaniku `a:`, `rawrite` vás požádá o vložení naformátované diskety a stisknutí (**Enter**). Následně je zobrazen postup kopírování. Akce může být zrušena pomocí stisku (**Ctrl**)-**C**). Další obrazy disket (`modules1`, `modules2`, `modules3`, a `modules4`) mohou být vytvořeny stejným způsobem.

### 3.5.2 Vytváření startovací diskety v operačním systému typu UNIX

V Unixovém operačním systému nebo v Linuxu potřebujete CD-ROM mechaniku, disketovou mechaniku a disketu (3,5"). Postup pro vytvoření startovacích disket:

1. Pokud potřebujete disketu nejprve naformátovat, použijte:

```
fdformat /dev/fd0u1440
```

2. Připojte CD1 (například do `/media/cdrom`). V současné verzi již není nutné CD připojovat ručně.
3. Přejděte do adresáře `boot` na CD:

```
cd /media/cdrom/boot
```

4. Vytvořte startovací disketu pomocí následujícího příkazu:

```
dd if=bootdisk1 of=/dev/fd0 bs=8k
```

5. Opakujte postup s obrazy `bootdisk2` a `bootdisk3`.

Soubor `README` v adresáři `boot` obsahuje další informace o obrazech disket. Tento soubor si můžete přečíst s pomocí příkazů `more` nebo `less`.

Další obrazy disket (`modules1`, `modules2`, `modules3`, a `modules4`) mohou být vytvořeny stejným způsobem. Tyto diskety jsou třeba zejména pokud máte USB nebo SCSI zařízení, popřípadě síťovou nebo PCMCIA kartu, které je třeba zpřístupnit během instalace. Disketa s moduly může být také třeba v případě použití speciálního souborového systému v průběhu instalace.

Použití vlastního jádra v průběhu instalace je trochu složitější. V takovém případě zapište na disketu standardní obraz `bootdisk` a poté přepište soubor s jádrem `linux` vašim vlastním (více informací v souboru `/usr/share/doc/packages/yast2-installation/vendor.html`).

### 3.5.3 Zavádění systému z diskety (SYSLINUX)

Zavádění systému je zahájeno zavaděčem SYSLINUX (`syslinux`). Když je systém zaveden, SYSLINUX spustí minimalizovanou detekci hardware, která se skládá z několika hlavních kroků:

1. Program otestuje jestli BIOS podporuje VESA2.0kompatibilní framebuffer a nastartuje jádro s patřičnými parametry.
2. Je přečtena informace o monitoru (DDC info).
3. První blok prvního harddisku (MBR) je načten pro namapování BIOS identifikace na jména zařízení v Linuxu v průběhu konfigurace zavaděče. Program se pokusí číst MBR pomocí lba32 funkcí pro zjištění jejich podpory v BIOSu.

**Tip**

Pokud podržíte klávesu (Shift) po čas nastartování SYSLINUX, všechny výše popsané kroky budou přeskočeny. Pro ladění můžete vložit řádku

```
verbose 1
```

do souboru `syslinux.cfg` uloženém na disketě, zavaděč pak zobrazí informace o všech probíhajících krocích.

**Tip**

Pokud počítač nespouští z diskety, můžete se pokusit změnit pořadí médií pro zavádění systému v BIOSu na A, C, CDROM.

### 3.5.4 Použití CD2 pro zavádění systému

► x86

CD2 je také možné použít pro zavádění systému. Na rozdíl od CD1, které používá bootovatelný ISO obraz CD, CD2 zavádí systém z obrazu 2.88MB diskety. Použijte CD2 v případě, že jste si jistí, že systém může startovat z CD, ale zavedení systému z CD1 nefunguje (jako náhradní variantu). ◀

### 3.5.5 Podporované CD mechaniky

Většina CD mechanik je podporována. Pokud dojde k problémům při spouštění instalace z CD, použijte ke spuštění CD 2.

Podpora SCSI CD-ROM mechanik záleží na podpoře SCSI řadiče ke kterému je mechanika připojena. Podporované řadiče jsou uvedeny v databázi hardwarové podpory na <http://cdb.suse.de>. Pokud váš řadič není podporován a váš harddisk je připojen ke stejnému řadiči, máte problém.

Mechaniky CD-ROM připojené přes USB jsou podporovány také. Jestliže váš BIOS nepodporuje zavádění z USB, startujte instalaci z disket. Před zaváděním z disket se ujistěte, že veškerá vámi požadovaná USB zařízení jsou připojena a napájena.

### 3.5.6 Instalace ze síťového zdroje

V některých případech není možné provést standardní instalaci s použitím CD-ROM mechaniky. Například není mechanika podporována, protože se jedná o

starší proprietární ty. Jiný počítač, například laptop, nemusí mít CD-ROM mechaniku vůbec, má ale síťovou kartu. SUSE LINUX umožňuje instalovat počítače bez CD mechaniky pomocí síťového spojení, většinou za použití protokolů NFS nebo FTP pomocí Ethernetu. Popis instalačního postupu najdete v části 3.5.6 na předchozí straně.

Tento postup není pokryt instalační podporou. Následující postup je doporučen jen pro zkušené uživatele.

Pro instalaci systému SUSE LINUX ze síťového zdroje jsou třeba dva kroky:

1. Data potřebná pro instalaci (CD nebo DVD disky) musí být zpřístupněny na počítači, který bude fungovat jako zdroj pro instalaci.
2. Instalovaný počítač musí být schopen zavést systém například z diskety a musí mít síťovou kartu.

Instalovat lze přes různé protokoly např. NFS nebo FTP. Informace o instalaci najdete v části 3.2 na straně 85.

## 3.6 Přirazování trvalých souborů zařízení SCSI zařízením

Když je systém zaveden, SCSI zařízení mají přiřazena soubory zařízení (devices) víceméně dynamicky. Pokud se počet zařízení nemění, nepředstavuje to v podstatě problém. Nicméně pokud je přidán nový SCSI harddisk a je detekován v pořadí před staršími harddisky, bude mu přiřazeno jedno ze starých zařízení a záznamy v tabulce připojení v souboru `/etc/fstab` již nebudou odpovídat skutečnosti.

Pro obejítí tohoto problému lze použít soubor `boot.scsidev`. `boot.scsidev` řídí nastavení SCSI zařízení v průběhu zavádění systému a přiřazuje trvalá jména zařízení z adresáře `/dev/scsi/`. Tato jména pak lze použít v `/etc/fstab`.

V expertním mód editoru úrovně běhu aktivujte službu `boot.scsidev` pro úroveň běhu B. Odkazy nutné pro vygenerování jmen v průběhu zavádění systému jsou vytvářeny v adresáři `/etc/init.d/boot.d`.

## 3.7 Konfigurace LVM

Tato část krátce popisuje základy použití LVM a jeho nejdůležitější vlastnosti využitelné za mnoha různých okolností. V části 3.7.2 na straně 97 se dozvíte, jak nakonfigurovat LVM pomocí nástroje YaST.

### Varování

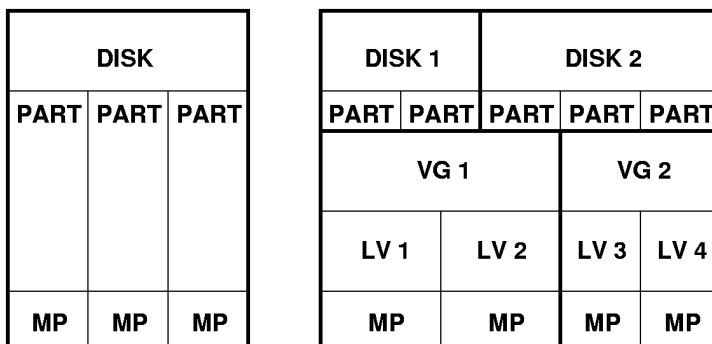
Použití LVM může představovat zvýšené riziko, např. ztráty dat. Může též zvýšit riziko pádu aplikací a dalších chybových stavů. Před nasazením LVM nebo změnou konfigurace svazků zazálohujte všechna důležitá data. Nikdy nepracujte bez zálohy.

### Varování

### 3.7.1 Správce logických svazků

Správce logických svazků (LVM) umožňuje flexibilní využití místa na pevných discích. LVM byl vyvinut, protože je často potřeba změnit rozdělení místa na pevných discích až po instalaci systému, v době, kdy již byly vytvořeny diskové oddíly. Protože je obtížné měnit diskové oddíly na běžícím systému, poskytuje LVM tzv. *virtuální skupinu svazků* (VG, volume group), ze které se podle potřeby vyčleňují *logické svazky* (LV, logical volumes). Operační systém přistupuje k logickým svazkům místo fyzických oddílů. Virtuální skupiny svazků mohou být tvořeny více než jedním diskem, mohou se skládat z mnoha disků nebo jejich částí. LVM tak abstrahuje od fyzické podstaty disků a umožňuje tak flexibilnější a snadnější změny segmentace oproti fyzickému přerozdělování disků. Více informací o fyzickém rozdělování disků najdete v částech 1.5.4 na straně 11 a 2.8.9 na straně 63.

Obrázek 3.3 na následující straně srovnává fyzické rozdělování (vlevo) s použitím LVM (vpravo). Vlevo byl jeden fyzický disk rozdělen na tři fyzické oddíly (PART). Všechny tyto oddíly mají přiřazen bod připojení (MP), takže k nim může operační systém přistupovat. Vpravo byly dva disky rozděleny na dva a tři fyzické oddíly. Byly definovány dvě skupiny svazků (VG1 a VG2). VG1 obsahuje dva oddíly z prvního disku a jeden z druhého disku. VG2 obsahuje zbývající dva oddíly z druhého disku. V rámci LVM se fyzické oddíly zařazené do skupiny svazků nazývají *fyzické svazky* (PV). Ve skupinách svazků byly definovány čtyři logické svazky (LV1 až LV4), ke kterým může operační systém přistupovat pomocí asociovaných bodů připojení. Hranice mezi logickými svazky nemusí odpovídat hranici žádného fyzického svazku. Podívejte se například na hranici mezi LV1 and LV2 v našem příkladě.



*Obrázek 3.3: Fyzické oddíly versus LVM*

Vlastnosti LVM:

- Několik pevných disků nebo oddílů lze sloučit do jednoho velkého logického svazku (LV).
- Nastane-li nedostatek volného místa v logickém svazku (např. `/usr`), lze ho při vhodné konfiguraci bez problémů rozšířit.
- Pomocí LVM lze dokonce přidat pevné disky nebo logické svazky za běhu systému. Podmínkou je ovšem hardware podporující tzv. hot swap.
- Logický svazek lze pomocí "stripping" režimu rozdělit mezi několik fyzických svazků. Pokud jsou tyto fyzické svazky na různých discích, lze dosáhnout zvýšení výkonu podobně jako v případě RAID0.
- Funkce snapshot umožňuje vytvoření konzistentní zálohy běžících systémů (zejména serverů).

LVM se vyplatí používat na intenzivně využívaných domácích počítačích nebo malých serverech. Pokud máte rychle se rozšiřující množství dat, např. databáze či MP3 archívy, je LVM ideálním řešením. Umožňuje použití souborových systémů větších, než je velikost pevného disku. Další výhodou je skutečnost, že lze použít až 256 logických svazků. Mějte však na paměti, že se práce s LVM velmi liší od práce s běžnými oddíly. Instrukce a další informace o použití LVM jsou dostupné v oficiálním LVM HOWTO dokumentu na adrese <http://tldp.org/HOWTO/LVM-HOWTO/>.



V systému s jádrem 2.6 je možno používat LVM verze 2, který je zpětně kompatibilní s předcházející verzí a umožňuje správu dříve vytvořených logických svazků. Při vytváření nových svazků je však třeba rozhodnout, zda použít nový nebo starší, zpětně kompatibilní, formát. LVM verze 2 nevyžaduje žádné jaderné záplaty. Využívá mapovač zařízení (device mapper) integrovaný v jádře 2.6. Tato verze jádra podporuje pouze LVM verze 2. Proto, kdykoliv budeme mluvit o LVM, máme na mysli LVM verze 2.

### 3.7.2 Konfigurace LVM pomocí nástroje YaST

YaST modul pro konfiguraci LVM je dostupný z expertního YaST modulu pro rozdělování disků (viz 2.8.9 na straně 63). Tento profesionální nástroj pro rozdělování disku umožňuje vytvářet, upravovat a mazat oddíly pro použití v rámci LVM. Oddíl pro LVM v něm vytvoříte kliknutím na 'Vytvořit' → 'Neformátovat' a výběrem ID '0x8E Linux LVM'. Jakmile máte vytvořeny všechny potřebné oddíly pro LVM, klikněte na 'LVM...'. Tím se zahájí konfigurace LVM.

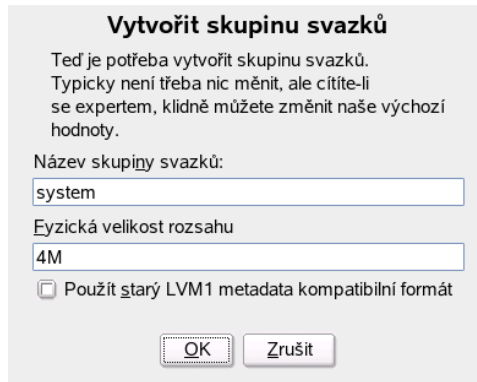
#### Vytváření skupin svazků

Pokud na systému ještě neexistuje žádná skupina svazků, budete požádáni o její vytvoření (viz 3.4 na následující straně). Další skupiny můžete vytvořit pomocí 'Přidat skupinu', obvykle ale stačí jedna. Pro skupinu svazků v systému SUSE LINUX se doporučuje použít název `system`. 'Fyzická velikost rozsahu' určuje velikost fyzického bloku ve skupině svazků. Veškerý prostor v rámci skupiny svazků se obhospodaruje v takto velkých blocích. Výchozí hodnota je 4MB, což umožňuje fyzické a logické svazky o maximální velikosti 256GB. Pokud potřebujete logické svazky větší, zvýšte hodnotu rozsahu na 8, 16 nebo 32MB.

#### Konfigurace fyzických svazků

Jakmile je vytvořena skupina svazků, objeví se seznam všech oddílů typu `Linux LVM` nebo `Linux native`. Nejsou zobrazeny odkládací ani DOS oddíly. Pokud je oddíl již zařazen do skupiny svazků, je v seznamu uvedeno její jméno. Nezařazené oddíly jsou označeny pomocí `--`.

Pokud je skupin svazků více, nastavte patřičnou skupinu v nabídce vlevo nahoře. Tlačítka vpravo nahoře umožňují vytvářet a mazat skupiny svazků. Smazat můžete pouze skupiny bez přiřazených oddílů. Oddíly zařazené do skupiny svazků se nazývají fyzické svazky (PV).



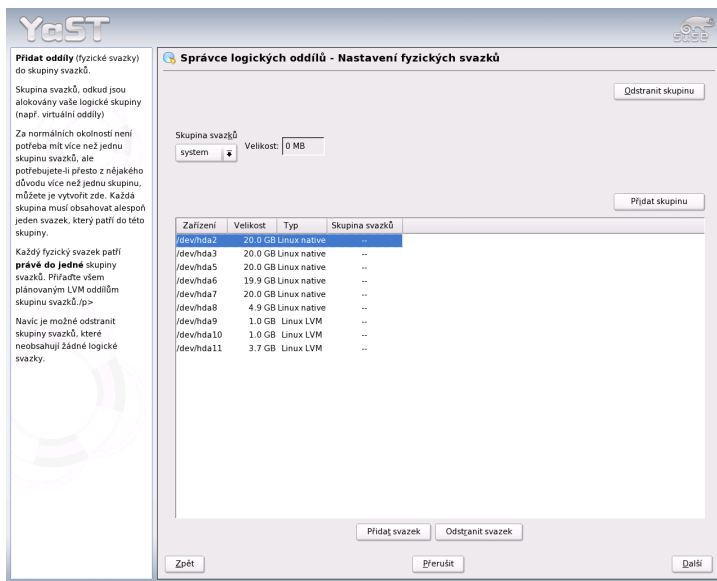
*Obrázek 3.4: Vytváření skupiny svazků*

Dosud nezařazený oddíl do zvolené skupiny svazků přidáte jednoduše. Nejprve klikněte na oddíl v seznamu a pak na tlačítko 'Přidat svazek'. V seznamu se v položce oddílu objeví název skupiny svazků. Zařad'te do skupiny všechny oddíly určené pro LVM. Jinak by místo na oddílu zůstalo nevyužito. Před opuštěním dialogu musíte přiřadit každé skupině svazků alespoň jeden fyzický svazek. Po přiřazení všech svazků klikněte na 'Další'.

### Konfigurace logických svazků

Po naplnění skupiny svazků fyzickými svazky je třeba definovat logické svazky, které bude operační systém používat. V nabídce vlevo nahoře zvolte patřičnou skupinu svazků. Hned vedle je zobrazeno volné místo v aktuální skupině svazků. Seznam níže obsahuje všechny logické svazky v aktuální skupině. Jsou zde zobrazeny všechny běžné linuxové oddíly s bodem připojení, všechny odkládací oddíly a všechny existující logické svazky. Logické svazky můžete dle libosti přidávat, upravovat a odebírat pomocí tlačítek pod seznamem. V každé skupině svazků vytvořte alespoň jeden logický svazek.

Nový logický svazek vytvoříte kliknutím na 'Přidat'. Otevře se dialog, ve kterém musíte zadat potřebné údaje. Zadejte velikost, typ souborového systému a bod připojení. Na logickém svazku se vytvoří souborový systém, např. ext2 nebo reiserfs, kterému je přidělen bod připojení. Soubory uložené na tomto logickém svazku jsou pak v systému dostupné v odpovídajícím adresáři. Je také možné rozmístit data v logickém svazku na několik fyzických svazků (striping). Pokud



Obrázek 3.5: Nastavení fyzického svazku

jsou tyto fyzické svazky umístěny na různých pevných discích, zvýší se výkon při čtení i zápisu (podobně jako u RAID0). Striping LV s  $n$  "proužky" (stripes) lze ovšem správně vytvořit jen tehdy, pokud lze požadované místo rovnoměrně rozdělit mezi  $n$  fyzických svazků. Pokud jsou například k dispozici jen dva fyzické svazky, nelze vytvořit logický svazek se třemi "proužky".

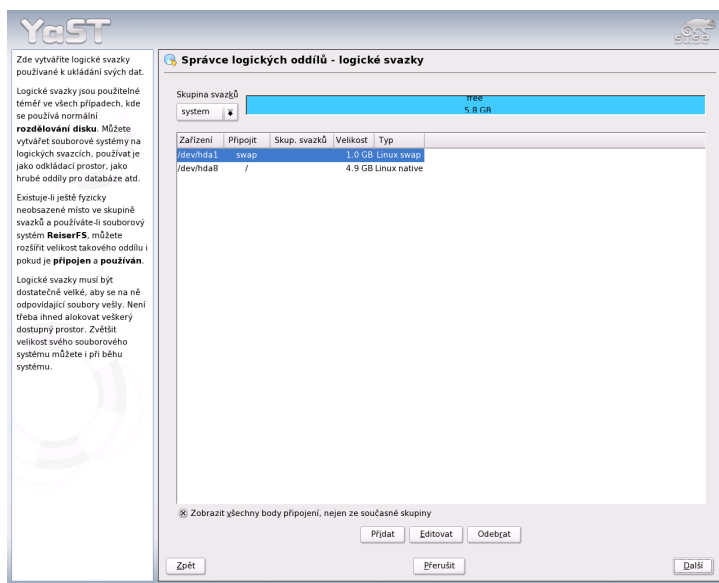
## Varování

### Striping

YaST v této fázi nemůže ověřit správnost vašeho nastavení ohledně stripingu. Chyby se projeví až později, když je LVM implementován na disk.

## Varování

Pokud jste na systému LVM již nakonfigurovali, můžete zadat existující logické svazky. Před pokračováním jim přiřadíte body připojení. Kliknutím na 'Další' se vrátíte do dialogu YaST 'Rozdělování disku pro experty'.



Obrázek 3.6: Správa logických svazků

### Přímá správa LVM

Pokud máte již LVM nakonfigurováno a jen chcete něco změnit, existuje jiná cesta. V nástroji YaST zvolte 'Systém' → 'LVM'. Otevře se dialog, který umožňuje veškeré nastavení zmíněné výše s výjimkou fyzického přerozdělování disku. Zobrazuje dva seznamy, jeden s existujícími fyzickými svazky, druhý s logickými svazky. Pracovat s nimi můžete dříve výše popsány postupy.

## 3.8 Konfigurace softwarového RAIDu

Smyslem polí RAID (redundant array of inexpensive disks – pole nepříliš drahých disků s možností redundance) je zkombinovat více diskových oddílů do jednoho velkého *virtuálního* pevného disku s vyšším výkonem a lepším zabezpečením dat. Jedna z těchto výhod je však při použití RAIDu uplatněna na úkor druhé. Většina řadičů RAID používá protokol SCSI, ten totiž umí adresovat velké

**Obrázek 3.7:** Vytváření logických svazků

množství disků efektivněji než řadiče IDE a je vhodnější pro paralelní zpracování příkazů. Nicméně existují i RAID řadiče podporující IDE nebo SATA disky. Více informací viz databázi hardwaru na adrese <http://cdb.suse.de>.

Podobné úkoly jako poměrně nákladný hardwarový RAID řadič dokáže plnit i RAID softwarový. SUSE LINUX, s pomocí konfiguračního nástroje YaST, nabízí možnost spojit několik pevných disků do jednoho softwarového RAID pole – velmi výhodné alternativy k hardwarovému RAIDu. RAID umožňuje aplikovat různé strategie kombinace disků do RAID pole, každá má jiný cíl, výhody a charakteristiky. Tyto varianty jsou známy jako tzv. typ RAIDu (*RAID level*).

### 3.8.1 Běžné typy polí RAID

**RAID0** Tento typ pole zlepšuje výkon při přístupu k datům rozložením datových bloků každého souboru na více pevných disků. Ve skutečnosti se nejedná o RAID v pravém slova smyslu, neboť neprobíhá žádné zabezpečování dat, nicméně se termín *RAID0* pro tento režim ujal. RAID0, spojuje dva

nebo více pevných disků v jeden virtuální disk. Výkon je velmi vysoký, ale výpadek jediného disku znamená selhání celého pole a ztrátu dat.

**RAID1** Tento typ pole poskytuje přiměřený stupeň ochrany dat, protože jsou kopírována na další disk v poměru 1:1. Metoda je též známá pod názvem *zrcadlení disku*. Pokud je některý disk zničen, kopie jeho obsahu je stále přístupná na dalším disku. Všechny disky kromě jednoho mohou být zničeny, aniž by byla data ohrožena. Výkon při zápisu dat je ve srovnání se samostatným pevným diskem kvůli kopírování dat o 10-20% nižší, ale čtení je naopak podstatně výkonnější, neboť se data načítají paralelně z více disků současně. Obecně se dá říci, že RAID 1 poskytuje zhruba dvojnásobný přenos při čtení a zhruba stejný při zápisu ve srovnání se samostatnými disky.

**RAID2 a RAID3** Nejedná se o typické implementace RAIDu. RAID2 rozkládá data na úrovni bitů, nikoliv bloků. RAID3 data rozkládá na úrovni bytů, má vyhrazený disk pro paritní data a nedokáže obsloužit více současných požadavků. Tyto typy se používají jen vzácně.

**RAID4** RAID4 pracuje na úrovni bloků, podobně jako RAID0, ale je vybaven vyhrazeným diskem pro paritní data. V případě havárie disku jsou paritní data použita k jeho obnově. Paritní disk ale může znamenat ztrátu výkonu při zápisu dat. Nicméně se RAID4 občas používá.

**RAID5** RAID5 je kompromisem mezi typy 0 a 1, co se týče výkonu i zabezpečení dat. Kapacita pole je rovná kapacitě všech použitých disků bez jednoho. Data jsou rozdělena na jednotlivých discích podobně jako v případě pole RAID0, ale navíc se o bezpečnost dat starají tzv. *paritní bloky*, které jsou vytvořeny na jednom z diskových oddílů. Paritní bloky jsou navzájem spojeny pomocí logického XOR, v případě výpadku jednoho z disků tak mohou být data obnovena z odpovídajících paritních bloků a ostatních dat. Při používání pole typu RAID5 nesmí dojít k výpadku více než jednoho disku současně. V zájmu ochrany dat je tedy nutné vadný disk co nejrychleji nahradit.

**Další RAID typy** Existuje mnoho dalších typů RAIDu (RAIDn, RAID10, RAID0+1, RAID30, RAID50 atd.), některé z nich jsou proprietárními implementacemi výrobců hardware. Nejsou příliš rozšířené, a proto tu nejsou vysvětleny.

### 3.8.2 Konfigurace softwarového RAIDu pomocí YaST

Konfigurace softwarového RAIDu v YaSTu je přístupná z modulu ‘Rozdělování disku pro experty’, popsaného v části 2.8.9 na straně 63. Tento profesionální nástroj pro rozdělování disku umožňuje upravovat a mazat existující oddíly a vytvářet nové pro použití v softwarovém RAIDu. RAID oddíl vytvoříte kliknutím na ‘Vytvořit’ → ‘Neformátovat’ a výběrem ‘0xFD Linux RAID’ jako identifikátoru oddílu. Pro RAID0 a RAID1 jsou zapotřebí alespoň dva oddíly, pro RAID1 se obvykle více než dva oddíly nepoužívají. Pokud chcete RAID5, musíte použít alespoň tři oddíly, které by měly mít všechny stejnou velikost. Oddíly pro RAID by měly být umístěny na různých fyzických discích, aby se předešlo ztrátě dat v případě selhání disku (RAID1 a 5) nebo dosáhlo vyššího výkonu RAID0. Po vytvoření všech oddílů pro RAID klikněte na ‘RAID’ → ‘Vytvořit RAID’. Zahájíte tak konfiguraci RAIDu.

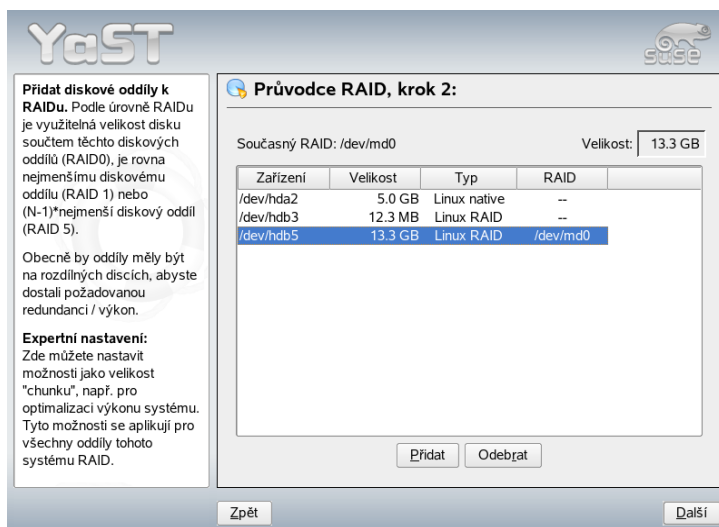
V dalším dialogu vyberte mezi RAID levely 0, 1 nebo 5 (viz 3.8.1 na straně 101). Po kliknutí na ‘Další’ se zobrazí dialog, který obsahuje přehled všech oddílů typu „Linux RAID“ nebo „Linux native“ (viz 3.8 na následující straně). Odkládací (swap) ani DOS oddíly nejsou zobrazeny. Pokud je oddíl přiřazen do RAID svazku, je zobrazeno jméno RAID zařízení (např. `/dev/md0`). Nepřiřazené oddíly jsou označeny „--“.

Chcete-li nepřřižený oddíl přidat do RAID svazku, klikněte v seznamu na oddíl a pak na ‘Přidat’. Jméno RAID zařízení se zobrazí vedle vybraného oddílu. Přiřaďte všechny oddíly určené pro RAID. Jinak by místo na nich zůstalo nevyužité. Po přiřazení všech oddílů klikněte na ‘Další’, dostanete se tak do dialogu, ve kterém můžete vyladit výkon (viz 3.9 na straně 105).

Stejně jako v případě konvenčních oddílů vyberte typ souborového systému, případné šifrování a bod připojení. Zaškrtnutí volby ‘Perzistentní superblok’ zabezpečuje rozeznání RAID oddílů při startu systému. Po ukončení konfigurace tlačítkem ‘Konec’ si prohlédněte vytvořené zařízení `/dev/md0`, případně další označená jako *RAID*, v expertním modulu pro rozdělování disku.

### 3.8.3 Řešení problémů

Zda byl některý z oddílů zapojených do RAIDu poškozen, zjistíte v souboru `/proc/mdstats`. Pokud nastala chyba, vypněte systém a vyměňte poškozený pevný disk za nový, obsahující stejné oddíly jako disk původní. Pak restartujte systém a zadejte příkaz `mdadm /dev/mdX --add /dev/sdX`. ‘X’ nahraďte patřičnými identifikátory zařízení. Tím bude nový disk automaticky zapojen do pole RAID a data budou obnovena.



*Obrázek 3.8: Oddíly RAID*

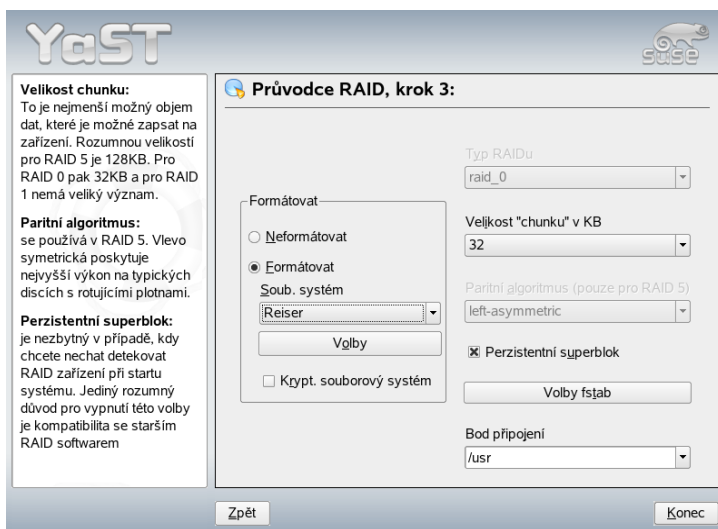
### 3.8.4 Další informace

Pokyny ke konfiguraci a další informace o softwarovém RAIDu naleznete v následujícím HOWTO dokumentu:

- `/usr/share/doc/packages/raidtools/Software-RAID.HOWTO.html`
- <http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html>

nebo v konferenci věnované linuxovému RAIDu: <http://www.mail-archive.com/linux-raid@vger.rutgers.edu>.





Obrázek 3.9: Nastavení souborového systému



# Aktualizace systému a správa balíčků

SUSE LINUX nabízí možnost aktualizovat stávající systém, aniž by bylo nezbytné ho znovu instalovat. Přitom je třeba rozlišovat mezi *aktualizací jednotlivých balíčků* a *celkovou aktualizací systému*. Balíčky lze také doinstalovat ručně pomocí RPM.

4.1	Aktualizace systému SUSE LINUX . . . . .	108
4.2	Od verze k verzi . . . . .	109
4.3	RPM the Package Manager . . . . .	120

## 4.1 Aktualizace systému SUSE LINUX

Existuje známý jev, že se software verzi od verze rozrůstá. Proto je dobré podívat se *před* aktualizací příkazem `df`, jak jsou diskové oddíly zaplněny. Pokud máte dojem, že by na to jeho kapacita nestačila, zálohujte data a proveďte přerozdělení disku. Neexistuje žádná univerzální rada, kolik místa budete potřebovat, to závisí na způsobu stávající instalace, vybraném softwaru a na tom, z které verze aktualizujete.

### 4.1.1 Přípravy

Před začátkem aktualizace byste měli zálohovat konfigurační soubory na jiné médium (streamer, disketa, výměnný disk, ZIP mechanika, vypálit na CD). V první řadě se jedná o soubory v adresáři `/etc`, dále v adresáři `/var/lib` (např. News nebo XDM). Kromě toho zálohujte také soubory z domovských adresářů.

Než spustíte samotnou aktualizaci, poznamenejte si, jaký máte kořenový diskový oddíl `/`, což zjistíte příkazem `df/`

V příkladu výstupu je kořenovým oddílem `/dev/hda2`:

```
tux@linux:~>df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       1.9G  189M  1.7G  10%  /dos
/dev/hda2       8.9G   7.1G  1.4G  84%  /
/dev/hda5       9.5G   8.3G  829M  92%  /home
```

### 4.1.2 Možné problémy

Po přechodu na novou verzi se můžete setkat s různými problémy. Zde najdete jejich popis.

#### Kontrola `passwd` a `group` v `/etc`

Před aktualizací se ujistěte, že soubory `/etc/passwd` a `/etc/group` neobsahují žádné chyby v syntaxi. To provedete jako uživatel `root` pomocí ověřovacích nástrojů `pwck` a `grpck`. Zjištěné chyby opravte.

## PostgreSQL

Před aktualizací databáze PostgreSQL balík musíte vydumpovat databázi více v `pg_dump`. Tento postup je nutné dodržovat je v případě, že byla databáze PostgreSQL před aktualizací *používána*.

### 4.1.3 Aktualizace pomocí YaST

Postupujte jako u instalace podle postupu uvedeného v části 4.1.1 na předchozí stránce pak systém aktualizujte následujícím způsobem:

1. Spusťte instalaci podle postupu popsaného v části 1.1 na straně 4. V programu YaST, nastavte jazyk. Místo 'Nová instalace' zvolte 'Aktualizace stávajícího systému'.
2. YaST zjistí, zda se na disku nenachází více kořenových oddílů. Pokud ne, pokračuje dále. Pokud na disku máte více oddílů, musíte zvolit kořenový oddíl a potvrdit výběr stisknutím tlačítka 'Další'. YaST načte starý `fstab` a pokusí se připojit zde uvedené oddíly.
3. Nyní můžete vytvořit zálohu systémových souborů. Pokud již nemáte vlastní zálohu, doporučujeme vám tuto volbu využít. Záloha může být později velmi užitečná.
4. Vyberte rozsah aktualizace systému (např. 'Standardní systém'). Drobné nesrovnalosti můžete později upravit pomocí programu YaST.

## 4.2 Od verze k verzi

V následujících odstavcích bude popsáno, jaké detaily se změnilo od jedné verze k následující. V tomto přehledu bude např. uvedeno, zda se změnilo základní nastavení, zda došlo k přesunutí konfiguračních souborů na nové místo, nebo jestli se změnilo chování důležitých programů. Jsou zde uvedeny pouze věci, se kterými se uživatel resp. administrátor běžně setká. Tento seznam není v žádném případě úplný a vyčerpávající.

Problémy jednotlivých verzí jsou zveřejněny okamžitě po jejich odhalení. Důležité updaty jednotlivých balíčků najdete na stránce <http://www.novell.com/products/linuxprofessional/downloads/>. Jejich instalace provádí pomocí YaST Online Update (YOU) viz 2.3.2 na straně 38.

## 4.2.1 Změny z 8.1 na 8.2

Problémy a zvláštnosti: <http://portal.suse.com/sdb/cz/2003/03/bugs82.html>.

- 3D podpora karet s čipy nVidia (změna): `NVIDIA_GLX` a `NVIDIA_kernel` již nejsou součástí distribuce (včetně skriptů `switch2nvidia_glx`). Místo toho prosím použijte instalátor společnosti nVidia pro *Linux IA32*, který naleznete na <http://www.nvidia.com>. Následně pak použijte YaST pro aktivaci 3D podpory.
- Při nové instalaci bude použit místo `inetd` program `xinetd`. Konfigurační adresář je `/etc/xinetd.d`. Při aktualizaci zůstane zachován `inetd`.
- PostgreSQL je nyní k dispozici ve verzi 7.3. Při přechodu z verze 7.2.x doporučujeme `dump/restore` příkazem `pg_dump`. Pokud vaše aplikace přistupují k systémovým katalogům, pak je třeba provést ještě další úpravy, protože 7.3 již zavádí schémata. Podrobné informace naleznete na <http://www.ca.postgresql.org/docs/momjian/>
- PostgreSQL je nyní pouze ve verzi 7.3. pro přechod z verzí 7.2.x je určen `dump/restore` s příkazem `pg_dump`. Pokud vaše aplikace vyžaduje systémový katalog, musíte provést ještě další úpravy, kterými zavedete schéma verze 7.3. Více informací najdete na stránce [http://www.ca.postgresql.org/docs/momjian/upgrade\\_tips\\_7.3](http://www.ca.postgresql.org/docs/momjian/upgrade_tips_7.3)
- Verze 4 programu `stunnel` již nepodporuje na příkazové řádce žádné parametry. Je však poskytován spolu se skriptem `/usr/sbin/stunnel3_wrapper`, který parametry příkazové řádky pro `stunnel` dokáže konvertovat do konfiguračního souboru. Jeho použití je následující (položku *OPTIONS* nahraďte parametry):

```
/usr/sbin/stunnel3_wrapper stunnel OPTIONS
```

Konfigurační soubor se zároveň vypíše do standardního výstupu, aby bylo možné se seznámit se syntaxí pro zápis do trvalého konfiguračního souboru.

- `openjade` (`openjade`) je nyní DSSSL engine, který se používá místo `jade` (`jade_dsl`), když je spuštěn `db2x.sh` (`docbook-toys`). Z důvodů kompatibility jsou jednotlivé programy také bez předpony *o*.

Pokud je nějaká aplikace závislá na adresáři `jade_dsl` a tam umístěných souborech, pak je třeba buď ji přesměrovat na `/usr/share/sgml/openjade` nebo vytvořit odkaz (jako `root`):

```
cd /usr/share/sgml
rm jade_dsl
ln -s openjade jade_dsl
```

Abyste zabránili konfliktu s `rszsz`, jmenuje se příkaz `sx` i nadále `s2x`, resp. `sgml2xml` nebo `osx`.

## 4.2.2 Změny z 8.2 na 9.0

Problémy a zvláštnosti:

- Došlo ke změně verze správce balíků RPM na verzi 4. Nové balíky se nyní vytvářejí příkazem `rpmbuild`. Příkaz `rpm` je nadále používán pro instalaci, aktualizaci a dotazy.
- Pro nastavení *tisku* přibyl balík *footmatic-filters*. Obsah byl získán z balíku `cups-drivers`, aby bylo možné filtry používat i v případě, že není nainstalován CUPS. Díky tomu nyní lze prostřednictvím programu YaST získat nastavení nezávislé na tiskovém systému (CUPS, LPRng). Balík obsahuje konfigurační soubor `/etc/foomatic/filter.conf`.
- I při nasazení LPRng/lpdltru jsou nyní vyžadovány balíky `footmatic-filters` a `cups-drivers`.
- XML zdroje balíků jsou zpracovávány pomocí záznamů v `/etc/xml/suse-catalog.xml`. Tento soubor nesmí být změněn příkazem `xmlcatalog`, protože by mohlo dojít k přemazání komentářů nutných pro aktualizaci. Soubor `/etc/xml/suse-catalog.xml` je zpracován pomocí výrazu `nextCatalogv /etc/xml/catalog`, aby nástroje jako `xmllint` nebo `xsltproc` automaticky našli lokální zdroje.

## 4.2.3 Změny z 9.0 na 9.1

Problémy a zvláštnosti najdete popsané v článku v databázi instalační podpory na stránce <http://portal.suse.com>.

- SUSE LINUX používá jádro řady 2.6. Jádro řady 2.4 již není k dispozici a je možné, že pokud používáte programy, vyžadující starší jádro, tyto programy přestanou fungovat. Ze změnou jádra souvisí i následující změny:
  - ▷ Zavádění modulů se nyní nastavuje v souboru `/etc/modprobe.conf`. Soubor `/etc/modprobe.conf` se přestal používat. YaST dokáže do určité míry starý soubor převést (pomocí skriptu `/sbin/generate-modprobe.conf`).
  - ▷ Moduly mají nyní příponu `.ko`.
  - ▷ IDE vypalovačky již pro vypalování nepotřebují modul `ide-scsi`.
  - ▷ Z parametrů modulů ALSA byla odstraněna přímona `snd_`.
  - ▷ `/proc` byl nahrazen novým `sysfs`.
  - ▷ Správa napájení (především ACPI) lze nyní nastavit i prostřednictvím programu YaST.
- NGPT a linuxthreads

Programy linkované proti NGPT (*Next Generation POSIX Threading*) již s glibc 2.3.x nepoběží. Všechny takto postižené programy, které nejsou součástí distribuce SUSE LINUX musí být kompilovány s podporou linuxthreads nebo NPTL (*Native POSIX Thread Library*).

Problémy s NPTL mohou nastat také na systémech se starší implementací linuxthreads, pokud nenastavíte následující proměnnou prostředí (*kernel-version* nahraďte příslušnou verzí jádra):

```
LD_ASSUME_KERNEL=kernel-version
```

Možné jsou tyto verze:

- ▷ 2.2.5 (i386, s390): linuxthreads bez Floating Stacks
- ▷ 2.4.1 (AMD64, i586, i686): linuxthread s Floating Stacks

Poznámky k jádru a linuxthreads s *Floating Stacks*:

Programy používající `errno`, `h_errno` a `_res`, potřebují hlavičkové soubory (`errno.h`, `netdb.h` a `resolv.h`). C++ programy s podporou multithread, potřebují ke správnému chodu nastavit proměnnou prostředí `LD_ASSUME_KERNEL=2.4.1`.

NPTL (*Native POSIX Thread Library*) je v systému obsažena jako balíček Thread. NPTL slouží k zajištění binární kompatibility se starší knihovnou linuxthreads.



- Jako výchozí kódování je pro systémy použit standard *UTF-8*. Při instalaci se zadá také národní kódování ve formátu *NarodniKodovani.UTF-8* (např. *cs\_CZ.UTF-8*).
- Nástroje z balíku *coreutils* jako *tail*, *chown*, *head*, *sort* se řídí POSIX standardem z roku 2001 (*Single UNIX Specification, version 3 == IEEE Std 1003.1-2001 == ISO/IEC 9945:2002*) ale již ne standardem z roku 1992. Staré nastavení můžete získat pomocí proměnné prostředí:

```
_POSIX2_VERSION=199209
```

(Nové nastavení je *200112* a je převzato z *\_POSIX2\_VERSION*.)

SUSE standard je dostupný na stránce (zdarma po registraci) <http://www.unix.org/>

Současné nastavení:

**Tabulka 4.1:** Srovnání POSIX 1992 a POSIX 2001

POSIX 1992	POSIX 2001
<code>chown tux.users</code>	<code>chown tux:users</code>
<code>tail +3</code>	<code>tail -n 3</code>
<code>head -1</code>	<code>head -n 1</code>
<code>sort +3</code>	<code>sort -k 4</code>
<code>nice -10</code>	<code>nice -n 10</code>
<code>split -10</code>	<code>split -l 10</code>

### Tip

Software třetích stran se novým standardem ještě nemusí řídit. V takovém případě nastavte proměnnou prostředí takto: *\_POSIX2\_VERSION=199209*.

### Tip

- Soubor */etc/gshadow* byl odstraněn. Důvody pro tento krok jsou tyto:
  - ▷ Nemá žádnou podporu v *glibc*.

- ▷ Soubor nemá žádné oficiální rozhraní a propojení. Toto propojení nemá ani systém shadow.
- ▷ Většina aplikací kontrolujících heslo skupiny ignoruje tento soubor z výše uvedených důvodů.
- Podle FHS jsou nyní XML zdroje (DTD, Stylesheety atd.) nainstalované v adresáři `/usr/share/xml`. Z tohoto důvodu již tyto soubory nenajdete v adresáři `/usr/share/sgml`. V případě problémů je nutné vytvořit případný skript, upravit Makefile nebo tzv. oficiální katalogy (především `/etc/xml/catalog` popř. `/etc/sgml/catalog`).

#### 4.2.4 Změny z 9.1 na 9.2

Více informací získáte v anglickém článku *Known Problems and Special Features in SUSE LINUX 9.2* v databázi instalační podpory SUSE Support Database na stránce <http://portal.suse.com> po zadání klíčových slov *special features*.

#### Aktivace firewallu během instalace

Aby byla zvýšena bezpečnost systému, je na konci instalace v návrhu aktivován firewall SuSEFirewall2. Po spuštění firewallu jsou zavřeny všechny porty. Potřebné porty lze otevřít v dialogu návrhu.

V případě síťového přístupu během instalace příslušný modul programu YaST otevře potřebné TCP a UDP porty na interních i externích rozhraních. Pokud potřebujete jiné nastavení, proveďte je v modulu firewallu programu YaST po instalaci.

*Tabulka 4.2: Porty důležitých služeb*

Služba	Port
HTTP server	Firewall je nastaven podle konfigurace (pouze TCP)
Mail (postfix)	smtp 25/TCP
Samba server	netbios-ns 137/TCP; netbios-dgm 138/TCP; netbios-ssn 139/TCP; microsoft-ds 445/TCP
DHCP server	bootpc 68/TCP
DNS server	domain 53/TCP; domain 53/UDP

- " -	Plus zvláštní podpora pro port mapper v aplikaci SuSEFirewall2
Port mapper	sunrpc 111/TCP; sunrpc 111/UDP
NFS server	nfs 2049/TCP
- " -	Plus port mapper
NIS server	Aktivuje portmap
TFTP	tftp 69/TCP
CUPS (IPP)	ipp 631/TCP; ipp 631/UDP

---

## KDE a podpora IPv6

Ve výchozím nastavení KDE není podpora IPv6 povolena. Povolit ji můžete v modulu editor souborů `/etc/sysconfig` programu YaST. Důvod, proč není podpora IPv6 povolena, spočívá ve skutečnosti, že IPv6 adresy nejsou správně podporovány všemi poskytovateli internetového připojení. Chybná podpora může vést k chybovým hlášením a vysokým prodlevám při načítání stránek.

## YaST Online Update a "delta balíčky"

YaST Online Update nyní podporuje zvláštní druh RPM balíčků, které obsahují pouze odlišné části spustitelných souborů. Tato nové technologie výrazně zmenšuje velikost opravných balíčků a tím také délku jejich stahování. Nastavení potřebná k používání "delta balíčků" můžete provést v souboru `/etc/sysconfig/onlineupdate`. Podrobnější informace najdete v souboru `/usr/share/doc/packages/deltarpm/README`.

## Konfigurace tiskového systému

Na konci instalace (proposal dialog) je nutné na firewallu otevřít port pro tiskový systém. CUPS používá porty 631/TCP a 631/UDP. Pracovní stanice by měla mít tyto porty zavřené. V případě tisku přes LPD nebo SMB musí být otevřený také port 515/TCP (starý LPD protokol).

## Přechod na X.Org

Přechod z XFree86 na X.Org je zjednodušen kompatibilitou odkazy se starými jmény na nové důležité soubory a příkazy

**Tabulka 4.3: Příkazy**

<b>XFree86</b>	<b>X.Org</b>
XFree86	Xorg
xf86config	xorgconfig
xf86cfg	xorgcfg

**Tabulka 4.4: Soubory s logy v adresáři /var/log**

<b>XFree86</b>	<b>X.Org</b>
XFree86.0.log	Xorg.0.log
XFree86.0.log.old	Xorg.0.log.old

Při přechodu na X.Org bal samozřejmě balíček XFree86\* změněn na xorg-x11\*.

### **Emulátory terminálu pro X11**

Vyřadili jsme z distribuce řadu emulátorů, protože nejsou již spravované nebo jsou nefunkční ve výchozím prostředí např. nepodporují UTF-8. SUSE LINUX nabízí standardní terminály jako xterm, terminály prostředí KDE a GNOME a ml-term (Multilingual Terminal Emulator for X), který může být náhradou za aterm a eterm.

### **Změny v balíčku powersave**

Došlo ke změně konfiguračních souborů v /etc/sysconfig/powersave:

**Tabulka 4.5: Splynutí konfiguračních souborů do /etc/sysconfig/powersave**

<b>Staré</b>	<b>Součástí</b>
/etc/sysconfig/powersave/common	common
	cpufreq

```
events
battery
sleep
thermal
```

---

Soubor `/etc/powersave.conf` zastaral. Existující proměnné byly přesunuty do souborů v tabulce 4.5 na předchozí straně. Pokud jste měnili *events* proměnné v `/etc/powersave.conf`, musíte nyní provést v `/etc/sysconfig/powersave/events`.

Stavy uspání se změnilly z:

- uspat (ACPI S4, APM suspend)
- standby (ACPI S3, APM standby)

na:

- uspat na disk (ACPI S4, APM suspend)
- uspat do ram (ACPI S3, APM suspend)
- standby (ACPI S1, APM standby)

## OpenOffice.org (OOo)

**Cesty:** OOo se nyní instaluje místo do adresáře `/opt/OpenOffice.org` do adresáře `/usr/lib/ooo-1.1`. Výchozí adresář pro uživatelská nastavení je `~/.ooo-1.1` místo původního `~/OpenOffice.org1.1`.

**Wrapper:** Některé OOo komponenty jsou spouštěny novými wrappery. Nová jména jsou uvedena v následující tabulce 4.6 na této straně.

*Tabulka 4.6: Wrapper*

Starý	Nový
<code>/usr/X11R6/bin/OOo-calc</code>	<code>/usr/bin/oocalc</code>
<code>/usr/X11R6/bin/OOo-draw</code>	<code>/usr/bin/oodraw</code>

/usr/X11R6/bin/OOo-impress	/usr/bin/ooimpress
/usr/X11R6/bin/OOo-math	/usr/bin/oomath
/usr/X11R6/bin/OOo-padmin	/usr/sbin/oopadmin
/usr/X11R6/bin/OOo-setup	-
/usr/X11R6/bin/OOo-template	/usr/bin/oofromtemplate
/usr/X11R6/bin/OOo-web	/usr/bin/ooweb
/usr/X11R6/bin/OOo-writer	/usr/bin/oowriter
/usr/X11R6/bin/OOo	/usr/bin/ooffice
/usr/X11R6/bin/OOo-wrapper	/usr/bin/ooo-wrapper

---

Wrapper nyní podporuje volbu `--icons-set` pro přepnutí ikon mezi KDE a GNOME. Následující volby již nejsou podporovány:  
`--default-configuration`, `--gui`, `--java-path`, `--skip-check`,  
`--lang` (jazyk je nastaven podle locales), `--messages-in-window` a  
`--quiet`.

**Podpora KDE a GNOME:** Rozšíření pro KDE a GNOME jsou dostupné v balíčcích `OpenOffice_org-kde` a `OpenOffice_org-gnome`.

### Zvukový směšovač kmix

Jako výchozí zvukový směšovač je nastaven kmix. Pro high-end hardware jsou dostupné starší směšovače jako QAMix/KAMix, `envy24control` (pouze ICE1712) nebo `hdspmixer` (pouze RME Hammerfall).

## 4.2.5 Změny z 9.2 na 9.3

Více informací získáte v anglickém článku *Known Problems and Special Features in SUSE LINUX 9.3* v databázi instalační podpory SUSE na stránce <http://portal.suse.com> po zadání klíčových slov *special features*.

### Spuštění ruční instalace s promptu jádra

Instalační nabídka již neobsahuje položku 'Manual Installation' (ruční instalace). Ruční instalaci v linuxrc spustíte zadáním parametru `manual=1`. Tento způsob instalace není ve většině již nutný. Instalační parametry jako např. instalační zdroj můžete zadat přímo jako parametr.

## Síťové ověřování a Kerberos

Místo aplikace `heimdal` je nyní výchozí pro síťové ověřování systém Kerberos. Existující `heimdal` konfiguraci nelze automaticky převést. Během aktualizace systému se vytvoří záložní soubory uvedené v tabulce 4.7 na této straně.

*Tabulka 4.7: Záložní soubory*

Starý soubor	Záložní soubor
<code>/etc/krb5.conf</code>	<code>/etc/krb5.conf.heimdal</code>
<code>/etc/krb5.keytab</code>	<code>/etc/krb5.keytab.heimdal</code>

Klientská konfigurace (`/etc/krb5.conf`) je velmi podobná nastavení `heimdal`. Pokud jste neprováděli žádné zvláštní nastavení, stačí položku `kpasswd_server` nahradit za `admin_server`.

Data serveru (`kdc/kadmind`) převzít nelze. Po aktualizaci je `heimdal` databáze stále dostupná v `/var/heimdal`; MIT kerberos databáze se nachází v `/var/lib/kerberos/krb5kdc`.

## Konfigurační soubor X.Org Configuration File

SaX2, nástroj pro nastavení grafického prostředí, zapisuje konfiguraci X.Org do souboru `/etc/X11/xorg.conf`. Při čisté instalaci se již nevytvoří symbolický odkaz na `XF86Config`.

## Konfigurace PAM

**common-auth** výchozí nastavení PAM sekce `auth`

**common-account** výchozí nastavení PAM sekce `account`

**common-password** výchozí nastavení PAM pro změnu hesla

**common-session** výchozí nastavení PAM pro správu sezení

Protože je jednodušší tímto způsobem upravovat a spravovat nastavení, měli byste do těchto souborů přesunout také nastavení ze svých aplikací. Pokud pak některou z aplikací nainstalujete později, automaticky se aplikují již provedené změny a správce systému již nemusí nastavení provádět ručně.

Změny jsou jednoduché. Pokud máte následující konfigurační soubor (výchozí pro většinu aplikací):

```

#%PAM-1.0
auth      required      pam_unix2.so
account   required      pam_unix2.so
password  required      pam_pwcheck.so
password  required      pam_unix2.so      use_first_pass use_authtok
#password required      pam_make.so      /var/yp
session   required      pam_unix2.so

```

změňte ho takto:

```

#%PAM-1.0
auth      include      common-auth
account   include      common-account
password  include      common-password
session   include      common-session

```

## 4.3 RPM the Package Manager

Distribuce SUSE LINUX používá RPM. Databáze RPM poskytuje detailní informace o nainstalovaných balících a tím usnadňuje práci uživatelům, systémovým administrátorům a v neposlední řadě i tvůrcům balíků. Hlavní příkazy systému RPM jsou `rpm` a `rpmbuild`.

`rpm` funguje v pěti módech:

- Nainstaluje, aktualizuje a beze zbytku odinstaluje balíky ve formátu RPM.
- Umožňuje dotazy ohledně balíků, včetně závislostí a spravuje databázi instalovaných RPM balíků.
- Přestaví v případě potřeby RPM databázi.
- Překontroluje integritu balíky.
- Podepisuje RPM balíky.

Příkaz `rpmbuild` aplikaci přeloží ze zdrojových kódů a zabalí je pro instalaci. Archivy RPM jsou zabalené ve speciálním binárním formátu. Skládají se ze souborů k instalaci a různých meta informací, které `rpm` používá během instalace pro konfiguraci stávajících softwarových balíků nebo je uloží do databáze RPM za účelem dokumentace. Archivy RPM mají zpravidla příponu `.rpm`. Aplikace `rpm` může spravovat balíky kompatibilní s LSB.



**Tip**

Pro vývoj softwaru je potřeba řada komponent (knihovny, hlavičkové soubory atd.), které jsou umístěny v samostatných balících. Tyto balíky jsou potřebné pouze pro vývoj a nijak neovlivňují běžný chod systému. Poznáte je podle toho, že ve jménu balíčku obsahují `-devel` např. `alsa-devel`, `gimp-devel` a `kdelibs-devel`.

**Tip**

### 4.3.1 Ověření balíku

RPM balíky SUSE podepisovány pomocí GnuPG:

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing
Key <build@suse.de> Key fingerprint = 79C1 79B2 E1C8
20C1 890F 9994 A84E DAE8 9C80 0ACA
```

```
rpm --verbose --checksig apache-1.3.12.rpm
```

je možné zkontrolovat signaturu rpm balíku a tak určit, zda balík pochází opravdu od SUSE nebo z jiného důvěryhodného zdroje. Toto je vhodné zvláště pro balíky, které si stahujete z Internetu. Náš veřejný klíč je standardně uložen v `/root/.gnupg/`.

### 4.3.2 Správa balíků -- instalace, aktualizace a smazání

V běžném případě je instalace balíků RPM velice jednoduchá:

```
rpm -i JmenoBaliku.rpm
```

Pomocí tohoto standardního příkazu bude balík nainstalován pouze v případě, že jsou v pořádku závislosti a že nedojde k žádným konfliktům. Při ohlášení chyby vyhledá rpm chybějící závislosti, resp. balíky. Databáze RPM zajišťuje, aby nedošlo ke konfliktům -- je pravidlem, že určitý soubor patří vždy jen do jednoho balíku. Zadáním voleb lze přinutit rpm, aby to ignoroval, ale pak je třeba přesně vědět, co děláme, aby nedošlo k ohrožení možnosti aktualizovat systém.

Volba `-U` resp. `--upgrade` je určena pro aktualizaci balíků. Pomocí ní je možné smazat starší verzi stejných balíků a nainstalovat novější verzi. Zároveň se rpm opatrně pokouší editovat konfigurační soubory následujícím způsobem:

- Pokud nebyl konfigurační soubor změněn systémovým administrátorem, pak `rpm` nainstaluje odpovídajícím způsobem novou verzi instalovaného souboru. Není třeba žádných zásahů administrátora.
- Pokud *před aktualizací* došlo ke změně konfiguračního souboru, RPM bude instalovaný soubor zálohovat s příponou `.rpmorig` nebo `.rpmsave` -- avšak pouze pokud se instalovaný soubor a nová verze liší. Tehdy je třeba upravit nové konfigurační soubory podle záložních kopií (`.rpmorig` nebo `.rpmsave`). Potom by měly být tyto záložní kopie okamžitě odstraněny, aby nebránily budoucí aktualizaci. Přípona `.rpmorig` se používá, když databáze RPM soubor nezná, v opačném případě se použije `.rpmsave`. Jinak řečeno, `.rpmorig` se používá pro aktualizaci z cizího formátu na RPM a `.rpmsave` při aktualizaci ze staršího RPM na novější RPM verzi.

Aktualizace s volbou `-U` *není* pouhou náhradou za odinstalování pomocí `-e` a následnou instalaci pomocí `-i`. Pokud je to možné, dávejte vždy přednost volbě `-U`.

Po každé aktualizaci je třeba zkontrolovat záložní kopie s příponou `.rpmorig` nebo `.rpmsave` -- jsou to staré konfigurační soubory. Pokud je to možné, převezměte vaše úpravy ze starých souborů do nových a potom záložní kopie (`.rpmorig` resp. `.rpmsave`) smažte.

Budete-li chtít odinstalovat balík, zadejte:

```
rpm -e JmenoBaliku
```

Příkaz `rpm` však odstraní balík pouze pokud nenajde žádné závislosti. Proto není například teoreticky možné smazat `Tcl/Tk` tak dlouho, dokud ho bude ke svému běhu využívat některý z dalších programů -- RPM to hlídá s pomocí své databáze.

Pokud ve výjimečném případě nelze balík odstranit, přestože *žádné* závislosti neexistují, může pomoci aktualizovat databázi RPM volbou `--rebuilddb`.

### 4.3.3 RPM a opravy

Aby byl systém vždy naprosto bezpečný, je nutné pravidelně aplikovat opravy. Dříve bylo možné chybu v programu odstranit pouze současným přepisem celého RPM balíku. I při celkem malé chybě, která se týkala jediného souboru, bylo nutné balík kompletně přepsat. Od verze SUSE 8.1 umožňuje SUSE instalovat do balíku jen nové funkce a opravy bez nutnosti kompletního přepisu.

Výhody si můžeme demonstrovat na programu pine:

**Je RPM určena pro váš systém?** abyste dokázali na tuto otázku odpovědět, musíte zjistit verzi nainstalovaného balíku. Pro program pine to provedete příkazem `rpm -q pine`.

Zda je opravné RPM určené pro verzi vašeho programu pine zjistíte příkazem:

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
pine = 4.44-195
pine =
```

Tato oprava je určena pro tři různé verze programu pine. Jedna z verzí se shoduje s naší nainstalovanou verzí, takže oprava je určena i pro náš případ.

**Jaké soubory oprava přepíše?** Soubory, které budou přepisovány zjistíte v RPM opravu. Použijte příkaz:

```
rpm -qp1 pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
```

Pokud jste již opravu nainstalovali a chcete informaci získat z již nainstalovaného systému, zadejte:

```
rpm -qP1 pine
/etc/pine.conf
/etc/pine.conf.fixed
```

Odpovídající výstup je v příkladu hned pod příkazem.

**Jak opravné RPM nainstalovat?** S opravným RPM se pracuje jako s každým obyčejným RPM balíkem. Jediný rozdíl spočívá v tom, že již na systému musíte mít nainstalovaný balík, pro který je oprava určena.

**Jaké opravy jsou již nainstalovány a pro jaké verze balíčků?**

Seznam již nainstalovaných oprav zobrazíte příkazem `rpm -qPa`. Pokud je jako v našem příkladu nainstalovaný pouze jeden opravný RPM, bude seznam vypadat takto:

```
rpm -qPa
```

Na déle běžícím systému s řadou oprav a aktualizací budete možná potřebovat zjistit, jaká verze byla původně nainstalována. I tato informace se dá z RPM databáze získat. Například pro program `pine` příkazem `rpm -q --basedon pine`.

Více informací o opravných RPM najdete v manuálových stránkách `rpm` a `rpmbuild`.

### 4.3.4 Delta RPM balíčky

„Delta RPM“ obsahují rozdíl (to je „delta“) mezi starou a novou verzí RPM balíčku. Aplikací delta balíčku na starý RPM balíček získáte zcela nový balíček. Pokud nemáte kopii balíčku můžete delta RPM aplikovat i na již nainstalovaný balíček. Velkou výhodou delta RPM balíčků je jejich menší velikost oproti standardním opravným balíčků. Menší velikost totiž znamená mnohem rychlejší stažení z internetu. Nevýhodou je vyšší zatížení systému při jejich aplikaci. Pokud chcete delta balíčky používat při online update, nastavte v souboru `/etc/sysconfig/onlineupdate` proměnnou `YOU_USE_DELTAS` na hodnotu „yes“.

Hlavními příkazy pro práci s delta balíčky jsou `prepdeltarpm`, `writedeltarpm` a `applydeltarpm`. Následujícím příkazem vytvoříte delta RPM balíček nazvaný `new.delta.rpm` (z balíčku `stary.rpm` a novy `.rpm`):

```
prepdeltarpm -s seq -i info old.rpm > old.cpio
prepdeltarpm -f new.rpm > new.cpio
```

```
xdelta delta -0 old.cpio new.cpio delta
```

```
writedeltarpm new.rpm delta info new.delta.rpm
rm old.cpio new.cpio delta
```

Příkazem `applydeltarpm` můžete, pokud máte starou verzi balíčku nainstalovanou v systému, získat novou verzi RPM balíčku:

```
applydeltarpm new.delta.rpm new.rpm
```

Jestliže nechcete přistupovat k souborovému systému, použijte pro vytvoření volbu `-r`:

```
applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

Technické podrobnosti najdete v souboru `file:///usr/share/doc/packages/deltarpm/README`.

### 4.3.5 Zadání dotazu

Pomocí volby `-q` je možné zadat dotaz a prohlédnout si tak archiv RPM (volba `-p JmenoBaliku`) nebo se dotázat databáze RPM na instalované balíky. Druh požadovaných informací se zadá přepínači v tabulce 4.8 na této straně.

*Tabulka 4.8: Nejdůležitější volby při RPM dotazování*

<code>-i</code>	Zobrazit informace o balíku
<code>-l</code>	Zobrazit seznam souborů
<code>-f CeleJmenoSouboru</code>	Dotaz na balík obsahující soubor vypasný s úplnou cestou
<code>-s</code>	Zobrazit stavové informace (implicitně <code>-l</code> )
<code>-d</code>	Seznam dokumentačních souborů (implicitně <code>-l</code> )
<code>-c</code>	Seznam konfiguračních souborů (implicitně <code>-l</code> )
<code>--dump</code>	Zobrazit detailní informace o souboru (použít s <code>-l</code> , <code>-c</code> nebo <code>-d</code> !)
<code>--provides</code>	Seznam virtuálních balíčků, které tento balík poskytuje
<code>--requires, -R</code>	Seznam balíčků, virtuálních balíčků a souborů, které tento balík vyžaduje
<code>--scripts</code>	Zobrazit skripty pro instalaci a deinstalaci

**Příkaz:**

```
rpm -q -i rpm
```

```

Name       : rpm                      Relocations: (not relocateable)
Version    : 3.0.3                    Vendor: SUSE GmbH, Germany
Release    : 47                       Build Date: Fri Dec 10 13:50:27
Install date: Tue Dec 14 12:57 1999   Build Host: Cauchy.suse.de
Group      : unsorted                 Source RPM: rpm-3.0.3-47.src.rpm
Size       : 5740847                  License: GPL
Packager   : feedback@suse.de
Summary    : RPM Package Manager

```

```
Description :
RPM Package Manager is the main tool for managing software packages
of the SUSE Linux distribution.
[...]
```

Volba `-f` je funkční pouze v případě, že znáte kompletní název souboru včetně cesty. Může být zadán libovolný počet hledaných souborů, např.:

```
rpm -q -f /bin/rpm /usr/bin/wget
```

vede k tomuto výsledku:

```
rpm-3.0.3-3
wget-1.5.3-55
```

Pokud znáte pouze část názvu souboru, musíte si pomoci skriptem příkazového interpretu. Hledaný soubor se zadává při volání tohoto skriptu jako parametr. Použít můžete např. následující skript:

```
#!/bin/sh
for i in `rpm -q -a -l | grep $1 `; do
    echo "\b\slash"$i\b\slash" je v balíku:"
    rpm -q -f $i
    echo ""
done
```

Příkazem `rpm -q --changelog rpm` můžete zobrazit žádaný seznam informací (aktualizace, konfigurace, změny, atd.) o jednotlivých balících, např. o balíku *rpm*.

Pomocí databáze RPM je možné provádět kontroly. Ty je možné provádět volbou `-V` (stejný význam jako `-y` nebo `--verify`). Pomocí této volby zobrazí program `rpm` všechny soubory v balíku, u kterých došlo ke změně, v porovnání s originálem balíku. Program `rpm` používá osm různých znaků na označení nalezených změn v jednotlivých souborech:

**Tabulka 4.9:** Příznaky druhů změn souboru

---

5

kontrolní součet MD5

S	velikost souboru
L	symbolický odkaz
T	čas změny
D	major a minor číslo zařízení
U	uživatel
G	skupina
M	mód (přístupová práva a typ)

---

Tyto znaky se navzájem kombinují v řetězec. U konfiguračních souborů se navíc zobrazí znak `c`. Pokud například změníte `/etc/wgetrc`, který obsahuje `wget`, dostanete:

```
rpm -V wget
S.5....T c
```

Soubory databáze RPM jsou v adresáři `/var/lib/rpm`. Při velikosti stromu `/usr` kolem 1 GB může databáze zabírat kolem 30 MB -- zvláště po kompletní aktualizaci. Pokud vám bude připadat, že se databáze příliš rozrostla, lze ji obnovit pomocí volby `--rebuilddb`. Hodí se předtím zálohovat (samozřejmě někam jinam) stávající databázi.

Kromě toho vytváří skript `cron.daily` každý den zabalené kopie databáze v `/var/adm/backup/rpmdb`. Počet kopií určuje `MAX\_RPMDB\_BACKUPS` v `/etc/sysconfig/cron` (standardní počet je 5).

Je zde třeba počítat až s 3 MB pro každou zálohu (při 1 GB velkém `/usr`). To je třeba brát v úvahu při vytváření kořenového diskového oddílu. Pokud má `/var` zvláštní diskový oddíl, je třeba toto zohlednit při vytváření oddílu `/var`.

### 4.3.6 Instalace a překlad zdrojových balíčků

Všechny zdrojové kódy distribuce SUSE Linuxu mají příponu `.spm` -- jde o tzv. zdrojová RPM.

## Tip

Zdrojové balíky dokáže nainstalovat i YaST, avšak nejsou pak označeny jako ostatní řádné balíky, neboť v databázi RPM je pouze *spustitelný* software, což zdrojové kódy nejsou.

## Tip

V `/usr/src/packages` musí existovat následující pracovní adresáře pro rpm (pokud jste neprovedli žádná vlastní nastavení, např. v `/etc/rpmrc`):

**SOURCES** pro soubory `.tar.gz` atd., obsahující originální zdrojové kódy, a pro soubory `.dif`, obsahující úpravy specifické pro danou distribuci.

**SPECS** pro soubory `.spec`, které kontrolují proces sestavení binárního balíku

**BUILD** kde se zdrojové kódy rozbalují, upravují a překládají

**RPMS** kde se ukládají hotové binární balíky

**SRPMS** kde jsou zdrojové balíky

Pokud použijete pro instalaci zdrojového balíku YaST, komponenty potřebné pro sestavovací proces se nainstalují do `/usr/src/packages`. Zdrojový kód a úpravy do se nainstalují do adresáře **SOURCES** a odpovídající soubor `.spec` do **SPECS**.

## Důležité

Prosím nedělejte pomocí RPM žádné experimenty s důležitými systémovými součástmi jako jsou `glibc`, `rpm`, `sysvinit` atd. Riskujete tím ztrátu funkčnosti vašeho systému.

## Důležité

Pro náš následující příklad vybereme balík `wget .spm`. Poté, co se zdrojový balík `wget .spm` nainstaluje, by měly vzniknout například následující soubory:

- `/usr/src/packages/SPECS/wget.spec`
- `/usr/src/packages/SOURCES/wget-1.4.5.dif`
- `/usr/src/packages/SOURCES/wget-1.4.5.tar.gz`



Příkazem `rpmbuild -b X /usr/src/packages/SPECS/wget.spec` se spustí překlad. Proměnná `X` označuje různé stupně pokročilosti instalace. Jednotlivé možnosti nám podá např. `rpmbuild --help` nebo dokumentace k RPM. Hlavní z nich jsou:

- bp** Příprava zdrojového kódu v adresáři `/usr/src/packages/BUILD` -- rozbalení a úpravy
- bc** totéž jako `-bp`, navíc s překladem
- bi** totéž jako `-bc`, navíc s instalací. (Pozor -- pokud instalovaný balík nepodporuje BuildRoot, může dojít během instalace k přepisu konfiguračních souborů!)
- bb** totéž jako `-bi`, navíc s vytvořením tzv. binárního RPM. Po úspěšném překladu bude v `/usr/src/packages/RPMS`.
- ba** totéž jako `-bb`, navíc s vytvořením tzv. zdrojového RPM. Po úspěšném překladu bude v `/usr/src/packages/SRPMS`.

Pokud společně s `-bc` (resp. `-bi`) zadáte volbu `--short-circuit`, `rpm` vykoná pouze překlad, resp. instalaci, bez předchozích fází. S pomocí tohoto příkazu je tedy možné přeskočit určité kroky.

Vytvořené binární RPM se instaluje pomocí `rpm -i` nebo lépe `rpm -U`, aby došlo k zápisu do databáze RPM.

### 4.3.7 Další nástroje pro práci s archivy a databází RPM

Program Midnight Commander dokáže procházet archiv RPM a pracovat s jeho součástmi. Zachází přitom s balíkem RPM, jakoby se jednalo o souborový systém. Při používání `mc` můžete zobrazit informace obsažené v záhlaví (přístupném zde jako soubor `HEADER`) klávesou `(F3)` a kopírovat části archivu klávesou `(F5)`.

`xrpm` je název grafického správce balíčků RPM, který je napsaný v Pythonu a podporuje příkazy pro přístup přes FTP.

KDE obsahuje nástroj `kpackage`, což je grafické rozhraní pro obsluhu různých formátů balíčků, včetně RPM. GNOME obsahuje podobný nástroj `gnorpm`.



# Oprava systému

Kromě mnoha modulů pro instalaci a nastavení SUSE Linuxu, nabízí YaST také možnost opravy nainstalovaného systému. Tato kapitola popisuje postup takové opravy. YaST Oprava systému poskytuje přístup k oddílům a umožňuje zkušenému uživateli opravit poškozený systém.

5.1	Automatická oprava . . . . .	132
5.2	Vlastní nastavení . . . . .	134
5.3	Expertní nástroje . . . . .	134
5.4	Záchranný systém SUSE . . . . .	135

Protože se předpokládá, že poškozený systém nelze spustit, a protože oprava běžícího systému je obtížná, spouští se YaST Oprava systému z instalačního CD nebo DVD. Postupujte stejně jako při instalaci systému popsané v kapitole 1 na straně 3, ale v dialogu nabízejícím typy instalace vyberte 'Opravit nainstalovaný systém'.

---

## Důležité

### Použití správného instalačního média

Použité instalační médium by mělo *přesně* odpovídat nainstalovanému systému, který chcete opravit.

---

## Důležité

V dalším kroku vyberte způsob opravy: automatickou opravu, použití vlastního nastavení a nebo expertní nástroje.

## 5.1 Automatická oprava

Tento způsob opravy je nejlepší v případě, že neznáte příčinu nefunkčnosti systému. Jeho výběrem spustíte podrobnou analýzu nainstalovaného systému, která zabere vzhledem k velkému množství testů poměrně hodně času. Postup je znázorněn v dolní části obrazovky pomocí dvou ukazatelů průběhu. Horní ukazatel znázorňuje průběh právě běžícího testu, dolní průběh celé analýzy. Okno se záznamy (logy) v horní části obrazovky zobrazuje informace o probíhajících testech a jejich výsledky. Viz 5.1 na následující straně. Následující hlavní testy jsou prováděny vždy. Každý obsahuje množství podtestů.

### Kontrola tabulky oddílů na všech discích

Zkontroluje platnost a konzistenci tabulek oddílů na všech rozpoznaných pevných discích.

**Kontrola odkládacího oddílu** Jsou vyhledány a otestovány odkládací oddíly. V případě potřeby je nabídnuta jejich aktivace. Nabídka by v zájmu zrychlení opravy systému měla být přijata.

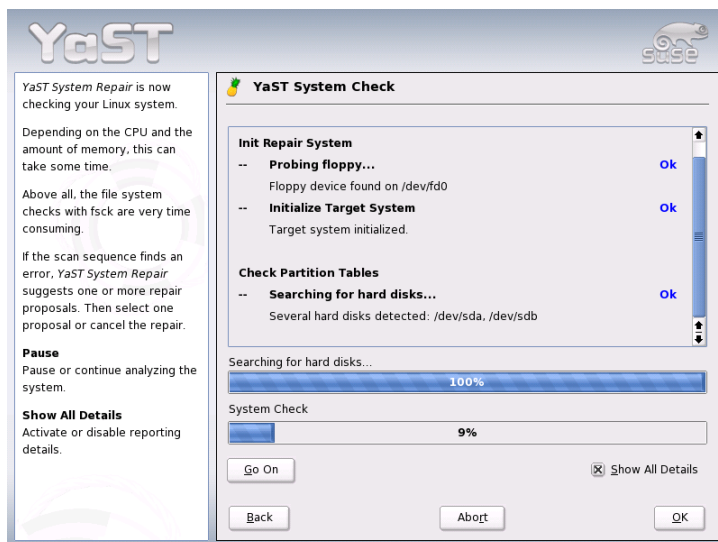
**Kontrola souborového systému** Všechny rozpoznané souborové systémy jsou podrobeny specifickému testu.

### Kontrola položek souboru `/etc/fstab`

Kontrola úplnosti a konzistence položek v souboru `/etc/fstab`. Připojení všech platných oddílů.

**Kontrola konfigurace zavaděče** Kontrola úplnosti a konzistence konfigurace zavaděče (GRUB nebo LILO). Testována jsou také zařízení se startovacím adresářem a kořenovým systémem spolu s dostupností modulů initrd.

**Kontrola databáze balíčků** Kontrola balíčků z výběru minimální instalace, které zajišťují funkčnost minimálního systému. Pokud chcete, můžete překontrolovat také všechny základní balíčky. To však může trvat delší čas.



*Obrázek 5.1: Automatický režim opravy*

Při nalezení chyby se testování zastaví a zobrazí se dialog s informacemi o nalezené chybě a nabídkami možných řešení. Není možné zde popsat všechny možnosti. Přečtěte si pozorně informace na obrazovce a vyberte z nabídky požadovanou akci. V případě pochybností můžete opravu odmítnout. V takovém případě zůstane systém beze změn. Žádná oprava není provedena automaticky bez potvrzení uživatelem.

## 5.2 Vlastní nastavení

Automatická oprava popsaná v předchozí části provádí všechny dostupné testy systému. Je to užitečné, pokud neznáte rozsah poškození. Vlastní nastavení je naopak nejvhodnější pro případ, kdy víte, ve které části došlo k chybě. Pokud zvolíte ‘Vlastní nastavení’, zobrazí se seznam testů. Všechny dohromady odpovídají automatické opravě. V případě vlastního nastavení ale můžete deaktivovat testy týkající se oblastí, které jsou v pořádku. Kliknutím na ‘Další’ spustíte zaškrtnuté (aktivované) testy.

Samostatně nejsou dostupné všechny typy testů. Test souboru `/etc/fstab` je například vždy spojen s kontrolou souborového systému včetně oddílu swap. YaST tyto závislosti řeší automaticky.

## 5.3 Expertní nástroje

Pokud SUSE LINUX velmi dobře znáte a máte přesnou představu o tom, co je třeba v systému opravit, můžete přímo používat potřebné nástroje volbou ‘Expertní nástroje’. K dispozici jsou následující nástroje:

**Instalovat nový zavaděč** Touto volbou spustíte modul nastavení zavaděče. Další informace jsou uvedeny v části 8.4 na straně 176.

### **Spustit nástroj pro rozdělování disku**

Výběrem této volby spustíte modul pro dělení disku. Podrobnosti najdete v části 2.8.9 na straně 63.

**Opravit souborový systém** Pomocí této volby spustíte nástroj pro kontrolu souborového systému. Můžete nechat překontrolovat všechny oddíly nebo zadat jen jeden vybraný.

**Obnovení ztracených oddílů** Je možné pokusit se o rekonstrukci poškozených tabulek oddílů. Nejprve se zobrazí seznam rozpoznaných pevných disků. Kliknutím na ‘Spustit’ zahájíte analýzu. Může to trvat poměrně dlouho, v závislosti na výkonu počítače a velikosti pevného disku.

## Varování

### Rekonstrukce tabulky oddílů

Obnovení tabulky oddílů je složitá operace založena na analýze obsahu disku. Po úspěšném rozpoznání oddílů jsou nalezené obnovené oddíly vloženy do přestavěné tabulky disků. Obnovení oddílů nemusí být úspěšné ve všech případech.

## Varování

### Uložit systémové nastavení na disketu

Tato volba uloží důležité systémové soubory na disketu. V případě poškození těchto souborů je pak bude možné z diskety obnovit.

**Ověřit instalované programy** Tímto se ověří konzistence databáze balíčků a dostupnost nejdůležitějších balíčků. Poškozené nainstalované balíčky lze pomocí tohoto nástroje přeinstalovat.

## 5.4 Záchranný systém SUSE

SUSE LINUX obsahuje záchranný systém nezávislý na instalačním médiu, kdy se v případě nouze můžete *zvenčí* dostat ke všem svým linuxovým oddílům. Záchranný systém může být načítán ze sítě, CD či dokonce ze SUSE FTP serveru. Dokonce i z CD bootovatelný SUSE LINUX *LiveEval* CD můžete použít jako svůj záchranný systém. Záchranný systém zahrnuje některé pomocné programy určené k odstranění následků systémových katastrof, jako bývají nedostupné disky, nekonzistentní konfigurační soubory, atd.

Jedním z nástrojů záchranného systému je aplikace Parted, kterou používáme pro změny velikosti oddílů, když nechceme použít k dané úpravě oddílů příslušný modul YaSTu. Více informací o programu Parted naleznete na <http://www.gnu.org/software/parted/>.

### 5.4.1 Spouštění záchranného systému

Vložte první CD či DVD SUSE LINUXu do mechaniky, ze které bude systém nabíhat. Zapněte počítač. Vyberte z instalační nabídky položku 'Rescue System' ('Záchranný systém'). Záchranný systém se rozbálí, načte do RAM disku jako nový kořenový souborový systém, připojí se a spustí, přičemž tento postup je nezávislý na použitém médiu. Po těchto fázích je připraven k použití.

## 5.4.2 Práce v záchranném systému

V záchranném systému jsou k dispozici pod klávesovými zkratkami **(Alt) + (F1)** až **(Alt) + (F3)** tři virtuální konzole. Je možné se přihlásit bez hesla jako `root`. Pro zobrazení zpráv jádra a programu `syslog` na tzv. systémové konzoli použijte kombinaci **(Alt) + (F10)**.

V adresáři `/bin` naleznete množství užitečných shellových nástrojů. Je mezi nimi i program `mount`. Adresář `sbin` obsahuje také důležité souborové a síťové nástroje pro diagnostiku a opravy souborového systému. (Např., `e2fsck`). V adresáři jsou také nejdůležitější binární soubory sloužící k údržbě systému, jako jsou `fdisk`, `mkfs`, `mkswap`, `mount`, `mount`, `init` a `shutdown`, spolu s `ifconfig`, `route` a `netstat`, které se Vám jistě budouhodit při údržbě sítě. Adresář `/usr/bin` obsahuje vi editor, `grep`, `find`, `less`, a `telnet`.

### Přístup do normálního systému

K připojení systémů SUSE LINUX pomocí záchranného systému použijte adresář - přípojný bod `/mnt`. Můžete ovšem použít i jiný adresář, či si nějaký jiný vytvořit. Následující příklad demonstuje použití záchrany pro soubor `/etc/fstab` s následujícím obsahem:

<code>/dev/sdb5</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>/dev/sdb3</code>	<code>/</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>1</code>
<code>/dev/sdb6</code>	<code>/usr</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>2</code>

### Varování

Dbejte na pozorné dodržení pořadí kroků popisovaných v následující sekci, zvláště těch které se týkají připojování různých zařízení.

### Varování

Přístup k celému systému si zajistíte připojením systémů do adresáře `/mnt` při použití následujících příkazů:

```
mount /dev/sdb3 /mnt
mount /dev/sdb6 /mnt/usr
```

Nyní máte zajištěn přístup do celého systému a můžete např. opravit chyby v konfiguračních souborech, jako jsou chyby v `/etc/fstab`, `/etc/passwd`



a `/etc/inittab`. Konfigurační soubory naleznete v připojeném adresáři `/mnt/etc`, což je původně nedostupný `/etc`.

*Dříve než začnete obnovovat ztracené oddíly pomocí programu `fdisk` jednoduše tím, že si příslušné soubory začnete znovu nastavovat, vytiskněte si, nebo nako-pírujte znění souboru `/etc/fstab` a výsledek příkazu `fdisk -l`.*

## Oprava systémových souborů

Poškozené souborové systémy představují pro záchranný systém choulostivý problém. Obecně - tyto souborové systémy není možné opravit na stávajícím systému. V případě, že se vyskytnou skutečně závažné problémy, je možné, že se Vám dokonce nepodaří připojit kořenový souborový systém a spuštění systému končí hláškou:

```
kernel panic
```

Pak nezbývá než systém opravit *zvenčí* za použití záchranného systému.

V SUSE LINUX záchranném systému můžete najít programy `e2fsck` a `dumpe2fs` (který se používá jako diagnostický nástroj). Tyto programy by měly pomoci s většinou problémů. Když se vyskytnou větší problémy, nebývají mnohdy dostupné tolik potřebné manuálové stránky. Z tohoto důvodu je zahrnujeme do příručky, najdete je v apendixu C na straně 636.

Stane-li se, že souborový systém padne z důvodů *neplatného* superbloku, program `e2fsck` selže s velkou pravděpodobností také. Problém může být způsoben porušením samotného superbloku. Kopie superbloku se nacházejí každých 8192 bloků (tedy jde o bloky 8193, 16385, atd.) Jestliže máte zničený superblok použijte jednu z těchto kopií. Zajistí to např. příkaz `e2fsck -f -b 8193 /dev/zniceny_oddil`. Příznak `-f` donutí souborový systém zkontrolovat a přepsat chybu programu `e2fsck`, jako by byl superblok paměti netknutý a vše bylo v pořádku.



# **Část II**

## **System**



# 32- a 64-bitové aplikace v 64-bitovém prostředí

SUSE LINUX je dostupný pro několik 64-bitových platforem. To však nutně neznamená, že všechny v distribuci obsažené aplikace byly portovány na 64 bitů. SUSE LINUX podporuje spouštění 32-bitových aplikací v 64-bitovém prostředí. Tato kapitola nabízí základní přehled o podpoře 32-bitových aplikací na 64-bitových SUSE LINUX platformách. Vysvětluje, jak se 32-bitové aplikace spouští (podpora běhu) a jak by měly být 32-bitové aplikace kompilovány, aby mohly běžet ve 32- i 64-bitovém prostředí. Obsahuje také informace o API jádra a vysvětlení toho, jak mohou 32-bitové aplikace běžet pod 64-bitovým jádrem.

6.1	Podpora běhu aplikací . . . . .	142
6.2	Vývoj softwaru . . . . .	143
6.3	Kompilace softwaru pro jinou platformu . . . . .	143
6.4	Specifikace jádra . . . . .	144

SUSE LINUX pro 64-bitové platformy AMD64 a EM64T je navržen tak, že existující 32-bitové aplikace v 64-bitovém rozhraní běží bez problémů přímo po instalaci.

Díky této podpoře můžete používat své oblíbené 32-bitové aplikace bez čekání na jejich 64-bitové verze.

## 6.1 Podpora běhu aplikací

### Důležité

#### Konflikty mezi verzemi aplikací

Pokud je aplikace dostupná pro 32- i 64-bitové prostředí, vede současná instalace obou verzí obvykle k problémům. Rozhodněte se pouze pro jednu verzi a tu nainstalujte a používejte.

### Důležité

Pro správné spuštění potřebují aplikace řadu různých knihoven. Bohužel jsou jména 32- i 64-bitových knihoven totožná. Musí být proto rozlišeny jinak než jménem.

Pro zachování kompatibility s 32-bitovou verzí jsou všechny knihovny uloženy na stejných místech jako ve verzi 32-bitové. 32-bitová verze knihovny `libc.so.6` je dostupná na cestě `/lib/libc.so.6` ve 32- i 64-bitovém prostředí.

Všechny 64-bitové knihovny a objektové soubory jsou uloženy v adresářích pojmenovaných `lib64`. 64-bitový objektový soubor, který byste normálně očekávali v adresářích `/lib`, `/usr/lib` a `/usr/X11R6/lib` naleznete v adresářích `/lib64`, `/usr/lib64` a `/usr/X11R6/lib64`. Pro 32-bitové knihovny tak zůstává místo v adresářích `/lib`, `/usr/lib` a `/usr/X11R6/lib`. Jména obou verzí knihoven tak mohou zůstat totožná.

Podadresáře objektových adresářů jejichž obsah není závislý na velikosti slova jsou stále na stejných místech. Například X11 fonty jsou stále na obvyklém místě v adresáři `/usr/X11R6/lib/X11/fonts`. To odpovídá standardům LSB (Linux Standards Base) a FHS (File System Hierarchy Standard).

## 6.2 Vývoj softwaru

Cross kompilační vývojářské nástroje umožňují vytvářet 32- i 64-bitové objekty. Výchozí je kompilace 64-bitových objektů. Pomocí zvláštních přepínačů lze kompilovat i 32-bitové objekty. Pro GCC se používá přepínač `-m32`.

Všechny hlavičkové soubory musí být napsány v podobě nezávislé na architektuře. Nainstalované 32- a 64-bitové knihovny musí mít API (programovací aplikační rozhraní) odpovídající nainstalovaným hlavičkovým souborům. Běžné SUSE prostředí tomuto požadavku odpovídá. Pokud jste ručně aktualizovali knihovny, vyřešte si problémy sami.

## 6.3 Kompilace softwaru pro jinou platformu

Pro vývoj binárních souborů pro jinou platformu je nutné nainstalovat příslušné knihovny pro tuto druhou platformu. Tyto balíčky se nazývají `jmenorpm-32bit`. Můžete také potřebovat hlavičky a knihovny z balíčků `jmenorpm-devel` a vývojové knihovny pro druhou platformu z `jmenorpm-devel-32bit`.

Většina opensource programů používá konfiguraci založenou na `autoconf`. Chcete-li použít `autoconf` pro konfiguraci programu pro druhou architekturu, přepište normální nastavení spuštěním skriptu `configure` s přidánými proměnnými prostředí.

Následující příklad se vztahuje k AMD64 či EM64T systému s x86 jako druhou architekturou:

1. Nastavte `autoconf` k použití 32-bitového kompilátoru:

```
CC="gcc -m32"
```

2. Přikážete linkeru zpracovávat 32-bitové objekty:

```
LD="ld -m elf64_i386"
```

3. Nastavte assembler, aby vytvářel 32-bitové objekty:

```
AS="gcc -c -m32"
```

4. Určete, že knihovny pro `libtool` atd. jsou v `/usr/lib`:

```
LDFLAGS="-L/usr/lib"
```

5. Určete, že jsou knihovny uloženy v podadresáři `lib`:

```
--libdir=/usr/lib
```

6. Určete, že jsou používány 32-bitové X knihovny:

```
--x-libraries=/usr/X11R6/lib/
```

Ne všechny proměnné jsou potřeba pro každý program. Upravte je podle potřeby.

```
CC="gcc -m64"           \
LDFLAGS="-L/usr/lib64;" \
    .configure          \
    --prefix=/usr       \
    --libdir=/usr/lib64
make
make install
```

## 6.4 Specifikace jádra

64-bitová jádra pro AMD64 a EM64T poskytují 64- i 32-bitové jaderné ABI (binární aplikační rozhraní). To 32-bitové je identické s ABI odpovídajícího 32-bitového jádra. To znamená, že může 32-bitová aplikace komunikovat s 64-bitovým jádrem stejně, jako by komunikovala s 32-bitovým jádrem.

32-bitová emulace systémových volání pro 64-bitové jádro nepodporuje řadu API používaných systémovými programy. Závisí to na konkrétní platformě. Z tohoto důvodu musí být některé aplikace, jako např. `lspci` nebo programy pro administraci LVM, zkompileovány jako 64-bitové programy, aby správně fungovaly.

64-bitové jádro umí nahrát pouze 64-bitové jaderné moduly zkompileované přímo pro toto jádro. Nelze použít 32-bitové jaderné moduly.



**Tip**

Některé aplikace vyžadují zvláštní moduly nahrávané jádrem. Pokud potřebujete použít takovou 32-bitovou aplikaci na 64-bitovém systému, kontaktujte výrobce aplikace a SUSE, abyste se ujistili, že je dostupná 64-bitová verze modulu a 32-bitová kompilovaná verze jaderného API pro tento modul.

**Tip**



# Startování

Startování a inicializace unixového systému bývají oříškem i pro zkušeného administrátora. Tato kapitola přináší stručný úvod do velmi komplexní. Najdete zde také informace o úrovních běhu a systémové konfiguraci v `sysconfig`.

7.1	Startovací proces v Linuxu . . . . .	148
7.2	Program <code>init</code> . . . . .	151
7.3	Úrovně běhu . . . . .	151
7.4	Změna úrovně běhu . . . . .	153
7.5	Init skripty . . . . .	154
7.6	Editor úrovní běhu . . . . .	158
7.7	SuSEconfig a <code>/etc/sysconfig</code> . . . . .	159
7.8	YaST <code>sysconfig</code> Editor . . . . .	161

## 7.1 Startovací proces v Linuxu

proces startování Linuxu se skládá z několika úrovní, ve kterých se spouští různé procesy. Následující část pojednává o startovacím procesu a nejdůležitějších komponentech.

### 1. BIOS

První věc, která se stane po zapnutí počítače je, že BIOS (Basic Input Output System) převezme řízení, nastaví obrazovku a klávesnici na počáteční hodnoty, a otestuje paměť. V této chvíli systém ještě neví o žádných ukládacích či externích zařízeních. Poté systém načte z paměti CMOS (kde je uloženo nastavení BIOSu) současný čas a datum, a informace o nejdůležitějších periferních zařízeních. Po načtení CMOS by měl BIOS rozeznat první pevný disk včetně informací o jeho geometrii. Poté může z tohoto disku začít zavádět operační systém (dále jen OS).

### 2. Zavaděč

Nejdříve se nahraje počátečních 512 bytů z prvního segmentu pevného disku do paměti a spustí se kód, který je uložen na začátku tohoto segmentu. Tento kód, zavaděč, začne nahrávat zbytek operačního systému. Proto se tomuto segmentu disku obvykle říká Master Boot Record (MBR). Více informací o zavaděči najdete v kapitole 8 na straně 163.

### 3. Jádru a *initrd*

Po systémové kontrole zavaděč nahraje jádro a ramdisk (*initrd*) do paměti. Linuxové jádro umožňuje zavedení malého souborového systému do paměti, ve kterém se zajistí před připojením kořenového souboru spuštění několika programů. Pak dojde k rozbalení *initrd* a jeho připojení ve formě dočasného kořenové souborového systému. *initrd* obsahuje minimální linuxový systém s programem *linuxrc*, který je spuštěn ještě před připojením skutečného kořenového systému. Po úspěšném dokončení běhu *linuxrc* jádro, pokud je to možné, uvolní paměť zabranou *initrd* a spustí *init*. Více informací o *initrd* najdete v části 7.1.1 na následující straně.

### 4. *linuxrc*

tento program provádí všechny akce potřebné ke správnému připojení kořenového souborového systému jako např. zavedení správného modulu souborového systému a ovladačů pro diskové zařízení. Po úspěšném připojení kořenového souborového systému se *linuxrc* ukončí a jádro spouští program *init*. Více informací o *linuxrc* najdete v části 7.1.2 na straně 150.

## 5. *init*

*init* se stará o proces spouštění na několika úrovních. Popis *init* najdete v části 7.2 na straně 151.

### 7.1.1 *initrd*

*initrd* je malý (obvykle komprimovaný) souborový systém, který se zavádí do ramdisku jako dočasný kořenový systém. Obsahuje minimální linuxové prostředí umožňující vykonání programů před připojením skutečného kořenového systému. Zavádí se přímo do paměti a nemá jiné hardwarové nároky než dostatečnou velikost paměti. *initrd* vždy spouští *linuxrc*, který by měl být ukončen bez chybového návratového kódu.

Ještě před připojením kořenového souborového systému a spuštěním operačního systému potřebuje jádro zavést moduly. Může jít o moduly ovladačů diskových zařízení nebo třeba o moduly s podporou síťových souborových systémů (viz na následující straně). Moduly potřebné pro připojení kořenového souborového systému by také měl zavádět *linuxrc*. Aby vše proběhlo úspěšně, musí jádro obsahovat kód, který mu umožní číst souborový systém *initrd*.

Vytvořte *initrd* skriptem *mkinitrd*. V systému SUSE LINUX se zaváděné *initrd* zadávají do proměnné *INITRD\_MODULES* v souboru */etc/sysconfig/kernel*. Po instalaci se proměnná automaticky nastaví na správnou hodnotu (*linuxrc* uloží instalační nastavení). Moduly se pak zavádějí v pořadí, v jakém jsou zadány v *INITRD\_MODULES*. To je důležité především u systémů, kde je vyžadováno několik SCSI ovladačů současně. Změna pořadí modulů by vedla ke změně jmen disků. Nastaveny by tedy měly být pouze ovladače, které jsou potřebné pro připojení kořenového souborového systému. Při instalaci se však zapíšou všechny použité SCSI ovladače, protože jejich pozdější ruční zavádění by mohlo působit problémy.

#### Důležité

##### Update *initrd*

Zavaděč zavádí *initrd* současně s jádrem. Po reinstalaci *initrd* není GRUB nutné neinstalovat, protože GRUB vyhledává soubory v adresářích při startu.

#### Důležité

## 7.1.2 linuxrc

Hlavním účelem linuxrc je příprava pro připojení a přístupu ke kořenovému adresáři. V závislosti na nastavením vašeho systému je linuxrc také odpovědný za:

**Zavádění modulů jádra** V závislosti na vaší konfiguraci může být potřeba před zavést pro některé zařízení ovladače (nejdůležitější je obvykle pevný disk). Aby bylo možné např. správně přistupovat k disku, musí jádro zavést správný modul pro souborový systém.

**Správa nastavení RAID a LVM** Pokud konfiguruje kořenový souborový systém na RAID nebo LVM, linuxrc nastaví, aby bylo možné přistupovat k souborovému systému, LVM nebo RAID. Informace o RAIDu najdete v části 3.8 na straně 100, informace o LVM v3.7 na straně 95.

**Správa síťového nastavení** Jestliže používáte síťový souborový systém, linuxrc zajišťuje zavedení správných síťových modulů.

Pokud je linuxrc volán při startu jako část instalace, úlohy se od uvedených výše liší:

**Vyhledání instalačního média** Po startu systému se z instalačního média zavádí jádro a zvláštní initrd s instalátorem YaST. Instalátor YaST běžící v RAM souborovém systému potřebuje k instalaci informace o aktuálním umístění instalačního média.

### **Prvotní rozpoznání hardwaru a zavedení příslušných modulů**

Jak bylo zmíněno v části 7.1.1 na předchozí straně, spouštění systému je proces, kdy se zavádí minimum ovladačů, které mohou být použity s většinou hardwaru. linuxrc provádí první zjištění hardwaru, aby zjistil, jaké ovladače budou pro váš systém potřeba. Tyto moduly jsou pak zapsány do proměnné INITRD\_MODULES v souboru `/etc/sysconfig/kernel` a používány při příštích startech systému. Tyto moduly také linuxrc zavede během instalačního procesu.

### **Zavedení instalačního nebo záchranného systému**

Po rozpoznání hardwaru a zavedení správných modulů spouští linuxrc instalační proces a to buď systém s instalátorem YaST nebo záchranný systém.

**Spuštění programu YaST** Na závěr linuxrc spustí samotná YaST, který začne s instalací balíčků a nastavením systému.

### 7.1.3 Informace i initrd

Další informace o `initrd` najdete v souborech `/usr/src/linux/Documentation/ramdisk.txt` a a v manuálových stránkách `initrd(4)` a `mkinitrd(8)`.

## 7.2 Program init

Program `init` inicializuje všechny další procesy, představuje tedy otce všech procesů. Mezi všemi programy má zvláštní roli: spouští ho přímo jádro a je imunní proti signálu 9, který normálně ukončí každý proces. Všechny další procesy pak program `init` spouští buď sám, nebo některý z jeho potomků.

Program `init` se konfiguruje centrálně v souboru `/etc/inittab`, kde se definují *úrovně běhu* angl. *runlevel* (více v 7.3 na této straně) a kde se určí, které služby a démoni mají být na jednotlivých úrovních k dispozici. Podle údajů v souboru `/etc/inittab` pak program `init` spouští různé skripty, které jsou z důvodu přehlednosti umístěny ve společném adresáři `/etc/init.d`.

Celý postup startu systému (a stejně tak i jeho zastavení) má tedy na starost program (a stejnojmenný proces) `init`. Z tohoto hlediska lze chápat činnost jádra jako proces na pozadí, jehož úlohou je udržovat všechny ostatní procesy a přidělovat hardware a čas CPU podle požadavků ostatních programů.

## 7.3 Úrovně běhu

V Linuxu existují různé *úrovně běhu*, které definují, v jakém stavu se nachází systém. Standardní úroveň běhu, které systém dosáhne po startu, je uvedena v souboru `/etc/inittab` v položce `initdefault`. Obvykle je to úroveň 3 nebo 5 (viz tabulka 7.1 na následující straně). Alternativou je zadat požadovanou úroveň běhu při startu (např. ze startovací výzvy LILO). Všechny parametry, které jádro samo nepoužije, totiž předá beze změny procesu `init`.

Aby šlo později úroveň běhu změnit, lze zavolat program `init` s udáním požadované úrovně běhu (což je dovoleno pouze superuživateli).

Například příkazem `init 1` přejde systém do *jednouživatelského režimu* *single user mode*, vhodného pro správu systému. Po ukončení této práce administrátor opět zadá `init 3`, čímž systém přejde opět na normální úroveň běhu, na které běží potřebné služby a kde se mohou přihlašovat uživatelé.

Tabulka níže podává přehled o dostupných úrovních běhu.

## Důležité

### Úroveň běhu 2 s oddílem `/usr/` připojeným přes NFS

Nepoužívejte úroveň běhu 2, pokud je adresář `/usr` na oddílu připojeném přes NFS. Adresář `/usr` obsahuje programy důležité pro běh systému. Služba NFS není na úrovni běhu 2 aktivní (lokální víceuživatelský režim bez sítě) a systém by v důsledku neexistence adresáře `/usr` nefungoval korektně.

## Důležité

*Tabulka 7.1: Seznam platných úrovní běhu*

Úroveň běhu	Význam
0	Stop <i>System halt</i>
S	Jednouživatelský režim, US klávesnice <i>Single user mode</i>
1	Jednouživatelský režim <i>Single user mode</i>
2	Lokální víceuživatelský režim bez sítě <i>Local multiuser without remote network (např. NFS)</i>
3	Plně víceuživatelský režim se sítí <i>Full multiuser with network</i>
4	Nepoužito
5	Plně víceuživatelský režim se sítí a KDM (standard), GDM nebo XDM <i>Full multiuser with network and xdm</i>
6	Restart systému <i>System reboot</i>

Z uvedeného bezprostředně plyne, že systém se dá zastavit zadáním příkazu `init 0` nebo případně restartovat zadáním `init 6`.

Máte-li na počítači nainstalovaný systém X Window (kap. 11 na straně 213) a přejete-li si, aby se uživatel přihlašoval přímo v grafickém prostředí, můžete nastavit standardní úroveň běhu pomocí programu YaST na hodnotu 5. Předtím si ovšem vyzkoušejte příkazem `init 5`, zda se systém bude chovat podle vašich představ.



## Varování

### Změna `/etc/inittab`

Doporučuje se velká opatrnost, chcete-li do souboru `/etc/inittab` zasahovat ručně. Jeho poškození totiž může vést k neschopnosti systému řádně nastartovat. Pokud se to stane, je zde ještě možnost z výzvy zavaděče zadat parametr `init=/bin/bash`, čímž se vám objeví přímo výzva příkazového procesoru:

```
boot:linux init=/bin/bash
```

## Varování

## 7.4 Změna úrovně běhu

Při změně úrovně běhu se nejprve spustí tzv. *stop-skripty*, které ukončí činnost některých programů současně úrovně. Dále se spustí *start-skripty* nové úrovně, a tím se zpravidla spustí i řada programů.

Pro názornost zde ukážeme příklad změny úrovně běhu z hodnoty 3 na 5:

- Administrátor (uživatel `root`) sdělí procesu `init`, že se má změnit úroveň běhu:
 

```
init 5
```
- Podle konfiguračního souboru `/etc/inittab` `init` usoudí, že má spustit skript `/etc/init.d/rc` s novou úrovní běhu jakožto parametrem.
- Nyní volá program `rc` ty `stop` skripty současně úrovně běhu, jimž neodpovídají `start-skripty` v nové úrovni. V našem případě jsou to ty skripty, jež se nalézají v adresáři `/etc/init.d/rc3.d` (stará úroveň běhu byla 3) a začínají písmenem `K`. Jména `stop` skriptů začínají písmenem `K` *kill*, zatímco jména `startovacích` skriptů začínají písmenem `S` *start*. Po písmenu `K` následuje číslo, udávající pořadí, aby byly respektovány případné závislosti mezi programy.
- Nakonec se zavolají `startovací` skripty nové úrovně běhu, které v našem případě leží v adresáři `/etc/init.d/rc5.d` a začínají písmenem `S`. Rovněž zde se dodržuje pořadí.

Pokud se stane, že změníte úroveň běhu na úroveň právě běžící (tj. např. z úrovně 3 opět na úroveň 3), přečte program `init` pouze svůj konfigurační soubor `/etc/inittab` a zjistí, zda i v rámci téže úrovně nejsou nějaké změny. Pokud je najde, provede příslušné kroky (například spustí program `getty` pro další konzoli).

## 7.5 Init skripty

Skripty v adresáři `/etc/init.d` se dělí do dvou kategorií:

### Skripty, které program `init` volá přímo

to je případ startu a korektního zastavení systému (např. klávesovou kombinací `(Ctrl)-(Alt)-(Return)`)

Vykonání těchto skriptů je definováno v `/etc/inittab`.

### Skripty, které program `init` volá nepřímo

to se stane při změně úrovně běhu. Spustí se skript `/etc/init.d/rc` volající správné skripty ve správném pořadí.

Skripty pro změnu úrovně běhu se rovněž nalézají v adresáři `/etc/init.d`, ale volají se pomocí symbolických odkazů z jednoho z adresářů počínaje `/etc/init.d/rc0.d` až po `/etc/init.d/rc6.d`. To je velmi názorné a zabraňuje to duplicitě skriptů, použitých pro více úrovní běhu.

Každý z těchto skriptů se dá volat jako start-skript i stop-skript, rozlišují proto parametry `start` a `stop`.

Navíc rozlišují skripty parametry `restart`, `reload`, `force-reload` a `status`. Význam všech voleb je v následující tabulce.

*Tabulka 7.2: Přehled voleb init skriptů*

Volba	Význam
<code>start</code>	Spustit službu.
<code>stop</code>	Ukončit službu.
<code>restart</code>	Pokud služba běží, ukončit ji a znovu spustit, pokud neběží, pouze spustit.

<code>reload</code>	Znovu načíst konfiguraci služby, aniž by se zastavovala a spouštěla.
<code>force-reload</code>	Totéž jako <code>reload</code> , pokud to služba podporuje, jinak jako <code>restart</code> .
<code>status</code>	Zobrazit aktuální status.

#### Příklad:

Při opuštění úrovně běhu 3 je skript `/etc/init.d/rc3.d/K40network` jedním ze spuštěných skriptů. Program `/etc/init.d/rc` volá skript `/etc/init.d/network` s parametrem `stop`. Při vstupu do úrovně běhu 5 se spustí tentýž skript, ale s parametrem `start`.

Odkazy v podadresářích pro jednotlivé úrovně běhu slouží pouze k tomu, aby umožnily přiřadit skriptu úrovním běhu.

Vytvoření a odstranění potřebných odkazů provádí program `insserv` při instalaci a deinstalaci balíků. Podrobnosti najdete v manuálové stránce tohoto programu.

V dalším odstavci najdete krátký popis startovacího a ukončovacího skriptu spolu s řídicím skriptem:

**boot** Spouští se při startu systému přímo z programu `init`. Je nezávislý na požadované výsledné úrovni běhu a provádí se pouze jednou. Spustí se démon jádra, který zajistí zavedení modulů jádra. Zkontrolují se souborové systémy, zruší se některé nadbytečné soubory v adresáři `/var/lock` a síť se nakonfiguruje pro *loopback device* (pokud je to nastaveno v souboru `/etc/rc.config`). Dále se nastaví systémový a PnP hardware pomocí nástroje `isapnp`.

Pokud se stane chyba při automatické opravě souborového systému, má systémový administrátor možnost po zadání hesla zadat další informace přispívající k jejímu odstranění.

Dále se vykonají všechny skripty v adresáři `/etc/init.d/boot.d` začínající písmenem `S`. Je to proto vhodné místo pro vaše rozšíření o ty kroky, které by měl systém dělat pouze při startu.

Nakonec se spustí skript `boot.local`.

**boot.local** Zde můžete přidat další příkazy, které se mají provést při startu, než se začne zvyšovat úroveň běhu. Funkční obdobou v dosových systémech je soubor `AUTOEXEC.BAT`.

**boot.setup** Všeobecná nastavení při přechodu z jednouživatelského režimu *single user mode* na libovolnou vyšší úroveň běhu, například rozložení kláves a konfigurace konzole.

**halt** Tento skript se spouští při přechodech na úrovně běhu 0 nebo 6. Proto se může zavolat jak pod jménem `halt`, tak i `reboot`, a podle předaného jména se systém znovu nashutuje nebo ukončí.

**rc** Řídící skript pro změnu úrovně běhu. Spouští nejprve stop skripty současné úrovně a po nich start skripty nové úrovně.

Do této kostry můžete vhodně zasadit své vlastní skripty. Šablonu na to najdete v souboru `/etc/init.d/skeleton`. Pro konfiguraci spuštění vlastního skriptu v souboru `/etc/rc.config` zde vytvořte proměnnou `START_služba`. Dodatečné parametry lze uvést v případě potřeby také do souboru `/etc/rc.config` (viz např. skript `/etc/init.d/gpm`).

---

### Varování

Při vytvoření vlastních skriptů zachovejte opatrnost. Chybný skript může způsobit nefunkčnost systému.

---

Varování

## 7.5.1 Vkládání skriptů

V Linuxu není problém vytvářet vlastní skripty a poměrně jednoduše je integrovat do stávajícího prostředí. Informace o způsobu pojmenování, formátu a organizaci vlastních skriptů najdete ve specifikaci LSB a manuálových stránkách `init`, `init.d` a `insserv`. Zajímavé informace najdete také v manuálových stránkách `startproc` a `killproc`.

---

### Varování

#### Vytváření vlastních init skriptů

Chyby v `init` skriptech mohou vést k zamrznutí počítače. Věnujte prosím editaci těchto skriptů maximální pozornost a pokud je to možné, otestujte je. Užitečné informace o `init` skriptech najdete v části 7.3 na straně 151.

---

Varování

- Jako šablonu pro svůj nový init skript použijte soubor `/etc/init.d/skeleton`. Kopii tohoto souboru uložte pod novým jménem a editujte důležité položky jako program, jména souborů, cesty a další detaily. Šablonu samozřejmě můžete rozšířit o vlastní části.
- Blok `INIT INFO` je povinnou částí skriptu a měly by v něm být provedeny příslušné změny:

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

Na první řádce bloku `INFO` po řádce `Provides :`, uveďte jméno služby nebo programu kontrolovaného nově vytvářeným skriptem. V řádkách `Required-Start :` a `Required-Stop :` uveďte všechny služby, které je nutné spustit a zastavit před startem nebo spuštěním vaší nové služby.

Tyto informace budou později použity při generování jména a čísla skriptu v adresářích úrovně běhu. V `Default-Start` a `Default-Stop` uveďte úroveň běhu, kdy se služba má automaticky spustit nebo ukončit. Na konec do řádky `Description` napište krátký popis služby.

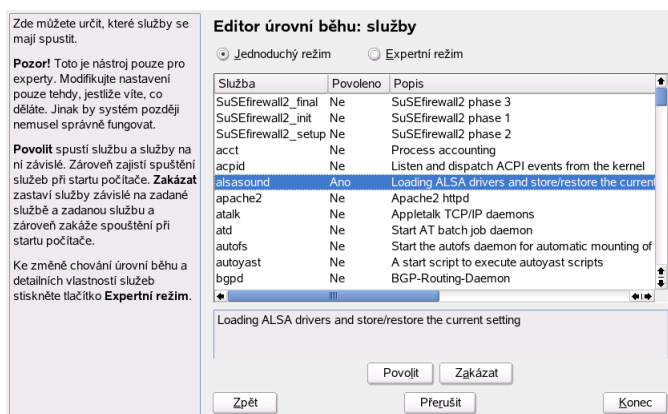
- Odkazy z `/etc/init.d/` do příslušného adresáře úrovně běhu (`/etc/init.d/rc?.d/`), vytvoříte zadáním příkazu `insserv jmeno_skriptu`. Program `insserv` používá hlavičku `INIT INFO` pro vytváření důležitých odkazů potřebných pro spuštění a zastavení skriptu v adresářích úrovně běhu (`/etc/init.d/rc?.d/`). Program se také stará o správné pořadí spuštění a zastavení v určených úrovních běhu. Pokud byste raději používali grafický nástroj, můžete použít editor úrovně běhu v programu YaST, popsany v sekci 7.6 na následující straně.

Pokud již skript v adresáři `/etc/init.d/` existuje, můžete ho do existujícího schématu úrovně běhu jednoduše integrovat pomocí programu `insserv` nebo povolením příslušné služby v programu YaST. Vámi provedené změny se projeví při následujícím restartu počítače, během kterého dojde k automatickému spuštění nové služby.

## 7.6 Editor úrovní běhu

Po spuštění tohoto modulu programu YaST se zobrazí seznam dostupných služeb a jejich stav (zda jsou povoleny či ne). Zvolit si můžete ze dvou režimů zobrazení 'Jednoduchý režim' nebo 'Expertní režim'. Jako výchozí je nastaven 'Jednoduchý režim', který je vhodný pro většinu situací.

V levém sloupci 'Jednoduchého režimu' je jméno služby, v prostředním stav služby a v pravém sloupci krátký popis služby. U zvolené služby je detailnější popis dostupný v okně pod seznamem. Službu povolíte tak, že ji označíte a kliknete na 'Povolit'. Pokud chcete službu zakázat, opět ji zvolte a klikněte na tlačítko 'Zakázat'.



Obrázek 7.1: Editor úrovní běhu

Pokud potřebujete o službách více informací a chtěli byste použít detailnější nastavení, vyberte 'Expertní režim'. V tomto režimu získáte informace o nastavené výchozí úrovni nebo-li `initdefault`, která říká, do jaké úrovně se má systém spustit při startu. Jako výchozí je nastavena úroveň 5 (Plný víceuživatelský režim se sítí a xdm). Vhodnou náhradou obvykle bývá úroveň 3 (Plný víceuživatelský režim se sítí).

YaST umožňuje výběr nové výchozí úrovně běhu (viz tabulka 7.1 na straně 152). Zároveň nabízí tabulku, kde můžete povolit nebo zakázat běh určité služby. V tabulce najdete všechny dostupné služby a démony. Příslušnou úroveň nastavíte

tak, že v řádce vybrané služby označíte příslušné pole úrovně běhu ('B', '0', '1', '2', '3', '5', '6' a 'S'), ve které se má služba spustit. Úroveň 4 není definována a můžete si ji nastavit podle svých potřeb. Jako poslední najdete v tabulce krátký popis služby nebo démona.

Pomocí 'Nastavit/Obnovit' můžete určit, co se má se zvolenou službou provést. Okamžitě můžete služby povolit či zakázat v 'Spustit/Zastavit/Načíst znovu'. Pokud po změnách chcete zobrazit aktuální stav, zvolte v 'Spustit/Zastavit/Načíst znovu' položku 'Znovu načíst stav'. Kliknutím na tlačítko 'Konec' uložíte změny.

### Varování

#### Změna úrovně běhu

Chybné nastavení úrovně běhu může vést k chybě systému. Před změnou úrovně běhu se prosím ujistěte, zda se tím neovlivní některá ze služeb důležitých pro váš systém.

### Varování

## 7.7 SuSEconfig a /etc/sysconfig

Prakticky celá konfigurace systému SUSE LINUX je otázkou centrálního konfiguračního adresáře `/etc/sysconfig`. Ve verzích starších než 8.0 byla konfigurace soustředěna do souboru `/etc/rc.config`. Tento soubor již není používán.

Každý ze skriptů v adresáři `/etc/init.d` načítá soubory z adresáře `/etc/sysconfig`, kde převezme platné hodnoty jednotlivých proměnných. Nastavení v `/etc/sysconfig` vede také k automatickému vytváření nebo změně některých dalších konfiguračních souborů skriptem `SuSEconfig`. Tak například po změnách v síťové konfiguraci se nově vytvoří soubor `/etc/host.conf`, protože na těchto změnách závisí.

Po ručních změnách v některém ze souborů v adresáři `/etc/sysconfig` musíte vždy zavolat program `SuSEconfig`, abyste tak zajistili, že se vaše změny rozšíří i do závislých konfiguračních souborů. Použijete-li na konfiguraci program `YaST`, nemusíte se o to starat, protože ten zavolá program `SuSEconfig` při korektním ukončení automaticky.

Tato koncepce vám umožní provést zásadní změny v konfiguraci, aniž byste museli restartovat počítač. Některé změny však jdou tak daleko, že je třeba restartovat alespoň některé jimi ovlivněné programy. To je typické například u konfigu-

race sítě, kde zadáním příkazů `rcnetwork stop` a `rcnetwork start` dosáhnete toho, že se změnou postižené programy restartují.

Doporučený postup změny systémového nastavení se skládá z následujících kroků:

1. Přejděte do jednouživatelského režimu *single user mode* (úroveň běhu 1) pomocí příkazu `init 1`.
2. Změňte konfigurační soubory podle své potřeby. Použít můžete svůj oblíbený textový editor nebo editor v programu YaST.

### Důležité

#### Manuální změna systémové konfigurace

Pokud ke změně *nepoužíváte* YaST, ujistěte se že jsou prázdné proměnné a proměnné skládající se z více položek v souborech v adresáři `/etc/sysconfig` v uvozovkách (`KEYTABLE=""`). Proměnné s jednou hodnotou není nutné uzavírat do uvozovek.

### Důležité

3. Aby se změny projevíly, spusťte `/sbin/SuSEconfig`. Pokud jste změny provedli pomocí programu YaST, spustí se SuSEconfig automaticky.
4. Vraťte se do původní úrovně běhu příkazem `init 3` (nahraďte 3 číslem vaší úrovně běhu).

Tento postup je nutné dodržovat při hlubších zásazích do systému, jako je například změna konfigurace sítě. V případě jednoduchých změn není zapotřebí přechod do *jednouživatelského režimu*, ale získáte tak jistotu, že u všech služeb došlo ke správnému spuštění.

### Tip

Automatickou konfiguraci programem SuSEconfig lze vypnout tak, že se proměnná `ENABLE_SUSECONFIG` v souboru `/etc/sysconfig/suseconfig` nastaví na hodnotu `no`. Je to ovšem i cesta, jak současně ztratit instalační podporu SUSE. Nevypínejte SuSEconfig, pokud chcete využít bezplatné instalační podpory. Autokonfiguraci je možné zakázat také pouze částečně.

### Tip



## 7.8 YaST sysconfig Editor

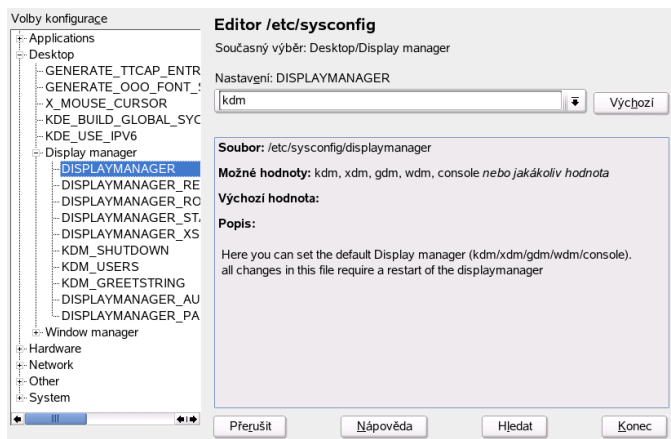
Nejdůležitější konfigurační soubory systému SUSE LINUX jsou uloženy v adresáři `/etc/sysconfig`. Sysconfig editor představuje způsob, jak zde uložená nastavení editovat s co nejvyšším pohodlím. Hodnoty lze měnit a v případě nutnosti také vkládat do vlastních konfiguračních souborů. Většinu nastavení není nutné nastavovat ručně. K nastavení dojde automaticky při instalaci příslušných balíků.

### Varování

#### Změna souborů v `/etc/sysconfig/`

Pokud nemáte se změnou konfiguračních souborů žádné zkušenosti, nemějte žádná nastavení v adresáři `/etc/sysconfig`. Chybný zásah do těchto souborů může vést k nefunkčnosti systému. Pokud je ruční editace nezbytná, věnujte pozornost komentářům u jednotlivých proměnných.

### Varování



Obrázek 7.2: Konfigurace systému pomocí editoru souborů sysconfig

Dialog YaST sysconfig editoru se skládá ze tří částí. V levé části jsou zobrazeny nastavitelné proměnné. Po volbě proměnné se v pravé části objeví aktuální na-

stavení zvolené proměnné. Pod tímto nastavením najdete krátký popis funkce proměnné, možné dosaditelné hodnoty, výchozí hodnotu a soubor, kde se tato proměnná nachází. Dialog také poskytuje informace o skriptech, které se po nastavení této proměnné spustí a službách, které se v důsledku nového nastavení mohou spustit. Po změně se YaST dotáže, zda si skutečně proměnnou přejete změnit. Nastavení uložíte kliknutím na 'Dokončit'.

# Startování systému a zavaděče

Tato kapitola popisuje různé metody startování linuxového systému. Nejdříve jsou však vysvětleny některé technické detaily tohoto procesu. Poté následuje detailní popis programů GRUB (současný zavaděč používaný v systému SUSE LINUX) a možnosti použití programu YaST. V této kapitole najdete také popis řešení některých problémů, které mohou u nastavení zavaděče GRUB nastat.

8.1	Startování . . . . .	164
8.2	Výběr zavaděče . . . . .	165
8.3	Startování systému se zavaděčem GRUB . . . . .	166
8.4	Konfigurace zavaděče pomocí programu YaST . . . . .	176
8.5	Odinstalace zavaděče LILO nebo GRUB . . . . .	179
8.6	Vytvoření startovacího CD . . . . .	181
8.7	Grafická konzole SUSE . . . . .	183
8.8	Řešení problémů . . . . .	183
8.9	Další informace . . . . .	185

Tato kapitla se zaměřuje na konfiguraci zavaděče GRUB. Proces startování systému je popsán v kapitole 7 na straně 147. Zavaděč je rozhraním mezi počítačem (BIOSem) a operačním systémem (SUSE LINUX). Nastavení zavaděče ovlivňuje spouštění nainstalovaných operačních systémů a dostupnos parametrů, kterými můžete spouštění ovlivnit.

Nejdůležitější pojmy používané v této kapitole jsou:

**Master Boot Record** Struktura MBR je standardizována a není závislá na použitém operačním systému. Prvních 446 bytů je rezervováno pro kód startovacího programu. Následujících 64 bytů je určeno pro uložení tabulky diskových oddílů, která obsahuje informace o maximálně 4 oddílech. Bez této tabulky nemůže být na disku žádný souborový systém - disk je bez této tabulky nepoužitelný. Poslední 2 byty musí obsahovat speciální magické číslo (AA55). MBR, který na této pozici obsahuje jiné číslo, může být BIOSem, a některými operačními systémy, posouzen jako neplatný.

**Zaváděcí sektory** Zaváděcí sektory jsou uloženy na každém diskovém oddílu jako první. Výjimku tvoří pouze rozšířené diskové oddíly, které jsou pouze kontejnery pro další oddíly. Zaváděcí sektory jsou velké 512 bytů, a slouží k uložení kódu pro spuštění operačního systému uloženého na tomto oddílu. Zaváděcí sektory na oddílech vytvořených z DOSu, OS/2, a Windows fungují přesně jak bylo popsáno (navíc obsahují některá základní data o struktuře souborového systému). V Linuxu, na rozdíl od jmenovaných OS, je tento sektor prázdný (i po vytvoření souborového systému), a Linuxový oddíl není schopen zavést sám sebe, i když oddíl obsahuje platný souborový systém s jádrem. Aby bylo možné zavést z tohoto oddílu Linux, musíme do tohoto sektoru uložit zaváděcí program. Zaváděcí sektor s platným zaváděcím kódem obsahuje na stejné pozici jako MBR (poslední 2 byty) shodné magické číslo (AA55).

## 8.1 Startování

V tom nejjednodušším případě, kdy se na počítači nachází pouze jeden operační systém, se startování chová podle pravidel popsaných výše. V případě více operačních systémů však přicházejí ke slovu následující postupy:

### Spouštění dalšího systému z externího média

Jeden z nainstalovaných systémů je spouštěn z disku, druhý pomocí zavaděče uloženého na externím médiu (disketa, CD, USB flash disk). Obvykle

tato metoda není nutná, protože GRUB umí spouštět i jiné operační systémy než Linux.

**Instalace zavaděče do MBR** Zavaděč umožňuje spouštění různých operačních systémů. Který bude spuštěn, si může vybrat ve startovací nabídce. Aby došlo ke spuštění jiného systému, musí být počítač restartován. Toto řešení je samozřejmě možné jen v případě, že je zavaděč kompatibilní se všemi operačními systémy, které chcete s jeho pomocí spouštět. GRUB, zavaděč systému SUSE LINUX, je schopný spouštět všechny obvyklé operační systémy. Ve výchozím nastavení je nainstalován do MBR.

### 8.1.1 Startování DOSu a Windows 9x

MBR DOSu na prvním pevném disku obsahuje informaci o tom, který oddíl je aktivní - tedy kde se má hledat kód pro zavedení operačního systému. Proto musí být DOS nainstalován na první pevný disk. Spustitelný kód v MBR (zavaděč prvního stupně) potom testuje, zda označený oddíl obsahuje platný zaváděcí sektor. Jestliže je vše v pořádku, spustí se odtud zavaděč druhého stupně. Odtud je možné nahrávat DOSové programy, a objeví se obvyklý DOSový prompt. V DOSu lze označit jako aktivní pouze primární diskové oddíly. Z toho důvodu nemůžete použít pro zavádění DOSu logické diskové oddíly, které jsou uvnitř rozšířených oddílů.

## 8.2 Výběr zavaděče

V systému SUSE LINUX je jako výchozí zavaděč použit GRUB. V některých případech, kdy je použit zvláštní hardware ve spojení s určitým softwarem, však může být mnohem vhodnější použití zavaděče LILO.

Zavaděč LILO se automaticky nainstaluje v případě aktualizace ze staršího systému SUSE LINUX, který používal jako výchozí zavaděč LILO. V nové instalaci se vždy nainstaluje zavaděč GRUB. Výjimkou jsou RAIDové systémy, které splňují jednu z následujících podmínek:

- Na CPU závislé RAID řadiče (např. řada řadičů Promise nebo Highpoint)
- Softwarový RAID
- LVM

Informace o instalaci a nastavení zavaděče LILO najdete v databázi instalační podpory pod heslem LILO.

## 8.3 Startování systému se zavaděčem GRUB

GRUB (GRand Unified Boot loader) podobně jako LILO pracuje ve dvou fázích. V první fázi se spustí kód velký pouze 512 bytů, který je zapsaný v MBR, zaváděcím sektoru diskového oddílu nebo na disketě. Druhá fáze spočívá ve spuštění většího programu vykonávajícího zavádění jako takové. Jedinou funkcí programu první fáze je zavést program druhé fáze.

Odsud již GRUB pracuje jinak než LILO, poněvadž program druhé fáze obsahuje kód pro čtení ze souborového systému. V současné době jsou podporovány tyto souborové systémy: Ext2, Ext3, ReiserFS, JFS, XFS, Minix a DOS FAT používaný Windows. GRUB tedy může přistupovat na souborové systémy již před vlastním startováním systému. Číst lze z těch zařízení, která jsou dostupná přes BIOS (disketové mechaniky a pevné disky). Ve výsledku to znamená, že provedené změny v konfiguraci programu GRUB nemusíme po každé změně zapsat reinstalací zavaděče. Při zavádění GRUB načte svůj soubor s menu a odsud zjistí, na kterých oddílech leží jádro a výchozí RAM disk (`initrd`), a je sám schopen tyto soubory najít.

Další výhodou programu GRUB je, že lze jednoduše měnit veškeré parametry startu systému před samotným startem. Pokud při zavádění zjistíte, že soubor s menu obsahuje chyby, je stále možné opravit tyto chyby za chodu. V programu GRUB také můžete zadávat příkazy interaktivně na příkazový řádek, takže lze startovat i systém, jenž není uveden v konfiguračním souboru.

### 8.3.1 Startovací menu

GRUB zobrazuje zaváděcí menu na grafické titulní obrazovce nebo v rozhraní textového režimu. Co bude obsahem této obrazovky, lze nastavit v souboru s menu `/boot/GRUB/menu.lst`. V tomto souboru jsou popsány veškeré informace o diskových oddílech a operačních systémech, které lze zvolit z nabídky při zavádění.

GRUB nahraje menu přímo ze souborového systému při každém startu systému. Pokud chcete změnit nastavení zavaděče, upravíte pouze menu soubor pomocí programu YaST nebo vaším oblíbeným editorem.

Soubor s menu obsahuje příkazy spouštěné při zavádění a jeho skladba je jednoduchá na pochopení. Každý řádek sestává z příkazu, volitelně následovaného parametry. Ty jsou odděleny mezerou stejně jako v shellu. Z historických důvodů lze u některých příkazů použít před jejich prvním parametrem `=`. Řádky začínající znakem hash `#` jsou považovány za komentáře.

Každý záznam, jenž se objeví v menu zavaděče, odpovídá jménu v menu souboru, které musí být uvozeno pomocí slova `title`. Jinými slovy: textový řetězec následující za `title` (včetně mezer) se zobrazí jako volitelná položka. Následující řádky až do další položky `title` pak reprezentují příkazy, které se provedou, pokud zvolíte tuto položku v menu.

Jednoduchý příklad takového příkazu je zřetěžené nahrání zavaděče jiného operačního systému. Příkaz se nazývá `chainloader` a jako parametr má obvykle zaváděcí blok jiného diskového oddílu. Zapsáno v notaci programu GRUB:

```
chainloader (hd0,3)+1
```

Jak GRUB pojmenovává zařízení je vysvětleno v sekci 8.3.1 na následující straně. Příklad uvedený výše odkazuje na první blok čtvrtého oddílu prvního disku.

Příkaz pro určení obrazu jádra je `kernel`. První parametr je cesta k obrazu jádra na diskovém oddíle. Zbylé argumenty se během zavádění předají jádru jako parametry pro start Linuxu.

Pokud jádro nemá zabudované nezbytné ovladače pro souborový systém nebo disk (aby mohlo přistupovat na kořenový oddíl), připojte také příkaz `initrd`. Tento příkaz má pouze jeden parametr, a to cestu k souboru `initrd`. Příkaz `initrd` musí být umístěn bezprostředně po příkazu `kernel`, protože jádro (nyní již zavedené) očekává nějaký obraz `initrd` na konkrétní adrese v paměti.

Příkaz `root` zjednodušuje určení, kde se nachází obrazy jádra a `initrd`. `root` má jako jediný parametr označení zařízení nebo diskového oddílu (v notaci GRUB).

GRUB následně připojí na začátek všech cest k souborům (jádra, `initrd` nebo jiných souborů, které výslovně neurčují cestu nebo zařízení) hodnotu svého parametru. Toto připojování se děje do nalezení dalšího příkazu `root`. Tento příkaz není použit v souboru `menu.lst`, který je generován během instalace.

Příkaz `boot` je automaticky proveden jako poslední u každé položky menu. Nemusí se tedy zapisovat jako příkaz do souboru s menu. Jestliže se však dostanete do situace, že musíte zadávat příkazy do příkazové řádky programu GRUB, nezapomeňte nakonec zadat příkaz `boot`. Příkaz nemá parametry a pouze spustí zavádění obrazu jádra nebo zřetěžené zavaděč (`chain loader`).

Jakmile máte vytvořen soubor s nabídkou položek odpovídajících jednotlivým OS, vyberte jednu jako implicitní pomocí příkazu `default`. Pokud nevyberete implicitní položku tímto příkazem, zavede se systém z první položky v menu (číslo 0). Lze také nastavit časovou prodlevu ve vteřinách, kdy můžete vybrat některou z položek. Řádky s příkazy `timeout` a `default` jsou obvykle umístěny před položky menu. Vzorový menu soubor je popsán v sekci 8.3.1 na následující straně.

## Konvence pojmenování pevných disků a oddílů

GRUB pojmenovává disky a oddíly podle jiných konvencí, než jste zvyklí v Linuxu, a jaké byste nejspíš očekávali (např. `/dev/hda1`). První disk je vždy odkazován jako `hd0`. Disketová mechanika se nazývá `fd0`.

### Důležité

#### Výpočet čísla oddílu

GRUB počítá diskové oddíly od nuly. `hd0,0` tedy odkazuje na první oddíl prvního disku. Označení odpovídá typickému stolnímu počítači s jedním diskem připojeným jako primární master disk. V Linuxu bychom se na něj odkazovali pomocí `/dev/hda1`.

### Důležité

Čtyři primární oddíly (které lze na disku vytvořit) jsou číslovány od 0 do 3 a logické oddíly jsou číslovány od 4 výš.

```
(hd0,0)   první primární oddíl prvního disku
(hd0,1)   druhý primární oddíl prvního disku
(hd0,2)   třetí primární oddíl prvního disku
(hd0,3)   čtvrtý primární oddíl prvního disku
(hd0,4)   první logický oddíl
(hd0,5)   druhý logický oddíl
...
```



## Důležité

### IDE, SCSI a RAID

GRUB nerozlišuje mezi IDE, SCSI nebo RAID zařízením. Veškeré pevné disky detekované BIOSem nebo diskovým řadičem jsou číslovány podle pořadí zavádění nastaveném v BIOSu.

## Důležité

Fakt, že disky jsou jinak adresovány Linuxem a jinak BIOSem, je problém jak pro LILO, tak pro GRUB. Oba programy používají podobný algoritmus pro mapování. Nicméně GRUB ukládá výsledek tohoto algoritmu do souboru (`device.map`), který lze editovat. Více informací o souboru `device.map` najdete v8.3.2 na straně 173.

V programu GRUB musí být cesta uvedena jako jméno zařízení, uzavřené do kulatých závorek, následovaná jménem souboru včetně plné cesty na tomto zařízení nebo oddílu. Cesta musí vždy začínat lomítkem. Například v systému s jedním IDE diskem a Linuxem uloženým na prvním oddílu, se odkážete na jádro takto:

```
(hd0,0)/boot/vmlinuz
```

### Vzorový soubor `menu.lst`

Následující příklad ukazuje, jak funguje soubor `menu.lst`.

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd
title windows
    chainloader (hd0,0)+1
title floppy
    chainloader (fd0)+1
title failsafe
    kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma
    apm=off acpi=off vga=normal nosmp maxcpus=0 3
    initrd (hd0,4)/initrd.shipped
```

Tento fiktivní stroj má zaváděcí Linuxový oddíl na `/dev/hda5`, kořenový oddíl na `/dev/hda7`, a instalaci Windows na `/dev/hda1`.

První část souboru definuje nastavení titulní obrazovky a standardní chování:

**\mbx{gfxmenu (hd0,4)/message}**

Obrázek zobrazený na pozadí je uložen na `/dev/hda5` a jmenuje se `message`.

**\mbx{color }** Barevné schéma: bílá pro popředí, modrá jako pozadí, černá jako popředí pro vybranou položku a světle šedá pro pozadí zvolené položku. Definice barev neovlivní titulní grafickou obrazovku definovanou pomocí `gfxmenu`, ale pouze standardní textové rozhraní programu GRUB. V systému SUSE LINUX se můžete z grafického menu do textového přepnout stisknutím (`Esc`).

**\mbx{default 0}** Implicitně se zavede první položka `title linux`.

**\mbx{timeout 8}** Časová prodleva 8 vteřin. Pokud uživatel nezvolí jinak, zavede se implicitní volba.

Obsáhlejší druhá část definuje zavádění jednotlivých operačních systémů:

- První položka (`title linux`) nastavuje zavádění systému SUSE LINUX. Jádro (`vmlinuz`) je uloženo na prvním disku na prvním logickém oddílu (v tomto případě zaváděcí oddíl). Následné parametry blíže určují kořenový oddíl a mód zobrazení při startování jádra. Kořenový oddíl je uveden podle Linuxové konvence, protože bude interpretován samotným jádrem (a ne programem GRUB). Obraz `initrd` je uložen na stejném logickém oddíle prvního disku.
- Druhá položka (`title windows`) je odpovědná za zavedení Windows, které jsou nainstalované na prvním oddíle prvního disku (`hd0, 0`). Příkaz `chainloader +1` způsobí, že GRUB načte a spustí první sektor definovaného oddílu.
- Další záznam povoluje zavádění systému z disketové mechaniky bez zásahů do BIOSu.
- Položka `failsafe` zavádí jádro Linuxu s mnoha přesně specifikovanými parametry jádra, aby bylo možné zavést systém na problematickém hardwaru.

Konfigurační soubor s menu můžete kdykoliv změnit. GRUB automaticky při příštím restartu načte tyto změny ze souboru. Abyste provedli permanentní změny v nastavení zavádění systému, použijte odpovídající modul programu YaST, nebo váš oblíbený editor. Pokud chcete změnit pouze jednorázově chování programu GRUB při zavádění, využijte jeho příkazovou řádku.

## Změna položek v menu při startu

Grafické rozhraní dovoluje nejen zvolit položku pro zavedení systému (pomocí kurzorových kláves), ale umožňuje vám také zadat přídavné parametry pro jádro na příkazový řádek (pokud jste vybrali položku s Linuxem). Toto umí i LILO, avšak GRUB jde ještě o krok dál. Pokud stisknete (**Esc**), přepnete se do textového módu. Nyní stiskem (**E**) vstoupíte do editovacího režimu. Zde můžete přímo měnit nastavení vybrané položky, které bude platné pouze pro toto zavádění systému. Žádná změna se nezapíše do souboru.

### Důležité

#### Rozložení klávesnice během fáze zavádění

V době zavádění systému můžete použít pouze americké rozložení klávesnice. Dejte pozor na jiné umístění znaků.

### Důležité

Po zapnutí režimu editace použijte kurzorové klávesy pro výběr položky, kterou chcete upravit. Nyní stiskněte (**E**). Upravte parametry (diskové oddíly, cesty k souborům), které mají chybné hodnoty a ovlivňují proces zavádění. Opusťte režim editace stiskem (**Enter**) a jděte zpět do menu, kde můžete spustit zavádění systému s upravenými parametry. GRUB zobrazuje v dolní části obrazovky rady ohledně dalších možných činností.

Aby byly změny trvalé, upravte soubor `menu.lst` jako uživatel `root`, a přidejte libovolné parametry jádra oddělené mezerou na konec existujícího řádku:

```
title linux
    kernel (hd0,0)/vmlinuz root=/dev/hda3 parametry_jadra
    initrd (hd0,0)/initrd
```

Při příštím startování systému GRUB použije tyto nové parametry. Další možností, jak předat jádru přídavné parametry, je pomocí modulu programu YaST. Veškeré argumenty napište na konec řádku, oddělené mezerou.

## Zástupné znaky a zadání jádra ke spuštění

Pokud se podílíte na vývoji jádra nebo používáte jádro vlastní, musíte, aby se systém správně spouštěl, buď změnit položky v `menu.lst` nebo zadat příslušné parametry do startovacího promptu. Nyní máte možnost se těmito procedurám vyhnout použitím *zástupných znaků*. S jejich pomocí se všechna jádra vyhovující kritériím, automaticky vloží do startovací nabídky.

Pro použití zástupných znaků stačí dodržovat pravidla při pojmenování obrazů jader a `initrd` a nová položka v souboru `menu.lst`. Předpokládejme, že máme systém s jádry a příslušnými `initrd`:

```
initrd-default
initrd-test
vmlinuz-default
vmlinuz-test
```

Abyste jak `linux-default`, tak `linux-test` vložili do souboru `menu.lst` musíte zadat:

```
title linux-*
  wildcard (hd0,4)/vmlinuz-*
  kernel (hd0,4)/vmlinuz-* root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-*
```

V tomto příkladě GRUB vyhledá dostupná jádra na oddíle (hd0,4) a doplní do souboru `menu.lst`:

```
title linux-default
  wildcard (hd0,4)/vmlinuz-default
  kernel (hd0,4)/vmlinuz-default root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-default
title linux-test
  wildcard (hd0,4)/vmlinuz-test
  kernel (hd0,4)/vmlinuz-test root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-test
```

Problémy mohou nastat, pokud jste obrazy jader pojmenovali jiným než obvyklým způsobem, který nevyhovuje zadaným kritériím hledání, nebo pokud některý ze souborů neexistuje. Problémy s nastavením a používáním zástupných znaků nespádají do instalační podpory.

### 8.3.2 Soubor `device.map`

Výše zmíněný soubor `device.map` mapuje zařízení pojmenovaná podle notace programu GRUB na jména podle Linuxové notace. Pokud váš systém má jak IDE tak SCSI zařízení, GRUB zkouší určit pořadí zavádění podle určitého algoritmu. Bohužel GRUB není schopen získat tuto informaci z BIOSu. Ukládá proto pořadí zařízení, ze kterých se zavádí systém do souboru `/boot/GRUB/device.map`. Na systémech kde je BIOS nastaven tak, aby zaváděl OS z IDE disků a až poté z SCSI, by soubor vypadal takto:

```
(fd0)  /dev/fd0
(hd0)  /dev/hda
(hd1)  /dev/hdb
(hd2)  /dev/sda
(hd3)  /dev/sdb
```

Jestliže GRUB zavádí systém podle `device.map` a narazí na problém, zkontrolujte pořadí zařízení v tomto souboru, a případně změňte jejich pořadí v GRUB shellu. Jakmile nainstalujete systém, můžete změnit pořadí v modulu konfigurace zavaděče programu YaST, nebo ve vašem oblíbeném editoru.

Po změnách provedených v souboru `device.map` musíte aktualizovat instalaci zavaděče. To provedete následujícími příkazy:

```
GRUB -batch < /etc/GRUB.conf
```

### 8.3.3 Soubor `/etc/grub.conf`

Kromě souborů `menu.lst` a `device.map` GRUB používá pro uložení svého nastavení také soubor `GRUB.conf`. V tomto souboru jsou uložena data o místech, kam má příkaz GRUB uložit kód zavaděče:

```
root (hd0,4)
install /GRUB/stage1 d (hd0) /GRUB/stage2 0x8000
(hd0,4) /GRUB/menu.lst
quit
```

Druhá a první řádka jsou napsané v jedné řádce. Jednotlivé údaje mají následující význam:

**\mbox{root (hd0, 4) }** Tato položka říká programu GRUB, že veškeré následující příkazy se týkají prvního logického oddílu na prvním disku, na kterém jsou uloženy soubory pro zavádění.

**\mbox{install parametr}** Zde se říká, že GRUB má spustit svůj interní příkaz `install` a určuje, kam uložit kód. Zavaděč prvního stupně zapsat do MBR prvního disku (`/GRUB/stage1 d (hd0)`), a na paměťovou adresu `0x8000` nahrát zavaděč druhé fáze (`/GRUB/stage2 0x8000`). Poslední parametr (`(hd0, 4) /GRUB/menu.lst`) ukazuje, kde je uložen soubor `s menu`.

### 8.3.4 GRUB shell

GRUB sestává ze dvou částí: zavaděče a běžného Linuxového programu (`/usr/sbin/GRUB`). Tomuto programu se také říká `GRUB shell`. Program obsahuje interní příkazy pro zapsání kódu zavaděče na disk nebo disketu (`install a setup`). Jinými slovy, tyto vnitřní příkazy můžete spustit v rámci GRUB shellu na běžícím Linuxovém stroji. Nicméně tyto příkazy jsou také dostupné během zavádění pomocí programu GRUB - ještě před tím, než je nastartován Linux. Díky tomu je mnohem jednodušší opravit vadný systém.

Výše zmíněný algoritmus pro mapování zařízení se použije pouze tehdy, pokud GRUB spouští svůj shell. GRUB načte soubor `device.map` a namapuje jména používaná programem GRUB na Linuxová jména. Každé zařízení je na jednom řádku. Pokud máte potíže se zaváděním systému, zkontrolujte zda pořadí zařízení uvedených v `device.map` koresponduje s nastavením v BIOSu počítače. Soubor najdete v adresáři `/boot/GRUB/`. Chcete-li vědět o tomto tématu více, přečtěte si sekci 8.3.2 na předchozí straně.

### 8.3.5 Nastavení hesla pro zavádění

Protože GRUB umí během zavádění přistupovat na různé souborové systémy, můžeme ho použít i pro čtení souborů, které by za normálních okolností nebyly přístupné - na běžícím systému by uživatel potřeboval mít oprávnění uživatele `root`. Abyste tomuto zamezili, nastavte si heslo pro zavaděč GRUB. Tímto můžete zabránit neautorizovaným osobám v přístupu k souborům během zavádění, a předejít zavedení jiného než implicitního operačního systému.

Heslo vytvoříte tak, že se přihlásíte jako `root` a provedete následující kroky:

1. Spustíte GRUB shell a zašifrujete heslo:

```
GRUB> md5crypt
Password: ****
Encrypted: $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

2. Vložte zašifrovaný řetězec do globální sekce souboru `menu.lst`:

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password -md5 $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

Od teď nelze spouštět příkazy programu GRUB při zavádění systému bez znalosti hesla. Oprávnění získáte po stisknutí **P** a zadání hesla. Uživatelé ale stále mohou zavádět libovolné nainstalované OS bez omezení.

3. Abyste zamezili zavedení některých operačních systémů, přidejte ke každé položce, kterou chcete mít chráněnou heslem, řádek `lock`. Jako v následujícím příkladě:

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

Po restartování počítače se při pokusu o zavedení OS z takto označené položky zobrazí chybová hláška:

```
Error 32: Must be authenticated
```

Česky tedy:

```
Chyba 32: Musíte zadat heslo
```

Vraťte se do menu stisknutím **Enter**. Zde stiskněte **P** a zadejte heslo. Vybraný OS (v našem případě Linux) se zavede po zadání hesla.

---

## Důležité

### Heslo pro zavádění a úvodní obrazovka

Nastavení hesla vypne implicitní zobrazování grafické úvodní obrazovky (boot splash screen).

---

## Důležité

## 8.4 Konfigurace zavaděče pomocí programu YaST

Tento modul programu YaST zjednodušuje konfiguraci nastavení zavaděče. Neměli byste ale s tímto modulem experimentovat pokud nerozumíte základním konceptům, ke kterým se vztahuje. Přečtěte si odpovídající části *Příručka správce systému* před tím, než budete měnit konfiguraci zavaděče. Následující text pokrývá hlavně standardně instalovaný zavaděč GRUB.

---

## Důležité

### Změna nastavení

Neměňte způsob ani nastavení zavádění systému u běžícího počítače, pokud si nejste opravdu jisti, že víte co děláte.

---

## Důležité

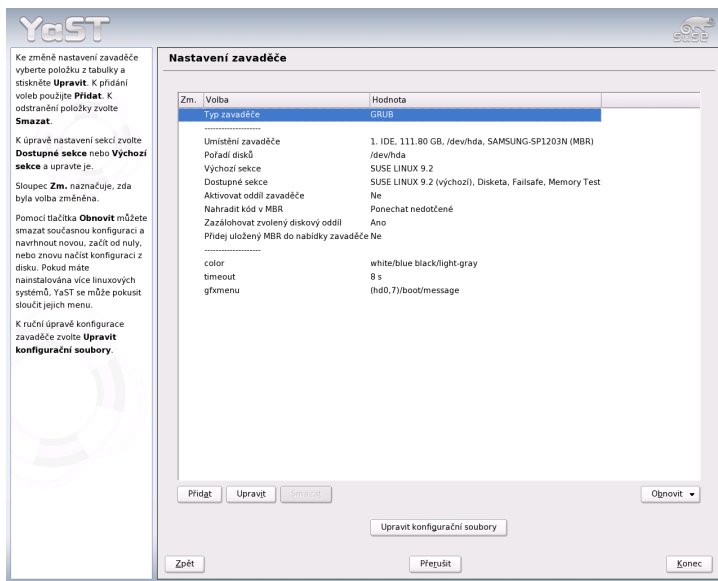
V Řídicím středisku programu YaST vyberte 'Systém' → 'Konfigurace zavaděče'. Bude načtena a zobrazena stávající konfigurace zavaděče, a můžete provést potřebné změny (viz obr.8.1 na následující straně).

### 8.4.1 Obrazovka nastavení zavaděče

Tabulka obsahující konfigurační data má tři sloupce. Levý sloupec 'Zm.' zobrazuje informaci o tom, která nastavení uvedená v prostředním sloupci byla změněna. Pro přidání volby klikněte na 'Přidat'. Ke změně hodnoty existujícího nastavení ho vyberte myší a klikněte na 'Upravit'. Pokud nechcete použít už existující volbu, vyberte ji a klikněte na 'Smazat'.

Volba 'Obnovit' vpravo pod konfiguračním oknem nabízí následující možnosti:





**Obrázek 8.1:** Konfigurace zavaděče pomocí programu YaST

**Navrhnout novou konfiguraci** Vygeneruje návrh nové konfigurace. Starší verze linuxových operačních systémů nebo jiné systémy které budou nalezeny na oddílech v počítači budou přidány do menu zavaděče, což vám umožní zavést Linux nebo jeho starší zavaděč. Tato volba vás pak zavede do druhého menu zavaděče.

**Začít od nuly** Pomůže vám vytvořit zcela novou konfiguraci. Nebude vygenerován žádný návrh.

**Znovu načíst konfiguraci z disku** Pokud jste již udělali nějaké změny a nejste spokojeni s výsledkem, můžete znovu načíst stávající konfiguraci.

#### Navrhnout a sloučit s existujícími menu GRUB

Pokud je již v počítači instalován jiný operační systém nebo starší Linux na jiném oddílu, menu bude vygenerováno se zohledněním starších položek, i s položkami nového systému SUSE LINUX. Celá operace může zabrat určitý čas. Tuto volbu není možné použít, pokud máte v počítači instalován zavaděč LILO.

**Obnovit MBR disku** MBR (master boot record) zaváděcí sektor disku, který byl uložen na pevný disk, bude zapsán zpět na jeho místo.

Pro editaci relevantních konfiguračních souborů v textovém editoru použijte položku 'Upravit konfigurační soubory' pod konfiguračním oknem. Pro editaci souboru jej vyberte, proveďte změny a klikněte na 'OK' pro uložení změn. Konfiguraci zavaděče můžete ukončit bez uložení kliknutím na tlačítko 'Přerušit'. 'Zpět' vás zavede zpět do hlavního okna.

## 8.4.2 Volby nastavení zavaděče

Pro méně zkušené uživatele je konfigurace pomocí programu YaST mnohem jednodušší než přímá editace konfiguračních souborů. Vyberte požadovanou volbu a klikněte na 'Upravit' pro otevření dialogu ve kterém můžete změnit nastavení tak jak potřebujete. Klikněte na 'OK' pro potvrzení změn a návratu do hlavního menu, kde můžete upravit ostatní volby. Dostupnost jednotlivých voleb závisí na použitém zavaděči. Následující výčet obsahuje některé z voleb zavaděče GRUB:

**Typ zavaděče** Tuto volbu můžete použít pro přepínání mezi zavaděčem GRUB a LILO. Zobrazí se vám nové konfigurační okno ve kterém můžete specifikovat jak bude tato změna provedena. Například převedení stávající konfigurace zavaděče GRUB do podobné konfigurace zavaděče LILO. Některá nastavení mohou ale být ztracena pokud neexistuje ekvivalentní náhrada dané volby. Můžete také vytvořit novou konfiguraci od začátku nebo vygenerovat a upravit návrh nové konfigurace.

Pokud spustíte konfiguraci zavaděče z běžícího systému, můžete nahrát nastavení z disku. Pokud se v průběhu úprav rozhodnete pro návrat k originálnímu zavaděči, je ještě možné nahrát jeho původní konfiguraci. Nicméně tato varianta je dostupná pouze do opuštění modulu konfigurace zavaděče.

**Umístění zavaděče** Použitím tohoto dialogu můžete specifikovat umístění zavaděče: do hlavního zaváděcího sektoru (MBR), do zaváděcího sektoru bootovacího oddílu (je-li k dispozici), do zaváděcího sektoru kořenového oddílu nebo na disketu. Pokud chcete zadat jiné umístění, vyberte 'Ostatní'. Více informací o zavaděči GRUB najdete v *Příručka správce systému*.

**Pořadí disků** Pokud má váš počítač více než jeden disk, můžete zadat jejich pořadí pro zavádění systému jak je nastaveno v BIOSu.

**Výchozí sekce** V této volbě můžete nastavit které jádro nebo operační systém se má spouštět pokud nebude v zaváděcím menu vybrána jiná volba. Tento systém je zaveden po uplynutí nastavené několikavteřinové prodlevy. Vyberte tuto volbu a klikněte na 'Upravit', zobrazí se vám seznam všech položek ze zaváděcího menu. Vyberte jednu položku z menu a klikněte na 'Nastavit jako výchozí'. Klikněte na 'Upravit' a proveďte případné další změny dalších parametrů.

**Dostupné sekce** Umožňuje editaci položek stávajícího zaváděcího menu. Pokud kliknete po výběru této položky na 'Upravit', otevře se dialog shodný s editačním oknem zobrazeným po volbě 'Výchozí sekce'.

**Aktivovat oddíl zavaděče** Tato volba aktivuje oddíl jehož startovací sektor obsahuje zavaděč, nezávisle na adresáři, ve kterém jsou uloženy další soubory zavaděče (`/boot` nebo kořenový adresář `/`).

**Nahradit kód v MBR** Vyberte jestli chcete přepsat kód v hlavním zaváděcím sektoru, což může být nezbytné pokud jste změnil umístění zavaděče.

**Zálohovat zvolený diskový oddíl** Zazálohuje změněné oblasti diskového oddílu (typicky zaváděcí sektor).

#### **Přidat uložený MBR do nabídky zavaděče**

Přidá dříve uložený hlavní zaváděcí sektor (MBR) do nabídky zavaděče.

Dále můžete změnit položku 'timeout', která specifikuje délku prodlevy při startu systému ve vteřinách, po jejímž uplynutí je zaveden systém specifikovaný volbou 'Výchozí sekce'. Další volby můžete přidávat pomocí tlačítka 'Přidat'. Používání dalších voleb vyžaduje hlubší rozsah znalostí a není pokryto tímto textem. Více informací lze nalézt v dokumentaci zavaděče GRUB, respektive LILO (`grub(8)` nebo `lilo(8)`). Podrobný manuál pro zavaděč GRUB je k dispozici na adrese <http://www.gnu.org/software/grub/manual/>.

## **8.5 Odinstalace zavaděče LILO nebo GRUB**

Při odinstalaci programů GRUB a LILO se do zaváděcího sektoru (kde sídlí zavaděč) musí nahrát původní obsah. SUSE LINUX uchovává platnou původní zálohu obsahu tohoto sektoru. YaST modul pro zavaděče lze použít pro vytvoření

zálohy, integraci této zálohy do menu zavaděče a nebo pro obnovení standardního MBR. Tento modul je popsán v kapitole věnující se instalaci systému.

### Varování

Záloha zaváděcího sektoru se stane neplatnou, jestliže na oddíl kde leží zaváděcí sektor nainstalujeme nový souborový systém. Tabulka rozdělení diskových oddílů v záloze MBR je nepoužitelná, pokud jsme od doby vytvoření zálohy změnili rozložení oddílů. Tyto staré zálohy jsou jako časovaná bomba. Je lepší je mazat hned jak změníme rozložení disku.

Varování

## 8.5.1 Obnova MBR (DOS, Win9x/ME, OS/2)

Obnovit MBR DOSu, OS/2 nebo Windows je velice snadné. Pouze zadejte příkaz DOSu (který je dostupný od verze 5.0):

```
fdisk /MBR
```

nebo na OS/2:

```
fdisk /newmbr
```

Tyto příkazy zapíší do MBR pouze prvních 446 bytů (kód zavaděče) a ponechají tabulku rozdělení disků nedotčenou. Pokud však je MBR označen jako neplatný kvůli špatnému magickému číslu), nastaví se tabulka na hodnotu nula. Po obnově MBR zkontrolujte zda je požadovaný oddíl nastaven jako zaváděcí (znovu pomocí fdisk). Tento příznak požaduje kód startující DOS, Windows a OS/2.

## 8.5.2 Obnova MBR v Windows XP

Zaveďte systém z instalačního CD Windows XP a stiskněte během startu (**Ⓡ**) pro spuštění konzole pro zotavení. Vyberte vaši instalaci Windows XP ze seznamu a zadejte heslo administrátora. Poté z příkazové řádky spusťte příkaz `FIXMBR` a poté potvrďte stiskem **y**. Nyní restartujte počítač pomocí příkazu `exit`.

## 8.5.3 Obnova MBR v Windows 2000

Zaveďte systém z instalačního CD Windows 2000 a stiskněte (**Ⓡ**) a poté v dalším menu (**Ⓢ**). Zvolte ze seznamu vaši instalaci Windows 2000 a zadejte heslo pro administrátora. Do promptu zadejte příkaz `FIXMBR` a potvrďte tuto volbu pomocí **y**. Následně můžete restartovat počítač pomocí `exit`.

### 8.5.4 Zavedení systému Linux po obnovení MBR

Po obnovení standardního Windows MBR můžete nastavit jeden z Linuxových zavaděčů, abyste mohli dále používat instalovaný Linuxový systém.

#### GRUB

I když je nainstalován v MBR, ukládá GRUB svá data pro zaváděcí fázi 1 na linuxový oddíl. Po obnovení MBR pomocí YaST nebo ve Windows s nástroji zmíněnými výše, musíte označit oddíl, kde leží GRUB, jako aktivní.

#### LILO

Po obnovení MBR můžete znovu nainstalovat LILO, pokud máte uložený záložní soubor. Nejprve zkontrolujte jestli velikost souboru je přesně 512 bytů a poté obnovte sektor (nejdříve však provedeme zálohu do +jmeno-noveho-souboru). Pomocí příkazů:

- Jestliže LILO leží na oddíle yyyy (např. hda1, hda2,...):

```
dd if=/dev/yyyy of=jmeno-noveho-souboru bs=512 count=1
dd if=jmeno-souboru-se-zalohou of=/dev/yyyy
```

- Jestliže LILO leží v MBR na disku zzz (např., hda, sda):

```
dd if=/dev/zzz of=jmeno-noveho-souboru bs=512 count=1
dd if= of=jmeno-souboru-se-zalohou /dev/zzz bs=446
count=1
```

Poslední příkaz je bezpečná verze předešlého - nepřepisuje tabulku oddílů. Nyní opět označte oddíl jako aktivní pomocí programu `fdisk`.

## 8.6 Vytvoření startovacího CD

V některých případech se může stát, že nelze systém spustit pomocí standardních zavaděčů LILO nebo GRUB na instalovaných do MBR disku. V takových případech obvykle nastupuje použití startovací diskety. U novějších jader je však vytvoření startovací diskety kvůli nedostatku místa na disketě často nemožné. Pokud máte k dispozici vypalovací mechaniku, můžete si místo startovací diskety vytvořit startovací CD.

K vytvoření startovacího CD se zavaděčem GRUB je potřeba zvláštní forma *stage2* nazývaná *stage2\_eltorito* a upravený soubor *menu.lst*. Klasické soubory *stage1* a *stage2* nejsou potřebné.

Vytvořte si adresář určený pro obsah ISO obrazu.

```
cd /tmp
mkdir iso
```

V adresáři */tmp* si vytvořte podadresář GRUB :

```
mkdir -p iso/boot/grub
```

Překopírujte soubor *stage2\_eltorito* do adresáře *grub* :

```
cp /usr/lib/grub/stage2_eltorito iso/boot/grub
```

Překopírujte jádro (*/boot/vmlinuz*), *initrd* (*/boot/initrd*) a soubor */boot/message* do adresáře *iso/boot/* :

```
cp /boot/vmlinuz iso/boot/
cp /boot/initrd iso/boot/
cp /boot/message iso/boot/
```

Aby byly tyto soubory dostupné pro GRUB, překopírujte soubor *menu.lst* do adresáře *iso/boot* a upravte jednotlivé položky tak, aby ukazovaly na CD mechaniku. To uděláte tak, že všechny odkazy na pevný disk (např. *(hd\*)*) zaměňte za jméno CD mechaniky (*(cd)*):

```
gfxmenu (cd)/boot/message
timeout 8
default 0

title Linux
    kernel (cd)/boot/vmlinuz root=/dev/hda5 vga=794 resume=/dev/hda1
    splash=verbose showopts
    initrd (cd)/boot/initrd
```

ISO můžete například vytvořit následujícím příkazem:

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -o grub.iso iso
```

Soubor `grub.iso` vypalte svým oblíbeným vypalovacím programem na CD.

## 8.7 Grafická konzole SUSE

Od verze SUSE LINUX 7.2, má SUSE při nastavení parametru `vga=<value>` první konzoli grafickou. V případě instalace pomocí programu YaST se tento parametr nastaví automaticky. Grafickou konzoli lze vypnout třemi způsoby:

**Vypnutí grafiky podle potřeby** V příkazové řádce zadejte příkaz `echo 0 >/proc/splash`. Opět ji aktivujete příkazem `echo 1 >/proc/splash`.

**Vypnutí jako přednastavená možnost**

Stačí přidat do konfigurace zavaděče parametr jádra `splash=0`. Více informací najdete v kapitole 8 na straně 163. Pokud dáváte přednost textovému režimu z předchozích verzí, zadejte `vga=normal`.

**Vypnutí úplně a na vždy** Přeložte si jádro a vypněte volbu 'Use splash screen instead of boot logo' v menu 'frame-buffer support'.

### Tip

Pokud si vypnete podporu pro framebuffer, pak se splash screen vypne automaticky. V případě, že si budete sami kompilovat jádro, nebudete pro takto upravené jádro moci využít instalační podporu.

### Tip

## 8.8 Řešení problémů

V této části jsou popsány nejčastější problémy související s používáním zavaděče GRUB a jejich řešení. Řešení nejčastějších problémů najdete v databázi instalační

podpory <http://portal.suse.de/sdb/cz/index.html>. Můžete použít také funkci hledání. Při hledání v <https://portal.suse.com/PM/page/search.pm> použijte klíčová slova jako GRUB, boot a zavaděč.

**GRUB a XFS** XFS neponechá na oddílu žádné místo pro `stage1`, proto nenastavujte XFS oddíl jako umístění zavaděče. Tento problém se dá vyřešit vytvořením zvláštního startovacího oddílu, který nebude naformátován na XFS.

**GRUB a JFS** Kombinace zavaděče GRUB a souborového systému JFS bývá problematická. Doporučujeme použít zvláštní startovací oddíl (`/boot`) a naformátovat jej např. na Ext2. Pak GRUB nainstalujte na tento oddíl.

#### **GRUB Hláška `\mbox{GRUB Geom Error}`**

GRUB zjišťuje geometrii připojeného disku při startu systému. Občas BIOS vrátí nekorektní informace a GRUB nahlásí chybu `GRUB Geom Error`. V takovém případě použijte zavaděč LILO nebo proveďte update BIOSu. Podrobnější informace o tomto problému najdete v databázi instalační podpory pod klíčovým slovem LILO.

GRUB tuto chybu hlásí také v případě instalace linuxového systému na BIOSem neregistrovaném disku. `stage1` se zavede, ale `stage2` není nalezen. Tento problém vyřešíte registrací disku v BIOSu.

#### **Kombinovaný systém s IDE i SCSI nestartuje**

Během instalace může YaST špatně detekovat startovací sekvenci disků (a vy ji nemůžete opravit). Například GRUB může `/dev/hda` označit jako `hd0` a `/dev/sda` jako `hd1`, přestože je startovací sekvence v BIOSu nastavena jinak (SCSI před IDE).

V takovém případě použijte příkazovou řádku zavaděče GRUB. Trvalé změny provedete po spuštění systému editací souboru `device.map`. Pak překontrolujte jména zařízení v souborech `/boot/GRUB/menu.lst` a `/boot/GRUB/device.map` a přeinstalujte zavaděč příkazem:

```
grub -batch < /etc/grub.conf
```

**Start Windows z druhého disku** Některé operační systémy jako např. Windows umí startovat pouze z prvního disku. Pokud takový operační systém chcete nainstalovat na jiný než první disk, musíte pozměnit logické pořadí disků v konfiguračním souboru zavaděče.

...

```
title windows
```



```
map (hd0) (hd1)
map (hd1) (hd0)
chainloader (hd1,0)+1
...
```

Ve výše uvedeném příkladu startuje Windows z druhého disku. Z tohoto důvodu je přenastaveno logické pořadí disků pomocí `map`. Tato změna nijak neovlivní soubor nabídku zavaděče GRUB, takže je nutné ještě provést zvláštní nastavení pro `chainloader`.

## 8.9 Další informace

Více informací o programu GRUB v angličtině, němčině a japonštině získáte na adrese <http://www.gnu.org/software/grub/>. Online manuál je pouze v angličtině. Můžete se také podívat na stránky podpory zákazníků na adrese <http://portal.suse.com/sdb/cz/index.html> a vyhledávat informace podle klíčového slova GRUB.



# Linuxové jádro

Jádro je v systému odpovědné za celou řadu procesů. V této kapitole nenajdete návod, jak se stát linuxovým *hackerem*, ale pouze informace jak bezbolestně provést update jádra, jak jádro překompilovat a nainstalovat. Také zde najdete popis některých základních parametrů jádra a postup, jak v případě potřeby spustit systém s předcházející verzí jádra.

9.1	Update jádra . . . . .	188
9.2	Zdrojové texty jádra . . . . .	189
9.3	Konfigurace jádra . . . . .	189
9.4	Moduly jádra . . . . .	190
9.5	Nastavení konfigurace jádra . . . . .	193
9.6	Překlad jádra . . . . .	193
9.7	Instalace jádra . . . . .	194
9.8	Úklid po překladu jádra . . . . .	194

Jádro nebo-li kernel SUSE najdeme na korektně nainstalovaném systému v adresáři `/boot`. Pokud nechcete experimentovat s různými vlastnostmi nebo ovladači, není obvykle potřeba překládat vlastní jádro.

Chování nainstalovaného jádra lze ovlivnit parametry jádra. Například parametr `desktop` nastavuje kratší intervaly předělování tiků časovače, což vede k subjektivnímu zrychlení systému. Informace o parametrech najdete v adresáři `/usr/src/linux/Documentation` po instalaci balíčku `kernel-source`.

S jádrem je nainstalováno několik souborů `Makefiles`. Zvolte nastavení hardwaru a vlastností jádra. K těmto nastavením je potřeba skutečně velmi dobře znát svůj počítač.

## 9.1 Update jádra

Update jádra získáte ve formě RPM balíku z FTP serveru společnosti SUSE nebo některého z jeho mirrorů, např.: `ftp://ftp.gwdg.de/pub/linux/suse/`. Pokud nevíte, jaké jádro v současné době používáte, můžete to zjistit příkazem:

```
cat /proc/version
```

Zároveň si můžete nechat vypsát k jakému balíku vaše aktuální jádro `/boot/vmlinuz` patří:

```
rpm -qf /boot/vmlinuz
```

Před instalací nového jádra je vhodné zazálohovat si `initrd` současného jádra i samotné jádro. To provedete jako uživatel `root` následujícími příkazy:

```
cp /boot/vmlinuz-$(uname -r) /boot/vmlinuz.old  
cp /boot/initrd-$(uname -r) /boot/initrd.old
```

Balík s jádrem nainstalujete příkazem:

```
rpm -Uvh Jmeno_baliku
```

Od verze 7.3 je jako standardní souborový systém používán `reiserfs`, jehož podporu je nutné umístit na ramdisku. To uděláte příkazem `mk_initrd`. U aktuální verze se tento příkaz provede automaticky při instalaci jádra.

Abyste mohli spustit starší jádro, musíte správně nastavit zavaděč (více informací najdete v 8 na straně 163). Pak již můžete spustit systém s novým jádrem.

Reinstalace jádra z instalačního CD nebo DVD je podobná, pouze RPM jádra překopírujete z adresáře `boot` na CD 1 nebo DVD. Dále pokračujte podle postupu výše. Pokud systém ohlásí, že již máte nainstalováno jádro novější než instalovaná verze, přidejte k instalačnímu příkazu ještě volbu `--force`.

## 9.2 Zdrojové texty jádra

Pro vlastní sestavení jádra musí být nainstalovány následující balíky: zdrojové texty `kernel-source`, překladač jazyka C `gcc`, GNU binutils `binutils` a hlavičkové (include) soubory pro překladač jazyka C `glibc-devel`. Instalace překladače jazyka C je vhodná i všeobecně, protože jazyk C k unixovým systémům historicky patří.

Zdrojové texty jádra se očekávají v adresáři `/usr/src/linux`. Pokud je hodláte modifikovat a přejete si mít na disku více verzí zdrojových textů spolu s odpovídajícími přeloženými jádry, je pak zvykem přidělit jim jiná jména ve společném adresáři `/usr/src` (např. `/usr/src/linux1`, `/usr/src/linux2`) a jakožto `/usr/src/linux` vytvořit pouze odkaz, ukazující na právě aktivní verzi. Tento způsob instalace zajišťuje i YaST.

Důvod, proč je vhodné zachovávat jednotnou cestu ke zdrojovým souborům `/usr/src/linux` je ten, že je v tomto adresáři potřebuje mít celá řada programů, která by pak nepracovala. Jedná se zejména o systémové programy, které při svém překladu vyžadují informace ze zdrojových textů jádra.

## 9.3 Konfigurace jádra

Konfigurace jádra najdete v souboru `/boot/vmlinuz.config`. Tuto konfiguraci můžete podle vlastního přání změnit. Nejdříve jako uživatel `root` proveďte příkaz:

```
cp /boot/vmlinuz.config /usr/src/linux/.config
```

Pak přejděte do adresáře `/usr/src/linux` a spusťte příkaz `make oldconfig`.

Alternativním postupem, jak získat konfiguraci současného jádra je příkaz:

```
zcat /proc/config.gz >gt; /usr/src/linux/.config
```

Konfigurační nástroje jádra nastavení načtou ze souboru `.config`. Tento soubor však popisuje pouze jádro a nikoli moduly, které obsahoval `kernmod`. Pokud chcete překládat nové moduly, musíte je vybrat ručně.

Jádro lze konfigurovat třemi způsoby:

- Z příkazové řádky
- Z menu v textovém módu
- Z menu pod X Window

### 9.3.1 Konfigurace z příkazové řádky

Ke konfiguraci jádra vstupte do adresáře `/usr/src/linux` a zadejte následující příkaz:

```
make config
```

Dále budete dotazováni na celou řadu vlastností, které má mít nové jádro. Odpovědět se dá: buď jednoduše *ano* -- (y) a *ne* -- (n), případně ještě *module* -- (m). Poslední případ říká, že ovladač nebude pevně spojen s jádrem, ale přeložen jako samostatný modul. Jak již bylo vysvětleno, moduly potřebné pro start musí být součástí jádra, a proto u nich odpovíte vždy (y). Stisknutí kterékoli jiné klávesy mimo těchto tří vypíše krátkou nápovědu o právě konfigurované volbě.

### 9.3.2 Konfigurace v textovém módu

Pohodlnější je konfigurace jádra pomocí menu, to se dělá příkazem `make menuconfig`. Výhoda je v tom, že nemusíte kvůli jedné otázce procházet celý dialog nebo ho opakovat po jediné chybě.

### 9.3.3 Konfigurace pod X Window

Pokud máte nainstalován systém X Window `xf86` a rovněž Tcl/Tk `tcl` a `tk`), můžete zadat grafickou alternativu předchozí možnosti příkazem `make xconfig`.

Pod X Window je konfigurace jádra ještě příjemnější. Nezapomeňte přitom pracovat jako uživatel `root`.

## 9.4 Moduly jádra

Aby zařízení pracovalo, musí pro něj v systému existovat *ovladač*, pomocí kterého k němu systém (v Linuxu jádro) přistupuje. Možné způsoby integrace ovladačů do systému lze:

- Ovladač může být zakompilován přímo do jádra. Takové jádro se pak nazývá *monolitické* (*in kernel*). některé ovladače jsou dostupné pouze v této formě.

- Ovladače lze zavést do jádra na požádání jako moduly. Takové jádro se pak nazývá *modulární*. Má tu velkou výhodu, že se zavedou pouze potřebné ovladače a neobsahuje tak nic nepotřebného.

Který ovladač se zakompiluje do jádra a který jako modul je definováno v konfiguraci jádra. V zásadě by mělo platit, že části, které nejsou přímo potřeba běhu k systému, by měly být zaváděny jako moduly. Tak se zajistí, že jádro není příliš velké pro zavedení BIOSem nebo zavaděčem. Ovladače pro `ext2`, SCSI mechaniky a SCSI subsystém by měly být zakompilovány do jádra. Naopak podpora `isofs`, `msdos` nebo zvuku patří k typickým částem zaváděným jako moduly.

Moduly jádra se nacházejí v adresáři `/lib/modules/<verze>`, kde `verze` je aktuální verze jádra.

### 9.4.1 Detekce hardwaru příkazem `hwinfo`

Příkazem `hwinfo` můžete zjistit hardware vašeho systému a zvolit správné ovladače. Rychlou nápovědu k příkazu získáte zadáním `hwinfo--help`. Pokud potřebujete např. informaci o SCSI zařízeních, zdajte příkaz `hwinfo--scsi`. Všechny tyto informace také samozřejmě najdete v modulu informací o hardwaru programu YaST.

### 9.4.2 Práce s moduly

Pro práci s moduly se používají tyto příkazy:

**insmod** Příkazem `insmod` se zadaný modul zavede. Hledá se přitom v adresáři `/lib/modules/verze_jadra`. (Tento příkaz i následující se však většinou nevolají samostatně, ale obecnějším příkazem `modprobe`, viz dále.)

**rmmmod** Odstraní zadaný modul. To ovšem není možné, pokud je tento modul používán. Například není možné odstranit modul `isofs`, pokud je stále ještě připojeno CD.

**depmod** Tento příkaz vytvoří soubor se jménem `modules.dep` v adresáři `/lib/modules/verze_jadra`, kde jsou definovány závislosti mezi jednotlivými moduly. Tím se zajistí, že při zavedení určitého modulu se také automaticky zavedou všechny závislé moduly.

**modprobe** Zavádí a odstraňuje moduly s ohledem na vzájemné závislosti. Poskytuje též řadu dalších služeb, jako postupné zkoušení více modulů stejného typu, než se jeden osvědčí. Na rozdíl od zavádění programem `insmod` pracuje program `modprobe` se souborem `/etc/modprobe.conf`. V současné době představuje `modprobe` doporučený nástroj k zavádění modulů. Podrobné vysvětlení jeho jednotlivých možností najdete na příslušných manuálových stránkách.

**lsmod** Ukazuje, které moduly jsou právě zavedeny a kolik dalších modulů je používá. Moduly, zavedené kernelovým démonem, jsou označeny jako `autoclean`, což naznačuje, že budou automaticky odstraněny, pokud nejsou používány a vyprší jim povolená doba nečinnosti.

**modinfo** Zobrazí informace o modulu. Protože jde o informace získané přímo od modulu, zobrazují se pouze informace z modulu. Mohou obsahovat jméno autora, popis, licenci, parametry, závislosti a aliasy.

### 9.4.3 Soubor `/etc/modprobe.conf`

Zavádění modulů ovlivňují soubory `/etc/modprobe.conf` a `/etc/modprobe.conf.local` a adresář `/etc/modprobe.d`. Více najdete v manuálové stránce `manmodprobe.conf`. V tomto souboru musí být zadány všechny parametry modulů přistupujících k hardwaru. Některé moduly, např. ovladač CD mechaniky nebo síťové karty, mohou vyžadovat zvláštní parametry. Možné parametry jsou popsány ve zdrojových kódech jádra. Po instalaci balíčku `kernel-source` najdete potřebné informace v adresáři `/usr/src/linux/Documentation`.

### 9.4.4 Kmodzavaděč modulů jádra

Zavaděč modulů jádra je jeden z nejelegantnějších způsobů práce s moduly. `Kmod` (*kernel module loader*) zajišťuje sledování na pozadí a stará se o správné zavádění potřebných modulů pomocí příkazu `modprobe`.

`Kmod` aktivujete volbou 'Kernel module loader' (`CONFIG_KMOD`) v konfiguraci jádra. `Kmod` neodstraňuje moduly automaticky. Omezení pro něj představuje pouze velikost RAM. Z toho důvodu je pro servery se zvláštními funkcemi lepším řešením monolitické jádro s několika ovladači.



## 9.5 Nastavení konfigurace jádra

Dokumentace k jednotlivým detailům konfigurace jádra se nachází u zdrojových textů jádra v adresáři `/usr/src/linux/Documentation`. Zde máte také jistotu, že se jedná o poslední dokumentaci k instalované verzi.

## 9.6 Překlad jádra

Doporučujeme generovat rovnou komprimované jádro `bzImage`. Pomáhá to také v případech, kdy systém nezvládne pracovat s velkým jádrem `zImage` v obyčejném binárním tvaru a hlásí `Kernel too big` nebo `System is too big`.

Po konfiguraci jádra podle vašich představ spustíte překlad:

```
make clean
make bzImage
```

Tyto dva příkazy lze napsat na příkazovou řádku:

```
make clean bzImage
```

```
make clean bzImage
```

Po úspěšné kompilaci najdete jádro v `/usr/src/linux/arch/<arch>/boot`. Obraz jádrasoubor obsahující jádrose jmenuje `bzImage`.

Pokud soubor s jádrem nemůžete najít, došlo pravděpodobně během kompilace k chybě. V interpretu Shell lze výstup hlášení kompilace jádra zapisovat do souboru `kernel.out`:

```
make bzImage 2> &1 | tee kernel.out
```

V případě nastavení kompilace části ovladačů ve formě modulů, musíte moduly zvlášť překompilovat příkazem `make modules`.

## 9.7 Instalace jádra

Po kompilaci jádra je nutné jádro umístit tak, aby bylo spustitelné. Jádro se musí nacházet v adresáři `/boot`. To provedete příkazem:

```
INSTALL_PATH=/boot make install
```

Dále je potřeba nainstalovat moduly jádra. zadejte příkaz `make modules_install`, kterým je přkopírujete do adresáře `/lib/modules/<verze>`. Pokud jste kompilovali stejnou verzi jádra jako bylo již instalované, dojde k přepsání starých modulů. Staré moduly však lze s původním jádrem kdykoliv doinstalovat z CD.

### Tip

Abyste předešly nečekaným efektům, ujistěte se, že jsou moduly zkompilevané přímo do jádra odstraněny z `/lib/modules/<verze>`. Toto je jeden z hlavních důvodů, proč se kompilace jádra doporučuje pouze pokročilejším uživatelům.

### Tip

Staré jádro (nyní `/boot/vmlinuz.old`) můžete pomocí zavaděče GRUB spustit zadáním položky `Linux.old` do konfiguračního souboru zavaděče `/boot/grub/menu.lst`. Postup je podrobně popsán v kapitole 8 na straně 163. GRUB není potřeba narozdíl od zavaděče LILO reinstalovat.

Soubor `/boot/System.map` obsahuje symboly jádra požadované moduly pro úspěšné spuštění. Soubor je závislý na aktuálním jádře. Proto pokud jste překompilevaly a nainstalovali nové jádro, přkopírujte soubor `/usr/src/linux/System.map` do adresáře `/boot`. Soubor se vytváří nově pro každé kompilované jádro. Chybové hlášení *System.map does not match current kernel* je obvykle zapříčiněno chybějícím souborem `System.map` v adresáři `/boot`.

## 9.8 Úklid po překladu jádra

Pokud nebudete výhledově znovu překládat a přejete si smazat přeložené zdrojové moduly, abyste ušetřili místo na disku, napíšete:

```
cd /usr/src/linux
make clean
```

Pokud naopak přeložené soubory na disku ponecháte, zrychlí se tím příští překlad, protože program `make` zajistí, aby se překládaly pouze změny.



# Speciální vlastnosti SUSE LINUXu

V kapitole probereme informace o detailně popíšeme některé balíčky softwaru a speciální vlastnosti, jakými je bootování z *initrd*, linuxrc a záchranný systém. Součástí je i část věnovaná lokalizaci systému a s ní souvisejícím specifickým nastavením (I18N a L10N).

10.1	Nápověda k některým zvláštním balíčkům . . . . .	198
10.2	Virtuální konzole . . . . .	207
10.3	Mapování klávesnice . . . . .	207
10.4	Lokální přizpůsobení I18N and L10N . . . . .	208

## 10.1 Návod k některým zvláštním balíčkům

Zde najdete důležité informace o balíčcích jako je například `bash` či `cron` a příkazům `ulimit` a `free`.

### 10.1.1 Balíček `bash` a `/etc/profile`

1. `/etc/profile`
2. `~/.profile`
3. `/etc/bash.bashrc`
4. `~/.bashrc`

Osobní nastavení si každý uživatel může zapsat do souboru `~/.profile` nebo do `~/.bashrc`. Aby bylo nastavení těchto souborů správné, je nezbytné zkopírovat základní nastavení z `/etc/skel/.profile` nebo `/etc/skel/.bashrc` do domovského adresáře uživatele. Je doporučeno překopírovat `/etc/skel` ihned po updatu. Aby nedošlo k ztrátě osobních nastavení, doporučuje se nejdříve provést následující příkazy:

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

Osobní nastavení je pak nutné překopírovat zpět z `*.old`.

### 10.1.2 Balíček `cron`

Tabulky programu `cron` se nyní nacházejí v `/var/cron/tabs`. `/etc/crontab` nyní slouží jako rozsáhlý konfigurační soubor systémové tabulky. Zde zadávejte jako uživatel `root` jméno počítače, který by měl v určitý čas spouštět některé příkazy podle časové tabulky. Tabulky specifické pro balíček, uložené v `/etc/cron.d`, mají stejný formát. Více v `man cron`.

Ukázka údajů v `/etc/crontab` uživatele `root`):

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

/etc/crontab nelze spustit příkazem `crontab -e`. Musíte jej nejdříve otevřít v editoru, pak změnit a uložit.

Mnoho balíčků instaluje skripty příkazového řádku do adresářů `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, a `/etc/cron.monthly`, instrukce jsou kontrolovány skriptem `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` je z hlavní systémové tabulky (`/etc/crontab`) spouštěn každých patnáct minut. To zajistí spuštění všech správných procesů včas.

úlohy, které jsou vykonávány denně, jsou z důvodů přehlednosti rozděleny do několika samostatných skriptů (`aaa_base`, `/etc/cron.daily` obsahují komponenty `backup-rpmdb`, `clean-tmp`, či `clean-vi`).

### 10.1.3 Soubory logů: logrotate a balíčky

V systému je spuštěno mnoho služeb (*démonů*), které pravidelně zaznamenávají stav systému a určitých událostí do záznamů (logovacích souborů). Tímto způsobem může administrátor pravidelně zkontrolovat stav systému v určitém časovém okamžiku, najít problémy a chyby funkcí, řešit a ladit je s velkou precizností. Záznamy - logy jsou uloženy v adresáři `/var/log`, přesně dle specifikace FHS a denně nabírají nové a nové záznamy. Balíček `logrotate` umožňuje zvýšení počtu těchto souborů a lepší kontrolu systému.

## Nastavení

Nastavení logrotate je uloženo v souboru `/etc/logrotate.conf`. Položka `include` specifikuje další soubory pro čtení. SUSE LINUX zajišťuje instalování jednotlivých balíčků do `/etc/logrotate.d` např. `apache2` (`/etc/logrotate.d/apache2`) `syslog` (`/etc/logrotate.d/syslog`).

Příklad z `/etc/logrotate.conf`:

```
# podívejte se na "man logrotate" pro další detaily
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#    monthly
#    create 0664 root wtmp
#    rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotate je kontrolován programem cron a bývá volán denně. `/etc/cron.daily/logrotate`.

---

### Důležité

Volba `create` umožňuje načíst veškerá nastavení vytvořená administrátorem v souboru `/etc/permissions*`. Zajišťuje, že nedojde ke konfliktu žádných nastavení.

---

**Důležité**



### 10.1.4 Manuálové stránky

Manuálové stránky GNU aplikací (jako je např. tar) již nejsou delší dobu spravovány a aktualizovány. Byly nahrazeny info stránkami. Pro zjištění základních příkazů programů, použijte parametr `--help`, který poskytuje rychlý přehled info stránek. Ty ovšem poskytují mnohem hlubší pohled na jednotlivé možnosti programů a vysvětlují příkazové instrukce. info je hypertextový systém vyvíjený v rámci projektu GNU. Úvod do info stránek zobrazíte jednoduchým vypsáním `info info` na příkazovou řádku. Info stránky lze prohlížet editorem Emacs, i přímo při jeho spuštění pomocí `emacs -f info`, nebo přímo v konzoli příkazem `info`. Programy jako `tkinfo`, `xinfo`, lze jednoduše prohlížet pomocí nápovědy SUSE.

### 10.1.5 Příkaz ulimit

Díky příkazu `ulimit` (*user limits*) je možné nastavit využívání zdrojů systému a zároveň si je nechat zobrazit. `ulimit` je užitečný zvláště pro omezení paměti využívané aplikacemi. Můžete zabránit aplikaci v nadměrném čerpání zdrojů, aby nemohlo dojít k zamrznutí systému.

`ulimit` může být používán s mnoha volbami. Využívání paměti omezíte některou z voleb z tabulky 10.1 na této straně.

*Tabulka 10.1: ulimit: Přidělení zdrojů uživateli*

<code>-m</code>	maximální velikost fyzické paměti
<code>-v</code>	maximální velikost virtuální paměti
<code>-s</code>	maximální velikost zásobníku
<code>-c</code>	maximální velikost core souborů
<code>-a</code>	zobrazení limitů

Nastavní platná pro celý systém zapisujte do `/etc/profile`. Zde musíte povolit vytváření core souborů, které jsou potřebné při *ladění*. Normální uživatelé hodnoty uvedené v `/etc/profile` měnit nemohou, mohou si ovšem vytvořit speciální nastavení ve vlastním `~/.bashrc`.

Příklad omezení paměti v `~/.bashrc`:

```
# Omezení fyzické paměti:
ulimit -m 98304

# Omezení virtuální paměti:
ulimit -v 98304
```

Velikost paměti musí být zadána v KB. Více informací najdete v `man bash`.

---

### Důležité

Některé shelly příkaz `ulimit` nepodporují. V tom případě využijte PAM `pam_limits`, který nabízí podobné možnosti pro omezování přidělených prostředků.

---

Důležité

## 10.1.6 Příkaz `free`

Pokud chcete zjistit, kolik paměti RAM je momentálně používáno, může být výstup programu `free` trochu matoucí. Podstatné informace naleznete v souboru `/proc/meminfo`. V moderních operačních systémech jako je Linux, se již uživatelé nedostatku paměti nemusí obávat. Koncepce *dostupné RAM* zdědil Linux z období řízení unifikovaného přístupu k paměti. Slogan *volná paměť je špatná paměť* padne Linuxu jako ulitý. Výsledkem je vlastnost systému, kdy je nesmyslné byt i jen mluvit o volné či nepoužívané paměti.

Jádro v podstatě nemá přímé informace o aplikačních či uživatelských datech. Místo toho obsluhuje aplikace a uživatelská data pomocí *stránkování*. V případě nedostatku paměti budou načítány na odkládací oddíl nebo do souborů, ze kterých je možné je číst příkazem `mmap`. (viz `man mmap`).

Jádro obsahuje také jiné cache, např. *slab cache*, používanou pro uložení síťového přístupu. To může vést k situaci, kdy jsou informace v souboru `/proc/meminfo` odlišné od reality. K většině, ale ne ke všem, lze přistupovat přes `/proc/slabinfo`.

### 10.1.7 Soubor `/etc/resolv.conf`

Rozpoznávání doménových jmen je řešeno souborem `/etc/resolv.conf`. Více najdete v kapitole 24 na straně 409.

Soubor je aktualizován výlučně skriptem `/sbin/modify_resolvconf`, což znamená, že žádný jiný program nesmí soubor `/etc/resolv.conf` upravovat přímo. Přísnost tohoto pravidla zaručuje konzistentní stav konfigurace sítě.

### 10.1.8 Nastavení programu GNU Emacs

GNU Emacs je komplexním pracovním prostředím. Více informací je k dispozici na <http://www.gnu.org/software/emacs/>. Následující sekce popisují konfigurační soubory načítané při startu GNU Emacs.

Během startu načítá Emacs množství souborů s nastaveními uživatele, administrátora systému a distributora lokalizace a předkonfigurovaných vlastností. Inicializační soubor `~/ .emacs` se nainstaluje do home adresářů uživatelů z adresáře `/etc/skel`. `.emacs` posléze čte ze souboru `/etc/skel/ .gnu-emacs`. Pro vlastní úpravy programu by si měl uživatel zkopírovat `.gnu-emacs` do svého domovského adresáře. Požadované změny by měl provést zde:

```
cp /etc/skel/.gnu-emacs ~/ .gnu-emacs
```

`.gnu-emacs` definuje soubor `~/ .gnu-emacs-custom` jako `custom-file`. V případě, že uživatel dělá změny nastavení pomocí `customize` nastavení, ukládají se pak tyto změny do `~/ .gnu-emacs-custom`.

V systému SUSE LINUX, `emacs` balíček instaluje soubor `site-start.el` do adresáře `/usr/share/emacs/site-lisp`. Soubor `site-start.el` je načítán ještě *předtím*, než je načten konfigurační soubor `~/ .emacs`. Mezi mnoha službami, které soubor `site-start.el` zajišťuje, je také automatické nahrávání speciálních konfiguračních souborů obsažených v přídatných balíčcích programu Emacs (to jsou balíčky jako např. `psgml`). Konfigurační soubory tohoto typu jsou také umístěny v `/usr/share/emacs/site-lisp`, a vždy začínají `suse-start-`. Místní administrátor může specifikovat nastavení velmi široce v souboru `default.el`.

Více informací o těchto souborech najdete v info souboru pod Emacs *Inicializačním souborem*: `info:/emacs/InitFile`. Na druhou stranu zde najdete taktéž informace o tom, jak v případě potřeby tyto soubory vypnout.

Komponenty programu Emacs jsou rozděleny do několika balíčků:

- Základní balík `emacs`.
- Obvykle by měl být instalován `emacs-x11`. Balíček obsahuje program s podporou X11 .
- `emacs-nox` naopak podporu X11 *neobsahuje*.
- `emacs-info`: Jde o online dokumentaci v info formátu.
- `emacs-el` obsahuje nekompilovanou knihovnu souborů v jazyce Emacs Lisp. Vyžadovány pro běh programu nejsou tyto soubory nejsou.
- Množství přídatných balíčků, které instalujete v případě potřeby:
  - `emacs-auctex` (pro LaTeX)
  - `psgml` (pro SGML a XML)
  - `gnuserv` (pro fungování jako klient a server) a další.

## 10.1.9 Krátký úvod do editoru vi

Editor vi je v unixovém světě považován za velmi komfortní a výkonný editor, jehož ovládání je mnohem ergonomičtější, než ovládání většiny textových editorů s grafickým rozhraním a podporou myši.

### Režimy práce

Editor vi používá tři hlavní režimy práce: *vkládací* (insert) režim, *příkazový* (normalní, command) režim a *řádkový* (ex) režim. Klávesy mají v různých režimech práce různé funkce. Za běžných okolností se vi po startu nachází v *příkazovém* režimu. První věc, kterou se uživatel musí naučit, je přepínat mezi jednotlivými pracovními režimy:

### Přechod z příkazového do vkládacího režimu

Možností je několik, patří mezi ně zápis **(A)** (z anglického append, text bude vložen za aktuální pozici kurzoru), **(I)** (z anglického insert, text bude vložen na aktuální pozici kurzoru) nebo **(O)** (text bude vložen na začátek nové řádky vytvořené za aktuální řádkou).

### Přechod z vkládacího do příkazového režimu

Stisknutí klávesy (Esc) ukončí *vkládací* režim a způsobí přechod do *příkazového* režimu.

Pokud se editor vi nachází ve *vkládacím* režimu, nelze ukončit, a proto je důležité zapamatovat si klávesu (Esc) pro jeho opuštění.

### Přechod z příkazového režimu do řádkového režimu

*Řádkový* režim editoru vi lze aktivovat zápisem dvojtečky (: ) během práce v příkazovém režimu. *Řádkový* režim je obdobou nezávislého řádkově orientovaného textového editoru využitelného k řadě jednoduchých i složitých úkolů.

### Přechod z řádkového režimu do příkazového režimu

Po vykonání příkazu v *řádkovém* režimu se editor automaticky vrátí do *příkazového* režimu. Pokud nechcete vykonat žádný příkaz, smažte dvojtečku v příkazovém řádku pomocí klávesy (←). Editor se tak vrátí do *příkazového* režimu.

Přepnout vi z *vkládacího* do *řádkového* režimu přímo, aniž by se nejdříve přepnulo do *příkazového* režimu, nelze.

Editor vi má vlastní způsob ukončení. Nemůže být ukončen během práce ve *vkládacím* režimu. Nejprve je nutno *vkládací* režim ukončit stiskem klávesy (Esc). Dále jsou dvě možnosti:

1. *Ukončení bez uložení změn:* aby byl editor ukončen bez uložení jakýchkoliv změn v souborech, запиšte v *příkazovém* režimu sekvenci (: ) (Q) (!). Vykřičník (!) způsobí, že bude vi ukončen bez ohledu na jakékoliv změny v editovaných souborech.
2. *Uložit změny a ukončit editor:* Možností, jak uložit změny a ukončit editor, je několik. V *příkazovém* režimu запиšte sekvenci (Z) (Z). V *řádkovém* režimu запиšte sekvenci (: ) (W) (Q). Příkaz (W) v *řádkovém* režimu znamená *zapsat* do souboru, (Q) znamená *ukončit* editor.

### Editor vi v akci

Editor vi je možno používat jako jakýkoliv běžný editor. Ve *vkládacím* režimu, lze vkládat text nebo text pomocí kláves (←) a (Enter) mazat.

Kurzorem lze pohybovat pomocí kurzorových kláves.

Tento způsob ovládání může nicméně způsobit v určitých situacích problémy, neboť některé terminály používají speciální klávesové kódy. Tehdy se hodí *příkazový* režim. Přejděte z *vkádacího* do *příkazového* režimu stiskem klávesy (Esc). V *příkazovém* režimu lze pohybovat kurzorem pomocí kláves (H), (J), (K) a (L). Klávesy mají následující funkce:

- (H) přesunout kurzor o jeden znak doleva
- (J) přesunout kurzor o jeden řádek dolů
- (K) přesunout kurzor o jeden řádek nahoru
- (L) přesunout kurzor o jeden řádek doprava

Příkazy lze v *příkazovém* režimu různě modifikovat. Chcete-li příkaz vykonat několikrát, jednoduše vložte množství opakování před vlastní příkaz. Chcete-li například, aby se kurzor posunul o pět znaků doprava, stiskněte (5)(L).

### Další informace

Editor vi podporuje širokou paletu funkcí a příkazů. Umožňuje používat makra, uživatelem definované klávesové zkratky, pojmenované buffery a další užitečné vlastnosti. Podrobný popis přesahuje možnosti tohoto krátkého manuálu. SUSE LINUX obsahuje vim (vi improved), zdokonalenou verzi editoru vi. Podrobnější informace o tomto editoru najdete na mnoha místech:

- Program vimtutor je interaktivní průvodce editorem vim.
- Přímo v editoru vim můžete získat nápovědu k mnoha tématům pomocí příkazu :help.
- Kniha o editoru vim je dostupná online na adrese <http://www.truth.sk/vim/vimbook-OPL.pdf>.
- Webové stránky projektu vim na adrese <http://www.vim.org> obsahují novinky, odkazy na poštovní konference a další dokumentaci.
- Množství informačních zdrojů o editoru vim je dostupných na internetu: <http://www.selflinux.org/selflinux/html/vim.html>, <http://www.linuxgazette.com/node/view/9039>, [http://www.apmaths.uwo.ca/~xli/vim/vim\\_tutorial.html](http://www.apmaths.uwo.ca/~xli/vim/vim_tutorial.html). Na adrese <http://linux-universe.com/HOWTO/Vim-HOWTO/vim-tutorial.html> jsou odkazy na další průvodce.

## Důležité

### Licence editoru VIM

Editor VIM je šířen jako tzv. *charityware*, což znamená, že autoři za něj pro sebe nežádají žádné peníze, nicméně vyzývají uživatele k finanční podpoře dobročinného projektu na pomoc chudým dětem z Ugandy.

Více informací lze získat na webových stránkách <http://iccf-holland.org/index.html>, <http://www.vim.org/iccf/> a <http://www.iccf.nl/>.

## Důležité

## 10.2 Virtuální konzole

Linux je víceúlohový víceuživatelský systém. Tyto vlastnosti systému oceníte dokonce i na obyčejné uživatelské stanici. V textovém režimu je k dispozici šest virtuálních konzolí. Přepínání mezi nimi zajišťuje kombinace **(Alt)-(F1)** až **(Alt)-(F6)**. Sedmá konzole je rezervována pro X11. Počet konzolí je možné změnit v souboru `/etc/inittab`.

K přepnutí z X11 do konzole použijte kombinaci kláves **(Ctrl)-(Alt)-(F1)** až **(Ctrl)-(Alt)-(F6)**. Stisknutím kláves **(Alt)-(F7)** se vrátíte zpět do X11.

## 10.3 Mapování klávesnice

Standardizace mapování klávesnice si vynutila změny v následujících souborech:

```
/etc/inputrc
/usr/X11R6/lib/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/X11R6/lib/X11/app-defaults/XTerm
/usr/share/emacs/<VERSION>/site-lisp/term/*.el
```

Změny se týkají pouze aplikací, které používají `terminfo` nebo těch aplikací, jejichž konfigurační soubory se mění v systému přímo, jako je (`vi`, `less`, atd.). Ostatní aplikace ne od SUSE by měly být přizpůsobeny tomuto původnímu nastavení.

Pod systémem X může být ovládání pomocí (klávesových zkratk) zpřístupněno přes kombinaci kláves `(Ctrl)-(Shift)` (pravý). Podívejte se na příslušný příkaz v souboru `/usr/X11R6/lib/X11/Xmodmap`.

Další nastavení jsou možná přes "X rozšíření klávesnice" (XKB). Toto rozšíření používají také prostředí GNOME (`gswitchit`) a KDE (`kxkb`). Informace o XKB najdete v souboru `/etc/X11/xkb/README` a dalších dokumentech v adresáři.

Detailní informace o čínských, japonských a korejských (CJK) specifických klávesových zkratkách najdete na stránkách Mika Fabiana zde: <http://www.suse.de/~mfabian/suse-cjk/input.html>.

## 10.4 Lokální přizpůsobení I18N and L10N

SUSE LINUX je mezinárodní systém, který se dá velmi flexibilně přizpůsobit lokálním potřebám. Internacionální charakter (*I18N*) jinými slovy umožňuje specifický přístup k lokalizaci (*L10N*).

Lokální nastavení pro národní jazyky je zajištěno proměnnými `LC_` definovanými v souboru `/etc/sysconfig/language`. Nejde přitom pouze o určení jazyka pro komunikaci s jednotlivými aplikacemi a *prostředí programů v původním jazyce*, ale také o *zprávy systému, znakové sady, pořadí při abecedním třídění, formát časových údajů, desetinných čísel a peněžních částek*. Každou z těchto kategorií můžete definovat přímo její proměnnou, nebo nepřímo hlavní proměnnou v souboru `language` (podívejte se na manuálové stránky `man locale`).

1. `RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`, `RC_LC_NUMERIC`, `RC_LC_MONETARY`: Tyto proměnné se exportují do prostředí příkazového interpretu bez předpony `RC_` a určují jednotlivé z lokalizačních kategorií. Soubory, kterých se to týká najdete v seznamu níže. Nastavení proměnných zjistíte výpisem příkazu `locale`.
2. `RC_LC_ALL`: Pokud je nastavena tato proměnná, přepíše svou hodnotou výše uvedené proměnné.
3. `RC_LANG`: Pokud není nastavena žádná z výše uvedených proměnných, je výchozí hodnotou. Defaultně SUSE LINUX nastavuje pouze `RC_LANG`. Tato vlastnost pomáhá uživatelům zavést své vlastní hodnoty.



4. `ROOT_USES_LANG`: V případě nastavení na `no`, `root` pracuje `root` v prostředí standardu POSIX.

Ostatní proměnné můžete nastavit YaSTem v editoru souborů `sysconfig`. Hodnota těchto proměnných obsahuje kód jazyka, země, kódování a modifikátoru. Individuální komponenty jsou připojitelné pomocí speciálních znaků:

```
LANG=<language> [ [_<COUNTRY>] .<Encoding> [@<Modifier>] ]
```

## 10.4.1 Některé příklady

Nastavení jazyka a kódů země by mělo jít ruku v ruce. Jazyková nastavení jsou dle standardu ISO639 (<http://www.evertype.com/standards/iso639/iso639-en.html> a <http://www.loc.gov/standards/iso639-2/>). Kódy zemí naleznete v ISO3166, podívejte se na ([http://www.din.de/gremien/nas/nabd/iso3166ma/codlstpl/en\\_listpl.html](http://www.din.de/gremien/nas/nabd/iso3166ma/codlstpl/en_listpl.html)). Smysl má nastavit hodnoty, jejichž popis užití naleznete v `/usr/lib/locale`. Další soubory s popisy můžete vytvořit ze souborů v adresáři `/usr/share/i18n` použitím příkazu `localedef`. Soubor s popisem `cs_CZ.UTF-8` (pro naši krásnou zemi) vytvoříte takto:

```
localedef -i cs_CZ -f UTF-8 cs_CZ.UTF-8
```

**LANG=cs\_CZ.UTF-8** Toto je defaultní nastavení, když je v průběhu instalace vybrána čeština. Jestliže zvolíte jiný jazyk, bude tento jazyk také s kódovacím UTF-8.

**LANG=cs\_CZ.ISO-8859-2** Takto nastavíme proměnnou na češtinu, zemi na Českou republiku a znakovou sadu na ISO-8859-2. Řetězec definující znakovou sadu, kterou je v našem případě ISO-8859-2 pak bude načítán programy jako je Emacs.

SuSEconfig čte proměnnou ze souboru `/etc/sysconfig/language` a zapisuje nezbytné změny do `/etc/SuSEconfig/profile` a do `/etc/SuSEconfig/csh.cshrc`. Pak přečte `/etc/SuSEconfig/profile`, nebo data načte ze zdroje, kterým je `/etc/profile`. `/etc/SuSEconfig/csh.cshrc` hledá svůj zdroj v `/etc/csh.cshrc`. Toto uspořádání umožňuje široké spektrum nastavení i pro velký systém.

Uživatelé také mohou přepisovat původní hodnoty v systému editací `~/` `.bashrc` ve svém home adresáři. Jako příklad je možné uvést zobrazování programových hlášek v češtině `cs_CZ` do španělštiny, což znamená použít `LC_MESSAGES=es_ES`.

## 10.4.2 Nastavení jazykové podpory

Dle pravidel pro kategorii *Messages* pak systém zprávy ukládá v příslušném jazykovém adresáři (v našem případě *cs*) jako zálohu. Jestliže nastavíte *LANG* na *cs\_CZ* a soubor *zpráv* ukládáte do */usr/share/locale/en\_US/LC\_MESSAGES*, který neexistuje, systém jej bude dále ukládat do souboru */usr/share/locale/en/LC\_MESSAGES*.

Řetěz zálohových souborů můžete nadefinovat např. pro slovenštinu a češtinu, či pro galštinu, španělštinu a portugalštinu:

```
LANGUAGE="cs_SK:cs_CZ"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

Jestli toužíte po tradiční norštině *nynorsk* a *bokmål* namísto a s dodatečnou zálohou pro *no*), proveďte úpravu proměnné do tohoto tvaru:

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

or

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

Poznamenejme nakonec, že v norštině je odlišný i parametr *LC\_TIME*.

### Vyskytující se problémy

Pro správnou práci s desetinnými čísly v češtině nestačí pouze nastavit proměnnou *LANG* na *cs*. Aby např. knihovna *glibc* našla správnou hodnotu v souboru */usr/share/locale/en\_US/LC\_NUMERIC*, je třeba nastavit přímo proměnnou *LC\_NUMERIC* na hodnotu *cs\_CZ*.

### Další informace

- *The GNU C Library Reference Manual*, Kapitola *Locales and Internationalization*, kterou najdeme v *glibc-info*.
- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, na <http://www.cl.cam.ac.uk/~mgk25/unicode.html>.
- *Unicode-Howto*, autor Bruno Haible je v souboru `file:/usr/share/doc/howto/en/html/Unicode-HOWTO.html`.



# Systém X Window

Systém X Window (X11 nebo X server) se stal prakticky standardem grafického uživatelského rozhraní v unixových systémech. Je to síťový systém, který umožňuje, aby se programy spuštěné na jednom počítači zobrazovaly na jiném počítači připojeném jakoukoli sítíovou technologií, ať už v LAN nebo Internetu.

V této kapitole se pojednává o optimalizaci prostředí vašeho systému X Window, základech práce s fonty v systému SUSE LINUX a o konfiguraci OpenGL a 3D. Konfigurace myši a klávesnice pomocí modulů YaST je popsána v *Uživatelské příručce*.

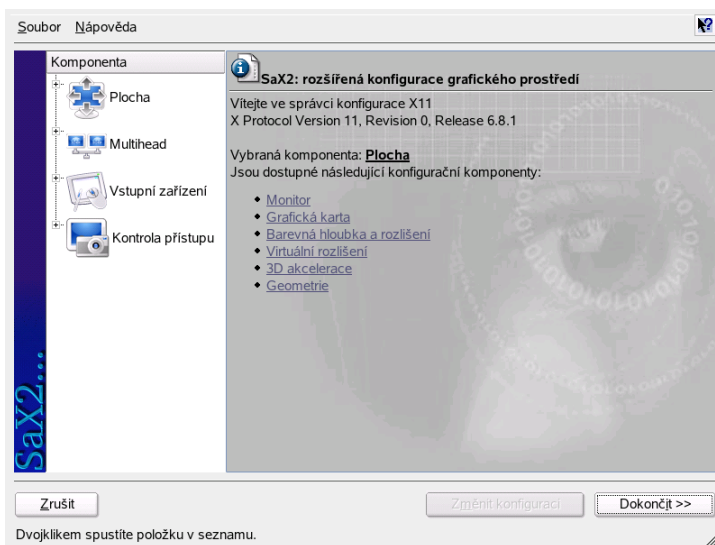
11.1	Nastavení X11 pomocí SaX2 . . . . .	214
11.2	Optimalizace systému X Window . . . . .	223
11.3	Instalace a konfigurace fontů . . . . .	228
11.4	Konfigurace OpenGL – 3D . . . . .	234

## 11.1 Nastavení X11 pomocí SaX2

X server se stará o komunikaci mezi hardwarem a softwarem. Pracovní prostředí (KDE nebo GNOME) a mnoho správců oken používá X server pro interakci s uživatelem.

Grafické prostředí se nastavuje během instalace. Chcete-li později změnit nastavení, spusťte SaX2. Aktuální nastavení je uloženo a můžete se k němu kdykoliv vrátit. Při konfiguraci se použijí jako výchozí aktuální hodnoty, které můžete změnit: rozlišení obrazovky, barevná hloubka, obnovovací frekvence a výrobce a typ monitoru, pokud byl rozpoznán.

Pokud jste nainstalovali novou grafickou kartu, objeví se malý dialog s dotazem na aktivaci podpory 3D. Kliknutím na 'Změnit' se spustí ve zvláštním okně SaX2, konfigurační nástroj pro vstupní a zobrazovací zařízení. Toto okno je zobrazeno na obrázku 11.1 na této straně.



*Obrázek 11.1: Hlavní okno SaX2*

Vlevo jsou čtyři hlavní položky: 'Plocha', 'Multihead', 'Vstupní zařízení' a 'Kontrola přístupu'. V sekci 'Plocha' nastavíte grafickou kartu, monitor, rozlišení ob-

razovky, barevnou hloubku a velikost a umístění obrazu. Klávesnici, myš, dotykovou obrazovku a grafický tablet lze nastavit v sekci 'Vstupní zařízení'. V sekci 'Multihead' lze nastavit více obrazovek (viz 11.1.7 na straně 219). V sekci 'Kontrola přístupu' je možné nastavit ovládání kurzoru pomocí numerické klávesnice.

Vyberte váš monitor a grafickou kartu. Obvykle systém monitor i grafickou kartu automaticky rozpozná. Pokud se tak stalo, nemusíte zde nic dalšího nastavovat. Pokud systém váš monitor nerozpoznal, vyberte váš typ monitoru ze seznamu v dalším dialogu, nebo zadejte technické parametry uvedené v manuálu, který jste dostali s monitorem. Alternativně můžete použít některý z připravených režimů VESA.

V hlavním okně klikněte na 'Dokončit' a vyzkoušejte nové nastavení. Tím se zajistí konfigurace vhodná pro vaše zařízení. Pokud nemáte stabilní obraz, ihned ukončete test stisknutím klávesy (Esc) a snižte obnovovací frekvenci nebo rozlišení a barevnou hloubku. Nehledě k tomu, zda jste vaše nové nastavení testovali, se toto nové nastavení projeví až po restartu X serveru.

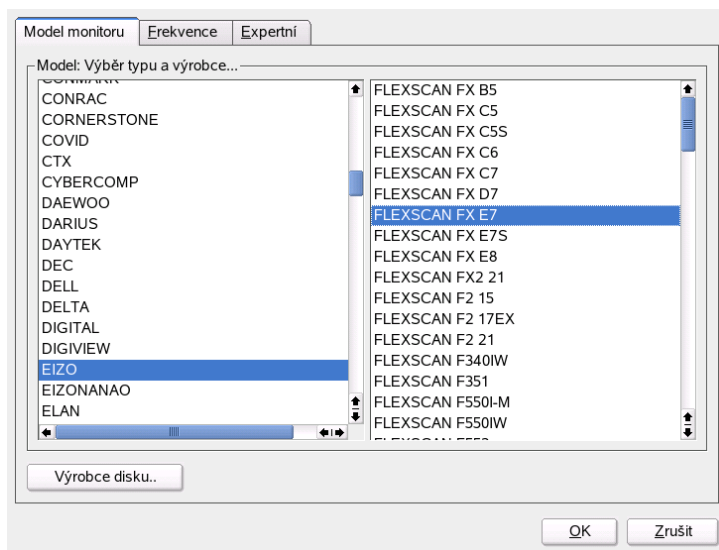
### 11.1.1 Plocha

Výběrem 'Změnit konfiguraci' → 'Vlastnosti' se zobrazí okno se záložkami 'Model monitoru', 'Frekvence' a 'Expertní'.

**'Model monitoru'** V levé části okna vyberte výrobce, v pravé části model. Pokud máte disketu s linuxovými ovladači pro váš monitor, nainstalujte je kliknutím na 'Disk s ovladači'.

**'Frekvence'** Zadejte horizontální a vertikální frekvenci vašeho monitoru. Vertikální frekvence je pouze jiné označení pro obnovovací frekvenci obrazovky. Obvykle jsou vhodná rozmezí nastavena automaticky podle typu monitoru a není třeba nic měnit.

**'Expertní'** Zde můžete změnit některá nastavení obrazovky. V horní nabídce zvolte, kterou metodu chcete použít pro výpočet rozlišení obrazovky a geometrie obrazu. Nastavení měňte pouze pokud nemáte stabilní obraz. Navíc zde můžete zapnout úsporný režim DPMS.



*Obrázek 11.2: Výběr monitoru*

## Varování

### Konfigurace frekvencí monitoru

Ačkoliv většinou mají monitory bezpečnostní pojistku, měli byste být při ručním zadávání frekvencí velice opatrní. Zadáním nevhodných hodnot můžete poškodit váš monitor. Pokud si nejste jisti, nahlédněte do manuálu k monitoru.

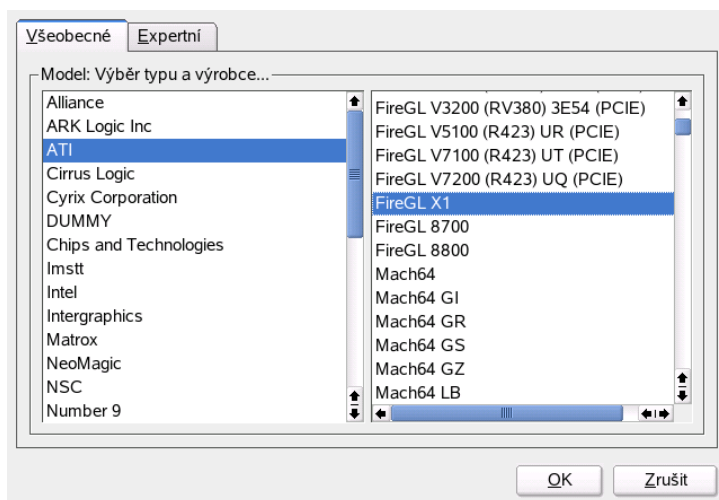
**Varování**

### 11.1.2 Grafická karta

Dialog grafické karty má dvě záložky: 'Všeobecné' a 'Expertní'. V záložce 'Všeobecné' na levé straně vyberte výrobce vaší karty a na pravé straně model.

V záložce 'Expertní' najdete rozšířené možnosti konfigurace. Na pravé straně můžete otočit obraz (užitečné u některých TFT obrazovek). Záznamy ID obrazovky jsou užitečné tehdy, pokud používáte více obrazovek. Obvykle zde není třeba nic





Obrázek 11.3: Výběr grafické karty

měnit. Pokud přesto změníte některé hodnoty, měli byste přesně vědět, co děláte. Více informací najdete v manuálu vaší grafické karty.

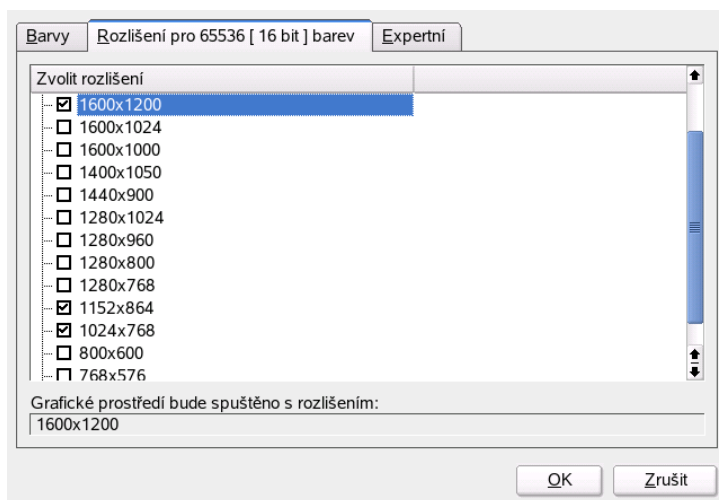
### 11.1.3 Barevná hloubka a rozlišení

V této sekci najdete tři karty: 'Barvy', 'Rozlišení', a 'Expertní'.

**'Barvy'** V závislosti na vašem vybavení zvolte barevnou hloubku. Možnosti jsou 16, 256, 32768, 65536, nebo 16.7 milionů barev (4, 8, 15, 16 nebo 24 bitů). Pro přiměřeně kvalitní zobrazení zvolte nejméně 256 barev.

**'Rozlišení'** Při rozpoznávání hardwaru je nastavena taková kombinace rozlišení a barevné hloubky, kterou dokáže váš monitor zobrazit. Díky tomu hrozí pouze malé nebezpečí, že SUSE LINUX poškodí váš hardware. Pokud ale měníte toto nastavení ručně, pak byste si měli přečíst dokumentaci k hardwaru.

**'Expertní'** Kromě rozlišení nabízených v předchozím dialogu si zde můžete přidat vlastní rozlišení, která budou následně zahrnuta do výběrové tabulky.



*Obrázek 11.4: Konfigurace rozlišení*

### 11.1.4 Virtuální rozlišení

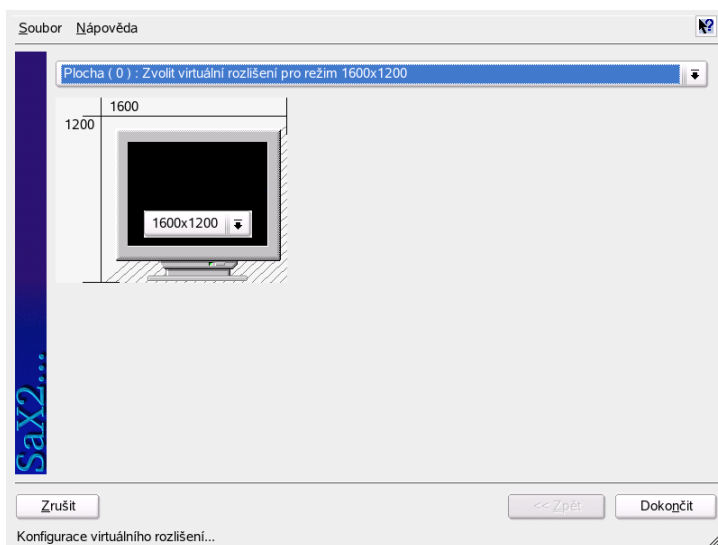
Každá pracovní plocha má rozlišení, které se vykresluje na celou plochu monitoru. Navíc máte možnost nastavit si pracovní plochu větší, než je viditelná plocha obrazovky. Pokud posunete ukazatel myši za okraj pracovní plochy, zobrazí se skrytá (virtuální) část plochy. Můžete si tedy zvětšit svou pracovní plochu.

Virtuální rozlišení můžete nastavit dvěma způsoby. Tažením myši: posuňte ukazatel myši nad obrázek monitoru tak, aby se změnil v křížek. Držte stisknuté levé tlačítko myši a posuňte kurzor tak, abyste zvětšili šrafovanou plochu na požadovanou velikost. Tato metoda je vhodná, pokud si nejste zcela jisti, jak velkou pracovní plochu chcete používat.

Výběrem z nabídky zobrazující aktuální virtuální rozlišení můžete také zvolit požadované virtuální rozlišení.

### 11.1.5 3D Akcelerate

Volitelně zde můžete zapnout 3D akceleraci vaší grafické karty.



Obrázek 11.5: Konfigurace virtuálního rozlišení

### 11.1.6 Geometrie

V těchto dvou záložkách můžete přesně nastavit velikost a pozici obrazu, viz obrázek 11.6 na následující straně. Jestliže máte nastaveno více obrazovek, můžete další nastavit přechodem na další obrazovku tlačítkem 'Následující obrazovka'. Nakonec stiskněte 'Uložit' a vaše nastavení se uloží.

### 11.1.7 Multihead

Jestliže jste nainstalovali více než jednu grafickou kartu, nebo vaše karta podporuje výstup na více obrazovek, můžete si zde nastavit připojení více monitorů. Dvě zapojené obrazovky se obvykle označují jako *dualhead*. Více obrazovek pak jako *multihead*. SaX2 sám najde více připojených grafických karet a připraví pro ně vhodnou konfiguraci. Doladit tuto konfiguraci můžete v nabídkách 'Režim s více monitory' a 'Rozložení obrazovky'. Na výběr máte tři různé režimy: 'Tradiční multihead' (výchozí), 'Klonovaný multihead', a 'Xinerama'.



*Obrázek 11.6: Úprava geometrie obrazu*

**Tradiční multihead** Každý monitor se chová jako nezávislá jednotka. Myši přejíždíte z obrazovky na obrazovku.

**Klonovaný multihead** V tomto režimu všechny monitory zobrazují stejný obraz. Kurzor myši je viditelný pouze na hlavní obrazovce.

**Xinerama** Veškeré obrazovky dohromady vytvářejí jednu velkou plochu. Okna programů lze rozmístit na všechny obrazovky nebo změnit velikost, aby se zobrazily na více monitorech.

Rozložení jednotlivých obrazovek v prostředí multihead lze měnit myší v dialogu 'Rozložení obrazovek', posunováním po mřížce. Standardně jsou monitory vyrovnány vedle sebe v pořadí, v jakém byly konfigurovány jednotlivé grafické karty, v řadě zleva doprava. Po dokončení nastavení otestujte tlačítkem 'Test'.

Linux v současnosti nepodporuje 3D zobrazení v prostředí Xinerama multihead. Pokud zvolíte režim Xinerama, SaX2 vypne podporu 3D.

### 11.1.8 Vstupní zařízení

**Myš** Pokud systém vaši myš nenalezl, vyberte model ručně. Pro zjištění přesného typu nahlédněte do dokumentace k výrobku. Stačí zvolit model ze seznamu podporovaných myší a stisknout na numerické klávesnici (5).

**Klávesnice** V horní části dialogu nastavte typ klávesnice. Poté nastavte, jakou chcete používat klávesovou mapu (v každé zemi jsou určitá tlačítka rozmístěna na různých klávesách). Vaše nastavení můžete ověřit v testovacím políčku.

Pro aktivaci a uložení vašich změn klikněte na 'Dokončit'.

**Dotyková obrazovka** V současné době podporuje X.org pouze dotykové obrazovky společností Microtouch a Elo TouchSystems. SaX2 bohužel nemůže automaticky rozpoznat dotykový panel. Poznává monitor, ne dotykový panel. Dotykový panel je považován za vstupní zařízení.

Při konfiguraci postupujte takto: spusťte SaX2 a zvolte 'Vstupní zařízení' → 'Dotyková obrazovka'. Klikněte na 'Přidat novou dotykovou obrazovku' a vyberte model. Konfiguraci uložíte kliknutím na 'Dokončit'. Konfiguraci není třeba testovat.

Dotykové obrazovky obvykle nabízí spoustu možností pro konfiguraci a obvykle je potřeba je nejdříve zkalibrovat. V Linuxu bohužel pro tento účel neexistuje obecný nástroj. Při instalaci se však nastaví vhodné standardní hodnoty, které by měly být dostačující. Normálně není potřeba další konfigurace.

**Tablet** X.org momentálně podporuje pouze několik grafických tabletů. Pomocí SaX2 můžete nastavit tablety připojené přes USB nebo sériový port. Z hlediska konfigurace se jedná pouze o další vstupní zařízení, jako je myš.

Spusťte SaX2 a vyberte 'Vstupní zařízení' → 'Tablet'. Klikněte na 'Přidat' a z následujícího dialogu vyberte výrobce vašeho zařízení. Pokud máte připojené pero nebo gumu, zaškrtněte na pravé straně odpovídající políčko. Jestliže je tablet připojen přes sériový port, ověřte jeho hodnotu. `/dev/ttyS0` odpovídá prvnímu sériovému portu. `/dev/ttyS1` odpovídá druhému sériovému portu. Další porty používají obdobný zápis. Konfiguraci uložíte kliknutím na 'Dokončit'.

## 11.1.9 AccessX

Pokud nemáte k vašemu počítači připojenou myš, spusťte SaX2 a aktivujte AccessX. Tak budete moci řídit ukazatel myši pomocí numerické klávesnice. Popis funkcí kláves naleznete v tabulce 11.1 na následující straně. Posuvníkem můžete nastavit rychlost pohybu ukazatele při stisku klávesy.

**Tabulka 11.1:** AccessX ovládání myši pomocí numerické klávesnice

Klávesa	Popis
①	Aktivuje levé tlačítko myši.
②	Aktivuje prostřední tlačítko myši.
③	Aktivuje pravé tlačítko myši.
⑤	Klikne zvoleným (viz výše) tlačítkem. Jestliže není vybrané žádné tlačítko, klikne levým. Volba tlačítka je po kliknutí nastavena na výchozí.
④	Chová se jako ⑤, ale provede dvojklik.
⑩	Chová se jako ⑤, ale drží tlačítko stisknuté.
Del	Pustí dříve stisknuté (pomocí ⑩) tlačítko myši.
⑦	Pohyb kurzoru nahoru doleva.
⑧	Posunuje kurzor nahoru.
⑨	Pohyb nahoru doprava.
④	Posun doleva.
⑥	Pohyb doprava.
①	Pohyb kurzoru dolů doleva.
②	Posun dolů.
③	Posun kurzoru dolů doprava.

### 11.1.10 Joystick

V nástroji YaST, vyberte položku 'Hardware' a klikněte na ikonu 'Joystick'. V dialogu, který se otevře, vyberte výrobce a model vašeho joysticku. Tlačítkem 'Test' ověřte správnou funkci joysticku. Testovací dialog obsahuje tři grafy pro osy pohybu a čtyři značky pro čtyři standardní tlačítka joysticku. Při pohybu joystickem nebo stisku jeho tlačítek byste měli v dialogu zaznamenat odpovídající reakci. Protože jsou joysticky obvykle připojeny ke zvukové kartě, můžete k tomuto modulu přistupovat i z nastavení zvukové karty.

## 11.2 Optimalizace systému X Window

X.Org je open source implementace X Window systému. Je vyvíjena "X.Org Foundation", která je také odpovědná za vývoj nových technologií a standardů X Window systému.

Abyste maximálně využili možností svého hardwaru (myš, grafická karta, monitor, klávesnice), můžete nastavení ručně optimalizovat. Podrobnější informace o nastavení X Window systému najdete v souborech v adresáři `/usr/share/doc/packages/Xorg` a manuálových stránkách, ke kterým můžete přistupovat například příkazem `manxorg.conf`.

Program SaX2 umožňuje i náročné zásahy do konfigurace X Window, nicméně abyste naplno využili schopnosti vašeho hardwaru jako jsou myš, grafická karta, monitor nebo klávesnice, může být nutná ruční editace konfiguračního souboru. Některé aspekty tohoto procesu budou vysvětleny v následujícím textu. Podrobnější informace o konfiguraci systému X Window získáte v manuálových stránkách - viz příkaz `manxorg.conf`, k užítku vám mohou být i soubory v adresáři `/usr/share/doc/packages/xf86`.

### Varování

Při konfiguraci systému X Window buďte opatrní. Nikdy X Window nespouštějte před dokončením jeho řádné konfigurace, protože chybná konfigurace může způsobit neodstranitelné škody na vašem hardwaru (to se vztahuje zejména na monitory s pevnou frekvencí, které se však dnes už téměř nepoužívají). Autoři této knihy a společnost SUSE LINUX AG není za takovéto škody odpovědná. Následující informace byly pečlivě ověřovány, to ovšem nezaručuje, že všechny zde popsane postupy jsou správné a nemohou poškodit váš hardware.

### Varování

V následujících odstavcích je popsána struktura konfiguračního souboru `/etc/X11/xorg.conf`. Tento soubor je členěn na sekce uvedené klíčovým slovem `Section <designation>` a ukončené klíčovým slovem `EndSection`. Níže naleznete stručný přehled nejdůležitějších sekcí.

Ve výchozím nastavení vytváří programy SaX2 a `xf86config` konfigurační soubor `xorg.conf` v adresáři `/etc/X11`. To je hlavní konfigurační soubor systému X Window. Zde se nachází veškerá nastavení vaší grafické karty, myši a monitoru.

Každá sekce souboru `xorg.conf` popisuje určitou část konfigurace a má následující podobu:

```
Section název
    položka 1
    položka 2
    položka n
EndSection
```

Rozlišovány jsou následující typy sekcí:

*Tabulka 11.2: Sekce `/etc/X11/xorg.conf`*

Typ sekce	Popis
Files	Tato sekce obsahuje cesty použité pro vyhledávání fontů a tabulku RGB.
ServerFlags	Zde se zadávají obecné volby pro X server.
InputDevice	Zde se konfigurují vstupní zařízení jako klávesnice a speciální zařízení (touchpady, joysticky atd.). Důležitými položkami jsou: <code>Driver</code> a volby určující položky <code>Protocol</code> a <code>Device</code> .
Monitor	Popisuje použitý monitor: jméno, na které později odkazuje definice <code>Screen</code> , dále šířka pásma a obě mezní synchronizační frekvence ( <code>HorizSync</code> a <code>VertRefresh</code> ). Frekvence se zadávají v MHz, kHz, nebo Hz. Server obvykle odmítne jakékoli zobrazovací parametry, které neodpovídají specifikaci monitoru. Cílem je zabránit náhodnému nastavení monitoru na příliš vysokou řádkovou nebo snímkovou frekvenci.
Modes	Zde jsou uloženy zobrazovací parametry pro různá rozlišení obrazovky. Jejich hodnoty jsou obvykle vypočteny programem <code>SaX2</code> na základě údajů zadaných uživatelem a obvykle je není třeba měnit. Ruční zásah může být nutný např. při použití monitoru s pevnými frekvencemi. Podrobnější popis jednotlivých parametrů by byl nad rámec této knihy, ale najdete ho např. v dokumentu <code>HOWTO /usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz</code> .



Device	Zde je definována konkrétní grafická karta v systému, na kterou je prostřednictvím jejího názvu odkazováno v jiných sekcích konfiguračního souboru.
Screen	Zde je definován vztah mezi sekcemi Monitor a Device, jimiž je tvořena nezbytná konfigurace systému XFree. V podsekcích Display je určena barevná hloubka a škála rozlišení obrazovky použitelná pro danou hloubku.
ServerLayout	V této sekci je definována použitá kombinace vstupních zařízení ze sekce InputDevice a zobrazovacích zařízení (sekce Screen), ať už je v systému jedna grafická karta nebo se jedná o režim multihead (více karet provozovaných zároveň).

O sekcích Monitor, Device, a Screen se podrobněji dočtete dále. Informace o ostatních sekcích naleznete například v manuálových stránkách XFree86 a `xorg.conf`.

Konfigurační soubor `xorg.conf` může obsahovat více různých sekcí Monitor a Device. V souboru může existovat i více sekcí typu Screen. V sekci ServerLayout, která po nich následuje, je pak určeno, které sekce budou skutečně použity.

## 11.2.1 Sekce Screen

Nyní se pozastavíme u sekce Screen, která je styčným místem sekce Monitor a sekce Device a určuje, jaké kombinace barevné hloubky a rozlišení obrazovky budou použity. Příklad sekce Screen:

```
Section "Screen"
    DefaultDepth 16
    SubSection "Display"
        Depth 16
        Modes "1152x864" "1024x768" "800x600"
        Virtual 1152x864
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
```

```

        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth      8
        Modes       "1280x1024"
    EndSubSection
    Device         "Device[0]"
    Identifier     "Screen[0]"
    Monitor        "Monitor[0]"
EndSection

```

V řádce `Identifier` (zde `Screen[0]`) je dán jednoznačný název této sekce, na nějž je odkazováno v následující sekci `ServerLayout`. Řádky `Device` a `Monitor` určují kombinaci grafické karty a monitoru, pro které je tato sekce `Screen` platná a ve skutečnosti jsou to jen odkazy na odpovídající sekce `Device` a `Monitor` konfiguračního souboru. Těm se budeme více věnovat později.

Řádkou `DefaultDepth` nastavíte barevnou hloubku, se kterou se spustí X server, pokud nebude explicitně stanoveno jinak. Každé barevné hloubce odpovídá jedna podsekce `Display`. Na řádce `Depth` je této podsekci přiřazena konkrétní barevná hloubka, jejíž hodnoty mohou být 8, 15, 16, 24 a 32. Všechny moduly X serveru však nepodporují všechny hodnoty. Pro některé grafické karty znamenají hodnoty 24 a 32 totéž, zatímco u jiných udává hodnota 24 tzv. packed-pixel 24 bpp mód a 32 tzv. padded-pixel 32 bpp mód.

Nastavené barevné hloubce odpovídá seznam rozlišení obrazovky v sekci `Modes`. Tento seznam je zpracováván zleva doprava X serverem, který přiřadí danému rozlišení příslušný řádek `Modeline` se zobrazovacími parametry. Jejich hodnoty jsou závislé na schopnostech grafické karty a monitoru. Výsledný řádek je tedy předurčen obsahem sekce `Monitor`.

První nalezené platné rozlišení je tzv. `Default mode` a X server se s ním pustí. Během jeho provozu se pak dá kombinací kláves `(Ctrl) + (Alt) + (+)` (na numerické klávesnici) přepínat mezi hodnotami v seznamu směrem doprava, zatímco kombinací kláves `(Ctrl) + (Alt) + □` procházíme seznam směrem vlevo. Tím se dá měnit rozlišení obrazovky i za běhu X serveru.

Poslední řádka podsekce `Display` s označením `Depth 16` udává barevnou hloubku a přímo ovlivňuje maximální velikost virtuální obrazovky. Ta je dále závislá na velikosti videopaměti, nikoli na maximálním rozlišení monitoru. Moderní grafické karty mají jsou osazeny pamětí o dostatečné velikosti, lze tedy používat velké virtuální obrazovky. Pokud má grafická karta videopaměť např. o 16 MB, lze při barevné hloubce 32 bitů vytvořit virtuální obrazovku o velikosti až 2048x248 bodů. Zejména u moderních akcelerovaných karet však není doporučeno použít veškerou dostupnou paměť na virtuální obrazovku, neboť jejich

paměť slouží také jako vyrovnávací paměť pro uložení fontů a grafických objektů.

## 11.2.2 Sekce Device

Tato sekce popisuje konkrétní grafickou kartu. Soubor `xorg.conf` může obsahovat více těchto sekcí, které jsou odlišeny hodnotou řádku `Identifier`. Máte-li více grafických karet, sekce jsou očíslovány tak, že první karta bude `Device[0]`, druhá karta `Device[1]` atd. Následující výpis je příklad konfigurace sekce `Device` u počítače s jednou kartou Matrox Millennium PCI:

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"
    Driver         "mga"
    Identifier     "Device[0]"
    VendorName     "Matrox"
    Option         "sw_cursor"
EndSection
```

Při konfiguraci pomocí `SaX2` bude vaše sekce `Device` vypadat podobně. Položky `Driver` a se liší podle hardwaru ve vašem počítači a `BusID` jsou zjištěny programem `SaX2` automaticky. Hodnota `BusID` představuje pozici na sběrnici `PCI` nebo `AGP`, ve které je instalována grafická karta. Odpovídá hodnotě zjištěné příkazem `lspci` (nenechte se nicméně zmást tím, že `X` server zde používá dekadické hodnoty a program `lspci` hodnoty hexadecimální).

V sekci `Driver` přiřadíte grafické kartě ovladač. Máte-li např. kartu Matrox Millennium, nazývá se modul ovladače `mga`. `X` server pak hledá daný modul v podadresáři s ovladači uvedeném v položce `ModulePath` v sekci `Files`. Ve výchozím stavu po instalaci to je adresář `/usr/X11R6/lib/modules/drivers`. Pokud ke jméně modulu přidáte `_drv.o`, získáte jméno souboru s ovladačem, v případě modulu `mga` bude tedy zaveden soubor `mga_drv.o`.

Chování `X` server nebo ovladačů lze ovlivnit dalšími volbami. Příkladem je například volba `sw_cursor` ze sekce `Device`, která zakáže hardwarový kurzor myši a simuluje ho hardwarově. Různé ovladače mohou mít implementovány různé volby. Popis voleb dostupných u konkrétního ovladače najdete v adresáři `/usr/X11R6/lib/X11/doc` (máte-li nainstalován balík `XFree-doc`. Popis obecně platných voleb obsahují také manuálové stránky (`manxorg.conf` a `manXFree86`).

### 11.2.3 Sekce Monitor a Modes

Podobně jako každá sekce `Device` popisuje jednu grafickou kartu, popisují sekce `Monitor` a `Modes` jeden monitor. Konfigurační soubor může obsahovat libovolné množství těchto sekcí (lišících se minimálně v jejich symbolických jménech). V sekci `SystemLayout` je pak určeno, která ze sekcí `Monitor` je platná.

Nastavení monitoru by měli provádět pouze zkušení uživatelé. Nejdůležitějšími položkami sekcí `Monitor` jsou horizontální a vertikální frekvence monitoru pro dané rozlišení.

#### Varování

Pokud nerozumíte principům spolupráce monitoru a grafické karty, hodnoty frekvencí neměňte, neboť to zejména u starších monitorů může vést až k jejich zničení.

#### Varování

Pokud si troufáte ručně měnit navrženou konfiguraci monitoru, měli byste věnovat pozornost dokumentaci `/usr/X11/lib/X11/doc`. Velký význam má zejména část popisující režimy monitoru, manipulaci s horizontální a vertikální frekvencí a funkci grafických komponent systému.

V dnešní době se s ručním nastavením frekvencí monitoru prakticky nesetkáte. Při použití moderního monitoru schopného přizpůsobit obraz libovolné frekvenci generované grafickou kartou v určitém rozsahu (dnes v tomto režimu pracuje naprostá většina monitorů), dokáže X server zpravidla zjistit rozsah frekvencí a optimální rozlišení pomocí DDC přímo od monitoru. Této možnosti využívá i konfigurační program `SaX2`. Pokud se to nepodaří, může být využito i X serverem nabízené módy `VESA`, jenž fungují prakticky pro jakékoli kombinace monitorů a grafických karet.

## 11.3 Instalace a konfigurace fontů

V systému `SUSE LINUX` je instalace dalších fontů velmi jednoduchá. Stačí když fonty přepokopírujete do určité adresářové struktury `X11`, (viz odstavec 11.3.1 na straně 232), tak aby je mohl používat nový systém pro zobrazování fontů - `xft`. Instalační adresář s fonty by tedy měl být podadresářem adresářů, jenž jsou uvedeny v `/etc/fonts/fonts.conf` (viz odstavec 11.3.1 na následující straně).

Fonty můžete (jako uživatel `root`) překopírovat ručně do adresáře jako je např. `/usr/X11R6/lib/X11/fonts/truetype`. Instalaci fontů lze provést také pomocí Ovládacího centra KDE - položka Vzhled a motivy->Písmo.

Místo kopírování fontů můžete vytvořit také symbolické odkazy na fonty, které jsou uloženy na připojeném diskovém oddílu se systémem Windows. Pak stačí spustit příkaz `SuSEconfig--module fonts`.

Příkaz `SuSEconfig--module fonts` spustí skript `/usr/sbin/fonts-config`, který zajistí instalaci fontů. Pokud vás zajímá, co přesně tento skript dělá, podívejte se do jeho manuálové stránky např. příkazem `(man fonts-config)`.

Ať už se jedná o písma bitmapová, TrueType, OpenType nebo Type1 (Postskriptová), tento postup je stejný. Fonty všech těchto typů mohou být umístěny v jednom adresáři. Jedinou výjimkou jsou tzv. CID-keyed fonty (tyto fonty umožňují kombinovat znaky různých kódování a používají se pro japonštinu, čínštinu a podobné jazyky). U těchto písem se instalační postup poněkud liší, viz odstavce 11.3.1 na straně 233.

### 11.3.1 Systémy písem

XFree používá dva naprosto rozdílné systémy písem: původní *X11 Core-Font systém* a nově navržený *Xft/fontconfig*. V následující části si stručně popíšeme jejich charakteristiku.

#### Xft

Při vývoji Xft byl od počátku kladen důraz na podporu škálovatelných písem včetně jejich vyhlazování. Na rozdíl od X11 Core písem nejsou písma spravována X serverem, ale jednotlivými aplikacemi. Jednotlivé programy získaly přímý přístup ke konfiguračním souborům písem a tím i kontrolu na interpretaci jednotlivých znaků. Zároveň je díky tomu zaručeno, že tisk z těchto programů bude vypadat přesně tak, jak vidíte na obrazovce.

V systému SUSE LINUX obě velká grafická prostředí KDE a GNOME, program Mozilla i řada dalších aplikací již standardně Xft používá a tento systém je dnes používán více než tradiční X11-Core.

Systém Xft používá při vyhledávání písem a jejich interpretaci knihovnu `fontconfig`. Její chování lze ovlivnit globálním konfiguračním souborem `/etc/fonts/fonts.conf` a uživatelskými konfiguračními soubory `~/.fonts.conf`. Každý konfigurační soubor musí začínat touto hlavičkou:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

a končit patičkou

```
</fontconfig>
```

Každý adresář s fonty je v konfiguračním souboru definován na samostatném řádku následujícím způsobem:

```
<dir>/usr/local/share/fonts/</dir>
```

Není však nutné přidávat do souboru nový záznam pro každý adresář. Jako výchozí uživatelský adresář s písmy je v `/etc/fonts/fonts.conf` nastaven adresář `~/ .fonts`. Chcete-li si tedy nainstalovat další písma, nakopírujte je do `~/ .fonts` ve svém domovském adresáři.

Můžete zde také definovat pravidla určující vzhled písem. Takto například vypnete vyhlazování pro všechna písma:

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

Vyhlazování pro konkrétní písma pak povolíte např. takto:

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

Ve svém výchozím nastavení používá většina aplikací písma `sans-serif` (nebo jejich ekvivalent `sans`), `serif`, nebo `monospace`. Nejde o skutečné fonty, ale o aliasy, které podle jazykového nastavení teprve ukazují na konkrétní písma.

Uživatel si může vytvořit vlastní soubor `~/ .fonts.conf` a nasměrovat zde tyto aliasy na svá oblíbená písma:

```

<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
</alias>
<alias>
  <family>monospace</family>
  <prefer>
    <family>FreeMono</family>
  </prefer>
</alias>

```

Protože systém aliasů používají téměř všechny aplikace, ovlivní tyto změny celý systém. Máte tak možnost centrálně nastavit používání vašich oblíbených písem a nemusíte měnit konfiguraci v každé aplikaci zvlášť.

Příkazem `fc-list` získáte seznam nainstalovaných písem. Pokud vás zajímá pouze určitý typ písem, např. škálovatelný (`:outline=true`) s hebrejskými znaky (`:lang=he`), obsahující ve jméně slovo (`family`) a chcete znát jeho styl (`style`), řez (`weight`) a název souboru, v němž se písmo nachází, zadejte příkaz:

```
fc-list ":lang=he:outline=true" family style weight file
```

Výstup tohoto příkazu může vypadat např. takto:

```

/usr/X11R6/lib/X11/fonts/truetype/FreeSansBold.ttf: FreeSans:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBoldOblique.ttf: FreeMono:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSerif.ttf: FreeSerif:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBoldItalic.ttf: FreeSerif:style=BoldItalic:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansOblique.ttf: FreeSans:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifItalic.ttf: FreeSerif:style=Italic:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoOblique.ttf: FreeMono:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMono.ttf: FreeMono:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSans.ttf: FreeSans:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBold.ttf: FreeSerif:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBoldOblique.ttf: FreeSans:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBold.ttf: FreeMono:style=Bold:weight=200

```

Důležité parametry příkazu `fc-list` jsou:

*Tabulka 11.3: Vybrané parametry příkazu `fc-list`*

Parametr	Popis a možné hodnoty
<code>family</code>	Název rodiny písma, např., <code>FreeSans</code> .
<code>foundry</code>	Výrobce písma, např., <code>urw</code> .
<code>style</code>	Styl písma, např. <code>Medium</code> , <code>Regular</code> , <code>Bold</code> , <code>Italic</code> , <code>Heavy</code> .
<code>lang</code>	Jazyky, které písmo podporuje, např. <code>cs</code> pro češtinu, <code>ja</code> pro japonštinu, <code>zh-TW</code> pro tradiční čínštinu, <code>zh-CN</code> pro zjednodušenou čínštinu atd.
<code>weight</code>	Tloušťka písma, např., 80 pro normální, 200 pro tučné.
<code>slant</code>	Šikmost 0 pro normální písmo, 100 pro kurzívu.
<code>file</code>	Název souboru s písmem.
<code>outline</code>	<code>true</code> pokud se jedná o obrysová písma, <code>false</code> pro ostatní.
<code>scalable</code>	<code>true</code> pokud se jedná o škálovatelná písma, <code>false</code> pro ostatní.
<code>bitmap</code>	<code>true</code> u bitmapových písem, <code>false</code> u ostatních.
<code>pixelsize</code>	Velikost písma v pixelech. Má význam pouze u bitmapových písem.

## Systém písem X11 Core

Systém X11 Core byl navržen v~roce 1987 pro zpracování monochromatických bitmapových písem v X11R1. Dnes podporuje kromě bitmapových písem i škálovatelná písma jako jsou fonty Type1, TrueType, OpenType a písma typu CID-keyed. Již velmi dlouho jsou podporována také unicodová písma. Zdaleka však nenabízí takové možnosti jako Xft/fontconfig.

Například u škálovatelných písem není implementována podpora antialiasingu. Zpracování fontů se znaky v mnoha jazycích může trvat déle. Také použití Unicodových písem vede ke zpomalení a vyžaduje více paměti.

Systém písem X11 Core zdědil několik slabín. Je zastaralý a nedá se rozumným způsobem rozšiřovat. Z důvodu zpětné kompatibility je stále zachováván při životě, nicméně je vhodné ho nahradit moderním systémem Xft/fontconfig, pokud je to možné.

X server dokáže zpracovat pouze adresáře splňující jednu z následujících podmínek:



- Adresář je uveden v direktivě `FontPath` v části `Files` konfiguračního souboru `/etc/X11/XF86Config`.
- Adresář obsahuje platný soubor `font.dir` (vytvořený skriptem `SuSEconfig`).
- Adresář není za běhu X serveru vyřazen ze seznamu adresářů s fonty příkazem `xset-fp`.
- Adresář je zařazen za běhu X serveru do seznamu adresářů s fonty příkazem `xset+fp`.

Pokud X server už běží, lze nově nainstalované (tj. do příslušných adresářů nako-pírované) fonty zpřístupnit příkazem `xsetfp rehash`. Tento příkaz je spuštěn skriptem `SuSEconfig--module fonts`.

Příkaz `xset` potřebuje přímý přístup k běžícímu X serveru, skript `SuSEconfig--module fonts` tedy musí být spuštěn ze shellu, který k němu přístup má. Toto lze nejjednodušeji zajistit získáním administrátorských oprávnění, tj. zadáním příkazu `su and` hesla uživatele `root`. Příkaz `su` předá přístupu oprávnění uživatele, který spustil X server, administrátorskému shellu. Korrectní instalaci písem a jejich dostupnost prostřednictvím systému X11 core fontů ověříte příkazem `xlsfonts`, jenž vrátí právě seznam všech dostupných písem. SUSE LINUX používá ve výchozím nastavení kódování UTF-8. Je tedy vhodné dávat přednost fontům typu Unicode, jež poznáte tak, že ve výstupu příkazu `xlsfonts` bude jméno fontu končit na `iso10646-1`. Seznam všech Unicodových písem nainstalovaných na vašem systému získáte příkazem `xlsfonts | grep iso10646-1`. Protože téměř všechna písma typu Unicode ze systému SUSE LINUX obsahují alespoň znaky evropských abeced, nahradilo kódování Unicode předchozí kódování `iso-8859-*`.

### Písma s kódováním CID (CID-Keyed)

Narozdíl od jiných typů písem nelze písma s kódováním CID umístěna v libovolném adresáři. Musíte je instalovat do adresáře `/usr/share/ghostscript/Resource/CIDFont`. Pro Xft/fontconfig nehraje sice umístění fontů žádnou roli, ale Ghostscript a systém fontů X11 Core vyžadují, aby se nacházela právě zde.

#### Tip

Další informace o fontech v prostředí X11 najdete na stránce <http://www.xfree86.org/current/fonts.html>.

#### Tip

## 11.4 Konfigurace OpenGL – 3D

### 11.4.1 Podpora hardware

SUSE LINUX používá pro 3D podporu několik OpenGL ovladačů. Jejich přehled se nachází v tabulce 11.4 na této straně:

*Tabulka 11.4: Karty s podporou 3D*

Ovladač OpenGL	Podporovaný hardware
nVidia	čipové sady nVidia: všechny kromě Riva 128(ZX)
DRI	3Dfx Voodoo Banshee, 3Dfx Voodoo-3/4/5, Intel i810/i815/i830M, Intel 845G/852GM/855GM/865G, Matrox G200/G400/G450/G550, ATI Rage 128(Pro)/Radeon

Při instalaci nové karty do systému pomocí programu YaST nebo již při prvotní konfiguraci systému lze aktivovat 3D podporu. Pokud YaST nerozpozná vaši kartu automaticky, můžete ji vybrat sami ze seznamu. Výjimkou jsou grafické čipy společnosti nVidia. Originální ovladač s 3D podporou pro tyto čipy není v distribuci z licenčních důvodů obsažen, a pokud vyžadujete podporu 3D, musíte si ho nejprve stáhnout a nainstalovat – nejsnáze pomocí YOU (YaST Online Update).

Pokud neprovádíte novou instalaci, ale aktualizaci, nebo přidáváte akcelerátor Voodoo Graphics (či Voodoo-2), postup při nastavení 3D podpory se poněkud liší podle toho, jaký ovladač OpenGL použijete. Více informací najdete v následujících odstavcích.

### 11.4.2 Ovladače OpenGL

Prostřednictvím programu SaX2 lze OpenGL nVidia a DRI ovladače jednoduše konfigurovat. V případě karet společnosti nVidia je třeba nejprve stáhnout originální ovladač. Příkazem `3Ddiag` ověříte, zda byla instalace a konfigurace nVidia nebo DRI ovladače úspěšná.

Z bezpečnostních důvodů mají k 3D hardwaru přístup jen uživatelé patřící do skupiny `video`, proto se přesvědčete, že všichni lokální uživatelé jsou členy této skupiny. V opačném případě se ovladač OpenGL přepne do režimu tzv. software-renderingu (vykreslování obrazu má na starosti software a nikoli hardware), což se významně projeví na rychlosti aplikací využívajících OpenGL. Pokud příslušní uživatelé do skupiny `video` nepatří (což ověříte např. příkazem `id`), je vhodné je do skupiny přidat, např. programem `YaST`.

### 11.4.3 Diagnostický nástroj 3Ddiag

Diagnostický nástroj `3Ddiag` slouží v systému SUSE LINUX ke kontrole konfigurace podpory pro 3D. Jedná se o program, který je nutno spouštět z příkazové řádky. Seznam voleb tohoto příkazu získáte zadáním `3Ddiag-h`.

Program zkontroluje, zda jsou nainstalovány balíky zajišťující 3D podporu a zda jsou použity správné knihovny OpenGL popř. rozšíření GLX. Pokud ve výstupu programu najdete hlášení *failed*, řiďte se jeho dalšími instrukcemi. Pokud je všechno v pořádku, objeví se pouze zpráva *done*.

### 11.4.4 Testování OpenGL

Funkčnost OpenGL můžete vyzkoušet programem `glxgears` popř. pomocí her `tuxracer` nebo `armagetron` (balíčky mají stejné názvy). Při aktivované podpoře 3D by měly být hry hratelné i na slabších počítačích, bez této podpory poběží hry pomalu – obraz bude trhaný. Dalším prostředkem, který ověří, zda má váš systém podporu pro 3D, je příkaz `glxinfo|grep direct`, jehož výsledkem by měl být řádek `direct rendering: Yes`.

### 11.4.5 Řešení problémů

Pokud máte s OpenGL nějaké problémy (např. hry jsou trhané), zkontrolujte programem `3Ddiag` konfiguraci OpenGL, a pokud se objeví hlášení *failed*, odstraňte daný problém podle instrukcí. Pokud opravný zásah nepomohl, popřípadě se ve výstupu `3Ddiag` žádná závada neobjevila, přičemž váš problém s 3D přetrvává, nahlédněte do protokolových souborů `X.org`.

Často zjistíte, že se v protokolovém souboru `X.org /var/log/Xorg.0.log` objevuje hláška `DRI is disabled`, jejíž přesnou příčinu lze objevit zevrubným zkoumáním protokolového souboru. Je to však úkol pro zkušeného uživatele.

Pokud se s tím setkáte, většinou se o chybu v konfiguraci nejedná, neboť program 3Ddiag by ji již odhalil. Pak vám obvykle zbývá jediná možnost – používat DRI ovladač v režimu softwarového renderingu, tj. bez využití podpory 3D, kterou obsahuje váš hardware. Pokud dochází k chybám v zobrazení nebo jsou aplikace používající OpenGL nestabilní, bude lepší, když 3D podporu vypnete pomocí SaX2 úplně.

### **11.4.6 Instalační podpora**

Pokud pomineme režim softwarového renderingu v ovladači DRI, jsou všechny linuxové ovladače OpenGL ve vývojovém stádiu a jsou tedy považovány za experimentální. Ovladače byly však zařazeny do distribuce, protože poptávka po podpoře 3D v Linuxu je vysoká. Vzhledem ke stavu ovladačů však nejsme schopni zajistit instalační podporu uživatelům ohledně konfigurace hardwarové akcelerace 3D, ani řešení podobných problémů. Ve výchozím nastavení X serveru není hardwarová akcelerace zapnuta, a pokud se při jejím používání setkáváte s nějakými problémy, doporučujeme ji úplně vyřadit.

### **11.4.7 Dodatečná online dokumentace**

Informace o DRI naleznete v souboru `/usr/X11R6/lib/X11/doc/README.DRI (xorg-x11-doc)`. Více informací o instalaci nVidia ovladačů naleznete na adrese <http://ftp.suse.com/pub/suse/i386/supplementary/X/nvidia-installer-HOWTO.html>.

# Obsluha tisku

V této kapitole najdete obecné informace o práci s tiskárnami a jejich provozu v síti. Zvláštní důraz je kladen na tiskový systém CUPS. Podrobná část o řešení problémů popisuje nejčastější problémy s tiskem a způsob, jak se jim vyhnout.

12.1	Příprava . . . . .	238
12.2	Práce tiskového systému . . . . .	239
12.3	Způsoby a protokoly pro připojení tiskáren . . . . .	240
12.4	Instalace softwaru . . . . .	240
12.5	Konfigurace tiskárny . . . . .	241
12.6	Nastavení aplikací . . . . .	246
12.7	Zvláštní vlastnosti v systému SUSE LINUX . . . . .	247
12.8	Řešení problémů . . . . .	252

## 12.1 Příprava

CUPS je standardní tiskový systém v systému SUSE LINUX a je vysoce uživatelsky orientovaný. V mnoha případech je kompatibilní s LPRng nebo ho je možno poměrně jednoduše přizpůsobit. LPRng je v systému obsažen z důvodů kompatibility.

Tiskárny je možno rozlišovat na základě jejich rozhraní, jako např. USB tiskárny či síťové tiskárny, nebo podle tiskových jazyků. Při nákupu tiskárny se ujistěte, zda je tiskárna vybavena vhodným podporovaným rozhraním a tiskovým jazykem. Podle tiskového jazyka lze tiskárny rozdělit do následujících třech tříd:

**Postscriptové tiskárny** PostScript je tiskový jazyk, ve kterém se v Linuxu a Unixu zpracovávána většina tiskových úloh a který je podporován interním tiskovým systémem. Je to jazyk poměrně starý a velmi efektivní. Pokud umí tiskárna zpracovat přímo postscriptové soubory a není nutné je převádět přes další meziformáty, velmi se snižuje riziko chyb. Protože jsou postscriptové tiskárny zatíženy vysokými licenčními poplatky, jsou obvykle o něco dražší než tiskárny bez podpory tohoto jazyka.

### Standardní tiskárny (jazyky typu PCL a ESC/P)

Ačkoliv i tyto jazyky jsou poměrně staré, stále se vyvíjejí, aby pokryly nové vlastnosti tiskáren. V případě známých jazyků může tiskový systém pomocí Ghostscriptu konvertovat postscriptové úlohy do patřičného jazyka. Tento proces se označuje jako interpretace. Nejznámější jazyky jsou PCL (užívaný zejména tiskárnami HP a jejich klony) a ESC/P (používaný tiskárnami Epson). Jsou obvykle v Linuxu podporovány a tiskový výstup je kvalitní. Linux nicméně nemusí podporovat některé nové a zvláštní vlastnosti tiskáren. S výjimkou ovladačů `hpijs` vyvíjených HP v současnosti žádní výrobci tiskáren nedodávají linuxové ovladače dostupné pod opensource licencí. Cena těchto tiskáren se pohybuje ve střední kategorii.

### Proprietární tiskárny (obvykle GDI tiskárny)

Pro proprietární tiskárny je obvykle k dispozici pouze ovladač pro operační systém Windows. Nepodporují žádný běžný tiskový jazyk a jazyky, které užívají, se mění s každým novým modelem tiskárny. Viz 12.8.1 na straně 252.

Před nákupem nové tiskárny si projděte následující informační zdroje a ověřte si, jak dobře je v Linuxu podporována.

- <http://cdb.suse.de/> nebo databáze tiskáren pro SUSE LINUX
- <http://www.linuxprinting.org/> databáze tiskáren na LinuxPrinting.org
- <http://www.cs.wisc.edu/~ghost/> stránky projektu Ghostscript
- `/usr/share/doc/packages/ghostscript/catalog.devices` ovladače obsažené v systému

Online databáze obsahují vždy aktuální informace o podpoře jednotlivých tiskáren v Linuxu. Distribuce však může obsahovat pouze ovladače dostupné před jejím vydáním. Navíc tiskárny, které jsou dnes označeny jako *perfectly supported* (výborně podporované), nemusely takovou podporu mít v době vydání distribuce. Proto databáze nemusí vždy přesně odpovídat podpoře tiskáren v distribuci SUSE Linuxu.

## 12.2 Práce tiskového systému

Uživatel vytvoří tiskovou úlohu. Tisková úloha sestává z dat, která se mají vytisknout, a z informací pro spooler, jako je jméno tiskárny nebo tiskové fronty, a, volitelně, informací pro filtr, jako jsou volby specifické pro tiskárnu.

Každá tiskárna má vlastní tiskovou frontu. Spooler drží tiskovou úlohu ve frontě, dokud požadovaná tiskárna není připravená přijmout data. Jakmile je tiskárna připravená, pošle jí spooler data skrze filtr a backend.

Filtr zkonvertuje data, která chce uživatel vytisknout (ASCII, PostScript, PDF, JPEG atd.) do dat určených pro tiskárnu. (PostScript, PCL, ESC/P atd.). Vlastnosti tiskárny jsou popsány v PPD souborech. PPD soubor obsahuje volby a parametry specifické pro daný typ tiskárny. Filtr zajistí, aby byly volby vybrané uživatelem zapnuty.

Pokud používáte postscriptovou tiskárnu, zkonvertuje filtr data do PostScriptu specifického pro tiskárnu. To nevyžaduje tiskový ovladač. Pokud používáte nepostscriptovou tiskárnu, zkonvertuje filtr data do formátu specifického pro tiskárnu pomocí programu Ghostscript. To vyžaduje použití ghostscriptového tiskového ovladače vhodného pro vaši tiskárnu. Backend přijme data specifická pro tiskárnu a odešle je tiskárně.

## 12.3 Způsoby a protokoly pro připojení tiskáren

Existuje mnoho různých možností, jak připojit tiskárnu k počítači. Konfigurace systému CUPS nerozlišuje mezi lokálními a síťovými tiskárnami. Lokální tiskárny musí být připojeny tak, jak popisuje jejich výrobce v dodaném manuálu. CUPS podporuje připojení přes sériové, USB, paralelní a SCSI rozhraní. Více informací o připojování tiskáren naleznete v článku *CUPS in a Nutshell* v databázi podpory na adrese <http://portal.suse.com>. Článek naleznete vyhledáním termínu *cups* ve vyhledávacím dialogu.

---

### Varování

#### Kabelové připojení k počítači

Při připojování tiskárny k počítači pamatujte na to, že pouze USB zařízení mohou být připojována či odpojována za provozu. Před změnou jiných typů připojení by měl být systém vypnut.

---

Varování

## 12.4 Instalace softwaru

PPD (PostScript Printer Description) je počítačový jazyk popisující vlastnosti postscriptových tiskáren, např. rozlišení a další možnosti, jako je duplexní jednotka. Pro využití různých vlastností tiskáren v systému CUPS je takový popis nutný. Bez souboru PPD by byla data odeslána tiskárně v nezpracovaném stavu, což je obvykle nežádoucí. Během instalace systému SUSE LINUX je předinstalováno množství PPD souborů, které umožňují použít i tiskárny bez podpory jazyka PostScript.

Nejlepším způsobem konfigurace postscriptové tiskárny je získání patřičného PPD souboru. Mnoho jich je dostupných v balíčku `manufacturer-PPDs`, který je součástí standardní instalace (viz 12.7.4 na straně 250 a 12.8.2 na straně 253).

Nové PPD soubory lze ukládat do adresáře `/usr/share/cups/model/` nebo je přidat do tiskového systému pomocí nástroje YaST (viz 12.5.1 na straně 242). Pak je možné vybraný PPD soubor zvolit při instalaci tiskárny.



Pokud výrobce tiskárny chce instalovat celé softwarové balíčky, nikoliv pouze modifikovat konfigurační soubory, buďte velmi opatrní. Taková instalace znamená nejen ztrátu podpory poskytované SUSE, ale také může změnit funkci tiskových příkazů a způsobit nefunkčnost při práci se zařízeními jiných výrobců. proto takovou instalaci nedoporučujeme.

## 12.5 Konfigurace tiskárny

Po připojení tiskárny k počítači a instalaci softwaru musíte tiskárnu nainstalovat do systému. To by mělo být provedeno nástroji dodanými se systémem SUSE LINUX. Protože SUSE LINUX klade velký důraz na bezpečnost, mají nástroje třetích stran často potíže s bezpečnostními nastaveními a působí mnohdy více potíží než užitku.

### 12.5.1 Lokální tiskárny

Pokud je při vašem přihlášení rozpoznána nenakonfigurovaná lokální tiskárna, spustí se pro její konfiguraci YaST. Dialogy jsou stejné jako v následujícím popisu konfigurace.

Chcete-li nakonfigurovat tiskárnu, zvolte v nástroji YaST 'Hardware' → 'Tiskárna'. Tím se otevře hlavní okno pro konfiguraci tiskárny, v jehož horní části je zobrazen seznam rozpoznávaných zařízení. V dolní části jsou zobrazeny již nakonfigurované fronty. Pokud nebyla vaše tiskárna rozpoznána, nastavte ji ručně.

#### Důležité

Pokud YaST neobsahuje položku 'Tiskárna', není zřejmě nainstalován balíček `yast2-printer`. Doinstalujte ho a restartujte YaST.

#### Důležité

#### Automatická konfigurace

Pokud lze tiskárnu automaticky rozpoznat, umí ji YaST automaticky nakonfigurovat. Je však zapotřebí, aby databáze tiskáren obsahovala ID tiskárny, kterou YaST rozpoznal. Pokud se ID liší, vyberte model tiskárny ručně.

Každá konfigurace by měla být otestována pomocí testovací funkce YaSTu. Vytisknutá testovací stránka obsahuje důležité informace o testované konfiguraci.

## Ruční konfigurace

Pokud vás automatická konfigurace z nějakého důvodu neuspokojuje, nastavte tiskárnu ručně.

Je nutné nastavit následující parametry:

**Způsob připojení (Port)** Konfigurace hardwarového připojení závisí na tom, zda byl YaST schopný tiskárnu automaticky rozpoznat. Pokud se tak stalo, dá se předpokládat, že připojení je na hardwarové úrovni v pořádku a není třeba ho dále nastavovat. Pokud YaST tiskárnu nerozpoznal, může to znamenat problém s hardwarovým připojením. Pak je nutné připojení upravit manuálně.

**Jméno fronty** Jméno fronty se používá při vydávání tiskových příkazů. Mělo by být relativně krátké a skládat se pouze z malých písmen a číslic.

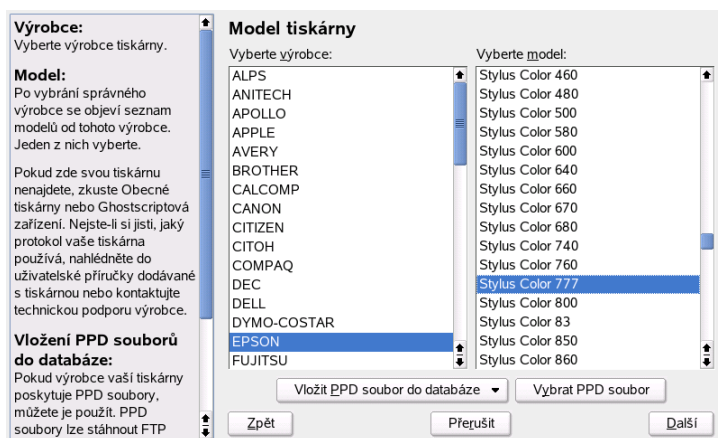
**Model tiskárny a PPD soubor** Všechny parametry specifické pro model tiskárny, jako typ používaného Ghostscript ovladače nebo filtrační parametry ovladače, jsou uloženy v PPD souboru (PostScript Printer Description). Viz 12.4 na straně 240.

Pro mnoho typů tiskáren je dostupných více PPD souborů, například tehdy, když s daným modelem funguje více Ghostscript ovladačů. Při výběru výrobce a modelu tiskárny YaST sám zvolí vhodný PPD soubor. Pokud je pro tiskárnu k dispozici více PPD souborů, vybere YaST obvykle ten, který je označen jako doporučený (*recommended*). Tento výchozí PPD soubor můžete změnit po kliknutí na 'Upravit'.

V případě nepostscriptových tiskáren jsou všechna data specifická pro tiskárnu vytvářena Ghostscript ovladačem. Proto je nastavení ovladače nejdůležitějším faktorem ovlivňujícím kvalitu tiskového výstupu. Tisk je ovlivněn jak druhem Ghostscript ovladače (PPD souboru), tak i pro něj nastavenými volbami. Pokud je to nutné, změňte další volby (dostupné díky PPD souboru) po kliknutí na 'Upravit'.

Nastavení tisku vždy zkontrolujte vytištěním testovací stránky. Pokud je výstup špatný, například obsahuje několik prázdných stránek, zastavte tisk odstraněním papírů z tiskárny a následným přerušením tisku v YaSTu.

Pokud databáze tiskáren neobsahuje vaši tiskárnu, můžete přidat nový PPD soubor kliknutím na 'Vložit PPD soubor do databáze' nebo použít některý z obecných PPD souborů a zprovoznit tiskárnu pomocí standardního tiskového jazyka. Učiníte tak volbou výrobce tiskárny 'UNKNOWN MANUFACTURER' (neznámý výrobce).



Obrázek 12.1: Výběr modelu tiskárny

**Pokročilé nastavení** Za běžných okolností není třeba do pokročilého nastavení zasahovat.

### Konfigurace tiskárny pomocí příkazové řádky

Chcete-li tiskárnu konfigurovat ručně pomocí nástrojů pro příkazovou řádku, které jsou popsány v části 12.5.3 na straně 245, potřebujete URI (Uniform Resource Identifier) zařízení. To se skládá z backendu, například `usb`, a parametrů, jako `/dev/usb/lp0`. Plné URI může například být `parallel:/dev/lp0` (tiskárna na prvním paralelním portu) nebo `usb:/dev/usb/lp0` (první rozpoznaná tiskárna na USB portu).

### 12.5.2 Síťové tiskárny

Síťová tiskárna může podporovat různé protokoly, někdy dokonce více protokolů najednou. Přestože je většina protokolů standardizována, někteří výrobci protokoly modifikují, protože chtějí nabídnout funkce, které standard nepodporuje. Nabídnou k tiskárně ovladače pro několik málo systémů, na nichž tak odstraní problémy s protokolem. Bohužel, linuxové ovladače jsou dodávány jen

zřídka. V současné době nelze předpokládat, že v Linuxu bude fungovat libovolný protokol. Proto je někdy k dosažení funkčnosti třeba experimentovat s nastavením.

CUPS podporuje protokoly `socket`, `LPD`, `IPP` a `smb`:

**socket** *Socket* je připojení, během kterého jsou data posílána na TCP/IP socket bez předchozího navazování spojení (*handshaking*). Mezi běžně používané porty socketů se řadí 9100 a 35. Příklad URI zařízení je `socket://host-printer:9100/`.

**LPD (Line Printer Daemon)** Spolehlivý protokol LPD je popsán v dokumentu RFC1179. Při použití tohoto protokolu jsou některé údaje spojené s tiskovou úlohou (např. ID tiskové fronty) zasílány před vlastními tiskovými daty. Proto musí být při konfiguraci LPD protokolu pro datový přenos specifikována tisková fronta. Implementace různých výrobců jsou většinou natolik flexibilní, že je možné používat jakékoliv jméno fronty. V případě potřeby by správné jméno mělo být uvedeno v manuálu tiskárny. Obvykle se používají jména jako LPT, LPT1, LP1 apod. LPD fronta může být samozřejmě nastavena v systému CUPS i na jiných linuxových či unixových počítačích. Číslo portu pro službu LPD je 515. Příklad URI je `lpd://host-printer/LPT1`.

**IPP (Internet Printing Protocol)** IPP je poměrně nový (1999) protokol založený na HTTP. Při použití IPP je přenášeno více dat spojených s úlohou než u jiných protokolů. CUPS používá protokol IPP pro vnitřní datové přenosy. Je to upřednostňovaný protokol pro předávací frontu mezi dvěma CUPS servery. Jméno tiskové fronty je nutno nastavit správně. Používaný port je 631. Příklad URI je `ipp://host-printer/ps` nebo `ipp://host-cupsserver/printers/ps`.

**SMB (Windows Share)** CUPS umožňuje tisk i na sdílených tiskárnách Windows. Používaný protokol je SMB. Používané porty jsou 137, 138 a 139. URI může být například `smb://Uzivatel:Heslo@PracovniSkupina/Server/Tiskarna`, `smb://Uzivatel:Heslo@Pocitac/Tiskarna` nebo `smb://Server/Tiskarna`.

Protokol, který tiskárna podporuje, musí být určen před vlastní konfigurací. Pokud výrobce potřebné informace neuvádí, lze protokol odhadnout příkazem `nmap` (balíček `nmap`). Program `nmap` hledá na tiskárně otevřené porty. Například:

```
nmap -p 35,137-139,515,631,9100-10000 IP_tiskarny
```

### 12.5.3 Konfigurace

Konfiguraci lze provést pomocí nástroje YaST nebo pomocí nástrojů pro příkazovou řádku.

#### Konfigurace CUPS v síti pomocí YaST

Síťové tiskárny by měly být konfigurovány nástrojem YaST, který je nejlépe vybaven pro práci s bezpečnostními omezeními systému CUPS. (viz kapitola 12.7.2 na straně 248).

Více informací o instalaci CUPS v síti naleznete v článku *CUPS in a Nutshell* v databázi podpory na adrese <http://portal.suse.com>.

#### Konfigurace pomocí nástrojů pro příkazovou řádku

CUPS lze nakonfigurovat i přes příkazovou řádku nástroji jako `lpadmin` a `lpoptions`. Pokud jste již učinili přípravné práce (máte PPD soubor a znáte jméno zařízení), pokračujte následujícím způsobem:

```
lpadmin -p fronta -v URIzařízení \  
-P PPDsoubor -E
```

Volbu `-E` nepoužívejte jako první. U všech CUPS příkazů znamená `-E` jako první argument použití šifrovaného spojení. Pro zprovoznění tiskárny musí být argument `-E` použit tak jako v následujících příkladech:

```
lpadmin -p ps -v parallel:/dev/lp0 -P \  
/usr/share/cups/model/Postscript.ppd.gz -E
```

Příklad pro síťovou tiskárnu:

```
lpadmin -p ps -v socket://192.168.1.0:9100/ -P \  
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

#### Úprava voleb

Během instalace systému jsou určité volby nastaveny jako výchozí. Volby lze pak pro jednotlivé tiskové úlohy měnit (v závislosti na tiskovém nástroji) nebo je měnit trvale, například pomocí YaST. Pomocí nástrojů pro příkazovou řádku toho dosáhnete následujícím způsobem:

1. Nejprve zobrazte všechny volby:

```
lptions -p fronta -l
```

Příklad:

```
Resolution/Output Resolution: 150dpi *300dpi 600dpi
```

Aktivovaná výchozí volba je označena hvězdičkou.

2. Změňte volbu příkazem `lpadmin`:

```
lpadmin -p fronta -o Resolution=600dpi
```

3. Zkontrolujte nové nastavení:

```
lptions -p fronta -l
```

```
Resolution/Output Resolution: 150dpi 300dpi *600dpi
```

## 12.6 Nastavení aplikací

Aplikace tisknou do tiskových front podobným způsobem jako příkazy z příkazové řádky. Pro tisk z aplikací není nutné přenastavovat tiskárnu, tisk bude prováděn pomocí již nastavených front.

### 12.6.1 Tisk z příkazové řádky

Pro tisk z příkazové řádky zadejte příkaz `lp -d <jmeno_fronty> <jmeno_souboru>`, kde `<jmeno_fronty>` nahradíte jménem tiskové fronty, kterou chcete použít, a `<jmeno_souboru>` nahradíte jménem souboru, který si přejete vytisknout.

### 12.6.2 Tisk z aplikací pomocí příkazů příkazové řádky

Některé aplikace používají pro tisk příkaz `lp`. V takovém případě do tiskového dialogu aplikace zadejte správný tiskový příkaz (obvykle bez jména `<souboru>`), např. `lp -d <jmeno_fronty>`. Aby tento postup fungoval také v programech z prostředí KDE, musíte v ovládacím centru KDE v nastavení tiskáren povolit 'Tisk pomocí externího programu'. V opačném případě nelze příkaz zadat.

### 12.6.3 Použití tiskového systému CUPS

Nástroje jako `xpp` nebo `kprinter` z prostředí KDE poskytují grafické rozhraní pro výběr tiskových front, nastavení voleb systému CUPS a nastavení vlastností tiskáren pomocí PPD souboru. Aplikaci `kprinter` můžete použít jako standardní tiskové rozhraní také pro ostatní (ne z KDE) programy zadáním příkazu `kprinter` nebo `kprinter--stdin` jako tiskového příkazu v těchto aplikacích. Volba příkazu je závislá na chování programu. Pokud je nastaven správně, program spustí při každém tisku dialog aplikace `kprinter`, ve kterém můžete zvolit požadovanou frontu a další tiskové volby. Samozřejmě je nutné, aby nativní nastavení tisku aplikace s programem `kprinter` nekolidovalo a aby tiskové volby byly nastavované pouze přes `kprinter`.

## 12.7 Zvláštní vlastnosti v systému SUSE LINUX

V SUSE Linuxu je v systému CUPS řada zajímavých vlastností. O těch nejdůležitějších se píše v následujícím textu:

### 12.7.1 CUPS server a firewall

Existuje několik možností, jak nastavit CUPS jako klienta síťového serveru.

- Ke každé frontě na síťovém serveru můžete nastavit lokální frontu, přes kterou lze přeposílat tiskové úlohy na správný server. Tento přístup nelze obecně doporučit, neboť v případě změny konfigurace na serveru je nutno přenastavit i všechny klienty.
- Tiskové úlohy je též možno přeposílat přímo na jeden síťový server. Při použití tohoto typu konfigurace nespouštějte lokálního démona CUPS. `lp` (a odpovídající knihovni volání dalších programů) umožňuje zasílat úlohy přímo na síťový server. Tuto konfiguraci však nelze použít, pokud chcete používat lokální tiskárnu.
- Démon CUPS může naslouchat oznamovacím IPP paketům vysílaným síťovými servery pro oznámení dostupných front. Je to nejlepší možná CUPS konfigurace pro tisk na vzdálených CUPS serverech. Existuje ovšem riziko, že útočník vyšle falešné IPP pakety a lokální démon pak zašle tisková

data na podvrženou frontu. Při používání této konfigurace musí být port 631/UDP otevřen pro příchozí pakety.

YaST může použít dvě metody vyhledávání CUPS serverů. Může skenovat všechny počítače na síti a zjišťovat, zda nabízejí službu CUPS, nebo může naslouchat IPP paketům (metoda popsaná výše). Takto jsou také během instalace vyhledávány CUPS servery nabízející služby. Druhá metoda vyžaduje otevření portu 631/UDP pro příchozí pakety.

Výchozí nastavení firewallu zakazuje naslouchat IPP oznamovacím paketům na všech rozhraních. Proto nemůže fungovat druhá metoda vyhledávání vzdálených front ani třetí metoda pro přístup ke vzdáleným frontám. Je tedy potřeba změnit nastavení firewallu. Je možné některé ze síťových rozhraní nastavit jako vnitřní (na kterém je port defaultně otevřen) nebo explicitně otevřít port na vnějším rozhraní. Z bezpečnostních důvodů není žádný z portů ve výchozím nastavení otevřen. Otevření portu pro konfiguraci vzdálených front druhou metodou může znamenat bezpečnostní riziko.

Nabídnuté nastavení firewallu je nutno změnit, aby mohl CUPS server během instalace detekovat vzdálené fronty. Jinou možností je oskenovat všechny lokální počítače nebo nakonfigurovat fronty ručně. Z důvodů zmíněných výše to však nedoporučujeme.

## 12.7.2 Administrátor webového frontendu CUPS

Pro administraci přes webový frontend (CUPS) nebo nástroj pro administraci tiskáren v KDE je nutné nastavit uživatele `root` jako CUPS administrátora, CUPS administrační skupinu `sys` a CUPS heslo. Učinit tak může uživatel `root` následujícím příkazem:

```
lppasswd -g sys -a root
```

Pokud toto nastavení neprovedete, nebude možná administrace přes webové rozhraní nebo administrační nástroj v KDE, protože autentizace bez nastavení CUPS administrátora selže. Jako CUPS administrátor může být nastaven i jakýkoliv jiný uživatel (viz 12.7.3 na této straně).

## 12.7.3 Změny v tiskové službě CUPS (cupsd)

Tyto změny byly poprvé provedeny v systému SUSE LINUX 9.1.



## cupsd běží pod uživatelem lp

Při spuštění se program `cupsd` přepne z běhu pod uživatelem `root` na uživatele `lp`. Tím je dosaženo vyšší bezpečnosti, protože služba CUPS tak běží jen s potřebnými právy.

Nicméně autentizace (lépe řečeno kontrola hesla) nemůže být provedena přes `/etc/shadow`, protože uživatel `lp` k němu nemá přístup. Místo toho je použita autentizace specifická pro CUPS přes soubor `/etc/cups/passwd.md5`. Proto je do tohoto souboru nutné vložit CUPS administrátora, CUPS administrační skupinu `sys` a heslo. Provést to může uživatel `root` následujícím příkazem:

```
lppasswd -g sys -a CUPS-administrátor
```

Pokud běží `cupsd` pod uživatelem `lp`, nemůže vygenerovat soubor `/etc/printcap`, neboť nemá právo zapisovat do adresáře `/etc/`. Místo toho `cupsd` vytvoří `/etc/cups/printcap`. Aby nebyla ohrožena funkce aplikací, které umí číst jména front pouze z `/etc/printcap`, je `/etc/printcap` symbolickým odkazem na `/etc/cups/printcap`.

Když `cupsd` běží pod uživatelem `lp`, nelze otevřít port 631. Proto nelze použít příkaz `rc cups reload`. Místo něj použijte `rc cups restart`.

## Obecná funkce BrowseAllow a BrowseDeny

Přístupová práva nastavená pro `BrowseAllow` a `BrowseDeny` platí pro všechny pakety zaslané na `cupsd`. Výchozí nastavení v souboru `/etc/cups/cupsd.conf` jsou následující:

```
BrowseAllow @LOCAL
BrowseDeny All
```

a

```
<Location />
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 127.0.0.2
  Allow From @LOCAL
</Location>
```

Při tomto nastavení mohou ke `cupsd` na CUPS serveru přistupovat pouze `LOCAL` počítače, tj. počítače, jejichž IP adresa náleží non-PPP rozhraní (přesněji rozhraní, jehož `IFF_POINTOPOINT` příznak není nastaven) a jejichž adresa náleží do stejné sítě jako CUPS server. Pakety z ostatních počítačů jsou okamžitě odmítnuty.

## **cupsd je defaultně aktivní**

Ve standardní instalaci je cupsd automaticky aktivní, což umožňuje pohodlný přístup ke CUPS frontám bez manuálního nastavování. Dvě předchozí vlastnosti (viz 12.7.3 na předchozí straně a 12.7.3 na předchozí straně) jsou podmínkou k tomuto automatickému spuštění, neboť jinak by nebyla zajištěna dostatečná bezpečnost.

## **12.7.4 PPD soubory v různých balíčcích**

V této části jsou popsány zdroje PPD souborů a jejich použití.

### **Konfigurace tiskáren pouze pomocí PPD souborů**

Modul pro konfiguraci tiskáren nástroje YaST nastavuje CUPS fronty pouze s využitím PPD souborů v `/usr/share/cups/model/`. Vhodný PPD soubor vybírá YaST porovnáním modelu tiskárny zjištěného během rozpoznávání hardwaru a modelů v PPD souborech v adresáři `/usr/share/cups/model/`. Za tímto účelem si YaST vytváří databázi modelů tiskáren získaných z PPD souborů. Když vyberete model ze seznamu výrobců a typů tiskáren, bude automaticky přiřazen vhodný PPD soubor.

Konfigurace s využitím pouze PPD souborů a žádných jiných informací má výhodu v tom, že je možné PPD soubory v adresáři `/usr/share/cups/model/` volně modifikovat. Modul YaST pro nastavení tiskáren si všímá všech změn a obnovuje svou databázi. Pokud například máte jen postscriptové tiskárny, nepotřebujete Foomatic PPD soubory z balíčku `cups-drivers` ani Gimp-Print PPD z balíčku `cups-drivers-stp`. Místo toho můžete prostě přkopírovat PPD soubory pro vaše postscriptové tiskárny přímo do adresáře `/usr/share/cups/model/` (pokud nejsou již součástí balíčku `manufacturer-PPDs`).

### **PPD soubory v balíčku cups**

Obecné PPD soubory v balíčku `cups` byly doplněny upravenými Foomatic PPD soubory pro tiskárny PostScript level 1 a 2:

- `/usr/share/cups/model/Postscript-level1.ppd.gz`
- `/usr/share/cups/model/Postscript-level2.ppd.gz`

## PPD soubory v balíčku cups-drivers

Normálně je pro nepostscriptové tiskárny používán Foomatic tiskový filtr `foomatic-rip` spolu s Ghostscriptem. Vhodné Foomatic PPD soubory s položkami `*NickName: ... Foomatic/Ghostscript driver` a `*cupsFilter: ... foomatic-rip` jsou umístěny v balíčku `cups-drivers`.

YaST upřednostňuje Foomatic PPD soubory za následujících podmínek:

- Foomatic PPD soubor s položkou `*NickName: ... Foomatic ... (recommended)` odpovídá modelu tiskárny.
- Balíček `manufacturer-PPDs` neobsahuje vhodnější PPD soubor (viz níže).

## Gimp-Print PPD soubory v balíčku cups-drivers-stp

Místo `foomatic-rip` lze s mnoha nepostscriptovými tiskárnami použít CUPS filtr `rastertoprinter` z projektu Gimp-Print. Tento filtr a vhodné Gimp-Print PPD soubory jsou dostupné v balíčku `cups-drivers-stp`. Gimp-Print PPD soubory jsou umístěné v adresáři `/usr/share/cups/model/stp/` a mají položky `*NickName: ... CUPS+Gimp-Print` a `*cupsFilter: ... rastertoprinter`.

## PPD soubory od výrobců tiskáren v balíčku manufacturer-PPDs

Balíček `manufacturer-PPDs` obsahuje PPD soubory od výrobců tiskáren, pokud jsou uvolněny pod dostatečně volnou licenci. Postscriptové tiskárny by měly být nakonfigurovány s příslušným PPD souborem od výrobce, protože jsou tak dostupné všechny funkce tiskárny. YaST upřednostňuje PPD soubor z balíčku `manufacturer-PPDs` za následujících podmínek:

- Výrobce a model tiskárny zjištěný během detekce hardwaru odpovídá výrobcí a modelu tiskárny uvedeným v PPD souboru z balíčku `manufacturer-PPDs`.
- PPD soubor z balíčku `manufacturer-PPDs` je jediný vhodný PPD soubor pro danou tiskárnu nebo existuje Foomatic PPD soubor s položkou `*NickName: ... Foomatic/Postscript (recommended)`, který rovněž odpovídá dané tiskárně.

YaST nepoužije žádný soubor z balíčku `manufacturer-PPDs` v následujících případech:

- PPD soubor z balíčku `manufacturer-PPDs` neodpovídá výrobci a modelu tiskárny. To se může stát v případě, že balíček obsahuje jen jeden PPD soubor pro několik podobných tiskáren.
- Foomatic PostScript PPD soubor není *recommended* (doporučený). To může být v případě, kdy daná tiskárna nefunguje v postscriptovém režimu efektivně, například je v tomto režimu nespolehlivá pro nedostatek paměti či pomalá kvůli slabému procesoru. Dalším důvodem může být to, že tiskárna nepodporuje PostScript ve výchozí konfiguraci (je např. dostupný jako rozšiřující výbava).

Pokud je některý PPD soubor z balíčku `manufacturer-PPDs` pro postscriptovou tiskárnu vhodný, ale YaST ho nepoužije z výše zmíněných důvodů, zvolte vybraný model tiskárny v nástroji YaST ručně.

## 12.8 Řešení problémů

Následující odstavce se zabývají řešením nejčastějších hardwarových i softwarových problémů s tiskem.

### 12.8.1 Tiskárny bez podpory standardního tiskového jazyka

Tiskárny, které nepodporují žádný standardní tiskový jazyk, ale je s nimi možno komunikovat pouze pomocí speciálních kontrolních sekvencí, se nazývají *GDI tiskárny*. Takové tiskárny jsou funkční pouze s operačním systémem, ke kterému výrobce dodává ovladač. *GDI* je programovací rozhraní vyvinuté firmou Microsoft pro grafická zařízení. Problémem není programovací rozhraní jako takové, ale skutečnost, že pro komunikaci s *GDI* tiskárnami lze použít pouze proprietární jazyk specifický pro daný typ tiskárny.

Některé tiskárny lze používat v režimu *GDI* i v režimu standardního tiskového jazyka. Někteří výrobci dodávají ke *GDI* tiskárnám proprietární ovladače. Nevýhoda takových ovladačů ale spočívá v tom, že nemusí být vhodné pro všechny tiskové systémy či hardwarové platformy. Tiskárny podporující standardní tiskový jazyk jsou naopak na tiskovém systému či hardwarové platformě nezávislé. Často může být výhodnější zakoupit podporovanou tiskárnu se standardním tiskovým jazykem, než trávit čas snahou zprovoznit proprietární linuxový ovladač. Problém s ovladači se tak vyřeší jednou pro vždy a odstraní se nutnost instalovat a konfigurovat speciální ovládací software a shánět jeho nové verze v případě změn v tiskovém systému.

## 12.8.2 Pro postscriptovou tiskárnu není k dispozici vhodný PPD soubor

Pokud balíček `manufacturer-PPDs` neobsahuje pro vaši postscriptovou tiskárnu žádný vhodný PPD soubor, zkuste použít PPD soubor z CD s ovladači dodaného s tiskárnou nebo stáhněte soubor z webových stránek výrobce.

Pokud je PPD soubor k dispozici ve formě zip archívu (.zip) nebo samorozbalovacího zip archívu (.exe), rozbalte ho programem `unzip`. Přečtěte si licenční podmínky souboru a pomocí programu `cupstestppd` ověřte, zda odpovídá specifikaci *Adobe PostScript Printer Description File Format Specification, version 4.3*. Pokud program vrátí `FAIL`, jsou v PPD souboru závažné chyby, které mohou způsobit problémy. Proto by objevené chyby měly být odstraněny. Pokud je to nutné, požádejte výrobce tiskárny o vhodný PPD soubor.

## 12.8.3 Paralelní porty

Nejspolehlivější je připojit tiskárnu přímo k prvnímu paralelnímu portu a v BIOSu zvolit následující nastavení:

- I/O address: 378 (hexadecimal)
- Interrupt: irrelevant
- Mode: Normal, SPP nebo Output Only
- DMA: disabled

Pokud tiskárna na paralelním portu s tímto nastavením BIOSu nefunguje, explicitně vložte I/O adresu nastavenou v BIOSu do souboru `/etc/modprobe.conf` ve tvaru `0x378`. Pokud jsou paralelní porty dva a jejich I/O adresy jsou `378` a `278` (hexadecimálně), vložte je do souboru ve tvaru `0x378, 0x278`.

Pokud je volné přerušení 7, lze ho aktivovat zápisem nastavení uvedeným v příkladu 12.1 na této straně. Před aktivací přerušení zkontrolujte v souboru `/proc/interrupts`, jaká přerušení se již používají. Jsou tam zobrazena jen právě používaná přerušení, což závisí na právě aktivních hardwarových komponentách. Přerušení pro paralelní port nesmí být používáno žádným jiným zařízením. Pokud si nejste jisti, použijte `irq=none`.

**Příklad 12.1:** `/etc/modprobe.conf`: Režim přerušení pro první paralelní port

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

## 12.8.4 Připojení síťových tiskáren

**Identifikace síťových problémů** Připojte tiskárnu přímo k počítači. Nakonfiguruje ji pro účely testování jako lokální. Pokud funguje, problém je spojený se sítí.

**Kontrola TCP/IP sítě** TCP/IP síť a převod jmen musí být funkční.

**Kontrola vzdáleného lpd** Následujícím příkazem otestujte, zda je možné navázat TCP spojení s lpd (port 515) na vzdáleném počítači *<host>*:

```
netcat -z host; 515 && echo ok || echo selhalo
```

Pokud spojení s lpd nelze navázat, je možné, že lpd není aktivní, nebo, že jsou vážné problémy se sítí.

Jako uživatel *root* použijte následující příkaz k získání (možná velmi dlouhé) zprávy o stavu fronty *<queue>* na vzdáleném počítači *<host>*, za předpokladu, že je lpd aktivní a vzdálený počítač odpovídá na dotazy:

```
echo -e "\004queue" \  
| netcat -w 2 -p 722 host 515
```

Pokud lpd neodpovídá, může být neaktivní nebo může být problém se sítí. Pokud lpd odpoví, měla by odpověď ozřejmit, proč nelze na frontě *queue* na počítači *host* tisknout. Pokud dostanete odpověď jako v příkladu 12.2 na této straně, je problém způsobený vzdáleným lpd:

*Příklad 12.2: Chybové hlášení programu lpd*

```
lpd: your host does not have line printer access  
lpd: queue does not exist  
printer: spooling disabled  
printer: printing disabled
```

**Kontrola vzdáleného cupsd** Ve výchozím nastavení by měl CUPS server oznamovat své fronty každých třicet sekund na UDP portu 631. Následující příkaz testuje, zda je na síti přítomný CUPS síťový server.

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

Pokud síťový CUPS server skutečně existuje, vrátí se za čtyřicet sekund zpráva zobrazená v příkladu 12.3 na následující straně.

**Příklad 12.3: Oznámení síťového CUPS serveru**

```
ipp://pocitac.domena:631/printers/fronta
```

Následující příkaz lze použít k otestování možnosti navázání TCP spojení s cupsd (port 631) na vzdáleném počítači *<host>*:

```
netcat -z host 631 && echo ok || echo selhalo
```

Pokud nelze spojení navázat, je cupsd neaktivní nebo jsou závažné problémy se sítí. Příkaz `lpstat -h host -l -t` vrátí (možná velmi dlouhou) zprávu o stavu všech front na vzdáleném počítači *<host>*, pokud je cupsd aktivní a počítač odpovídá na dotazy.

Následující příkaz lze použít k otestování, zda fronta *<queue>* na počítači *<host>* přijme tiskovou úlohu sestávající z jednoho znaku carriage return (nový řádek). Vytisknuto by nemělo být nic, jen možná vysunut jeden prázdný list papíru.

```
echo -en "\r" \
| lp -d queue -h host
```

**Řešení problémů se síťovou tiskárnou nebo zařízením *print server box*.**

Při velkém množství tiskových úloh se občas objeví problémy se spoolery běžícími v zařízení *print server box*. Problém nelze řešit přímo, ale můžete spooler obejít adresováním tiskárny přímo přes TCP soket (viz 12.5.2 na straně 243).

Abyste mohli tuto metodu použít, musíte znát příslušný port na zařízení *print server box*. Když je tiskárna zapnuta a připojena k tomuto zařízení, lze TCP port určit krátce po zapnutí zařízení pomocí programu `nmap`. Příkaz `nmap <IP_adresa>` může mít pro zařízení *print server box* následující výstup:

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

Tento výstup značí, že tiskárnu připojenou k zařízení lze adresovat přes TCP soket na portu 9100. Ve výchozím nastavení kontroluje nmap jen běžně používané porty uvedené v `/usr/share/nmap/nmap-services`. Chcete-li zkontrolovat všechny možné porty, použijte příkaz `nmap -p<od_portu>-<do_portu><IP_adresa>`. Může to ale trvat poměrně dlouho. Další informace naleznete v manuálové stránce `nmap`.

K otestování, zda lze tiskárnu na určitém portu adresovat, zašlete na příslušný port následujícím příkazem řetězce nebo soubory k tisku:

```
echo -en "\rAhoj\r\f" | netcat -w 1 IP_adresa port
cat soubor | netcat -w 1 IP_adresa port
```

### 12.8.5 Problém s tiskem bez chybového hlášení

Tiskový systém považuje úlohu za hotovou v okamžiku, kdy dokončí přenos dat příjemci (tiskárně). Pokud zpracování na tiskárně z nějakého důvodu selže (pokud například tiskárna nedokáže zpracovat data specifická pro určitou tiskárnu), tiskový systém se o tom nedozví. Pokud není tiskárna schopna vytisknout data specifická pro tiskárnu, použijte jiný, pro vaši tiskárnu vhodnější, PPD soubor.

### 12.8.6 Nepřístupné fronty

Pokud datový přenos k příjemci z nějakého důvodu i po několika pokusech selže, oznámí CUPS backend (např. `usb` nebo `socket`) tiskovému systému (přesněji `cupsd`) chybu. Backend rozhoduje o tom, kolik pokusů o přenos dat má smysl, a kdy prohlásí spojení za nemožné. Protože v takovém případě by další pokusy byly zbytečné, zablokuje `cupsd` na příslušné frontě tisk. Jakmile odstraníte zdroj problémů, musí systémový administrátor reaktivovat tisk na frontě příkazem `/usr/bin/enable`.

### 12.8.7 Rušení tiskových úloh

Pokud síťový CUPS server oznamuje fronty klientským počítačům přes prohlížení sítě a na klientovi je vhodně nastaven `cupsd`, přijímá od aplikací tiskové úlohy klientský `cupsd` a přeposílá je programu `cupsd` na serveru. Když `cupsd` tiskovou úlohu přijme, je jí přiřazeno nové číslo. Proto je číslo úlohy jiné na klientovi a jiné na serveru. Protože je tisková úloha obvykle přeposílána ihned, nelze



ji zrušit pomocí čísla na klientovi. Klientský `cupsd` považuje tiskovou úlohu za dokončenou v okamžiku jejího přeposlání na server.

Chcete-li úlohu na serveru zrušit, použijte příkaz `lpstat -h tiskovyserver -o` ke zjištění čísla úlohy na serveru (za předpokladu, že server úlohu dosud nedokončil, tj. neposlal ji na tiskárnu). Pomocí takto získaného čísla můžete úlohu na serveru zrušit:

```
cancel -h tiskovyserver fronta-cisloulohy
```

### 12.8.8 Vadné tiskové úlohy a chyby v přenosu dat

Tiskové úlohy ve frontách zůstávají i když vypnete a zapnete tiskárnu nebo restartujete počítač během tisku. Vadné tiskové úlohy je nutno odstranit z fronty pomocí příkazu `cancel`.

Pokud je tisková úloha vadná nebo se objeví chyba v komunikaci mezi počítačem a tiskárnou, vytiskne tiskárna mnoho listů papíru s nečitelnými znaky, neboť není schopná data správně zpracovat. Vypořádat se s ní můžete následujícím způsobem:

1. Chcete-li tisk zastavit, vyjměte z inkoustových tiskáren papír nebo, u tiskáren laserových, otevřete zásobníky papíru. Kvalitní tiskárny mají pro zastavení tisku zvláštní tlačítko.
2. Tisková úloha může ve frontě přetrvávat, neboť úlohy jsou odstraňovány, až když jsou odeslány celé. Příkazem `lpstat -o` (nebo `lpstat -h <tiskovy-server> -o`) zjistíte, která fronta se právě tiskne. Úlohu pak odstraníte příkazem `cancel <fronta>-<cislo-ulohy>` (nebo `cancel -h <tiskovy-server> <fronta>-<cislo-ulohy>`).
3. Někdy je část dat tiskárně odesílána i v případě, že tisková úloha byla z fronty odstraněna. Ověřte si, zda pro frontu stále běží CUPS backend proces, a pokud ano, ukončete ho. Například (v případě tiskárny na paralelním portu) lze použít příkaz `fuser -k /dev/lp0`, který ukončí všechny procesy přistupující k tiskárně (či přesněji k paralelnímu portu).
4. Tiskárnu resetujte jejím vypnutím. Po chvilce do ní vložte papír a zapněte ji.

## 12.8.9 Hledání problémů v tiskovém systému CUPS

Chcete-li identifikovat problém v tiskovém systému CUPS, použijte následující postup:

1. Nastavte `LogLevel debug` v souboru `/etc/cups/cupsd.conf`.
2. Zastavte `cupsd`.
3. Odstraňte `/var/log/cups/error_log*`, vyhněte se tak prohledávání příliš velkého protokolového souboru.
4. Spust'te `cupsd`.
5. Zopakujte činnost, která vedla k problému.
6. Zkontrolujte záznamy v souboru `/var/log/cups/error_log*`. Měly by vést k odhalení problému.

## 12.8.10 Další informace

Řešení mnoha specifických problémů najdete v Databázi podpory. Pokud vám ničí nervy tiskárny, přečtete si v této databázi články *Installing a Printer* a *Printer Configuration from SUSE LINUX 9.2*, které naleznete vyhledáním slova „printer“.

# Mobilita v Linuxu

Tato kapitola pojednává o používání Linuxu ve světě mobilních počítačů. Krátce si představíme různé oblasti a dostupná zařízení, najdete část o potřebných aplikacích i informace o možnostech minimalizace spotřeby. Na konci najdete odkazy na nejdůležitější zdroje informací.

13.1	Notebooky . . . . .	260
13.2	Mobilní hardware . . . . .	265
13.3	Mobilní telefony a kapesní počítače . . . . .	266
13.4	Další informace . . . . .	267

Většina lidí si při slově mobilita představí notebooky, kapesní počítače a mobilní telefony. Tato kapitola se však zaměřuje také na další zařízení jako jsou externí disky, flash disky nebo digitální fotoaparáty, které můžete připojovat jak k notebookům, tak k pracovním stanicím.

## 13.1 Notebooky

### 13.1.1 Zvláštní hardwarové vlastnosti notebooků

Z důvodů důrazu na mobilitu, minimální prostorové nároky a spotřebu energie se hardware notebooků od obyčejných stolních počítačů v mnoha ohledech odlišuje. Výrobci mobilních zařízení vyvinuli standard PCMCIA (*Personal Computer Memory Card International Association*), který pokrývá oblast paměťových karet, síťových rozhraní jako síťové karty a modemy a externích disků. Informace o implementaci tohoto standardu v Linuxu, potřebných nastaveních, dostupných aplikacích a řešení možných problémů najdete v kapitole 14 na straně 269.

### 13.1.2 Snížení spotřeby energie

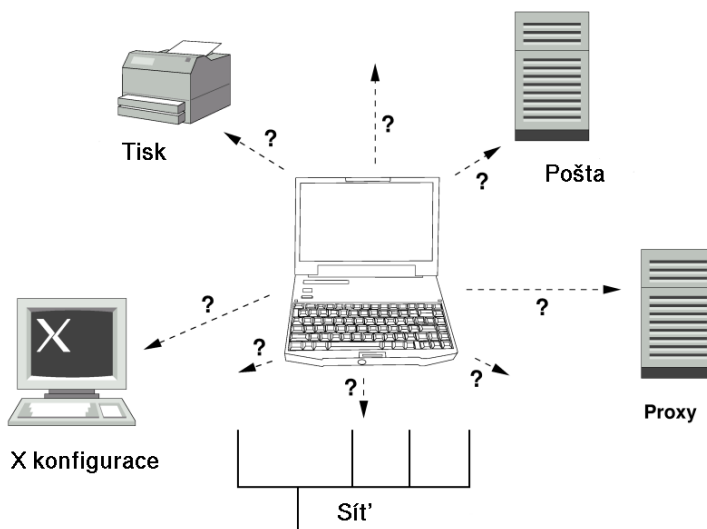
Řada komponent je již od výrobce navržena a optimalizována tak, aby měla v případě napájení z baterií co nejnižší spotřebu energie. Podíl takto upravených komponent na úspoře energie je přinejmenším stejně tak důležitý jako schopnosti operačního systému. SUSE LINUX řadu metod úspory spotřeby energie při napájení z baterie. Následující seznam možných způsobů snížení spotřeby je seřazen podle významu dopadu na spotřebu:

- Zpomalení rychlosti CPU
- Vypnutí monitoru během nečinnosti
- Ruční nastavení parametrů monitoru
- Odpojení nepoužívaných zařízení (USB CD-ROM, externí myš, nepoužívané PCMCIA karty, atd..)
- Zastavení disku při nečinnosti

Podrobnější informace o správě napájení v systému SUSE LINUX a používání modulu správy napájení programu YaST najdete v kapitole 16 na straně 285.

### 13.1.3 Změny nastavení systému

V mobilním prostředí se systém často potřebuje přizpůsobovat novým podmínkám. Mnoho služeb závisí na pracovním prostředí a při změnách je nutné přenastavit jejich klienty. SUSE LINUX dokáže obstarat i takové situace.



*Obrázek 13.1: Integrace notebooku do sítě*

Služby měněné přenášením mezi domácí a podnikovou sítí mohou být následující:

**Nastavení sítě** Nastavení sítě obsahuje IP adresu, jmenné služby, připojení k internetu a připojení k dalším sítím.

**Tisk** V závislosti na síti, do které je notebook nastaven, musí být správně nastavená databáze tiskáren a příslušný tiskový server.

**Email a proxy** Musí být nastaven správný seznam serverů.

**Nastavení grafického prostředí** Pokud např. v zaměstnání připojujete notebook k externímu monitoru, musí být dostupné příslušné nastavení v grafickém prostředí.

SUSE LINUX nabízí dvě možnosti, které lze kombinovat, jak notebook přizpůsobit aktuálnímu prostředí.

**SCPM** SCPM (*system configuration profile management*) umožňuje jednotlivá nastavení obsahující konfigurační soubory ukládat do tzv. *profilů*. Profily lze vytvářet pro různé situace. Jsou užitečné při potřebě změn prostředí (domácí síť, podniková síť). Mezi profily se lze jednoduše přepínat. Informace o SCPM najdete v kapitole 15 na straně 277. Přepínání mezi profily v KDE umožňuje applet Profile Chooser. Aplikace vyžaduje před přepnutím profilu zadání hesla uživatele root.

**SLP** SLP (*service location protocol*) zjednodušuje připojení notebooku do existující sítě. Bez SLP je obvykle potřeba znát pro nastavení řadu údajů. V případě SLP jsou všechny potřebné informace vysílány po síti a aplikace si vše nastaví samy automaticky. SLP lze používat také pro instalaci systému. Podrobnější informace o SLP najdete v části 23 na straně 405.

Význam SCPm spočívá v povolení a správě snadno reprodukovatelných systémových podmínek. SLP významně usnadňuje síťové nastavení.

### 13.1.4 Software

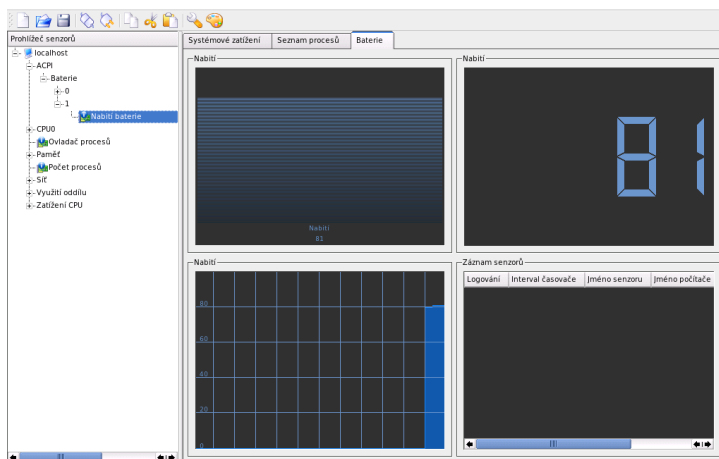
V oblasti mobilních zařízení je řada oblastí, které vyžadují zvláštní aplikace: monitorování systému (především stav baterií), synchronizace dat, bezdrátová komunikace v periferiemi nebo bezdrátové připojení k internetu. V této sekci najdete informace o nejdůležitějších aplikacích.

#### Monitorování systému

V systému SUSE LINUX najdete dva monitorovací nástroje prostředí KDE. Stav nabití baterií a status napájení zobrazuje applet KPowerSave na hlavním panelu. Komplexní systém monitorování poskytuje KSysguard. Pokud používáte prostředí GNOME, budete používat GNOME ACPI (jako applet) a Monitor systému.

**KPowerSave** KPowerSave je applet, který zobrazuje stav baterií a status napájení na hlavním panelu v prostředí KDE. V případě připojení do sítě je zobrazena malá zástrčka. Po přechodu na napájení z baterie se objeví ikonka baterie. Z kontextové nabídky aplikace lze po zadání hesla uživatele root otevřít modul správy napájení programu YaST. V tomto modulu můžete nastavit chování správy napájení. Informace o modulu správy napájení programu YaST najdete v kapitole 16 na straně 285.

**KSysguard** KSysguard je nezávislá aplikace pro monitorování systému. Monitoruje ACPI (stav baterie), zatížení procesoru, síťový provoz, rozdělení disku a využití paměti. Může monitorovat a zobrazovat libovolné systémové procesy. Způsob zobrazení a filtrování lze upravit. Lze monitorovat různé parametry v několika stránkách nebo přes síť sbírat data z několika počítačů současně. KSysguard může běžet jako démon na počítači bez prostředí KDE. Více informací o tomto programu najdete v nápovědě.



*Obrázek 13.2: Monitorování stavu baterií pomocí KSysguard*

## Synchronizace dat

Pokud střídavě pracujete na notebooku bez síťového připojení a na pracovní stanici v síti, je nezbytné nutné zajistit, abyste na obou počítačích měli všechna aktuální data. To zahrnuje poštovní složky, adresáře i jednotlivé soubory. Řešením je synchronizace dat, kterou můžete provádět následujícími způsoby:

**Synchronizace emailů** Používejte pro ukládání zpráv v podnikové síti IMAP účty. Ke zprávám lze přistupovat libovolným klientem, který umí pracovat také s odpojeným IMAP účtem jako např. Mozilla Thunderbird Mail, Evolution nebo KMail. Klienta je nutné nastavit tak, aby byla vždy použita shodná složka Odeslané. Tím zajistíte, že synchronizace proběhne bez

problémů, a budete mít vždy aktuální data a zprávy budou mít správný status. Abyste vždy měli přehled o neodeslaných zprávách, používejte místo systémových MTA jako postfix nebo sendmail SMTP služby implementované ve svém poštovním klientovi.

**Synchronizace souborů a adresářů** Pro synchronizaci dat mezi pracovní stanicí a notebookem je k dispozici celá řada aplikací. Podrobnější informace najdete v kapitole 31 na straně 499.

## Bezdrátová komunikace

Stejně jako doma nebo v kanceláři lze zapojit počítač do klasické sítě, lze notebooky propojit s ostatními notebooky, periferiemi, mobilními telefony nebo kapesními počítači pomocí bezdrátové technologie. Linux tři typy bezdrátové komunikace:

**WLAN** WLAN je jako bezdrátová technologie s největším dosahem jediná vhodná volba pro budování rozsáhlých sítí. Lze ji použít k propojování nezávislých stanic nebo k připojení k internetu. Zařízení nazývané přístupový bod může hrát úlohu základní stanice sítě a zprostředkovávat přístup do internetu. Mobilní uživatel se může mezi přístupovými body přepínat a přistupovat do sítě přes bod, který mu umožňuje nejkvalitnější přístup. Stejně jako u mobilních telefonů je možný přístup kdykoliv. Podrobnější informace najdete v části 17.1 na straně 310.

**Bluetooth** Bluetooth je bezdrátová technologie s kratším dosahem. Obvykle je používána pro komunikaci mezi počítači a kapesními počítači nebo také místo IrDA pro komunikaci s mobilními telefony. Touto technologií lze také propojovat více počítačů bez nutnosti dohledu na jednotlivá zařízení. Bluetooth je také používána u bezdrátových myší a klávesnic. Bližší informace o Bluetooth najdete v části 17.2 na straně 317.

**IrDA** IrDA je bezdrátová technologie s nejkratším dosahem. Obě komunikační strany musí být v dohledu. Překážky jako zdi vedou k nefunkčnosti spojení. Jedním z využití IrDA je přenos souborů z mobilního telefonu do notebooku a naopak. Propojena pomocí IrDA je pouze část mezi notebookem a telefonem. Přenos na delší vzdálenosti je již veden mobilní sítí. Dalším obvyklým využitím IrDA je bezdrátové odesílání tiskových úloh na tiskárnu. Více informací o IrDA najdete v části 17.3 na straně 327.



### 13.1.5 Ochrana dat

V ideálním případě by měla být data na notebooku chráněna několika způsoby. Možné oblasti zajištění jsou následující:

**Ochrana proti krádeži** Pokud je to možné, můžete počítač zajistit fyzicky. V obchodech je dnes k dispozici řada různých typů zabezpečení.

**Bezpečnost dat v systému** Důležitá data by neměla být šifrovaná jen během přenosu, ale také na disku. Tím zajistíte, že v případě krádeže nedojde k jejich zneužití. Popis vytváření šifrovaného souborového systému najdete v části 34.3 na straně 564.

**Síťová bezpečnost** Každý přenos dat by měl být bezpečný. Základní informace o Linuxu a sítích najdete v části 34.4 na straně 567. O bezpečnosti v bezdrátových sítích pojednává kapitola 17 na straně 309.

## 13.2 Mobilní hardware

SUSE LINUX podporuje automatickou detekci mobilních disků připojených přes firewire (IEEE 1394) nebo USB. Termín mobilní disky zde zahrnuje všechny typy firewire nebo USB disků, flash disků a digitálních kamer. Všechna tato zařízení jsou po připojení automaticky detekována systémem hotplug, subfs a submount zajišťují automatické připojení zařízení do souborového systému. Ruční připojování a odpojování zařízení již není používáno. Po ukončení programu, který přistupovat k zařízení, stačí disk jednoduše odpojit od počítače.

**Externí disky (USB a Firewire)** Po rozpoznání systémem jsou externí disky dostupné v seznamu připojených zařízení po kliknutí na ikonu 'Můj počítač' (KDE) nebo 'Počítač' (GNOME). Na externím disku můžete libovolně vytvářet, přejmenovávat a mazat adresáře i soubory. Disk lze přejmenovat kliknutím na ikonu disku pravým tlačítkem a volbou příslušné 'Přejmenovat'. Nové jméno bude dostupné pouze ve správci souborů, skutečné jméno zařízení nastavené systémem jako např. `/media/usb-xxx` nebo `/media/ieee1394-xxx` zůstane nezměněno.

**USB flash disky** K flash diskům systém přistupuje jako k externím diskům. Přejmenovat je lze ve správci souborů.

### Digitální fotoaparáty (USB a Firewire)

Digitální fotoaparáty rozpoznané systémem jsou často ve správci souborů zobrazeny jako externí disky. KDE umožňuje přístup k obrázkům uloženým ve fotoaparátu zadáním URL `camera: /`. Obrázky lze upravovat například pomocí programu digikam nebo GIMP. V prostředí GNOME lze použít Nautilus. Jednoduchý nástroj pro správu a úpravu obrázků je GThumb. Pro pokročilé úpravy je určen GIMP. Programy digikam a GIMP a Nautilus jsou popsány v uživatelské příručce, kde je digitální fotografii věnována celá kapitola.

### Důležité

#### Bezpečnost mobilních diskových zařízení

Výměnné pevné disky a flash disky jsou stejně jako notebooky častým cílem zlodějů. Aby nedošlo k jejich zneužití, doporučujeme na nich vytvořit šifrovaný souborový systém viz. 34.3 na straně 564.

### Důležité

## 13.3 Mobilní telefony a kapesní počítače

Pracovní stanice a notebooky mohou komunikovat s mobilními telefony pomocí IrDA nebo Bluetooth. Některé modely podporují oba protokoly, jiné pouze jeden. Použití těchto protokolů je popsáno v 13.1.4 na straně 264. Nastavení nutná na straně mobilního telefonu najdete v manuálu svého telefonu. Nastavení na straně Linuxu je popsáno v částech 17.2 na straně 317 a 17.3 na straně 327.

Podporu pro synchronizaci s kapesními počítači Palm obsahují programy Evolution a Kontact. Připojení zařízení je v obou případech prováděno pomocí průvodce. Po nastavení Palm Pilota je nutné zadat typ synchronizovaných dat /adresy, schůzky, atd.). Obě aplikace jsou popsány v uživatelské příručce.

Program KPilot je součástí aplikace Kontact nebo jako nezávislý nástroj. Pro synchronizaci kontaktů lze použít také program KitchenSync.

Další informace o aplikacích Evolution a Kontact najdete v uživatelské příručce.

## 13.4 Další informace

Hlavní zdroj informací i Linuxu na mobilních zařízeních najdete na stránce <http://tuxmobil.org/>. Podrobnosti o notebookích, kapesních počítačích, mobilních telefonech a dalších zařízeních jsou roztržiděné do jednotlivých podsekcí.

Podobnou stránku jako <http://tuxmobil.org/> věnovanou pouze notebookům a kapesním počítačům najdete na adrese <http://www.linux-on-laptops.com/>.

SUSE spravuje emailovou konferenci věnovanou notebookům. Základní informace najdete na stránce <http://lists.suse.com/archive/suse-laptop/>. V této konferenci uživatelé a vývojáři probírají problematiku systému SUSE LINUX a mobilních počítačů. Konference je vedena v německém jazyce, ale běžně jsou zodpovídány také dotazy v angličtině.

V případě problémů se správou napájení na notebooku se systémem SUSE LINUX doporučujeme nejdřív prostudovat soubor `README` v adresáři `/usr/share/doc/packages/powersave`. Tento soubor obsahuje nejnovější informace vývojářů a testerů, které již nebylo možné zařadit do oficiální dokumentace.



# Linux a notebooky

U notebooků se setkáváme s řadou hardwarových zvláštností, jako je řízení spotřeby infračervený port (IrDA), karty PCMCIA a Bluetooth. Tyto komponenty nacházíme příležitostně i u stolních počítačů a protože se funkčně neliší od provedení v notebooku, bude jejich použití a konfigurace popsána společně v této kapitole.

14.1	Hardware . . . . .	270
14.2	Software . . . . .	270
14.3	Konfigurace . . . . .	271
14.4	Problémové notebooky . . . . .	272
14.5	Další informace . . . . .	275

## 14.1 Hardware

Zkratka PCMCIA znamená *Personal Computer Memory Card International Association* a používá se všeobecně pro hardware a odpovídající software tzv. karet PCMCIA, u kterých rozlišujeme dva základní typy:

### Klasické karty PCMCIA (též PC-karty):

To je zatím nejběžnější typ, kde se používá 16 bitová sběrnice. Jsou dnes již cenově dostupné a obvykle fungují bez problémů a mají stabilní podporu.

**Karty CardBus:** Jedná se o nový standard. Používají 32 bitovou sběrnici a jsou proto rychlejší, také ovšem dražší. Protože je však přenos dat často omezen i druhou stranou spojení, nemusí se náklady na ně vyplatit. Existuje zatím několik ovladačů na tyto karty, v závislosti na použitém řadiči PCMCIA však dosud nemusí být zcela stabilní.

Pokud je služba PCMCIA aktivní, dozvíte se o typech Linuxem rozpoznaných karet příkazem `cardctl ident`. Seznam podporovaných karet naleznete v souboru `SUPPORTED.CARDS` v adresáři `/usr/share/doc/packages/pcmcia`. Zde se nachází i aktualizovaná verze `PCMCIA-HOWTO`.

Další důležitou komponentou je řadič PCMCIA, nazývaný též PCMCIA/CardBus-bridge. Ten vytváří spojení mezi kartou a sběrnici PCI, ve starších počítačích sběrnici ISA. Tyto řadiče jsou téměř vždy kompatibilní s čipem Intel i82365. Typ řadiče lze zjistit příkazem `pcic_probe`. Jedná-li se o zařízení PCI, podá nám zajímavé informace i příkaz `lspci -vt`.

## 14.2 Software

Všechny potřebné ovladače a programy, pokud již nejsou integrovány v jádru, obsahuje `pcmcia`. Základ tvoří moduly `pcmcia_core`, `i82365` (nebo `yenta_socket`) a ds. Tyto moduly se normálně spouštějí automaticky při startu systému. Inicializují řadič PCMCIA a podporují základní funkce.

### 14.2.1 Cardmanager

Aby se karty PCMCIA daly vyměňovat za běhu, musí zde být démon, který dohlíží na aktivity v zásuvkách PCMCIA. To provádí program *Cardmanager* nebo

Hotplug systém jádra. Pokud je karta zasunuta, rozpozná *Cardmanager* resp. hotplug její typ a funkci a zavede příslušný modul. Pomocí příkazu `lsmod` zjistíme, který modul byl zaveden. Po úspěšném zavedení všech modulů se spustí zvolené instalační skripty, které například vybudují síťové spojení. Pokud se karta opět vysune, *Cardmanager*, hotplug pomocí stejných skriptů řádně ukončí aktivitu karty. Poté se nepotřebné moduly opět odstraní.

Teoreticky se tedy dá karta PCMCIA kdykoli vyjmout. To platí velmi dobře pro karty síťové, modemové a ISDN, pokud přes ně zrovna neprobíhá aktivní komunikace. Potíže však nastávají u souborových systémů, připojených přes kartu PCMCIA, např. jako jsou oddíly externích médií nebo jako adresáře NFS. Zde je třeba nejprve zajistit, aby tato zařízení byla synchronizována (tj. byla jim vyprázdněna vyrovnávací paměť) a pak řádně odpojena. Linux totiž nemůže předvídat, kdy za běhu kartu vytáhneme, a proto je potřeba mu to s předstihem oznámit. Pomoci nám může příkaz `cardctl eject`. Ten deaktivuje všechny karty PCMCIA v notebooku.

## 14.3 Konfigurace

PCMCIA je možné ručně spustit za běhu příkazem `rcpcmcia start`.

Protože výběr správných ovladačových modulů pro danou kartu zajistí *Cardmanager*, resp. hotplug -- další nastavení, týkající se vlastností hardwaru, již není zapotřebí.

Další konfiguraci `pcmcia` můžete provést v `/etc/sysconfig/pcmcia`, kde se nachází pár voleb s podrobnou nápovědou.

### 14.3.1 Ethernet, bezdrát (wireless) a Token Ring

Síťové připojení na Ethernet nebo Token Ring nakonfigurujeme pohodlně pomocí instalátoru YaST. Provádí se stejně, jako konfigurace klasické síťové karty, ale je třeba zde uvést, že se jedná o PCMCIA.

### 14.3.2 ISDN

Karty PCMCIA typu ISDN se konfiguruji podobně jako ostatní karty. Tzv. modemy ISDN existují i v provedení PCMCIA. Jsou to modemové nebo multifunkční karty s dodatečným adaptérem pro připojení k ISDN a zachází se s nimi jako s modemem.

### 14.3.3 Modem

U modemových karet PCMCIA obvykle nepotřebujeme nastavovat nic navíc. Jakmile zasuneme modemovou kartu, je použitelná jako zařízení `/dev/modem`. Konfigurace tohoto zařízení provádí také YaST.

### 14.3.4 SCSI a IDE

Odpovídající moduly ovladačů zavede Cardmanager. Jakmile zasuneme kartu PCMCIA typu SCSI nebo IDE, jsou připojená zařízení použitelná. Rovněž se pro ně dynamicky určí jméno zařízení (*device name*).

Informace o podporovaných kartách PCMCIA pro SCSI a IDE najdeme v adresářích `/proc/scsi` a `/proc/ide`.

#### Důležité

Externí disky, mechaniky CD a podobná zařízení je třeba zapnout, než k nim připojenou kartu PCMCIA zasuneme do počítače. Nezapomeňte přitom na správné kabelové zakončení u zařízení SCSI. *Pozor:* Než vysunete kartu PCMCIA pro SCSI nebo IDE, je třeba odpojit jejich souborové systémy. Pokud na to zapomenete, dostanete se na ně příště pouze až po restartu systému.

#### Důležité

Linux se dá také instalovat celý na takovémto externím zařízení, pouze startování je pak náročnější. Tehdy je zapotřebí použít *startovací disketu*, obsahující jádro a startovací ramdisk (`initrd`).

Soubor `initrd` obsahuje virtuální souborový systém, na kterém jsou všechny potřebné moduly PCMCIA a programy. Startovací disketa pro SUSE LINUX resp. její obraz jsou tak vytvořeny, a proto z nich můžete startovat externí instalaci. Zavadět podporu PCMCIA při každém startu ručně je však nepohodlné. Proto si pokročilí uživatelé vytvoří startovací disketu na míru podle PCMCIA--HOWTO v odst. 5.3 Startování ze zařízení PCMCIA.

## 14.4 Problémové notebooky

Některé notebooky mají potíže s určitými kartami PCMCIA, z čehož většinu lze odstranit pouhou důsledností. Nejprve je třeba zjistit, zda se problém týká spíše



karty nebo základního systému PCMCIA. K tomu stačí nejprve spustit počítač bez zasunuté karty. Pokud vše běží, pak teprve zasuneme kartu. Všechna důležitá hlášení najdeme v souboru `/var/log/messages`. Průběžné pozorování těchto informací umožňuje příkaz

```
tail -f /var/log/messages
```

Tímto způsobem lze určit typ chyby.

### 14.4.1 Základní systém PCMCIA nefunguje

Pokud systém přestane komunikovat již při startu po hlášení *PCMCIA: Starting services:* nebo se chová podivně, zkuste potlačit spuštění PCMCIA při příštím startu zadáním *NOPCMCIA=yes* ze startovacího promptu zavaděče. K dalšímu vymezení problému je potřeba ručně spustit tři základní moduly. K tomu slouží příkazy `modprobe -t pcmcia_core`, `modprobe -t pcmcia-external i82365` u externích PCMCIA, resp. `modprobe -t pcmcia yenta_socket` u jaderného PCMCIA `modprobe -t ds`. Kritické moduly jsou první a druhý.

Objeví-li se problém při zavedení modulu `pcmcia_core`, pomůže nám `pcmcia_core`. Volby, které jsou tam popsány, vyzkoušíme nejprve pomocí příkazu `modprobe`. Jako příklad můžeme odpojit podporu APM pro modul PCMCIA, protože s ním mohou být občas problémy. Na to použijete volbu `do_apm=0`, která APM deaktivuje:

```
modprobe -t pcmciacore do_apm=0
```

V případě úspěchu zapíšete do proměnné *PCMCIA\_CORE\_OPTS* v souboru `/etc/sysconfig/pcmcia`:

```
PCMCIA_CORE_OPTS="do_apm=0"
```

Od této chvíle již APM nepracuje a pokud ho potřebujete obnovit, musíte zadat `do_apm=1`.

Rovněž může v ojedinělých případech dojít ke konfliktu některých komponent při testování volného rozsahu IO. To lze obejít volbou `probe_io=0`.

V případě více voleb použijeme k jejich oddělení mezery:

```
PCMCIA_CORE_OPTS="do_apm=0 probe_io=0"
```

Pokud se chyba objevuje při zavádění modulu `i82365`, pomůže nám `i82365`. Tato chyba je následkem konfliktu zdrojů *resource conflict*, tj. dvě zařízení si nárokují stejné přerušování, IO port nebo paměťový rozsah. Modul `i82365` zdroje sice kontroluje, může však naneštěstí přestat reagovat právě při tom. Tak se stává, že u některých počítačů vede test IRQ 12 (zařízení typu PS/2) k zablokování myši,

případně i klávesnice. V tomto případě pomáhá parametr *irq\_list=seznam\_pripustnych\_IRQ*. Seznam by měl obsahovat všechny IRQ, které se smějí použít. Napíšeme tedy například

```
modprobe i82365 irq_list=5,7,9,10
```

nebo umístíme natrvalo do souboru */etc/rc.config* řádku:

```
PCMCIA_PCIC_OPTS="irq_list=5,7,9,10"
```

Dále jsou zde soubory */etc/pcmcia/config a /etc/pcmcia/config.opts*, které používá Cardmanager. Nastavení v těchto souborech se použijí pro zavádění modulů ovladačů karet PCMCIA. V souboru */etc/pcmcia/config.opts* lze rovněž přiřadit nebo zakázat všechny IRQ, IO porty a paměťové rozsahy. Rozdíl oproti volbě *irq\_list* je ten, že zde zakázané zdroje sice pak nepoužije karta PCMCIA, ale budou stále ještě kontrolovány modulem *i82365*.

## 14.4.2 Karta PCMCIA nefunguje správně

Zde jsou tři možnosti chyby: karta nebyla správně detekována, používá nedostupné zdroje nebo se nechová dle očekávání.

**ádná reakce po vložení karty** Pokud systém po vložení karty nereaguje a nepomůže ani ruční zadání příkazu *cardctlinsert*, může jít o špatnou alokaci přerušení PCI zařízení. Pokud jde o tento problém, mohou mít problémy i jiná zařízení např. síťová karta. V takovém případě může pomoci parametr jádra *pci=noacpi*.

**Karta nebyla detekována** Pokud nebyla karta detekována, najdete v souboru */var/log/messages* hlášení *unsupported Card in Slot x*. Toto hlášení znamená, že správce karet nebyl schopný k vaší kartě přiřadit žádný ovladač. Pro toto přiřazení je potřebný soubor */etc/pcmcia/config* popř. */etc/pcmcia/\*.conf*. Databázi ovladačů lze snadno rozšířit existující položky, kterou použijete jako šablonu. Podrobnosti o své kartě zjistíte zadáním příkazu *cardctlident*. Další informace o tomto tématu najdete v PCMCIA HOWTO (sekce 6) a manuálových stránkách *pcmcia*. Po editaci souborů obnovte přiřazení ovladačů příkazem *rcpcmciareload*.

**Ovladač se nezavedl** Jedním z důvodů této situace může být nekorektní záznam pro přiřazení ovladače v databázi. K tomu může dojít např. tehdy, pokud výrobce použije jinou čipovou sadu u již vyráběného modelu. Některé karty pak mohou pracovat pouze s jiným než předzvoleným ovladačem. V takovém případě budete potřebovat podrobné informace o své

kartě. Někdy je užitečné požádat o pomoc v některé linuxové emailové konferenci nebo si vyžádat rozšířenou podporu.

Pro CardBus karty musí být v souboru `/etc/sysconfig/hotplug` nastavena proměnná `HOTPLUG_DEBUG=yes`.

Další možnou příčinou je konflikt při přidělení systémových prostředků. U většiny karet je jedno, s jakým IRQ, I/O portem a rozsahem paměti pracují, ale existují výjimky. V takovém případě testujte systém vždy pouze s jednou zapojenou kartou a ostatní vyjměte (např. zvukovou kartu, IrDA modem, tiskárnu...). Přidělení systémových prostředků můžete sledovat jako uživatel `root` pomocí příkazu `lsdev`. Z výstupu můžete zjistit, jaké prostředky jsou používány. Použití jednoho IRQ několika PCI zařízeními je obvykle bez problémů.

Řešením je nastavení vhodných parametrů ovladače. Seznam parametrů získáte příkazem `modinfo <jmeno_ovladace>`. Pro většinu ovladačů jsou také k dispozici manuálové stránky.

Po nalezení vhodných parametrů proveďte nastavení systémových zdrojů v souboru `/etc/pcmcia/config.opts`. Například pro modul `pcnet_cs` používající IRQ 5 zadejte následující:

```
module pcnet_cs opts irq_list=5
```

**Chybné rozhraní** Pokud dojde k chybnému nastavení rozhraní, překontrolujte nastavení rozhraní a jméno pomocí příkazu `getcfg`. V souboru `/etc/sysconfig/network/config` nastavte proměnnou `DEBUG` a v souboru `/etc/sysconfig/hotplug` proměnnou `HOTPLUG_DEBUG` na `yes`. Pokud tento postup nepomůže, zadejte do skriptu vykonávaného sprvcem karet nebo hotplugem řádku `set-vx`. Po tomto nastavení bude výstup skriptu zaznamenáván do systémového logu. Pokud naleznete kritickou sekci skriptu, otestujte příslušné příkazy v terminálu.

## 14.5 Další informace

Podrobnější informace o používání notebooků v Linuxu naleznete na <http://linux-laptop.net>. Velmi dobrým zdrojem informací o Linuxu na mobilních počítačích je také <http://mobiliX.org/> (MobiliX -- Mobile Computers and Unix). Zde naleznete, kromě jiného, také Laptop-Howto a IrDA-Howto.



# Správa profilů

V této kapitole je popsán SCPM (system configuration profile management). S pomocí SCPM můžete svůj počítač přizpůsobit různým pracovním prostředím nebo odlišným hardwarovým konfiguracím. SCPM spravuje pro různé situace skupinu systémových souborů. Díky tomu umožňuje rychlé přepnutí mezi systémovými profily bez nutnosti jejich ručního přenastavení.

15.1	Základní terminologie . . . . .	278
15.2	Nastavení SCPM . . . . .	279
15.3	Volba profilu při startu . . . . .	283
15.4	Problémy a jejich řešení . . . . .	283
15.5	Další informace . . . . .	284

Jsou situace, kde je nezbytné změnit systémovou konfiguraci. Pokud často provozujete svůj počítač v prostředích, kde potřebujete různá nastavení systému, možná by se vám hodilo uložit si tato nastavení a obnovit je později, kdykoliv je to potřeba. To to je typická situace například pro uživatele notebooků, kteří pracují na různých místech. Také si lze představit stolní počítač, který chcete dočasně provozovat s jinou konfigurací. V takových případech byste rádi měli záložní mechanismus, který uloží současná systémová konfigurační data a uloží je do profilu. Tímto způsobem lze potom kdykoliv tuto konfiguraci obnovit.

Hlavní doménou SCPM je nastavit síť na notebookech. Předpokládejme tedy, že máte notebook a chcete jej připojit ke své domácí i firemní síti a používat jej nezávisle, když jste na cestách. Toto obvykle vyžaduje nakonfigurovat systém tak, aby zapadl do různých sítí. Například potřebujete DHCP klienta v kanceláři a pevnou IP adresu doma. Dále máte třeba v kanceláři spuštěné služby jako xntpd, NIS klienta, ale doma pouze automounter, ale žádná z těchto služeb není potřeba, pokud cestujete. Pro tyto případy vám SCPM pomůže zvládnout rozdílné konfigurace a jednoduše se mezi nimi přepínat.

SCPM toho ale umí daleko víc. Je velmi konfigurovatelný; zvládne skoro všechny možné scénáře, kdy je potřeba uložit a obnovit data v různých verzích. Dokonce jej lze použít pro spouštění skriptů v závislosti na profilech, mezi kterými je přepínáno. Více informací najdete v příslušných info stránkách.

## 15.1 Základní terminologie

Dřív než začnete používat SCPM, seznamte se prosím se základními pojmy používanými v modulu programu YaST.

- Pod *systémovou konfigurací nebo nastavením* rozumíme souhrn nastavení počítače. Všechna důležitá nastavení jako např. připojení disků, nastavení sítě, časové zóny nebo rozložení klávesnice.
- *Profil* nebo také *konfigurační profil* je nastavení systému, které bylo uloženo pod určitým jménem.
- *Aktivním profilem* rozumíme profil, který je zrovna používán. Neznamená to však, že je systém nastaven právě podle tohoto profilu, protože každý uživatel má možnost si svůj systém z určité části poupravit.
- *Zdroje* jsou v pojetí SCPM všechny části spravované systémovou konfigurací. Může jít o soubory nebo odkazy. Pojem zahrnuje také systémové služby, které v jednom profilu běží a v jiném jsou vypnuté.

- Zdroje jsou organizovány do *Skupiny zdrojů*. Tyto skupiny jsou sestaveny podle určitých logických kritérií. Znamená to, že s určitou službou obsahují také její konfigurační soubory. To umožňuje spravovat zdroje bez znalosti konfiguračních souborů jednotlivých služeb.

## 15.2 Nastavení SCPM

V zásadě jsou dostupné dvě rozhraní pro nastavení SCPM. Balíček `scpm` obsahuje rozhraní pro příkazovou řádku. ‘Správce profilů’ programu YaST je určen pro grafické prostředí. Obě rozhraní mají stejnou funkčnost, ale znalost rozhraní příkazové řádky vám výrazně usnadní pochopení modulu programu YaST. Následující popis bude proto zaměřen především na textové prostředí.

### 15.2.1 Spuštění SCPM a definice skupin zdrojů

SCPM musíte nejdřív aktivovat. To provedete příkazem `scpmenable`. Při prvním spuštění dochází k inicializaci SCPM. Inicializace je časově náročnější a může zabrat několik sekund. SCPM deaktivujete a tím zabráníte nechtěnému přepnutí profilů příkazem `scpm disable`.

Standardně SCPM obsahuje nastavení pro síť, tisk a grafické prostředí. Před použitím odpovídajícího nastavení musíte nejdřív aktivovat příslušné skupiny zdrojů. Dostupné skupiny zobrazíte příkazem:

```
scpm list_groups
```

Pokud si chcete nechat vypsat pouze aktivní skupiny, zadejte příkaz:

```
scpm list_groups -a
```

Uvedené příkazy musíte vykonávat jako uživatel `root`.

```
scpm list_groups -a
```

```
nis           Network Information Service client
mail          Mail subsystem
ntpd          Network Time Protocol daemon
xf86          X-Server settings
autofs        Automounter service
network       Basic network settings
printer       Printer settings
```

Skupiny aktivujete popř. deaktivujete příkazem:

```
scpm activate_group JMENO
```

popř.

```
scpm deactivate_group JMENO.
```

Část JMENO nahradíte jménem zvolené skupiny. Skupiny lze spravovat také prostřednictvím správce profilů programu YaST.

## 15.2.2 Vytváření a přepínání profilů

Po aktivaci SCPM se spustí profil default. Seznam všech dostupných profilů získáte příkazem `scpm list`. Pouze jeden ze všech dostupných profilů může být aktivní. Jméno aktivního profilu získáte příkazem `scpm active`. Profil default je základní profil, ze kterého jsou všechny ostatní odvozeny. Před spuštěním správy profilů proto nastavte všechna nastavení, která chcete mít v profilech dostupná. Příkazem `scpm reload` uložíte všechny změny na systému do aktivního profilu. Profil default si pak můžete ponechat nebo ho smazat.

Jsou dvě možnosti, jak vytvořit nový profil. Nový profil (zde work) např. odvozený od profilu např. default vytvoříte příkazem `scpm copy default work`. Příkazem `scpm switch work` se do nového profilu můžete přepnout a provést další nastavení. V některých případech je však výhodné vytvořit profil z již existujícího právě používaného nastavení. To provedete pomocí příkazu `scpm add work`. Po zadání tohoto příkazu budete mít aktuální nastavení systému uložené v profilu work a ten bude označen jako aktivní; \dasheisst že příkaz `scpm reload` uloží změny do profilu work.

Profily lze samozřejmě také přejmenovávat a mazat. K tomu použijte příkazy `scpm rename x y` a `scpm delete x`. K přejmenování např. work na prace použijte příkaz `scpm rename work prace`. Aktivní profil nelze smazat.

Další příkazy:

**scpm list** zobrazení seznamu dostupných profilů

**scpm active** zobrazení aktivního profilu

**scpm add Jmeno** uložení aktuálního nastavení systému do profilu a nastavení tohoto profilu jako aktivního

**scpm copy Jmeno NoveJmeno** kopírování profilu



**scpm rename Jmeno NoveJmeno** přejmenování profilu

**scpm delete Jmeno** smazání profilu

Poznámka k modulu programu YaST: Při prvním spuštění máte k dispozici pouze nabídku 'Volby'. Až po spuštění správy profilů, získáte možnost vybrat si jeden z předdefinovaných profilů, který se uloží jako profil `default`. Až pak získáte další možnosti úpravy.

### 15.2.3 Přepínání mezi profily

Pokud se chcete přepnout do jiného profilu použijte příkaz (zde `work`):

```
scpm switch work
```

Tímto příkazem vypnete aktivní profil a nastavíte nový. Před nastavením nového profilu můžete také právě aktivní profil zcela deaktivovat.

Při této změně SCPM porovná aktuální nastavení s novým profilem. Pak musí určit, které služby se budou restartovat a jaké konfigurační souboru bude potřeba načíst. Následně se spustí akce, která se jeví jako částečný systémový restart, kdy se restartují všechny měněné služby, ale zbytek systému funguje dál.

Nyní se spustí tyto akce:

- Systémové služby budou zastaveny.
- Zápis všech změněných zdrojů (např. \konfigurační soubory).
- Systémové služby se (znovu) spustí.

### 15.2.4 Rozšířené nastavení

Ke každému profilu lze napsat krátký popis, který se zobrazí po zadání příkazu `scpm list`. Pro aktivní profil nastavíte popis příkazem:

```
scpm set description "text"
```

Pro neaktivní profil musíte zadat ještě jméno profilu, takže pro profil `work` bude příkaz vypadat takto:

```
scpm set description "text" work
```

Někdy je při vypínání či zapínání profilu nutné vykonat akce ještě po ukončení služeb či před jejich spuštěním. Pro každý profil jsou proto dostupné čtyři programy nebo skripty, které se vykonávají v různých fázích při přepnutí. Tyto body jsou následující:

**prestop** před zastavením služby při ukončení profilu

**poststop** po zastavení služby při ukončení profilu

**prestart** před spuštěním služby při aktivaci profilu

**poststart** po spuštění služby při aktivaci profilu

Přepnutí z profilu `work` na `home` funguje takto:

- Prestop akce profilu `work`
- Zastavení služeb
- Poststop akce profilu `work`
- Změna nastavení
- Prestart akce profilu `home`
- Spuštění služeb
- Poststart akce profilu `home`

Tyto akce lze vykonat příkazem `set`. Použití je takové:

```
scpm set prestop JmenoSouboru  
scpm set poststop JmenoSouboru  
scpm set prestart JmenoSouboru  
nebo
```

```
scpm set poststart JmenoSouboru
```

Všechny tyto příkazy vykonává uživatel `root`.

## Varování

Protože tyto skripty mohou obsahovat citlivé informace o systému, měly by být čitelné pouze pro administrátora systému. Nejvhodnější je tedy nastavit souboru práva na `-rwx----` `root root`. (`chmod 700 JmenoSouboru` a `chown root.root JmenoSouboru`).

## Varování

Všechna nastavení provedená pomocí `set` lze získat příkazem `get`. Například příkaz `scpm get poststart` vypíše jméno poštovního programu nebo krátkou informaci, pokud není nic nastaveno.

Příkazy `set` a `get` lze aplikovat také na profil. K tomu účelu musíte zadat jméno profilu. Například:

```
scpm get prestop JmenoSouboru work
nebo
scpm get prestop work.
```

## 15.3 Volba profilu při startu

Profil při startu systému zvolíte tak, že během zobrazení startovacího seznamu stisknete klávesu **(F4)** a ze seznamu zvolíte požadovaný profil. Po seznamu se lze pohybovat pomocí šipek. Start do zvoleného profilu spustíte stisknutím klávesy **(Enter)**. Zvolený profil je pak použit jako startovací parametr.

## 15.4 Problémy a jejich řešení

SCPM není v současné době stále ještě možné aktualizovat spolu se systémem. problém spočívá ve skutečnosti, že se konfigurační soubory nacházejí na celé řadě míst, kam mechanismus aktualizace nemůže zasahovat. SCPM je však schopné aktualizaci rozpoznat a po jejím provedení vám nahlásí:

Vaše instalace se změnila nebo je neznámá

V takovém případě stačí SCPM reinicializovat příkazem:

```
scpm -f enbale
```

Některé profily však mohou být při aktualizaci zcela ztraceny. V takovém případě není jiná cesta, než je znovu vytvořit.

Za určitých okolností se může stát, že SCPM při pokusu o přepnutí profilu přestane pracovat. K tomuto stavu může dojít např. při nenadálém vypnutí systému. Při spuštění SCPM obdržíte hlášení, že je SCPM zamčen. Tato služba chrání data v databázi SCPM v případě, že dojde k problémům se systémem. V takovém případě smažte soubor příkazem:

```
rm /var/lib/scpm/#LOCK
```

a obnovte SCPM zadáním:

```
scpm -s reload.
```

Pak již budete moci bez problémů pracovat.

### 15.4.1 Změna nastavení skupiny zdrojů

Změna v nastavení skupiny v již inicializovaném SCPM nepředstavuje v zásadě žádný problém. Po změně nebo smazání skupiny pouze musíte zadat příkaz:

```
scpm rebuild
```

Tento příkaz zavede do skupiny nové zdroje a smaže ty, které jste se rozhodli odstranit. Pokud provádíte změny pomocí programu YaST, není výše uvedený příkaz nutný. Programem YaST provedete všechna nutná nastavení a příkazy automaticky.

## 15.5 Další informace

Nejnovější dokumentace je dostupná na infostránkách SCPM, které si můžete prohlédnout např. pomocí programu Konqueror nebo Emacs (`konquerorinfo:scpm`). Na příkazové řádce pomocí příkazu `info` nebo `pinfo`. Informace od vývojářů jsou dostupné v souboru `/usr/share/doc/packages/scpm`.

# Správa napájení

V této kapitole najdete stručný úvod do správy napájení v systému Linux. Popsány jsou oba v současné době používané standardy APM (Advanced Power Management) a ACPI (Advanced Configuration and Power Interface).

16.1	Funkce šetření spotřeby . . . . .	286
16.2	APM . . . . .	287
16.3	ACPI . . . . .	288
16.4	Zastavení disku . . . . .	294
16.5	Balík powersave . . . . .	295
16.6	Modul správy napájení programu YaST . . . . .	304

Narozdíl od APM používaného pouze pro správu napájení, je ACPI nástroj umožňující získávání informací o hardwaru a jeho nastavení. V moderních počítačích je tak například možné nastavit frekvenci procesoru podle situace a dosáhnout tím významné úspory energie, což je velmi užitečné především u mobilních zařízení napájených z baterií.

Všechny technologie správy napájení vyžadují podporu v BIOSu a vhodný hardware. Řada moderních notebooků, pracovních stanic a serverů tyto podmínky splňuje. APM je dnes již používáno jen na starších počítačích. Protože se skládá především z funkcí implementovaných v BIOSu, je závislý na hardwaru. To platí také o ACPI, který je však mnohem komplexnější. Z toho důvodu je nemožné upřednostnit jednu technologii před druhou. Jednoduše otestujte potřebné funkce obou technologií na svém počítači a zvolte tu nejlepší.

---

### Důležité

#### Správa napájení procesorů AMD64

U procesorů AMD64 a 64 bitového jádra je podporován pouze ACPI.

---

Důležité

## 16.1 Funkce šetření spotřeby

Celá řada funkcí, které správa napájení poskytuje, má největší uplatnění v oblasti mobilních počítačů. Nejdůležitější jsou tyto:

**Úsporný režim *standby*** V tomto režimu se pouze vypne displej a u novějších počítačů se sníží příkon procesoru.

#### Uspání do paměti (*suspend to memory*)

V tomto režimu se stav systému uloží *do paměti* a počítač (kromě této paměti) přestane pracovat. Spotřeba je pak nepatrná, takže pak počítač (podle typu) vydrží v tomto režimu pracovat na baterii 12 hodin až několik dní. Tento režim má oproti vypnutí tu výhodu, že je opět pohotový po několika sekundách přesně v tom místě, kde skončil, aniž by bylo potřeba znovu startovat a zavádět potřebné programy. U Linuxu, který *nepotřebuje* být čas od času restartován z důvodu obnovení stability -- jako některé nejmenované systémy -- je tato možnost zvláště zajímavá. U moderních notebooků stačí jen zaklapnout víko, aby přešly do suspendovaného režimu. Opětovným odklopením víka notebook opět ožije.

**Uspání na disk (*hibernation, suspend to disk*)**

V tomto režimu počítač doslova přezimuje období své nečinnosti. Současný stav se nejprve uloží *na disk* a počítač se pak sám vypne. Zpětné probuzení ze zimního spánku do stavu před uspáním pak ovšem trvá mezi 30 až 90 sekundami. The state prior to the suspend is restored. Někteří výrobci nabízejí různé hybridní varianty (např. RediSafe v IBM Thinkpadech). Odpovídající ACPI režim je S4. V Linuxu je *uspání na disk* prováděno rutinami nezávislými na APM a ACPI.

**Kontrola stavu baterií** Velmi užitečné.

**Automatické vypnutí po zastavení systému**

Hodí se i pro stolní počítače. Po zastavení systému *shutdown* se počítač (elektricky) vypne.

**Vypínání disku** Šetří významně spotřebu a u hlučných disků i vaše nervy. Je ovšem třeba brát ohled na editory, které v pravidelných intervalech nemilosrdně budí disk na záložní kopie.

Některé z těchto funkcí podporuje již samotný BIOS. Úsporný režim *standby* a odstavení *suspend to memory* realizují notebooky klávesovou kombinací nebo detekcí zaklapnutého víka. Tyto funkce jsou nezávislé na operačním systému, při vhodném jádru a nainstalovaných balících je však můžeme navíc volat i pomocí linuxových příkazů.

## 16.2 APM

Některé funkce již obsahuje APM BIOS. Uspání a probuzení dokáže aktivovat mnoho notebooků pomocí klávesové kombinace nebo uzavřením víka. K tomu nejsou zapotřebí žádné funkce poskytované operačním systémem.

Podpora APM je přímo součástí standardního jádra a je automaticky aktivována v případě, že při startu je nalezen APM-BIOS a deaktivována podpora ACPI parametrem `acpi=off`. Když chcete vypnout podporu APM při startu, můžete to udělat parametrem `apm=off`. Zda je APM aktivováno, zjistíte velice jednoduše příkazem `cat /proc/apm`. Pokud se zobrazí řádek s různými čísly, pak je vše v pořádku.

Protože se některé implementace BIOSu nedrží platných standardů, dochází k zajímavému chování. Něco je možné obejít parametry při startu systému. Můžete použít např.:

**on/off** Zapnout/vypnout podporu APM

**(no-)allow-ints** Povolit během spouštění funkcí BIOSu přerušeni

**(no-)broken-psr** BIOS má vadnou funkci `GetPowerStatus`

**(no-)realmode-power-off** Procesor se přepne před ukončením chodu do reálného režimu

**(no-)debug** Hlášení APM jsou protokolována v syslogu

**(no-)power-off** Po shutdownu se počítač vypne

**bounce-interval=n** Čas v setinách sekundy, kdy po přijetí výsledku uspání budou další požadavky ignorovány

**idle-period=n** Čas v setinách vteřiny po kterém bude sdělena (ne)aktivita systému.

APM démon (`apmd`) již není používán. Jeho funkce jsou součástí nového démona `powersaved`, který podporuje také ACPI a nastavení frekvence CPU.

## 16.3 ACPI

ACPI je zkratka z *Advanced Configuration and Power Interface*. ACPI umožňuje operačnímu systému nastavit a kontrolovat spotřebu jednotlivých hardwarových součástí. Svou funkcí nahrazuje jak PnP tak APM. Část ACPI zodpovědná za inicializaci hardwaru není v této kapitole popsána.

BIOS poskytuje tabulku obsahující informace o jednotlivých komponentech a metodách přístupu. Tyto informace pak použijte operační systém např. k přiřazení přerušeni či aktivaci nebo deaktivaci tohoto zařízení. Jaké operace může operační systém provést, záleží na implementaci BIOSu. Záznamy ACPI o nalezení a použití tabulky najdete v souboru `/var/log/boot.msg`. Detekované a zavedené ACPI tabulky jsou zapsány do `/var/log/boot.msg`. Více o této problematice najdete v části 16.3.4 na straně 292.



### 16.3.1 ACPI v praxi

Když jádro detekuje při startu ACPI BIOS, ACPI se automaticky aktivuje (a APM deaktivuje). Některé starší počítače važdují pro spuštění ACPI zadání parametru jádra `acpi=on`. Počítač musí podporovat ACPI 2.0 nebo vyšší. Zda se ACPI aktivovalo, zjistíte ze záznamu jádra v souboru `/var/log/boot.msg`.

Zavádění modulů obstarává startovací skript ACPI démona. Pokud se při zavádění některého modulu objeví problémy, je možné ho vyřadit zápisem v souboru `/etc/sysconfig/powersave/common`.

Hlášení modulů, která vám umožní zjistit detekované komponenty, najdete v systémovém záznamu (`/var/log/messages`).

`/proc/acpi` nyní obsahuje řadu souborů s informacemi o stavu systému a možných změnách. Některé funkce se stále vyvíjejí a nejsou stále plně funkční. Podpora řady dalších funkcí je závislá na implementaci výrobce.

Všechny soubory (kromě `dsdt` a `fadt`) lze číst pomocí příkazu `cat`. V řadě souborů lze nastavení měnit, použít můžete např. příkaz `echo`. U nastavení vhodných hodnot pro X Window bude příkaz vypadat takto:

```
echoX ><soubor>.
```

K přístupu k těmto informacím vždy používejte příkaz `powersave`. Nejdůležitější soubory s nastaveními správy napájení jsou:

**`/proc/acpi/info`** Základní informace o ACPI

**`/proc/acpi/alarm`** Doba, kdy má dojít k probuzení. Doba je nastavena pomocí příkazu `echo year-month-day hour:minute:second > /proc/acpi/alarm`. Nastavení je bezpředmětné v případě, že probuzení nefunguje.

**`/proc/acpi/sleep`** Poskytuje informace o možných stavech usnutí. V současné době jsou funkční pouze S1 (standby) a S5 (vypnout, neuklízet): `echo 1 > /proc/acpi/sleep`.

**`/proc/acpi/event`** Zde jsou ukládány záznamy o všech událostech. Ty jsou vykonávány démony 'acpid' nebo 'ospmid'. Pokud k souboru nepřístupuje žádný démon, události lze číst příkazem `cat /proc/acpi/event` (ukončení stisknutím **Ctrl-C**).

**`/proc/acpi/dsdt` a `/proc/acpi/fadt`** tento soubor obsahuje ACPI tabulky DSDT a FADT. Soubor lze číst pomocí `acpidmp`, `acpidisasm` a `dmdecode`.

Příklad: `acpidmp DSDT | acpidisasm`.

**/proc/acpi/ac\_adapter/AC/state** Je připojen AC adaptér?

**/proc/acpi/battery/BAT\*/{alarm,info,state}**

Detailní informace o stavu baterií.

**/proc/acpi/button** Tento adresář obsahuje informace o přepínačích.

**/proc/acpi/fan/FAN/state** Ukazuje aktivitu větráčku. Lze ho také manuálně vypnout/spustit zapsáním 0 (zapnutý) nebo 3 (vypnutý) do tohoto souboru. V případě vysoké teploty může jádro toto nastavení přepsat.

**/proc/acpi/processor/CPU\*/info** Informace o úsporách energie procesoru.

**/proc/acpi/processor/CPU\*/power** Informace o stavu procesoru.

**/proc/acpi/processor/CPU\*/performance**

Zde můžete získat informace nebo nastavit výkon -- využijte Speedstep nebo PowerNow procesoru.

**/proc/acpi/processor/CPU\*/throttling** Zde se dá povolit lineární prbrždění procesoru.

**/proc/acpi/processor/CPU\*/limit** Nastavení limitů při použití omezení výkonu a přibrždění procesoru. Nacházejí se zde jak systémové tak uživatelské limity. Příkazem `echo 1:5 > /proc/acpi/processor/CPU*/limit` předejdete použití stavů P0 nebo T0--T4.

**/proc/acpi/thermal\_zone/** Poddadresáře pro jednotlivé teplotní zóny. termální zóna je oblast s určitými teplotními vlastnostmi, číslem a jménem určeným výrobcem zařízení. Velká část funkcí bohužel není implementována. Nejvhodnější ovládání je stále přímo prostřednictvím BIOSu. Některé z následujících nastavení mohou být pouze teoretické.

**/proc/acpi/thermal\_zone/\*/temperature**

Současná teplota teplotní zóny.

**/proc/acpi/thermal\_zone/\*/state** Stav může být ok, aktivní nebo pasivní chlazení. Vše je ok v případě ovládání větráčku nezávisle na ACPI.

**/proc/acpi/thermal\_zone/\*/cooling\_mode**

Volba výchozího chlazení v případě nasazení kontroly ACPI. Může být aktivní (méně úsporné, ale výkonnější) nebo pasivní (méně výkonné, ale úsporné).

**/proc/acpi/thermal\_zone/\*/trip\_points**

Nastavení teploty pro pasivní nebo aktivní chlazení, uspání nebo bezpečnostní vypnutí.

**/proc/acpi/thermal\_zone/\*/polling\_frequency**

Hodnota v `temperature` není automaticky obnovována se změnou teploty, přepněte na 'polling mode'. Příkaz `echo X > /proc/acpi/thermal_zone/*/polling_frequency` zapíše aktuální hodnotu každých X second. Nastavením X=0 polling deaktivujete.

Žádné z těchto nastavení není nutné provádět ručně. Použít můžete buď přímo Powersave démona (`powersaved`) nebo některou z aplikací jako `powersave`, `kpowersave` nebo `wmpowersave`. Více informací najdete v části 16.3.3 na následující straně. Protože `powersaved` obsahuje všechny funkce staršího démona `acpid`, není již démon `acpid` potřebný.

## 16.3.2 Nastavení výkonu CPU

V případě procesoru lze snížit spotřebu energie třemi různými způsoby a v závislosti na operačním režimu lze tyto metody kombinovat. Nižší spotřeba vede k nižšímu zahřívání procesoru a méně častému spouštění větráčků.

**Frekvence a napětí** Technologie nastavení frekvence a napětí `PowerNow!` a `Speedstep` byly navrženy společnostmi AMD a Intel. Tyto technologie jsou implementována také v procesorech jiných výrobců. Současné snížení frekvence a napětí vede k více jak lineárním úsporám energie, což znamená, že při snížení frekvence na polovinu, je spotřeba energie méně než poloviční. Technologie jsou závislé na APM nebo ACPI a vyžadují pro nastavení frekvence příslušného démona. Nastavení lze provést v adresáři `/sys/devices/system/cpu/cpu*/cpufreq/`.

**Přiškrcení** Pomocí této technologie lze přenastavit procento signálů časovače pro CPU. V případě 25% přiškrcení je vynechán každý čtvrtý impuls a k procesoru se dostane pouze 87.5% obvyklých signálů. Uspora energie je však menší než lineární. Obvykle se přiškrcování používá, pokud není dostupná změna frekvence nebo je nutné dosáhnout maximální úspory energie. Tato technologie vyžaduje kontrolu zvláštním procesem. Systémové rozhraní je v `/proc/acpi/processor/*/throttling`.

**Uspání procesoru** Operační systém v případě nečinnosti procesor uspí zasláním příkazu `halt`. Uspání má stavy C1, C2 a C3. V neekonomičtějším stavu C3 je zastavena také synchronizace vyrovnávací paměti procesoru a operační paměti. Tento stav je tedy možné nastavit pouze v případě, že žádné zařízení nepřistupuje k operační paměti a nemění její obsah. Některé ovladače vylučují uvedení do stavu C3. Aktuální stav můžete zjistit v souboru `/proc/acpi/processor/*/power`.

Změna frekvence a přiškrcování jsou účinné pouze při velkém zatížení procesoru, protože u nevytíženého procesoru je automaticky aplikován ekonomický režim C. V případě pracujícího procesoru je doporučená metoda spoření energie změna frekvence. Ve většině případů totiž není procesor zcela vytížen a může bez problémů pracovat i na nižší frekvenci. Obvykle je nejvhodnější dynamická změna frekvence. Statické nastavení má význam pouze pokud stálá nižší frekvence vede k významným úsporám energie nebo pokud je potřeba, aby byl počítač dobře chlazený a tichý.

Přiškrcování je metoda poslední volby, např. v případě potřeby maximální vydrže baterií. Některé systémy nemusí při větším přiškrcení běžet korektně. Přiškrcení nemá žádný smysl, pokud je procesor málo vytížen.

V systému SUSE LINUX jsou tyto technologie kontrolovány pomocí démona Powersave. Nastavení je popsáno v části 16.5 na straně 295.

### 16.3.3 Nástroje ACPI

K dispozici je řada více či méně komplexních ACPI nástrojů pro zobrazení informací jako např. stav baterií nebo teplota (`acpi`, `klaptopdaemon`, `wmacpimon` atd.), nástrojů umožňujících přístup ke struktuře `/proc/acpi` nebo pomáhajících monitorovat změny (`akpi`, `acpiw`, `gtkacpiw`) a také nástroje pro editaci ACPI tabulek v BIOSu (balíček `pmttools`).

### 16.3.4 Možné problémy

V zásadě se můžete setkat se dvěma základními typy problémů. V prvním případě může jít o selhání podpory ACPI v jádře. V takovém případě, hned jak bude k dispozici oprava, můžete problém vyřešit stažením a instalací novějšího typu jádra. Druhý typ problému je spojen s BIOSem počítače. Ne všichni výrobci bohužel správně dodržují ACPI specifikaci. Jejich zařízení pak nefungují správně.

Zařízení s chybnou implementací ACPI jsou zařazeny na černou listinu linuxového jádra. Jádro pak pro tato zařízení ACPI nepoužije.

První krok, který byste při řešení problému s ACPI měli udělat, je update BIOSu. Tím můžete vyřešit mnoho problémů. Pokud se počítač nespouští správně, můžete použít jeden z parametrů jádra:

**pci=noacpi** Nepoužívat ACPI pro nastavená PCI zařízení.

**acpi=oldboot** Provést jen základní nastavení. Nepoužívat ACPI k ničemu jinému.

**acpi=off** Vypnout ACPI.

V dalším kroku pečlivě prostudujte startovací záznamy. To můžete udělat např. příkazem `dmesg | grep -2i acpi` (nebo si nechte zobrazit všechny záznamy, protože chyba může být zapříčiněna něčím jiným). Pokud při parsování ACPI tabulky dojde k chybě, lze přepsat nejdůležitější tabulku – DSDT. To způsobí, že DSDT BIOSu bude ignorována. Jde však o značně složitý úkol, který by měl provádět pouze expert. Pro některé počítače jsou opravené DSDT tabulky dostupné na Internetu.

Při nastavení jádra máte možnost nastavit vytváření ladicích zpráv ACPI. Pokud jste překompilovali a nainstalovali jádro s ACPI laděním, mohou být výpisy jádra cennými informacemi při hledání chyby.

V případě problémů s BIOSem nebo hardwarem je vždy užitečné kontaktovat výrobce zařízení. Ne všichni výrobci jsou sice schopni poskytnout pomoc v případě podpory Linuxu, ale vždy je dobré je o svém problému informovat. Pokud se výrobce setká s větším počtem stížností na funkci svého výrobku, je větší pravděpodobnost, že chybu opraví. Pokud chcete, můžete také informovat výrobce svého hardwaru, že vám na něm Linux funguje bez jakýchkoliv problémů.

## Další informace

Dodatečnou dokumentaci najdete na následujících stránkách:

- <http://www.cpqlinux.com/acpi-howto.html> (podrobné ACPI HOWTO a DSDT opravy)
- <http://www.intel.com/technology/iapc/acpi/faq.htm> (ACPI FAQ @Intel)
- <http://acpi.sourceforge.net/> (ACPI4Linux projekt)
- <http://www.poupinou.org/acpi/> (DSDT opravy od Bruna Ducrota)

## 16.4 Zastavení disku

Pokud se disk nepoužívá, lze ho pod Linuxem zastavit. Slouží k tomu program `hdparm`, se kterým lze nastavit i další funkce disku. Volbou `-y` se disk okamžitě suspenduje, volbou `-Y` se úplně vypne. Příkazem `hdparm -S 6` se disk vypne po 30 sekundách nečinnosti. (Číslo 6 znamená počet intervalů po 5 sekundách, tj.  $6 \cdot 5 = 30$  sekund. Hodnota 0 zastavování disku zruší. U větších hodnot je větší multiplikátor, přesněji viz manuálovou stránku.)

Pokud chcete nastavit suspendování závisle na provozu z baterií nebo z elektrické sítě, najdete potřebné proměnné v souboru `/etc/rc.config.d/apmd.rc.config`. Proměnná `APMD_CHECK_TIME` pak musí být nastavena na hodnotu 0.

Často se stává, že zastavování disku je nepraktické, protože mnoho programů na něj ukládá dočasná data nebo záložní kopie -- například editory. V některých případech to lze řešit, například, jak již bylo popsáno, použitím příkazu `tailf LogSoubor` při zobrazování narůstajícího výpisu.

Uvedení disku do klidu však vůbec není tak jednoduché, jak se z popisu výše může zdát. V Linuxu neustále probíhá celá řada procesů, které zapisují nebo ukládají na disk. Všechna data se před zápisem nejdříve shromáždí v zásobníku paměti. Tento zásobník spravuje **Kernel Update Daemon** (kupdated). Jakmile jsou data v zásobníku určitou dobu, dojde k vyprázdnění zásobníku zápisem na disk. Velikost zásobníku je dynamická a závisí na velikosti operační paměti. Aby byla zajištěna co největší bezpečnost dat, stará se kupdated o tom, aby byla data na disk zapisována v pravidelných krátkých intervalech. Každých 5 sekund kontroluje zásobník a volá `bdflush`, pokud zásobník obsahuje data starší než 30 sekund nebo je zaplněn více než z 30 procent. Pokud máte stabilní systém, můžete toto nastavení změnit.

---

### Důležité

#### Bezpečnost dat

Změna nastavení Kernel Update démona může vést k ohrožení bezpečnosti dat. Pokud si nejste jistí, jaké důsledky budou změny mít, raději je neprovádějte.

---

### Důležité

Nastavení timeoutu disku a intervalu démona kupdated s hodnotami zaplnění zásobníku nastavíte v souboru `/etc/sysconfig/powermanagement`.

Nastavení provedete dvakrát. Jednou pro provoz s baterií a jednou pro provoz s připojením do sítě. Další informace o tomto tématu najdete v souboru `/usr/share/doc/packages/powersave`.

Pomocí `bdflush` zapisují na disk metadata také žurnálovací souborové systémy jako ReiserFS nebo Ext3. Pro ošetření tohoto zápisu existuje podpora v jádře. Tato podpora byla vyvinuta především pro mobilní zařízení. Podrobnější popis této problematiky najdete v souboru `/usr/src/linux/Documentation/laptop-mode.txt`.

Další zápis na disk mohou provádět také aplikace, se kterými právě pracujete. Například naprostá většina textových editorů si vytváří bezpečnostní kopie právě editovaného textu. Pokud by došlo k pádu programu, můžete tak obnovit editovaný soubor. Toto ukládání se však provádí během editace textu a neustále aktivuje disk. Na druhou stranu, pokud deaktivujete ukládání bezpečnostní kopii, riskujete bezpečnost souboru.

Zvláštní nastavení vhodné pro situace, kdy potřebujete mít disk co nejvíce v klidu, má také démon `postfix`. Jde o proměnnou `POSTFIX_LAPTOP`. Pokud tuto proměnnou nastavíte na hodnotu `yes`, maximálně se omezí přístup `postfix` k disku. Aktivace tohoto parametru však nemá větší význam, pokud prodloužíte interval pro `kupdated`.

## 16.5 Balík powersave

`powersave` je jedním z nejužitečnějších balíčků určených především pro notebooky, kde je velmi důležité kontrolovat stav baterií a proces napájení systému. Řada funkcí je užitečná i pro běžnou pracovní stanici (např. `Suspend/Standby`, funkce `ACPI` a možnost zastavení IDE disků).

Balíček slučuje všechny funkce správy napájení. Podporuje hardware, který využívá technologie `ACPI`, `APM`, `PowerNow!` a např. i technologii `SpeedStep`. Obsahuje funkce balíčků:

- `apmd`
- `acpid`
- `ospm`
- `cpufreqd`

- `cpuspeed`
- `powersave`

Z toho důvodu není možné, pokud chcete používat `powersave`, spouštět zároveň démony obsažené ve výše jmenovaných balíčcích.

Doporučujeme vám používat `powersave` i v případě, že hardware nepodporuje všechny uvedené technologie. Případné změny hardwaru démon rozpozná automaticky.

---

## Důležité

### Informace o powersave

Mimo této kapitoly najdete velmi užitečné informace o `powersave` také v souboru `/usr/share/doc/packages/powersave/README_POWERSAVE`.

---

Důležité

## 16.5.1 Konfigurace powersave

Nastavení `powersave` je rozděleno do několika souborů:

### `/etc/sysconfig/powersave/common`

Soubor ze základním nastavením démona `powersave`. V tomto souboru lze například významně zkrátit zapis démona do záznamů (do souboru `/var/log/messages`) nastavením nižší hodnoty proměnné `POWERSAVE_DEBUG`.

### `/etc/sysconfig/powersave/events`

Soubor potřebný pro zpracování systémových událostí. Každé události lze přiřadit externí akci nebo akce nebo akce vykonávané přímo démonem. V případě externích akcí se démon snaží spustit některý ze skriptů uložený v adresáři `/usr/lib/powersave/scripts/`. Předdefinované interní akce jsou:

- `ignore`
- `throttle`
- `dethrottle`
- `suspend_to_disk`
- `suspend_to_ram`



- `standby`
- `do_suspend_to_disk`
- `do_suspend_to_ram`
- `do_standby`

`throttle` přiškrcuje procesor na hodnotu zadanou v proměnné `POWERSAVE_MAX_THROTTLING`. Tato proměnná je závislá na aktuálním schématu. `dethrottle` nastavuje procesor na plný výkon. `suspend_to_disk`, `suspend_to_ram` a `standby` zachycují systmové události režimu uspání. Tyto tři akce jsou odpovědné především za uspávání, ale vždy by měly být asociovány se zvláštními systémovými událostmi.

Adresář `/usr/lib/powersave/scripts` obsahuje skripty pro následující akce:

**notify** Upozornění o události na textové konzoli, v grafickém prostředí nebo zvukovým signálem.

**screen\_saver** Aktivace spořiče obrazovky.

**switch\_vt** Užitečná akce v případě, že se po probuzení nebo standby režimu nechová korektně obrazovka.

**wm\_logout** Uložení všech nastavení a logy z GNOME, KDE nebo jiného grafického prostředí a provede odhlášení.

**wm\_shutdown** Uložení nastavení GNOME nebo KDE a vypnutí systému.

V případě nastavení proměnné `POWERSAVE_EVENT_GLOBAL_SUSPEND2DISK="prepare_suspend_to_disk do_suspend_to_disk"`, provedou se při uspání na disk dva skripty nebo akce v zadaném pořadí. Démon `powersaved` spustí externí skript `/usr/lib/powersave/scripts/prepare_suspend_to_disk`. Po úspěšném vykonání tohoto skriptu provedete démon interní akci `do_suspend_to_disk` a po té, co skript odstraní kritické moduly, počítač uspí.

Akci tlačítka (uspání) lze pozměnit v proměnné `POWERSAVE_EVENT_BUTTON_SLEEP="notify suspend_to_disk"`. V takovém případě budou uživatelé o uspání informováni externím skriptem `notify`. Následně je generována událost `POWERSAVE_EVENT_GLOBAL_SUSPEND2DISK` vedoucí k akcím popsaným výše a bezpečnému uspání systému. Skript `notify` lze upravit pomocí proměnné `POWERSAVE_NOTIFY_METHOD` v souboru `/etc/sysconfig/powersave/common`.

#### **/etc/sysconfig/powersave/cpufreq**

Soubor obsahuje proměnné pro nastavení optimalizace dynamického nastavení frekvence procesoru.

#### **/etc/sysconfig/powersave/battery**

Omezení baterie a další pro specifická nastavení baterie.

#### **/etc/sysconfig/powersave/sleep**

V tomto souboru se aktivuje uspávání, nastavují kritické moduly, které je nutné pře uspáním odstranit ze systému, a určují služby, jež je nutné před uspáním nebo před režimem standby zastavit. Po probuzení počítače jsou zadané moduly opět zavedeny do systému a služby spuštěny. Proces uspání lze z důvodů bezpečného uložení souborů odložit. Výchozí nastavení se ve většině případů týká USB a PCMCIA modulů. Selhání uspání nebo režimu standby je obvykle zapříčiněno některým z modulů. Více informací o zjišťování příčin selhání najdete v části 16.5.3 na straně 301.

#### **/etc/sysconfig/powersave/thermal**

Aktivace chlazení a kontroly teploty. Podrobnosti o tomto tématu najdete v souboru `/usr/share/doc/packages/powersave/README.thermal`.

#### **/etc/sysconfig/powersave/scheme\_\***

Různá schémata správy napájení závislá na situaci nasazení počítače. Mimo již přednastavených schémat se zde ukládají také vlastní schémata.

## **16.5.2 Konfigurace APM a ACPI**

U APM a ACPI můžete provádět nastavení uspávání, vlastní hodnoty pro sledování stavu baterií a samozřejmě různé režimy práce lišící se např. spotřebou energie nebo hlučností.

### **Uspání a probuzení**

Protože režim uspání na některých počítačích stále nefunguje, je ve výchozím nastavení vypnutý. Dostupné jsou tři typy ACPI uspání a dva typy APM uspání:

#### **Uspání na disk (ACPI S4, APM suspend)**

Uložení obsahu paměti na disk. Počítač se zcela vypne a nespotřebovává elektrickou energii.

#### **Uspání do RAM (ACPI S3, APM suspend)**

Uložení stavu všech zařízení do operační paměti. Počítač potřebuje elektrickou energii pouze pro operační paměť.

**Standby (ACPI S1, APM standby)** Vypnutí některých zařízení (funkce závislá na výrobci).

Aby usnutí, standby a probuzení proběhly bez problémů, ujistěte se, že máte v souboru `/etc/sysconfig/powersave/events` následující nastavení (výchozí nastavení systému SUSE LINUX):

```
POWERSAVE_EVENT_GLOBAL_SUSPEND2DISK=
    "prepare_suspend_to_disk do_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_SUSPEND2RAM=
    "prepare_suspend_to_ram do_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_STANDBY=
    "prepare_standby do_standby"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2DISK=
    "restore_after_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2RAM=
    "restore_after_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_RESUME_STANDBY=
    "restore_after_standby"
```

### Uživatелеm definovaný stav baterie

V souboru `/etc/powersave.conf` můžete nastavit tři hodnoty týkající se kapacity baterií. Jde o stavy v procentech, při jejichž dosažení buď dojde k hlášení o stavu baterií nebo se spustí nějaká akce.

```
POWERSAVED_BATTERY_WARNING=20
POWERSAVED_BATTERY_LOW=10
POWERSAVED_BATTERY_CRITICAL=5
```

Jaké akce se spustí, lze nastavit v souboru `/etc/powersave.conf`. Typy akcí nastavíte v souboru `/etc/sysconfig/powersave/common`:

```
POWERSAVE_EVENT_BATTERY_NORMAL="ignore"
POWERSAVE_EVENT_BATTERY_WARNING="notify"
POWERSAVE_EVENT_BATTERY_LOW="notify"
POWERSAVE_EVENT_BATTERY_CRITICAL="suspend"
```

Další možnosti nastavení najdete v komentářích konfiguračního souboru.

## Nastavení spotřeby na různé režimy práce

Svůj systém můžete nastavit tak, aby se při různých způsobech napájení, choval jiným způsobem. Tak můžete dočasně z důvodů šetření energie snížit výkon svého systému, a po připojení do sítě ho pak zase zvýšit. Konkrétními příklady změn nastavení jsou frekvence procesoru, aktivita disku, spořicí funkce a další vlastnosti.

V souboru `/etc/powersave.conf` můžete prostřednictvím `powersave_proxy` nastavit různé spořicí kroky. V souboru `/etc/sysconfig/powersave/common` k nim můžete nastavit různé scénáře (nazývané ‘schéma’ nebo ‘profily’):

```
POWERSAVE_AC_SCHEME="performance"
POWERSAVE_BATTERY_SCHEME="powersave"
```

‘Schémata’ jsou uložena do jednotlivých souborů v adresáři `/etc/sysconfig/powersave`. Jméno se vždy skládá z částí: `scheme_FJmenoSchemata`. V našem případě máme dvě schémata `scheme_performance` a `scheme_powersave`. předkonfigurována jsou schémata `performance`, `powersave` a `acoustic`. Již existující schémata můžete kdykoliv měnit pomocí programu YaST. Pomocí programu YaST můžete také schémata vytvářet a mazat.

## Další funkce ACPI

Pokud používáte ACPI, můžete si nastavit ‘ACPI tlačítka’ (‘Power’, ‘Sleep’ a ‘Otevření’, ‘Zavření’). Příslušné akce pro `powersave_proxy` lze nastavit v souboru `/etc/powersave.conf`. Jednotlivé akce jsou nastavené v souboru `/etc/sysconfig/powersave/common`. Více informace o nastavení najdete v komentářích těchto konfiguračních souborů.

### **POWERSAVE\_EVENT\_BUTTON\_POWER="wm\_shutdown"**

Po stisknutí klávesy ‘Power’ se ukončí nastavený správce oken (KDE, GNOME, fwm...).

### **POWERSAVE\_EVENT\_BUTTON\_SLEEP="suspend"**

Po stisknutí klávesy ‘Sleep’ dojde k usnutí notebooku.

### **POWERSAVE\_EVENT\_BUTTON\_LID\_OPEN="ignore"**

Při otevření notebooku nedojde k žádné akci.

### **POWERSAVE\_EVENT\_BUTTON\_LID\_CLOSED="screen\_saver"**

Při zavření notebooku se aktivuje spořič obrazovky.

Nastavení procesoru můžete provést prostřednictvím proměnných

`POWERSAVED_CPU_LOW_LIMIT` a `POWERSAVED_CPU_IDLE_TIMEOUT`.

### 16.5.3 Možné problémy

V následující části najdete nejčastější dotazy a problémy související s používáním powersave.

#### Obecný postup určení příčiny problémů

Nejdříve se podívejte do souboru `/var/log/messages`. Do tohoto souboru se zapisuje řada chybových hlášení systému. Pokud v tomto souboru nic nenajdete, nastavte v souboru `/etc/sysconfig/powersave/common` proměnnou `DEBUG` na hodnotu 7 nebo 15. Pak restartujte démona. Všechna chybová hlášení powersave se pak budou zapisovat do souboru `/var/log/messages`.

#### ACPI je aktivován, ale klávesy ani stav baterie nereaguje podle nastavení

Zda se jedná o problémy související s ACPI zjistíte pomocí příkazu `dmesg` zadáním:

```
dmesg|grep -i acpi
```

Jestliže najdete nějaká chybová hlášení, updatujte BIOS. Novou verzi BIOSu najdete na stránkách výrobce své základní desky.

V případě, že chyba přetrvává i po updatu BIOSu, vyhledejte na stránkách pro svůj systém také aktuální tabulku DSDT a nahraďte jí tabulku v BIOSu:

- Ze stránky <http://acpi.sourceforge.net/dsdt/tables> si stáhněte DSDT tabulku. Ujistěte se, že jde o správný a překompilovaný soubor (obsahuje příponu `.aml` (ACPI Machine Language)). Pokud jste pro svůj systém našli takový soubor, pokračujte krokem 3.
- Pokud jste našli tabulku s příponou `.asl` (ACPI Source Language), musíte ji nejdříve pomocí `iasl` z balíčku `pmtools` překompilovat. Zadejte příkaz:

```
iasl -sa JmenoSouboru.asl
```

Nejnovější verzi programu `iasl` (Intel ACPI Compiler) najdete na stránce <http://developer.intel.com/technology/iapc/acpi/>.

- Překopírujte soubor `DSDT.aml` do systému (v našem případě `/etc/DSDT.aml`). Editujte soubor `/etc/sysconfig/kernel` a zadejte zde cestu k DSDT souboru. Spusťte příkaz:

```
mkinitrd
```

Tímto příkazem zajistíte, že se tabulka zavede ještě před startem jádra.

## Důležité

Náhrada DSDT tabulky vyžaduje pokročilejší znalosti správy počítače. Při nesprávném postupu může dojít k nefunkčnosti systému.

## Důležité

### Nefunguje nastavení CPU frekvence.

Překontrolujte v dokumentaci, zda je u vašeho procesoru tato funkce podporována a zda jsou zavedeny všechny potřebné moduly a nastavené správné parametry těchto modulů. Všechny potřebné informace najdete v souboru `/usr/src/linux/Documentation/cpu-freq/*`. Pokud je potřeba nastavit určité parametry, proveďte změny v souboru `/etc/sysconfig/powersave/common` pomocí proměnných `CPUFREQD_MODULE` a `CPUFREQD_MODULE_OPTS`.

### Nelze uspávat a budit počítač

V současné době je známo několik problémů s uspáváním a probouzením na systémech používajících ACPI:

- Systémy s více jak 1 GB RAM nemají v současné době podporu uspání.
- Víceprocesorové systémy nebo systémy s procesorem P4 nemají v současné době podporu uspání.

Problém může spočívat také v chybné implementaci DSDT. V takovém případě nahraďte novou DSDT podle postupu uvedeného v Aktivovala jsem ACPI, ale klávesy ani stav baterie nereaguje podle nastavení?

Pro APM i ACPI systémy:

Při pokusu o zavedení problémového modulu zamrzne proxy a nedojde k pokynu k uspání. To samé může nastat v okamžiku, kdy službu nebo modul nejde zastavit. V obou případech se můžete pokusit najít problémový modul pomocí úprav v souboru `/etc/sysconfig/powersave/common`:

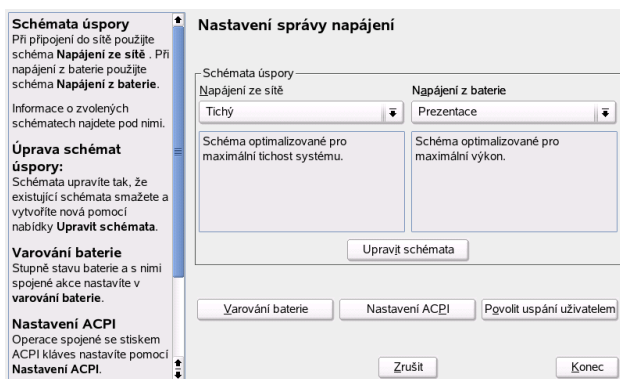
```
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND=" "  
POWERSAVE_UNLOAD_MODULES_BEFORE_STANDBY=" "  
POWERSAVE_SUSPEND_RESTART_SERVICES=" "  
POWERSAVE_STANDBY_RESTART_SERVICES=" "
```

**Powersave u ACPI nesprávně rozpoznává stav baterií.**

Při používání ACPI systém získává informace o stavu baterie od BIOSu. Výhoda tohoto řešení spočívá v tom, že stav baterií není nutné načítat nepřetržitě a tak je snížena zátěž systému a tím i jeho spotřeba. Může se však stát, že k přenosu informací mezi BIOSem a systémem nedochází. V takovém případě nastavte v souboru `/etc/powersave.conf` proměnnou `POWERSAVED_FORCE_BATTERY_POLLING` na hodnotu `yes`.

## 16.6 Modul správy napájení programu YaST

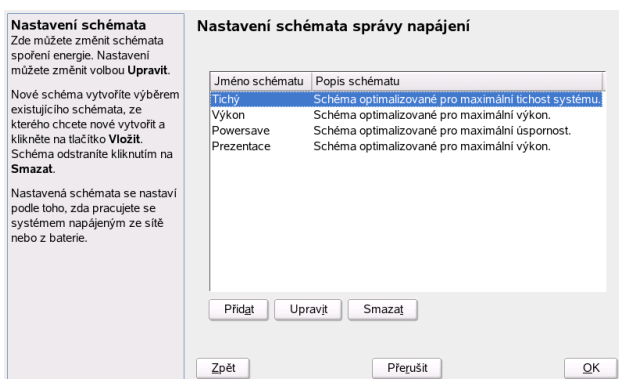
S modulem správy napájení programu YaST lze provést všechna výše zmíněná nastavení. Spustíte jej volbou 'Systém' → 'Správa napájení'. Modul správy napájení je zobrazen na obrázku 16.1 na této straně.



*Obrázek 16.1: Výběr schéma*

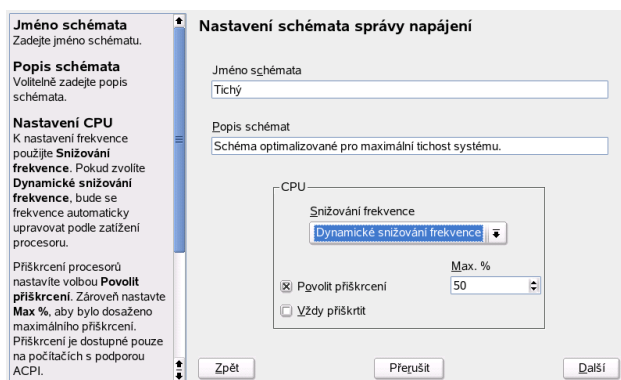
V dialogu správy napájení zvolte schéma, které chcete používat. Pokud chcete přidat nové schéma nebo upravit stávající, klikněte na tlačítko 'Upravit schéma'. Otevře se dialog podobný obrázku 16.2 na následující straně.





*Obrázek 16.2: Přehled existujících schémat*

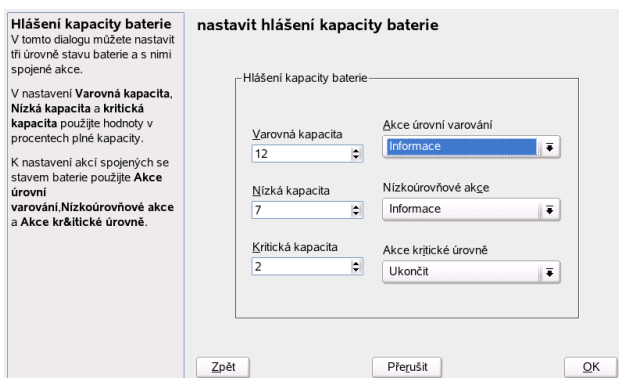
V seznamu schémat vyberte to, které chcete upravit a klikněte na tlačítko 'Upravit'. Nové schéma přidáte kliknutím na tlačítko 'Přidat'. Dialog, který se otevře, můžete vidět na obrázku 16.3 na následující straně.



*Obrázek 16.3: Přidání schéma*

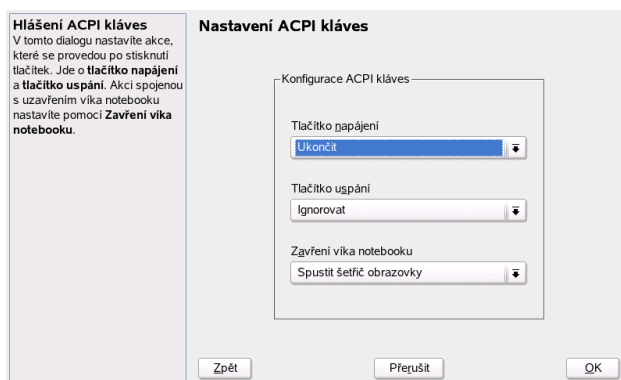
Nejdřív u upravovaného nebo nového schématu zadejte jméno a popis. Definujte ovládání výkonu procesoru. Nastavit můžete změnu frekvence CPU a přiškrcování. V následujícím dialogu nastavte politiku disku a chlazení. Některé metody politiky chlazení nemusí být podporovány BIOSem. Přesnější informace o používání větráčků a pasivním chlazení najdete v souboru `/usr/share/doc/packages/powersave/README.thermal`. Po nastavení požadovaných hodnot klikněte na tlačítko 'Další'. V následujícím dialogu nastavte spoření monitoru. Po nastavení všech hodnot se vraťte do úvodního dialogu kliknutím na tlačítko 'OK'. Nově vytvořené schéma aktivujete a modul ukončíte kliknutím na tlačítko 'OK'.

Obecná nastavení správy napájení lze provést také z dialogu ‘Varování baterie’, ‘Nastavení ACPI’ nebo ‘Povolit uspání uživatelem’. Kliknutím na ‘varování baterie’ se otevře dialog zobrazen na obrázku 16.4 na této straně.



**Obrázek 16.4:** Nabíjení baterie

Po překročení určené kapacity napájení BIOS varuje operační systém. V tomto dialogu můžete nastavit tři různé typy limitů: ‘Varovná kapacita’, ‘Nízká kapacita’ a ‘Kritická kapacita’. Po překročení těchto limitů se provedou k nim přidružené akce. U prvních dvou se obvykle jedná o varování. Třetí limit vede k vypnutí počítače, protože není možné nadále napájet systém. Po nastavení limitů a jejich akcí se vraťte do úvodního dialogu kliknutím na tlačítko ‘OK’.



**Obrázek 16.5:** *Nastavení ACPI*

ACPI tlačítka nastavíte v dialogu dostupném po kliknutí na ‘Nastavení ACPI’. Dialog je znázorněn na obrázku 16.5 na této straně. Nastavení ACPI tlačítek určuje, jak bude systém reagovat na stisknutí určitých tlačítek jako tlačítko uspávání nebo také zavření víka notebooku. Po nastavení limitů a jejich akcí se vraťte do úvodního dialogu kliknutím na tlačítko ‘OK’.

Kliknutím na tlačítko ‘Povolit uspání uživatelem’ vyvoláte dialog, ve kterém můžete nastavit možnosti uživatelů používat funkce uspání a probouzení. Po nastavení limitů a jejich akcí se vraťte do úvodního dialogu kliknutím na tlačítko ‘OK’. Dalším kliknutím na tlačítko ‘Konec’ aktivujete všechny změny ve správě napájení.

# Bezdrátová komunikace

V linuxovém systému si můžete zvolit, jakým způsobem bude váš notebook komunikovat s ostatními počítači, mobilem nebo periferními zařízeními. Pro připojení počítače do sítě nejspíš zvolíte WLAN (*Wireless LAN*). Bluetooth slouží nejčastěji k připojení jednotlivých periférií (myš, klávesnice), mobilů, PDA a připojení počítačů. IrDA je nejčastěji používána při komunikaci s PDA nebo mobilním telefonem. V této kapitole najdete informace o základním nastavení všech tří možností.

17.1	Bezdrátové sítě . . . . .	310
17.2	Bluetooth . . . . .	317
17.3	IrDA Infrared Data Association . . . . .	327

## 17.1 Bezdrátové sítě

Bezdrátové sítě jsou významnou součástí mobilní výpočetní techniky. V současné době má velká část notebooků integrovanou WLAN kartu. Standard 802.11 bezdrátové komunikace WLAN karet byl připraven organizací IEEE. Původně umožňovat maximální rychlost 2 Mb/s. Prošel však řadou změn, které umožnily rychlost zvýšit. Tyto změny definují podrobnosti jako modulaci, přenosový výstup a rychlosti:

*Tabulka 17.1: Přehled různých WLAN standardů*

Jméno	Pásmo (GHz)	Max. přenosová rychlost (Mb/s)	Poznámka
802.11	2.4	2	Zastaralý
802.11b	2.4	11	nejrozšířenější
802.11a	5	54	Méně obvyklý
802.11g	2.4	54	Zpětně kompatibilní s 11b

Dostupné jsou také proprietární variace 802.11b např. od společnosti Texas Instruments s maximální přenosovou rychlostí 22 Mb/s (standard někdy označovaný jako 802.11b+). Rozšíření těchto karet však není velké.

### 17.1.1 Hardware

Karty 802.11 nejsou systémem SUSE LINUX podporovány. Podporována je ale většina karet používajících protokoly 802.11a, 802.11b a 802.11g. Nově karty obvykle podporují standard 802.11g, ale dostupné jsou také karty s podporou 802.11b. Podporovány jsou karty obsahující následující čipové sady:

- Lucent/Agere Hermes
- Intel PRO/Wireless 2100
- Intersil Prism2/2.5/3
- Intersil PrismGT

- Atheros 5210, 5211, 5212
- Atmel at76c502, at76c503, at76c504, at76c506
- Texas Instruments ACX100

Podporována je také řada již nevyráběných starších karet. Vyčerpávající seznam WLAN karet a čipových sad je dostupný na stránce *AbsoluteValue Systems*: [http://www.linux-wlan.org/docs/wlan\\_adapters.html.gz](http://www.linux-wlan.org/docs/wlan_adapters.html.gz). Seznam různých WLAN čipových sad najdete na stránce <http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz>.

Některé karty vyžadují při zavádění ovladače nahrání obrazu s firmwarem. To je případ karet Intel PRO/Wireless 2100 (Centrino), Intersil PrismGT, Atmel a ACX100. Firmware lze snadno doinstalovat pomocí YaST online updatu. Více informací o této problematice najdete v souboru `/usr/share/doc/packages/wireless-tools/README.firmware`.

## 17.1.2 Funkce

Tato část popisuje základní aspekty bezdrátového síťování, operační režimy a způsoby ověřování a šifrování.

### Operační režimy

Bezdrátové sítě lze označit jako spravované (*managed*) nebo ad-hoc sítě. Spravované sítě mají kontrolní bod označovaný obvykle jako přístupový bod. V tomto režimu (také označovaném jako *infrastructure*), jsou všechny stanice připojené přes přístupový bod, který zároveň slouží jako připojení k Ethernetu. Ad-hoc sítě žádný přístupový bod nemají. jednotlivé stanice komunikují přímo mezi sebou. Protože je v ad-hoc sítích výrazně omezený rozsah vysílání a počet stanic, je přístupový bod vhodnějším řešením. Jako přístupový bod lze použít naprostou většinu WLAN karet.

Protože je bezdrátovou sítí snadnější odposlouchávat a kompromitovat než sítí klasickou, řada standardů obsahuje ověřovací a šifrovací metody. V původní verzi standardu IEEE 802.11 jsou popsány pod termínem WEP. WEP však nebyl dostatečně bezpečný (viz. 17.1.5 na straně 316) a tak WLAN výrobci (sdružení do skupiny známé jako *Wi-Fi Alliance*) definovali nové rozšíření WPA, které mělo odstranit slabiny WEP. Pozdější standard IEEE 802.11i (také nazývaný WPA2, protože WPA je založeno na 802.11i) obsahoval nejen WPA, ale také řadu dalších ověřovacích a šifrovacích metod.

## Ověřování

Aby bylo zajištěno, že dojde pouze k ověřeným připojením, obsahují spravované sítě několik ověřovacích mechanismů:

**Otevřený** Otevřený (anglicky *open*) systém nevyžaduje ověření. Do sítě se může připojit každá stanice, ale může být použito WEP šifrování (viz. 17.1.2 na této straně).

**Sdílený klíč (podle IEEE 802.11)** Při této proceduře je používán pro ověření WEP klíč. Tento postup však není doporučován, protože je poměrně náchylný na útoky zvenčí. Vše, co potencionální útočník potřebuje k úspěšnému průniku, je naslouchat komunikaci. Během ověřovacího procesu si obě strany vyměňují stejné informace. Jednou v šifrované a jednou v nešifrované formě. Tak je poměrně jednoduché s pomocí příslušných nástrojů rekonstruovat použitý klíč. Vzhledem k použití klíče pro ověřování i šifrování není tato metoda zvýšení bezpečnosti sítě. Stanice se správným WEP klíčem se může přihlásit do sítě a šifrovat a dešifrovat provoz. Stanice bez klíče nemůže dešifrovat příchozí pakety ani komunikovat.

**WPA-PSK (podle IEEE 802.1x)** WPA-PSK (PSK je zkratka z *Pre-Shared Key*) pracuje podobně jako sdílený klíč. Stanice i přístupový bod používají jeden klíč. Klíč má 256 bitů a obvykle je zadáván jako heslo. tento systém nepotřebuje komplexní správu klíčů jako WPA-EAP a je vhodný pro běžné domácí používání. Proto se někdy o WPA-PSK mluví jako o WPA *home* nebo-li *domácím* WPA.

**WPA-EAP (podle IEEE 802.1x)** WPA-EAP ve skutečnosti není ověřovací systém ale protokol transportu ověřovacích informací. WPA-EAP je používán v podnikovém prostředí. V domácím prostředí je používán zřídka. Z toho důvodu se o WPA-EAP mluví jako o WPA *Enterprise* nebo-li *podnikovém* WPA.

## Šifrování

Existuje řada šifrovacích metod, které se používají k zamezení čtení datových paketů neautorizovanými osobami a přístupu do sítě. Nejdůležitější jsou tyto:

**WEP (definován v IEEE 802.11)** Tento standard používá šifrovací mechanismus RC4 původně s délkou klíče 40 bitů, později s 104 bity. Zda je délka deklarována jako 64 bitů nebo 128 bitů často závisí na tom, zda je zahrnut také 24 bitový inicializační vektor. Tento standard má řadu slabin a klíče mohou být cílem případného útoku. Přesto je WEP vždy lepší než žádné šifrování.



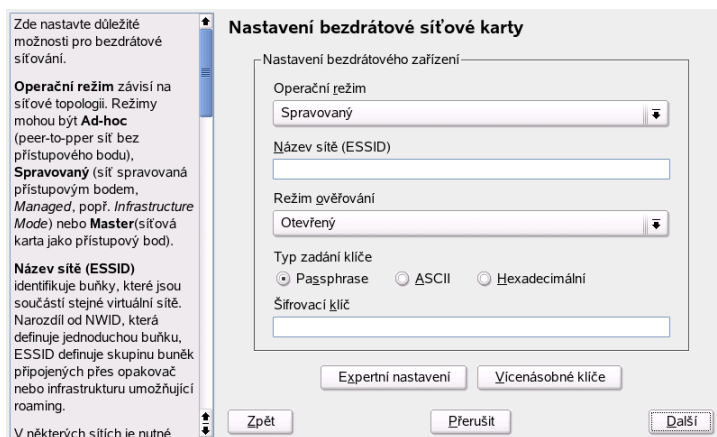
### TKIP (definován v WPA/IEEE 802.11i)

Tento protokol správy klíčů definovaný v standardu WEP používá stejný šifrovací algoritmus jako WEP, ale neobsahuje jeho chyby. Nový klíč je generován pro každý datový paket, což výrazně snižuje pravděpodobnost úspěšného útoku. TKIP se používá současně s WPA-PSK.

**CCMP (definován v IEEE 802.11i)** CCMP popisuje správu klíčů. Obvykle je používán současně s WPA-EAP, ale lze jej používat také s WPA-PSK. Šifrování se řídí podle AES a je silnější než RC4 nebo WEP standard.

## 17.1.3 Nastavení pomocí programu YaST

Bezdrátovou síťovou kartu nastavíte pomocí programu YaST v nabídce 'Síťová zařízení' → 'Síťová karta'. V části 'Konfigurace sítě', nastavte typ zařízení na 'bezdrátová technologie' a klikněte na tlačítko 'Další'.



*Obrázek 17.1: YaST: nastavení bezdrátové síťové karty*

V dialogu 'nastavení bezdrátové síťové karty' na obr. 17.1 na této straně provedete základní nastavení:

**Operační režim** Stanici lze zařadit do sítě ve třech různých režimech. Zvolený režim je závislý na typu sítě: 'Ad-hoc' (peer-to-peer bez přístupového

bodu), 'Spravované' (spravovaná síť s přístupovým bodem) nebo 'Master' (karta je používána jako přístupový bod).

**Jméno sítě (ESSID)** Aby mohly stanice v jedné síti spolu komunikovat, musí používat stejné ESSID. Pokud žádné nezvolí, karta automaticky nastaví některé z dostupných, to však nemusí být to, které chcete používat.

**Režim ověřování** Zvolte vhodný režim ověřování pro svou síť: 'Otevřený', 'Sdílený klíč', nebo 'WPA-PSK'. Pokud zvolíte 'WPA-PSK', musíte nastavit jméno sítě.

**Expertní nastavení** Stisknutím tohoto tlačítka otevřete dialog expertního nastavení, ve kterém můžete provést podrobnější nastavení. Popis tohoto dialogu najdete níže.

Po provedení základního nastavení je síť připravená pro připojení do WLAN.

---

## Důležité

### Bezpečnost v bezdrátových sítích

Ujistěte se, že svou síť chráníte některých ověřovacím a šifrovacím mechanismem. Nešifrované WLAN připojení umožňuje třetím stranám zachytit vaše data. I slabá ochrana (WEP) je lepší než žádná. Více najdete v částech 17.1.2 na straně 312 a 17.1.5 na straně 316.

---

## Důležité

V závislosti na zvoleném režimu ověřování umožňuje YaST nastavení doladit. U režimu 'Otevřený' nelze nic dalšího nastavit, jedná se o nešifrovaný provoz bez ověřování.

**WEP klíče** Nastavte typ vstupu klíče. Na výběr máte z 'Passphrase', 'ASCII' nebo 'Hexadecimal'. Kliknutím na 'Vícenásobné klíče' můžete nastavit až čtyři klíče. Délka klíče může být '128 bitů' nebo '64 bitů'. Výchozí nastavení je '128 bitů'. Jeden ze čtyř klíčů v seznamu můžete označit a kliknutím na tlačítko 'Nastavit jako výchozí' nastavit jako výchozí. Pokud žádný klíč jako výchozí nenastavíte, bude jako výchozí použit první vložený klíč v seznamu. Pokud výchozí klíč smažete, musíte jako výchozí označit jiný klíč. Kliknutím na tlačítko 'Upravit' lze měnit již existující klíče nebo vytvářet nové. V dialogu úpravy budete mít k dispozici všechny typy zadání klíče ('Passphrase', 'ASCII' nebo 'Hexadecimal'). Při výběru 'Passphrase' zadejte slovo nebo řetězec znaků, ze kterých se má klíč vytvořit. U 'ASCII' je vyžadováno zadání pěti znaků pro 64 bitový klíč, 13 znaků pro 64 bitový nebo

26 znaků pro 128 bitový. U 'Hexadecimal' zadejte deset znaků pro 64 bitový klíč nebo 26 pro 128 bitový.

**WPA-PSK** Pro WPA-PSK klíč zvolte vstupní metodu 'Passphrase' nebo 'Hexadecimal'. U režimu 'Passphrase' zadejte 8 až 63 znaků, u režimu 'Hexadecimal' 64 znaků.

Základní nastavení opustíte kliknutím na 'Expertní nastavení'. Volby expertního nastavení jsou následující:

**Kanál** Nastavení kanálu WLAN karty je nutné pouze v režimech 'Ad-hoc' a 'Master'. Ve 'spravovaném' režimu karta dostupné kanály automaticky vyhledá. V Master režimu nastavte, který kanál bude nabízet služby přístupového bodu. Výchozí nastavení je 'Automatický'.

**Přenosová rychlost** Podle výkonnosti vaší sítě můžete nastavit přenosovou rychlost mezi body. Ve výchozím nastavení 'Auto' se systém pokusí použít nejvyšší možnou rychlost. Některé WLAN karty změnu přenosové rychlosti nepodporují.

**přístupový bod** V prostředí s více přístupovými body lze jeden zvolit zadáním MAC adresy.

**Použit správu napájení** Pokud jste na cestách, je zvýšíte výdrž baterií použitím správy napájení. Více informací o správě napájení najdete v kapitole 16 na straně 285.

### 17.1.4 Dostupné programy

hostap (balíček `hostap`) je používán k nastavení WLAN karty jako přístupového bodu. Více informací o tomto programu najdete na domovské stránce jeho projektu (<http://hostap.epitest.fi/>).

kismet (balíček `kismet`) je nástroj pro analýzu WLAN provozu. Tento nástroj vám může pomoci také při odhalování pokusů o průnik do sítě. Více informací najdete na stránce <http://www.kismetwireless.net/> a v manuálové stránce.

### 17.1.5 Tipy a triky nastavení WLAN

Při nastavování bezdrátové sítě se vám může hodit některý z následujících tipů:

## Stabilita a rychlost

Výkon a rychlost bezdrátové sítě závisí na čistotě signálu. překážky jako např. zdi výrazně snižují kvalitu signálu. Se slábnutím signálu se snižuje přenosová rychlost. Sílu signálu můžete překontrolovat pomocí nástroje iwconfig na příkazové řádce nebo pomocí kwifimanager v prostředí KDE. Pokud máte s kvalitou signálu problémy, proveďte nastavení na jiné zařízení nebo se pokuste nasměrovat anténu vašeho přístupového bodu. Přídavné antény lze připojit k řadě PCMCIA WLAN karet. Přenosová rychlost specifikovaná výrobcem (např. 54 Mb/s) je maximální teoretická hodnota. V praxi obvykle získáte něco přes polovinu této hodnoty.

## Bezpečnost

Pokud nastavujete bezdrátovou síť, uvědomte si, že každý v dosahu vysílání může, pokud nepoužíváte šifrování, bez problémů zachytit váš signál. Všechny karty a přístupové body podporují WEP šifrování. Tato metoda ochrany však není naprosto bezpečná a obsahuje možná slabá místa připravená pro potenciální útočníky. WEP je obvykle dostatečná metoda ochrany pro běžné domácí používání. Mnohem bezpečnější je metoda WPA-PSK, která však není dostupná na přístupových bodech a routerech. Na některých zařízeních jí lze použít po updatu firmwaru, nicméně řada zařízení WPA v Linuxu vůbec nepodporuje. Během psaní tohoto článku bylo WPA možné používat pouze s kartami založenými na čipech Atheros nebo Prism2/2.5/3. WPA pracovalo pouze s ovladačem hostap (viz. 17.1.6 na následující straně). Pokud není WPA k dispozici, je WEP lepší než žádné šifrování. V podnikové sféře s vysokými nároky na bezpečnost by bezdrátová síť měla používat WPA.

### 17.1.6 Možné problémy

Pokud WLAN karta neodpovídá, překontrolujte, zda máte potřebný firmware. Více o této problematice najdete v části 17.1.1 na straně 310.

#### Více síťových zařízení

Moderní notebooky mívají síťovou i wlan kartu. Pokud obě zařízení nastavíte na DHCP (automatické přiřazení adresy), může dojít k problémům při přiřazování výchozí brány a resolvování jmen. Problém s resolvováním odhalíte snadno tak, že sice můžete poslat ping na adresu routeru, ale nemůžete brouzdat po internetu.

## Problémy s kartami Prism2

Pro zařízení s čipovou sadou Prism2 je k dispozici několik ovladačů. Kombinace různých ovladačů a různých karet vedou k různé kvalitě příjmu. WPA je dostupné pouze s ovladačem hostap. Pokud vaše karta nepracuje správně nebo chcete používat WPA, přečtěte si `/usr/share/doc/packages/wireless-tools/README.prism2`.

## WPA

Podpora WPA byla poprvé implementována v systému SUSE LINUX. Protože je linuxová podpora WPA stále ve vývoji, podporuje YaST pouze nastavení WPA-PSK. S řadou karet WPA stále nepracuje. Některé karty potřebují pro podporu WPA update firmwaru. Pokud chcete WPA používat, prostudujte si `/usr/share/doc/packages/wireless-tools/README.wpa`.

### 17.1.7 Další informace

Řadu informací o bezdrátových sítích najdete na stránce Jeana Tourrilhes, který vytvořil linuxové nástroje nástroje pro práci s bezdrátovými sítěmi (*Wireless Tools*), na adrese [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html).

## 17.2 Bluetooth

Bluetooth je technologie, která umožňuje propojovat různá zařízení jako mobilní telefony, PDA, notebooky nebo připojovat periférie (např. myši a klávesnice). Své jméno tato technologie získala podle dánského krále Haralda Modrozubého (Bluetooth). Logo Bluetooth je odvozeno od run pro písmena „H“ (podobá se hvězdě) a „B“.

Na rozdíl od IrDA není nutné, aby na sebe zařízení *viděla* a lze propojovat navzájem více zařízení. Pomocí této technologie je možné dosáhnout přenosové rychlosti 720 Kb/s (v aktuální verzi 1.2). Čistě teoreticky lze tento způsob připojení používat i v případě takových překážek, jakou je zeď. V praxi samozřejmě záleží na tloušťce a materiálu, ze kterého je zeď postavena, a třídě zařízení. Maximální dosah této technologie je podle třídy 10 až 100 metrů.

## 17.2.1 Základy

V následující části najdete informace o principech Bluetooth. Seznámíte se s potřebným softwarem a způsobem komunikace Bluetooth rozhraní s vaším systémem a samozřejmě i s Bluetooth profily.

### Software

Abyste mohli využívat Bluetooth, potřebujete Bluetooth adaptér (nejčastěji je integrovaný přímo v zařízení), ovladač a *Bluetooth Protocol Stack*.

Linuxové jádro již základní podporu Bluetooth obsahuje. Jako *Protocol Stack* slouží Bluez systém. Balíčky potřebné k používání Bluetooth:

- `bluez-libs`
- `bluez-bluefw`
- `bluez-pan`
- `bluez-sdp`
- `bluez-utils`

### Základní informace

Systém Bluetooth se skládá ze čtyř propojených vrstev:

**Hardware** Adaptér a příslušný ovladač v linuxovém jádře.

**Konfigurační soubory** Používané pro nastavení Bluetooth systému.

**Démoni** Služby nastavené v konfiguračním souborech a poskytující služby.

**Aplikace** Aplikace využívající služby démonů a ovládané uživateli.

Po vložení Bluetooth adaptéru systém hotplug zavede odpovídající ovladač. Po zavedení ovladače systém překontroluje konfigurační soubory, zda může Bluetooth spustit. Pokud ano, dojde ke spuštění služby a s ní spojených démonů. Z bezpečnostních důvodů je ve výchozím nastavení služba Bluetooth vypnuta.

## Profil

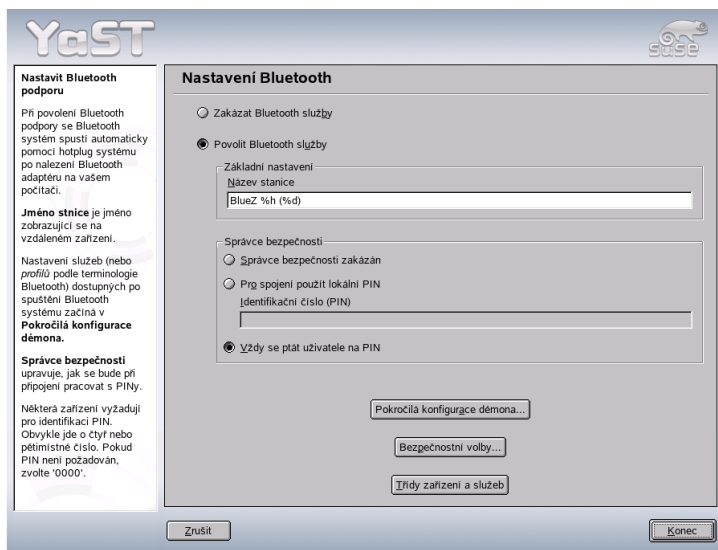
V Bluetooth jsou služby definovány pomocí profilů jako např.. transportní profil nebo základní tiskový profil. Aby zařízení mohlo používat službu jiného zařízení, musí rozumět stejnému profilu informace, která často chybí v balíčku zařízení a v manuálu. někteří výrobci se však nechrání definic profilů, což vede k tomu, že je komunikace mezi jednotlivými zařízeními často velmi problematická.

## 17.2.2 Nastavení

V následující části se dozvíte, jak nastavit Bluetooth na vašem počítači.

### Nastavení Bluetooth pomocí programu YaST

Podporu Bluetooth nastavíte pomocí Bluetooth modulu programu YaST viz. obr.17.2 na této straně. Pokud je pak systémem hotplug detekován Bluetooth adaptér, je Bluetooth automaticky spuštěn s nastaveními provedenými v tomto modulu.



Obrázek 17.2: YaST konfigurace Bluetooth

První krok nastavení v programu YaST představuje povolení spuštění služby Bluetooth. Po povolení služby Bluetooth na nutné provést dvě nastavení. První nastavení se týká položky 'Jméno stanice'. Jde o jméno, které se zobrazí při připojení k systému (počítači) na zařízení. Použít můžete dvě proměnné `%h` pro jméno počítače (užitečné, např. pokud je jméno přiřazováno dynamicky přes DHCP) a `%d` vracející číslo rozhraní (užitečné, pokud připojujete více Bluetooth zařízení na jednu). Pokud do tohoto pole nastavíte `Laptop %h` a přes DHCP získá stanice jméno `unit123`, budou k počítači všechna vzdálená zařízení přistupovat jako k `Laptop unit123`.

Další povinnou částí je 'Správce bezpečnosti', kde nastavujete chování svého systému při připojení vzdáleného zařízení. Správce bezpečnosti můžete zakázat, povolit lokální PIN nebo vyžadovat PIN vždy. Rozdíl mezi posledními dvěma položkami spočívá v tom, že v prvním případě se systém sice na PIN zeptá, ale pokud zařízení PIN neodešle nebo jen PIN chybný, spojení se přesto uskuteční. V druhém případě je PIN vyžadován vždy a spojení bez PINu nebo se zadáním chybného PINu se neprovedou. Z bezpečnostních důvodů vám doporučujeme použít třetí možnost, která navíc umožňuje používat pro různá zařízení různé PINy.

Dostupné služby (v Bluetooth nazývané *profily*) nastavíte v dialogu 'Pokročilá konfigurace démona'. Služby lze povolit kliknutím na tlačítko 'Povolit' a zakázat kliknutím na tlačítko 'Zakázat'. V případě potřeby přenastavení služby jí upravíte jejím výběrem ze seznamu a kliknutím na tlačítko 'Upravit'. Pokud nejste se službou blíže seznámeni, neměňte nastavení. Po provedení všech nastavení ukončete dialog kliknutím na tlačítko 'OK'.

Dialog bezpečnostních nastavení, kde můžete nastavit šifrování, ověřování a nastavení skenování, vyvoláte v hlavním dialogu kliknutím na tlačítko 'Bezpečnostní volby'. Zpět do hlavního dialogu se po nastavení vrátíte kliknutím na tlačítko 'OK'. Všechna nastavení aktivujete kliknutím na tlačítko 'Konec'.

Z hlavního dialogu je dostupný také dialog 'Zařízení a třídy služeb'. Bluetooth zařízení jsou rozdělena do různých „tříd zařízení“. Zvolte pro svůj počítač správné zařazení jako „pracovní stanice“ nebo „notebook“. Nastavení třídy zařízení není tak důležité jako nastavení „třídy služeb“. Některá zařízení, např. mobilní telefony, totiž při špatně zvolené třídě služeb neumožňují využít všechny služby. Zvolit můžete několi tříd zároveň. Obvykle není vhodné předvolit všechny třídy současně. Ve většině případů je dostačující výchozí nastavení.

Pokud chcete nastavit síť, aktivujte v nabídce 'Pokročilá konfigurace démona' profil 'PAND' a nastavte režim služby pomocí tlačítka 'Upravit'. Aby byla síť funkční, je nutné, aby na jednom počítači byl profil pand nastaven na 'naslouchací' režim a na druhém počítači na 'vyhledávací'. Výchozí nastavení je 'Listen'.



Upravte pand podle své potřeby. Dále nastavte rozhraní `bnepX` (X je číslo pořadí zařízení v systému) v modulu ‘Síťová zařízení’ → ‘Síťová karta’.

## Ruční konfigurace

Konfigurační soubory jednotlivých komponent Bluez systému se nacházejí v adresáři `/etc/bluetooth`. Výjimku představuje soubor `/etc/sysconfig/bluetooth` s nastaveními pro start komponent, který je upravován programem YaST module.

Konfiguraci popsanou v následujícím odstavci můžete provádět pouze jako uživatel `root`. V současné době zatím neexistuje žádný grafický konfigurační nástroj. Veškerá nastavení se provádějí pomocí editace textových souborů.

Při prvním spojení se nabídne zabezpečení pomocí PIN. PIN je číslo, které slouží např. u mobilních telefonů jako základní ochrana před nepovolanou manipulací s telefonem. Abyste mohli ovládat dva přístroje současně, musí mít oba stejný PIN. Na straně počítače PIN nastavíte v souboru `/etc/bluetooth/pin`. Bez ohledu na nainstalovaný počet externích zařízení umí Linux v současné době pracovat pouze s jedním PINem. Ovládání několika zařízení s různými PINy najednou není v současné době podporováno. Pokud tedy chcete ovládat více zařízení najednou, musí tato zařízení mít všechna stejný PIN, nebo vypnete ověřování pomocí čísla PIN.

## Důležité

### Bezpečnost Bluetooth spojení

Bez ohledu na to, zda používáte PIN nebo ne, není spojení pomocí Bluetooth naprosto bezpečné!

## Důležité

V konfiguračním souboru `/etc/bluetooth/hcid.conf` lze provést řadu různých nastavení (např. jméno zařízení nebo režim bezpečnosti). Výchozí nastavení však obvykle není nutné měnit. Soubor obsahuje také popis jednotlivých voleb.

Aktivaci Bluetooth provedete v souboru `/etc/bluetooth/hcid.conf`. Zde můžete také změnit různá nastavení jako jméno zařízení či bezpečnostní režim. Soubor obsahuje u každé proměnné vysvětlující komentář.

Důležitou proměnnou je `security auto`. Pomocí této proměnné nastavujete použití PINu. V případě problémů se u tohoto nastavení použití PINu samo vypne. Pokud nechcete PIN používat vůbec, nastavte proměnnou na `none`. Z bezpečnostních důvodů by výchozí nastavení mělo být `user`. Uživatel pak bude při každém připojení požádán o PIN.

Zajímavé jsou také proměnné vázající se k zařízení. Pomocí těchto proměnných můžete zadat, pod jakým jménem bude zařízení připojeno k počítači. Dále jsou zde definována také jednotlivé třídy jako `notebook`, `server` atd. včetně ověřování a připojení.

### 17.2.3 Systémové komponenty a programy pro práci s Bluetooth

Bluetooth je možné používat pouze ve spojení s různými službami. Ke spuštění potřebujete minimálně dva démony:

**hcid** (*Host Controller Interface*) -- k vytvoření a rušení spojení.

**sdpd** (*Service Discovery Protocol*) -- k zjištění dostupných služeb.

Pokud nejsou démoni spuštěni automaticky při startu systému, lze je oba aktivovat příkazem `rcbluetoothstart`. Tento příkaz musí být vykonán s právy uživatele `root`.

Následující text obsahuje stručný popis nejdůležitějších příkazů pro práci s Bluetooth. Ačkoliv je v současnosti pro ovládání Bluetooth dostupná řada grafických programů, může se vám znalost programů příkazové řádky hodit.

Některé příkazy lze vykonat pouze jako uživatel `root`. Jde například o příkaz `l2ping<adresa_zarizeni>`, kterou se testuje připojení vzdáleného zařízení.

#### hcitool

Prostřednictvím `hcitool` lze jednoduše určit, zda jde o lokální nebo vzdálené zařízení. Zařízení zobrazíte příkazem:

```
hcitool dev
```

Příkaz vypíše na každou řádku jedno zařízení ve formátu `JmenoRozhrani AdresaZarizeni`.

Příkazem `hcitool AdresaZarizeni` zjistíte jméno zařízení vzdáleného zařízení. Může jít například o další počítač, který má potřebné informace o třídě a jménu zařízení uložené v `/etc/bluetooth/hcid.conf`. V případě lokálních zařízení vám tento příkaz vrátí chybové hlášení.

Vzdálené zařízení se vyhledává pomocí příkazu `hcitoolinq`. U každého zařízení získáte tři údaje: adresu zařízení, offset hodin a třídu zařízení. Adresa je

důležitá, protože ji ostatní příkazy používají pro identifikaci cílového zařízení. Offset hodin slouží pouze k technickým účelům. Třída určuje typ zařízení a typ služby ve formě hexadecimálního čísla.

Příkaz `hcitool jmeno<adresa-zarizeni>` se používá k určení jména vzdáleného zařízení. V případě vzdáleného počítače je jméno stejné s informacemi v `/etc/bluetooth/hcid.conf`. Zadání lokální adresy povede k chybě výstupu.

## hciconfig

Příkazem `/usr/sbin/hciconfig` získáte informace o lokálních zařízeních. Bez argumentů příkaz zobrazí informace o zařízení jako jméno (`hciX`), fyzickou adresu (dvanácti místné číslo ve formátu `00:12:34:56:78`) a informace o přenesených datech.

`hciconfighci0jmeno` zobrazí jméno vrácené systémem po dotazu na vzdálené zařízení. Změnu nastavení lze provést s pomocí údajů získaných příkazem `hciconfig`, například `hciconfighci0nameTEST` nastaví jméno na `TEST`.

## sdptool

Informace o tom, jaká služba je pro určité zařízení dostupná, získáte pomocí `sdptool`.

Příkaz `sdptool browse AdresaZarizeni` předá všechny služby jednomu zařízení se zadanou adresou.

Naproti tomu příkaz `sdptool search Sluzba` vyhledá jednu určitou službu.

Příkaz se dotáže na všechna dostupná zařízení a vypíše jejich služby spolu s krátkým popisem těchto služeb. Seznam všech dostupných služeb získáte zadáním příkazu `sdptool` bez parametrů.

## 17.2.4 Grafické aplikace

V prohlížeči Konqueror získáte seznam lokálních a vzdálených Bluetooth zařízení zadáním URL `sdp: /`. Dvojklikem na zařízení zobrazíte informace o zařízení. Pokud na zařízení umístíte kurzor, zobrazí se na stavové liště prohlížeče informace o profilu služby. Kliknutím na službu vyvoláte dialog nabízející uložení, použití služby (zařízení musí být aktivováno) nebo zrušení akce. Pokud nechcete, aby se dialog příště opět objevil a došlo přímo k vykonání služby, zatrhněte nabídku příště dialog nezobrazovat. Některá zařízení nejsou doposud podporována. Jiná vyžadují doinstalování dodatečných balíčků.

## 17.2.5 Příklady

Abyste si udělali přehled, co všechno je možné s Bluetooth dělat, připravili jsme pro vás několik příkladů.

### Propojení počítačů R1 a R2

V prvním příkladě si ukážeme, jak se nastavuje připojení mezi dvěma počítači. Potřebovat k tomu budeme pand (*Personal Area Networking*). Všechny příkazy z tohoto příkladu je nutné zadávat jako uživatel `root`. K nastavení síťového připojení bude potřebný také příkaz (`ip`).

Na jednom z počítačů spusťte `pand` (v našem případě označen jako R1) příkazem:

```
pand -s
```

Na druhém počítači R2 získejte adresu pomocí příkazu:

```
hcitool ing
```

Spojení pak navážete zadáním příkazu:

```
pand -c AdresaZarizeni
```

Zjistěte jaké zařízení systém nastavil pro připojení příkazem:

```
ip link show
```

získáte výstup v následujícím formátu:

```
bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000  
    link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

Zařízení `bnep0` byste měli přiřadit IP adresu.

To uděláte např. pomocí následujícím příkazů (na R1):

```
ip addr add 192.168.1.3/24 dev bnep0  
ip link set bnep0 up
```

a na R2:

```
ip addr add 192.168.1.4/24 dev bnep0  
ip link set bnep0 up
```

R1 je z R2 viditelný na adrese IP `192.168.1.3`. Na počítač R2 se z počítače R1 můžete přihlásit příkazem:

```
ssh 192.168.1.4.
```

Příkaz `ssh` bude fungovat i pod normálním uživatelem.

## Datový transfer z mobilního telefonu na počítač

V dalším příkladě se ukážeme, jak překopírovat obrázek z fotoaparátu mobilního telefonu (bez dodatečných nákladů např. za MMS) na disk počítače. Prosím uvědomte si, že každý typ telefonu má jinou strukturu nabídky, ale v základech je postup podobný na všech typech telefonů. Aby bylo možné z telefonu na počítač přistupovat, na počítači musí být aktivována služba Obex-Push. O to se stará démon `opd` z `bluez-utils`. Službu spustíte příkazem:

```
opd --mode OBEX --channel 10 --daemonize --path /tmp --sdp
```

Důležité jsou zde dva parametry. Parametr `--sdp` aktivuje `sdpd`. Parametr `--path /tmp` říká, kam budou data ukládána, v našem příkladu do adresáře `/tmp`. Samozřejmě si můžete zvolit jiný adresář, do kterého máte práva zápisu.

Nyní je potřebné spustit na telefonu Bluetooth připojení. Postup najdete v manuálu vašeho mobilního telefonu. Nezapomeňte nastavit na počítači v souboru `/etc/bluetooth/pin` PIN. Po úspěšném připojení pošlete pomocí Bluetooth obrázky na počítač. Postup zasílání obrázků najdete opět v manuálu mobilního telefonu. Mimo obrázků můžete samozřejmě přenášet také např. hudební soubory.

### 17.2.6 Řešení možných problémů

Pokud máte s nastavením Bluetooth problémy, projděte nejdřív následující seznam postupů. pamatujte, že k chybě může docházet jak na straně počítače, tak na straně zařízení. Pokud máte možnost, otestujte funkčnost zařízení s jiným adaptérem.

#### Je ve výstupu příkazu `hcitooldev` uvedeno lokální zařízení?

Pokud ve výstupu není lokální zařízení uvedeno, nespustil se `hcid` nebo nebylo rozpoznáno Bluetooth zařízení. Příčin může být vícero, zařízení může být porouchané nebo chybí správný ovladač. Notebooky s integrovaným Bluetooth adaptérem mají často pro bezdrátová zařízení vypínač. Zda je nutné zařízení nejdřív fyzicky zapnout zjistíte v manuálu svého notebooku. Restartujte Bluetooth příkazem `rcbluetoothrestart` a podívejte se do souboru `/var/log/messages`, zda systém nevypisuje chyby.

#### Nepotřebuje Bluetooth adaptér soubor s firmwarem?

Pokud ano, nainstalujte balíček `bluez-bluefw` a restartujte Bluetooth příkazem `rcbluetoothrestart`.

**Vrací příkaz hcitooling jiná zařízení?**

Proveďte tento test vícekrát než jednou. Může docházet k interferenci s jiným zařízením používajícím stejnou frekvenci.

**Souhlasí PIN?** Překontrolujte, zda zadaný PIN (v souboru `/etc/bluetooth/pin`) souhlasí se zařízením.

**Vidí vzdálené zařízení počítač?** Pokuste se navázat spojení ze vzdáleného zařízení. Překontrolujte, zda zařízení vidí počítač.

**Nezdaří se síťové propojení počítačů z příkladu 1.**

Příčin může být několik. Jedním může být skutečnost, že jeden nebo oba počítače nerozumí protokolu SSH. Otestujte, zda na sebe počítače vidí příkazy:

```
ping 192.168.1.3
```

a

```
ping 192.168.1.4
```

Pokud proběhnou příkazy bez problémů, ujistěte se, že běží sshd.

Další příčina může spočívat v tom, že jste nastavili jiné adresy, než jsou uvedeny v příkladu nebo jste pro oba počítače nastavili stejnou IP adresu. Změňte IP adresy.

**Nedošlo k rozpoznání počítače jako cíle z propojení počítače a mobilního telefonu z příkladu 2.**

Ujistěte se, že mobil rozpoznal službu Obex-Push na počítači. V nabídce mobilu je obvykle pro takové účely položka, která zobrazuje dostupné služby. Návod najdete v manuálu svého mobilního telefonu. Pokud není služba Obex-Push zobrazena, je problém na straně počítače u programu opd. Ujistěte se, že je opd spuštěn a že máte práva zápisu do zadaného adresáře.

**Je možné kopírovat také z počítače na mobilní telefon?**

Ano, kopírování je možné, pokud nainstalujete program obexftp a použijete příkaz:

```
obexftp -b AdresaZarizeni -B 10 -p Obrazek.
```

Tento postup byl testován na telefonech Siemens a Sony Ericsson a u jiných typů nemusí být funkční.

### 17.2.7 Další informace

Obsáhlý přehled různých návodů na používání a nastavení Bluetooth najdete na stránce <http://www.holtmann.org/linux/bluetooth/>.

- Oficiální howto integrace Bluetooth protocolu do jádra: <http://bluez.sourceforge.net/howto/index.html>
- Připojení k PDA PalmOS: <http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html>

## 17.3 IrDA Infrared Data Association

IrDA (Infrared Data Association) je průmyslový standard pro bezdrátovou komunikaci v infračerveném spektru. Řada dnešních laptopů obsahuje IrDA kompatibilní vysílač a přijímač, umožňující spojení s dalšími zařízeními, jako jsou tis-kárny, modemy, LAN nebo jiné laptopy. Přenosová rychlost sahá od 2400 bps až do 4 Mbps.

IrDA má dva operační režimy. Standardní režim, SIR, přistupuje k zařízení přes sériové rozhraní. Tento režim pracuje na naprosté většině systémů a je dostačující pro většinu požadavků. Rychlejší režim, FIR, vyžaduje pro IrDA čip zvláštní ovladač. Z důvodů neexistence ovladače nejsou ve FIR režimu podporovány všechny čipy. Režim nastavíte v BIOSu svého počítače. V BIOSu také zjistíte, které seriové zařízení bude v SIR režimu používáno.

Informace o IrDA najdete v IrDA HOWTO Wenera Heusera na stránce <http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html>. Další odkazy jsou dostupné na stránkách Linux IrDA projektu <http://irda.sourceforge.net/>.

### 17.3.1 Software

Všechny potřebné moduly jsou již obsaženy v jádře. Nezbytné aplikace pro podporu infračerveného portu a protokolu IrDA jsou součástí balíčku `irda`. Po instalaci balíku naleznete dokumentaci v souboru `/usr/share/doc/packages/irda/README`.

### 17.3.2 Konfigurace

IrDA systém se automaticky nespouští při startu systému. Ke změně tohoto nastavení použijte editor úrovně běhu v programu YaST, případně příkaz `chkconfig`. Každých několik sekund vysílá IrDA "průzkumný paket", kterým vyhledává periferní zařízení ve svém okolí. Tento proces je náročný na spotřebu energie a snižuje výdrž baterií. Z tohoto důvodu je ve výchozím nastavení podpora IrDA vypnuta a měla by být spouštěna pouze v případě potřeby. Ručně ji spustíte příkazem `rcirda start` a vypnete příkazem `rcirda stop`. Všechny potřebné moduly se zavedou automaticky.

Soubor `/etc/syconfig/irda` obsahuje pouze jedinou proměnnou `IRDA_PORT`, pomocí které je nastaveno zařízení rozhraní v SIR režimu. Tuto proměnnou nastavuje skript `/etc/irda/drivers`.

### 17.3.3 Použití

K tisku přes infračervený port pošlete data do souboru zařízení `/dev/irldpt0`. Tento soubor se chová stejně jako normální tiskový port `/dev/lp0`, jediný rozdíl je bezdrátový přenos.

Tiskárnu na tomto portu můžete konfigurovat pomocí YaST stejně jako na paralelním nebo sériovém portu. Při tisku dbejte na to, aby byla vždy zachována přímá viditelnost mezi počítačem a tiskárnou a aby byla aktivována podpora IrDA.

Pro komunikaci s jinými počítači, mobilními telefony a dalšími zařízeními použijte soubor zařízení `/dev/ircomm0`. Například s mobilním telefonem Siemens S25 můžete použít program `wvdial` a mít tak bezdrátové spojení na Internet.

Bez dalších nastavení lze přistupovat pouze k zařízením podporující tiskový nebo IrCOMM protokol. Zařízení s podporou protokolu IROBEX (např. 3Com Palm Pilot) vyžadují zvláštní aplikace jako `irobexpalm` a `irobexreceive`. Více informací o této problematice najdete v IR-HOWTO. Podporovaný protokol zařízení najdete ve výstupu příkazu `irdadump` v závorkách za jménem příslušného zařízení. Podpora IrLAN protokolu je stále ve vývoji a není stabilní.

### 17.3.4 Možné potíže

Pokud zařízení nereagují na IrDA, přihlaste se jako `root` a příkazem `irdadump` se přesvědčte, zda váš počítač zařízení rozpoznal:



irdadump

V případě tiskárny Canon BJC-80 v dosahu počítače se objeví v pravidelných intervalech zprávy, které ukazují výstup na obrazovku:

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                        hint=8804 [ Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* erde
                        hint=0500 [ PnP Computer ] (21)
```

Pokud se výstup neobjeví nebo zařízení neodpovídá, prověřte konfiguraci IrDA. Používáte správný port? Někdy se infraport najde jako `/dev/ttyS2` nebo `/dev/ttyS3` nebo je použito jiné přerušení než 3, což se většinou dá nastavit v BIOSu konfigurovaného notebooku.

Dále je důležité si uvědomit, že IrDA komunikuje pouze se zařízeními, podporujícími protokoly `Printer` nebo `IrCOMM`. Na podporu protokolu `IROBEX` potřebujete ještě programy `irobex_palm3` a `irobex_receive` a pak můžete komunikovat například s 3Com Palm Pilot. Všechny protokoly podporované zařízením se zobrazí ve výstupu z příkazu `irdadump` za jménem zařízení v hranatých závorkách. Podpora protokolu `IrLAN` je zatím ve vývoji a očekává se v budoucích verzích Linuxu.

Pokud potřebujete zkontrolovat, zda IrDA port vysílá infračervené záření, můžete k tomu použít některou z běžných videokamer, které bývají narozdíl od lidských očí citlivé i v infračervené oblasti.



# Hotplug systém

Podpora pro hotplug v systému SUSE LINUX byla vyvinuta ve spolupráci s projektem *Linux Hotplug*, ale vyznačuje se několika odlišnostmi. Hlavní rozdíl spočívá v tom, že není použit multiplexor událostí `/etc/hotplug.d`, ale hotplug skripty se spouštějí přímo. Je-li to možné, jsou pro inicializaci či zastavení hotplug zařízení použity skripty `/sbin/hwup` a `/sbin/hwdown`.

18.1	Zařízení a rozhraní . . . . .	332
18.2	Hotplug události . . . . .	333
18.3	Hotplug agenti . . . . .	334
18.4	Automatické nahrávání modulů . . . . .	335
18.5	Hotplug PCI zařízení . . . . .	336
18.6	Startovací skripty coldplug a hotplug . . . . .	337
18.7	Analýza chyb . . . . .	337

Hotplug systém se nepoužívá jen pro zařízení, která mohou být připojena a odpojena během provozu systému, ale také pro zařízení detekovatelná až po spuštění linuxového jádra. Zařízení a jejich rozhraní jsou vložena do souborového systému `sysfs`, připojeného pod `/sys`. Dokud není jádro zavedeno, inicializují se pouze naprosto nezbytná zařízení, jako sběrnice, startovací disky a klávesnice.

Obvykle je přítomnost zařízení zjištěna ovladačem, který spustí hotplug událost. Ta je zpracována vhodnými skripty. Pro zařízení, která nelze detekovat automaticky, se používá coldplug a statická konfigurace.

Kromě několika historických výjimek je většina zařízení inicializována při startu systému nebo v okamžiku připojení. Inicializace obvykle vede k registraci rozhraní. Registrace rozhraní spouští hotplug události, které rozhraní automaticky nakonfigurují. Dříve se zařízení inicializovala na základě konfiguračních dat. Dnes systém vyhledává vhodné konfigurační údaje na základě existujících zařízení. Postup při inicializaci byl tedy převrácen, čímž je umožněno pružnější použití hotplug zařízení.

Nejdůležitější vlastnosti hotplug systému se nastavují ve dvou souborech. První z nich, `/etc/sysconfig/hotplug`, obsahuje proměnné ovlivňující chování hotplug a coldplug systému. Všechny proměnné jsou opatřeny vysvětlujícími komentáři. Druhý soubor, `/proc/sys/kernel/hotplug`, obsahuje jméno spustitelného programu volaného jádrem. Konfigurace zařízení je uložena v adresáři `/etc/sysconfig/hardware`. Od verze SUSE LINUX 9.3 je tento soubor prázdný, protože `udev` získává zprávy hotplugu přes netlink soket.

## 18.1 Zařízení a rozhraní

Hotplug systém spravuje zařízení a rozhraní. Zařízení je spojeno se sběrnicí či rozhraním. Sběrnici lze pokládat za vícenásobné rozhraní. Rozhraní propojuje zařízení navzájem nebo s aplikací. K dispozici jsou také virtuální zařízení jako například síťové tunely. Jednotlivá zařízení ke své práci obvykle vyžadují ovladač v podobě modulu jádra. Rozhraní jsou většinou reprezentována nody zařízení vytvořenými `udev`. K pochopení celkové koncepce je nutné rozlišovat mezi zařízeními a rozhraním.

Zařízení vložené do souborového systému `sysfs` najdete v `/sys/devices`, rozhraní v `/sys/class` nebo `/sys/block`. Všechna rozhraní v `sysfs` by měla být prolinkována s příslušným zařízením. Bez tohoto propojení by nebylo možné zjistit, které rozhraní patří ke zařízení a nebylo by možné najít vhodné nastavení.

Zařízení se adresují pomocí popisu zařízení. Popisem může být cesta k zařízení v souborovém systému `sysfs` (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0`), místo připojení (`bus-pci-0000:02:00.0`), jedinečné identifikační číslo (`id-32311AE03FB82538`) nebo obdobný údaj.

Rozhraní se dříve adresovala pomocí jmen. Ta ale odrážela pořadí existujících zařízení a mohla se měnit, kdykoliv bylo nějaké zařízení přidáno nebo odstraněno. Proto je možné rozhraní adresovat také popisem přidruženého zařízení. Z kontextu obvykle jasně plyne, zda se popis týká samotného zařízení nebo jeho rozhraní. Mezi typické příklady zařízení, rozhraní a jejich popisů patří:

**PCI síťová karta** Zařízení je připojeno na PCI sběrnici (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0` nebo `bus-pci-0000:02:00.0`) a má síťové rozhraní (`eth0`, `id-00:0d:60:7f:0b:22` nebo `bus-pci-0000:02:00.0`). Síťové rozhraní je využíváno síťovými službami nebo připojeno k virtuálnímu síťovému zařízení jako je tunel nebo síť VLAN.

**PCI SCSI řadič** Zařízení (`/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0` nebo `bus-scsi-1:0:0:0`) vytvářející několik fyzických rozhraní ve formě sběrnice (`/sys/class/scsi_host/host1`).

**SCSI pevný disk** Zařízení (`/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0` nebo `bus-scsi-1:0:0:0`) s několika rozhraními (`/sys/block/sda*`).

## 18.2 Hotplug události

Každé zařízení a každé rozhraní má přidruženu *hotplug událost*, která je zpracovávána `udev` a odpovídajícím agentem. Hotplug události jsou spouštěny jádrem v okamžiku připojení zařízení nebo ve chvíli, kdy ovladač zaregistruje rozhraní. Od verze SUSE LINUX 9.3 všechny hotplug události zachytává a rozděljuje `udev`. Zda má naslouchat přímo zprávám `netlink` z jádra nebo z `/sbin/udevsend` se nastavuje v `/proc/sys/kernel/hotplug`. Po té, co `udev` vykoná svou práci (viz 19 na straně 339), vyhledá se podle typu události `hotplug agent` v `/etc/hotplug.d/`.

## 18.3 Hotplug agenti

Hotplug agent je spustitelný program provádějící patřičné akce jako odpověď na hotplug událost. Agenti jsou umístěni v adresáři `/etc/hotplug.d/jmeno_udalosti` `/etc/hotplug.d/default`. Všechny programy s příponou `.hotplug` v podadresáři jsou vykonávány v abecedním pořadí.

Pokud chcete, aby některé události byly ignorovány, jednoduše nastavte příslušný soubor jako nespustitelný. Stejného účinku dosáhnete změnou přípony `.hotplug`.

Agenti pro zařízení obvykle nahrávají jaderné moduly, ale mohou volat i další příkazy. V systému SUSE LINUX se o to starají programy `/sbin/hwup` nebo `/sbin/hwdown`, které hledají vhodnou konfiguraci v adresáři `/etc/sysconfig/hardware` a aplikují ji. Chcete-li zabránit inicializaci určitého zařízení, vytvořte příslušný konfigurační soubor s nastavením startovací metody (start mode) `manual` nebo `off`. Pokud `/sbin/hwup` nenalezne žádnou konfiguraci, nahraje automaticky moduly. Více informací najdete v kapitole 18.4 na následující straně. Další informace o programu `/sbin/hwup` najdete v souboru `/usr/share/doc/packages/sysconfig/README` a v manuálové stránce programu `hwup`.

Agenti pro rozhraní se spouští nepřímo pomocí `udev`. `udev` nejprve vytvoří pro zařízení příslušný uzel, ke kterému může systém přistupovat. `udev` umožňuje rozhraním přidělit trvalá jména. Podrobnosti viz 19 na straně 339. Následně jednotliví agenti rozhraní nastaví. Postup pro vybraná rozhraní je popsán dále.

### 18.3.1 Aktivace síťových rozhraní

Síťová rozhraní jsou inicializována pomocí `/sbin/ifup` a deaktivována pomocí `/sbin/ifdown`. Podrobnosti jsou popsány v souboru `/usr/share/doc/packages/sysconfig/README` a v manuálové stránce příkazu `ifup`.

Pokud má počítač několik síťových zařízení s různými ovladači a ta se při startu systému nahrají v jiném pořadí, mohou se označení rozhraní změnit. Proto SUSE LINUX spravuje události pro PCI síťová zařízení s využitím fronty. Tuto vlastnost lze vypnout nastavením proměnné `HOTPLUG_PCI_QUEUE_NIC_EVENTS=no` v souboru `/etc/sysconfig/hotplug`.

Nejlepší způsob, jak dosáhnout konzistence označení rozhraní, je určit jména jednotlivých rozhraní v konfiguračních souborech. Podrobnosti naleznete v souboru `/usr/share/doc/packages/sysconfig/README`. I když síťové rozhraní není nod zařízení, umí s ním od verze SUSE LINUX 9.3 `udev` pracovat.

### 18.3.2 Aktivace zařízení pro ukládání dat

Rozhraní k zařízením pro ukládání dat musí být připojena (přimontována), jinak není možno k zařízení přistupovat. Tento proces lze plně automatizovat nebo předem nakonfigurovat. Konfigurace se provádí v proměnných `HOTPLUG_DO_MOUNT`, `HOTPLUG_MOUNT_TYPE` a `HOTPLUG_MOUNT_SYNC` v souboru `/etc/sysconfig/hotplug` a v souboru `/etc/fstab`.

Plně automatický chod lze zapnout nastavením proměnné `HOTPLUG_DO_MOUNT=yes`. Proměnná `HOTPLUG_MOUNT_TYPE` přepíná mezi módem `subfs` a `fstab`.

Je-li nastavena proměnná `HOTPLUG_MOUNT_TYPE=subfs`, je vytvořen podadresář adresáře `/media`, jehož jméno je odvozeno od vlastností zařízení. Médium je při přístupu automaticky připojováno a odpojováno pomocí `submountd`. Zařízení je v tomto módu možno jednoduše fyzicky odpojit ve chvíli, kdy zhasne přístupová kontrolka.

Je-li nastavena proměnná `HOTPLUG_MOUNT_TYPE=fstab`, zařízení pro ukládání dat jsou připojována (přimontována) klasickým způsobem pomocí příslušného záznamu v souboru `/etc/fstab`. Proměnná `HOTPLUG_MOUNT_SYNC` umožňuje nastavit přístup v synchronním nebo asynchronním módu. V asynchronním módu je přístup pro zápis rychlejší, neboť je používána vyrovnávací paměť. Nicméně neopatrné odpojení zařízení může způsobit ztrátu dat. V synchronním módu jsou všechna data zapsána okamžitě, ale přístup trvá delší dobu. Zařízení musí být odpojeno manuálně příkazem `umount`.

Při použití posledně dvou zmíněných módů je doporučeno využít trvalých jmen zařízení, neboť klasická jména zařízení se mohou měnit v závislosti na inicializační sekvenci. Více informací viz 19 na straně 339.

## 18.4 Automatické nahrávání modulů

Pokud nelze zařízení inicializovat pomocí `/sbin/hwup`, agent se snaží nalézt vhodný ovladač v *mapách modulů*. Nejprve prohledá mapy obsažené v `/etc/hotplug/*.handmap`. Pokud tam ovladač nenalezne, hledá v `/lib/modules/<verze_jadra>/modules.*map`. Aby byl použit jiný než standardní ovladač pro dané jádro, nastavte ho v prvním načítaném souboru `/etc/hotplug/*.handmap`.

USB agent rovněž hledá uživatelské ovladače v souborech `/etc/hotplug/usb.usermap` a `/etc/hotplug/usb/*.usermap`. Uživatelské ovladače jsou

programy obsluhující přístup k zařízení a nahrazující v této úloze jaderné moduly. Je tak možné pro určitá zařízení volat spustitelné programy.

V případě zařízení PCI se nejprve `pci.agent` dotáže programu `hwinfo` na ovladače. Pouze pokud `hwinfo` žádné ovladače nezná, prohledá agent `pci.handmap` a mapu jádra. Protože ale `hwinfo` tato místa již prohledal, dotaz jistě selže. `hwinfo` má dodatečnou databázi ovladačů, nicméně nahrává i `pci.handmap`, aby se ujistil, že byla aplikována veškerá mapování.

Agent `pci.agent` může být omezen pouze na zařízení určitého typu nebo na moduly ovladačů z určitého podadresáře `/lib/modules/<verze_jadra>/kernel/drivers`. V prvním případě lze do proměnných `HOTPLUG_PCI_CLASSES_WHITELIST` a `HOTPLUG_PCI_CLASSES_BLACKLIST` v souboru `/etc/sysconfig/hotplug` vložit třídy PCI zařízení uvedené na konci souboru `/usr/share/pci.ids`. V druhém případě lze v proměnných `HOTPLUG_PCI_DRIVERTYPE_WHITELIST` a `HOTPLUG_PCI_DRIVERTYPE_BLACKLIST` uvést jeden nebo více adresářů. Moduly z vyřazených adresářů nejsou nahrávány. Prázdný whitelist v obou případech znamená, že jsou povoleny všechny možnosti kromě možností uvedených v blacklistu. Moduly, které nemají být agentem nikdy nahrány, uveďte v souboru `/etc/hotplug/blacklist`. Každý modul zapište na samostatnou řádku.

Pokud je v mapovém souboru nalezeno vhodných modulů více, je nahrán pouze první z nich. Aby byly nahrány všechny moduly, nastavte proměnnou `HOTPLUG_LOAD_MULTIPLE_MODULES=yes`. Nicméně je pro takové zařízení lépe vytvořit zvláštní konfiguraci v `/etc/sysconfig/hardware/hwcfg-*`.

Modulů nahrávaných pomocí `hwup` se toto nastavení netýká. K automatickému nahrávání modulů dochází jen ve výjimečných případech, které budou v budoucích verzích systému SUSE LINUX dále omezeny.

## 18.5 Hotplug PCI zařízení

Některé počítače umožňují hotplug i pro PCI zařízení. Aby se této vlastnosti dalo plně využít, musí být nahrány zvláštní jaderné moduly, které ovšem mohou působit problémy na počítačích bez podpory hotplug pro PCI zařízení. Podporu hotplug PCI nelze bohužel automaticky detekovat, a proto musí být nastavena manuálně. Učíníte tak nastavením proměnné `HOTPLUG_DO_REAL_PCI_HOTPLUG` v souboru `/etc/sysconfig/hotplug` na hodnotu `yes`.



## 18.6 Startovací skripty coldplug a hotplug

`boot.coldplug` je zodpovědný za všechna zařízení, která nejsou automaticky detekována a pro která nejsou generovány žádné hotplug události. Nedělá nic jiného, než že pro každou statickou konfiguraci zařízení, která je pojmenovaná jako `/etc/sysconfig/hardware/hwcfg-static-*`, volá `hwup`. Lze pomocí něj dosáhnout i změny pořadí inicializace vestavěných zařízení oproti použití `hotplug`, neboť `coldplug` je spuštěn dříve než `hotplug`.

## 18.7 Analýza chyb

### 18.7.1 Log soubory

Pokud není určeno jinak, posílá `hotplug` do systémového logu pouze pár nejdůležitějších zpráv. Chcete-li získat více informací, nastavte proměnnou `HOTPLUG_DEBUG` v souboru `/etc/sysconfig/hotplug` na hodnotu `yes`. Pokud tuto proměnnou nastavíte na hodnotu `max`, bude zaznamenáván každý shellový příkaz `hotplug` skriptů. Důsledkem bude výrazné zvětšení souboru `/var/log/messages`, do kterého `syslog` ukládá všechny zprávy. Protože se `syslog` během startu systému spouští až po `hotplug` a `coldplug`, může se stát, že první zprávy nebudou v logu uloženy. Pokud jsou tyto zprávy pro vás důležité, nastavte použití jiného log souboru pomocí proměnné `HOTPLUG__SYSLOG`. Více informací o této problematice naleznete v souboru `/etc/sysconfig/hotplug`.

### 18.7.2 Problémy při startu systému

Pokud počítač zamrzne během startu systému, vypněte `hotplug` nebo `coldplug` zadáním `NOHOTPLUG=yes` nebo `NOCOLDPLUG=yes` na výzvu při startu systému. Vzhledem k deaktivaci systému `hotplug` nevzdává jádro žádné hotplug události. V běžícím systému můžete aktivovat `hotplug` příkazem `/etc/init.d/boot.hotplugstart`. Všechny dosud generované události tak budou vydány a zpracovány. Nechcete-li události ve frontě přijmout, zapište do souboru `/proc/sys/kernel/hotplug` cestu `/bin/true` a po chvíli ji přepište na `/sbin/hotplug`. Deaktivace `coldplug` způsobí, že nebudou aplikovány statické konfigurace. Můžete je aplikovat později zadáním příkazu `/etc/init.d/boot.coldplugstart`.

Chcete-li zjistit, zda je za problém zodpovědný některý modul nahrávaný pomocí `hotplug`, zadejte na výzvu při startu systému `HOTPLUG_TRACE=<N>`. Jména všech modulů, které se mají nahrát, jsou pak vypisována na obrazovku dříve, než se skutečně, po  $\langle N \rangle$  sekundách, nahrají. Do průběhu nahrávání nemůžete nijak zasahovat.

### 18.7.3 Zapisovač událostí

Skript `/sbin/hotplugeventrecorder` je programem `/sbin/hotplug` spuštěn při každé události. Pokud existuje adresář `/events`, jsou do něj jako jednotlivé soubory ukládány všechny hotplug události. Mohou být tak znovu použity pro testovací účely. Pokud adresář neexistuje, není nic zaznamenáváno.

# Dynamické uzly zařízení pomocí udev

Linuxové jádro 2.6 přineslo nové řešení v *uživatelském prostoru* umožňující používat v dynamickém adresáři `/dev` pro zařízení stálá a konzistentní označení: `udev`. Předchozí implementace `/dev` pomocí `devfs` již není podporována a byla nahrazena implementací založenou na `udev`.

19.1	Tvorba pravidel . . . . .	340
19.2	Automatizace pomocí NAME a SYMLINK . . . . .	341
19.3	Regulární výrazy v klíčích . . . . .	341
19.4	Výběr klíčů . . . . .	342
19.5	Konzistentní pojmenování zařízení pro hromadné uchovávání dat . . . . .	343

Tradičně byly v linuxových systémech v adresáři `/dev` umístěny uzly (device nodes) pro všechny možné typy zařízení, bez ohledu na jejich skutečnou existenci. Adresář proto zabíral velké množství místa. Příkaz `devfs` přinesl významné zlepšení, neboť díky němu mají v adresáři `/dev` své uzly pouze ta zařízení, která skutečně existují.

Nový způsob vytváření uzlů přinesl příkaz `udev`. Ten porovná informace dostupné ze systému souborů `sysfs` s daty zadanými uživatelem ve formě pravidel. `sysfs` je nový souborový systém dostupný v jádře 2.6. Poskytuje základní informace o zařízeních připojených k systému. Souborový systém `sysfs` je připojený jako `/sys`.

Pravidla není nutno vytvářet. Pokud je k systému připojeno zařízení, je vytvořen příslušný uzel, Pravidla ovšem umožňují změnit jména uzlů. Lze tak nahradit nesrozumitelná jména jmény snadno zapamatovatelnými a dosáhnout konzistentních jmen zařízení, když je připojeno více zařízení stejného typu.

Dvě připojené tiskárny budou například, není-li určeno jinak, označeny jako `/dev/lp0` a `/dev/lp1`. Které tiskárně bude přiřazen který uzel závisí na pořadí, v jakém jsou zapnuty. Jiným příkladem jsou externí zařízení pro ukládání dat, jako USB disky. Příkaz `udev` umožňuje přesně zvolit cesty vkládané do `/etc/fstab`.

## 19.1 Tvorba pravidel

Pravidla načítá `udev` ze souboru `/etc/udev/udev.rules` ještě předtím, než vytvoří uzly v adresáři `/dev`. Pokud odpovídá více pravidel, použije první z nich. Komentáře jsou v souboru uvozeny znakem hash (`#`). Pravidla jsou zapisována v následujícím formátu:

```
klíč, [klíč,...] NAME [, SYMLINK]
```

Každé pravidlo musí obsahovat alespoň jeden klíč, neboť pravidla jsou zařízením přiřazována právě pomocí těchto klíčů. Rovněž je nezbytné určit jméno (parametr `name`). To je totiž přiřazeno uzlu zařízení vytvořenému v adresáři `/dev`. Volitelný parametr `symlink` umožňuje vytvoření uzlů i na dalších místech. Pravidlo pro tiskárnu může vypadat například takto:

```
BUS="usb", SYSFS{serial}="12345", NAME="lp_hp", SYMLINK="printers/hp"
```

V příkladu jsou použity dva klíče `BUS` a `SYSFS{serial}`. Tyto klíče říkají `udev`, aby porovnal sériové číslo obsažené v klíči se sériovým číslem zařízení připojeného na USB sběrnici. Pokud oba klíče souhlasí, přiřadí zařízení jméno `lp_hp` v adresáři `/dev`. Navíc na něj vytvoří symbolický odkaz `/dev/printers/hp`. Adresář `printers` se vytvoří automaticky. Tiskové úlohy bude možno posílat jak na `/dev/printers/hp`, tak i na `/dev/lp_hp`.

## 19.2 Automatizace pomocí NAME a SYMLINK

Parametry `NAME` a `SYMLINK` umožňují využití operátorů pro automatické přiřazení hodnot, které odkazují na informace jádra o příslušném zařízení. Následující jednoduchý příklad objasňuje princip:

```
BUS="usb", SYSFS{vendor}="abc", SYSFS{model}="xyz", NAME="kamera%n"
```

Operátor `%n` v parametru `name` bude nahrazen číslem zařízení `kamera`, např. `kamera0` nebo `kamera1`. Další užitečný operátor je `%k`, který je nahrazován standardním jménem zařízení v jádře, např. `hda1`. Seznam všech operátorů je k dispozici v manuálové stránce `udev`.

## 19.3 Regulární výrazy v klíčích

V interpretu příkazů lze používat regulární výrazy a zástupné znaky. Např. znak `*` lze použít místo libovolných znaků a znak `?` lze použít místo právě jednoho libovolného znaku.

```
KERNEL="ts*", NAME="input/%k"
```

Toto pravidlo přiřazuje standardní jméno ve standardním adresáři zařízení jehož označení začíná písmeny "ts". Podrobné informace o použití regulárních výrazů viz manuálová stránka `udev`.

## 19.4 Výběr klíčů

Důležité je pro každé udev pravidlo vybrat dobrý klíč. Následují příklady běžně používaných klíčů:

**BUS** typ sběrnice

**KERNEL** jméno zařízení používané jádrem

**ID** číslo zařízení na sběrnici (např. ID na sběrnici PCI)

**PLACE** fyzické místo připojení zařízení (např. USB)

Klíče ID a PLACE jsou užitečné, obvykle se ale používají klíče BUS, KERNEL, a SYSFS{...}. Konfigurace udev umožňuje použít i klíče volající externí skripty a vyhodnocující jejich výsledky. Další informace lze získat pomocí příkazu `man udev`.

Souborový systém `sysfs` obsahuje v adresářovém stromu malé soubory s informacemi o hardwaru. Každý soubor obvykle obsahuje jednu informační položku, jako je jméno zařízení, výrobce nebo sériové číslo. Každý z těchto souborů může být použit jako hodnota klíče. V jednom pravidlu však mohou být použity jako klíče pouze soubory nacházející se ve stejném adresáři.

Přitom může být užitečný příkaz `udevinfo`. Je potřeba nalézt podadresář `/sys`, který odpovídá příslušnému zařízení a obsahuje soubor `dev`. Ty se všechny nacházejí v adresáři `/sys/block` nebo `/sys/class`. Pokud pro zařízení již uzal existuje, může vám `udevinfo` ušetřit kus práce. Příkaz `udevinfo -q path -n /dev/sda` vypíše `/block/sda`. To znamená, že hledaný adresář je `/sys/block/sda`. Nyní zavolejte `udevinfo` příkazem `udevinfo -a -p /sys/block/sda`. Oba příkazy lze rovněž sloučit v jeden: `udevinfo -a -p `udevinfo -q path -n /dev/sda``. Toto je část výstupu:

```
BUS="scsi"
ID="0:0:0:0"
SYSFS{detach_state}="0"
SYSFS{type}="0"
SYSFS{max_sectors}="240"
SYSFS{device_blocked}="0"
SYSFS{queue_depth}="1"
SYSFS{scsi_level}="3"
SYSFS{vendor}="          "
SYSFS{model}="USB 2.0M DSC      "
SYSFS{rev}="1.00"
SYSFS{online}="1"
```

Z výstupu příkazu si vyberte vhodné klíče, které se nebudou měnit. Pamatujte, že nelze použít klíče z různých adresářů.

## 19.5 Konzistentní pojmenování zařízení pro hromadné uchovávání dat

SUSE LINUX obsahuje skripty, které pomáhají přiřadit pevným diskům a dalším úložným zařízením vždy stejná jména, `/sbin/udev.get_persistent_device_name.sh` je obalovací skript (wrapper). Nejprve zavolá `/sbin/udev.get_unique_hardware_path.sh`, který zjistí hardwarovou cestu k příslušnému zařízení. Skript `/sbin/udev.get_unique_drive_id.sh` zjistí sériové číslo. Oba výstupy jsou následně předány `udev`, který vytvoří symbolický odkaz na uzel zařízení v adresáři `/dev`. Obalovací skript lze rovněž přímo použít v `udev` pravidlech. Následující příklad pro SCSI může být zobecněn i pro USB nebo IDE (musí být zapsán na jedné řádce):

```
BUS="scsi",  
PROGRAM="/sbin/udev.get_persistent_device_name.sh",  
NAME="%k", SYMLINK="%c{1+}"
```

Jakmile je nahrán ovladač pro zařízení pro hromadné uchovávání dat, zaregistruje všechny dostupné pevné disky v jádře. Každý z nich spustí blokovou hotplug událost, která volá `udev`. `udev` nejdříve načte pravidla aby zjistil, zda je potřeba vytvořit symbolický odkaz.

Pokud je ovladač nahrán prostřednictvím `initrd`, hotplug události se ztratí. Nicméně jsou všechny informace uloženy v souborovém systému `sysfs`. Utilita `udevstart` vyhledá všechny zařízení v `/sys/block` a `/sys/class` a spustí `udev`.

Existuje také startovací skript `boot.udev`, který během startu systému znovu vytvoří všechny uzly zařízení. Tento startovací skript musí být aktivován pomocí editoru úrovně běhu YaST nebo příkazem `insserv boot.udev`.

**Tip**

Mnoho programů a nástrojů spoléhá na skutečnost, že `/dev/sda` je SCSI pevný disk a `/dev/hda` je IDE pevný disk. Pokud tomu tak není, přestanou fungovat. YaST je na těchto nástrojích závislý, takže pracuje pouze jaderným označením zařízení.

**Tip**



# Souborové systémy

Linux podporuje řadu různých souborových systémů. V této kapitole najdete krátký přehled těch nejpobulárnějších včetně jejich popisu, výhod a příkladů vhodného nasazení. Zároveň se zde dočtete o podpoře LFS (*Large File Suppnebot*) v Linuxu.

20.1	Termíny . . . . .	346
20.2	Hlavní souborové systémy Linuxu . . . . .	346
20.3	Některé další podporované souborové systémy . . . . .	352
20.4	Podpora souborů větších než 2 GB . . . . .	353
20.5	Další informace . . . . .	354

## 20.1 Termíny

**metadata** Interní datová struktura souborového systému, která zajišťuje okamžité organizování a přístupnost dat na disku. Lze je nazvat také daty o datech. Prakticky všechny souborové systémy metadata používají a jejich struktura bývá jedním z důvodů odlišných výkonů.

**inod** Inody obsahují různé informace o souboru, včetně velikosti, počtu odkazů, data a času vytvoření, změny a posledního přístupu, stejně jako ukazatele na diskové bloky, kde je soubor skutečně uložen.

**žurnál** Žurnál je struktura na disku obsahující záznam o změnách metadat souborového systému. Žurnálování má významnou zásluhu na obnově souborového systému v případě poškození a kontrole konzistence při startu. Při kontrole jsou obnovovány pouze žurnály.

## 20.2 Hlavní souborové systémy Linuxu

Před několika lety byla volba souborového systému v Linuxu otázkou několika vteřin (bud' Ext2 nebo ReiserFS). Jádra řad 2.4 a 2.6 nabízejí však mnohem víc.

Při volbě souborového systému je především v situacích, kdy je požadován maximální výkon, nutné uvážit, jaké aplikace hodláte používat. Každý souborový systém má své výhody i nevýhody, které je nutné přitom brát v úvahu. Ani ten nejlepší souborový systém však nedokáže nahradit rozumné zálohování.

Termíny integrity dat nebo konzistence dat používané v této kapitole, nemají nic společného s konzistencí uživatelských dat (dat zapisovaných aplikacemi do souborů). Zda jsou data pro aplikace konzistentní, si kontrolují přímo aplikace.

---

### Důležité

#### Nastavení souborového systému

Všechna zde uvedená nastavení lze snadno provést pomocí programu YaST.

---

Důležité

## 20.2.1 Ext2

Historie Ext2 sahá až do počátečních dnů Linuxu. Jeho předchůdce Extended souborový systém byl implementován v dubnu roku 1992 v Linuxu 0.96c. Od té doby prošel Extended souborový systém celou řadou změn až k Ext2, nejpoužívanějšímu linuxovému souborovému systému. Z trůnu ho sesadil až příchod žurnálovacích souborů.

Ext2 neumožňuje dynamickou alokaci inodů. Znamená to, že datové bloky, do jsou data ukládána, jsou vždy stejně velké. Tato skutečnost může vést k nehospodárnému využívání diskového prostoru.

Základní přehled vlastností Ext2 vám pomůže porozumět tomu, proč byl tento souborový systém (a v některých oblastech stále ještě je) nejoblíbenějším linuxovým souborovým systémem.

**Spolehlivost** Od počátků svého vzniku Ext2 prošel celou řadou testů a zlepšení. To může být důvod, proč se jeví tak spolehlivým. Pokud systém není možné korektně odpojit, spustí se `e2fsck`, který začne kontrolovat data souborového systému. Metadata jsou spojována do konzistentního stavu a chybná nebo poškozená data nebo bloky dat jsou zapisována do příslušného souboru (nazývaného `lost+found`). Na rozdíl od žurnálovacích souborových systémů `e2fsck` nekontroluje jen pozměněná data, ale celý systém. To u dnešních disků samozřejmě zabere mnoho času. Protože však není nutné spravovat žurnály a používá mnohem méně paměti, je v některých případech rychlejší než ostatní souborové systémy.

**Jednoduchý upgrade** Souborový systém Ext2 tvoří z velké části podklad pro souborový systém další generace Ext3. Jeho spolehlivost byla elegantně zkombinována s výhodami žurnálování.

## 20.2.2 Ext3

Ext3 navrhl Stephen Tweedie. Na rozdíl od všech ostatních novějších souborových systémů není Ext3 založen na zcela nových základech. Jeho vývoj byl založen na Ext2. Tyto dva souborové systémy tak k sobě mají velmi blízko. Není proto problém vystavět Ext3 na již existujícím systému Ext2. Největší rozdíl, který tyto dva systémy odlišuje, je především podpora žurnálování v Ext3.

Ext3 nabízí tyto nejvýznamnější výhody:

**Jednoduchý upgrade z Ext2** Ext3 je založen na kódu Ext2 a sdílí s ním formát dat na disku. Z toho důvodu je přechod z Ext2 na Ext3 velmi jednoduchý. Obnova při poškození a kontrola tohoto systému je extrémně rychlá a bezpečná. Pokud z nějakého důvodu Ext3 nevyhovuje vašim požadavkům, není problém vrátit se zpět k Ext2. Downgrade je stejně jednoduchý jako upgrade. Stačí čistě odpojit souborový systém Ext3 a pak ho připojit jako Ext2.

**Spolehlivost a výkon** Naprostá většina žurnálovacích souborů je metadata-only. To znamená, že metadata jsou vždy udržována v konzistentním stavu, což ale není vždy garancí konzistentnosti samotných dat souborového systému. Ext3 je navržen tak, aby se staral jak o metadata tak o samotná data. Stupeň této péče lze nastavit. Povolení Ext3 v režimu *data=journal* poskytuje maximální bezpečnost (integritu dat), ale žurnálování dat i metadat může vést k výraznému zpomalení systému. Jednou z novějších záležitostí je režim *data=ordered*, který zajišťuje integritu dat i metadata, ale žurnálování provádí pouze u metadat. Ovladač souborového systému sbírá všechny bloky dat, které náleží k určitému updatu metadat. Tyto bloky jsou seskupovány do transakcí a ty jsou pak před updatem metadat zapsány na disk. Výsledkem je zajištění konzistence dat i metadat bez viditelného zvýšení zatížení systému. Třetí volbou je režim *data=writeback*, který umožňuje zapsat data po zapsání metadat do žurnálu. Tato volba vykazuje nejlepší hodnoty při měření výkonu. Zároveň dokáže zajistit obnovu dat při narušení integrity souborového systému. Pokud pro Ext3 nenastavíte žádný režim, použije se *data=ordered*.

Přechod z Ext2 na Ext3 na již existujícím systému se skládá ze dvou kroků:

**Žurnály** Přihlaste se jako `root` a zadejte příkaz `tune2fs -j`. Tak vytvoříte žurnál Ext3 s výchozími parametry. Pokud chcete nastavit délku žurnálu, zadejte místo předešlého příkazu příkaz `tune2fs -J` spolu s volbami *size=* a *device=*. Více informací o programu `tune2fs` najdete v jeho manuálové stránce (*man 8 tune2fs*).

### Nastavení typu souborového systému v `/etc/fstab`

Aby byl Ext3 správně rozpoznáván, je nutné ho uvést v souboru `/etc/fstab`. U položky diskového oddílu, u které jsme souborový systém změnili, musíte změnit typ souborového systému z *ext2* na *ext3*. Změna se projeví po restartu počítače.

### 20.2.3 ReiserFS

Ten souborový systém byl jednou z hlavních novinek jádra 2.4. Pro SUSE jádra předcházející řady 2.2.x byl dostupný jako jaderný patch od verze 6.4. ReiserFS vznikl díky Hansi Reiserovi a týmu vývojářů společnosti Namesys jako alternativa ke starému Ext2. Může se pochlubit lepším využitím disku, rychlejším přístupem a mnohem lepší a rychlejší opravou dat. Zaměřuje se však na péči o metadata, ale ne o samotná data. Následující verze vy již měly obsahovat také datové žurnálování (do žurnálu jsou zapisovány informace o metadatach i aktuálních datech).

Výhody souborového systému ReiserFS:

**Lepší využití disku** V ReiserFS jsou všechna data organizována ve strukturách nazývaných B<sup>\*</sup> stromy. Stromová struktura umožňuje lepší využití disku, protože malé soubory lze umístit přímo do listu stromu, místo rozmístění po celém disku a spravovat pak ukazatele na umístění dat. Data navíc nejsou umísťována do bloků s pevnou velikostí (obvykle 1 nebo 4 kB), ale do bloků potřebné velikosti. Další výhoda ReiserFS spočívá v dynamickém alokování inodů. To umožňuje oproti starším systémům vyšší flexibilitu.

**Vyšší diskový výkon** U malých souborů najdete informace o datech souboru a stat\_data (inode) vedle sebe. Lze je přečíst jednou jednoduchou diskovou IO operací, což znamená, že je potřeba pouze jeden přístup na disk.

**Rychlá obnova po poškození** V případě havárie počítače a poškození souborového systému lze souborový systém ve většině případů opravit během několika sekund. Žurnálování také urychluje pravidelné kontroly konzistence souborového systému.

**Žurnálování** ReiserFS podporuje také žurnálování podobné tomu popsanému v části věnované Ext3 section, 20.2.2 na straně 347. Výchozí režim je data=ordered. tento režim zajišťuje jak integritu metadata, tak samotných dat, ale žurnálování používá pouze u metadata.

### 20.2.4 Reiser4

Krátce po vydání jádra 2.6 se rodina žurnálovacích souborových systémů rozšířila o nového člena: Reiser4. Reiser4 je od svého předchůdce (version 3.6) zcela odlišný. Představuje koncept modulů vylepšujících funkčnost souborového systému a vylepšenou bezpečnost.

**Bezpečnostní koncept** Při návrhu souborového systému Reiser4 věnovali vývojáři zvláštní pozornost funkcím spjatým s bezpečností. Reiser4 je proto vybaven řadou bezpečnostních modulů. Jedním z nejvýznamnějších jsou „položky“ souboru. V současnosti jsou ACLs definovány pro každý soubor. V systému s velkým počtem souborů každý soubor obsahuje potřebné informace o právech každého uživatele, skupiny či aplikace. V systému Reiser4 jsou tyto soubory rozděleny do menších jednotek („položky“). Přístupová práva mohou být pro každou položku a uživatele nastavena zvlášť, čímž je umožněna mnohem lepší správa přístupu. Jako příklad může posloužit soubor `/etc/passwd`. Práva zápisu do tohoto souboru má pouze uživatel `root`, ostatní uživatelé mají jen práva pro čtení. S využitím položek souborového systému Reiser4 můžete rozdělit soubor na řadu položek (jednu pro každého uživatele), takže uživatel může editovat vlastní data, ale nesmí měnit data ostatních uživatelů. Tento koncept sebou přináší jak vyšší bezpečnost, tak také pružnost.

#### **Rozšiřitelnost prostřednictvím modulů**

V systému Reiser4 je řada běžných i rozšířených funkcí prováděna moduly. Lze je snadno přidávat, takže není nutné kvůli nové funkci kompilovat nové jádro nebo formátovat disk.

#### **Lepší výkon souborového systému díky delayed alokaci**

Stejně jako u XFS podporuje Reiser4 delayed alokaci viz 20.2.6 na následující straně.

## **20.2.5 JFS**

*JFS Journaling file system* byl navržen společností IBM. První testovací verze JFS se v linuxové komunitě objevila na jaře roku 2000. Verze 1.0.0 vyšla roku 2001. JFS byl navržen pro výkonné servery a proto byl velký důraz kladen na jeho výkonnost. Jako plně 64 bitový souborový systém, JFS podporuje větší velikost souborů i oddílů.

Vlastnosti JFS:

#### **Výkonné žurnálování**

JFS klade stejně jako ReiserFS důraz pouze na metadata. Stejně jako ReiserFS při opravě kontroluje pouze změny v metadatech, což vede k vysoké úspoře času. Konkurenční operace vyžadují současně záznam lze spojit do jedné skupiny a tak vícenásobnými operacemi zápisu redukovat ztráty výkonu.

**Vynikající organizace adresářů** JFS používá dva typy organizace adresářů. Pro malé adresáře umožňuje ukládání obsahu přímo v inodu. U větších adresářů používá B<sup>+</sup> stromy.

#### **Lepší využití prostoru díky dynamické alokaci inodů**

JFS šetří váš čas — inody jsou alokovány automaticky.

## **20.2.6 XFS**

Původně společnost SGI spustila vývoj tohoto systému na začátku roku 1990 pro svůj operační systém IRIX OS. XFS měl být výkonným 64-bitovým žurnálovacím souborovým systémem určeným pro ty nejnáročnější výpočetní úlohy. XFS dosahuje vynikajících výsledků při práci s velkými soubory a špičkovým hardwarem. Stejně jako jiné žurnálovací systémy jako např. ReiserFS však kontroluje pouze integritu metadat.

Rychlý pohled na hlavní vlastnosti XFS ukáže, proč je tak dobrým souborovým systémem pro náročné výpočetní úlohy:

#### **Vysoká stabilita díky využití alokačních skupin**

Při vytvoření souborového systému XFS je souborový systém rozdělen do osmi nebo více lineárních částí stejné velikosti. Ty jsou označovány jako alokační skupiny. Na alokační skupiny lze pohlížet jako na souborový systém v souborovém systému. Jednotlivé alokační skupiny na sobě nejsou nijak závislé, takže jádro může současně adresovat několik skupin najednou. Tato funkce pak vede k vysokému výkonu souborového systému XFS.

#### **Vysoký výkon podpořený účinnou správou diskového prostoru**

Volný prostor a inody jsou spravovány B<sup>+</sup> stromy vně alokačních skupin. Využívání B<sup>+</sup> stromů zvyšuje výkon. S XFS je spojena funkce delayed alokace. XFS při alokaci dělí proces na dvě části. Transakce jsou uloženy v RAM a je pro ně rezervována předpokládaná velikost prostoru. XFS nerozhoduje, kde přesně budou data uložena (bloky souborového systému). Toto rozhodnutí je odloženo na poslední možnou chvíli. Některá data se tak vůbec nedostanou na disk, protože dříve než XFS rozhodne o jejich uložení, zastarají. Tímto způsobem je zvyšován výkon při zápisu a redukována fragmentace souborového systému. Vzhledem ke strategii delayed alokace je však XFS mnohem náchylnější ke ztrátám dat při pádu systému než jiné souborové systémy.

### Prelokace souborového systému jako prevence fragmentace

Před zápisem dat do souborového systému, XFS rezervuje (prelokuje) volný prostor potřebný pro soubor. Tak je maximálně redukována fragmentace souborového systému. Zároveň dojde ke zvýšení výkonu, protože jednotlivé soubory nejsou rozmístěny po celém souborovém systému.

## 20.3 Některé další podporované souborové systémy

Následující tabulka shrnuje některé další souborové systémy podporované Linuxem. Jedná se především o takové souborové systémy, které jsou podporovány z důvodů kompatibility s jinými systémy nebo typy médií.

*Tabulka 20.1: Typy souborových systému v Linuxu*

cramfs	<i>Komprimovaný souborový systém ROM souborový systém: systém pouze ke čtení.</i>
hpfs	<i>High Performance souborový systém: IBM OS/2 standard souborový systém — systém pouze ke čtení.</i>
iso9660	Standardní souborový systém na CD.
minix	První linuxový souborový systém používaný v Linuxu. Dnes se používá prakticky pouze pro diskety s ovladači.
msdos	<i>fat</i> , souborový systém používaný systémem DOS. Dnes je používán řadou dalších operačních systému.
ncpfs	souborový systém pro připojení svazků Novellu přes síť.
nfs	<i>Síťový souborový systém: Síťový souborový systém umožňuje uložení dat na jednom počítači, na který pak mohou přes síť přistupovat uživatelé z jiných počítač.</i>
smbfs	<i>Server Message Block: síťový souborový systém umožňující přístup po síti používaný systémy Windows.</i>
sysv	Používaný systémy SCO UNIX, Xenix a komerční unixové systémy pro PC.



ufs	Používaný systémy BSD, SunOS a NeXTstep. Podporuje pouze režim <i>read-only</i> .
umsdos	<i>UNIX na MSDOS</i> : aplikovaný na normálním <i>fat</i> souborovém systému. Unixové funkčnosti (přístupová práva, odkazy, dlouhá jména souborů) dosahuje vytvářením zvláštních souborů.
vfat	Virtual FAT: rozšíření souborového systému <i>fat</i> (podporuje dlouhá jména souborů).
ntfs	<i>Windows NT souborový systém</i> , pouze ke čtení.

## 20.4 Podpora souborů větších než 2 GB

Původně podporovaná maximální velikost linuxového souboru je 2 GB. Před příchodem multimediálních souborů a rozsáhlých databází se tato velikost zdála dostatečná. Především velmi rychlý rozmach digitálního zpracování médií sebou přinesl nutnost poupravit jádro a knihovnu C tak, aby bylo možné pracovat také se soubory většími než 2 GB. V současné době již LFS podporují prakticky všechny novější souborové systémy.

Následující tabulka poskytuje přehled současných omezení velikostí linuxových souborů a souborových systémů v jádrech řady 2.4.

*Tabulka 20.2: Maximální velikost souborových systémů*

Souborový systém	Velikost souboru [Byte]	Velikost souborového systému [Byte]
Ext2 or Ext3 (velikost bloku 1 kB)	$2^{34}$ (16 GB)	$2^{41}$ (2 TB)
Ext2 or Ext3 (velikost bloku 2 kB)	$2^{38}$ (256 GB)	$2^{43}$ (8 TB)
Ext2 or Ext3 (velikost bloku 4 kB)	$2^{41}$ (2 TB)	$2^{44}$ (16 TB)
Ext2 or Ext3 (velikost bloku 8 kB) (systémy s 8 kB stránkami jako Alpha)	$2^{46}$ (64 TB)	$2^{45}$ (32 TB)

ReiserFS v3	$2^{46}$ (64 GB)	$2^{45}$ (32 TB)
XFS	$2^{63}$ (8 EB)	$2^{63}$ (8 EB)
JFS (velikost bloku 512 bytů)	$2^{63}$ (8 EB)	$2^{49}$ (512 TB)
JFS (velikost bloku 4 kB)	$2^{63}$ (8 EB)	$2^{52}$ (4 PB)
NFSv2 (na straně klienta)	$2^{31}$ (2 GB)	$2^{63}$ (8 EB)
NFSv3 (na straně klienta)	$2^{63}$ (8 EB)	$2^{63}$ (8 EB)

---

## Důležité

### Omezení linuxového jádra

Existují také omezení jádra:

V tabulce 20.2 na předchozí straně najdete omezení velikosti disku.

Jádro 2.6 má následující vlastní omezení velikosti souborových systémů a souborů:

**Velikost souboru** Na 32 bitových systémech nemohou být soubory větší než 2 TB (241 bytů).

**Velikost souborového systému** Souborové systémy mohou být velké 2 na 73 73 bytů. Tohoto limitu v současné době ani reálně nelze kvůli omezením hardwaru dosáhnout.

## Důležité

## 20.5 Další informace

Každý z uvedených souborových systémů je spravován vlastním projektem, který má vlastní internetové stránky obsahující veškerou dostupnou dokumentaci a také emailovou konferenci.

- <http://e2fsprogs.sourceforge.net/>
- <http://www.zipworld.com.au/~akpm/linux/ext3/>

- <http://www.namesys.com/>
- <http://oss.software.ibm.com/developerworks/opensource/jfs/>
- <http://oss.sgi.com/projects/xfs/>

Srovnávací tutoriál linuxových souborových systému najdete na stránkách *IBM developerWorks*:

<http://www-106.ibm.com/developerworks/library/l-fs.html>

Srovnání linuxových žurnálovacích souborových systémů najdete v článku od Juan I. Santos Florido uveřejněného v *Linuxgazette*:

<http://www.linuxgazette.com/issue55/flneboido.html>.

Pokud byste rádi získali další informace o LFS v Linuxu, doporučujeme vám stránky Andrease Jaegera: [http://www.suse.de/aj/linux\\_lfs.html](http://www.suse.de/aj/linux_lfs.html).



# Autentizace pomocí PAM

Linux používá PAM (Pluggable Authentication Modules – připojovatelné autentizační moduly) při procesu autentizace jako zprostředkující vrstvu mezi uživatelem a aplikací. PAM moduly jsou dostupné v celém systému, takže mohou být použity libovolnou aplikací. Tato kapitola se věnuje popisu funkce modulárního autentizačního mechanismu a jeho konfiguraci.

21.1	Struktura PAM konfiguračního souboru . . . . .	358
21.2	Konfigurace PAM pro sshd . . . . .	360
21.3	Konfigurace PAM modulů . . . . .	362
21.4	Další informace . . . . .	364

Systémoví administrátoři a programátoři často potřebují omezit přístup k určitým částem systému nebo použití určitých funkcí aplikace. Bez využití PAM by aplikace musely být upraveny, kdykoliv je zaveden nový autentizační mechanismus (jako LDAP nebo SAMBA). To je však časově náročný a k chybám náchylný proces. Problémům se lze vyhnout oddělením aplikací od autentizačního procesu a převedením autentizační funkce na centrálně spravované moduly. Kdykoliv je pak potřeba zavést nový autentizační mechanismus, stačí upravit nebo napsat příslušné PAM moduly.

Každý program závislý na mechanismu PAM má svůj vlastní konfigurační soubor v adresáři `/etc/pam.d/<jmenoprogramu>`. Tyto soubory určují, jaké PAM moduly mají být použity při autentizaci. Navíc pro většinu PAM modulů existují globální konfigurační soubory uložené v adresáři `/etc/security` (např. `pam_env.conf`, `pam_pwcheck.conf`, `pam_unix2.conf`, `time.conf`). Ty určují přesné chování modulů. Každá aplikace používající PAM modul ve skutečnosti volá sadu PAM funkcí, které následně zpracují údaje v různých konfiguračních souborech a vrátí výsledek volající aplikaci.

## 21.1 Struktura PAM konfiguračního souboru

Každý řádek PAM konfiguračního souboru obsahuje nejvýše čtyři sloupce:

```
<Typ modulu> <Kontrolní příznak> <Jméno modulu> <Parametry>
```

Moduly PAM jsou zpracovávány postupně za sebou. Různé moduly mají různé účely. Jeden modul například kontroluje správnost hesla, jiný ověřuje umístění, z kterého je k systému přistupováno, a další načítá uživatelsky specifická nastavení. PAM obsahuje čtyři různé typy modulů:

**\mbx{auth}** Účelem modulu tohoto typu je autentizovat uživatele. Obvykle se tak činí ověřením hesla, ale lze toho dosáhnout i s pomocí čipových karet nebo biometrie (otisků prstů či rozpoznání oční duhovky).

**\mbx{account}** Moduly tohoto typu ověřují, zda má uživatel obecné oprávnění využít příslušnou službu. Například lze s jejich pomocí zajistit, aby se k systému nemohl přihlásit nikdo pod uživatelským jménem, jehož účet vypršel.

**\mbox{password}** Smyslem tohoto typu modulu je umožnit změnu autentizačního tokenu. Tímto tokenem je ve většině případů heslo.

**\mbox{session}** Moduly tohoto typu jsou zodpovědné za správu a konfiguraci uživatelských relací. Jsou spuštěny před a po autentizaci, aby zaznamenaly pokusy o přihlášení do systémových logů a nakonfigurovaly uživatelsky specifické prostředí (poštovní účty, domovský adresář, systémová omezení atd.).

Druhý sloupec obsahuje kontrolní příznaky, které ovlivňují chování spuštěných modulů:

**\mbox{required}** Modul s tímto příznakem musí být úspěšně zpracován dříve, než proběhne autentizace. Selže-li modul s příznakem `required`, musí být zpracovány všechny ostatní moduly se stejným příznakem dříve, než je uživatel informován o neúspěšnosti pokusu o autentizaci.

**\mbox{requisite}** Moduly s tímto příznakem musí být, stejně jako moduly s příznakem `required`, úspěšně zpracovány. Nicméně v případě selhání modulu s příznakem `requisite` je uživatel okamžitě informován a nejsou zpracovávány žádné další moduly. Pokud je zpracování úspěšné, jsou zpracovávány i další moduly, stejně jako v případě modulů s příznakem `required`. Příznak `requisite` lze použít jako základní filtr pro ověření podmínek nezbytných pro korektní autentizaci.

**\mbox{sufficient}** Pokud je úspěšně zpracován modul s tímto příznakem, dostane volající aplikace okamžitou zprávu o úspěšnosti autentizace a žádné další moduly nejsou zpracovávány. Platí to však jen tehdy, pokud již dříve nedošlo k selhání modulu s příznakem `required`. Selhání modulu s příznakem `sufficient` nemá žádné přímé důsledky, všechny další moduly jsou zpracovávány v běžném pořadí.

**\mbox{optional}** Úspěch ani selhání modulu s tímto příznakem nemá žádné přímé důsledky. Toho se využívá v případě modulů, jejichž jediným účelem je zobrazit zprávu (například oznámení o příchozí poště).

**include** Tento příznak slouží ke vložení souboru udaného jako argument.

Pokud se modul nachází v implicitním adresáři `/lib/security` (`/lib64/security` na 64-bitových platformách se systémem SUSE LINUX), nemusí být cesta explicitně stanovena. Čtvrtý sloupec může obsahovat parametry předávané modulu, jako např. `debug` (umožňuje ladění programu) nebo `nullok` (dovoluje použití prázdných hesel).

## 21.2 Konfigurace PAM pro sshd

Následující praktický příklad ukazuje konfiguraci PAM pro sshd:

### *Příklad 21.1: Konfigurace PAM pro sshd*

```
%PAM-1.0
auth    include      common-auth
auth    required      pam_nologin.so
account include      common-account
password include      common-password
session include      common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README.SuSE)
#session optional    pam_resmgr.so fake_ttyname
```

Typická PAM konfigurace aplikace (v našem případě sshd) obsahuje čtyři vkládací příkazy (`include`) odkazující na konfigurační soubory čtyř typů modulů: `common-auth`, `common-account`, `common-password` a `common-session`. Tyto čtyři soubory obsahují výchozí konfiguraci pro každý typ modulů. Toto vkládání zajišťuje automatické použití aktuálního výchozího nastavení. Dříve bylo třeba všechny konfigurační soubory pro všechny aplikace upravit ručně, kdykoli došlo k aktualizaci PAM. Nyní existuje centrální konfigurace; jsou-li v ní provedeny změny, automaticky se dědí PAM konfiguracemi jednotlivých služeb. První vkládaný soubor (`common-auth`) volá dva moduly typu `auth`: `pam_env` a `pam_unix2`. Viz 21.2 na této straně.

### *Příklad 21.2: Výchozí konfigurace pro auth sekci*

```
auth    required      pam_env.so
auth    required      pam_unix2.so
```

První z nich, `pam_env`, nahraje soubor `/etc/security/pam_env.conf` a nastaví proměnné prostředí specifikované v tomto souboru. To lze využít k nastavení proměnné `DISPLAY` na správnou hodnotu, neboť modul `pam_env` zná místo, ze kterého probíhá přihlašování. Druhý, `pam_unix2`, zkontroluje přihlašovací jméno a heslo podle `/etc/passwd` a `/etc/shadow`.

Po úspěšném zavolání modulů z `common-auth` zkontroluje třetí modul, `pam_nologin`, zda existuje soubor `/etc/nologin`. Pokud existuje, nesmí se přihlásit



nikdo kromě superuživatele `root`. Všechny `auth` moduly jsou zpracovány dříve než `sshd` dostane informaci o výsledku přihlašování. Protože všechny `auth` moduly mají příznak `required`, musí být všechny úspěšně zpracovány před tím, než `sshd` dostane zprávu o výsledku autentizace. Pokud některý z modulů selže, stejně musí být zpracována celá sada, a teprve potom `sshd` dostane zprávu o negativním výsledku.

Jakmile jsou všechny `auth` moduly úspěšně zpracovány, přijde na řadu další vkládací (`include`) příkaz, tentokrát ten, který je uvedený v 21.3 na této straně. Soubor `common-account` obsahuje jen jeden modul, `pam_unix2`. Pokud `pam_unix2` zjistí, že uživatel existuje, dostane `sshd` zprávu o úspěchu a je zpracována další sada modulů (`password`) – viz 21.4 na této straně.

### *Příklad 21.3: Výchozí konfigurace pro `account` sekci*

```
account required      pam_unix2.so
```

### *Příklad 21.4: Výchozí konfigurace pro `password` sekci*

```
password required      pam_pwcheck.so  nullok
password required      pam_unix2.so    nullok use_first_pass use_authtok
#password required     pam_make.so     /var/yp
```

PAM konfigurace `sshd` zahrnuje pouze vkládací (`include`) příkaz odkazující na výchozí konfiguraci `password` modulů v souboru `common-password`. Tyto moduly se musí úspěšně zpracovat (příznak `required`) kdykoliv aplikace vyžaduje změnu autentizačního tokenu. Změna hesla či jiného tokenu vyžaduje bezpečnostní kontrolu. Tu zajišťuje modul `pam_pwcheck`.

Po něm použitý modul `pam_unix2` přenáší hesla z modulu `pam_pwcheck`, takže se uživatel nemusí znovu autentizovat. Také tím znemožňuje obejít kontroly prováděné modulem `pam_pwcheck`. Moduly typu `password` by měly být používány vždy, když jsou moduly `account` či `auth` nakonfigurovány tak, aby upozorňovaly na vypršení hesla.

### *Příklad 21.5: Výchozí konfigurace pro `session` sekci*

```
session required      pam_limits.so
session required      pam_unix2.so
```

Jako poslední krok jsou volány moduly typu `session` z `common-session`, jejichž úkolem je nastavit relaci pro konkrétního uživatele. Opětovné použití modulu `pam_unix2` nemá žádné praktické důsledky, neboť je volán s parametrem `none`, který je nastaven v konfiguračním souboru tohoto modulu (`pam_unix2.conf`). Modul `pam_limits` zpracovává soubor `/etc/security/limits.conf`, ve kterém mohou být definována omezení pro využívání určitých systémových zdrojů. Moduly typu `session` jsou volány podruhé při odhlášení uživatele.

## 21.3 Konfigurace PAM modulů

Některé PAM moduly jsou konfigurovatelné. Příslušné konfigurační soubory jsou umístěny v adresáři `/etc/security`. Tato kapitola stručně popisuje konfigurační soubory vztahující se k předchozímu příkladu s `sshd`, tj. `pam_unix2.conf`, `pam_env.conf`, `pam_pwcheck.conf` a `limits.conf`.

### 21.3.1 `pam_unix2.conf`

Běžná autentizace založená na heslech je řízená PAM modulem `pam_unix2`. Ten může přistupovat k potřebným údajům v `/etc/passwd`, `/etc/shadow`, NIS mapách, NIS+ tabulkách nebo v LDAP databázi. Chování modulu lze ovlivnit individuálním nastavením PAM pro jednotlivé aplikace nebo globálně úpravou souboru `/etc/security/pam_unix2.conf`. Velmi jednoduchý konfigurační soubor pro tento modul ukazuje příklad 21.6 na této straně.

#### *Příklad 21.6: `pam_unix2.conf`*

```
auth:    nullok
account:
password:    nullok
session:    none
```

Parametr `nullok` pro moduly `auth` a `password` znamená, že jsou povolena prázdná hesla. Uživatelé také mohou měnit hesla ke svým účtům. Parametr `none` modulu typu `session` znamená, že nebudou logovány žádné zprávy modulu (to je implicitní nastavení). Další konfigurační možnosti jsou popsány v komentářích v samotném souboru a v manuálové stránce `pam_unix2(8)`.

### 21.3.2 pam\_env.conf

Tento soubor lze použít k nastavení standardizovaného uživatelského prostředí, kdykoliv je zavolán modul `pam_env`. Proměnné prostředí lze nastavit pomocí následující syntaxe:

```
VARIABLE [DEFAULT=[hodnota]] [OVERRIDE=[hodnota]]
```

**VARIABLE** Jméno proměnné prostředí, která má být nastavena.

**[DEFAULT=[hodnota]]** Implicitní hodnota proměnné.

**[OVERRIDE=[hodnota]]** Hodnota, na kterou se modul `pam_env` dotáže a kterou přepíše implicitní hodnotu.

Obvyklým příkladem implicitní hodnoty, jež má být modulem `pam_env` přepsána, je proměnná `DISPLAY`, která se mění při každém vzdáleném přihlášení. Viz příklad 21.7 na této straně.

#### *Příklad 21.7: pam\_env.conf*

```
REMOTEHOST    DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY       DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

První řádka nastavuje proměnnou `REMOTEHOST` na hodnotu `localhost`. Tato hodnota je použita, pokud modul `pam_env` nemůže zjistit jinou hodnotu. Proměnná `DISPLAY` obsahuje hodnotu proměnné `REMOTEHOST`. Další informace lze získat z komentářů v souboru `/etc/security/pam_env.conf`.

### 21.3.3 pam\_pwcheck.conf

Tento konfigurační soubor je určen pro modul `pam_pwcheck`, který z něj načítá nastavení pro všechny moduly typu `password`. Nastavení z tohoto souboru jsou načtena před PAM nastaveními pro jednotlivé aplikace. Pokud nemá aplikace nastavení definováno specificky, použije se toto globální nastavení. Příklad 21.8 na této straně přikazuje modulu `pam_pwcheck` povolit prázdná hesla a jejich změnu. Více nastavení je zmíněno v souboru `/etc/security/pam_pwcheck.conf`.

#### *Příklad 21.8: pam\_pwcheck.conf*

```
password:      nullok
```

### 21.3.4 limits.conf

V souboru `limits.conf`, který je načítán modulem `pam_limits`, lze nastavit systémová omezení pro jednotlivé uživatele nebo jejich skupiny. Umožňuje nastavit pevná omezení, která nelze v žádném případě překročit, a měkká omezení, která mohou být překročena dočasně. Syntaxe souboru a další možnosti nastavení jsou popsány v komentářích.

## 21.4 Další informace

V adresáři `/usr/share/doc/packages/pam` naleznete následující dokumentaci:

**Soubory README** V kořenu adresáře jsou obecně zaměřené README dokumenty. Podadresář `modules` obsahuje README dokumenty zabývající se dostupnými PAM moduly.

### **The Linux-PAM System Administrators' Guide**

Tento dokument obsahuje vše, co by měl systémový administrátor o PAM vědět. Zabývá se širokým okruhem témat, od syntaxe konfiguračních souborů, až po bezpečnostní aspekty. Dokument je dostupný ve formátech PDF, HTML a jako prostý text.

### **The Linux-PAM Module Writers' Manual**

Tento dokument shrnuje PAM moduly z pohledu vývojáře. Poskytuje informace o vývoji PAM modulů v souladu se standardy. Je dostupný ve formátech PDF, HTML a jako prostý text.

### **The Linux-PAM Application Developers' Guide**

Tato příručka obsahuje vše, co potřebuje znát vývojář aplikací používajících PAM knihovny. Je dostupný ve formátech PDF, HTML a jako prostý text.

Thorsten Kukuk napsal množství PAM modulů pro SUSE LINUX a některé informace o nich zveřejnil na adrese: <http://www.suse.de/~kukuk/pam/>

**Část III**

**Služby**



# Základy síťování

Linux je dítě Internetu. Nabízí proto samozřejmě všechny potřebné funkce pro integraci do všech typů sítí. Linuxový protokol TCP/IP má řadu funkcí a poskytuje řadu služeb, které zde popisujeme. Přístup k síti pomocí síťové karty, modemu nebo jiného zařízení lze nakonfigurovat nástrojem YaST. Je možná i manuální konfigurace. V této kapitole jsou popsány pouze základní síťové mechanismy a konfigurace.

22.1	IP adresy a směrování . . . . .	370
22.2	IPv6 – Internet další generace . . . . .	373
22.3	Překlad jmen . . . . .	381
22.4	Konfigurace síťového připojení pomocí YaST . . . . .	382
22.5	Manuální konfigurace sítě . . . . .	392
22.6	smpppd jako pomocník s vytáčeným připojením . . . . .	402

Linux a jiné unixové operační systémy používají především tzv. TCP/IP protokol. V tomto případě se nejedná o jeden, ale o celou skupinu síťových protokolů, která poskytuje různé služby. Protokoly uvedené v tabulce 22.1 na této straně slouží k výměně dat mezi dvěma stroji přes TCP/IP. TCP/IP sítě tvoří navzájem provázanou celosvětovou síť známou pod jménem Internet.

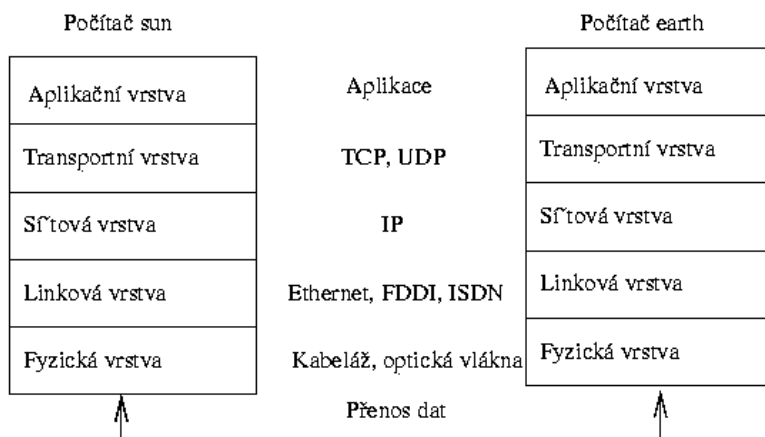
RFC dokumenty (Request for comments) popisují různé internetové protokoly a související procedury operačního systému a aplikací. Pokud si tedy chcete prohloubit své znalosti o určitém protokolu, pak je pro vás odpovídající RFC dokument to pravé. RFC naleznete na internetové adrese <http://www.ietf.org/rfc.html>

**Tabulka 22.1:** Různé protokoly z rodiny TCP/IP

Protokol	Popis
TCP	(angl. <i>Transmission Control Protocol</i> ) Spojovací zabezpečený protokol. Přenášena data jsou aplikací odesílána jako datový tok a samotný operační systém je upravuje do formátu vhodného pro přenos. Data pak přichází cílové aplikaci opět ve formě datového toku tak, jak byla odeslána. TCP zajišťuje, že se po cestě žádná data neztratí. TCP se používá tam, kde je důležité pořadí dat.
UDP	(angl. <i>User Datagram Protocol</i> ) Nezabezpečený protokol. Data jsou odesílána ve formě paketů. Není garantováno pořadí příchodu dat příjemci a stejně tak se může stát, že se některé pakety ztratí. UDP se hodí pro datově orientované aplikace (např. přenos multimédií) a nemá žádné prodlevy způsobené ověřováním tak, jak je tomu u TCP.
ICMP	(angl. <i>Internet Control Message Protocol</i> ) Jedná se o servisní protokol, který sděluje stav chyb a řídí chování počítačů při přenosu TCP/IP dat. Navíc podporuje ICMP echo režim, který používá program ping.
IGMP	(angl. <i>Internet Group Management Protocol</i> ) Tento protokol řídí chování počítačů při IP multicast. Naneštěstí IP multicast přesahuje rozsah této publikace.



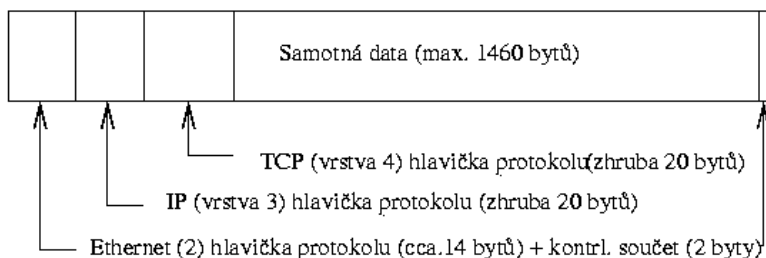
Jak je vidět v tabulce 22.1 na této straně, výměna dat probíhá v několika vrstvách. Vlastní síťová vrstva představuje nezabezpečený přenos dat pomocí IP (angl. *Internet Protocol*). Nad IP je TCP (angl. *Transmission Control Protocol*), který, do jisté míry, zajišťuje bezpečnost přenášených dat. IP sám je zase nadstavbou hardwarového protokolu, např. Ethernetu.



**Obrázek 22.1:** Zjednodušený model vrstev TCP/IP

Takřka všechny hardwarové protokoly jsou paketově orientovány. Je tedy třeba přenášená data zabalit do malých paketů a není možné posílat vše v jednom. Proto také TCP/IP pracuje s menšími datovými jednotkami. Maximální velikost jednoho TCP/IP paketu je skoro 64 KB (kilobytů). Obvykle jsou tyto pakety značně menší, protože limitujícím faktorem je síťový hardware. Takže např. maximální velikost datových paketů v Ethernetu je zhruba 1500 bytů. Tomu také odpovídá velikost TCP/IP paketů, pokud jsou data posílána přes Ethernet. Pokud posíláte větší objem dat, musí je operační systém rozdělit do více paketů a ty pak poslat.

Aby mohla každá vrstva plnit přidělenou funkci, musí přidat doplňující informace do paketu. Ty jsou uloženy v *hlavičce* paketu. Každá vrstva připojí malý blok dat, tzv. hlavičku protokolu (angl. *protocol header*). Paket v ethernetové síti může vypadat jako na obrázku 22.2 na následující straně. Kontrolní součet je umístěn na konci paketu, ne na začátku. To usnadňuje život hardwaru.



*Obrázek 22.2: TCP/IP paket v Ethernetu*

Pokud chce nějaká aplikace posílat data přes síť, pak proběhnou data jednotlivými vrstvami, které jsou (s výjimkou hardwarové vrstvy) implementovány do linuxového jádra. Každá z vrstev upraví data tak, aby mohla být předána níže položené vrstvě. Nejnižší vrstva je pak zodpovědná za posílání dat. Při příjmu dat probíhá to samé, ale v opačném gardu. Paket je zde loupán jako cibule a v každé vrstvě jsou odstraňovány hlavičky protokolu. Čtvrtá vrstva pak připravuje data pro aplikaci na cílovém počítači. Přitom komunikuje každá vrstva pouze s vrstvou přímo nad, resp. pod ní. Aplikace se tedy nemusí starat o to, zda data půjdou přes 100 MB FDDI síť nebo 56 kbit vytáčenou linku. Stejně tak je např. transportní vrstvě jedno, zda jsou posílaná data správně zabalena.

## 22.1 IP adresy a směrování

Následující část je věnována protokolu IPv4. Informace o IPv6 naleznete v části 22.2 na straně 373.

### 22.1.1 IP adresa

Každý počítač v internetové síti má jednoznačnou 32bitovou (4 bajty) adresu. Ta může vypadat jako v příkladu 22.1 na této straně

*Příklad 22.1: Zápis IP adres*

```
IP adresa (binárně):   11000000 10101000 00000000 00010100
IP adresa (decimálně): 192.      168.      0.      20
```

Tyto čtyři byty jsou v desítkové soustavě odděleny tečkou. IP adresa je přiřazena každému počítači, resp. každému síťovému rozhraní, takže už nemůže být použita v jakémkoliv jiném počítači na celém světě. Sice existují výjimky z tohoto pravidla, ale zde nehrají žádnou roli.

Také Ethernetové karty obsahují jednoznačnou adresu, tzv. MAC (angl. *Media Access Control*). Ta je 48 bitů dlouhá, celosvětově jedinečná a je výrobcem kartě jednoznačně přidělena. Má ale jeden obrovský nedostatek. MAC adresy netvoří hierarchický systém, ale jsou přidělovány víceméně náhodně. Není je proto možné používat pro adresování vzdálených počítačů. Rozhodující úlohu ale tyto adresy hrají při komunikaci počítačů v lokální síti (a jsou součástí hlavičky paketů pro druhou vrstvu).

A nyní zpět k IP adresám. Jak již napovídá výše uvedený text, tvoří IP adresy hierarchický systém. Do poloviny devadesátých let byly IP adresy pevně členěny do jednotlivých tříd. Tento systém se ukázal jako neflexibilní a proto se přestal používat. Používá se pouze směrování bez tříd (CIDR – Classless Inter Domain Routing).

## 22.1.2 Síťové masky a směrování

Protože počítač s IP adresou 192.168.0.1 nemůže vědět, kde se nachází počítač s IP adresou 192.168.0.20, byly zavedeny síťové masky. Zjednodušeně řečeno síťové masky sdělují počítači s IP adresou, co je uvnitř a co vně. Počítače, které se nacházejí uvnitř (ve stejné části počítačové sítě) spolu mohou komunikovat přímo. Při přístupu k počítačům nacházejícím se vně je třeba použít tzv. bránu (angl. *gateway*) nebo router. Protože má každé síťové rozhraní svou IP adresu, může to být poměrně komplikované.

Předtím, než se paket vydá na svou cestu, proběhne v počítači následující proces. Cílová adresa je se síťovou maskou binárně spojena pomocí operátoru AND. Také adresa odesílatele je spojena se síťovou maskou pomocí operátoru AND. Pokud je k dispozici více síťových rozhraní, pak jsou zpravidla ověřeny všechny adresy odesílatele. Výsledky spojení adres (AND) jsou pak porovnány. Pokud jsou tyto výsledky zcela shodné, nachází se cílový počítač ve stejné části sítě. V opačném případě je třeba použít bránu. To znamená, že čím více 1 bitů se nachází v síťové masce, tím méně počítačů je přímo dostupných. V následující tabulce je uvedeno několik příkladů:

*Příklad 22.2: Spojování IP adres se síťovou maskou*

```

IP adresa      (192.168.0.20):  11000000 10101000 00000000 00010100
sít'ová maska (255.255.255.0):  11111111 11111111 11111111 00000000
-----
výsledek  (binární):           11000000 10101000 00000000 00000000
výsledek  (decimální):         192.      168.      0.      0

IP adresa      (213.95.15.200): 11010101 10111111 00001111 11001000
sít'ová maska (255.255.255.0):  11111111 11111111 11111111 00000000
-----
výsledek  (binární):           11010101 10111111 00001111 00000000
výsledek  (decimální):         213.      95.      15.      0

```

Sít'ová maska se zapisuje, tak jako IP adresa, ve formě decimálních čísel oddělených tečkami. Protože má sít'ová maska také velikost 32 bitů, jsou jednotlivá čísla psána za sebe. Které počítače jsou bránou nebo které oblasti adres jsou přístupné přes které sít'ové rozhraní, je třeba nakonfigurovat.

A následuje další příklad – všechny počítače připojené na jeden ethernetový kabel se nacházejí *ve stejné části sítě* a jsou přímo přístupné. I když je v Ethernetu rozdělují tzv. switche a bridge, je možné k počítačům přistupovat přímo.

Pokud chcete překlenout delší vzdálenost, není již možné použít Ethernet. Pak je třeba IP pakety převést na jiný hardware (např. FDDI nebo ISDN). Taková zařízení se nazývají routery, resp. brány. Linuxový počítač může plnit i tyto úlohy, tato volba se označuje jako `ip_forwarding`.

Pokud je nakonfigurována brána, je paket poslán na odpovídající gateway. Ta se pak pokusí paket přeposlat dále. To se opakuje na každém dalším počítači tak dlouho, než paket dosáhne cílový počítač nebo vyprší jeho *životnost* TTL (angl. *time to live*).

**Tabulka 22.2:** Vyhrazené adresní prostory

Adresa	Popis
Základní sít'ová adresa	Sít'ová maska spojená (AND) s libovolnou adresou v síti, tedy výsledek z tabulky 22.2 na předchozí straně. Tuto adresu nelze přiřadit žádnému počítači.
Oznamovací adresa	Ta říká: hovoř se všemi počítači v této části sítě. Získá se binární inverzí sít'ové masky a spojením výsledku se základní sít'ovou adresou pomocí operace OR. Náš příklad vede k výsledku 192.168.0.255. Ani tato adresa nemůže být přiřazena žádnému počítači.

Lokální počítač

Adresa 127.0.0.1 odkazuje na každém počítači na tzv. loopback device. Pomocí této adresy je možné navázat spojení s vlastním počítačem.

Protože je třeba, aby byly IP adresy jedinečné, nemůžete si zvolit libovolné adresy. Abyste i přesto mohli postavit síť na bázi IP adres, existují tři oblasti, které můžete ihned použít. S těmito adresami se ale bez překladu adres nemůžete připojit k Internetu. Tyto adresové oblasti jsou definovány v RFC 1597 a jejich seznam si můžete prohlédnout v tabulce 22.3 na této straně.

**Tabulka 22.3:** Neveřejné adresní rozsahy

síť / síťová maska	oblast
10.0.0.0 / 255.0.0.0	10.x.x.x
172.16.0.0 / 255.240.0.0	172.16.x.x - 172.31.x.x
192.168.0.0 / 255.255.0.0	192.168.x.x

## 22.2 IPv6 – Internet další generace

Díky vynálezu WWW začal Internet, a tím i počet počítačů komunikujících pomocí TCP/IP, v posledních patnácti letech exponenciálně růst. Podle informací CERN (<http://public.web.cern.ch/>) vzrostl jejich počet z několika tisíc v roce 1990 na zhruba 100 000 000 v současnosti.

Jak již víte, má IP adresa pouze 32 bitů. Protože není z organizačních důvodů možné používat mnoho adres z 32 bitového adresního prostoru, je počet adres již nedostačující. Pouze pro připomenutí - Internet se skládá z podsítí, které jsou dále členěny. Ty se skládají vždy z mocniny dvou mínus 2 použitelných adres. Pokud tedy chcete připojit k Internetu 128 počítačů, pak potřebujete podsíť s 256 síťovými adresami, ze kterých můžete použít pouze 254 adres. Dvě adresy není možné použít, protože jedna je broadcast a druhá základní adresa sítě.

Aby se maximálně využívaly současné adresy v IPv4, používá se DHCP nebo NAT (angl. *Network Address Translation*). Tyto nástroje, spolu s veřejnými a neveřejnými adresními prostory, částečně řeší nedostatek adres. Nevýhodou těchto

metod je náročnější konfigurace, protože pro korektní nastavení počítače v IPv4 sítích potřebujete množství informací, jako je vlastní IP adresa, síťová maska, adresa brány a podle potřeby také nameserver. Všechny tyto informace musíte vědět.

S IPv6 je omezený adresní prostor a komplikovaná konfigurace minulostí. V následujících odstavcích si přiblížíme základní přednosti IPv6 a způsob přechodu od starého k novému protokolu.

## 22.2.1 Přednosti IPv6

Největší výhodou nového protokolu je enormní rozšíření adresního prostoru, protože IPv6 obsahuje místo 32bitových adres 128bitové adresy.

IPv6 adresy se neliší od svých předchůdců pouze délkou, ale také vnitřní strukturou, která obsahuje informace o systému a síti. Více v části 22.2.2 na následující straně.

Dalšími důležitými přednostmi nového protokolu jsou:

**Automatická konfigurace** IPv6 zavádí v síťování princip *Plug and Play*, protože nový systém se do lokální sítě integruje bez nutnosti manuální konfigurace. Autokonfigurační mechanismus zjistí vlastní adresu z informací, které obdrží prostřednictvím ND (*Neighbor Discovery*) protokolu ze sousedních routerů. Tento proces nevyžaduje žádný zásah ze strany správce sítě a oproti DHCP v IPv4 sítích má tu výhodu, že není nutné udržovat centrální server.

**Mobilita** IPv6 umožňuje, aby jednomu síťovému rozhraní bylo přiděleno více adres. Tím pádem budete mít jako uživatel systému jednoduše přístup k různým sítím. Tuto funkci je možné porovnat s roamingem u mobilních telefonů. Pokud se nacházíte se svým mobilem v zahraničí, připojí se telefon automaticky k cizí síti. Je zcela jedno, kde jste. Máte zaručenou dostupnost prostřednictvím běžného telefonního čísla a můžete telefonovat v cizích sítích, jako by to byly domovské sítě.

**Bezpečná komunikace** Zatímco v IPv4 patří zabezpečení komunikace pouze mezi doplňkové funkce, obsahuje IPv6 IPSec pro bezpečnou komunikaci.

**Zpětná kompatibilita** Rychlý přechod celého Internetu na IPv6 není realistický. Proto je důležité, že obě verze mohou koexistovat v jednom systému. Koexistence obou je možná díky používání kompatibilních adres (IPv4 lze převést na IPv6). Je také možné použít různé tunely (viz část 22.2.3 na straně 379). Prostřednictvím tzv. *Dual-Stack-IP* je možná podpora obou protokolů na jednom systému. Každý z obou protokolů používá vlastní síťový stack, takže nikdy nedojde ke kolizi.

**Multicasting** Zatímco v IPv4 sítích posílají některé služby (např. SMB) své pakety prostřednictvím všesměrového vysílání všem počítačům v lokální síti, je v IPv6 dostupný zcela jiný způsob. Pomocí multicastu je možné komunikovat se skupinou počítačů, tedy ne nutně se všemi jako v případě broadcast. Která skupina to bude, záleží na aplikaci. Existují však i určité předdefinované skupiny, jako jsou *všechny nameservery* (angl. *all nameservers multicast group*) nebo *všechny routery* (angl. *all routers multicast group*).

## 22.2.2 Adresování v IPv6

Jak již bylo uvedeno, má současný IP protokol dvě výrazné nevýhody. První je blížící se nedostatek IP adres a druhým složitá správa routování, jejíž složitost stále narůstá. První problém odstraňuje IPv6 rozšířením adresního prostoru na 128 bitů. Řešení druhého problému leží v hierarchické adresní kultuře, sofistikovaných mechanismech pro přiřazování adresy v síti a možnosti používání více adres pro jedno rozhraní, které zajišťuje přístup do různých sítí (tzv. multiho-ming).

Existují tři důležité typy IPv6 adres:

**Unicast** Adresy tohoto typu patří právě jednomu síťovému rozhraní. Pakety s adresou tohoto typu jsou směrovány přímo na příjemce. Unicast adresy se používají pro komunikaci s jednotlivými počítači v lokální síti nebo Internetu.

**Multicast** Adresy tohoto typu odkazují na skupinu rozhraní. Pakety s touto adresou jsou doručeny všem členům skupiny. Multicast používají především různé síťové služby, aby komunikovaly s určitou skupinou počítačů.

**Anycast** Adresy tohoto typu odkazují na skupinu rozhraní. Pakety s adresou tohoto typu jsou odeslány členu skupiny, který je podle směrovacích protokolů nejbližší odesílateli. Anycast adresy se používají v případě, kdy je vyhledáván server poskytující určité síťové služby. Všechny servery určitého typu obdrží stejnou anycast adresu. Pokud tedy terminál vyžaduje službu, odpoví ten server, který je podle směrovacího protokolu počítače nejbližší. Pokud tento server neodpovídá, je kontaktován další nejbližší.

IPv6 adresa sestává z osmi bloků po 16ti bitech, které jsou odděleny dvojtečkou a jsou v hexadecimálním zápise. Počáteční nulové byty (v rámci bloku) je možné vypustit, uprostřed nebo na konci musí být zachovány. Více než čtyři nulové byty

za sebou je možné nahradit :: (tzv. *collapsing*). V každé adrese je však možné :: použít maximálně jednou. Příklad 22.3 na této straně obsahuje tři různé ekvivalentní zápisy.

**Příklad 22.3: Sample IPv6 Address**

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :      0 :      0 :      0 :      0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```

Každá část IPv6 adresy má definovaný význam. První byty tvoří prefix a vypovídají o typu adresy. Prostřední část adresuje síť nebo je bez významu. Konec adresy tvoří tzv. host část. Síťová maska se určuje v IPv6 délkou prefixu a zapisuje se za lomítko na konci adresy. Adresa zobrazená v příkladu 22.4 na této straně obsahuje informaci, že prvních 64 bitů tvoří síťovou část adresy a posledních 64 bitů část týkající se počítače. Jinými slovy, 64 značí, že je síťová maska tvořena 64 1-bitovými hodnotami z levé části. Stejně jako v případě IPv4 je IP adresa kombinována pomocí AND s hodnotami síťové masky, aby se zjistilo, zda jsou počítače ve stejné části sítě.

**Příklad 22.4: IPv6 adresa s vyznačenou délkou prefixu**

```
fe80::10:1000:1a4/64
```

IPv6 rozpoznává různé prefixy s definovaným významem (viz tabulka 22.4 na této straně).

**Tabulka 22.4: Různé IPv6 prefixy**

Prefix (hexadecimálně.)	Definice
00	IPv4 adresy a IPv4 over IPv6 adresy. Jedná se o adresy zpětně kompatibilní s IPv4. Vhodný router musí ještě převést IPv6 paket na IPv4. Tento prefix používají i další speciální adresy, jako je loopback smyčka.



První číslice 2 nebo 3	<i>Aggregatable Global Unicast Address</i> – Stejně jako IPv4 lze síť IPv6 dělit na jednotlivé části. Aktuálně je možné použít následující adresní prostory: 2001::/16 ( <i>production quality address space</i> ) a 2002::/16 ( <i>6to4 address space</i> ).
fe80::/10	Tzv. <i>link-local</i> adresy. Adresy s tímto prefixem není možné routovat a jsou dostupné pouze v rámci podsítě.
fec0::/10	Tzv. <i>site-local</i> adresy. Tyto adresy je sice možné směrovat, ale pouze v rámci organizace. Tím tedy odpovídají tyto adresy současným <i>privátním</i> adresním prostorům (např. 10.x.x.x).
ff	<i>Multicast</i> ) IPv6 adresy.

Unicast adresy jsou vystavěny ze tří stupňů:

**Public Topology** První část (která obsahuje také výše uvedený prefix) slouží pro směrování paketů v prostředí Internetu. Zde jsou obsaženy informace o poskytovateli nebo instituci, která zajišťuje připojení k Internetu.

**Site Topology** Druhá část obsahuje směrovací informace o podsíti, ke které paket náleží.

**Interface ID** Třetí díl pak jednoznačně určuje rozhraní, pro které je paket určen. To umožňuje použít MAC adresy jako součást adresy. Protože jsou celosvětově jedinečné a pevně přidělené výrobcem hardwaru, znamená to velké zjednodušení konfigurace. Ve skutečnosti se prvních 64 bitů skládá z tzv. **EUI-64** tokenu, kde se odejme posledních 48 bitů MAC adresy a zbylých 24 bitů tvoří speciální informace, které vypovídají o typu tokenu. To také umožňuje přiřadit **EUI-64** token zařízením bez MAC adresy, jako jsou PPP a ISDN spojení.

Na základě této struktury existuje 5 různých typů IPv6 unicast adres:

**:: (unspecified)** Tuto adresu používá počítač jako zdrojovou adresu, když poprvé inicializuje síťové rozhraní a nemá ještě žádné informace o vlastní adrese.

**::1 (loopback)** Adresa pro smyčku loopback.

**Adresy kompatibilní s IPv4** IPv6 adresa sestává z IPv4 adresy a 96-bitového prefixu samých nul. Tento typ kompatibilních adres se používá při tunelování (viz odst. 22.2.3 na následující straně). IPv4/IPv6 počítače tak mohou komunikovat s ostatními počítači, které se nacházejí v čistě IPv4 síti.

**IPv4 adresy mapované na IPv6** Tento typ specifikuje čistě IPv4 adresy v IPv6 zápisu.

**Lokální adresy** Existují dva typy adres pro lokální používání:

**link-local** Tento typ adres je vyhrazen pouze pro používání v lokálních částech sítě. Routery nesmí předávat pakety s touto zdrojovou nebo cílovou adresou do Internetu nebo jiné části sítě. Tyto adresy jsou označeny speciálním prefixem ( $\text{fe80}::/10$ ) a ID rozhraní síťové karty. Střední část adresy obsahuje nulové byty. Tento druh adres se používá autokonfiguračními programy, které komunikují s počítači ve stejném segmentu sítě.

**site-local** Tento typ adres je možné směřovat mezi jednotlivými podsítěmi, ale pouze v rámci sítě, nesmí se použít v rámci Internetu. Takové adresy se používají pro intranet a jsou ekvivalentem pro privátní adresy v IPv4. Kromě definovaného prefixu ( $\text{fec0}::/10$ ) a ID rozhraní obsahují tyto adresy 16-bitové pole s informacemi o ID segmentu sítě. Zbytek je vyplněn nulovými byty.

Navíc obsahuje IPv6 další vynález a to možnost přiřadit jednomu síťovému rozhraní více síťových adres. To má tu výhodu, že je k dispozici více sítí. Jedna z nich může být nakonfigurována zcela automaticky pomocí MAC adresy a známého prefixu, výsledkem je dosažitelnost všech počítačů v IPv6 síti (pomocí link-local adresy) okamžitě po jejím zprovoznění. Pokud je součástí IP adresy MAC adresa, jsou jednotlivé IP adresy celosvětově unikátní. Jediné variabilní části adresy jsou ty, které určují topologii (*site topology* a *public topology*) v závislosti na síti, ve které se počítač právě nachází.

Pokud se počítač pohybuje mezi jednotlivými sítěmi, potřebuje minimálně dvě adresy. Jedna je jeho domovská adresa skládající se z ID rozhraní, informací o domovské síti a odpovídajícího prefixu. Domovská adresa je statická a neměnná. Všechny pakety, které jsou určeny pro tento počítač, mu budou doručeny, ať se fyzicky nachází kdekoli. To umožňují zcela nové funkce IPv6, tzv. *Stateless Auto-configuration* a *Neighbor Discovery*. Přenosný počítač může tedy mít kromě domovské adresy jednu nebo více adres, které patří sítím, ve kterých se počítač právě

nachází. Těmto adresám se říká *Care-of Address*. V domácí síti mobilního počítače musí existovat instance, která bude komunikaci směřovanou na jeho domovskou adresu dále přeposílat, pokud se nalézá v jiné síti. Tuto funkci přebírá v IPv6 tzv. *Home Agent*. Ten pak vytvoří tunel, kterým posílá pakety. Pakety, které mají jako cílovou *Care-of Address*, mohou putovat bez okliky přes Home agenta.

### 22.2.3 IPv4 versus IPv6 – cestování mezi světy

Přechod všech počítačů připojených k Internetu z IPv4 na IPv6 není možné provést okamžitě, spíše je pravděpodobné, že starý a nový protokol budou koexistovat dlouhou dobu. Sdílení na jednom počítači je řešeno pomocí *Dual Stack*, zůstává ale otázkou, jak bude komunikovat IPv6 počítač s IPv4 počítačem a jak přenášet IPv6 přes stávající IPv4 síť. Odpovědí na tyto otázky je tzv. tunelování a používání kompatibilních adres (viz 22.2.2 na straně 375).

Jednotlivé ostrůvky IPv6 v moři IPv4 sítí si vyměňují svá data pomocí tunelů. Při tunelování jsou IPv6 pakety zabaleny do IPv4 paketů, aby je bylo možné přenášet v IPv4 sítích. Tunel je definován jako spojení mezi dvěma IPv4 konci. Pakety musí obsahovat IPv6 cílovou adresu (nebo odpovídající prefix) a IPv4 adresu počítače na konci tunelu. V jednoduchých případech se konfiguruje takové tunely ručně a říká se jim *statické*.

Pokud není ruční vytváření tunelů reálné kvůli jejich vysokému počtu, existují tři různé způsoby pro vytváření *dynamických tunelů*:

**6over4** IPv6 pakety jsou automaticky zabaleny do IPv4 paketů a poslány přes IPv4 síť, kde je aktivován multicasting. IPv6 se tedy zdá, že celý Internet je pouze velká LAN. Nevýhodou tohoto řešení je špatná škálovatelnost a také skutečnost, že IP multicasting není dostupný v celém Internetu. Toto řešení se hodí pro malé firmy a organizace, které mají možnost provádět IP multicasting. Více informací naleznete v RFC 2529.

**6to4** Zde jsou IPv4 adresy automaticky generovány z IPv6 adres. Tak mohou jednotlivé ostrůvky IPv6 komunikovat prostřednictvím IPv4. Problém ale nastává při komunikaci s čistě IPv4 počítači. Více viz RFC 3056.

**IPv6 Tunnel Broker** Tento postup se používá pro speciální servery, které vytvářejí uživatelům tunely automaticky a je popsán v RFC 3053.

## Důležité

### Iniciativa 6Bone

Uprostřed starobylého Internetu existuje *6Bone* ([www.6bone.net](http://www.6bone.net)), což je celosvětová síť IPv6 podsítí, které jsou navzájem spojeny tunely. V rámci 6Bone sítě se testuje IPv6. Softwaroví vývojáři a poskytovatelé, kteří vyvíjí nebo poskytují IPv6 služby, mohou tyto segmenty použít pro testování, aby získali důležité zkušenosti s protokolem. Bližší informace naleznete na stránkách projektu 6Bone.

## Důležité

### 22.2.4 Konfigurace IPv6

Pokud chcete používat IPv6, není za běžných okolností třeba na pracovních stanicích provádět žádné změny. Musí však být zavedena podpora pro IPv6 v jádře. Jako uživatel `root` ji zavedete příkazem `modprobe ipv6`.

Protože se IPv6 z velké části konfiguruje samo, bude síťové kartě přiřazena adresa v *link-local* síti. Standardně není třeba mít na pracovní stanici směrovací tabulku. Pro směrování se používá *Router Advertisement Protocol*, pomocí kterého se pracovní stanice dotazují na prefix a brány, které mají být používány. K nastavení směrovače pro IPv6 slouží program `radvd`. Tento program pak sdělí pracovním stanicím prefixy pro IPv6 adresy a informace o směrování. Pro automatické nastavení adres a směrování lze také použít program `zebra`.

Informace o nastavení různých typů tunelů pomocí souborů `/etc/sysconfig/network` naleznete v manuálové stránce `ifup` (`man ifup`).

### 22.2.5 Další informace

Přehled v této kapitole neobsahoval všechny podrobnosti o IPv6. Pro hlubší studium můžete využít následující literaturu:

<http://www.ngnet.it/e/cosa-ipv6.php>

Série dokumentů, kde jsou velice dobře vysvětleny základy IPv6. Dobrý úvod do problematiky.

<http://www.bieringer.de/linux/IPv6/>

Dokument `Linux-IPv6-HOWTO` a mnoho odkazů.

<http://www.6bone.de/> Připojení k IPv6 pomocí tunelů.

<http://www.ipv6.org/> Vše o IPv6.

RFC 2640 Úvod do IPv6.

**IPv6 Essentials** Kniha popisující všechny důležité aspekty IPv6. Silvia Hagen: *IPv6 Essentials*. O'Reilly & Associates, 2002 (ISBN 0-596-00125-8).

## 22.3 Překlad jmen

DNS se stará o to, abyste si nemuseli pamatovat žádné IP adresy. V Linuxu se o tento převod stará specializovaný software, který se nazývá *bind*. Počítač, na kterém se tento převod realizuje, je *nameserver* (jmenný server). Názvy tvoří také hierarchický systém, kde jsou jednotlivé části názvu oddělovány tečkou. Tato hierarchie je nezávislá na hierarchii IP adres.

Jako celé jméno můžeme použít např. `laurent.suse.de`. Jedná se o tzv. *fully qualified domain name (FQDN)*, plně kvalifikované doménové jméno. Je zapsáno ve formátu název počítače.doména. Doména (v našem případě `suse.de`) obsahuje tzv. *TLD (Top level domain)* `de`.

Z historických důvodů je přiřazování TLD trochu zamotané. Proto jsou v USA používány domény první úrovně složené ze tří písmen, v ostatním světě pak národní ISO dvoupísmenné domény. Od roku 2000 jsou k dispozici další TLD pro speciální oblasti, které se skládají i z více písmen (např. `.info`, `.name`, `.museum` atd.).

V kamenných dobách Internetu (před rokem 1990) se používal soubor `/etc/hosts`, kde byly uvedeny názvy všech počítačů, které existovaly na Internetu. To se ukázalo, při rychle rostoucím počtu připojených počítačů, jako nepraktické. Proto byla navržena distribuovaná databáze, která obsahuje názvy počítačů spolu s jejich IP adresami. Jelikož je databáze distribuovaná, nemusí znát všechny počítače, místo toho se zeptá jmenného serveru vyšší úrovně, zda náhodou počítač neznají. To ale neznamená, že nemůžete soubor použít pro překlad adres, např. v lokální podsíti.

Na vrcholu hierarchie nameserverů se nachází tzv. kořenový nameserver *root nameserver*. Tento nameserver spravuje top level domény a běží v tzv. *Network Information Centers*, zkráceně (NIC). Informace o českém správci domény naleznete na adrese <http://www.nic.cz>, případně obecnější informace na adrese <http://www.internic.net/>.

Pomocí DNS nemusíte převádět pouze názvy počítačů, DNS toho zvládne daleko více. Např. nameserver ví, který počítač přebírá pro celou doménu e-mailů, tzv. *Mail exchanger (MX)*.

Aby dokázal i váš počítač převádět IP adresy, musí mít přístup alespoň k jednomu nameserveru (a znát jeho IP adresu). Konfiguraci nameserveru můžete pohodlně provést pomocí YaST. Pokud používáte vytáčenou linku, pak se může stát, že nemusíte ručně konfigurovat žádný nameserver. Protokol používaný pro vytáčené linky vám poskytne adresu nameserveru při navazování spojení. Konfigurace přístupu k nameserveru je popsána v kapitole 24 na straně 409.

Těsně spojený s DNS je protokol *whois*. Se stejnojmenným programem *whois* máte možnost rychle zjistit, kdo je za určitou doménu odpovědný.

## 22.4 Konfigurace síťového připojení pomocí YaST

Počítač musí být vybaven podporovanou síťovou kartou. Většinou je síťová karta rozpoznána již při instalaci a je nahrán vhodný ovladač. Jestli je karta správně připojena, zjistíte příkazem `ipaddress list eth0`. Pokud se zobrazí všechny informace o síťovém zařízení `eth0` a nikoliv chybové hlášení, je karta nainstalována správně.

Pokud máte jadernou podporu pro síť implementovanou jako modul, což je v jádře SUSE výchozí, musí být jméno modulu zadáno v souboru `/etc/sysconfig/hardware/hwcfg-*`. Pokud v něm není nic uvedeno, *hotplug* automaticky zvolí ovladač. *Hotplug* přiřadí ovladač pro vestavěnou i *hotplug* síťovou kartu.

### 22.4.1 Konfigurace síťové karty pomocí YaST

Po spuštění modulu zobrazí YaST obecný dialog pro nastavení sítě. V horní části je seznam dosud nenakonfigurovaných síťových karet. Všechny správně automaticky rozeznané karty jsou v seznamu uvedené pod svým jménem. Nerozpoznaná zařízení jsou uvedena jako 'Jiné (nerozpoznáno)'. Ve spodní části je zobrazen seznam již nakonfigurovaných zařízení spolu s typem sítě a adresou. Můžete nakonfigurovat novou kartu nebo změnit existující konfiguraci.

## Ruční konfigurace síťové karty

Konfigurace síťové karty, která nebyla automaticky rozpoznána (tedy je uvedená pod 'Jiné (nerozpoznáno)'), sestává z následujících částí:

**Konfigurace sítě** Nastavte typ zařízení rozhraní a jméno konfigurace. Typ zařízení vyberte z nabízených možností. Jméno konfigurace nastavte podle potřeby. Obvykle je možno použít výchozí hodnoty. Informace o konvencích používaných při pojmenovávání konfigurací naleznete v manuálové stránce `getcfg`.

**Modul jádra** 'Jméno nastavení hardwaru' specifikuje jméno souboru `/etc/sysconfig/hardware/hwcfg-*`, ve kterém je obsaženo hardwarové nastavení vaší síťové karty, např. jméno vhodného jaderného modulu. Pro PCMCIA a USB hardware obvykle YaST nabídne užitečná jména. Jméno nabízené pro ostatní hardware má obvykle smysl jen v případě, že je karta konfigurována pomocí `hwcfg-static-0`.

Pokud je síťová karta zařízení PCMCIA nebo USB, zaškrtněte příslušné políčko a opusťte dialog pomocí tlačítka 'Další'. Pokud není, klikněte na 'Zvolit ze seznamu' a vyberte správný typ karty. YaST automaticky vybere správný jaderný modul. Opusťte dialog pomocí tlačítka 'Další'.

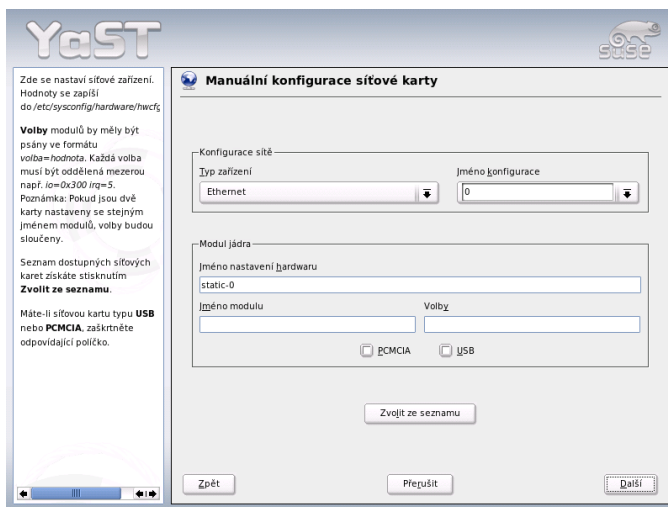
## Nastavení síťové adresy

Vyberte z nabízených možností typ zařízení a jméno konfigurace podle svých potřeb. Obvykle lze použít výchozí hodnoty. V manuálové stránce `getcfg` naleznete informace o konvencích používaných při pojmenovávání konfigurací.

Pokud jste jako typ zařízení rozhraní vybrali 'Bezdrátová technologie', nastavte v následujícím dialogu ('Nastavení bezdrátové síťové karty') operační režim, název sítě (ESSID) a údaje o šifrování. Kliknutím na 'OK' konfiguraci dokončíte. Podrobný popis konfigurace WLAN karet naleznete v kapitole 17.1.3 na straně 313. V případě ostatních rozhraní pokračujte nastavením síťové adresy:

### 'Automatické přidělení adresy (pomocí DHCP)'

Pokud na vaší síti běží DHCP server, můžete se na něj spolehnout a nechat nastavit síťovou adresu automaticky. Tato volba je vhodná také v případě, kdy jste připojeni přes DSL linku bez přidělené statické adresy. Pokud se rozhodnete použít DHCP, vyberte z nabídky 'Rozšířené' položku 'Nastavení DHCP klienta' a nastavte podrobnosti. Nastavte, zda má být požadována všesměrová odpověď a identifikátory, které se mají používat. Ve



Obrázek 22.3: Konfigurace síťové karty

výchozím nastavení identifikují DHCP servery rozhraní podle hardwarové adresy síťové karty. Pokud ale různí virtuální klienti komunikují přes jedno rozhraní, je pro rozlišení nutné nastavit identifikátory.

**‘Nastavení statické adresy’** Pokud máte statickou IP adresu, zaškrtněte příslušnou položku v dialogu a zadejte IP adresu a síťovou masku podsítě. Přednastavená maska by měla vyhovovat běžné domácí síti.

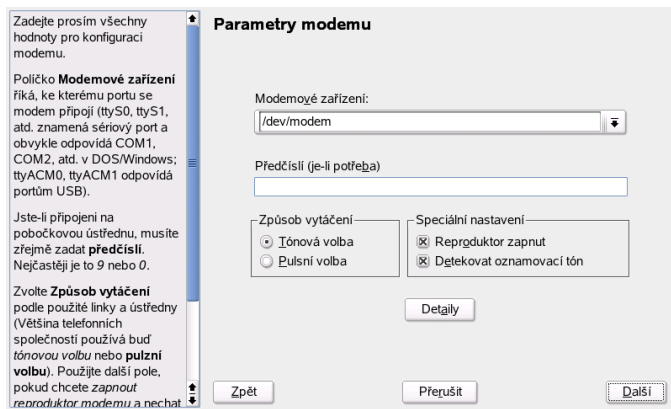
Dialog opusťte kliknutím na ‘Další’ nebo pokračujte nastavením jména počítače, nameserveru a podrobností o směrování (viz části 2.6.6 na straně 55 a 22 na straně 367).

‘Rozšířené’ umožňuje nastavit podrobnosti. V položce ‘Detailní nastavení’ zaškrtněte ‘Ovládání uživatelem’, pokud chcete, aby měl běžný uživatel kontrolu nad síťovou kartou (nikoliv pouze root). V případě mobilního použití to umožňuje uživateli flexibilně reagovat na změnu podmínek, neboť může sám aktivovat a deaktivovat rozhraní. Dále lze v tomto dialogu nastavit způsob ‘Aktivace zařízení’ a MTU (Maximum Transmission Unit).



## 22.4.2 Modem

V Řídicím středisku YaST, v sekci 'Síťová zařízení', zvolte modul 'Modem'. Pokud nebyl modem rozpoznán automaticky, otevřete dialog pro ruční konfiguraci ('Konfigurovat...') a v políčku 'Modemové zařízení' zadejte rozhraní, ke kterému je modem připojen.



Obrázek 22.4: Konfigurace modemu

Pokud jste připojeni přes pobočkovou ústřednu (PBX), může být nutné zadat volací předčísli. Obvykle je to nula. Podrobné informace naleznete v dokumentaci k vaší ústředně. Vyberte také, zda se má používat tónová nebo pulzní volba, zda má být zapnut reproduktor a zda má modem vyčkat, dokud nedetekuje oznamovací tón. Poslední z voleb by v případě připojení přes pobočkovou ústřednu neměla být zapnuta.

V dialogu, který se otevře po kliknutí na 'Detaily', nastavte přenosovou rychlost a inicializační řetězce pro modem. Nastavení měňte pouze tehdy, pokud modem nebyl automaticky rozpoznán nebo pokud vyžaduje pro funkci zvláštní nastavení. To obvykle nastává při použití ISDN terminálového adaptéru. Chcete-li umožnit kontrolu nad modemem (možnost aktivace a deaktivace) uživatelům bez pravomocí superuživatele, zaškrtněte 'Ovládání uživatelem'. V položce 'Regulární výraz vytáčeného předčísli' zadejte regulární výraz, kterému musí odpovídat hodnota zadaná uživatelem v položce 'Vytáčené předčísli' programu KInternet. Pokud je pole pro regulární výraz ponecháno prázdné, uživatel bez ad-

ministrátorských pravomocí nebude moci nastavit jiné předčísli. Dialog opusťte kliknutím na 'OK'.

V dalším dialogu vyberte vašeho poskytovatele připojení k Internetu (ISP). Chcete-li poskytovatele vybrat z přednastaveného seznamu, vyberte položku 'Země'. Druhou možností je kliknout na tlačítko 'Nový' a zadat údaje o vašem poskytovateli ručně. Potřebné údaje zahrnují jméno poskytovatele, telefonní číslo a jméno a heslo, které vám poskytovatel přidělil. Pokud chcete být před každým připojením dotazováni na heslo, zaškrtněte položku 'Vždy se ptát na heslo'.

Poslední dialog umožňuje nastavit další volby pro spojení:

**'Vytáčení na vyžádání'** Pokud povolíte vytáčení na vyžádání, nastavte alespoň jeden jmenný server (nameserver).

**'Modifikovat DNS po spojení'** Tato volba je implicitně zapnuta, což znamená, že je nameserver automaticky aktualizován při každém připojení na Internet.

**Automaticky obnovit DNS** Pokud poskytovatel při navazování připojení nevysílá adresu jmenného serveru (DNS), zakažte 'Automaticky obnovit DNS' a zadejte DNS ručně.

**'Hloupý režim'** Hloupý režim vypne detekci všech výzev na straně dial-in serveru. Pokud je navázání spojení pomalé nebo vůbec nefunguje, zkuste tuto volbu.

**'Vnější rozhraní firewallu'** Volbou 'Vnější rozhraní firewallu' aktivujete firewall a nastavíte toto rozhraní jako externí. Vaše vytáčená připojení k Internetu tak budou chráněna před možnými útoky z vnější sítě.

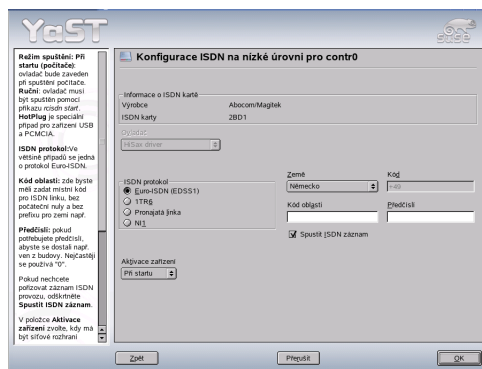
**'Čas nečinnosti (v sekundách)'** Tato volba určuje čas v sekundách, po kterém se spojení přeruší, nejsou-li přenášena žádná data (0 znamená nekonečno).

**Detaily IP** Kliknutím na tlačítko otevřete dialog pro nastavení IP adresy. Pokud váš poskytovatel připojení nepoužívá dynamické přidělování IP adres, zakažte volbu 'Dynamická IP adresa' a vložte lokální IP adresu svého počítače a vzdálenou IP adresu (na adresy se zeptejte svého poskytovatele). Volbu 'Výchozí směrování' ponechte zaškrtnutou a dialog ukončete kliknutím na 'OK'.

Kliknutím na 'Další' se vrátíte k původnímu dialogu, který zobrazuje souhrn konfigurace modemů. Dialog zavřete kliknutím na 'Konec'.

### 22.4.3 ISDN

Tento modul použijte ke konfiguraci jedné nebo více ISDN karet. Pokud YaST kartu nedetekoval, vyberte ji ručně. Je možno nastavit více rozhraní, ale i jedno rozhraní může být nastaveno pro více ISP. V následujících dialogích nastavte volby ISDN nutné pro správnou funkci karty.



Obrázek 22.5: Konfigurace ISDN

V dialogu zobrazeném na obrázku 22.5 na této straně vyberte požadovaný protokol. Implicitní je 'Euro-ISDN (EDSS1)', ale pro starší nebo větší ústředny použijte '1TR6'. Pokud se nacházíte v USA, vyberte 'NI1'. V příslušném poli nastavte zemi. V sousedním poli se objeví příslušný kód. Zadejte 'Kód oblasti' a (pokud potřebujete) 'Předčíslo'.

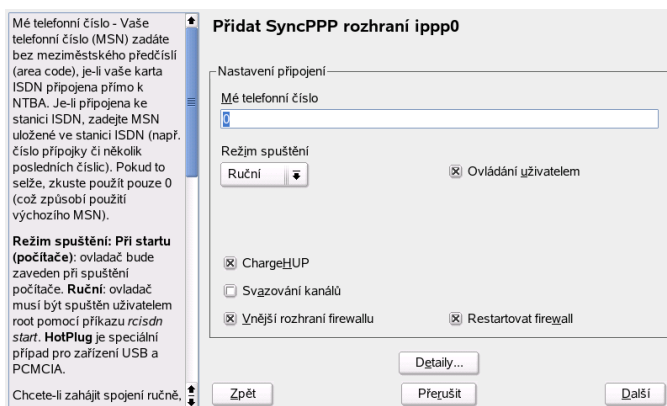
'Režim spuštění' určuje, jak je ISDN rozhraní spouštěno: 'Při startu' znamená, že je ISDN ovladač zaváděn vždy při startu systému. Je-li zvoleno 'Ručně', musí být ovladač zaveden uživatelem root pomocí příkazu `rcisdn start`. 'Hotplug' se používá pro zařízení PCMCIA nebo USB, ovladač se nahraje po připojení zařízení. Jste-li s nastavením hotovi, stiskněte 'OK'.

V následujícím dialogu vyberte pro ISDN kartu rozhraní a k němu poskytovatele připojení. Rozhraní může být typu SyncPPP nebo RawIP, většina poskytovatelů však dnes používá níže popsany SyncPPP.

Číslo, které je třeba vložit do pole 'Mé telefonní číslo', závisí na konkrétní situaci:

#### ISDN karta přímo připojena do telefonní zásuvky

Standardní ISDN linka poskytuje tři telefonní čísla (tzv. vícenásobné účast-



**Obrázek 22.6:** Konfigurace ISDN rozhraní

nické číslo, MSN). Pokud účastník požaduje čísel více, může jich být až deset. Jedno z těchto čísel je na tomto místě nutné vybrat a nastavit, ale bez kódu oblasti. Pokud vložíte nesprávné číslo, váš telefonní operátor automaticky použije první z čísel přidělených vaší ISDN lince.

### ISDN karta připojená k telefonní ústředně

Konfigurace opět závisí na instalovaném zařízení:

1. Menší ústředny určené k domácímu použití obvykle pro interní hovory používají protokol Euro-ISDN (EDSS1). Tyto ústředny mají vnitřní sběrnici S0 a pro připojená zařízení používají interní čísla. Použijte jedno z interních čísel. Měli byste moci použít alespoň jedno z čísel ústředny, kterým je umožněno přímé volání ven. Pokud to nefunguje, zkuste jednu nulu. Další informace naleznete v dokumentaci dodané s vaší ústřednou.
2. Větší ústředny určené pro firmy obvykle pro vnitřní hovory používají protokol 1TR6. Jejich MSN (vícenásobné účastnické číslo) se nazývá EAZ a obvykle odpovídá přímému volacímu číslu. Pro nastavení v Linuxu by mělo stačit použít poslední číslici EAZ. Pokud to nefunguje, vyzkoušejte všechny číslice od 1 do 9.

Chcete-li spojení ukončovat těsně před započtením další tarifní jednotky (impulzu), zaškrtněte 'ChargeHUP'. Nemusí však fungovat s každým poskytovatelem.

lem. Můžete také povolit ‘svazování kanálů’ (multilink PPP). Zaškrtnutím volby ‘Vnější rozhraní firewallu’ aktivujete SuSEfirewall2 a nastavíte toto rozhraní jako externí. Chcete-li povolit běžným uživatelům aktivaci a deaktivaci rozhraní, zaškrtněte volbu ‘Ovládání uživatelem’.

Výběrem ‘Detaily...’ otevřete dialog s pokročilým nastavením, které není určeno pro běžné domácí uživatele. Pokračujte proto k dalšímu dialogu stisknutím tlačítka ‘Další’.

V dalším dialogu nastavte IP adresu. Pokud vám poskytovatel připojení nepřidělil pevnou IP adresu, zvolte ‘Dynamická IP adresa’. V opačném případě zadejte lokální IP adresu (adresa vašeho počítače) a vzdálenou IP adresu podle specifikace vašeho poskytovatele. Pokud má být toto rozhraní používáno jako výchozí pro směrování paketů, zaškrtněte volbu ‘Výchozí směrování’. Na každém počítači může být jako výchozí nastaveno pouze jedno rozhraní. Pokračujte stisknutím tlačítka ‘Další’.

Následující dialog umožňuje nastavit zemi, ve které se nacházíte, a poskytovatele připojení (ISP). V seznamu jsou pouze operátoři dostupní přes službu Call-by-Call (volba operátora předčíslem). Pokud v seznamu není váš poskytovatel, zvolte ‘Nový’. Tím se otevře dialog ‘Volby poskytovatele’, do kterého vložte příslušné údaje. Ujistěte se, že jste do telefonního čísla nevložili žádné mezery nebo čárky. Zadejte uživatelské jméno a heslo přidělené poskytovatelem a stiskněte ‘Další’.

Chcete-li na samostatné pracovní stanici používat ‘Vytáčení na vyžádání’, zadejte jmenný server (nameserver, DNS). Většina poskytovatelů podporuje dynamický DNS, což znamená, že adresa jmenného serveru je zaslána poskytovatelem vždy v okamžiku připojení. Na samostatné pracovní stanici je ovšem i v takovém případě uvést zástupnou adresu, např. 192.168.22.99. Pokud poskytovatel dynamický DNS nepodporuje, musíte zadat IP adresu jmenného serveru poskytovatele. Pokud chcete, můžete v položce ‘Čas nečinnosti (v sekundách)’ zadat i dobu, po které se spojení automaticky přeruší, nejsou-li přenášena žádná data. Nastavení potvrďte zvolením ‘Další’. YaST zobrazí přehled nastavených rozhraní. Stisknutím ‘Konec’ nastavení aktivujete.

## 22.4.4 Kabelový modem

V některých zemích (v Rakousku, USA, ale i u nás) je běžný přístup na Internet přes síť kabelové televize). Účastník sítě obvykle dostane modem, který je na jedné straně připojen k rozvodu kabelové televize a na druhé straně k síťové kartě počítače (pomocí kabelu 10Base-T kroucený pár).

V závislosti na instrukcích od vašeho poskytovatele připojení zvolte při konfiguraci síťové karty buď 'Automatické přidělení adresy (pomocí DHCP)' nebo 'Nastavení statické adresy'. Dnes většina poskytovatelů používá DHCP. Statická adresa je obvykle volitelnou doplňkovou službou.

## 22.4.5 DSL

Chcete-li nakonfigurovat zařízení DSL, zvolte modul 'DSL' ze sekce 'Síťová zařízení' nástroje YaST. Modul sestává z několika dialogů, v nichž je třeba nastavit parametry DSL linky založené na některém z následujících protokolů:

- PPP přes Ethernet (PPPoE)
- PPP přes ATM (PPPoATM)
- CAPI pro ADSL (Fritz karty)
- Point-to-Point Tunneling Protocol (PPTP) – Rakousko

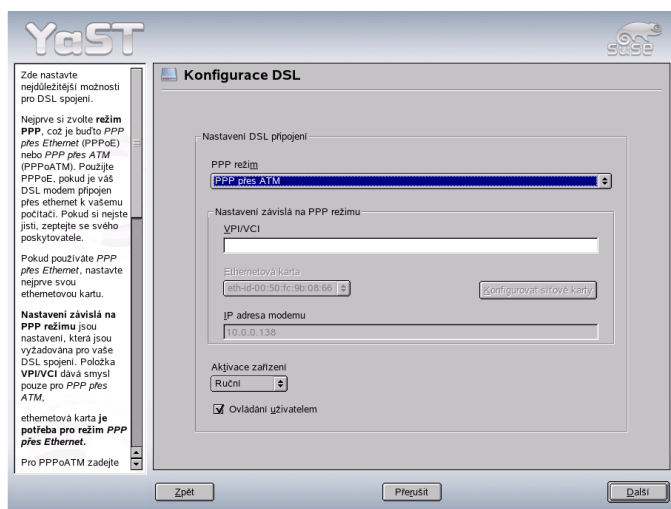
Konfigurace DSL připojení založeného na PPPoE nebo PPTP vyžaduje předem správně nastavenou síťovou kartu. Pokud ještě karta není nastavena, nastavte ji volbou 'Konfigurovat síťové karty' (viz 22.4.1 na straně 382). V případě DSL připojení sice mohou být adresy automaticky přidělovány, ale nikoliv pomocí DHCP. Proto volbu 'Automatické přidělení adresy (přes DHCP)' ponechte nezaškrtnutou. Místo toho zadejte statickou fiktivní adresu rozhraní, např. 192.168.22.1. V poli 'Síťová maska podsítě' zadejte 255.255.255.0. Pokud nastavujete samostatnou pracovní stanici, ujistěte se, že je položka 'Výchozí brána' (v dialogu 'Směrování') prázdná.

### Tip

Hodnoty 'IP Adresa' a 'Síťová maska podsítě' jsou pouze zástupné a nereprezentují DSL připojení jako takové. Slouží pouze k inicializaci síťové karty.

### Tip

Konfiguraci DSL (viz obrázek 22.7 na následující straně) začněte výběrem PPP režimu a ethernetové karty, ke které je modem připojen (obvykle je to `eth0`). Pak ze seznamu 'Aktivace zařízení' zvolte způsob aktivace DSL připojení. Pokud chcete povolit běžným uživatelům aktivaci či deaktivaci rozhraní pomocí programu KInternet, zaškrtněte položku 'Ovládání uživatelem'. V dalším dialogu



Obrázek 22.7: Konfigurace DSL

zvolte zemi a poskytovatele připojení (ISP). Podrobnosti nastavení v dalších dialogích závisí na dosud provedeném nastavení, proto jsou v následujících odstavcích jen krátce zmíněny. Podrobnosti se dozvíte z nápovědy přímo v jednotlivých dialogích.

Chcete-li používat ‘Vytáčení na vyžádání’ na samostatné pracovní stanici, zadejte adresu jmenného serveru (nameserver, DNS). Většina poskytovatelů podporuje dynamický DNS – IP adresa jmenného serveru je zasílána poskytovatelem při každém připojení. Pro samostatnou stanici však v takovém případě zadejte zástupnou adresu, např. 192.168.22.99. Pokud váš poskytovatel dynamický DNS nepodporuje, zadejte adresu, kterou vám dodal.

‘Čas nečinnosti (v sekundách)’ určuje dobu síťové neaktivity, po které bude spojení automaticky přerušeno. Vhodná je hodnota mezi 60 a 300 sekundami. Pokud je zakázáno ‘Vytáčení na vyžádání’, může být užitečné nastavit dobu nečinnosti rovnou nule, což znemožní automatické přerušování spojení.

Chcete-li nastavit T-DSL, postupujte stejně jako při nastavení DSL. Pouze při výběru poskytovatele připojení zvolte ‘T-Online’. YaST otevře dialog pro nastavení T-DSL, ve kterém vyplňte některé doplňující informace vyžadované T-DSL, jako ID linky, T-Online číslo, uživatelský kód a heslo. Všechny potřebné údaje jste do-

stali při přihlášení ke službě T-DSL.

## 22.5 Manuální konfigurace sítě

Manuální konfigurace sítě by měla být používána pouze jako nouzové řešení nebo ve speciálních případech. Jinak je lepší využít YaST. Zde uvedené informace o konfiguraci sítě ale mohou být užitečné i při práci s YaSTem.

Všechny vestavěné i hotplug (PCMCIA, USB, některé PCI) síťové karty jsou detekovány a konfigurovány pomocí hotplug systému. Systém chápe síťovou kartu dvěma různými způsoby: jako fyzické zařízení a jako rozhraní. Připojení nebo rozpoznání zařízení spustí hotplug událost, která zahájí inicializaci zařízení pomocí skriptu `/sbin/hwup`. Pokud je síťová karta inicializována jako nové síťové rozhraní, jádro vyvolá další hotplug událost, která pomocí `/sbin/ifup` rozhraní nastaví.

Jádro přiděluje jména rozhraní podle časového pořadí jejich registrace. O přidělených jménech rozhoduje inicializační sekvence. Když jedna z několika síťových karet selže, čísla všech následujících karet se posunou. V případě skutečných hotplug karet (připojitelných za běhu systému) rozhoduje okamžik (pořadí) připojení k systému.

Pro zvýšení flexibility byla oddělena konfigurace zařízení (hardware) a rozhraní; a přiřazování konfigurací k zařízením a rozhraním již není založeno na jménech rozhraní. Konfigurace zařízení jsou uloženy v souborech `/etc/sysconfig/hardware/hwcfg-*`, zatímco v souborech `/etc/sysconfig/network/ifcfg-*` jsou uloženy konfigurace rozhraní. Jména konfigurací jsou přiřazována tak, že popisují zařízení a rozhraní, s nimiž jsou spojeny. Protože dříve používané přiřazování ovladačů ke jménům rozhraní vyžadovalo stálá jména rozhraní, nelze přiřazování jmen nadále provádět v souboru `/etc/modprobe.conf`. Uvedení aliasu v tomto souboru může nyní mít nepříjemné vedlejší účinky.

Jména konfiguračních souborů (vše, co následuje po `hwcfg-` či `ifcfg-`) mohou na jednotlivá zařízení odkazovat pomocí použité sběrnice, ID zařízení nebo jména rozhraní. Například konfigurace PCI karty může být `bus-pci-0000:02:01.0` (sběrnice PCI) nebo `vpid-0x8086-0x1014-0x0549` (identifikační číslo produktu). Jméno příslušného rozhraní může být `bus-pci-0000:02:01.0` nebo `wlan-id-00:05:4e:42:31:7a` (MAC adresa).

Chcete-li přiřadit konfiguraci libovolné kartě určitého typu (pokud je v tu chvíli připojena jen jedna karta tohoto typu), místo konkrétní kartě, zvolte méně specifické jméno konfigurace. Například, konfigurace se jménem `bus-pcmcia` bude



použita libovolnou PCMCIA kartou. Chcete-li rozsah použití omezit, přidejte na začátek jména typ rozhraní, např. `wlan-bus-usb` bude přiřazeno všem WLAN kartám na USB portu.

Systém vždy použije tu konfiguraci, která zařízení nebo rozhraní nejlépe popisuje. Nejvhodnější konfiguraci vyhledává program `/sbin/getcfg`. Výstup programu obsahuje veškeré informace použitelné pro popis zařízení. Podrobnosti o pravidlech tvorby jmen konfigurací naleznete v manuálové stránce `getcfg`.

Vzhledem k popsané metodě jsou síťová rozhraní vždy správně nakonfigurována bez ohledu na pořadí inicializace. Nicméně jméno rozhraní na pořadí inicializace stále závisí. Jsou dva způsoby, jak zajistit spolehlivý přístup k rozhraní určité síťové karty:

- `/sbin/getcfg-interface <jméno konfigurace>` vrací jméno rozhraní asociovaného s danou konfigurací. V některých konfiguračních souborech tak lze místo nestálého jména rozhraní použít jméno konfigurace (např. `firewall`, `dhcpd`, směrování nebo různá virtuální síťová rozhraní, tunely).
- Rozhraním, jejichž konfigurace jméno rozhraní neobsahuje, můžete trvalé jméno přiřadit proměnnou `PERSISTENT_NAME=<pname>` v příslušné konfiguraci (`ifcfg-*`). Trvalá jména (`<pname>`) by ovšem neměla být stejná, jako jména automaticky přidělovaná jádrem. Proto nejsou povolena jména jako `eth*`, `tr*`, `wlan*`, `qeth*`, `iucv*` atd. Místo nich používejte `net*` nebo popisná jména jako `vnejsi`, `vnitrni` či `dmz`. Trvalá jména je možné přiřadit rozhraní pouze vzápětí po jeho registraci, což znamená, že je nutné znovu zavést ovladač síťové karty nebo spustit příkaz `hwup <popis zařízení>`. Příkaz `rcnetwork restart` není v tomto případě dostatečný.

## Důležité

### Použití trvalých jmen rozhraní

Použití trvalých jmen zatím nebylo důkladně otestováno. Proto se může stát, že některé aplikace nebudou schopny s volně vybranými jmény rozhraní zacházet. Pokud na podobný problém narazíte, dejte nám vědět na adrese <http://www.suse.de/feedback>. Pokud upřednostňujete komunikaci v českém jazyce, napište nám na adresu [feedback@suse.cz](mailto:feedback@suse.cz)

## Důležité

`ifup` vyžaduje existenci rozhraní, protože neinicializuje hardware. Inicializaci hardwaru má na starost příkaz `hwup` (spouštěný pomocí `hotplug` nebo `coldplug`). Jakmile je zařízení inicializováno, je pomocí `hotplug` automaticky spuštěn `ifup`. Rozhraní je spuštěno, pokud je startovací režim nastaven na `onboot`, `hotplug` nebo `auto` a služba `network` je spuštěna. Dříve inicializaci hardwaru spouštěl příkaz `ifup<jméno rozhraní>`. Nyní je postup opačný. Nejprve je inicializována hardwarová komponenta, pak následují ostatní akce. Tímto způsobem lze pomocí existující sady konfigurací optimálně nakonfigurovat měnící se množství zařízení.

Tabulka 22.5 na této straně shrnuje nejdůležitější skripty účastníci se síťové konfigurace. Tam kde je to možné, jsou rozlišeny podle toho, zda se týkají hardwaru nebo rozhraní:

*Tabulka 22.5: Skripty pro manuální síťovou konfiguraci*

Fáze konfigurace	Příkaz	Funkce
Hardware	<code>hw{up, down, status}</code>	Skripty <code>hw*</code> jsou spouštěny systémem <code>hotplug</code> , aby inicializovaly zařízení, zrušily inicializaci nebo zjistily stav zařízení. Více informací naleznete v manuálové stránce <code>hwup</code> .
Rozhraní	<code>getcfg</code>	Skript <code>getcfg</code> lze použít ke zjištění jména rozhraní asociovaného s určitým jménem konfigurace nebo popisem zařízení. Více informací naleznete v manuálové stránce <code>getcfg</code> .
Rozhraní	<code>if{up, down, status}</code>	Skripty <code>if*</code> spouští existující síťová rozhraní nebo vrací stav určeného rozhraní. Více informací naleznete v manuálové stránce <code>ifup</code> .

Další informace o systému `hotplug` a trvalých jménech rozhraní naleznete v kapitolách 18 na straně 331 a 19 na straně 339.

## 22.5.1 Konfigurační soubory

Zde je uveden přehled síťových konfiguračních souborů, jejich formátů a funkcí.

### **/etc/sysconfig/network/hwcfg-\***

Tyto soubory obsahují hardwarovou konfiguraci síťových karet a dalších zařízení. Obsahují potřebné parametry, jako je jaderný modul, režim spouštění a asociace se skripty. Více informací najdete v manuálové stránce `hwup`. Bez ohledu na existující hardware jsou při spuštění `coldplug` aplikovány konfigurační soubory `hwcfg-static-*`.

### **/etc/sysconfig/network/ifcfg-\***

Tyto soubory obsahují data pro jednotlivá síťová rozhraní. Obsahují např. režim spouštění a IP adresu. Možné parametry jsou popsány v manuálové stránce `ifup`). Navíc lze, pokud chcete obecné nastavení použít jen pro jedno rozhraní, používat v `ifcfg-*` souborech všechny proměnné ze souborů `dhcp`, `wireless`, a `config`.

### **/etc/sysconfig/network/config, dhcp, wireless**

Soubor `config` obsahuje obecné nastavení chování skriptů `ifup`, `ifdown` a `ifstatus`. Soubor `dhcp` obsahuje nastavení pro DHCP. Soubor `wireless` obsahuje nastavení pro bezdrátové síťové karty. Proměnné v těchto souborech jsou dobře okomentovány. Všechny proměnné z těchto souborů je možné použít také v `ifcfg-*`, kde mají vyšší prioritu.

### **/etc/sysconfig/network/routes,ifroute-\***

Zde je nastaveno statické směrování TCP/IP paketů. Všechny statické směrovací záznamy vyžadované různými systémovými úlohami lze nastavit v souboru `/etc/sysconfig/network/routes`: pro směrování k počítači, skrze bránu nebo k síti. Pro všechna rozhraní, která potřebují individuální směrování, je možné vytvářet samostatné konfigurační soubory `/etc/sysconfig/network/ifroute-*` (hvězdičku nahraďte názvem rozhraní). Záznamy ve směrovacích konfiguračních souborech vypadají následovně:

```
DESTINATION      GATEWAY NETMASK  INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION      GATEWAY PREFIXLEN INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION/PREFIXLEN GATEWAY -      INTERFACE [ TYPE ] [ OPTIONS ]
```

Pokud není uveden parametr GATEWAY, NETMASK, PREFIXLEN nebo INTERFACE, je nutné místo něj napsat -. Položky TYPE a OPTIONS nejsou povinné.

V prvním sloupci (DESTINATION) je uveden cíl směrovacího záznamu. Může zde být IP adresa sítě nebo počítače. Pokud je dostupný nameserver, pak také celý název sítě nebo počítače

Druhý sloupec (GATEWAY) slouží pro uvedení výchozí brány nebo brány, skrze kterou se přistupuje k počítači, resp. síti. Ve třetím sloupci se uvádějí síťové masky pro síť nebo počítače za bránou, např. 255.255.255.255.

Poslední sloupec má smysl pro síť připojené k lokálnímu počítači, jako např. loopback, ethernet, ISDN, PPP či dummy zařízení. Musí v něm být zapsáno jméno zařízení.

### **/etc/resolv.conf**

V tomto souboru je specifikována doména, do které počítač patří (klíčové slovo *search*). Je uvedena též adresa nameserveru, ke kterému se má přistupovat (klíčové slovo *nameserver*). Lze uvést i více domén. Při převodu jména, které není plně kvalifikováno, se k němu postupně připojují jednotlivé položky *search*. Více nameserverů lze uvést zápisem více řádků začínajících klíčovým slovem *nameserver*. Komentáře jsou uvozeny znaky #. YaST zapisuje nastavení nameserveru do tohoto souboru. 22.5 na této straně ukazuje příklad skutečného souboru `/etc/resolv.conf`.

#### ***Příklad 22.5: /etc/resolv.conf***

```
# Our domain
search example.com
#
# We use slunce (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

Některé služby, jako pppd (wvdial), ipppd (isdn), dhcp (dhcpcd a dhclient), pcmcia a hotplug, modifikují soubor `/etc/resolv.conf` pomocí skriptu `modify_resolvconf`. Pokud byl soubor skriptem `/etc/resolv.conf` dočasně změněn, obsahuje komentář informující o službě, která změnu provedla, místu, kde je uložena záloha původního souboru a o způsobu, jakým můžete zamézit automatickým změnám souboru. Pokud je soubor `/etc/resolv.conf` změněn vícekrát, obsahuje všechny změny ve vnořené podobě. Změny lze korektně vrátit i v jiném pořadí, než byly učiněny. Mezi služby, které toho využívají, patří isdn, pcmcia a hotplug.

Pokud se stane, že je služba ukončena nestandardním způsobem, lze k obnovení původního souboru použít `modify_resolvconf`. Při startu systému se rovněž kontroluje, zda není přítomen modifikovaný `resolv.conf` (např. po pádu systému), případně je původní nezměněný soubor `resolv.conf` obnoven.

YaST pomocí `modify_resolvconf` kontroluje, zda byl `resolv.conf` modifikován, a případně varuje uživatele, že se provedené změny po obnovení souboru ztratí. Navíc YaST sám `modify_resolvconf` nepoužívá, což znamená, že změna souboru `resolv.conf` provedená pomocí YaST má stejnou váhu jako manuální editace. V obou případech je změna trvalá, zatímco změny provedené výše zmíněnými službami jsou pouze dočasné.

### **/etc/hosts**

V tomto souboru (viz 22.6 na této straně) se jménům počítačů přiřazují IP adresy. Pokud se nepoužívá nameserver, musíte zde uvést všechny počítače, na které chcete mít přístup pomocí jména. Každý počítač je na zvláštní řádce, sestávající postupně z IP adresy, plně kvalifikovaného jména počítače a jména počítače. IP adresa musí být uvedena na začátku řádky, položky musí být odděleny mezerami nebo tabulátory. Komentáře začínají znakem #.

#### *Příklad 22.6: /etc/hosts*

```
127.0.0.1 localhost
192.168.0.20 slunce.example.com slunce
192.168.0.1 zeme.example.com zeme
```

### **/etc/networks**

V tomto souboru se nastavuje převod jmen sítí na síťové adresy. Formát je podobný jako u souboru `hosts`, pouze síťová jména jsou první a za nimi následují adresy. Viz 22.7 na této straně.

#### *Příklad 22.7: /etc/networks*

```
loopback      127.0.0.0
localnet      192.168.0.0
```

## **/etc/host.conf**

Tento soubor kontroluje převod jmen pomocí *resolver* knihovny. Používá se pouze programy slinkované proti *libc4* nebo *libc5*. Novější *glibc* programy se nastavují v souboru `/etc/nsswitch.conf`. Každý parametr je uveden na samostatném řádku a komentáře jsou uvozeny znakem `#`. Přípustné parametry jsou uvedeny v tabulce 22.6 na této straně. Ukázku souboru `/etc/host.conf` si můžete prohlédnout v příkladu 22.8 na této straně.

**Tabulka 22.6:** Parametry pro `/etc/host.conf`

<code>order hosts, bind</code>	Stanoví, v jakém pořadí se volají služby pro převod jména počítače na IP adresu. Možné argumenty jsou (odděleny mezerami nebo čárkami):  <code>hosts</code> : prohledávat soubor <code>/etc/hosts</code>  <code>bind</code> : použít nameserver  <code>nis</code> : použít NIS
<code>multi on/off</code>	Stanoví, zda počítač, uvedený v <code>/etc/hosts</code> smí mít více IP adres.
<code>nospoof on spoofalert on/off</code>	Tyto parametry mají vliv pouze na <i>spoofing</i> nameserveru.
<code>trim název domény</code>	Zadané jméno domény se při převodu oddělí od jména počítače (pokud ovšem jméno počítače obsahovalo doménu). Tato volba se hodí, pokud jsou v souboru <code>/etc/hosts</code> jen jména z lokální domény, které by však měla být rozpoznatelná i s připojenou doménou.

**Příklad 22.8:** `/etc/host.conf`

```
# We have named running
order hosts bind
```

```
# Allow multiple addrs
multi on
```

### **/etc/nsswitch.conf**

S GNU CLibrary 2.0 můžete nyní využívat tzv. *Name Service Switch* (NSS). (Viz `man 5 nsswitch.conf` a manuál *The GNU C Library Reference Manual*.)

V souboru `/etc/nsswitch.conf` je uvedeno pořadí dotazů. Soubor `nsswitch.conf` si můžete prohlédnout v příkladu 22.9 na této straně. Komentáře jsou uvozeny znaky `#`. V tomto příkladu uvedená položka `hosts` znamená, že po dotazu na `/etc/hosts` (files) je proveden dotaz pomocí DNS (viz kapitolu 24 na straně 409).

#### ***Příklad 22.9: /etc/nsswitch.conf***

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

Databáze dosažitelné pomocí NSS jsou uvedeny v tabulce 22.7 na této straně. V budoucnu se navíc počítá s parametry `automount`, `bootparams`, `netmasks` a `publickey`. Konfigurační volby pro databáze jsou uvedeny v tabulce 22.8 na následující straně.

***Tabulka 22.7: Databáze dosažitelné pomocí /etc/nsswitch.conf***

aliases	Poštovní aliasy pro sendmail; viz <code>man5 aliases</code> .
ethers	Ethernetové adresy.
group	Uživatelské skupiny pro <code>getgrent</code> . Viz <code>man5 group</code> .

<code>hosts</code>	Jména počítačů a IP adresy pro <code>gethostbyname</code> a podobné funkce.
<code>netgroup</code>	Platný seznam počítačů a uživatelů v síti pro účely kontroly přístupových práv, viz <code>man5 netgroup</code> .
<code>networks</code>	Jména a adresy sítí pro <code>getnetent</code> .
<code>passwd</code>	Uživatelská hesla pro <code>getpwent</code> ; viz <code>man5 passwd</code> .
<code>protocols</code>	Síťové protokoly pro <code>getprotoent</code> ; viz <code>man5 protocols</code> .
<code>rpc</code>	Jména a adresy <i>Remote procedure call</i> pro <code>getrpcbyname</code> a podobné funkce.
<code>services</code>	Síťové služby pro <code>getservent</code> .
<code>shadow</code>	Stínová hesla uživatelů pro <code>getspnam</code> ; viz <code>man5 shadow</code> .

---

***Tabulka 22.8: Konfigurační možnosti NSS databází***

---

<code>files</code>	Přímý přístup k souborům, například <code>/etc/aliases</code> .
<code>db</code>	Přístup přes databázi.
<code>nis, nisplus</code>	NIS, viz kapitola 25 na straně 429.
<code>dns</code>	Lze použít pouze jako rozšíření <code>hosts</code> a <code>networks</code> .
<code>compat</code>	Lze použít pouze jako rozšíření <code>passwd</code> , <code>shadow</code> a <code>group</code> .

---



### **/etc/nscd.conf**

Pomocí tohoto souboru se konfiguruje program `nscd` (Name Service Cache Daemon). Viz `man8 nscd` a `man5 nscd.conf`. Ve výchozím nastavení jsou položky `passwd` a `groups` programem `nscd` ukládány do vyrovnávací paměti. Je to důležité pro výkon adresářových služeb jako je NIS nebo LDAP, protože jinak by bylo nutné používat síťové spojení pro každý přístup ke jménům nebo skupinám. Položka `hosts` ukládána do vyrovnávací paměti není, protože používaný mechanismus znemožňuje lokálním počítačům odpovědím na dotazy důvěřovat. Místo ukládání do vyrovnávací paměti programem `nscd` použijte DNS server s ukládáním do vyrovnávací paměti.

Je-li aktivována vyrovnávací paměť (cache) pro `passwd`, trvá zpravidla 15 sekund, než je systému znám nově založený lokální uživatel. Opětovným spuštěním programu `nscd` se tato doba čekání dá zkrátit. Slouží k tomu příkaz `rcnscdrestart`.

### **/etc/HOSTNAME**

Tento soubor se čte různými skripty při startu systému. Smí obsahovat jedinou řádku se jménem počítače (bez domény).

## **22.5.2 Startovací skripty**

Kromě výše popsaných konfiguračních souborů existuje řada skriptů, které spouští síťové programy během startu systému. Jsou spuštěny v okamžiku, kdy systém přejde do některé *víceuživatelské úrovně běhu* (viz tabulka 22.9 na této straně).

**Tabulka 22.9:** *Některé startovací skripty pro síťové programy*

<code>/etc/init.d/network</code>	Tento skript se stará o konfiguraci síťových rozhraní. Hardware musí být inicializováno předem pomocí <code>/etc/init.d/coldplug</code> (přes <code>hotplug</code> ). Pokud nebyla spuštěna služba <code>network</code> , nejsou implementována žádná síťová rozhraní.
----------------------------------	--

<code>/etc/init.d/inetd</code>	Spouští program <code>xinetd</code> . <code>xinetd</code> umožňuje na systému používat serverové služby. Například spouští <code>vsftpd</code> při každé inicializaci FTP spojení.
<code>/etc/init.d/portmap</code>	Spouští portmapper potřebný pro RPC server, např. NFS.
<code>/etc/init.d/nfsserver</code>	Spouští NFS server.
<code>/etc/init.d/sendmail</code>	Řídí proces <code>sendmail</code> .
<code>/etc/init.d/ypserv</code>	Spouští NIS server.
<code>/etc/init.d/ypbind</code>	Spouští klienta NIS.

---

## 22.6 smpppd jako pomocník s vytáčeným připojením

Většina uživatelů nemá pro internetové připojení vyhrazenou pevnou linku, ale používají vytáčené připojení. V závislosti na metodě vytáčení (ISDN nebo DSL) se o spojení stará program `ipppd` nebo `pppd`. Všechno, co je potřeba pro připojení k Internetu, je správné spuštění těchto programů.

Pokud používáte paušální připojení, jednoduše spustíte příslušného démona. Stav připojení pak lze kontrolovat pomocí apletu v KDE nebo z příkazové řádky. Pokud je internetové připojení poskytováno jiným počítačem, tzv. bránou, můžete chtít připojení kontrolovat po síti.

Právě pro kontrolu vytáčeného připojení po síti je určen program `smpppd`. Tento program poskytuje jednotné rozhraní pro řadu programů a plní dvě funkce. První je volání programu `pppd` nebo `ipppd` spolu s kontrolou vlastností vytáčeného připojení. Druhou je správa více poskytovatelů Internetu a přenos informací o aktuálním stavu připojení. Pokud používáte vytáčené připojení pro soukromou síť, můžete program `smpppd` ovládat také po síti.

## 22.6.1 Konfigurace smpppd

Připojení prostřednictvím smpppd je automaticky nakonfigurováno YaSTem. Programy pro vytáčení kinternet a cinternet jsou také předkonfigurovány. Manuální nastavení smpppd je potřeba pouze pro aktivaci zvláštních funkcí, jako je např. vzdálené ovládání po síti.

Konfigurační soubor smpppd je `/etc/smpppd.conf`. Ve výchozím nastavení není vzdálená kontrola povolena. Nejdůležitější volby v tomto souboru jsou:

**open-inet-socket = <yes | no>** Ke kontrole smpppd po síti musí být nastavena na `yes`. Port, na kterém smpppd naslouchá, je 3185. Pokud je tento parametr nastaven na `yes`, musí být příslušně nastaveny také parametry `bind-address`, `host-range` a `password`.

**bind-address = <ip>** Pokud má počítač více IP adres, nastavte zde adresu, na které má smpppd přijímat spojení.

**host-range = <min ip> <max ip>** Parametr `host-range` se používá k nastavení rozsahu sítě. Přístup pomocí smpppd je povolen pouze počítačům z tohoto rozsahu.

**password = <heslo>** Nastavením hesla omezíte přístup pouze pro autorizované uživatele. Pokud nenastavíte žádné heslo, mohou smpppd používat všichni klienti. Heslo je uloženo v textové podobě, nepřeceňujte proto jeho bezpečnost.

**slp-register = <yes | no>** Tento parametr rozhoduje o zveřejňování smpppd služby v síti pomocí SLP.

Více informací o smpppd najdete v manuálových stránkách `man 8 smpppd` a `man 5 smpppd.conf`.

## 22.6.2 Programy kinternet, qinternet a cinternet a vzdálené použití

Programy kinternet, qinternet a cinternet lze používat pro ovládání lokálního i vzdáleného smpppd. Program cinternet je textová alternativa grafického programu kinternet. Program qinternet je v podstatě totéž jako kinternet, ale nepoužívá knihovny KDE, takže není na KDE závislý. Abyste mohli tyto programy používat se vzdáleným smpppd, upravte ručně nebo pomocí programu kinternet konfigurační soubor `/etc/smpppd-c.conf`. V tomto souboru jsou používány pouze tři volby:

**sites = <seznam\_mist>** Zde nastavte, kde mají frontendy hledat program smpppd. Frontendy testují volby v pořadí zde uvedeném. Volba `local` nařizuje připojení k lokálnímu smpppd. Volba `gateway` ukazuje na smpppd na bráně. Připojení lze nastavit ve volbě `server`. Volba `slp` nařizuje použití smpppd nalezeného přes SLP.

**server = <server>** Zde nastavíte jméno počítače, na kterém běží smpppd.

**password = <heslo>** Zde zadejte heslo pro smpppd.

Pokud je program smpppd aktivní, můžete otestovat přístup. To provedete příkazem `cinternet --verbose --interface-list`. Pokud narazíte na jakýkoliv problém, přečtěte si prosím manuálové stránky `man8 cinternet` a `man5 smpppd-c.conf`.

# SLP služby v síti

SLP (*Service Location Protocol*) byl vyvinut pro zjednodušení konfigurace klientů v lokální síti. Taková konfigurace (včetně všech požadovaných služeb) vyžaduje detailní znalost serverů dostupných v síti. SLP informuje všechny klienty v síti o dostupnosti služeb. Aplikace, které SLP podporují, mohou tyto informace využít a provést automatickou konfiguraci.

SUSE LINUX podporuje instalaci s využitím instalačních zdrojů dostupných pomocí SLP a obsahuje řadu systémových služeb s integrovanou podporou SLP. YaST i Konqueror poskytují pro SLP příslušné uživatelské rozhraní. SLP můžete využít k poskytování centrálně řízených služeb klientům, např. instalačního serveru, YOU serveru, souborového serveru nebo tiskového serveru.

## 23.1 Registrace vlastních služeb

Mnoho aplikací v systému SUSE LINUX má podporu SLP integrovanou pomocí knihovny `libslp`. Pokud služba nebyla přeložena s podporou SLP a chcete, aby byla přes SLP dostupná, použijte jeden z následujících postupů:

### Statická registrace pomocí `/etc/slp.reg.d`

Pro každou službu vytvořte zvláštní registrační soubor. Následující příklad ukazuje soubor pro registraci skenovací služby:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
```

```
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

Nejdůležitější řádek souboru je řádek obsahující *URL služby*, který začíná řetězcem `service:.` Obsahuje typ služby (`scanner.sane`) a adresu, na které je služba na serveru dostupná. `<$HOSTNAME>` je automaticky nahrazeno úplným jménem počítače. Za dvojtečkou následuje číslo TCP portu, na kterém je služba dostupná. Následuje kód jazyka, ve kterém má být služba dostupná, a doba registrace v sekundách, obojí oddělené čárkou. Dobu registrace zadávejte v rozmezí 0 až 65535. 0 registraci znemožňuje, 65535 ruší veškerá omezení.

Registrační soubor také obsahuje dvě proměnné: `watch-tcp-port` a `description`. První váže SLP oznámení služby na to, zda služba skutečně běží (slpd kontroluje stav služby). Druhá obsahuje přesnější popis služby pro zobrazení ve vhodných prohlížečích.

#### Statická registrace pomocí `/etc/slp.reg`

Jediným rozdílem oproti postupu popsanému výše je seskupení všech služeb v jednom centrálním souboru.

#### Dynamická registrace pomocí `slptool`

Pokud chcete zaregistrovat službu pro SLP z proprietárního skriptu, použijte příkaz `slptool` jako frontend.

## 23.2 SLP frontendy v systému SUSE LINUX

SUSE LINUX obsahuje několik frontendů, které umožňují kontrolovat a využívat SLP informace přes síť:

**slptool** `slptool` je jednoduchý program pro příkazový řádek využitelný pro SLP dotazy v síti nebo pro oznamování proprietárních služeb. Příkaz `slptool--help` vypíše všechny dostupné volby a funkce programu. Příkaz `slptool` lze volat i ze skriptů, které zpracovávají SLP informace.

**Konqueror** Používáte-li Konqueror jako síťový prohlížeč, můžete zobrazit služby dostupné v lokální síti zadáním adresy `slp:/`. Kliknutím na ikony v hlavním okně získáte podrobné informace o příslušné službě. Pokud v Konqueroru zadáte adresu `service:/`, spojíte se kliknutím na ikonu s příslušnou službou.

## 23.3 Aktivace SLP

Pokud chcete nabízet služby, musí na systému běžet `slpd`. Pro pouhé dotazování na služby není nutné tohoto démona spouštět. Jako většina systémových služeb v SUSE Linuxu, je i `slpd` démon řízen samostatným `init` skriptem. Implicitně je démon neaktivní. Chcete-li démona aktivovat na dobu trvání relace, spusťte ho jako `root` příkazem `rcslpdstart` nebo zastavte příkazem `rcslpdstop`. Volbami `restart` a `status` provedete restart a kontrolu stavu. Pokud chcete, aby byl `slpd` aktivní vždy po startu systému, spusťte jako `root` příkaz `insservslpd`. Tím bude `slpd` automaticky zařazen mezi služby spouštěné při startu systému.

## 23.4 Další informace

O SLP jsou dostupné následující zdroje informací:

**RFC 2608, 2609, 2610** RFC 2608 definuje SLP, RFC 2609 detailně popisuje URL služeb a RFC 2610 se zabývá DHCP přes SLP.

**<http://www.openslp.com>** Domovská stránka projektu OpenSLP.

**`file:/usr/share/doc/packages/openslp/*`**

Tento adresář obsahuje všechnu dostupnou dokumentaci k SLP, včetně `README`. SUSE s detaily o systému SUSE LINUX, výše zmíněných RFC a dvou úvodních HTML dokumentů. Programátoři, kteří mají zájem o využití služeb SLP, by si měli nainstalovat balíček `openslp-devel`, ve kterém je programátorská příručka (*Programmers Guide*).





# DNS Domain Name System

Síťová služba DNS (*Domain Name Service*) se používá k překladu doménových jmen a jmen počítačů na odpovídající IP adresy. Tím se například jménu počítače zeme přiřadí IP adresa 192.168.0.1. Před spuštěním vlastního nameserveru si nastudujte obecné informace o DNS v části 22.3 na straně 381. Následující příklad konfigurace se týká nameserveru BIND.

## 24.1 Konfigurace pomocí YaST

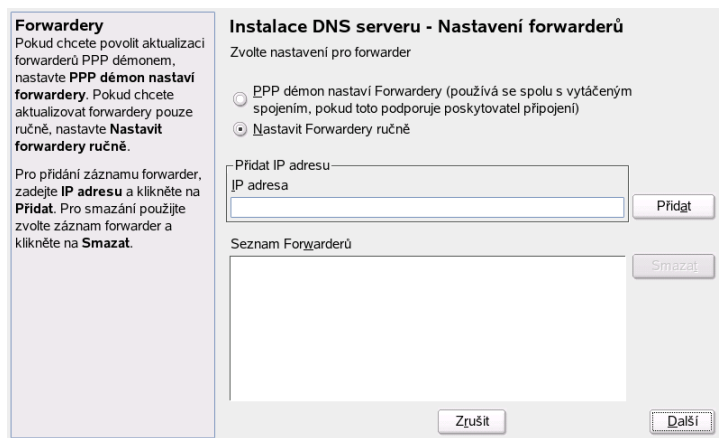
DNS modul nástroje YaST lze použít ke konfiguraci DNS serveru pro lokální síť. Při prvním spuštění modulu se spustí průvodce základním nastavením serveru. Zodpovězením dotazů získáte jednoduchou ale funkční konfiguraci DNS serveru. V expertním režimu je možno nastavit pokročilejší volby.

### 24.1.1 Průvodce konfigurací

Průvodce nastavením sestává ze tří dialogů a umožňuje přechod do expertní konfigurace.

#### Instalace DNS serveru nastavení forwarderů

Při prvním spuštění modulu spatříte dialog zobrazený na obrázku 24.1 na následující straně. Umožňuje volbu mezi nastavením forwarderů pomocí PPP démona při vytáčeném spojení přes DSL nebo ISDN ('PPP démon nastaví forwardery') a manuálním nastavením forwarderů ('Nastavit forwardery ručně').



*Obrázek 24.1: Instalace DNS serveru Nastavení forwarderů*

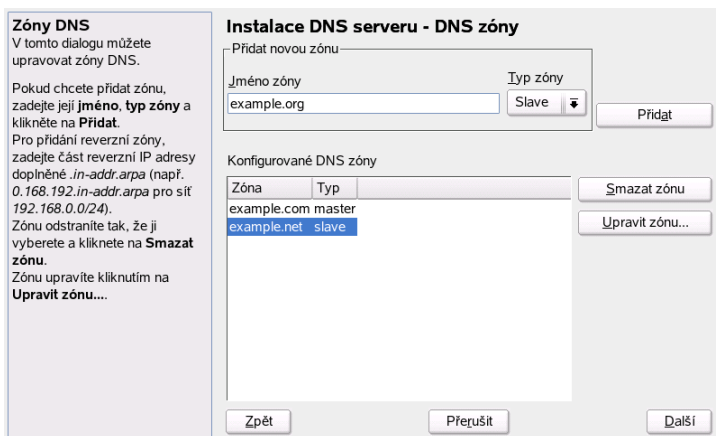
**DNS zóny** Tento dialog sestává z několika částí a je zodpovědný za správu souborů zón popsaných v části 24.7 na straně 421. Pro vytvoření nové zóny zadejte v položce 'Jméno zóny' její jméno. Chcete-li přidat reverzní zónu, musí jméno končit řetězcem `.in-addr.arpa`. Dále specifikujte 'Typ zóny' (master nebo slave) a klikněte na tlačítko 'Přidat'. Viz obrázek 24.2 na následující straně. Další nastavení zóny lze provést po kliknutí na tlačítko 'Upravit'. Chcete-li zónu odstranit, použijte tlačítko 'Smazat'.

**Dokončit průvodce** V posledním dialogu můžete ve firewallu otevřít port pro DNS a rozhodnout, zda má být DNS server automaticky spouštěn po startu systému. Lze odsud také přejít do expertního režimu konfigurace. Viz obrázek 24.3 na straně 412).

## 24.1.2 Expertní nastavení

V expertním režimu zobrazuje YaST okno s množstvím konfiguračních možností. Jejich nastavením získáte DNS server se všemi základními funkcemi:

**Spuštění** V položce 'Spuštění' nastavte, zda se má DNS server spouštět při startu systému automaticky nebo ručně. Chcete-li DNS server spustit okamžitě, stiskněte tlačítko 'Spustit DNS server'. Chcete-li jej zastavit, stiskněte



*Obrázek 24.2: Instalace DNS serveru: DNS zóny*

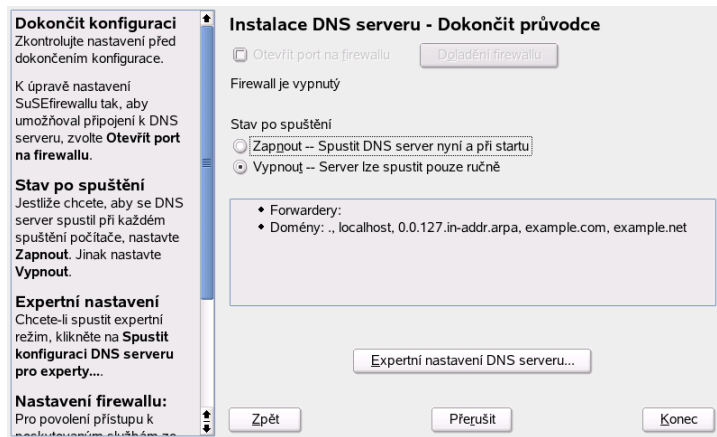
‘Zastavit DNS server’. Chcete-li uložit nastavení, stiskněte ‘Uložit nastavení a restartovat DNS server’.

Port pro DNS můžete na firewallu otevřít zaškrtnutím ‘Otevřít port na firewallu’. Změnit nastavení firewallu lze po stisknutí tlačítka ‘Doladění firewallu’.

**Forwardery** Jedná se o stejný dialog jako je ten, který se objeví po spuštění průvodce (viz kapitola na straně 409).

**Logování** V této sekci můžete nastavit co a jak má DNS server zapisovat do logů (protokolových souborů). V položce ‘Typ logování’ vyberte kam má DNS server logy zapisovat. Na výběr je mezi systémovým logem `/var/log/messages` (vyberte ‘Zapisovat do syslog’) a libovolným jiným souborem (vyberte ‘Zapisovat do souboru’, specifikujte jméno souboru, jeho maximální povolenou velikost a počet verzí souboru, který bude uchováván).

V položce ‘Další logování’ můžete zaškrtnout následující volby: ‘Logovat dotazy’ zapisuje *veškeré* dotazy klientů, což může způsobit extrémní nárůst velikosti souboru. Proto aktivace této volby bývá rozumná pouze pro účely ladění. Volba ‘Logovat aktualizace zón’ zapisuje datové přenosy při aktualizaci zón mezi DHCP a DNS servery. Chcete-li zapisovat přenosy mezi



*Obrázek 24.3: Instalace DNS serveru Dokončit průvodce*

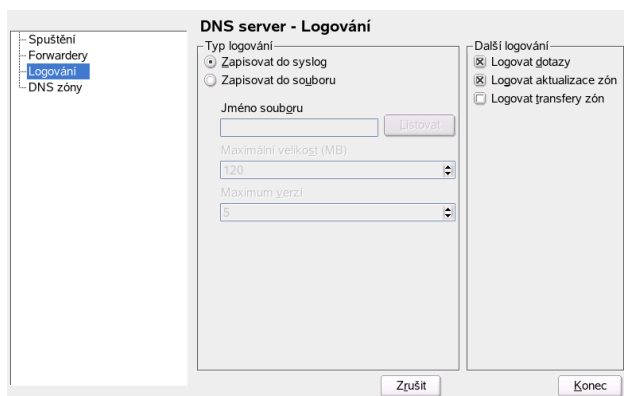
primárním a sekundárním serverem (master, slave), aktivujte volbu ‘Logovat transfery zón’. Viz obrázek 24.4 na následující straně.

**DNS zóny** Tento dialog je popsán v části věnované průvodci konfigurací. Viz 24.1.1 na straně 409.

**Editor slave zón** Tento dialog se objeví, pokud v předchozím dialogu zvolíte možnost ‘Upravit’ pro některou slave zónu. V položce ‘Master DNS server IP’ nastavte IP adresu serveru, ze kterého má slave získávat data. Chcete-li povolit transport zón, zaškrtněte ‘Povolit transport zón’. Pro omezení přístupu k serveru vyberte ze seznamu ACL. Viz 24.5 na straně 414.

**Editor master zón** Tento dialog se objeví, pokud v dialogu popsaném v části na této straně zvolíte možnost ‘Upravit’ pro některou master zónu. Skládá se z několika karet: ‘Základní’ (ta je otevření první), ‘NS záznamy’, ‘MX záznamy’, ‘SOA’ a ‘Záznamy’.

**Editor zón (NS záznamy)** V tomto dialogu můžete nastavit alternativní nameservery. Ujistěte se, že je v seznamu uveden i váš vlastní nameserver. Nový nameserver přidáte tak, že zadáte adresu serveru do pole ‘Přidat nameserver’ a kliknete na ‘Přidat’. Viz 24.7 na straně 416.



Obrázek 24.4: DNS server Logování

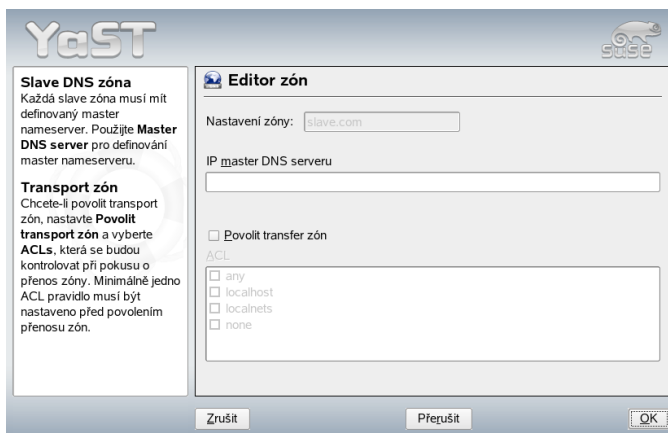
**Editor zón (MX záznamy)** Chcete-li pro zónu přidat poštovní server, zadejte do příslušných polí jeho adresu a prioritu. Potvrďte stisknutím tlačítka 'Přidat'. Viz obrázek 24.8 na straně 417.

**Editor zón (SOA)** Na této kartě můžete vytvořit záznamy SOA (*Start Of Authority*). Vysvětlení jednotlivých voleb naleznete v části 24.6 na straně 422. Změny SOA záznamů nejsou podporovány pro dynamické zóny spravované přes LDAP.

**Editor zón (Záznamy)** Na této kartě se nastavuje překlad jmen. V položce 'Klíč záznamu' zadejte jméno, v rozbalovací nabídce vpravo vyberte jeho typ. 'A-Překlad doménového jména' představuje hlavní záznam. Jeho hodnotou by měla být IP adresa. 'CNAME' je alias pro doménové jméno. 'NS' a 'MX' použijte pro záznamy rozšiřující informace zadané na kartách 'NS záznamy' a 'MX záznamy'. Hodnotou pro poslední tři typy je existující A záznam. Typ 'PTR' je určen pro reverzní zóny. Je opakem A záznamu.

## 24.2 Spuštění nameserveru BIND

Nameserver BIND (*Berkeley Internet Name Domain*) je v SUSE Linuxu již předkonfigurovaný, takže ho můžete spustit ihned po instalaci. Pokud máte fungující internetové připojení a do `/etc/resolv.conf` jako adresu nameserveru



Obrázek 24.5: DNS server Editor slave zón

pro `localhost` vložíte `127.0.0.1`, máte k dispozici překlad jmen na IP adresy bez nutnosti znát IP adresu DNS serveru poskytovatele připojení. BIND tak ale provádí překlad jmen prostřednictvím root nameserveru, což je výrazně pomalejší. Výhodnější je uvést IP adresu DNS serveru poskytovatele do konfiguračního souboru `/etc/named.conf` v položce `forwarders`. Získáte tak efektivní a bezpečný překlad. Takto nastavený nameserver běží v tzv. *caching-only* režimu. Skutečným DNS serverem se stane v případě, že nastavíte příslušné zóny.

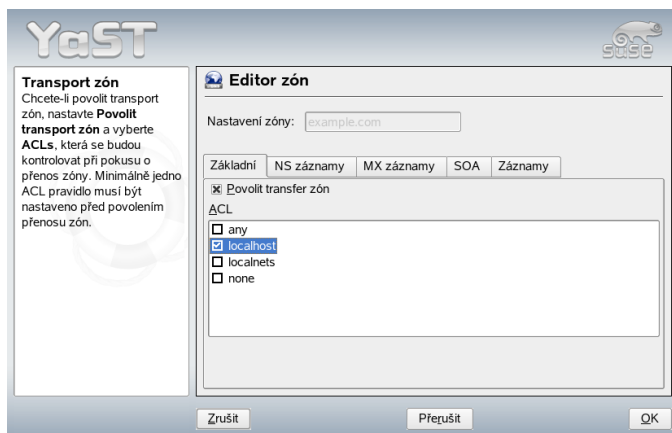
## Tip

### Automatické přizpůsobení informací o nameserveru

Informace o nameserveru lze, v závislosti na typu internetového nebo síťového připojení, automaticky přizpůsobovat aktuální situaci. Tuto vlastnost aktivujete nastavením proměnné `MODIFY_NAMED_CONF_DYNAMICALLY` v souboru `/etc/sysconfig/network/config` na `yes`.

## Tip

Nezřizujte však oficiální domény, které nemáte řádně registrovány. Nečiňte tak ani pokud jste sice vlastníky domény, ale tu spravuje poskytovatel, protože BIND nebude forwardovat (přeposílat dále) dotazy na tuto doménu. Takže třeba webový server umístěný u poskytovatele nebude pro vlastní doménu přístupný.



**Obrázek 24.6:** DNS server Editor zón (Základní)

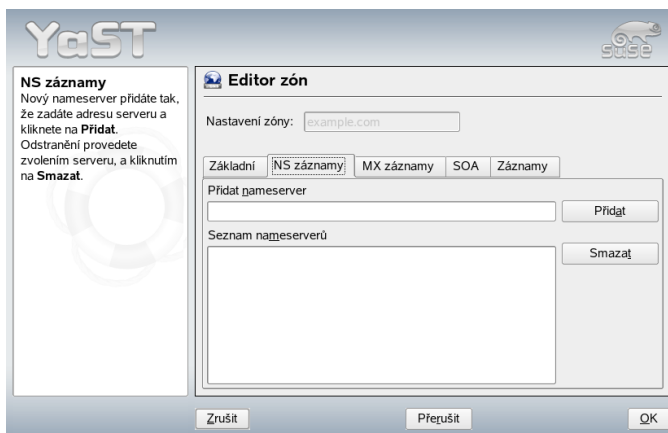
Nameserver může spustit uživatel `root` příkazem `rcnamedstart`. Pokud se vpravo zobrazí zeleně `done`, spustil se úspěšně proces nameserveru `named`.

Na lokálním počítači je možné fungování nameserveru ihned vyzkoušet programy `host` nebo `dig`, které by jako výchozí server měly vrátit `localhost` s adresou `127.0.0.1`. Pokud tomu tak není, pak je pravděpodobně v `/etc/resolv.conf` uveden špatný nameserver nebo tento soubor vůbec neexistuje. Zkuste příkaz `host 127.0.0.1`, který by měl fungovat vždy. Pokud se zobrazí chybové hlášení, otestujte příkazem `rcnamedstatus`, zda `named` vůbec běží. Jestliže nameserver není spuštěn nebo vykazuje chybné chování, naleznete obvykle příčinu v protokolovém souboru `/var/log/messages`.

Chcete-li používat nameserver poskytovatele nebo vlastní nameserver běžící ve vlastní síti jako forwarder, pak je třeba v části `options` mezi `forwarders` uvést jeho/jejich IP adresy. Adresy uvedené v příkladu 24.1 na této straně jsou pouze ukázkové.

**Příklad 24.1:** Volby pro přeposílání v souboru `named.conf`

```
options {
    directory "/var/lib/named";
    forwarders { 10.11.12.13; 10.11.12.14; };
    listen-on { 127.0.0.1; 192.168.0.99; };
```



*Obrázek 24.7: DNS server Editor zón (NS záznamy)*

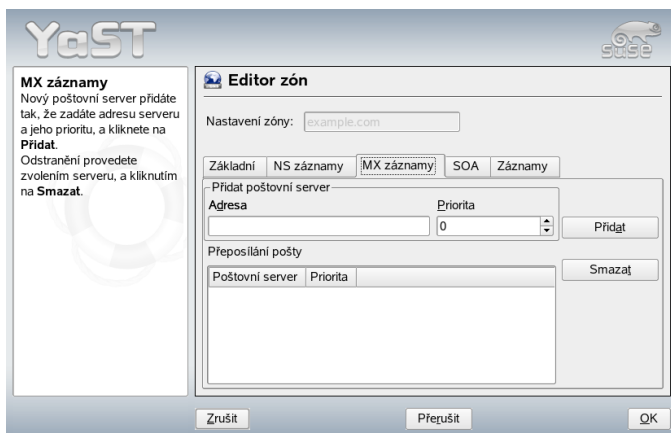
```
allow-query { 127/8; 192.168.0/24; };
notify no;
};
```

Položka `options` je následována položkami pro jednotlivé zóny, `localhost`, `0.0.127.in-addr.arpa` a položkou `type hint` pod `.`, která by měla být vždy přítomná. Příslušné soubory není nutno měnit a měly by pracovat tak, jak jsou. Ujistěte se, že je každá položka ukončena znakem `;`, a že jsou správně umístěny složené závorky. Změníte-li soubor `/etc/named.conf` nebo soubor zóny, přikazte programu BIND pomocí příkazu `rndc reload`, aby soubor znovu načtl. Dosáhnete toho také zastavením a novým spuštěním serveru příkazem `rndc restart`. Server můžete zastavit také příkazem `rndc stop`.

## 24.3 Konfigurační soubor `/etc/named.conf`

Všechna nastavení pro BIND se provádějí v souboru `/etc/named.conf`. Nicméně data pro zóny, jako názvy počítačů, IP adresy atd. jsou uloženy v se-





**Obrázek 24.8:** DNS server Editor zón (MX záznamy)

parátních souborech v adresáři `/var/lib/named`. Bližší informace jsou uvedeny v následujícím textu.

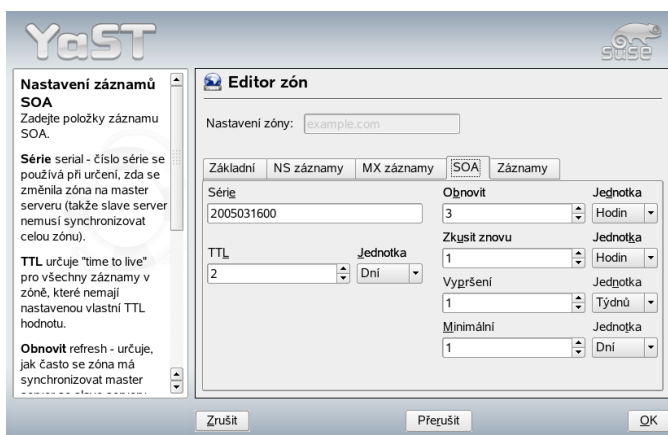
Konfigurační soubor `/etc/named.conf` se dělí na dvě oblasti. Obecná nastavení jsou v části `options`, v části `zone` jsou položky pro jednotlivé domény. Kromě toho je zde volitelně také oblast `logging` a položky typu `acl` (Access Control List). Komentáře začínají znakem `#` či znaky `//`. Minimalistický `/etc/named.conf` je uveden v příkladu 24.2 na této straně.

### *Příklad 24.2: Jednoduché nastavení souboru /etc/named.conf*

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
```



Obrázek 24.9: DNS server Editor zón (SOA)

```
type master;
file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};
```

## 24.4 Nejdůležitější konfigurační volby v sekci options

**directory** "<adresář>"; Udává adresář, ve kterém BIND hledá soubory s daty o jednotlivých zónách. Obvykle je to adresář /var/lib/named.

**forwarders** { <IP adresa>; }; Určuje IP adresy jednoho nebo více nameserverů (většinou nameserverů poskytovatele), na které jsou DNS dotazy přeposílány v případě, že je není možné zodpovědět přímo. Řetězec <IP adresa> nahradí IP adresou, např. 10.0.0.1.

**forward first;** Tato volba způsobuje, že je DNS dotaz ihned, před dotazováním na root nameserveru, přeposílán. Místo `forward first` lze použít `forward only`, pak nebude root nameserver dotazován vůbec.

**listen-on port 53 { 127.0.0.1; <IP adresa>; };**

Tato položka sděluje BINDu, na kterém síťovém rozhraní a portu má poslouchat dotazy klientů. `port 53` je standardní a není třeba jej explicitně uvádět. Zadáním adresy `127.0.0.1` povolíte dotazy z počítače `localhost`. Pokud je tato položka zcela vynechána, jsou standardně použita všechna rozhraní.

**listen-on-v6 port 53 {any; };** Tato položka sděluje BINDu, aby naslouchal clientským požadavkům přes protokol IPv6. Jedinou alternativou k `any` je `none` (nenaslouchat IPv6 požadavkům). Server akceptuje pouze IPv6 adresy typu `wild card`.

**query-source address \* port 53;** Tato volba se používá pokud firewall blokuje externí DNS dotazy. BIND pak komunikuje přes port 53 a ne přes porty vyšší než 1024.

**query-source-v6 address \* port 53;** Tato volba určuje, jaký port má být použit pro IPv6 dotazy.

**allow-query { 127.0.0.1; <sítě>; };** Tato volba určuje sítě, ze kterých mohou klienti posílat DNS dotazy. Řetězec `<sítě>` nahraďte adresou, např. `192.168.1/24`. Číslo `/24` je zkrácený zápis síťové masky `255.255.255.0`.

**allow-transfer { ! \*; };** Tato volba řídí, které počítače mohou požadovat transfer zóny. V uvedeném příkladu jsou takové požadavky zcela zakázány pomocí `! *`. Pokud by zde tato položka nebyla, bylo by možné provádět transfer zóny odkudkoliv a bez omezení.

**statistics-interval 0;** Bez této položky generuje BIND každou hodinu několik řádků do protokolového souboru `/var/log/messages`. Nula potlačuje tento výstup, jinak je možné uvádět čas v minutách.

**cleaning-interval 720;** Tato položka určuje, v jakém časovém odstupu bude BIND mazat svou cache (vyrovnávací paměť). Smazání cache vždy vygeneruje zápis do `/var/log/messages`. Čas se udává v minutách a výchozí hodnotou je 60 minut

**interface-interval 0;** BIND pravidelně prohledává síťová rozhraní a hledá nová či odpojená rozhraní. Nula zamezí tomuto hledání a BIND bude pracovat

pouze s rozhraními, která nalezne při startu. Čas se udává v minutách a výchozí hodnotou je 60 minut.

**notify no;** Parametr `no` zabraňuje informování ostatních nameserverů při změně data pro zónu nebo restartu nameserveru.

## 24.5 Konfigurace v sekci logging

BIND má široké možnosti protokolování (logování) různých událostí. Výchozí nastavení by mělo vyhovovat ve většině případů. Příklad 24.3 na této straně obsahuje nejjednodušší možnou formu nastavení a zakazuje logování zcela:

*Příklad 24.3: Položka zakazující protokolování*

```
logging {  
    category default { null; };  
};
```

## 24.6 Struktura souboru odkazujícího na data pro zóny

*Příklad 24.4: Data zóny moje-domena.cz*

```
zone "moje-domena.cz" in {  
    type master;  
    file "moje-domena.zone";  
    notify no;  
};
```

Za `zone` je uveden název spravované domény, zde tedy `moje-domena.cz`, následovaný `in` a složenými závorkami, které obsahují volby pro tuto zónu (viz 24.5 na následující straně). Chcete-li definovat sekundární (*slave zone*), změňte `type` na `slave` a uveďte nameserver, který spravuje zónu jako `master` (ale sám může být `slave` jiného serveru).

### Příklad 24.5: Data zóny *jina-domena.cz*

```
zone "jina-domena.cz" in {  
    type slave;  
    file "slave/jina-domena.zone";  
    masters { 10.0.0.1; };  
};
```

Volby pro nastavení zón:

**type master;** Volba `master` určuje, že je zóna spravována lokálním nameserverem. To předpokládá správně vytvořený soubor pro zónu.

**type slave;** Zóna je transferována z jiného nameserveru. Volba musí být použita společně s volbou `masters`.

**type hint;** Zóna `.` typu `hint` se používá pro specifikaci root nameserveru. Můžete ponechat výchozí nastavení.

**file "moje-domena.zone" nebo "slave/jina-domena.zone";**

Tato volba specifikuje soubor, ve kterém jsou uložena data pro doménu. V případě zóny typu `slave` není potřeba, neboť potřebné údaje jsou získány z jiného nameserveru. Aby byly primární (master) a sekundární (slave) soubory odlišeny, používá se pro sekundární soubory zvláštní adresář `slave`.

**masters { <IP adresa serveru>; };** Tuto položku je třeba uvádět pouze u sekundárních (slave) zón. Specifikuje nameserver, ze kterého jsou získávána data o zóně.

**allow-update { ! \*; };** Tato volba určuje práva externích uživatelů pro zápis do konfigurace. To je obvykle z bezpečnostních důvodů nevhodné. Chybí-li tato položka, nebo je-li použit zápis uvedený výše, je zápis zakázán.

## 24.7 Struktura souboru s daty pro zónu

Používají se dva druhy souborů s daty zóny. Jedny slouží pro přiřazení IP adresy počítačům a druhé pak pro reverzní převod, tedy pro přiřazení názvu počítače k IP adrese.

## Tip

### Použití tečky v souborech s daty zóny

V souborech s daty zóny má velký význam tečka (.). Jsou-li názvy počítačů uvedeny bez tečky na konci, je vždy doplňována zóna. Proto je třeba již kompletní názvy počítačů uvedené i s doménou ukončit tečkou tak, aby nebyla doména uvedena dvakrát. Chybějící tečky nebo jejich špatné umístění jsou často příčinou chyb v konfiguraci nameserveru.

## Tip

Ukážeme si soubor `world.zone` odpovědný za doménu `world.cosmos`:

### *Příklad 24.6: Soubor `/var/lib/named/world.zone`*

```
1 $TTL 2D
2 world.cosmos. IN SOA      gateway root.world.cosmos. (
3                       2003072441 ; serial
4                       1D         ; refresh
5                       2H         ; retry
6                       1W         ; expiry
7                       2D )       ; minimum
8
9                       IN NS      gateway
10                      IN MX      10 sun
11
12 gateway             IN A        192.168.0.1
13                      IN A        192.168.1.1
14 sun                  IN A        192.168.0.2
15 moon                 IN A        192.168.0.3
16 earth                IN A        192.168.1.2
17 mars                 IN A        192.168.1.3
18 www                  IN CNAME    moon
```

**Řádek 1:** `$TTL` definuje standardní délku platnosti TTL (*Time To Live*), která platí pro všechny položky v tomto souboru. V našem případě jsou to dva dny (2D).

**Řádek 2:** Zde začíná SOA záznam:

- Na prvním místě je uveden název spravované domény `world.cosmos` ukončený tečkou (jinak by zóna byla přidána ještě jednou. Alternativním řešením je použití zavináče (@), který znamená použití zóny z `/etc/named.conf`.
- Za `IN SOA` je uveden název primárního (*master*) nameserveru pro danou zónu. Jméno `gateway` bude rozšířeno na `gateway.world.cosmos`, protože není ukončeno tečkou.
- Následuje e-mailová adresa osoby odpovědné za nameserver. Protože zavináč má v tomto souboru zvláštní význam, používá se místo něj tečka. Adresa `root@world.cosmos` se tedy zapíše jako `root.world.cosmos..` Na konci je opět nutné uvést tečku.
- Řádka končí levou závorkou (, která uzavírá, spolu s následující pravou závorkou ), řádky tvořící SOA záznam.

**Řádek 3:** Obsahuje tzv. sériové číslo (*serial number*), které se má při každé změně v souboru zvýšit. Slouží sekundárním nameserverům pro porovnávání konfigurace s primárním nameserverem. Jako formát čísla se ujal `YYYYMMDDNN`.

**Řádek 4:** Položka `refresh rate` udává časový interval, po jehož uplynutí sekundární server kontroluje `serial number` na primárním serveru. V našem případě jeden den (1D).

**Řádek 5:** Položka `retry rate` udává časový interval, po jehož uplynutí se sekundární server opět pokusí kontaktovat primární server v případě, že se původní kontakt z důvodu chyby neuskutečnil. Zde dvě hodiny (2H).

**Řádek 6:** Položka `expiration time` udává dobu, po jejímž uplynutí sekundární nameserver smaže data z cache, pokud nemůže kontaktovat primární server. Zde jeden týden (1W).

**Řádek 7:** Poslední SOA položka určuje tzv. `negative caching TTL`, čas po který mají ostatní servery uchovávat v cache negativně vyřízené dotazy.

**Řádek 9:** Položka `IN NS` udává nameserver odpovědný za doménu. Také zde platí, že `gateway` expanduje na `gateway.world.cosmos`, protože je bez tečky na konci. Řádků podobných tomuto může být více, jeden pro primární a další pro sekundární nameservery. Pokud není `notify` v souboru `/etc/named.conf` nastaven na `no`, pak budou všechny zde uvedené nameservery informovány o změnách dat zóny.

**Řádek 10:** MX záznam určuje poštovní server pro doménu `world.cosmos`. Tento server poštu přijímá a dále zpracovává, resp. přeposílá. V uvedeném příkladě to je server `sun.world.cosmos`. Kromě názvu serveru se uvádí preferenční hodnota (zde 10) v případě většího počtu MX položek bude pošta zaslána serveru s nejnižším číslem a teprve při problémech s doručením bude použit server s vyšší hodnotou.

**Řádky 12 až 17:** Zde jsou uvedeny vlastní adresní záznamy přiřazující jménům počítačů IP adresy. Názvy počítačů jsou uváděny bez tečky a budou tak rozšířeny o doménu. Více IP adres se používá u počítačů, které mají více síťových karet. Pokud je použita tradiční (IPv4) adresa, je záznam označen písmenem A. Záznamy s IPv6 adresou jsou označeny jako A6. (Dříve se IPv6 adresy označovaly jako AAAA, což je již zastaralé.)

**Řádek 18:** Alias `www` je použit k adresování počítače `moon` (CNAME = *canonical name*).

Pro *reverzní převod* (*reverse lookup*) IP adres na názvy počítačů se používá pseudodoména `in-addr.arpa`. Je připojena k obrácenému zápisu adresy. Ze `192.168.1` se tak stane `1.168.192.in-addr.arpa`, viz příklad 24.7 na této straně.

### *Příklad 24.7: Zpětný převod*

```
1 $TTL 2D
2 1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
3                               2003072441           ; serial
4                               1D                     ; refresh
5                               2H                     ; retry
6                               1W                     ; expiry
7                               2D )                   ; minimum
8
9                               IN NS                 gateway.world.cosmos.
10
11 1                               IN PTR               gateway.world.cosmos.
12 2                               IN PTR               earth.world.cosmos.
13 3                               IN PTR               mars.world.cosmos.
```

**Řádek 1:** Položka `$TTL` definuje standardní délku platnosti TTL (*Time To Live*), která platí pro všechny položky v tomto souboru. V našem případě jsou to dva dny (2D).

**Řádek 2:** Reverzní převod je nastaven pro síť `192.168.1.0`. Protože se zde zóna nazývá `1.168.192.in-addr.arpa`, nechceme ji připojovat za názvy počítačů, a proto je píšeme celé včetně domény a s tečkou na konci.



**Řádek 3-7:** Viz předchozí příklad pro `world.cosmos`.

**Řádek 9:** I zde je uveden nameserver, který odpovídá za zónu. Tentokrát je uveden včetně domény a s tečkou na konci.

**Řádek 11-13:** Pointer záznamy, které uvádějí k IP adrese náležející názvy počítačů. Uvádí se pouze poslední pozice IP adresy bez tečky. Připojením zóny (bez `.in-addr.arpa`) vznikne kompletní IP adresa v obráceném pořadí.

Přenosy zón mezi různými verzemi BINDu by měly být bezproblémové.

## 24.8 Dynamická aktualizace údajů o zóně

Termín *dynamická aktualizace* se vztahuje na mechanismy, kterými jsou záznamy v souborech zón na primárním (master) serveru přidávány, měněny nebo mazány. Tyto mechanismy jsou popsány v dokumentu RFC2136. Dynamická aktualizace je pro každou zónu nastavována individuálně přidáním volitelného pravidla `allow-update` nebo `update-policy`. Dynamicky aktualizované zóny by neměly být upravovány ručně.

Záznamy, které se mají na serveru aktualizovat, přenesete příkazem `nsupdate`. Přesná syntaxe je popsána v manuálové stránce (`man 8 nsupdate`). Z bezpečnostních důvodů by všechny aktualizace měly být prováděny s využitím TSIG klíčů popsaných v kapitole 24.9 na této straně.

## 24.9 Bezpečné transakce

Bezpečné transakce lze zajistit pomocí transakčních signatur (TSIG) založených na sdílených tajných klíčích (TSIG klíčích). V této sekci je popsáno, jak tyto klíče vytvořit a používat.

Bezpečné transakce jsou potřeba pro komunikaci mezi různými servery a pro dynamickou obnovu zónových dat. Kontrola pomocí klíčů je mnohem bezpečnější než pouhá kontrola pomocí IP adres.

TSIG klíč můžete vygenerovat následujícím příkazem (podrobnosti viz `mandnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

Tím se vytvoří dva soubory se jmény podobnými následujícím:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

Samotný klíč (např. řetězec `ejIkuCyyGJwwuN3xAteKgg==`) se nachází v obou souborech. Aby mohl být používán pro transakce, musí být druhý soubor (`Khost1-host2.+157+34265.key`) přenesen na vzdálený počítač (nejlépe bezpečnou cestou, např. pomocí `scp`). Na vzdáleném serveru musí být vložen do souboru `/etc/named.conf`, čímž se umožní bezpečná komunikace mezi oběma počítači (`host1` a `host2`):

```
key host1-host2. {  
    algorithm hmac-md5;  
    secret "ejIkuCyyGJwwuN3xAteKgg==";  
};
```

## Varování

### Přístupová práva k `/etc/named.conf`

Ujistěte se, že přístupová práva k souboru `/etc/named.conf` jsou správně nastavena (a omezena). Výchozí práva pro tento soubor jsou `0640`, vlastníkem souboru je `root` a skupina je `named`. Jinou možností je přesunout klíče do jiného souboru s patřičně nastavenými právy, který se pak do souboru `/etc/named.conf` vkládá.

## Varování

Aby mohl server `host1` používat klíč pro `host2` (jehož adresa je `192.168.2.3`), musí soubor `/etc/named.conf` na serveru obsahovat následující pravidlo:

```
server 192.168.2.3 {  
    keys { host1-host2. ; };  
};
```

Obdobné nastavení je třeba učinit i v konfiguračních souborech na počítači `host2`.

Kromě seznamů správy přístupu (ACL, *Access Control Lists* neplést s ACL souborového systému) definovaných pro jednotlivé IP adresy a rozsahy adres přidejte pro zvýšení bezpečnosti TSIG klíče. Příslušný záznam v konfiguraci by měl vypadat asi takto:

```
allow-update { key host1-host2. ;};
```

K tomuto tématu naleznete více informací v příručce *BIND Administrator Reference Manual* v části `update-policy`.

## 24.10 DNSSEC

DNSSEC, bezpečné DNS, je popsáno v RFC2535. Nástroje pro práci s DNSSEC jsou probírány v BIND manuálu.

Bezpečná zóna musí mít přiřazen jeden nebo více zónových klíčů, generovaných pomocí `dnssec-keygen`, stejně jako klíče počítačů. V současnosti se pro tvorbu klíčů používá algoritmus DES. Veřejné klíče by měly být vloženy do příslušného zónového souboru pomocí pravidla `$INCLUDE`.

Příkazem `dnssec-makekeyset` jsou všechny klíče spojeny do jedné sady, která pak musí být bezpečným způsobem přenesena do rodičovské (nadřazené) zóny. Tam je sada podepsána pomocí `dnssec-signkey`. Soubory generované tímto příkazem jsou použity k podepsání zón pomocí `dnssec-signzone`, čímž jsou vytvořeny soubory, které se vloží do `/etc/named.conf` každé zóny.

## 24.11 Další informace

Další informace naleznete v příručce *BIND Administrator Reference Manual* nainstalované v adresáři `/usr/share/doc/packages/bind/`. Zvažte i studium RFC dokumentů zmiňovaných v tomto manuálu a příslušných manuálových stránek. Soubor `/usr/share/doc/packages/bind/README`. SuSE obsahuje aktuální informace o BINDu v systému SUSE LINUX.



# NIS Network Information Service

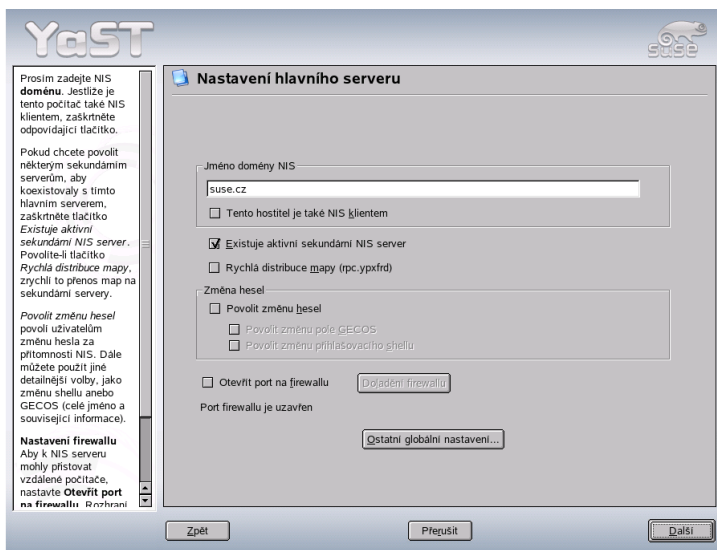
Jakmile přistupuje v síti více unixových počítačů ke společným prostředkům, je třeba zajistit, aby bylo všude společné označení uživatelů a skupin. Síť musí být pro každého uživatele transparentní – ať pracuje na kterémkoli z těchto počítačů, vždy by měl najít stejné prostředí. To umožňují služby *NIS* a *NFS*. *NFS* slouží pro přístup k souborovým systémům přes síť a je popsán v kapitole 26 na straně 435.

*NIS* (Network Information Service) je databázová služba poskytující síťový přístup k obsahu souborů `/etc/passwd`, `/etc/shadow` a `/etc/group`. *NIS* lze použít i k dalším účelům (např. pro zpřístupnění souborů `/etc/hosts` nebo `/etc/services`), ale to je nad rámec tohoto textu. *NIS* se také často nazývá *YP* (žluté nebo zlaté stránky).

## 25.1 Konfigurace NIS serveru

Konfiguraci zahájíte výběrem YaST modulu ‘NIS server’ v části ‘Síťové služby’. Pokud ve vaší síti dosud neexistuje žádný NIS server, zvolte v dialogu ‘Instalovat a nastavit NIS hlavní server’. Pokud již NIS server máte (hlavní server, master), můžete přidat sekundární (slave) NIS server (např. pro konfiguraci nové podsítě). Nejprve popíšeme konfiguraci hlavního serveru.

Pokud některé balíčky chybí, vložte instalační zdroj, doinstalují se automaticky. V horní části dialogu (viz 25.1 na následující straně) zadejte jméno domény. Zařazením položky ‘Tento hostitel je také NIS klientem’ zvolte, zda má být server zároveň i NIS klientem (to umožňuje uživatelům přihlašovat se a přistupovat k datům z NIS serveru).

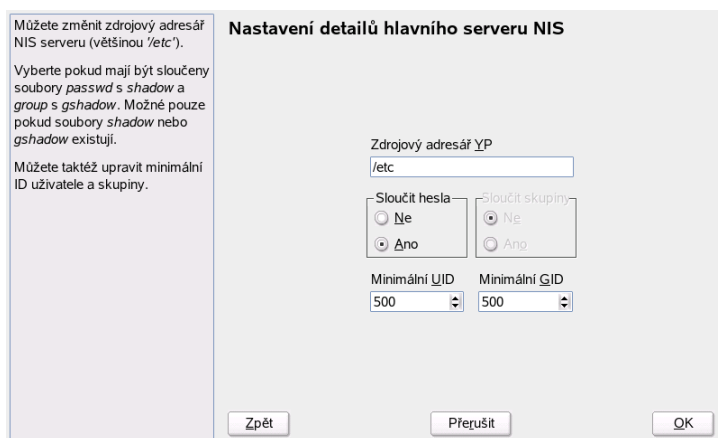


*Obrázek 25.1: Nástroj pro nastavení NIS serveru*

Pokud chcete později v síti vytvořit sekundární NIS server (slave), nezapomeňte zaškrtnout tlačítko ‘Existuje aktivní sekundární NIS server’. Kromě toho byste měli zapnout i položku ‘Rychlá distribuce mapy’, která zajistí velmi rychlý přenos informací z hlavního (master) NIS serveru na sekundární (slave).

Jestliže chcete uživatelům v síti (lokálním i spravovaným pomocí NIS serveru) povolit změnu vlastních hesel uložených na NIS serveru (příkazem `yppasswd`), vyberte ‘Povolit změnu hesel’. ‘Povolit změnu pole GECOS’ umožní uživatelům měnit i nastavení jména a adresy (příkazem `ypchfn`). ‘Povolit změnu přihlašovacího shellu’ dovoluje uživatelům zvolit přihlašovací shell příkazem `ypchsh` (např. `sh` místo `bash`).

Tlačítkem ‘Ostatní globální nastavení’ přejdete do dialogu ‘Nastavení detailů hlavního serveru NIS’ (viz obrázek 25.2 na následující straně), kde můžete změnit zdrojový adresář NIS serveru (výchozím adresářem je `/etc`). Aby byly synchronizovány soubory `/etc/passwd` a `/etc/shadow` nebo `/etc/group` a `/etc/gshadow`, zvolte ‘Ano’. Nastavit můžete i minimální ID uživatele a skupiny. Nastavení potvrdíte kliknutím na tlačítko ‘OK’. Vráťte se do původního dialogu, kde můžete pokračovat stisknutím tlačítka ‘Další’.



**Obrázek 25.2:** Změna adresáře a synchronizace souborů NIS serveru

Pokud jste předtím aktivovali tlačítko 'Existuje aktivní sekundární NIS server', pak je třeba nyní uvést název či názvy počítačů, které budou fungovat jako sekundární servery. Pokračujte tlačítkem 'Další'. Pokud sekundární servery nepoužíváte, je tento dialog vynechán.

V dalším dialogu můžete upravit mapy, které budou z NIS serveru přeneseny na klienty. Výchozí nastavení většinou není třeba měnit.

Stisknutím tlačítka 'Další' se přenesete do dalšího dialogu (viz 25.3 na následující straně). V něm nastavte, které sítě mohou přistupovat k NIS serveru. Obvykle je to vaše vnitřní síť. V takovém případě nastavte následující dvě položky:

```
255.0.0.0      127.0.0.0
0.0.0.0        0.0.0.0
```

První položka umožňuje přístup z vašeho počítače (NIS serveru). Druhá umožňuje přístup každému, kdo má přístup do lokální sítě.

Prosím zadejte, kterým hostitelům je povoleno dotazovat se NIS serveru.

Adresa bude povolena, pokud **síť** odpovídá bitovému AND adresy *hostitele* a *síťové masky*.

Musí existovat položka se *síťovou maskou* 255.0.0.0 a *sítí* 127.0.0.0, pokud mají být povolena spojení z localhost.

Pokud je zadána *síťová maska* 0.0.0.0 a *síť* 0.0.0.0, povolíte tím přístup všem hostitelům.

### Nastavení dotazů hostitelů na NIS servery

Síťová maska podsítě	Síť
255.0.0.0	127.0.0.0
0.0.0.0	0.0.0.0

Zpět

Přidat

Editovat

Smazat

Přerušit

Konec

*Obrázek 25.3: Nastavení přístupových práv k NIS serveru*

## Důležité

### Automatické nastavení firewallu

Pokud máte aktivovaný firewall (SuSEfirewall2) a zvolili jste ‘Otevřít port na firewallu’, YaST upraví nastavení firewallu pro NIS server povolením portmap služby.

## Důležité

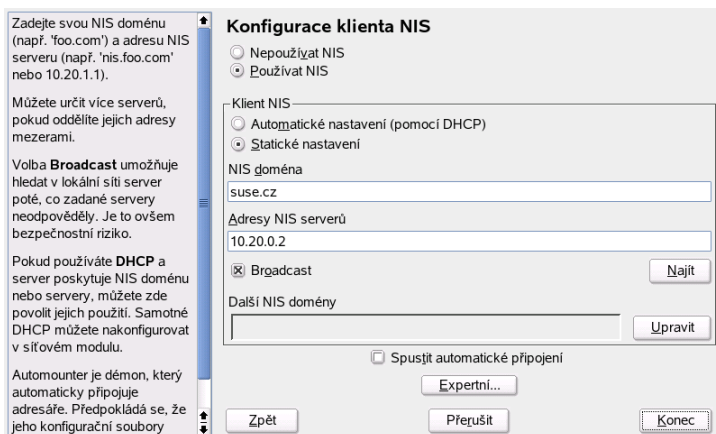
## 25.2 Konfigurace NIS klientů

Pro konfiguraci NIS klienta použijte YaST modul ‘Klient NIS’. Zvolíte-li používání NIS nebo, v závislosti na okolnostech, automounter, otevře se tento dialog. Zvolte, zda má stanice pevnou IP adresu nebo zda ji má získat z DHCP serveru. DHCP server nastaví také NIS doménu a NIS server. Více informací o DHCP najdete v části 27 na straně 441. V případě používání pevné IP adresy nastavte NIS doménu a NIS server ručně (viz 25.4 na následující straně). NIS server v síti můžete vyhledat pomocí volby ‘Najít’.



Zadat lze i více domén s tím, že jedna bude nastavena jako výchozí. K zadání dalšího serveru použijte tlačítko 'Upravit'.

Aby nebylo možné z jiného počítače zjistit, který NIS server vaše stanice používá, zakažte v expertním nastavení volbu 'Odpovídat vzdáleným počítačům'. Pokud zvolíte 'Poškozený server', může klient přijímat odpovědi serveru na neprivilegovaném portu. Další informace získáte v manuálové stránce ypbind.



*Obrázek 25.4: Nastavení domény a adresy NIS serveru*



# NFS sdílené souborové systémy

Jak již bylo uvedeno v kapitole 25 na straně 429, NFS (spolu s NIS) zabezpečují transparentnost sítě pro uživatele. NFS umožňuje počítačům sdílet souborové systémy v síti – uživatel pak vidí stejné prostředí nezávisle na tom, odkud se přihlásí.

Podobně jako NIS, představuje i NFS nesymetrickou službu – rozlišuje se NFS server a NFS klient. Počítač může vykonávat obě tyto úlohy, tj. exportovat do sítě své vlastní souborové systémy a připojovat (mount) souborové systémy jiných počítačů. Centrální server NFS mívá obvykle velkou diskovou kapacitu. Jednotliví klienti si z něho připojují povolené adresářové stromy ke svému souborovému systému.

---

## Důležité

### Potřeba DNS

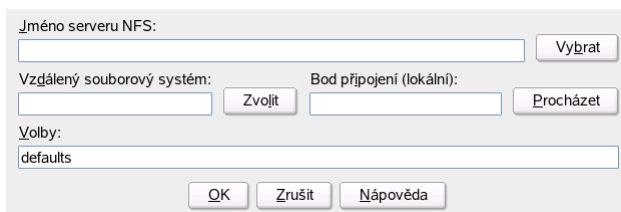
Teoreticky lze export provádět pouze pomocí IP adres. Abyste zabránili prodávám, potřebujete funkční DNS systém. Je to potřeba minimálně pro účely logování, neboť mountd démon provádí reverzní překlady.

---

## Důležité

## 26.1 Importování souborových systémů pomocí YaST2

Každý oprávněný uživatel může připojit NFS adresáře ke svému systému. Nejjednodušší je použít YaST modul 'Klient NFS'. Zvolte 'Přidat' a uveďte potřebné informace: jméno NFS serveru, adresář, který chcete importovat, a bod připojení (adresář), ve kterém se importovaná data zobrazí. Viz 26.1 na této straně.



*Obrázek 26.1: Nastavení NFS klienta v programu YaST*

## 26.2 Ruční import souborových systémů

Importovat systém souborů ze serveru NFS je snadné. Jediným předpokladem je, aby běžel RPC portmapper (může ho spustit uživatel `root` příkazem `rcportmapstart`). Je-li tento předpoklad splněn, lze vzdálené souborové systémy připojovat stejně snadno jako lokální souborové systémy příkazem `mount` s následující syntaxí:

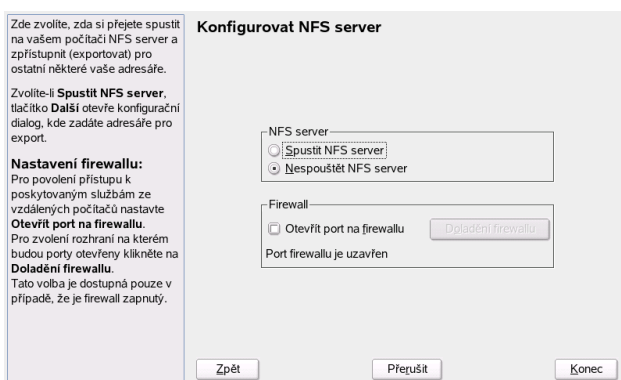
```
mount jmeno-serveru:vzdalena-cesta lokalni-cesta
```

Uživatelské adresáře ze serveru slunce se například importují následujícím příkazem:

```
mount slunce:/home /home
```

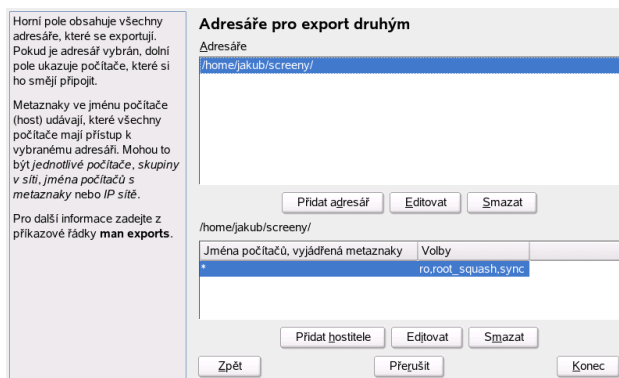
## 26.3 Exportování souborových systémů pomocí YaST

S pomocí programu YaST můžete svůj počítač proměnit v NFS server – server exportující adresáře a soubory na všechny ostatní počítače s povoleným přístupem. Lze tak poskytnout aplikace všem účastníkům v síti, aniž by bylo nutné tyto aplikace instalovat na jednotlivé pracovní stanice. Server nainstalujete tak, že spustíte YaST a zvolíte ‘Síťové služby’ → ‘NFS server’. Objeví se dialog zobrazený na obrázku 26.2 na této straně).



*Obrázek 26.2: Nástroj pro nastavení NFS serveru*

V dialogu zvolte položku ‘Spustit NFS server’ a klikněte na tlačítko ‘Další’. V horním poli se zadávají soubory a adresáře k exportu. Dolní pole je určeno pro seznam počítačů s povoleným přístupem. Dialog je zobrazen na obrázku 26.3 na následující straně. Klientské počítače lze specifikovat čtyřmi způsoby: jako jednotlivý počítač, skupinu v síti, jméno počítače s metaznaký nebo IP síť. Podrobný popis najdete v manuálové stránce `manexports`. Nastavení dokončíte kliknutím na ‘Konec’.



*Obrázek 26.3: Nastavení NFS serveru v programu YaST*

## Důležité

### Automatické nastavení firewallu

Pokud máte aktivovaný firewall (SuSEfirewall2) a zvolili jste 'Otevřít port na firewallu' v prvním dialogu, YaST automaticky upraví nastavení firewallu a povolí službu `nfs`.

## Důležité

## 26.4 Ruční export souborových systémů

Pokud nechcete pro konfiguraci NFS serveru použít YaST, ujistěte se, že na NFS serveru běží následující systémy:

- RPC portmapper (`portmap`)
- RPC mount démon (`rpc.mountd`)
- RPC NFS démon (`rpc.nfsd`)

Aby se tyto služby spouštěly při startu systému automaticky pomocí skriptů `/etc/init.d/portmap` a `/etc/init.d/nfsserver`, zadejte příkazy

`insserv/etc/init.d/nfsserver` a `insserv/etc/init.d/portmap`. V konfiguračním souboru `/etc/exports` určete, které souborové systémy mají být exportovány kterým počítačům.

Pro každý exportovaný adresář je potřeba jeden řádek, na kterém jsou specifikovány počítače, kterým se má exportovat, a jejich oprávnění. Automaticky jsou exportovány i všechny podadresáře. Oprávněné počítače se obvykle zadávají plnými jmény, včetně domény. Také je možno použít zástupné znaky (wildcards) jako `*` a `?` (chovají stejně jako v `bash`i). Pokud se nezadá žádný počítač, mohou adresář importovat všechny počítače, podle zadaných přístupových práv.

Přístupová práva se zadávají do závorek za jméno počítače. Nejdůležitější volby jsou ukázány v tabulce 26.1 na této straně.

*Tabulka 26.1: Přístupová práva k exportovaným souborům*

volba	význam
<code>ro</code>	Souborový systém se exportuje pouze pro čtení (výchozí).
<code>rw</code>	Souborový systém se exportuje pro čtení i zápis.
<code>root_squash</code>	Uživatel <code>root</code> daného počítače nemá rootovská práva pro tento souborový systém. Dosáhne se toho změnou <code>user-ID 0</code> na <code>user-ID 65534</code> , a to se přiřadí uživateli <code>nobody</code> (výchozí volba).
<code>no_root_squash</code>	Zachovat rootovská práva (opak předchozího).
<code>link_relative</code>	Nahradit absolutní symbolické odkazy (začínající <code>/</code> ) odpovídající posloupností <code>../</code> . Tato volba má smysl jen tehdy, je-li připojen úplný systém souborů počítače (výchozí volba).
<code>link_absolute</code>	Symbolické odkazy zůstávají nezměněny.
<code>map_identity</code>	Na klientovi budou stejná uživatelská ID jako na serveru (výchozí volba)
<code>map_daemon</code>	Klient a server nemají odpovídající ID uživatelů. To se sdělí programu <code>nfsd</code> , aby vytvořil konverzní tabulku pro ID. Předpokladem je spuštění démona <b>uidd</b>

Soubor `exports` může vypadat například tak, jak je uvedeno v příkladu 26.1 na této straně. Soubor `/etc/exports` je používán démoni `mountd` a `nfsd`. Pokud soubor změníte, `mountd` a `nfsd` restartujte příkazem `rcnfsserverrestart`.

***Příklad 26.1: /etc/exports***

```
#
# /etc/exports
#
/home          slunce(rw)    venuse(rw)
/usr/X11       slunce(ro)    venuse(ro)
/usr/lib/texmf slunce(ro)    venuse(rw)
/              zeme(ro,root_squash)
/home/ftp      (ro)
# End of exports
```



# DHCP

## 27.1 DHCP protokol

Protokol DHCP (*Dynamic Host Configuration Protocol*) umožňuje centrální nastavení sítě na serveru místo individuální konfigurace jednotlivých stanic. Klient, který používá DHCP, nemá kontrolu nad svou statickou IP adresou, ta je mu automaticky přidělována DHCP serverem.

Jednotlivé klienty je možné identifikovat podle hardwarové adresy síťové karty, tzv. MAC adresy, a tak jim, kdykoliv se spojí se serverem, přiřadit stejné nastavení. I přes dynamické přidělování IP adres je tak možno pro jednotlivé počítače zachovat stále stejné IP adresy (i když se počítače připojí až po delší době). Nefunguje to ale v případě, kdy je v síti více počítačů než adres; tehdy jsou adresy přidělovány podle potřeby.

Použití DHCP přináší dvě výhody. Zaprvé je možné jednoduše provádět i velice rozsáhlé změny v síti a spravovat všechny konfigurační soubory centrálně bez nutnosti individuální konfigurace všech klientů. Druhou výhodou je možnost velice jednoduchého připojování nových počítačů k síti. Připojovaným počítačům je automaticky přidělena IP adresa z vyčleněného adresního prostoru. To je požehnaní zejména pro notebooky, které se pravidelně připojují do různých sítí.

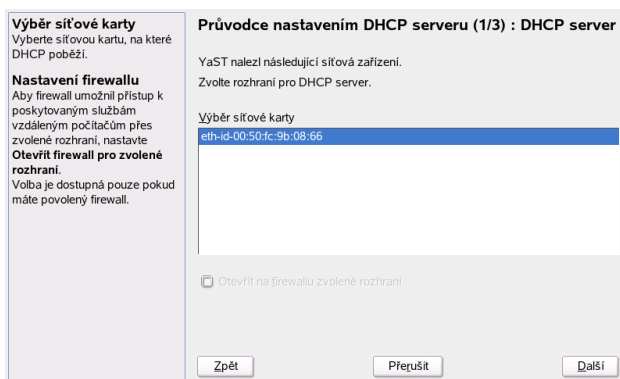
Kromě IP adres a síťových masek je možné spravovat také názvy počítačů a domén, používané brány a adresy nameserverů, které jsou pak sdělovány klientům. Navíc je možné centrálně konfigurovat i např. server pro synchronizaci času (xntp) nebo tiskový server.

## 27.2 Konfigurace DHCP serveru pomocí nástroje YaST

YaST DHCP modul umožňuje nastavit vlastní DHCP server pro lokální síť. Modul pracuje ve dvou různých režimech, jednoduchém a expertním:

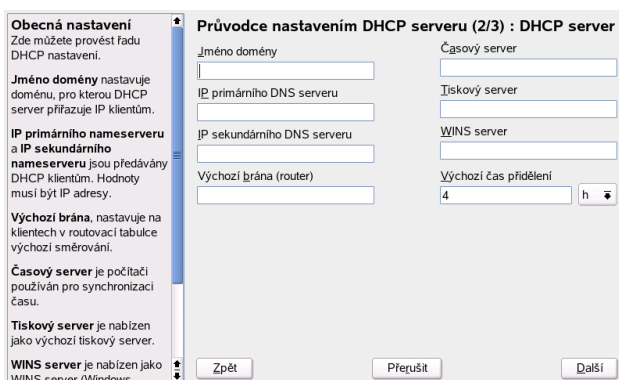
Při prvním spuštění modulu vyvolá YaST čtyřdílného průvodce, který vám pomůže provést základní konfiguraci DHCP serveru.

**Výběr síťové karty** V prvním kroku YaST zjistí, jaká jsou na vašem systému dostupná síťová rozhraní, a zobrazí jejich seznam. Ze seznamu vyberte rozhraní, na kterém má DHCP server naslouchat, a otevřete pro toto rozhraní firewall zaškrtnutím položky ‘Otevřít na firewallu zvolené rozhraní’. Viz 27.1 na této straně.



Obrázek 27.1: DHCP server: Výběr karty

**Obecná nastavení** V jednotlivých polích zadejte podrobnosti o klientech, které má DHCP server spravovat. Je třeba určit jméno domény, adresu časového serveru, adresu primárního a sekundárního DNS serveru, adresu tiskového serveru, WINS serveru (v případě smíšené sítě zahrnující počítače se systémem Linux i Windows), adresu výchozí brány a výchozí čas přidělení adresy. Viz 27.2 na následující straně.



Obrázek 27.2: DHCP server: Obecná nastavení

**Dynamické DHCP** V tomto kroku nastavte, jak mají být klientům přiřazovány dynamické IP adresy. Určete rozsah, ze kterého budou adresy přidělovány. Všechny adresy musejí mít stejnou masku. Nastavte rovněž dobu přidělení adresy, po jejímž uplynutí musí počítač zažádat o prodloužení přidělení. Můžete také určit maximální dobu, po kterou je IP na serveru blokována pro klienta ('Max. čas přidělení'). Viz obrázek 27.3 na následující straně).

#### Ukončení konfigurace a nastavení režimu spouštění

V posledním dialogu konfiguračního průvodce zvolte, jak má být DHCP server spouštěn – automaticky při startu operačního systému nebo manuálně v případě potřeby (např. pro testovací účely). Klikněte na 'Konec', konfigurace DHCP serveru se tak dokončí. Viz obrázek 27.4 na straně 445.

## 27.3 DHCP softwarové balíčky

Pro systém SUSE LINUX je k dispozici jak DHCP server, tak i klientský DHCP software. V systému SUSE LINUX je DHCP server `dhcpd` od konzorcia ISC (Internet Software Consortium). Na straně klienta lze použít program `dhclient` (rovněž od ISC) nebo klientského démona z balíčku `dhcpd`.

SUSE LINUX standardně používá `dhcpd`, který je velmi snadno nastavitelný, spouští se automaticky při startu systému a okamžitě hledá DHCP server. Ke své

**Rozsah IP adres**  
Zde nastavíte nejvyšší IP adresu a nejnižší IP adresu z rozsahu přidělovaného klientům. Tyto adresy musí mít stejnou masku. Například 192.168.1.1 a 192.168.1.64

**Přidělení**  
Zde můžete nastavit výchozí čas přidělení aktuálního rozsahu IP adres, kterým nastavíte optimální obnovování IP klientů.

**Max. čas přidělení** (volitelné)  
nastavuje maximální dobu, pro kterou je IP na DHCP serveru blokováno pro klienta.

**Průvodce nastavením DHCP serveru (3/3) : DHCP server**

Rozsah IP adres

Nejvyšší IP adresa:

Nejnižší IP adresa:

Přidělení

Čas přidělení

4 h

Max. čas přidělení

2 Dni

Zpět Přerušit Další

*Obrázek 27.3: DHCP server: Dynamické DHCP*

práci nepotřebuje žádný konfigurační soubor a ve většině případů pracuje bez nutnosti jakéhokoliv zásahu. Pro složitější případy použijte ISC dhclient, který se nastavuje pomocí konfiguračního souboru `/etc/dhclient.conf`.

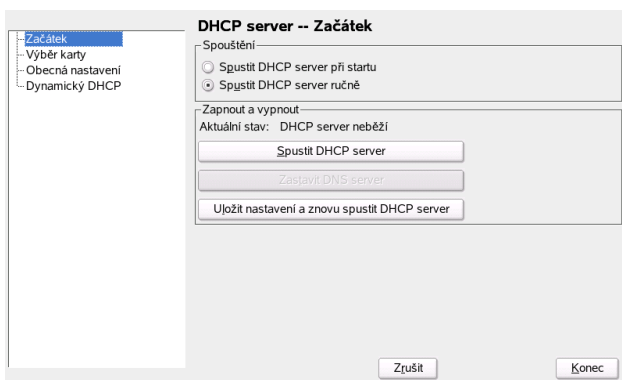
## 27.4 DHCP server dhcpd

Srdcem každého DHCP systému je démon *Dynamic Host Configuration Protocol Daemon* (dhcpd). Pronajímá adresy a kontroluje jejich používání tak, jak je nastaveno v konfiguračním souboru `/etc/dhcpd.conf`. Změnou parametrů a hodnot uvedených v tomto souboru lze ovlivnit chování programu. Podívejte se na jednoduchý příklad konfiguračního souboru `/etc/dhcpd.conf` v 27.1 na této straně:

*Příklad 27.1: Konfigurační soubor `/etc/dhcpd.conf`*

```
default-lease-time 600;          # 10 minutes
max-lease-time 7200;            # 2 hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
```



*Obrázek 27.4: DHCP server: Spouštění systému*

```
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.10 192.168.1.20;
    range 192.168.1.100 192.168.1.200;
}
```

Tento jednoduchý konfigurační soubor stačí k tomu, abyste prostřednictvím DHCP mohli přidělovat v síti IP adresy. Nezapomeňte na středníky na konci každé řádky, bez kterých není možné `dhcpd` spustit!

Jak je vidět z výše uvedeného příkladu, soubor je rozdělen do tří bloků. V první části je uvedeno, na kolik vteřin bude IP adresa standardně počítači přidělena (`default-lease-time`), nezažádá-li o jiný časový úsek. Po uplynutí této doby musí počítač zažádat o prodloužení. Druhá položka určuje maximální dobu, o kterou si počítač může zažádat (`max-lease-time`).

V druhé části jsou nastaveny některé obecné síťové parametry:

- Volbou `option domain-name` je definována výchozí doména sítě.
- `option domain-name-servers` může obsahovat až tři DNS servery, které slouží pro převod IP adres na názvy počítačů (a obráceně). V ideálním případě máte již v systému nebo v síti provozuschopný jmenný server (nameserver). Ten by měl pro každou dynamickou adresu definovat jméno počítače a naopak. Více informací o konfiguraci nameserverů viz 24 na straně 409.
- `option broadcast-address` určuje, jakou oznamovací (*broadcast*) adresu má použít dotazující se počítač.
- `option routers` určuje, kam mají být zasílány pakety, které nejsou určeny počítači v lokální síti (podle zdrojové a cílové adresy a masky podsítě). U malých sítí je tento směrovač obvykle bránou k Internetu.
- `option subnet-mask` určuje síťovou masku pro klienty.

Poslední část souboru definuje síť, včetně masek podsítě. Nakonec je zde uveden rozsah adres, které bude DHCP démon přiřazovat klientům. V našem příkladu může být klientům přiřazena libovolná adresa mezi 192.168.1.10 a 192.168.1.20 nebo mezi 192.168.1.100 a 192.168.1.200.

Pokud jste provedli tato nastavení, měli byste být schopni spustit DHCP démona příkazem `rcdhcpdstart`. Démon tak bude okamžitě připraven k provozu. Pro kontrolu syntaxe konfiguračního souboru můžete použít příkaz `rcdhcpdcheck-syntax`. Pokud nastanou problémy a server skončí s chybou nebo nevrátí po startu `done`, podívejte se na systémová hlášení do protokolového souboru `/var/log/messages`, případně na desátou konzoli (~~C~~trí-~~A~~lt-~~F~~10).

Ve výchozím nastavení systému SUSE LINUX se DHCP démon z bezpečnostních důvodů spouští ve chroot prostředí. Aby démon našel konfigurační soubory, musí být do chroot prostředí zkopírovány. Obvykle si s tím nemusíte lámat hlavu, protože příkaz `rcdhcpdstart` soubory automaticky zkopíruje.

## 27.4.1 Počítač s pevnou IP adresou

Jak jsme zmínili výše, DHCP lze nastavit tak, aby určitý počítač dostal při každém požadavku přednastavenou statickou adresu. Explicitně určené adresy mají

přednost před dynamickými adresami vybíranými z přiděleného rozsahu. Navíc statická adresa nikdy nevypřší, jak se to může stát s adresou dynamickou, například v případě, kdy je nedostatek adres a server je potřeby mezi počítači přerozdělit.

K identifikaci počítače, který má mít přidělovánu *statickou* adresu, používá dhcpd celosvětově unikátní hardwarovou adresu (MAC). Hardwarová adresa sestává z šesti párů šestnáctkových číslic (např. 00:00:45:12:EE:F4). Pokud jsou do konfiguračního souboru 27.1 na straně 444 přidány řádky podobné těm z příkladu 27.2 na této straně, bude danému počítači vždy přidělováno stejné nastavení.

### *Příklad 27.2: Additions to the Configuration File*

```
host zeme {  
  hardware ethernet 00:00:45:12:EE:F4;  
  fixed-address 192.168.1.21;  
}
```

Jméno počítače (host *<jmenopocitace>*), v našem příkladu *zeme*) se vkládá na první řádek. Hardwarová (MAC) adresa se zapisuje na řádek druhý. Na linuxových strojích lze MAC adresu zjistit příkazem (v případě síťového zařízení *eth0*) *ifstatuseth0*. Pokud není karta aktivní, aktivujte ji příkazem *ifupeth0*. Výstup příkazu *ifstatus* by měl obsahovat řádek podobný následujícímu:

```
link/ether 00:00:45:12:EE:F4
```

Při nastavení uvedeném v příkladu výše bude počítači se síťovou kartou s MAC adresou 00:00:45:12:EE:F4 automaticky přiřazena IP adresa 192.168.1.21 a jméno *zeme*. Na řádce s MAC adresou je zapsán i typ hardwaru, většinou *ethernet*. Je ale podporován i *token-ring* často se vyskytující v systémech IBM.

## 27.4.2 Zvláštnosti v systému SUSE LINUX

Pro zvýšení bezpečnosti je SUSE verze ISC DHCP serveru opatřena *non-root/chroot* záplatou Ari Edelkinda. Server tak může běžet s uživatelským ID *nobody* ve *chroot* prostředí (*/var/lib/dhcp*). Aby to bylo skutečně možné, musí se konfigurační soubor *dhcpd.conf* nacházet v adresáři */var/lib/dhcp/etc*. Startovací skript ho tam automaticky zkopíruje.

Tuto vlastnost lze nastavit v souboru `/etc/sysconfig/dhcpd`. Chcete-li spouštět dhcpd bez prostředí chroot, nastavte v něm proměnnou `DHCPD_RUN_CHROOTED` na `no`.

Chcete-li aby dhcpd překládal jména počítačů i z prostředí chroot, musí se zkopírovat i některé další soubory:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

Tyto soubory jsou startovacím skriptem kopírovány do adresáře `/var/lib/dhcp/etc/`. Kopie je nutno brát v úvahu při dynamické modifikaci souborů skripty jako např. `/etc/ppp/ip-up`. Pokud však konfigurační soubor specifikuje pouze IP adresy (a nikoliv jména počítačů), nemusíte se tím zabývat.

Pokud ve vaší konfiguraci potřebujete do chroot prostředí kopírovat další soubory, nastavte je v proměnné `DHCPD_CONF_INCLUDE_FILES` v souboru `etc/sysconfig/dhcpd`. Aby mohl DHCP server v prostředí chroot zaznamenávat údaje do protokolových souborů i po restartu syslog démona, musíte do proměnné `SYSLOGD_PARAMS` v souboru `/etc/sysconfig/syslog` vložit volbu `"-a /var/lib/dhcp/dev/log"`.

## 27.5 Další informace

Více informací o DHCP najdete na stránkách *Internet Software Consortium* (<http://www.isc.org/products/DHCP/>). Řada důležitých informací je také v manuálových stránkách `dhcpd`, `dhcpd.conf`, `dhcpd.leases` a `dhcp-options`.



# Synchronizace času pomocí xntp

NTP (Network Time Protocol) je protokol pro synchronizaci systémového času po síti. Počítače mohou s jeho pomocí získávat informaci o času z přesných časových serverů. Takto seřízený počítač pak může poskytovat informaci o přesném čase dalším počítačům v síti. Cíle jsou dva – zajistit přesnou informaci o absolutním čase a synchronizovat čas všech strojů v síti.

28.1	Nastavení xntp v síti . . . . .	450
28.2	Nastavení lokálních referenčních hodin . . . . .	451
28.3	Nastavení NTP klienta v programu YaST . . . . .	451

Nastavení správného a jednotného času v síti je důležité v řadě situací. Počítače samozřejmě obsahují vlastní hardwarové hodiny. Jejich čas se však může u různých počítačů lišit. Takové časové rozdíly pak mohou způsobit řadu problémů např. při práci s databázemi. Také v síti je obvykle potřeba mít čas na jednotlivých strojích synchronizovaný. Lze ho nastavit ručně, ale to není dobrý přístup. Síťové řešení tohoto problému nabízí program `xntp`. Neustále upravuje systémový čas pomocí údajů ze spolehlivých časových serverů v síti. Navíc umožňuje spravovat lokální referenční hodiny, např. rádiem řízené.

## 28.1 Nastavení `xntp` v síti

Výchozí nastavení `xntp` respektuje jako referenční čas lokální hodiny počítače. Použití těchto (BIOS) hodin je však pouze náhradní řešení pro případ, kdy není dostupný spolehlivější zdroj. Nejjednodušší způsob, jak přistupovat k časovému serveru, je zadat server do položky `server` v konfiguračním souboru `/etc/ntp.conf`. Např. pokud má být čas synchronizován podle serveru `ntp.example.com`, do souboru `/etc/ntp.conf` vložte řádek `server ntp.example.com`.

Chcete-li používat serverů více, vložte pro každý z nich samostatný řádek začínající klíčovým slovem `server`. Po spuštění `xntpd` příkazem `rcxntpd start` trvá asi hodinu, než se čas stabilizuje a vytvoří se *drift soubor* korigující lokální hardwarové hodiny. Pomocí *drift souboru* lze spočítat a opravit systematickou chybu hardwarových hodin okamžitě po spuštění počítače. Tím je zajištěna vysoká stabilita systémového času.

Jsou dva možné způsoby využití NTP na klientovi. Prvním je dotazování se na přesný čas na časovém serveru v pravidelných intervalech. Pokud je ale klientů hodně, může to pro server znamenat velkou zátěž. Druhou možností je čekat na vysílání časových údajů serveru v síti. Nevýhodou je, že kvalita vysílajícího serveru není známá a server vysílající nesprávné časové údaje může způsobit vážné problémy.

Pokud je čas vysílán po síti, nepotřebujete znát jméno serveru. Stačí do souboru `/etc/ntp.conf` vložit řádek `broadcastclient`. Chcete-li používat pouze jeden nebo několik známých serverů, vložte jejich jména do řádky začínající slovem `servers`.

## 28.2 Nastavení lokálních referenčních hodin

Program xntp obsahuje také ovladač pro připojení lokálních referenčních hodin. Seznam podporovaných hodin najdete po nainstalování balíčku xntp-doc v souboru `file:/usr/share/doc/packages/xntp-doc/html/refclock.htm`. Každý ovladač je označen vlastním číslem. Konfigurace xntp se pak provádí pomocí pseudo IP adres. Údaje o hodinách se vloží do souboru `/etc/ntp.conf`, jako by šlo o standardní síťový časový server. Jsou jim přiřazeny speciální IP adresy ve formátu `127.127.t.u`. Hodnota `t` označuje typ hodin a určuje výběr použitého ovladače, zatímco `u` (unit) specifikuje použité rozhraní.

Jednotlivé ovladače mají specifické konfigurační parametry. Podrobnosti o jednotlivých typech hodin naleznete v souboru `/usr/share/doc/packages/xntp-doc/html/driverNN.htm` (kde NN je číslo ovladače). Například hodiny typu 8 (radiové hodiny připojené přes sériové rozhraní) vyžadují přídatný modul. Modul Conrad DCF77 má např. režim 5. Aby byly tyto hodiny používány jako primární referenční zdroj, je nutné použít klíčové slovo `prefer`. Kompletní položka pro nastavení modulu Conrad DCF77 v konfiguračním souboru se proto napíše takto:

```
server 127.127.8.0 mode 5 prefer
```

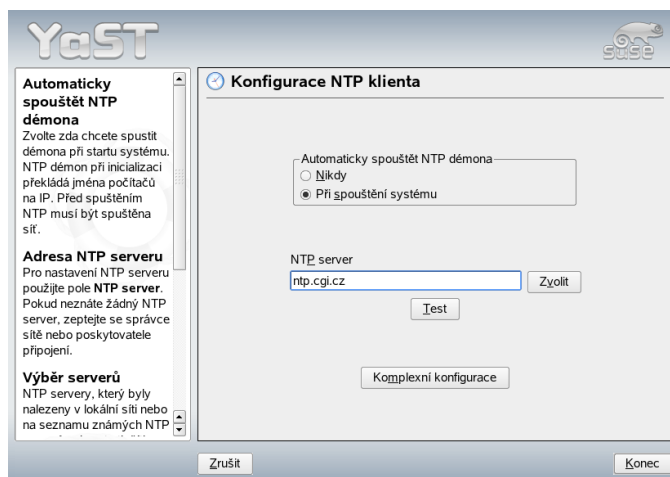
Ostatní hodiny se nastavují podobně. Příklady najdete v dokumentaci xntp v `/usr/share/doc/packages/xntp-doc/html`.

## 28.3 Nastavení NTP klienta v programu YaST

Nastavení NTP klienta můžete v systému SUSE LINUX provést pomocí nástroje YaST. Na výběr máte z rychlé nebo komplexní konfigurace.

### 28.3.1 Rychlé nastavení NTP klienta

Rychlé nastavení NTP klienta se skládá ze dvou kroků. V prvním je nutné nastavit spouštění xntpd, ve druhém zadat NTP server. Chcete-li, aby se xntpd spouštěl automaticky při startu systému, vyberte 'Při spouštění systému'. Pak klikněte na 'Zvolit'. Tím se otevře druhý dialog, ve kterém vyberte vhodný server.



*Obrázek 28.1: YaST: Konfigurace NTP klienta*

Na výběr máte ‘Lokální síť’ nebo ‘Veřejný NTP server’. Zvolte nejvhodnější server a otestujte nastavení tlačítkem ‘Test’. Pokud test dopadl dobře, potvrďte výběr tlačítkem ‘OK’.

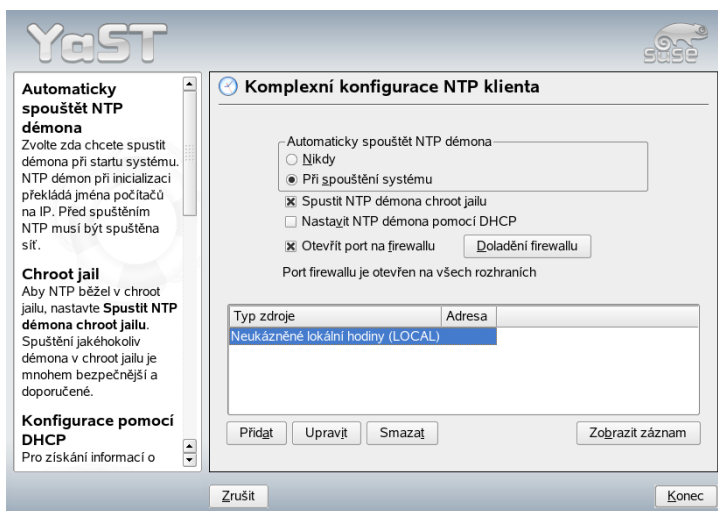
### 28.3.2 Komplexní nastavení NTP klienta

Komplexní nastavení NTP klienta je dostupné v hlavním dialogu ‘NTP klient’ po nastavení spouštění kliknutím na tlačítko ‘Komplexní konfigurace’ (viz 28.1 na této straně).

V ‘Komplexní konfiguraci NTP klienta’ lze nastavit, zda se má xntpd spouštět v chroot jail. Tímto nastavením výrazně zvýšíte bezpečnost systému, protože v případě napadení xntpd nebude mít útočník k dispozici přístup do systému. Volba ‘Nastavit NTP démona pomocí DHCP’ zajistí získání NTP serverů pro NTP klienta přes DHCP.

Jednotlivé časové servery a další časové zdroje najdete v tabulce pod volbami. Můžete je ‘Přidat’, ‘Upravit’ nebo ‘Smazat’.

Nový zdroj časových informací zadáte kliknutím na ‘Přidat’. Vyberte požadovaný typ zdroje a klikněte na tlačítko ‘Další’. Vybrat si můžete z následujících typů zdrojů:



Obrázek 28.2: YaST: Komplexní konfigurace NTP klienta

**Server** Zvolíte-li tuto volbu, zadejte v následujícím dialogu NTP server (viz 28.3.1 na straně 451). Aktivujte ‘Použít pro počáteční synchronizaci’, pokud chcete provádět synchronizaci s tímto serverem při startu systému. V dalším poli můžete zadat dodatečné volby. Více informací najdete v adresáři `/usr/share/doc/packages/xntp-doc`.

**Rovnocenný** Zde můžete místo serveru zvolit jinou klientskou stanici, se kterou bude navázán symetrický vztah. Další dialog je podobný jako v případě volby ‘Server’.

**Radio hodiny** U radio hodin musíte v následujícím dialogu zadat typ hodin, číslo jednotky, jméno zařízení a další volby. Doladění provedete kliknutím na ‘Kalibrace ovladače’. Další informace najdete v souboru `/usr/share/doc/packages/xntp-doc/html/refclock.htm`.

**Vysílání** Časové informace lze vysílat po síti. Pokud tak chcete činit, je v tomto dialogu nutné zadat adresu, na kterou mají být časové údaje vysílány. Nepoužívejte vysílání, pokud nemáte spolehlivý časový zdroj, např. rádiem řízené hodiny.

**Přijímání vysílaných paketů** Jestliže má klient přijímat vysílané pakety, zadejte v tomto poli adresu, ze které mají být přijímány pakety.

# LDAP adresářové služby

LDAP (Lightweight Directory Access Protocol) je sada protokolů určených ke správě a přístupu k informačním adresářům. LDAP lze využít k mnoha účelům, jako je správa uživatelů a skupin, správa systémové konfigurace nebo správa adres. V této kapitole jsou popsány základy funkce LDAP a jeho konfigurace pomocí nástroje YaST.

29.1	LDAP versus NIS . . . . .	457
29.2	Struktura adresářového stromu LDAP . . . . .	458
29.3	Konfigurace LDAP serveru pomocí slapd.conf . . . . .	460
29.4	Správa dat v LDAP adresáři . . . . .	465
29.5	YaST LDAP klient . . . . .	468
29.6	Další informace . . . . .	474

V síťovém prostředí je velmi důležité uchovávat důležité informace na dostupném místě a v uspořádané podobě. To lze zajistit adresářovou službou, která, podobně jako zlaté stránky, poskytuje informace ve strukturované a přehledné formě s možností snadného vyhledávání.

V ideálním případě server všechna data uloží do adresáře a pomocí jednotného protokolu je pak distribuuje všem klientům. Data jsou strukturována tak, aby s nimi mohla pracovat celá řada různých aplikací. Není tak nutné, aby každá kalendářová aplikace či poštovní klient udržoval nezávislou databázi, stačí vytvořit jednu centrální. Tím se uspoří čas a náklady na údržbu několika databází. Použitím otevřeného a standardizovaného protokolu LDAP navíc zajistíte, že tato data budou dostupná pro různé typy aplikací a klientů.

Pojmem adresář v této souvislosti rozumíme databázi optimalizovanou pro rychlé a efektivní čtení a vyhledávání, která má tyto vlastnosti:

- Aby bylo umožněno vícenásobné čtení v maximálním objemu, je zápis omezen na aktualizace administrátorem databáze. Běžné typy databází jsou optimalizovány pro zápis maximálního množství dat v krátkém čase.
- Protože jsou možnosti zápisu značně omezeny, slouží adresářové služby především pro uchovávání neměnných *statických informací*. V normální databázi se naopak data mění velmi často (dynamická data). Např. telefonní číslo společnosti se nemění tak často jako účetní údaje.
- Administrace statických dat vyžaduje jen výjimečné aktualizace a změny. Při práci s dynamickými daty, jako např. zůstatky na účtech, je kladen vysoký důraz na konzistenci dat. Pokud je například z jednoho účtu odečtena částka a připsána na jiný, musí obě operace proběhnout současně v rámci jedné transakce. Databáze takové transakce podporují, ale adresářové služby nikoliv. Krátkodobé nekonzistence nevedou u adresářové služby k žádným závažným problémům.

Adresářové služby jako LDAP nejsou navrženy pro podporu komplexní aktualizace a dotazovacího mechanismu. Přístup musí být rychlý a jednoduchý.

Řada adresářových služeb existovala a dosud existuje jak na platformě Unix, tak mimo ní. Několika příklady jsou Novell NDS, Microsoft ADS, Banyan Street Talk a OSI standard X.500. LDAP byl původně navržen jako verze DAP (Directory Access Protocol) navrženého pro přístup k X.500. Standard X.500 se zabývá hierarchickou organizací adresářové struktury.



LDAP je zjednodušená verze DAP, která neobsahuje některé funkce DAP, což umožňuje úspory zdrojů. Použití protokolu TCP/IP usnadňuje spojení aplikací se službou LDAP.

LDAP je dnes samostatným řešením pracujícím bez podpory X.500. LDAPv3 (verze protokolu v balíčku `openldap2`) podporuje tzv. *referrals*, které umožňují vytváření distribuovaných databází. Nové je také využití SASL (Simple Authentication and Security Layer).

LDAP není omezen na X.500 servery, jak bylo původně v plánu. Opensource server `slapd` dokáže ukládat objektové informace v lokální databázi. Díky rozšíření `slurpd` je možné LDAP servery replikovat.

Balíček `openldap2` obsahuje následující programy:

**slapd** LDAPv3 server spravující informace v databázi typu BerkeleyDB.

**slurpd** Program pro replikaci změn dat z lokálního serveru na ostatní LDAP servery v síti.

**Další nástroje pro správu** `slapcat`, `slapadd`, `slapindex`.

## 29.1 LDAP versus NIS

Unixoví administrátoři pro převod jmen a distribuci dat v síti tradičně používají službu NIS. Konfigurační data se nacházejí v souborech v adresáři `/etc`: `group`, `hosts`, `mail`, `netgroup`, `networks`, `passwd`, `printcap`, `protocols`, `rpc` a `services`, odkud jsou distribuována klientům v síti. Tyto soubory lze velmi jednoduše spravovat, protože jde o prosté textové soubory. Správa většího množství dat je ovšem náročnější vzhledem k neexistující strukturalizaci. Služba NIS je určena pouze pro unixové systémy, což znesnadňuje nasazení v heterogenních sítích.

Na rozdíl od NIS není služba LDAP omezená jen na čistě unixové sítě. LDAP podporují Windows servery (od verze 2000) a podporu nabízí také Novell.

LDAP je vhodný všude, kde je zapotřebí centrálně spravovat datovou strukturu, např.:

- Náhrada NIS.
- Směrování pošty (`postfix`, `sendmail`).

- Adresář pro poštovní klienty jako je Mozilla, Evolution či Outlook.
- Administrace popisů zón BIND9 name serveru.

Tento seznam by mohl být mnohem delší, protože LDAP je na rozdíl od NIS rozšiřitelný. Jasně definovaná hierarchická struktura dat usnadňuje administraci velkého množství dat.

## 29.2 Struktura adresářového stromu LDAP

LDAP adresář má stromovou strukturu. Všechny záznamy (zvané objekty) adresáře mají v hierarchii jasně definovanou pozici. Tato struktura je označována jako *informační adresářový strom* (DIT, directory information tree). Kompletní cesta k určité položce se nazývá *jedinečné jméno* nebo-li DN (distinguished name). Jednotlivé nody této cesty se nazývají *relativní jedinečné jméno* nebo-li RDN (relative distinguished name). Objekty mohou být dvou typů:

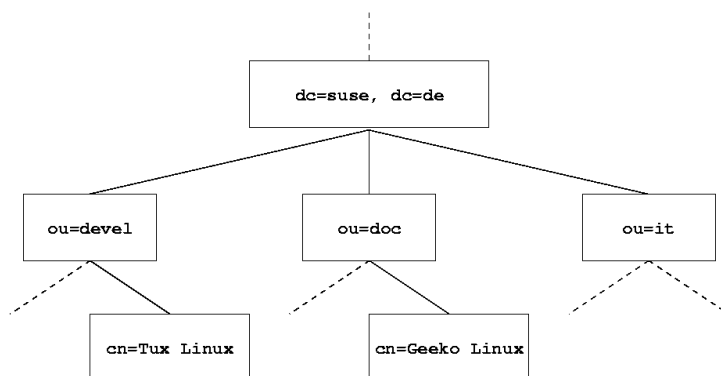
**kontejner** Tyto objekty mohou obsahovat další objekty. Mezi tyto objekty patří `root` (kořenový element adresářového stromu), `c` (country, země), `ou` (organizational unit, organizační jednotka) a `dc` (domain component, doménová komponenta).

**list** Tyto objekty se nalézají na samém okraji větve a nemají žádné podobjekty. Jde např. o `person`, `InetOrgPerson` nebo `groupofNames`.

Na samém vrcholu adresářové struktury stojí objekt `root`. Ten obsahuje podobjekty `c` (country), `dc` (domain component) nebo `o` (organization). Vztahy mezi objekty v LDAP stromu jsou zřejmé z obrázku 29.1 na následující straně.

Diagram obsahuje fiktivní informační adresářový strom. Každý obdélník na obrázku představuje jeden záznam. Úplné validní *jedinečné jméno* (DN) smyšleného SUSE zaměstnance jménem `Geeko Linux` je v našem případě `cn=Geeko Linux, ou=doc, dc=suse, dc=de`. Je vytvořeno přidáním RDN `cn=Geeko Linux` k DN předcházejícího záznamu `ou=doc, dc=suse, dc=de`.

Obecná pravidla určující, jaké typy objektů mají být ukládány v DIT, jsou daná tzv. schématem (*scheme*). Typ objektu je určen *objektovou třídou*. Objektová třída určuje vlastnosti, které objekt *musí* nebo *může* mít. Schéma proto musí obsahovat definici všech objektových tříd a atributů. K dispozici je několik obecných schémat (viz RFC 2252 a 2256). Samozřejmě je možné vytvořit si schéma vlastní, které bude více vyhovovat vašim požadavkům.



**Obrázek 29.1:** Struktura LDAP adresáře

Tabulka 29.1 na této straně nabízí krátký přehled tříd objektů ze schémat `core.schema` a `inetorgperson.schema` použitých v příkladu. Najdete zde také atributy a platné hodnoty těchto atributů.

**Tabulka 29.1:** Běžně používané objektové třídy a atributy

Objektová třída	Význam	Příklad záznamu	Povinné atributy
dcObject	<i>domainComponent</i> (komponenta domény)	suse	dc
organizationalUnit	<i>organizationalUnit</i> (organizační jednotka)	doc	ou
inetOrgPerson	<i>inetOrgPerson</i> (osobní data pro intranet nebo internet)	Geeko Linux	sn a cn

Příklad 29.1 na této straně ukazuje výtah ze schématu s vysvětlením:

**Příklad 29.1:** Výtah ze `schema.core` (řádky jsou dodatečně očíslovány)

```

#1 attributetype (2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
#2     DESC 'RFC2256: organizational unit this object belongs to'
#3     SUP name )

```

Typ atributu `organizationalUnitName` a odpovídající objektová třída `organizationalUnit` zde slouží jako příklad. Řádka 1 obsahuje jméno atributu a jeho unikátní identifikátor OID (*object identifier*) (číselný údaj) a zkratku atributu.

Řádka 2 obsahuje krátký popis atributu (`DESC`). Je zde uveden i odkaz na příslušný RFC. `SUP` v řádce 3 uvádí nadřazený typ atributu, ke kterému tento atribut náleží.

Samotná definice objektové třídy `organizationalUnit` začíná na řádce 4. Stejně jako definice atributu obsahuje OID a jméno třídy. Na řádce 5 je krátký popis objektové třídy. Řádka 6 (`SUP top`) udává, že tato objektová třída není závislá na jiné objektové třídě. Řádka 7 začínající řetězcem `MUST` udává všechny atributy, které objekt typu `organizationalUnit` *musí* obsahovat. Řádka 8 začínající řetězcem `MAY` udává typy atributů, které *mohou* být s touto objektovou třídou používány.

Velmi hezký úvod do schémat najdete v dokumentaci OpenLDAP. Pokud je nainstalován, najdete ho v souboru `/usr/share/doc/packages/openldap2/admin-guide/index.html`.

## 29.3 Konfigurace LDAP serveru pomocí `slapd.conf`

Konfigurace LDAP serveru se nachází v souboru `/etc/openldap/slapd.conf`. Zde jsou popsány jednotlivé položky konfigurace. Položky začínající znakem `#` jsou zakomentované a tedy neaktivní. Pokud je chcete aktivovat, musíte znak smazat.

### 29.3.1 Globální nastavení v `slapd.conf`

*Příklad 29.2: `slapd.conf`: Include příkaz pro schéma*

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/inetorgperson.schema
```

První příkazy `slapd.conf` zobrazené v příkladu 29.2 na předchozí straně určují schéma LDAP adresáře. K základnímu povinnému schématu (zde `core.schema`) lze přidávat i dodatečná schémata (v našem případě `inetorgperson.schema`). Další schémata naleznete v adresáři `/etc/openldap/schema`. Pro nahrazení služby NIS službou LDAP budete potřebovat dvě schémata – `rfc2307.schema` a `cosine.schema`. Informace o této problematice najdete v dokumentaci OpenLDAP.

**Příklad 29.3:** *slapd.conf: pidfile a argsfile*

```
pidfile /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

Tyto dva soubory obsahují PID (process ID) a některé argumenty, se kterými je spouštěn `slapd`. Žádné změny zde nejsou potřeba.

**Příklad 29.4:** *slapd.conf: Kontrola přístupu*

```
# Sample Access Control
#       Allow read access of root DSE
# Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
# access to dn="" by * read
#       access to * by self write
#               by users read
#               by anonymous auth
#
# if no access controls are present, the default is:
#       Allow read by all
#
# rootdn can always write!
```

Příklad 29.4 na této straně ukazuje část souboru `slapd.conf`, která se týká nastavení přístupu k adresáři LDAP na serveru. Nastavení uvedená zde v globální sekci souboru `slapd.conf` jsou platná až do okamžiku vytvoření nastavení v části specifické pro databázi. Ta mají přednost před globálními nastaveními. V našem příkladě mají všichni uživatelé práva pro čtení, ale pouze administrátor (`rootdn`) může do této databáze zapisovat. Nastavení přístupových práv v LDAP je poměrně složité téma, nabízíme proto několik tipů:

- Každé pravidlo pro přístup má následující strukturu:

access to <what> by <who> <access>

- *<what>* nahradíte objektem nebo atributem, ke kterému se má přistupovat. Jednotlivé větve adresáře mohou být chráněny vlastními pravidly. Pokud chcete, můžete chránit části adresáře pomocí regulárních výrazů. Program `slapd` vyhodnocuje všechna pravidla v pořadí, v jakém jsou uvedena v konfiguračním souboru. Obecnější pravidla by měla být uvedena později – uplatněno je první platné pravidlo, ostatní jsou ignorována.
- *<who>* určuje, komu bude přiznán přístup do oblastí určených pomocí *<what>*. Lze použít i regulární výrazy. `slapd` opět ukončí vyhodnocování `who` po nalezení první shody, proto by obecnější pravidla měla být uvedena později. Možná jsou nastavení uvedená v tabulce 29.2 na této straně

*Tabulka 29.2: Uživatelské skupiny a jejich přístupová práva*

Tag	Význam
*	všichni uživatelé bez výjimky
anonymous	neautentizovaní uživatelé
users	autentizovaní uživatelé
self	uživatelé spojení s cílovým objektem
dn.regex=<regex>	všichni uživatelé vyhovující regulárnímu výrazu

- *<access>* určuje typ přístupu. Možná nastavení najdete v tabulce 29.3 na této straně.

*Tabulka 29.3: Typy přístupu*

Tag	Význam
none	bez přístupu
auth	spojení se serverem
compare	porovnávání
search	vyhledávání pomocí filtrů
read	čtení

write                      zápis

---

slapd porovnává požadavky klientů s nastavením přístupových práv v souboru `slapd.conf`. Klientovi je přístup povolen jen v případě, že splňuje požadavky pro přístup (má požadovaná nebo vyšší práva). Pokud klient vyžaduje vyšší práva, než mu jsou přiřazena, je mu odmítnut přístup.

Příklad 29.5 na této straně ukazuje jednoduché nastavení přístupových práv pomocí regulárního výrazu:

**Příklad 29.5:** *slapd.conf: Příklad nastavení přístupových práv*

```
access to dn.regex="ou=([^\,]+),dc=suse,dc=de"
by dn.regex="cn=administrator,ou=$1,dc=suse,dc=de" write
by user read
by * none
```

V tomto příkladu má práva zápisu do záznamu `ou` pouze administrátor. Všichni ostatní autentizovaní uživatelé mají práva ke čtení. Ostatní uživatelé nemají žádný přístup.

## Tip

### Vytvoření přístupových pravidel

Pokud chybí pravidlo `access to` nebo neexistuje vyhovující proměnná `by`, není přístup povolen. Jsou přiznána jen výslovně uvedená přístupová práva. Jestliže nezádáte vůbec žádné pravidlo, nastaví se výchozí přístupová práva, tj. právo zápisu pro administrátora a právo čtení pro všechny ostatní.

## Tip

Podrobné informace a příklady nastavení přístupových práv k LDAP naleznete v dokumentaci balíčku `openldap2`.

Kromě nastavení přístupových práv v centrálním konfiguračním souboru (`slapd.conf`) je k dispozici také ACI (Access Control Information). ACI umožňuje ukládání informací o jednotlivých objektech LDAP stromu. Tento způsob kontroly přístupu je však stále ještě považován za experimentální. Viz <http://www.openldap.org/faq/data/cache/758.html>.

## 29.3.2 Nastavení specifická pro databázi v souboru slapd.conf

*Příklad 29.6: slapd.conf: Nastavení specifická pro databázi*

```
database ldbm
suffix "dc=suse,dc=de"
rootdn "cn=admin,dc=suse,dc=de"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slapd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index objectClass eq
```

Na prvním řádku této sekce (viz 29.6 na této straně) je určen typ databáze (v našem případě LDBM). Na druhé řádce (`suffix`) je určeno, za jakou část LDAP stromu server zodpovídá. Následující `rootdn` určuje administrátora serveru. Zde nastavený uživatel nepotřebuje mít LDAP záznam nebo existovat jako běžný uživatel. Heslo administrátora je nastaveno v položce `rootpw`. Místo `secret` můžete použít hash administrátorského hesla vytvořený pomocí programu `slapasswd`. Položka `directory` určuje adresář (v souborovém systému), ve kterém je uložena databáze. Poslední část, `index objectClass eq`, určuje, že index bude udržován pro všechny objektové třídy. Podle zkušeností zde lze nastavit atributy, které uživatelé nejčastěji vyhledávají. Access pravidla nastavená v této sekci se použijí místo pravidel globálních.

## 29.3.3 Spuštění a zastavení serveru

Je-li server plně nakonfigurovaný a jsou-li vytvořeny všechny požadované záznamy, jak je popsáno v sekci 29.4 na následující straně, spusťte server jako uživatel `root` příkazem `rcldap start`. Ručně server zastavíte příkazem `rcldap stop`. Stav běžícího LDAP serveru zjistíte příkazem `rcldap status`.

Pokud chcete LDAP server spouštět automaticky při startu systému, použijte k nastavení editor úrovní běhu systému nástroje YaST (viz 7.6 na straně 158). Automatické spuštění při startu systému můžete zajistit také pomocí příkazu `insserv` (viz 7.5.1 na straně 156).



## 29.4 Správa dat v LDAP adresáři

OpenLDAP nabízí pro správu dat v LDAP adresáři celou řadu nástrojů. Čtyři nejdůležitější nástroje pro vkládání, mazání, vyhledávání a úpravy dat jsou popsány dále.

### 29.4.1 Vkládání dat do LDAP adresáře

Pokud je LDAP server správně nakonfigurován, tedy pokud jsou v souboru `/etc/openldap/slapd.conf` nastaveny položky `suffix`, `directory`, `rootdn`, `rootpw` a `index`, pokračujte vkládáním záznamů. K tomu OpenLDAP nabízí nástroj `ldapadd`. Objekty je z praktických důvodů vhodné vkládat po větších celcích. Vhodný je například LDIF formát (LDAP Data Interchange Format). LDIF je jednoduchý textový soubor obsahující páry atributhodnota. Dostupné objektové třídy a atributy jsou definované ve schématech uvedených v souboru `slapd.conf`. LDIF soubor k vytvoření hrubé kostry obrázku 29.1 na straně 459 by vypadal asi tak, jak je uvedeno v příkladu 29.7 na této straně:

#### *Příklad 29.7: Příklad LDIF souboru*

```
# The SUSE Organization
dn: dc=suse,dc=de
objectClass: dcObject
objectClass: organization
o: SUSE AG dc: suse

# The organizational unit development (devel)
dn: ou=devel,dc=suse,dc=de
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=suse,dc=de
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=suse,dc=de
objectClass: organizationalUnit
ou: it
```

## Důležité

### Kódování LDIF souborů

LDAP pracuje s UTF-8 (Unicode). Používejte proto editor s podporou UTF-8 (např. Kate nebo novější verze editorů Emacs či Vim). Jestliže použijete editor bez podpory UTF-8, budou se špatně zobrazovat znaky s českou diakritikou. Pokud potřebujete převést do UTF-8 již existující text, použijte program `recode`.

## Důležité

Soubor se ukládá s příponou `.ldif` a serveru se předává příkazem:

```
ldapadd -x -D <dn administrátora> -W -f <soubor>.ldif
```

První parametr, `-x`, vypíná ověřování pomocí SASL. Parametr `-D` specifikuje uživatele, který operaci volá. Za touto volbou musí následovat DN administrátora tak, jak je uvedeno v souboru `slapd.conf`. V našem případě jde o `cn=admin,dc=suse,dc=de`. Přepínač `-W` obojde zadávání hesla přímo na příkazovém řádku (v prostém textu) a zobrazí zvláštní výzvu k zadání hesla. Jde o heslo ze souboru `slapd.conf` (`rootpw`). Parametrem `-f` předáte jméno souboru. Ukázkou běhu programu `ldapadd` si můžete prohlédnout v příkladu 29.8 na této straně.

### *Příklad 29.8: Použití `ldapadd` s `example.ldif`*

```
ldapadd -x -D cn=admin,dc=suse,dc=de -W -f example.ldif
```

```
Enter LDAP password:
adding new entry "dc=suse,dc=de"
adding new entry "ou=devel,dc=suse,dc=de"
adding new entry "ou=doc,dc=suse,dc=de"
adding new entry "ou=it,dc=suse,dc=de"
```

Data jednotlivých uživatelů lze připravit v oddělených LDIF souborech. Příklad 29.9 na této straně přidává do LDAP adresáře uživatele Tux:

### *Příklad 29.9: LDIF data uživatele Tux*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@suse.de
uid: tux
telephoneNumber: +49 1234 567-8
```

LDIF soubor může obsahovat libovolné množství objektů. Jednotlivé větve stromu je tak možné vložit do databáze najednou nebo po částech. Pokud se některé části mění častěji, je vhodné je oddělit zvlášť.

## 29.4.2 Úprava dat v LDAP adresáři

K úpravě dat se používá příkaz `ldapmodify`. Nejjednodušší způsob je změnit příslušný LDIF soubor a ten pak předat serveru. Pokud byste např. chtěli změnit telefonní číslo kolegy Tuxe z +49 1234 567-8 na +49 1234 567-10, změňte LDIF soubor tak, jak je uvedeno v příkladu 29.10 na této straně:

### *Příklad 29.10: Upravený LDIF soubor `tux.ldif`*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Upravený soubor importujete do adresáře na serveru příkazem:

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W -f tux.ldif
```

Vlastnosti lze měnit i přímo následujícím postupem:

- Spusťte příkaz `ldapmodify` a zadejte heslo:

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W
Enter LDAP password:
```

- Při zadání změn je nutné dodržovat syntaxi. Příkazy pro náš případ vypadají takto:

```
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Více informací o `ldapmodify` a příslušné syntaxi najdete v jeho manuálové stránce (`ldapmodify(1)`).

### 29.4.3 Vyhledávání a čtení dat z LDAP adresáře

OpenLDAP poskytuje nástroj `ldapsearch` pro vyhledávání a čtení dat z LDAP adresáře. Jednoduchý dotaz má následující syntaxi:

```
ldapsearch -x -b dc=suse,dc=de "(objectClass=*)"
```

Parametrem `-b` nastavíte vyhledávací základnu (*search base*), tj. sekci stromu, která má být prohledána, v našem případě `dc=suse, dc=de`.

Volba `-x` zapíná jednoduchou autentizaci. `(objectClass=*)` určuje, že budou čteny všechny objekty v adresáři. Tento příkaz je vhodný např. k ověření správnosti záznamů po vytvoření nového adresářového stromu. Více informací najdete v manuálové stránce `ldapsearch(1)`.

### 29.4.4 Mazání dat z LDAP adresáře

Nechtěné záznamy smažete pomocí příkazu `ldapdelete`. Syntaxe je podobná jako u příkazů uvedených výše. Např. celý záznam `Tux Linux` smažete příkazem:

```
ldapdelete -x -D cn=admin,dc=suse,dc=de -W cn=Tux \
Linux,ou=devel,dc=suse,dc=de
```

## 29.5 YaST LDAP klient

YaST obsahuje modul pro nastavení ověřování uživatelů pomocí LDAP. Pokud jste tuto vlastnost nepovolili během instalace systému, spusťte modul volbou 'Síťové služby' → 'Klient LDAP'. YaST automaticky povolí změny PAM a NSS vyžadované LDAP (jak je popsáno dále) a nainstaluje potřebné soubory.

## 29.5.1 Standardní procedura

Pro pochopení funkce YaST Klient LDAP modulu je nutné znát procedury probíhající na klientském počítači. Při aktivaci LDAP pro ověřování v síti nebo po spuštění YaST Klient LDAP modulu se nainstalují balíčky `pam_ldap` a `nss_ldap` a nastaví se dva související konfigurační soubory. `pam_ldap` je PAM modul odpovědný za přenos dat mezi přihlašovacím procesem a LDAP sloužícím jako zdroj autentizačních dat. Nainstaluje se modul `pam_ldap.so` a přizpůsobí se PAM konfigurace (viz 29.11 na této straně).

### *Příklad 29.11: pam\_unix2.conf přizpůsobený pro LDAP*

```
auth:      use_ldap nullok
account:   use_ldap
password:  use_ldap nullok
session:   none
```

Pokud nastavujete ručně další služby, aby používaly LDAP, vložte PAM LDAP modul do PAM konfiguračního souboru odpovídajícího dané službě v adresáři `/etc/pam.d`. Konfigurační soubory upravené pro jednotlivé služby lze nalézt v adresáři `/usr/share/doc/packages/pam_ldap/pam.d/`. Zkopírujte potřebné soubory do adresáře `/etc/pam.d`.

`glibc` rozpoznávání jmen mechanismem `nsswitch` se nasazení LDAP přizpůsobuje pomocí `nss_ldap`. V adresáři `/etc/` je při instalaci tohoto balíčku vytvořen nový přizpůsobený soubor `nsswitch.conf`. Více se o práci s `nsswitch.conf` dozvíte v části 22.5.1 na straně 395. V souboru `nsswitch.conf` musí být řádky uvedené v příkladu 29.12 na této straně.

### *Příklad 29.12: Přizpůsobení v souboru nsswitch.conf*

```
passwd: compat
group: compat

passwd_compat: ldap
group_compat: ldap
```

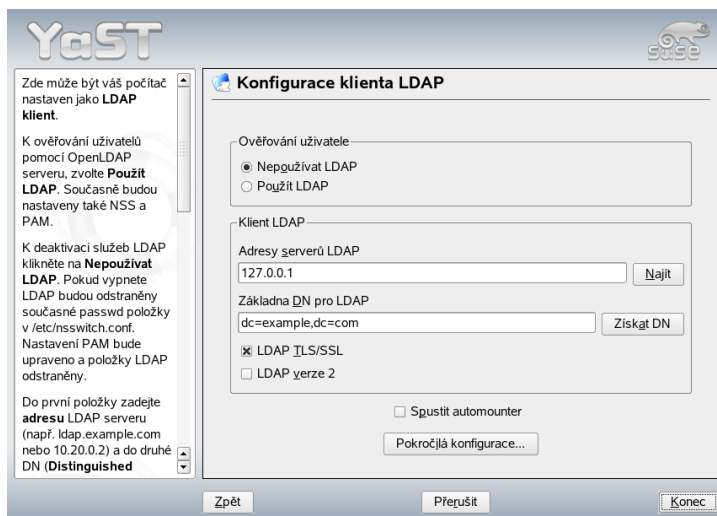
Tyto řádky přikazují resolver knihovně `glibc` nejprve vyhodnotit soubory v adresáři `/etc` a pak se připojit k LDAP serveru jako zdroji autentizačních a uživatelských dat. Mechanismus můžete otestovat přečtením uživatelské databáze

příkazem `getent passwd`. Výsledek by měl obsahovat lokální uživatele vašeho systému i uživatele uložené na LDAP serveru.

Abyste zabránili běžným uživatelům spravovaným přes LDAP přihlásit se k serveru pomocí `ssh` nebo `login`, musí soubory `/etc/passwd` a `/etc/group` obsahovat následující řádek: `+: :: : /sbin/nologin` v `/etc/passwd` a `+: :: : v` v `/etc/group`.

## 29.5.2 Konfigurace LDAP klienta

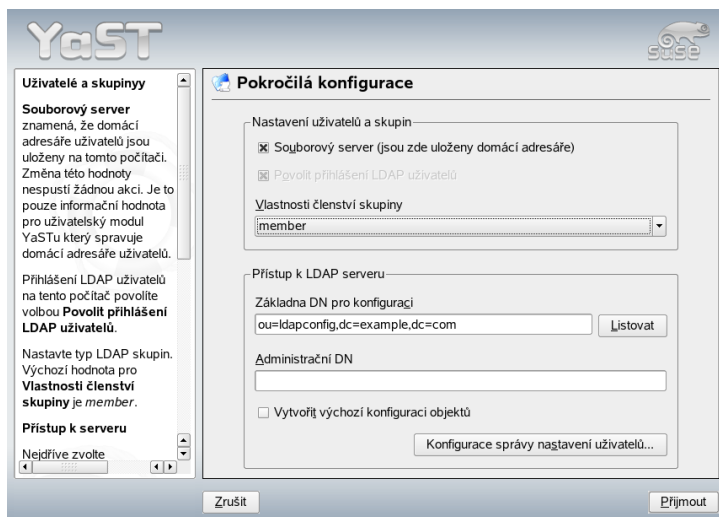
Jakmile jsou `nss_ldap`, `pam_ldap`, `/etc/passwd` a `/etc/group` YaSTem upraveny, lze pokračovat v konfiguraci za pomoci prvního dialogu modulu YaST. Viz obrázek 29.2 na této straně.



*Obrázek 29.2: YaST: Konfigurace LDAP klienta*

V prvním dialogu aktivujte použití LDAP pro autentizaci uživatelů. V položce 'Základna DN pro LDAP' zadejte prohledávací základnu, ve které jsou na serveru uložena data. IP adresu LDAP serveru zadejte v položce 'Adresy serverů LDAP'. Můžete zadat více serverů oddělených mezerou. Chcete-li automaticky

připojovat adresáře, zaškrtněte ‘Spustit automounter’. Chcete-li jako administrátor upravit data na serveru, klikněte na ‘Pokročilá konfigurace’. Viz obrázek 29.3 na této straně.



Obrázek 29.3: YaST: Pokročilá konfigurace

Další dialog má dvě části: V horní části lze provést obecné nastavení uživatelů a skupin. V dolní části se nastavují data potřebná pro přístup k LDAP serveru. Nastavení uživatelů a skupin obsahuje následující položky:

**Souborový server** Pokud je aktuální systém souborový server pro uživatelské adresáře (/home), povolte tuto volbu.

**Povolit přihlášení LDAP uživatelů** Povoláním této volby umožníte uživatelům spravovaným přes LDAP přihlásit se do vašeho systému.

**Vlastnosti členství skupiny** Zde nastavte typ LDAP skupiny. Výchozí je ‘member’, další možností je ‘uniquemember’.

V dolní části nastavte údaje potřebné pro konfiguraci a přístup k LDAP serveru, tj. ‘Základna DN pro konfiguraci’, pod kterou jsou uloženy všechny konfigurační objekty, a ‘Administrativní DN’.

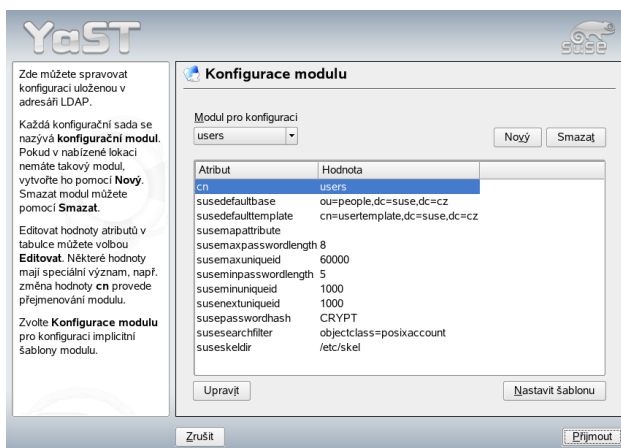
Chcete-li editovat položky na serveru, klikněte na 'Konfigurace správy nastavení uživatelů'. V dialogu, který se objeví, zadejte heslo pro autentizaci na serveru. Bude vám umožněn přístup ke konfiguračním modulům na serveru v souladu s ACL a ACI.

## Důležité

### Použití YaST klienta

YaST LDAP klienta použijte k přizpůsobení YaST modulů pro správu uživatelů a skupin a k jejich případnému rozšíření. Navíc je možné definovat předlohy s výchozími hodnotami jednotlivých atributů pro usnadnění registrace údajů. Tato nastavení jsou sama uložena jako LDAP objekty v LDAP adresáři. Registrace uživatelských dat je stále prováděna pomocí běžných YaST formulářů. Údaje se ukládají jako objekty v LDAP adresáři.

## Důležité



Obrázek 29.4: YaST: Konfigurace modulu

V dialogu pro konfiguraci modulu (29.4 na této straně) lze vybírat a upravovat existující konfigurační moduly a vytvářet a upravovat šablony. Chcete-li upravit hodnotu v konfiguračním modulu nebo modul přejmenovat, vyberte příslušný modul v nabídce. Objeví se seznam všech jeho povolených atributů i s hodno-

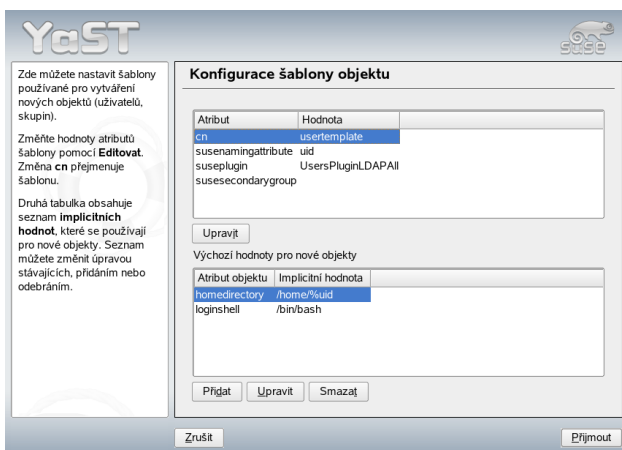


tami. Obsahuje i atributy povolené schématem, ale nepoužité.

Chcete-li změnit hodnotu atributu, vyberte atribut ze seznamu a klikněte na 'Upravit'. Provedené změny potvrdíte tlačítkem 'OK'.

Chcete-li přidat nový modul, klikněte na 'Nový'. Zadejte jméno a objektovou třídu nového modulu (bud' `suseuserconfiguration` nebo `susegroupconfiguration`). Uzavřením dialogu tlačítkem 'OK' přidáte nový modul do seznamu existujících modulů. Kliknutím na 'Smazat' vybraný modul smažete.

Pokud byly předem definovány, obsahují YaST moduly pro správu uživatelů a skupin šablony se smysluplnými výchozími hodnotami. Chcete-li šablonu upravit, klikněte na 'Nastavit šablonu'. Dialog pro nastavení šablon je rozdělen na dvě části. Horní část obsahuje obecné atributy šablony. Upravte je podle potřeby a nebo nechte prázdné. Prázdné atributy budou na LDAP serveru smazány.



Obrázek 29.5: YaST: Konfigurace šablony objektu

Druhá část ('Výchozí hodnoty pro nové objekty') obsahuje všechny atributy odpovídajícího LDAP objektu (v tomto případě konfigurace uživatelů či skupin), pro které se definuje standardní hodnota. Lze přidávat nové a mazat již existující atributy a jejich standardní hodnoty, případně je měnit či mazat. Šablonu zkopírujete změnou hodnoty `cn`. Šablonu spojíte s modulem nastavením hodnoty atributu `susedefaulttemplate` příslušného modulu na DN upravené šablony.

## Tip

Výchozí hodnoty lze vytvářet z jiných atributů pomocí proměnných místo přímého zadání hodnoty. Například při vytváření nového uživatele lze použít `cn=%sn %givenName` a vytvářet tak automaticky hodnotu z `sn` a `givenName`.

## Tip

Jsou-li moduly a šablony správně nastaveny, můžete registrovat nové uživatele a skupiny běžným způsobem v nástroji YaST.

### 29.5.3 Uživatelé a skupiny Konfigurace pomocí YaST

Registrace údajů o uživateli a skupinách se od postupu bez použití LDAP liší jen minimálně. Následující text se vztahuje k registraci uživatelů. Registrace skupin je analogická.

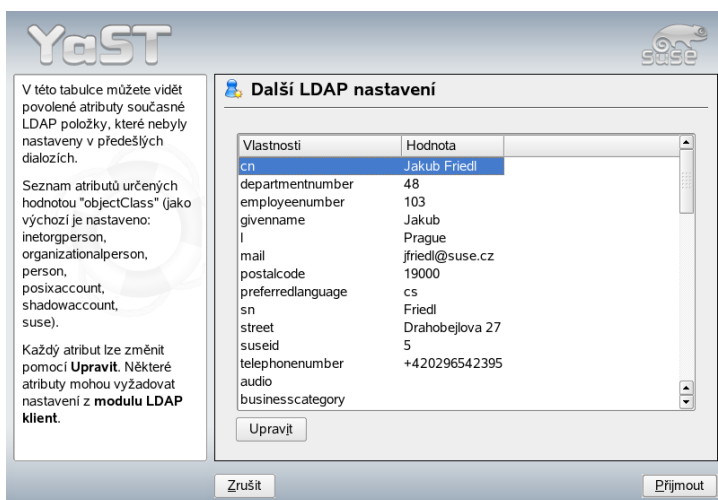
Spustíte YaST modul pro administraci uživatelů pomocí 'Bezpečnost a uživatelé' → 'Správce uživatelů'. Chcete-li prohlížet, přidávat či upravovat LDAP uživatele, klikněte na tlačítko 'Nastavit filtr' vpravo dole a vyberte 'LDAP uživatelé'. Při úpravě údajů o stávajícím uživateli nebo při zakládání nového uživatele pak máte v dialogu k dispozici kartu 'Pluginy'. Kliknete-li v ní na 'Upravit další vlastnosti LDAP uživatele' a pak na tlačítko 'Spustit', objeví se formulář pro zadání údajů specifických pro LDAP (29.6 na následující straně). Vyberte atributy, jejichž hodnotu chcete upravit, a klikněte na 'Upravit'. Zavřením dialogu, který se objeví po kliknutí na 'Přijmout', se vrátíte k hlavnímu dialogu správy uživatelů.

První dialog správy uživatelů obsahuje nabídku 'LDAP volby'. Ta umožňuje použít vyhledávací LDAP filtry a nebo přejít do modulu pro konfiguraci LDAP uživatelů a skupin výběrem 'Správa LDAP uživatelů a skupin'.

## 29.6 Další informace

Tato kapitola neobsahuje řadu témat, jako např. konfiguraci SASL nebo replikaci LDAP serveru, která umožňuje rozložit zatížení na několik strojů. Velmi vyčerpávajícím způsobem je toto nastavení popsáno v *OpenLDAP 2.1 Administrator's Guide* (viz níže).

Velmi rozsáhlou dokumentaci najdete přímo na stránkách projektu OpenLDAP:



Obrázek 29.6: YaST: Další LDAP nastavení

**OpenLDAP Faq-O-Matic** Sbírka otázek a odpovědí týkajících se instalace, konfigurace a správy OpenLDAP je dostupná na adrese <http://www.openldap.org/faq/data/cache/1.html>.

**Quick Start Guide** Jednoduchá instalační příručka LDAP serveru je dostupná na adrese <http://www.openldap.org/doc/admin21/quickstart.html> nebo přímo na vašem počítači v souboru `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`.

### OpenLDAP 2.2 Administrator's Guide

Detailní informace o konfiguraci LDAP včetně kontroly přístupu a šifrování. Příručka je dostupná na adrese <http://www.openldap.org/doc/admin22/> nebo přímo na vašem počítači v souboru `/usr/share/doc/packages/openldap2/admin-guide/index.html`

IBM vydalo o LDAP tyto červené knihy:

**Understanding LDAP** Základní principy LDAP. Kniha je dostupná na adrese <http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf>.

**LDAP Implementation Cookbook** Tato příručka je zaměřená především na administraci *IBM SecureWay Directory*. Obsahuje však také základní informace o LDAP. Naleznete ji na adrese <http://www.redbooks.ibm.com/redbooks/pdfs/sg245110.pdf>.

Tištěné knihy o LDAP:

- Howes, Smith, and Good: *Understanding and Deploying LDAP Directory Services*. Addison-Wesley, 2. Aufl., 2003. (ISBN 0-672-32316-8)
- Hodges: *LDAP System Administration*. O'Reilly & Associates, 2003. (ISBN 1-56592-491-6)

Vynikajícím referenčním manuálem pro LDAP jsou RFC dokumenty 2251–2256.

# Webový server Apache

Jedním z nejrozšířenějších webových serverů na všech platformách je Apache (zdroj: <http://www.netcraft.com>). Apache je často používán spolu s operačním systémem Linux, databází MySQL a programovacími jazyky PHP a Perl. Této kombinaci se často říká *LAMP*.

V následující kapitole se vám pokusíme stručně přiblížit jeho principy, instalaci, základní konfiguraci a dostupné moduly. Jsou zmíněny i virtuální servery.

30.1	Základy . . . . .	478
30.2	Nastavení HTTP serveru pomocí YaST . . . . .	479
30.3	Moduly Apache . . . . .	479
30.4	Vlákna (threads) . . . . .	480
30.5	Instalace . . . . .	481
30.6	Nastavení . . . . .	482
30.7	Používání Apache . . . . .	487
30.8	Aktivní obsah . . . . .	488
30.9	Virtuální servery . . . . .	492
30.10	Bezpečnost . . . . .	495
30.11	Možné problémy . . . . .	497
30.12	Další dokumentace . . . . .	497

## 30.1 Základy

V této části jsou popsány základní principy funkce webového serveru a používané protokoly. Jsou zde představeny i nejdůležitější funkce webového serveru.

### 30.1.1 Webový server

Webový server zasílá na požádání klientům *HTML* stránky. Tyto stránky mohou být uloženy v adresáři (pasivní nebo statické stránky) nebo na požádání vytvořeny (aktivní obsah).

### 30.1.2 HTTP

Klienty obvykle rozumíme webové prohlížeče jako Konqueror nebo Mozilla. Komunikace mezi klientem a serverem obvykle probíhá podle protokolu *Hyper Text Transfer Protocol* (HTTP). Současná verze HTTP 1.1 je popsána v RFC 2068 a v aktualizaci RFC 2616. Tyto dokumenty jsou k dispozici na stránce `http://www.w3.org`.

### 30.1.3 URL

Klienti pro dotazy používají URL stránek. Například `http://www.novell.com/index_us.html`. URL se skládá z:

**Protokolu** Nejpoužívanější protokoly:

**http://** HTTP protokol

**https://** Bezpečná šifrovaná verze HTTP protokolu

**ftp://** FTP protokol pro přenos souborů

**Domény** V našem příkladě `www.novell.com`. Doménu lze rozdělit do dvou částí. První část (*>www*) ukazuje na počítač. Vlastní doménu tvoří druhá část (`novell.com`). Společně tvoří tzv. FQDN (Fully Qualified Domain Name).

**Zdroje** V našem případě `index_us.html`. Tato část specifikuje úplnou cestu ke zdroji. Zdroje mohou být soubory, ale i CGI skripty, stránky v Javě atd.

Díky různým mechanismům prohledávání domén (jako DNS) je dotaz doručen správnému počítači. Apache pak ze své adresářové struktury doručí aktivní zdroj (v našem případě stránka `index_us.html`). V našem případě je zdroj přímo v hlavním adresáři serveru. Zdroje lze však umístit také do podadresářů, např. `http://support.novell.com/linux/`

Cesta k souboru je relativní vzhledem k hodnotě `DocumentRoot`, kterou lze nastavit v konfiguračním souboru. Popis najdete v části 30.6.2 na straně 484.

### 30.1.4 Automatický výstup výchozí stránky

Pokud neuvedete výchozí stránku, Apache automaticky připojí obvyklé jméno. Ve většině případů se jedná o `index.html`. Tato funkce včetně jmen stránek, které má server používat, může být nakonfigurována podle popisu v části 30.6.2 na straně 485.

## 30.2 Nastavení HTTP serveru pomocí YaST

Apache snadno nastavíte pomocí programu YaST. Nastavení vyžaduje alespoň základní znalosti o nastavení webového serveru. Po výběru 'Síťové služby' → 'HTTP server' vás může YaST před samotným nastavením webového serveru požádat o doinstalování potřebných balíčků. Po úspěšné instalaci se zobrazí konfigurační dialog.

Nejdřív povolte spuštění serveru zatrhnutím položky 'Povoleno'. Zaškrtnutím 'Na zvolených portech otevřít firewall' otevřete potřebné porty. Ve spodní části okna ('Nastavení/Shrnutí') lze nastavit vlastnosti HTTP serveru: 'Naslouchat na' (výchozí je `port 80`), 'Moduly', 'Výchozí server' a 'Servery'. Zvolenou položku změníte kliknutím na tlačítko 'Upravit'.

Nejdřív překontrolujte nastavení položky 'Výchozí server' a případně ji přizpůsobte svému serveru. Pak aktivujte potřebné moduly v položce 'Moduly'. Dostupné jsou také další dialogy umožňující detailnější nastavení např. vytváření virtuálních serverů.

## 30.3 Moduly Apache

Pomocí modulů lze Apache rozšířit o řadu funkcí např. o schopnost pracovat s CGI skripty v různých jazycích. Mimo tradičních jazyků jako Perl a PHP jsou k

dispozici také jazyky Python a Ruby. Použít lze mimo jiné i moduly pro bezpečný přenos dat (secure sockets layer - SSL), ověřování uživatelů, rozšířené logování a mnoho dalších.

S dostatkem know-how můžete Apache pomocí vlastních modulů přizpůsobit libovolným požadavkům. Více informací najdete v části 30.12.4 na straně 498.

Modularizace Apache dospěla tak daleko, že je moduly řešeno v podstatě vše kromě nejjednodušších úkolů. Dospělo to tak daleko, že dokonce samotné HTTP je zpracováváno moduly. Apache proto vůbec nemusí fungovat jako webserver. S patřičnými moduly může sloužit úplně jiným účelům. Byl například nasazen jako poštovní server (POP3).

Moduly Apache podporují řadu dalších užitečných funkcí:

**Virtuální servery** Podpora funkce virtuálního serveru znamená, že na jednom počítači s jednou instancí Apache lze provozovat více webů, které se návštěvníkům jeví jako samostatné servery. Virtuální servery mohou používat různé IP adresy nebo jména. Tak ušetříte výdaje za další hardware a software.

**Flexibilní přepis URL** Apache nabízí řadu možností, jak manipulovat a přepisovat URL. Více informací najdete v dokumentaci Apache.

**Content Negotiation** Apache umí klientovi (prohlížeči) doručit stránku ve stavu, který odpovídá jeho zobrazovacím schopnostem. Například starým prohlížečům nepodporujícím rámce pošle stránku bez rámců. Pokud jste ochotni připravit JavaScript zvlášť pro každý typ prohlížeče, můžete takto obejít případné nekompatibility v jeho implementaci.

**Flexibilní nakládání s chybami** Apache na chybu, například chybějící stránku, dokáže reagovat flexibilně a odpovídajícím způsobem. Odpověď je možno generovat i dynamicky, například pomocí CGI.

## 30.4 Vlákna (threads)

Vlákno je jednoduchý proces. Výhoda vláken leží v nižší spotřebě zdrojů, čímž se zvyšuje výkon. Nevýhodou je, že aplikace musí být tzv. thread-safe. To znamená:

- Funkce (nebo metody v objektově orientovaných aplikacích) musí být re-entrantní (vícenásobně přístupné) – funkce se stejným vstupem vždy vrátí stejný výstup, i když je současně vykonávána jiným vláknem. Funkce tedy musí být navrženy tak, aby mohly být vykonávány současně více vlákny.



- Přístup ke zdrojům (obvykle proměnným) musí být řízen tak, aby současně běžící vlákna nepřicházela do konfliktu.

Apache 2 přistupuje k dotazům jako odděleným procesům, nebo, ve smíšeném režimu, jako kombinaci procesů a vláken. Za zpracování dotazů jako procesů zodpovídá MPM *prefork*, za zpracování jako vláken MPM *worker*. Výběr MPM můžete provést při instalaci (viz 30.5 na této straně). Třetí režim – *perchild* – není zatím vyzrálý a není proto v naší distribuci dostupný.

## 30.5 Instalace

### 30.5.1 Výběr balíků v programu YaST

Vše, co pro základní instalaci potřebujete, je nainstalovat balík obsahující Apache, tj. `apache2`. Navíc nainstalujte jeden z balíčků s MPM (multiprocessing module), např. `apache2-prefork` nebo `apache2-worker`. Pokud zvolíte MPM, pamatujte, že MPM s podporou vláken (worker) nelze použít s balíkem `mod_php4`, protože některé knihovny z tohoto balíčku stále nesplňují podmínku bezpečnosti vláken.

### 30.5.2 Aktivace Apache

Apache se po instalaci nespouští automaticky. Je nutné ho aktivovat v editoru úrovní běhu. Pokud ho chcete spouštět vždy při startu, zvolte v editoru úrovní běhu úroveň 3 a 5. Zda je Apache aktivní, zjistíte zadáním adresy `http://localhost/` ve svém prohlížeči. Je-li aktivní, zobrazí se testovací stránky obsažené v balíčku (pokud je nainstalován) `apache2-example-pages`.

### 30.5.3 Moduly pro aktivní obsah

Abyste mohli používat aktivní obsah, musíte mít nainstalován modul s podporou příslušného jazyka, který se rozhodnete používat. K dispozici máte `apache2-mod_perl` pro Perl, `mod_php4` pro PHP a `mod_python` pro Python. Použití těchto modulů je popsáno v části 30.8.4 na straně 489.

### 30.5.4 Další doporučené balíky

V některých případech je vhodné doinstalovat rozšířenou dokumentaci, kterou najdete v balíčku `apache2-doc`. Po instalaci balíčku a spuštění serveru lze k dokumentaci přistupovat přímo přes URL `http://localhost/manual`.

Pro vývoj nových modulů nebo jejich kompilaci potřebujete balíček `apache2-devel` a vývojové nástroje. Ty zahrnují `apxs` nástroje popsané v části 30.5.5 na této straně.

### 30.5.5 Instalace modulů pomocí `apxs`

Příkaz `apxs2` je důležitý nástroj pro vývojáře modulů. Díky tomuto příkazu je možné jedním příkazem překompilovat i nainstalovat požadovaný nový modul (včetně provedení potřebných změn v konfiguračních souborech). Tímto příkazem lze instalovat také moduly dostupné jako objektové soubory (koncovka `.o`) nebo statické knihovny (koncovka `.a`). Ze zdrojového kódu příkaz `apxs2` vytvoří DSO (Dynamic Shared Object), který může Apache používat jako modul.

Instalaci modulu ze zdrojového kódu lze provést příkazem jako `apxs2 -c -i -a mod_foo.c`. Další volby tohoto příkazu jsou popsány v manuálové stránce. Moduly je pak třeba aktivovat v souboru `/etc/sysconfig/apache2` položkou `APACHE_MODULES`, jak je popsáno v části 30.6.1 na následující straně.

`apxs2` je dostupný v několika verzích: `apxs2`, `apxs2-prefork` a `apxs2-worker`. `apxs2` instaluje moduly tak, aby je mohly používat všechny MPM. Ostatní programy instalují moduly tak, že mohou být používány pouze příslušnými MPM. `apxs2` instaluje moduly do `/usr/lib/apache2`. `apxs2-prefork` instaluje moduly do `/usr/lib/apache2-prefork`.

## 30.6 Nastavení

Pokud potřebujete zvláštní nastavení, proved'te je po instalaci Apache. V naprosté většině případů můžete Apache používat, jak je. Apache lze nastavit pomocí YaST a SuSEconfig nebo přímou editací souboru `/etc/apache2/httpd.conf`.

### 30.6.1 Konfigurace pomocí skriptu SuSEconfig

Nastavení v `/etc/sysconfig/apache2` jsou do konfiguračních souborů Apache zapisována pomocí skriptu SuSEconfig. Předkonfigurovaná nastavení by měla být vhodná pro většinu běžných nasazení. Soubor obsahuje u každé proměnné vysvětlující komentář.

#### Vlastní konfigurační soubory

Místo zápisu změn přímo do konfiguračního souboru `/etc/apache2/httpd.conf` si s pomocí proměnné `APACHE_CONF_INCLUDE_FILES` můžete vytvořit vlastní konfigurační soubor (např. `httpd.conf.local`). Tento soubor pak bude interpretován hlavním konfiguračním souborem. Tak si zachováte vlastní nastavení i v případě přepsání souboru `/etc/apache2/httpd.conf` během reinstalace serveru.

#### Moduly

Moduly instalované programem YaST mohou být aktivovány zapsáním jména modulu do seznamu pod proměnnou `APACHE_MODULES`. Tato proměnná se nachází v souboru `/etc/sysconfig/apache2`.

#### Návěští

`APACHE_SERVER_FLAGS` se používá k nastavení návěští (flag), které aktivují či deaktivují určité části konfiguračního souboru. Pokud je sekce v konfiguračním souboru vymezena takto:

```
<IfDefine návěští>
.
.
.
</IfDefine>
```

aktivuje se pouze nastavením příslušného návěští v proměnné `ACTIVE_SERVER_FLAGS`: `ACTIVE_SERVER_FLAGS = návěští`. Tímto způsobem lze bez problémů aktivovat či deaktivovat poměrně rozsáhlé části konfiguračního souboru např. pro testovací účely.

## 30.6.2 Ruční nastavení

Konfigurační soubor `/etc/apache2/httpd.conf` umožňuje změny, které nejsou dostupné nastavením v `/etc/sysconfig/apache2`. V této části si popíšeme některé parametry, které lze v tomto souboru nastavit. Jsou zmíněny v pořadí, v jakém se nacházejí v konfiguračním souboru.

### DocumentRoot

Jedno ze základních nastavení je `DocumentRoot` určující adresář s obsahem webu. Pro výchozí virtuální server je nastaven na `/srv/www/htdocs`. Obvykle toto nastavení není nutné měnit.

### Timeout

Nastavení timeoutu pro dotazy.

### MaxClients

Maximální počet klientů, jejichž požadavky může Apache vyřizovat současně. Výchozí nastavení je 150, ale tato hodnota může být pro vytíženější weby malá.

### LoadModule

`LoadModule` určuje moduly, které se mají nahrát. Pořadí nahrávání je určeno přímo moduly. Uvádějí se zde i soubory obsahující moduly.

### Port

Určuje port, na kterém Apache naslouchá. Obvykle jde o port 80, výchozí port služby HTTP. Za normálních okolností byste toto nastavení neměli měnit. Jedním z důvodů, proč by Apache měl naslouchat na jiném portu, je test nové verze webových stránek. V takovém případě je platná verze stránek stále dostupná na portu 80.

Jiným důvodem je dostupnost stránek pouze na intranetu (z bezpečnostních důvodů). V takovém případě nastavte např. 8080 a zablokujte externí přístup na port firewallem. Tak bude server chráněn proti externím přístupům.

### Directory

Nastavení přístupových práv pro adresář. Tato položka existuje i pro `DocumentRoot`. Jméno adresáře musí být změněno vždy, když je změněn *DocumentRoot*.

## DirectoryIndex

Zde určíte, v jakém souboru má Apache hledat výchozí stránku. Jako výchozí je nastavena `index.html`. Pokud pak zadáte například `http://www.xyz.com/foo/bar` a adresář `foo/bar/` obsahuje soubor `index.html`, Apache vrátí klientovi tuto stránku.

## AllowOverride

Každý adresář Apache, ze kterého jsou doručovány dokumenty, může obsahovat soubor, který může přepisovat globální nastavení a nastavení přístupových práv adresáře. Tato nastavení se aplikují rekurzivně na aktuální adresář a jeho podadresáře, dokud nejsou přepsány jiným podobným souborem v podadresáři. Nastavení v souboru umístěném v *DocumentRoot* je aplikováno globálně. Obvykle jsou tyto soubory pojmenovány `.htaccess`, ale to lze změnit (viz 30.6.2 na následující straně).

Pomocí `AllowOverride` nastavte, zda jsou tyto přepisy globálního nastavení povoleny. Možné hodnoty jsou `None`, `All` a jakákoliv kombinace `Options`, `FileInfo`, `AuthConfig` a `Limit`. Význam hodnot je popsán v dokumentaci Apache. Bezpečné výchozí nastavení je `None`.

## Order

Určuje pořadí, ve kterém jsou aplikována nastavení pro *Allow* a *Deny*. Výchozí nastavení je:

```
Order allow,deny
```

Tak je nejprve aplikováno nastavení pro povolení přístupu, pak pro zákaz. Význam záznamu:

***allow all*** povolí veškerý přístup a určí výjimky

***deny all*** zakáže veškerý přístup a určí výjimky

Příklad pro `deny all`:

```
Order deny,allow
Deny from all
Allow from example.com
Allow from 10.1.0.0/255.255.0.0
```

## AccessFileName

Zde zadejte jméno pro soubory, které mohou přepisovat globální nastavení práv a další pro adresáře doručované Apachem (viz 30.6.2 na předchozí straně). Výchozí nastavení je `.htaccess`.

## ErrorLog

Určuje jméno souboru, kam se zapisují chybová hlášení Apache. Výchozí nastavení je `/var/log/httpd/errorlog`. Chybová hlášení virtuálních serverů (viz 30.9 na straně 492) jsou do tohoto souboru zapisována také, pokud ovšem nebyl v sekci *VirtualHost* nastaven jiný, zvláštní soubor.

## LogLevel

Chybová hlášení jsou rozdělena do několika úrovní závažnosti. Toto nastavení určuje, jaké stupně budou zapisovány. Nastavením určitého stupně se budou zapisovat chybová hlášení tohoto stupně a vyšší. Výchozí nastavení je `warn`.

## Alias

Použitím aliasu můžete určit zkratku adresáře pro přímý přístup. Například alias `/manual/` umožňuje přístup do `/srv/www/htdocs/manual` i v případě, že je `DocumentRoot` nastaven na jiný adresář než `/srv/www/htdocs`. Pomocí aliasu `http://localhost/manual` je pak možný přímý přístup do zmíněného adresáře. U adresáře určeného v `Alias` můžete potřebovat nastavit práva, učiníte tak pomocí direktivy `Directory`. Viz 30.6.2 na straně 484.

## ScriptAlias

Tato položka je podobná položce `Alias`. Navíc říká, že soubory v cílovém adresáři jsou CGI skripty.

## Server-Side Includes

Server-side includes lze aktivovat vyhledáním SSI ve všech spustitelných souborech. To provedete tímto příkazem:

```
<IfModule mod_include.c>  
XBitHack on </IfModule>
```

Aby byl soubor s SSI vykonatelný, použijte příkaz `chmod +x` `<JmenoSouboru>`. Jinou možností je explicitně zadat typ souborů, ve kterých se má SSI hledat. To lze učinit následující instrukcí:

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

Není rozumné nastavit `.html`, protože by Apache SSI vyhledával ve všech stránkách (včetně těch, kde žádné určité nejsou) a došlo by ke značnému zvýšení zátěže. SUSE LINUX již tyto položky obsahuje, není proto nutné v tomto směru nic měnit.

### UserDir

S pomocí modulu `mod_userdir` a direktivy `UserDir` můžete nastavit jméno adresáře, ze kterého se v případě jeho existence v domovském adresáři jednotlivých uživatelů budou stránky automaticky publikovat Apachem. Toto chování lze nastavit také skriptem `SuSEconfig` nastavením proměnné `HTTPD_SEC_PUBLIC_HTML` na `yes`. Výsledkem je následující položka v souboru `/etc/apache2/mod_userdir.conf` (který je interpretován souborem `/etc/apache2/httpd.conf`).

```
<IfModule mod_userdir.c>
UserDir public_html
</IfModule>
```

## 30.7 Používání Apache

Abyste zobrazili statické webové stránky, stačí je umístit do správného adresáře. V SuSE LINUXu jde o adresář `/srv/www/htdocs`. Několik pokusných stránek je zde již nainstalováno. Tak si můžete ověřit, zda Apache běží správně. Tyto soubory můžete přepsat nebo smazat. Pro běh Apache nejsou nutné. CGI skripty jsou instalovány do `/srv/www/cgi-bin`.

Během svého běhu Apache zapisuje zprávy do souborů `/var/log/httpd/access_log` nebo `/var/log/apache2/access_log`. V těchto zprávách je uvedeno, jaké zdroje byly žádány, jaké doručeny, v jakém čase a jakou metodou (GET, POST atd.). Chybové zprávy jsou zapisovány do souboru `/var/log/apache2.`

## 30.8 Aktivní obsah

Apache nabízí několik způsobů, jak klientovi doručit aktivní obsah. Aktivní obsah HTML stránek je generován v závislosti na datech získaných od klienta. Např. vyhledávače poskytují seznam stránek na základě dotazu uživatele.

Apache generuje aktivní obsah třemi způsoby:

**SSI (Server Side Includes)** Jde o příkazy přímo v HTML stránce zapsané jako speciální komentáře. Apache komentáře interpretuje, vytvoří příslušný obsah a výsledek pošle jako část HTML stránky.

**CGI (Common Gateway Interface)** Programy v určitém adresáři. Apache jim předá parametry obdržené od klienta a klientovi vrátí výstup těchto programů. To je poměrně jednoduchý způsob, neboť lze snadno přizpůsobit mnoho existujících programů pro příkazovou řádku, aby takto spolupracovaly s Apachem.

**Moduly** Apache nabízí rozhraní pro vykonání jakéhokoliv modulu. Moduly jsou programy pracující s informacemi získanými od Apache. Apache umožňuje modulům přístup k důležitým informacím jako HTTP hlavičkám. Moduly lze použít kromě generování aktivních stránek také k jiným funkcím (například ověřování uživatele). Jejich výhodou je vysoký výkon a možnosti překonávající SSI i CGI.

Normálně jsou CGI skripty vykonávány přímo serverem Apache pod uživatelským ID jejich vlastníka. Naopak moduly jsou kontrolovány interpretem, který je v serveru Apache obsažen. Není tak nutné pro každý dotaz spouštět a ukončovat samostatný proces (což zvyšuje zátěž). Skript je interpretem spuštěn pod ID webserveru.

Toto řešení má i své chyby. CGI skripty jsou totiž oproti modulům velmi robustní. Při jejich použití nemají chyby při správě zdrojů a paměti tak ničivé následky jako u modulů, neboť dojde k ukončení programu po vyřízení požadavku. Při použití modulů může dojít ke kumulaci chyb. Pokud server běží bez restartu delší dobu, mohou se chyby hromadit a vést k nestabilitě systému.

### 30.8.1 SSI

Server-side includes jsou příkazy ve zvláštních komentářích vykonávané Apachem. Výsledek je zahrnut ve výstupu. Například aktuální datum lze zahrnout pomocí `<!--#echo var="DATE_LOCAL" -->`. Znak # na konci otvírací



značky (`<!--`) říká indiánovi, že se jedná o SSI direktivu a nikoliv o obyčejný komentář.

SSI lze aktivovat několika způsoby. Nejjednodušší je vyhledat SSI ve všech spustitelných souborech. Jiná možnost je určit, ve kterých souborech se má SSI hledat. Obojí je vysvětleno v části 30.6.2 na straně 486.

## 30.8.2 CGI

CGI je zkratka z anglického *Common Gateway Interface*. Díky CGI je server schopný zasílat mimo klasických statických stránek také dynamicky generované stránky. Tak je možné vytvářet stránky, které jsou výsledkem výpočtu nebo hledání v databázi. V závislosti na obdržené proměnné je server schopný vytvářet na každý dotaz zvláštní stránky lišící se obsahem.

Hlavní výhoda technologie CGI je jednoduchost. Programy jsou obvykle uloženy v určitém adresáři a spouštěny serverem jako jakékoliv jiné programy v systému. Server pak zašle výstup programu ze standardního výstupu (`stdout`) klientovi.

Teoreticky mohou být CGI napsány v libovolném programovacím jazyce. Obvykle jsou k tomuto účelu používány skriptovací jazyky jako Perl nebo PHP. Pokud je rychlost kritická, může být vhodnější C/C++.

V nejjednodušším případě hledá indián tyto programy ve zvláštním adresáři (`cgi-bin`). Ten lze nastavit v konfiguračním souboru (viz 30.6 na straně 482). Pokud je potřeba, mohou být nastaveny další takové adresáře. Je však nebezpečné umožnit Apache spouštět programy uživatele. Pokud jsou CGI omezeny na adresář `cgi-bin`, může administrátor lépe kontrolovat jejich obsah.

## 30.8.3 GET a POST

Vstupní parametry mohou být serveru doručeny pomocí *GET* nebo *POST*. V závislosti na použité metodě předává server hodnoty skriptu různým způsobem. Při *POST* budou parametry předávány přes standardní vstup (`stdin`). (Program vstup obdrží stejným způsobem, jako by byl spuštěn z příkazové řádky.) U metody *GET* použije server k předání proměnnou prostředí `QUERY_STRING`.

## 30.8.4 Generování aktivního obsahu pomocí modulů

Pro webový server Apache je dostupných mnoho různých modulů. Termín *modul* je zde používán ve dvou různých významech. První představuje moduly integro-

vané přímo do Apache a ošetřující zvláštní funkce, jako je podpora programovacích jazyků.

Druhý význam je spojen s programovacími jazyky. Moduly zde odkazují na nezávislou skupinu funkcí, tříd a proměnných. Tyto moduly jsou integrovány do programu a poskytují různé funkce, jako např. CGI moduly pro skriptovací jazyky. Tyto moduly umožňují CGI programování poskytováním různých funkcí, jako jsou metody čtení parametrů dotazů a metody pro HTML výstup.

## 30.8.5 mod\_perl

Perl je populární a prověřený skriptovací jazyk. Existuje pro něj řada modulů a knihoven včetně knihovny pro rozšíření konfiguračního souboru Apache. Domovská stránka Perlu se nachází na adrese Řada knihoven je dostupná v Comprehensive Perl Archive Network (CPAN) na adrese <http://www.cpan.org/>.

### Nastavení mod\_perl

Modul `mod_perl` nastavíte instalací příslušného balíčku (viz 30.5 na straně 481). Po instalaci se v konfiguračním souboru automaticky objeví všechny důležité položky (viz `/etc/apache2/mod_perl-startup.pl`). Informace o *mod\_perl* jsou dostupné na stránce <http://perl.apache.org/>.

### mod\_perl versus CGI

V nejjednodušším případě spustíte předešlý CGI skript jako `mod_perl` skript dotazem z jiné adresy. Konfigurační soubor obsahuje aliasy, které odkazují na stejný adresář a vykonají každý zde obsažený skript prostřednictvím buď CGI nebo `mod_perl`. Všechny položky již v konfiguračním souboru existují. Alias pro CGI je:

```
ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
```

Položky pro `mod_perl` jsou:

```
<IfModule mod_perl.c>
# Provide two aliases to the same cgi-bin directory,
# to see the effects of the 2 different mod_perl modes.
# for Apache::Registry Mode
ScriptAlias /perl/ "/srv/www/cgi-bin/"
# for Apache::Perlrun Mode
ScriptAlias /cgi-perl/ "/srv/www/cgi-bin/"
</IfModule>
```

Pro `mod_perl` jsou potřebné také následující položky. Tyto položky se již v konfiguračním souboru nacházejí.

```
#
# If mod_perl is activated, load configuration information
#
<IfModule mod_perl.c>
PerlRequire /usr/include/apache/modules/perl/startup.perl
PerlModule Apache::Registry

#
# set Apache::Registry Mode for /perl Alias
#
<Location /perl>
SetHandler perl-script
PerlHandler Apache::Registry
Options ExecCGI
PerlSendHeader On
</Location>

#
# set Apache::PerlRun Mode for /cgi-perl Alias
#
<Location /cgi-perl>
SetHandler perl-script
PerlHandler Apache::PerlRun
Options ExecCGI
PerlSendHeader On
</Location>

</IfModule>
```

Tyto položky vytvoří aliasy pro režimy *Apache::Registry* a *Apache::PerlRun*. Rozdíly mezi těmito režimy jsou následující:

***Apache::Registry*** Všechny skripty jsou překompilovány a uloženy do vyrovnávací paměti. Každý skript je pak používán jako obsah subrutiny. Přestože tak získáte vysoký výkon, jsou zde i nevýhody. Skript je nutné napsat s extrémní opatrností, protože proměnné a subrutiny mezi jednotlivými požadavky přetrvávají. Znamená to, že vždy musíte každou proměnnou ošetřit tak, aby se před použitím rutiny dalším dotazem vynulovala. Například pokud ve skriptu uložíte jako proměnnou číslo bankovní karty, bez vynulování se může stát, že se číslo karty použije i u dalšího zákazníka.

***Apache::PerlRun*** Skripty jsou pro každý požadavek rekompileovány. Všechny proměnné mezi požadavky mizí. Proto `Apache::PerlRun` nevyžaduje

tak pečlivé programování, ale je pomalejší než `Apache::Registry`. Stále je však mnohem rychlejší než CGI (navzdory podobnostem), protože není spouštěn zvláštní proces pro interpret.

### 30.8.6 mod\_php4

PHP je jazyk vyvinutý speciálně pro webové servery. Na rozdíl od jiných jazyků, které využívají pro své příkazy samostatné soubory (skripty), PHP lze vložit přímo do HTML stránky (podobně jako SSI). PHP interpret zpracuje vložené PHP příkazy a vygeneruje výsledek do webové stránky.

Domovskou stránku PHP najdete na adrese <http://www.php.net/>. Pro použití PHP musíte nainstalovat balíčky `mod_php4-core` a `apache2-mod_php4`.

### 30.8.7 mod\_python

Python je objektově orientovaný jazyk s velmi jasnou a čitelnou syntaxí. Neobvyklou ale velmi užitečnou vlastností je struktura programu závislá na odsazení. Jednotlivé bloky od sebe nejsou odděleny složenými závorkami (jako v C a Perlu) ani jinými oddělovači (jako `begin` a `end`), ale stupněm odsazení. Pro podporu hada potřebujete balíček `apache2-mod_python`.

Více informací o tomto jazyce najdete na stránce <http://www.python.org/>. Informace o `mod_python` jsou dostupné na <http://www.modpython.org/>.

### 30.8.8 mod\_ruby

Ruby je poměrně nový objektově orientovaný jazyk s prvky Perlu a Pythonu. Stejně jako Python má jasnou a transparentní syntaxi. Na druhou stranu obsahuje zkratky jako `$.r` pro číslo poslední řádky načtené ze vstupního souboru, což je vlastnost, kterou někteří programátoři vítají a jiní nenávidí. Koncept Ruby částečně převzal ze Smalltalku.

Domovskou stránku Ruby najdete na adrese <http://www.ruby-lang.org/>. Apache modul má domovskou stránku <http://www.modruby.net/>.

## 30.9 Virtuální servery

Virtuální servery umožňují hostovat na jednom počítači více domén. Je to spolehlivý a ověřený způsob, jak ušetřit náklady na administraci zvláštního serveru

pro každou doménu. Apache nabízí hned několik možností, jak virtuální servery nastavit:

- Virtuální server založený na jménu.
- Virtuální server založený na IP.
- Vícenásobné instance Apache na jednom počítači.

### 30.9.1 Virtuální server založený na jménu

Virtuální server založený na jménu hostuje na jedné instanci Apache několik domén. Není nutné nastavovat žádné další IP adresy. Jedná se o nejjednodušší a nejčastěji používanou možnost. Důvody proti této konfiguraci najdete v dokumentaci Apache.

Konfigurace se provádí přímo v konfiguračním souboru `/etc/apache2/httpd.conf`. Abyste aktivovali virtuální server založený na jménu, musíte zadat `NameVirtualHost *`. Nastavení `*` způsobí, že bude Apache přijímat všechny příchozí požadavky. Pak nastavte jednotlivé servery:

```
<VirtualHost *>
    ServerName www.example.com
    DocumentRoot /srv/www/htdocs/example.com
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com-error_log
    CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>

<VirtualHost *>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>
```

`VirtualHost` musí být nastaven i pro originální doménu serveru (`www.example.com`). Originální doména i dodatečná doména (`www.myothercompany.com`) jsou v našem příkladě hostovány na stejném serveru.

Stejně jako v `NameVirtualHost` je v direktivách `VirtualHost` použita `*`. Apache používá pole "host" v HTTP hlavičce pro spojení požadavků s virtuálním serverem. Požadavek je doručen tomu virtuálnímu serveru, jehož nastavení v `ServerName` odpovídá údajům v HTTP hlavičce.

Pro direktivy `ErrorLog` a `CustomLog` nemusí záznamy obsahovat jméno domény. Použijte jméno podle vlastní volby.

`Serveradmin` obsahuje e-mailovou adresu osoby, která má být kontaktována v případě problémů. Apache tuto adresu předává klientům v případě potíží.

## 30.9.2 Virtuální server založený na IP

Alternativou serveru založeného na jménu je nastavení více IP adres pro jeden jediný počítač. V takovém případě jediná instance Apache hostuje více domén s různými IP adresami. V následujícím příkladu si ukážeme konfiguraci Apache používajícího vlastní IP adresu (`192.168.1.10`) plus další dvě dodatečné IP adresy (`192.168.1.20` a `192.168.1.21`). Tento konkrétní příklad funguje pouze na intranetu, protože se jedná o privátní adresy, které nejsou na Internetu směrovány.

### Nastavení IP aliasů

Aby Apache mohl pracovat s více IP, musí počítač přijímat požadavky na více IP. Tomu se říká multi-IP hosting. Tato funkce vyžaduje podporu IP aliasingu v jádře. Tato podpora je v SUSE Linuxu výchozí.

Pokud je v jádře povolen IP aliasing, lze pomocí příkazů `ifconfig` a `route` nastavovat další IP adresy počítače. Tyto příkazy musí vykonávat uživatel `root`. V následujícím příkladě budeme předpokládat, že počítač již má vlastní IP adresu (např. `192.168.1.10`), která je přiřazena zařízení `eth0`.

Příkazem `ifconfig` bez parametrů zjistíte IP adresu počítače. Další IP nastavíte příkazem:

```
ip addr add 192.168.1.20/24 dev eth0
```

Všechny IP adresy používají stejné síťové fyzické zařízení (`eth0`).

### Virtuální počítače s IP

Jakmile je na počítači nastaveno IP aliasování nebo má počítač více síťových karet, můžete nastavit virtuální servery Apache. Pro každý virtuální server musíte vložit vlastní blok `VirtualHost`:

```
<VirtualHost 192.168.1.20>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.21>
    ServerName www.anothercompany.com
    DocumentRoot /srv/www/htdocs/anothercompany.com
    ServerAdmin webmaster@anothercompany.com
    ErrorLog /var/log/apache2/www.anothercompany.com-error_log
    CustomLog /var/log/apache2/www.anothercompany.com-access_log common
</VirtualHost>
```

Proměnná `VirtualHost` se používá pouze pro dodatečné domény. Výchozí doména (`www.example.com`) je nastavena zvlášť v `DocumentRoot` mimo bloky `VirtualHost`.

### 30.9.3 Vícenásobné instance Apache

Při použití výše zmíněných metod může administrátor jedné domény číst data ostatních domén. Abyste jednotlivé domény oddělili, musíte spustit další instance Apache, které budou používat zvláštní nastavení uživatele, skupiny a dalších proměnných v konfiguračním souboru.

V konfiguračním souboru nastavte proměnnou `Listen` na IP adresy obsluhované jednotlivými instancemi Apache. V našem případě bude zápis pro první instanci:

```
Listen 192.168.1.10:80
```

A pro další dvě instance:

```
Listen 192.168.1.20:80
Listen 192.168.1.21:80
```

## 30.10 Bezpečnost

### 30.10.1 Minimalizace rizika

Pokud Apache nepotřebujete, deaktivujte jeho spouštění v editoru úrovní běhu nebo ho oddinstalujte. Pokud chcete bezpečnostní rizika minimalizovat úplně,

vypněte i další serverové služby. Platí to zejména pro počítače používané jako firewally. Na těch pokud možno nespouštějte žádné služby.

## 30.10.2 Přístupová práva

### DocumentRoot by měl patřit uživateli root

Jako výchozí vlastník adresáře *DocumentRoot* (`/srv/www/htdocs`) a adresáře CGI je nastaven uživatel `root`. Pokud je adresář zapisovatelný pro všechny, může do něj umístit soubory jakýkoliv uživatel. Tyto soubory pak budou vykonány Apachem pod uživatelem `wwwrun`. Apache by neměl mít práva zápisu do adresářů s daty a skripty, které dodává. Proto by neměl být vlastníkem těchto adresářů uživatel `wwwrun`, ale jiný uživatel (např. `root`).

Aby mohli do adresáře s dokumenty umístit své soubory také jiní uživatelé, musí mít práva k zápisu. Takové řešení však není bezpečné. Pokud máte možnost, vytvořte raději nový adresář, kam budou mít práva zápisu všichni (např. `/srv/www/htdocs/miscellaneous`).

### Publikování dokumentů z domovských adresářů

Jiný způsob, jak zajistit, aby uživatelé mohli publikovat své stránky, je určení jednoho přesného jména adresáře v domovském adresáři, kam se mají stránky určené k publikaci ukládat. Jméno tohoto podadresáře je obvykle `~/public_html`. To je výchozí nastavení v systému SUSE LINUX. Více viz část 30.6.2 na straně 487.

Webové stránky pak můžete zobrazit zadáním jména uživatele v URL, pomocí části `~uživatel`. K zobrazení obsahu adresáře `public_html` uživatele `tux` zadejte do prohlížeče adresu `http://localhost/~tux`.

## 30.10.3 Aktualizace

Pokud provozujete webový server, který je veřejně přístupný, nezanedbávejte pravidelnou aktualizaci. Snažte se pravidelně získávat informace o bezpečnostních chybách a problémech. Zdroje, které vám v tom pomohou, najdete v části 30.12.3 na straně 498.



## 30.11 Možné problémy

Pokud se objeví problémy, např. Apache odmítne zobrazit stránku nebo ji nezobrazí správně, mohou vám pomoci následující postupy. Nejprve se podívejte do souboru `/var/log/apache2/error_log`, zda neobsahuje zprávy vysvětlující problém.

Spolehlivý postup je sledovat záznamy v konzoli a sledovat reakci na přístup k serveru. Lze tak učinit příkazem `tail -f /var/log/apache2/*_log` zadáním uživatelem `root`.

Podívejte se také do databáze chyb na stránce <http://bugs.apache.org/>. Přihlaste se do uživatelské konference Apache dostupné na adrese <http://httpd.apache.org/userslist.html>. Doporučujeme také novinky (newsgroup) [comp.infosystems.www.servers.unix](http://comp.infosystems.www.servers.unix).

Pokud jste stále nenalezli řešení a jste si jisti, že se jedná o chybu v Apache, nahlaste ji na <http://www.suse.de/feedback/> a nebo, česky, na [feedback@suse.cz](mailto:feedback@suse.cz)

## 30.12 Další dokumentace

Apache je velmi rozšířený webserver. Proto existuje mnoho dokumentace a mnoho webových stránek nabízí nápovědu a podporu.

### 30.12.1 Apache

Apache je dodáván s velmi obsáhlou dokumentací. Instalace dokumentace je popsána v části 30.5 na straně 481. Po instalaci můžete k dokumentaci přistupovat prostřednictvím svého prohlížeče na adrese <http://localhost/manual>. Nejnovější dokumentaci najdete na domovské stránce Apache <http://httpd.apache.org>.

### 30.12.2 CGI

Více informací CGI získáte z těchto stránek:

- <http://apache.perl.org/>

- <http://perl.apache.org/>
- <http://www.modperl.com/>
- <http://www.modperlcookbook.org/>
- <http://www.fastcgi.com/>
- <http://www.boutell.com/cgiic/>

### 30.12.3 Bezpečnost

Poslední opravy pro balíčky SUSE najdete na stránce <http://www.novell.com/linux/security/securitysupport.html>. Navštěvujte tuto adresu v pravidelných intervalech. Zde se také můžete přihlásit do e-mailové konference o bezpečnosti, v rámci které vám budou zasílána upozornění o bezpečnostních chybách a opravách.

Apache tým zcela otevřeně informuje o všech chybách. Oznamuje nejnověji objevené chyby a snaží se co nejdřív vydat příslušnou opravu na stránce [http://httpd.apache.org/security\\_report.html](http://httpd.apache.org/security_report.html). Pokud objevíte bezpečnostní chybu (předtím překontrolujte výše zmíněné stránky, zda již nebyla hlášena), pošlete nám prosím hlášení na email [feedback@suse.cz](mailto:feedback@suse.cz) nebo přímo (anglicky) na [security@apache.org](mailto:security@apache.org).

### 30.12.4 Další zdroje

V případě problémů navštivte Databázi instalační podpory na stránce <http://portal.suse.com/>. Novinky o webovém serveru Apache najdete na stránce <http://www.apacheweek.com/>.

Historie Apache je popsána v dokumentu [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html). Zde najdete i důvod pro pojmenování *Apache*.

Informace o aktualizaci z 1.3 na 2.0 najdete na stránce <http://httpd.apache.org/docs-2.0/en/upgrading.html>.

# Synchronizace souborů

Řada lidí používá více počítačů najednou jeden počítač doma, jeden nebo více počítačů v práci a laptop nebo PDA na cestách. Dříve či později budete potřebovat upravovat určitý soubor na všech počítačích, ale současně mít všude k dispozici aktuální verzi bez nutnosti ručního kopírování souborů.

31.1	Programy pro datovou synchronizaci . . . . .	500
31.2	Výběr vhodného programu . . . . .	502
31.3	Úvod do Unison . . . . .	505
31.4	Úvod do programu CVS . . . . .	507
31.5	Úvod do Subversion . . . . .	510
31.6	Úvod do rsync . . . . .	513
31.7	Úvod do mailsync . . . . .	514

## 31.1 Programy pro datovou synchronizaci

Pro počítače trvale připojené do rychlé sítě není synchronizace dat žádným problémem. V takovém případě je nejjednodušší cestou nasazení síťového souborového systému, jako je NFS, který umožňuje ukládat všechna data na serveru a přistupovat k nim z klientských stanic v síti. Toto řešení je však vyloučené v případě pomalejší nebo dočasné sítě. I na laptopu potřebujete lokální kopii všech důležitých souborů. Tehdy přichází na řadu synchronizace souborů. Ta zajistí, že pokud je soubor na jakémkoliv počítači změněn, dojde k aktualizaci souboru na všech ostatních počítačích. Automaticky lze synchronizaci provádět pomocí programů scp nebo rsync. Ne vždy je však tento způsob žádoucí, protože může dojít např. k přepsání novější verze starší.

### Varování

#### Riziko ztráty dat

Dřív než začnete používat systém k synchronizaci dat, seznamte se s funkcemi zvoleného programu a proveďte několik testů. U zvláště důležitých dat proveďte zálohu.

### Varování

Ruční synchronizace je vysoce časově náročná a náchylná k chybám. Tomu lze předejít automatizací. Zde vám některé z programů, které takovou automatizaci umožňují, krátce představíme. Pokud se pro některý z nich rozhodnete, nezapomeňte si pročíst jeho dokumentaci.

### 31.1.1 Unison

Unison není síťový souborový systém. Soubory jsou jednoduše ukládány a upravovány lokálně. Program Unison pak po ručním spuštění provede synchronizaci dat. Při první synchronizaci se na obou počítačích vytvoří databáze obsahující kontrolní součty, časová razítka a informace o přístupových právech jednotlivých zvolených souborů. Při dalším spuštění již program Unison rozpozná, které soubory se mají synchronizovat, a navrhne přenos na jiný počítač. Obvykle lze všechny návrhy akceptovat.

### 31.1.2 CVS

CVS je nejčastěji používán pro správu verzí zdrojových kódů programů. Nabízí možnost udržování kopie souborů na řadě počítačů. Použitelný je samozřejmě

také pro synchronizaci dat. CVS spravuje centrální sklad dat na serveru. Neukládají se jen samotné soubory, ale také záznamy o změnách. Změny se provádějí lokálně a odesílají se do centrálního skladu odkud mohou být stahovány ostatními uživateli. Odeslání i stažení změn vyžaduje aktivní účast uživatele.

CVS je odolný proti chybám, které nastanou v případě současného odesílání ze dvou různých počítačů. Všechny změny spojuje, ale pokud ke změnám dojde současně na jedné řádce, nahlásí konflikt. Databáze zůstává i v případě konfliktu v konzistentním stavu. Konflikty jsou viditelné a řešitelné pouze na klientských stanicích.

### 31.1.3 subversion

Na rozdíl od CVS, které se vyvinulo živelně, je subversion pečlivě navržený projekt, technicky zdokonalený následník CVS.

Program subversion byl zdokonalen v mnoha směrech. CVS umí z historických důvodů pracovat jen se soubory a nikoliv s adresáři, zatímco subversion udržuje i historii adresářů, které lze kopírovat a přejmenovávat stejně jako soubory. Ke každému adresáři i souboru lze navíc přiřadit metadata, pro která je taktéž udržována historie verzí. Na rozdíl od CVS podporuje subversion transparentní přístup přes speciální síťové protokoly, např. WebDAV (Web-based Distributed Authoring and Versioning). WebDAV rozšiřuje funkčnost HTTP protokolu o zápis do souborů na vzdálených webových serverech s možností spolupráce.

Při vývoji subversion byly využity již existující programy. Proto je společně se subversion vždy používán webserver apache a rozšíření WebDAV.

### 31.1.4 mailsync

Program mailsync se používá pouze k synchronizaci elektronické pošty ve schránkách na různých serverech. Synchronizovat lze jak lokální schránky, tak schránky IMAP.

Zprávy jsou synchronizovány či mazány v závislosti na ID zprávy obsaženém v hlavičce. Synchronizace je možná mezi jednotlivými schránkami nebo skupinami schránek.

### 31.1.5 rsync

Pokud není potřeba správa verzí, ale je potřeba synchronizovat rozsáhlé adresářové struktury přes pomalou síť, je vhodné použít nástroj rsync, který nabízí

dobrý mechanismus pro přenos změn v souborech, a to nejen textových, ale i binárních. Aby rsync zjistil změny v souborech, rozdělí je na jednotlivé bloky, ze kterých spočítá kontrolní součty.

Zjišťování změn je poměrně náročná činnost. Systémy, na kterých se má synchronizace provádět, by měly být náležitě vybaveny. Důležitý je zejména dostatek operační paměti.

## 31.2 Výběr vhodného programu

Při výběru vhodného programu byste měli zvážit následující hlediska:

### 31.2.1 Klient-Server vs. Peer-to-Peer

Pro distribuci dat se používají dva odlišné modely. V prvním modelu všichni klienti synchronizují data s centrálním serverem, který musí být alespoň čas od času pro klienty dostupný. Tento model používá subversion, CVS a WebDAV.

Druhou možností je synchronizace dat mezi klienty navzájem. Tak pracuje např. unison. Program rsync obvykle pracuje v klientském režimu, ale každý klient může fungovat i jako server.

### 31.2.2 Přenositelnost

CVS, subversion a unison jsou dostupné také ve verzích pro jiné operační systémy včetně Unixu a Windows.

### 31.2.3 Interaktivní vs. automatický

V programech subversion, CVS, WebDAV a unison synchronizaci spouští uživatelé ručně. Mají nad ní tak větší kontrolu. Pokud však uživatelé synchronizují v příliš dlouhých intervalech, zvyšuje se pravděpodobnost konfliktu.

### 31.2.4 Konflikty: výskyt a řešení

Konflikty jsou v CVS a subversion vzácné i v případě spolupráce velkého množství lidí na rozsáhlém projektu. Je to díky tomu, že změny v souborech jsou slučovány po jednotlivých řádcích. Když konflikt přeci jen nastane, je postižen pouze jeden klient. Konflikty se v CVS i subversion dají obvykle snadno řešit.

Unison oznamuje konflikty a umožňuje vyjmout postižené soubory ze synchronizace. Slučování změn je však obtížnější než v aplikacích subversion a CVS.

Na rozdíl od subversion či CVS, ve kterých lze přijmout změny v případě konfliktu alespoň částečně, přijme WebDAV změny pouze pokud je vše v pořádku.

Aplikace rsync se o konflikty vůbec nestará. Uživatel je zodpovědný za ruční řešení veškerých konfliktů a za to, aby omylem nepřepsal žádné soubory. Na druhou stranu lze dodatečně zapojit systém správy verzí, jako např. RCS.

### 31.2.5 Výběr a vkládání souborů

Ve standardní konfiguraci synchronizuje unison celý adresářový strom. Nové soubory přidáné do adresářového stromu jsou automaticky synchronizovány.

V subversion nebo CVS musí být nové soubory explicitně přidány příkazem `svnadd` či `cvsadd`. Znamená to větší uživatelskou kontrolu nad synchronizací, ale na druhou stranu se nové soubory často přehlédnou, zejména v případě, kdy je souborů mnoho a otazníky ve výstupu příkazů `svnupdate` a `svnstatus` nebo `cvsupdate` nejsou uživatelem zpozorovány.

### 31.2.6 Historie

Další funkcí subversion a CVS je možnost rekonstrukce starých verzí. Ke každé změně je možno doplnit krátkou poznámku. Vývoj všech souborů lze později snadno vysledovat na základě záznamů o změně obsahu a poznámek. To je neocenitelná pomoc zejména v případě vědeckých prací a zdrojových programových kódů.

### 31.2.7 Objem dat a požadavky na diskový prostor

Při synchronizaci je nutné mít na všech klientech dostatek místa pro data. V případě subversion a CVS budete navíc potřebovat místo na serveru pro repositář. Historie souborů je také uložena na serveru a vyžaduje další prostor. U textových souborů se ukládají pouze pozměněné řádky. Binární soubory se ukládají celé, pro uložení každé změny tedy vyžadují tolik místa, kolik zabírá celý soubor.

### 31.2.8 GUI

Unison nabízí pro zobrazení navrhovaného postupu synchronizace grafické uživatelské prostředí. Můžete v něm návrh přijmout či vyjmout jednotlivé soubory ze synchronizace. V textovém režimu lze interaktivně přijímat jednotlivé procedury.

Zkušení uživatelé obvykle pracují se subversion či CVS přes příkazovou řádku. Pro Linux však k těmto programům existují i grafická prostředí, jako např. cervisia. V jiných operačních systémech existují podobné programy, např. wincvs. Mnoho vývojářských nástrojů, jako např. kdevelop, a textových editorů, jako např. emacs, podporuje CVS či subversion. Řešení konfliktů je s těmito nástroji obvykle o poznání jednodušší.

### 31.2.9 Uživatelská přívětivost

Programy unison a rsync se používají poměrně snadno a jsou vhodné pro začátečníky. CVS a subversion jsou poněkud obtížnější. Vyžadují, aby uživatel pochopil vztah mezi repositářem a lokálně umístěnými daty. Změny by nejprve měly být sloučeny s repositářem lokálně pomocí příkazu `cvsupdate` nebo `svnupdate`. Pak musí být data odeslána zpět do repositáře příkazem `cvscscommit` nebo `svncommit`. Pokud uživatel pochopí tento princip, bude pro něj i použití CVS či subversion snadné.

### 31.2.10 Bezpečnost

Data by během přenosu měla být chráněna proti nedovolené manipulaci. Unison, subversion, CVS i rsync lze používat spolu s ssh (Secure Shell). Pokud chcete svým datům zajistit maximální bezpečnost, vyhněte se používání rsh (Remote Shell). V nedůvěryhodných nebo otevřených sítích nepoužívejte s CVS *pserver*. Program subversion použitý spolu se serverem apache již obsahuje bezpečnostní mechanismy.

### 31.2.11 Ochrana proti ztrátě dat

CVS je vývojáři používán velmi dlouho a je extrémně stabilní. Protože ukládá historii projektu, je CVS chráněn i proti chybám uživatelů jako je např. nechtěné smazání souboru. Ačkoliv není subversion tak rozšířená jako CVS, je již běžně nasazována do produkčního prostředí, například sama při svém vývoji.



Unison patří k novějším programům, ale vyznačuje se vysokou stabilitou. Je však mnohem citlivější na chyby uživatelů. Např. smazaný soubor nelze po synchronizaci obnovit.

**Tabulka 31.1:** Funkce synchronizačních nástrojů: -- = velmi nízká, - = nízká nebo žádná, o = střední, + = dobrá, ++ = výborná, x = dostupná

	unison	CVS/subv.	rsync	mailsync
Klient/server	rovnocenné	C-S/C-S	C-S	rovnocenné
Přenositelnost	Lin,Un*x,Win	Lin,Un*x,Win	Lin,Un*x,Win	Lin,Un*x
Interaktivita	x	x/x	x	-
Rychlost	-	o/+	+	+
Konflikty	o	++/++	o	+
výběr soub.	adresář	výběr/soub., adr.	adresář	mailbox
Historie	-	x/x	-	-
Místo na disku	o	--	o	+
GUI	+	o/o	-	-
Obtížnost	+	o/o	+	o
Útoky	+(ssh)	+/(ssh)	+(ssh)	+(SSL)
Ztráta dat	+	++/++	+	+

## 31.3 Úvod do Unison

Unison je vynikající řešení pro synchronizaci a přenos adresářového stromu. Synchronizace je prováděna v obou směrech a lze ji kontrolovat pomocí přehledného grafického rozhraní. V případě potřeby je k dispozici ovládání přes příkazovou řádku. Synchronizaci lze automatizovat tak, že není potřebný žádný zásah uživatele. Takové nastavení již vyžaduje určité zkušenosti.

### 31.3.1 Požadavky

Unison je nutné nainstalovat na server i na klienty. *Serverem* se zde rozumí vzdálený počítač (na rozdíl od CVS, viz 31.1.2 na straně 500).

V následujících příkladech je Unison používán spolu s ssh. ssh klient musí být nainstalován na klientovi a ssh server na serveru.

### 31.3.2 Používání Unison

Podstatou práce Unison je asociace dvou adresářů (*kořeny*, *roots*). Tato asociace je symbolická nejde o online spojení. V našem příkladu je asociace následující:

Klient:	/home/tux/dir1
Server:	/home/geeko/dir2

Synchronizovat se budou dva výše uvedené adresáře. Uživatel má na klientovi uživatelské jméno tux a na serveru geeko. Před zahájením práce je vhodné otestovat komunikaci klientserver příkazem:

```
unison -testserver /home/tux/dir1 ssh://geeko@server//homes/geeko/dir2
```

Problémy, které mohou nastat:

- Nekompatibilita verzí Unison na klientu a serveru.
- Server nepovoluje SSH připojení.
- Některá z uvedených cest neexistuje.

Pokud vše funguje, vynechejte volbu `-testserver`. Během první synchronizace Unison nezná vztahy mezi adresáři a navrhne směr přenosu jednotlivých souborů a adresářů. Šipka ve sloupci 'Action' indikuje směr přenosu. Otazník znamená, že Unison nedokáže určit směr přenosu, protože obě verze byly změněny nebo jsou nové.

Kurzorovými klávesami (šipkami) můžete nastavit směr přenosu jednotlivých položek. Pokud jsou nastaveny správné směry pro všechny položky, potvrďte nastavení kliknutím na 'Go'.

Vlastnosti Unison (například, zda má v jasných případech provést synchronizaci automaticky) lze nastavit při spuštění programu v příkazové řádce parametry. Seznam parametrů získáte příkazem: `unison--help`.

### Příklad 31.1: Soubor `~/unison/example.prefs`

```
root=/home/tux/dir1
root=ssh://wilber@server//homes/wilber/dir2
batch=true
```

Pro každou dvojici se vytváří záznam (log) v uživatelském adresáři `~/unison`. Konfigurace se také ukládá v tomto adresáři (např. `~/unison/example.prefs`). Při startu synchronizace zadejte na příkazovém řádku soubor s konfigurací jako parametr: `unisonexample.prefs`.

### 31.3.3 Další informace

Velmi užitečná je oficiální dokumentace Unison. Kompletní manuál najdete na stránce <http://www.cis.upenn.edu/~bcpierce/unison/> a v SUSE balíčku `unison`.

## 31.4 Úvod do programu CVS

CVS je velmi užitečný v případě časté editace textových souborů velkým počtem uživatelů. CVS lze použít i pro netextová data, ale za cenu velkých požadavků na prostor na serveru, protože budou ukládány všechny verze souborů celé. Navíc v takových případech není dostupná řada užitečných funkcí. Synchronizace pomocí CVS vyžaduje na rozdíl od Unison existenci jednoho centrálního serveru, ke kterému se mohou připojit všichni klienti.

### 31.4.1 Konfigurace CVS serveru

*Server* je místo, kde jsou uloženy všechny platné soubory včetně nejnovějších verzí. Jako server lze používat libovolnou pracovní stanici. Pokud je to možné, měli byste provádět pravidelné zálohování tohoto serveru.

Při konfiguraci serveru je vhodné nastavit přístup pro uživatele přes SSH. Pokud je uživatel serveru znám např. jako `tux` a CVS je nainstalován jak na klientovi, tak na serveru, je nutné na straně klienta nastavit následující proměnné prostředí:

```
CVS_RSH=ssh CVS_ROOT=tux@server:/serverdir
```

Příkazem `cvssinit` lze inicializovat CVS server ze strany klienta. Tento příkaz je třeba provést pouze jednou.

Nakonec musí být synchronizaci přiřazeno jméno. Na klientovi vytvořte adresář, který bude obsahovat soubory spravované pomocí CVS. Jméno adresáře bude také jméno synchronizace. V našem případě používáme adresář pojmenovaný `synchome`. Jméno synchronizace nastavíme v tomto adresáři příkazem:

```
cvss import synchome tux novak
```

Řada CVS příkazů vyžaduje komentář. Pro tento účel CVS spouští editor (definovaný proměnnou prostředí `$EDITOR` nebo `vi`, pokud jste žádný editor nenastavili). V editoru můžete doplnit komentář jako v následujícím příkladě:

```
cvss import -m 'toto je test' synchome tux novak
```

## 31.4.2 Používání CVS

Od tohoto okamžiku lze k repositáři přistupovat ze všech klientů a stahovat jeho obsah pomocí příkazu `cvssco synchome`. Voláním tohoto příkazu se vytvoří na klientském počítači podadresář `synchome`. Změny provedené v tomto adresáři (tento adresář nebo některý z jeho podadresářů musí být aktuálním adresářem) odešlete do repositáře příkazem `cvsscommit`.

Implicitně jsou na server zasílány všechny soubory včetně podadresářů. Chcete-li zaslat pouze jednotlivé soubory nebo adresáře, určete je příkazem `cvsscommit soubor1 adresar1`. Nové soubory a adresáře musí být do repositáře vloženy příkazem `cvssadd soubor1 adresar1` dříve, než jsou zaslány na server příkazem `cvsscommit soubor1 adresar1`.

Pokud přejdete k jiné pracovní stanici, proveďte checkout synchronizačního repositáře, pokud jste tak neučinili na této stanici již dříve (viz výše).

Synchronizaci se serverem zahájíte příkazem `cvssupdate`. Jednotlivé soubory a adresáře synchronizujete příkazem `cvssupdate soubor1 adresar1`. Rozdíly mezi aktuálními lokálními soubory a soubory na serveru získáte příkazem `cvssdiff` nebo `cvssdiff soubor1 adresar1`. Příkaz `cvss-nq update` použijte, pokud chcete zjistit, jaké soubory budou synchronizací ovlivněny.

Během synchronizace jsou používány následující stavové symboly:

- U** Lokální verze byla aktualizována verzí ze serveru. To se týká všech souborů, které jsou na serveru, ale na lokálním systému chyběly.

- M** Lokální verze souboru obsahuje oproti serveru změny. Pokud byly změny i na serveru, bylo je možné sloučit s lokálními změnami. Nedošlo ke konfliktu.
- P** Byla aktualizována lokální verze. Nepřenesl se celý soubor, ale byl použit tzv. patch (záplata).
- C** Lokální verze je v konfliktu s verzí na serveru.
- ?** Soubor v CVS repositáři neexistuje.

Stav označený písmenem **M** upozorňuje na lokálně změněný soubor. Bud' nahrajte lokální soubor na server nebo lokální soubor odstraňte a proveďte znovu update chybějící soubor bude nahrán ze serveru. Pokud budete nahrávat lokálně změněný soubor, který byl mezitím změněn ve stejné řádce i na serveru, může dojít ke konfliktu označenému písmenem **C**.

V takovém případě v souboru vyhledejte konfliktní značky a rozhodněte se mezi verzemi. Je to poměrně nepříjemná práce, takže někdy může být lepší rezignovat na své změny, lokální soubor smazat a pomocí příkazu `cvsup` nahrát aktuální verzi ze serveru.

### 31.4.3 Další informace

Zde jsme vám poskytli pouze krátký úvod do možností CVS. Rozsáhlou dokumentaci naleznete na následujících adresách:

<http://www.cvshome.org/>  
<http://www.gnu.org/manual/>

## 31.5 Úvod do Subversion

Subversion je svobodný opensource systém pro správu verzí, který je často považován za nástupce staršího systému CVS. To znamená, že funkce známé z CVS jsou běžně dostupné i v subversion, avšak bez nutnosti potýkat se s omezeními a nevýhodami CVS. O některých vlastnostech jsme psali již v kapitole 31.1.3 na straně 501.

### 31.5.1 Instalace Subversion serveru

Instalace skladovací databáze na serveru je poměrně snadná. Subversion k tomuto účelu nabízí speciální administrační nástroj. Chcete-li vytvořit nový repositář (skladovací databázi), použijte příkaz:

```
svnadmin create /cesta/k/repositari
```

Další možnosti lze zjistit pomocí příkazu `svnadminhelp`. Na rozdíl od CVS není subversion založená na RCS, nýbrž na Berkeley databázi. Proto se ujistěte, že repositář neinstalujete na vzdálené souborové systémy (např. NFS, AFS, Windows SMB). Databáze totiž vyžaduje POSIX kompatibilní zamykací mechanismy, které nejsou na těchto souborových systémech podporovány.

Příkaz `svnlook` poskytuje informace o stávajícím repositáři.

```
svnlook info /cesta/k/repositari
```

Server musí být nastaven tak, aby umožnil uživatelům přístup k repositáři. Použijte k tomu buď Apache webserver s WebDAV nebo `svnserve`, což je server dodávaný spolu se subversion. Jakmile je `svnserve` spuštěn, je repositář přístupný na příslušné URL přes protokol `svn://` nebo `svn+ssh://`. Uživatelé, kteří se musejí při použití `svn` autentizovat, lze nastavit v souboru `/etc/svnserve.conf`.

Výběr mezi servery Apache a `svnserve` záleží na mnoha faktorech. Doporučujeme proto nastudovat si příručku k subversion. Více se o ní dozvíte v části 31.5.3 na straně 512.

## 31.5.2 Použití a provoz

K přístupu do repozitáře použijte příkaz `svn` (podobně jako příkaz `cvcs`). Obsah poskytovaný správně nastaveným serverem s odpovídajícím repozitářem je přístupný jakýmkoliv klientem jedním z následujících příkazů:

```
svn list http://svn.example.com/cesta/k/projektu
```

nebo

```
svn list svn://svn.example.com/cesta/k/projektu
```

Uložit existující projekt do aktuálního adresáře (`check out`) lze příkazem `svncheckout`:

```
svn checkout http://svn.example.com/cesta/k/projektu jmenoprojektu
```

Checkout vytvoří na klientovi nový podadresář `jmenoprojektu`. V něm lze následně provádět operace se soubory (přidávání, kopírování, přejmenovávání, mazání):

```
svn add soubor
svn copy starysoubor novysoubor
svn move starysoubor novysoubor
svn delete soubor
```

Tyto příkazy lze rovněž použít na adresáře. Program `subversion` navíc umí zaznamenat vlastnosti souboru či adresáře:

```
svn propset license GPL foo.txt
```

Předchozí příklad nastaví hodnotu `GPL` vlastnosti `license`. Vlastnosti lze zobrazit příkazem `svnproplist`:

```
svn proplist --verbose foo.txt
Properties on 'foo.txt':
license : GPL
```

Změny lze na server uložit příkazem `svncommit`. Ostatní uživatelé se mohou synchronizovat příkazem `svnupdate`.

Na rozdíl od CVS lze stav pracovního adresáře zobrazit bez přístupu k repozitáři pomocí `svnstatus`. Lokální změny jsou zobrazeny v pěti sloupcích, z nichž nejdůležitější je první:

- " Žádné změny.
- 'A' Objekt bude přidán.
- 'D' Objekt bude smazán.
- 'M' Objekt byl změněn.
- 'C' Objekt je v konfliktu.
- 'I' Objekt byl ignorován.
- '?' Objekt není verzovacím systémem spravován.
- !' Objekt chybí. Tento příznak značí, že byl objekt smazán či přesunut bez použití příslušného příkazu `svn`.
- '~' Objekt je spravován jako soubor, ale byl nahrazen adresářem, nebo naopak.

Druhý sloupec zobrazuje stav vlastností. Význam všech sloupců je popsán v příručce k subversion.

Příkaz `svnhelp` použijte, pokud chcete získat popis parametrů příkazu:

```
svn help proplist
proplist (plist, pl): List all properties on files, dirs, or revisions.
usage: 1. proplist [PATH...]
       2. proplist --revprop -r REV [URL]

1. Lists versioned props in working copy.
2. Lists unversioned remote props on repos revision.
```

### 31.5.3 Další informace

Prvním místem, kde hledat další informace, je domovská stránka projektu subversion na adrese <http://subversion.tigris.org/>. Velmi doporučujeme také příručku, která je dostupná online na adrese <http://svnbook.red-bean.com/svnbook/index.html> nebo po instalaci balíčku `subversion-doc` v souboru `file:///usr/share/doc/packages/subversion/html/book.html`.



## 31.6 Úvod do rsync

Program rsync je užitečný, pokud je potřeba pravidelně přenášet velké množství dat, která se příliš nemění. To je často případ záloh nebo staging serverů. Tyto servery obsahují kompletní adresářové stromy webserverů, které jsou pravidelně zrcadleny na webserver v demilitarizované zóně.

### 31.6.1 Konfigurace a provoz

Program rsync lze provozovat ve dvou různých režimech. Může být používán k archivování nebo kopírování dat. K tomu je na cílovém systému potřeba pouze vzdálený interpret příkazů, např. ssh. Program rsync lze ale používat také jako démon, který poskytuje adresáře na síti.

Základní provozní režim rsync nevyžaduje žádné zvláštní nastavení. Program rsync umožňuje přímo zrcadlit celé adresáře na jiný systém. Následující příkaz například vytvoří zálohu domovského adresáře uživatele tux na záložním serveru slunce:

```
rsync -baz -e ssh /home/tux/ tux@slunce:backup
```

A tímto příkazem se adresář nahraje zpět:

```
rsync -az -e ssh tux@slunce:backup /home/tux/
```

Použití se příliš neliší od běžného kopírovacího nástroje, jako např. scp.

Program rsync by ale měl být používán v režimu *rsync*, který umožňuje plně využívat všechny jeho funkce. Lze tak učinit spuštěním démona rsyncd na jednom ze systémů. Démon se konfiguruje v souboru `/etc/rsyncd.conf`. Pokud například chcete aby byl adresář `/srv/ftp` dostupný přes rsync, použijte následující konfiguraci:

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

```
[FTP]
    path = /srv/ftp
    comment = An Example
```

Po provedení konfigurace spusťte `rsyncd` příkazem `rcrsyncdstart`. Program `rsyncd` může být spouštěn i automaticky během startu systému. To nastavíte v editoru úrovní běhu pomocí nástroje YaST nebo ručně příkazem `insservrsyncd`. Program `rsyncd` může být také spuštěn pomocí `xinetd`, je to však doporučeno jen na serverech, které `rsyncd` používají jen výjimečně.

Konfigurace v použitém příkladu rovněž vytváří protokolový soubor `/var/log/rsyncd.log`, ve kterém jsou zaznamenávána všechna spojení.

Přenos z klientského systému lze otestovat příkazem:

```
rsync -avz slunce::FTP
```

Tento příkaz vypíše všechny soubory v adresáři `/srv/ftp` na serveru. Požadavek je zaznamenán v souboru `/var/log/rsyncd.log`. Pro zahájení skutečného přenosu specifikujte cílový adresář. Aktuální adresář zapište jako `..`. Například:

```
rsync -avz slunce::FTP .
```

Implicitně se při synchronizaci pomocí `rsync` nemazou žádné soubory. Pokud si chcete smazání souborů vynutit, musíte použít parametr `--delete`. Pokud si chcete být jistí, že nebudou smazány žádné novější soubory, použijte parametr `--update`. Veškeré konflikty je nutné řešit manuálně.

## 31.6.2 Další informace

Důležité informace o `rsync` naleznete v manuálových stránkách (`manrsync` a `manrsyncd.conf`). Technický popis funkce `rsync` naleznete v souboru `/usr/share/doc/packages/rsync/tech_report.ps`. Novinky o `rsync` najdete na webové stránce projektu na adrese <http://rsync.samba.org/>.

## 31.7 Úvod do mailsync

Program `mailsync` se používá zejména pro tři úlohy:

- Synchronizace lokálně uložených poštovních zpráv se zprávami uloženými na serveru.
- Přenos schránek na jiný server nebo převod do jiného formátu.
- Kontrola integrity schránky a vyhledávání duplikátů.

### 31.7.1 Konfigurace a použití

mailsync rozlišuje mezi samotnými schránkami (store) a kanály mezi schránkami (channel). Definice schránek a kanálů jsou uloženy v `~/.mailsync`. Následující odstavce vysvětlují použití schránek (store) na několika příkladech.

Jednoduchá definice může vypadat takto:

```
store saved-messages {  
    pat      Mail/saved-messages  
    prefix   Mail/  
}
```

Mail/ je podadresář v domovském adresáři uživatele, který obsahuje zprávy včetně složky saved-messages. Pokud program mailsync spustíte příkazem mailsync-m saved-messages, vypíše seznam zpráv ve složce saved-messages.

Při nastavení:

```
store localdir {  
    pat      Mail/*  
    prefix   Mail/  
}
```

vypíše příkaz mailsync-m localdir všechny zprávy ve složce Mail/. Příkaz mailsynclocaldir naopak vypíše jména složek.

Příklad specifikace pro IMAP server:

```
store imapinbox {  
    server {mail.edu.harvard.com/user=gulliver}  
    ref    {mail.edu.harvard.com}  
    pat    INBOX  
}
```

Uvedený příklad specifikuje pouze hlavní složku na IMAP serveru. Pro pod-složky bude vypadat takto:

```
store imapdir {
server {mail.edu.harvard.com/user=gulliver}
ref {mail.edu.harvard.com}
pat INBOX.*
prefix INBOX.
}
```

Pokud IMAP server podporuje šifrované připojení, měla by jeho specifikace vypadat takto:

```
server {mail.edu.harvard.com/ssl/user=gulliver}
```

nebo, pokud je certifikát neznámý:

```
server {mail.edu.harvard.com/ssl/novalidate-cert/user=gulliver}
```

Nyní je možné složky v Mail/ připojit k podadresářům na IMAP serveru:

```
channel folder localdir imapdir {
msinfo .mailsync.info
}
```

Program mailsync používá soubor `msinfo` k zaznamenávání již synchronizovaných zpráv.

Příkaz `mailsyncfolder` provede následující:

- Expanduje schéma schránky na obě strany.
- Ze získaných jmen složek odstraní předponu.
- V párech synchronizuje složky (pokud neexistují, vytvoří je).

Složka `INBOX.sent-mail` na IMAP serveru je synchronizována s lokální složkou `Mail/sent-mail` (pokud existují definice uvedené výše). Synchronizace mezi jednotlivými složkami se provádí následovně:

- Pokud zpráva existuje na obou stranách, nic se neděje.
- Pokud zpráva existuje jen na jedné straně a je nová (není uvedena v souboru `msinfo`), je přenesena.

- Pokud zpráva existuje jen na jedné straně a je stará (je již uvedena v souboru `msinfo`), je smazána (neboť byla očividně na jedné straně úmyslně smazána).

Pokud chcete s předstihem vědět, které zprávy budou během synchronizace přeneseny a které smazány, spusťte `mailsync` pomocí `mailsyncfolder localdir`. Tímto příkazem získáte seznam všech zpráv, které jsou na lokálním počítači nové, a seznam všech zpráv, které budou na IMAP serveru během synchronizace smazány. Podobně příkazem `mailsyncfolder imapdir` získáte seznam všech zpráv, které jsou nové na straně IMAP serveru, a zpráv, které budou během synchronizace smazány na lokálním počítači.

### 31.7.2 Možné problémy

V případě ztráty dat je nejbezpečnější metodou smazat příslušný soubor se záznamy `msinfo`. Tak budou všechny soubory existující na jedné straně považovány za nové a přeneseny během další synchronizace.

Synchronizace zahrnuje pouze zprávy s ID. Zprávy, které ID nemají, jsou ignorovány, tzn. nejsou ani přenášeny ani mazány. Chybějící ID je většinou důsledkem chyby programu při vytváření nebo odesílání zprávy.

Na některých IMAP serverech je hlavní složka adresována pomocí `INBOX` a podsložky pomocí náhodně zvoleného jména (na rozdíl od `INBOX` a `INBOX.jmeno`). Proto pro takové IMAP servery nelze nastavit vzorec jen pro podsložky.

Po úspěšném přenosu zpráv na IMAP server nastaví ovladače schránky (c-client) používané programem `mailsync` zvláštní příznak. Z tohoto důvodu nejsou některé programy, jako např. `mutt`, schopny rozpoznat tyto zprávy jako nové. Nastavení tohoto příkazu lze zakázat volbou `-n`.

### 31.7.3 Další informace

Další informace najdete po instalaci balíčku `mailsync` v souboru `README` v adresáři `/usr/share/doc/packages/mailsync/`. V této souvislosti věnujte také pozornost RFC 2076 *Common Internet Message Headers*.



# Samba

Pomocí balíku Samba lze doplnit libovolný unixový počítač o funkce výkonného souborového a tiskového serveru pro DOS, OS/2 a Windows počítače. Postupem doby se Samba vyvinula ve složitý a komplexní produkt. V této kapitole najdete popis základního nastavení Samby a konfigurace pomocí modulu programu YaST.

Podrobné informace jsou dostupné v digitální podobě. Příkazem `apropos samba` zobrazíte dostupné manuálové stránky. Pokud je Samba nainstalována, najdete další dokumentaci a příklady v adresáři `/usr/share/doc/packages/samba`. V podadresáři `examples` najdete okomentovaný příklad konfigurace (`smb.conf.SuSE`).

Balíček `samba` verze 3 obsahuje řadu novinek a zlepšení, z nichž nejvýznamnější jsou:

- Podpora Active Directory.
- Výrazně vylepšená podpora Unicode.
- Přepracovaný interní autentizační mechanismus.
- Vylepšená podpora tiskového systému pro Windows 200x/XP.
- Možnost nastavení jako serveru domény Active-Directory.
- Možnost migrace z NT4 domény na Samba doménu.

---

## Tip

### Migrace na Sambu verze 3

Pokud chcete migrovat ze Samby 2.x na Sambu 3, musíte být maximálně opatrní. Aby nedošlo k chybě, věnujte prosím pozornost dokumentu *Samba-HOWTO-Collection*. Najdete ho po instalaci balíčku `samba-doc` v souboru `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.

---

## Tip

Samba používá SMB protokol (server message block) založený na službách Net-BIOSu. Díky tlaku společnosti IBM Microsoft tento protokol uveřejnil, a tak je možné připojit se do domén sítě Microsoft. Protože Samba pracuje na základě TCP/IP protokolu, musí být tento protokol nainstalován na všech klientech.

NetBIOS je softwarové rozhraní (API) pro komunikaci mezi počítači poskytující tzv. *name service* umožňujícím počítačům připojeným k síti rezervovat si pro sebe jména, sloužící k oboustranné identifikaci. Pro přidělování nebo kontrolu jmen zde není žádná centrální autorita. Každý počítač v síti smí mít libovolný počet jmen, pokud se tato jména již nepoužívají jiným počítačem. Rozhraní NetBIOS lze implementovat v různých síťových architekturách. Jedna z implementací, která je těsně svázána se síťovým hardwarem, se nazývá NetBEUI (bývá však často zaměňována za NetBIOS). Síťové protokoly implementované v NetBIOSu pocházejí z IPX od společnosti Novell (NetBIOS via TCP/IP) a TCP/IP.

Všechny běžné operační systémy, jako Mac OS X, Windows nebo OS/2, podporují protokol SMB. Na všech počítačích musí být nainstalovaný TCP/IP protokol. Samba poskytuje klienta pro různé UNIXové systémy. Pro Linux existuje jaderný modul umožňující integraci SMB zdrojů na systémové úrovni.

SMB servery poskytují hardwarové místo klientům ve formě sdílení (shares). Sdílení zahrnuje adresář na serveru včetně podadresářů. Je exportováno pod zadaným jménem. Jako jméno sdílení lze nastavit jakékoliv jméno, nemusí to být jméno sdíleného adresáře. Tiskárna má také přiděleno jméno. Klienti pak k tiskárně přes její jméno přistupují.

## 32.1 Nastavení serveru

Nejdříve je třeba nainstalovat balíček `samba`. Ručně pak můžete spustit službu příkazem `rcnmb start && rcsmb start` a pomocí `rcsmb stop && rcnmb stop` ji opět ukončit.



Hlavní konfigurační soubor Samby je `/etc/samba/smb.conf`. Skládá se ze dvou logických částí. V části `[global]` jsou obecná a centrální nastavení. V části `[share]` se nastavují individuální sdílení souborů a tiskáren. Rozdělení mezi tyto dvě sekce zvyšuje přehlednost konfiguračního souboru.

### 32.1.1 Sekce (global)

Aby ostatní počítače s Windows mohly přistupovat prostřednictvím SMB k vašemu Samba serveru, vyžadují následující parametry ze sekce `[global]` určité úpravy v závislosti na nastavení sítě.

**workgroup = TUX-NET** Samba serveru je pomocí této řádky přiřazena pracovní skupina. TUX-NET nahraďte správným jménem skupiny ve vašem síťovém prostředí. Samba server se objeví pod svým DNS jménem, pokud ovšem není používáno jiným strojem v síti. Pokud DNS jméno není dostupné, nastavte jméno serveru pomocí `netbiosname=MYNAME` (viz `mansmb.conf`).

**os level = 2** Podle tohoto parametru se bude Samba server rozhodovat, zda se stane LMB (*Local Master Browser*) pro svou pracovní skupinu. Nízká hodnota zajistí, že existující windowsová síť nebude rušena špatně nakonfigurovanou Sambou. Bližší informace k této volbě naleznete v souborech `BROWSING.txt` a `BROWSING-Config.txt`, které najdete v podadresáři `textdocs` dokumentace balíku.

Pokud ještě neprovozujete SMB server (např. ve Windows NT, 2000, XP) a sambový server by měl v lokální síti udržovat informace o jménech dostupných systémů, zvýšte `os level` na vyšší hodnotu (např. 65). Váš Samba server se tak stane LMB.

Při změnách této hodnoty byste měli být obzvláště opatrní, protože můžete rušit komunikaci ve stávající síti. Nejprve si nastavení otestujte v izolované síti nebo o víkendu.

**wins support a wins server** Pokud chcete integrovat Sambu do windowsové sítě, kde již běží WINS server, tak položku `wins server` odkomentujte a uveďte IP adresu WINS serveru.

Pokud jsou windowsové systémy provozovány v oddělených podsítích a měly by se přesto vidět, potřebujete WINS server. Sambu proměníte na takový WINS server nastavením volby `wins support = yes`. Pozor na to, abyste tuto položku aktivovali pouze na jednom serveru. Volby `wins server` a `wins support` nesmí být v souboru `smb.conf` nikdy povoleny současně.

### 32.1.2 Sdílení

V následujících příkladech si ukážeme, jak sdílet CD mechaniku a domovské adresáře uživatelů.

**[cdrom]** Aby nedošlo ke zneužití CD mechaniky, je ve výchozím nastavení deaktivována pomocí komentáře (zde středník). Odstraněním středníků v prvním sloupci můžete CD-ROM sdílet.

#### *Příklad 32.1: Sdílení CD-ROM*

```
[cdrom]
;      comment = Linux CD-ROM
;      path = /media/cdrom
;      locking = No
```

**[cdrom] a comment** Položka `[cdrom]` je jméno, které bude vidět na SMB klientech. Pomocí `comment` můžete sdílení podrobněji popsat.

**path = /media/cdrom** Exportuje adresář `/media/cdrom`.

Vzhledem k velmi přísné implicitní konfiguraci je tento způsob exportování omezen na lokální uživatele. Ostatním umožníte přístup volbou `guest ok = yes`. Protože tato volba umožňuje přístup ke čtení všem, je potřeba s ní zacházet velice opatrně. Hlavně při jejím používání v sekci `[global]`.

**[homes]** Zvláštní postavení má export domovských adresářů. Pokud má uživatel na linuxovém souborovém serveru platný účet a vlastní domovský adresář, pak se může jeho klient po zadání platného uživatelského jména a hesla připojit

#### *Příklad 32.2: Sdílení domovských adresářů*

```
[homes]
      comment = Home Directories
      valid users = %S
      browseable = No
      read only = No
      create mask = 0640
      directory mask = 0750
```

**[homes]** Při připojení uživatele k SMB serveru je automaticky vytvořeno sdílení pomocí direktivy `[homes]`. Výsledné jméno sdílení je shodné s uživatelským jménem a vytvoří se pouze, pokud již neexistuje sdílení se stejným jménem.

**valid users = %S** %S je po úspěšném spojení nahrazen konkrétním jménem sdílení. V případě sdílení `[homes]` je to vždy jméno uživatele. Důsledkem je omezení používání home pouze na jeho vlastníka.

**browseable = No** Toto nastavení činí sdílení neviditelným v síťovém prostředí.

**read only = No** Samba má přenastaven zápis u exportovaných dat na `read only = Yes`. Pokud má být adresář přístupný pro zápis, pak je třeba nastavit `read only = No`, což je totéž jako `writeable = Yes`.

**create mask = 0640** Systémy nezaložené na MS Windows NT nedokáží pracovat s UNIXovými přístupovými právy a tím pádem ani nastavit tato práva při vytváření souborů. Parametr `create mask` nastavuje přístupová práva všech nově vytvořených souborů. Toto nastavení se týká pouze těch sdílení, do kterých mají uživatelé právo zápisu. Výše uvedená hodnota nastavuje právo pro čtení a zápis vlastníka souboru a práva pro čtení pro všechny uživatele z vlastníkovy skupiny. Nastavením `valid users = %S` zamezíte ostatním členům skupiny přístupu ke čtení i v případě, že to práva povolují. Aby měla celá skupina práva ke čtení či zápisu, je nutné řádku `valid users = %S` zakomentovat.

### 32.1.3 Bezpečnostní úrovně

SMB protokol vychází z prostředí DOS/Windows a bere ohledy na problematiku bezpečnosti. Proto je možné přístup ke každému exportovanému adresáři ochránit heslem. SMB rozlišuje tři různé způsoby:

#### Share Level Security (`security = share`):

Heslo je stejné pro všechny uživatele, je vázáno na sdílení. Každý, kdo toto heslo zná, má ke sdílení přístup.

**User Level Security (`security = user`):** Každý uživatel má vlastní heslo. Po registraci server přiděluje uživateli přístup jen k tomu povoleným sdílením.

### Server Level Security (security = server):

Samba před klienty předstírá práci v uživatelském režimu. Nicméně předává všechny hesla k ověření jinému serveru v uživatelském režimu. Toto nastavení vyžaduje další parametr (`password server=`).

Uvedená nastavení jsou aplikována na celý server. Není možné nastavit individuální sdílení s různými bezpečnostními stupni. Můžete však pro každou IP adresu nastavenou na systému spustit vlastní Samba server.

Více informací o této problematice najdete v Samba HOWTO Collection. U vícenásobného serveru na jednom počítači věnujte pozornost volbám `interfaces` a `bind interfaces only`.

#### Tip

Pro jednoduchou správu Samba serverů existuje program `swat`. Ten používá pro konfiguraci Samba serveru jednoduché webové rozhraní. Po spuštění prohlížeče ho najdete na adrese `http://localhost:901`, kde se přihlaste jako uživatel `root`. Nezapomeňte, že `swat` je také potřeba aktivovat v souborech `/etc/xinetd.d/samba` a `/etc/services`. K tomu musíte v souboru `/etc/xinetd.d/samba` nastavit parametr `disable` na hodnotu `no` (`disable = no`). Další informace o `swat` najdete v jeho manuálové stránce.

#### Tip

## 32.2 Samba jako přihlašovací server

V sítích, kde je převaha windowsových klientů, je často žádoucí, aby se směl uživatel přihlásit pouze s platným účtem a heslem. Toto je možné zajistit pomocí Samba serveru. V čistě windowsové síti je to úloha NT serveru, který je konfigurován jako Primary Domain Controller (PDC). Proto je třeba provést změny v obecné *globals* části konfiguračního souboru `smb.conf` uvedené v příkladu 32.3 na této straně.

### *Příklad 32.3: Globální sekce smb.conf*

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

Pokud se pro verifikaci používají šifrovaná hesla, musí si s tím Samba umět poradit. To umožňuje položka `encrypt passwords = yes` v části `[globals]` (v Sambě 3 je to výchozí nastavení). Kromě toho je třeba převést uživatelské účty a hesla do šifrovaného formátu vhodného pro Windows. To provedete příkazem `smbpasswd -a name`. Protože v doménové koncepci Windows NT potřebují i samotné počítače doménový účet, vytvořte ho následujícími příkazy:

***Příklad 32.4: Nastavení účtu počítače***

```
useradd hostname\$\n\nsmbpasswd -a -m hostname
```

Příkazem `useradd` je přidán znak dolaru. Příkaz `smbpasswd` ho vkládá automaticky, pokud je použit parametr `-m`. Komentovanou ukázkovou konfiguraci včetně automatizace výše uvedených činností najdete v souboru `/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`.

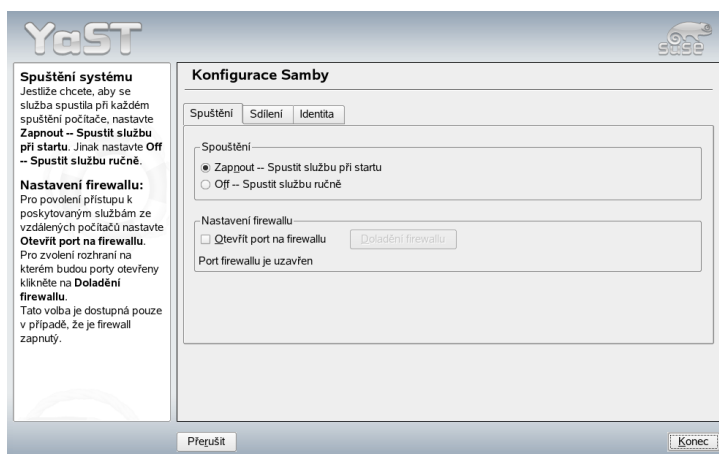
***Příklad 32.5: Automatizované nastavení účtu počítače***

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \n\n-s /bin/false %m\$\n\n
```

Aby mohla Samba tento skript vykonat, zvolte Samba uživatele s požadovanými administrátorskými právy. Vyberte jednoho uživatele a přidejte ho do skupiny `ntadmin`. Pak můžete všechny uživatele patřící do této linuxové skupiny obdařit statutem `Domain Admins` pomocí příkazu:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

Více informací naleznete ve dvanácté kapitole `Samba-HOWTO-Collection` v souboru `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.



Obrázek 32.1: Konfigurace Samby start

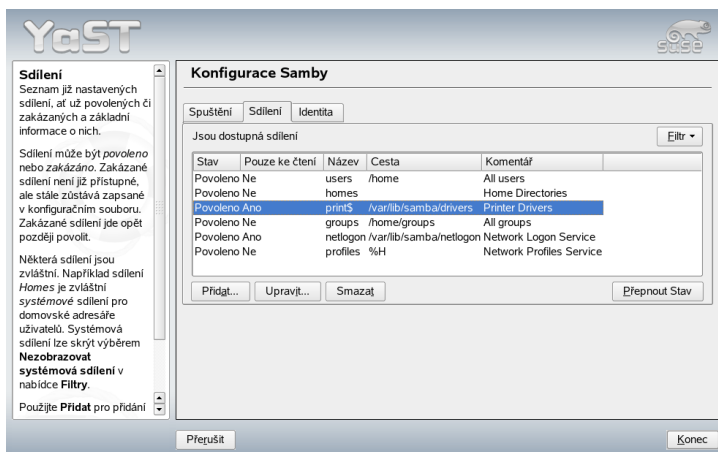
## 32.3 Konfigurace Samba serveru pomocí programu YaST

Na začátku nastavení Samba serveru zvolte doménu nebo pracovní skupinu, kterou bude server spravovat. V položce 'Pracovní skupina nebo jméno domény' můžete zadat existující nebo zcela novou doménu či skupinu. V dalším kroku nastavte, zda má server plnit úlohu PDC (Primary Domain Controller) nebo BDC (Backup Domain Controller).

Na kartě 'Spuštění' spusťte Sambu (viz 32.1 na této straně) a v části 'Nastavení firewallu' aktivujte 'Otevřít port na firewallu'. Na všech rozhraních tak dojde k otevření portů pro služby `netbios-ns`, `netbios-dgm`, `netbios-ssn` a `microsoft-ds`. Pokud potřebujete upřesnit nastavení, klikněte na tlačítko 'Doladění firewallu'.

Na kartě 'Sdílení' (viz 32.2 na následující straně) nastavte sdílení Samby. U jednotlivých položek lze tlačítkem 'Změnit stav' přepínat mezi stavem 'Zakázáno' a 'Povoleno'. Nové sdílení zadáte kliknutím na 'Přidat'.

Na kartě 'Identita' (viz 32.3 na straně 528) lze nastavit doménu počítače ('Základní nastavení') a jméno v SMB síti ('NetBIOS jméno počítače').



Obrázek 32.2: Konfigurace Samby sdílení

## 32.4 Nastavení klienta

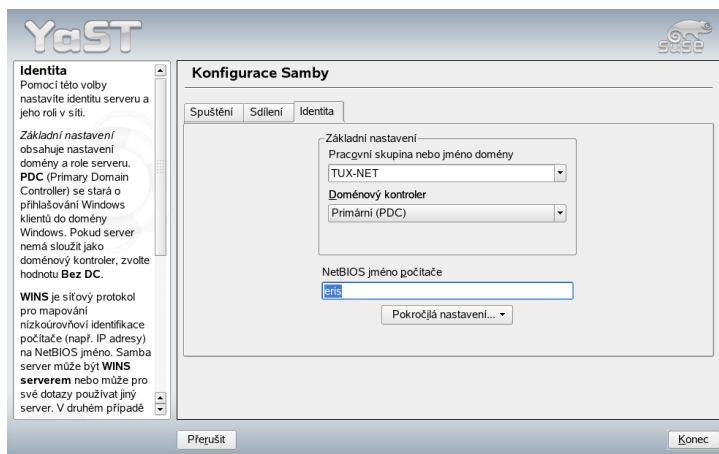
Upozorňujeme, že server Samba je dosažitelný pro klienta pouze prostřednictvím protokolu TCP/IP. NetBEUI ani IPX nejsou pro Sambu v současnosti použitelné.

### 32.4.1 Nastavení Samba klienta pomocí YaST

Samba klienta nastavíte pro přístup ke zdrojům Samba serveru (soubory nebo tiskárny) následovně. V dialogu 'Pracovní skupina' zadejte doménu nebo pracovní skupinu. Všechny dostupné domény a skupiny zjistíte kliknutím na tlačítko 'Procházet'. Skupinu vyberete označením myši. Pokud zvolíte 'Použít SMB informace také pro autentizaci v Linuxu', budou uživatelé ověřováni přes Samba server. Nastavení aktivujete kliknutím na tlačítko 'Konec'.

### 32.4.2 Windows 9x a ME

Windows 9x a ME již sice podporu TCP/IP obsahují, avšak dosud nikoli jako výchozí nastavení. Proto pro přidání protokolu TCP/IP klikněte na 'Ovládací panel', dále 'Systém' a vyberte 'Přidat', 'Protokoly', z nich vyberte 'Microsoft' →



*Obrázek 32.3: Konfigurace Samby identita*

‘TCP/IP’. Po restartu počítače s Windows najdete Samba server dvojitým poklepáním na ikonu ‘Síť’ na pracovní ploše Windows.

Abyste mohli použít tiskárnu na Samba serveru, stačí nainstalovat standardní ovladač tiskárny (popřípadě ovladač Apple-PostScript) pro odpovídající verzi Windows. Nejlepší je provázat ho s tiskovou frontou, která přijímá úlohy ve formátu PostScript.

## 32.5 Optimalizace

Optimalizaci nabízí `socket options`. Přednastavení, která jsou součástí příkladové konfigurace se zaměřují především na lokální ethernetovou síť. Další podrobnosti naleznete v příslušné části manuálových stránek `smb.conf` a v manuálové stránce `socket (7)`. Další informace naleznete v `SambaHOWTOCollection` v kapitole věnované ladění výkonu.

Standardní konfigurace v `/etc/samba/smb.conf` není samozřejmě vhodná pro všechny sítě a způsoby nasazení, proto je třeba ji ještě upravit podle místních podmínek. Protože je ale tato optimalizace závislá na mnoha faktorech, neexistuje žádné univerzální řešení. Komentovaný příklad konfiguračního souboru



`examples/smb.conf`. SuSE obsahuje užitečné informace pro přizpůsobení místním podmínkám.

Samba HOWTO Collection obsahuje návod pro řešení nejčastějších problémů. V části V (Part V) pak najdete podrobný návod, který vás krok za krokem provede kontrolou konfigurace.



# Proxy server Squid

Squid je na linuxových/unixových platformách nejrozšířenější proxy cache. Zde si popíšeme, jak ho konfigurovat, řekneme si, jaké má systémové požadavky a mnoho dalšího. Stranou nezůstane ani konfigurace transparentní proxy, zpracování statistik programy calamaris a cachemgr a filtrování internetových stránek pomocí squidGuard.

Squid funguje jako burzián. Přijímá požadavky od klientů (v tomto případě internetových prohlížečů) a ty pak předává dál odpovídajícím serverům poskytovatele. Když se požadovaný objekt vrátí, nechá si pro sebe jednu kopii, kterou uloží v diskové cache a druhou doručí zpět klientovi. Výhoda se projeví v okamžiku, kdy bude druhý uživatel požadovat stejný objekt – v tom případě není třeba stránku stahovat znovu, ale nahraje z cache. Výsledkem je nepoměrně rychlejší vyřízení požadavku a navíc dochází k úspoře kapacity linky.

Squid nabízí velké spektrum funkcí, např. hierarchické dělení proxy serveru, které rozkládá zátěž systému, vytváření pravidel pro přístup klientů, správu přístupových práv k jednotlivým stránkám a také statistiky nejčastěji používaných internetových stránek, chování uživatelů při surfování apod. Squid není genericou proxy. Standardně pouze zprostředkovává HTTP spojení. Kromě toho podporuje protokoly FTP, Gopher, SSL a WAIS, ale žádné další internetové protokoly typu Real Audio, News nebo videokonference. UDP protokol používá pouze pro podporu komunikace mezi různými cache. Z tohoto důvodu nejsou podporovány ani žádné další programy postavené na tomto protokolu.

## 33.1 Informace o proxy-cache

Proxy cache Squid lze využít různými způsoby. Spolu s firewallem může zlepšit bezpečnost. Lze použít více proxy společně. Umí také určit, jaké objekty se vy-

platí cachovat a na jak dlouho.

### 33.1.1 Squid a bezpečnost

Squid můžete provozovat spolu s firewallem a zabezpečit vnitřní síť před vnější sítí pomocí proxy. Firewall odmítne všechny přístupy ke službám z vnějšku kromě přístupu ke Squid. Všechna webová spojení musí být zprostředkována proxy.

Pokud konfigurace firewallu obsahuje DMZ, měla by proxy pracovat v této zóně. V takovém případě je důležité, aby všechny počítače v DMZ zasílaly logy počítačům ve vnitřní síti. Možnost implementace tzv. *transparentní* proxy je popsána v části 33.5 na straně 542.

### 33.1.2 Vícenásobná cache

Můžete nakonfigurovat více cache, které si vyměňují objekty. Snižuje se tak zátěž systému a zvyšuje pravděpodobnost nalezení objektu již v lokální síti. Můžete také vytvořit hierarchicky uspořádané cache, takže je cache schopná předat požadavek na objekt jiné cache na stejné úrovni nebo cache nadřazené – která pak vyřídí požadavek prostřednictvím jiné cache nebo stáhne objekt přímo ze zdroje.

Volba správné topologie je velice důležitá, protože by nemělo dojít ke zvýšení celkového síťového provozu. U velké sítě je možné nakonfigurovat proxy server pro každou podsíť a tu pak spojit s nadřazenou cache, která je opět napojena na proxy ISP (poskytovatele).

Tato komunikace je řízena prostřednictvím ICP (*Internet Cache Protocol*), který je vystavěn nad UDP. Výměna dat mezi jednotlivými cache se provádí prostřednictvím HTTP (*Hyper Text Transmission Protocol*) založeném na TCP.

Aby byl nalezen nejlepší server pro požadované objekty, posílá cache všem proxy stejné hierarchie tzv. ICP dotaz. Ostatní proxy pak odpoví prostřednictvím ICP buď *⟨HIT⟩* v případě, že objekt našly, nebo *⟨MISS⟩* v případě, že ho nenašly. V případě nálezu více HITů se proxy rozhodne, ze které cache bude stahovat. Toto rozhodování se provádí na základě rychlosti odpovědi. Když všechny cache ohlásí MISS, pak bude dotaz předán nadřazené cache.

Abyste zabránili vícenásobnému ukládání objektů v různých cache lokální sítě – používají se jiné ICP protokoly, jako je např. *CARP Cache Array Routing Protocol* nebo *HTCP Hyper-Text Cache Protocol*. Čím více objektů je v síti udržováno, tím větší je pravděpodobnost nálezu požadovaného.

### 33.1.3 Přechovávání objektů z Internetu

Ne všechny objekty v síti jsou statické. Existuje velké množství dynamicky generovaných CGI stránek, počítadel a SSL dokumentů, které nejsou ukládány v cache, protože jsou měněny při každém přístupu.

A u všech ostatních objektů je třeba zvážit, jak dlouho by měly zůstat v cache. Kvůli tomu mají objekty v cache přiřazeny různé stavy. V hlavičkách pak obsahují informace jako *⟨Last modified⟩* (datum poslední změny) nebo *⟨Expires⟩* (datum expirace), případně informaci o zákazu cachování objektu. Objekty v cache jsou odstraňovány převážně kvůli nedostatku místa, kde se používají algoritmy jako je LRU (*Least Recently Used*). Ten v podstatě maže nejdéle nepoužité objekty.

## 33.2 Systémové požadavky

Nejdříve by měla být určena zátěž systému. Je třeba věnovat zvláštní pozornost špičkám, které mohou být i 4x vyšší, než je denní průměr. Pokud si nejste jisti, pak je lepší nadhodnotit systémové požadavky, protože nevhodný hardware pro Squid může vést k výraznému poklesu výkonu. V následujícím textu jsou jednotlivé části seřazeny podle důležitosti.

### 33.2.1 Pevný disk

Při ukládání do meziskladu (cache) hraje rychlost zápisu velkou roli. Proto byste měli tomuto faktoru věnovat velkou pozornost. U pevných disků je nejdůležitější doba přístupu (náhodného), která je udávána v milisekundách. Protože bloky dat se kterými Squid pracuje jsou poměrně malé, je přístupová doba disku důležitější než jeho datová průchodnost. Pro účely proxy jsou lepší disky s vysokými otáčkami, neboť umožňují rychlejší pozicování hlavičky. Rychlost systému lze zvýšit využitím více disků současně, případně použitím RAID.

### 33.2.2 Velikost diskové cache

Pokud máte malou cache, pak je pravděpodobnost HITu velmi nízká, protože cache se velice rychle zaplní, a pak jsou starší objekty přepisovány novějšími. Pokud ale máte *⟨1 GB⟩* pro cache a uživatel potřebuje každý den pouze *⟨10 MB⟩*, pak máte minimálně sto dní, než se vám cache zaplní.

Nejjednodušší je určit velikost cache podle rychlosti připojení. Pokud máte 1 Mbit/s linku, pak bude maximální přenosová rychlost 128 KB/s. Za předpokladu, že veškerý datový přenos skončí v cache, máte za jednu hodinu uloženo více než 450 MB. Pokud bychom pokračovali a řekli bychom, že pracovní den má 8 hodin a pořád by byla linka plně využita, pak je to za jeden den 3,6 GB. Protože však nebývá linka vytížená na 100%, budou stačit zhruba 2 GB.

### 33.2.3 RAM

Velikost potřebné paměti pro Squid je závislá na počtu objektů, které se nachází v cache. Squid ukládá cachovací odkazy a často používané stránky v paměti tak, aby mohly být požadavky rychleji vyřizovány. Protože RAM je mnohem rychlejší než pevný disk.

Squid má v paměti také další data, např. tabulku se všemi použitými IP adresami, s nejčastěji používanými zásobníky, objekty a pak také seznamy s informacemi o přístupu a mnoho dalšího.

Proto je důležité, aby měl Squid dostatek operační paměti. Pokud by musel začít swapovat, tj. odkládat méně často používané části operační paměti do vyhrazeného diskového oddílu, dramaticky by klesl výkon. Pro správu cache v paměti můžete využít `cachemgr.cgi`, který je popsán v části 33.6 na straně 544.

### 33.2.4 CPU

Proxy nepotřebuje příliš výkonný procesor. Pouze během kontroly obsahu cache se zvyšuje zatížení procesoru. Pokud byste chtěli použít víceprocesorové stroje, pak nedosáhnete zvýšení výkonu Squidu. Lepší je přidat disky a operační paměť.

## 33.3 Spuštění squidů

Program Squid má SUSE LINUX již předkonfigurovaný, takže ho můžete spustit hned po instalaci. Předpokladem bezproblémového startu je správně nastavená síť – tj. aby byl nastaven alespoň nameserver a bylo možné pingnout. Problémy se mohou objevit v okamžiku, kdy používáte dynamickou DNS konfiguraci. V tom případě by alespoň nameserver měl mít platný zápis, protože pokud Squid nenajde v `/etc/resolv.conf` DNS server – tak se vůbec nespustí.

### 33.3.1 Příkazy pro spuštění squidů

Pro spuštění se přihlaste jako uživatel `root` a zadejte příkaz `rcsquidstart`. Při prvním spuštění se vytvoří adresářová struktura v `/var/squid/cache`, což provádí automaticky spouštěcí skript `/etc/init.d/squid` a může to trvat řádově několik vteřin až minut. Pokud se zobrazí zelené `done`, byla proxy úspěšně spuštěna. Na lokálním systému můžete funkčnost squidů ihned v prohlížeči nastavením proxy na `localhost` a portu na `3128`.

Abyste zpřístupnili Squid všem uživatelům, bude potřeba upravit konfigurační soubor `/etc/squid/squid.conf` tak, že změníte položku `http_access deny all` na `http_access allow all`. Mějte ale na mysli, že tím otevřete proxy všem, proto byste měli nastavit ACL. Bližší informace naleznete v části 33.4.2 na straně 539.

Změny v konfiguračním souboru `/etc/squid/squid.conf`, je potřeba načíst příkazem `rcsquidreload`. Nebo můžete Squid úplně restartovat příkazem `rcsquidrestart`.

Příkaz `rcsquidstatus` slouží ke zjištění, zda proxy běží. Zastavit ji můžete příkazem `rcsquidstop`. Může to chvíli trvat, protože Squid čeká až půl minuty (volba `shutdown_lifetime` v souboru `/etc/squid/squid.conf`), než přeruší spojení s klienty a zapíše data na disk.

#### Varování

Pokud ukončíte squidů tak, že ho zabijete příkazem `kill` nebo `killall`, může dojít k poškození cache, kterou je pak potřeba smazat, aby bylo možné squidů znovu spustit. `indexterm>`

#### Varování

Pokud Squid zemře krátce po úspěšném spuštění, zkontrolujte, zda není špatně nastaven nameserver či zda nechybí soubor `/etc/resolv.conf`. Squid zaznamenává důvod selhání spuštění do protokolového souboru `/var/squid/logs/cache.log`. Pokud se má Squid spouštět při startu systému automaticky, použijte editor úrovně běhu YaST a aktivujte Squid v požadovaných úrovních, viz 2.8.7 na straně 62.

Při odinstalování proxy se neodstraní ani cache, ani protokolové soubory. Je potřeba ručně smazat adresář `/var/cache/squid`.

### 33.3.2 Lokální DNS server

Lokální DNS server je výhodný i v případě, že nespravuje vlastní doménu. Stačí, když funguje pouze jako caching-only DNS a umí bez zvláštní konfigurace zpracovat DNS dotazy, resp. je předat root nameserveru (viz 24.2 na straně 413). Jak toho dosáhnete, záleží na tom, zda jste zvolili dynamické DNS při konfiguraci připojení k Internetu.

**Dynamické DNS** Za běžných okolností, při použití dynamického DNS, je DNS server nastaven poskytovatelem během navazování spojení. Lokální soubor `/etc/resolv.conf` je upraven automaticky. Toto chování je způsobeno nastavením `sysconfig` proměnné `MODIFY_RESOLV_CONF_DYNAMICALY` na `YES`. Nastavte ji YaST `sysconfig` editorem (viz 7.8 na straně 161) na `NO`. Pak zadejte lokální DNS server do souboru `/etc/resolv.conf`: IP adresu `127.0.0.1` pro `localhost`. Tak Squid při startu vždy nalezne lokální DNS server.

Aby byl přístupný nameserver poskytovatele, zadejte ho v konfiguračním souboru `/etc/named.conf` spolu s jeho IP adresou do položky `forwarders`. Při použití dynamického DNS to lze automatizovat nastavením proměnné `MODIFY_NAMED_CONF_DYNAMICALY` na `YES`.

**Statické DNS** Při použití statického DNS nedochází během navazování spojení k žádným úpravám DNS, takže není třeba upravovat žádné `sysconfig` proměnné. Musíte ovšem do souboru `/etc/resolv.conf` zadat lokální DNS server, jak je výše popsáno. Navíc musíte ručně zadat statický DNS server poskytovatele (s IP adresou) do souboru `/etc/named.conf` (položka `forwarders`).

---

#### Tip

##### DNS a firewall

Pokud máte spuštěný firewall, ujistěte se, že skrze něj mohou DNS požadavky projít.

---

Tip



## 33.4 Konfigurační soubor `/etc/squid/squid.conf`

Všechna nastavení Squid proxy serveru jsou zapsána v souboru `/etc/squid/squid.conf`. Pro první spuštění Squida není třeba v tomto souboru provádět žádné změny, ale externím klientům bude zamítnut přístup. Proxy bude dostupná pouze pro `localhost`. Výchozí port je 3128. Předinstalovaný soubor `/etc/squid/squid.conf` obsahuje podrobné komentáře s popisy voleb a mnoho příkladů. Téměř všechny položky začínají znakem `#` (komentář) a obsahují podrobné informace. Zadané hodnoty jsou téměř vždy shodné s výchozími, takže odstranění komentáře bez změny hodnoty má pětšinou minimální vliv. Lepší je ale příklady nechat beze změny a zadat volby se změněnými parametry na nový řádek pod příklad. Tak budete mít přehled o výchozích hodnotách a vámi provedených změnách.

### Tip

#### Přízpůsobení konfiguračního souboru po aktualizaci

Pokud jste aktualizovali Squid ze starší verze, doporučuje se upravit nový `/etc/squid/squid.conf` a jen do něj zadat změny provedené ve starším souboru. Pokud byste použili starší konfigurační soubor přímo, riskujete, že nebude správně fungovat, protože některé volby se mezi verzemi mění.

### Tip

### 33.4.1 Základní nastavení

**http\_port 3128** Toto je port, na kterém poslouchá Squid požadavky klientů. Přednastaven je na `<3128>`, ale běžný je také port `<8080>`. Další porty můžete přidat (odděluje je mezerou).

**cache\_peer <hostname> <type> <proxy-port> <icp-port>**

Zde uveďte nadřazenou proxy, např. když chcete využívat proxy poskytovatele. Jako `<hostname>` uveďte jméno a IP adresu používané proxy.

Jako `<type>` zadejte `<parent>`. Jako číslo portu poskytovatele (`<proxy-port>`) se nejčastěji používá `<8080>`. `<icp-port>` můžete nastavit na `<7>` nebo `<0>`, pokud neznáte ICP port nadřazené proxy a její používání není dohodnuto s poskytovatelem. Navíc byste za čísla portů měli zapsat volby `default`

a `no-query`, čímž zamezíte používání ICP protokolu. Squid se pak vůči proxy poskytovatele chová jako obyčejný webový prohlížeč.

**cache\_mem 8 MB** Tato položka stanoví, kolik operační paměti bude Squid pro cache používat. Přednastaveno je `<8 MB>`.

**cache\_dir ufs /var/cache/squid 100 16 256**

Položka `cache_dir` určuje adresář, do kterého budou na disku ukládány jednotlivé objekty. Číslo za cestou k adresáři znamená: maximální velikost cache v MB; počet podadresářů; a počet podadresářů podadresářů. Parametr `ufs` by měl zůstat beze změny. Přednastavenými hodnotami pro velikost cache jsou 100 MB diskového prostoru v adresáři `/var/cache/squid`, kde bude vytvořeno 16 adresářů, každý z nich se 256 podadresářů. Při vyčleňování místa na disku byste si měli nechat dostatek rezerv, rozumné je vytvářet cache o velikosti 50 až 80 procent volného místa. Kromě toho byste měli poslední dvě čísla (počty adresářů) zvětšovat velice opatrně, protože režie adresářových struktur může snížit výkon systému. Pokud máte pro cache více disků, můžete vytvořit odpovídající množství řádků s definicí `cache_dir`.

**cache\_access\_log /var/log/squid/access.log**

Cesta k protokolovému souboru.

**cache\_log /var/log/squid/cache.log** Cesta k protokolovému souboru.

**cache\_store\_log /var/log/squid/store.log**

Cesta k protokolovému souboru.

Tyto tři volby definují cesty k protokolovým souborům a není třeba je měnit. Pouze v případě, že je cache velice často dotazována, se může hodit přesunout protokolové soubory na jiný disk.

**emulate\_httptd\_log off** Změnou na `on` získáte čitelné protokolové soubory, se kterými si ale neporadí některé programy, které mají na starosti vyhodnocování.

**client\_netmask 255.255.255.255** Touto položkou můžete maskovat IP adresy zapisované do logů a skrýt tak identitu klientů. Pokud zde napíšete např. `255.255.255.0`, tak bude poslední pozice IP adresy vynulována.

**ftp\_user Squid@** Zde nastavte heslo, které bude Squid používat pro anonymní FTP login. Může mít smysl uvést zde platnou e-mailovou adresu, protože některé FTP servery její platnost kontrolují.

**cache\_mgr webmaster** Tato volba slouží pro uvedení e-mailové adresy, na kterou se pošle zpráva v případě neočekávaného pádu. Přednastaveno je *<webmaster>*.

**logfile\_rotate 0** Squid umí také rotovat uložené protokolové soubory, pokud ho spustíte s volbou `squid-k rotate`. Soubory jsou číslovány, jakmile se dojde k nastavené hodnotě, přepíše se nejstarší soubor. Výchozí nastavení je 0, protože pro archivaci a mazání protokolových souborů používá SUSE LINUX cron úlohu nastavenou v `/etc/logrotate/squid`.

**append\_domain <domain>** Volbou `append_domain` můžete určit, která doména bude automaticky připojena v případě, že není žádná uvedena. Nejčastěji se zde uvádí vlastní doména, takže stačí v prohlížeči uvést *www* a dostanete se na vlastní webserver.

**forwarded\_for on** Když nastavíte na *<off>*, odstraní Squid IP adresu a jméno počítače klienta z HTTP dotazu.

**negative\_ttl 5 minutes; negative\_dns\_ttl 5 minutes**

Ve standardním případě není třeba toto nastavení upravovat. Pokud ale máte vytáčenou linku, pak se může stát, že Internet nebude po nějakou dobu přístupný. To je tím, že si Squid poznamenává neúspěšné dotazy a brání se znovu dotazovat, i když je již spojení s Internetem obnoveno. V tom případě změňte *<minutes>* na *<seconds>* a nechte znovu načíst stránku v prohlížeči.

**never\_direct allow <acl\_name>** Pokud chcete zabránit tomu, aby Squid vyřizoval požadavky přímo z Internetu, pak použijte tuto volbu. V tom případě je ale potřeba, aby existovala ještě další proxy, které bude Squid své požadavky zasílat. Tu je třeba nastavit ve volbě `cache_peer`. Pokud zadáte jako *<acl\_name>* `all`, pak zajistíte, že všechny požadavky budou předávány *nadřazené* proxy. To je třeba např. tehdy, když poskytovatel striktně trvá na využívání jeho proxy, nebo když je firewall nastaven tak, že nepovoluje přímý přístup k Internetu.

## 33.4.2 Volby pro kontrolu přístupu

Squid obsahuje velice sofistikovaný systém pro řízení přístupu k proxy. Pomocí ACL je velice dobře a jednoduše konfigurovatelný. V zásadě se jedná o seznam pravidel, která jsou jedno po druhém zpracovávány. ACL je třeba definovat předtím, než budou použita. Některá jsou již definována, jako je *all* a *localhost*. Ale

pouhým vytvořením ACL ještě nic neprovedete. Teprve, když ho použijete např. spolu s *http\_access*, tak se změny projeví.

**acl** *<acl\_name>* *<type>* *<data>* ACL potřebuje pro svou definici minimálně tři parametry. Název *acl\_name* může být libovolný. U *type* můžete zvolit z celé řady různých možností, které jsou uvedeny v části *ACCESS CONTROLS* souboru */etc/squid/squid.conf*. Jaká *data* uvést, záleží na typu ACL. Lze je také načíst ze souboru, například, přes jméno počítače, IP adresu nebo URL. Následují krátké příklady:

```
acl mujnet srcdomain .ma-domena.cz
acl ucitele src 192.168.1.0/255.255.255.0
acl studenti src 192.168.7.0-192.168.9.0/255.255.255.0
acl obed time MTWHF 12:00-15:00
```

**http\_access allow** *<acl\_name>* Volbou *http\_access* určíte, kdo může proxy používat a k čemu může na Internetu přistupovat. Zde využijete výše definovaná ACL nebo použijete ta přednastavená, tj. *localhost* a *all*. Přístup může být povolen nebo zakázán pomocí hodnot *deny* či *allow*. Můžete vytvořit celý seznam položek *http\_access*, které budou zpracovávány odshora dolů a podle toho, co se nače jako první bude přístup povolen nebo zakázán. Jako poslední položka by měl být vždy *http\_access deny all*. V následujícím příkladu povolíme přístup všem lokálním uživatelům, zatímco všem ostatním ho zakážeme.

```
http_access allow localhost
http_access deny all
```

V dalším příkladu (s využitím vlastních ACL) mají učitelé povolen stálý přístup k Internetu, zatímco studenti k němu mají přístup pouze od pondělí do pátku v čase oběda.

```
http_access deny localhost
http_access allow ucitele
http_access allow studenti obed time
http_access deny all
```

Volby *http\_access* byste, kvůli přehlednosti, měli psát pouze na jedno, předem určené, místo v souboru */etc/squid/squid.conf*. A to mezi řádky:

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
```

a uzavírající text:

```
http_access deny all
```

### **redirect\_program /usr/bin/squidGuard**

Tato volba slouží pro tzv. přesměrování, kdy jsou dotazy předávány externímu programu, v našem případě squidGuard, který dokáže zakázat přístup k určeným URL. Spolu s proxy autentizací a vhodnými ACL tak můžete velice precizně řídit přístup k Internetu pro různé skupiny. squidGuard je v separátním balíku a musí se tedy nainstalovat zvlášť.

### **authenticate\_program /usr/sbin/pam\_auth**

Pokud je třeba autentizovat uživatele při přístupu k proxy, můžete použít program pam\_auth. Při prvním přihlášení uživatele se spustí přihlašovací dialog, kde musí uživatel vložit uživatelské jméno a heslo. Navíc se stále vyžaduje ACL, připojit se mohou pouze klienti s platným loginem:

```
acl password proxy_auth REQUIRED
```

```
http_access allow password  
http_access deny all
```

Klíčové slovo *REQUIRED* za *proxy\_auth* můžete nahradit seznamem povolených jmen uživatelů nebo cestou k takovému seznamu.

### **ident\_lookup\_access allow <acl\_name>**

Tato volba zajistí, že za všechny klienty definované v ACL je proveden identifikační dotaz, který prověří identitu uživatele. Když nastavíte *acl\_name* na *<all>*, bude se provádět dotazování pro všechny klienty. Na klientech však musí běžet identifikační démon. V Linuxu můžete nainstalovat program pidentd, pro Windows existuje volně dostupný software, který si můžete stáhnout z Internetu. Aby byli připuštěni pouze klienti s úspěšným identifikačním dotazem *ident lookup*, je potřeba opět definovat vhodný ACL.

```
acl idenhosts ident REQUIRED
```

```
http_access allow idenhosts  
http_access deny all
```

Také zde je možné nahradit *REQUIRED* seznamem povolených jmen uživatelů. Používání *ident* může přístup výrazně zpomalit, protože kontrola se provádí při každém dotazu.

## 33.5 Konfigurace transparentní proxy

Standardně posílá prohlížeč na určitý port proxy serveru dotazy a proxy mu odpovídající objekty poskytuje, ať už se v cache nacházejí nebo ne. V praxi pak mohou nastat různé situace:

- Z bezpečnostních důvodů je lepší, když proxy používají všichni klienti.
- Je třeba, aby uživatelé používali proxy, i když o ní neví.
- Proxy se v síti přesunula, ale klienti by si měli i nadále zachovat svou starou konfiguraci.

V každém z těchto případů je vhodné nasadit transparentní proxy. Princip je přitom velice jednoduchý. Internetový prohlížeč pošle svůj požadavek. Na cestě sedí proxy, která tento požadavek zpracuje a odpověď odešle zpět prohlížeči, který vůbec netuší, že komunikuje s proxy a ne přímo se zdrojem. Celý proces je zcela transparentní.

### 33.5.1 Konfigurace jádra

Nejprve se ujistěte, že jádro proxy serveru podporuje transparentní proxy. Jádro systému SUSE LINUX tuto podmínku splňuje. Pokud tomu tak není, rekompilejte jádro s podporou transparentní proxy. Více informací najdete v kapitole 9 na straně 187.

### 33.5.2 Možnosti konfigurace v `/etc/squid/squid.conf`

Volby v souboru `/etc/squid/squid.conf` potřebné pro aktivaci transparentní proxy jsou následující:

- `httpd_accel_host virtual`
- `httpd_accel_port 80`  
Číslo portu, na kterém běží HTTP server.
- `httpd_accel_with_proxy on`
- `httpd_accel_uses_host_header on`

### 33.5.3 Konfigurace firewallu pomocí SuSEfirewall2

Všechny příchozí dotazy musí být pomocí firewallu přesměrovány na port Squida. K tomu můžete použít nástroj SuSEfirewall2. Jeho konfigurace se nachází v souboru `/etc/sysconfig/SuSEfirewall2`. Soubor je dobře komentovaný. I když chcete nastavit pouze transparentní proxy, je potřeba provést určitá nastavení ve firewallu:

- Rozhraní pro přístup k Internetu: `FW_DEV_EXT="eth1"`
- Rozhraní pro přístup k vnitřní síti: `FW_DEV_INT="eth0"`

Když jste definovali rozhraní pro přístup k jednotlivým sítím, je potřeba povolit porty a služby, které budou přístupné z vnější sítě. V našem příkladu jsou vně nabízeny jen webové služby:

```
FW_SERVICES_EXT_TCP="www"
```

Pak je třeba povolit porty a služby dostupné z vnitřní (bezpečné) sítě přes TCP i UDP:

```
FW_SERVICES_INT_TCP="domain www 3128"
FW_SERVICES_INT_UDP="domain"
```

Tím jsou povoleny webové služby a Squid, který běží standardně na portu 3128. Navíc je povolena služba DNS (*domain*). Pokud DNS povolovat nechcete, smažte ho z nastavení výše a nastavte následující volbu na `no`:

```
FW_SERVICE_DNS="yes"
```

Nejdůležitější je volba číslo 15:

#### *Příklad 33.1: Konfigurace firewallu: Volba 15*

```
#
# 15.)
# Which accesses to services should be redirected to a local port
# on the firewall machine?
#
# This can be used to force all internal users to surf via your
# Squid proxy, or transparently redirect incoming Web traffic to
```

```
# a secure Web server.
#
# Choice: leave empty or use the following explained syntax of
# redirecting rules, separated with spaces.
# A redirecting rule consists of 1) source IP/net,
# 2) destination IP/net, 3) original destination port and
# 4) local port to redirect the traffic to, separated by a colon,
# e.g. "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"
#
```

Komentáře popisují syntaxi. Nejdřív se vezme IP adresa a síťová maska interní sítě, ze které se bude přistupovat k proxy firewallu. Pak zadejte adresu a masku cíle, tj. kam jsou požadavky klientů posílány. V případě webového prohlížeče zvolte síť 0/0, což značí přístup kamkoliv. Pak nastavte originální port a port, na který jsou požadavky přesměrovávány. Protože Squid podporuje kromě HTTP i další protokoly, přeměrujte na proxy požadavky i z dalších portů, jako např. FTP (port 21), HTTPS nebo SSL (port 443). V našem příkladě jsou webové služby (port 80) přesměrovány na Squid proxy (port 3128). Pokud je sítí nebo služeb více, musí být v položce odděleny mezerou.

```
FW_REDIRECT_TCP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
FW_REDIRECT_UDP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
```

Firewall s novou konfigurací spustíte nastavením proměnné `START_FW` v souboru `/etc/sysconfig/SuSEfirewall12` na hodnotu "yes".

Pak spusťte Squid tak, jak je uvedeno v části 33.3 na straně 534. Zda vše funguje právně se můžete přesvědčit v protokolovém souboru `/var/log/squid/access.log`.

Zda jsou všechny porty nastaveny dobře zjistíte tak, že použijete z libovolného místa mimo vaši síť portscan, tj. že se pokusíte zjistit, které porty jsou otevřené. V našem případě by měl být otevřen pouze port 80. Ke skenování použijte např. program `nmap` (`nmap(-O IP_adresa)`).

## 33.6 cachemgr.cgi

Cache manager (`cachemgr.cgi`) je CGI program pro zpracování statistik spotřeby paměti běžící proxy Squid. Je to také pohodlný způsob správy cache.



### 33.6.1 Nastavení

Nejprve je třeba mít v systému běžící webový server. Zda server běží zjistíte jako uživatel root příkazem `rcapachestatus`. Pokud se zobrazí hlášení:

```
Checking for service httpd: OK
Server uptime: 1 day 5 hours 23 minutes 17 seconds
```

tak Apache běží. V opačném případě je třeba webový server spustit příkazem `rcapachestart`. Jako poslední krok je třeba zkopírovat `cachemgr.cgi` do adresáře `cgi-bin` Apache příkazem:

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi /srv/www/cgi-bin/
```

### 33.6.2 ACL cache manageru v `/etc/squid/squid.conf`

V originálním souboru jsou výchozí nastavení potřebná pro cache manager. První ACL je nejdůležitější, protože se cache manager snaží se Squidem komunikovat pomocí `cache_object` protokolu.

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

Měla by být nastavena i následující pravidla:

```
http_access allow manager localhost
http_access deny manager
```

Následující pravidla předpokládají, že web server a Squid běží na stejném počítači. Pokud komunikace mezi cache managerem a Squidem vychází ze strany web serveru na jiném počítači, nastavte další ACL, jak je uvedeno v příkladu 33.2 na této straně.

#### *Příklad 33.2: Přístupová pravidla*

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # webserver IP
```

Pak přidejte pravidla z příkladu 33.3 na této straně.

### *Příklad 33.3: Přístupová pravidla*

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

Nastavte heslo pro správce cache nutné pro přístup k rozšířeným volbám, jako vzdálenému zavření cache nebo zobrazení podrobných informací o cache. K tomu slouží položka `cachemgr_passwd` s heslem a seznam voleb, které budou zobrazeny po uvedení hesla. Tento seznam je uveden v komentáři v `/etc/squid/squid.conf`.

Pokaždé, když se změní konfigurace Squidu, je potřeba ho restartovat `rcsquidreload`.

### **33.6.3 Prohlížení statistik**

Podívejte se na `http://vas_server/cgi-bin/cachemgr.cgi`. Stiskněte 'continue' a nechte si zobrazit různé statistiky. Bližší informace o jednotlivých volbách naleznete v často kladených dotazech k programu Squid (FAQ) na adrese `http://www.squid-cache.org/Doc/FAQ/FAQ-9.html`

## **33.7 squidGuard**

Tato kapitola by měla být úvodem do konfigurace squidGuard a měla by vám představit možnosti jeho použití. Pro podrobné popisy jemných nuancí zde nebude dostatek místa. Hlubší informace naleznete na internetových stránkách `http://www.squidguard.org`.

squidGuard je svobodný, flexibilní a velice rychlý filtr pro Squida. Podporuje definování množství pravidel pro přístup s různými omezeními pro různé skupiny. Pro přesměrování používá squidGuard standardní rozhraní Squidu.

squidGuard můžete použít k následujícím úkolům:

- Omezení přístupu určitých uživatelů pouze k definovaným serverům nebo URL.

- Zakázání přístupu určitých uživatelů k určitým serverům nebo URL.
- Zamezení přístupu určitých uživatelů na základě regulárních výrazů nebo slov.
- Přesměrování ze zakázané URL na inteligentní CGI stránku.
- Přesměrování nepřihlášeného uživatele na registrační formulář.
- Náhrada reklamních banerů prázdným GIFem.
- Rozdílná pravidla přístupu v závislosti na čase, dni v týdnu a datu.
- Rozdílná pravidla pro jednotlivé skupiny uživatelů.

Ani squidGuard nebo Squid ale neumí:

- Filtrovat, cenzurovat nebo upravovat text v dokumentech.
- Filtrovat, cenzurovat nebo upravovat skriptovací jazyky (např. JavaScript nebo VBscript), které jsou součástí HTML.

Pro použití programu squidGuard musíte nejprve nainstalovat balíček squidGuard a pak upravit konfigurační soubor `/etc/squidguard.conf`. Pokud hledáte příkladové konfigurace, podívejte se na <http://www.squidguard.org/config/>. Později můžete zkoušet složitější konfigurace.

Pak vytvořte stránku *Přístup odmítnut* nebo CGI stránku, na kterou bude klient přesměrován v případě, že přistoupí na zakázanou stránku. I zde doporučujeme používat Apache.

Nyní musíte squidu říct, aby používal squidGuard. Stačí použít následující zápis v `/etc/squid/squid.conf`:

```
redirect_program /usr/bin/squidGuard
```

Další volba `redirect_children` nastavuje počet přesměrovacích procesů (squidGuardu), které na stroji poběží. Standardně dokáže squidGuard zpracovat 100000 požadavků za 10 vteřin na 500MHz Pentiu s 5900 doménami a 7880 URL. Proto se nedoporučuje nastavovat více než 4 procesy, protože pak zabírají pouze místo v paměti.

```
redirect_children 4
```

Nakonec necháte Squida znovu načíst konfiguraci příkazem `rcsquidreload`. Přišel čas otestovat nastavení v prohlížeči.

## 33.8 Vytvoření protokolů programem Calamaris

Calamaris je perlový skript, který vytváří hlášení o aktivitě cache. Tyto reporty jsou dostupné buď v ASCII nebo HTML. Calamaris využívá při sestavování protokolových souborů Squidu. Domovskou stránku projektu naleznete na <http://Calamaris.Cord.de/>.

Program se používá velice jednoduše. Přihlaste se jako uživatel `root` a použijte příkaz `cat access.log.soubory | calamaris <volby> > vystupnisoubor`. V případě, že zpracováváte více protokolových souborů, je důležité seřadit je chronologicky, nejstarší soubor první. Použitelné volby jsou následující:

- a Výstupem budou všechna dostupná hlášení.
- w Výstupem je protokol ve formátu HTML.
- l Nadpis nebo logo v záhlaví.

Další informace o různých volbách obsahuje manuálová stránka calamaris. Typickým příkladem použití je:

```
cat access.log.2 access.log.1 access.log | calamaris -a -w \  
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

Hlášení se tak uloží do adresáře webserveru.

Dalším nástrojem, který můžete použít pro generování hlášení o stavu cache, je SARG (Squid Analysis Report Generator). Další informace naleznete na stránkách <http://web.onda.com.br/orso/>.

## 33.9 Další informace o Squidu

Navštivte domovskou stránku <http://www.squid-cache.org/>. Naleznete tam uživatelskou příručku a rozsáhlý seznam často kladených dotazů (FAQ).

Navíc máte k dispozici HOWTO, které naleznete po nainstalování balíčku `howtoenh` v adresáři `/usr/share/doc/howto/en/mini/TransparentProxy.gz`. Využít můžete i konferenci `squid-users@squid-cache.org` nebo její archiv na adrese <http://www.squid-cache.org/mail-archive/squid-users/>.

**Část IV**

**Správa**



# Bezpečnost v Linuxu

Ke kontrole a směrování datového provozu ve své síti můžete použít také další mechanismy např. maškarádu, firewally nebo Kerbera. Secure Shell (SSH) umožňuje šifrované připojení na vzdálený počítač. Šifrování a další nástroje chrání vaše choulostivá data před nepovolanými uživateli. Mimo čistě technických informací v této kapitole najdete také základní informace o bezpečnostních aspektech linuxových sítí.

34.1	Firewall a maškaráda . . . . .	552
34.2	SSH: bezpečná práce v síti . . . . .	559
34.3	Šifrování diskových oddílů a souborů . . . . .	564
34.4	Bezpečnost a soukromí . . . . .	567

## 34.1 Firewall a maškaráda

Linux v síťovém prostředí umožňuje takovou manipulaci s pakety, která udržuje oddělené vnější a vnitřní síťové oblasti. Linuxový systém *netfilter* poskytuje prostředky pro vybudování efektivního firewallu udržujícího jednotlivé sítě odděleny. S pomocí *iptables* obecné tabulkové struktury pro definici pravidel umožňuje přesnou kontrolu, kterým paketům je dovoleno přejít přes síťové rozhraní. Takový paketový filtr lze snadno nastavit pomocí SuSEfirewall2 a odpovídajícího modulu YaST.

### 34.1.1 Filtrování paketů pomocí iptables

Komponenty netfilter a iptables jsou zodpovědné za filtrování a manipulaci s pakety a za překlad síťových adres (NAT). Filtrovací kritéria a všechny s nimi spojené akce jsou uloženy v řetězech (chains), se kterými jsou porovnávány všechny příchozí pakety. Řetězy jsou uloženy v tabulkách. Manipulaci s těmito tabulkami a sadami pravidel umožňuje příkaz *iptables*.

Linuxové jádro si udržuje tři tabulky, každou z nich pro jednu skupinu funkcí paketového filtru:

**filter** Tato tabulka obsahuje většinu filtrovacích pravidel, neboť implementuje *filtrování paketů* v užším slova smyslu. Určuje například, který paket může projít skrz (ACCEPT) a který je zahozen (DROP).

**nat** Tato tabulka určuje změny ve zdrojových a cílových adresách paketů. S její pomocí lze rovněž implementovat *maškarádu*, což je zvláštní případ NAT používaný pro propojení privátní sítě s Internetem.

**mangle** Pravidla v této tabulce umožňují měnit hodnoty uložené v IP hlavičkách (např. typ služby).

Výše zmíněné tabulky obsahují několik předdefinovaných řetězů (chains) pro porovnávání s pakety:

**\mbbox{PREROUTING}** Tento řetěz je aplikován na příchozí pakety.

**\mbbox{VSTUP (input)}** Tento řetěz je aplikován na pakety určené pro vnitřní systémové procesy.



`\mbox{FORWARD}` Tento řetěz je aplikován na pakety, které jsou na systému pouze směrovány.

`\mbox{VÝSTUP (output)}` Tento řetěz je aplikován na pakety, které pocházejí z vlastního systému.

`\mbox{POSTROUTING}` Tento řetěz je aplikován na všechny odchozí pakety.

Obrázek 34.1 na straně 578 znázorňuje cesty, po kterých se v systému může síťový paket pohybovat. Z důvodu jednoduchosti jsou tabulky zobrazeny jako části řetězců, ale ve skutečnosti jsou řetězy umístěny právě v tabulkách.

V nejjednodušším případě dorazí paket určený přímo pro systém na síťové rozhraní `eth0`. Paket je nejprve postoupen řetězu `PREROUTING` tabulky `mangle`, a pak řetězu `PREROUTING` tabulky `nat`. Následující krok určí, že cílem paketu je proces na vlastním systému. Po průchodu přes řetězy `INPUT` tabulek `mangle` a `filtr` dosáhne paket konečně svého cíle, pokud ovšem odpovídá pravidlům v tabulce `filtr`.

## 34.1.2 Základy maškarády

Maškaráda je linuxově specifická forma NAT (překladu síťových adres). Lze ji použít k propojení malé lokální sítě LAN (ve které počítače používají IP adresy z privátního rozsahu, viz 22.1.2 na straně 371) s Internetem. Aby se mohl počítač z LAN připojit k Internetu, musí být jeho privátní adresa přeložena na veřejnou, používanou v Internetu. O to se stará router (směrovač), který slouží jako brána mezi LAN a Internetem. Princip je jednoduchý: router má více než jedno síťové rozhraní, obvykle síťovou kartu a zvláštní rozhraní pro připojení k Internetu. Zatímco druhé spojuje router s vnějším světem, první, nebo i více takových, spojuje router s počítači v síti LAN. Počítače v síti LAN tak mohou posílat pakety, které nejsou určeny pro lokální síť, na router.

### Důležité

#### Použití správné síťové masky

Při nastavení sítě se ujistěte, že oznamovací adresa a maska sítě je nastavena pro všechny počítače stejně. Pokud to tak není, síť nefunguje správně, protože pakety nemohou být správně směrovány.

### Důležité

Kdykoliv počítač v lokální síti LAN pošle paket určený pro internetovou adresu, je poslán na implicitní router. Router však musí být správně nakonfigurován, aby

mohl pakety předávat dál. Z bezpečnostních důvodů to SUSE LINUX neumožňuje v implicitní instalaci. Chcete-li předávání povolit, nastavte proměnnou `IP_FORWARD` v souboru `/etc/sysconfig/sysctl` na `IP_FORWARD=yes`.

Cílový počítač vidí váš router, ale neví nic o počítači ve vaší interní síti, ze kterého paket pochází. Proto se tato technika nazývá maškaráda. Díky překladu adres je router prvním cílem všech paketů zaslaných jako odpověď. Router musí tyto pakety rozpoznat a přeložit jejich cílovou adresu tak, aby mohly být předány správnému počítači v lokální síti.

Vzhledem k tomu, že směrování příchozích paketů závisí na maškarádové tabulce, neexistuje způsob, jak otevřít přímé spojení s počítačem v lokální síti zvenku. Pro takové spojení není v tabulce žádný zápis. Navíc každé již navázané spojení má v tabulce přiřazený stavový zápis, takže zápis nemůže být použit jiným spojením.

Důsledkem jsou možné problémy s některými komunikačními protokoly, např. ICQ, cucme, IRC (DCC, CTCP) a FTP (PORT režim). Mnoho FTP klientů používá režim PASV. Tento pasivní režim působí při používání maškarády a filtrování paketů podstatně méně problémů.

### 34.1.3 Základy firewallu

*Firewall* je běžně používaný termín pro mechanismus zajišťující propojení sítí a kontrolující přenos dat mezi nimi. Přesně řečeno, mechanismus popsany v této části se jmenuje *paketový filtr*. Paketový filtr řídí datový tok podle určitých kritérií, jako je komunikační protokol, porty a IP adresy. To umožňuje zablokovat pakety, které by, vzhledem ke svým adresám, neměly být do vaší sítě doručeny. Pokud chcete povolit veřejný přístup k vašemu webserveru, musíte explicitně otevřít příslušný port. Paketový filtr nicméně nezkontroluje obsah paketů s legitimními adresami, jako například paketů pro webserver. I když by příchozí pakety byly například zaslány za účelem nabourání CGI programu na webserveru, paketový filtr je nechá normálně projít.

Efektivnější ale složitější mechanismus je kombinace několika typů systémů, jako například paketový filtr spolupracující s aplikační bránou nebo proxy. V takovém případě paketový filtr odmítá všechny pakety určené pro zakázané porty. Přijaty jsou pouze pakety určené pro aplikační bránu. Tato brána nebo proxy předstírá, že je klientem. V jistém smyslu lze takovou proxy považovat za maškarádu na úrovni protokolu používaného aplikací. Takovou proxy je například Squid, HTTP proxy server. Aby prohlížeč mohl využít proxy, musí být patřičně nakonfigurován. Všechny HTTP požadavky jsou obsluhovány proxy s využitím cache (vy-

rovnávací paměti) a stránky, které se v cache nenalézají, jsou staženy proxy z Internetu. Dalším příkladem je SUSE proxy-suite (proxy-suite), která poskytuje proxy pro protokol FTP.

Následující část se zabývá paketovým filtrem dodávaným se systémem SUSE LINUX. Další informace o filtrování paketů a firewallu naleznete v dokumentu Firewall HOWTO obsaženém v balíčku howto. Pokud je tento balíček nainstalovaný, můžete si dokument přečíst pomocí příkazu `less/usr/share/doc/howto/en/Firewall-HOWTO.gz`.

### 34.1.4 SuSEfirewall2

Skript SuSEfirewall2 čte proměnné z `/etc/sysconfig/SuSEfirewall2` a generuje sadu iptables pravidel. SuSEfirewall2 definuje tři bezpečnostní zóny:

**Vnější síť** Protože neexistuje žádný způsob, jak kontrolovat dění ve vnější síti, musí být počítače proti ní chráněny. Ve většině případů je vnější síť Internet, ale může to být i jiná nezabezpečená síť, například WLAN.

**Vnitřní síť** Privátní síť, nejčastěji LAN. Pokud počítače v této síti používají IP adresy z privátního rozsahu (viz 22.1.2 na straně 371), je pro přístup k vnější síti zapotřebí použít překlad síťových adres (NAT).

**Demilitarizovaná zóna (DMZ)** Ačkoliv jsou počítače v této zóně dosažitelné z vnitřní i vnější sítě, samy nemají do vnitřní sítě přístup. Tím se před vnitřní sítí vytvoří obranný val navíc.

Jakýkoliv síťový provoz, který není explicitně povolen filtračním pravidlem, je pomocí iptables zakázán. Proto musí být každé síťové rozhraní umístěno do jedné ze tří zón. Pro každou zónu je třeba určit, které služby a protokoly jsou povoleny. Pravidla jsou používána jen na pakety pocházející ze vzdálených počítačů. Lokálně generované pakety nejsou firewallem zachycovány.

Konfiguraci lze provést pomocí nástroje YaST (viz 34.1.4 na této straně). Lze ji provést i ručně v dobře okomentovaném souboru `/etc/sysconfig/SuSEfirewall2`. Navíc je řada příkladů nastavení dostupná v `/usr/share/doc/packages/SuSEfirewall2/EXAMPLES`.

### Konfigurace pomocí YaST

## Důležité

### Automatické nastavení firewallu

YaST automaticky spouští firewall na všech nakonfigurovaných rozhraních. Pokud je na systému nakonfigurován a aktivován server a v dialogích pro nastavení serveru použijete volbu 'Na zvolených portech a rozhraních otevřít firewall' nebo 'Na zvolených portech otevřít firewall', YaST automaticky upraví konfiguraci firewallu. Dialogy některých serverových modulů mají tlačítko 'Doladění firewallu' pomocí kterého lze aktivovat další služby a porty. Modul YaST pro firewall se používá pouze k aktivaci, deaktivaci či rekonfiguraci firewallu.

## Důležité

Dialogy pro nastavení firewallu v grafickém prostředí jsou dostupné v nástroji YaST zvolením položek 'Bezpečnost a uživatelé' → 'Firewall'. Konfigurace je rozdělena do sedmi částí, ke kterým lze přistupovat přímo stromové struktury vlevo.

**Začátek** V tomto dialogu můžete nastavit spouštění firewallu. Ve výchozím nastavení se SuSEfirewall2 spouští automaticky. Můžete ho ale spustit nebo zastavit v tomto dialogu ručně. Chcete-li použít svá nová nastavení, stiskněte 'Uložit nastavení a restartovat firewall'.

**Rozhraní** Seznam v tomto dialogu obsahuje všechna známá rozhraní. Chcete-li rozhraní odebrat ze zóny, klikněte na něj, stiskněte tlačítko 'Změnit' a vyberte 'Není přiřazena žádná zóna'. Chcete-li rozhraní přiřadit zóně, stiskněte 'Změnit' a vyberte některou z dostupných zón. Můžete také vytvořit zvláštní rozhraní s vlastním nastavením pomocí 'Vlastní'.

**Povolené služby** Toto nastavení potřebujete pouze, pokud chcete aby systém nabízel služby dostupné ze zóny, proti které je chráněn. Ve výchozím nastavení je systém chráněn pouze proti vnějším zónám. Explicitně povolte služby, které mají být počítačům ve vnější síti dostupné. Nejprve v nabídce 'Povolené služby pro vybranou zónu' zvolte zónu, pak přidejte služby, které pro ni mají být povoleny.

**Maškaráda** Maškaráda skrývá vnitřní síť před vnějšími sítěmi, jako je Internet, ale umožňuje počítačům z vnitřní sítě transparentně přistupovat k vnější síti. Požadavky z vnější do vnitřní sítě jsou zablokovány a požadavky z vnitřní sítě z vnějšku vypadají, jako by je vydával maškarádující server. Pokud mají být ve vnější síti dostupné služby stroje ve vnitřní síti, přidejte pro službu zvláštní přesměrovávací pravidlo.

**Broadcast** V tomto dialogu nastavte UDP porty, na kterých je povolen příjem broadcast paketů. K žádané zóně přidejte požadované porty nebo služby oddělené mezerami. Viz také `/etc/services`.

Lze zde také zapnout zaznamenávání nepřijatých broadcast paketů. To může působit problémy, neboť počítače s Windows generují velké množství broadcast paketů, kterým si o sobě dávají vědět. Jsou-li v síti takové počítače, mohou záznamy narůstat do velkých rozměrů.

**Podpora IPsec** V tomto dialogu nastavte, zda má být z vnější sítě dostupná služba IPsec. Po stisknutí tlačítka 'Podrobnosti' můžete nastavit, jak se má IPsec paketům důvěřovat.

**Úroveň logování** Existují dvě pravidla pro logování: zaznamenávání přijatých a nepřijatých paketů. Pro každou z obou skupin můžete zvolit mezi 'Zaznamenávat vše', 'Zaznamenávat pouze kritické' a 'Nezaznamenávat nic'.

Po ukončení konfigurace pokračujte stisknutím tlačítka 'Další'. Otevře se shrnutí nastavení konfigurace firewallu. V něm zkontrolujte všechna nastavení, služby, porty a protokoly, které byly povoleny. Chcete-li konfiguraci změnit, použijte tlačítko 'Zpět'. Tlačítkem 'Přijmout' konfiguraci uložíte.

## Ruční konfigurace

Následující odstavce popisují krok za krokem správný postup konfigurace. U každé konfigurační položky je uvedeno, zda je relevantní pro firewall nebo maškarádu. Nastavení týkající se DMZ (demilitarizované zóny) tu nejsou zmíněna. Jsou použitelná pouze ve složitějších sítích (obvykle podnikových), jejichž nastavení vyžaduje hlubokou znalost problematiky.

Nejprve pomocí editoru úrovní běhu YaST povolte `SUSEfirewall2` ve vámi používané úrovni (pravděpodobně 3 nebo 5). Tím se nastaví symbolické odkazy pro `SUSEfirewall2_*` skripty v adresářích `/etc/init.d/rc?.d/`.

**FW\_DEV\_EXT (firewall, maškaráda)** Zařízení připojené do Internetu. Pro modem vložte `ppp0`. Pro ISDN připojení použijte `ippp0`. Pro DSL připojení použijte `dsl0`. `auto` použijte pro rozhraní odpovídající výchozímu směrování.

**FW\_DEV\_INT (firewall, maškaráda)** Zařízení připojené k vnitřní, privátní síti (např. `eth0`). Pokud firewall chrání jen počítač, na kterém běží, a nikoliv vnitřní síť, ponechte prázdné.

**FW\_ROUTE (firewall, maškaráda)** Pokud chcete používat maškarádu, nastavte `yes`. Vnitřní počítače nebudou z vnější sítě viditelné, protože jejich privátní adresy (např. 192.168.x.x) nejsou v Internetu vůbec směrovány.

U firewallu bez maškarády nastavte `yes` pouze v případě, že chcete povolit přístup do vnitřní sítě. Pak ale musí mít počítače ve vnitřní síti platné IP adresy. V běžném případě byste neměli přístup zvenku povolovat!

**FW\_MASQUERADE (maškaráda)** Pokud potřebujete maškarádu, uveďte zde `yes`. Tím vnitřní počítače získají v podstatě přímý přístup k Internetu. Uvědomte si, že přistupovat k Internetu je bezpečnější skrze proxy. Maškaráda není pro službu poskytovanou proxy serverem potřeba.

**FW\_MASQ\_NETS (maškaráda)** Zde uveďte počítače či sítě, které budou maškarádovány. Jednotlivé položky odděluje mezerou, např.

```
FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
```

**FW\_PROTECT\_FROM\_INT (firewall)** Nastavením `yes` zabezpečíte váš firewall před vnitřní sítí. Pak je třeba služby z interní sítě explicitně povolovat. Viz `FW_SERVICES_INT_TCP` a `FW_SERVICES_INT_UDP`.

**FW\_SERVICES\_EXT\_TCP (firewall)** Zadejte TCP porty, které mají být přístupné. V případě domácí pracovní stanice, která nemá nabízet žádné služby, ponechte prázdné.

**FW\_SERVICES\_EXT\_UDP (firewall)** Ponechte prázdné, pokud ovšem neprovádíte UDP službu, která by měla být z venku přístupná. Mezi takové služby patří DNS, IPSec, TFTP, DHCP a další. V takovém případě vložte potřebné UDP porty.

**FW\_SERVICES\_INT\_TCP (firewall)** Tato proměnná určuje služby dostupné pro vnitřní síť. Zápis je stejný jako v případě `FW_SERVICES_EXT_TCP`, jen je nastavení použito pro *vnitřní síť*. Proměnnou je potřeba nastavit, pouze pokud je `FW_PROTECT_FROM_INT` nastavená na `yes`.

**FW\_SERVICES\_INT\_UDP (firewall)** Viz `FW_SERVICES_INT_TCP`.

Jakmile firewall nastavíte, otestujte ho. Sady pravidel vytvoříte jako uživatel `root` příkazem `SuSEfirewall12 start`. Pak se pokuste připojit telnetem z externího počítače, abyste ověřili, zda bude připojení skutečně odmítnuto. Následně si prohlédněte soubor `/var/log/messages`, ve kterém byste měli nalézt něco podobného:

```
Mar 15 13:21:38 linux kernel: SFW2-INext-DROP-DEFAULT IN=eth0
OUT= MAC=00:80:c8:94:c3:e7:00:a0:c9:4d:27:56:08:00 SRC=192.168.10.0
DST=192.168.10.1 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=15330 DF PROTO=TCP
SPT=48091 DPT=23 WINDOW=5840 RES=0x00 SYN URG=0
OPT (020405B40402080A061AFEB0000000001030300)
```

Mezi další balíčky, kterými můžete otestovat nastavení firewallu, patří `nmap` a `nessus`. Po jejich nainstalování k nim naleznete dokumentaci v adresářích `/usr/share/doc/packages/nmap` a `/usr/share/doc/packages/nessus-core`.

### 34.1.5 Další informace

Aktuální informace a další dokumentaci o balíčku `SuSEfirewall2` naleznete v `/usr/share/doc/packages/SuSEfirewall2`. Domovská stránka projektu `netfilter` a `iptables` na adrese <http://www.netfilter.org> poskytuje velké množství dokumentace v různých jazycích.

## 34.2 SSH: bezpečná práce v síti

V dnešní době, kdy je více a více počítačů instalovaných do prostředí sítě, je často nezbytné, aby se k nim dalo vzdáleně přistupovat. Obvykle to znamená, že se uživatel přihlásí – zašle přihlašovací jméno a heslo. Pokud jsou však tyto údaje zasílány přes síť jako prostý text, může se stát, že je cestou někdo odposlechne a získá přístup k účtu uživatele, aniž by o tom oprávněný uživatel věděl. Kromě toho, že útočník takto získá přístup k souborům uživatele, může se dostat i k účtu uživatele `root` nebo napadat další počítače. V minulosti se přihlašovalo na vzdálené počítače programem `telnet`, který nenabízí žádné bezpečnostní mechanismy pro utajení přenášených údajů. Podobné chování mají i další často používané programy pro vzdálený přístup, např. `ftp`.

SSH naproti tomu nabízí ochranu přenášených informací. Šifruje jak přihlašovací údaje (login a heslo), tak i veškerou další komunikaci mezi počítači. Útočník stále může odposlouchávat, ale bez znalosti šifrovacího klíče nemůže získat původní obsah zasílaných dat. SSH tedy umožňuje bezpečně komunikovat se vzdálenými systémy přes nezabezpečenou síť, jako je např. Internet. Sada programů, které se v systému SUSE LINUX starají o zabezpečení vzdáleného přístupu, se jmenuje `OpenSSH`.

### 34.2.1 Balíček OpenSSH

SUSE LINUX instaluje balíček OpenSSH automaticky. Programy ssh, scp a sftp jsou pak dostupné jako alternativa programů telnet, rlogin, rsh, rcp a ftp. Ve výchozím nastavení je síťový přístup k systému možný jen pomocí OpenSSH nástrojů a pouze v případě, že je povolen na firewallu.

### 34.2.2 Program ssh

Program ssh vám umožní připojovat se na vzdálené stroje a interaktivně pracovat. Nahrazuje telnet i rlogin. Program slogin je jen symbolický odkaz na ssh. Například na vzdálený počítač slunce se můžete přihlásit pomocí příkazu `sshslunce`. Vzdálený systém vás požádá o heslo (které máte nastavené na vzdáleném počítači slunce).

Po úspěšném přihlášení můžete pracovat s příkazovým řádkem na vzdáleném stroji, nebo spouštět interaktivní aplikace, např. YaST. Pokud máte na vzdáleném počítači nastavené jiné přihlašovací jméno než na lokálním počítači, můžete se přihlásit s použitím jiného přihlašovacího jména příkazem `ssh-l augustynka slunce` nebo `sshaugustynka@slunce`.

Navíc můžete pomocí ssh spouštět příkazy na vzdáleném systému, stejně jako s programem rsh. Na následujícím příkladě si ukážeme, jak spustit příkaz `uptime` na počítači slunce, a jak vytvořit adresář se jménem `tmp`. Výstup programů se zobrazí na terminálu lokálního počítače zeme.

```
ssh slunce "uptime; mkdir tmp"
tux@slunce's password:
1:21pm up 2:17, 9 users, load average: 0.15, 0.04, 0.02
```

Uvozovky jsou nezbytné, aby byly obě instrukce zaslány jedním příkazem. Jen tak se druhý příkaz spustí na počítači slunce.

### 34.2.3 Bezpečné kopírování pomocí scp

Program scp kopíruje soubory na vzdálený počítač. Je to bezpečná a šifrovaná náhrada za program rcp. Například příkaz `scpdopis.tex slunce` zkopíruje soubor `dopis.tex` z aktuálního adresáře lokálního počítače zeme na počítač slunce. Pokud máte na počítači slunce jiné uživatelské jméno než



na počítači zeme, zadejte uživatelské jméno pro vzdálený počítač ve formátu `username@host`. Pro tento příkaz neexistuje volba `-l`.

Po zadání správného hesla začne scp přenášet soubor a zobrazuje při tom stav přenosu jako rostoucí řadu hvězdiček. Navíc zobrazuje i odhadovaný čas trvání přenosu. Tyto výstupy můžete vypnout použitím parametru `-q`.

Program scp také zvládá rekursivní kopírování celých adresářů. Příkaz `scp -r src/ slunce:backup/` zkopíruje obsah adresáře `src/` včetně jeho podadresářů do adresáře `backup/` na počítači `slunce`. Pokud tento adresář neexistuje, scp ho automaticky vytvoří.

Parametrem `-p` řeknete scp, aby neměnil časové údaje u souborů. Volba `-C` zapne kompresi dat při přenosu, takže sníží velikost přenášených dat (zvýší se tím ale zatížení procesoru).

## 34.2.4 Bezpečný přenos souborů pomocí sftp

Program sftp lze použít místo scp pro bezpečný přenos souborů. Během sftp relace můžete používat některé z příkazů známých z ftp. Program sftp se hodí hlavně pro situace, kdy předem neznáte názvy souborů na vzdáleném počítači.

## 34.2.5 SSH démon (sshd) – strana serveru

Pro práci s SSH klienty ssh a scp musí v pozadí běžet SSH server (démon) naslouchající na TCP/IP portu 22. Démon při prvním spuštění generuje tři páry klíčů. Každý pár sestává ze soukromého a veřejného klíče. Proto se jedná o tzv. proceduru založenou na veřejném klíči. Aby byla zaručena bezpečnost komunikace pomocí SSH, musí mít přístup k soukromému klíči pouze administrátor systému. Ve standardní instalaci jsou přístupová práva k souborům podle toho nastavena. Soukromé klíče jsou potřebné pouze lokálně pro démona SSH a nesmíte je nikomu poskytnout. Veřejné části klíče (soubory s příponovou `.pub`) jsou zasílány klientům požadujícím spojení; mohou je číst všichni uživatelé.

Spojení je vždy iniciováno klientem. Čekající démon si s klientem nejdříve vymění identifikační data (zjistí jakou verzi protokolu, případně jaký program a port, používá protějščí strana). Protože na požadavek odpovídá potomek hlavního procesu démona SSH, může současně běžet více různých SSH spojení.

Pro komunikaci mezi serverem a klientem podporuje program OpenSSH verzi 1 i 2 protokolu SSH. Nově instalovaný systém SUSE LINUX používá standardně verzi 2. Pokud chcete u staršího systému po aktualizaci i nadále používat verzi

1, držte se instrukcí popsanych v souboru `/usr/share/doc/packages/openssh/README.SuSE`. V tomto dokumentu také najdete informace o tom, jak v několika krocích přejít z prostředí verze SSH 1 na verzi SSH 2.

Pokud používáte SSH verze 1, zasílá server svůj veřejný klíč stroje a klíč serveru, který je SSH démonem znovu vytvářen každou hodinu. Oba umožňují SSH klientovi zašifrovat libovolně zvolený klíč relace, který je zaslán SSH serveru. SSH klient také serveru oznámí, jaký šifrovací algoritmus používá.

Verze 2 protokolu SSH nevyžaduje klíč serveru. Obě strany používají pro výměnu klíčů algoritmus Diffie-Helman.

Pokud chcete rozšifrovat klíč relace, musíte znát soukromý klíč stroje i serveru, které nelze odvodit z veřejných klíčů. Pouze kontaktovaný SSH démon může rozšifrovat klíč relace pomocí svých soukromých klíčů (více viz `man /usr/share/doc/packages/openssh/RFC.nroff`). Počáteční fázi relace můžete podrobně sledovat, pokud zapnete u klienta SSH tzv. "užvaněný" režim volbou `-v`.

Výchozí je verze 2 SSH protokolu. Verzi 1 můžete vynutit přepínačem `-1`. Klient si po prvním kontaktu se serverem ukládá jeho veřejný klíč stroje do souboru `~/.ssh/known_hosts`. Tak se zabráni útokům cizích serverů s falešnými jmény a IP adresami (tzv. "man-in-the-middle" útok). Takový útok je odhalen buď díky klíči stroje nepřítomnému v `~/.ssh/known_hosts` nebo díky neschopnosti serveru rozšifrovat klíč relace kvůli tomu, že nemá odpovídající soukromé klíče.

Doporučujeme vám zálohovat na bezpečné místo veřejný i soukromý klíč (uloženy jsou v `/etc/ssh/`). Můžete tak odhalit manipulace s klíči, a pokud budete muset reinstalovat systém, můžete opět použít staré klíče. Tak ušetříte uživatele znepokojivých varování o změně klíče. Pokud se v případě varování o změně klíče ověří, že se skutečně jedná o správný SSH server, musí uživatel odstranit existující záznam o tomto serveru ze souboru `~/.ssh/known_hosts`.

## 34.2.6 Mechanismus ověřování pomocí SSH

Vlastní autentizace, v nejjednodušší formě, sestává z vložení hesla, jak bylo uvedeno výše. Cílem SSH bylo přinést snadno použitelný, ale bezpečný, software. Protože cílem je nahradit `rsh` a `rlogin`, SSH musí poskytovat autentizační metodu vhodnou pro každodenní použití. SSH toho dosahuje pomocí dalšího páru klíčů generovaného uživatelem. Balíček SSH k tomuto účelu obsahuje pomocný program `ssh-keygen`. Příkazem `ssh-keygen -t rsa` nebo `ssh-keygen -t dsa` se vygeneruje pár uživatelských klíčů a uživatel je dotázán, do jakého souboru se mají uložit.

Potvrďte standardní název a odpovězte na žádost o zadání hesla. I když vám program navrhne použít prázdné heslo, je lepší zadat netriviální heslo o délce 10 až 30 znaků. Potvrďte zopakováním hesla. Následně se uloží klíče do souborů, v našem příkladě do `id_rsa` (soukromý) a `id_rsa.pub` (veřejný) a program zobrazí celou cestu k souborům.

Pro změnu hesla u již vygenerovaných klíčů použijte (podle typu vašeho klíče) příkaz `ssh-keygen -p -t rsa` nebo `ssh-keygen -p -t dsa`. Nyní si na vzdáleném počítači, kam se chcete přihlašovat, uložte váš veřejný klíč (`id_rsa.pub`) do souboru `~/.ssh/authorized_keys`. Při přihlášení pak budete dotázáni na heslo ke klíči. Pokud se tak nestane, překontrolujte, zda jste vše správně uložili.

Tato procedura může vypadat složitěji, než samotné přihlašování pomocí přihlašovacího jména a hesla. SSH ale nabízí další nástroj, program `ssh-agent`, který si pamatuje privátní klíče během sezení. Celé sezení (X session) se musí spustit jako potomek programu `ssh-agent`. Nejjednodušší cestou je nastavit na začátku konfiguračního souboru `.xsession` proměnnou `usessh` na `yes` a přihlásit se přes KDM nebo XDM. Eventuálně spusťte X Window pomocí příkazu `ssh-agentstartx`.

Nyní můžete používat `ssh` nebo `scp` jako obvykle. Pokud jste uložili na vzdálené počítače váš veřejný klíč, nebude po vás systém vyžadovat heslo. Nezapomeňte ale, pokud odejdete od počítače, zamknout váš desktop (např. pomocí `xlock`).

Veškeré změny SSH protokolu 2 oproti dřívější verzi jsou popsány v souboru `/usr/share/doc/packages/openssh/README.SuSE`.

### 34.2.7 X server, ověřování a přeposílací mechanismy

Kromě vylepšení bezpečnostních mechanismů popsaných výše, SSH také zjednodušuje používání vzdálených aplikací pro X server. Jestliže spustíte `ssh` s parametrem `-X`, proměnná `DISPLAY` se na vzdáleném stroji nastaví na hodnotu počítače, odkud se přihlašujete, a veškerý výstup X aplikací bude přeposílán na vzdálený počítač přes existující `ssh` spojení. Navíc tyto aplikace spuštěné vzdáleně a zobrazované lokálně nemohou být díky přenosu přes `ssh` odposlechnuty útočníkem.

Pokud při spouštění přidáte parametr `-A`, bude se `ssh-agent` autentizační mechanismus přenášet i na stroje, na které se připojíte. Můžete se tedy bez zadávání hesel přihlašovat na další počítače. Stačí abyste všude uložili váš veřejný klíč.

Oba tyto mechanismy jsou standardně vypnuty, ale lze je kdykoliv zapnout v systémovém souboru `/etc/ssh/sshd_config` nebo v uživatelském souboru `~/.ssh/config`.

Program `ssh` můžete také použít pro přesměrování TCP/IP spojení. V následujícím příkladě SSH přesměruje SMTP a POP3 port:

```
ssh -L 25:slunce:25 zeme
```

Tedy každé SMTP spojení, které půjde na port 25 (SMTP) počítače `zeme`, je přes šifrovaný kanál přesměrováno na SMTP port počítače `slunce`. To se může hodit, pokud nepoužíváte SMTP server s funkcemi SMTP-AUTH nebo POP-before-SMTP. Z jakéhokoliv místa připojeného k síti lze veškerý poštovní provoz přesměrovat na hlavní poštovní server. Stejně tak lze přesměrovat POP3 spojení (port 110) z počítače `zeme` na počítač `slunce` pomocí příkazu:

```
ssh -L 110:slunce:110 zeme
```

Oba dva příkazy musíte spustit jako superuživatel `root`, protože jde o přesměrování privilegovaných portů. Elektronická pošta je normálními uživateli odesílána a přijímána pomocí existujícího SSH spojení. SMTP a POP3 host musí být nastaven na `localhost`. Další informace naleznete v manuálových stránkách k jednotlivým programům a v adresáři `/usr/share/doc/packages/openssh`.

## 34.3 Šifrování diskových oddílů a souborů

### 34.3.1 Vhodné nasazení

Každý uživatel má data, u kterých si přeje, aby k nim neměl přístup nikdo jiný. Čím více mobilní jste, tím opatrnější byste měli být při práci s daty. Při přímém nebo síťovém přístupu třetí strany k vašim datům je vždy vhodné řešení šifrování souborů. V následující části najdete popis nastavení šifrování a jeho možné použití v různých situacích.

**Notebooky** Pokud pracujete na cestách se svým notebookem nebo ho často převážíte z místa na místo, je vhodné šifrovat diskový oddíl s daty. V případě ztráty nebo krádeže notebooku jsou pak vaše data v bezpečí před nepovolanou osobou.

**Vyměnitelná média** U USB flash disku nebo externího disku je pravděpodobnost ztráty nebo krádeže mnohem pravděpodobnější než u celého notebooku. V takovém případě šifrování souborů uchrání vaše data před čtením nepovolanými osobami.

### 34.3.2 Nastavení šifrovaného souborového systému pomocí YaST

YaST nabízí možnost vytvoření šifrovaných souborů nebo diskových oddílů jak během instalace, tak na již nainstalovaném systému. Šifrované soubory lze bez problémů vytvářet bez ohledu na rozdělení disku. V případě šifrovaného diskového oddílu musíte nejdříve vytvořit příslušný diskový oddíl. Výchozí rozvržení rozdělení disku během instalace neobsahuje žádný šifrovaný diskový oddíl. Šifrovaný diskový oddíl je nutné vytvořit v rozdělování disku pro experty.

#### Vytvoření šifrovaného oddílu při instalaci

##### Varování

##### Zadání hesla

Věnujte pozornost zprávám systému o bezpečnosti hesla při zadávání hesla pro šifrovaný diskový oddíl. Heslo si dobře zapamatujte, bez jeho zadání se nedostanete k datům na šifrovaném diskovém oddíle.

##### Varování

Vytvoření šifrovaného diskového oddílu najdete v dialogu rozdělování disku programu YaST popsaném v části 2.8.11 na straně 68. Stejně jako při vytváření normálního diskového oddílu klikněte na tlačítko 'Vytvořit'. Pak zadejte parametry nového diskového oddílu (formátování a bod připojení). Dále pokračujte kliknutím na 'Krypt. souborový systém'. V následujícím dialogu zadejte heslo, které bude vyžadované před připojením šifrovaného diskového oddílu. Nastavení dokončíte kliknutím na tlačítko 'OK'. Systém vás požádá před připojením oddílu o zadání hesla pro připojení oddílu.

Pokud nechcete, aby byl šifrovaný diskový oddíl připojený během startu systému, stiskněte místo zadání hesla klávesu (Enter). Stejně postupujte u dalších požadavků o zadání hesla pro připojení diskového oddílu. Šifrovaný diskový oddíl nebude připojen a systém bude pokračovat ve startu. Jde o jeden ze způsobů ochrany vašich dat, protože po připojení šifrovaného diskového oddílu je obsah tohoto oddílu přístupný všem uživatelům.

Pokud chcete souborový systém připojovat pouze v případě jeho potřeby, zvolte 'Nepřipojovat při spuštění' v dialogu 'Volby fstab'. Diskový oddíl pak nebude automaticky připojován během startu systému. Kdykoliv ho pak můžete připojit příkazem `mount <jmeno_oddilu> <bod_pripojeni>`. Zadejte heslo. Aby k datům nemohli přistupovat další uživatelé, po ukončení práce odpojte diskový oddíl příkazem `umount jmeno_oddilu`.

## Vytvoření šifrovaného oddílu na běžícím systému

### Varování

#### Aktivace šifrování na běžícím systému

Šifrovaný diskový oddíl lze vytvořit také v již běžícím systému. Vytvoření šifrovaného oddílu na již existujícím oddílu povede ke ztrátě dat na zvoleném oddílu.

### Varování

Na běžícím systému zvolte v ovládacím centru programu YaST 'Systém' → 'Rozdělování disku'. Výběr dialogu potvrďte kliknutím na tlačítko 'Ano'. Místo tlačítka 'Vytvořit' použitého v předcházejícím nastavení klikněte na tlačítko 'Editovat'. Další postup je stejný.

### Šifrované soubory

Mimo šifrovaných oddílů je možné v dialogu rozdělování disku vytvářet šifrované soubory. Pod tabulkou diskových oddílů klikněte na tlačítko 'Vytvořit šifrovaný soubor' a zvolte 'Vytvořit šifrovaný soubor'. Zadejte cestu k souboru spolu s předpokládanou velikostí. Odsouhlaste výchozí nastavení pro formátování a typ souborového systému, zadejte bod připojení a nastavte, zda má být šifrovaný souborový systém připojen během startu systému.

## 34.3.3 Šifrování obsahu vyměnitelného média

Vyměnitelná média jako externí pevné disky a USB flash disky rozpoznává YaST jako normální pevný disk. Je tedy možné na nich šifrovat soubory nebo celé diskové oddíly stejným způsobem uvedeným výše. Protože k jejich připojení dochází obvykle pouze na omezenou dobu při práci, nenastavujte připojení těchto zařízení během startu systému.

## 34.4 Bezpečnost a soukromí

Jednou z hlavních vlastností linuxových a unixových systémů je schopnost obsluhovat více uživatelů najednou (víceuživatelský systém) a umožnit jim současně spouštět více úloh (multitasking). Navíc je tento operační systém sít'ově transparentní. Uživatelé často neví, zda data či aplikace, které používají, jsou umístěny lokálně na jejich počítači, nebo v síti.

Multiuživatelská podstata systému vyžaduje možnost oddělení dat jednotlivých uživatelů. Je nutno zajistit soukromí a bezpečí. Bezpečnost dat byla důležitým problémem již před vznikem počítačových sítí. Stejně jako dnes bylo vždy nejdůležitější zajistit bezpečnost dat v případě havárie nebo ztráty paměťového média, např. pevného disku.

Tato část se zabývá především otázkami soukromí, ale je nutno si uvědomit, že každá kvalitní bezpečnostní politika musí pamatovat na pravidelné, funkční a ověřené zálohy dat. Bez nich budete mít problém obnovit data nejen v případě havárie hardwaru, ale také v případě podezření na nedovolenou manipulaci s nimi.

### 34.4.1 Lokální a sít'ová bezpečnost

Existuje řada způsobů přístupu k datům:

- osobní komunikace s lidmi, kteří mají požadované informace nebo přístup k počítači
- přímý fyzický přístup k počítači
- přes sériovou linku
- přes počítačovou sít'

Ve všech těchto případech by se uživatel měl autentizovat dříve, než mu data budou zpřístupněna. Webový server nemusí být chráněn tak přísně, ale stále je nutné zajistit, aby neznámému uživateli neposkytl choulostivá data.

Ve výše uvedeném seznamu je první případ ten, který vyžaduje nejvíce komunikace mezi lidmi, jako např. tehdy, pokud kontaktujete zaměstnance banky a musíte ho přesvědčit, že bankovní účet je skutečně váš. Požádá vás o podpis, PIN nebo heslo, kterým si ověří vaši identitu. V některých případech se může podařit, na

základě několika málo známých skutečností a psychologie, získat důvěru informované osoby a postupně z ní vymámit další a další potřebné informace, aniž by si to vůbec uvědomila. Hackeři tuto techniku nazývají *sociální inženýrství*. Proti této technice se můžete zabezpečit jedinečně vzděláváním a školením svých zaměstnanců v užívání jazyka a komunikaci s lidmi. Před vlastním útokem na počítačové systémy se útočníci často snaží získat zajímavé informace od recepční, servisních techniků, nebo dokonce od rodinných příslušníků. V mnoha případech je útok založený na sociálním inženýrství odhalen příliš pozdě.

Útočník může použít i tradiční cestu a snažit se dostat přímo k vašemu hardwaru. Proto by počítače měly být chráněny proti nedovolené manipulaci, aby nikdo nepovolaný nemohl odstraňovat, vyměňovat nebo poškozovat jejich součásti. To platí i pro zálohy dat, síťové a elektrické kabely. Zabezpečte také startování systému, protože existuje několik dobře známých klávesových kombinací schopných vyvolat neobvyklé chování. Chraňte se použitím hesel pro BIOS i zavaděč systému.

Na mnoha místech se stále používají sériové terminály připojené k sériovým portům. Na rozdíl od síťových rozhraní nezávisí jejich komunikace s počítačem na síťovém protokolu. Používají jednoduchý kabel nebo infračervený paprsek, který přenáší informace v podobě nezašifrovaných znaků. Kabel je nejslabším článkem systému: lze k němu připojit starší tiskárnu a nastavit ji tak, aby tiskla veškerou přenášenou komunikaci. Místo tiskárny lze samozřejmě použít i jiné metody útoku.

Lokální čtení souboru na počítači vyžaduje jiná přístupová pravidla než otevření síťového spojení se serverem. Je rozdíl mezi lokální a síťovou bezpečností. Hranice je tam, kde se data musí balit do paketů, aby byla zaslána na jiné místo.

## **Lokální bezpečnost**

Lokální bezpečnost závisí na fyzickém prostředí, ve kterém počítač běží. Umístěte stroj v prostředí, které bezpečnostním požadavkům odpovídá. Hlavním cílem lokální bezpečnosti je zajistit, aby byli uživatelé odděleni a nemohli navzájem zneužívat svá práva a identity. To je obecné pravidlo, které je třeba mít na pozoru, ale nejdůležitější je v případě uživatele `root`, který má nad systémem absolutní kontrolu. Může totiž používat identitu kteréhokoliv dalšího uživatele, aniž by znal jeho heslo, a číst jakýkoliv lokálně uložený soubor.

## **Hesla**

Na Linuxu nejsou hesla ukládána jako text a uživatelem vložené heslo není jednoduše porovnáváno s heslem uloženým v systému. Kdyby tomu tak bylo, byly



by všechny účty v počítači kompromitovány v okamžiku, kdy by někdo nepovolaný získal přístup k patřičnému konfiguračnímu souboru. Místo toho je uložené heslo zašifrované a při každém vložení je zašifrováno znovu – porovnávají se pak dva zašifrované řetězce. V případě, že zašifrovaná hesla nelze převést zpět do původního tvaru, to významně zvyšuje bezpečnost.

Používá se k tomu speciální jednosměrný algoritmus, tzv. *trapdoor algorithm*. Útočník, i když by získal zašifrované heslo, není schopný algoritmus otočit a získat nezašifrovanou podobu hesla. Musel by testovat všechny možné kombinace písmen a dalších znaků, dokud by nenašel kombinaci, která při zašifrování dává stejný výsledek jako původní heslo. Pokud jsou hesla tvořena osmi znaky, je takových kombinací velmi mnoho.

V sedmdesátých letech se věřilo, že je tato metoda bezpečnější díky relativní pomalosti použitého algoritmu, který k zašifrování jednoho hesla vyžadoval několik sekund. Počítače se však natolik zrychlily, že dnes zvládnou podobných operací za sekundu milióny. Proto zašifrovaná hesla nesmějí být běžným uživatelům viditelná (běžní uživatelé nesmí mít možnost číst soubor `/etc/shadow`). Je také velmi důležité zajistit, aby hesla nebyla snadno uhodnutelná, pro případ že by se tento soubor v důsledku chyby stal viditelným. Není také příliš užitečné měnit heslo typu „tantalize“ na „t@nt@1lz3“.

Záměna některých písmen za podobné znaky není dostatečně bezpečná, protože programy pro odhalování hesel používající slovníky umí provádět i podobné záměny. Lepší je použít slovo bez obecného významu, něco, co dává smysl jen vám osobně. Například první písmena slov nějaké věty nebo názvu knihy, například *Kniha Jméno růže, kterou napsal Umberto Eco* by vedla k bezpečnému heslu *KJrk-nUE8*. Hesla typu *cernakocka* nebo *zuzana76* může snadno uhádnout i někdo, kdo vás téměř nezná.

## Start systému

Systém nastavte tak, aby nemohl být spuštěn z diskety nebo CD. Bud' mechaniky úplně odstraňte, nebo nastavte BIOS tak, aby spouštěl systém výhradně z pevného disku, a zajistěte BIOS heslem. Linux je obvykle spouštěn zavaděčem, který umožňuje jádru předávat různé parametry. Zakažte ostatním tyto parametry používat nastavením dalšího hesla v souboru `/boot/grub/menu.lst` (viz 8 na straně 163). Je to pro bezpečnost systému velmi důležité, protože jádro samotné běží s pravomocemi uživatele `root` a navíc je to právě jádro, kdo tyto pravomoci dále přiděluje.

## Souborová přístupová práva

Obecným pravidlem je pracovat vždy s nejpřísnějšími možnými nastaveními práv, které umožňují vykonat potřebný úkol. Například pro čtení a psaní pošty rozhodně nejsou potřeba práva uživatele `root`. Pokud by v poštovním programu byla chyba, mohla by být zneužita k útoku, který by měl přesně ta práva, jako měl program při svém spuštění. Výše zmíněné pravidlo pomáhá minimalizovat škody v podobných případech.

Práva téměř čtvrt miliónu souborů obsažených v systému SUSE LINUX jsou velmi pečlivě zvolena. Administrátor by při instalaci dodatečných souborů a programů měl dávat na nastavení práv velký pozor. Zkušený a bezpečnostních pravidel znalý administrátor vždy používá spolu s příkazem `ls` volbu `-l`, což jim umožňuje okamžitě odhalit špatně nastavená přístupová práva. Špatně nastavená práva souboru mohou vést nejen ke změně či smazání souboru, ale mohou být spuštěny s právy superuživatele, nebo, v případě konfiguračních souborů, programy je mohou použít s právy superuživatele. To významně zvyšuje možnosti útočníka. Tento typ útoku se nazývá "kukaččí vejce", neboť je program spuštěn ("vysezen") jiným uživatelem ("ptákem"), podobně jako když kukačka oklame jiné ptáčky a donutí je tak starat se o svou snůšku.

SUSE LINUX obsahuje v adresáři `/etc` soubory `permissions`, `permissions.easy`, `permissions.secure` a `permissions.paranoid`. Smyslem těchto souborů je definovat zvláštní práva, jako adresáře, do kterých může zapisovat kdokoliv, nebo, v případě souborů, `setuser` ID bit (programy s nastaveným `setuser` ID bitem neběží pod uživatelem, který je spustil, nýbrž s právy vlastníka souboru, nejčastěji uživatele `root`). Administrátor může přidávat vlastní nastavení do souboru `/etc/permissions.local`.

Který z těchto souborů se bude používat konfiguračními programy nastavíte pomocí 'Nastavení bezpečnosti' nástroje YaST. Více se dozvíte v komentářích souboru `/etc/permissions` nebo v manuálové stránce příkazu `chmod`.

## Přetečení zásobníku a chyby typu `format string`

Vždy, když program zpracovává data, která mohla být změněna uživatelem, je třeba být na pozoru. Je to však spíše problém programátorů než běžných uživatelů. Programátor musí zajistit, aby aplikace zpracovávala data správným způsobem, aniž by zapisovala do paměťových oblastí, které jsou pro data příliš malé. Program by měl také předávat data konzistentním způsobem přes k tomu určená rozhraní.

K *přetečení zásobníku* (buffer overflow) může dojít tehdy, pokud se při zápisu do paměťového zásobníku nevezme v úvahu jeho velikost. V určitých případech

data (vytvořená uživatelem) zabírají více místa, než zásobník obsahuje. Důsledkem je, že jsou zapsána za hranici zásobníku. To může za určitých okolností znamenat vykonání instrukcí zadaných uživatelem (nikoliv programátorem) místo pouhého zpracování dat. Chyba tohoto typu může mít velmi závažné následky, zvláště pokud je program spuštěn se zvláštními právy (viz 34.4.1 na předchozí straně).

Chyby typu *format string* fungují trochu jinak, ale následky jsou podobné. Ve většině případů se tyto chyby zneužívají v programech, které běží se zvláštními právy (setuid a setgid), což ovšem také znamená, že se můžete chránit odebráním těchto práv. Nejlepší je aplikovat pravidlo o použití nejnižších možných oprávnění (viz 34.4.1 na předchozí straně).

Protože se tyto chyby týkají zpracování uživatelských dat, lze je zneužívat bez přístupu k lokálnímu účtu. Často je lze zneužívat i po síti. Proto jsou důležité z hlediska místní i síťové bezpečnosti.

## Viry

Ačkoliv někteří lidé říkají opak, viry existují i na Linuxu. Nicméně známé linuxové viry jsou pouze pokusné laboratorní exempláře vyvinuté jako důkaz jejich možné existence. V divoké přírodě nikdo nikdy žádné linuxové viry nespatriil.

Viry nemohou přežít a šířit se bez hostitele. Takovým hostitelem může být program nebo důležitý datový prostor, např. MBR disku, který musí být pro kód viru zapisovatelný. Vzhledem ke své multiuživatelské podstatě může Linux omezit práva zápisu k určitým souborům, zejména důležitým systémovým souborům. Proto zvyšujete pravděpodobnost napadení virem, pokud provádíte běžnou práci jako uživatel `root`. Naproti tomu, pokud používáte zmíněné pravidlo o nejnižších možných oprávněních, je pravděpodobnost infekce zanedbatelná.

Mimo to byste nikdy neměli bezhlavě spouštět program z neznámého internetového zdroje. SUSE RPM balíčky obsahují digitální podpis potvrzující jejich původ. Virová infekce je typickým příznakem administrátorů a uživatelů s nízkým povědomím o bezpečnosti. Takoví dokáží ohrozit i systém, který byl navržen jako vysoce bezpečný.

Nezaměňujte viry s červy. Červi jsou čistě síťové potvůrky, které nevyžadují pro své šíření hostitele.

## Síťová bezpečnost

Síťová bezpečnost je důležitá pro ochranu proti útokům pocházejícím z vnější. Běžná přihlašovací procedura zahrnující dotaz na uživatelské jméno a heslo je

stále místní bezpečnostní záležitost. V případě přihlašování po síti je nutno rozlišit mezi dvěma bezpečnostními aspekty. To, co se odehrává před vlastním přihlášením, je záležitost sít'ové bezpečnosti, to co se děje po vlastním přihlášení, je záležitost lokální bezpečnosti.

## X Window System a X autentizace

Jak bylo zmíněno na začátku, je sít'ová transparentnost jednou z hlavních charakteristik unixových systémů. X, okenní systém unixových systémů, toho umí využívat úžasným způsobem. Při použití X není problém přihlásit se na vzdálený stroj a spustit tam grafický program, jehož výstup je zasílán přes síť zpět k vám a zobrazen na vašem počítači.

Pokud se má X klient vzdáleně zobrazit, musí X server chránit zdroje (obrazovku) před neoprávněným přístupem. Klientská aplikace musí dostat určitá práva. Systém X Window to umí zařídit dvěma způsoby. První se nazývá kontrola přístupu na straně hosta (host-based access control), druhou je kontrola přístupu pomocí cookies (cookie-based access control). První spoléhá na IP adresu počítače, ze kterého běží klient, a je ovládána programem xhost. Program xhost uloží IP adresu klienta do malé databáze X serveru. Spoléhání na IP adresu však není nijak zvlášť bezpečné. Na počítači navíc může pracovat další uživatel, který může prvnímu uživateli ukrást přístup k X serveru. Z důvodů nízké bezpečnosti zde proto tuto metodu nebudeme popisovat. Pokud se s ní přesto chcete blíže seznámit, najdete informace v manuálové stránce xhost.

V případě kontroly pomocí cookies se generuje řetězec, který zná pouze X server a správný uživatel. Jde o něco jako občanský průkaz. Koláček (slovo se nevztahuje k obyčejným koláčkům, ale k čínským koláčkům pro štěstí, na kterých je epigram) je při přihlášení uložen v souboru `.Xauthority` v domovském adresáři uživatele a je dostupný všem X klientům vyžadujícím X server pro zobrazení okna. Soubor `.Xauthority` lze otestovat programem `xauth`. V případě přejmenování souboru `.Xauthority` nebo jeho smazání není možné otevřít žádné nové okno nebo X klienta. Více se o bezpečnostních mechanismech X Window dovíte v manuálové stránce `Xsecurity` (`manXsecurity`).

SSH (secure shell) lze využít ke kompletnímu šifrování sít'ového připojení k X serveru, aniž by to uživatel pocítil. Tomuto přeposílání se říká X forwarding. Jde o simulaci X serveru na straně serveru a nastavení příslušné proměnné na straně vzdáleného klienta. Další podrobnosti o SSH najdete v části 34.2 na straně 559.

## Varování

Pokud si nejste jistí bezpečností počítače, na kterém pracujete, nepoužívejte X forwarding. Pokud ho přesto aktivujete, může případný útočník například zneužít vaše SSH připojení k napadení X serveru a odposlechu klávesnice.

## Varování

### Přetečení zásobníku a chyby typu "format string"

Jak bylo vysvětleno v části 34.4.1 na straně 570, přetečení zásobníku i chyby typu "format string" jsou záležitosti místní i síťové bezpečnosti. Stejně jako v případě lokálních útoků, i zde jsou tyto chyby nejčastěji zneužívány k získání pravomocí superuživatele. I když se nepodaří přímo toto, může útočník získat nepriviligovaný lokální účet a ten použít ke zneužívání dalších potenciálních bezpečnostních slabín systému.

Přetečení zásobníku a chyby typu "format string" zneužitelné po síti jsou nepochybně nejčastějším typem vzdálených útoků. Programy zneužívající nově nalezených chyb tohoto typu (tzv. exploits) se často distribuují v konferencích věnovaných bezpečnosti. Lze je použít bez znalosti vlastního kódu. Během let se ukázalo, že jejich dostupnost vede k bezpečnějším systémům, prostě proto, že výrobci operačních systémů byli donuceni se bezpečností zabývat. V případě svobodného softwaru má ke zdrojovým kódům přístup kdokoli (SUSE LINUX je dodáván s kompletními zdrojovými kódy) a tak může kdokoli, kdo našel bezpečnostní chybu, dodat patřičnou záplatu.

### Zahlcení (Denial of Service)

Smyslem útoků typu zahlcení (Denial of Service – DoS) je zablokovat serverový program nebo dokonce celý systém, čehož lze dosáhnout několika způsoby: přetížením serveru, jeho zaměstnáním nesmyslnými pakety nebo zneužitím přetečení zásobníku. DoS útok má často jediný účel – zlikvidovat určitou službu v síti. Zmizení služeb může znamenat další ohrožení, například útoky typu *man-in-the-middle* (odposlech, přebírání TCP spojení, předstírání adresy) či otravu DNS.

### Útoky typu "Man-in-the-Middle"

Každý vzdálený útok, při kterém se útočník vplete *mezi* dva komunikující počítače, se řadí mezi tzv. *man-in-the-middle* útoky. Většina útoků toho to typu má

jedno společné – oběť vůbec netuší, že se děje něco zlého. Existuje mnoho různých variant těchto útoků, útočník například může zachytit požadavek na spojení a přeposlat ho cílovému stroji. Oběť se tak spojí s nežádoucím protějškem, ale nic netuší, protože ten předstírá skutečný cíl spojení.

Nejjednodušší forma takového útoku je tzv. *sniffing*, při kterém útočník jen odposlouchává okolní síťový provoz. Složitější "man-in-the-middle" útok může znamenat převzetí již existujícího spojení (*hijacking*). Aby tak mohl útočník učinit, musí po nějakou dobu analyzovat pakety, aby mohl předpovědět TCP sekvenci daného spojení. V okamžiku, kdy útočník spojení převzme, si oběť problému všimne, protože se její spojení ukončí a dostane chybové hlášení. Skutečnost, že existují protokoly nezabezpečené šifrováním, útočníkům život jenom usnadňuje.

*Spoofing* je útok, při kterém jsou pakety pozměněny, aby obsahovaly falešná data, obvykle IP adresu. Většina aktivních útoků závisí na možnosti zaslání takových falešných paketů, což je něco, co na linuxovém stroji může udělat pouze superuživatel (`root`).

Mnoho zmíněných útoků se kombinuje s útoky typu DoS. Pokud má útočník možnost určitý počítač, byť na krátkou dobu, vyřadit z provozu, usnadňuje to aktivní útok, protože počítač nebude schopný s útokem po určitou dobu interferovat.

## Otrava DNS

Otrava DNS (poisoning) nastává, když útočník pozmění cache DNS serveru [pomocí podvržených DNS paketů], který pak předává informace oběti, která je vyžaduje. Mnoho serverů udržuje s dalšími počítači důvěrné vztahy na základě IP adres nebo jmen. Útočník těmto důvěrným vztahům musí dobře porozumět, aby se mohl vydávat za jeden z důvěryhodných strojů. Obvykle proto útočník analyzuje pakety ze serveru. Současně často musí použít dobře načasovaný DoS útok. Obranou je zde použití šifrovaných spojení a ověřování identity počítačů, s nimiž je navazováno spojení.

## Červi

Červi jsou často zaměňováni s viry, rozdíl mezi nimi je však jasný. Na rozdíl od virů není červí život závislý na hostiteli. Místo toho se specializují na co možná nejrychlejší šíření sítí. Červi, kteří se v minulosti objevili, jako Ramen, Lion či Adore, využívali dobře známých bezpečnostních děr v programech jako `bind8` nebo `lprNG`. Ochrana proti červům je poměrně snadná. Protože mezi objevem bezpečnostní díry a výskytem červů uplyne nějaký čas, může mít svědomitý administrátor dávno instalovány potřebné bezpečnostní opravy.

### 34.4.2 Bezpečnostní tipy a triky

Je velmi důležité být informován o novinkách na poli bezpečnosti a o nejnovějších bezpečnostních problémech. Jedním z nespolehlivějších způsobů ochrany systému je instalace všech aktualizací balíčků, které bezpečnostní zprávy doporučují. Bezpečnostní oznámení pro SUSE jsou publikována v poštovní konferenci, do které se můžete přihlásit na adrese <http://www.novell.com/linux/security/securitysupport.html>. Tato konference ([suse-security-announce@suse.de](mailto:suse-security-announce@suse.de)) je skvělým zdrojem informací o bezpečnostních aktualizacích a mezi její aktivní členy patří řada odborníků z bezpečnostního týmu SUSE.

Poštovní konference [suse-security@suse.de](mailto:suse-security@suse.de) je dobrým místem pro diskusi o bezpečnostních problémech, které vás zajímají. Můžete se do ní přihlásit na výše uvedené webové stránce.

[bugtraq@securityfocus.com](mailto:bugtraq@securityfocus.com) je jedna z nejznámějších bezpečnostních konferencí na světě. Doporučujeme její sledování. Denně je v jejím rámci publikováno přibližně 15 až 20 příspěvků. Více informací najdete na stránce <http://www.securityfocus.com>.

Následující základní bezpečnostní pravidla vám mohou přijít vhod:

- V souladu s pravidlem o použití nejprísnějších omezení práv, jaká ještě umožňují provést požadovanou úlohu, neprovádějte svou běžnou činnost jako superuživatel `root`. Snížíte tak pravděpodobnost infekce virem nebo kukaččím vejcem. Také ochráníte sami sebe před vlastními chybami.
- Pokud je to možné, pracujte na vzdáleném stroji s využitím šifrovaného spojení. Používání `ssh` (secure shell) místo programů `telnet`, `ftp`, `rsh` a `rlogin` by mělo být samozřejmostí.
- Vyhněte se autentizačním metodám založeným pouze na IP adrese.
- Snažte se udržovat nejdůležitější balíčky spojené se sítí aktuální. Přihlaste se do konferencí, které vás budou informovat o potřebných aktualizacích takových programů (`bind`, `sendmail`, `ssh` atd.). Totéž platí pro programy ovlivňující lokální bezpečnost.
- Optimalizujte soubor `/etc/permissions` podle potřeb vašeho systému. Pokud odeberete program `setuid` bit, může přestat správně pracovat. Na druhou stranu přestane být potenciální bezpečnostní dírou do vašeho systému. Podobně je tomu se soubory a adresáři, do kterých může každý zapisovat.

- Zakažte všechny síťové služby, které na serveru nutně nepotřebujete. Tím zvýšíte bezpečnost systému. Porty, na kterých se naslouchá, lze vyhledat programem `netstat`. Používejte ho spolu s následujícími volbami: `netstat -ap` nebo `netstat -anp`. Volba `-p` zobrazí, který proces obsadil který port pod jakým jménem.

Výstup programu `netstat` porovnejte s pečlivým skenem portů provedeným z jiného počítače. Vynikající program k tomuto účelu je `nmap`, který nejen prověří porty na vašem stroji, ale dokáže odhadnout, jaké služby za nimi čekají. Skenování portů však může být považováno za nepřátelský akt, proto nikdy nic takového nedělejte na počítači bez výslovného souhlasu administrátora. Uvědomte si také, že je nutné oskenovat nejen TCP, ale i UDP porty (volby `-sS` a `-sU`).

- Program `tripwire` umožňuje monitorovat integritu souborů na vašem systému. Tento program je součástí distribuce SUSE LINUX. Databázi tohoto programu zašifrujte, aby s ní nikdo nepovolaný nemohl provádět psí kusy. Navíc si její kopii uložte mimo počítač na externí datové médium, které není připojené přes síť.

- Mějte se na pozoru při instalaci softwaru třetích stran. Byly případy, kdy útočník vložil trojského koně do balíčku s bezpečnostní aktualizací programu. Naštěstí byl brzy odhalen. Pokud instalujete binární balíček, nesmíte mít nejmenší pochybnosti o jeho původu.

RPM balíčky distribuce SUSE jsou elektronicky podepsány (gpg). Klíč, který SUSE k podepisování používá, je následující:

ID:9C800ACA 2000-10-19 SUSE Package Signing Key <build@suse.de>

Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA

Příkazem `rpm --checksig balicek.rpm` můžete zkontrolovat kontrolní součet a podpis nenainstalovaného balíčku. Klíč naleznete na prvním instalačním CD i na většině světových klíčových serverů.

- Pravidelně kontrolujte zálohy vašich uživatelských i systémových souborů. Pokud nemáte zálohy ověřené, mohou být nepoužitelné a bezcenné.
- Kontrolujte záznamy v protokolových souborech (logy). Pokud je to možné, napište si malý skript hledající podezřelé záznamy. Není to jednoduchá úloha, ale jen vy můžete vědět, které záznamy jsou ve vašem systému podezřelé.



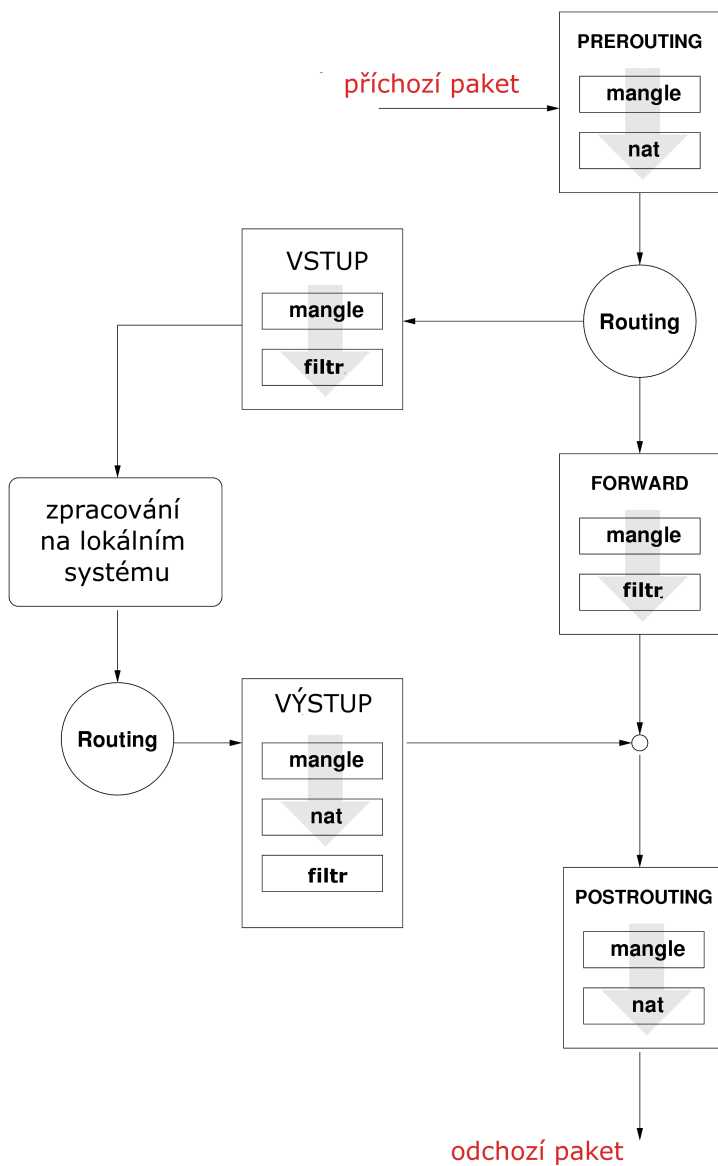
- Pomocí `tcp_wrapper` omezte přístup ke službám běžícím na vašem stroji, takže získáte kontrolu nad tím, kterým IP adresám je povolen přístup ke službě. Více informací o tomto nástroji najdete v manuálových stránkách `tcpd` a `hosts_access` (`man 8 tcpd` a `man hosts_access`).
- Pro zvýšení bezpečnosti `tcpd` (`tcp_wrapper`) použijte `SuSEfirewall`.
- Bezpečnostní opatření by měla být vícenásobná: dvakrát zobrazená zpráva je lepší než žádná zpráva.

### 34.4.3 Ústřední adresa pro hlášení bezpečnostních problémů

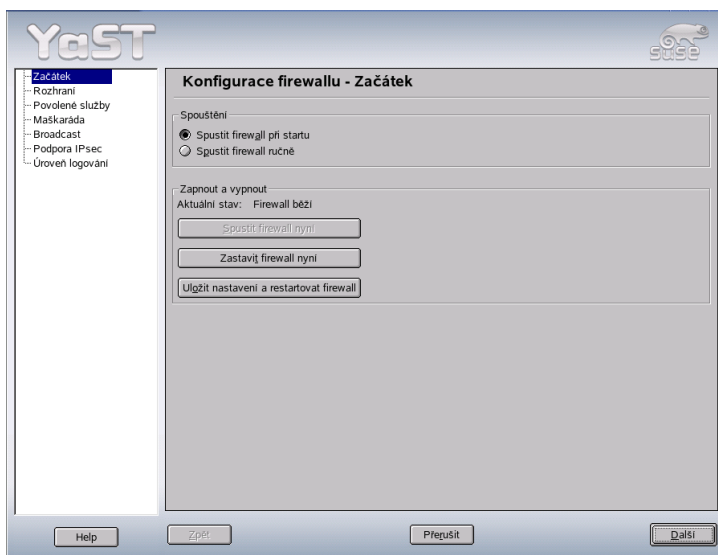
Pokud objevíte bezpečnostní problém (nejprve prosím zkontrolujte dostupné aktualizace), napište e-mail na adresu `security@suse.de`. Přiložte prosím podrobný popis problému a číslo verze postiženého balíčku. SUSE vám odpoví co nejrychleji. Doporučujeme poštu šifrovat pomocí `pgp`. Klíč SUSE je:

```
ID:3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>  
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

Tento klíč si můžete stáhnout i ze stránky <http://www.novell.com/linux/security/securitysupport.html>.



**Obrázek 34.1:** *iptables: Možné cesty paketu*



*Obrázek 34.2: YaST: Konfigurace firewallu*



# ACLs v Linuxu

V této kapitole je popsáno pozadí a funkce POSIX ACLs pro linuxové souborové systémy. Zároveň zde získáte informace o používání a výhodách ACLs (*Access Control Lists*).

35.1	Výhody ACLs . . . . .	582
35.2	Definice . . . . .	582
35.3	Používání ACLs . . . . .	583
35.4	Výhledy . . . . .	592
35.5	Další informace . . . . .	592

## 35.1 Výhody ACLs

V tradičním linuxovém systému má každý objekt tři typy přístupových práv. Jde o práva ke čtení (r, zápisu w a vykonání x) pro každý ze tří typů uživatelů (vlastníka, skupinu a ostatní). Navíc lze nastavit *user id*, *group id* a *sticky* bit.

Toto pojetí je zcela dostačující v naprosté většině situací. Ve velmi rozsáhlých systémech a zvláštních typech aplikací však naráží na řadu limitů.

ACLs vznikly právě proto, aby tyto situace ošetřily rozšířením tradičního pojetí přístupových práv o další vlastnosti. Pomocí ACLs je možné možné nastavit přístupová práva pouze pro určité uživatele nebo skupiny, kteří nejsou vlastníky objektu ani nepatří do příslušné skupiny. Access Control Lists jsou součástí jádra a mají podporu v souborových systémech ReiserFS, Ext2, Ext3, JFS a XFS. Díky ACLs můžete nastavovat přístupová práva, aniž byste museli zároveň zasahovat do celého systému přístupových práv.

Výhody ACLs si uvědomíte především při náhradě serveru s Windows za server s Linuxem. Řada stanic v síti může pracovat se systémem Windows i po migraci a systém Linux bude těmto stanicím poskytovat tiskové a souborové služby pomocí Samby.

Díky podpoře ACLs v Sambě lze práva nastavit jak na linuxovém serveru tak na stanicích Windows (pouze Windows NT a vyšší). Pomocí programu winbindd lze nastavovat práva uživatelů, kteří existují pouze na straně Windows a na linuxovém serveru nemají účet. Access Control Lists je nastaven pomocí getfacl a setfacl pouze na straně serveru.

## 35.2 Definice

**Třídy uživatelů** Tradiční koncept POSIX používá v souborovém systému tři *třídy* přístupových práv. Vlastníka, skupinu vlastníka a ostatní. Pro každou z těchto tří tříd lze nastavit bity dávající práva ke čtení (r, zápisu w a vykonání x).

**Přístupové ACLs** Přístupová práva skupin a uživatelů jsou pro všechny typy objektů souborového systému (soubory a adresáře) omezeny přístupovými ACLs.

**Výchozí ACL** Výchozí ACLs se nastavuje pouze u adresářů. Omezuje nastavení přístupových práv u nově vytvářených podadresářů a souborů.

**Položka ACL** Každý ACLs se skládá ze skupiny položek. ACLs položky se skládají z typu (viz. tabulka 35.1 na následující straně), ukazatelem na skupinu nebo uživatele a nastavením práv. Pro některé typy položek musí být ukazatel na skupinu nebo uživatele prázdný.

## 35.3 Používání ACLs

V následující části si na příkladech ukážeme používání ACLs a jejich interakci s tradičním systémem přístupových práv. Popíšeme postup pro vytvoření vlastních ACLs a také syntaxi ACLs.

ACLs dělíme na dva základní typy. *Minimální* ACLs obsahují položku pro typ uživatele (owner), skupinu vlastníka (owner group) a ostatní (other) s konvenčními přístupovými bity pro soubory a adresáře. *Rozšířené* ACLs jde ještě dál. Musí obsahovat nastavení položky *mask* a musí obsahovat více položek pro typy *named user* a *named group*. V tabulce 35.1 na následující straně najdete přehled různých typů možných ACLs položek.

**Tabulka 35.1: Typy ACL položek**

Typ	Zápis
owner	user::rwx
named user	user:name:rwx
owning group	group::rwx
named group	group:name:rwx
mask	mask::rwx
other	other::rwx

Práva definována v položce *owner* a *other* jsou vždy platná. S vy jímku položky *mask* všechny ostatní položky (*named user*, *owning group*, a *named group*) mohou být neaktivní nebo maskované. Platné jsou v případě, že jsou součástí jak určité položky, tak masky. Pokud jsou pouze součástí masky, jsou neaktivní. Tento mechanismus je demonstrován v tabulce 35.2 na této straně.

**Tabulka 35.2: Maskování práv**

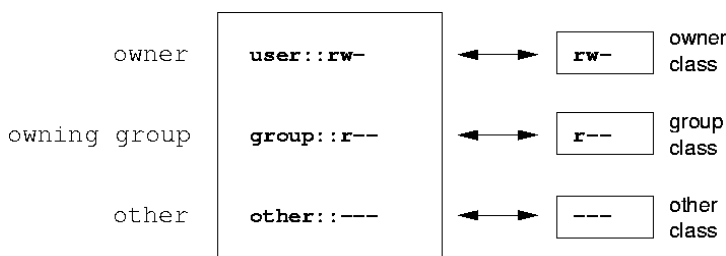
Typ položky	Zápis	Práva
named user	user:jane:r-x	r-x
mask	mask::rw-	rw-
	effective permissions:	r--

### 35.3.1 ACL položky a přístupové bity

V systému s ACLs existují minimální a rozšířené ACLs, první jsou znázorněny na obrázku 35.1 na následující straně, druhé na 35.2 na následující straně. V následujících příkladech si ukážeme dva případy minimálních a rozšířených ACLs.

V obou případech jsou práva *třídy owner* mapována na ACL položky *owner*. Stejně tak jsou na příslušnou položku mapována také práva *třídu other*. V obou případech je však jiné mapování na *třídu group*.





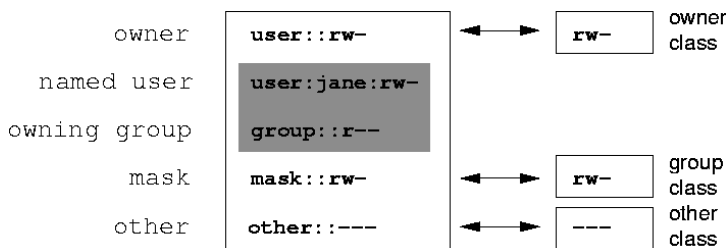
*Obrázek 35.1: Minimální ACL: ACL úpoložky porovnávány podle přístupového bitu*

#### V případě minimálních ACLs bez masky

jsou práva třídy *group* mapována na ACLs položku *owning group*.

#### V případě rozšířených ACLs s maskou

jsou práva třídy *group* mapována na položku *mask*.



*Obrázek 35.2: Rozšířené ACL: ACL položky porovnávány podle přístupového bitu*

Mapování zajišťuje hladký chod aplikací s podporou ACLs spolu s aplikacemi bez této podpory. Práva zde nezmiňovaná buď nejsou nastavena pomocí ACLs nebo jsou neaktivní. Pokud dojde ke změně přístupových bitů, dojde ke změně ACLs a vice versa.

### 35.3.2 Adresář s ACL přístupem

Princip přístupu ACLs je znázorněn v následujícím příkladě:

- Vytvoření objektu souborového systému (v našem případě adresáře)
- Změna ACL
- Maskování

1. Před vytvořením adresáře použijte příkaz `umask` k nastavení výchozích práv:

```
umask 027
```

Příkaz `umask 027` nastaví výchozí přístupová práva tak, že vlastníkově dá všechna práva (0, skupině zakáže zápis 2 a ostatním nedá práva žádná 7). `umask` zároveň maskuje všechny přístupové bity a deaktivuje je. Více informací o tomto příkazu získáte z jeho manuálových stránek (`man umask`).

Zdejte příkaz `mkdir`. Výsledkem je vytvoření adresáře `mydir` s přístupovými právy nastavenými prostřednictvím `umask`. Následujícím příkazem překontrolujete, zda jsou práva nastavena správně:

```
ls -dl mydir
```

```
drwxr-x- ... tux project3 ... mydir
```

2. Zjistěte počáteční nastavení ACL a vložte nové hodnoty pro uživatele a skupiny.

```
getfacl mydir
```

```
user::rwx
group::r-x
other::---
```

Výstup příkazu `getfacl` velmi jasně ukazuje nastavení bitů a ACL položek popsaných v části 35.3.1 na straně 584. První tři řádky zobrazují jméno adresáře, vlastníka a jeho skupinu. Následující tři řádky obsahují ACL položky *owner*, *owning group* a *other*. V tomto případě má adresář minimální ACL nastavení a pomocí příkazu `getfacl` jsme získali stejný výpis jako v případě použití prostého `ls`.

V první změně ACL přidáme práva pro čtení, zápis a vykonání pro dalšího uživatele se jménem `jane` a další skupiny `djungle`.

```
setfacl -m
```

```
user:jane:rwx,group:djungle:rwx mydir
```

Parametrem `-m` příkazu `setfacl` říkáme, že má změnit ACLs. Parametr je následován hodnotami (jednotlivé položky jsou odděleny dvojtečkami). Poslední částí příkazu je jméno adresáře, na který se mají změny aplikovat.

Příkazem `getfacl` si můžete nechat vypsat výsledné nastavení ACLs.

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:djungle:rwx
mask::rwx
other:---
```

Jako další nastavení pro uživatele `jane` a skupinu `djungle` byla vytvořena položka `mask`. Tato položka automaticky redukuje všechny položky v *třídě* `group` na společný základ.

*Maska* definuje maximální efektivní přístupová práva pro všechny položky v *třídě* `group`. To obsahuje *named user*, *named group* a *owning group*. Přístupové bity *třídě* `group` lze zobrazit příkazem `ls -dl mydir`.

```
ls -dl mydir
```

```
drwxrwx- ... tux project3 ... mydir
```

První sloupec mimo obvyklého výstupu obsahuje také `+`, který indikuje existenci *rozšířených* ACLs.

- Podle výstupu příkazu `ls` obsahuje položka `mask` práva k zápisu. V tradičním pojetí by to znamenalo, že má *vlastnická skupina* (zde `project3`) také práva zápisu do adresáře `mydir`. Přístupová práva *vlastnické skupiny* však souhlasí s nastavením v *mask*, které jsou v našem příkladě `r-x` (viz. tabulka 35.2 na straně 584). Dodatečné nastavení tak nebude mít na dosavadní nastavení žádný vliv.

Editujte položku *mask* příkazem `setfacl` nebo `chmod`.

```
chmod g-w mydir
```

```
ls -dl mydir
```

```
drwxr-x---+ ... tux project3 ... mydir
```

```
getfacl mydir
```

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx          # effective: r-x
group::r-x
group:djungle:rwx      # effective: r-x
mask::r-x
other:---
```

Po vykonání příkazu `chmod` bude odstraněn bit pro zápis z *třídy group* a výstup příkazu `ls` ukazuje, že musí být změněn i bity masky. Práva zápisu jsou opět omezeny pouze na vlastníka adresáře `mydir`. Výstup příkazu `getfacl` tuto skutečnost potvrzuje. Výstup obsahuje komentář pro všechny položky, kde přístupové bity nesouhlasí s originálním nastavením, protože jsou filtrovány pomocí položky *mask*. Původní nastavení lze kdykoliv vrátit příkazem `chmod`:

```
chmod g+w mydir
ls -dl mydir

drwxrwx---+ ... tux project3 ... mydir

getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:djungle:rwx
mask::rwx
other:---
```

### 35.3.3 Adresář s výchozími ACL

Adresáře mohou mít zvláštní typ ACL tzv. výchozí ACL. Výchozí ACL nastavuje přístupová práva ke všem podřízeným adresářům s nastavenými výchozími ACL. Výchozí ACL se nastavuje přístupové ACL jak u adresářů tak v nich obsažených souborech.

## Vliv výchozích ACL

S výchozím ACL je pracováno různě podle toho, na jaký typ objektu je uplatňován:

- ACL podadresáře se skládá z výchozího ACL, jeho vlastního výchozího ACL a přístupového ACL adresáře.
- Přístupová práva souboru se skládají z jeho vlastních ACL a výchozího ACL.

Všechny objekty souborového systému používají při nastavení přístupových práv parametr `mode`, který definuje přístupová práva nově vytvářených objektů.

- Pokud rodičovský adresář nemá nastavené výchozí ACL, nastaví se přístupové bity podle hodnoty parametru `mode` příkazu `umask`.
- Pokud má rodičovský adresář nastavené výchozí ACL, nově vytvářený objekt převezme přístupová práva od parametru `mode` a z výchozího ACL. `Umask` je ignorován.

## Aplikace výchozích ACLs

Následující tři kroky ilustrují operace pro adresáře a výchozí ACLs:

- vytvoření výchozího ACL pro aktuální existující adresář
- Vytvoření podadresáře v adresáři s nastavených výchozím ACL
- Vytvoření souboru v adresáři s výchozím ACL

### 1. Vložení výchozí ACLs do existujícího adresáře `mydir`:

```
setfacl -d -m group:djungle:r-x mydir
```

Parametr `-d` příkazu `setfacl` zajistí změny (parametr `-m`) ve výchozím ACLs.

Podívejme se blíže na výstup příkazu `getfacl mydir`:

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
```

```
group::r-x
group:djungle:rwX
mask::rwX
other:---
default:user::rwX
default:group::r-x
default:group:djungle:r-x
default:mask::r-x
default:other:---
```

`getfacl` vrátí jak přístupová ACL tak výchozí ACL. Výchozí ACL je tvořeno řádkami začínajícími na `default`. Po nastavení výchozího ACL příkazem `setfacl` pro skupinu `djungle` příkaz `setfacl` automaticky přepokopíruje všechny ostatní položky k nastavení platného výchozího ACL. Nastavení výchozího ACL nebude mít na existující objekty žádný okamžitý vliv. Ovlivňovat bude pouze nově vytvářené objekty po nastavení výchozího ACL. Tyto nové objekty budou mít přístupová práva skládající se pouze z výchozího ACL rodičovského adresáře.

2. Nyní použijte příkaz `mkdir` k vytvoření podadresáře v adresáři `mydir`, který bude mít stejné ACLs.

```
mkdir mydir/mysubdir
getfacl mydir/mysubdir

# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwX
group::r-x
group:djungle:r-x
mask::r-x
other:---
default:user::rwX
default:group::r-x
default:group:djungle:r-x
default:mask::r-x
default:other:---
```

Jak jsme očekávali, nově vytvořený podadresář `mysubdir` má přístupová práva rodičovského adresáře. Nastavení přístupových práv `mysubdir` je stejné jako `mydir`.

3. Použití příkazu `touch` k vytvoření souboru v adresáři `mydir`:

```
touch mydir/myfile
ls -l mydir/myfile
```

```
-rw-r-----+ ... tux project3 ... mydir/myfile
```

```
getfacl mydir/myfile
```

```
# file: mydir/myfile
# owner: tux
# group: project3
user::rw-
group::r-x      # effective:r--
group:djungle:r-x # effective:r--
mask::r--
other::---
```

Důležitým je v tomto příkladě příkaz `touch` s režimem s hodnotou `0666`, což znamená, že nově vytvářené soubory mají nastaveno právo pro čtení a zápis pro všechny třídy uživatelů a `umask` ani ACLs nenastavují žádná další omezení (viz. 35.3.3 na straně 589).

V důsledku to znamená, že všechna přístupová práva neobsažená v režimu hodnoty jsou odstraněny z ACLs položky. Přestože nebyla z ACLs *třídy* *group* odstraněna žádná práva, položka *mask* byla změněna k maskování jiným způsobem než s nastaveným režimem.

Tato vlastnost zajišťuje bezchybnou funkci ACLs aplikací např. kompilátorů. Můžete tak vytvářet souboru s omezenými přístupovými právy a zároveň je označit jako vykonatelné. Pomocí `mask` mechanismu zajistí, že k nim budou mít práva pouze ti správní uživatelé a skupiny.

### 35.3.4 ACL kontrolní algoritmus

Všechny procesy a aplikace projdou před tím, než je jim povolen přístup k objektům chráněným ACLs kontrolním algoritmem. ACLs jsou testovány na následující sekvence: *owner*, *named user*, *owning group* nebo *named group* a *other*. Přístup je pak řízen s nejlepším výsledkem ve prospěch procesu. Sekvence nelze slučovat.

Tento algoritmus je samozřejmě mnohem komplikovanější, pokud objekt patří do více skupin s různými vlastnostmi. V takovém případě algoritmus náhodně vybere ze skupin, které mají požadované vlastnosti. Je jedno, jaká z položek bude vést k výsledku *access granted*. Pokud algoritmus nenajde žádnou vhodnou skupinu, výsledkem bude *access denied*.

## 35.4 Výhledy

Jak bylo napsáno výše, ACLs umožňuje mnohem podrobnější nastavení přístupových práv. ACLs lze v případě potřeb kombinovat se starým konceptem nastavení přístupových práv. Některé důležité aplikace však stále ACLs nepodporují. Mimo programu star například stále není k dispozici zálohovací program s plnou podporou ACLs.

Základní příkazy (`cp`, `mv`, `ls` atd.) ACLs podporují, ale řada editorů a správců souborů na (např. Konqueror). Při kopírování souborů v Konqueroru dojde ke ztrátě jejich ACLs. Při změně v editorech jsou někdy ACLs zachovány, jindy ne. Důvodem je různý zálohovací režim editorů. Možnosti jsou tyto:

- Pokud editor zapisuje změny do originálního souboru, jsou ACLs zachovány.
- Pokud editor vytváří nový soubor s pozměněným obsahem starého souboru a pak provádí přejmenování na původní jméno, dojde ke ztrátě ACLs bez ohledu na to, zda editor ACLs podporuje.

Aplikací s podporou ACL se objevuje stále více, takže se dá předpokládat, že Linux dokáže plně využít této funkce již v nejbližší době.

## 35.5 Další informace

Detailní informace o ACLs získáte na následujících stránkách [http://sdb.suse.de/en/sdb/html/81\\_acl.html](http://sdb.suse.de/en/sdb/html/81_acl.html), <http://acl.bestbits.at/> a v manálových stránkách příkazů `getfacl`, `acl(5)` a `setfacl(1)`



# Nástroje monitorování systému

Aktuální stav systému lze zjistit pomocí mnoha různých nástrojů. Najdete zde také nástroje potřebné pro každodenní práci včetně jejich nejdůležitějších parametrů.

36.1	Seznam otevřených souborů: lsof . . . . .	595
36.2	Přístup uživatelů k souborům: fuser . . . . .	596
36.3	Vlastnosti souboru: stat . . . . .	597
36.4	USB zařízení: lsusb . . . . .	597
36.5	SCSI zařízení: scsiinfo . . . . .	598
36.6	Procesy: top . . . . .	599
36.7	Seznam procesů: ps . . . . .	599
36.8	Strom procesů: pstree . . . . .	602
36.9	Kdo co dělá: w . . . . .	603
36.10	Využití paměti: free . . . . .	603
36.11	Systémové hlášení jádra: dmesg . . . . .	604
36.12	Souborový systém a jeho využití: mount, df a du . . . . .	604
36.13	Souborový systém /proc . . . . .	605
36.14	vmstat, iostat a mpstat . . . . .	607
36.15	procinfo . . . . .	608
36.16	PCI zdroje: lspci . . . . .	609
36.17	Systémová volání běžícího programu: strace . . . . .	609
36.18	Volání knihoven běžícím příkazem: ltrace . . . . .	611
36.19	Zjištění vyžadovaných knihoven: ldd . . . . .	611

36.20	Dodatečné informace o ELF binárních souborech . . . .	612
36.21	Meziprocesová komunikace: ipcs . . . . .	612
36.22	Měření času: time . . . . .	613

U každého příkazu je současně uveden také příklad výstupu. Na první řádce příkladu je vždy příkaz (po znaku dolaru). Komentáře jsou uzavřeny v závorkách [...]. U dlouhých řádek, pokud je to potřeba, je zalomení. Zalomení dlouhých řádek se provádí pomocí znaku zpětného lomítka (\).

```
$ command -x -y
output line 1
output line 2
output line 3 is annoyingly long, so long that \
    we have to break it
output line 3
[...]
output line 98
output line 99
```

Popis každého z nástrojů je pouze stručný, aby bylo možné zmínit co největší množství užitečných příkazů. Podrobnější informace o každém příkazu najdete v jeho manuálové stránce. U většiny příkazů lze také použít krátkou nápovědu zadáním parametru `--help`.

## 36.1 Seznam otevřených souborů: lsof

Seznam všech souborů otevřených procesem s ID  $\langle PID \rangle$  získáte zadáním parametru `-p`. Například všechny soubory otevřené aktuálním shellem zjistíte příkazem:

```
$ lsof -p $$
COMMAND PID USER  FD  TYPE DEVICE SIZE      NODE NAME
zsh      4694  jj   cwd   DIR    0,18    144 25487368 /use/jj/t (totan:/real-home/jj)
zsh      4694  jj   rtd   DIR    3,2      608      2 /
zsh      4694  jj   txt   REG    3,2    441296   20414 /bin/zsh
zsh      4694  jj   mem   REG    3,2   104484   10882 /lib/ld-2.3.3.so
zsh      4694  jj   mem   REG    3,2   11648   20610 /usr/lib/zsh/4.2.0/zsh/rlimits.so
[...]
zsh      4694  jj   mem   REG    3,2   13647   10891 /lib/libdl.so.2
zsh      4694  jj   mem   REG    3,2   88036   10894 /lib/libnsl.so.1
zsh      4694  jj   mem   REG    3,2   316410  147725 /lib/libncurses.so.5.4
zsh      4694  jj   mem   REG    3,2   170563   10909 /lib/tls/libm.so.6
zsh      4694  jj   mem   REG    3,2  1349081   10908 /lib/tls/libc.so.6
zsh      4694  jj   mem   REG    3,2     56   12410 /usr/lib/locale/de_DE.utf8/LC_TELEPHONE
[...]
zsh      4694  jj   mem   REG    3,2     59   14393 /usr/lib/locale/en_US/LC_NUMERIC
zsh      4694  jj   mem   REG    3,2  178476   14565 /usr/lib/locale/en_US/LC_CTYPE
zsh      4694  jj   mem   REG    3,2   56444   20598 /usr/lib/zsh/4.2.0/zsh/computil.so
zsh      4694  jj    0u   CHR 136,48      50 /dev/pts/48
zsh      4694  jj    1u   CHR 136,48      50 /dev/pts/48
zsh      4694  jj    2u   CHR 136,48      50 /dev/pts/48
zsh      4694  jj   10u   CHR 136,48      50 /dev/pts/48
```

Ve výše uvedeném příkladu byla použita proměnná shellu `$$`, kde `$$` vrací ID aktuálního shellu.

Bez parametru vypíše příkaz `lsuf` všechny otevřené soubory. Obvykle jde o velmi velké množství souborů. Jejich počet zjistíte příkazem:

```
$ lsuf | wc -l
3749
```

Seznam používaných znakových zařízení:

```
$ lsuf | grep CHR
sshd      4685      root  mem   CHR    1,5          45833 /dev/zero
sshd      4685      root  mem   CHR    1,5          45833 /dev/zero
sshd      4693       jj   mem   CHR    1,5          45833 /dev/zero
sshd      4693       jj   mem   CHR    1,5          45833 /dev/zero
zsh       4694       jj    0u   CHR  136,48          50 /dev/pts/48
zsh       4694       jj    1u   CHR  136,48          50 /dev/pts/48
zsh       4694       jj    2u   CHR  136,48          50 /dev/pts/48
zsh       4694       jj   10u   CHR  136,48          50 /dev/pts/48
X          6476      root  mem   CHR    1,1          38042 /dev/mem
lsuf      13478       jj    0u   CHR  136,48          50 /dev/pts/48
lsuf      13478       jj    2u   CHR  136,48          50 /dev/pts/48
grep      13480       jj    1u   CHR  136,48          50 /dev/pts/48
grep      13480       jj    2u   CHR  136,48          50 /dev/pts/48
```

## 36.2 Přístup uživatelů k souborům: fuser

Předpokládejme, že chcete odpojit souborový systém připojený k `/mnt`:

```
$ mount -l | grep /mnt
/dev/sda on /mnt type ext2 (rw,noexec,nosuid,nodev,noatime,user=jj)
```

Pokus o odpojení selže:

```
$ umount /mnt
umount: /mnt: device is busy
```

Proces, který k adresáři `/mnt` přistupuje, zjistíte příkazem:

```
$ fuser -v /mnt/*

          USER          PID ACCESS COMMAND
/mnt/notes.txt
          jj            26597 f....  less
```

Po ukončení procesu `less` spuštěného z jiného terminálu půjde souborový systém bez problémů odpojit.

## 36.3 Vlastnosti souboru: stat

Příkazem `stat` zobrazíte vlastnosti souboru:

```
$ stat xml-doc.txt
  File: `xml-doc.txt'
  Size: 632          Blocks: 8          IO Block: 4096   regular file
Device: eh/14d Inode: 5938009      Links: 1
Access: (0644/-rw-r--r--)  Uid: (11994/   jj)   Gid: (   50/   suse)
Access: 2004-04-27 20:08:58.000000000 +0200
Modify: 2003-06-03 15:29:34.000000000 +0200
Change: 2003-07-23 17:48:27.000000000 +0200
```

Pomocí parametru `--filesystem` získáte podrobnosti o souborovém systému, jehož je soubor součástí:

```
$ stat . --filesystem
  File: "."
    ID: 0          Namelen: 255      Type: ext2/ext3
Blocks: Total: 19347388  Free: 17831731  Available: 16848938  Size: 4096
Inodes: Total: 9830400   Free: 9663967
```

Pokud používáte z shell (zsh), musíte zadat `/usr/bin/stat`, protože z shell obsahuje zabudovaný příkaz `stat` s jinými parametry a jiným typem výstupu:

```
% type stat
stat is a shell builtin
% stat .
device 769
inode 4554808
mode 16877
nlink 12
uid 11994
gid 50
rdev 0
size 4096
atime 1091536882
mtime 1091535740
ctime 1091535740
blksize 4096
blocks 8
link
```

## 36.4 USB zařízení: lsusb

Příkazem `lsusb` získáte výpis všech připojených USB zařízení. S parametrem `-v` bude výpis podrobnější. Program načítá informace z adresáře `/proc/bus/usb/`. V následujícím příkladu si můžete prohlédnout výpis příkazu `lsusb` po připojení flash disku. Zařízení je vypsáno na poslední řádce.

```

Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 018: ID 0402:5634 ALi Corp.

```

## 36.5 SCSI zařízení: `scsiinfo`

Příkazem `scsiinfo` můžete získat výpis všech připojených SCSI zařízení. Všechna SCSI zařízení vypíšete přidáním parametru `-l` (podobné informace můžete získat také s pomocí příkazu `lsscsi`). V následujícím příkladu výstupu `scsiinfo -i /dev/sda` můžete vidět informace o disku `/dev/sda`.

```

Inquiry command
-----
Relative Address                0
Wide bus 32                     0
Wide bus 16                     1
Synchronous neg.               1
Linked Commands                 1
Command Queueing                1
SftRe                           0
Device Type                     0
Peripheral Qualifier            0
Removable?                      0
Device Type Modifier            0
ISO Version                     0
ECMA Version                    0
ANSI Version                    3
AENC                            0
TrmIOP                          0
Response Data Format             2
Vendor:                         FUJITSU
Product:                        MAS3367NP
Revision level:                 0104A0K7P43002BE

```

Podrobnější informace získáte zadáním parametru `-a`. Ve výstupu je pak vypsán také seznam chyb na disku, který obsahuje dvě tabulky chybných bloků: první

je tabulka od výrobce (manufacturer table) a druhá obsahuje chyby, ke kterým došlo během používání disku (grown table). Pokud se počet položek v druhé tabulce zvyšuje, je čas uvažovat o výměně disku.

## 36.6 Procesy: top

Příkaz `top` zobrazí každé dvě sekundy obnovovaný seznam procesů. Program ukončíte stisknutím klávesy `Q`. Pokud chcete program automaticky ukončit po zobrazení prvního seznamu, spusťte ho s parametrem `-n 1`:

```
$ top -n 1
top - 14:19:53 up 62 days, 3:35, 14 users, load average: 0.01, 0.02, 0.00
Tasks: 102 total, 7 running, 93 sleeping, 0 stopped, 2 zombie
Cpu(s): 0.3% user, 0.1% system, 0.0% nice, 99.6% idle
Mem: 514736k total, 497232k used, 17504k free, 56024k buffers
Swap: 1794736k total, 104544k used, 1690192k free, 235872k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  Command
1426 root        15   0 116m  41m  18m  S  1.0   8.2   82:30.34 X
20836 jj         15   0  820   820   612  R  1.0   0.2    0:00.03 top
   1 root        15   0  100    96    72  S  0.0   0.0    0:08.43 init
   2 root        15   0     0     0     0  S  0.0   0.0    0:04.96 keventd
   3 root        34  19     0     0     0  S  0.0   0.0    0:00.99 ksoftirqd_CPU0
   4 root        15   0     0     0     0  S  0.0   0.0    0:33.63 kswapd
   5 root        15   0     0     0     0  S  0.0   0.0    0:00.71 bdflush
    [...]
1362 root        15   0  488   452  404  S  0.0   0.1    0:00.02 nsd
1363 root        15   0  488   452  404  S  0.0   0.1    0:00.04 nsd
1377 root        17   0   56     4     4  S  0.0   0.0    0:00.00 mingetty
1379 root        18   0   56     4     4  S  0.0   0.0    0:00.01 mingetty
1380 root        18   0   56     4     4  S  0.0   0.0    0:00.01 mingetty
```

Stisknutí klávesy `F` během běhu příkazu `top` vstoupíte do nabídky umožňující změnu formátu výstupu.

Zadáním parametru `-U <UID>` a uživatelského jména, získáte seznam procesů zadaného uživatele. `<UID>` je ID uživatele. Následující příkaz vypíše `<UID>` uživatele zadaného uživatelského jména a jeho procesy:

```
$ top -U $(id -u UzivatelскеJmeno)
```

## 36.7 Seznam procesů: ps

Zadáním příkazu `ps` získáte seznam procesů. S parametrem `r` omezíte výpis pouze na aktuální procesy využívající počítačový čas:

```
$ ps r
  PID TTY          STAT       TIME COMMAND
 22163 pts/7      R           0:01 -zsh
   3396 pts/3      R           0:03 emacs new-makedoc.txt
 20027 pts/7      R           0:25 emacs xml/common/utilities.xml
 20974 pts/7      R           0:01 emacs jj.xml
 27454 pts/7      R           0:00 ps r
```

Tento parametr se zadává *bez* minus před písmenem. Některé příkazy se někdy píší s minus a někdy bez. Správný zápis obvykle najdete v manuálové stránce. Návod vypsáný příkazem `ps --help` bývá obvykle velmi stručný.



Počet běžících příkazů např. emacs zjistíte příkazem:

```
$ ps x | grep emacs
1288 ?      S      0:07 emacs
3396 pts/3  S      0:04 emacs new-makedoc.txt
3475 ?      S      0:03 emacs .Xresources
20027 pts/7  S      0:40 emacs xml/common/utilities.xml
20974 pts/7  S      0:02 emacs jj.xml

$ pidof emacs
20974 20027 3475 3396 1288
```

Parametr -p seřadí procesy podle ID:

```
$ ps www -p $(pidof xterm)
  PID TTY          STAT       TIME COMMAND
 9025 ?            S          0:01 xterm  -g 100x45+0+200
 9176 ?            S          0:00 xterm  -g 100x45+0+200
29854 ?            S          0:21 xterm  -g 100x75+20+0 -fn \
-B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1
 4378 ?            S          0:01 xterm  -bg MistyRose1 -T root -n root -e su -l
25543 ?            S          0:02 xterm  -g 100x45+0+200
22161 ?            R          0:14 xterm  -g 100x45+0+200
16832 ?            S          0:01 xterm  -bg MistyRose1 -T root -n root -e su -l
16912 ?            S          0:00 xterm  -g 100x45+0+200
17861 ?            S          0:00 xterm  -bg DarkSeaGreen1 -g 120x45+40+300
19930 ?            S          0:13 xterm  -bg LightCyan
21686 ?            S          0:04 xterm  -g 100x45+0+200 -fn \
lucidasanstypewriter-12
23104 ?            S          0:00 xterm  -g 100x45+0+200
26547 ?            S          0:00 xterm  -g 100x45+0+200
```

Seznam procesů můžete naformátovat podle vlastních potřeb. Seznam všech možností získáte příkazem -L. Podle využití paměti procesy seřadíte příkazem:

```
$ ps ax --format pid,rss,cmd --sort rss
  PID  RSS  CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
   17     0 [kblockd/0]
[...]
```

```
10164 5260 xterm
31110 5300 xterm
17010 5356 xterm
3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth /var/lib/xdm/authdir/au
```

## 36.8 Strom procesů: pstree

Příkaz `ps tree` zobrazí běžící procesy ve stromovém výpisu:

```
$ pstree
init--+-atd
      |-3*[automount]
      |-bdf flush
      |-cron
      [...]
      |-usb-storage-1
      |-usb-storage-2
      |-10*[xterm---zsh]
      |-xterm---zsh---mutt
      |-2*[xterm---su---zsh]
      |-xterm---zsh---ssh
      |-xterm---zsh---pstree
      |-ypbind---ypbind---2*[ypbind]
      `--zsh---startx---xinit4--+-X
                                `--ctwm--+-xclock
                                       |-xload
                                       `--xosview.bin
```

Parametrem `-p` získáte ke jménům procesů také jejich ID. S parametrem `-a` vypíše příkaz také parametry příkazů:

```
$ pstree -pa
init,1
  |-atd,1255
  [...]
  `--zsh,1404
      `--startx,1407 /usr/X11R6/bin/startx
          `--xinit4,1419 /suse/jj/.xinitrc [...]
              |-X,1426 :0 -auth /suse/jj/.Xauthority
              `--ctwm,1440
                  |-xclock,1449 -d -geometry -0+0 -bg grey
                  |-xload,1450 -scale 2
                  `--xosview.bin,1451 +net -bat +net
```

## 36.9 Kdo co dělá: w

Příkazem `w` zjistíte uživatele přihlášené na počítači a jejich činnosti. Například:

```
$ w
15:17:26 up 62 days,  4:33, 14 users,  load average: 0.00, 0.04, 0.01
USER      TTY      LOGIN@  IDLE   JCPU   PCPU WHAT
jj        pts/0    30Mar04  4days  0.50s  0.54s xterm -e su -l
jj        pts/1    23Mar04  5days  0.20s  0.20s -zsh
jj        pts/2    23Mar04  5days  1.28s  1.28s -zsh
jj        pts/3    23Mar04  3:28m  3.21s  0.50s -zsh
[...]
jj        pts/7    07Apr04  0.00s   9.02s  0.01s w
jj        pts/9    25Mar04  3:24m  7.70s  7.38s mutt
[...]
jj        pts/14   12:49   37:34   0.20s  0.13s ssh totan
```

Podle poslední řádky je uživatel `jj` k počítači `totan` připojen pomocí secure shellu (`ssh`). U vzdáleně připojených uživatelů a jiných systémů získáte informace o vzdáleném počítači parametrem `-f`.

## 36.10 Využití paměti: free

Nástrojem `free` zjistíte využití RAM. Zobrazeny jsou jak informace o využití paměti, tak o volné paměti (a swapu):

```
$ free
              total        used        free     shared    buffers     cached
Mem:          514736       273964       240772          0       35920       42328
-/+ buffers/cache:      195716       319020
Swap:         1794736       104096       1690640
```

Údaje v MB získáte zadáním parametru `-m`:

```
$ free -m
              total        used        free     shared    buffers     cached
Mem:             502         267         235          0         35         41
-/+ buffers/cache:         191         311
Swap:             1752         101        1651
```

Následující řádka obsahuje skutečně zajímavé informace:

Jde o paměť zásobníků a vyrovnávací paměti. Parametrem `-d <n>` zadáte, aby došlo k obnovení výpisu každých  $\langle n \rangle$  sekund. Například `free -d 1.5` obnoví výpis každé 1,5 sekundy.

## 36.11 Systémové hlášení jádra: dmesg

Linuxové jádro uchovává systémová hlášení v paměti omezené velikosti (standardně 2 na 14 B). Tato hlášení zobrazíte příkazem `dmesg`:

```
$ dmesg
[...]
sdc : READ CAPACITY failed.
sdc : status = 1, message = 00, host = 0, driver = 08
Info fld=0xa00 (nonstd), Current sd00:00: sense key Not Ready
sdc : block size assumed to be 512 bytes, disk size 1GB.
sdc: test WP failed, assume Write Enabled
sdc: I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
unable to read partition table
I/O error: dev 08:20, sector 0
nfs: server totan not responding, still trying
nfs: server totan OK
```

Poslední řádka indikuje dočasné problémy s NFS serverem `totan`. Řádky před ní jsou spojeny se zasunutím USB flash disku.

Starší události najdete v souborech `/var/log/messages` a `/var/log/warn`.

## 36.12 Souborový systém a jeho využití: mount, df a du

Příkaz `mount` souborový systém (zařízení a typ) a jeho body připojení:

```
$ mount
/dev/hdb2 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /data type ext2 (rw)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
automount(pid1012) on /suse type autofs \
    (rw,fd=5,pgrp=1012,minproto=2,maxproto=3)
totan:/real-home/jj on /suse/jj type nfs \
    (rw,nosuid,rsize=8192,wsiz=8192,hard,intr,nolock,addr=10.10.0.1)
```

Informaci o využití místa získáte příkazem `df`. S parametrem `-h` (nebo `--human-readable`) získáte výstup v uživatelsky přívětivém formátu.

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdb2       7.4G  5.1G  2.0G   73% /
/dev/hda1       74G   5.8G   65G    9% /data
shmfs           252M    0   252M    0% /dev/shm
totan:/real-home/jj 350G  324G   27G   93% /suse/jj
```

Uživatelé NFS serveru `totan` by měli neodkladně promazat své domovské adresáře. Celkovou velikost všech souborů a podadresářů vypisuje příkaz `du`. S parametrem `-s` vypíše pouze celkovou velikost bez dalších detailů. Parametr `-h` povede k uživatelsky přívětivému výstupu. Zadáním příkazu:

```
$ du -sh ~
361M    /suse/jj
```

získáte velikost svého domovského adresáře.

## 36.13 Souborový systém /proc

V adresáři `/proc` se nachází pseudo souborový systém, do kterého jádro ve formě virtuálních souborů ukládá důležité informace. Například k informacím o typu procesoru můžete přistoupit příkazem:

```
$ cat /proc/cpuinfo
processor      : 0
vendor_id    : AuthenticAMD
cpu family    : 6
```

```

model          : 8
model name     : AMD Athlon(tm) XP 2400+
stepping       : 1
cpu MHz        : 2009.343
cache size     : 256 KB
fdiv_bug       : no
[...]

```

Využití přerušení zjistíte příkazem:

```

$ cat /proc/interrupts
          CPU0
 0:   537544462          XT-PIC  timer
 1:     820082          XT-PIC  keyboard
 2:           0          XT-PIC  cascade
 8:           2          XT-PIC  rtc
 9:           0          XT-PIC  acpi
10:     13970          XT-PIC  usb-uhci, usb-uhci
11:  146467509          XT-PIC  ehci_hcd, usb-uhci, eth0
12:   8061393          XT-PIC  PS/2 Mouse
14:   2465743          XT-PIC  ide0
15:    1355          XT-PIC  ide1
NMI:           0
LOC:           0
ERR:           0
MIS:           0

```

Některé důležité soubory a jejich obsah:

**/proc/devices** dostupná zařízení

**/proc/modules** zavedené moduly jádra

**/proc/cmdline** příkazová řádka jádra

**/proc/meminfo** podrobné informace o využití paměti

**/proc/config.gz** gzip archiv s konfigurací běžícího jádra

Další informace najdete v souboru `/usr/src/linux/Documentation/filesystems/proc.txt`. Informace o běžících procesech najdete v adresáři `/proc/<NNN>`, kde `<NNN>` je ID (PID) příslušného procesu. Proces a jeho částečnou charakteristiku najdete v `/proc/self/`:

```
$ ls -l /proc/self
lrwxrwxrwx 1 root root 64 Apr 29 13:52 /proc/self -> 27585
```

```
$ ls -l /proc/self/
total 0
dr-xr-xr-x 2 jj suse 0 Apr 29 13:52 attr
-r----- 1 jj suse 0 Apr 29 13:52 auxv
-r--r--r-- 1 jj suse 0 Apr 29 13:52 cmdline
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 cwd -> /suse/jj/t
-r--r--r-- 1 jj suse 0 Apr 29 13:52 delay
-r----- 1 jj suse 0 Apr 29 13:52 environ
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
dr-x----- 2 jj suse 0 Apr 29 13:52 fd
-rw----- 1 jj suse 0 Apr 29 13:52 mapped_base
-r--r--r-- 1 jj suse 0 Apr 29 13:52 maps
-rw----- 1 jj suse 0 Apr 29 13:52 mem
-r--r--r-- 1 jj suse 0 Apr 29 13:52 mounts
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 root -> /
-r--r--r-- 1 jj suse 0 Apr 29 13:52 stat
-r--r--r-- 1 jj suse 0 Apr 29 13:52 statm
-r--r--r-- 1 jj suse 0 Apr 29 13:52 status
dr-xr-xr-x 3 jj suse 0 Apr 29 13:52 task
-r--r--r-- 1 jj suse 0 Apr 29 13:52 wchan
```

Adresy spustitelných adres a knihoven jsou v souboru maps:

```
$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 22890 /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 22890 /bin/cat
0804d000-0806e000 rwxp 0804d000 00:00 0
40000000-40016000 r-xp 00000000 03:02 10882 /lib/ld-2.3.3.so
40016000-40017000 rw-p 00015000 03:02 10882 /lib/ld-2.3.3.so
40017000-40018000 rw-p 40017000 00:00 0
4002b000-40135000 r-xp 00000000 03:02 10908 /lib/tls/libc.so.6
40135000-4013d000 rw-p 0010a000 03:02 10908 /lib/tls/libc.so.6
4013d000-40141000 rw-p 4013d000 00:00 0
bffffe000-c00000000 rw-p bffffe000 00:00 0
fffffe000-fffff000 ---p 00000000 00:00 0
```

## 36.14 vmstat, iostat a mpstat

Nástroje vmstat slouží ke zjištění informací o virtuální paměti. Údaje získávají ze souborů /proc/meminfo, /proc/stat a /proc/\*/stat. Jde o velmi užitečné nástroje při zjišťování slabín ve výkonu počítače.

S pomocí příkazu iostat můžete získat informace o procesoru, I/O zařízeních a diskových oddílech. Údaje jsou čteny z /proc/stat a /proc/partitions. Výstup může být velmi užitečný např. při ladění zátěže vstupních a výstupních operací mezi disky. Příkaz mpstat vypisuje statistiky související s CPU.

## 36.15 procinfo

Souhrn všech důležitých informací a systému /proc získáte příkazem procinfo:

```
$ procinfo
Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU [roth.suse.de]
```

Memory:	Total	Used	Free	Shared	Buffers
Mem:	516696	513200	3496	0	43284
Swap:	530136	1352	528784		

Bootup: Wed Jul 7 14:29:08 2004 Load average: 0.07 0.04 0.01 1/126 5302

user :	2:42:28.08	1.3%	page in :	0
nice :	0:31:57.13	0.2%	page out:	0
system:	0:38:32.23	0.3%	swap in :	0
idle :	3d 19:26:05.93	97.7%	swap out:	0
uptime:	4d 0:22:25.84		context :	207939498

irq 0:	776561217 timer	irq 8:	2 rtc
irq 1:	276048 i8042	irq 9:	24300 VIA8233
irq 2:	0 cascade [4]	irq 11:	38610118 acpi, eth0, uhci_hcd
irq 3:	3	irq 12:	3435071 i8042
irq 4:	3	irq 14:	2236471 ide0
irq 6:	2	irq 15:	251 ide1

Po zadání parametru `-a` vypíše příkaz všechny informace. S parametrem `-n<N>` bude výpis obnovován každých N sekund. program ukončíte stisknutím klávesy **Q**.

Ve výchozím nastavení jsou zobrazeny hodnoty kumulativně. Parametr `-d` provede k výpisu změněných hodnot. Příkazem `procinfo -dn5` získáte hodnoty změněné za posledních 5 sekund:

Memory:	Total	Used	Free	Shared	Buffers	Cached
Mem:	0	2	-2	0	0	0
Swap:	0	0	0			

Bootup: Wed Feb 25 09:44:17 2004 Load average: 0.00 0.00 0.00 1/106 31902

user :	0:00:00.02	0.4%	page in :	0	disk 1:	0r	0w
nice :	0:00:00.00	0.0%	page out:	0	disk 2:	0r	0w
system:	0:00:00.00	0.0%	swap in :	0	disk 3:	0r	0w
idle :	0:00:04.99	99.6%	swap out:	0	disk 4:	0r	0w
uptime:	64d 3:59:12.62		context :	1087			

irq 0:	501 timer	irq 10:	0 usb-uhci, usb-uhci
irq 1:	1 keyboard	irq 11:	32 ehci_hcd, usb-uhci,
irq 2:	0 cascade [4]	irq 12:	132 PS/2 Mouse
irq 6:	0	irq 14:	0 ide0
irq 8:	0 rtc	irq 15:	0 ide1
irq 9:	0 acpi		



## 36.16 PCI zdroje: lspci

Příkaz `lspci` vypíše PCI zdroje:

```
$ lspci
00:00.0 Host bridge: VIA Technologies, Inc. \
    VT8366/A/7 [Apollo KT266/A/333]
00:01.0 PCI bridge: VIA Technologies, Inc. \
    VT8366/A/7 [Apollo KT266/A/333 AGP]
00:0b.0 Ethernet controller: Digital Equipment Corporation \
    DECchip 21140 [FasterNet] (rev 22)
00:10.0 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.1 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.2 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.3 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 82)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8235 ISA Bridge
00:11.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B \
    PIPC Bus Master IDE (rev 06)
00:11.5 Multimedia audio controller: VIA Technologies, Inc. \
    VT8233 AC97 Audio Controller (rev 50)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. \
    MGA G550 AGP (rev 01)
```

Podrobnější výpis získáte zadáním parametru `-v`:

```
$ lspci -v
[...]
01:00.0 \
    VGA compatible controller: Matrox Graphics, Inc. MGA G550 AGP (rev 01) \
    (prog-if 00 [VGA])
    Subsystem: Matrox Graphics, Inc. Millennium G550 Dual Head DDR 32Mb
    Flags: bus master, medium devsel, latency 32, IRQ 10
    Memory at d8000000 (32-bit, prefetchable) [size=32M]
    Memory at da000000 (32-bit, non-prefetchable) [size=16K]
    Memory at db000000 (32-bit, non-prefetchable) [size=8M]
    Expansion ROM at <unassigned> [disabled] [size=128K]
    Capabilities: <available only to root>
```

Informace o jménech zařízení jsou uložena v souboru `/usr/share/pci.ids`.  
PCI ID neobsažené v tomto souboru jsou označena jako **Unknown device**.

Parametr `-vv` povede k vypsání všech dostupných informací. Čistě numerické hodnoty získáte zadáním parametru `-n`.

## 36.17 Systémová volání běžícího programu: strace

Nástroj `strace` umožňuje zjistit všechna systémová volání běžících procesů:

```
$ strace -e open ls

execve("/bin/ls", ["ls"], [/* 88 vars */]) = 0
uname({sys="Linux", node="edison", ...}) = 0
brk(0) = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0x40017000
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...}) = 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=53, ws_col=110, ws_xpixel=897, ws_ypixel=693}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...}) = 0
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
getdents64(3, /* 5 entries */, 4096) = 160
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0x40018000
write(1, "ltrace-ls.txt myfile.txt strac...", 41) = 41
munmap(0x40018000, 4096) = 0
exit_group(0) = ?
```

Pro vypísání všech pokusů o otevření určitého souboru (např. myfile.txt) stačí napsat:

```
$ strace -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
open("/lib/tls/librt.so.1", O_RDONLY) = 3
open("/lib/libacl.so.1", O_RDONLY) = 3
open("/lib/libselinux.so.1", O_RDONLY) = 3
open("/lib/tls/libc.so.6", O_RDONLY) = 3
open("/lib/tls/libpthread.so.0", O_RDONLY) = 3
open("/lib/libattr.so.1", O_RDONLY) = 3
open("/proc/mounts", O_RDONLY) = 3
[...]
open("/proc/filesystems", O_RDONLY) = 3
open("/proc/self/attr/current", O_RDONLY) = 4
```

K výpisu potomků určitého procesu použijte parametr `-f`. Chování i výstup příkazu lze ovlivnit. Podrobnější informace získáte v manuálové stránce `man strace`.

## 36.18 Volání knihoven běžícím příkazem: `ltrace`

Příkazem `ltrace` získáte výpis všech volání knihoven procesu. Příkaz je používán podobně jako `strace`. Zadáním parametru `-c` získáte počet a trvání volání knihoven:

```
$ ltrace -c find /usr/share/doc
% time      seconds    usecs/call   calls      errors syscall
-----
 86.27      1.071814         30      35327             write
 10.15      0.126092         38       3297          getdents64
  2.33      0.028931          3      10208          lstat64
  0.55      0.006861          2       3122          1 chdir
  0.39      0.004890          3       1567          2 open
[...]
```

% time	seconds	usecs/call	calls	errors	syscall
86.27	1.071814	30	35327		write
10.15	0.126092	38	3297		getdents64
2.33	0.028931	3	10208		lstat64
0.55	0.006861	2	3122	1	chdir
0.39	0.004890	3	1567	2	open
[...]					
0.00	0.000003	3	1		uname
0.00	0.000001	1	1		time
100.00	1.242403		58269	3	total

## 36.19 Zjištění vyžadovaných knihoven: `ldd`

Pomocí příkazu `ldd` zjistíte jaké dynamické knihovny vyžaduje určitá dynamicky linkovaná aplikace. Pro příkaz `ls` bude výstup vypadat takto:

```
$ ldd /bin/ls
linux-gate.so.1 => (0xffffe000)
librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)
libacl.so.1 => /lib/libacl.so.1 (0x40033000)
libseline.so.1 => /lib/libselinux.so.1 (0x40039000)
libc.so.6 => /lib/tls/libc.so.6 (0x40048000)
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4015d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

Staticky linkované aplikace nevyžadují žádné dynamické knihovny:

```
$ ldd /bin/sash
not a dynamic executable
```

```
$ file /bin/sash
/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
for GNU/Linux 2.2.5, statically linked, stripped
```

## 36.20 Dodatečné informace o ELF binárních souborech

Obsah spustitelných binárních souborů lze číst pomocí nástroje `readelf`. Funguje také pro ELF soubory vytvořené pro jinou hardwarovou architekturu:

```
$ readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                   ELF32
  Data:                     2's complement, little endian
  Version:                  1 (current)
  OS/ABI:                   UNIX - System V
  ABI Version:              0
  Type:                     EXEC (Executable file)
  Machine:                  Intel 80386
  Version:                  0x1
  Entry point address:      0x8049b40
  Start of program headers: 52 (bytes into file)
  Start of section headers: 76192 (bytes into file)
  Flags:                    0x0
  Size of this header:      52 (bytes)
  Size of program headers:  32 (bytes)
  Number of program headers: 9
  Size of section headers:  40 (bytes)
  Number of section headers: 29
  Section header string table index: 26
```

## 36.21 Meziprocesová komunikace: `ipcs`

Příkazem `ipcs` získáte seznam používaných IPC zdrojů:

```
$ ipcs
----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x000027d9   5734403   toms       660        64528      2
0x00000000   5767172   toms       666        37044      2
```

```
0x00000000 5799941      toms      666      37044      2

----- Semaphore Arrays -----
key          semid      owner      perms      nsems
0x000027d9 0          toms      660      1

----- Message Queues -----
key          msqid      owner      perms      used-bytes  messages
```

## 36.22 Měření času: time

Čas potřebný pro vykonání určitého příkazu lze zjistit pomocí příkazu `time`. Tento příkaz je dostupný ve dvou variantách buď jako zabudovaný příkaz shellu nebo jako program (`/usr/bin/time`).

```
$ time find . > /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s
```



**Část V**

**Přílohy**







# Dokumentace a zdroje informací

Pro SUSE LINUX existuje řada informačních zdrojů, které vám pomohou při práci a nastavení vašeho systému. Některé z těchto zdrojů jsou specifické pouze pro SUSE, ale většina je obecná. Některé tyto zdroje budete mít přístupné na svém systému okamžitě při instalaci, jiné jsou přístupné pouze na Internetu.

## SUSE dokumentace

Řadu důležitých podrobných informací najdete ve svých knížkách. Digitální podobu knížek ve formátech HTML nebo PDF najdete v RPM balíčcích `suselinux-adminguide_cs` a `suselinux-adminguide_cs-pdf`). Knihy jsou ve standardní instalaci nainstalovány v adresáři `/usr/share/doc/manual/`. Přistupovat k nim můžete například prostřednictvím centra nápovědy SUSE.

## The Linux Documentation Project (LDP)

Linux - dokumentační projekt (viz. <http://www.tldp.org/>) byl založen dobrovolníky starajícími se o linuxovou distribuci. LDP obsahuje HOWTO, FAQy a příručky uveřejněné pod svobodnými licencemi.

HOWTO je návod, který krok za krokem popisuje určité nastavení. V HOWTO je například popsán způsob nastavení DHCP serveru, ale již ne instalace Linuxu. Jedná se o obecný návod, který lze připojit ke každé distribuci. HOWTO v ASCII

formátu jsou obsaženy v balíčku `howto`. V případě, že dáváte přednost HTML formátu, nainstalujte si balíček `howtoenh`.

FAQy (*frequently asked questions*) jsou sbírky často kladených dotazů a jejich odpovědí z různých emailových konferencí. Jde například o otázky typu *Co je LDAP?* nebo *Co je RAID?*. Odpovědi jsou zpravidla velmi stručné.

Příručky jsou dokumenty, které určitou problematiku popisují mnohem podrobněji a hlouběji než HOWTO a FAQy. Může jít například o programování jádra nebo kompletní správu sítě. Hlavním cílem je podání co nejobsáhlejší a nejpodrobnější informace o daném tématu.

Některé části TLDP dokumentace jsou dostupné i v jiných formátech jako PDF, jednoduchá a strukturovaná HTML publikace, PostScript, SGML nebo XML zdroj. Standardně je veškerá dokumentace dostupná v angličtině a některé dokumenty jsou překládány do jednotlivých národních jazyků.

## Manuálové a info stránky

Manuálové stránky (*man page*) poskytují nápovědu pro příkazy, systémová volání, formáty souborů atd. Obvykle jsou rozděleny do několika sekcí pojednávajících o jménu, syntaxi, volbách a souborech.

Manuálovou stránku zobrazíte pomocí příkazu `man` následovaným jménem příkazu, jehož stránku si přejete zobrazit. Např. příkaz `man ls` zobrazí manuálovou stránku příkazu `ls`. Po dokumentu se můžete nahoru a dolů pohybovat pomocí šipek. Čtení ukončíte stisknutím klávesy `Q`. Manuálovou stránku vytisknete příkazem `card`, např. `card ls` pro příkaz `ls`. Jednoduchou nápovědu příkazu `card` (balíček `a2ps`) zobrazíte zadáním tohoto příkazu s parametrem `--help`.

Některá typy dokumentace jsou dostupné také ve formátu `info` např. `grep`. Info stránky příkazu `grep` zobrazíte příkazem `info grep`.

Info stránky jsou mnohem podrobnější než manuálové stránky. Jsou rozděleny do několika *nodů* a lze je číst v prohlížečích info stránek (podobných HTML prohlížeči). V info stránkách se můžete pohybovat pomocí kláves `P` (předchozí stránka) a `N` (následující stránka). Klávesou `Q` příkaz `info` a tím i čtení ukončíte. Další klávesy jsou popsány v dokumentaci `info` (příkaz `info info`).

Jak manuálové tak info stránky lze číst v prohlížeči Konqueror. V poli určeném pro zadání adresy napište `man:<prikaz>` nebo `info:<prikaz>`.

## Standardy a specifikace

Standardy a specifikace lze dohledat na řadě míst.

**[www.linuxbase.org](http://www.linuxbase.org)** *The Free Standards Group* je nezávislá nezisková organizace zaměřující se na svobodný software. Spravuje několik důležitých standardů jako např. LSB (*Linux Standard Base*).

**<http://www.w3.org>** *The World Wide Web Consortium* (W3C) je pravděpodobně jednou z nejznámějších standardizačních organizací. Byla založena v říjnu roku 1994 Timem Berners-Leem a zaměřuje se na webové technologie. W3C šíří specifikace HTML, XHTML a XML. Věnuje se jak otevřeným standardům tak standardům závislým na řešeních výrobce. Webové standardy jsou uveřejňovány jako doporučení (*W3C recommendations* - REC).

**<http://www.oasis-open.org>** OASIS (*Organization for the Advancement of Structured Information Standards*) je mezinárodní konzorcium zaměřující se na vývoj bezpečnostních standardů pro web, internetový obchod, internetové obchodní transakce, logistiku a spolupráci trhů.

**<http://www.ietf.org>** *The Internet Engineering Task Force* (IETF) je založena na spolupráci vývojářů a uživatelů. Zaměřuje se především na vývoj architektury internetu a s ním spojené protokoly.

Každý IETF standard je publikován jako RFC (Request for Comments) a poskytnut volně veřejnosti. Je celkem šest typů RFC: proposed standardy, draft standardy, internetové standardy, experimentální protokoly, informativní dokumenty a historické standardy. Pouze první tři (proposed, draft a full) jsou brány jako skutečné IETF standardy (viz. <http://www.ietf.org/rfc/rfc1796.txt>).

**<http://www.ieee.org>** *The Institute of Electrical and Electronics Engineers* (IEEE) se stará o standardy z oblasti informatiky, telekomunikací, lékařství atd..IEEE jsou zpoplatněny.

**<http://www.iso.org>** Mezinárodní organizace pro normy ISO (*International Organization for Standards*) je světově největší vydavatel standardů působící ve více než 140 zemích. ISO standardy jsou zpoplatněny.

**<http://www.din.de>, <http://www.din.com>**  
Český normalizační institut je organizace odpovědná za normy v České republice.



# SUSE LINUX FAQ

## Informace

### Jsou SUSE manuály dostupné také ve formě PDF nebo HTML souborů?

Manuály najdete v digitální podobě na CD. Po instalaci příslušných balíčků jsou k dispozici také v Centru nápovědy SUSE. Centrum nápovědy spustíte současným stisknutím kláves **(Alt)-(F2)** a zadáním příkazu `susehelp`. HTML verze je součástí balíčků `suselinux-adminguide_cs` a `suselinux-userguide_cs`. Po instalaci je najdete v adresářích `/usr/share/doc/manual/suselinux-adminguide_en` popř. `/usr/share/doc/manual/suselinux-userguide_en`. PDF verze je součástí balíčků `suselinux-adminguide_cs-pdf` a `suselinux-userguide_cs-pdf`.

### Kde se dají o systému SUSE LINUX získat další informace?

Mnoho užitečných informací o vlastnostech a různých aplikacích najdete systému SUSE LINUX najdete v manuálu. Dokumentace jednotlivých aplikací je uložena v adresáři `/usr/share/doc/packages` a *HowTo* v `/usr/share/doc/howto/en`. Soubory můžete číst např. pomocí `: less /usr/share/doc/howto/en/DOS-to-Linux-HOWTO.txt.gz`

Čtení ukončíte stisknutím klávesy **(Q)**.

## Hardware

**Je můj hardware podporovaný?** Podporovaný hardware je pravidelně uveřejňován na stránce <http://cdb.suse.de>. In-

formace o hardwaru lze získat také zadáním příkazu `less /usr/share/doc/howto/en/Hardware-HOWTO.gz`.

## Instalace

### **Jaké jsou systémové požadavky systému SUSE LINUX?**

Systémové požadavky najdete na stránce <http://www.novell.com/products/linuxprofessional/>.

### **Kolik místa na disku Linux potřebuje?**

Pro standardní instalaci je doporučená velikost diskového oddílu 2 GB. Počítat musíte také s místem pro svá data. Pro normální práci obvykle postačuje něco kolem jednoho GB, v případě práce s multimédií jako např. editace nahrávek z kamery nebo ukládání velkého počtu obrázků, je rozumné počítat s prostorem i několik GB. Určitý prostor na disku bude vyžadovat také odkládací prostor tzv. swap, který by měl dvojnásobnou velikost než RAM počítače.

### **Co se myslí pojmem rozdělování disku?**

Rozdělováním disku se rozumí dělení disku na části tzv. diskové oddíly. Ve výchozím nastavení SUSE LINUX vyžaduje alespoň dva diskové oddíly, jeden pro kořenový adresář, druhý pro swap.

### **Jaký je doporučený souborový systém?**

Výběr souborového systému závisí na účelu, kterému bude váš počítač sloužit. Pro domácí použití a malé domácí servery můžete použít souborový systém ReiserFS nebo Ext3. Více informací o souborových systémech najdete v Příručce správce systému.

### **Kde se dají získat informace o softwaru obsaženém v systému SUSE LINUX?**

Seznam všech balíčků najdete na stránce <http://www.novell.com/products/linuxpackages/professional/index.html>. Popis balíčků najdete samozřejmě také v modulu správce softwaru programu YaST.

### **Je možné mít Linux na stejném počítači s Windows? Jak vytvořit dual boot systém?**

**Jak Linux smazat?** Linuxové diskové oddíly smažete pomocí příkazu `fdisk`. Nejdřív jej spusťte v Linuxu. Pak spusťte systém v DOSu nebo Windows spusťte `fdisk /MBR`.

# Správa systému

## Jsem jediná osoba používající počítač, musím se přihlašovat?

Linux je víceuživatelský systém. Uživatelé jsou identifikováni podle uživatelského jména a hesla. Pokud chcete, aby byl určitý uživatel přihlášen ihned po spuštění systému bez nutnosti zadávat identifikační údaje, zvolte při instalaci 'Automatické přihlášení'. Přihlašovat se pak musíte pouze jako `root` (pro instalaci nových programů a správu systému).

Automatické přihlášení na již nainstalovaném systému nastavíte jako uživatel `root` v programu YaST v modulu 'Bezpečnost a uživatelé' → 'Správce uživatelů'. Označte uživatele, u kterého chcete automatické přihlášení, klikněte na 'Expertní volby' a zvolte 'Nastavení přihlášení' → 'Automatické přihlášení'. Po restartu KDM bude zvolený uživatel automaticky přihlášen do systému.

## Varování

Automatické přihlášení představuje nezanedbatelné bezpečnostní riziko. Používejte ho proto pouze na systémech, které mají skutečně pouze jednoho uživatele a kde nemáte uložena žádná choulostivá data, která by mohla být zneužita.

## Varování

**Hrozí v Linuxu viry?** Pro Linux žádné skutečné viry prakticky neexistují. Linuxové virové skenery jsou určeny pro kontrolu příchozí pošty na poštovních serverech poskytujících služby klientům Windows. Existují však i jiný způsoby, jak může váš systém dojít k úhoně (např. porucha hardwaru), proto nezanedbávejte zálohování důležitých dat a nastavení.

## Nemůžu najít žádný `.exe` soubor. Kde jsou všechny aplikace?

V Linuxu spustitelné soubory nemají žádnou příponu. Řada důležitých spustitelných souborů se nachází např. v adresářích `/usr/bin` a `/usr/X11R6/bin`.

**Jak se poznají spustitelné příkazy?** Po vypsání `ls -l` získáte seznam všech souborů v aktuálním adresáři, jména spustitelných souborů budou napsána zeleně a v prvním sloupci budou mít `x`.

## Jak spustit aplikaci nebo službu při startu systému?

Pro automatické spuštění služeb při startu systému použijte YaST Editor

úrovni běhu. Informace o principech spouštění najdete v kapitolách věnovaných startu systému.

V GNOME spusťte Ovládací centrum GNOME a zvolte 'Systém' → 'Sezení'. Klikněte na záložku 'Programy při spuštění' a zadejte aplikace, které si přejete spouštět při přihlášení.

V KDE spusťte Konqueror a otevřete adresář `.kde/Autostart` ve svém domovském adresáři. Myší chytněte ikonu programu, který si přejete spouštět při přihlášení, přetáhněte ji do okna adresáře a zvolte 'Vytvořit odkaz na tomto místě'. Aplikace se automaticky spustí při dalším přihlášení do KDE.

### **Mám pouze zdrojové kódy programu. Jak je mám nainstalovat?**

Než se pustíte do vlastní kompilace jakéhokoliv programu, podívejte se, zda již neexistuje hotové RPM pro vaši verzi systému SUSE LINUX. Dobrými zdroji programů jsou:

- `ftp://ftp.suse.com/pub/projects`
- `ftp://ftp.suse.com/pub/suse/i386/supplementary`
- `http://rpmfind.net`
- `http://packman.links2linux.org/`

Pokud žádné RPM nenajdete, rozbalte archiv např. příkazem `tar xvzf name.tar.gz` a přečtěte si instrukce v souborech `INSTALL` nebo `README`. Prosím vezte na vědomí, že kompilace vlastních aplikací nespadá do bezplatné instalační podpory.

### **Musím si sám/sama kompilovat jádro?**

Ne, především pokud s Linuxem začínáte, vyhněte se vlastní kompilaci jádra. Distribuční jádro obvykle poskytuje vše, co potřebujete. V případě vlastního jádra vám navíc nejsme schopni poskytnout bezplatnou instalační podporu.

**Jak defragmentovat disk?** Linuxové souborové systémy jsou navrženy tak, aby fragmentaci disku zabráňovaly. V každém případě se však ujistěte, že nepoužíváte víc než osmdesát procent kapacity každého diskového oddílu. Přeplnění disku totiž znamená zvýšení fragmentace u prakticky každého souborového systému, linuxové nevyjímaje. Informace o zaplnění disku získáte buď v některém k tomu určeném grafickém programu nebo příkazem `df -h`.



## Potřebuji více místo pro Linux. Jak přidám další disk?

Více místa získáte jednoduchým připojením dalšího disku do systému. předpokládáme například, že potřebujete více místa pro adresář `/opt`, který se bude nacházet na dodatečně připojeném disku. Adresář přemístíte následujícím způsobem

1. Nainstalujte disk podle návodu výrobce. Pak spusťte Linux.
2. Přihlaste se jako uživatel `root` a vytvořte si adresář `/opt2`. Použít můžete např. příkaz:

```
mkdir /opt2
```

3. Rozdělte a naformátujte nový disk pomocí modulu 'Rozdělování disku' programu YaST. Nově vytvořenému a naformátovanému oddílu přiřaďte bod připojení `/opt2`. V našem příkladu budeme předpokládat, že půjde o oddíl `/dev/hdb1`.

Pokud nechcete používat YaST, máte k dispozici programy `fdisk` pro rozdělení a `mke2fs` pro naformátování. Návod k jejich používání najdete v jejich manuálových stránkách.

4. Zadejte tyto příkazy:

```
cd /opt
cp -axv . /opt2
```

Nyní odstraňte starý adresář `/opt` a vytvořte prázdný bod připojení příkazy:

```
mv /opt /opt.old
mkdir /opt
```

V libovolném editoru upravte řádku: `/etc/fstab`.

```
/dev/hdb1          /opt2          ext2          defaults      1      1
```

na

```
/dev/hdb1          /opt           ext2          defaults      1      1
```

Pokud jste použili jiný souborový systém, nebude řádka obsahovat parametr `ext2`, ale např. `ext3` nebo `reiserfs`. Tento parametr ponechte nezměněný podle nastavení vašeho systému. Měňte pouze bod připojení `/opt2` na `/opt`.

Restartujte počítač.

5. Po restartu se `/dev/hdb1` připojí do `/opt`. Pokud vše funguje bez problémů, můžete smazat `/opt.old` příkazy:

```
cd /  
rm -fr opt.old
```

### Jak zjistit, kolik místa je v Linuxu na disku volného?

Nejrychlejší způsob nabízí příkaz `df -hT`. Volba `-h` převede velikost do lidsky přívětivého formátu (e.g. 1K, 234M, 2G) a `-T` vypíše typ souborového systému.

V KDE si můžete spustit v hlavní nabídce 'Systém' → 'Monitor' → 'Informační centrum'. Informace o disku najdete pod položkou 'Úložná zařízení'.

## Aplikace

**Jak nainstalovat aplikace?** Programy, které jsou součástí instalačních médií systému SUSE LINUX je nejvhodnější nainstalovat pomocí programu YaST. Instalaci programů může provádět pouze uživatel `root`.

### Jak zadávat v GNOME/KDE příkazy?

V prostředí KDE klikněte na ikonu monitoru s mušlí v hlavním panelu. Další emulátory textové konzole nebo-li terminály můžete najít v hlavní nabídce. V GNOME zvolte terminál v hlavní nabídce. Pokud chcete jen spustit určitý program, stiskněte klávesy **(Alt)-(F2)** ta zadejte příkaz.

### Jak se z textové konzole dostat do grafického prostředí?

Ve výchozí instalaci máte k dispozici celkem šest virtuálních konzolí, na které se můžete z grafického prostředí přepnout stisknutím kláves **(Ctrl)-(Alt)-(F1)** to **(F6)**. Zpět do grafického prostředí se vrátíte stisknutím kláves **(Alt)-(F7)**.

## Řešení problémů

**Kde se dají číst systémové zprávy?** Systémové zprávy nebo-li logy se nachází v adresáři `/var/log/`. Abyste si je mohli přečíst, musíte se přihlásit jako uživatel `root`. Nejdůležitější informace se nacházejí v souboru `messages`. Pokud z tohoto souboru chcete zobrazit jen nejnovější zprávy tak, jak přibývají, zadejte příkaz `tail -f /var/log/messages`. Zprávy systému během startu se zapisují do souboru `boot.msg`.

Jestliže chcete vidět, jaké procesy zrovna na vašem systému běží, zadejte příkaz `top`. K informacím z adresáře `/proc` lze přistupovat příkazem `procinfo`. Hezké grafické znázornění vyřízení procesu a využití paměti nabízí např. `xosview`.

**Jak najít určitý soubor?** Při hledání můžete použít grafické nástroje prostředí KDE/GNOME nebo příkaz `find`. Informace o příkazu `find` najdete v manuálové stránce `find(1)`.

**Hledám určitý soubor (např. `libfoo.so.2`). Ve kterém je balíčku?**

V grafickém prostředí můžete pro vyhledávání použít funkci hledání v modulu instalace balíčků programu YaST.

V textovém prostředí soubor najdete pomocí příkazu `pin` (Package Information):

```
pin jmeno_souboru
```

Více informací o hledání najdete v manuálové stránce `pin(1)`.

**Můj počítač je značně nestabilní a zatuhává. Můžu zmáčknout bez obav reset?**

Pokud systém reaguje velmi pomalu nebo vůbec na pohyb myši či stisknutí klávesy, nemusí to nutně znamenat, že zatuhl. Je docela možné, že klávesnici nebo myš blokuje jen špatně fungující program, ale zbytek systému běží dál bez chyb. V případě, že funguje klávesnice, přepněte se do textové konzole současným stisknutím kláves `(Ctrl)-(Alt)-(F2)`. Pomocí příkazů `ps` a `top` zjistíte viníka a ukončete ho příkazem `killall <jmeno_programu>`. Pokud by tento příkaz program neukončil, použijte `killall -9 <jmeno_programu>`.

Jestliže nefunguje klávesnice a máte možnost vzdáleného přístupu (sériový terminál, síť), přihlaste se na počítač vzdáleně a postupujte stejným způsobem jako na konzoli.

Může se stát, že počítač zatuhne tak, že se na něj nelze přihlásit ani vzdáleně nebo prostě možnost vzdáleného přihlášení nemáte. Než zmáčknete tlačítko reset, počkejte deset sekund a ujistěte se, že počítač nevykazuje žádnou diskovou aktivitu (nesvítí kontrolka disku).

Abyste předešli poškození dat, zajistěte bezpečný zápis všech dat na disk. Funkci, která vám to umožní, aktivujete jako uživatel `root` v souboru `/etc/sysconfig/sysctl` nastavením proměnné `ENABLE_SYSRQ` na `yes`. V případě problémů pak můžete uložit obsah vyrovnávací paměti a odpojit souborový systém stisknutím kláves `(Alt)-(SysRQ)-(U)` a pak bez obav o data stisknout reset.

Nemůžu se na počítač dostat příkazem `telnet`. Vždy hlásí *Login incorrect*.

Pravděpodobně se snažíte přihlásit jako uživatel `root`. Z bezpečnostních důvodů není možné se přes `telnet` jako tento uživatel přihlásit. Přihlaste se jako normální uživatel a po přihlášení se můžete přepřihlásit příkazem `su`. Mnohem lepší a bezpečnější řešení je však použít místo `telnetu` `ssh`, který celé připojení šifruje.

## Kontaktování SUSE

V systému SUSE LINUX je chyba. Jak ji nahlásit?

Než chybu nahlásíte, proveďte online update. Je totiž možné, že chybu již někdo odhalil a byla opravena. Zároveň překontrolujte, zda máte vyřešeny závislosti balíčků, a také se podívejte do databáze instalační podpory (<http://portal.suse.com/sdb/cz/index.html>), zda již není váš problém znám. Pokud v databázi instalační podpory není popsáno žádné řešení a problém po updatu trvá, napište nám popis chyby na adresu <mailto:feedback@suse.cz?subject=FAQ>. Prosíme zasílejte nám chyby v jednotlivých zprávách, velmi nám tím usnadníte jejich zpracování, a také urychlíte vzájemnou komunikaci. Popis by měl obsahovat následující části:

- Stručný popis chyby.
- Popis, jak problém reprodukovat.
- Chybové hlášení, které systém po chybě vypíše (pokud existuje).
- Popis svého hardwaru.

V případě, že máte problémy s aplikací, kterou jste nenainstalovali z instalačních médií systému SUSE LINUX, obraťte se s žádostí o řešení problému na výrobce této aplikace.

Co je **mirror**? Proč by se nemělo vše stahovat přímo z [ftp.suse.com](http://ftp.suse.com)?

Každý FTP server dokáže v určité době nabídnout své služby pouze určitému množství lidí. Aby nedocházelo k jeho přetížení a tím i nedostupnosti, je možné vytvořit další FTP servery se stejným obsahem jako má původní server, a směřovat uživatele na tyto servery. Těmto serverům se říká *mirrory* nebo-li zrcadla.

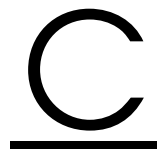
Server [ftp.suse.com](http://ftp.suse.com) patří k nejvytíženějším serverům vůbec, proto má řadu *mirrorů*. Pokud se chcete vyhnout pomalému stahování nebo častým výpadkům spojení, doporučujeme vám je využít. Jejich seznam najdete na

stránce [http://www.novell.com/products/linuxprofessional/downloads/ftp/int\\_mirrors.html](http://www.novell.com/products/linuxprofessional/downloads/ftp/int_mirrors.html). Abyste dosáhli největší možné rychlosti a spolehlivosti, zvolte si nejbližší dostupný mirror (nejlépe v zemi, kde se nacházíte).

**Mám problém, na který zde není odpověď. Je možné ji v některém z příštích manuálů publikovat?**

Jsme velmi rádi, že nám posíláte připomínky k našim manuálům. Pokud jste v této části nenašli odpověď na svou otázku, můžete nám ji zaslat na adresu <mailto:feedback@suse.cz?subject=FAQ> a my se pokusíme ji v některé z budoucích příruček uveřejnit. Pokud potřebujete pouze radu, obraťte se na svého poskytovatele podpory.





# Kontrola souborového systému

## Manuálová stránka reiserfsck

REISERFSCK(8)

REISERFSCK(8)

### NAME

reiserfsck - check a Linux Reiserfs file system

### SYNOPSIS

```
reiserfsck [ -afprVy ] [ --rebuild-sb | --check | --fix-  
fixable | --rebuild-tree | --clean-attributes ] [ -j |  
--journal device ] [ -z | --adjust-size ] [ -n | --nolog ]  
[ -l | --logfile file ] [ -q | --quiet ] [ -y | --yes ] [  
-S | --scan-whole-partition ] [ --no-journal-available ]  
device
```

### DESCRIPTION

Reiserfsck searches for a Reiserfs filesystem on a device, replays any necessary transactions, and either checks or repairs the file system.

device is the special file corresponding to the device or partition (e.g /dev/hdXX for IDE disk partition or /dev/sdXX for SCSI disk partition).

### OPTIONS

--rebuild-sb

This option recovers the superblock on a Reiserfs partition. Normally you only need this option if mount reports "read\_super\_block: can't find a reiserfs file system" and you are sure that a Reiserfs file system is there.

`--check`

This default action checks file system consistency and reports but does not repair any corruption that it finds. This option may be used on a read-only file system mount.

`--fix-fixable`

This option recovers certain kinds of corruption that do not require rebuilding the entire file system tree (`--rebuild-tree`). Normally you only need this option if the `--check` option reports "corruption that can be fixed with `--fix-fixable`". This includes: zeroing invalid data-block pointers, correcting `st_size` and `st_blocks` for directories, and deleting invalid directory entries.

`--rebuild-tree`

This option rebuilds the entire file system tree using leaf nodes found on the device. Normally you only need this option if the `--check` option reports "corruption that can be fixed only during `--rebuild-tree`". You are strongly encouraged to make a backup copy of the whole partition before attempting the `--rebuild-tree` option.

`--clean-attributes`

This option cleans reserved fields of Stat-Data items.

`--journal device , -j device`

This option supplies the device name of the current file system journal. This option is required when the journal resides on a separate device from the main data device (although it can be avoided with the expert option `--no-journal-available`).

`--adjust-size, -z`



This option causes reiserfsck to correct file sizes that are larger than the offset of the last discovered byte. This implies that holes at the end of a file will be removed. File sizes that are smaller than the offset of the last discovered byte are corrected by `--fix-fixable`.

`--logfile file, -l file`

This option causes reiserfsck to report any corruption it finds to the specified log file rather than `stderr`.

`--nolog, -n`

This option prevents reiserfsck from reporting any kinds of corruption.

`--quiet, -q`

This option prevents reiserfsck from reporting its rate of progress.

`--yes, -y`

This option inhibits reiserfsck from asking you for confirmation after telling you what it is going to do, assuming yes. For safety, it does not work with the `--rebuild-tree` option.

`-a, -p` These options are usually passed by `fsck -A` during the automatic checking of those partitions listed in `/etc/fstab`. These options cause reiserfsck to print some information about the specified file system, check if error flags in the superblock are set and do some light-weight checks. If these checks reveal a corruption or the flag indicating a (possibly fixable) corruption is found set in the superblock, then reiserfsck switches to the fix-fixable mode. If the flag indicating a fatal corruption is found set in the superblock, then reiserfsck finishes with an error.

`-V` This option prints the reiserfsprogs version and exit.

`-r, -f` These options are ignored.

#### EXPERT OPTIONS

DO NOT USE THESE OPTIONS UNLESS YOU KNOW WHAT YOU ARE DOING. WE ARE NOT RESPONSIBLE IF YOU LOSE DATA AS A RESULT OF THESE OPTIONS.

##### `--no-journal-available`

This option allows reiserfsck to proceed when the journal device is not available. This option has no effect when the journal is located on the main data device. NOTE: after this operation you must use reiserfstune to specify a new journal device.

##### `--scan-whole-partition, -S`

This option causes `--rebuild-tree` to scan the whole partition, not only used space on the partition.

#### EXAMPLE OF USING

1. You think something may be wrong with a reiserfs partition on `/dev/hda1` or you would just like to perform a periodic disk check.
2. Run `reiserfsck --check --logfile check.log /dev/hda1`. If `reiserfsck --check` exits with status 0 it means no errors were discovered.
3. If `reiserfsck --check` exits with status 1 (and reports about fixable corruptions) it means that you should run `reiserfsck --fix-fixable --logfile fixable.log /dev/hda1`.
4. If `reiserfsck --check` exits with status 2 (and reports about fatal corruptions) it means that you need to run `reiserfsck --rebuild-tree`. If `reiserfsck --check` fails in some way you should also run `reiserfsck --rebuild-tree`, but we also encourage you to submit this as a bug report.
5. Before running `reiserfsck --rebuild-tree`, please make a backup of the whole partition before proceeding. Then run `reiserfsck --rebuild-tree --logfile rebuild.log /dev/hda1`.
6. If the `--rebuild-tree` step fails or does not recover what you expected, please submit this as a bug report. Try to provide as much information as possible and we will try

to help solve the problem.

#### EXIT CODES

reiserfsck uses the following exit codes:

- 0 - No errors.
- 1 - File system errors corrected.
- 4 - File system fatal errors left uncorrected,  
reiserfsck --rebuild-tree needs to be launched.
- 6 - File system fixable errors left uncorrected,  
reiserfsck --fix-fixable needs to be launched.
- 8 - Operational error.
- 16 - Usage or syntax error.

#### AUTHOR

This version of reiserfsck has been written by Vitaly Fertman <vitaly@namesys.com>.

#### BUGS

There are likely to be some bugs. Please report bugs to the ReiserFS mail-list <reiserfs-list@namesys.com>.

#### TODO

Faster recovering, signal handling, i/o error handling, etc.

#### SEE ALSO

mkreiserfs(8), reiserfstune(8) resize\_reiserfs(8), debu greiserfs(8),

Reiserfsprogs-3.6.9

April 2003

REISERFSCK(8)

# Manuálová stránka e2fsck

E2FSCK(8)

E2FSCK(8)

## NAME

e2fsck - check a Linux second extended file system

## SYNOPSIS

```
e2fsck [ -pacnyrdfvstDFSV ] [ -b superblock ] [ -B block
size ] [ -l|-L bad_blocks_file ] [ -C fd ] [ -j external-
journal ] [ -E extended_options ] device
```

## DESCRIPTION

e2fsck is used to check a Linux second extended file system (ext2fs). E2fsck also supports ext2 filesystems containing a journal, which are also sometimes known as ext3 filesystems, by first applying the journal to the filesystem before continuing with normal e2fsck processing. After the journal has been applied, a filesystem will normally be marked as clean. Hence, for ext3 filesystems, e2fsck will normally run the journal and exit, unless its superblock indicates that further checking is required.

device is the device file where the filesystem is stored (e.g. /dev/hdc1).

## OPTIONS

-a This option does the same thing as the -p option. It is provided for backwards compatibility only; it is suggested that people use -p option whenever possible.

### -b superblock

Instead of using the normal superblock, use an alternative superblock specified by superblock. This option is normally used when the primary superblock has been corrupted. The location of the backup superblock is dependent on the filesystem's blocksize. For filesystems with 1k blocksizes, a backup superblock can be found at block 8193; for filesystems with 2k blocksizes, at block 16384; and for 4k blocksizes, at block 32768.

Additional backup superblocks can be determined by using the mke2fs program using the -n option to print out where the superblocks were created. The -b option to mke2fs, which specifies blocksize of

the filesystem must be specified in order for the superblock locations that are printed out to be accurate.

If an alternative superblock is specified and the filesystem is not opened read-only, e2fsck will make sure that the primary superblock is updated appropriately upon completion of the filesystem check.

**-B blocksize**

Normally, e2fsck will search for the superblock at various different block sizes in an attempt to find the appropriate block size. This search can be fooled in some cases. This option forces e2fsck to only try locating the superblock at a particular blocksize. If the superblock is not found, e2fsck will terminate with a fatal error.

**-c**

This option causes e2fsck to run the badblocks(8) program to find any blocks which are bad on the filesystem, and then marks them as bad by adding them to the bad block inode. If this option is specified twice, then the bad block scan will be done using a non-destructive read-write test.

**-C fd**

This option causes e2fsck to write completion information to the specified file descriptor so that the progress of the filesystem check can be monitored. This option is typically used by programs which are running e2fsck. If the file descriptor specified is 0, e2fsck will print a completion bar as it goes about its business. This requires that e2fsck is running on a video console or terminal.

**-d**

Print debugging output (useless unless you are debugging e2fsck).

**-D**

Optimize directories in filesystem. This option causes e2fsck to try to optimize all directories, either by reindexing them if the filesystem supports directory indexing, or by sorting and compressing directories for smaller directories, or for filesystems using traditional linear directories.

**-E extended\_options**

Set e2fsck extended options. Extended options are comma separated, and may take an argument using the

equals ('=') sign. The following options are supported:

- ea\_ver=extended\_attribute\_version  
Assume the format of the extended attribute blocks in the filesystem is the specified version number. The version number may be 1 or 2. The default extended attribute version format is 2.
- f Force checking even if the file system seems clean.
- F Flush the filesystem device's buffer caches before beginning. Only really useful for doing e2fsck time trials.
- j external-journal  
Set the pathname where the external-journal for this filesystem can be found.
- l filename  
Add the block numbers listed in the file specified by filename to the list of bad blocks. The format of this file is the same as the one generated by the badblocks(8) program. Note that the block numbers are based on the blocksize of the filesystem. Hence, badblocks(8) must be given the blocksize of the filesystem in order to obtain correct results. As a result, it is much simpler and safer to use the -c option to e2fsck, since it will assure that the correct parameters are passed to the badblocks program.
- L filename  
Set the bad blocks list to be the list of blocks specified by filename. (This option is the same as the -l option, except the bad blocks list is cleared before the blocks listed in the file are added to the bad blocks list.)
- n Open the filesystem read-only, and assume an answer of 'no' to all questions. Allows e2fsck to be used non-interactively. (Note: if the -c, -l, or -L options are specified in addition to the -n option, then the filesystem will be opened read-write, to permit the bad-blocks list to be updated. However, no other changes will be made to the filesystem.)
- p Automatically repair ("preen") the file system without any questions.

- r This option does nothing at all; it is provided only for backwards compatibility.
- s This option will byte-swap the filesystem so that it is using the normalized, standard byte-order (which is i386 or little endian). If the filesystem is already in the standard byte-order, e2fsck will take no action.
- S This option will byte-swap the filesystem, regardless of its current byte-order.
- t Print timing statistics for e2fsck. If this option is used twice, additional timing statistics are printed on a pass by pass basis.
- v Verbose mode.
- V Print version information and exit.
- y Assume an answer of 'yes' to all questions; allows e2fsck to be used non-interactively.

#### EXIT CODE

The exit code returned by e2fsck is the sum of the following conditions:

- 0 - No errors
- 1 - File system errors corrected
- 2 - File system errors corrected, system should be rebooted
- 4 - File system errors left uncorrected
- 8 - Operational error
- 16 - Usage or syntax error
- 32 - E2fsck canceled by user request
- 128 - Shared library error

#### SIGNALS

The following signals have the following effect when sent to e2fsck.

##### SIGUSR1

This signal causes e2fsck to start displaying a completion bar. (See discussion of the -C option.)

##### SIGUSR2

This signal causes e2fsck to stop displaying a completion bar.

#### REPORTING BUGS

Almost any piece of software will have bugs. If you manage to find a filesystem which causes e2fsck to crash, or which e2fsck is unable to repair, please report it to the author.

Please include as much information as possible in your bug report. Ideally, include a complete transcript of the e2fsck run, so I can see exactly what error messages are displayed. If you have a writeable filesystem where the transcript can be stored, the script(1) program is a handy way to save the output of e2fsck to a file.

It is also useful to send the output of dumpe2fs(8). If a specific inode or inodes seems to be giving e2fsck trouble, try running the debugfs(8) command and send the output of the stat(1u) command run on the relevant inode(s). If the inode is a directory, the debugfs dump command will allow you to extract the contents of the directory inode, which can sent to me after being first run through uuen code(1).

Always include the full version string which e2fsck displays when it is run, so I know which version you are running.

AUTHOR

This version of e2fsck was written by Theodore Ts'o  
<tytso@mit.edu>.

SEE ALSO

mke2fs(8), tune2fs(8), dumpe2fs(8), debugfs(8)

E2fsprogs version 1.34

July 2003

E2FCK(8)





# Manuálová stránka xfs\_check

xfs\_check(8)

xfs\_check(8)

## NAME

xfs\_check - check XFS filesystem consistency

## SYNOPSIS

xfs\_check [ -i ino ] ... [ -b bno ] ... [ -s ] [ -v ] xfs\_special

xfs\_check -f [ -i ino ] ... [ -b bno ] ... [ -s ] [ -v ] file

## DESCRIPTION

xfs\_check checks whether an XFS filesystem is consistent. It is normally run only when there is reason to believe that the filesystem has a consistency problem. The filesystem to be checked is specified by the xfs\_special argument, which should be the disk or volume device for the filesystem. Filesystems stored in files can also be checked, using the -f flag. The filesystem should normally be unmounted or read-only during the execution of xfs\_check. Otherwise, spurious problems are reported.

The options to xfs\_check are:

- f Specifies that the special device is actually a file (see the mkfs.xfs -d file option). This might happen if an image copy of a filesystem has been made into an ordinary file.
- s Specifies that only serious errors should be reported. errors are those that make it impossible to find major data structures in the filesystem. This option can be used to cut down the amount of output when there is a serious problem, when the output might make it difficult to see what the real problem is.
- v Specifies verbose output; it is impossibly long for a reasonably-sized filesystem. This option is intended for internal use only.
- i ino Specifies verbose behavior for a specific inode. For instance, it can be used to locate all the blocks associated with a given inode.
- b bno Specifies verbose behavior for a specific filesystem block. instance, it can be used to determine what a specific block is used for. The block number is a "file system block number". Conversion between disk addresses (i.e. addresses reported by xfs\_bmap) file system blocks may be accomplished using xfs\_db's convert command.

Any non-verbose output from xfs\_check means that the filesystem has an incon-

sistency. The filesystem can be repaired using either `xfs_repair(8)` to repair the filesystem in place, or by using `xfsdump(8)` and `mkfs.xfs(8)` to dump the filesystem, make a new filesystem, then use `xfsrestore(8)` to restore the data onto the new filesystem. Note that `xfsdump` may fail on a corrupt filesystem. However, if the filesystem is mountable, `xfsdump` can be used to try and backup important data before repairing the filesystem with `xfs_repair`. If the filesystem is not mountable though, `xfs_repair` is the only viable option.

## DIAGNOSTICS

Under one circumstance, `xfs_check` unfortunately might dump core rather than produce useful output. If the filesystem is completely corrupt, a core dump might be produced instead of the message `xxx is not a valid filesystem`.

If the filesystem is very large (has many files) then `xfs_check` might run out of memory. In this case the message out of memory is printed.

The following is a description of the most likely problems and the associated messages. Most of the diagnostics produced are only meaningful with some understanding of the structure of the filesystem.

`agf_freeblks n, counted m in ag a`

The freeblocks count in the allocation group header for allocation group a doesn't match the number of blocks counted free.

`agf_longest n, counted m in ag a`

The longest free extent in the allocation group header for allocation group a doesn't match the longest free extent found in the allocation group.

`agi_count n, counted m in ag a`

The allocated inode count in the allocation group header for allocation group a doesn't match the number of inodes counted in the allocation group.

`agi_freecount n, counted m in ag a`

The free inode count in the allocation group header for allocation group a doesn't match the number of inodes counted free in the allocation group.

`block a/b expected inum 0 got i`

The block number is specified as a pair (allocation group, block in the allocation group). The block is used multiple times (shared), between multiple inodes. This message usually appears as the first message of the next type.

`block a/b expected type unknown got y`

The block is used multiple times (shared).

`block a/b type unknown not expected`

The block is unaccounted for (not in the freelist and not in use).

link count mismatch for inode nnn (name xxx), nlink m, counted n  
The inode has a bad link count (number of references in directories)

rtblock b expected inum 0 got i  
The block is used multiple times (shared), between multiple inodes  
This message usually follows a message of the next type.

rtblock b expected type unknown got y  
The real-time block is used multiple times (shared).

rtblock b type unknown not expected  
The real-time block is unaccounted for (not in the freelist and not in use).

sb\_fdblocks n, counted m  
The number of free data blocks recorded in the superblock doesn't match the number counted free in the filesystem.

sb\_frexents n, counted m  
The number of free real-time extents recorded in the superblock doesn't match the number counted free in the filesystem.

sb\_icount n, counted m  
The number of allocated inodes recorded in the superblock doesn't match the number allocated in the filesystem.

sb\_ifree n, counted m  
The number of free inodes recorded in the superblock doesn't match the number free in the filesystem.

## SEE ALSO

mkfs.xfs(8), xfsdump(8), xfsrestore(8), xfs\_ncheck(8), xfs\_repair(8), xfs(5).

xfs\_check(8)

# Manuálová stránka jfs\_fsck

jfs\_fsck(8)

jfs\_fsck(8)

## NAME

jfs\_fsck - initiate replay of the JFS transaction log, and check and repair a JFS formatted device

## SYNOPSIS

```
jfs_fsck [ -afnpvV ] [ -j journal_device ]  
[--omit_journal_replay          ] [ --replay_journal_only ] device
```

## DESCRIPTION

jfs\_fsck is used to replay the JFS transaction log, check a JFS formatted device for errors, and fix any errors found.

device is the special filename corresponding to the actual device to be checked (e.g. /dev/hdb1).

jfs\_fsck must be run as root.

## WARNING

jfs\_fsck should only be used to check an unmounted file system or a file system that is mounted READ ONLY. Using jfs\_fsck to check a file system mounted other than READ ONLY could seriously damage the file system!

## OPTIONS

If no options are selected, the default is -p.

- a      Autocheck mode - Replay the transaction log. Do not continue fsck processing unless the aggregate state is dirty or the log replay failed. Functionally equivalent to -p. Autocheck mode is typically the default mode used when jfs\_fsck is called at boot time.
- f      Replay the transaction log and force checking even if the file system appears clean. Repair all problems automatically.
- j journal\_device      Specify the journal device.

- n      Open the                    file system read only. Do not replay the transaction log. Report errors, but do not repair them.
  
- omit\_journal\_replay  
      Omit the replay of the transaction log. This option should not be used unless as            a last resort (i.e. the log has been severely corrupted and replaying it causes further problems).
  
- p      Automatically repair ("preen") the file system. Replay the transaction log. Do not continue fsck processing unless the aggregate state is dirty or the log replay failed. Functionally equivalent to -a.
  
- replay\_journal\_only  
      Only replay the transaction log.                    Do not continue with a full file system check if the replay fails or if the file system is still dirty even after a journal replay. In general, this option should only be used for debugging purposes as it could leave the file system in an unmountable state. This option cannot be used with            -f, -n, or --omit\_journal\_replay.
  
- v      Verbose messaging - print details and debug statements to stdout.
  
- V      Print version information and                    exit (regardless of any other chosen options).

#### EXAMPLES

Check the 3rd partition on the 2nd hard disk, print extended information to stdout, replay            the transaction log, force complete jfs\_fsck checking, and give permission to repair all errors:

```
jfs_fsck -v -f /dev/hdb3
```

Check the 5th partition on the 1st hard disk, and report, but do not    repair, any errors:

```
jfs_fsck -n /dev/hda5
```

EXIT CODE

The exit code returned by `jfs_fsck` represents one of the following conditions:

- 0 No errors
- 1 File system errors corrected and/or transaction log replayed successfully
- 2 File system errors corrected, system should be rebooted if file system was mounted
- 4 File system errors left uncorrected
- 8 Operational error
- 16 Usage or syntax error
- 128 Shared library error

#### REPORTING BUGS

If you find a bug in JFS or `jfs_fsck`, please report it via the bug tracking system ("Report Bugs" section) of the JFS project web site:  
<http://oss.software.ibm.com/jfs>

Please send as much pertinent information as possible, including the complete output of running `jfs_fsck` with the `-v` option on the JFS device.

#### SEE ALSO

`fsck(8)`, `jfs_mkfs(8)`, `jfs_fscklog(8)`, `jfs_tune(8)`,  
`jfs_logdump(8)`, `jfs_debugfs(8)`

#### AUTHORS

Barry Arndt ([barndt@us.ibm.com](mailto:barndt@us.ibm.com))  
William Braswell, Jr.

`jfs_fsck` is maintained by IBM.  
See the JFS project web site for more details:  
<http://oss.software.ibm.com/jfs>

October 29, 2002

`jfs_fsck(8)`

# GNU licence

## GNU GENERAL PUBLIC LICENSE

Verze 2, červen 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place – Suite 330, Boston, MA 02111-1307, USA

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave,  
Cambridge, MA 02139, USA

Každému je dovoleno kopírovat a distribuovat doslovné kopie tohoto licenčního dokumentu, ale není dovoleno jej změnit.

### Preamble

Licence u většiny softwaru jsou navrženy tak, že vám vezmou možnost sdílet tento software a měnit jej. Účelem licence GNU General Public License je naproti tomu zaručit vám svobodu sdílet a měnit volný software – aby bylo zajištěno, že bude zdarma pro všechny jeho uživatele. Tato licence General Public License platí pro většinu softwaru nadace Free Software Foundation a jakýkoli jiný program, jehož autoři přistoupí k užívání této licence. (Některý další software nadace Free Software Foundation je místo toho kryt licencí GNU Library General Public License.) Licenci můžete využít i u svých programů.

Mluvíme-li o volném softwaru, hovoříme o svobodě, nikoli o ceně. Naše licence General Public Licenses jsou navrženy tak, aby vám zajistily možnost distribuovat kopie volného softwaru (a za tuto službu si účtovat, pokud máte zájem), možnost získání zdrojového kódu nebo možnost si jej opatřit v případě zájmu a

možnost měnit software nebo používat jeho části v nových volných programech. Licence dále zajistí, abyste věděli o tom, že tyto možnosti máte.

Abychom ochránili vaše práva, potřebujeme vytvořit omezení, která všem zakazují odepřít vám tato práva nebo žádat vás, abyste se jich vzdali. Důsledkem těchto omezení je jistá odpovědnost, kterou musíte přijmout, pokud distribuujete kopie softwaru nebo jej modifikujete.

Pokud například distribuujete kopie takového programu, ať už zdarma či za poplatek, musíte příjemcům poskytnout veškerá práva, která máte vy sami. Rovněž musíte zajistit, aby i oni obdrželi nebo měli možnost získat zdrojový kód. Dále jim musíte ukázat tyto podmínky, aby věděli, jaká práva mají.

Vaše práva chráníme dvěma kroky: (1) zajistíme copyright softwaru a (2) nabídneme vám tuto licenci, která vám dává právní svolení ke kopírování, distribuci a případně modifikaci softwaru.

V zájmu ochrany každého autora i naší vlastní ochrany si chceme být jisti, že každý chápe, že na tento volný software není poskytována žádná záruka. Je-li software modifikován někým jiným a předán dál, chceme, aby jeho příjemci věděli, že to, co mají, není originál, tak aby se žádné problémy vzniklé vinou jiných osob neodrazily na reputaci původního autora.

A konečně, každý volný program je neustále ohrožován softwarovými patenty. Chceme zabránit nebezpečí, že redistributoři volného programu budou samostatně získávat patentové licence, v důsledku čehož se program v podstatě stane jejich vlastnictvím. Abychom tomu zabránili, jasně jsme definovali, že každý patent musí být licencován pro volné použití kýmkoli, nebo nesmí být licencován vůbec.

Přesné podmínky a okolnosti pro kopírování, distribuci a modifikaci jsou uvedeny níže.

## **PODMÍNKY A OKOLNOSTI PRO KOPÍROVÁNÍ, DISTRIBUCI A MODIFIKACI**

0. Tato licence se vztahuje na jakýkoli program nebo jiné dílo obsahující upozornění uvedené držitelem autorských práv, které říká, že tento program nebo jiné dílo smí být distribuovány za podmínek této licence General Public License. Výraz *program* v dalším textu odkazuje na jakýkoli takový program nebo dílo a výraz *dílo založené na programu* znamená buď program, nebo jakékoli dílo z něj odvozené podle zákona o autorských právech; a to dílo obsahující program nebo



jeho část, ať už doslovnou nebo s úpravami, a případně s překladem v jiném jazyce. (Překlad je dále zahrnován bez omezení do pojmu *úprava*.) Každý držitel licence je osloven *vy*.

Jiné aktivity než kopírování, distribuce a modifikace nejsou touto licencí pokryty; spadají mimo její rámec. Akt provozu programu není omezen a výstup z programu je kryt pouze tehdy, představuje-li jeho obsah dílo založené na programu (nezávisle na tom, zda byl vytvořen provozem programu). Jestli je to pravda, to závisí na tom, co program dělá.

**1.** Můžete kopírovat a distribuovat doslovné kopie zdrojového kódu programu v podobě, v jaké jste jej obdrželi, na jakémkoli médiu, za předpokladu, že zřetelně a vhodným způsobem zveřejníte na každé kopii příslušné upozornění na autorská práva a odmítnutí záruky, všechna upozornění vztahující se k této licenci a k absenci jakékoli záruky zachováte neporušená a všem dalším příjemcům programu poskytnete kopii této licence současně s programem.

Za fyzický skutek přenosu kopie si můžete účtovat poplatek a podle vlastního uvážení můžete za poplatek nabídnout ochrannou záruku.

**2.** Můžete modifikovat svou kopii nebo kopie programu nebo jakékoli jeho části, čímž vytvoříte dílo založené na programu, a kopírovat a distribuovat takové úpravy nebo dílo za podmínek uvedených výše v části 1, pokud splníte rovněž všechny tyto podmínky:

1. a) ke všem upraveným souborům musíte připojit výrazné upozornění informující o skutečnosti, že jste soubory změnili vy, a o datu každé takové změny;
2. b) u každého vámi distribuovaného nebo publikovaného díla, které jako celek nebo částečně obsahuje program nebo jakoukoli jeho část, nebo je z programu či jeho části odvozeno, musíte zaručit, že bude bezplatně jako celek licencováno všem třetím stranám za podmínek daných touto licencí;
3. c) pokud modifikovaný program za normálních okolností při provozu čte interaktivně příkazy, musíte zajistit, aby při spuštění provozu pro takové interaktivní použití tím nejobvyklejším způsobem vytiskl nebo zobrazil oznámení obsahující příslušné upozornění na autorská práva a upozornění na neexistenci záruky (nebo upozornění informující o tom, že záruku poskytuje) a skutečnost, že uživatelé mohou program za těchto podmínek dále šířit, a informující uživatele, jak si může prohlédnout kopii této licence. (Výjimka: je-li program sám interaktivní, ale normálně takové oznámení netiskne, nemusí oznámení tisknout ani vaše dílo založené na programu.)

Tyto požadavky se vztahují na upravené dílo jako celek. Pokud nejsou identifikovatelné části takového díla odvozeny od programu a lze je rozumně považovat za nezávislá a samostatná díla sama o sobě, pak se tato licence a její podmínky na tyto části nevztahují, distribuujete-li je jako samostatná díla. Pokud ale distribuujete tytéž části jako součást celku, který představuje dílo založené na programu, musí distribuce celku podléhat podmínkám této licence, jejíž povolení pro ostatní držitele licence se rozšiřují na úplný celek, a tedy na každou jeho jednotlivou část, bez ohledu na to, kdo ji napsal.

Záměrem této části tedy není nárokovat práva na dílo napsané výhradně vámi, nebo tato vaše práva popírat. Cílem je spíše uplatnit právo kontrolovat distribuci odvozených nebo kolektivních děl založených na programu.

Kromě toho platí, že pouhé sdružení jiného díla, jež není založeno na programu, s programem (nebo dílem založeným na programu) na svazku ukládacího nebo distribučního média nepřevádí toho jiné dílo pod rámec této licence.

3. Program (nebo dílo na něm založené podle části 2) můžete kopírovat a distribuovat v objektovém kódu nebo spustitelné formě za podmínek částí 1 a 2 uvedených výše za předpokladu, že zároveň učiníte jedno z následujících:

1. dílo doplníte kompletním odpovídajícím strojově čitelným zdrojovým kódem, který musí být distribuován za podmínek částí 1 a 2 uvedených výše na médiu obvykle používaném k předávání softwaru; nebo
2. dílo doplníte písemnou nabídkou s alespoň tříletou platností na poskytnutí kompletní strojově čitelné kopie odpovídajícího zdrojového kódu jakékoli třetí straně za poplatek, který nepřevýší vaše náklady na fyzickou distribuci zdrojového kódu, k šíření za podmínek částí 1 a 2 uvedených výše na médiu obvykle používaném k předávání softwaru; nebo
3. dílo doplníte informací, kterou jste obdrželi v souvislosti s nabídkou na distribuci odpovídajícího zdrojového kódu. (Tato alternativa je dovolena pouze u nekomerční distribuce a pouze pokud jste program obdrželi v objektovém kódu nebo ve spustitelné formě s takovou nabídkou v souladu s odstavcem b uvedeným výše.) Zdrojový kód díla znamená formu díla, preferovanou pro provádění úprav díla. U spustitelného díla znamená kompletní zdrojový kód veškerý zdrojový kód všech modulů, které obsahuje, plus jakékoli doplňkové soubory s definicemi rozhraní plus skripty použité k řízení kompilace a instalace spustitelného díla. Existuje však zvláštní výjimka – distribuovaný zdrojový kód nemusí obsahovat nic

z toho, co je distribuováno normálně (ve zdrojové nebo binární formě) s hlavními komponentami (kompilátor, jádro atd.) operačního systému, v jehož prostředí je spustitelné dílo provozováno, pokud taková komponenta sama nedoprovází spustitelné dílo.

Pokud je distribuce spustitelné formy díla nebo objektového kódu řešena nabídnutím přístupu umožňujícího zkopírování z určeného místa, potom je nabídnutí ekvivalentního přístupu umožňujícího zkopírování zdrojového kódu ze stejného místa chápáno jako distribuce zdrojového kódu, ačkoli třetí strany nejsou nuceny kopírovat zdrojový kód spolu s kódem objektovým.

4. Program nesmíte kopírovat, upravovat, sublicencovat nebo distribuovat jinak, než jak je výslovně uvedeno v této licenci. Jakýkoli jiný pokus kopírovat, modifikovat, sublicencovat nebo distribuovat program je neplatný a automaticky zruší vaše práva daná touto licencí. Platí však, že stranám, které od vás obdržely kopie nebo práva v rámci této licence, nebudou jejich licence zrušeny, dokud budou tyto strany plně dodržovat licenční podmínky.

5. Nepožaduje se po vás, abyste licenci přijali, neboť jste ji nepodepsali. Nic jiného vám však nezaručí dovolení upravovat nebo distribuovat program nebo díla z něj odvozená. Tyto činnosti jsou zakázány zákonem, pokud nepřijmete tuto licenci. Modifikací nebo distribucí programu (nebo jakéhokoli díla založeného na programu) proto dáváte najevo přijetí této licence, abyste tak mohli činit, a všech jejích podmínek a okolností pro kopírování, distribuci nebo modifikaci programu nebo děl na něm založených.

6. Pokaždé, když program (nebo jakékoli dílo na programu založené) distribuujete dále, obdrží příjemce automaticky licenci původního poskytovatele licence pro kopírování, distribuci a modifikaci programu podléhající těmto podmínkám a okolnostem. Nesmíte uvalovat žádná další omezení na uplatňování práv, která jsou zde zaručena, příjemcem. Nejste odpovědní za prosazení dodržování této licence třetími stranami.

7. Pokud jsou na vás následkem soudního rozsudku nebo obvinění z porušení patentu nebo z jakéhokoli jiného důvodu (bez omezení na patentové otázky) uvaleny podmínky (ať už soudním příkazem, dohodou nebo jinak), které jsou v rozporu s podmínkami této licence, nezbavují vás povinnosti dodržovat podmínky této licence. Pokud nemůžete distribuci provádět tak, abyste zároveň vyhověli svým závazkům plynoucím z této licence a jakýmkoli jiným relevantním závazkům, potom v důsledku toho nesmíte program distribuovat vůbec. Pokud by například patentová licence všem, kdo získali kopie přímo nebo nepřímo od vás, zakazovala další distribuci programu bez autorských honorářů, pak jediný

způsob, jak vyhovět tomuto požadavku i této licenci, by bylo upustit zcela od distribuce programu.

Jestliže je nějaký úsek této části neplatný nebo nevynutitelný za nějakých konkrétních okolností, aplikuje se zbytek této části, a část jako celek se aplikuje za jiných okolností.

Účelem této části není navádět vás, abyste porušovali jakékoli patenty nebo jiné majetkoprávní nároky, nebo popírali platnost jakýchkoli takových nároků; jediným účelem této části je chránit integritu distribučního systému volného softwaru, který je implementován pomocí praktik veřejné licence. Mnoho lidí věnovalo štědré příspěvky na široké spektrum softwaru distribuovaného s využitím tohoto systému. Tito lidé se přitom spoléhali na konzistentní aplikaci systému. Záleží na autorovi/dárci, zda se rozhodne, že chce distribuovat software pomocí nějakého jiného systému, a držitel licence mu tuto volbu nemůže vnutit.

Účelem této části je důkladně vyjasnit, co je chápáno jako důsledek zbytku této licence.

**8.** Je-li distribuce a případně použití programu v některých zemích omezeno – buď patenty, nebo autorskými právy na rozhraní, pak původní držitel autorských práv, který zavede program pod tuto licenci, může přidat explicitní geografické omezení distribuce vyjímající tyto země, tak aby distribuce byla povolena jen ve státech, které nejsou takto vyňaty, a mezi nimi. V takovém případě obsahuje licence omezení, stejně jako by byla napsána v těle licence.

**9.** Nadace Free Software Foundation může čas od času publikovat revidované a případně nové verze licence General Public License. Takové nové verze budou svou povahou podobné verzi současné, ale mohou se lišit v drobnostech daných reakcí na nové problémy nebo zájmy.

Každé verzi je přiděleno charakteristické číslo verze. Pokud je v programu specifikováno číslo verze této licence, které se vztahuje k němu a *jakékoli další verzi*, máte možnost postupovat podle podmínek a okolností uvedených buď v dané verzi, nebo v jakékoli další verzi publikované nadací Free Software Foundation. Pokud v programu není specifikováno číslo verze této licence, můžete si zvolit jakoukoli její verzi, která kdy byla publikována nadací Free Software Foundation.

**10.** Jestliže chcete začlenit části programu do jiných volných programů, jejichž distribuční podmínky se liší, napište autorovi a požádejte jej o svolení. U softwaru, jehož autorská práva vlastní nadace Free Software Foundation, napište nadaci Free Software Foundation; zde někdy děláme výjimky. Naše rozhodnutí bude dáno dvěma cíli – zachováním volného statusu všech odvozenin z našeho volného softwaru a obecnou propagační sdílení a opětovného použití softwaru.

## ŽÁDNÁ ZÁRUKA

11. PROTOŽE JE LICENCE K PROGRAMU POSKYTOVÁNA ZDARMA, NENÍ NA TENTO PROGRAM POSKYTOVÁNA ŽÁDNÁ ZÁRUKA DO ROZSAHU POVOLENÉHO PLATNÝM ZÁKONEM. NENÍ-LI PÍSEMNĚ UVEDENO JINAK, DRŽITELÉ AUTORSKÝCH PRÁV A PŘÍPADNĚ JINÉ STRANY POSKYTUJÍ PROGRAM TAK JAK JE BEZ ZÁRUKY JAKÉHOKOLI DRUHU, AŽ VYJÁDŘENÉ EXPLICITNĚ ČI NIKOLI, VČETNĚ, ALE NIKOLI POUZE, IMPLICITNÍCH ZÁRUK PRODEJNOSTI A VHODNOSTI PRO URČITÝ KONKRÉTNÍ ÚČEL. CELÉ RIZIKO V SOUVISLOSTI S KVALITOU A VÝKONEM PROGRAMU LEŽÍ NA VÁS. POKUD SE UKÁŽE, ŽE JE PROGRAM VADNÝ, BERETE NA SEBE NÁKLADY NA VEŠKERÝ NEZBYTNÝ SERVIS, OPRAVY NEBO KOREKCE.

12. ŽÁDNÝ DRŽITEL AUTORSKÝCH PRÁV NEBO JAKÁKOLI JINÁ STRANA, KTERÁ MŮŽE MODIFIKOVAT A POPŘÍPADĚ DÁLE DISTRIBUOVAT PROGRAM TAK, JAK JE POVOLENO VÝŠE, NEBUDE V ŽÁDNÉM PŘÍPADĚ ODPOVĚDNÁ ZA ŠKODY VÁM ZPŮSOBENÉ, VČETNĚ JAKÝCHKOLI OBECNÝCH, ZVLÁŠTNÍCH, NÁHODNÝCH NEBO VYPLÝVAJÍCÍCH ŠKOD VZNIKLYCH Z POUŽITÍ PROGRAMU NEBO NEMOŽNOSTI JEJ POUŽÍT, LEDAŽE BY TO VYŽADOVAL PLATNÝ ZÁKON NEBO TAK BYLO DOHODNUTO PÍSEMNOU FORMOU (VČETNĚ, ALE NIKOLI POUZE, ZTRÁTY DAT, NEBO PORUŠENÍ PŘESNOSTI DAT, NEBO ZTRÁT, KTERÉ JSTE UTRPĚLI VY NEBO TŘETÍ STRANY, NEBO SELHÁNÍ PROGRAMU PŘI PROVOZU S JAKÝMKOLI JINÝMI PROGRAMY), DOKONCE I KDYŽ TAKOVÝ DRŽITEL NEBO JINÁ STRANA BYLI POUČENI O MOŽNOSTI VZNIKU TAKOVÝCH ŠKOD.



# Slovník pojmů

## **ACL (Access Control List)**

Rozšíření klasického systému přístupových práv k souborům a adresářům. Umožňuje přesnější nastavení přístupových práv.

## **adresář (v souborovém systému)**

Struktura obsahující soubory či další adresáře (podadresáře). Adresáře v souborovém systému jsou hierarchicky uspořádány ve stromové struktuře sloužící k organizaci souborů.

## **ADSL (Asymmetric Digital Subscriber Line)**

Rychlý přenosový protokol využívající telefonní síť.

## **AGP (Accelerated Graphics Port)**

Vysokorychlostní sběrnice pro grafické karty poskytující větší šířku přenosového pásma než PCI. Grafické karty založené na AGP mohou přímo, bez účasti procesoru, přistupovat do operační paměti počítače (RAM).

## **ATAPI (Advanced Technology Attachment Packet Interface)**

ATAPI je typ CD-ROM mechaniky, který se připojuje řadiči E(IDE). Kromě mechanik typu ATAPI existují i SCSI CD mechaniky, připojené přes SCSI řadič.

## **BIOS**

Malý program spuštěný po zapnutí nebo restartu počítače. Je zodpovědný za inicializaci hardwarových komponent. Většina BIOSů umožňuje úpravy nízkourovňových systémových parametrů přes interaktivní uživatelské rozhraní. Programový kód je uložen v paměti typu ROM.

**cesta**

Nezaměnitelný popis umístění souboru v systému souborů.

**CPU (Central Processing Unit)**

Viz procesor.

**DDC (Direct Display Channel)**

Standard používaný při komunikaci mezi monitorem a grafickou kartou, pomocí kterého jsou kartě předávány různé parametry, například typ nebo rozlišení monitoru.

**démon**

Démon je program, který běží na pozadí a v případě potřeby se aktivuje. Např. HTTP démon (httpd) odpovídá na HTTP požadavky.

**DNS (Domain Name System)**

Protokol pro převod jmenných adres na IP adresy a naopak.

**domovský adresář**

Soukromý adresář v souborovém systému patřící určitému uživateli (obvykle /home/<uživatelskéjméno>). Do domovského adresáře má plný přístup pouze uživatel, kterému patří, a superuživatel root.

**e-mail (elektronická pošta)**

Elektronický způsob přenosu pošty mezi uživateli v síti. E-mailová adresa se zapisuje ve formátu `uživatel@domena.org`.

**EIDE (Enhanced Integrated Drive Electronics)**

Vylepšení standardu IDE, které umožňuje použití pevných disků s kapacitou větší než 512 MB.

**ethernet**

Rozšířený standard používaný pro přenos dat v lokálních počítačových sítích.

**EXT2 (Second Extended File System)**

Souborový systém podporovaný Linuxem.

**FAQ (Frequently Asked Questions)**

Zkratka pro dokumenty s odpověďmi na často kladené otázky.



## **firewall**

Mechanismus pro filtraci síťového provozu chrání lokální síť proti nepovolenému přístupu z vnějšku.

## **FTP (File Transfer Protocol)**

Protokol pro přenos souborů po síti založený na TCP/IP.

## **GNOME (GNU Network Object Model Environment)**

Uživatelsky přívětivé grafické pracovní prostředí pro Linux.

## **GNU (GNU is Not Unix)**

GNU je projekt *Nadace pro svobodný software (Free Software Foundation)*. Cílem projektu GNU je vytvořit svobodný operační systém kompatibilní s Unixem. Podstatné přitom není, aby byl systém k dispozici *zdarma*, ale aby s ním bylo možno *svobodně* nakládat: volně distribuovat, měnit a modifikovat. Dnes již klasický Manifest GNU (<http://www.gnu.org/gnu/manifesto.html>) vysvětluje myšlenky, na kterých je projekt postaven. Právně je GNU software chráněn Obecnou veřejnou licencí GNU neboli *GPL* (<http://www.gnu.org/copyleft/gpl.html>) a Obecnou knihovní licencí GNU neboli *LGPL* (<http://www.gnu.org/copyleft/lgpl.html>). Jádro Linuxu, které je šířeno pod licencí *GPL*, z projektu GNU (zejména z nástrojů vyvinutých v jeho rámci) těží, ale není s ním totožné.

## **GPL (GNU General Public License)**

Viz GNU.

## **GRUB (GRand Unified Bootloader)**

Malý program instalovaný v zaváděcím sektoru pevného disku, který umožňuje spustit nejen Linux, ale případně i další operační systémy.

## **hostname**

Jméno počítače. V Linuxu je to obvykle jméno, pod kterým je počítač dosažitelný na síti.

## **HTML (Hypertext Markup Language)**

Nejdůležitější jazyk používaný pro tvorbu obsahu webových stránek. Formátovací příkazy jazyka HTML určují vzhled dokumentu a jeho zobrazení v prohlížeči.

**HTTP (Hypertext Transfer Protocol)**

Síťový protokol definující způsob žádostí a přenos webových dokumentů. Dokumenty jsou obvykle HTML stránky nabízené serverem a vyžádané uživatelem pomocí prohlížeče.

**IDE (Integrated Drive Electronics)**

Standard pro připojení pevných disků.

**Internet**

Celosvětová počítačová síť založená na komunikačním protokolu TCP/IP.

**interpret příkazů**

Interaktivní program umožňující zadávat příkazy. Existuje několik různých interpretů příkazů, jako např. Bash, Zsh a tcsh. Každý má svůj zvláštní programovací jazyk. Shell je jiný název pro interpret příkazů.

**IP adresa**

Jedinečná 32-bitová adresa počítače v TCP/IP síti. Obvykle zapisovaná jako čtyři čísla oddělená tečkami (např. 192.168.10.1).

**IRQ (Interrupt Request)**

Požadavek na nějakou akci vyslaný hardwarovou součástí nebo programem. Většinu IRQ obsluhuje operační systém.

**ISDN (Integrated Services Digital Network)**

Rozšířený standard pro rychlý digitální přenos dat po telefonní síti.

**jádro**

Jádro (též kernel) je základem operačního systému. Alokuje paměť, spravuje souborové systémy, obsahuje ovladače, které umožňují komunikaci s hardwarem, a řídí procesy a síťování.

**KDE (K Desktop Environment)**

Uživatelsky přívětivý grafický desktop pro Linux.

**klient**

Program nebo počítač v síťovém prostředí, který se připojuje k a požaduje informace ze serveru.

## **konzole**

Dříve synonymum pro terminál. V Linuxu je několik *virtuálních konzolí*, které umožňují používat obrazovku pro několik navzájem nezávislých souběžných sezení bez použití grafického rozhraní.

## **kořenový adresář**

Základní adresář v systému souborů. V UNIXu je kořenový adresář označován jako /.

## **kurzor**

Kurzor je znak, který označuje místo pro vložení textu.

## **LAN (Local Area Network)**

LAN je místní počítačová síť, obvykle poměrně malá.

## **LILO (Linux Loader)**

Malý program instalovaný v zaváděcím sektoru pevného disku, který spouští Linux nebo jiný operační systém.

## **Linux**

Vysoce výkonný operační systém unixového typu distribuovaný volně za podmínek daných GNU GPL licencí. Název Linux odkazuje na tvůrce systému (zkratka z *LINUsův uniX*, jímž je Linus Torvalds. Ačkoliv se termín Linux v užším slova smyslu vztahuje pouze na vlastní jádro, obecně se obvykle používá pro označení celého operačního systému.

## **manuálové stránky**

Tradiční dokumentace unixových systémů. Číst ji lze zadáním příkazu `man`. Slouží jako referenční příručka.

## **MBR (Master Boot Record)**

První fyzický sektor pevného disku, jehož obsah je nahrán do paměti a spuštěn BIOSem. Spuštěný kód nahraje z diskového oddílu operační systém a nebo spustí důmyslnější zavaděč, například LILO nebo GRUB.

## **MD5**

Algoritmus pro generování kontrolních součtů. Kontrolní součty jsou vytvářeny způsobem, který v podstatě znemožňuje vytvořit soubor s jiným obsahem ale stejným kontrolním součtem.

### **MP3**

Velmi účinný způsob komprese zvukových dat, který přináší až desetinásobné zmenšení souboru ve srovnání s nekomprimovaným zvukovým souborem.

### **multitasking**

Schopnost operačního systému spouštět více programů současně,

### **NFS (Network File System)**

Protokol pro přístup k souborovému systému sdílenému po síti.

### **NIS (Network Information Service)**

Centrální systém pro administraci uživatelů v sítích. Umožňuje správu uživatelských jmen a hesel v celé síti.

### **oddíl**

Logicky nezávislá část pevného disku, která může obsahovat jiný souborový systém nebo odkládací místo (swap).

### **odhlášení**

Proces ukončení interaktivní relace v linuxovém systému.

### **odkaz**

Odkaz je ukazatel na soubor, hojně používaný na internetu i v linuxovém souborovém systému. V Linuxu se rozlišují *pevné odkazy* (hard link) a *symbolické odkazy*. Zatímco *pevné odkazy* ukazují na přesnou pozici v souborovém systému, symbolické odkazy ukazují pouze na příslušné jméno.

### **operační paměť**

Fyzická paměť s velmi rychlým přístupem. Často se označuje též jako RAM (Random Access Memory).

### **operační systém**

Viz kernel.

### **ovladač**

Část operačního systému zodpovědná za komunikaci s hardwarovými komponentami.

**plug and play**

Protokol pro automatickou detekci a konfiguraci hardwarových komponent.

**proces**

Běžící program, někdy též nazývaný úloha.

**procesor**

Procesor (CPU) je mikročip provádějící strojový kód uložený v operační paměti. Je to mozek počítače.

**prohlížeč**

Program zobrazující obsah lokálních souborů nebo webových stránek.

**proměnná prostředí**

Prvek prostředí interpretu příkazů.

**prostředí**

Sada proměnných prostředí a jejich hodnot udržovaná interpretem příkazů (shellem). Uživatel může měnit (a rušit) hodnoty existujících proměnných prostředí a nastavovat nové proměnné. Trvalé přiřazení hodnot lze zajistit pomocí konfiguračních souborů interpretu příkazů.

**protokol**

Standard určující pravidla hardwarové, softwarové nebo síťové komunikace. Protokolů existuje velké množství. Mezi nejznámější patří HTTP a FTP.

**proxy**

Obvykle počítač, který slouží jako mezisklad dat přenášných z internetu. Pokud je jeden dokument vyžadován vícekrát, podruhé může být dodán mnohem rychleji. Počítače, které toho chtějí využívat, musí být nastaveny tak, aby požadavky vysílaly přes proxy.

**přihlášení**

Autentizace uživatele pomocí uživatelského jména a hesla nutná k přístupu do počítače nebo sítě.

**příkazová řádka**

Textový režim zadávání příkazů počítači.

**připojení (přimontování)**

Vložení souborového systému do systémového adresářového stromu.

**přístupová práva**

Přístupová práva souboru určují, zda může uživatel nebo skupina číst, zapisovat nebo spouštět soubor či adresář. Přístupová práva obvykle nastavuje systémový administrátor.

**RAM (Random Access Memory)**

Viz operační paměť.

**ReiserFS**

Souborový systém, který umožňuje rychlou opravu případných nesrovnalostí, které se mohou objevit, pokud není souborový systém před vypnutím operačního systému správně odpojen. ReiserFS je optimalizovaný pro práci s velkým množstvím malých souborů.

**root**

Účet superuživatele. Superuživatel má všechna práva. Tento účet je používán pro administraci a neměl by být používán pro běžnou práci.

**SCSI (Small Computer Systems Interface)**

Standard používaný pro připojení pevných disků a dalších zařízení, např. skenerů nebo páskových mechanik.

**server**

Počítač nebo program, jehož úkolem je nabízet služby, obvykle po síti. Službou může být například nabízení souborů, překlad jmen nebo vykreslování grafiky.

**síť**

Síť představuje propojení více počítačů umožňující přenášet mezi nimi data. Počítač posílající požadavek se často označuje jako klient, odpovídající počítač (např. posílající dokument) jako server.

**SMTP (Simple Mail Transfer Protocol)**

Protokol pro přenos elektronické pošty po síti.

**správce oken**

Program běžící nad X Window systémem umožňující akce jako změna velikosti oken nebo jejich přesun. Je také zodpovědný za dekorace oken, tj. jejich titulky, okraje atd. Chování správce oken lze nastavit podle potřeb uživatele.

**SSH (Secure SHell)**

Program pro vzdálené přihlášení využívající šifrování. Je to bezpečnější alternativa telnetu.

**SSL (Secure Socket Layer)**

Šifrovaný protokol pro přenos HTTP dat.

**startování**

Proces od zapnutí počítače až do chvíle, kdy je systém připraven k použití.

**superuživatel**

Viz root.

**swap**

Swap, jinak též odkládací místo. Diskový oddíl používaný k ukládání momentálně nepotřebných stránek paměti.

**systémový administrátor**

Viz root.

**šířka pásma**

Nejvyšší použitelná kapacita přenosového kanálu. Obvykle se používá v souvislosti se sítíovými spoji.

**TCP/IP**

Internetový komunikační protokol, který se stále častěji používá i v lokálních sítích.

**telnet**

Telnet je protokol pro komunikaci se vzdálenými počítači. Pro vzdálené přihlášení byl nahrazen SSH, které nabízí bezpečnější šifrované spojení.

**terminál**

Původně označení pro kombinaci klávesnice a monitoru připojenou k centrálnímu počítači. Dnes se tak na pracovních stanicích označují programy (xterm atd.), které emulují skutečný terminál.

**Tux**

Jméno tučňáka, maskota Linuxu (viz <http://www.sjbaker.org/tux/>).

**účet**

Účet je definovaný uživatelským nebo přihlašovacím jménem a heslem. Účet odpovídá uživatelskému ID (UID).

**úloha**

Viz proces.

**UNIX**

UNIX je (ochranná známka a) typ operačního systému.

**URL (Uniform Resource Locator)**

Určení zdroje v síti. Sestává z označení protokolu (např. `http://`), jména počítače a domény (např. `www.suse.cz`) a dokumentu (např. `cz/company/index.html`). Celá URL pak vypadá takto: `http://www.suse.cz/cz/company/index.html`.

**uživatelský adresář**

Viz domovský adresář.

**VESA (Video Electronics Standard Association)**

Průmyslové konsorcium, které mimo jiné určuje důležité standardy pro zobrazovací systémy.

**víceuživatelský**

Víceuživatelské systémy umožňují současnou práci několika uživatelů.

**výzva**

Krátký (nastavitelný) řetězec znaků zobrazený na začátku každé příkazové řádky. Obvykle obsahuje informaci o aktuálním pracovním adresáři.



## **WWW (World Wide Web)**

Web je soubor navzájem provázaných hypertextových dokumentů, obrázků a dalších souborů dostupných pomocí protokolu HTTP. Web je možné procházet a prohlížet pomocí specializovaného programu, kterému se říká webový prohlížeč.

## **X Window System**

X Window je síťový systém pro zobrazení grafického uživatelského rozhraní na různých počítačových platformách. Poskytuje základní grafické prvky jako čáry a pravoúhelníky. Je to zprostředkující vrstva mezi hardwarem a správcem oken.

## **X11**

Verze 11 X Window systému.

## **YaST (Yet another Setup Tool)**

YaST je nástroj pro pohodlnou instalaci a nastavení SUSE Linuxu.

## **YP**

Viz NIS.

## **záloha**

Záloha je kopie dat vytvořená za účelem obnovy v případě jejich ztráty nebo poškození. Záloha všech důležitých souborů by měla být vytvářena pravidelně.

## **záložka**

Položka ve sbírce URL adres.

## **zástupný znak**

Zástupný znak nahrazuje jeden (symbol: ?) nebo více (symbol: \*) neznámých znaků. Též metaznak. Zástupné znaky jsou součástí regulárních výrazů.



# Index

## Symboly

úroveň běhu .....	<i>viz</i> runlevel, <i>viz</i> runlevel
Řídicí středisko .....	37
časová zóna .....	60
často kladené dotazy .....	621
šifrování	
- oddíly .....	564
- soubory .....	564
64-bitový Linux .....	141
- podpora běhu aplikací .....	142
- specifikace jádra .....	144
- vývoj softwaru .....	143

## A

ACLs .....	581–592
- definice .....	582
- přístupové bity .....	584
- podpora .....	592
- používání .....	583
ACLs	
- kontrolní algoritmus .....	591
- masky .....	587
- přístup .....	585
- struktura .....	583
- výchozí .....	588
ACPI	
- vypnutí .....	6
adresa	
- IP .....	370
- MAC .....	371
aktualizace .....	107–111, 129
- na vyšší verzi .....	42
- online .....	38–41

- passwd a group .....	108
- systému .....	42
- YaST .....	109
- z CD .....	42
- z příkazové řádky .....	40
- zálohování .....	108
- zvukové směšovače .....	118
Apache .....	477–498
- apxs .....	482
- bezpečnost .....	495–496
- CGI .....	489
- content negotiation .....	480
- DocumentRoot .....	484
- flags .....	483
- instalace .....	481
- konfigurace .....	482–487
- logování .....	486, 487
- moduly .....	479
· aktivace .....	483
· mod_perl .....	490
· mod_php4 .....	492
· mod_python .....	492
· mod_ruby .....	492
· nahrávání .....	484
- návěští .....	483
- práva .....	496
- problémy .....	497
- Squid .....	545
- SSI .....	486, 488
- výchozí stránka .....	479
- virtuální servery .....	480, 492–495
autentizace	
- PAM .....	357–364

## B

balíky	
- RPM	<i>viz</i> RPM
Bash	
- .bashrc	198
- .profile	198
- profil	198
bezpečnost	56, 57, 567–577
- útoky	573–574
- červi	574
- šifrovaný souborový systém	265
- chyby	570, 573
- DNS	574
- firewall	552
- hesla	568–569
- hlášení problémů	577
- inženýrství	567
- lokální	568–571
- nastavení	56–57
- práva	570
- RPM podpisy	576
- sériové terminály	568
- síť	571–574
- Samba	523
- Squid	532
- SSH	559–564
- startování	568
- tcpd	577
- tipy a triky	575
- viry	571
- X	572
BIND	413–425
BIOS	5
Bluetooth	264, 317–327
- bluez	318
- hciconfig	323
- síť	320
- YaST	319
bootdisk	58
booting	644
bttv	50

## C

CardBus	<i>viz</i> hardware, CardBus
cardmgr	270
CD	
- zavádění systému	4
CD mechaniky	
- podporované	93
chybové zprávy	

- bad interpreter	66
- permission denied	66
chyby	
- hlášení	628
CJK	208
coldplug	337
commands	
- jfs_fsck	644
crashes	641, 644
cron	198
cryptofs	564
CVS	500, 507–509

## D

deltarpm	124
DHCP	441–448
- balíčky	443
- dhcpd	444–446
- konfigurace pomocí YaST	442
- server	444–446
- statické přiřazování adres	446
digitální fotoaparáty	265
disk	
- defragmentace	624
- hdparm	294
- přidání	625
- rozdělování	68
- správa napájení	294
disketa	
- formátování	91
- s moduly	58
- startovací	58, 90, 91
- rawrite	91
- záchranná	58
diskové oddíly	68, 622
- šifrování	564
- fdisk	180
- LVM	64
- parametry	64
- RAID	64
- swap	65
- tabulka diskových oddílů	164
- typy	11
- vytváření	11, 64
- vytvoření	63
- změna velikosti Windows	16
DMA	46
DNS	381
- řešení problémů	415
- bezpečnost	574

- BIND .....	413–425
- domény .....	396
- konfigurace .....	409
- logování .....	420
- Mail Exchanger .....	382
- nameservery .....	396
- NIC .....	381
- options .....	418
- přeposílání .....	415
- server .....	55
- spouštění .....	415
- squid .....	536
- top level domain .....	381
- volby .....	418
- zóny .....	420
· soubory .....	421
dokumentace .....	617
- info stránky .....	618
- LDP .....	617
- man stránky .....	618
Domain Name System .....	<i>viz</i> DNS
DOS .....	
- sdílení souborů .....	519
drift soubor .....	450
driver na CD .....	75

## E

e-mail .....	<i>viz</i> pošta
e2fsck .....	636
editor úrovní běhu .....	158
editory .....	
- Emacs .....	203–204
- vi .....	204
Emacs .....	203–204
- .emacs .....	203
- default.el .....	203
Evolution .....	266

## F

FAQ .....	621
FHS .....	
- SGML .....	114
- XML .....	114
file systems .....	
- jfs_fsck .....	644
filtrování paketů .....	<i>viz</i> firewall
firewall .....	57, 552
- filtrování paketů .....	552, 554
- Squid .....	543
- SuSEfirewall2 .....	552, 555

Firewire (IEEE1394) .....	
- disky .....	265
flash disky .....	265

## G

GNOME .....	
- příkazy .....	626
GPL .....	647
grafické karty .....	
- 3D .....	234–236
· instalační podpora .....	236
· ovladače .....	234
· podpora .....	234
- ovladače .....	227
grafické uživatelské rozhraní .....	214–222
grafika .....	
- 3D .....	234–236
· 3Ddiag .....	235
· diagnostika .....	235
· problémy .....	235
· SaX .....	234
· testování .....	235
- GLIDE .....	234–236
- OpenGL .....	234–236
· ovladače .....	234
· testování .....	235
GRUB .....	72, 163–185
- /etc/grub.conf .....	173
- řešení problémů .....	183
- GRUB Geom Error .....	184
- GRUB shell .....	174
- heslo pro zavedení .....	174
- informace .....	185
- JFS a GRUB .....	184
- jména oddílů .....	168
- jména zařízení .....	168
- menu .....	166
- odinstalace .....	179
- omezení .....	165
- parametry jádra .....	171
- správa spouštění .....	164
- start z kombinovaného IDE/SCSI .....	
· systému .....	184
- zástupné znaky .....	172

## H

harddisk .....	<i>viz</i> disk
hardware .....	
- CardBus .....	270
- CD mechanika .....	45

- informace .....	45
- ISDN .....	387
- karta PCMCIA .....	270
- konfigurace .....	45
- podporovaný .....	621
- SCSI zařízení .....	94
HDD .....	<i>viz disk</i>
hotplug .....	270, 331–338
- agent .....	333, 334
· PCI .....	336
· rozhraní .....	334
· USB .....	335
· zařízení .....	334
- analýza chyb .....	337
- blacklist .....	336
- jména zařízení .....	332
- log soubory .....	337
- mapové soubory .....	335
- moduly .....	
· automatické nahrávání .....	335
- PCI .....	336
- síťová zařízení .....	334
- události .....	333
- whitelist .....	336
- zařízení pro ukládání dat .....	335
- zapisovač událostí .....	338

## I

I18N .....	208
IDE DMA .....	46
inetd .....	55, 110
info stránky .....	201, 618
informace o hardwaru .....	45
init .....	151
- skripty .....	154–157
- vkládání skriptů .....	156
instalační podpora .....	74
- 3D grafické karty .....	236
instalační zdroj .....	37
instalace .....	
- balíků .....	43
- bezpečné nastavení .....	7
- GRUB .....	166
- RPM .....	43
- ruční .....	118
- softwaru .....	43
- textový režim .....	88
- VNC .....	87
- YaST .....	3–33
- ze sítě .....	93

Internet .....	51
- cinternet .....	403
- dial-up .....	402–404
- DSL .....	390
- ISDN .....	387
- kinternet .....	403
- qinternet .....	403
- smpppd .....	402–404
- T-DSL .....	391
- webový server .....	<i>viz Apache</i>

IP adresa .....	370
- třídy adres .....	371

IP adresy .....	
- dynamické přidělování .....	441
- IPv6 .....	
· konfigurace .....	380
- maškaráda .....	553
- privátní .....	373
IrDA .....	264, 327–329
- konfigurace .....	328
- spuštění .....	328
- zastavení .....	328

## J

jádro .....	188–195
- 2.6 .....	112
- cache .....	202
- démon .....	192
- instalace .....	194
- kmod .....	192
- kompilace .....	193
- konfigurace .....	189–193
- modprobe.conf .....	192
- moduly .....	190–192
· kompilace .....	193
· síťové karty .....	382
- omezení .....	354
- příliš velké .....	193
- překlad .....	188, 193
- parametry .....	188
- update .....	188
- zavaděč modulů .....	192
- zdrojové kódy .....	189
jade .....	<i>viz SGML, openjade</i>
jazyk .....	
- výběr .....	60
jfs_fsck .....	644
jmenný server .....	
- DNS .....	409
joystick .....	46

- konfigurace ..... 222

## K

### kódování

- UTF-8 ..... 113  
- výchozí ..... 113

křížový ovladač ..... 46

karta PCMCIA ..... 270

### karty

- grafické ..... 216  
  · ovladače ..... 227  
- síť ..... 382

### KDE

- příkazy ..... 626

### klávesnice

- klávesy ..... 60  
- mapování ..... 207  
  · kombinace kláves ..... 208  
  · skládání ..... 208  
- rozložení ..... 207  
- X rozšíření klávesnice ..... 208  
- XKB ..... 208

Kmod ..... viz jádro, zavaděč modulů

konfigurační soubory ..... 62, 395

- .bashrc ..... 198, 202  
- .emacs ..... 203  
- .htaccess ..... 485  
- .mailsync ..... 515  
- .profile ..... 198  
- .xsession ..... 563  
- /boot/GRUB/menu.lst ..... 166  
- /etc/grub.conf ..... 173  
- /etc/gshadow ..... 113  
- /etc/inittab ..... 151  
- /etc/powersave.conf ..... 117  
- acpi ..... 289  
- adresář sysconfig ..... 62  
- apache2 ..... 483  
- config ..... 189, 395  
- crontab ..... 198  
- csh.cshrc ..... 210  
- dhclient.conf ..... 444  
- dhcp ..... 395  
- dhcpd.conf ..... 444  
- exports ..... 439  
- fstab ..... 66, 136, 625  
- group ..... 108  
- host.conf ..... 398  
  · alert ..... 398  
  · multi ..... 398

  · nospoof ..... 398  
  · order ..... 398  
  · trim ..... 398

- HOSTNAME ..... 401

- hosts ..... 381, 397

- hotplug ..... 332

- httpd.conf ..... 482, 484

- hwdm ..... 336

- hwup ..... 334

- ifcfg-\* ..... 395

- inittab ..... 151, 207

- inputrc ..... 208

- irda ..... 328

- jazyk ..... 208, 210

- kernel ..... 149

- logrotate.conf ..... 200

- modprobe.conf ..... 192

- named.conf ..... 414, 416–425, 536

- networks ..... 397

- nscd.conf ..... 401

- nsswitch.conf ..... 399, 469

- ntp.conf ..... 450

- pam\_unix2.conf ..... 469

- passwd ..... 108

- powersave ..... 289

- práva ..... 575

- profil ..... 198

- profile ..... 201

- profily ..... 210

- resolv.conf ..... 203, 396, 414, 534

- routes ..... 395

- samba ..... 524

- services ..... 524, 543

- slapd.conf ..... 460

- smb.conf ..... 519, 520

- smppd.conf ..... 403

- smpppd-c.conf ..... 403

- squid.conf ... 535, 537, 540, 542, 545, 547

- squidguard.conf ..... 547

- sshd\_config ..... 564

- sysconfig ..... 159, 160

- termcap ..... 208

- wireless ..... 395

- XF86Config .. viz konfigurační soubory,

  xorg.conf

- xorg.conf ..... 119, 223

  · Device ..... 227

  · Monitor ..... 228

  · obrazovka ..... 225

konfigurace ..... 159

- Řídící středisko .....	37
- Apache .....	482–487
- bezpečnosti .....	56–57
- CD mechanika .....	45
- disku .....	68
- DNS .....	409
- DSL .....	390
- grafické karty .....	216
- GRUB .....	166
- hardwaru .....	45
- IPv6 .....	380
- IrDA .....	328
- ISDN .....	387
- joystick .....	222
- kabelového modemu .....	389
- Linuxu .....	62
- modemu .....	385
- monitoru .....	214
- myši .....	47
- NFS .....	54
- NTP	
· klienta .....	55
- routing .....	395
- síť .....	51–55, 382
· manuální .....	392–402
- Samba .....	520–524
· klient .....	527
· klienta .....	55
· serveru .....	55
- skeneru .....	47
- směrování .....	395
- softwaru .....	37
- Squid .....	537
- SSH .....	559
- systému .....	35–75
- T-DSL .....	391
- tisk .....	241–243
- X .....	214
- zavaděče .....	72
Kontakt .....	266
konzole	
- grafická	
· vypnutí .....	183
- přepínání .....	207
- počte .....	207
KPilot .....	266
KPowersave .....	262
KSysguard .....	263

## L

L10N .....	208
LDAP .....	455–476
- úprava dat .....	467
- ACL .....	461
- administrace	
· skupin .....	474
· uživatelů .....	474
- adresářový strom .....	458
- konfigurace serveru .....	460
- kontrola přístupu .....	463
- ldapadd .....	465
- ldapdelete .....	468
- ldapmodify .....	467
- ldapsearch .....	468
- mazání dat .....	468
- vkládání dat .....	465
- vyhledávání dat .....	468
- YaST	
· moduly .....	470
- YaST LDAP klient .....	468
LFS soubory	
- velikost .....	353
licence .....	<i>viz</i> GPL
Lightweight Directory Access Protocol .....	<i>viz</i>
LDAP	
LILO .....	165
- konfigurace .....	72
- odinstalace .....	179
Linux	
- odinstalace .....	179, 622
- síť .....	367
- sdílení souborů s jiným OS .....	519
linuxrc .....	85
- ruční instalace .....	118
loader .....	<i>viz</i> zavaděče
locale	
- UTF-8 .....	113
logování .....	420
- logrotate	
· nastavení .....	200
logrotate .....	199
logy .....	626
- apache2 .....	487, 497
- httpd .....	486, 487, 497
- Squid .....	535, 538, 544
- startování .....	75
- systémový .....	75
- Unison .....	507
- X.org .....	235



- zprávy .....	415
LSB (Linux Standard Base)	
- instalace balíků .....	120
LVM	
- YaST .....	95

## M

maškaráda .....	553
- konfigurace pomocí SuSEfirewall2 ..	555
manuálové stránky .....	201, 618
Master Boot Record .....	<i>viz</i> MBR
MBR .....	164
- obnova .....	180
mirrory .....	628
mobilita .....	259–267
- digitální fotoaparáty .....	265
- externí disky .....	265
- Firewire (IEEE1394) .....	265
- kapesní počítače .....	266
- mobily .....	266
- notebooky .....	260
- ochrana dat .....	265
- PDA .....	266
- USB .....	265
mobily .....	266
modemy	
- kabelové .....	389
- YaST .....	385
moduly	
- příkazy .....	191
- překlad .....	193
- zacházení .....	191
monitor	
- nastavení .....	214
monitorování systému .....	262
- KPowerSave .....	262
- KSysguard .....	263
mountd .....	438, 440
myš	
- konfigurace .....	47

## N

nápověda	
- FAQ .....	621
- info stránky .....	201
- manuálové stránky .....	201
- X11 .....	228
named .....	415
nameserver .....	<i>viz</i> DNS
- BIND .....	413

nastavení .....	62
- grafické karty .....	216
- joystick .....	222
- PAM .....	119
- sítě .....	382
- SSH .....	559
- tisk .....	241
- X .....	214

NAT .....	<i>viz</i> maškaráda
nestabilní systém .....	627
NetBIOS .....	520
Network Information Service .....	<i>viz</i> NIS
NFS .....	435
- export .....	438
- export souborů .....	437
- firewall .....	432, 438
- import souborů .....	436
- klient .....	54
- mount .....	436
- oprávnění .....	439
- připojení .....	436
- server .....	54, 437
nfsd .....	438, 440
NIS .....	429–433
- klient .....	432
- master .....	429–432
- slave .....	429–432
notebooky .....	260–265, 269
- hardware .....	260
- IrDA .....	327–329
- PCMCIA .....	260
- SCPM .....	261, 277
- SLP .....	262
- správa napájení .....	260, 285–295
- správa profilů .....	277
NSS .....	399
- databáze .....	399
NTP	
- klient .....	55
nVidia .....	110

## O

obrazovka	
- rozlišení .....	226
ochrana dat .....	265
oddíly	
- fstab .....	66
odinstalace	
- GRUB .....	179
- LILO .....	179

- Linuxu .....	179
odkládací oddíl .....	69
odstranění softwaru .....	43
OpenSSH .....	<i>viz SSH</i>
oprava systému .....	131
OS/2	
- sdílení souborů .....	519
ověřování	
- Kerberos .....	119
ovladače na CD .....	75

## P

písmo .....	229
- CID-keyed .....	233
- X11 core .....	232
- Xft .....	229
příkazy	
- chown .....	113
- depmod .....	191
- e2fsck .....	636
- fdisk .....	180
- fonty-konfigurace .....	229
- free .....	202
- hciconfig .....	323
- hdparm .....	294
- head .....	113
- hotplug .....	333
- hwinfo .....	336
- insmod .....	191
- ldapadd .....	466
- ldapdelete .....	468
- ldapmodify .....	467
- ldapsearch .....	468
- lp .....	246
- lsmod .....	192
- modinfo .....	192
- modprobe .....	192
- nice .....	113
- online_update .....	79
- rmmod .....	191
- rpm .....	120
- rpmbuild .....	111, 120, 127
- scp .....	560
- sftp .....	561
- slptool .....	406
- smbpasswd .....	525
- sort .....	113
- ssh .....	560
- ssh-agent .....	563
- ssh-keygen .....	562
- sx .....	111
- tail .....	113
- udev .....	339
- xfs_check .....	641
přístupová práva	
- ACLs .....	582–592
- přístupová práva k souborům .....	200
- Samba .....	523
připojení k síti .....	51
připojovatelné autentizační moduly ..	<i>viz PAM</i>
PAM .....	357–364
- nastavení .....	119
paměť	
- RAM .....	202
parametry jádra .....	171
PCMCIA .....	260, 270
- Cardmanager .....	270
- cardmgr .....	270
- Ethernet .....	271
- IDE .....	272
- IrDA .....	327–329
- ISDN .....	271
- konfigurace .....	271–275
- modem .....	272
- problémy .....	272
- SCSI .....	272
- software .....	270
- Token Ring .....	271
PCMCIA karty ..	<i>viz hardware, karta PCMCIA</i>
PDA .....	266
pevný disk .....	<i>viz disk</i>
pošta .....	52
- konfigurace .....	52
- MTA .....	52
- postfix .....	52
- sendmail .....	52
- soubory .....	501
- mailsync .....	514–517
- synchronizace .....	263
podpora	
- vytvoření dotazu .....	74
portmap .....	438
porty	
- 53 .....	419
- skenování .....	544
postfix .....	52
power management .....	<i>viz správa napájení</i>
powersave .....	295
- konfigurace .....	296
- probuzení .....	298

- standby .....	298
- suspend .....	298
- uspání .....	298
procmail .....	52
program	
- překlad .....	127
programy	
- instalace .....	626
proměnné	
- prostředí .....	208
protokolové soubory	
- Unison .....	507
protokoly	
- FTP .....	478
- HTTP .....	478
- HTTPS .....	478
- ICMP .....	368
- IGMP .....	368
- IPv6 .....	373
- LDAP .....	455
- SLP .....	405
- SMB .....	520
- TCP/IP .....	368
- UDP .....	368
proxy .....	<i>viz Squid</i> , 532
- transparentní .....	532, 542
- výhody .....	531

## R

RAID	
- softwarový .....	100
- YaST .....	100
reiserfsck .....	631
reverzní převod .....	424
RFC .....	368
routing .....	395–396
routování .....	<i>viz</i> směrování
rozložení kláves .....	60
RPC mount démon .....	438
RPC NFS démon .....	438
RPC portmapper .....	438
RPM .....	120–129
- bezpečnost .....	176
- deltarpm .....	124
- instalace .....	43, 121
- LSB .....	120
- mazání .....	121
- odstranění .....	43
- opravy .....	122
- překlad .....	127

- PGP .....	121
- správa .....	43, 120
- verze 4 .....	111
- vytváření .....	111
- zdrojové .....	127
rsync .....	501, 513
runlevel .....	62, 151
- přechod .....	151, 159
- typy .....	152
- YaST .....	158
- změna .....	153

## S

síťování .....	367
síťové adresy	
- IPv4 .....	370
- IPv6 .....	373
- překlad jmen .....	381
síťové služby .....	55
síťový souborový systém .....	<i>viz</i> NFS
sítě .....	367, <i>viz</i> TCP/IP
- bezdrátové .....	264
- Bluetooth .....	264, 320
- DHCP .....	441
- DNS .....	381
- IP adresa .....	370
- IrDA .....	264
- konfigurační soubory .....	395–401
- konfigurace .....	382–402
- IPv6 .....	380
- localhost .....	373
- nastavení .....	51–55, 382
- oznamovací adresa .....	372
- reverzní převod .....	424
- síťové masky .....	371
- SLP .....	405
- směrování .....	370, 371
- WLAN .....	264
- YaST .....	382
- základní síťová adresa .....	372
Samba .....	519–529
- bezpečnost .....	523–524
- instalace .....	520
- jména .....	520
- klient .....	55, 520, 527–528
- konfigurace .....	520–524
- nápověda .....	529
- NetBIOS .....	520
- optimalizace .....	528
- přístupová práva .....	523

- přihlášení .....	524	skener .....	47
- práva .....	523	- řešení problémů .....	48
- sdílení .....	520, 522	skripty .....	
- server .....	55, 520–524	- boot.udev .....	343
- SMB .....	520	- init.d .....	156, 401
- spuštění .....	520	· network .....	401
- swat .....	524	· nfsserver .....	402
- TCP/IP .....	520	· portmap .....	402
- tisk .....	528	· sendmail .....	402
- tiskárny .....	520	· xinetd .....	402
- ukončení .....	520	· ypbind .....	402
SaX .....	214	· ypserv .....	402
- 3D .....	218	- irda .....	328
- multihead .....	219	- mkinitrd .....	149
- rozlišení monitoru .....	217	- modify_resolvconf .....	203, 396
SCPM .....	67, 277	- SuSEconfig .....	159–160
- nastavení .....	279	skupiny .....	
- notebooky .....	261	- správa .....	56
- přepínání profilů .....	280	SLP .....	262, 405
- spuštění .....	279	- Konqueror .....	406
- zdroje .....	279	- registrace služeb .....	405
scripty .....		- slptool .....	406
- init.d .....		směrování .....	370, 395
· nfsserver .....	439	- maškaráda .....	553
· portmap .....	439	- síťová maska .....	371
· squid .....	535	- statické .....	395
SCSI zařízení .....		SMB .....	<i>viz Samba</i>
- konfigurace .....	94	software .....	
- soubory, přiřazování .....	94	- instalace .....	43
security .....		- odstranění .....	43
- startování .....	569	- Správce programů .....	43
security level .....	523	souborové systémy .....	345–355
sendmail .....	52	- šifrování .....	564
server .....		- access control lists .....	582–592
- CUPS .....	248	- e2fsck .....	636
- LDAP .....	455	- Ext2 .....	347
- NFS .....	438	- Ext3 .....	347–348
- NIS .....	429	- FAT .....	17
- poštovní .....	52	- JFS .....	350–351
- proxy .....	531	- kontrola .....	631
- Samba .....	519	- limity .....	353
- souborový .....	54, 519	- NTFS .....	17, 19
- tiskový .....	247	- oprava .....	631, 636
- webový .....	477	- podporované .....	352–353
- X .....	213	- Reiser4 .....	349–350
servery .....		- ReiserFS v3 .....	349
- instalační .....	82	- reiserfsck .....	631
Service Location Protocol .....	<i>viz SLP</i>	- sysfs .....	332
SGML .....		- termíny .....	346
- openjade .....	110	- výběr .....	346

- XFS .....	351–352	- práva .....	535, 539
- xfs_check .....	641	- problémy .....	535
Souborový systém FAT .....	17	- proxy cache .....	531
soubory		- RAM .....	534
- .exe .....	623	- reporty .....	548
- šifrování .....	564	- spuštění .....	534
- jádra .....	201	- squidGuard .....	546
- logy .....	199	- statistiky .....	544, 546
- synchronizace		- stav objektů .....	533
· CVS .....	500, 507	- transparentní proxy .....	542, 544
· mailsync .....	501, 514	- ukládání .....	533
· rsync .....	501	- vlastnosti .....	531
· subversion .....	501	- zastavení .....	535
· Unison .....	500, 505	SSH .....	559–564
- větší než 2 GB .....	353	- autentizační mechanizmy .....	562
- velikost .....	353, 354	- démon .....	561
spindown .....	294	- páry klíčů .....	561, 562
správa		- scp .....	560
- profilů .....	277	- server .....	561
- skupin .....	56	- sftp .....	561
- uživatelů .....	56	- ssh .....	560
správa napájení .....	260, 285–303	- ssh-agent .....	563
- ACPI .....	285, 288–293	- ssh-keygen .....	562
- APM .....	285, 287	- sshd .....	561
- disk .....	294	- X .....	563
- frekvence CPU .....	295	ssh .....	628
- powersave .....	295	startování .....	147
- rychlost CPU .....	295	- CD .....	164
- YaST .....	304	- disketa .....	90, 164
správce		- DOS .....	165
- profilů .....	67	- grafické	
Správce logických svazků .....	<i>viz</i> LVM	· vypnutí .....	183
Squid .....	531	- graphic .....	183
- adresáře .....	535	- GRUB .....	166–185
- Apache .....	545	- initrd	
- bezpečnost .....	532	· vytváření .....	149
- cache .....	532	- konfigurace .....	22
· poškozená .....	535	- rawrite .....	91
· vícenásobná .....	532	- souborový systém .....	631, 636
· velikost .....	533	- správa .....	164
- cachemgr.cgi .....	544, 546	- USB .....	164
- Calamaris .....	548	- Windows .....	165
- CPU .....	534	- zaváděcí sektory .....	164
- firewall .....	543	- zaváděče .....	165
- konfigurace .....	537	startovací disketa .....	58, 164
- kontrola přístupu .....	539, 545	Subversion .....	501, 510
- logy .....	535, 538, 544	SUSE LINUX	
- odinstalování .....	535	- instalace .....	85
- operační paměť .....	534	swap .....	69
- pevný disk .....	533	synchronizace	

- pošta .....	501
- soubory .....	499–517
· CVS .....	500, 507–509
· mailsync .....	501, 514–517
· rsync .....	501
· subversion .....	501
· Unison .....	500, 505–507
synchronizace času .....	449
- konfigurace .....	450
- xntp .....	449
synchronizace dat .....	264
- e-mail .....	263
- Evolution .....	266
- Kontakt .....	266
- KPilot .....	266
sysconfig .....	62
systém .....	57
- aktualizace .....	107–111, 129
- konfigurace .....	35–75
- lokalizace .....	208
- využívání omezených zdrojů .....	201
- X Window .....	<i>viz X</i>
- záchrana .....	135
systémová právy .....	626
systémové soubory .....	
- oprava .....	137
systémy písem .....	229
- písma s kódováním CID .....	233
- písma X11 core .....	232
- Xft .....	229
<b>T</b> .....	
TCP/IP .....	368
- ICMP .....	368
- IGMP .....	368
- přenosový model .....	369
- pakety .....	369
- TCP .....	368
- UDP .....	368
telefonní ústředna .....	388
telnet .....	628
tisk .....	237, 241–243
- řešení problémů .....	
· síť .....	254
- CUPS .....	247
- footmatic filtry .....	111
- fronty .....	242
- GDI tiskárny .....	252
- Ghostscript ovladač .....	242
- konfigurace pomocí YaST .....	241

- kprinter .....	247
- LPRng .....	111
- ovladače .....	242
- příkazová řádka .....	246
- připojení .....	242
- port .....	242
- PPD soubor .....	242
- síť .....	
· řešení problémů .....	254
- Samba .....	520
- testovací stránka .....	242
- xpp .....	247
- z aplikace .....	246

TrueType ..... *viz X, TrueType fonty*  
TV karty ..... 50

## U

uživatelé .....	
- /etc/passwd .....	360, 470
- správa .....	56
udev .....	339
- automatizace .....	341
- klíče .....	342
- mass storage .....	343
- pravidla .....	340
- regulární výrazy .....	341
- startovací skript .....	343
- sysfs .....	342
- udevinfo .....	342
- YaST .....	343
- zástupné znaky .....	341
ulimit .....	201
- nastavení .....	201
update .....	<i>viz aktualizace, viz update</i>
USB .....	
- disky .....	265
- flash disky .....	265
UTF-8 .....	113
uzly zařízení .....	
- udev .....	339

## V

virtuální paměť .....	65
VNC .....	
- instalace .....	87
vstupní metody .....	
- CJK .....	208

## W

webový server

- Apache .....	<i>viz</i> Apache
whois .....	382
Windows	
- sdílení souborů .....	519
WLAN .....	264

## X

X .....	213
- 3D .....	218
- bezpečnost .....	572
- fonty .....	228
- fonty TrueType .....	228
- multithread .....	219
- nápověda .....	228
- nastavení .....	214
- optimalizace .....	223–228
- ovladače .....	227
- písma s kódováním CID .....	233
- písma X11 core .....	232
- SaX2 .....	223
- SSH .....	563
- systémy písem .....	229
- virtuální obrazovka .....	226
- xfb6config .....	223
- Xft .....	229
- xft .....	228
- znakové sady .....	228
X rozšíření klávesnice .....	<i>viz</i> klávesnice, X rozšíření klávesnice
X11 .....	<i>viz</i> X
xfs_check .....	641
Xft .....	229
xinetd .....	55, 110
XKB .....	<i>viz</i> klávesnice, X rozšíření klávesnice
XML	
- Katalog .....	111
- openjade .....	110
xorg.conf	
- barevná hloubka .....	226
- Cesty k fontům .....	224
- Depth .....	226
- Display .....	226
- Modeline .....	226
- Modes .....	226
- Monitor .....	224, 226
- parametry zobrazení .....	224
- sekce Device .....	226
- sekce InputDevice .....	224
- Sekce Modes .....	224
- sekce ServerFlags .....	224

## Y

### YaST

- úroveň běhu .....	62
- Řídicí středisko .....	37
- časová zóna .....	60
- 3D .....	234
- aktualizace .....	27, 109
- online .....	38, 78
- aktualizace systému .....	42
- aktualizace z CD .....	42
- backup .....	57
- bezpečnost .....	56–57
- Bluetooth .....	319
- CD mechanika .....	45
- dělení disku .....	63
- DHCP .....	442
- diskový prostor .....	12
- DMA .....	46
- DNS server .....	55
- dotaz na podporu .....	74
- DSL .....	390
- Editor úrovní běhu .....	158
- grafická karta .....	214, 216
- grafické uživatelské rozhraní ..	214–222
- hardware .....	45
- informace o hardwaru .....	45
- instalační server .....	82
- instalace .....	3–33
- Internet .....	51
- ISDN .....	387
- joystick .....	46, 222
- kabelový modem .....	389
- konfigurace .....	35–75
- konfigurace linuxu .....	62
- konfigurace pevného disku .....	68
- konfigurace sítě .....	26, 51–55
- konfigurace zavaděče .....	72, 176
- LDAP klient .....	468
- LVM .....	95
- modem .....	385
- monitor .....	214
- myš .....	10, 47
- návrh instalace .....	9
- ncurses .....	75
- NFS klient .....	54
- NFS server .....	54
- NIS klient .....	29, 432
- NTP	
- klient .....	55
- obnova .....	57

- oprava systému .....	131
- ovladače na CD .....	75
- patch CD .....	42
- RAID .....	100
- režim spouštění .....	22
- restore .....	57
- root heslo .....	25
- routing .....	55
- rozdělování disky .....	11
- rozložení kláves .....	60
- rozložení klávesnice .....	10
- runlevel .....	62
- síťová karta .....	382
- Samba .....	
· klient .....	55, 527
· server .....	55
- SCPM .....	67
- skener .....	47
- software .....	37
- spouštění .....	4
- správa napájení .....	304
- správa skupin .....	56
- správa uživatelů .....	56
- správce profilů .....	67
- spuštění .....	36
- start systému .....	4
- startovací disketa .....	58
- sysconfig .....	62
- sysconfig editor .....	161
- systém .....	57
- systémový log .....	75
- systémový protokol .....	75
- T-DSL .....	391
- test paměti .....	7
- textový mód .....	
· odstraňování problémů .....	89
- textový režim .....	75–80, 88
· moduly .....	78
- tisk .....	241–243
- TV karty .....	50
- typ instalace .....	8, 21
- update .....	42, 109
- update softwaru .....	38
- výběr jazyka .....	8, 36, 60
- YOU .....	38

- záchranný systém .....	7
- záloha .....	57
- závislosti balíků .....	22
- zavaděč .....	72
- zdroj .....	37
- zvuk .....	49
YOU .....	<i>viz aktualizace online</i>
YP .....	<i>viz NIS</i>

## Z

záchraný systém .....	7, 135
- používání .....	136
- spouštění .....	135
záloha .....	
- obnova v YaST .....	57
- vytváření v YaST .....	57
zálohování .....	
- aktualizace .....	108
záznamy .....	<i>viz logy</i>
- Unison .....	507
- zprávy .....	558
zóna .....	
- časová .....	60
zóny .....	420
zatuhávání .....	627
zavádění systému .....	
- BIOS .....	5
- konfigurace .....	
· YaST .....	176–179
- MBR .....	164
- z CD .....	4
- z CD2 .....	93
- z diskety .....	90, 92
- zavaděč .....	178
· umístění .....	178
zavaděče .....	72, 163
- GRUB .....	165
- LILO .....	165
zdroj instalace .....	37
zdrojový kód .....	
- překlad .....	127
zvuk .....	49
zvuková karta .....	49
zvukové směšovače .....	118