Prosinec 1997

Linuxové noviny

Úvodem

Pavel Janík ml., 8. prosince 1997

Právě jste začali číst již třetí číslo měsíčníku Linuxové noviny. Grafická úprava nového čísla se příliš neliší od čísla předchozího, což je dobrým znamením, že Linuxové noviny dostávají svoji tvář. Jedinou výraznější změnou, které si jistě všimnete, je přítomnost českého překladu anglických textů.

Dostali jsme několik poznámek i k obsahu Linuxových novin. Za všechny děkujeme a budeme se snažit vyhovět.

I přes veškerou snahu se nám bohužel nepodařilo udržet rozsah prosincového čísla pod deset stránek (za což bych se chtěl omluvit hlavně těm, kteří mi psali, že se jim Linuxové noviny nevejdou na nástěnku).

A nyní již k vlastnímu obsahu. V prosincovém čísle naleznete druhé pokračování článku Jana „Yenyi“ Kasprzaka ➡Úvod do RPM o systémech správy balíků. Leoš Bitto nám sdělí své zážitky z instalace Red Hat Linux 5.0. Pokud se chcete dovědět něco zajímavého o distribuci Debian, nezapomeňte si přečíst článek ➡Debian GNU/Linux – Users' for users. Jeho autorem není nikdo jiný než prozatím jediný český oficiální maintainer Debianu – Milan Zamazal.

Jan Pazdziora nás seznámí s možnostmi jazyka Perl a archívu ➡CPAN – Comprehensive Perl Archive Network.

Vzhledem ke vzrůstajícímu zájmu uživatelů operačního systému Linux o databáze jsme pro vás připravili článek o relačním databázovém serveru ➡Adabas D pro Linux.

Jistě víte, že procesory Intel Pentium obsahují chybičky, ale víte opravdu jaké a jak se jim bránit? Pokud ne, Jan Kasprzak vám jistě poradí v článku nazvaném ➡Chyba v procesorech Intel Pentium – softwarové řešení. Na čem zrovna pracuje Linus Torvalds a kterou distribuci používá na svých počítačích se dovíte v článku ➡Interview s Linusem Torvaldsem.

Prosincové číslo také obsahuje návrh stanov Českého sdružení uživatelů operačního systému Linux (CZLUG).

Linuxové noviny toho ale obsahují mnohem více, proto vás již déle nebudu zdržovat od zajímavého čtení.

Free software a svobodné myšlení

Pavel Hanák, 5. prosince 1997

Poslední tři měsíce byly hodně ve znamení myšlenek free softwaru, Linuxu, GNU apod. Vznikla Česká nadace pro podporu free softwaru. Jaká bude její budoucnost, to se teprve ukáže. Vše závisí jen na lidech, tedy na nás, na tom, jak budeme schopni podporovat ideje, které jsou s free softwarem spojeny. V této souvislosti se mi stále zřetelněji vyjevuje paralela mezi free softwarem a free myšlením. My všichni víme, že slůvko „free“ neznamená primárně „zdarma“. Mnohem více jde o svobodu sdílení myšlenek, které se zformovaly v konkrétní programy či programový kód. Z toho mi ale také vyplývá, že má-li mít free software příznivou budoucnost, je nutné, aby si každý příznivec free softwaru uvědomil, že je nezbytné zachovat si co nejsvobodnější myšlení.

Nyní by se mohlo ozvat: „To je přece jasné, to snad není nutno nám připomínat. Taková výzva by se spíše měla obracet na ten dav, který jde slepě za vším, co mu předhodí Bill Gates.“ Není to však tak docela pravda. I nám se často stává, že přestaneme uvažovat svobodně. Co tím tedy míním a kde se často skrývá nebezpečí?

Nebezpečí se skrývá v nás, v našich slabostech, návycích, stereotypech, pohodlnosti, lenosti, ješitnosti apod. Dalším významným slůvkem v této souvislosti je slovo „strnulost“. Snad bude na tomto místě vhodné uvést příklad.

Mějme člověka, který pochopí, že není něco v pořádku ve způsobu, kterým se v dnešní době ve světě prosazuje a uplatňuje software. Že tento způsob má nakonec ten důsledek, že se často neuplatní ty nejlepší myšlenky, ba naopak, že v podstatě dochází k potlačování kvalitativně lepšího softwaru, aby nemohl ohrožovat pozice kvalitativně horšího softwaru od komerčně silné firmy. Je přirozené, že takový člověk hledá východisko, hledá alternativu. Dnes už tuto alternativu může nalézt, protože existuje free software. Najde si tedy ty programy, které potřebuje, dle svých možností se snaží i rozvíjet vlastnosti těchto programů a je spokojen. Ale právě zde se už skrývají zárodky nebezpečí. Původně „široké“ a volné myšlení se zúží na jím v praxi poznané věci, původní snaha o poznávání nového se hodně redukuje, někdy až ustrne, s pokračujícími zkušenostmi se i připojují „železné košile“ návyků, a připojí-li se k tomu ještě naše ješitnost, není nic neobvyklého, že se třeba ze dvou lidí, kteří na počátku hledání zastávali podobná stanoviska, stanou posléze protivníci jenom proto, že jeden si vybral jednu jednotlivost a druhý nějakou jinou. Přitom zapomínají, že vycházejí z jednoho základu a že celý problém spočívá pouze v tom, že zapomněli na celek a nejvyšší prioritu přisoudili detailu. Lpění na tomto detailu pak člověku brání v rozletu jeho myšlení a původně pohyblivé myšlení se stává strnulejší a strnulejší.

Vrátím se opět k free softwaru, jehož nejpodstatnějším aspektem je podle mého názoru samotná jeho ideová podstata, jak jsem uvedl. Bylo by dobré, kdybychom se dokázali vyhnout procesu, který jsem popsal ve výše uvedeném příkladu a který se v lidských dějinách již mnohokrát opakoval (např. u křesťanství se jisté skupiny lidí lpěním na bezvýznamných maličkostech, různých rituálech, úkonech apod., dostali až k nenávistným náboženským válkám, ačkoli původně vycházeli ze stejného základního principu; a to vše proto, že zapomněli, nebo kvůli ješitnosti chtěli zapomenout na základní stavební kameny a myšlenky).

Jsem přesvědčen, že free software je nám pomocí nejen ryze praktickou (ve smyslu užitečnosti programů jako Linux, Emacs, TeX, XFree atd.), ale že nám má být základní idea free softwaru stále i připomenutím toho, abychom své myšlení udržovali svěží.

Linuxové noviny připravuje a vydává Česká nadace pro podporu free softwaru

Měsíc v comp.os.linux.announce

Pavel Janík ml., 1. prosince 1997

Jeremy D. Impson (jdimpson@syr.edu) vytvořil program urlmon [1], který dokáže monitorovat změny na WWW stránkách i na FTP serverech. Obsahuje dokonce skript, který umožňuje monitorovat všechny servery obsažené v záložkách (bookmark) prohlížeče Netscape Navigator.

Používáte pro čtení Usenet news prohlížeč Netscape Navigator a zdá se vám, že jeho ovládání není příliš user-friendly? Potom byste se měli podívat na Netscape Newsgroup selector [2], jehož autorem je Bob van der Poel (bvdpoel@awinc.com). Program je napsán v tcl/tk a umožňuje jednoduché přihlášení do news skupin.

Tim Berger (timb@transmeta.com) vytvořil anti-spamovací patch pro sendmail 8.8.8. Jmenuje se spamcan a můžete jej najít na adrese [3]. Patch umožňuje eliminovat spam pomocí regulárních výrazů.

Pokud jste systémovým administrátorem nějakého počítače, jistě uvítáte novou verzi programu xosview [4]. Xosview umožňuje sledování využití procesoru, paměti, swap, sítě, přerušení a sériových portů. Samozřejmostí je sledování „load average". Nová verze je kompatibilní s jádry řady 2.0 i 2.1.

2. listopadu vyšla nová verze XEmacsu s číslem 19.16, která obsahuje pouze opravy chyb z 19.15. Pokud již používáte XEmacs 20, tak vás jistě bude zajímat, že XEmacs 20.3 by měl vyjít ještě tento rok. XEmacs naleznete na FTP serveru [5].

Emarit Ranu (drranu@holly.ColoState.EDU) vytvořil jednoduchý skript, který vypisuje aktuální adresáře běžících procesů. Skript se jmenuje procdirs a najdete na adrese [6].

Marcin Dalecki oznámil novou verzi svého prohlížeče GNU-info dokumentace. Program minfo 1.7.19 je k dispozici i staticky slinkován s knihovnou Motif. Naleznete jej na adrese [7].

Cheap Bytes nabízí CD s distribucí Slackware 3.4 za neuvěřitelnou cenu $1.99. Více informací naleznete na stránce [8].

H. J. Lu (hjl@gnu.ai.mit.edu) oznámil novou verzi balíku binutils. Nová verze má číslo 2.8.1.0.15 a obsahuje hlavně opravy chyb z předchozí verze. Nově je obsažena i podpora pro Linux/ARM.

Dave Airlie (david.airlie@ul.ie) oznámil vytvoření PAM modulu pro autentikaci v doméně pomocí NT serveru. Modul umožňuje uchovávat pouze jedna hesla pro NT a Linux. Naleznete jej na adrese [10].

Společnost MathTools Ltd. oznámila novou verzi programu Matcom V3, který umožňuje kompilovat zdrojové texty pro Matlab do C++. Program si můžete nahrát na adrese [11].

Jeff Garzik (jeff.garzik@spinne.com) vytváří stránku, která se zabývá hrami pro X-Window System. Její adresa je [12].

Karsten Ballueder (karsten@phy.hw.ac.uk) oznámil novou verzi zálohovacího balíku KBackup, který je podle svých autorů jedním z nejvýkonnějších. KBackup umí komprimovat i šifrovat archívy, podporuje „multi volume" i „inkrementální zálohy". KBackup je nyní k dispozici na adrese [13] za podmínek uvedených v GPL.

5. listopadu byla oznámena první beta verze balíku Matrice W3 System [14], který je kompletním řešením prezentací na WWW. Systém je založen na serveru Apache a databázovém serveru MySQL.

Michel Eyckmans (eyckmans@imec.be) vytvořil další verzi programu Xautolock [15]. Tento program sleduje aktivity uživatele a po určité době nečinnosti automaticky spustí zvolený program. Umožňuje také automatické odhlášení po stanovené době.

Linux Counter [16] je jediným seriózním projektem zabývajícím se sečítáním uživatelů operačního systému Linux. V současné době je zaregistrováno více než 53 tisíc uživatelů z více než 130 zemí světa. Zaregistrujte se také!

Andrew Weeks (ccsaw@bath.ac.uk) oznámil vytvoření programu, který umí konvertovat True Type fonty do postscriptu. Program naleznete na adrese [17].

Nová stránka Linux Network Address Translation je na adrese [18]. Naleznete zde informace o různých druzích NAT – 1:1, 1:x, x:1 a použití k zlepšení bezpečnosti. Stránka obsahuje také odkazy na další zajímavá místa o IP Masquerading a přesměrování portů.

Allin Cottrell (cottrell@wfu.edu) vytvořil novou emailovou konferenci, která se bude zabývat problémy s novou knihovnou glibc. Přihlásit se můžete zasláním textu `subscribe glibc-linux` na adresu: `Majordomo@ricardo.ecn.wfu.edu`.

Roberto Alsina (ralsina@unl.edu.ar) vytvořil podporu jazyka C pro knihovnu Qt. Projekt je zatím v počátcích, ale zdá se, že je i použitelný. Autor uploadoval balík na FTP servery *sunsite.unc.edu* a *ftp.troll.no*.

Woven Goods jsou již ve verzi 1.3. Jedná se o balík aplikací určených pro práci s World Wide Web. Balík obsahuje např. Netscape Communicator 4.03 a Acrobat Reader. Woven Goods jsou nakonfigurovány pro Slackware 3.4. Kompletní balík naleznete na adrese [19].

Gaspar Sinai (gsinai@gol.com) vytvořil editor yutex, který podporuje Unicode. Naleznete jej na adrese [20].

Stephan Schiessling oznámil novou verzi sofie [21] – 3d engine pro tvorbu her. Distribuovaný balík obsahuje i demo verzi hry.

Emacspeak 98 je novou verzí balíku, který umožňuje slepým přístup nejen k lokálním informacím, ale i k informacím uloženým na síti Internet. Emacspeak 98 (Labrador) podporuje Emacs 20. Balík najdete na adrese [22].

V Bolívii (doména `.bo`) vznikla první emailová konference zaměřená na Linux.

Představitel projektu Debian – Bruce Perens (bruce@debian.org) vyzval všechny uživatele operačního systému Linux, aby se podíleli na vývoji distribuce, která již byla ve vesmíru. Pokud se chcete stát vývojářem distribuce Debian, podívejte se na stránku [23].

> Can you get your operating system fixed when you need it?
>
> ---
>
> Můžete si upravit váš operační systém, pokud to potřebujete?

Lars Wirzenius oznámil další verzi System Administrator's Guide, tentokráte s číslem 0.6. Přibyla kapitolka o `initu`. Novou verzi si můžete stáhnout ze zrcadel Linux Documentation Project.

Společnost Xi Graphics [24] oznámila novou verzi X-serveru Accelerated-X verze 4.1. Nová verze podporuje více než 500 grafických karet včetně nejnovějších produktů

ATI, Matrox nebo S3. X-server podporuje jak PseudoColor, HighColor, tak i TrueColor módy.

Alex Shnitman (alexsh@linux.org.il) oznámil vytvoření programu news2mail, který umožňuje číst Usenet news přes elektronickou poštu. Balík naleznete na adrese [25].

Charles Booher (ctb1@ix.netcom.com) oznámil vytvoření utilit pro práci s RSA. Najdete je na adrese [26].

Marjorie Richardson (mlr@ssc.com) oznámila, že listopadové číslo Linux Gazette nevyjde z důvodů personálních problémů. Prosincové číslo by již mělo vyjít.

David Mandelstam (dm@sangoma.com) ze společnosti Sangoma Technologies Inc. oznámil driver Wanpipe 2.0.0, který podporuje karty Sangoma S508 a Sangoma S508/FT1. Podporuje interní routovací karty Wanpipe pro WAN sítě Frame Relay, PPP a X.25. Bližší informace naleznete na adrese [27].

Návrh stanov CZLUG (Czech Linux Users' Group)

Jan Kasprzak, Petr Šnajdr, Pavel Janík ml., 7. prosince 1997

V současné době připravujeme založení sdružení uživatelů operačního systému Linux (CZLUG). Protože chceme všem dát možnost vyjádřit se k podobě a činnosti budoucího sdružení, uveřejňujeme i v Linuxových novinách kompletní znění připravovaných stanov sdružení. Stanovy jsou k dispozici i na WWW na adrese [28]. Na adrese [29] je také WWW formulář sloužící ke zjištění zájmu o různé druhy členství v CZLUG.

Prosím udělejte si chvíli čas a pozorně si přečtěte návrh stanov a sdělte nám veškeré své náměty a postřehy.

Poslání CZLUG

CZLUG je neformální Linux Users' Group v České republice při České nadaci pro podporu free software (dále jen CZFSF). Základním posláním je sdružovat vývojáře, správce a uživatele operačního systému Linux, vytvářet vhodné podmínky pro rozvoj, využití a propagaci operačního systému Linux, pořádat vzdělávací akce pro své členy i širokou veřejnost jako jsou přednášky, odborné semináře atd., a to samostatně nebo v návaznosti na akce CZFSF.

Základní ustanovení

1. CZLUG je dobrovolné sdružení uživatelů operačního systému Linux pod záštitou CZFSF.
2. CZLUG není právním subjektem, ale organizačně samostatným projektem CZFSF pro podporu a rozvoj operačního systému Linux.
3. Oficiální název je České sdružení uživatelů Linuxu (anglicky Czech Linux Users' Group) a jeho zkrácená verze je CZLUG.

Činnost sdružení

1. Sdružuje uživatele operačního systému Linux v České republice.
2. Spolupracuje s institucemi a organizacemi využívajícími operačního systému Linux a obdobně zaměřenými skupinami.
3. Pořádá školení, diskuse, exkurze, konference, sympozia a semináře. Vyvíjí expertní, publikační, vydavatelskou činnost.
4. Podporuje nebo zadává vytváření programového vybavení pro operační systém Linux.
5. Podporuje nebo přímo spolupracuje na vytváření archívů software (zejména operačního systému Linux) a jejich údržbu.
6. Napomáhá rozvoji a propagaci operačního systému Linux.
7. Otázky finanční, sponzorství a propagaci řeší pouze ve spolupráci s CZFSF.

Organizační struktura CZLUG

1. Nejvyšším orgánem CZLUG je valné shromáždění řádných a zástupců kolektivních členů.
2. Výkonným orgánem sdružení je výbor CZLUG (dále jen „výbor") volený v počtu 3 až 7 členů. Výbor je volen a odvoláván valným shromážděním a to nadpoloviční většinou. Výbor musí svolávat valné shromáždění minimálně jednou ročně s přesahem maximálně dvou měsíců do následujícího roku. Funkční období výboru je nejvýše tříleté.
3. Výbor volí ze svého středu předsedu, který pověřuje jednotlivé členy vybranými úkoly tak, aby bylo respektováno následující:
   (a) operativní a efektivní činnost CZLUG.
   (b) základní myšlenka a stanovy CZLUG.
4. Ve výboru je možno dohodnout rozhodnutím výboru kratší funkční období předsedy, nejméně ale 1 rok.
5. V rámci možností CZLUG je možno proplácet nutné výdaje na činnost výboru i dalších členů.
6. Výbor řídí CZLUG v období mezi valnými shromážděními a za svou činnost je odpovědný valnému shromáždění.
7. Minimálně jednou ročně dává výbor všem svým členům a statutárnímu zástupci CZFSF informaci o své činnosti a rámcových plánech do budoucna.
8. Výbor je oprávněn nadpoloviční většinou hlasů výboru svolat z vážných důvodů vyjímečné valné shromáždění i více než 1x do roka.
9. Za CZLUG jedná jeho předseda. Oficiální vystupování CZLUG je možné pouze za spolupráce s CZFSF.
10. Výbor řeší veškeré otázky na zasedání výboru. Z tohoto zasedání je nutno pořídit zápis, který podepisuje člen pověřený jeho zpracováním a předseda.
11. Výbor je oprávněn zvolit i jinou než osobní formu zasedání výboru. Příslušný zápis z jednání pak členové výboru ověří předem dohodnutou formou elektronického podpisu.
12. O závažných otázkách se hlasuje v rámci výboru. K hlasování se vyžaduje přítomnost předsedy. Hlasování je platné, když se ho zúčastní minimálně 50% členů výboru. Výbor může stanovit i elektronickou formu hlasování. Příslušná výměna veřejných klíčů musí být provedena minimálně 7 dní před hlasováním.

13. Hlasování je možné i v rámci valné hromady. K platnosti hlasování je potřebná účast minimálně 50% přítomných členů. Výbor může povolit i hlasování elektronickou formou. Otázky pak musí být zveřejněny alespoň 7 dní před konáním valné hromady spolu s pravidly hlasování.

Členství v CZLUG

1. CZLUG má své členy řádné, kolektivní a spolupracující převážně z České republiky, ale i ze zahraničí (například ze Slovenské republiky).
2. Řádným a spolupracujícím členem CZLUG se může stát každý, kdo souhlasí se stanovami CZLUG, podá příslušnou přihlášku a zaplatí příslušný členský příspěvek na jeden rok.
3. Kolektivním členem se může stát každá instituce nebo organizace, která souhlasí se stanovami CZLUG, podá příslušnou přihlášku a zaplatí příslušný členský příspěvek.
4. Řádní, spolupracující i kolektivní členové jsou přijímáni usnesením výboru CZLUG na základě přihlášky určující druh členství.
5. Člen CZLUG je každý, kdo je přijat za člena CZLUG a plní řádně členské povinnosti.
6. Členství v CZLUG zaniká vystoupením nebo vyloučením člena. Vyloučit člena může výbor sdružení pokud pro toto hlasuje minimálně polovina členů CZLUG a to pro vážné důvody, zejména neplnění povinností člena sdružení.

Práva člena CZLUG

1. Na vyžádání a dle možností CZLUG získat volně šiřitelný software šířený v rámci CZLUG nebo sítě Internet.
2. Přednostní právo se účastnit konferencí a obdobných akcí pořádaných CZLUG a CZFSF.
3. Možnost publikovat články v rámci případné publikační činnosti CZLUG.
4. Přednostní právo na získání publikací nebo materiálů vzniklých činností CZLUG nebo CZFSF.
5. Řádní a kolektivní členové mají právo na slevy při získávání těchto produktů, jsou-li slevy stanoveny.
6. Právo hlasovat na valném shromáždění členů CZLUG.
7. Zástupce kolektivního člena má právo hlasovat na valném shromáždění s váhou tři hlasy.
8. Řádný člen má právo být volen do výboru CZLUG.
9. Všichni členové mají právo být pravidelně (alespoň jednou ročně) informováni o činnosti CZLUG. Tyto informace jsou šířeny především pomocí sítě Internet nebo v závažných případech rozesílány poštou.

Povinnosti člena CZLUG

1. Platit členský příspěvek ve výši a době stanovené valnou hromadou.
2. Informovat správce databáze členů CZLUG tak, aby tato databáze byla stále aktuální.
3. Informovat o důležitých skutečnostech (odjinud než z CZLUG) výkonný výbor.
4. Povinností řádného a kolektivního člena navíc je propagovat vhodně operační systém Linux a aktivně získávat další členy.
5. Povinností řádného člena je pomáhat dle svých možností svému okolí s provozem operačního systému Linux.
6. Kolektivní člen je povinen umožnit svým zaměstnancům nebo členům přístup k operačnímu systému Linux a zasadit se v rámci svých možností o co nejširší nasazení Linuxu na vlastní výpočetní technice.

Hospodaření CZLUG

Hospodářská činnost CZLUG je zajišťována:

1. Z příspěvků členů.
2. V návaznosti na hospodaření CZFSF.

Výši členských příspěvků stanoví hlasování na valné hromadě CZLUG. Neproběhne-li hlasování nebo nedojde-li ke schválení nové výše členských příspěvků, zůstávají tyto údaje stejné jako v předchozím období.

Závěrečná ustanovení

1. Stanovy Czech Linux Users' Group je možno změnit pouze na základě hlasování valného shromáždění.
2. CZLUG zaniká usnesením valného shromáždění vysloví-li se pro tento návrh alespoň dvě třetiny všech členů.

Příloha ke stanovám CZLUG – Výše členských příspěvků

1. Členské příspěvky spolupracujícího člena činí 20 Kč ročně.
2. Členské příspěvky řádného člena činí 220 Kč ročně, pro studenty 150 Kč ročně.
3. Členské příspěvky kolektivního člena činí 1000 Kč ročně, v případě škol a neziskových organizací 500 Kč ročně.

Red Hat Linux 5.0 – konečně?

Pavel Janík ml., 19. listopadu 1997

18. listopadu se v tichosti změnila úvodní stránka WWW serveru společnosti Red Hat [9]. Žádná announce, dokonce ani v *redhat-announce*.

Již dlouho očekávaná verze 5.0 distribuce Red Hat je na světě (tedy bude). Společnost Red Hat oznámila, že nová verze bude uvolněna první prosincový den (nebo 31. listopadu?).

Red Hat Linux 5.0 by měl usnadnit instalaci novým uživatelským prostředím pro rozdělení pevného disku, novým rozpoznáváním a konfigurací technického vybavení počítače, podporou zvuků a *Kickstart* instalací, která je vhodná zvláště pro systémové administrátory.

The remarkably effective and efficient Internet-based cooperative development model under which Red Hat Linux is developed, improved, and enhanced, continues to expand rapidly.

Pozoruhodně výkonný a působivý způsob, jakým se díky spolupráci prostřednictvím Internetu vyvíjí, zlepšuje a rozšiřuje Red Hat Linux, se dále rychle rozvíjí.
Bob Young, prezident společnosti Red Hat Software Inc.

Red Hat Linux 5.0 je založen stejně jako Mustang i Thunderbird na nové knihovně C (Glibc), která lépe podporuje internacionalizaci a obsahuje podporu threadů. Nový Red Hat obsahuje také několik předkonfigurovaných správců oken.

Official Red Hat Linux 5.0 má navíc oproti 5.0 internetový vysílací server pro Real Audio, komerční X-server společnosti Metro a zálohovací software BRU 2000-PE.

Red Hat Linux 5.0 (Hurricane) – první dojmy

Leoš Bitto, 7. prosince 1997

Jak jste si už možná všimli, byla vypuštěna do světa nová verze populární linuxové distribuce Red Hat. Minulá verze nesla číslo 4.2 a označení Vanderbilt, nová má číslo 5.0 a označení *Hurricane*. V čem že se tedy liší? Začněme instalací.

Hned při nabootování z příslušné diskety (lidé vlastnící příslušný CD disk mohou bootovat přímo z něj) uvidíte, že se skutečně cosi změnilo. Až příliš stručná hláška z Vanderbiltu byla nahrazena přímo sofistikovaným menu systémem o celých šesti stránkách. Je pravda, že většina toho byla dříve v extra dokumentaci také, ale přece se před instalováním nebude někdo zdržovat čtením dokumentace. Okamžitě mne zaujala volba F2, Kickstart. Podle dokumentace má umožňovat naprogramovat dopředu celou instalaci tak, že nakonec pouze přijdete k počítači, nabootujete z příslušné diskety a jen se koukáte. Výhoda je jasná: jednou si dáte práci s vytvořením potřebného instalačního skriptu a pak ho můžete použít na spoustě počítačů. V Red Hat Linuxu 4.2 se mi rozhodně nelíbilo při každé instalaci vybírat více méně to samé. Pokud mne od toho Kickstart v Red Hat 5.0 osvobodí, sláva mu!

Během bootování jsem s povděkem zaregistroval, že je použito jádro Linuxu 2.0.32. Oproti verzi 2.0.31 a starším má totiž opravené dvě zásadní chyby. Za prvé je odolné proti tzv. „teardrop“ útoku, kdy posláním špatného IP paketu bylo možno Linux (a spoustu dalších OS a rádoby-OS) zastavit či restartovat. A za druhé obchází tzv. „F00F“ chybu ve všech procesorech Intel Pentium a Intel Pentium MMX, která jakémukoliv uživateli dává možnost procesor zastavit, což je rozhodně špatně.

Odkud že prý budeme instalovat? Z lokálního disku, kam jsme instalaci nějakým způsobem dostali (můj případ)? Z CDROM? Po síti (FTP, NFS)? Oproti Red Hat 4.2 zmizela volba instalovat po síti pomocí protokolu SMB, která mi stejně připadala taková nějaká divná. Z počítače s Windows 95 či NT se mi to stejně nikdy nepovedlo, šlo to jen z unixové Samby. Ale z unixu už to můžu instalovat pomocí FTP nebo NFS. Takže té volby není zas taková škoda.

Další věc, co je v Red Hat 5.0 navíc je „klikací fdisk“, který se jmenuje *Disk Druid*. Důvod je jasný – klasický linuxový fdisk je pro nezkušeného uživatele docela odstrašující program, takže jeho funkce byly zabaleny do okének a uživatel se třeba hned nelekne a neuteče. Já jsem ale s radostí pustil starý dobrý fdisk a pomocí několika přesně mířených příkazů zajistil vše potřebné. V Disk Druidovi bych možná tabulátorem skákal po čudlících ještě doteď. :-)

Tak. Teď už jen vybrat, který disk naformátovat a který rozhodně ne, překontrolovat to, ještě jednou překontrolovat a jedeme dál. Jak se dalo předpokládat, čeká na nás širší výběr software k nainstalování než v Red Hat 4.2. Nově je zařazena např. databáze PostgreSQL či démon gated, od spousty programů se dle očekávání objevily nové verze. Např. XFree 3.3.1, Apache 1.2.4. Nemá cenu zde vypisovat všechny konkrétní změny, pokud se někdo v četbě podobného typu vyžívá, doporučuji ke studiu např. telefonní seznam, je podobného rázu. Red Hat 5.0 pro Intel má totiž 230 MB v 482 souborech RPM. Prostě je tam toho spousta a kdo hledá, velice často najde. Pro srovnání – Red Hat 4.2 měl 190 MB v 458 RPM souborech.

Další překvapení čeká v okamžiku detekování hardware. Tam, kde se dřív instalační program ptal, zda mám připojenou myš, a na kterém portu, mi teď jen oznámil že myš mám, a to na COM1 a zeptal se jen na typ. Dobrá, trefil se. Doufám jen, že tahle autodetekce nebude dělat problémy v jiných hardwarových konfiguracích. Dále jsem se dozvěděl, že mám videokartu S3 Trio 64 a že se pro ni hodí speciální akcelerovaný X-Server. No, to jsem všechno už věděl, ale to, že si to zjistí samo, se mi líbí. U PCI karet se autodetekci dá celkem věřit, každá karta je schopná se celkem inteligentně představit a nemělo by docházet k nedorozumění. Toho by podle dokumentace měl instalační program využívat, známé PCI karty by měl prý detekovat spolehlivě.

Opravdu se mi líbilo, že si mohu jednoduše zvolit, které serverové služby se mají při bootování spouštět. Ve starších verzích Red Hatu se to totiž dalo dělat pouze z jednoho konkrétního programu z X-window. Nebo ručně, což rozhodně mělo k elegantnosti daleko.

Zbývá snad už jen nastavit heslo pro roota, nainstalovat bootloader (známé LILO) a systém je připraven k používání.

Nebo ne?

Ne, alespoň v mém případě nebyl. Bohužel. Hlavní vinu na tom měla asi nejpřevratnější novinka co se v Red Hat Linuxu 5.0 vyskytla – systémová knihovna glibc (plným jménem GNU libc 6). Byla vybrána kvůli tomu, že v Linuxu postupně nahrazuje knihovnu libc verze 5. Ta má některé nedostatky, které je bohužel nutné řešit radikálním řezem, i za cenu ztráty plné zpětné kompatibility. Ve skutečnosti to není tak hrozné, jak se může na první pohled zdát – většinu programů stačí jen s glibc znovu zkompilovat a prostě fungují. Red Hat 5.0 je momentálně jedinou distribucí Linuxu, která plně používá glibc místo libc 5.

Bohužel jsem zjistil, že mi nefunguje kompilátor jazyka C – při jakékoliv větší kompilaci spadne a naříká, že „got signal 11“. To obvykle znamená hardwarové problémy (RAM, CPU apod.), ale to zjevně nebyl můj případ, protože Red Hat 4.2 mi na tom samém počítači běhal bez problémů. Po postupném vyptávání se jsem došel k názoru, že je to asi tím, ze mám procesor Cyrix 6x86 a nové GCC s novým glibc si s ním prostě neporadí.

Feela (Ondřej Filip) měl úplně stejný problém. A to dokonce ještě dřív než já, protože našel čas na upgrade svého Linuxu ráno, já až odpoledne. Instaloval pomocí volby „upgrade“ z funkčního Red Hat 4.2, takže jsem si myslel, že je to tím. Já jsem tedy zazálohoval vše potřebné a instaloval jsem komplet znovu. A GCC padalo stejně. Nakonec jsme si řekli, že to bude asi procesorem (oba máme Cyrix 6x86), protože nikomu jinému to nedělalo, a nikdo jiný Cyrix neměl. Večer jsme se dohodli, že to pošleme e-mailem jako oficiální oznámení chyby.

Zatímco já jsem hledal patřičnou e-mailovou adresu, Feela neváhal a poslal mail přímo Eriku Troanovi. Obratem dostal odpověď.

> Get the gcc snapshot from sunsite.unc.edu:/pub/Linux/GCC and try it. Please let me know if it works or not. I've heard of this before, but I don't have any Cyrix chips to try it on. I've given this advice to others, but I never heard back from them...
>
> ---
>
> Stáhni si gcc ze sunsite.unc.edu:/pub/Linux/GCC a zkus ho. Dej mi prosím vědět, zda to funguje. Slyšel jsem o tom již dříve, ale bohužel nemám k dispozici žádný procesor Cyrix. Stejnou radu jsem dal i jiným, ale ještě jsem od nich nic neslyšel.
> Erik Troan

Po nainstalování souboru `gcc-2.7.2.3-glibc` z doporučené adresy nám oběma ten problém zmizel. Feela odepsal, že už je to v pořádku, takže se dá očekávat, že bude zanedlouho k dispozici oficiální update gcc ve formátu RPM. Tohle všechno se seběhlo během pár hodin. Takže až mi zase někdo bude vysvětlovat, že Linux nemá vůbec žádnou podporu, budu se už jen shovívavě usmívat. . . :-)

Proč jsem ale potřeboval funkční GCC? V první řadě abych si mohl zkompilovat ssh, protože staré binárky odmítaly fungovat, že prý „segmentation fault“. Daly si ale domluvit pomocí

```
rpm --rebuild ssh-*.src.rpm
rpm -e ssh
rpm -i ssh
```

Nic jiného jsem už kompilovat nemusel. V konferenci *linux@muni.cz* si navíc někdo stěžoval, že nefunguje ncpmount (utilita na připojování novellovských disků). Prý k opravě stačilo smazat rpm z Red Hat 5.0 a nainstalovat starší z Red Hat 4.2, které ještě nepoužívá glibc.

Oproti tomu, co jsem čekal za komplikace při přechodu z libc 5 na glibc, celá instalace vlastně proběhla celkem hladce. Upgrade na glibc se časem stejně nevyhne nikdo, tak proč to neudělat hned teď. Navíc když k tomu je aktuální distribuce Linuxu! A řekl bych, že za chvíli se začnou rojit i připravené soubory .rpm speciálně pro Red Hat 5.0 se vším potřebným, takže i to `rpm --rebuild` si budete moci odpustit.

Tak co na závěr? Doporučit, nedoporučit? Red Hat 5.0 ještě není tak výborně odladěná distribuce jako Red Hat 4.2, ale má nové vlastnosti, které se vám určitě budou líbit. Pokud požadujete bezproblémový chod serveru a chcete, aby všechno fungovalo hned, sáhněte stále po Red Hat 4.2. Jinak rozhodně uvažujte o Red Hat 5.0, je prostě novější a pokrok nezastavíte.

Debian GNU/Linux – Users' for users

Milan Zamazal, 2. prosince 1997

Debian GNU/Linux

Debian GNU/Linux je v současné době spolu s Red Hat Linuxem jedna ze dvou nejvýznamnějších základních distribucí Linuxu. Z čistě technického hlediska se jedná o distribuci založenou na původním správci balíků dpkg obsahující v současné době přes 1300 balíků ve vývojové verzi. Poslední verze distribuce (1.3) vyšla v červnu letošního roku, novou verzi 2.0 lze očekávat v první polovině příštího roku. Verze 1.3 oficiálně podporuje pouze platformy na bázi procesorů Intel, verze 2.0 by již měla zahrnovat i oficiální varianty pro ostatní linuxovské platformy.

V tomto článku nebudu hovořit o uživatelské stránce distribuce, neboť to si může každý vyzkoušet sám. Nemíním zde také srovnávat Debian s jinými distribucemi, neboť jak uvidíme dále, z hlediska Debianu neexistuje žádná konkurenční distribuce. Uživatel, který chce mít objektivní srovnání technických vlastností různých distribucí udělá nejlépe, když si příslušné distribuce nainstaluje, zkusí je nějakou dobu používat a posoudí sám, co mu nejlépe vyhovuje. Chtěl bych se tedy tentokrát zaměřit hlavně na vysvětlení toho, co to vlastně Debian je, čím se od ostatních distribucí liší po netechnické stránce a proč je tato distribuce svým způsobem výjimečná.

Debian je podobně jako jádro Linuxu dítkem Internetu. Jedním z popudů jeho vzniku byla nespokojenost se stavem distribuce SLS a touha vytvořit novou distribuci na principu free softwaru. Jedním ze sloganů vztahujících se k Debianu je „users for users“, což znamená, že Debian je distribuce tvořená svými uživateli. V současné době je vývoj Debianu plně distribuovaný – Debian je tvořen více než 200 lidmi z mnoha zemí, Českou republiku a Slovensko nevyjímaje. Každý z těchto lidí se může zapojit do různých činností – například tvorby a údržby balíků (každý balík v Debianu má svého nezávislého maintainera), testování distribuce, tvorby manuálů a jiných technických dokumentů, administrativních činností (chcete-li se stát vedoucím tohoto projektu, stačí vám vyhrát volby), programování debianovských utilit, zaučování nových vývojářů, atd. Distribuce a její vývoj je plně otevřená, veškeré technické otázky jsou řešeny prostřednictvím veřejného mailing listu, kde se k nim vyjadřuje kdokoliv z vývojářů a nejen jich, a podílet se na vývoji Debianu může prakticky kdokoliv. Ať už jako oficiální maintainer (což je nutnou podmínkou zejména v případě, chcete-li vytvořit a/nebo udržovat nějaký balík) nebo pouze formou své účasti v diskusích na vývojářském mailing listu. To je zásadní rozdíl oproti ostatním nejvýznamnějším distribucím, které jsou tvořeny malými uzavřenými týmy, a velmi významně to, ať už pozitivně či negativně, ovlivňuje výslednou podobu distribuce.

Nejvýznamnější charakteristikou Debianu je to, že se jedná o *kompletně free softwarovou* distribuci. Možná namítnete, že i ostatní distribuce jsou založeny převážně na free softwaru. Rozdíl je však právě ve slovech „kompletně“ a „převážně“. Debian má svoji free softwarovou podobu jako cíl, nikoliv pouze jako vedlejší efekt. Vývojáři Debianu schválili tzv. *Debian Social Contract* a *Debian Free Software Guidelines* [30]. V těchto dokumentech se zavazují, že Debian vždy zůstane plně free a otevřenou distribucí založenou kompletně na free softwaru a definuje se zde pojem free softwaru z hlediska Debianu. Zjednodušeně řečeno, free softwarem z hlediska Debianu je software, který neklade žádná významná omezení na uživatele (jako například nedostupnost zdrojových textů, zákaz modifikace, zákaz používání pro nějaké účely, platnost licence pouze pro určité platformy, apod.). Tím je mimo jiné zajištěno, že každý program obsažený v Debianu bude možno v rámci Debianu nebo na něm založené distribuci bez problémů používat a spravovat. Uživatel nebo vývojář, který se rozhodne pro Debian, tak má zaručeno, že nikdy nenarazí na proprietární omezení. Tím se Debian liší od všech ostatních nejvýznamnějších distribucí. Cílem Debianu je prakticky dokázat, že lze vybudovat kompletní systém založený na free softwaru. Debian si touto svojí zatím velmi úspěšnou snahou získal respekt free softwarové komunity a je bezpochyby jedním ze současných nejvýznamnějších free softwarových projektů. Přesto je do značné míry podporován i non-free software – na debianovském FTP je například dostupná řada non-free programů v podobě debianovských balíků.

Další významnou odlišností Debianu je to, že se jedná o distribuci nekomerční. Vývojáři Debianu pracují na distribuci ve svém volném čase a takřka výhradně zdarma. Pro většinu z nich je práce na Debianu zajímavým koníčkem, nikoliv existenční nutností. Debian tím pádem nezná pojem vnucování se uživatelům, místo toho je službou uživatelům. Debian nemrhá prostředky na marketing a chytání uživatelů na líbivé věci, místo toho implementuje myšlenky svých vývojářů-uživatelů. Debian je prost reklamních sloganů obsahujících více vykřičníků než informací. Cílem Debianu není ani finanční zisk, ani nejvyšší počet uživatelů, ani vysoké umístění na žebříčku popularity.

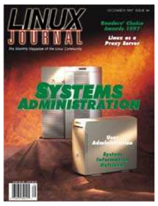
Výše uvedené přístupy přinesly některé pozoruhodné výsledky. Debian byl první distribucí, která přišla s inkrementálními upgrady. Jeho správce balíků dpkg měl jako první některé zajímavé vlastnosti jako například závislosti mezi balíky, které se dnes zdají být již samozřejmostí. Debian je velmi stabilní a bezpečnou distribucí, tyto vlastnosti mají vždy vyšší prioritu než například rychlost vydávání nových verzí. A v neposlední řadě je Debian nejrozsáhlejší základní distribucí co se týče množství softwaru, který obsahuje. Důkazem kvalit Debianu je jistě i to, že se zúčastnil zatím dvou letů amerického raketoplánu a je velmi pravděpodobné, že to nebylo naposledy.

Když to vše shrneme, Debian znamená naprostou svobodu v tvorbě softwaru. Nemusíte nikoho přemlouvat, aby ve své distribuci zohlednil vaše požadavky. Můžete své představy do distribuce zabudovat sami. Chcete-li se „vrtat“ ve své distribuci nejen na vlastním počítači nebo jste důslednými příznivci free softwaru nebo pouze toužíte mít instalovánu stabilní a bezpečnou distribuci, Debian je distribuce pro vás.

Bylo by škoda, kdyby těchto vlastností Debianu nebylo využito ve prospěch uživatelů z českých zemí a Slovenska. Proto se v rámci projektů České nadace pro podporu free softwaru chystá širší účast českých, moravských a slovenských uživatelů na internacionalizaci Debianu. Cílem je dostat Debian do podoby distribuce na úrovni systému plně podporující požadavky česky a slovensky mluvících uživatelů. Pomoci lze překlady, programy, radami. *Vytvořme si* kvalitní free distribuci s českým a slovenským prostředím! Spuštění projektu se očekává začátkem příštího roku a bude oznámeno v *cz.comp.linux*. Zajímá-li vás tento projekt blíže, můžete se ozvat na adresu [31].

Linux Journal

Pavel Janík ml., 1. prosince 1997



Prosincové číslo měsíčníku Linux Journal je určeno hlavně pro systémové administrátory.

Velice zajímavý je článek Petera Eltona o Linuxu jako proxy serveru. Autor zde popisuje jak postupoval při konfiguraci linuxového proxy serveru. Použil přitom balíků TIS fwtk a Socks5.

Druhé pokračování článku Michaela J. Hammela o grafickém programu GIMP je zaměřeno na základní principy tohoto skvělého nástroje pro grafiky (okna, dialogová okna a kurzory). Dovíme se také, jaké formáty GIMP „umí“ a jaké ne.

Článek Davida Bandela je určen především pro začínající administrátory. Zabývá se správou uživatelských účtů a základními administrátorskými úkony. Bohužel o shadow je zde pouze jediný odstaveček.

Nakladatelství O'Reilly & Associates vydalo knihu Jeffa Trantera nazvanou *Linux Multimedia Guide*. Recenzi provedl Michael J. Hammel. Kniha není určena pouze pro konzumenty multimediální kultury, ale také pro programátory multimediálních aplikací, čímž se z ní stává zajímavý zdroj informací pro všechny.

Marjorie Richardson vede virtuální rozhovor s Larry Austinem, zakladatelem společnosti VA Research. Tato společnost je dodavatelem pracovních stanic s operačním systémem Linux.

V prosincovém čísle také naleznete vyhlášení výsledků ankety Readers' Choice Awards, která měla určit nejpoužívanější produkty v různých kategoriích. Mezi distribucemi vyhrál Red Hat Linux před Debianem (Red Hat Linux dostal od čtenářů Linux Journalu skoro třikrát více hlasů než Debian). Nejpoužívanějším vývojovým prostředkem se stal kompiler `gcc` těsně následovaný Perlem. Jasným vítězem kategorie shellů se stal Bourne Again Shell (`bash`). `Vi` jasně ukázal, že je používanějším editorem než Emacs. GIMP neměl v grafických programech téměř žádnou konkurenci. Nejlepším WWW serverem byl zvolen *Linux v2 Information Headquarters* `http://www.linuxhq.com/`. Nejpoužívanější platformou pro Linux je bezpochyby Intel. Mezi hrami dominuje Quake před Doomem. Nejpoužívanějším poštovním klientem je `pine`.

Ingo Molnar, Gadi Oxman a Miguel de Icaza se v Kernel Korner zaměřili na vysvětlení principů a implementace podpory RAID a ovladače MD v jádře.

Andrew G. Morgan popisuje implementaci a používání PAM modulů pro vývoj aplikací.

> The Linux implementation of PAM is the only fully functional version of PAM publicly available.
>
> Implementace PAM v Linuxu je jedinou plně funkční verzí PAM, která je veřejně dostupná.
> Andrew G. Morgan

V rubrice Product Review se nám představí VAR Station II. Počítač postavený na procesoru 266 MHz Intel Pentium II.

Joel D. Hart nám ukáže, že Linux je možné používat i v armádě.

Dalším z představovaných produktů je i zálohovací program BRU 2000, kterému se jistě budeme věnovat v jednom z dalších čísel. Tento program je součástí Red Hat Official Linux.

Linux Journal obsahuje i mnoho jiných zajímavých článků.

Co nového na sunsite.unc.edu?

Pavel Janík ml., 1. prosince 1997

X11

X11/desktop/X-Files-1.41b.tar.gz, filemanager pro X-Window system umožňující souborové operace se soubory `.zip` a `.tar`

apps

apps/math/graphs/chart-2.0.0.tar.gz, jednoduchý neinteraktivní program pro kreslení grafů pro svgalib a X-Window

libs

libs/AppConfig-971117.tar.gz, knihovna pro obsluhu konfiguračních voleb aplikací

system

system/mail/mailhandlers/Autoresponder.v02.tgz, program pro automatické odpovídání na došlé dopisy

system/mail/listservs/BeroList-2.5.2.tar.gz, jednoduše konfigurovatelný listserver

system/admin/idle/Clobberd-3.2-RELEASE.tgz, program umožňující regulovat činnost uživatelů

system/ups/apcupsd-2.9.6.tar.gz, monitor pro APC BackUPS nebo SmartUPS v „dumb“ módu

system/bbs/blackbox-0.73b.tar.gz, BBS pro Linux

system/boot/loaders/chos-0.8.tar.gz, zaváděcí program, založený na systému menu

utils

utils/scripts/basepak-1.1.tar.gz, skript pro konverzi mezi šestnáctkovou, desítkovou a dvojkovou soustavou

utils/file/QuickTree-971024.tgz, utilitka podobná DOSovému příkazu TREE

games

games/arcade/tetris/Xsquarez-0.1.tar.gz, hra podobná Tetrisu

games/arcade/tetris/Xtris-0.3.tar.gz yet, Tetris pro X

games/arcade/tetris/aftetris-1.11.tar.gz, další klon Tetrisu

games/arcade/tetris/altris-1.2.2.tgz, sbírka her podobných hře Tetris

Adabas D pro Linux

Miroslav Šonka, 19. listopadu 1997

Adabas D je relační databázový systém, který se vyznačuje vysokým výkonem, snadnou administrací a úsporným využitím systémových prostředků. Spadá do kategorie výkonných databázových systémů jako např. Informix, Oracle, Sybase aj. Vzhledem k tomu, že se jedná o velmi „mladý“ produkt (první provozní verze byla uvolněna v roce 1984), má některé podstatné výhody (např. nepracuje s tzv. tablespaces, dovoluje opět použít uvolněný prostor databáze atd.), které umožňují při zachování výkonu výrazně snížit tzv. „Cost of Ownership“. Pro operační systém Linux je třeba poznamenat, že nepracuje v prostředí emulace SCO (iBCS).

Systém zahrnuje všechny komponenty pro provoz v prostředí klient/server a pro přístup z Internetu (nástroje pro administraci a dotazování, integraci s MS Office, komunikace v síti, distribuce dat, replikace dat, trigery, uložené procedury, předkompilátory C, C++ a Cobol, rozhraní ODBC, WebDB, přizpůsobení SQL režimu Oracle, DB2 a ANSI). Pro provoz v sítích není třeba pořizovat dodatečné komponenty. Samozřejmostí je možnost využití třídění dle národních konvencí.

Systém ADABAS D je schopen obecně pracovat s různým vývojovým prostředím, kromě prostředí Natural společnosti Software AG je možno použít Uniface, Power Builder, Visual Basic, MS-Access, FoxPro, a další. Existují brány do databázových systémů ADABAS C a DB2.

ADABAS D je moderní databázový systém, ve kterém je v plném rozsahu implementován relační model včetně podpory domén, primárních klíčů, Outer Joins, modifikovatelných JOIN – VIEWS, relační integrity, triggerů a uložených procedur. Podporuje použití velkých neformátovaných dat (BLOB) pomocí databázového typu LONG. Jednoduchá administrace a vysoká výkonnost vytváří z Adabasu D ideální databázový systém v oblasti klient/server.

Adabas D se vyznačuje:

1. vysokou výkonností a spolehlivostí
2. velmi jednoduchou administrací
3. prováděním administračních funkcí za provozu
4. dynamickou optimalizací využití prostoru databáze
5. podporou multitaskingu a multithreadingu
6. minimalizaci I/O operací prováděním asynchronních operací Logging a Group Commits
7. zamykáním dat na úrovni vět
8. optimalizací dotazů
9. podporou distribuce databáze
10. značnou odolností proti chybám hardware i v rámci distribuované databáze
11. robustností vzhledem k udržení konzistence dat

Komprimace dat a efektivní správa volné paměti zajišťují úsporné využití diskové kapacity. Sofistikované algoritmy zajišťují rovnoměrné vytížení všech disků. Tabulky a indexy se bez nutnosti administrativních zásahů dynamicky přizpůsobují provozním potřebám. Díky dynamické správě paměti je zajištěna rovnoměrná rychlost zpracování během celého života databáze.

Automatická reorganizace a zálohování dat umožňují nepřetržitý provoz 24 hodin denně 7 dní v týdnu. Adabas D je odolný proti chybám hardware a poskytuje systém zálohování on-line. V konfiguraci klient/server je možné Adabas D nasadit jako centrální databázi, nebo vytvořit databázi distribuovanou s rozdělením dat na více síťových uzlů, čímž vznikne výkonný virtuální server. Distribuovaná databáze se vůči uživatelům chová jako centrální. Je toho dosaženo použitím dvoufázového potvrzení (Two Phase Commit), jenž realizuje distribuované transakce, globální optimalizací a také nezávislostí na technických i programových prostředcích.

Adabas D umožňuje udržování kopií tabulek ve všech uzlech distribuované databáze. Zvyšuje se tak rychlost přístupu k datům a snižuje zátěž sítě. Replikovaná data jsou automaticky aktualizována podle hodnot v mateřských tabulkách. Konzistence dat a rozpracovaných transakcí je v distribuované databázi udržena i v případě dočasných poruch sítě.

Adabas D podporuje četné dialekty SQL, například ANSI, Oracle a DB2. Tím je zajištěna přenositelnost mnoha aplikací z jiných prostředí.

Adabas D disponuje souhrnným konceptem autorizace, jenž podporuje čtyři uživatelské třídy a sloupcově orientovaná přístupová práva. Pro každého uživatele tak lze vymezit množinu přístupných dat a znemožnit neautorizovaný přístup k datům.

Pokud aplikace resp. vývojové prostředí nekomunikuje se serverem ADABASu D přímo (což ulehčuje možnost použití dialektů ORACLE, DB2 a ANSI) je možno použít rozhraní ODBC. Driver ODBC vyhovuje specifikaci Level 2 a byl vyvinut přímo společností Software AG tak, aby byl maximálně využit výkon serveru Adabas D. Mimoto systém obsahuje modul pro zpřístupnění dat z Adabasu D do produktů společnosti Microsoft (Excel, Word, atd.)

Komponenta Adabas WebDB zprostředkuje přístup Web Serveru na databázový systém Adabas D z některého z protokolů CGI, NSAPI nebo ISAPI.

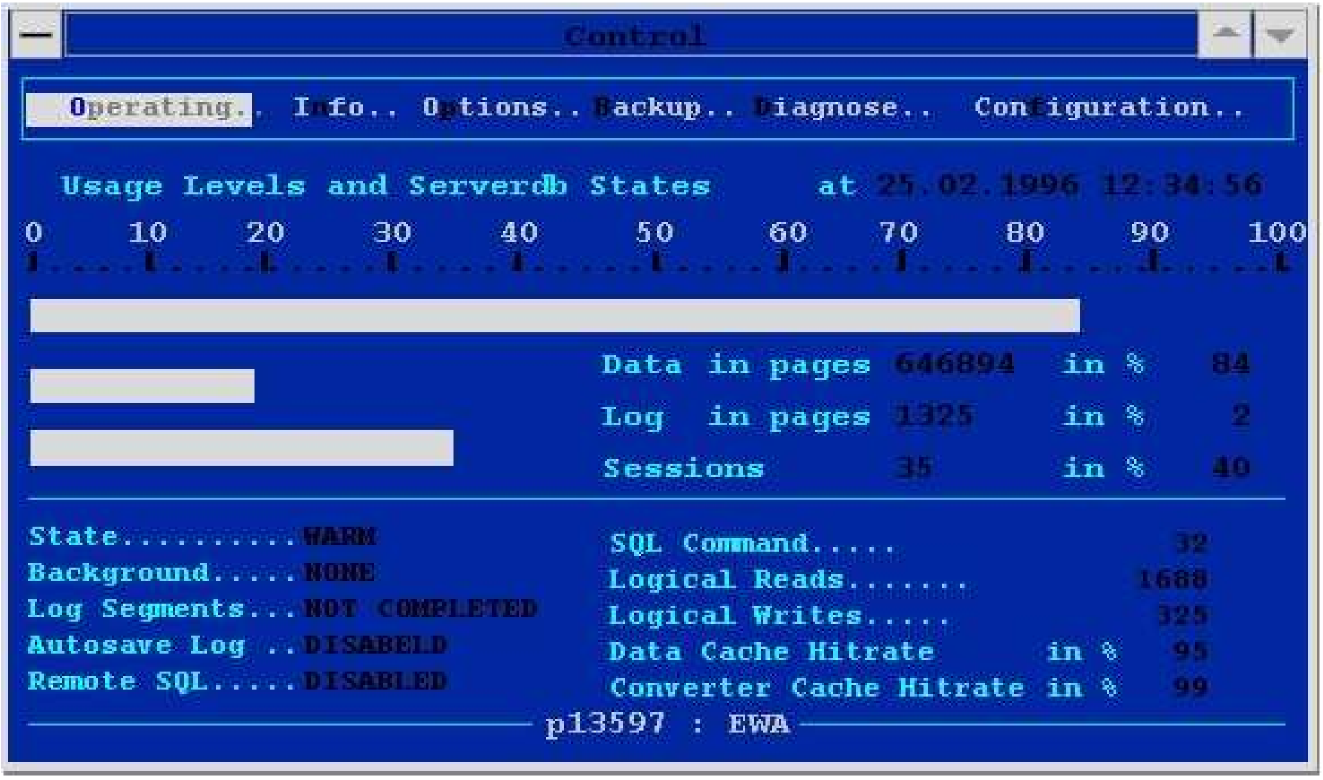
Jako názorný příklad vlastností a funkcionality Adabasu D může sloužit jednoduchá aplikace NORTHWIND (viz obrázek), vyvinutá v prostředí Visual Basicu, se kterou se zájemci mohou blíže seznámit na CD Adabas D Personal Edition. Toto CD je šířeno bezplatně na požádání a obsahuje funkčně omezenou verzi tohoto databázového systému. Struktura databáze byla vytvořena za pomocí administračního programu DOMAIN a propojení vývojového prostředí s databází zajišťuje ODBC.

Autor je zaměstnancem společnosti Software AG [32].

Úvod do RPM

Jan „Yenya“ Kasprzak, 6. listopadu 1997

V předchozím čísle jsem se věnoval problematice balíků softwaru obecně a informacím o tom, co by správce balíků měl umět. V tomto čísle se podrobněji podíváme na systém RPM (*RedHat Package Manager*) z Red Hat Linuxu.



Současný systém RPM je výsledkem dlouhého vývoje. Je založen na starších systémech Red Hat Softwaru, jako je RPP a PM, a také na správci balíků PMS, použitého v distribuci BOGUS blahé paměti. První verze RPM byly psány v Perlu a nepodporovaly více architektur. Použití Perlu také komplikovalo nasazení RPM v omezeném prostředí, jako jsou například bootovací diskety při instalaci systému.

V současné době se používá RPM verze 2, což je systém psaný v jazyce C, přenositelný (a také přenesený) na nejrůznější architektury, s dostatečně rychlým a robustním designem databáze. Formát databáze i RPM balíků je navržen rozšiřitelně, lze tedy přidávat nové vlastnosti. V poslední době to například byly závislosti mezi balíky nebo balíky nezávislé na architektuře (to jsou ty, které neobsahují žádné binární programy závislé na architektuře; příkladem je balík `ghostscript-fonts` nebo `howto`).

Cíle návrhu RPM

- **Jednoduché instalování a rušení balíků**. Preinstalační a post-instalační skripty dávají autorovi balíku kompletní kontrolu nad tím, jak se balík bude instalovat (například `qmail` přidává určité uživatele do `/etc/passwd`, `ssh` může při instalaci vygenerovat RSA pár klíčů a podobně). Totéž platí pro skripty před a po odinstalování balíku. Je-li balík navržen správně, může pro svoji instalaci i odinstalování učinit vše potřebné.
- **Kontrola instalovaných balíků**. Správce balíků umí nejen zjistit, které soubory patřící danému balíku se změnily, ale umí také například vyvolat verifikační skript, který umožní například zkontrolovat syntaxi konfiguračních souborů.
- **Jednoduchá tvorba balíků**. Tvorba a údržba balíků v RPM není o mnoho složitější, než kompilování a instalace ze zdrojových textů. Pokud se tato instalace provádí na více počítačích, nebo dokonce chcete čas od času instalovat novější verzi daného softwaru, RPM značně šetří čas: ve zdrojovém RPM balíku je totiž poznamenáno, jaké všechny úpravy autor dělal na původní zdrojové distribuci softwaru, než vznikla kompilace a později i běhu schopná instalace.
- **Práce na více architekturách**. RPM zamezí duplikaci práce při kompilování a používání téhož softwaru na více architekturách. Pokud je kompilace nebo instalace balíku na jiné platformě odlišná od původní platformy, má RPM možnost ve zdrojovém balíku uvádět příkazy, které se provedou jen na určitých architekturách.

Uvnitř RPM balíku

Nebudu zde popisovat přesně definici struktury RPM souboru, ale přesto stojí za to alespoň rámcově vědět, co všechno RPM balík obsahuje.

- **Jméno balíku**. Tato položka je zde proto, aby bylo možné RPM soubory ukládat i na svazcích, které nepodporují dostatečně dlouhá jména souborů (jako například FAT nebo Minix FS). Jako příklad jména si uveďme třeba `ssh-1.2.21-1i.i386.rpm`.

  Jméno balíku se obvykle skládá z těchto komponent:

  - **Jméno softwaru** (zde: `ssh`). Odpovídá názvu programu nebo programového systému, který je obsažen v daném balíku nebo souhrnně pojmenovává soubory v tomto balíku (třeba `XFree86-100dpi-fonts`).
  - **Verze softwaru** (zde: `1.2.21`). Identifikuje verzi programu, který je uvnitř daného balíku.
  - **Verze balíku** (*package release*; zde: `1i`). Odpovídá zhruba tomu, kolikrát byl balík znovu zabalen s různými modifikacemi a úpravami a zveřejněn. Pokud se například najde chyba v post-instalačním skriptu, může autor balíku vydat nový release balíku.
  - **Architektura** (zde: `i386`). Specifikuje, pro kterou platformu je balík určen. Navíc poslední verze RPM podporují speciální architekturu noarch, která značí, že balík lze instalovat na libovolné platformě (například již zmiňovaný balík `ghostscript-fonts`). V RPM balíku je kromě hardwarové architektury ještě uveden operační systém, pro který je balík určen. Tento atribut se ale neprojeví v názvu balíku.

- Informace o balíku jako celku:
  - datum a čas vytvoření balíku.
  - datum a čas instalace balíku, jedná-li se o již nainstalovaný balík.
  - celková velikost všech souborů v balíku.
  - jméno stroje, na kterém byl balík vytvořen.
  - jméno člověka nebo firmy, která balík distribuuje.
  - jméno distribuce, ke které balík patří.
  - skupina softwaru, do které balík náleží (například balík `ssh` patří do skupiny Applications/Networking). Seznam skupin a podskupin zveřejňuje Red Hat Software. Skupina se používá k tomu, aby uživatel byl schopen procházet balíky tematicky patřící k sobě.
  - MD5 kontrolní součet celého balíku.
  - libovolný (i nulový) počet digitálních PGP podpisů. Tyto podpisy slouží k ověření autora balíku. Balíky distribuované Red Hat Software jsou podepsány jedním PGP klíčem, takže je možné je zkontrolovat. PGP klíč softwaru naleznete na [38]
  - popis balíku – jednořádková informace o balíku (*summary*) a jednak libovolně dlouhý popis (*description*).
  - instalační a verifikační scripty.
- Informace o jednotlivých souborech:
  - jméno souboru s cestou
  - obsah souboru
  - přístupová práva
  - vlastník a skupina souboru
  - MD5 kontrolní součet každého souboru

V příštím čísle se budeme věnovat jednotlivým parametrům programu RPM pro instalaci, rušení a upgradování balíků a pro získávání informací o balících.

Interview s Linusem Torvaldsem

Tom Schaefer, 1. listopadu 1997

Následující text je emailové interview s Linusem Torvaldsem, tvůrcem operačního systému Linux. Protože je velice zaneprázdněn, děkujeme mu, že nám věnoval několik minutek, ve kterých nám sdělil na čem právě pracuje. Originál tohoto interview naleznete na adrese [33].

TS: *Proč jsi byl vzhůru tak dlouho do noci?*
Linus: Pracoval jsem na různých věcech; současný SMP kód a dcache jsou mnou těžce ovlivněny. A kromě Linuxu mám také svoji SkutečnouPráci® a samozřejmě rodinu.

TS: *Co je pravdy na pověstech, že chceš pokračovat s jádrem Linuxu na komerční bázi?*
Linus: Nic, vůbec nic. Já jsem nikdy nic takového neslyšel a jsem poněkud překvapen. Autorská práva jádra mi neumožňují komercializaci, ani kdybych chtěl, což nechci. Protože jsem hlavním vlastníkem autorských práv, doporučuji všem lidem, aby do hlavičky svých kódů vkládali svoje jména – výsledkem toho je, že existuje několik lidí, kteří jsou majiteli autorských práv pro různé části jádra.

Proto může každý klidně spát s přesvědčením, že pokud bych chtěl komercializovat Linux, musel bych k tomu mít souhlas tolika lidí, že to prostě není možné.

Kromě toho, kdyby čert chtěl a já bych některý kernel komercializoval, tak by se neztratila GPL nad staršími jádry, která by nepřestala fungovat a jistě by se našli lidé, kteří by pokračovali v jejich vývoji, protože si myslí, že by Linux měl zůstat free.

Výsledkem tedy je, že opravdu není čeho se obávat.

TS: *Jaký je stav podpory multiprocesorových systémů v jádře?*
Linus: Linux/SMP není příliš starý, současná stabilní jádra (tj. jádra řady 2.0) obsahují tradiční „první přiblížení“ SMP – jednoduchý zámek jádra.

Jednoduchý zámek určený k ochraně dat jádra je nejjednodušší cestou jak udělat podporu pro SMP, ale je zde problém s jistým typem zátěže (vše, co intenzivně používá volání jádra, např. web server). Ale i tak to pracuje celkem pěkně (např. pro paralelní `make` atd.). Ale ještě to není dobré finální řešení.

Dobrou zprávou je, že to všichni vědí a že cesta jednoduchého zámku je pouze prvním přiblížením. Současná vývojová jádra (řada 2.1) už obsahuje implementaci násobného zámku SMP (multi-lock), která je poměrně zdařilá.

V současné době je velká část jádra zamykána jednoduchým zámkem, ale velký rozdíl je v tom, že to již není koncepční nutností, ale pouze výsledkem pomalého přechodu z původního modelu SMP na nový.

Proto bylo ve verzích řady 2.0 dost nemožné přesunout nějakou část jádra mimo zámek. Verze 2.2 už bude mít mnoho důležitých subsystémů oddělených (obsluha procesů, přerušení, signály apod.)

TS: *Existuje mnoho Linuxových projektů, vidíš nějaký jako významnější než ostatní?*
Linus: Pokud bych si měl vybrat nějaký, na kterém bych chtěl pracovat, stále bych pokračoval ve vývoji jádra, a to jednoduše z toho důvodu, že zde leží mé osobní zájmy. Ale jsem šťastný, že mohu říci, že v současné době je většina skutečně důležité práce konána v uživatelském prostoru a jádro bylo přesunuto tam, kde by mělo být: důležitá část systému, ale lidé už o něm nepřemýšlí tak často jako dříve, a to jednoduše proto, že si zvykli na jeho přítomnost.

Z projektů v uživatelském prostoru se mi nejvíce líbí *wine* (emulátor prostředí MS Windows) a *egcs* (pokračovatel projektu GNU C kompileru). Ale jsou jistě i jiné zajímavé projekty, jako např. NC společnosti Corel, který je založen na Linuxu běžícím na StrongARM (bližší informace jsou v článku ➡NC společnosti Corel).

TS: *Pravděpodobně nejčastější otázka: Kterou distribuci Linuxu považuješ osobně za nejlepší a proč?*
Linus: Já osobně používám Red Hat Linux, částečně protože jí důvěřuji po technické stránce. Distribuce Red Hat je velice propracovaná po stránce bezpečnosti na Internetu; ze svých zkušeností mohu říci, že společnost Red Hat je zdaleka nejodpovědnějším prodejcem (a to nejen ve světě Linuxu) co se týče bezpečnosti. Dalším důvodem je, že jsem líný a je tak jednoduché upgradovat balíky automaticky přes FTP, když je nalezena další bezpečnostní díra v sendmailu. . .

Jiným (a to dost podstatným) problémem je, že mám doma tři stanice DEC Alpha a Red Hat je jednou z mála distribucí Linuxu, která je k dispozici na více architekturách. Proto jsou možnosti mé volby poněkud omezeny.

Samozřejmě nechci vychvalovat pouze Red Hat – velice se mi líbí třeba SuSe. Vypadá docela dobře a ačkoli byla původně německá (jejich první CD bylo celé v němčině – mohu tento jazyk dekódovat, pokud bych chtěl, ale nemohu jej číst), dnes již má i anglickou verzi. Částečně také proto, že měli jazykové problémy, se nyní snaží brát vážně problémy internacionalizace.

Dost také záleží na tom, co od distribuce chcete. Caldera má jiné cíle než třeba Red Hat nebo Suse. Kvůli jejich pěknému desktopu jsem ji nainstaloval na počítač mé ženy (ale jakmile si na něj zvykla, ihned jsem její počítač přeinstaloval na Red Hat jenom proto, že jsem líný jako veš a protože mám na svých počítačích Red Hat a nechtěl jsem se starat o dva různé systémy, když se tomu mohu vyhnout. . .). Pro některé lidi může být ale Caldera kvůli jejich podpoře Netware jedinou volbou.

TS: *Nyní žiješ ve Spojených státech. Můžeš nám říci, kde pracuješ, co tvoje rodina atd.?*
Linus: Ano, nyní žiji ve Spojených státech, přesněji v Santa Clara, CA. My jsme se přestěhovali v tomto roce a jsme zde velice šťastní. Počasí je zde o poznání lepší než v Helsinkách, a mně se líbí moje práce.

(„My“ znamená já, moje žena Tove, naše deset měsíců stará dcera Patricia a naše dvě kočky Randi a Misu. Ano, když jsme se stěhovali, vzali jsme s sebou naše kočky.)

Pracuji pro maličkou společnost Transmeta a skoro oficiální prohlášení říká, že vyrábíme „produkty“. Pokud by se někdo zeptal, jaké „produkty“, odpověď by pravděpodobně zněla „pěkné produkty“. Tak a teď znáte všechny podrobnosti.

(Ale ne, ve skutečnosti to nesouvisí s Linuxem: Chtěl jsem dělat něco nového někde, kde mi dovolí pracovat jak na Linuxu tak i na něčem jiném, a společnost Transmeta je ráda, že pokračuji ve vývoji Linuxu nesvázán řetězy – pravděpodobně proto, že Linux hojně používají na svých počítačích.)

Kterak jádro přeložiti

Dan Ohnesorg, 28. listopadu 1997

Úvodem si řekněme několik důvodů, proč se vůbec jádra překládají. Drtivá většina distribucí v současné době obsahuje modulární jádro, které se skládá ze základního jádra, jehož možnosti se rozšiřují nahráváním modulů.

Prvním důvodem pro překlad je přeložení základních ovladačů, které jádro potřebuje pro náš hardware, přímo do jádra. To má jednak výhodu v jednodušší konfiguraci systému, protože není nutné natahovat moduly ručně, resp. ze startovacích skriptů a navíc některé ovladače umí v nemodulární verzi najít svůj hardware bez nutnosti zadávat další konfigurační parametry při startu jádra. Zvláště u serverů bych překlad ovladačů do jádra velmi doporučoval. Má to i další výhodu, takové jádro je možné nahrát na jednu disketu a pokud nám nejde namountovat hlavní disk, můžeme nastartovat z této diskety a máme všechno přístupné.

Dalším důvodem pro překlad je zavedení podpory pro specifické zařízení. Poměrně typickým zástupcem této skupiny je řadič pevného disku s obvodem UMC, u kterého je možné po překladu jádra s podporou pro UMC dosáhnout několikanásobného zvýšení rychlosti čtení z disku. U karet NE* je možné nastavovat adresy, na kterých se karta hledá a ovladač lze potom použít i na noname PCI karty kompatibilní např. s NE2500 bez dalších parametrů při startu jádra.

Další skupinou důvodů jsou různé upgrady jádra. Ty jsou nutné buď pro zacelení bezpečnostních děr nebo pro odstranění chyb. Případně pro rozšíření možností jádra. Tady jmenujme např. patche pro možnost čtení dat z CD ve formátu joliet nebo podporu tzv. FAT32.

Dalším důvodem pro kompilaci může být optimalizace jádra z hlediska velikosti nebo výkonu. Optimalizací na výkon se vůbec nebudeme zabývat, není to nic jednoduchého. Jediné co lze snadno ovlivnit je optimalizace jádra pro konkrétní procesor. Všechna jádra používaná pro prvotní instalaci jsou přeložena pro 386. Pokud máme Pentium Pro, je škoda jeho možností nevyužít a nepoužívat vylepšené instrukce. Z hlediska velikosti se jádro optimalizuje tím, že se přeloží jen skutečně potřebné věci, ale v dnešní době dimenzování počítačů na produkty firmy Microsoft je ušetření několika desítek kilobytů paměti zanedbatelné, zvláště u systému, který nemá omezení 640 kilobytů paměti na běh programů. Je třeba upozornit na to, že jádro optimalizované pro Pentium nemusí běžet na intelských klonech od AMD nebo Cyrix, ty potřebují jádro přeložené pro 386.

Pokud jsem Vás přesvědčil, že byste měli s Vaším jádrem něco udělat, čtěte dále, neboť se konečně dostáváme k tomu jak to udělat.

Získání a příprava jádra

Primárním server, odkud je možné jádro získat je `ftp://ftp.kernel.org`. Jádro se nachází v adresáři `/pub/linux/kernel` a dále podle verze buď v adesáři `v2.0` nebo `v2.1`. Verze, které mají na druhé pozici liché číslo jsou *vývojové* a sudé jsou tzv. *stabilní*. Výhodou vývojových verzí je obvykle širší podpora speciálního hardware, ale nejsou tolik vyzkoušené a proto se nehodí pro kritické aplikace a pro začátečníky také ne. Sice zřídka, ale přece se občas dostane do světa vývojová verze, která nejde přeložit bez oprav přímo ve zdrojových textech. V adresáři s jádrem se vždy vyskytuje soubor `LATEST-IS-2.0.32`, kde číslo v názvu udává poslední verzi jádra. Jádro si musíme stáhnout na náš počítač. Vhodný postup je následující.

```
cd /usr/src
ftp ftp.kernel.org
ftp
Login@Pocitac.Domena.cz
cd /pub/linux/kernel/v2.0
binary
get linux-2.0.32.tar.gz
bye
```

Potom jádro rozbalíme

```
tar xvzf linux-2.0.32.tar.gz
```

Parametry příkaru tar znamenají eXtract=rozbal, Verbose=piš co se děje, gunZip=dekomprimuj, File=načti archív ze souboru.

Tip: Pokud používáte distribuce Red Hat nebo Debian, nainstalujte si balík kernel source, pomocí tzv. *dependencies* (závislostí) se zároveň instalují všechny programy pro překlad potřebné. Jádro, které již pravděpodobně bude poněkud zastaralé, můžete potom smazat.

Aby mohlo být jádro přeloženo musí existovat následující odkazy. To se ověří např. následujícími příkazy:

```
cd /usr/include
ls -l linux asm scsi
```

které musí zobrazit příslušné symbolické odkazy

```
asm -> /usr/src/linux/include/asm/
linux -> /usr/src/linux/include/linux/
scsi -> /usr/src/linux/include/scsi/
```

Pokud tyto odkazy neexistují, je nutné je vytvořit, a to následovně:

```
cd /usr/include
rm -rf asm linux scsi
ln -s /usr/src/linux/include/asm asm
ln -s /usr/src/linux/include/linux linux
ln -s /usr/src/linux/include/scsi scsi
```

Dále je nutné mít v záloze kernel starý. Je to sice nepopulární, ale zálohovat je nutné. Je lepší dělat zálohy hned dvě. První na disketu, to je velmi snadné, nejdříve je ale nutné vědět, kde se kernel v naší distribuci nachází, bude to buď v `/vmlinuz` nebo `/boot/vmlinuz`, efektivní metoda jak to zjistit je použití příkazu

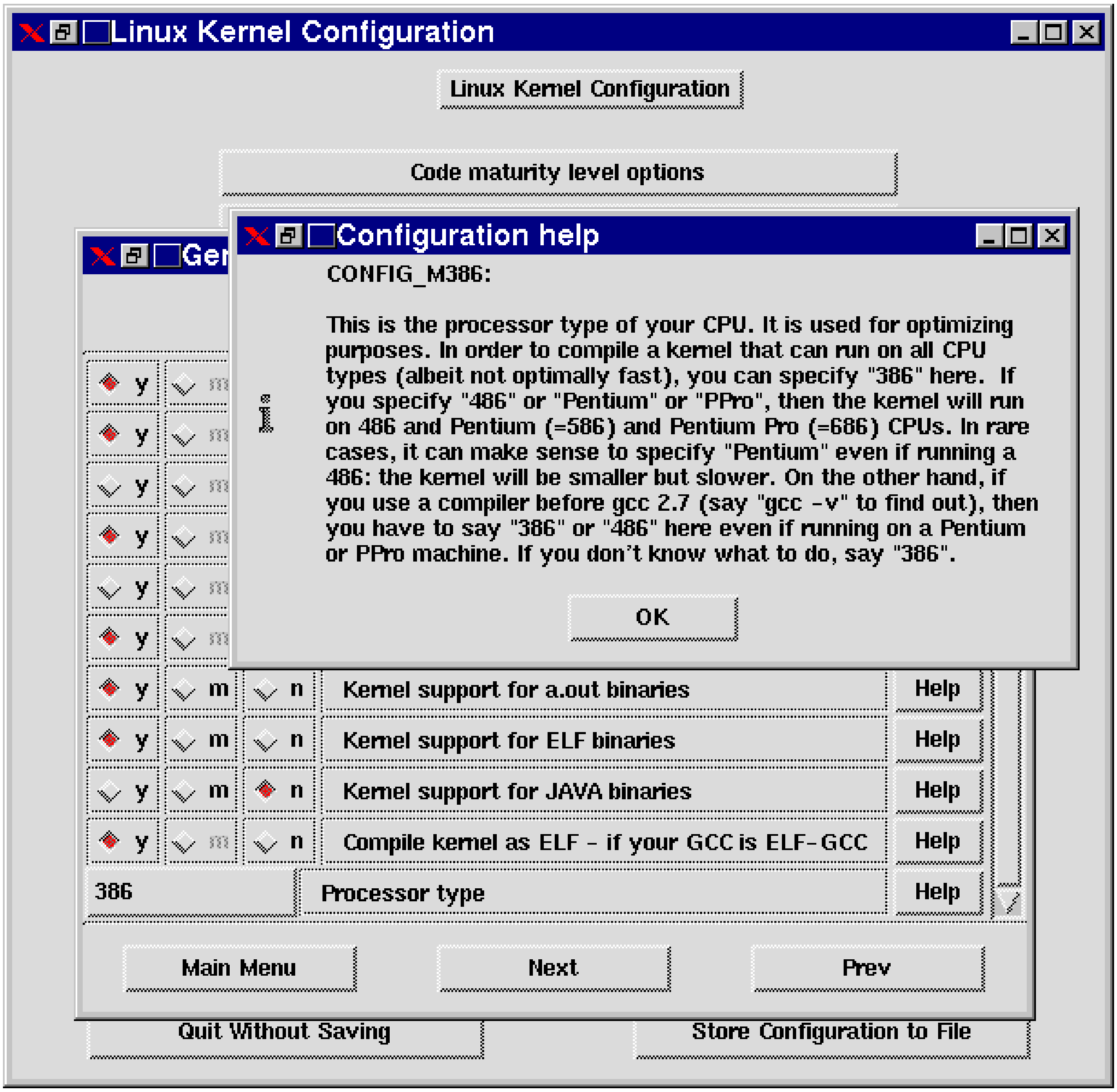
```
more /etc/lilo.conf
```

a najít řádek na kterém je napsáno `image=....`

Jádro překopírujeme na disketu, která neobsahuje žádná podstatná data (v průběhu akce bude smazána) příkazem

```
dd if=/boot/vmlinuz of=/dev/fd0
```

přičemž první parametr nahradíme skutečnou cestou k jádru a druhý reprezentuje disketu v mechanice. Tímto je vytvořena bootovací disketa. Dále je vhodné zkopírovat jádro do zálohy, např. takto:




```
cp /boot/vmlinuz /boot/vmlinuz.old
```

a opravit konfigurační soubor `/etc/lilo.conf`. Pro úpravu je vhodné použít např. textový editor `joe`, který má ovládání obdobné jako textový editor Word Star, z nehož vychází ovládání většiny dosových programů. Editor je nutné instalovat zvlášť, distribuce jej obvykle nemají ve standardních volbách. Výsledný soubor by měl vypadat přibližně tak, jak je uvedeno na výpisu ➡Soubor lilo.conf. Pokud je v originále v sekci image nějaký další parametr, jako např. `initrd=`, musí být také v sekci old.

Nyní je bezpodmínečně nutné zadat příkaz `lilo`. To je nutné po každé změně jádra nebo konfiguračního souboru `/etc/lilo.conf`.

Nyní je příprava k překladu hotová.

Konfigurace jádra

Jádro lze konfigurovat v zásadě třemi způsoby. První funguje téměř vždy, ale není tak komfortní jako další dvě varianty. Vyvolá se příkazem

```
make config
```

v adresáři `/usr/src/linux`. Postupně se dotazuje na různé parametry a požadavky uživatele.

Mnohem lepší jsou další dvě varianty. Nejkomfortnější z nich je příkaz

```
make xconfig
```

který je nutné spouštět z běžícího X-serveru, tedy nejlépe z xtermu. Po chviličce překladu se objeví okno, ve kte-

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50

image=/boot/vmlinuz
 label=linux
 root=/dev/hda2
 read-only

image=/boot/vmlinuz.old
 label=old
 root=/dev/hda2
 read-only
```

Výpis 1: Soubor lilo.conf

rém lze postupně vybrat všechny parametry a požadavky. V případě potřeby se ke každé volbě dá zobrazit nápověda. Tam kde uživatel neví co vybrat, lze doporučit nechat původní nastavení. U některých položek je možné kliknout do sloupce M a ovladač nebo něco jiného se přeloží jako modul.

Poslední variantou je

```
make menuconfig
```

Opět pomocí systému menu umožňuje vybrat vše potřebné.

Při konfiguraci jádra je nutné znát velice přesně konfiguraci počítače. Pokud počítač obsahuje noname karty, je vhodné vědět, jaké obsahují čipy. Není vyloučeno, že čip má v jádře podporu.

Jemné doladění lze provést ještě přímo ve zdrojových textech jednotlivých ovladačů. Drtivá většina z nich je rozdělena na dvě části, první je určena k editaci a druhá ne. Co je možné editovat se pozná snadno, obvykle je to bohatě popsané a okomentované. Např. u síťových karet je možné nastavit vlastní adresy, na kterých se má karta hledat. Je potřeba dát pozor na to, že adresy, které píše počítač při startu jsou v hexadecimálním tvaru a musí se zadávat ve tvaru 0x000, třeba pro síťovou kartu sídlící na typické adrese 300 je nutné zadat 0x300.

Pokud je vše nastaveno, lze spustit překlad příkazem

```
make
```

Pro vyzkoušení jádra se překlad spustí příkazem

```
make bzdisk
```

s vloženou novou disketou v mechanice `a:`. To zajistí, že se nové jádro po překladu nakopíruje na disketu a z ní je možné pokusně nabootovat. Hlášky, které systém při startu vypsal si lze prohlédnout stiskem kláves SHIFT a PGUP případně PGDOWN. (Tato kombinace nefunguje, pokud v průběhu startu natahujete obrazovkové fonty, např. češtinu. Natažení totiž maže buffer videokarty.) Pokud je vše v pořádku a počítač se rozběhne, je možné přeložit jádro přímo na disk, to se dělá příkazem

```
make bzlilo
```

Pokud byl některý ovladač vybrán jako modul, je nutné přeložit moduly. Tady mohou nastat dvě situace. Buď překládáte jádro stejné verze jako provozujete, nebo se jedná o jádro novější. V prvním případě přejmenujte adresář `/lib/modules/verze_jádra` na `/lib/modules/verze_jádra.old`, moduly tam uložené by jinak nesouhlasily s verzí jádra a nebyly by splněny všechny závislosti. V obou případech je možné po vyzkoušení jádra adresář definitivně smazat. Potom je již možné přeložit moduly. K tomu slouží příkaz

```
make modules
```

Pokud proběhne úspěšně, je možné moduly instalovat příkazem

```
make modules_install
```

Jak se moduly používají

Modul lze do jádra vložit dvěma způsoby – manuálně, nebo si jej jádro pomocí démona, který se jmenuje `kerneld`, natáhne samo. Druhá metoda není všemocná, ale pro natažení podpory PPP protokolu při startu pppd ji lze použít. Chování `kerneld` se řídí jeho konfiguračním souborem `/etc/conf.modules`. Příklad takového souboru je na výpisu ➡Soubor /etc/conf.modules.

```
# používaný SCSI řadič
alias scsi_hostadapter aha1542

# používaná síťová karta
alias eth0 3c509

# modul pro ax25 zatím není k dispozici
alias net-pf-3 off

# pokud nepoužíváte IPX
alias net-pf-4 off

# pokud nepoužíváte AppleTalk
alias net-pf-5 off

# zvláštní nastavení síťové karty
options 3c509 io=0x300 irq=10

# zvláštní nastavení jiných karet
options cdu31a cdu31a_port=0x1f88 \
 sony_pas_init=1
```

Výpis 2: Soubor /etc/conf.modules

Pokud nejsou uvedeny aliasy pro net-pf-?, hlásí kernel při startu např.

```
cannot locate net-pf-4 module
```

což není žádná zásadní závada. Pro úplnost budiž uvedeno, že soubor `conf.modules` se může jmenovat i `modules.conf`, protože `kerneld` hledá obě varianty.

Alias zpřístupňuje jádru modul pod dalším jménem. Jádro při potřebě ovladače síťové karty natahuje modul `eth0`, který neexistuje a proto je nutné pomocí aliasu „označit“ ten správný. Řádek obsahující `options` definuje standardní parametry pro start modulu. Pokud je modul natahován z příkazové řádky, mají na ní uvedené parametry přednost.

Druhou možností je vložit modul do jádra ručně. K tomu lze použít dva příkazy, vhodnější je

```
modprobe název_modulu případné_parametry
```

Tento příkaz také využívá parametry nastavené v konfiguračním souboru `/etc/conf.modules`. Umí si sám najít, které další moduly jsou případně pro činnost požadovaného modulu nutné a natáhne je. Pokud má korektně fungovat, je nutné před prvním použitím vytvořit zadáním příkazu

```
/sbin/depmod -a
```

seznam závislostí mezi moduly. Většina distribucí tento příkaz spouští při každém startu.

Další možností je příkaz `insmod`, který dokáže vložit modul o něco „agresivněji" než `modprobe`. `Modprobe` tento příkaz volá pro natažení modulu po kontrole verze a závislostí. Pro začátečníka nemá tento příkaz, podle mého názoru, žádný význam.

Pokud není modul po nějakou dobu používán a byl natažen přes `kerneld`, je automaticky z jádra vyjmut. Většinou se tak stane po 1 minutě. Ručně natažený modul zůstává v jádře trvale. Odstraní se příkazem

```
rmmod jméno_modulu
```

Pokud jméno není známé, nebo chceme vědět jaké moduly jádro aktuálně používá, zadáme příkaz

```
lsmod
```

Informace o modulech v jádře lze získat také příkazem

```
cat /proc/modules
```

Závěrem

Nakonec bych ocitoval několik rad, které lze najít v dokumentaci k jádru, která je uložena v adresáři `/usr/src/linux/Documentation`. Obsahuje mnoho pokynů ke konfiguraci konkrétních periferií, nastavení jednotlivých karet, kompatibilitě jednotlivých výrobků a podobně.

Nejdůležitější jsou dva soubory – `README`, ve kterém je popsána instalace jádra a `Changes`, který obsahuje popis novinek a požadavků pro překlad jádra. Speciálně v případě vývojové řady je často nutné upgradovat některé programy, jako `gcc`, linker apod. Jak na to, a kde se nové verze nachází je popsáno právě v tomto souboru.

Vývojáři radí nepoužívat zbytečně ovladače, které nejsou potřeba. Některé karty se totiž mohou při hledání zařízení cizím ovladačem zaseknout a potom není možné kartu ovládat ani korektním ovladačem.

Dále není vhodné zapínat ladící parametry jádra (Kernel Profiling), protože takové jádro je pomalejší a potenciálně méně stabilní. Tyto volby jsou určeny především pro vyhledávání chyb a ladění jádra.

Přeji vám mnoho štěstí při kompilaci jádra.

CPAN – Comprehensive Perl Archive Network

Jan Pazdziora, 8. prosince 1997

Perl, jazyk na zpracování textů, správu systému či přístup k databázím, má mnoho možností a funkcí již ze své definice, velice významná je ale možnost použít moduly, které řeší přímo náš úkol, nebo je jen stačí patřičným způsobem rozšířit. Všechny volně dostupné moduly, spolu s Perlovskými distribucemi a dokumentací, jsou shromážděny na CPANu, archívu mirrorovaném na mnoha serverech světa. Tento příspěvek přiblíží obsah archívu a také naznačí, které úkoly se v Perlu řeší tak často, že pro ně existují již hotová řešení.

Kde je CPAN

Vstupním bodem k CPANu je URL [35], které nás přesměruje k nejbližšímu mirroru. Pokud s defaultním přesměrováním nejsme spokojeni, můžeme odstranit poslední lomítko a vybrat si nejvhodnější server ručně. Na TEN-34-CZ je možno zamířit například na mirror [36].

Kategorie modulů

Mnoho z úkolů, které se chystáme v Perlu udělat, už pravděpodobně dělal někdo před námi. Soubor `modules/00modlist.long.html` obsahuje jednak obšírný úvod do problematiky, tedy například i informaci o tom, co to vlastně modul je, jednak seznam existujících modulů, rozdělený na části podle zaměření a se stručnými komentáři ke každému modulu.

Uveďme zde skupiny, do nichž jsou moduly neformálně roztříděny:

1. **základní Perlovské moduly, rozšíření jazyka Perl a nástroje pro tvorbu dokumentace**
2. **vývojové nástroje**
3. **rozhraní k operačním systémům**
4. **sítě a síťování, meziprocesová komunikace**
5. **datové typy a nástroje pro práci s nimi**
6. **databázová rozhraní**
7. **uživatelská rozhraní**
8. **rozhraní a emulace jiných programovacích jazyků**
9. **soubory, filesystémy, zamykání souborů**
10. **zpracování řetězců, textů, parsování, hledání**
11. **zpracování parametrů příkazové řádky a konfiguračních souborů**
12. **internacionalizace a lokalizace**
13. **autentikace, bezpečnost a kryptování**
14. **World Wide Web, HTML, HTTP, CGI, MIME**
15. **nástroje pro tvorbu serverů a daemonů**
16. **archivace, komprese, konverze**
17. **obrázky, kreslení a grafika**
18. **pošta a Usenet news**
19. **řízení výpočtu (zpětná volání a zpracování výjimek)**
20. **filehandly a vstupně/výstupní proudy**
21. **MS Windows**
22. **různé**

Další obsah CPANu

Kromě modulů najdeme na CPANu jednak zdrojové distribuce Perlu samotného v podadresáři `src`, jednak dokumentaci k Perlu včetně FAQ v adresáři `doc`. Snad stojí za to upozornit například na `doc/perlref-*`, což je vysázená referenční příručka v tom formátu, v jakém ji prodává nakladatelství O'Reilly, nebo podadresář `doc/FMTEYEWTK`, Far More Than Everything You Ever Wanted To Know.

Při hledání řešení na CPANu může pomoci také vyhledávání na URL [37] s nejrůznějšími kritérii hledání.

Distribuce modulů

Každý modul či balík modulů je šířen jako archív (buď .tar.gz nebo .zip) a je umístěn v adresáři svého autora. Adresáře autorů jsou v podadresáři `authors` a odkazy většinou jdou přes `authors/id` s krátkou identifikací každého člověka. Součástí názvu bývá číslo verze a pokud distribuce obsahuje `README`, je toto na CPANu automaticky dostupné vedle daného archívu.

Jak instalovat

Tar rozbalíme, přepneme se do vytvořeného adresáře a spustíme

```
perl Makefile.PL
make
make test
make install
```

samozřejmě pokud předchozí fáze proběhly v pořádku. Prvním bodem by jistě mohla být četba dokumentace a doporučení k instalaci, ale kdo čte manuály, pokud nejsou žádné problémy, že ;-) Příkaz `perl Makefile.PL` vytvoří `Makefile`, který je nadále svázán (cestami a voláním perlu) s tím interpretem, který byl volán na `.PL`. Můžeme tedy mít na disku více verzí Perlu a `Makefile.PL` zajistí, že instalujeme moduly k té správné.

Pokud se objeví problémy, je vhodné si přečíst návod, zda jsme neměli nastavit nějakou proměnnou prostředí (kupříkladu `$ORACLE_HOME`) či se na instalaci mnohdy předcházenou kompilací jinak připravit. Pokud dojde k chybě až při testech, je možné je pustit znovu jako

```
make test TEST_VERBOSE=1
```

což nám může pomoci vyhledat chybu na naší straně.

Pokud jsme si jisti, že chyba není u nás, je vhodné poslat podrobný popis chyby spolu s verzemi Perlu a modulu autorovi, resp. osobě či mailing-listu zodpovědnému za údržbu. Reakce je většinou velmi rychlá – pokud o chybě informujeme, máme naději na její rychlé odstranění, pokud ne, nemáme si nač stěžovat.

Automatická instalace

S distribucí Perlu přichází i modul `CPAN`, který zjednodušuje instalace a upgrade nových verzí modulů na našem lokálním stroji. Nejjednodušší spuštění

```
perl -MCPAN -e shell
```

nám nabídne interaktivní prompt, který dovoluje: vyhledávat moduly na CPANu a porovnávat jejich verze s verzemi instalovanými, volby (`i` a `m`); najít moduly, které mají na CPANu vyšší číslo verze (volba `r`); spustit instalaci (`install`), která ale proběhne jen pokud byl v pořádku jak `make`, tak `make test`; vytvořit seznam nainstalovaných modulů, abychom pak na jiném stroji či jiné platformě byli schopni rychle nainstalovat stejnou konfiguraci (`autobundle`, `recompile`).

Modul `CPAN` při instalaci modulu zajistí stažení distribuce z námi zvoleného CPAN mirroru, přičemž použije buď externí programy nebo knihovnu `LWP`. Porovná kontrolní součet .taru, rozbalí ho a pak provede posloupnost výše uvedených instalačních kroků. Parametry fungování tohoto správce instalace jsou uloženy na disku, takže se po jeho ukončení neztrácejí.

Dokumentace

Perl podporuje dokumentaci přímo v souboru modulu ve formátu POD (Plain Old Documentation). Při instalaci je tato dokumentace automaticky přeformátována do podoby manových stránek a ty jsou nainstalovány spolu s vlastním kódem modulů. Dokumentace je tedy on-line dostupná i poté, co jsme distribuční balík v zájmu šetření místem z disku smazali.

CGI, MIME

Jednou z oblastí, kde je popularita Perlu velice vidět, jsou CGI skripty, neboť množství systémových funkcí spolu s regulárními výrazy a textovými funkcemi dávají možnost rychle se dobrat požadovaného cíle. Obraty jako

```
s/%([0-9a-fA-F]{2})/pack("c",hex($1))/ge;
```

jsou rychlé a efektivní. Ovšem dělat po dvacáté tu samou věc rychle omrzí, a proto existuje například modul `CGI`, který objektovým způsobem nabízí většinu funkcí pro zpracování vstupu všemi HTTP metodami, manipulaci s předanými parametry i tvorbu například HTML odpovědi. Výhodou je jednak čisté řešení, jednak jistá záruka, že kód vyvíjený a podporovaný Perlovskou komunitou se chová podle psaných i nepsaných standardů a pamatuje i na neobvyklé situace.

Typický skript pak může být

```
use CGI;

my $q = new CGI;    # načte vstup

if ($q->param('name') and # test parametrů
   $q->remote_user() eq $allow) {
  print $q->header(
    -type=>'text/html; charset=iso-8859-2',
    -expires=>'+3d');          # odpověď

  print $q->start_html(-title=>' ...
```

Podobně například pro práci s MIME formáty je pravděpodobně lepší přímo použít prověřený modul, například `MIME::Base64` či `MIME::Decoder`, než opakovaně vyvíjet obdobný kód. Volnost Perlu jak v otázce typů, tak například při práci s velkými objemy dat, nám vždy dovoluje obrátit se v případě nutnosti na standardní Perlovské prostředky, kterými uděláme rychle potřebné, a pouze pro rutinní části skriptu použijeme funkce z modulu.

Z CGI a HTML tedy snadno odskočíme k modulu GD, kterým v reálném čase nakreslíme .gif a ten pošleme spolu s HTTP hlavičkami na výstup, případně si můžeme například tvorbu grafů zjednodušit nadstavbou nad GD, modulem GIFgraph. Nebo můžeme pomocí Mail::Folder procházet archív diskusní skupiny a pro přibližné vyhledání použijeme buď regulární výraz či některý z modulů Text::.

LWP, Net

Na úrovni přístupu někam po síti pracují moduly Net::, kde za dvojtečkou jsou jména protokolů jako FTP, Time či NNTP. Při jejich použití opět vytvoříme konstruktorem objekt a k němu voláme metody, které se promítají na odpovídající akce použitého protokolu, například

```
use Net::NNTP;
my $nntp = new Net::NNTP;
my $group = 'comp.lang.perl.misc';
my @gr;
if (@gr = $nntp->group($group)) {
```

Nad protokolem HTTP jsou postaveny moduly LWP::, Library for WWW access in Perl. Poskytují několik úrovní složitosti (a možností), od jednoduchého

```
perl -MLWP::Simple -e \
    'getprint "http://www.perl.com/"'
```

přes třídu HTTP::Request až po široce konfigurovatelný LWP::UserAgent.

Databáze

Přístup k databázovým strojům z Perlu se v poslední době stává velmi rozšířenou aplikací. Existuje specifikace rozhraní a modul DBI (database interface), které definuje skupinu metod pro připojení k databázi, zaslání SQL dotazu a čtení odpovědi. Pro všechny hlavní databáze pak existují databázové ovladače, DBD::, které převádí tato unifikovaná volání na volání API jednotlivých databází.

Příklad použití

```
use DBI;
my $dbh = DBI->connect("dbi:Oracle:prod",
              $user, $pass)
    or die $DBI::errstr;
my $sth = $dbh->prepare("insert into cust
                          values (?, ?, ?)")
    or die $dbh->errstr();
```

následované provedením příkazu. Takto psaný program je přenesitelný pod jinou databázi pouhou změnou parametru ve volání connect. Protože ale je více než jeden způsob, jak věci dělat, můžeme samozřejmě používat i nestandardní rozšíření definovaná databází a ovladačem, která mohou být rychlejší, ale také neportabilní.

Jádro většiny ovladačů je psáno v jazyce C a pro jeho kompilaci je nutná alespoň minimální část vývojového prostředí k dané databázi. Vždy je nutné přečíst si README, kde jsou požadavky pro instalaci konkrétního ovladače vyjmenovány.

Jeden z velmi zajímavých ovladačů je DBD::pNET, který funguje jako proxy pro přístup k databázi na vzdáleném stroji. Mějme například Informix na Solarisu, s vývojovým prostředím, k němuž chceme přistupovat z Linuxového stroje. Informix pro Linux nemáme, a proto nemůžeme zkompilovat driver na Linuxu a použít přímo síťovou podporu databáze (navíc mnohdy za speciální peníze). Můžeme ale zkompilovat DBD na Solarisu a spustit na něm tzv. pNETagenta, daemona, který požadavky přicházející z Linuxového stroje pošle databázi a zpět zprostředkuje výsledky. Přenos dat je možno mít kryptovaný a přístup povolený například jen pro konkrétní stroje.

NC společnosti Corel

Pavel Janík ml., 1. listopadu 1997

Společnost Corel Corporation dne 27. října vydala tiskovou zprávu, která oznamuje jejich nový produkt – je to NC (*Network Computer*). Ale proč o něm chci psát v Linuxových novinách? Existuje pro to jeden prostý důvod. Operačním systémem tohoto NC je Linux na procesoru Digital StrongARM. Hlavními přednostmi jejich produktu jsou:

- omezení nákladů na distribuci softwaru. Corel Video Network Computer (to je oficiální název jejich produktu) aplikuje síťový model ukládání softwaru, dat a nastavení uživatelského prostředí na centrálním serveru. Soubory a komponenty aplikací jsou natahovány v závislosti na požadavcích uživatele, což zajišťuje, že všichni uživatelé mají v jednom okamžiku tutéž verzi softwaru.
- trvalá cache a synchronizace. NC poskytuje možnosti jak pracovat i v případě odpojení od centrálního serveru. Protože aplikace a soubory jsou zrcadleny na NC z centrálního serveru, může uživatel pracovat i po jeho odpojení. Pokud se spojení znovu naváže, jsou data synchronizována s informacemi uloženými na serveru.
- minimální zatížení sítě. Aplikace jsou ze serveru stahovány pouze jednou, poté jsou spouštěny z lokálního pevného disku a nezatěžují zbytečně síť.
- úspora energie. NC má možnost přejít do tzv. „sleep módu", ve kterém se spotřebovává podstatně méně energie. Všechna data a aplikace jsou dočasně uložena pokud není počítač opět uveden do normálního provozu pouhým stiskem tlačítka.

Corel NC je založen na procesoru Digital StrongARM SA-110. Obsahuje 32 MB paměti (paměť je rozšiřitelná na 64 MB) a alespoň 500 MB pevný disk. NC disponuje dvěma ethernetovými porty, z nichž jeden je 10/100Base-T, IrDA transceiverem, paralelními a sériovými porty, 4MB video pamětí pro XVGA a NTSC/PAL video výstup. Multimediální podpora je zajištěna plně stereofonním mikrofonem a reproduktory. Samozřejmě nechybí klávesnice a myš.

V tiskové zprávě se dále hovoří o důvodech, které vedli společnost Corel k použití operačního systému Linux:

- Linux je vysoce stabilní – ideální pro podnikání
- výkonný a robustní
- podpora multitaskingu a multithreadingu
- real-time služby
- široká podpora – více než 100000 vývojářů na celém světě má k dispozici zdrojové kódy

TIP: Jak efektivně hledat ve zdrojových textech jádra?

Pavel Janík ml., 11. prosince 1997

Problém hledání ve zdrojových textech jádra operačního systému Linux nabývá rozměrů hlavně v poslední době, kdy např. velikost posledních vývojových jader (2.1.7x) se pohybuje těsně pod hranicí 10 megabytů (samozřejmě po komprimaci).

Možná se zeptáte, proč vůbec (a co vlastně) hledat v takové spoustě zdrojových textů. Odpověď je jednoduchá – spousta uživatelů Linuxu se zajímá o detaily implementace jednotlivých funkcí operačního systému apod. Ve zdrojových textech se můžete mimo jiné dočíst i spoustu zajímavých informací o lidech, kteří Linux píší (např. věk Linuse Torvaldse). Zdrojové texty jsou místy i vtipné (zvláště pokud je jejich autorem Dave Miller ...).

```
 /* Amuse the user. */
 printk(
"                 \\|/ ____ \\|/\n"
"                 \"@'/ ,. \\`@\"\n"
"                 /_| \\__/ |_\\\n"
"                     \\__U_/\n");
```

Výpis 3: Dave S. Miller, arch/sparc/kernel/traps.c

Jak tedy efektivně hledat? Existuje několik způsobů. Asi ten nejméně efektivní je použít programu `grep` k hledání nějakého regulárního výrazu. To ale není řešení, protože program `grep` není napsán pro prohledávání desítek megabytů textu. K tomu je určen např. program `glimpse` [41] umožňující po předchozí indexaci velmi rychle vyhledávat pomocí regulárních výrazů. Popíšeme si tedy typický postup při vyhledávání.

Jak je již výše napsáno, musíme nejprve vytvořit počáteční indexový soubor. K tomu slouží program `glimpseindex`, který je součástí instalace balíku `glimpse`.

```
glimpseindex /usr/src/linux
```

Indexový soubor se vytvoří ve vašem domovském adresáři (pomocí parametru `-H` můžete nastavit jiný adresář). Nyní máme již vše připraveno k rychlému hledání. Zkusíme tedy pro zajímavost najít ve zdrojových textech výskyt slova BogoMIPS:

```
glimpse BogoMIPS
```

Výstupem by nám měl být seznam souborů obsahujících hledané slovo, včetně řádků, které jej obsahují. Programy `glimpse` a `glimpseindex` mají mnohem více možností, které bohužel nelze postihnout v tomto článku.

Další možností, jak vyhledávat ve zdrojových textech jádra, je použít službu *Linux Source Search* [42], která je vystavěna nad balíkem `glimpse`.

Pokud chcete např. hledat pouze proměnnou nebo funkci v jádře, je k dispozici i aplikace *Linux Source Index* [43].

Chyba v procesorech Intel Pentium – softwarové řešení

Jan Kasprzak, 21. listopadu 1997

V předchozím čísle Linuxových novin byla uveřejněna zpráva o tom, že byla objevena nová chyba v procesorech Intel Pentium. Tato chyba umožňuje běžnému uživateli zablokovat procesor tak, že je nutno provést studený start počítače. Této chybě se začalo říkat „F00F bug“ podle hodnoty prvních dvou bajtů chybné instrukce.

Dnes je situace podstatně lepší. Existuje způsob, jak softwarově danou chybu odstínit. Stojí za povšimnutí, že k dnešnímu dni (21. listopadu 1997) byla podle stránky společnosti Intel [40] k dispozici softwarová záplata pouze pro Linux a BSDI. (Hovořím zde o systémech s jejichž vývojáři Intel spolupracoval. Z ostatních systémů má softwarovou záplatu na tuto chybu například FreeBSD. Na uvedené stránce jsou zmiňovány tyto systémy nebo tito výrobci: BSDI, IBM, Linux, Microsoft, NCR, Novell, SCO, Sequent, SunSoft a Unisys.) Ostatní systémy včetně například Solarisu a SCO UNIXu mají příslušnou záplatu pouze „ve stádiu vývoje“ nebo „testů“.

Je zajímavé, co se zde píše o operačních systémech Microsoftu:

> “Microsoft has worked closely with Intel to understand and characterize the effects of the recently uncovered Pentium processor erratum, and we're in the process of studying the implementation of potential workarounds in order to meet the needs of our customers,” said Moshe Dunie Vice-President Windows Operating Systems Division at Microsoft. “Since this erratum can only be exploited by a program that was developed with malicious intent and deliberately uses this illegal instruction, following common-sense computing practices, such as not downloading or running executables from unknown sources, can protect a user from this problem.”
>
> ---
>
> „Microsoft a Intel těsně spolupracovali ve snaze pochopit nedávno odhalenou chybu procesoru Pentium a určit její dopad. Právě teď studujeme různé možnosti jak chybu obejít, abychom co nejlépe vyšli vstříc potřebám našich zákazníků“, řekl Moshe Dunie, viceprezident divize Windows Operating Systems ve firmě Microsoft. „Vzhledem k tomu, že tato chyba může být využita pouze programem, který byl napsán se zlým úmyslem a záměrně používá tuto nedovolenou instrukci, stačí k ochraně uživatelů před tímto problémem chovat se podle zdravého rozumu, třeba nestahovat a nespouštět programy z neznámých zdrojů.“

Tedy Microsoft se zřejmě vzdal myšlenky na to, že by jeho operační systémy mohly být někdy nasazeny ve víceuživatelském prostředí.

Nyní se budu věnovat mechanismu, jakým chybná instrukce pravděpodobně pracuje a navrhovaným softwarovým řešením. Samozřejmě není v silách jednoho člověka pochopit milióny tranzistorů v Pentiu, ale z chování CPU při této chybě lze leccos odpozorovat. Popisované mechanismy se tedy snaží odhadnout, co se v CPU přibližně děje při výskytu této instrukce. Nekladu si žádné nároky na správnost níže uvedených mechanismů, nicméně s jejich využitím se podařilo chybu izolovat a odstínit.

Inkriminovanou instrukci s operačním kódem `f0 0f c7` by bylo možné zapsat jako `lock cmpxchg8b` s adresovacím módem, který není pro tuto instrukci platný. Tato instrukce je „téměř legální“. Co to znamená? Na to, že jde o neplatnou instrukci, a že by se tedy měla vyvolat příslušná výjimka, procesor přijde až poměrně pozdě, když už je provádění instrukce uvnitř CPU v pokročilém stavu. Tento stav se vyznačuje tím, že procesor už zamknul datovou sběrnici, jak mu přikazuje instrukční prefix `lock`. A zde právě nastává problém. Procesor zjistí, že instrukce je neplatná, a chce vyvolat příslušné přerušení. Toto v první řadě znamená podívat se do tabulky přerušení (IDT, *Interrupt Description Table*), na které adrese začíná obsluha přerušení „neplatná instrukce“. Pokus o přístup k IDT ale selže, protože CPU má zamčenou sběrnici! Někde v tomto místě se procesor zastaví.

Jak takovouto chybu ošetřit? K přístupu na sběrnici nedochází, pokud se příslušná část IDT právě nachází v primární cache procesoru. První a nejstarší softwarové záplaty na tuto chybu fungovaly tak, že zajistily, aby se v cache objevil ovladač pro obsluhu výjimky „neplatná instrukce“ (například vykonáním nějaké skutečně neplatné instrukce), a pak zamkly obsah cache. Toto fungovalo, ale výkon procesoru bez primární cache klesl až někam k průměrným procesorům 486.

Pozdější úpravy využívaly toho, že jiné výjimky mají obsluhu zamčené sběrnice vyřešeny lépe než výjimka „neplatná instrukce“. Pokud se například stránka IDT, ve které byl vektor přerušení „neplatná instrukce“, nenachází v operační paměti, Pentium toto rozpozná a vyvolá výjimku číslo 8 – „dvojitý výpadek“ (*double fault*). Jak tohoto ale dosáhnout? Stačí zarovnat začátek IDT tak, aby prvních sedm položek IDT (pro výjimky 0 až 6, přičemž „neplatná instrukce“ má číslo 6) bylo v jedné stránce a ostatní byly v následující stránce paměti. Tedy aby tabulka IDT začínala těsně před koncem stránky. Dále je nutno upravit obsluhu výjimky číslo 8 tak, aby byla schopna analyzovat situaci a případně zavolat výjimky číslo 0 až 6. A poslední věc je označit první stránku IDT jako neplatnou.

V případě pokusu o vykonání neplatné instrukce se tedy zavolá výjimka číslo 8 „dvojitý výpadek“, která pak zpětně zjistí o co jde a zavolá příslušnou výjimku z intervalu 0 až 6. Toto může udělat dvěma způsoby. Buď to zjistí příslušnou informaci z chybového slova a pak zavolá příslušnou obsluhu jako funkci (což byl způsob, jakým byla situace řešena v Linuxu) nebo se první stránka IDT namapuje do paměti a daná instrukce se restartuje (samozřejmě až po kontrole, jestli náhodou nešlo o instrukci `f0 0f c7 cx`. Toto byl způsob, jakým doporučoval chybu izolovat Intel. První způsob měl problémy s obsluhou speciálních případů, jako je například ladící bod na místě neplatné instrukce a podobně. Ladící a krokovací funkce zde nefungovaly úplně korektně. Naproti tomu druhý způsob (navrhovaný Intelem) měl problémy se samomodifikujícím se kódem (například na SMP stroji – dojde k přerušení, namapuje se IDT, ale někdo jiný změní zatím instrukci, na které došlo k přerušení na `f0 0f c7 cx`. Restartuje se instrukce, ale IDT je namapovaná a procesor se zablokuje). Linus Torvalds napsal na linux-kernel

> Charles is right about the problems, although they don't actually apply to the current Linux version of the patch. They _do_ apply to the intel one, exactly because the intel one tries to be more clever, and fails in subtle ways.
>
> ---
>
> Co se týče těchto problémů, má Charles pravdu, ačkoli se vlastně nevztahují k současné Linuxové verzi toho patche. Vztahují se ale k Intelovské verzi, právě proto, že ten jejich patch se snaží být chytřejší a nezvládá to.
> Linus Torvalds

Obsloužení chyby F00F jedním ze dvou výše popsaných způsobů stálo přibližně 55 taktů při vyvolání kterékoli z výjimek 0 až 6. Záplaty provedené tímto způsobem se objevily v Linuxu 2.0.32 (vlastně již v pre-2.0.32-4) a 2.1.63.

Jak je vidět, odstranění takovéto chyby popisovaným způsobem nebylo nijak triviální. Naštěstí Linus ve spolupráci s Intelem vymyslel ještě lepší mechanismus odstínění chyby. V CPU se totiž v průběhu vyvolání výjimky pro neplatnou instrukci nejen neresetuje zamčení sběrnice, ale také následující přístup do paměti se provádí jako read-write přesně tak, jak by jej prováděla skutečná instrukce `lock cmpxchg8b`. Takže pokud je IDT namapována pouze pro čtení, dojde k zavolání výjimky „porušení ochrany paměti“. Příslušná obslužná rutina pak zjistí, že byl proveden pokus o zápis do IDT na položku číslo 6 a detekuje tak jednoznačně pokus o zneužití chyby F00F.

> The new workaround actually depends on the Intel bug not only forgetting to clear the "lock" state of the instruction, it also forgets to clear the fact that the instruction tries to
> do a read-modify-write cycle. So it will not only do the IDT access as a locked cycle, it will do it as a locked cycle that requires write permissions.
>
> ---
>
> Nová záplata vlastně závisí na chybě Intelu, která nejenže zapomene zrušit „lock“ stav instrukce, ale také zapomene změnit informaci o tom, že se instrukce pokouší přistoupit do paměti způsobem čti-modifikuj-zapiš. Takže procesor nejenže vykoná přístup k IDT se zamčenou sběrnicí, ale je vykonán zamčený přístup vyžadující navíc právo zápisu.
> Linus Torvalds

Tímto způsobem se kompletně odstíní chyba F00F a navíc tato záplata nemá žádný vliv na výkon systému (na rozdíl od předchozích řešení). V současné době je k dispozici záplata na jádro 2.0.32 i 2.1.x, využívající tohoto mechanismu.

Operační systém Linux byl jedním z prvních, které měly k dispozici záplatu na hardwarovou chybu Pentia. Tato chyba pro současný Linux již nepředstavuje žádné nebezpečí.

Linuxové noviny a jejich šíření

(Pavel.Janik@freesoft.cz). Autorská práva k jednotlivým článkům zůstávají jejich autorům.

Tento text může být šířen a tisknut 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 Pavel.Janik@freesoft.cz. 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. a Juraji Bednárovi za poskytnutí diskového prostoru pro Linuxové noviny.

Linuxové noviny můžete najít na akademické síti CESNET [44], na síti IBM Global Network na adrese [45] nebo na serveru časopisu Netáčik [46], který je připojen do slovenského SIXu.

Odkazy

[1] Urlmon
`http://sunsite.unc.edu/pub/Linux/apps/www/mirroring`

[2] Netscape News selector
`http://www2.awinc.com/users/bvdpoel`

[3] Spamcan
`http://consult.ml.org/~timb/spamcan`

[4] Xosview
`ftp://sunsite.unc.edu/pub/Linux/X11/xutils/status`

[5] XEmacs
`ftp://ftp.xemacs.org`

[6] Procdirs
`ftp://sunsite.unc.edu/pub/Linux/system/misc`

[7] MInfo
`http://sub994.sub.uni-goettingen.de`

[8] Cheap Bytes
`http://www.cheapbytes.com`

[9] Red Hat Software, Inc.
`http://www.redhat.com`

[10] Pam NT
`ftp://ftp.csn.ul.ie/pub/linux/pam_smb`

[11] Matcom V3
`http://www.mathtools.com`

[12] X games
`http://www.spinne.com/x/games`

[13] KBackup
`ftp://sunsite.unc.edu/pub/linux/system/backup`

[14] Matrice W3 System
`http://www.matrice.it/beta_eng.html`

[15] Xautolock
`ftp://sunsite.unc.edu/pub/Linux/X11/screensavers`

[16] Linux Counter
`http://counter.li.org`

[17] True Type font convertor
`http://www.bath.ac.uk/~ccsaw/fonts`

[18] Linux Network Address Translation
`http://linas.org/linux/load.html`

[19] Woven Goods for Linux
`ftp://ftp.fokus.gmd.de/pub/Linux/woven`

[20] Yutex editor
`ftp://sunsite.unc.edu/pub/Linux/apps/editors/X`

[21] Sofie 3D Game Engine
`ftp://sunsite.unc.edu/pub/Linux/games`

[22] Emacspeak
`http://cs.cornell.edu/home/raman`

[23] Debian Developer Program
`http://www.debian.org/developers_corner.html`

[24] Accelerated-X server
`http://www.xig.com`

[25] News2mail
`http://alexsh.home.ml.org/news2mail.html`

[26] Utility pro práci s RSA
`http://www.filesafety.com`

[27] Sangoma Technologies Inc.
`http://www.sangoma.com`

[28] Návrh stanov CZLUG
`http://www.linux.cz/czlug/stanovy.html`

[29] Anketa na téma CZLUG
`http://www.linux.cz/czlug/pruzkum.html`

[30] Debian Social Contract
`http://www.debian.org/social_contract.html`

[31] Milan Zamazal
`mailto:pdm@fi.muni.cz`

[32] Software AG
`http://www.sag.cz`

[33] A Quick Update with Linus Torvalds
`http://www.inxutil.com/inx_linux/inx_linus_intv.html`

[34] Zdrojové texty jádra
`ftp://ftp.kernel.org/pub/linux/kernel`

[35] Comprehensive Perl Archive Network
`http://www.perl.com/CPAN/`

[36] CPAN na síti TEN-34
`http://www.fi.muni.cz/ftp/pub/cpan`

[37] Hledání v archívu CPAN
`http://theory.uwinnipeg.ca/search/cpan-search.html`

[38] Red Hat PGP klíč
`ftp://ftp.fi.muni.cz/pub/linux/distributions/redhat/redhat-5.0/i386/RPM-PGP-`

[39] Intel F00F Bug
`http://support.intel.com/support/processors/pentium/ppiie/index.htm`

[40] Softwarové řešení F00F bug
`http://support.intel.com/support/processors/pentium/ppiie/software.htm`

[41] Glimpse
`http://glimpse.cs.arizona.edu`

[42] Linux Source Search
`http://www.math.muni.cz/cgi-bin/LSDSearch`

[43] Linux Source Index
`http://www.math.muni.cz/LSDIndex`

[44] Linuxové noviny na síti CESNET
`ftp://ftp.fi.muni.cz/pub/linux/local/noviny`

[45] Linuxové noviny na síti IGN
`ftp://ftp.inet.cz/pub/People/Pavel.Janik/noviny`

[46] Linuxové noviny na Slovensku
`ftp://netacik.sk/pub/linux/cz-noviny`