

GRUB – zavaděč systému

Piotr Machej

Co vlastně je ten zavaděč systému (ang. *bootloader*)? Potřebuji jej vůbec ke štěstí? Jedná se o program umístěný v úvodních sektorech pevného disku (např. v *MBR – Master Boot Record*). Po spuštění počítače a provedení testů předává *BIOS* řízení právě programu obsaženému v *MBR* disku označenému jako startovací, tj. našemu zavaděči systému. Jeho úkolem je nahrát operační systém do paměti a předat mu řízení. V současné době většina rozšířených zavaděčů systému umožňuje volbu jednoho ze systémů nainstalovaných na disku. Nejoblíbenějšími programy tohoto typu, dodávanými spolu s distribucemi GNU/Linuxu, jsou *LILLO* a *GRUB (GRand Unified Bootloader)*. V tomto článku se pokusím přiblížit druhý z uvedených nástrojů.

Příklad použití

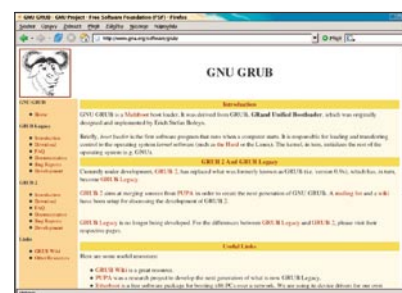
Jsem traditionalista a neskrývám to. Z tohoto důvodu jsem velmi dlouho (zvláště na vysoké škole) spouštěl GNU/Linux z diskety. Spolubydlící na pokoji dlouho nemohli přijít na to, proč se mi spouští jiný systém než jim. A nač vůbec spouštět něco, na čem si nejde zahrát *Doom*? Později jsem začal používat *LILLO*. Díky němu jsem si už nemusel pamatovat, ve které kapse jsem nechal disketu. Navíc v té době byl můj počítač již v bezpečí před úskoky studentstva, mohl jsem si tedy dopřát špetku luxusu. Přestože jsem měl u později používaných distribucí možnost přejít na *GRUB*, několik let jsem tvrdohlavě zůstával u *LILLO*. Pro změnu mě přesvědčila právě moje lenost... Jednou,

když jsem už poněkoli káté zapomněl zadat příkaz `lilo` po provedení změny v konfiguračním souboru a musel jsem hledat záchranou disketu, řekl jsem si: „Tak tohle by stačilo!“ Nainstaloval jsem si *GRUB* a od té doby si nemám nač stěžovat. Dokonce když přijedou příbuzní, neexistuje síla, která by jim umožnila spustit systém, se kterým si nemají co hrát – alespoň dokud se jim nepodaří uhádnout heslo.

Instalace

Balíčky s programem *GRUB* najdete prakticky ve všech distribucích. Ba co víc, při instalaci většiny distribucí si můžeme vybrat zavaděč, který chceme používat. Pokud se rozhodneme pro *GRUB*, systém se hned postará o to, aby byl tento program odpovídajícím způsobem nakonfigurován. Pokud jsme se však tehdy pro jeho instalaci nerozhodli, budeme to muset nyní udělat sami.

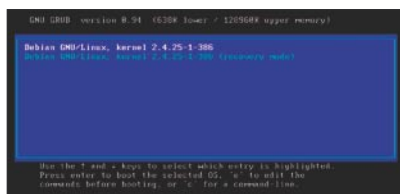
Nejdříve ze všeho by bylo dobré se ujistit, že *GRUB* opravdu nemáme nainstalovaný. V případě *Auroxu* k tomuto



Obrázek 1. Verze GRUBu, využívaná v současné době v distribucích, již není dále vyvíjena

Na DVD

Na DVD se nacházejí nástroje, o nichž je v článku řeč.



Obrázek 2. Základní menu GRUBu nevypadá příliš přitažlivě

účelu použijeme příkaz `rpm -q grub`. Pokud se objeví hlášení, že balíček `grub` není nainstalován, budeme muset tento stav změnit. Balíček s `GRUBem` si můžeme nainstalovat z několika různých zdrojů. Prvním z nich jsou CD, z nichž jsme si instalovali naši distribuci. V případě *Auroxu* se příslušný balíček nachází na prvním CD, po jeho vložení do jednotky tedy můžeme zadat příkaz:

```
# mount /mnt/cdrom/
# rpm -Uvh /mnt/cdrom/Aurox/RPMS/grub-0.94-5.i386.rpm
```

Druhým zdrojem jsou katalogy balíčků. Pokud využíváme jeden ze systémů správy balíčků, můžeme použít odpovídající příkaz, např.:

```
# yum install grub
# apt-get grub install
# emerge grub
```

Zadáme samozřejmě jen jeden z uvedených příkazů. Analogickým způsobem můžeme využít systém správy balíčků pro aktualizaci naší verze `GRUBu` (např. příkazem `yum update grub`).

Pak máme ještě třetí možnost. Pokud nám záleží na tom, abychom měli nejnovější verzi, můžeme si ji stáhnout z domácích stránek projektu (<http://www.gnu.org/software/grub/>). Jelikož již byly zahájeny práce na `GRUB 2`, byla starší verze (řada *0.9.x*, která se stále ještě používá v distribucích) přejmenována na *GRUB Legacy*. Máme zde k dispozici pouze zdroje, jejichž instalace se provádí běžným postupem – po stažení a rozbalení souboru (tar `xzvf grub-0.95.tar.gz`) jdeme do takto vytvořeného adresáře a zadáme příkazy: `./configure` a `make`. Následně, po získání oprávnění administrátora (příkaz `su`), zadáme poslední příkaz: `make install`. Lepším řešením však je využít balíčků dostupných pro naši distribuci.

Nyní musíme nakonfigurovat náš `GRUB` a pak jej umístit na disk.

Základní konfigurace

Hlavním konfiguračním souborem `GRUBu` je `/boot/grub/grub.conf`. Obvykle existují také dva symbolické odkazy na tento soubor: `/boot/grub/menu.lst` a `/etc/grub.conf` (takto je to v *Auroxu* – v jiných distribucích může být hlavním konfiguračním souborem jeden ze tří uvedených). Není to sice nezbytně nutné, jelikož můžeme použít příkazový řádek `GRUBu`, ale správně nakonfigurovaný soubor nám velmi usnadní práci. Díky tomu nebudeme muset po každém spuštění počítače zadávat řadu příkazů a v případě, že máme nainstalovaných více systémů, budeme moci využít pohodlné menu.

Záchranná disketa

Velice důležité je postarat se o to, abychom měli záchrannou disketu nebo CD. Disketu můžeme vytvořit během instalace distribuce. Pokud jsme to neudělali, můžeme zpravidla využít první distribuční CD. Obvykle nevzniká potřeba použít záchrannou disketu, ale při změně zavaděče se mohou objevit problémy, kvůli nimž se nám nepodaří systémy nainstalované na disku spustit.

Spuštění GNU/Linuxu

Nejdříve si vyzkoušejme použití `GRUBu` pro spuštění našeho GNU/Linuxu. K tomuto účelu bychom měli mít v souboru `/boot/grub/grub.conf` alespoň čtyři řádky:

```
title Aurox Linux (2.4.22-1)
root (hd0,7)
kernel /vmlinuz-2.4.22-1 ro root=LABEL=/
initrd /initrd-2.4.22-1.img
```

První z nich obsahuje po výrazu `title` text, který se objeví v menu. Nejlepší bude, když zde uvedeme název systému (distribuce) a v závorce verzi jádra. Samozřejmě můžeme přidat i další informace (např. zda je používán *FrameBuffer*), které pomohou identifikovat položku v menu.

Další řádek udává, na kterém diskovém oddílu se nachází adresář `/boot/`. V našem příkladu je umístěn na zvláštním diskovém oddílu `/dev/bda8` a hlavní systémový adresář `/` se nachází na diskovém oddílu `/dev/bda9`. O názvech diskových oddílů používaných `GRUBem` si můžete více přečíst v rámečku *Názvosloví disků a diskových oddílů*.

V dalším řádku určíme, které jádro chceme nahrát. Po slově *kernel* zadáme cestu k souboru s jádrem (v našem příkladu *vmlinuz-2.4.22-1*), uvedenou ve vztahu k diskovému oddílu zapsanému o řádek výše. Jelikož v našem příkladu máme pro adresář `/boot/` zvláštní diskový oddíl, zadáme cestu `/vmlinuz-2.4.22-1`. Po názvu souboru s jádrem můžeme zadat parametry jádra. V našem případě uvedeme diskový oddíl s hlavním souborovým systémem (*root*). Vysvětlení hodnoty `LABEL=/` se nachází v rámečku *Názvosloví disků a diskových oddílů*. Všimněme si, že výraz *root* má v tomto řádku jiný význam než v předchozím. Jedná se zde o parametr jádra, určující hlavní souborový systém, kdežto slovo *root* v druhém řádku je příkazem `GRUBu`, jehož prostřednictvím můžeme zjistit, ve vztahu ke kterému zařízení budou popsány další přístupové cesty.

Poslední řádek není nezbytný, pokud ale máme vytvořený soubor *initrd* vhodný pro naše jádro, měli bychom ho využít. Příkaz `initrd` obsažený v tomto řádku nahraje `ramdisk`. To, zda jej využijeme, bude záviset na konfiguraci našeho jádra.

Všeobecné volby

Kromě čtyř uvedených řádků stojí za to hned nyní přidat několik dalších, které určí, jak se bude chovat naše menu. Jedná se o následující řádky:

```
default=0
timeout=30
```

Tyto řádky musíme umístit před (zdurazňuji – před) řádky uvedenými dříve, hned na začátek souboru `/boot/grub/grub.conf`. První z nich určuje, která položka menu bude zvolena implicitně. Jak víme, `GRUB` počítá od nuly, v našem příkladu tedy vybereme první (a prozatím i jedinou) položku v menu.



Obrázek 3. Dokumentaci `GRUBu` najdeme také na WWW stránkách

Druhá volba určuje, že *GRUB* bude čekat na naše rozhodnutí 30 sekund (můžeme nastavit jinou hodnotu). Pokud nestiskneme žádnou klávesu, automaticky spustí položku uvedenou ve volbě *default*. Pokud jsme použili volbu *timeout*, aniž bychom nastavili volbu *default*, bude spuštěna první definovaná položka v menu.

Ukládání GRUBu na disk

Nyní již můžeme uložit *GRUB* na disk (pokud jsme to ještě neudělali). Obvykle jej budeme chtít umístit na první disk. K tomu slouží příkaz `grub-install /dev/hda`.

Pokud později změníme obsah konfiguračního souboru `/boot/grub/grub.conf`, nebudeme tento příkaz muset opět zadávat. *GRUB* automaticky najde změny v souboru a při dalším spuštění počítače zobrazí odpovídající menu. V podstatě jediný případ, kdy bychom mohli mít zájem znovu zadat tento příkaz, je okamžik aktualizace *GRUBu* na novou verzi.

Stojí za to znovu spustit počítač a přesvědčit se, jak funguje náš *GRUB*. Mělo by se nám zobrazit jednoduché menu s jednou položkou na výběr. Po jejím potvrzení klávesou `[Enter]` (nebo po 30 sekundách čekání) by se měl nahrát náš GNU/Linux.

Spouštění různých systémů

Možnosti *GRUBu* nejvíce oceníme, máme-li nainstalovány také jiné operační systémy – mnoho uživatelů systému v barvách tučňáka používá také *MS Windows* (třeba jenom proto, aby si občas zahráli) nebo *FreeBSD* (např. kvůli studiu nebo práci).

Jiné linuxové distribuce

Jestliže chceme pomocí *GRUBu* spouštět na našem počítači také jiné linuxové distribuce, postupujeme stejně, jako v případě popsaném v kapitole *Spouštění GNU/Linuxu*. Stačí dopsat další sekci začínající řádkem *title* a obsahující odpovídající volby. Stejný způsob můžeme využít i tehdy, chceme-li v naší distribuci otestovat nové jádro. Předpokládejme, že jsme si je sami zkompilevali a pojmenovali je *vmlinuz-2.4.22-1a*. V tomto případě můžeme do souboru `/boot/grub/grub.conf` nakonec přidat následující řádky:

Názvosloví disků a diskových oddílů

V konfiguračním souboru *GRUBu* (`/boot/grub/grub.conf`) obvykle nepoužíváme pro označení disků a diskových oddílů *hda*, *hdb2* apod., jak to známe z GNU/Linuxu. Namísto toho se zde setkáváme s označeními jako (*fd0*) nebo (*hd1,2*). Ze začátku to může být poněkud matoucí, ale lze si na to rychle zvyknout.

Popis (*fd0*) označuje první disketovou jednotku (je třeba pamatovat na závorky a na to, že mezi znaky není mezera). V případě pevných disků odpovídá název (*hd0*) disku *hda*, název (*hd1*) označuje disk *hdb* atd. Vidíme tedy, že namísto písmen se zde používají číslice, přičemž číslování začíná od nuly. Pro určení konkrétního diskového oddílu mu přiřazujeme číslo (opět začínáme na nule) po čárce. Pro *hda1* tedy máme (*hd0,0*), pro *hda7* – (*hd0,6*) a pro *hdc3* – (*hd2,2*). V podstatě si stačí pamatovat, že číslování začínáme od 0 a ne od 1. Pokud máme pochybnosti, můžeme si přiřazení názvů k diskům ověřit v souboru `/boot/grub/device.map`. U počítače, který má jen jednu disketovou jednotku a jeden pevný disk, by měl tento soubor mít následující obsah:

```
(fd0) /dev/fd0
(hd0) /dev/hda
```

U některých distribucí (např. u *Red Hatu* a z něj odvozených – *Fedora* a *Auroxu*)

```
title Aurox Linux (2.4.22-1a)
root (hd0,7)
kernel /vmlinuz-2.4.22-1a ro root=LABEL=/
initrd /initrd-2.4.22-1a.img
```

Pokud jsme nevygenerovali *ramdisk*, vynecháme poslední z těchto řádků a namísto `root=LABEL=/` zadáme název příslušného diskového oddílu (v našem příkladu `root=/dev/hda9`). Pokud na to zapomeneme, přestane se náš systém nahrávat ve fázi připojování hlavního souborového systému.

Windows, DOS

Nejjednodušším způsobem, jak spouštět systémy z rodiny *MS Windows*, je umístit na konec souboru `/boot/grub/grub.conf` následující řádky:

```
title DOS nebo Windows
rootnoverify (hd0,0)
chainloader +1
```

Příkaz *title* již známe. V řádku *rootnoverify* uvedeme diskový oddíl, na kterém

se rozšířilo odkazování na diskový oddíl pomocí labelu. Máme-li v našem konfiguračním souboru řádek v podobě: `kernel /vmlinuz-2.4.22-1 ro root=LABEL=/`, všimněme si textu `root=LABEL=/`. Nahrazuje běžný text v podobě `root=/dev/hda1` (pokud se náš adresář / nachází na diskovém oddílu *hda1*). Možná přemýšlíte o tom, co z toho máme, a proč se to takto používá. Tento zápis umožňuje volně stěhovat disk v rámci našeho počítače, např. v případě přidání nové jednotky nebo vypalovačky. Pokud používáme labely, nebudeme muset upravovat konfigurační soubory (tj. *grub.conf* a *etc/fstab*).

Vzniká zde ale otázka, jak načíst a vytvořit labely pro diskové oddíly. Slouží k tomu příkaz `e2label /dev/hda1`, ve kterém namísto `/dev/hda1` vložíme název diskového oddílu *ext2* nebo *ext3*. Tento příkaz vyvolá zobrazení labelu asociovaného s daným oddílem. Pokud chceme přiřadit diskovému oddílu nový label, použijeme zápis `e2label /dev/hda1 label`.

Když už jsme u toho, je třeba podotknout, že label můžeme použít jen tehdy, nahráváme-li také jádro odpovídající *ramdisk*. Slouží k tomu příkaz `initrd /initrd-2.4.22-1.img`, nacházející se pod již zmiňovaným řádkem konfiguračního souboru.

se nachází systém, jenž chceme nahrát. Tento příkaz se od příkazu *root* liší tím, že zde uvedený diskový oddíl nebude připojován. Příkaz *chainloader* předává ovládání dalšímu zavaděči. Jak to funguje? Pokud máme na diskovém oddílu `/dev/hda1` nainstalován jen jeden systém (např. *MS-DOS* nebo *MS Windows 98*), dojde k jeho spuštění. Jsou-li na disku nainstalovány systémy *MS Windows 98* a *MS Windows XP*, bude zavaděč nainstalovaný systémem *MS Windows XP* umět obsloužit oba. V příkazu *rootnoverify* tedy stačí určit diskový oddíl, na kterém je nainstalován zavaděč *MS Windows XP*. Díky tomu se nám po zvolení položky *DOS* nebo *Windows* v menu *GRUBu* zobrazí další menu, kde si budeme moci vybrat mezi *MS Windows 98* a *MS Windows XP*.

FreeBSD

Systémy z rodiny *FreeBSD* můžeme spouštět analogickým způsobem, jako *MS Windows* a *MS-DOS*, tj. s použitím příkazů *rootnoverify* a *chainloader*.

Můžeme také odkázat přímo na program `/boot/loader` pomocí následujících řádků v souboru `/boot/grub/grub.conf`:

```
title FreeBSD
root (hd1,a)
kernel /boot/loader
```

Vidíme, že se zde používají jiné názvy. Vyplyvá to ze skutečnosti, že *FreeBSD* pojmenovává své diskové oddíly podle písmen abecedy a navíc zavádí pojem *slice* pro označení nativních diskových oddílů *FreeBSD*. Pokud tedy máme *FreeBSD* nainstalováno na prvním oddílu druhého disku, měl by vyhovovat zápis (*bd1,a*). Tím zajistíme, že bude vyhledán první *slice*, na kterém se nachází diskový oddíl *a*. Chceme-li přesně určit, který *slice* má být připojen, můžeme použít formát (*bd1,0,a*).

Spouštění Memtest86+

Jednou za čas je dobré otestovat paměť našeho počítače, aby byly případné problémy včas odhaleny. Dvěma nejlepšími programy určenými pro tento účel jsou *Memtest86* a *Memtest86+*. *GRUB* nám může pomoci s jejich spuštěním.

Stačí si vybrat jeden z uvedených programů, jelikož oba navzájem čerpají ze svých zdrojových kódů. Osobně doporučuji *Memtest86+*. Nejdříve je třeba si stáhnout příslušný soubor z domácí stránky programu (<http://www.memtest.org/>). Tam musíme najít sekci *Download (Pre-built &*

ISOs), v níž se nacházejí již kompilované verze. Následně ze sekce odpovídající nejnovější verzi (v okamžiku vzniku tohoto článku to byla verze *1.40*) stáhneme soubor, který se nachází pod odkazem *Download -> Pre-Compiled Bootable Binary (.gz)*. Soubor má cca 40 kB, jeho stahování by tedy nemělo trvat příliš dlouho. Pro jistotu si můžeme ověřit, zda je soubor v pořádku. Přejdeme tedy do adresáře, v němž se tento soubor nachází a zadáme příkaz:

```
# md5sum memtest86+-1.40.bin.gz
```

Pokud jste si stáhli novější verzi, bude samozřejmě třeba odpovídajícím způsobem upravit název souboru. Ve výsledku by měl být vrácen znakový řetězec (tzv. *MD5* otisk), který srovnáme se znakovým řetězcem nacházejícím se vedle odkazu na soubor na domácí stránce programu. Pokud jsou oba tyto řetězce stejné, můžeme přistoupit k dalším činnostem.

Je čas rozbalit soubor. Provedeme to pomocí příkazu `gzip -d memtest86+-1.40.bin`. Následně si zajistíme práva superuživatele (zadáme příkaz `su -` a heslo uživatele `root`), a poté zkopírujeme soubor `memtest86+-1.40.bin` do adresáře `/boot/`. Nyní nám zbývá už jen odpovídajícím způsobem upravit konfigurační soubor *GRUBu*. Otevřeme jej v našem oblíbeném editoru (např. *Vim*) a na konec přidáme dva řádky:

```
title MEMTEST
root (hd0,7)
kernel /memtest86+-1.40.bin
```

Namísto *MEMTESTu* můžeme samozřejmě použít jiný program, který nám vyhovuje. Nezapomínejme však, že se uvedené řádky vztahují stále k našemu příkladu. Pokud máme adresář `/boot/` na jiném diskovém oddílu, budeme muset odpovídajícím způsobem upravit parametry příkazů `root` a `kernel`.

Po uložení změn můžeme restartovat systém a v našem menu by se měla objevit nová volba – *MEMTEST*. Rozhodneme-li se pro ni, spustí se program *Memtest86+* a hned začne testovat paměť. Volby programu jsou dostupné po stisknutí klávesy [c]. Pomocí klávesy [ESC] můžeme restartovat počítač, pokud nemáme čas provést úplný test.

Zkrášlujeme GRUB

Pokud se nám nelíbí implicitní vzhled menu *GRUBu*, můžeme si jej zkusit trochu oživit. Jednou z možností je přidání nebo změna obrázku zobrazovaného na pozadí. Kromě toho si můžeme podle svého vkusu upravit barvy písma.

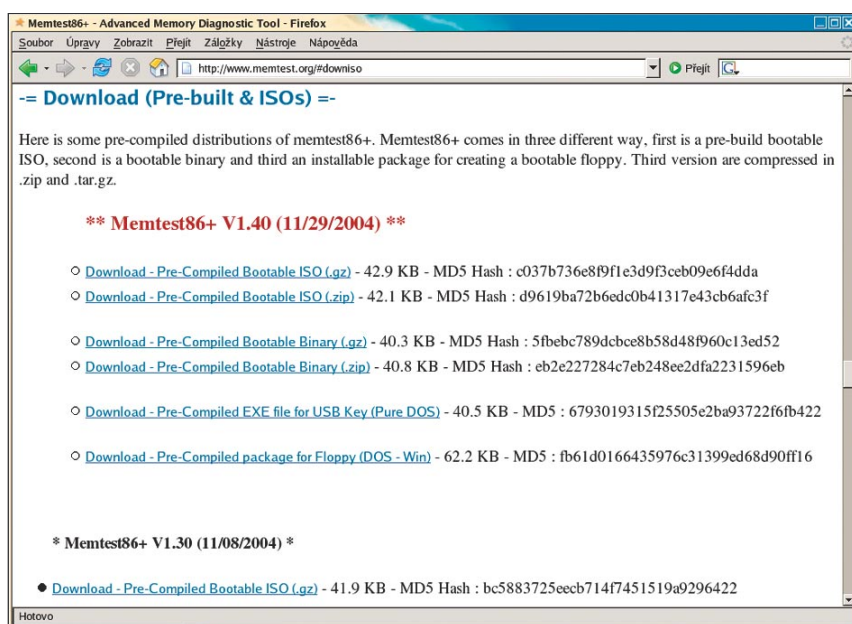
Většina distribucí dodává vlastní pozadí pro *GRUB*, v konfiguračním souboru tedy s největší pravděpodobností již máme příslušnou volbu. Je to řádek, který vypadá podobně jako tento:

```
splashimage=(hd0,7)/grub/splash.xpm.gz
```

Tento řádek přiřazuje volbě `splashimage` soubor `splash.xpm.gz` nacházející se v adresáři `/grub/` na diskovém oddílu `/dev/bda8` (v našem příkladu se tedy tento soubor nachází v adresáři `/boot/grub/`).

Pokud v našem souboru žádný podobný řádek nemáme, je třeba jej přidat a uvedené hodnoty odpovídajícím způsobem upravit. Tento řádek by se měl nacházet pod prvním řádkem `title`. Můžeme jej vložit hned pod volbu `timeout`.

Otázka ale je, kde vzít obrázky. Jak je vidět, ve volbě `splashimage` nezadáme soubory v běžných formátech, jako jsou například *PNG* nebo *JPEG*. Namísto toho zde máme soubor ve formátu *XPM* komprimovaný programem *Gzip*. To ale není jediné omezení. Náš obrázek může mít jen 14 barev a rozlišení 640x480 pixelů. Možná to nezní moc povzbudivě, ale ujišťuji vás, že lze připravit zajímavé obrá-



Obrázek 4. Memtest86+ nezabírá moc místa, ale je velmi užitečný

zky, které těmto požadavkům vyhovují. Začneme tím, že si prohlédneme několik příkladů takových obrázků.

Volba pozadí

Na stránce <http://vision.featia.net/linux/grubsplash/> najdeme několik příkladů obrázků, které lze použít jako pozadí v GRUBu. Máme zde k dispozici jak soubory ve formátu GIF (díky nim si můžeme obrázek hned prohlédnout), tak odpovídající soubory ve formátu XPM.GZ. Stáhněme si je všechny na disk. Obrázky dodávané spolu s distribucí se nacházejí v adresáři `/boot/grub/`, ale kdybychom k nim umístili i ty naše, mohl by tam vzniknout nepořádek. Namísto toho tedy vytvoříme pomocí příkazu `mkdir /boot/grub/splashes/` podadresář `splashes/` (samozřejmě k tomu potřebujeme mít oprávnění uživatele `root`). Následně do tohoto adresáře zkopírujeme všechny stažené soubory s koncovkou `.xpm.gz`.

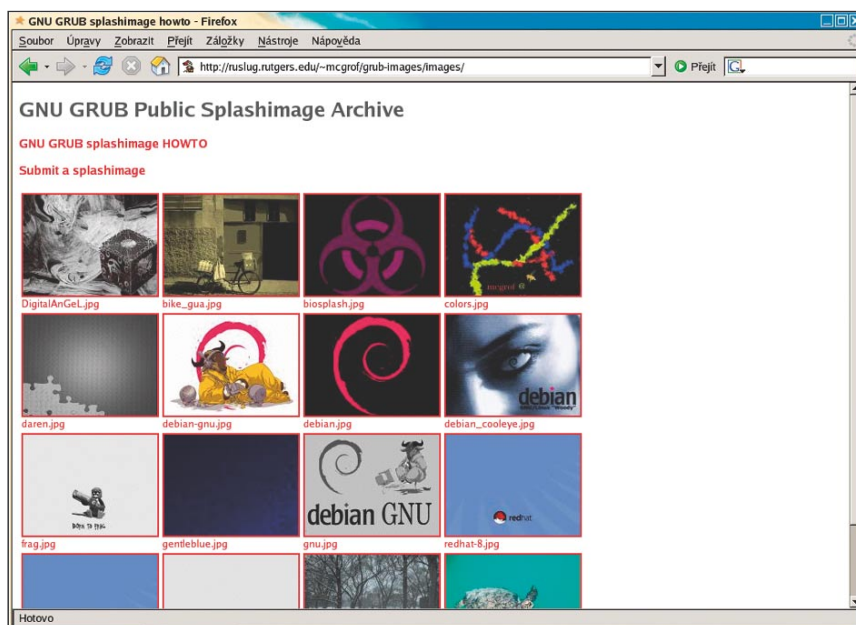
Mohlo by se zdát, že nyní bude potřeba upravit soubor `/boot/grub/grub.conf` a v řádku s volbou `splashimage` zadat novou cestu k souboru. Samozřejmě to můžeme udělat, například vložením tohoto řádku:

```
splashimage=(hd0,7)/grub/splashes/ ←
                               seachair.xpm.gz
```

Pokud však chceme sami trochu experimentovat a vyzkoušet si, které pozadí nám bude nejvíce vyhovovat, mohlo by být opakované restartování systému a upravování jednoho řádku únavné. Naštěstí to lze udělat i jinak. Po restartu systému a zobrazení menu GRUBu stiskneme klávesu `[c]`. Tím přejdeme do příkazového řádku GRUBu, který je rozsáhleji popsán v kapitole *Obsluha*. V tomto řádku nejdříve zadáme `root (hd0,7)` a následně stiskneme `[Enter]`. Pamatujme, že tímto příkazem určujeme



Obrázek 5. Obrázek pozadí bychom měli volit tak, aby bylo menu čitelné



Obrázek 6. Hezké obrázky můžeme vytvořit i s pouhými 14 barvami

diskový oddíl, se kterým se bude pracovat v dalších příkazech. Poté zadáme `splashimage /grub/splashes/seachair.xpm.gz` a opět stiskneme klávesu `[Enter]`. A už se nám zobrazuje vybrané pozadí. Klávesou `[Esc]` se ještě můžeme vrátit k menu, abychom si je prohlédli. Pokud si budeme chtít vyzkoušet jiná pozadí, opět stiskneme klávesu `[c]` a dále postupujeme stejně, jako v předchozím případě (nemusíme opět zadávat příkaz `root` – stačí jen samotné `splashimage`). Dobrá. Určitě jste ale už zapomněli, jak se jednotlivé soubory jmenovaly... Nevadí. Příkazový řádek GRUBu podporuje doplňování názvů, můžeme tedy zadat pouze `splashimage /` (nezapomeňme na značku `/`), a poté stisknout klávesu `[Tab]`. Zobrazí se nám seznam dostupných názvů adresářů a souborů. Takto přidáme pokračování cesty `(/grub/splashes/` nebo `/boot/grub/splashes/` – v závislosti na naší konfiguraci) a opět stiskneme klávesu `[Tab]` pro zobrazení názvů souborů s pozadími.

Až si nakonec jedno z nich vybereme, můžeme spustit systém (klávesou `[Esc]` se vrátíme až k menu a zvolíme položku s naším GNU/Linuxem) a následně dopsat příslušný řádek do souboru `/boot/grub/grub.conf`.

Tvorba vlastního pozadí

Kromě možnosti použít hotové soubory s pozadími, které můžeme najít na Internetu, by každý určitě rád věděl, jak si vytvořit své vlastní pozadí. Samozřejmě – nic tomu nebrání. Pokud chceme jen

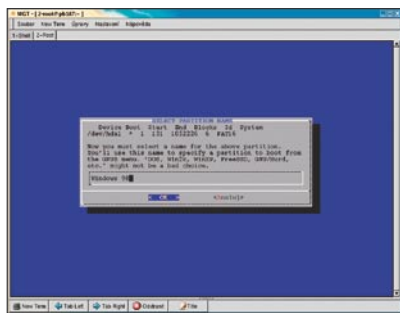
konvertovat nějaký obrázek na komprimovaný formát XPM, stačí zadat příkaz:

```
# convert -resize 640x480 -colors 14 ←
    picture.jpg picture.xpm
# gzip picture.xpm
```

Příkaz `convert` je součástí balíčku `ImageMagick`. Pokud není v našem systému přítomen, můžeme jej nainstalovat z distribučních CD nebo z repozitáře (analogickým způsobem, jako u instalace balíčku GRUB). Volba `-resize 640x480` mění rozlišení obrázku na požadovanou hodnotu a volba `-colors 14` redukuje škálu barev na čtrnáct. Názvy `picture.jpg` a `picture.xpm` patří dvěma souborům: vstupnímu (s konvertovaným obrázkem) a výstupnímu (s pozadím pro GRUB). Druhý příkaz provede komprimaci obrázku pomocí programu `Gzip`.

Uvedená metoda je rychlá, ale získané výsledky jsou často k pláči. Lepší řešení spočívá ve spuštění programu `Gimp`, načtení souboru, který chceme konvertovat a následném provedení změny velikosti obrazu (*Obrázek* → *Velikost obrázku*) a počtu barev (*Obrázek* → *Režim* → *Indexovaný*). Díky tomu můžeme experimentovat s různými nastaveními obrazu. Kromě toho se může ukázat, že bude nutné celý obrázek ručně upravit, než dospějeme k uspokojivému výsledku.

Jiným řešením je vytvořit prázdný obrázek o velikosti `640x480` se škálou 14 barev a pak si do něj nakreslit, co



Obrázek 7. Skript *Grubconf* nám může usnadnit konfiguraci GRUBu

budeme chtít. Ne každý však má umělecké nadání.

Volba metody je na nás. S vytvořeným souborem postupujeme stejně, jak je to popsáno v kapitole *Volba pozadí*.

Čtenáře by mohlo zajímat, proč komprimujeme pozadí před jejich využitím v GRUBu. Použít lze i soubory *XPM*, které nebyly komprimovány pomocí programu *Gzip*. Rozdíl pro nás bude v podstatě neznatelný, ale pro pořádek je lepší je komprimovat. Pro současné počítače je rychlejší načíst z disku komprimovaný (a tedy menší), soubor a poté jej rozbalit v paměti, než načíst z disku větší soubor, který nebyl podroben komprimaci.

Výběr barev

Kromě nastavení vlastního pozadí se můžeme pokusit změnit barvy písma používaného v menu. Jsou k tomu určeny volby *foreground* a *background*, které jsou nastavovány v úvodu souboru */boot/grub/grub.conf* (můžeme je umístit dokonce před volbu *splashimage*). Tyto volby přijímají jako parametry barvy zapsané v hexadecimálním formátu *RGB* (*Red*, *Green*, *Blue*). Budeme-li chtít získat bílá písmena na černém pozadí, použijeme volbu:

```
foreground fffffff
background 000000
```

Samozřejmě můžeme použít i jiné barvy, například červenou (*ff0000*), zelenou (*00ff00*), modrou (*0000ff*) a jiné. Pokud chceme pohodlným způsobem doladit barvy k našemu pozadí podle svého vkusu, můžeme použít stejnou metodu jako při volbě pozadí. Tentokrát po otevření příkazového řádku GRUBu zadáme odpovídající příkazy *foreground ffff00* a *background ff00ff* (samozřejmě při tom můžeme experimentovat s barvami).

Kromě toho zde existuje možnost změnit barvu podsvícení a pozadí

menu. Používá se k tomu příkaz *color*, který můžeme v konfiguračním souboru umístit před příkazy *foreground* a *background*. Seznam barev dostupných pro tento příkaz lze získat zadáním *help color* v příkazovém řádku GRUBu (a také v dokumentaci *GRUBu – info grub*). Příklad volání tohoto příkazu (poskytující celkem příjemné barvy) má podobu: *color cyan/blue white/blue*.

Obsluha

Pokud jsme GRUB nainstalovali na disk a vytvořili pro něj konfigurační soubor, mělo by nás během spouštění počítače uvítat menu GRUBu. Můžeme se v něm pohybovat pomocí šipek kurzoru a klávesou *[Enter]* vybrat položku, kterou chceme spustit. Kromě toho však máme k dispozici ještě několik jiných funkcí.

Úprava položky menu

Představme si, že chceme trochu experimentovat s ostatními parametry jádra. K tomuto účelu můžeme v konfiguračním souboru vytvořit novou sekci *title*. Lepší řešení však spočívá v úpravě parametrů jádra během startu systému. Jak to udělat? V menu označme naši distribuci GNU/Linuxu. V našem příkladu se jedná o řadu *Aurox Linux (2.4.22-1)*. Namísto stisku *[Enter]* však použijeme klávesu *[e]*. Zobrazí se nám obsah této sekce *title*. Mezi řádky se můžeme pohybovat podobně jako v menu. Pokud budeme chtít některý z řádků upravit, stiskneme *[Enter]*. Zkusme do řádku s příkazem *kernel* přidat volbu jádra *vga=791*. V našem příkladu by tento řádek vypadal takto:

```
kernel /vmlinuz-2.4.22-1 ro ↵
root=LABEL=/ vga=791
```

Změny potvrdíme klávesou *[Enter]*. Právě přidaná volba způsobí, že se systém nebude spouštět ve standardním textovém režimu, ale s použitím framebufferu nastaví rozlišení obrazovky na 1024x768 pixelů s více než 16 tisíci barvami.

Pokud budeme chtít přidat nový řádek, použijeme k tomu klávesu *[o]* nebo *[O]*. První z nich přidá nový řádek za právě označený řádek, druhá z nich před něj. Řádky, které nepotřebujeme, můžeme smazat pomocí klávesy *[d]*.

Až budeme s provedenými změnami spokojeni, stiskneme klávesu *[b]*, čímž

editovanou položku spustíme s provedenými změnami. Pokud jsme si změny rozmysleli, můžeme se pomocí klávesy *[Esc]* vrátit k hlavnímu menu.

Příkazový řádek

Ještě větší možnosti nám poskytuje příkazový řádek dostupný v GRUBu. Již jsme jej používali v kapitole *Volba pozadí*, nyní se mu ale budeme věnovat podrobněji. Pro přístup k němu použijeme v menu GRUBu klávesu *[c]*. Máme zde k dispozici celou řádu více či méně užitečných příkazů. Jejich seznam získáme jednoduše stiskem klávesy *[Tab]*. Jistě si všimnete, že se na seznamu nachází mnoho příkazů, které již znáte z tvorby menu – *chainloader*, *kernel*, *initrd*, *root* a další. Vidíme tedy, že příkazový řádek lze použít také pro spouštění systémů nainstalovaných na našem počítači. Kdybychom nevytvořili konfigurační soubor */boot/grub/grub.conf*, mohli bychom při startu počítače používat jen příkazový řádek GRUBu. Kdybychom v této situaci chtěli spustit *Memtest86+* (nainstalovaný dle výše uvedených pokynů), museli bychom zadat následující příkazy:

```
root (hd0,7)
kernel /memtest86+-1.40.bin
boot
```

Jediný rozdíl mezi obsahem *grub.conf* a příkazy zadávanými při ručním spouštění systému spočívá v tom, že byl v druhém případě přidán příkaz *boot* způsobující nahrání určeného jádra.

Je užitečné si pamatovat, že během zadávání příkazů a názvů souborů můžeme využít automatické doplňování – stačí stisknout klávesu *[Tab]* a budou zobrazeny všechny možnosti.

K čemu se nám ještě může hodit příkazový řádek? Pomocí příkazu *cat* můžeme zjistit obsah libovolného souboru, např. *cat /etc/fstab*. Vyplývá to ze skutečnosti, že ten, kdo má přístup k příkazovému řádku GRUBu, má právo číst prakticky jakýkoliv soubor (např. */etc/shadow* obsahující zakódované heslo). S tím si ale budeme lámat hlavu později, v kapitole *Zabezpečení*. Bez ohledu na možná rizika se nám příkaz *cat* může hodit, pokud budeme chtít během experimentování prověřit nějaká nastavení.

Dalším užitečným příkazem je *find* – dovoluje nám zjistit, na kterém dis-

kovém oddílu se nachází daný soubor. Měli bychom zadávat absolutní cestu k souboru, např. `find /boot/grub/stage1` nebo `find /grub/stage1` – tento příkaz nám vrátí název diskového oddílu, který bychom měli zadat v řádce `root` v konfiguračním souboru `GRUBu`.

Stojí za to seznámit se i s jinými příkazy. Velice se hodí nápověda, kterou můžeme získat zadáním `help název_příkazu`.

Příkazový řádek `GRUBu` lze spustit také z účtu uživatele `root` zadáním příkazu `grub`. V tomto případě budeme mít podobné možnosti, jako ve výše popísaném příkazovém řádce, ale samozřejmě některé příkazy nebudeme moci používat (týká se to například příkazu `splashimage`).

Zabezpečení

Menu nabízené `GRUBem` je velice pohodlné, ale občas poskytuje až příliš široké možnosti. Jak jsme se přesvědčili v předchozí kapitole, uživatel, který má přístup ke `GRUBu`, může libovolně měnit parametry volání systému a dokonce získat přístup k chráněným souborům. `GRUB` ale naštěstí dovoluje tento přístup zablokovat.

Slouží k tomu heslo, které brání v přístupu k editaci položky v menu, a také k příkazovému řádce. Pokud chceme, může být i spuštění jednotlivých systémů podmíněno zadáním hesla.

Chceme-li náš systém tímto způsobem zabezpečit, budeme k tomu samozřejmě potřebovat práva uživatele `root`. Nejdříve si vygenerujeme heslo. Spustíme příkazový řádek `GRUBu` – `grub`. Tentokrát použijeme příkaz `md5crypt`. Po jeho zadání budeme dotázáni na heslo. Zadáme je (snažme se, ať není příliš snadno uhádnutelné) a stiskneme `[Enter]`. Zobrazí se nám řádek se zakódovaným heslem.

Nyní budeme editovat náš konfigurační soubor (např. `vim /boot/grub/grub.conf`) – na jeho začátek přidáme následující řádek:

```
# password --md5 $1$ikE1sTM5yM1MYrx3Ms7
```

Namísto zde zadané hodnoty je samozřejmě třeba vložit výsledek provedení příkazu `md5crypt`. Použití hodnoty z tohoto příkladu není dobrý nápad, jelikož se jedná o zakódované slovo `password`. Jelikož zde uchováváme heslo, měli

bychom se ujistit, že soubor nebude možné snadno prohlížet. K tomuto účelu odpovídajícím způsobem upravíme přístupová práva k souboru:

```
# chmod 600 /boot/grub/grub.conf
```

Tato změna postačí k tomu, aby při spuštění počítače byly v menu `GRUBu` k dispozici jen dvě možnosti – spuštění jednoho ze systémů anebo stisknutí klávesy `[p]`. Pokud po jejím stisknutí zadáme správné heslo (to, které jsme uvedli v příkazu `md5crypt`), získáme přístup k možnostem popisovaným v kapitole *Obsluha*.

Uživatel však stále může spustit libovolnou položku z menu, a to nám nemusí vždy vyhovovat. Jelikož už jsme určili heslo, stačí nyní v sekci `title`, kterou chceme zablokovat, umístit příkaz `lock`. Musí se nacházet hned pod řádkem `title`, jelikož uživatel může provádět všechny příkazy až do okamžiku, kdy narazí na `lock`. V tomto okamžiku bude dotázán na heslo a dokud je správně nezadá, nebude moci pokračovat prováděním dalších příkazů. Jak tedy vidíme, umístit příkaz `lock` na konec sekce `title` zpravidla postrádá smysl – lepší bude vložit jej na začátek.

Je třeba poznamenat, že toto zabezpečení není dokonalé. Nelze počítat s tím, že nás zaheslování `GRUBu` ochrání před nekalými záměry lidí, kteří mají bezprostřední přístup k počítači. V každém případě však může přispět k pohodlí používání počítače a zabezpečit nás před méně zkušenými uživateli.

Usnadnění konfigurace

Pokud nechceme ručně upravovat konfigurační soubor `GRUBu`, můžeme použít jeden z konfigurátorů. Nejlepším řešením se zdá být modul `Webminu` dostupný v sekci *Hardware->Zavaděč systému GRUB*. Kromě toho máme k dispozici skript `Grubconfig`, který pracuje v textovém režimu a program `GrubConf` určený pro grafické prostředí. První z nich je málo intuitivní, má ale několik zajímavých voleb a přitom nevyžaduje instalaci – stačí rozbalit archiv a můžeme jej spustit.

Ty, kteří si jádro kompilují sami, pravděpodobně zaujme program `Install Kernel`, který nás nejen zbavuje povinnosti provádět řadu příkazů, ale také automaticky aktualizuje konfiguraci

zavaděčů (kromě `GRUBu` podporuje také `LILO`).

Pokud si chceme úplně zjednodušit nastavování pozadí menu `GRUBu`, můžeme použít program `GrubTk`. Stačí v něm zadat jen název souboru s obrázkem a program se postará o zbytek.

Pokud vás zajímá, k čemu by se ještě `GRUB` dal použít, nainstalujte si hru `GRUB Invaders`. Je spuštěna `GRUBem` a pro své fungování nevyžaduje žádný operační systém.

Závěr

Mohlo by se zdát, že zavaděč systému je jen jednoduchý nástroj umožňující zvolit, který systém má být spuštěn. Přesto se ukazuje, že jde o program, který si můžeme poměrně snadno přizpůsobit svým potřebám. S jeho pomocí lze spustit nejen systém, ale také jiné programy, např. `Memtest86`. `GRUB` může fungovat také jako ochranná hradba bránící ostatním uživatelům v přístupu k našemu systému – my se však přitom nemusíme zříkat pohodlné obsluhy. Navíc nás `GRUB` může uvítat naším oblíbeným obrázkem. Stojí za to prostudovat jeho dokumentaci a seznámit se i s jeho dalšími funkcemi. Δ

Na Internetu:

- Domácí stránka `GRUBu`:
<http://www.gnu.org/software/grub/>
- GNU `GRUB` splashimage howto:
<http://ruslug.rutgers.edu/~mcgrof/grub-images/>
- Různá pozadí pro menu `GRUBu`:
<http://vision.featia.net/linux/grubsplash/>
- Domácí stránka `Memtest86`:
<http://www.memtest86.com/>
- Domácí stránka `Memtest86+`:
<http://www.memtest.org/>
- Domácí stránka skriptu `Grubconfig`:
<http://www.tux.org/pub/people/kent-robotti/looplunix/index.html>
- Domácí stránka programu `GrubConf`:
<http://grubconf.sourceforge.net/>
- Domácí stránka programu `Install Kernel`:
<http://installkernel.tripod.com/ik/index.html>
- Domácí stránka programu `GrubTk`:
<http://freshmeat.net/projects/grubtk/>
- Domácí stránka hry `GRUB Invaders`:
<http://www.erikyyy.de/invaders/>