



## Abíčko

Časopis serveru AbcLinuxu.cz

Srpen 2004

**unixshop**®

---

Vychází také na CD-ROM jako příloha časopisu

**PC WORLD**

# Editoriál

Vítejte u čtení časopisu Abíčko.

Abíčko vychází jako měsíční příloha serveru <http://www.abclinuxu.cz> a obsahuje výběr toho nejzajímavějšího obsahu, který zde byl v minulém měsíci publikován. Touto formou chceme předat čtenářům informace v snadno čitelné podobě vhodné i pro tisk.

Cílem serveru [AbcLinuxu.cz](http://www.abclinuxu.cz) je pomáhat všem uživatelům Linuxu, nezávisle na jejich zkušenostech, platformě či použité distribuci. Motorem, který nás pohání vpřed, je idea vzájemné pomoci a spolupráce. Proto i velkou část obsahu tvoří samotní uživatelé. Zapojit se může kdokoli, tedy i vy.

Na [AbcLinuxu.cz](http://www.abclinuxu.cz) najdete rozsáhlou databázi návodů na zprovoznění hardwaru pod Linuxem, velice aktivní diskusní fórum, podrobné návody a tutoriály, recenze, archiv ovladačů, informace o linuxovém jádře (včetně populárních Jaderných novin) i rozcestník po ostatních linuxových serverech.

Náměty na články zasílejte do konference našich autorů: [autori@abclinuxu.cz](mailto:autori@abclinuxu.cz).

Sponzoring Abíčka a jiné formy reklamy si objednávejte na adrese: [reklama@abclinuxu.cz](mailto:reklama@abclinuxu.cz).

Ostatní dotazy směřujte na adresu: [literakl@abclinuxu.cz](mailto:literakl@abclinuxu.cz).

Server [AbcLinuxu.cz](http://www.abclinuxu.cz) provozuje firma Stickfish s.r.o., která poskytuje profesionální služby v oblasti Linuxu firmám i jednotlivcům. Zabývá se hlavně bezpečností, instalacemi Linuxu a konfigurací síťových služeb. Více na <http://www.stickfish.cz>.

©2004 Stickfish s. r. o. a autoři článků

Editor a sazba: Vlastimil Ott

Pro nekomerční účely smíte tento dokument jakkoliv šířit v tištěné i digitální podobě. V ostatních případech nás požádejte o svolení na adrese [info@abclinuxu.cz](mailto:info@abclinuxu.cz).

## Typografické konvence

Ve výpisech zdrojových textů mohou být použity znaky `\\`. Značí přechod na nový řádek, který ovšem *není* součástí samotného zdrojového textu, byl přidán editorem z důvodu lepšího vzhledu případně nemožnosti text formátovat bez jejich použití.

# Obsah

<b>Editoriál</b>	<b>1</b>
<b>Obsah</b>	<b>2</b>
<b>Jak jsem se poprala s ABC Linuxem</b>	<b>4</b>
<b>Signály</b>	<b>6</b>
Poslanie signálu alebo Ako vraždiť bez obáv z polície .....	6
Handler alebo Čo číha na procesy a ich sebaobrana .....	6
<b>MDB Tools: MS Access trochu jinak</b>	<b>9</b>
MDB Tools se představuje .....	9
Instalace přede dveřmi .....	9
Otvíráme bránu .....	10
Utility příkazového řádku .....	11
ODBC ovladač .....	12
Gmdb2 .....	12
Export Schema .....	14
MDB Query Tool .....	15
K čemu je tedy klíč dobrý ? .....	15
Případová studie .....	15
Klíč, který odemyká data .....	16
<b>Zpomalte si procesor</b>	<b>17</b>
Proč zpomalovat? .....	17
Spotřeba .....	17
Nastavení frekvence procesoru .....	18
Závěr .....	18
<b>Inštalácia Slackware</b>	<b>20</b>
Úvod .....	20
Motivácia .....	20
Čo k tomu potrebujeme? .....	21
Ingredience .....	21
Inštalácia .....	21
Bootovanie .....	22
Rozdeľovanie disku .....	23
Inštalácia systému .....	26
Príprava .....	26
Inštalácia .....	29
Poinštaláčnã konfigurácia .....	30
Ja chcem nový software! .....	37
Nastavenie siete .....	38
<b>Jaderné noviny 264–267</b>	<b>40</b>
Aktualizace USB pro 2.6, 11 e-mailů .....	40
Pravidla pro psaní IDE ovladačů, 10 e-mailů .....	40
Nová kniha o virtuální paměti v Linuxu, 3 e-maily .....	41
UMSDOS bude možná vypuštěn z 2.6, 4 e-maily .....	41
SquashFS verze 2.0, 1 e-mail .....	42

Linus navrhuje nová pravidla označování původu záplat .....	42
Ovladač VIA Velocity Gigabit Ethernet, 3 e-mailů .....	43
Jak funguje geometrie disků v 2.4 a 2.6, 33 e-mailů .....	43
Správcovství SysFS, 2 e-mailů .....	45
Bezpečnostní 'NX' funkce v 2.6, 66 e-mailů .....	45
Zdokonalení obsluhy tlačítek myši v mousedev, 6 e-mailů .....	46
Podpora binárek SCO v Linuxu, 5 e-mailů .....	46
Nový nástroj pro kopírování paměti, kcopyd, 10 e-mailů .....	47
Stav JFFS2, 14 e-mailů .....	47
Konfigurační nápověda pro NAPI, 5 e-mailů .....	47
Další pokus o kompilační parametr pro Debian, 9 e-mailů .....	48
Aktualizace NTFS obsahuje přepisování rezidentních souborů .....	48
Časovač s vysokou přesností (HRT), 33 e-mailů .....	48
Podpora RSA Crypto API v řadě 2.6, 12 e-mailů .....	49
Linux 2.6.7, 40 e-mailů .....	49
Kdy používat GCC Built-Ins, 17 e-mailů .....	49
<b>Zprávičky</b> .....	<b>51</b>

# Jak jsem se poprala s ABC Linuxem

Andrea Králová

---

*Za poslední tři roky jsem prostřídala na svém počítači různé verze Windows – od 95 přes 98 až k XP. Že existují i jiné operační systémy, jako třeba Linux, jsem donedávna opravdu netušila.*

---

Pak jsem si díky známému, který mi občas poslal odkazy na nějaké články o Linuxu, sem tam něco přečetla, hlavně o těch „live“, ale zas tak moc mě to nezajímalo. Jednak jsem nevěřila, že by to takhle jednoduše fungovalo rovnou z CD, a hlavně jsem se hrozně bála, že mi to něco provede s počítačem. Nakonec jsem jednou náhodou uviděla v obchodě ABC Linux, vzpomněla si, o co jde a po chvilce váhání koupila – 119 Kč není tak moc. Nejdřív jsem si raději četla jen příručku, vypadalo to lákavě, takže jsem se odhodlala. No a pak nastalo něco spíš jako zlý sen (s poměrně šťastným koncem :), který vám teď popíšu. Co všechno tedy může zažít dosud Linuxem nepoznamenaná majitelka takové live distribuce?

Úvodní obrazovka se třemi tučňáky naběhla přesně, jako bylo napsáno v příručce a po mém váhavém Enter se po obrazovce rozběhl vesele barevný text, jak se detekovalo nastavení PC. Ale ouha: někde za myší („Mouse is generic PS/2 wheel Mouse At /dev/psaux“) se to celé zastavilo a dál nic. Co teď? Kombinovat různé parametry bootování, radí příručka. Takže jsem poctivě postupně zkoušela „Failsafe, knoppix 2, noagp ...“, ale výsledek žádný. Nakonec už ze zoufalství: „knoppix testcd“, což mě odměnilo opravdu potěšujícím sdělením: „Data on your CD medium is possibly incomplete or damaged, or your computer has bad RAM“. Následný „memtest 20M -1“, který by měli podle příručky provádět jen odborníci, což tedy nejsem, neukázal vůbec nic. Ukončila jsem ho po třech hodinách. Chyba nebyla asi ani ve mně, ani v Linuxu, ale v CD samotném: nefungovalo totiž ani na jiném počítači a nešlo ani zkopírovat.

Nevzdala jsem se, zakoupila nové CD a akci opakovala. Začalo to stejně špatně. Až když k řadě mnou stále zoufale opisovaných a kombinovaných parametrů přibylo „xvrefresh=50-120“, nechal se monitor (HP D2837) přemluvit a tentokrát nezhasl. A tak jsem mohla sledovat, že „Starting X11 ... Retrying with server Xfree86 (vesa) ... Retrying with server Xfree86 (fbdev). Error No suitable x-server found for your card ...“ (ATI 3D Rage Pro AGP 2x, s níž asi nikdo jiný, jak to vypadá, problémy nemá). Nakonec zafungovala rada z diskusního fóra vyzkoušet „xserver=XF86\_Mach64“, který jsem až později opravdu našla i mezi spoustou parametrů v příručce. ABC Linux se tedy nakonec pomocí „knoppix noagp xserver=XF86\_Mach64 xmodule=i810 nodd screen=800x600 svrefresh=50-120“ rozběhl do KDE, sláva. Grafika je mizerná a zvuk nejede vůbec, ale i tak jsem byla ráda, že se to nakonec podařilo. Tohle všechno se dělo na repasovaném Compaq DeskPro Pentium III, 700 Mhz, 128 RAM.

Následovalo připojení k internetu, což také nebylo tak jednoduché zjistit, protože se připojuji přes Eurotel prostřednictvím mobilního GSM modemu MC35i. Pro Windows existuje instalační CD, nastavení pro linux jsem hledala na stránkách Eurotelu marně a modem mi nefungoval. Až na diskusním fóru jsem zase po delším hledání našla, že do příkazů modemu do druhé řádky musím zadat „AT+cgdcont=1,„IP”, „internet”“ – že mě to hned nenapadlo.

Protože celý pevný disk je formátován v NTFS, zkoušela jsem si uložit nastavení na disketu. Z jakéhosi záhadného důvodu ale při startu zadaný parametr „knoppix myconfig=/dev/floppy“ ani „knoppix myconfig=scan“ (zadáno podle příručky i podle pokynu při zálohování) nenačetl vůbec nic. Až na fórech Knoppixu jsem se dozvěděla, že myconfig při načítání z diskety nefunguje a nikdy snad nefungoval a že „knoppix floppyconfig“ je ten správný parametr. Nešlo by to pro začátečníka ještě více zmateně? Vypadá to, že tohle při výrobě distribuce snad nikdo netestoval.

Musím uznat, že jsem si to představovala jednodušeji, ale každopádně je to docela zajímavý zážitek. Trošku mě mrazí při představě, že bych tohle všechno musela prodělat úplně sama bez přítele na internetu, který mi přece jen poradil, když jsem nevěděla jak dál. Ale jinak se mi Linux líbí, i když si zatím hraji jen s nastavením vzhledu v KDE. Příjemně mě překvapilo, že si nemusím instalovat žádné další programy, viděla jsem, že je tam skoro všechno. Trochu mě mrzí, že je něco v angličtině, ale anglicky se už učím. Dostala jsem se na Internet, spojím se přes ICQ a dokážu napsat dokument v OpenOffice.org (ten jsem si stáhla i pro Windows), víc zatím asi nepotřebuji. Už jsem se i rozhodla, že nějaký Linux půjde i na pevný disk, ale ještě nevím který. Zatím jsem si koupila Knoppix, Danix, Slax, Slackware 10.0 a Gentoo, takže myslím, že o mně ještě uslyšíte.

\*\*\*

# Signály

Rastislav Stanik

---

*Tento článok hovorí o signáloch. Hoci budeme spomínať volania jadra a knižničné funkcie, to čo sa tu dočítate sa bude hodiť aj neprogramátorom.*

---

Jadro operačného systému spravuje procesy. Rozhoduje o ich vzniku a zániku, pozastavení, pokračovaní a striedaní. Na túto prácu potrebuje udržiavať o procesoch rôzne druhy informácií. Tou, ktoré nás teraz bude zaujímať, je informácia o tom, kde sa naposledy prerušilo vykonávanie procesu (a teda, kde by sa nabudúce malo pokračovať), potom pole príznakov s hodnotou áno/nie a tabuľka `sighandlers` udržiajúca ukazovatele na funkcie. Každý príznak v poli príznakov má zodpovedajúci ukazovateľ v `sighandlers`. Keď jadro rozhodne o tom, že je vhodné, aby nejaký proces dostal na chvíľu slovo, najprv sa pozrie, či je niektorý z príznakov nastavený na „áno“. Ak je, tak namiesto toho, aby riadenie odovzdalo na miesto, kde bol program naposledy prerušený, zavolá funkciu, na ktorú ukazuje zodpovedajúci ukazovateľ zo `sighandlers`. Ak proces nebol prerušený, ale práve bežal, tak jadro preruší jeho vykonávanie a takisto riadenie odovzdá do funkcie odkazovanej zo `sighandlers`.

## Poslanie signálu alebo Ako vraždiť bez obáv z polície

Signál môže proces dostať z troch dôvodov:

1. Nieкто spustí príkaz `kill` a ako parameter mu odovzdá číslo procesu:

```
$ kill 1234
```

Príkaz `kill` môže alternatívne dostať cez parameter aj typ signálu, ktorý má danému procesu poslať. Typ signálu možno určiť buď číslom:

```
$ kill -2 1234
```

alebo symbolickým menom:

```
$ kill -INT 1234
```

2. Druhá varianta poslania signálu je vlastne to isté, ale s použitím volania knižničnej funkcie `kill()`:

```
int ok;
ok=kill(1234, 2);
```

3. Posledná možnosť je, že program urobí niečo ilegálne a signál mu pošle jadro. To sa môže stať napríklad pri delení nulou, či prístupe do pamäti cez nesprávne nastavený ukazovateľ a podobne.

## Handler alebo Čo číha na procesy a ich sebaobrana

Všetky položky v poli príznakov sú inicálne nastavené na `nie`. Na druhej strane všetky položky v tabuľke `sighandlers` nejaké nastavenie majú. Za normálnych okolností na väčšinu signálov proces zvyčajne zareaguje tým, že skončí. Toto správanie ale môže program ovplyvniť tým, že zavolá funkciu `signal(2)` (funkcia `signal()` je definovaná v norme ANSI, novšie systémy odporúčajú použiť podobnú funkciu `sigaction(2)`):

```
signal(SIGFPE,newhandler);
sigaction(SIGFPE,struct_newhandler,struct_oldhandler);
```

kde `newhandler()` musí byť deklarovaná takto:

```
void newhandler(int arg)
```

`struct_newhandler` a `struct_oldhandler` sú štruktúry, ktoré okrem iného obsahujú ukazovateľ na funkciu deklarovanú ako `newhandler()`.

Ako druhý parameter funkcii `signal()` možno (okrem ukazovateľa na novú obslužnú funkciu) poslať aj preddefinované hodnoty `SIG_IGN`, ktorá znamená ignorovanie signálu a `SIG_DFL`, ktorá znamená nastavenie signálu na výchozú obslužnú funkciu. Zavolaním funkcie `signal()` môže proces preddefinovať, ako zareaguje na nejaký signál. Po prijatí signálu sa ďalšie signály tohoto typu buď obsluhujú výchozou obslužnou funkciou (to platí pre staršie systémy) alebo sú blokové. Samozrejme obslužná funkcia môže obsluhu signálu opäť prestať. V prípade, že procesu je poslaných viacero signálov prv než sa vobec dostane k slovu, tak sa tieto signály radia do fronty a budú spracované v takom poradí, v akom boli poslané. Výchozí akcie nájdete v manuálovej stránke `signal(7)` a v tabuľke.

Spomínané signály definuje norma POSIX. Linux (a aj iné OS) pozná ešte ďalšie – viz `signal(7)`. Ako som spomínal, niektoré signály možno poslať z terminálu. To, akou klávesou alebo kombináciou kláves sa ten ktorý signál pošle, možno nadefinovať programom `stty`. Ten umožňuje aj zistiť aktuálne nastavenie:

```
$ stty -a
...
intr = ^C; quit = ^\; erase = ^?; kill = ^U;
...; start = ^Q; stop = ^S; susp = ^Z;
...
$ stty intr ^k
$ stty -a
...
intr = ^K;...
```

Signály `SIGQUIT`, `SIGILL`, `SIGABRT`, `SIGFPE` a `SIGSEGV` spôsobia vytvorenie súboru `core`. Je to obraz pamäte procesu v okamihu, keď dostal signál. Vytváranie `core` súboru môže ovplyvniť vstavaný príkaz shellu – `ulimits`.

Posielanie signálov prirodzene podlieha prístupovým právam. Užívateľ, ktorý nie je `root`, môže poslať signál len procesom, ktoré sám naštartoval alebo ich vlastní. V manuálovej stránke funkcie `kill(2)` sa tiež dočítate o možnostiach posielania signálov skupine procesov naraz (ak `pid` je 0, -1, alebo menšie ako 0)

Hoci signál `SIGKILL` nemožno blokovať, môže sa stať, že proces nereaguje ani na najmocnejší zo signálov. Je to vtedy, ak jadro nerozhodlo o pridelení procesorového času tomuto procesu, alebo proces čaká na dokončenie vstupno-výstupnej operácie.

Operačný systém ponecháva proces v tabuľke procesov dovtedy, kým ich rodičovský proces nespracuje `SIGCLD` poslaný po zániku procesu-potomka. Počkať na tento signál môže program pomocou systémového volania `wait(2)` alebo `waitpid(2)`. Ak to rodičovský proces neurobí, jeho potomok vidno vo výpise procesov ako „zombie“. Zvláštne; keď sa chcete zbaviť zombie, musíte si na ňu počkať ;-).



Meno signálu	Výchozí obsluha	Komentár
SIGHUP	Zánik riadiaceho (rodičovského) procesu	Démony ho často používajú ako signál, ktorým môže správca požiadať o znovunačítanie konfigurácie
SIGINT	Prerušenie z klávesnice	Tento signál dostane proces, ak na jeho riadiacom termináli stlačíte Ctrl-C. (Kombináciu kláves umožňuje predefinovať program <code>stty</code> )
SIGQUIT	Signál Quit z klávesnice	Podobne ako predchádzajúci signál ho možno poslať z klávesnice. Zvyčajne stlačením Ctrl-(Opäť predefinovateľné pomocou <code>stty</code> )
SIGILL	Ilegálna inštrukcia	Program vykonal ilegálnu inštrukciu
SIGABRT	Abort signál od funkcie <code>abort(3)</code>	Funkciu <code>abort()</code> môže program zavolať, ak chce okamžite ukončiť svoju činnosť – napríklad v dôsledku chyby – bez toho, aby sa vrátil zo všetkých funkcií
SIGFPE	Výnimka pri operácii s rádovou čiarkou	Napríklad delenie nulou
SIGKILL	Zabíjajúci signál	Bezpodmienečné zabitie procesu. Obsluhu tohoto signálu nemožno predefinovať/ignorovať pomocou <code>signal()</code>
SIGSEGV	Porušenie segmentácie pamäte	Neinicializovaný pointer, či pretečenie poľa a podobne.
SIGPIPE	Zápis do rúry, z ktorej nikto nečíta	Ak dva procesy spolu komunikujú cez rúru <code>pipe()</code> a ten čo číta, zanikne, ten druhý sa o tom môže dozvedieť odchytením signálu SIGPIPE
SIGALRM	Signál poslaný funkciou <code>alarm(2)</code>	Funkcia <code>alarm()</code> , požiada aby jadro poslalo procesu signál SIGALRM za <code>N</code> sekúnd
SIGTERM	Ukončujúci signál od terminálu	Poslaný pri zániku riadiaceho terminálu
SIGUSR1, SIGUSR2	Užívateľské signály	Tieto signály môže programátor predefinovať na svoje vlastné a špecifické akcie – napríklad zvýšenie a zníženie úrovne logovania
SIGSTOP	Pozastavenie procesu	Tento signál možno zvyčajne poslať pomocou Ctrl-Z (predefinovateľné pomocou <code>stty</code> ) – takýto proces môžete poslať do pozadia, alebo nechať pokračovať pomocou vstavaných príkazov shellu – <code>bg</code> a <code>fg</code> . Nemožno ho odchytiť/ignorovať pomocou <code>signal()</code>
SIGCONT	Pokračovanie pozastaveného procesu	Povolí pokračovanie procesu pozastaveného pomocou SIGSTOP

\*\*\*

# MDB Tools: MS Access trochu jinak

Stanislav Puffler

---

*Byl bych se vůbec nezlobil, kdyby byl Bill býval sám zdokumentoval a otevřel MS Access širšímu okruhu uživatelů a vývojářů, zejména pak nám tučňákům linuxovitým. Opět to ale musel udělat někdo jiný...*

---

## MDB Tools se představuje

Projekt MDB Tools se snaží zdokumentovat souborový formát MDB používaný v databázích [Access](#) [1] z dílny společnosti [Microsoft](#) [2] a snaží se také poskytnout sadu nástrojů a aplikací, které zpřístupní data uložená v tomto proprietárním formátu i uživatelům non-windows platform. Obecně umožňují tyto nástroje exportovat schéma databáze a data do DDL a souborů importovatelných do MS SQL Server, MS Access, MySQL, PostgreSQL, Oracle, Sybase a dalších. Balíček obsahuje i engine vykonávající jednoduché dotazovací příkazy SQL. MDB Tools umí data ze souboru .mdb pouze číst a exportovat, a to ještě s omezením (Jet 3 a 4), nicméně v další verzích je avizována i možnost zápisu do databáze. Poslední verze 0.5 obsahuje grafické uživatelské rozhraní a ODBC ovladač. Motto projektu zní „Odemkni svá data“, vezměme tedy onen pomyslný klíč a pojďme se podívat, kde všude si odemkneme.

## Instalace přede dveřmi

Než začnete vůbec instalovat MDB Tools, je nutno splňovat určité požadavky, které jsou specifikované na stránkách projektu i s odkazy na související stránky. Potřebné jsou tyto technické předpoklady: glib2.0, readline bison, flex, unixODBC, Gnome 2.0, libglade a wordexp.

MDB Tools je možné instalovat jako balíček RPM, DEB, jiný binární balíček nebo ze zdrojů. U RPM a DEB balíčku je instalace triviální. Zastavme se tedy až u instalace ze zdrojových kódů. Nemine nás trojice – konfigurace, sestavení a instalace. Konfiguraci je možné udělat dvěma způsoby. Konfigurace přímo z tarballu vypadá následovně:

```
$ zcat mdbtools-0.5.tgz | tar xvf -
$ cd mdbtools-0.5
$ ./configure
```

Chcete-li instalovat MDB Tools s podporou ODBC, je nutné použít argument `--with-unixodbc` s uvedením adresáře, ve kterém máte [unixODBC](#) [3] nainstalován.

```
$ zcat mdbtools-0.5.tgz | tar xvf -
$ cd mdbtools-0.5
$ ./configure --with-unixodbc=/usr/local
```

Hledáte-li tu nejaktuálnější verzi MDB Tools, použijte CVS. Tento krok využívá skript `autogen.sh`, tudíž je konfigurace trochu odlišná. Opět zde platí – nechcete-li podporu ODBC, odstraňte argument `--with-unixodbc`.

```
$ cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/mdbtools login
$ cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/mdbtools co
```

```
$ cd mdbtools  
$ ./autogen.sh --with-unixodbc=/usr/local
```

Sestavení a instalace jsou již standardní příkazy:

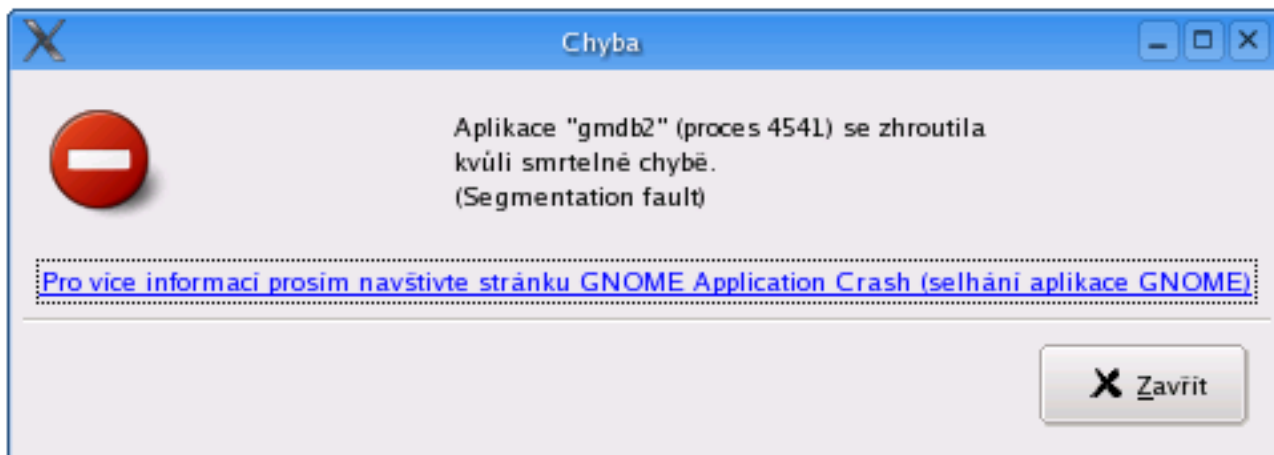
```
$ make  
$ su -  
# make install
```

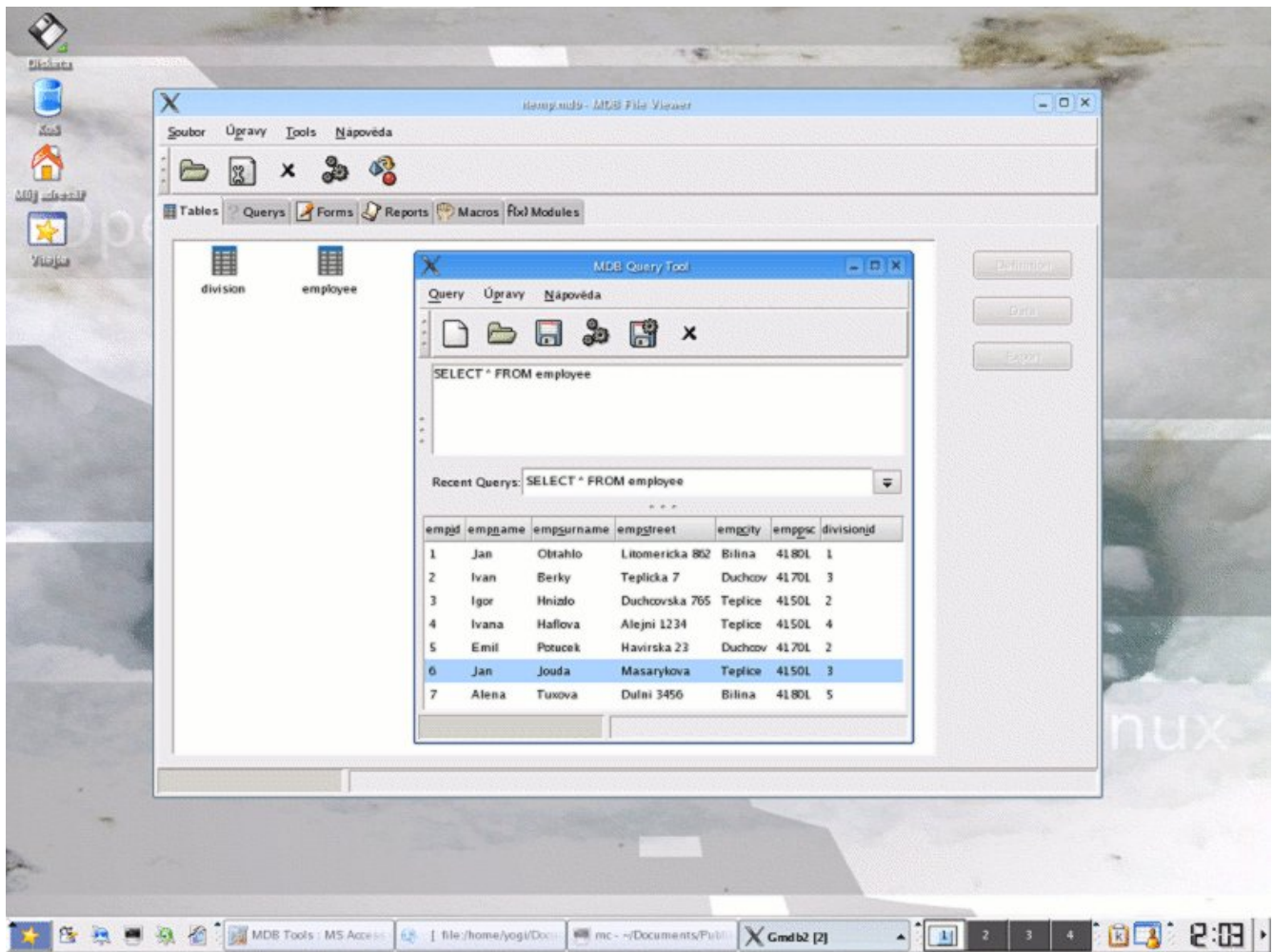
## Otvíráme bránu

Databáze MDB jsou soubory vytvořené pomocí MS Jet Database Engine. Tyto soubory jsou používány nejen v databázích MS Access, ale také například v produktech MS, jakými jsou webový server IIS, poštovní server Exchange, Money či Project. Setkal jsem se s nimi ale i u produktů třetích stran – příkladem budiž databáze sekundárních informací vztahujících se k audio nahrávkám telefonních hovorů v zařízení Racal Mirra. Tyto .mdb soubory umožňují klientské aplikaci indexovat hovory a přiřadit nahrávce .wav data typu počátek a konec hovoru, trvání, tel. linka a podobně.

MDB Tools je především knihovna pro přístup k datům z databázových souborů .mdb – knihovna má název LibMDB. Dále pak utility příkazového řádku pro získání schéma a dat ze souboru, tyto utility se nám hodí ke skriptování. Dále pak LibMDBSQL - vrstva nad LibMDB umožňující provádět nad daty v souboru dotazy SQL. LibMDBODBC je databázový ovladač, který využijeme ve spolupráci s programovacími jazyky jako Perl a PHP pomocí DBD:ODBC. K nativnímu ovladači je však nutno doinstalovat [unixODBC](#) [4], jelikož potřebuje ke své práci unixový ODBC driver manager. Konečně Gmdb2, což je GUI nad LibMDB a command line utilities, zobrazující objekty uložené v databázovém souboru a umožňující tak v grafickém rozhraní prohlížet schéma tabulek nebo dotazování se pomocí SQL – prostředí je docela zdařilé a přehledné.

Do budoucna se dle roadmapy chystají samá pěkná vylepšení a nové funkce jako například práce s indexy, možnost přidat nový záznam do tabulek, plná podpora zápisu pro LibMDB a LibMDBSQL, možnost extrahovat dotazy, VBA skripty, formuláře či vlastnosti tabulek, kontrola konzistence databáze, recovery tool, možnost přidávat vlastní tabulky do souboru a další. Z mého pohledu a díky testování těchto nástrojů bych však nejprve doporučoval pečlivě sledovat a řešit primárně bug reporty od uživatelů, jelikož chyb má zejména GUI stále dost a dost – příklad je na obrázku. Dalším nešvarem, jak už to tak bývá, je špatné zobrazování a vůbec práce s českými znaky. Myslím, že MDB Tools mají ambice být dobrou alternativou MS Accessu v prostředí Unixu a Linuxu – proto by jim slušela i lokalizace do vícero světových jazyků ;o).





## Utility příkazového řádku

Silným nástrojem unixu byly, jsou a budou programy pro příkazový řádek. MDB Tools nám nabízí několik utilit pro práci s databázovým souborem .mdb v linuxovém shellu. Většina dále jmenovaných programů má po instalaci své vlastní manuálové stránky s podrobným popisem a syntaxí. Prvním z nich je `mdb-tables`, který nedělá nic jiného, než vypíše tabulky z dané databáze.

Dalším příkazem je `mdb-schema`, který dokáže produkovat DDL výstup z MDB. Jazyk DDL (Data Definition Language) je standard a tudíž je možné pomocí schéma Access databáze následně vytvořit stejnou databázi v alternativním databázovém prostředí. Podporovány jsou aktuálně MS SQL Server, MS Access, Sybase, Oracle, PostgreSQL a MySQL. Exportovat je možné buď jednu tabulku nebo obsah databázového souboru MDB. Příkaz posílá výstup na obrazovku – doporučuji přeměrovat výstup pomocí roury do textového souboru s DDL.

```
mdb-schema soubor.mdb oracle > ddl_soubor.sql
```

`mdb-export` umí generovat CSV výstup pro určenou tabulku. Pomocí parametrů je možné nezobrazit v CSV hlavičku, určit si vlastní oddělovač, nezabalovat pole typu text, memo a date do uvozovek. Takový výstup je pak možné snadno importovat do jiných databázových prostředí a tabulkových procesorů typu OpenOffice.org Calc.

`mdb-ver` určí verzi databázového stroje, ve kterém byla databáze vytvořena. Výstup je buďto JET3 (Access 97) nebo JET4 (Access 2000/XP).

Programátoři a vývojáři databázových aplikací uvítají utility, které jim vygenerují hlavičkové soubory pro práci s databází v jazyce C. Jedná se o programy `mdb-header` a `mdb-parsecsv`.

Posledním, avšak velmi silným nástrojem `command utilities`, je samotné SQL rozhraní pro shell. Příkazem `mdb-sql` si tak můžete vaše data osahat tak, jak jste zvyklí. Díky několika parametrům je možné poslat na vstup `mdb-sql` příkazy ze souboru nebo naopak místo na STDOUT poslat výstup samotného `mdb-sql` do souboru pro pozdější použití. Opět je možné hrát si s hlavičkou, a tentokrát i s patičkou, vypínat a zapínat `pretty printing` dle vašich potřeb nebo použít vlastní oddělovače.

Nutno podotknout, že SQL je v Access značně osobité a znalci SQL Serveru nebo Oracle a jeho PL/SQL si asi na své zrovna nepřijdou. Navíc `mdb-sql` obsahuje jen malou podmnožinu příkazů jazyka SQL, kterou blíže definují manuálové stránky. Po zadání základního příkazu s jeho možnými parametry se dostanete do interaktivního režimu (tak jak je tomu například v utilitě `isql`) a můžeme začít s laborováním.

Příkazem `connect to` s názvem databáze se připojíte a můžete zadávat SQL příkazy. Nápomocné vám budou `list tables` a `describe table`. Příkazy SQL se spustí pomocí `go`, z databáze se odpojíte standardně pomocí příkazu `disconnect`.

## ODBC ovladač

LibMDBODBC je ovladač, který pracuje s manažerem ovladačů `unixODBC` [5]. Umožňuje MDB Tools přistupovat k datům z programovacích jazyků PHP či Perl pomocí `DBD::ODBC`. V aktuální chvíli je implementace ODBC v MDB Tools spíš experimentální záležitostí. Ovladač je postaven nad knihovnou `LibMDB` a `LibMDBSQL`.

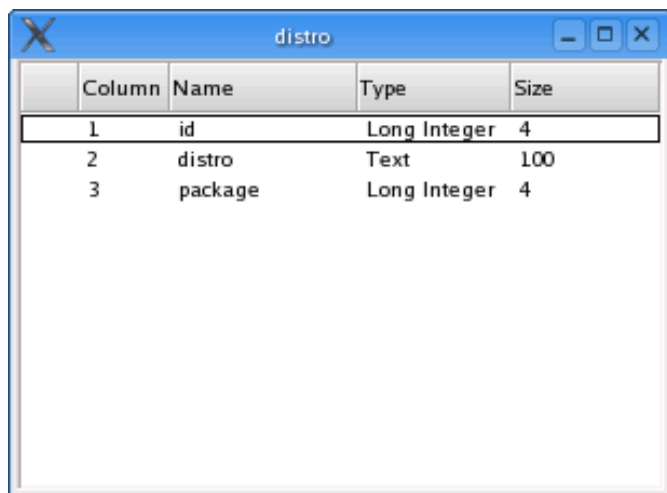
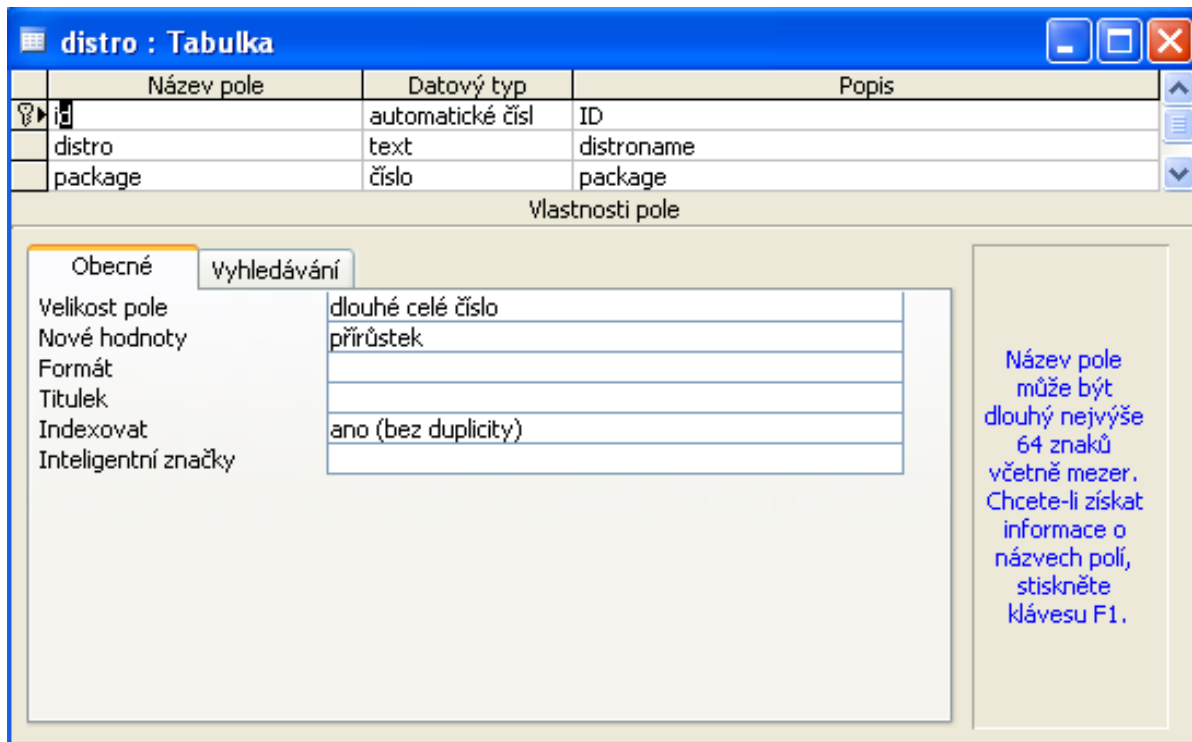
Projekt „`unixODBC`“ je samostatný a zcela nezávislý na MDB Tools a jeho cíl je vyvíjet a propagovat `unixODBC` jakožto standard ODBC na non-windows platformách a poskytovat vývojářům API pro přístup ke zdrojům dat, které v sobě zahrnují různé SQL servery a jiné zdroje dat, jež mají k dispozici `ODBC ovladače` [6]. `unixODBC` je vcelku dobře `zdokumentován` [7]. Zejména pak uživatelský manuál je přehledný, názorný a přitom stručný.

## Gmdb2

Gmdb2 je grafické uživatelské rozhraní pro práci s MDB Tools v X-Window. V jednotlivých záložkách je možné vidět databázové objekty v principu stejně jako v samotném MS Accessu. Není už však možné všechny podrobně prohlížet, natož editovat. Nechme se překvapit v dalších verzích MDB Tools.

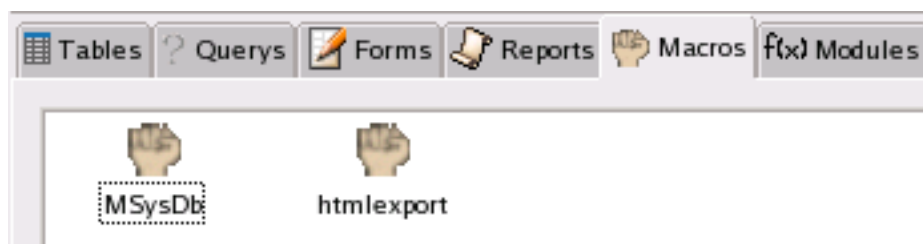
V prvním okně Gmdb2 jsou vidět dostupné tabulky databáze. Nejsou zde vidět tabulky systémové. Pomocí menu je pak možné zobrazit vlastnosti jednotlivé tabulky a data v tabulce uložená. Vlastnosti jsou tak trochu svéprávné, jelikož moje tabulka má například ve sloupci 1 definován typ pole jako automaticky přiřazované číslo, což je nativní typ Accessu. Gmdb2 však zobrazí Long Integer o velikosti 4. Stejná situace je i v shellu pomocí `mdb-schema`. Navíc definice tabulky je oproti zobrazení v samotném Accessu neúplná (indexy jsou avizované až v dalších verzích, ale chybí třeba i další atributy jednotlivých polí v tabulce).



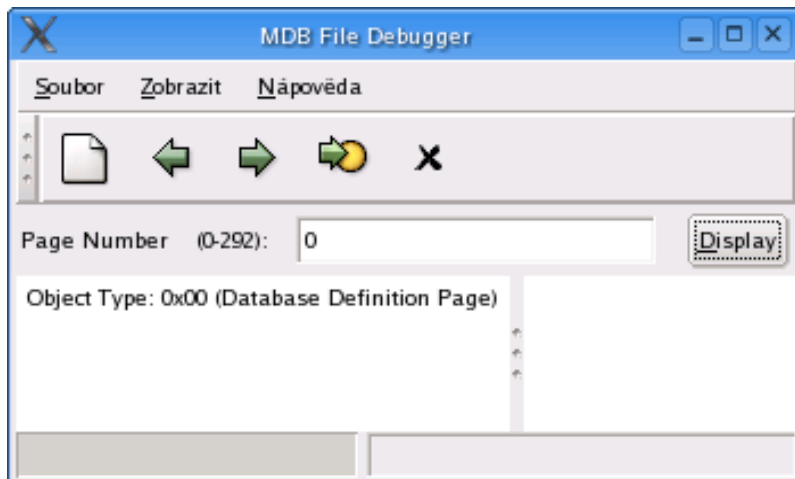
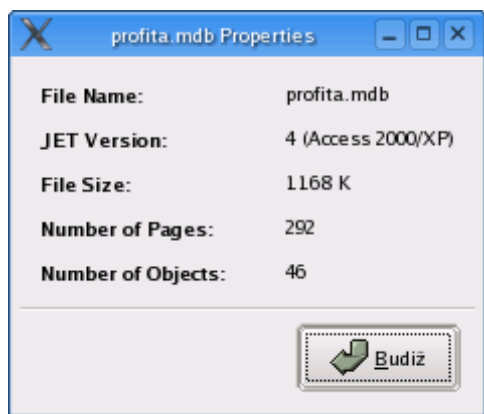


Prostě k tabulkám chybí mnoho informací jež se k ní v Accessu váží. Třeba primární klíč tabulky v souvislosti s relacemi považují za velmi důležitý a MDB Tools jej zatím bohužel neumí. Na druhou stranu je tento nástroj v neustálém vývoji, tak se snad dočkám. To hlavní z definice tabulky tam je. Další možnosti práce s tabulkou jsou zobrazení dat - budeme-li se držet terminologie MS Access, tak v datovém listu. Konečně pak možnost data z dané tabulky exportovat do textového souboru s možností nastavit řádkování, oddělovače polí, nastavit kvótu, znak kvóty a možnost zahrnout do výstupu hlavičku tabulky.

Ostatní záložky už jen zobrazují databázové objekty, ale nelze s nimi již dále pracovat. Jsou to: dotazy, formuláře, reporty, makra a moduly.



Gmdb2 umí ale i další věci, jako zobrazit verzi a informace o souboru .mdb nebo Debugger, který mi ovšem nedělá téměř nic jiného než padá a padá :o(. Vychytávkou jsou pak ovšem nástroje MDB Query Tool a Export Schema. Těm bych se rád věnoval podrobněji.



## Export Schema

Export Schema je GUI pro výše zmíněnou utilitu `mdb-schema` a umí tedy exportovat DDL databáze do textového souboru. Ve formuláři si vyberete umístění a název cílového souboru, tabulky, jejichž schéma byste chtěli exportovat – výchozí nastavení jsou všechny tabulky, dále pak dialekt (na výběr jsou již zmíněné Access, Oracle, Sybase, MS SQL Server, PostgreSQL). Volitelně pak můžete do výstupu zahrnout i relace (ty jsou kamenem úrazu, jelikož MDB Tools prostě relace ve výstupním souboru DDL nezobrazí, i když třeba vím, že je v databázi mám) a příkaz `DROP TABLE`, který slouží jako prevence v případě, že by již taková tabulka ve vaší databázi, do které se DDL importuje, existovala. Výsledný výstup pak může vypadat například takto:

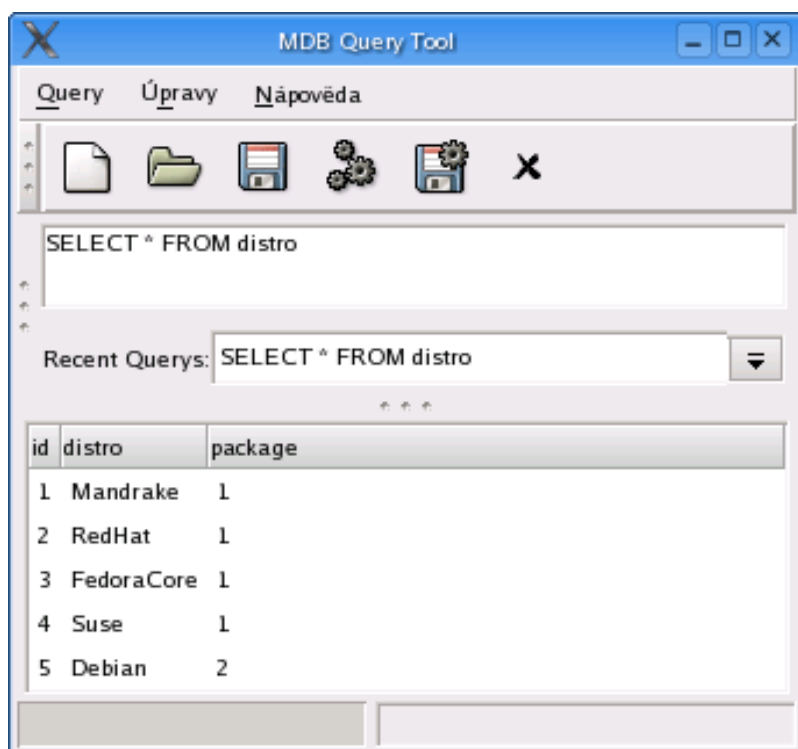
```
DROP TABLE distro;
CREATE TABLE distro
(
    id                int (4),
    distro            varchar (100),
    package           int (4)
);
-- CREATE ANY INDEXES ...

DROP TABLE package;
CREATE TABLE package
(
    package           int (4),
    packagename       varchar (100)
);
-- CREATE ANY INDEXES ...

-- CREATE ANY Relationships ...

relationships are not supported for sybase
```

## MDB Query Tool



MDB Query Tool je grafické uživatelské rozhraní pro dotazy jazyka SQL nad daty v tabulkách. Nečekejte žádné zázraky jako zvýrazňování syntaxe nebo dokonce automatické doplňování kódu, ale pro názornost a získání základních informací o datech uložených v databázi to stačí. SQL příkazy je možné ukládat do textového souboru a opětovně načíst a spustit. Výsledky dotazu můžete v tomto nástroji opět exportovat do formátovaného textového souboru. V principu stejným způsobem jako data z tabulek.

### K čemu je tedy klíč dobrý ?

Dokáží si představit situaci, kdy firma kompletně běžící na GNU/Linuxu (jak také jinak, že?), dostane od klienta databázi čehokoliv právě ve formátu MDB. Dejme tomu, že má data v databázi verifikovat, aktualizovat, doplnit či prostě jakkoliv změnit a poslat zákazníkovi zpět.

Úloha zdá se býti pro Linuxem vybavenou společnost zdánlivě složitá, ale s využitím MDB Tools není neřešitelná (pominu-li řešení Wine a zakoupení licence MS Access). Prostě a jednoduše se data exportují do formátu, který dokáže pojmout interní firemní databázový systém, data se zpracují a klient obdrží následně požadovaná data dejme tomu v .xls nebo .csv, který mu buď stačí nebo si ho zase pěkně naimportuje zpět do MS Accesu. Jediným, avšak dostatečně limitujícím faktorem je, že MDB Tools nemají zatím k dispozici českou lokalizaci (ani jinou než standardní anglickou) a nedokáží tudíž pracovat s daty v českém kódování.

### Případová studie

Dokáží si představit také mnohem složitější situaci – kdy po nás klient požaduje například online reporty z databáze MDB na webu. My upravujeme klientská data u nás ve firmě a klient chce mít představu, jak se data v jeho vlastní databázi mění, jaká je úspěšnost a podobné záležitosti. V prostředí MS Windows by byla tato situace dle mého názoru velmi obtížně řešitelná, pokud by byla „jen za pomoci Windows“ vůbec řešitelná.

V prostředí GNU/Linuxu bychom na to mohli vyžrát tak, že bychom soubor s databází umístili na SAMBA fileserver a zpřístupnili ho tak lidem, kteří by s ním v naší marketingové firmě měli pracovat. Soubor bychom si také zpřístupnili protokolem SMB ze stanice, na které bychom měli k dispozici MDB Tools. Pomocí démona crond a shell skriptu volajícího nástroje příkazové řádky z MDB Tools bychom zpracovali SQL dotazy do souboru MDB a jejich výsledek následně zpracovávali dejme tomu v PHP a poskytovali obsah webovému serveru.



Druhá možnost je samozřejmě přímo přes ODBC a DBD::ODBC opět s využitím PHP nebo Perlu. Pak už zbývá jen prezentační vrstva pro našeho náročného zákazníka a tou je webová stránka zpřístupněná ven do sítě internet pomocí směrování http požadavků z proxy našeho zákazníka na naši gateway. Tam už dejme tomu pomocí iptables směřujeme požadavek na výše zmíněný webserver a vesele ukazujeme, jak se daří naším pracovníkům plnit, upravovat či aktualizovat zákaznickou databázi.

To byla spíš taková „vědeckofantastická“ odbočka ;o). Pojdme k závěru.

## Klíč, který odemyká data

MDB Tools není univerzálním klíčem ke všem branám a dveřím v souborech .mdb vytvořených v databázi MS Access, ale některé dveře přeci otevře. A ty, které neotevře nyní, jistě brzy pokoří. Pokud ano, budeme opět o něco blíže k možnosti plného nahrazení MS Office Professional (jehož součástí právě Access je) linuxovými alternativami. Doufám, že se MDB Tools bude i nadále takto progresivně vyvíjet a dotáhne svou funkčnost třeba i dál než samotný MS Access. MDB Tools jsou vydávány pod licencí GNU General Public Licence, mají tedy ke kvalitnímu a přesto dynamickému vývoji ty nejlepší předpoklady. Osobně se zkusím podílet minimálně na lokalizaci do jazyka českého, pokud o to budou mít aktuální vývojáři zájem. Závěrem bych rád jen toliko: „Proč lézt někam oknem jako raubíř? Mám přeci klíč, zkusím to tedy raději dveřmi.“ ;o).

## Odkazy

- [1] <http://www.microsoft.com/office/access/default.asp>
- [2] <http://www.microsoft.com>
- [3] <http://www.unixodbc.com>
- [4] <http://www.unixodbc.com/download.html>
- [5] <http://www.unixodbc.com/manager.html>
- [6] <http://www.unixodbc.com/drivers.html>
- [7] <http://www.unixodbc.com/doc/>

\*\*\*

# Zpomalte si procesor

Pavel Beníšek

---

*K čemu může posloužit podtaktování procesoru a jak toho dosáhnout z uživatelského prostředí.*

---

Většina lidí, tedy hlavně hráčů, se snaží ušetřit na ceně procesoru, grafické karty nebo paměti. Ztracený výkon se potom snaží dohnat přetaktováním těchto komponent. To, že se mi to nelíbí, zde nechci rozpívat. Naopak se zaměřím na opačný problém - podtaktování.

## Proč zpomalovat?

Důvod mám jednoduchý. Dosažení maximální životnosti komponent. To se může hodit například na serveru skládaném z vlastních komponent, kde nikdy nemůžete zaručit bezvadný stav (hlavně mechanických částí, jako jsou ventilátory nebo pevné disky), a tak jim můžete alespon ulehčit. Snížení frekvence procesoru nám dá méně vyzářeného tepla, tím nižší teplotu ve skříní a následně s lepším chlazením i menší opotřebení komponent. Samozřejmě to není všespásná metoda, ale pouze jedna z dalších pomůcek.

Pro majitele notebooků je důvodem také spotřeba elektrické energie, která s frekvencí procesoru narůstá, takže lze několik desítek minut přidat výdrži na baterie. Přece jen pro psaní textu nebo úklid na pevném disku není třeba takt 2.5 GHz. Z vlastní zkušenosti mohu potvrdit, že pokud právě nekompiluji updaty systému (emerge systém v Gentoo Linuxu), tak bohatě vystačím s taktem 1,20 GHz, tedy polovičním. Vzhledem k dnešním výkonům procesorů je i při mé normální práci (desktop KDE a spuštěné aplikace Quanta Plus BE, Mozilla Firefox, Kmail, server Apache a LDAP a Pure-ftpd a databáze MySQL a PostgreSQL) výkon plně dostačující. A jak můžete sami posoudit, jedná se o plnohodnotné pracovní prostředí včetně spuštěných serverových aplikací.

Dalším důvodem může být šetření energií u PC postaveného pouze jako router, kde je výkon někdy až přemrštěný. Samozřejmě zůstává otázka, zda obsahuje podporovaný procesor.

## Spotřeba

Podarilo se mi najít pouze jedinou informaci o spotřebě energie v závislosti na frekvenci. Protože se mi na stránkách Intelu nepodařilo najít žádné relevantní informace k desktopové verzi procesoru Pentium 4, použil jsem informace o procesoru Pentium 4 M (mobilní verze), kde je frekvence ovládaná automaticky. Proto nebudou uvedená data přesně odpovídat, ale pro představu myslím postačí.

Procesor Pentium 4 M s pracovní frekvencí 2 GHz má tedy spotřebu 32 W. Při zpomalení na 1,2 GHz má odběr už jen 20,8 W. Tím jsme ušetřili třetinu spotřebované energie.

Nyní budu pracovat podle předpokladu, že poměr snížení frekvence ke snížení spotřeby u procesoru Pentium 4 M je podobný jako u ostatních procesorů.

Tedy např. procesor AMD 64 3400+ pracující na frekvenci 2200 MHz má udávanou spotřebu 89 W. Pokud tedy snížíme jeho takt na 1,32 GHz, pak by měla spotřeba klesnout na přibližně 58 W.

Protože však nemám k dispozici křivku závislosti frekvence na odběru (která jistě nebude lineární), další hodnoty lze pouze odhadovat. Pokud ji někdo má, bylo by zajímavé podělit se o ni s ostatními v diskuzi.

## Nastavení frekvence procesoru

Většinou se dá nastavit nižší frekvence i v BIOSu, což platí hlavně o procesorech Athlon. To je sice také možné, ale vzdáleně na tom pak už nic nezměníte. Navíc pokles bývá poměrně výrazný. Proto jsem hledal, zda existuje softwarové řešení. A světe div se: existuje a je velmi elegantní.

CPUSpeedy je script psaný v Pythonu. Umožňuje změnu pracovní frekvence procesoru v mezích možností jednotlivých procesorů, takže se nejedná o žádné praktiky ohrožující záruku.

CPUSpeedy pracuje s pomocí ovladače jádra CPUFreq. Takže ten musí být zkompileován. Také z toho vyplývá, že CPUSpeedy podporuje jen ty procesory, které jsou podporovány ovladačem CPUFreq.

Práce se skriptem je jednoduchá. Nabízí měnit frekvenci procesoru buď numerickou hodnotou následovanou mezerou a MHz/GHz, nebo hodnotami min, low, med, high, max. Pro výpis možných frekvencí použijeme:

```
#cpuspeedy -f
300000 KHz (300 Mhz)
600000 KHz (600 Mhz)
900000 KHz (900 Mhz)
1200000 KHz (1.20 GHz)
1500000 KHz (1.50 GHz)
1800000 KHz (1.80 GHz)
2100000 KHz (2.10 GHz)
2400000 KHz (2.40 GHz)
```

Ve výpisu zjistíme podporované hodnoty a jejich jmenné ekvivalenty, nebo dostaneme chybové hlášení, že není možné přistoupit k ovladači CPUFreq. Další nastavení je opravdu triviální:

```
#cpuspeedy med
cpuspeedy: CPU speed is now 1.50 GHz (62%)
```

Nastaví v mém případě 1.50 GHz. Můžeme si zjistit aktuální frekvenci parametrem `-s`.

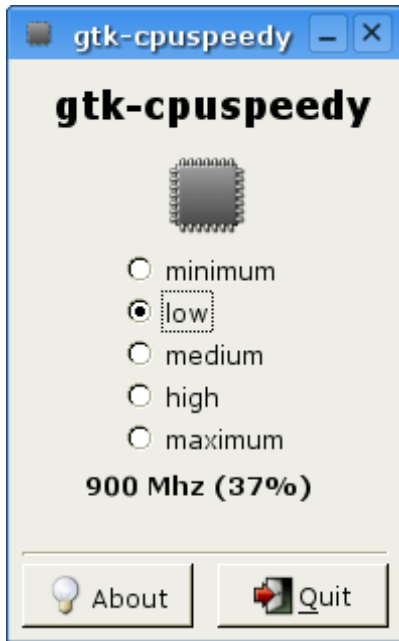
```
# cpuspeedy -s
cpuspeedy: CPU speed is 1.50 GHz (62%)

#cpuspeedy max
cpuspeedy: CPU speed is now 2.40 GHz (100%)
```

Obnoví maximální frekvenci procesoru (v mém případě 2.4 GHz).

## Závěr

Pro uživatele grafických prostředí existuje gtk nadstavba nazvaná gtk-cpuspeedy. Její použití je ještě jednodušší. Zvolení pracovní frekvence proběhne pouhým zaškrtnutím dané volby. Jak je vidět na obrázku, tak nenabízí všechny možné pracovní frekvence, ale jen pět základních.



CPUSpeedy je elegantní forma ovládání rychlosti CPU. Doufám, že si tento „anti-tuning“ oblíbíte stejně jako já. Jak samotné CPUSpeedy tak i grafickou nastavbu vytvořil Gabriele Giorgetti. Domovskou stránku projektu naleznete na stránce: <http://cpuspeedy.sourceforge.net> [1]. Asi nebudu sám, koho napadlo, že by se dalo řešit automatické zpomalování a zrychlování podle zatížení procesoru. A protože ani vy čtenáři, ani já jsem nebyl první, i na to existují hotová řešení. Ale o těch až příště.

## Odkazy

[1] <http://cpuspeedy.sourceforge.net>

\*\*\*

# Inštalácia Slackware

Andrej Krivulčík

---

*V tomto seriáli článkov si ukážeme, prečo a ako nainštalovať Slackware Linux (9.1). Neskôr ho skúsime aj trochu nakonfigurovať k obrazu svojmu a naučíme sa niekoľko tipov a trikov.*

---

## Úvod

Ako už býva pri takýchto seriáloch zvykom, v prvej časti sa budeme venovať skôr teoretickej stránke veci. K samotnej inštalácii sa dostaneme až neskôr.

## Motivácia

Najprv bychom sa mali pozastaviť nad tým, prečo (ne)inštalovať práve Slackware. Existuje predsa toľko rôznych distribúcií, mnohé sú „užívateľsky prítulnejšie“, majú grafickú inštaláciu, rôzne grafické konfiguračné nástroje (toto v Slacku nehľadajte). Navyše iné distribúcie (Red Hat/Mandrake, Debian) majú pomerne vyvinutý balíčkovací systém so závislosťami (na ktorý mnohí nadávajú), teda sa nemôže stať (ak o to explicitne nepožiadate), že si nainštalujete program bez knižníc, na ktorých závisí. Slack má síce akýsi balíčkovací systém (založený na .tgz archívoch), lenže tento nerieši závislosti, teda musíte vedieť, čo robíte, aby ste mali funkčný systém. Prečo teda inštalovať systém, ktorý postráda takéto vymoženosti?

Odpoveďou by mohla byť vec zvaná „čistota systému“, prípadne administrátorov „pocit kontroly nad systémom“.

Keďže iné distribúcie sa snažia fungovať na všetkých možných konfiguráciách automaticky, obsahujú množstvo rôznych skriptov a programov, ktoré ošetrojú najrôznejšie možné situácie. Je ich pochopteľne veľa, vykonávajú sa pri každej vhodnej príležitosti, teda štart systému je pomalší, rovnako tak jeho zastavenie, spúšťanie služieb atp. trvá dlhšie... V Slackware je naproti tomu zabezpečená základná funkčnosť, ktorá možno neobsahuje niektoré vymoženosti, no od administrátora sa očakáva, že si dokáže systém prispôbiť. Odmenou za jednorazovú konfiguráciu je systém, ktorý je prispôsobený počítaču bez rôznych „barličiek“. Navyše autor Slackware (Patrick Volkerding) si naozaj zakladá na dodržiavaní rôznych štandardov, teda to, čo je v Slackware, je pravdepodobne aj „správne“ (ono skoro nikomu na tom nezáleží, ale ten pocit je fajn :-)).

Distribúcie ako Mandrake sú určené prevažne pre začiatok a snažia sa im vyjsť maximálne možne v ústrety. Preto obsahujú rôzne grafické konfigurátory (napríklad Mandrake Control Center), pomocou ktorých sa dá nastaviť väčšina možností v systéme. Administrátor si spustí MCC, nastaví nastavenia dial-up pripojenia a ono to naozaj „nejak“ funguje. Lenže ani sám admin nevie, ako. Chce zmeniť niečo podrobnejšie? Nájde si príslušný konfiguračný súbor, pozmení a je to. Lenže po ďalšej úprave v MCC sa tento ochotne prepíše (už som takto niekoľkokrát prišiel o konfiguráciu)... Ak teda chceme mať systém, kde vieme, kde čo je, a ako to funguje, je Slackware tou pravou voľbou.

No na druhú stranu, prečo Slackware *neinštalovať*? Keďže vyžaduje viac konfigurácie od admina, je vhodné mať už nejaké skúsenosti s Linuxom. Keďže vyžaduje viac konfigurácie od admina, nie je vhodný pre lenivých, brain-dead adminov, alebo ľudí, ktorí nemajú veľmi čas.

Keďže vyžaduje viac konfigurácie od admina, nie je úplne vhodný pre ľudí, ktorí potrebujú rýchlo pracovať (potrebujem napísať a vytlačiť dokument v OpenOffice.org, je mi jedno, či mi bude systém každý deň štartovať o minútu dlhšie).

*Pravidlo:* Ak máte čas, chuť učiť sa a poznávať fungovanie systému, vôľu mať čistý systém, nad ktorým (snáď) budete mať kontrolu, linuxové znalosti alebo ľubovoľnú kombináciu z uvedeného, uvažujte nad Slackware. Slackware-pozitívny kamarát poruke je *veľké* plus. Ak s počítačom potrebujete rýchlo pracovať, je vám jedno, ako funguje (pokiaľ funguje) systém, ste pohodlní (a.k.a. leniví), asi to nebude stáť za to. Porozmýšľajte nad Mandrake, Red Hatom, SUSE, Microsoftom.

## Čo k tomu potrebujeme?

Ak ste sa rozhodli pre Slackware (správna voľba :-)), iste vás zaujíma, čo vás bude táto sranda stáť.

### Ingrediencie



**Slackware Linux** – ťažko nainštalujeme Linux bez Linuxu. Teda, kde ho zobrať? Objednať v [The Slackware Store](#) [1]; kúpiť na [qwert.cz](#) [2]; stiahnuť z ľubovoľného [mirroru](#) [3] (napríklad [tento rakúsky](#) [4]) a vypáliť; zohnať od kamaráta... Kompletný Slack je šírený na 4 CD, no na „normálnu“ inštaláciu nám postačia prvé dve. Na ďalších dvoch sú zdrojové kódy programov z prvých dvoch CD. Ak nemáte špeciálne požiadavky (pre ktoré by ste si museli/chceli kompilovať software sami), tieto dve CD nebudú potrebné (ja ich ani nemám). Všetko kompilovať sami nebudete a potrebný software sa vždy dá stiahnuť z netu (v novej verzii).

**Počítač** – V zásade stačí akýkoľvek x86 kompatibilný počítač. Ak nie je príliš nový, prípadne príliš starý, prípadne nepoužíva nejaký exotický hardware, pravdepodobne s ním nebudú žiadne problémy.

Demonštračná inštalácia bude prebiehať na stroji s parametrami: AMD Athlon 600 Mhz, 256 MB RAM, 10 GB HDD, ATI Rage 128 Pro 32 MB, Sound Blaster PCI128, interný winmodem „Super Voice“ s chipsetom Lucent, CDRom, CDRW, „štandardné“ periférie (myš, klávesnica, monitor). Takmer všetky komponenty sú kupované v roku 1999 a nie sú s nimi problémy.

Najväčší problém bude pravdepodobne s rozdelením diskového priestoru. Linux bude potrebovať svoju vlastnú partition (diskový oddiel). Ak nemáte osobitný disk, budete musieť zmenšiť diskový priestor existujúcemu systému a na uvoľnenom priestore vytvoriť novú partition pre Slacka (a ešte pravdepodobne partition pre swapovaciu pamäť). Uvoľnením priestoru na disku sa v tomto seriáli zaoberať nebudem, pretože Slacka budem inštalovať na osobitný disk.

**Čas** – okrem času potrebného na inštaláciu bude potrebný aj istý čas na konfiguráciu a vyladenie systému.

Teda, ak ste sa rozhodli Slackware si nainštalovať, môžete sa pustiť do zháňania potrebných ingrediencií (chýbať vám bude asi len Slack samotný), nabudúce už systém nainštalujeme a dostaneme sa k prvému bootu nového Slacku.

## Inštalácia

*Upozornení:* V tomto článku je opísaný postup, ako na počítač nainštalovať operačný systém. Tento proces môže spôsobiť (a pravdepodobne spôsobí) stratu dát. Je už na vašej zodpovednosti, aby tie dáta neboli životne dôležité, dôležité, ani „len trochu dôležité“. V žiadnom prípade nenesiem žiadnu zodpovednosť za prípadné ujmy na vašom psyché, materiálne ujmy, zdravotné problémy, rozhodené sociálne vzťahy a podobné vysoko pravdepodobné následky dodržiavania nižšie uvedeného postupu :-). Dopredu by som mal upozorniť na to, že článok nie je myslený ako vyčerpávajúci sprievodca inštaláciou. Je to popísaný postup, ktorým prejde „typická“ inštalácia (aj keď o význame tohto slova by sme tiež mohli veľa polemizovať) s niekoľkými radami pre užívateľa, keď si pri niektorom kroku nie je úplne istý, čo zvoliť.

## Bootovanie

Po vložení prvého inštalačného CD do mechaniky (z ktorej sa dá bootovať) na nás po chvíli vyskočí prvý prompt, pri ktorom si môžeme vybrať, aký kernel má naboťovať (obr. 1). Aby bolo možné nainštalovať Slacka na rôzne počítače, je dodaných niekoľko kernelov, každý má podporu nejakého zariadenia navyše (ak máte napríklad SCSI disky apod.). Ak inštalujete na obyčajný domáci počítač, je veľmi pravdepodobné, že nebudete potrebovať žiadny špeciálny kernel, preto odklepeme **Enter** a necháme kernel naboťovať. Ak nemáme možnosť naboťovať z CD, budeme musieť vytvoriť bootovaciu disketu (postup pre DOS je na CD v adresári **bootdisks**, súbor **README.TXT**, pod Linux použijeme **dd if=naša\_image of=/dev/fd0** s disketou v mechanike).

Po chvíli bootovania nás prekvapí ďalšia výzva, tentoraz si môžeme zapnúť podporu non-US (teda národných) klávesníc. Táto voľba je pekná, no my ju nevyžijeme, pretože fonty v konzole sa budú dať zmeniť až neskôr, a národná klavesnica nám je počas inštalačného procesu zbytočná. **Enter** Tretia výzva, ktorá nás čaká, je už login (obr. 2). Píše sa tam, že sa môžeme prihlásiť ako root, prečo tak neurobiť (dokonca stačí odklepnúť **Enter**). Na heslo sa nás pýtať nikto nebude, ak predsa, je to znak, že máme málo pamäte. Po prihlásení sa nám vypíšu ďalšie inštrukcie, ktorými sa budeme riadiť (obr. 3). Prvé dva odstavce vravia o upgradovaní existujúceho Slackware systému, teda to sa nás netýka. My budeme inštalovať nový systém z CD, teda potrebujeme spraviť dva kroky: rozdeliť disk a spustiť **setup** (v ktorom je viacero ďalších podkrokov).

```
ISOLINUX 2.06 2003-08-22 Copyright (C) 1994-2003 H. Peter Anvin
```

```
Welcome to Slackware version 9.1 (Linux kernel 2.4.22)!
```

```
If you need to pass extra parameters to the kernel, enter them at the prompt
below after the name of the kernel to boot (scsi.s etc). NOTE: In most cases
the kernel will detect your hardware, and parameters are not needed.
```

```
Here are some examples (and more can be found in the BOOTING file):
```

```
    hdx=cyls,heads,sects,wpcm,irq (needed in rare cases where probing fails)
or hdx=cdrom (force detection of an IDE/ATAPI CD-ROM drive) where hdx can be
any of hda through hdh.
```

```
In a pinch, you can boot your system from here with a command like:
```

```
For example, if the Linux system were on /dev/hda1.
```

```
boot: bare.i root=/dev/hda1 noinitrd ro
```

```
This prompt is just for entering extra parameters. If you don't need to enter
any parameters, hit ENTER to boot the default kernel "bare.i" or press [F2]
for a listing of more kernel choices.
```

```
boot: _
```

Obr.1



```
Welcome to the Slackware Linux installation disk! (version 9.1.0)

##### IMPORTANT! READ THE INFORMATION BELOW CAREFULLY. #####

- You will need one or more partitions of type 'Linux' prepared. It is also
  recommended that you create a swap partition (type 'Linux swap') prior
  to installation. For more information, run 'setup' and read the help file.

- If you're having problems that you think might be related to low memory (this
  is possible on machines with 16 or less megabytes of system memory), you can
  try activating a swap partition before you run setup. After making a swap
  partition (type 82) with cfdisk or fdisk, activate it like this:
    mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

slackware login: _
```

Obr. 2

```
- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

slackware login: root

Linux 2.4.22.

If you're upgrading an existing Slackware system, you might want to
remove old packages before you run 'setup' to install the new ones. If
you don't, your system will still work but there might be some old files
left laying around on your drive.

Just mount your Linux partitions under /mnt and type 'pkgtool'. If you
don't know how to mount your partitions, type 'pkgtool' and it will tell
you how it's done.

To partition your hard drive(s), use 'cfdisk' or 'fdisk'.
To activate PCMCIA/Cardbus devices needed for installation, type 'pcmcia'.
To activate network devices needed for installation, type 'network'.
To start the main installation, type 'setup'.

root@slackware :/#
```

Obr. 3

## Rozdeľovanie disku

Na rozdelenie disku sú tu dva možné nástroje: fdisk a cfdisk. Nie je tu žiadny grafický nástroj a tento krok je asi najproblematickejší z celého inštaláčného procesu. Ešte pred spustením nástroja si musíme ujasniť, ako budeme disk rozdeľovať. Potrebné budú minimálne dva diskové oddiely, jeden na Linux a druhý na swap (disková pamäť). Môžeme disk rozdeliť aj na viac oddielov, potom by boli užívateľské dáta na inom oddieli ako programy, logy by tiež mohli byť na inom. Takéto rozdelenie má význam, keď nám zaleží na rýchlosti, prípadne bezpečnosti systému, alebo keď používame viac



distribúcií naraz (dáta užívateľov len pripojíme v každom systéme). Pri domácom použití je toto riešenie zbytočné (navyše musíme správne odhadnúť veľkosť, pri zlom odhade máme buď veľa miesta zbytočne nevyužitého, alebo budeme musieť oddiely rezizovať (pri nedostatku miesta)). Preto disk rozdelíme iba na dva oddiely, jeden bude koreňový adresár (/), druhý bude spomínaný swap. Veľkosť swapu sa všeobecne odporúča nastaviť na dvojnásobok operačnej pamäte. Keďže máme dostatok miesta na disku, túto radu poslúchneme (ak máme veľa operačnej pamäte, ale málo diskového priestoru, swap môže byť aj oveľa menší – prevádzkoval som Mandrake (prostredie KDE) s cca 10 MB swap pri 256 MB RAM bez problémov).

Partitionovať budeme programom cfdisk. Spustíme ho príkazom `cfdisk /dev/hd?`, za otáznik dosadíme písmeno nášho disku – podľa zapojenia v počítači:

- primary master – `/dev/hda`
- primary slave – `/dev/hdb`
- secondary master – `/dev/hdc`
- secondary slave – `/dev/hdd`

U mňa mám disk na secondary slave, teda spustíme `cfdisk /dev/hdd`.

Na obr. 4 vidíme, že na disku sú dva oddiely (jeden je logický), sú to relikty zo systému Mandrake a v inštalačnom procese ich nemilosrdne zmažeme. V praxi by sme ani repartitionovať nemuseli, lenže to, že je swap na logickom oddieli, je zbytočné (význam by to malo, keby boli na disku viac ako 4 oddiely). A okrem toho si chceme ukázať proces delenia disku :-).

```

cfdisk 2.12

          Disk Drive: /dev/hdd
          Size: 10245537792 bytes, 10.2 GB
    Heads: 255   Sectors per Track: 63   Cylinders: 1245

-----
Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
hdd1     Boot      Primary   Linux ext3   9862.12
          Pri/Log   Free Space  0.04*
hdd5     Logical   Linux swap  378.34*

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition

```

Obr. 4

Pre oba oddiely teda vyberieme možnosť *DELETE*, čím ich zmažeme. Položkou *New – Primary* vytvoríme novú partíciu, veľkosť zvolíme tak, aby zostalo dostatok miesta na swap. Na swap necháme okolo 500 MB, teda veľkosť prvého oddielu bude okolo 9750 MB. Tejto prvej partícií zapneme *Bootable flag*. Vo zvyšku voľného miesta vytvoríme swap partition: *New – Primary*, odentrujeme výchozú veľkosť (celý zvyšok disku) a zmeníme jej **Type** z 83 (Linux native) na 82 (Linux swap). Teraz máme poslednú možnosť partitionovanie bez následkov zrušiť. My však vyberieme položku

*WRITE* a partition table bude zapísaná na disk (obr. 5). Po potvrdení zápisu tabuľky na disk sú dáta (pravdepodobne) nenávratne preč (ak sme teda zmenili rozdelenie disku).

System nás požiadal o reštart systému (obr. 6), aby sa partition table korektne načítala, vyhovieme mu preto a reštartujeme počítač príkazom `reboot` (alebo populárnym trojhmatom `Ctrl-Alt-Del`).

```
          cfdisk 2.12

          Disk Drive: /dev/hdd
          Size: 10245537792 bytes, 10.2 GB
          Heads: 255   Sectors per Track: 63   Cylinders: 1245

  Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
  hdd1      Boot        Primary    Linux         9746.96
  hdd2                        Primary    Linux swap    493.52
```

```
[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]
```

Write partition table to disk (this might destroy data)

Obr. 5

```
-----
  hdd1      Boot        Primary    Linux         9746.96
  hdd2                        Primary    Linux swap    493.52
```

```
[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]
```

Disk has been changed. program without writing partition table  
Reboot the system to ensure the partition table is correctly updated.

WARNING: If you have created or modified any  
DOS 6.x partitions, please see the cfdisk manual  
page for additional information.

root@slackware:/#

Obr. 6

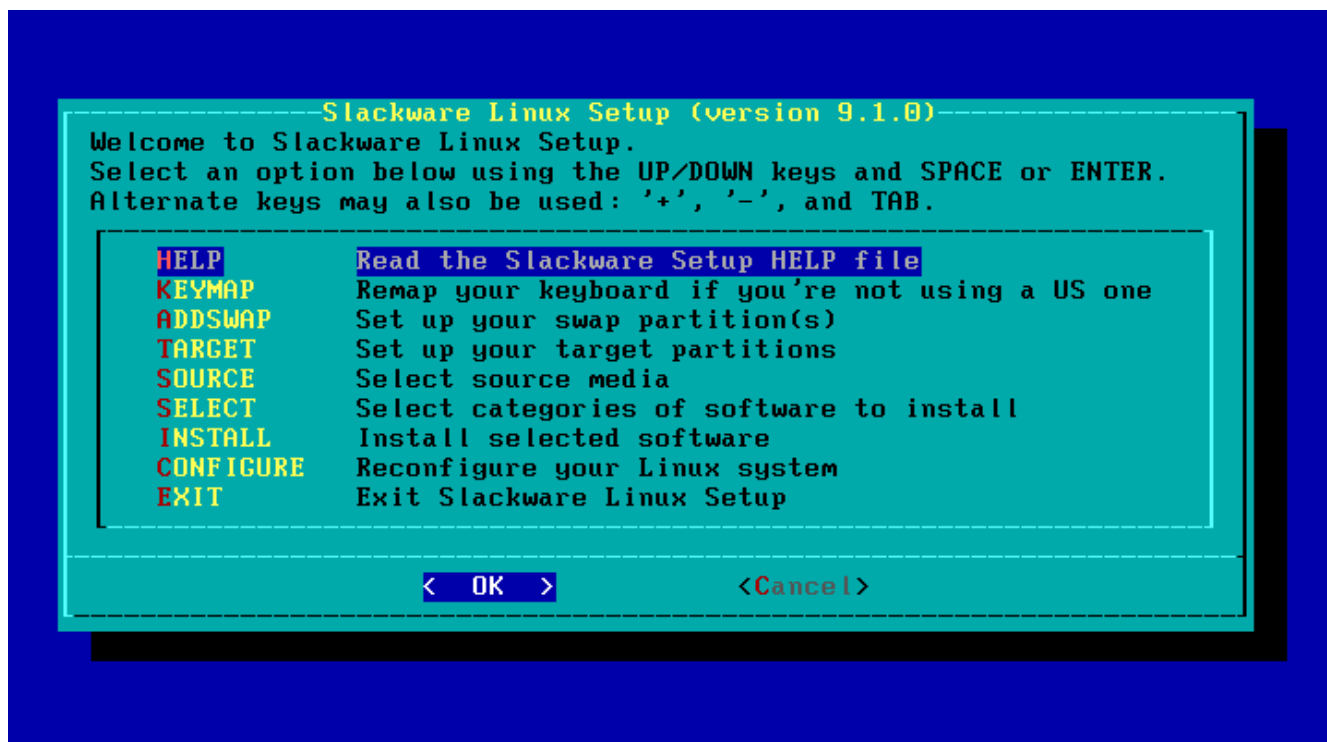
Rozdelenie disku máme teda úspešne za sebou, už nám zostáva „len“ samotná...

## Inštalácia systému

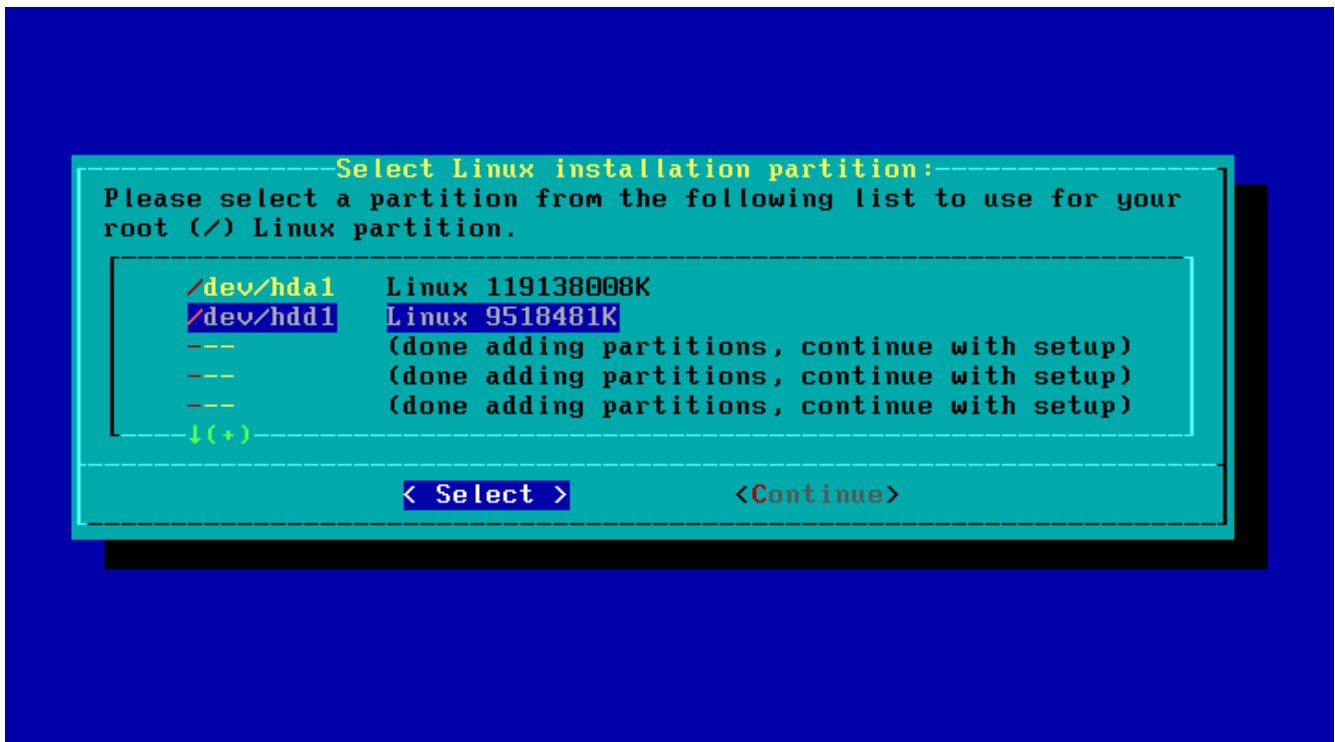
ktorá je vyriešená cez textovo-grafický inštalátor, ktorý spustíme príkazom `setup` (obr. 7). Na výber máme viacero položiek, ktorými nás inštalátor povedie. Na začiatok máme možnosť prečítať si súbor s nápodvedou (odporúčam), prípadne opäť si zmeniť klávesnicu (ak sme zmenili názor :-)). Treťou položkou v menu (`ADDSWAP`) máme možnosť nastaviť swap, čo aj urobíme.

### Príprava

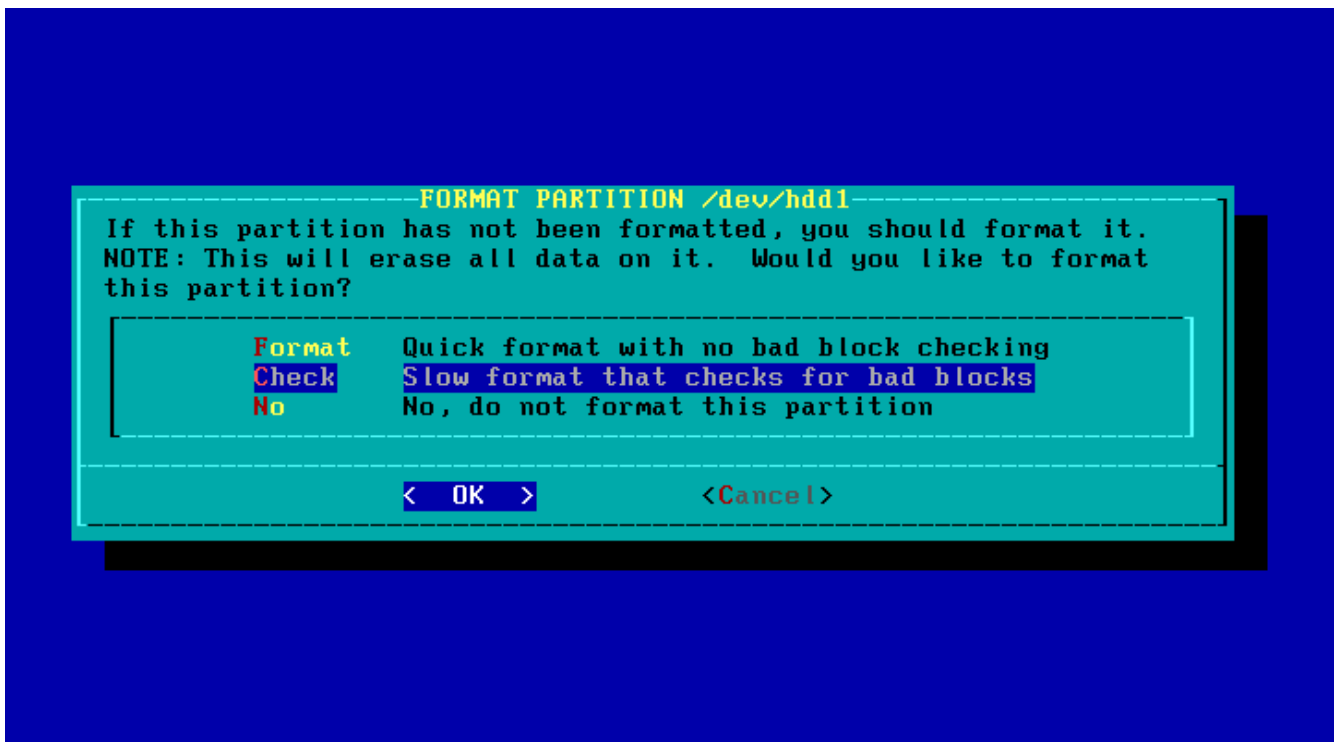
V ďalšom kroku sa nás inštalátor spýta, na ktorý diskový oddiel chceme inštalovať systém (obr. 8). Vyberieme práve vytvorený oddiel (na obrázku je ešte jeden oddiel (zo 120 GB disku, kde Slack už pekne dlho veselo úraduje)). Inštalátor sa nás opýta, či si želáme tento oddiel naformátovať (obr. 9). Keďže ešte naformátovaný nie je, vyberieme položku `Check`, ktorá nám okrem naformátovania diskový oddiel aj skontroluje. Pozor, doteraz ešte teoreticky bolo možné zachrániť dáta z daného disku. Po tomto kroku sa to stane veľmi problematické (ak vôbec možné)! Program sa nás opýta, aký súborový systém chceme na nový diskový oddiel naformátovať, jako vyzhová nám ponúkne ReiserFS (obr. 10), a nevidím dôvod, prečo ho nepoužiť. Je to moderný súborový systém, ktorý je spoľahlivý, stabilný, podporuje žurnálovanie (teda zápis na disk sa najprv poznačí do žurnálu, potom sa vykoná a ako vykonaný sa označí až po úspešnom dokončení. Keby počas procesu zápisu vypol prúd, k žiadnemu poškodeniu dát nedochádza. Pamätáte sa na kontrolovanie integrity súborového systému po nekorektnom odstavení ext2 alebo FAT\*? Tu ideme na istotu, adrenalínové športy si necháme mimo pracovnú dobu :-)), ak nemáme osobitný dôvod, prečo ho nepoužiť (možnosť čítať dáta z iného systému...), voľme ten. Prehľad nových súborových systémov, ich porovnanie, použitie a výhody nájdeme napríklad v [tomto seriáli](#) [5].



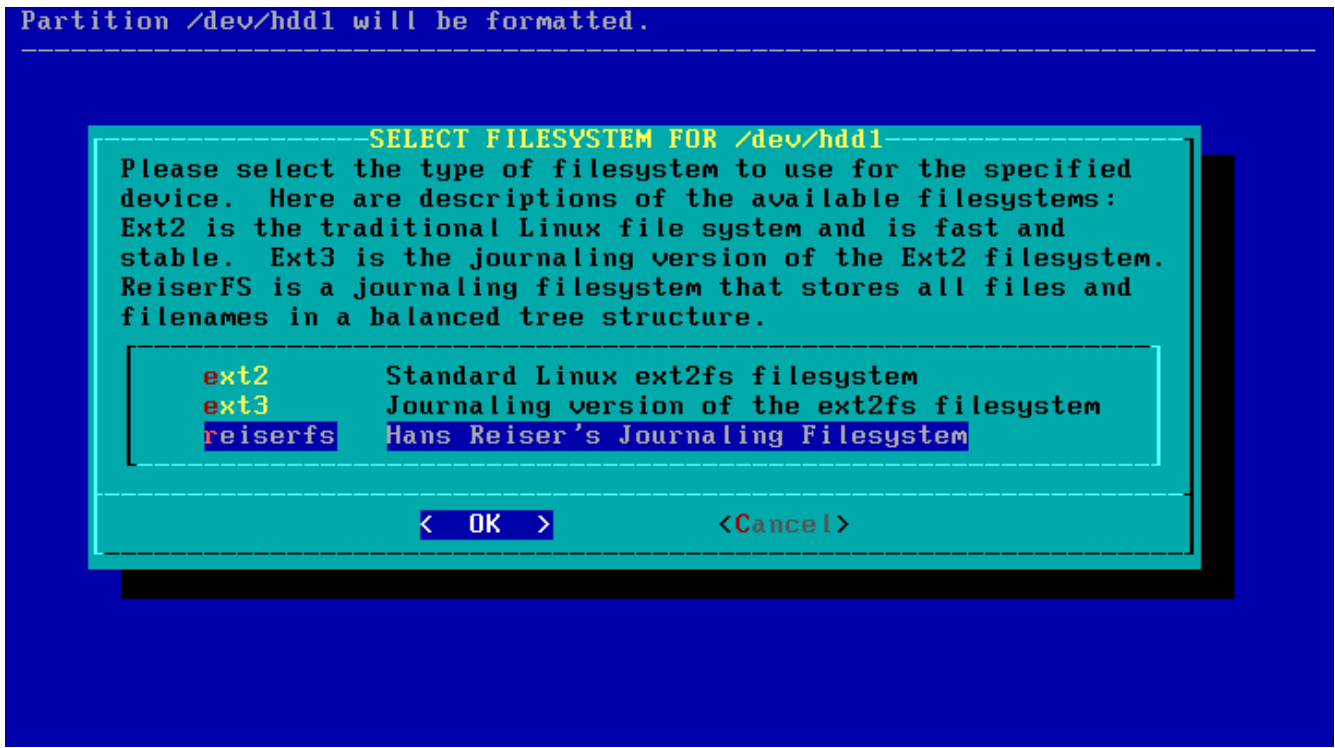
Obr. 7



Obr. 8



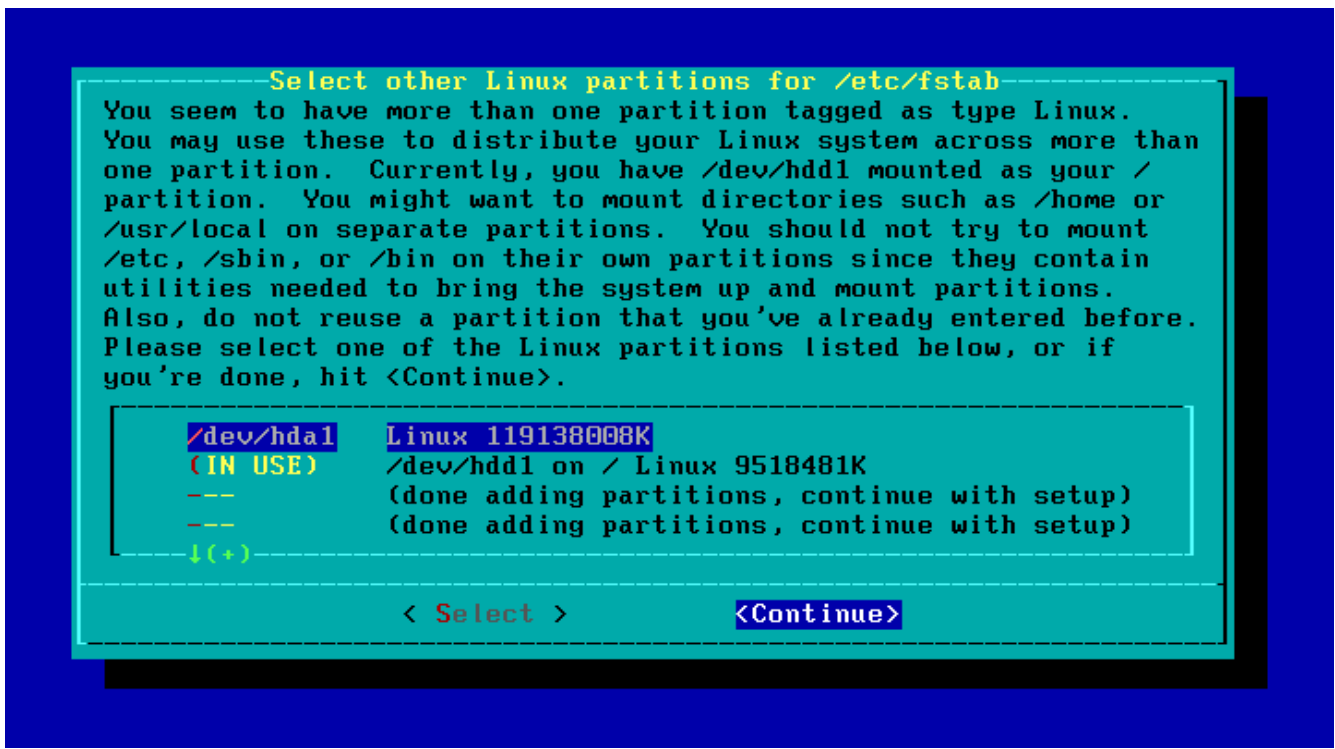
Obr. 9



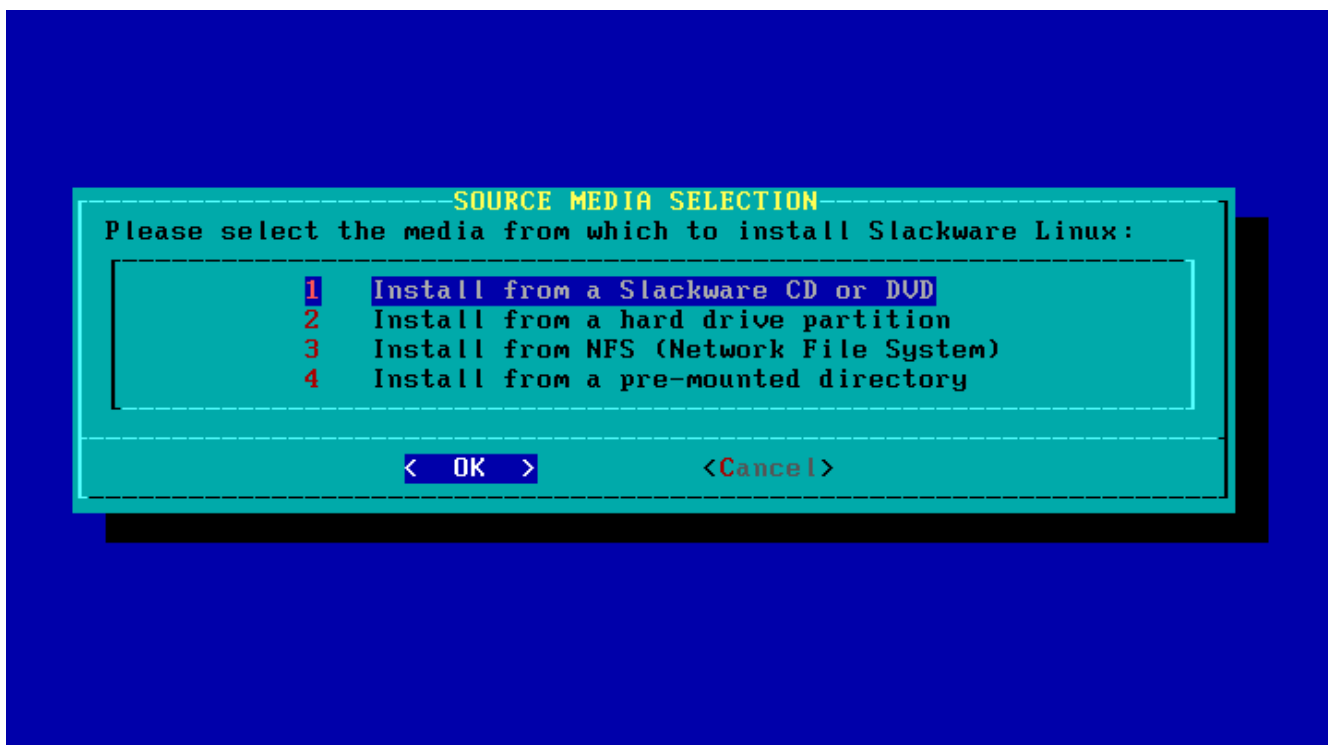
Obr. 10

Po naformátovaní oddielu inštalátor detekuje všetky ďalšie diskové oddiely označené pre Linux a ponúkne nám ich nastaviť, aby sme ich pripájali napríklad ako `/home`, `/usr/local`... Keďže v tomto prípade túto možnosť nevyužijeme a druhý linuxový diskový oddiel už jeden Linux obsahuje, zvolíme možnosť `Continue` bez výberu dodatočnej partície (obr. 11).

Po výbere cieľa si môžeme vybrať zdroj balíčkov; inštalujeme z CD, voľba je teda myslím jasná (obr. 12). Detekciu zariadenia necháme na automatiku, v prípade problémov pomôže manuálne nastavenie zdroja (zariadenia sú podľa umiestnenia na kábloch, pamätáte?).



Obr. 11

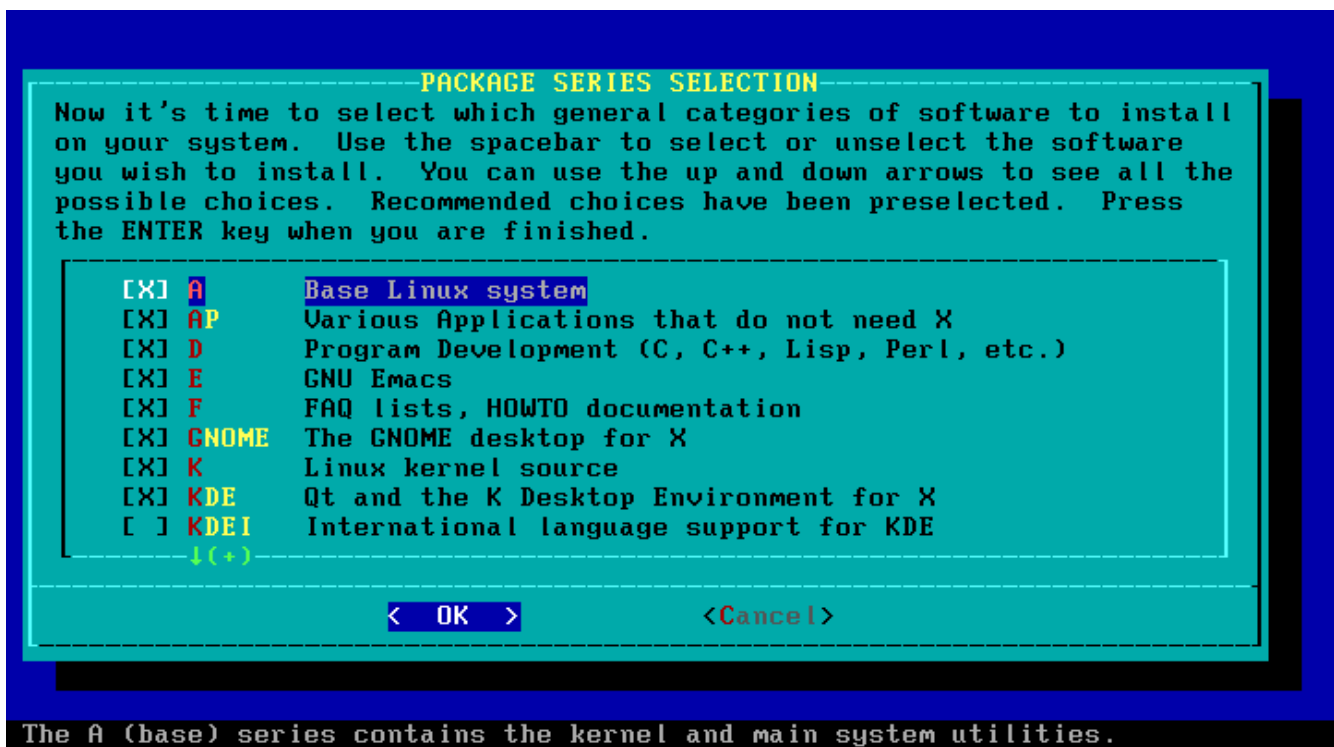


Obr. 12

## Inštalácia

Ďalšie dva kroky budeme vyberať software, ktorý chceme nainštalovať na disk. Prvým krokom je výber skupín balíčkov (obr. 13). Štandardne sú vybrané všetky okrem internacionalizácie KDE. Ak budeme chcieť KDE používať, bude vhodné túto skupinu vybrať tiež. V zásade platí, že ak máme dostatok voľného miesta, vybrať môžeme všetko. Zrušiť môžeme skupiny, o ktorých vieme, že nám budú zbytočné (napríklad pre nás vimistov bude celá skupina GNU Emacs na dve veci :-)). Ak máme málo miesta, vypínať budeme pravdepodobne: Kernel source, GNOME/KDE (alebo oboje), Emacs (?), TeX (ak nebudeme robiť DTP, je to zbytočne obsadených 150 MB).

Po výbere skupín balíčkov si vyberieme spôsob výberu balíčkov samotných (obr. 14). Ak máme veľa času, nervov a chceme hneď zo začiatku niečo pochytiť, odporúčam spôsob **newbie**. Inštalátor sa pýta na každý balíček osobitne, pričom pri každom napíše stručnú charakteristiku a obsadené miesto. Ak vieme, čo chceme (a naopak, čo nechceme), bude vhodná možnosť **menu** alebo **expert**, tu si môžeme vybrať balíky s jednoriadkovým opisom (rýchle, keď vieme, čo chceme). Keď vyberieme **expert**, vo výberoch sa nám budú zobrazovať aj balíčky nevyhnutné pre beh systému. Ak ich nechceme omylom vypnúť, vyberieme **menu**. Ak nemáme čas, alebo proste nechceme nič pokaziť, vyberieme voľbu **full**, ktorá nainštaluje všetko, čo sa vo vybratých skupinách nachádza. Je toho okolo 2 GB, teda pozor na miesto na disku :-). Položky **custom** a **tagpath** sú užitočné, keď chceme identický systém nainštalovať na viacero počítačov. Priznám sa, že som to neskúšal.



Obr. 13

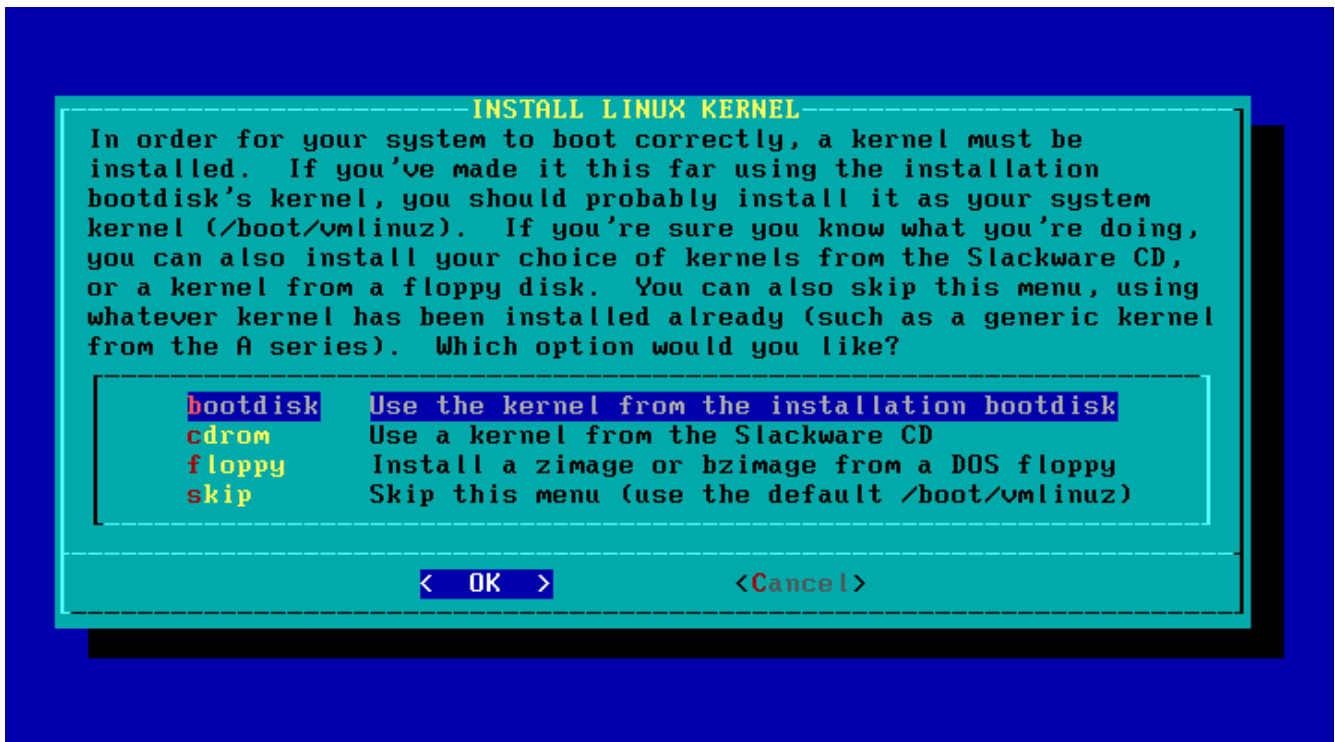


Obr. 14

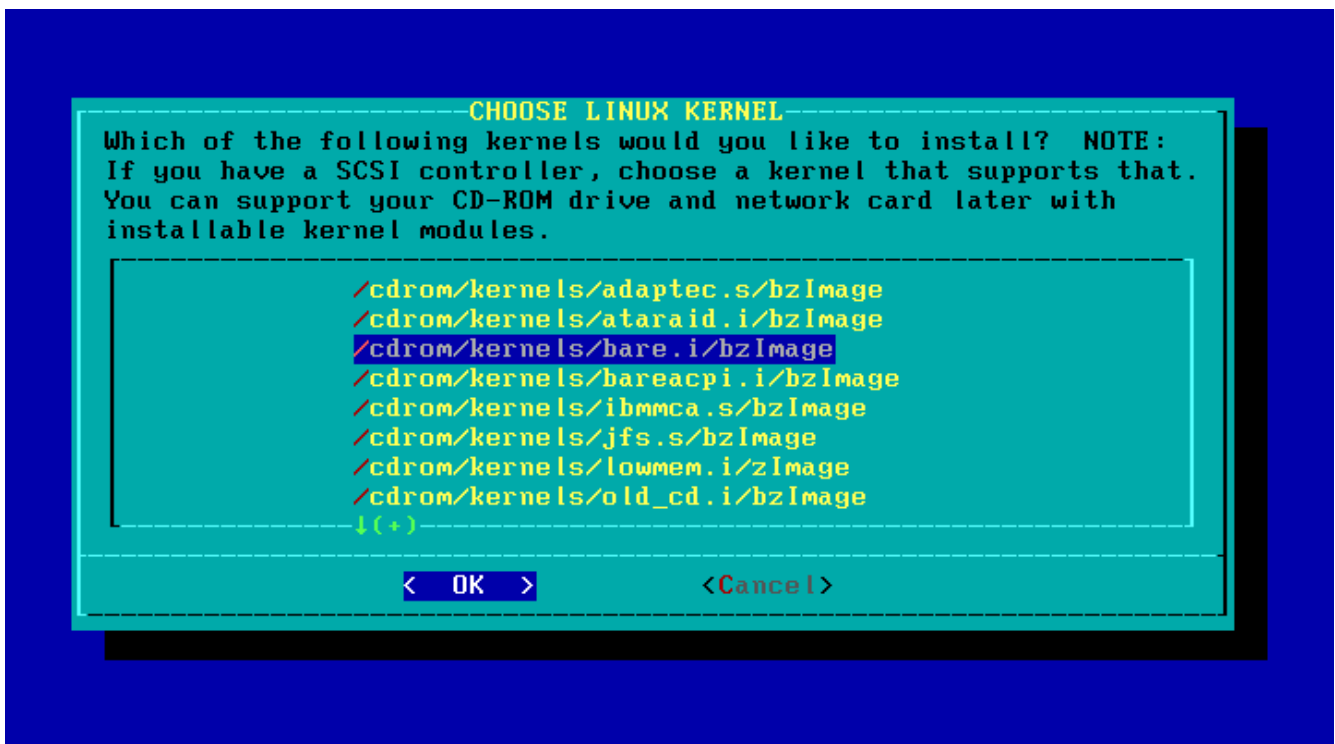
Po tomto kroku a vybratí balíčkov (ak sme nezvolili **full**) nasleduje inštalácia balíkov z CD. V priebehu si inštalátor asi vypýta druhé CD (na ktorom je KDE a GNOME), teda z obeda si občas odskočte skontrolovať priebeh inštalácie.

### Poinštaláčn konfigurcia

Po nainštalovan balíčkov sa ns inštaltor spta, aky kernel chceme v novom systme pouzvat (obr. 15). Ak sme doteraz nenarazili na iadny problm, nemme dvod nevybrať kernel, na ktorom fungujeme. Vyberieme teda druh polozku a z dlšieho menu vyberieme vychoz moznosť (obr. 16).



Obr. 15

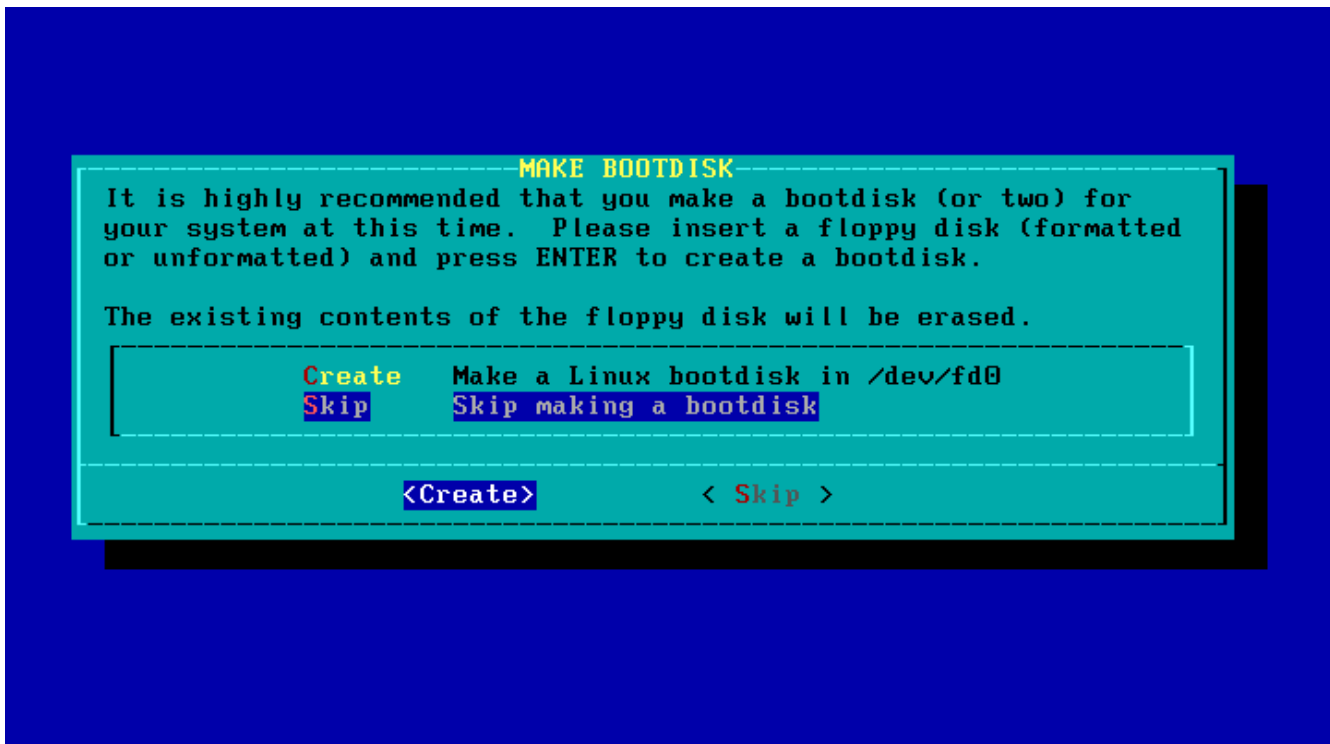


Obr. 16

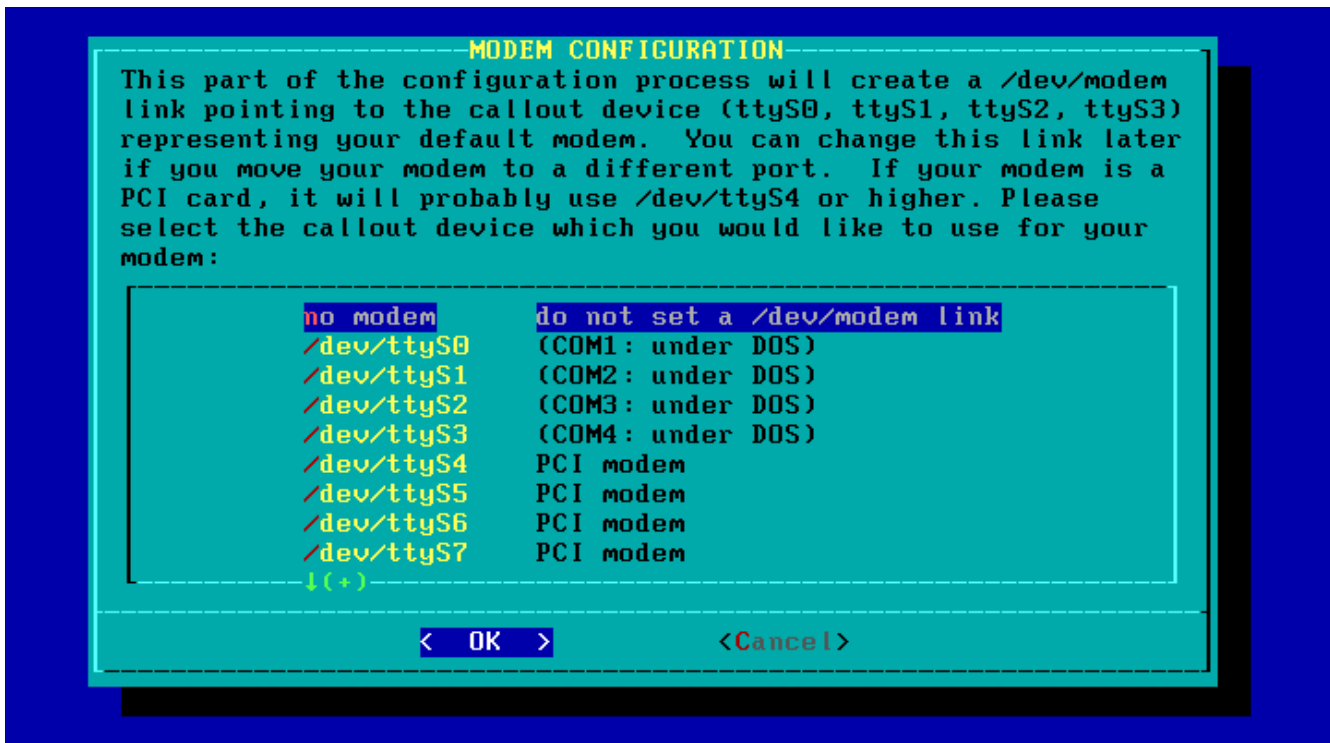
Teraz máme možnosť vytvoriť bootovaciu disketu. Nemám nič proti, ale naboťovať môžeme kedykoľvek z CD, teda túto možnosť preskočíme (skip) (obr. 17). Ak na inkriminovanom počítači kedykoľvek z CD naboťovať nemôžeme, odporúčam disketu vytvoriť (alebo dve, ako odporúča inštalátor) :-).

Ak máme modem, zaujímať nás bude nasledujúci krok (obr. 18). Tu si totiž vyberáme link `/dev/modem`, ktorý bude ukazovať na skutočné zariadenie modemu. V prípade absencie modemu, prípadne ak máme „winmodem“, vyberieme prvú možnosť a modem sa vynasnažíme rozbehať neskôr. To je aj môj prípad.





Obr. 17

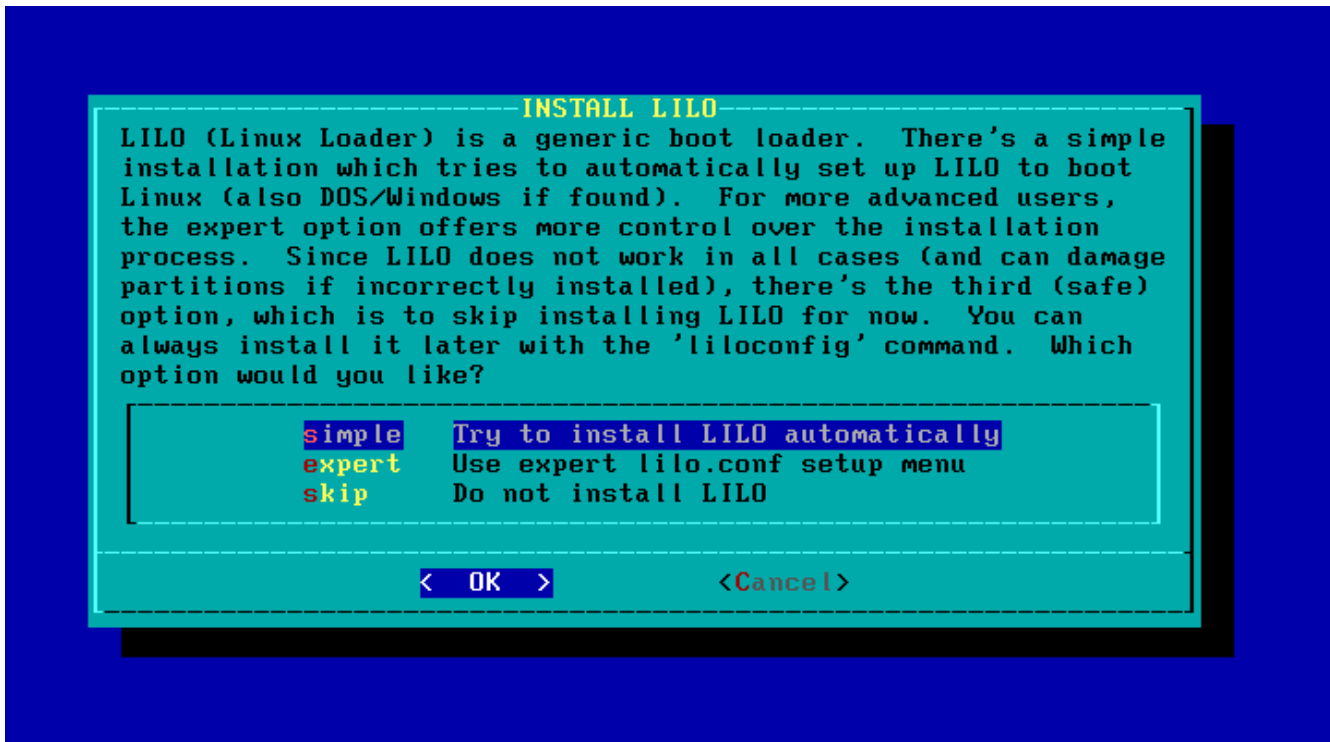


Obr. 18

Nasleduje možnosť vypnúť hotplug, ktorý umožňuje používanie zariadení pripojených za behu (USB kľúč, fotoaparát, tlačiareň...) bez potreby reštartovať počítač. Vypnúť ju má zmysel len vtedy, keď spôsobuje problémy.

V ďalšom menu (obr.19) máme možnosť nainštalovať LILO (LIInux LOader), vďaka ktorému nám Slack nabojuje. Priznám sa, že som LILO inštaloval ručne, pretože neviem, čo by s mojimi dvoma diskami urobila automatická inštalácia (viac o LILO sa môžeme dočítať v tomto skvelom článku [6]). Teoreticky by mala zafungovať možnosť simple, ale za nič neručím.

V nastaveniach myši (obr. 20) si vyberieme správny protokol myši (u mňa je to PS/2 – ps2), v ďalšom okne odklepeme spúšťanie programu gpm (ktorý umožňuje copy&paste myšou aj v textovej konzole).



Obr. 19



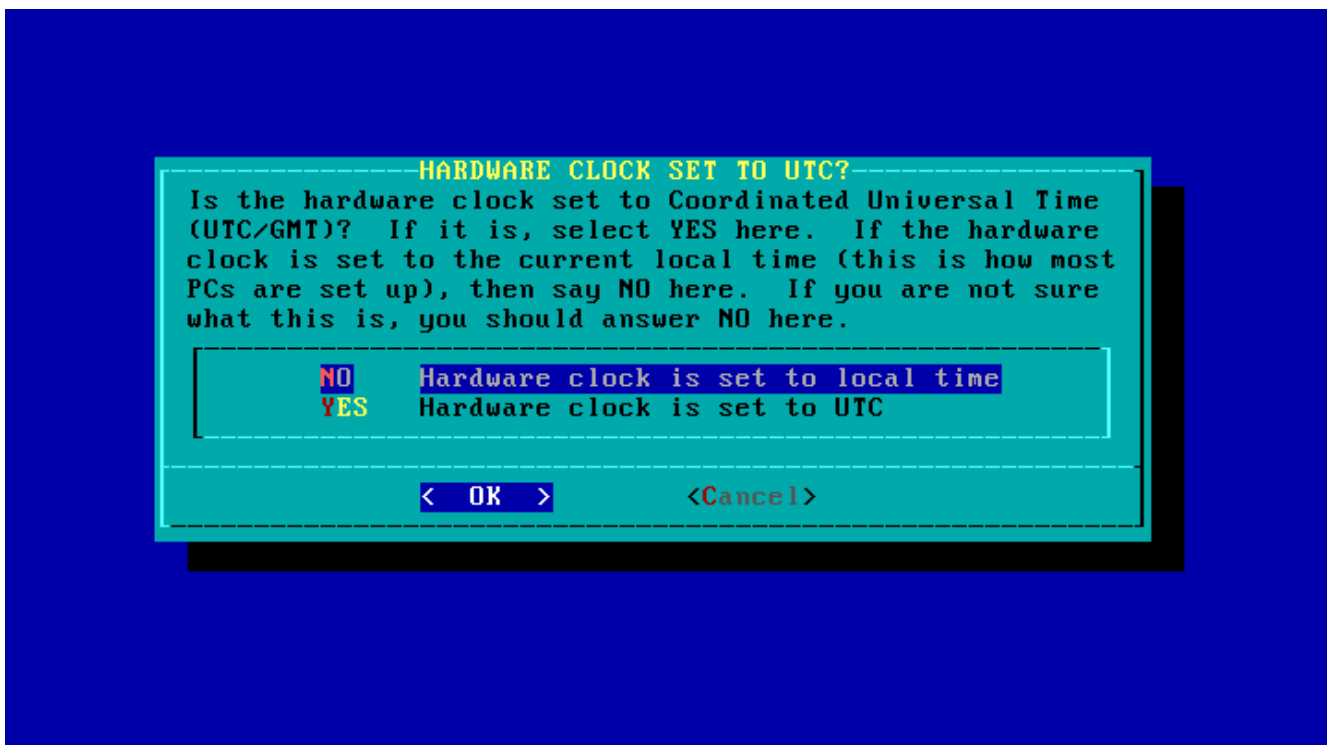
Obr. 20

Nastavenia siete bohužiaľ preskočíme (hlavne preto, lebo žiadnu sieť doma nemám). V ďalšom okne si vyberieme služby, ktoré sa budú spúšťať pri štarte systému (obr. 21). Čo potrebujeme, pozapínáme, ak si nie sme istí, môžeme nechať zapnuté výchozí hodnoty – spúšťanie služieb sa dá jednoducho zmeniť neskôr.

Po výbere služieb si môžeme vybrať konzolový font, aký len chceme (alebo tento krok preskočiť). Môžeme si taktiež nastaviť, či sú hardwarové hodiny nastavené na UTC (GMT) alebo na lokálny čas (obr. 22). Ak su nastavené na UTC, viete o tom. Ak si nie ste istí, vyberiete teda NO (local time). Rovnako aj vtedy, ak chcete, aby sa o počítač Slackware delil s nejakým systémom, ktorý nevie akceptovať nastavenie hardwarových hodín na UTC (áno, mám na mysli systémy z Redmondu). Potom vyberieme časovú zónu (Europe/Bratislava, Europe/Prague).



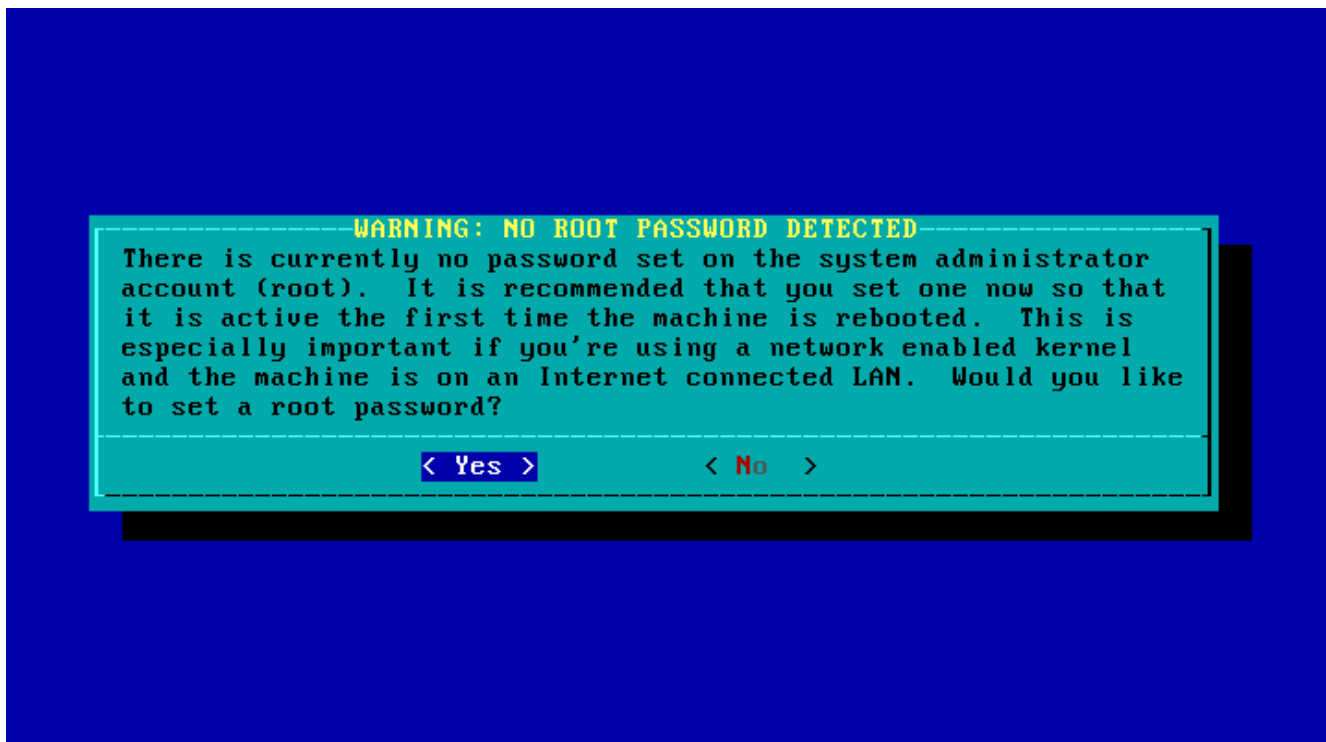
Obr. 21



Obr. 22

Vyberieme si aj, ktorý okenný manager sa má defaultne spôšťať po prihlásení užívateľa. Ak si neviete vybrať, pravdepodobne budete chcieť KDE alebo GNOME, ktoré sú najprepracovanejšie (ale aj najväčšie a najpomalšie („bloatware“)).

Inštalátor sa nás na záver spýta, či chceme nastaviť heslo roota, čo je *veľmi* dobrý nápad (obr. 23).



Obr. 23

Týmto je ukončená aj poinštaláčna konfigurácia, počítač môžeme reštartovať, a ak sme nič nepokazili, po reboote nás privíta zbrusu nový systém ready and willing. Zatiaľ len v textovom režime, a bez určitej základnej konfigurácie takmer nepoužiteľný.

Po prvom naboovaní nás privíta textový login. Prihlásime sa ako root (keďže iný užívateľ ešte neexistuje, je to celkom logické :-)) a keďže v textovom móde pravdepodobne nebudeme chcieť pracovať natrvalo, našou prvou zastávkou bude súbor `/etc/inittab`. Spustíme svoj najobľúbenejší textový editor a budeme editovať: v 24. riadku je východzí runlevel, ktorý asi budeme chcieť z 3 (textový login) zmeniť na 4 (grafický login – toto nie je preklep, v tomto sa Slackware líši od niektorých iných distribúcií (pokiaľ viem, aspoň Mandrake)). Po tejto zmene nás systém po najbližšom boote privíta grafickým loginom (gdm, kdm, xdm, podľa toho, čo sme nainštalovali (KDE, GNOME)). To, ktorý login manager chceme používať (štandardne sú v uvedenom poradí a spustí sa prvý nainštalovaný), môžeme zmeniť na konci súboru `/etc/rc.d/rc.4`.

Ak chceme aj v grafickom režime využívať viac ako jednu textový konzolu, mali by sme si všimnúť riadky 50 a viac. Za prvú dvojbodku pri konzole, ktorú chceme mať k dispozícii, pridáme číslo 4 (ja mám pri všetkých šiestich 12345).

Keď už sme v konzole (a v budúcnosti budeme prevažne pracovať v X), môžeme si ju odrazu aj poslovenčiť. Toto dosiahneme tým, že načítame klávesovú mapu (aby sme mohli písať po slovensky my) a fonty (aby nám konzola pekne po slovensky aj odpovedala). Toto dosiahneme tým, že spustíme príkazy:

```
loadkeys sk-qwerty
setfont lat2a-16
```

Aby nám konzoly hovorili po slovensky vždy, vložíme tieto príkazy do súborov `/etc/rc.d/rc.font` a `/etc/rc.d/rc.keymap` (ten prvý premenujeme z `rc.font.sample` a ten druhý musíme vytvoriť, rovnako mu aj prideliť práva na spúšťanie). Tieto súbory sa spúšťajú zo súboru `/etc/rc.d/rc.M` pri prechode systému do viacuzivateľského režimu. Pre podrobnejšie informácie o lokalizácii odporúčam prečítať [Slovak-HOWTO](#) [7], prípadne [Czech-HOWTO](#) [8]; ak sme nainštalovali dokumentáciu, tak aj `/usr/doc/Linux-HOWTOs/Slovak-HOWTO`.

Okrem nastavenia fontov a klávesovej mapy je dôležité aj nastavenie locales, ktoré nastavíme v súbore `/etc/profile.d/lang.sh` (ak používame `sh/bash` (používame `-`))). Ja osobne tam mám nastavenia:

```
export LANG=C
export LC_CTYPE=sk_SK
export LANGUAGE=C
export LC_NAME=sk_SK
export LC_ADDRESS=sk_SK
export LC_PAPER=sk_SK
```

Toto nastavenie zabezpečuje „rozumnú“ mieru poslovenčenia (slovenčina funguje, ale iba vtedy, keď chcem ja. Nemám napríklad rád poslovenčené chybové hlásenia). Pre podrobnosti opäť viď [Slovak/Czech-HOWTO](#).

Našou ďalšou zastávkou bude súbor `/etc/rc.d/rc.modules`, kde v 46. riadku odstránime znak `#` zo začiatku riadku (`sbin/modprobe apm`), ináč by nám nefungoval Advanced Power Management (čo znamená, že po vypnutí by sa ukončila činnosť systému, no ATX zdroj by sa nevypol).

Mali by sme si aj nastaviť X Window System, aby nám fungoval podľa očakávaní. Nebudem sa zaoberať podrobnou konfiguráciou (podrobný seriál práve vyšiel [tu na ABC](#) [9]). Môžeme použiť utilitu `xf86config` alebo `xf86cfg` (grafická verzia). `xf86config` sa nás bude interaktívne pýtať na naše požiadavky a podľa toho vygeneruje súbor `/etc/X11/XF86Config` (preto je vhodné poznať typ grafickej karty a frekvencie monitora (horizontálnu a vertikálnu – sú v manuáli)).

Ani sme sa nenazdali a prišiel čas na to, aby sme si vytvorili užívateľské konto. Všade sa to píše, no neodpustím si, aby som to nezdôraznil: účet roota sa používa na administráciu systému. Len na administráciu systému. Teda ak nemusíme, nepoužívame ho. Na vytvorenie obyčajného užívateľského účtu slúži príkaz `adduser`, ktorý sa nás interaktívne pýta na informácie o pridávanom užívateľovi. Po tomto kroku sa už môžeme prihlásiť do grafického prostredia ako užívateľ. Môžeme teraz počítač reštartovať (aby sa načítali latin2 fonty do všetkých konzolí apod.), alebo ako root spustíme príkaz `telinit 4`, ktorý prepne beh systému na úroveň 4.

Ak chceme hneď po nainštalovaní pracovať s nejakým 3D programom (prípadne sa zahrať nejakú 3D hru :-)), iste nás bude zaujímať spojzndenie akcelerovanej 3D grafiky. Vo svojom obľúbenom editore otvoríme súbor `/etc/X11/XF86Config`, v riadkoch 59 a 61 odkomentujeme načítanie modulov `glx` a `dri`. Aby 3D grafiku mohli využívať aj iní užívatelia ako root, na konci súboru odkomentujeme všetky tri riadky `Section "DRI"`.

Ak chceme mať „akcelerovanú“ aj myš (teda pri pomalom pohybe bude citlivejšia ako pri rýchlejších švihoch), a náš obľúbený window manager to nepodporuje (túto vlastnosť majú asi len KDE a GNOME), je pre nás vhodný program `xset` s parametrom `m`. Tento príkaz (ja používam `xset m 5 3`) vložíme do súboru `/etc/X11/xinit/xinitrc.nas-window-manager` pred spustením samotného window managera.

Ak chceme, aby počítač mohli vypínať užívatelia, ktorí nie sú root (čo je na počítačoch okrem serverov dosť praktická záležitosť), mali by sme zeditovať súbor `/etc/sudoers`. Tento súbor by sme mali editovať príkazom `visudo` (v praxi editor `vi`, ktorý zabezpečí, aby súbor needitovalo

viac rootov naraz a skontroluje jeho syntax (ináč by príkaz `sudo` odmietol fungovať – bavíme sa o vážnych bezpečnostných záležitostiach)). Ja mám v tomto súbore riadok

```
ALL    beast = NOPASSWD: /sbin/shutdown
```

To zabezpečí, že ktokoľvek môže napísať `sudo /sbin/shutdown -h now` a počítač sa bez komplikácii vypne. Štandardne by si vypýtal užívateľovo heslo, no takto to mám nastavené preto, aby som to mohol pridať do menu window manažera.

Tieto veci by na poinštaláciu konfiguráciu stačili. Viem, že to ani zďaleka nie je vyčerpávajúci popis možných nastavení, no snažil som sa zhrnúť nastavenia, ktoré bude chcieť spraviť takmer každý správca systému. Ďalšie nastavenia už veľmi závisia od konkrétnych požiadaviek a často aj od používaného window manažera (a keďže používam IceWM, ktorý ani nie je súčasťou distribúcie, moc do toho už nevidím). Písal som hlavne vlastné skúsenosti, ktoré nebolo vždy jednoduché získať a ktoré sa určite hodia. Uvítam postrehy čitateľov v diskusii (rád sa nechám poučiť :-)).

V priebehu písania tohto seriálu už vyšiel aj nový Slackware 10, no myslím, že verzia 9.1 hneď tak zo sveta nezíde. Dve veci ma dlho trápili, kým som na ne prišiel, a chcem sa s nimi podeliť.

**RPM** – `rpm` je program, ktorý umožňuje do Slacka doinštalovať aj `.rpm` balíky určené pre Red Hat/Mandrake. Veľmi užitočný program, no verzia v distribúcii je chybná. Aby sme túto užitočnú vlastnosť mohli využívať, musíme stiahnuť novú verziu.

**Tlač** – pravdepodobne potrebujete z času na čas aj niečo vytlačiť, no a ja som s tým mal veľmi veľké problémy. Niekoľko týždňov som kvôli tlačii bootoval do Mandrake, ešteže som toho veľa netlačil. Dôvodom nefungujúcej tlače (chybová hláška o nemožnosti sa pripojiť na počítač) bol konflikt medzi dvoma tlačovými daemonmi: `CUPS` a `lprng`. Preto treba ten z nich, ktorý nechceme používať, odinštalovať.

## Ja chcem nový software!

Okrem pohodlného spôsobu, keď si odniekiaľ stiahneme `.tgz` balíček pre našu verziu Slackware (pravdepodobne [www.linuxpackages.net](http://www.linuxpackages.net) [10]), existuje aj ten druhý. Využijeme ho hlavne pri programoch, ktoré sa distribuujú len ako zdrojový kód, ale aj pri tých, ktoré balíčky poskytujú, no nie pre Slacka. Stiahneme zdrojový kód, skompilujeme (spravidla `./configure; make`) a nainštalujeme `make install`. Problémom pri tomto postupe je to, že niektoré programy už v makefile nemajú cieľ `make uninstall`, teda sa takto odinštalovať nedajú, prípadne `make uninstall` poznajú, no ktovie, čo sa vlastne kde maže (navyše si musíme ponechať minimálne daný makefile). Toto je teda bezpečná cesta k strateniu kontroly nad systémom, ktorá sa nám tak páčila. Vystáva tak potreba kontrolovať, čo sme do systému od inštalácie doinštalovali, a ktorý súbor ku ktorému programu patrí. Na tento úkol je určený program `checkinstall`, ktorý nájdeme na druhom inštaláčnom CD, prípadne [na tejto adrese](#) [11].

Program sa používa tak, že namiesto normálneho nainštalovania (napr. `make install` – alebo hocijaký iný príkaz) – spustíme `checkinstall` s parametrom príkazu, ak to nie je práve `make install`. Program sa nás opýta na niekoľko informácií: ako má inštalovať dokumentáciu, či má z inštalácie vylúčiť niektoré súbory a podobne. Ak chceme nainštalovať dokumentačné súbory k programu, nakopírujeme ich do adresára `doc-pak` v adresári rozbaleného programu. Ak sa tu taký adresár nenachádza, program automaticky vyberie „štandardné“ dokumentačné súbory (README, INSTALL a podobne). Dokumentácia sa bude inštalovať do adresára `/usr/doc/nazov_programu-ver.zi.a`. Program sa nás opýta, aký balíček chceme vytvoriť – `.tgz` do Slackware, `.rpm` do Red Hat/Mandrake alebo `.deb` pre Debian.

Potom „akože“ vykoná inštaláciu – nainštaluje program do dočasného adresára a až potom daný program nainštaluje ako balíček, ktorý medzitým vytvorí. Takto sa nestratíme v nainštalovaných programoch, a navyše máme aj balíček priamo pre náš počítač. Toto je vhodné, ak sa staráme



o viacero rovnakých systémoch – na jednom program skompilujeme a nainštalujeme checkinstallom, na ostatných nám stačí nainštalovať novovytvorený balíček.

Ešte by som chcel upozorniť na jednu vec: neplette si `.tgz` balíčky pre Slackware s `.tgz` zázpí-ovanými tarballmi. Balíčky okrem skomprimovaných súborov obsahujú ešte aj skripty, ktoré sa vykonajú po nainštalovaní programom `installpkg` a podobne. A cesty k súborom sú v nich z pohľadu koreňového adresára.

## Nastavenie siete

Zatial som preskočil nastavenie siete, pretože by som písal o niečom, o čom som nemal páru. To sa zmenilo, v izbe mi pribudol jeden kus notebooku so sieťovým káblom, teda stačilo prikúpiť sieťovú kartu a môžeme sa do toho pustiť. Ešte by sme sa mohli pozastaviť pri tom, aké železo kúpiť (pokiaľ ešte žiadne nemáte). Pod [týmto seriálom](#) [12] sa rozprúdila dosť rozsiahla debata o tom, či je lepšie vsadiť na kvalitu, alebo kúpiť to najlacnejšie a dúfať, že to bude v poriadku. Ja som kúpil najlacnejšiu sieťovku v obchode, ktorá dokonca mala na krabici napísané, že má priložené aj ovládače pre Linux. Spokojný, že som za vodou, som ju doniesol domov a potešený som zistil, že má chipset realtek 8139C (pod Linuxom behá bezproblémovo). Konečné rozhodnutie je na vás, ak nepotrebuje extrémnu spoľahlivosť, myslím, že takéto riešenie stačí.

Po prepojení počítačov fyzickou vrstvou nám treba nakonfigurovať sieť na všetkých pripojených počítačoch (v mojom prípade na všetkých dvoch :-)). Toto v Slackovi môžeme robiť aj ručne, no je tu aj nástroj `netconfig` (ktorý by sa spustil aj pri inštalácii, keby sme nastavenie siete nepreskočili). Tento sa nás postupne spýta na nasledovné údaje: hostname (názov počítača), doménu, to, či budeme adresy získavať z DHCP servera, či budeme používať statické IP adresy "natvrdo", alebo počítač k žiadnej sieti pripojený nie je a nastaví iba loopback interface (pre komunikáciu počítača samého so sebou). Keďže DHCP server k dispozícii nemám, zvolíme možnosť „static IP“.

Potom zadáme údaje: IP adresa, netmask, gateway (brána – ak máme v sieti počítač, ktorý sprostredkúva pripojenie k Internetu alebo router. Ak nie, necháme prázdnu položku). Ak budeme pristupovať aj k Internetu, pri nasledovnej otázke zadáme, že k nameserveru pristupovať budeme. Do poskytnutej položky vyplníme IP adresu DNS servera (poskytne nám ju náš ISP). Teraz môžeme zadať len jeden DNS server, pre pridanie ďalších musíme editovať súbor `/etc/resolv.conf` ručne. Ak sme všetky údaje zadali správne, „Accept“ neme ich a dané údaje sa zapíšu do súborov `/etc/resolv.conf`, `/etc/HOSTNAME` a `/etc/rc.d/rc.inet1.conf`. V prípade, že počítač nechceme reštartovať, mali by sme nastaviť naše hostname, aby bolo nielen v súbore `/etc/HOSTNAME`. Teda spustíme príkaz `hostname meno_nášho_počítača`. Ak nepoužívame vlastný DNS server a chceme k počítačom pristupovať nielen cez IP adresy, bude nás zaujímať súbor `/etc/hosts`. V tomto sú názvy a aliasy pre jednotlivé IP adresy. Je vhodné mať tento súbor rovnaký na všetkých počítačoch v sieti :-). Ten môj vyzerá takto (bez komentárov):

```
127.0.0.1    localhost
192.168.0.1  beast.kennel beast
192.168.0.2  dragon.kennel dragon
```

Počítač, ktorý konfigurujeme, je 192.168.0.1 (beast), notebook je dragon (to chcem vidieť tie časy, keď takéto paskudy budú bývať v psej bude :-)). Po spustení skriptu `/etc/rc.d/inet1` (alebo reštarte počítača :-)) sa môžeme konečne dopingnúť na sieť. Pre podrobnosti o sieťach v Linuxe odporúčam [Net-HOWTO](#) [13] (alebo `/usr/doc/Linux-HOWTOs/Net-HOWTO`).

Aj keď toto je koniec tohto článku o inštalácii Slackware, je pravdepodobné, že to nebude môj posledný o Slackware. Tento systém sa mi páči tým viac, čím dlhšie ho používam. Rovnako sa mu stále viac dostávam pod kožu, a rád by som sa so svojimi skúsenosťami podelil aj s ostatnými. A to možno už aj o novom Slackware 10.

## Odkazy

- [1] <http://store.slackware.com>
- [2] <http://qwert.cz>
- [3] <http://www.slackware.com/getslack/>
- [4] <ftp://ftp.slackware.at/slackware-9.1-iso/>
- [5] <http://www.abclinuxu.cz/clanky/show/20310#fs>
- [6] <http://www.abclinuxu.cz/clanky/show/36780>
- [7] <http://www.tldp.org/HOWTO/Slovak-HOWTO/>
- [8] <http://www.penguin.cz/czech-howto/>
- [9] <http://www.abclinuxu.cz/clanky/show/59104>
- [10] <http://www.linuxpackages.net>
- [11] <http://asic-linux.com.mx/~izto/checkinstall/>
- [12] <http://www.abclinuxu.cz/clanky/show/20310#sit>
- [13] <http://www.tldp.org/HOWTO/Net-HOWTO/>

\*\*\*



## Jaderné noviny 264–267

Robert Krátký (264–266), Leoš Literák (267)

---

*Aktualizace USB pro 2.6. Pravidla pro psaní IDE ovladačů. Nová kniha o virtuální paměti v Linuxu. UMSDOS bude možná vypuštěn z 2.6. SquashFS verze 2.0. Linus navrhuje nová pravidla označování původu záplat. Ovladač VIA Velocity Gigabit Ethernet. Jak funguje geometrie disků v 2.4 a 2.6. Správcovství SysFS. Bezpečnostní 'NX' funkce v 2.6. Zdokonalení obsluhy tlačítek myši v mousedev. Podpora binárek SCO v Linuxu. Nový nástroj pro kopírování paměti, kcopyd. Stav JFFS2. Konfigurační nápověda pro NAPI. Další pokus o kompilační parametr pro Debian. Aktualizace NTFS obsahuje přepisování rezidentních souborů. Časovač s vysokou přesností (HRT). Podpora RSA Crypto API v řadě 2.6. Linux 2.6.7. Kdy používat GCC Built-Ins.*

---

### Aktualizace USB pro 2.6, 11 e-mailů

14. kvě–19. kvě

Greg KH napsal:

Tady jsou USB patche pro 2.6.62. Obsahují několik různých věcí:

- opravy přístupu k rozhraní
- opravy chyb
- pár nových ovladačů
- další spoustu opravených chyb

Všechny ty věci (s výjimkou několika malých dnešních patchů) už byly docela dlouho v -mm. Stahujte z: [bk://kernel.bkbits.net/gregkh/linux/usb-2.6](http://bk://kernel.bkbits.net/gregkh/linux/usb-2.6) Pro ty, kdo je chtějí vidět, se patche objeví ještě v linux-usb-devel.

Olaf Hering měl problémy s kompilací kódu pro aktuální oficiální strom a Linus Torvalds poradil: Nahraď všechna „led“ slovem „cytherm“. Ten kód stál stejně za houby a nikdy by nešel zkompilovat se zapnutým debugováním. Greg odpověděl, že pošle nový patch, ve kterém to bude opraveno.

### Pravidla pro psaní IDE ovladačů, 10 e-mailů

15. kvě–16. kvě

Bartłomiej Zolnierkiewicz dal dohromady nějaké pokyny pro psaní IDE ovladačů:

obecná pravidla:

- kód mimo adresář drivers/ide by neměl mít zapotřebí linkovat <linux/ide.h> – pokud to dělá, něco je špatně
- nevěřte rozšířenému omylu, že kód ovladačů může mít nižší kvalitu než kód hlavního kernelu
- nekopírujte bez rozmyslu ošklivý nebo pochybný kód (v IDE toho je stále ještě dost)
- IDE plně přechází model host ovladačů, takže pro své rozhraní napište host ovladač
- host ovladače by měly samostatně vyžadovat/uvolňovat I/O zdroje a nastavovat hwif→mmio na 2
- pamatujte na řazení [ordering] (pokazíte řazení a některé stroje vůbec nenabootují)
- používejte konferenci [linux-ide@vger.kernel.org](mailto:linux-ide@vger.kernel.org)

nové architektury:

- do `<asm/ide.h>` přidávejte jen to opravdu nejdůležitější a kód rozhraní dejte do `drivers/ide`
- `ide_init_hwif_ports()` je zastaralé a pomalu mizí, definujte `IDE_ARCH_NO_OBSOLETE_INIT` v `<asm/ide.h>`
- definujte `ide_default_irq()`, `ide_init_default_irq()` a `ide_default_io_base()` na (0)
- `ide_init_default_hwifs()` je pryč

ovladače pro specifické architektury:

- nezneužívejte `ide_default_irq()/ide_default_io_base()` z `<asm/ide.h>`
- váš ovladač by měl být v `drivers/ide/<vaše_architektura>/<název_vašeho_ovladače>`
- jako příklad mohou posloužit ovladače v `drivers/ide/arm` a `drivers/ide/legacy` (hlavně ovladače pro m68k a arm)
- nepoužívejte `ide_setup_ports()`

PCI ovladače:

- není potřeba ovladač rozdělovat na soubory `.c` a `.h`
- rozhraní `/proc/ide/` jsou zastaralá

Jeff Garzik připojil pár svých postřehů.

Především měl pocit, že popsané řešení situace s `ide_init_hwif_ports()` je krokem zpět. Argumentoval tím, že vzhledem k tomu, že jde o zastaralou funkci, bylo by lepší ji dát k dispozici pouze ovladačům, které definovaly `IDE_ARCH_OBSOLETE_INIT` (místo definování `IDE_ARCH_NO_OBSOLETE_INIT` pro nové ovladače, jak to říkal Bartłomiej). Tak by nové ovladače nemusely mít vlastně nesmyslnou definici jen proto, že některé starší ovladače dělaly věci jinak. Bartłomiej slíbil, že popis změní podle toho, jak to říkal Jeff.

## Nová kniha o virtuální paměti v Linuxu, 3 e-maily

15. kvě–18. kvě

Mel Gorman napsal:

Před dlouhou dobou jsem několika lidem řekl, že jsem byl požádán o napsání knihy založené na dokumentaci k virtuální paměti Linuxu, která je na <http://www.csn.ul.ie/~mel/projects/vm> [1]. Jsem rád, že vám mohu oznámit, že ta kniha už je hotová a k dostání v online obchodech (<http://www.phptr.com/title/0131453483> [2]). Ano, přihřívám si polívčičku. Je vydaná s licencí OPL podle podmínek Bruce Perens Open Book Series (<http://www.perens.com/books> [3]), což znamená, že bude k dispozici ke stažení po 90 dnech. Jinými slovy, chtěl jsem, aby byla snadno přístupná tak jako ta online dokumentace, ale máte možnost pořídit si i hezké tištěné vydání :).

Informace na té stránce tam zůstanou pro ty, kdo si je chtějí přečíst, ale v knize je toho mnohem více, včetně správy TLB, systému souborů sdílené paměti a o hodně více komentovaného kódu a dalšího extra materiálu. Každá kapitola také obsahuje informace o kernelu 2.6 - s platností kolem 2.6.0-test4, což by mělo každému pomoci pochopit i pozdější kód.

Bavilo mě na té knize pracovat a doufám, že konečný výsledek bude pro lidi užitečný.

Marc-Christian Petersen vyjádřil pocity mnoha, když odpověděl: Těžko napsat, jak moc tvé úsilí oceňuji. Díky!

## UMSDOS bude možná vypuštěn z 2.6, 4 e-maily

19. kvě–27. kvě

Adrian Bunk se zeptal, jestli někdo používá nebo vyvíjí UMSDOS, nebo jestli by to mohlo být z 2.6 jádra odstraněno. Jan-Benedict Glaw řekl, že by bylo fajn ponechat to v kernelu z historických

důvodů, coby jeden z těch starších způsobů jak ukázat Linux uživatelům DOSu. Ale uznal: **Totěž je však možné docílit (i když to dá mnohem více práce) umístěním ext2 filesystemu připojitelného pomocí loop a natažením z initrd. O dost komplikovanější, ne tak flexibilní (soubory připojené přes loop většinou nezvětšují velikost :).**

Mark Beyer poukázal na to, že některé embedded systémy to stále používají, takže by možná bylo dobré ponechat to v jádře kvůli zpětné kompatibilitě; ale Adrian odpověděl, že ve skutečnosti je to už teď nefunkční, a že pokud se to někdo nerozhodne spravovat, bude právě tohle ten rozhodující důvod, proč to odstranit.

## SquashFS verze 2.0, 1 e-mail

21. kvě

Phillip Lougher napsal:

Je mi potěšením oznámit první verzi Squashfs 2.0. Pod kapotou souborového systému Squashfs proběhlo mnoho změn, které vylepšují komprimaci. Squashfs 2.0 přináší koncept fragmentových bloků a velikost bloků je zvýšena na 64K. Tím se dosáhne 5–20 % úspory při komprimaci a umožňuje to Squashfs dosáhnout lepší komprimace než Cloop, ale při zachování efektivity I/O komprimovaného filesystemu. Kromě toho bylo navýšeno maximální množství UID a GID na 256. Squashfs proto lépe podporuje live CD.

Popis fragmentových bloků a dalších změn naleznete na stránkách projektu:

<http://squashfs.sourceforge.net> [4].

## Linus navrhuje nová pravidla označování původu záplat

22. kvě–27. kvě

Linus Torvalds obvykle diskuze sám nezačíná (nejedná-li se o oznámení nové verze jádra); tentokrát začal takto:

Hola!

Následující je podnět k diskuzi...

Někteří z vás již možná slyšeli o té bláznivé firmě, která se jmenuje SCO (neboli „Smoking Crack Organization“ /Společnost kuřáků cracku/), a které je zatěžko uvěřit, že open source pracuje lépe než jejich pět inženýrů. Jak už to vypadá, tak tvrdí nesmysly o tom, odkud pochází náš zdrojový kód, a dokonce říkají, že jsou majiteli kódu, který jsem zjevně napsal před více než desetiletím.

Lidi jim ta tvrzení celkem bez problému (vítěz soutěže o nejzdrženlivější vyjádření roku) vyvraceli, ale je pravdou, že něco z toho vyvracení zahrnovalo prohledávání archivů konferencí až do roku 1992, atd. Žádná velká zábava.

Například v případě souboru „ctype.h“ byla původnost práce prokázána díky příšerným chybám, které tehdy obsahoval. A protože teď už chyby neděláme (že ano?), tak bychom pravděpodobně měli mít i jiné způsoby dokumentace původu kódu.

Takže, abychom předešli podobným problémům za dalších deset let, navrhuji věnovat se více důsledné dokumentaci ne pouze původu patche (což už teď dokumentujeme dost dobře v changelogu), ale i cesty, kterou prošel. Proč celou cestu a ne pouze původce?

V současnosti není většina patchů odeslána přímo mně. To by se nejenom prostě nezvládalo, ale je i mnoho subsystémů, o kterých já nemám ponětí, a proto ani nemohu posoudit, nakolik je takový patch dobrý. Takže já většinou přijdu do styku se správcem těchto subsystémů a objeví-li se chyba, chci vidět jméno správce, ne nějakého náhodného vývojáře, o kterém ani nevím, jestli je i nadále aktivní nebo ne. Takže alespoň pro mě je ten řetěz vlastně více důležitý než skutečný původce.

A je tu ještě jedna věc. Konkrétně to, že když mi (nebo komukoliv jinému) někdo pošle patch, jediné co jasně vidím jsou informace o odesílateli – a to je ta část, které důvěřuji. Když mi pošle

patch Andrew, důvěřuji tomu, protože to přichází od něj – i kdyby původním autorem byl někdo, koho neznám. Takže cesta, kterou patch prošel, vlastně dokumentuje řetěz důvěry – všichni většinou známe ten „další krok“, ale nemusíme už znát celý ten řetěz.

Takže já navrhuji, abychom začali patche „podpisovat“ [signing off], aby byla zřejmá cesta, kterou se někam patch dostal, a aby byl dokumentován řetěz důvěry. Zároveň to prostředníkům umožní editovat patch, aniž by se někde „vytratilo“ jejich jméno – docela často nebývá patch, který nakonec dorazí do jádra, stejný jako ten původní, protože prošel několika vrstvami lidí.

Rád bych to ponechal velmi jednoduché a stručné a také by to mělo sedět s tím, jak si už teď patche předáváme. Podpis by se pouze přidal na konec popisu patche. Byla by to jediná řádka na konci (případně za podpisy jiných lidí). Vypadala by takto:

Signed-off-by: Nějaký Vývojář <nejaky@vyvojar.org>

Linus dále představil i náčrt krátkého textu, jehož akceptováním by vývojář potvrdil, že patch buď pochází přímo od něj a nebo o daném patchi ví, že je šířen pod příslušnou open source licenci, která mu dovoluje jej šířit dále.

V diskusi několik lidí představilo svoji představu o tom, jakým způsobem by šel navrhovaný systém vylepšit nebo vyšperkovat (např. pomocí používání PGP), ale Linus trval na tom, že nejde o faktickou ochranu autentičnosti, nýbrž o snahu dokumentovat celý proces – a proto je nejvýhodnější udržet systém co nejprostší.

Během několika dní se vyskytli i první vývojáři, kteří ve svých patchích nový systém podpisování použili.

## Ovladač VIA Velocity Gigabit Ethernet, 3 e-maily

26. kvě

Alan Cox napsal: Pročištěný ovladač VIA velocity portovaný na 2.6 najdete na <ftp://people.redhat.com/alan/Kernel/> [5].

Je to úplně první port na 2.6. V žádném případě ještě není pěkně uhlazený. Výsledky testování a patche mi prosím pošlejte (v současné době tuto konferenci nečtu). Mělo by to fungovat jak na 32, tak na 64 bitových little endian strojích. Na big endian to ještě fungovat nebude.

## Jak funguje geometrie disků v 2.4 a 2.6, 33 e-mailů

30. kvě–3. črv

Jeff Garzik napsal:

Vypadá to, že kód starající se ve 2.6 o geometrii znemožňuje duální bootování, protože Windows požadují „rozumné“ hodnoty CHS. Viz slashdot a <http://www.redhat.com/archives/fedora-devel-list/2004-May/msg00908.html> [6]. Ačkoliv se to teď svádí na Fedora Core, řekl bych, že je to spíše problém 2.6.x kernelu. Nepokusil se někdo dohledat csety, které způsobují takhle vážné narušení tabulky oddílů? Pokud ano, ušetřilo by mi to čas, kdy bych se po tom musel pít.

Andries Brouwer řekl, že ta diskuze na slashdotu se týká pouze problémů v uživatelském prostoru, a že fdisk, kterého je Andries správcem, funguje bezvadně. K tomu dodal: O geometrii disků a o situaci v 2.4 i 2.6 ti mohu povědět do detailů.

Jeff o ty detaily požádal a Andries odpověděl:

O těchto věcech už jsem napsal několik mnohostránkových textů. Viz např.:

<http://www.win.tue.nl/~aeb/linux/Large-Disk.html> [7]. Pokus o krátké shrnutí:

1. Hardware: V dávných dobách měly disky (MFM nebo RLL) geometrii popisující, kolik sektorů na stopu, kolik stop (cylindrů) a kolik hlav ta věc má.  
S příchodem IDE už neplatilo, že by disk měl pevně stanovenou geometrii: ATA příkaz INITIALIZE DRIVE PARAMETERS disku dnes řekne, jakou geometrii by měl mít.

Na IDE disk může být přístupováno buď v CHS nebo LBA režimu a geometrie určená nebo načtená pomocí ATA příkazu IDENTIFY definuje význam příkazu CHS. U LBA přístupu nehraje geometrie žádnou roli. SCSI disky nikdy žádnou praštěnou geometrii neměly. Linux používá LBA a nemusí se o geometrii starat.

2. DOS / Windows: Diskové rozhraní v DOSu používalo CHS. Používá-li disk LBA, ovladač musí CHS konvertovat na LBA, a proto potřebuje geometrii (alespoň H, S). Protože DOS (BIOS INT 13) měl 10 bitů pro C a 6 bitů pro S, zatímco ATA používá 4 bity pro H, mohlo takové schéma adresovat pouze o něco méně než  $2^{20}$  sektorů. Limit 528 MB.

Byly vymyšleny nejrůznější druhy převodních schémat proto, aby daly rozhraní BIOSu jinou geometrii než jakou používalo ATA rozhraní. Uživatel si převodní schéma zvolí v nastavení BIOSu. Geometrie už není svázána s diskem, ale BIOS ji zná. Existuje množství volání BIOSu, která hlásí různé verze geometrie.

3. Tabulky oddílů:

DOSová tabulka oddílů (viz např.: [http://www.win.tue.nl/~aeb/partitions/partition\\_tables.html](http://www.win.tue.nl/~aeb/partitions/partition_tables.html) [8]) udává umístění oddílů dvěma způsoby. Jeden způsob je ten, který používá Linux (počáteční sektor a délka, obojí ve 32 bitech). Ten druhý používá DOS a udává první a poslední sektor (obojí ve 24 bitech).

I když geometrie už nikoho nezajímá, stále je nutné tuto CHS políčka vyplnit. Pokud je na tom oddílu souborový systém FAT, boot sektor toho FAT filesystému také udává informace o velikosti oddílu.

4. Linux: Existuje ioctl HDIO\_GETGEO, které vrací geometrii disku a počáteční sektor (offset) oddílu. Dále ioctl BLKGETSIZE, které vrací velikost (v 512-bajtových sektorech) blokového zařízení v 32 bitech. A také ioctl BLKGETSIZE64, které vrací velikost (v bajtech) blokového zařízení v 64 bitech.

BLKGETSIZE je zjevně zastaralé – mělo by být všude nahrazeno BLKGETSIZE64.  $2^{41}$  B jsou 2 TB a některé RAIDy jsou větší.

HDIO\_GETGEO ioctl udává hlavy, sektory a cylindry – pole o 1, 1, 2 bajtech. Na jednu stranu je to rozumné – neexistuje žádné rozhraní, které by mohlo pro počet cylindrů používat více než 16 bitů. Na druhou stranu však stále existuje vadný software, který počítá velikost disku jako C\*H\*S a C, H, S získává pomocí HDIO\_GETGEO. Tak je to pochopitelně špatně a nepomůže zavedení nového ioctl – takový software musí být opraven, aby používal BLKGETSIZE64.

5. Používání geometrie pod Linuxem:

Zjednodušeně řečeno byla geometrie v Linuxu potřeba ze dvou důvodů: LILO (a podobné bootloadery) a `*fdisk`. Samozřejmě také programy, které se snaží emulovat aspekty DOSu/Windows, se zajímají o geometrii.

Důvod „LILO“ už je pryč – s parametry jako `linear` nebo `lba32` používá lilo při instalaci lineární čísla sektorů a na CHS je v případě potřeby převádí při bootu.

Důvod „`fdisk`“ už je také pryč – když se to všechno kolem geometrie tak hrozně zkomplikovalo, nikdo se už nesnažil geometrii rozumět. Místo toho se pouze odhadlo z tabulky oddílů, jakou geometrii používal program, který ji naposledy zapsal.

Jak geometrii zjistil kernel? Především následujícími třemi způsoby: (i) z bootovacích parametrů, (ii) z tabulky oddílů, (iii) dotazem na BIOS při bootu před přepnutím do chráněného režimu.

Informace (i) – explicitní specifikace – a (ii) – tabulka oddílů – jsou dostupné i z uživatelského prostředí, není potřeba kernel. Zůstává otázka, jestli je (iii) dobrý nápad.

Nefungovalo to vždy, ale často ano. (A samozřejmě pouze na i386.) Příklady selhání: kód, který jsme měli, se ptal na informace pouze o prvních dvou discích. Často kladený dotaz zněl: Mám dva identické disky, jakto, že mají jinou geometrii? Ten kód také selhal, pokud byly



v systému SCSI disky. Navíc si ten kód myslel, že první dva disky v BIOSu jsou hda a hdb. Ale proto existuje vysvětlení. Například mnoho lidí ponechávalo své velké disky mimo BIOS, protože ten by kvůli velkým diskům spadl.

Takže tohle (iii) bylo vždy trochu neuspořádané a byly s tím problémy. Ale pro většinu lidí to fungovalo.

6. Současnost Linux už se nesnaží vynalézat geometrii. Pokud někdo geometrii potřebuje, je sám zodpovědný za zvolení jednoho z mnoha konceptů a za určení hodnot. Většina softwaru se obrací na tabulku oddílů, což funguje.

Parted možná ještě nebyl aktualizován, aby to tak dělal, a proto jsem usuzoval, že problémy Fedory mohou pocházet právě z používání parted.

Je to ztráta? Neřekl bych, ale existuje minimálně jedno využití staré situace, které dnes selže: instalace Windows z linuxového média na úplně prázdný disk.

## Správčovství SysFS, 2 e-maily

1. črv–4. črv

Christian Gmeiner se pokoušel kontaktovat správce SysFS, ale nepodařilo se mu najít žádné informace o tom, kdo by to mohl být. Greg KH odpověděl: Já jsem teď de-facto správce sysfs a hlavního kódu ovladačů. Asi bych to měl konečně přidat do souboru MAINTAINERS...

## Bezpečnostní 'NX' funkce v 2.6, 66 e-mailů

02. črv–08. črv

Ingo Molnar za společnost Red Hat napsal:

Rádi bychom oznámili následující patch pro kernel: [http://redhat.com/ mingo/nx-patches/nx-2.6.7-rc2-bk2-AE](http://redhat.com/mingo/nx-patches/nx-2.6.7-rc2-bk2-AE) [9]

Tento patch zavádí využití funkce 'NX' u x86 – poprvé se objevila u AMD64 a podpora byla oznámena i Intelem. (Další výrobci x86 CPU, Transmeta a VIA, rovněž oznámili, že budou funkci podporovat. Podporu ve Windows Microsoft oznámil na příští service pack.) Funkce NX se označuje také jako 'Enhanced Virus Protection' (vylepšená ochrana před viry). Patch zajišťuje pro Linux plnou podporu této funkce.

Co ten patch dělá? Formát tabulky stránek současných x86 CPU nemá bit 'execute' (spustit). To znamená, že i když aplikace namapuje oblast paměti bez PROT\_EXEC, CPU stejně umožní, aby byl v této paměti spuštěn kód. Tato vlastnost bývá často zneužívána exploity, když se jim podaří umístit nepřátelský kód do takové paměti – např. díky přetečení bufferu.

Funkce NX toto mění a přidává do PAE formátu tabulky stránek bit 'dont execute' (nespouštět). Ale protože je výchozí hodnota nula (kvůli kompatibilitě), jsou všechny stránky při výchozím nastavení spustitelné a je potřeba jádro naučit tento bit využívat.

Je-li funkce NX podporovaná procesorem, opatchované jádro NX zapne a zavede omezení spustitelnosti v uživatelském prostoru – např. no-exec stack, no-exec mmap a datové oblasti. To znamená menší šanci, že přetečení stacku nebo bufferu způsobí exploity.

Dále ten patch také implementuje 'NX ochranu' pro kód kernelu: je spustitelný pouze kód kernelu a moduly – takže i přetečení v prostoru kernelu jsou těžší (v některých případech nemožná). Takhle je zastaven kód kernelu, který se pokouší spustit mimo stack:

```
kernel tried to access NX-protected page -- exploit attempt? (uid: 500)
Unable to handle kernel paging request at virtual address f78d0f40
printing eip:
...
```

Patch je založen na prototypovém NX patchi napsaném v Intelu pro 2.4 – zvláštní poděkování Sureshi Siddhovi a Junu Nakajimovi z Intelu. Existující podporu pro NX v 64-bitových x86\_64 kernelech napsal Andi Kleen a tento patch je modelován podle jeho kódu.

Také Arjan van de Ven poskytl mnoho připomínek a byl to on, kdo patch integroval do jádra Fedora Core 2. Testovací RPM jsou k dispozici ke stažení zde:

<http://redhat.com/arjanv/2.6/RPMS.kernel/> [10]

RPM balíky kernel-2.6.6-1.411 už mají NX patch aplikovaný. Rychlý návod, jak zkompileovat vanilla kernel ze zdrojáků s NX patchem: <http://redhat.com/mingo/nx-patches/QuickStart-NX.txt> [11] Následovalo mnoho technických připomínek a komentářů od lidí jako Christoph Hellwig, Andi Kleen, Rusty Russell a Gerhard Mack. Linus Torvalds se zeptal, nakolik tato funkce koliduje s funkčností starších programů, ale Ingo i Arjan potvrdili, že nežádoucí účinky tohoto typu jsou minimální. Linus tedy navrhl, aby byla NX automaticky zapnuta a pouze se přidal bootovací parametr, kterým by ji bylo možno v případě potřeby vypnout.

## Zdokonalení obsluhy tlačítek myši v mousedev, 6 e-mailů

4. črv–6. črv

Dmitry Torokhov napsal:

Mousedev v současné době kombinuje všechna hardwarová data o pohybu, která dorazí od chvíle, kdy uživatelský prostor naposledy data načetl, do jediného PS/2 paketu. Problém je, že při těžkém nebo i středním zatížení začneme data ztrácet, protože uživatelský prostor je nestíhá dostatečně rychle číst. Projevuje se to takto:

- ignorované stisky tlačítek, protože ve chvíli, kdy se uživatelský prostor dostane k tomu, aby data přečetl, tlačítko už je uvolněno;
- kliknutí se projeví na špatném místě – když se uživatelský prostor dostane k tomu, aby paket načetl, myš už se pohnula o půl obrazovky jinam.

Následující patch ten problém napravuje – začne shromažďovat nový paket pokaždé, když je uživatelský prostor pozadu a postavení tlačítek se změní. Velikost bufferu je 16 paketů, tj. až 8 párů událostí stisknutí/puštění, což by mělo být více než dost.

Patch je oproti Vojtěchově stromu a měl by být aplikovatelný i na -mm. Také jsem udělal kumulativní patch pro mousedev oproti 2.6.7-pre2: [http://www.geocities.com/dt\\_or/input/misc/mousedev-2.6.7-rc2-cumulative.patch.gz](http://www.geocities.com/dt_or/input/misc/mousedev-2.6.7-rc2-cumulative.patch.gz) [12]. Vojtěch Pavlík byl rád, že to někdo udělal.

## Podpora binárek SCO v Linuxu, 5 e-mailů

6. črv–9. črv

Steve Bergman chtěl spustit starší SCO binárku a povšiml si, že projekt [linux-abi.sf.net](http://linux-abi.sf.net) [13] se zdá být mrtvý. Zeptal se, jestli existují jiné projekty, které by pracovaly na spouštění cizích binárek pod Linuxem. Později napsal: Vypadá to, že existuje patch, vydaný právě dnes (a právě pro mě, hádám) na: [http://sf.net/tracker/index.php?func=detail&aid=968070&group\\_id=13130&atid=313130](http://sf.net/tracker/index.php?func=detail&aid=968070&group_id=13130&atid=313130) [14].

Na jiném místě Mike Jagdis komentoval:

iBCS bylo ukončeno, když jsem usoudil, že „dost“ výrobců již Linux považuje za platformu číslo 1 a ten zbytek jsou staré proprietární věci, které buď fungují nebo je příliš obtížné je opravit. Z iBCS se pak stalo linux-abi, které bylo portováno na novější jádra a byla přidána podpora UnixWare. Nemyslím, že by tu ještě bylo dost SYSV věcí na to, aby mělo smysl v linux-abi pokračovat...

SCO řeklo už dávno, že s linux-abi nemají žádný problém. Což je docela zajímavé, jelikož iBCS začalo víceméně jako způsob pro Erica Youngdalea, jak otestovat svůj ELF loader kód, a postupně se dostalo až do hlavního jádra. (iBCS CVS je pořád dostupné na <http://sf.net/projects/ibcs> [15])



– i kdyby nic jiného. Není to úplně od začátku. Mám pocit, že můj první import proběhl v roce 1993...)

## Nový nástroj pro kopírování paměti, kcopyd, 10 e-mailů

2. črv–10. črv

Alasdair G. Kergon poslal patch implementující něco s názvem kcopyd, ale nepřipojil žádné vysvětlení. Když se několik lidí, včetně Andrew Mortona, zeptalo, oč se jedná, přiblížil: **Démon pro efektivní kopírování oblastí blokových zařízení. Při kopírování může být určeno více cílových oblastí. Démon je navržen tak, aby měl dobrý výkon s malými kousky i velkými kusy.**

Kevin Corry dodal, že démon má asynchronní upozornění na dokončení.

## Stav JFFS2, 14 e-mailů

4. črv–14. črv

Michal Semler chtěl používat JFFS2 a zeptal se, kde hledat patche pro 2.4 a 2.6; Daniel Egger odpověděl:

Pokud se nemýlím, tak JFFS2 je obsaženo ve standardních jádrech. Spíše bych ale doporučil používat CVS verzi z oficiálního repozitáře, protože jsou v ní mnohá vylepšení.

Aby to šlo používat na non-MTD (Memory technology devices, např. flash chipy s přímým zápisem přiletované na některých deskách) zařízeních, budeš potřebovat emulační vrstvu, tj. pseudo blokové MTD zařízení. A také bude potřeba nějaký další diskový oddíl s ext2/ext3/reiserfs/FAT obsahující kernel pro Grub/LILO bootloader.

David Woodhouse dodal:

JFFS2 v kernelu 2.4 je stará stabilní větev.

Kód v 2.6 a v CVS umožňuje daleko rychlejší připojení, obzvláště proto, že podporuje NAND flash. Linusův strom je průběžně aktualizován – když jsem dost spokojený se stabilitou vývojového stromu v CVS a mám čas to sloučit a pročíst, abych zkontroloval, jestli všechny změny dávají smysl – což často znamená, že je některé potřeba předělat. Budeš-li používat verzi z jádra, neměl bys mít problém – pokud ano, dej mi vědět.

Daniel poznamenal: Původní verze v jádře 2.4 má zásadní problém vedoucí k poškození souborového systému – alespoň při použití s blkmtd na CF. Proto používám 2.4 a CVS snapshot, ne proto, že by byl rychlejší.

David požádal o specifičtější popsání té chyby, ale Daniel odpověděl, že to bohužel poskytnout nemůže. Obecně však vysvětlil: Při bootu se objevovaly (a zase mizely) záhadné oopsy jádra na různých místech zdrojového kódu. Nejprve mne napadlo, že by mohla být špatná paměť, ale to nebylo ono. Potom jsem si myslel, že kontrola logu (té dlouhé verze) by mohla přechodně přehřát pasivně chlazený procesor, ale i to jsme mohli škrtnout, když se problém opakoval i ve velmi chladném prostředí. Po únavném upgrade na CVS verzi však všechno krásně funguje v režimu „nikdy se nedotýkej běžícího systému“.

## Konfigurační nápověda pro NAPI, 5 e-mailů

4. črv–10. črv

Adrian Bunk přidal nápovědní text ke konfiguračnímu parametru NAPI:

NAPI je nové ovladačové API navržené pro snížení zátěže procesoru a přerušení v situacích, kdy ovladač z karty přijímá hodně paketů. Stále jde o trochu experimentální záležitost, a proto není ve výchozím nastavení zapnuta.

Je-li vaše odhadovaná Rx zátěž 10kpps nebo více nebo bude-li karta nasazena v potenciálně nepřátelských sítích (např. na firewallu), pak zadejte Y.

Více informací v <file:Documentation/networking/NAPI\_HOWTO.txt>. Jste-li na pochybách, zadejte N.

Patch se Davidu S. Millerovi a Jeffu Garzikovi líbil a zařadili jej tedy do fronty patchů k 2.6 pro Linuse Torvaldse a Andrew Mortona.

## Další pokus o kompilační parametr pro Debian, 9 e-mailů

7. črv–14. črv

Tentokrát napsal Wichert Akkerman:

Původně jsem o tom mluvil ještě před vydáním 2.6.0, ale bylo mi řečeno, abych počkal, až se věci trochu stabilizují. Z mého pohledu se to už stalo, takže to navrhuji znovu.

kbuild má už nějakou dobu parametr pro make k vytvoření RPM. A protože koncept balíčků jádra je hezky pohodlný, přidal jsem v přiloženém patchi parametr také pro DEB.

Jelikož pořád ještě tak docela nerozumím Makefile kbuildu, jsou některé části docela nahrubo, ale Mně to funguje(tm).

David Vrabel se zeptal: Proč tohle a ne utilita make-kpkg, která je v debianním balíčku kernel-package?

A Wichert odpověděl:

Několik důvodů:

1. funguje to na non-Debian systémech, které používají dpkg
2. je to \*mnohem\* jednodušší a rychlejší než make-kpkg

Na jiném místě Flavio Stanchina poznamenal: Ten nápad se mi moc líbí, ale tvůj patch pro Makefile mění docela dost věcí i v parametru clean, který, pokud vím, nijak s parametrem deb nesouvisí. Neprovedl jsi náhodou diff ze staršího stromu?

Wichert se praštil do čela a potvrdil, že je to tak: omylem v patchi ponechal jinou, nesouvisející práci. Vzápětí poslal aktualizovaný patch. Sam Ravnborg Wichertovi napsal:

Právě pracuji na infrastruktuře, která by lépe podporovala sestavování různých balíčků. Registruji požadavky na .tar.gz, tar.gz2 a také deb.

Doufám, že ke konci týdne pošlu pár patchů. Tvůj skript pak do té sady patchů zařadím.

## Aktualizace NTFS obsahuje přepisování rezidentních souborů

8. črv–10. črv

Anton Altaparmakov oznámil: Další vydání NTFS. Tato verze je milníkem v tom smyslu, že konečně umožňuje přepisování rezidentních, tj. velmi malých, souborů. Jako bonus už děláme i všechen potřebný úklid systémových souborů NTFS, aby byla zajištěna integrita dat, takže již po odpojení není potřeba spouštět ntfsfix.

## Časovač s vysokou přesností (HRT), 33 e-mailů

Geoff Levand informoval konferenci o dostupnosti [16] patchů pro POSIXový časovač s vysokou přesností (high-res timer) pro jádro 2.6.6. Základem pro tyto patche je hrtimers-2.6.5-1.0.patch George Anzingera ze stránky <http://sourceforge.net/projects/high-res-timers/> [17].

William Lee Irwin III reagoval s tím, že žije v domnění, že patche George Anzingera již byly do řady 2.6 začleněny. Jiný autor otvrdil, že aplikační rozhraní v kernelu již je, nicméně patche umožňující získat rozlišení na úrovni deseti mikrosekund nebyly začleněny.

## Podpora RSA Crypto API v řadě 2.6, 12 e-mailů

Joy Latten se zeptal:

Pracuje někdo na implementaci RSA šifrování do cryptoapi kernelu? Pokud ne, uvažuji o vytvoření takového projektu...

Kartikey Bhatt odpověděl, že by se rád k takovému projektu připojil. Kristian Sorensen také napsal: Projekt Umbrella ([umbrella.sf.net](http://umbrella.sf.net) [18]) také potřebuje mít možnost používat kryptografii s veřejným klíčem uvnitř linuxového kernelu (i když pouze pro verifikaci podpisu). Plánujeme proto implementovat RSA nebo ElGamal (portací z GPG) do kernelu. Nicméně pokud tento projekt vznikne, bude nás velmi zajímat. Na podzim pak budeme schopni přispět nějakými hodinami programování :-).

Jinde se James Morris zeptal, zda se Joy a spol. snaží vytvořit obecné asymetrické kryptografické API. Sergey odpověděl:

Je samostatné generické asymetrické crypto API skutečně nutné?

Model užití cryptoapi je nastavit klíč před (de)šifrováním. Takže není důvod mít `public_key_alg` se samostatnými funkcemi `public_encrypt` a `private_encrypt`, protože toto rozlišení je implikováno během použití metody `setkey`. Proto plánujeme prostě přidat další `crypto_alg` pro `rsa_1024`.

Tom St. Denis odpověděl:

Jako základ `digest` modulu budeš potřebovat mít možnost:

1. importovat veřejný klíč
2. uvolnit veřejný klíč z paměti (není důvod plýtvat dynamickými zdroji jako jsou velká čísla, když už nejsou používána)
3. podepisovat/ověřovat/šifrovat/dešifrovat, což vede k `exptmod` a PKCS #1 [v2.0] doplňování [padding]

Rozhraní přesně napodobující symetrickou stranu skutečně nebude fungovat na 100 %. Například symetrické operace obvykle nehavarují. Také metody ověřování/dešifrování mohou selhat natvrdo [kvůli nedostatku paměti na haldě] či naměkko [invalidní paket]. Dávalo by více smyslu navrhnout jednoduché API pro kryptografii s veřejným klíčem, než mrzačit API pro symetrické šifrování do něčeho kompatibilního.

Mimochodem, byl jsem kontaktován, zda mám zájem o uskutečnění tohoto projektu a nabízím svou `public domain` knihovnu `LibTomCrypt`. Navrhuji portovat ji jako kernel modul (po odstranění věcí, jež nejsou potřeba, jako je symetrické šifrování) a uvolnit pod GPL.

## Linux 2.6.7, 40 e-mailů

Linus Torvalds ohlásil Linux 2.6.7 s tím, že nejvíce pozornosti si zaslouží jednořádková oprava obtěžující výjimky plovoucí řádky. Tomas Szepe napsal, že `airo.ko` z 2.6.7 (oproti verzi 2.6.6) neumožňuje uživateli nastavit ESSID via „`echo myssid >/proc/driver/aironet/ethX/SSID`“. A dodal, že takové změny by se neměly provádět uprostřed stabilní řady.

Alexander Viro zaslal krátký patch s komentářem, že takovýmto změnám se říká chyby. Kromě toho originální verze byla chybová a snadno zneužitelná, takže oprava byla skutečně potřeba. Tomas vyzkoušel záplatu a potvrdil její funkčnost.

## Kdy používat GCC Built-Ins, 17 e-mailů

Arjan van de Ven napsal, že

`gcc 3.4` získalo podporu pro několik typických bitops jako zabudovaných direktiv. Jejich použití má oproti inline `asm` několik výhod:

- `gcc` může do nich lépe optimalizovat konstanty

- gcc může lépe přeuspořádat pořadí kódu
- gcc může alokovat registry

Otázka zní, zda považujeme tuto cestu za přínosnou či ne. Abych pomohl diskusi, příkládám patch, který v závislosti na verzi překladače umožní funkci `ffz()` používat tyto direktivy gcc. Než s tím ale budu pokračovat... má to cenu?

Ze začátku se ozývaly hlasy podpory, než Linus Torvalds napsal:

Pokud se jasně neprokáže, že kód gcc je lepší, pak to NEMÁ cenu.

Přidání takové podpory vede k zvýšeným nárokům na údržbu, protože stále musíme podporovat staré verze gcc a architektury, kde GCC optimalizuje HŮŘE než my. A upřímně, pochybuju, že najdete jediný případ, kdy GCC udělá znatelně lepší práci. Jinými slovy, pravidla ohledně GCC builtinů jsou:

- Používejte jen ty, jež jsou dost staré na to, aby je měla dostupná většina uživatelů (nejlépe všichni). Hlavním důvodem je, že gcc mívá chyby a často dělá předpoklady, které pro kernel neplatí (například „je v pořádku používat rutiny knihoven“).
- Používejte je, pouze pokud prokážete znatelné vylepšení. Teoretické argumenty prostě neplatí.

Až dosud jediným případem, jenž měl cenu, byly věci kolem `memcpy()` a i zde byla původní makra, která dělala téměř stejnou práci (ale byla tak škaredá, že stálo za to provést změnu).

## Odkazy

- [1] <http://www.csn.ul.ie/~mel/projects/vm>
- [2] <http://www.phptr.com/title/0131453483>
- [3] <http://www.perens.com/books>
- [4] <http://squashfs.sourceforge.net>
- [5] <ftp://people.redhat.com/alan/Kernel/>
- [6] <http://www.redhat.com/archives/fedora-devel-list/2004-May/msg00908.html>
- [7] <http://www.win.tue.nl/~aeb/linux/Large-Disk.html>
- [8] [http://www.win.tue.nl/~aeb/partitions/partition\\_tables.html](http://www.win.tue.nl/~aeb/partitions/partition_tables.html)
- [9] <http://redhat.com/ mingo/nx-patches/nx-2.6.7-rc2-bk2-AE>
- [10] <http://redhat.com/ arjanv/2.6/RPMS.kernel/>
- [11] <http://redhat.com/ mingo/nx-patches/QuickStart-NX.txt>
- [12] [http://www.geocities.com/dt\\_or/input/misc/mousedev-2.6.7-rc2-cumulative.patch.gz](http://www.geocities.com/dt_or/input/misc/mousedev-2.6.7-rc2-cumulative.patch.gz)
- [13] <http://linux-abi.sf.net>
- [14] [http://sf.net/tracker/index.php?func=detail&aid=968070&group\\_id=13130&atid=313130](http://sf.net/tracker/index.php?func=detail&aid=968070&group_id=13130&atid=313130)
- [15] <http://sf.net/projects/ibcs>
- [16] [http://tree.celinuxforum.org/pubwiki/moin.cgi/CELinux\\_5fPatchArchive](http://tree.celinuxforum.org/pubwiki/moin.cgi/CELinux_5fPatchArchive)
- [17] <http://sourceforge.net/projects/high-res-timers/>
- [18] <http://umbrella.sf.net>

\*\*\*

## Zprávičky

1.7.2004

*Petr Krčmář*

Vyšla první ostrá verze Mono 1.0 pro Linux, Unix, MacOS X a MSWindows. Můžete si přečíst co je nového nebo rovnou stahovat.

1.7.2004

*Petr Krčmář*

„Je válka prohlížečů zpět? aneb Jak Mozilla Firefox přetrumfla Internet Explorer“ je název velmi zajímavého rozboru aktuální situace na bitevním poli webových prohlížečů.

1.7.2004

*Petr Krčmář*

Potřebujete-li pořádně otestovat stabilitu systému, přicházejí vám na pomoc kolegové z týmu Linux Test Project, který vám umožní pořádně „stresovat jádro“.

1.7.2004

*Petr Krčmář*

Pamatujete firmu Loki Games? Zajímají vás některé hry, které (bohužel) ve verzi pro váš oblíbený systém neseženete? Můžete se podívat na LIFLG – Loki Installers for Linux Gamers.

2.7.2004

*Petr Krčmář*

Mplayer je univerzální multimediální přehrávač pro GNU/Linux a další platformy. Pokud jste jej ještě nevyzkoušeli, můžete se podívat na článek o jeho instalaci a provozu.

2.7.2004

*Petr Krčmář*

Podrobné srovnání distribucí Fedora, Mandrake a SUSE si můžete přečíst na flexbeta.net. Autor se věnuje mnoha aspektům od instalace přes boot až po jednotlivé softwarové balíky.

2.7.2004

*Petr Krčmář*

Analýza Sunu tvrdí, že otevření Javy celou platformu roztříští a znehodnotí. Konkurence zase hlásí, že vývoj v komunitě je příliš neefektivní a pomalý. Celá debata na toto téma každopádně ukazuje, že je Sun na správné cestě.

3.7.2004

*ALIUM*

Vyšly nové ovladače pro nVidii. Mimo jiné je již možné je používat na Fedora Core 2 (podpora 4K stack). Download.

4.7.2004

*Robert Krátký*

Na NewsForge argumentuje Roblino, že uživatelé Linuxu jsou rozmazlení. A vězí to v pohodlnosti běžné instalace – všechny základní software už bývá součástí distribuce.

4.7.2004

*David Nečas (Yeti)*

Tahovou strategii Battle for Wesnoth můžete od verze 0.8 (vyšla včera) pařit už i česky. Aktuálnější verzi překladu lze stáhnout ode mne.

5.7.2004

*Robert Krátký*

The Register informuje, že EFF vytáhla do boje proti 10 nejhorším (nejabsurdnějším) softwarovým patentům. Všechny zmiňované patenty jsou drženy firmami, které buď vyhrožovaly nebo již iniciovaly soudní pře s jednotlivci, neziskovými organizacemi nebo malými firmami.

6.7.2004

*Robert Krátký*

Libranet 2.8.1 je k dispozici ke stažení. Libranet je komerční distribuce založená na Debianu. Volně ke stažení bývá uvolněna pouze starší verze. Tentokrát si však na zkoušku můžete stáhnout i tu nejnovější.

6.7.2004

*Robert Krátký*

Fedora Project představil předběžný časový plán pro vydání FC3. V emailu zveřejněném na LWN je i nástin toho, na co se v nové verzi můžete těšit.

6.7.2004

*Robert Krátký*

Dlouhou recenzi, či spíše zprávu o stavu, v jakém se nachází kancelářský balík projektu KDE, KOffice, si můžete přečíst na zurgy.org.

6.7.2004

*Michal Novák*

Na serveru tucnacek.com se objevil článek zachycující atmosféru na letošním republikovém kole soutěže v programování. Asi se tam příště sám přihlásím :).

6.7.2004

*Ondra 'Kepi' Kudlík*

Na googlu si někdo nepěkně pohráł s českým překladem. Zkuste si dát něco vyhledat a podívejte se na tlačítko...

6.7.2004

*Robert Krátký*

Chcete vědět, Kdo napsal Linux? Tento článek vám to prozradí. A pokud se vám to nebude zdát



pravděpodobné, můžffice.org je recenze knihy OpenOffice.org pro zelenáče. Knihu napsal Pavel Satrapa, recenzi Martin Hassman.

**7.7.2004**

*Robert Krátký*

Na IT-Analysis.com si můžete přečíst rozhovor s technologickým šéfem a spoluzakladatelem VMware, firmy, jejíž produkt umožňuje spouštění operačních systémů ve virtuálním prostředí hostitelského OS (např. Windows na Linuxu). Recenze.

**7.7.2004**

*Vlastimil Ott*

Rádi byste si zahráli „staré“ hry pod Linuxem? Třeba Warcraft, resp. jeho novou verzi? Tak podepište petici k jeho portování na Linux!

**7.7.2004**

*Vlastimil Ott*

Pravděpodobně nejdéle běžící počítač typu PC v republice se nachází v jedné z laboratoří Fyzikálního ústavu Akademie věd ČR. Víte, jak dlouho běží? :-)

**8.7.2004**

*Robert Krátký*

Byla oznámena první betaverze KDE 3.3 „Klassroom“. Info stránka s odkazy na download. Seznam požadovaného software pro kompilaci.

**8.7.2004**

*Robert Krátký*

Na LinuxGazette.net vyšlo poměrně podrobné HOWTO popisující práci s GPG a frontendem KPGP: Front and Back: KPGP and GPG.

**8.7.2004**

*Robert Krátký*

Novell rozdává úctyhodný balík OS a software. Dohromady přes 10 GB na CD a DVD. Obsahuje mimo jiné SUSE Linux 9.1 DVD a Ximian Desktop.

**8.7.2004**

*Stanislav Puffler*

Zajímavý článek o „vpuštění“ Linuxu do domácnosti Linuxem neposkrvněných s podrobným popisem přípravy desktopu, výběrem a stažením obrazu distribuce, samotné instalace a konfigurace zveřejnil server reviews.cnet.com. Velmi přínosné čtivo pro dosud neodhodlané a neznalé.

**8.7.2004**

*Tomáš Hála*

Chcete-li přejít z Windows na Linux, naučte se nejprve místo nepřenositelných uživatelských programů a formátů od MS používat multiplat-

formní programy a otevřené formáty. V tom vám může pomoci CD s výběrem takových programů. Zvyknete-li si na jejich používání ještě ve známém prostředí, bude pozdější přechod mnohem snazší.

**9.7.2004**

*Robert Krátký*

Recenze nové verze WineX (umožňující hraní Win her na Linuxu) od fy. Transgaming, kterou přejmenovali na Cedega, je na OSNews. A je to recenze téměř nadšená.

**9.7.2004**

*Robert Krátký*

Mandraku se povedlo prodat svoji serverovou distribuci francouzské vládě (a nahradit tak na serverech NT). Na to, že to s MandrakeSoftem dlouho vypadalo nahnuté, se mají čile k světu.

**10.7.2004**

*Robert Krátký*

LinuxJournal popisuje, proč a jak zkompileovali vývojáři Red Hatu vývojové prostředí Eclipse tak, aby mohlo běžet nativně, tj. bez Java VM.

**10.7.2004**

*Robert Krátký*

Gentoo Linux vyhlásil soutěž o nejlepší (re)design jejich webových stránek. Z došlých návrhů bude vybráno pět, o kterých pak bude moci hlasovat komunita.

**11.7.2004**

*Robert Krátký*

Po vývojářích jádra se o název 64 bitové verze Linuxu hádali i programátoři Debianu. Nakonec zvítězilo označení amd64, ale ideální jméno to zjevně není. Více viz debianplanet.org.

**11.7.2004**

*Robert Krátký*

Šikovná sada nástrojů OpenOffice.org Tools (doplňující oblíbený kancelářský balík) je ve verzi 1.0.

**12.7.2004**

*Robert Krátký*

Po delší době opět „oficiální“ vydání jádra: 2.6.8-rc1 (oznámení), které má díky té velké pauze, kterou Linus přičítá hlavně stěhování, dost obsáhlý patch.um, které vám jistě pomůže s vašim slackem. Motto: Slackers helping slackers.

**12.7.2004**

*Stanislav Puffler*

Autoři hip-hopového muzikálu „The Bomb-itty of Errors“ si tentokrát vzali na paškál dvě osobnosti IT průmyslu – šéfa Apple Computer Steva

Jobse a Billa Gatese. Podle autorů muzikálu nazvaného „Nerds“ jde o muzikálovou softwarovou satiru. Představení muzikálu proběhne tento týden v Poughskeepsie ve státě New York. Zdroj: Živě.

**13.7.2004**

*Vlastimil Ott*

RedHat mění statistiky svých hospodářských výsledků za uplynulá léta. Na základě konzultace s novým poradcem tu přidali, tam zase ubrali. Suma sumárum to vyšlo nastejno, ale akcionáři stejně znervózněli. (Také zkráceně, německy.)

**14.7.2004**

*Robert Krátký*

Nedlouho po té, co se objevila první betaverze KDE 3.3, už je na světě live CD, které vám dovolí ji vyzkoušet a případně pomoci s vychytáváním chyb. OneBase Linux.

**14.7.2004**

*Robert Krátký*

Byla uvolněna Fedora Core 3 Test 1. Čtyřdiskovou distribuci můžete stahovat z oficiálních stránek.

**14.7.2004**

*Robert Krátký*

PHP 5.0 je venku. Seznam hlavních novinek najdete v oznámení na stránkách php.net. Stahovat můžete odsud.

**14.7.2004**

*Vlastimil Ott*

Úžasný archiv aplikací pro GNOME a programy pod GTK se nachází na GnomeFiles. K dispozici je přes dvě stě programů.

**14.7.2004**

*Robert Krátký*

Je to chvílka, ale DotDeb.org už má připraveny balíky PHP 5.0 pro Debian stable. V tamějším repozitáři najdete ještě aktuální MySQL.

**15.7.2004**

*Antonín Kolísek*

Na světě je další vepora amd64, podpora VIDIXu u nových Radeonů. Více o změnách viz News.

**15.7.2004**

*Robert Krátký*

Red Hatu se možná nevyplatí čachry, které prováděl se svými hospodářskými výsledky. Dnes byly oznámeny dvě společné žaloby, které na něj byly podány kvůli údajnému defraudování investorů uváděním nesprávných a zavádějících finančních informací. Více Triangle Business Journal.

**15.7.2004**

*Vlastimil Ott*

Na osnews vychází – jak je zvykem – review Slackware 10.0. Recenze vyznívá příznivě, i když zlé jazyky by mohly v otázce nový software kontra stabilní software najít téma k flamu. Keep slackin'!

**16.7.2004**

*Robert Krátký*

Jak zabezpečit Apache webserver se dočtete na Linux.com. Text navazuje na starší článek o instalaci Apache pomocí chroot.

**16.7.2004**

*Robert Krátký*

Java Enterprise System možná bude uvolněn s open source licencí. Sun se ještě nerozhodl, ale přihodil už tento produkt na seznam kandidátů pro „otevření“. Viz News.com.

**16.7.2004**

*Leoš Literák*

Pilujete-li zrovna před dovolenou slovíčka, mohl by se vám hodit tento malý program: (Java) [www.yan.cz/slovicka](http://www.yan.cz/slovicka).

**16.7.2004**

*satori*

Včera byl oficiálně uvolněn nový antivir firmy ALWIL Software avast! for Linux s integrovaným emailovým a souborovým skenerem.

**16.7.2004**

*ne-jsa*

Chtěl bych upozornit na velmi zajímavý projekt Xen, který umožňuje současný běh více OS. Z pohledu OS vypadá jako nová (virtuální) CPU architektura, takže je OS třeba portovat, což na oplátku přináší výkonnostní výhodu oproti např. UML. Momentálně exist

**17.7.2004**

*Robert Krátký*

Tom Walker na NewsForge ukazuje, jak snadné je nepochopit GPL. Z tolika nepřesností a nepravd, které dokázal nashromáždit do jediného článku, by se až zamotala hlava.

**18.7.2004**

*Robert Krátký*

Distrowatch informuje, že byl vypuštěn Mandrake Linux 10.1 alpha 1 (3 CD). Obsahuje to nejnovější, co je k dispozici (např. jádro 2.6.8-rc1 a gcc 3.4.1). Viz také časový plán pro vydání nové verze.



- 19.7.2004** *Jiří Hnídek*  
 Na adrese [www.blender3d.cz](http://www.blender3d.cz) vznikl nový český web o OpenSource programu Blender. Naleznete zde spoustu článků, diskuzní fórum, galerii, atd.
- 22.7.2004** *Petr Semelka*  
 Před dvěma dny vyšla nová verze Wine (implementace Windows API pod Linuxem) – 20040716. Ve této verzi je nově implementováno Microsoft Installer DLL a je opraveno hodně chyb. Stahovat můžete z domovské stránky.
- 23.7.2004** *tomasini*  
 Je to pár dní, co byla vydána stabilní verze oblíbeného souborového manažera Krusader. Současná verze 1.40 přináší drobné změny vzhledu a mnohem více možností nastavení, viz changelog.
- 23.7.2004** *Jan Polzer*  
 Nově vzniklý elektronický obchod Polzer SW nabízí za příjemné ceny linuxové distribuce a další volně šiřitelné operační systémy. Postupně budou přibývat také CD a DVD s velkými balíky volně šiřitelného softwaru.
- 23.7.2004** *Vladimír Slavík*  
 Kdyby programy byly slušnější a jejich chování uhlazenější, možná bychom byli tvořivější a produktivnější jako jejich uživatelé. Zajímavý článek na stránkách [newscientist.com](http://newscientist.com) popisuje projekt, kde dvě skupiny uživatele komentáře: 10, poslední 26.7.2004 20:32
- 26.7.2004** *Petr Krčmář*  
 Tux.org přinesl zajímavé srovnání tří live distribucí: Knoppix 3.4, PCLinuxOS 7a, Damn Small Linux 0.72.
- 26.7.2004** *Petr Krčmář*  
 Pokud vás zajímají live distribuce, můžete se podívat na rozsáhlou tabulku, která obsahuje snad všechno, s čím se v této oblasti můžete setkat.
- 26.7.2004** *Vlastimil Ott*  
 Norská firma Trolltech a zástupci KDE Free Qt Foundation uzavřeli smlouvu o dalším použití knihovny Qt. Smlouva nahrazuje původní dohodu z roku 1998.
- 26.7.2004** *Petr Krčmář*  
 Open Source Initiative (OSI) dnes podporuje více než 50 různých licencí, které splňují kritéria pro zařazení mezi tzv. open source licence. Všechny licence ale psali právníci a tak jim vývojář porozumí jen stěží. Jak se v nich tedy vyznat a vybrat si pro svůj projekt tu pravou? V tom vám může poradit například článek na Newsforge.
- 26.7.2004** *Honza Bartoš*  
 Vyšla verze 0.39 SVG vektorového grafického editoru Inkscape. Nově zvládá mimojiné také výplň vzorkem a zaoblování polygonů.
- 27.7.2004** *Leoš Literák*  
 Jiří Svoboda nám zaslal scan kousku článku z časopisu Květy, kde televizní komentátorka Daniela Drtinová popisuje své zážitky s Richardem Stallmanem.
- 27.7.2004** *Petr Krčmář*  
 Většina uživatelů tvrdí, že je Linux plně připraven pro použití na desktopu. Názor člena druhého tábora přináší jako rozsáhlou úvahu osnews.
- 27.7.2004** *Petr Krčmář*  
 Novell vyvíjí novou odlehčenou verzi SUSE Linuxu, která by mnabubřelou.
- 27.7.2004** *Vlastimil Ott*  
 Vývojáři testovali základní desky firmy Asus a dospěli k názoru – podpořenému arogantním chováním firemního supportu – že jsou pro uživatele zcela nevhodné. „Nekupujte je“, zní jejich rada. Souhrn (německy).
- 28.7.2004** *Petr Krčmář*  
 Australský deník The Sydney Morning Herald doporučuje používat raději Mozillu Firefox než Internet Explorer. Hlavními důvody jsou inovace, uživatelská přívětivost a bezpečnost.
- 28.7.2004** *Petr Krčmář*  
 Linux se chlubí nejširší podporou v oblasti souborových systémů. To ale neznamená, že může usnout na vavřínech. Už dnes je známo několik nových konceptů dalšího vývoje souborových systémů obecně. Kam tedy kráčejí a jak to vidí Linux si můžete přečíst v článku na [linux.com](http://linux.com).

**28.7.2004**

*šmejky*

Na MobilServeru vyšel článek s názvem „Linuxový mobil snů“. Jedná se o Motorolu A780.

**29.7.2004**

*Petr Krčmář*

Netcraft se zamýšlí nad tím, zda nemůže Mozilla Firefox potkat stejný osud jako kdysi Netscape. MozillaNews však oponuje tím, že se Mozilla již poučila z minulosti a stejných chyb se vyvaruje.

**29.7.2004**

*Petr Krčmář*

Pamatujete se ještě na ambiciózní projekt UnitedLinux, který měl standardizovat a sjednotit stávající distribuce? Co se s ním vlastně stalo, jak to celé bylo a kde se stala chyba vysvětluje Computerworld.

**29.7.2004**

*Petr Krčmář*

IronPython je nová implementace Pythonu pro .NET a Mono platformy. Nyní je k dispozici pod licencí CPL.

**30.7.2004**

*Petr Krčmář*

Společné úsilí Sun, RedHat a dalších vyústilo v GNOME Human Interface Guidelines 2.0, příručku pro vývojáře a grafiky, která říká jak vytvořit aplikaci, která správně zapadne do prostředí GNOME.

**30.7.2004**

*Petr Krčmář*

Máte-li zájem, můžete si poslechnout krátkou přednášku Warrena Sealeyho, Security Managera z Message Labs, která pojednává o problematice spamů a nastiňuje i některé možné obranné postupy.

**30.7.2004**

*Petr Krčmář*

Americká armáda by prý raději neměla do nového obranného systému začleňovat Linux. Nově vybudovaná síť v budoucnu spojí všechny vojenské složky do jednoho společného celku. Hlavním argumentem proti Linuxu je prý jeho nebezpečnost a přílišná otevřenost.

**30.7.2004**

*Vlastimil Ott*

Je to už dva dny, kdy vyšel nový BASH 3.0. Změny se týkají zejména skriptovacího jazyka, chování zabudovaných příkazů zůstává (až na výjimky) samozřejmě stejné.

**30.7.2004**

*Petr Krčmář*

Jak to vypadá, když dlouholetý uživatel Pine přechází na Mozillu Thunderbird, popisuje článek na osnews.com.

**30.7.2004**

*Martin Kysela*

Víte, že je dnes Den systémových administratorů? Tak ho pěkně oslavte!