

# Linuxové noviny



## Úvodem

Pavel Janík ml., 10. dubna 1998

Dva měsíce uběhly jako voda a opět je tu další číslo Linuxových novin, které vám přináší ty nejzajímavější informace ze světa Linuxu a vůbec dění kolem něj. To vše nezanesečné komerčními dezinformacemi a novinářským hyenismem (© Dušan Korýtko: -) — pouhé čisté informace.

Co vám tedy přináší toto číslo? Především je nutno konstatovat, že je zaměřeno na praktické otázky dnešního linuxového života, tedy na programování webových aplikací, bezpečnost, framebuffer a další. Ale postupně:

Jako v každém čísle nám Radek Vybíral představí nejnovější programy pro Linux v rubrice [Čerstvé maso pro Linux](#). Pokud vás některý představený program zaujme, napište nám o tom.

David Rohleder si na chvíli odpočal od překladu Packages v Debianu a přečetl si knihu „Bezpečnost v Unixu a Internetu v praxi“. O svých zážitcích se s námi podělí v článku [Recenze: Bezpečnost v Unixu a Internetu v praxi](#).

Róbert Dobozy se ujal rubriky Linux Journal, a tak si opět můžete přečíst, co se v nejčtenějším linuxovém periodiku objevil v uplynulých měsících (viz články [Linux Journal — marec](#) a [Linux Journal — apríl](#)).

Nathan L. Cutler dokončil překlad úvodního dokumentu ke kernelu 2.2 ([Báječný svět jádra v. 2.2 \(pokračování\)](#)), a tak si jej můžete přečíst i vy, kdož nevládnete angličtinou. S novými kernely souvisí i možnost zprovoznit vaši novou a jistě skvělou grafickou kartu v X Window systému pomocí tzv. framebufferu. Jak na to popisuje Robert Vojta v článku [Nové grafické karty a Xka](#).

Toto číslo je zaměřeno na web a na vývoj aplikací pro web. Hlavními články jsou [Zope Miloše Průdka](#), [Co je to HTTP a HTML?](#) a [Popis knihovny cgihtml, aneb jak na CGI z C](#) Marka Sezemskeho.

Dalším zajímavým článkem je [Rychlo-školička pro Makefile](#) Marka Sezemskeho.

No a nakonec jsem si samozřejmě nechal nejčtenější rubriku Linuxových novin :-). [Zasmáli jsme se!](#) je opět na svém místě a v tomto čísle dokonce i s článkem Věroše Kaplana o věčné rivalitě Pythonu a Perlu (a nejen o tom).

Přeji příjemné počtení a nezapomeňte nám napsat, co vám v Linuxových novinách chybí. Nebo je vše skvělé? Tak to je dobře :-)

PS.: Uvidíme se na Prudké'99...

## Čerstvé maso pro Linux

Radek Vybíral, 15. dubna 1999

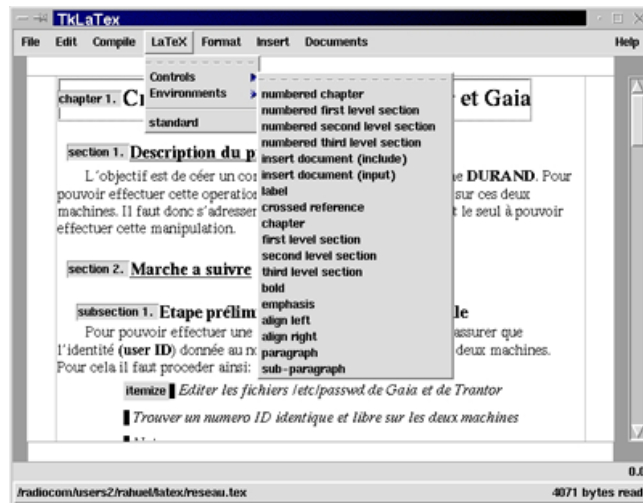
Již podruhé se na tomto místě setkáváte s nadílkou softwaru, který je možné provozovat pod operačním systémem Linux. Rozšířením oproti minulému dílu bude také uvedení softwaru, který je anoncován v diskusní skupině systému UseNet News: [comp.os.linux.announce](#), zkráceně c.o.l.a.

## Mít svůj vlastní sen... ehm, firewall

Jistě jste se dostali nebo třeba dostanete do situace, kdy vás někdo vyzve, abyste vyrobili z Linuxu zabezpečený počítač — firewall. Nástroje i podpora přímo v kernelu na to existují, ale největší problém je s definováním všech potřebných práv pro vstup/výstup do vnitřní sítě, která je potřeba hlídat. Pro řešení tohoto úkolu máte nyní pomocníka, skript jménem Mason (1), který dokáže řešit vše potřebné. Skript využívá logů od programů ipchains a ipfwadm, sleduje provoz na síti, má podporu pro PPP spojení a spoustu dalších užitečných vlastností. S jeho pomocí lze snadno sestavit potřebná pravidla pro provoz firewallu.

## Vždyť přeci psát v TeXu je tak snadné

TkLaTeX (2) je WYSIWYG nadstavba nad LaTeXem, která velmi usnadňuje psaní textů. Umožňuje vkládat nejpoužívanější formátovací značky přímo z menu. Pokud se přeci jen dostanete do situace, že editor nestačí, lze dokument editovat v klasickém módu. S pomocí toho editoru můžete vkládat i ty nejsložitější matematické vzorce.



## MP3 Radio? Ale jistě, ozvalo se ze sluchátka

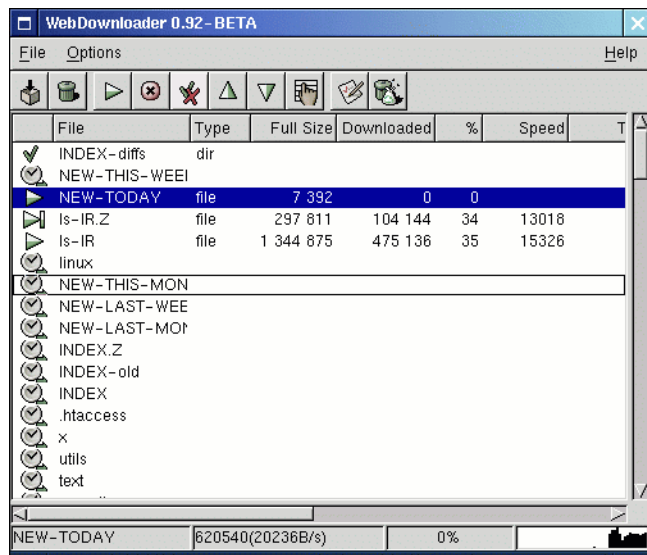
Pokud máte už nějakou sbírku skladeb v populárním formátu MP3, pro přehrávání musíte mít soubory MP3 přístupné na CD-ROM nebo na disku. IceCast (3) je server pro vysílání MP3 hudby prostřednictvím protokolu HTTP. Takže si můžete vyrobit MP3 rádio, které mohou využívat také vaši kolegové v práci. Na server může být napojeno více zdrojů, jako například výstup z radiokarty a podobně. Určitě stojí za pozornost, doporučuji z vlastní zkušenosti!



ry můžete také spravovat pomocí Project Managera a možností vypsaní ladících informací do zvláštního okna. Všichni programátoři všech zemí, **STAHUJTE!**

### Stahování programů načasovat: TEĎ

Stahujete také občas programy z Internetu? Pokud ano, pak jistě znáte problémy s tím, že se stahování přeruší, soubor je nekompletní. Večer jsou většinou linky volnější a tak je možné provést download večer. Pokud znáte program GetRight pro Windows, pak jistě nemusím nic dodávat, přesně tyto služby zajišťuje WebDownloader (13). Program využívá grafického prostředí GTK knihoven, takže práce s ním je velice příjemná.



### Vypalování CD je přeci zábava

Pokud vytváříte a vypalujete CD-ROM disky, pak jistě používáte řádkové utility mkisofs a cdrecord. Ne všem se ovšem zdá práce s těmito programy pohodlná, a proto je možné spustit grafickou nadstavbu nad těmito programy s názvem KisoCD (14). Jak název napovídá, jedná se o program běžící pod prostředím KDE, s jehož pomocí můžete vytvářet image pohodlně a přehledně pomocí menu. ■

- 1 Mason  
<http://users.dhp.com/~whisper/mason/>
- 2 TkIA<sub>T</sub>P<sub>X</sub>  
<http://gaia.univ-rennes1.fr/tklatex/>
- 3 IceCast  
<http://icecast.org/>
- 4 onShore TimeSheet  
<http://www.onshore-timesheet.org/>
- 5 PoPToP  
<http://www.moretonbay.com/vpn/pptp.html>
- 6 LREAD  
<http://www.it.fht-esslingen.de/~zimmerma/lread.htm>
- 7 Kdevelop  
<http://www.cs.uni-potsdam.de/~smeier/kdevelop/index.html>
- 8 Ted  
<http://www.nllgg.nl/Ted/>

- 9 LessTif  
<http://www.lesstif.org>
- 10 SpeedyCGI  
<http://www.daemoninc.com/speedycgi/>
- 11 PPDD  
<ftp://ftp.gwdg.de/pub/linux/misc/ppdd/>
- 12 TCL Developer Studio  
<http://www.star.spb.ru/~small/tcldev/>
- 13 WebDownloader  
<http://www.krasu.ru/soft/chuchelo/>
- 14 KisoCD  
<http://www.uni-karlsruhe.de/~um12/kisocd-index.html>

## Recenze: Bezpečnost v Unixu a Internetu v praxi

David Rohleder, 8. března 1999



Na konci minulého roku vyšlo několik knih týkajících se problematiky počítačové bezpečnosti. Rád bych napsal pár slov o nejobtímnější z nich. Je to překlad druhého vydání knihy Practical UNIX & Internet Security od Simsona Garfinkela a Gene Spafforda, který vyšel pod českým názvem Bezpečnost v Unixu a Internetu v praxi. Překlad vydalo v poslední době velmi aktivní nakladatelství Computer Press.

Tato kniha čítající více než 900 stran je v oblasti unixové bezpečnosti opravdovou biblí. Autoři se snaží seznámit čtenáře s celou širší problematikou bezpečnosti unixových systémů, od základního vysvětlení bezpečnostních opatření v systému, přes fyzickou bezpečnost, zálohování a bezpečnost jednotlivých programů až po právní problematiku.

Kniha není jen suchým popisem různých systémů bezpečnosti, její tvůrci jsou odborníci se značnou praxí, a tak je kniha proložena různými zajímavými příběhy a historkami ze života.

Tématiky je celý text rozdělen do šesti částí a několika příloh.

Část první — Základy počítačové bezpečnosti popisuje vznik a historii operačního systému Unix a obsahuje vysvětlení některých základních termínů. Věnuje se také volbě bezpečnostní strategie a managementu rizik.

Část druhá — Zodpovědnost uživatelů se věnuje uživatelským účtům, uživatelům, skupinám a superuživateli. Součástí je také popis unixového souborového systému včetně některých nestandardních rozšíření (ACL — „seznam řízení přístupu“). Poměrně velká část této kapitoly je věnována kryptografii. Na téměř padesáti stranách jsou rozloženy základy symetrických a asymetrických šifer, digitálních podpisů a hashovacích funkcí.

Část třetí — Bezpečnost systému se věnuje základním bezpečnostním praktikám, ke kterým patří zálohování, ochrana účtů, kontrola integrity a protokolovací mechanismy. Část je také věnována fyzické a personální bezpečnosti počítačových systémů, což je zajímavé zejména pro správce větších systémů.

Část čtvrtá — Bezpečnost sítě a Internetu popisuje problémy, které vznikají při připojení počítače k některému z druhů sítí, telefonní sítí počínaje a Internetem konče. Jsou zde dosti podrobně popsány systémy UUCP, WWW, RPC, NFS, NIS, NIS+ a Kerberos.

Část pátá — Pokročilá témata představuje systémy



ochrany vnitřních sítí před vnější sítí. Jsou to především firewally, wrappery a proxy a část je také věnována problematice tvorby suid programů.

Část šestá — Postup při napadení. Byli jste úspěšně napadeni? Tak toto je kapitola přesně pro vás. I když doporučuji přečíst před tím než se to stane. Jsou zde probrány postupy při napadení od odhalení průniku, zastavení útoku, až po zotavení se. Rovněž jsou zde shrnuty možnosti využití právních cest k potrestání útočnicka. Bohužel se vztahuje pouze na USA, takže našinec asi nebude schopen využít služeb FBI nebo tajné služby. Už jste přečetli skoro celou knihu a ještě nejste paranoidní? Nevadí, i na tuto možnost autoři mysleli a na závěr zařadili kapitolku o tom, komu můžete věřit.

V přílohách jsou pak uvedeny informace o důležitých souborech v systému, síťových portech a dalších tištěných a elektronických zdrojích informací.

### Hodnocení?

Autorům se opravdu podařilo obsáhnout velkou část problematiky bezpečnosti unixových systémů. Nezacházejí si ce příliš do podrobností, a také zde nenajdete návody pro útoky na počítačové systémy, ale prezentují množství mechanismů, jak se těmto útokům bránit.

Většina témat je probrána opravdu pečlivě, jistou výjimku podle mne představuje popis systému Kerberos a firewallů.

U Kerbera chybí podrobnější popis celého systému. Bez tohoto vysvětlení není téměř možné plně pochopit výhody jeho použití.

U části popisující konstrukci firewallů mi chybělo vysvětlení základních pojmů, např. hlaviček TCP/IP, a přesto autoři operují s využitím některých dost speciálních polí. Pokud se tedy chcete o firewallech dozvědět více, doporučuji jinou knihu (např. Chapman, Zwicky: Firewally, principy budování a udržování nebo Bellovin, Cheswick: Firewally a bezpečnost Internetu).

Překlad a sazba překladu si také zaslouží několik výtek. V překladu je poměrně dost chyb, v některých případech jsou takového charakteru, že to znemožňuje pochopení textu. Záměna mocnitelů u systému asymetrické kryptografie je toho jasným příkladem.

První (a dosti vážné) chyby se překladatel nebo sazeč dopustil ještě před samým začátkem knihy — v tiráži. Nejenže změnil autory knihy, dokonce přeložil text „Practical UNIX & Internet Security“ do češtiny jako „Microsoft Exchange in Business“ :-)

Mezi chyby sazeče bych zařadil výběr fontů v příkladech tak, že se slévají pomlčky a téměř není možné rozpoznat zpětné apostrofy od apostrofů normálních. V jednom z vložených článků chybí poslední řádek. A rozdělení slova Berkeley-y nepůsobí také nejlepším dojmem.

Přes všechny tyto výhrady si myslím, že by tato kniha rozhodně neměla chybět v knihovničce žádného správce unixového systému.

Celkové hodnocení: 4/5

## Odezva uživatele Linuxu

Lubomír Čevela, 28. března 1999

Jsem jedním z účastníků akce „Linux pod stromeček“, která se konala v Ostravě dne 13.12.1998. Nepatřím asi mezi typické uživatele tohoto systému. Asi před rokem jsem si

něco o Linuxu přečetl, zkusil jej a zcela propadl jeho kouzlu. Nevlastním žádný server a Linux mám zatím pouze doma na svém počítači. Myšlenka, že stovky programátorů celého světa se spojily pomocí Internetu a výsledkem jejich práce je operační systém výborné kvality, patří podle mého názoru do oblasti sci-fi. A přece Linux je tady, někým obdivovaný, někým zatracovaný, ale již uznávaný většinou firem, které v počítačovém světě něco znamenají.

Ještě se vrátím k akci v Ostravě. Přišel jsem trochu pozdě, vše již probíhalo a ocitl jsem se v atmosféře připomínající mi nejvíc studentská léta. Po přátelském rozhovoru s linuxovými odborníky se některé mé „záškysy“ v konfiguraci Linuxu staly minulostí. Za mříný peníz jsem obdržel CD s nejnovější verzí Linuxu a druhý z několika aplikacemi, u kterých stahování z Internetu po komutované lince se mi zdálo slovy klasika „řešením poněkud nešťastným“. Velice cenné bylo také setkání s běžnými uživateli, kteří jsou na tom se znalostí Linuxu podobně jako já.

Jako reakci na výzvu jednoho z pořadatelů této akce připojuji pár poznámek k mým problémům s Linuxem:

1. Největší problém, na který jsem narazil, byl problém s češtinou, především s počestněním systému X a StarOffice. Problémy jsou na Internetu skvěle dokumentovány, nejlépe je začít na adrese <http://www.penguin.cz/czech-howto>, přesto mohou představovat pro začínajícího uživatele nemilý rébus. Ideálním řešením by byla distribuce Linuxu, která by měla i v „X-ech“ tento problém zcela vyřešený.
2. Doma jsem trochu zápasil s „rozcházením“ zvukové karty. Mám SB AWE 32. Dnes vím, že stačilo pouze v konzole napsat „setup“ a nechat se vést konfiguračním programem, který není složitější než třeba u Duke Nukema 3D.
3. Zatím bohužel pod Linuxem chybí 100% český nebo počestněný kancelářský balík za přijatelnou cenu. Taková linuxová „Kancelář 602“ by se mi (a myslím, že nejenom mi) dost hodila. Používám StarOffice 5.0, doplněný modulem pro převod diakritiky CPcode od pánů Juruse a Kristofa. Až na mírné problémy s některými nečeskými fonty a s občasným padáním programu při tisku to docela jde. Přestože program čte výborně dokumenty např. Wordu 6.0 nebo Wordu 95, s Wordem 97 si zatím moc nerozumí.
4. Existuje sice spousta Window managerů, ale mně se nejvíc zamlouvají KDE a Gnome. Měl jsem dříve kombinace Afterstepu a panelu z Gnome, dnes používám asi méně obvyklou kombinaci KDE a panelu z Gnome. Ideální by bylo, kdyby si uživatel mohl během instalace systému vybrat jaký Window manager použije a nebylo mu nabídnuto jen uživatelsky dost nevládné fvwm nebo nezvyklý Afterstep. V tomto případě toužebně očekávám nový RedHat s WM Gnome (datum vydání je ohlášeno firmou Red Hat na 15.4.1999).
5. Při instalaci programů stažených z Internetu mám obvykle potíže s nesprávnou verzí programem požadované knihovny. V Linuxu jsem se naštěstí nesetkal s běžnou praxí pod Windows, tedy že vám nějaký agresivní program přepíše systémové knihovny. To pak vede k nesprávným funkcím systému, případně dalších aplikací. Ocením, když autor programu připojí k programovým souborům i stránku s odkazy na potřebné knihovny, eventuelně nabídne staticky linkovanou verzi.



6. Vytisknout něco z Linuxu, pokud zrovna nemáte post-scriptovou tiskárnu (mám Canon BJC-240) bývá někdy rovněž hlavolam.

Na závěr chci připojit poděkování lidem, kteří se podílejí na vytváření, dokumentaci a dalších věcech souvisejících s Linuxem, protože bez nich by opravdu patřil tento systém do oblastí vědeckofantastické literatury. ■

## Zope

Miloš Průdek, 11. dubna 1999

### Úvod

Každý správce webu, který usiluje o interaktivitu s uživatelem, se jednoho dne musí zamyslet nad přechodem na dynamické generování stránek. Základním nástrojem pro dynamické stránky jsou CGI skripty. Určitá výhoda CGI skriptů je v tom, že můžete použít programovací jazyk, který už umíte (C++, Perl). Jejich hlavní nevýhodou bývá nutnost vyvolat novou instanci pro každý skript. Z toho vyplývá větší spotřeba paměti a systémových prostředků.

Rychlejší a úspornější řešení je použití jazyka, který je přímo vložen do zdrojového HTML textu. Na serveru běží interpret, který čte HTML text, provádí příkazy, které jsou mu určeny, a uživateli je pak servírováno „čisté“ HTML, tj. bez příkazů určených pro interpret. Tuto technologii používá například ASP, vytvořené firmou Microsoft. Bohužel, ASP je proprietární řešení, závislé na webovém serveru IIS této firmy. Ve světě Linuxu existuje velmi dobrá alternativa: PHP3. Tento jazyk nabízí prakticky stejné možnosti jako ASP, ale není vázán na platformu MS Windows. PHP3 je populární zvláště v kombinaci s Open Source databázovými SQL systémy, jako je MySQL a Postgres.

Návrh stránek pomocí PHP3 nebo ASP ale není zrovna snadný, a připomíná všechno, jen ne WYSIWYG. Mezi komerční řešení, které se snaží tento problém řešit automatickým generováním ASP stránek s vizuálním systémem návrhu a programování patří například systémy Sun Microsystems NetDynamics a Allaire ColdFusion. Tyto poměrně silné nástroje jsou bohužel dosti drahé, a navíc se snaží programátora přivázat k proprietární platformě. První a zatím zřejmě jedinou alternativou k těmto systémům je právě Zope.

Historie systému Zope je poměrně zajímavá z licenčního hlediska. Firma Digital Creations vyvíjela tento systém (tehdy ještě pod názvem Principia) jako proprietární a tedy bez zveřejnění zdrojových kódů. V souvislosti s kapitálovou injekcí, kterou nedávno získala, přešla po dohodě s investorem na model Open Source. Zpočátku ale vyžadovala, aby každý website postavený na Zope uváděl ikonku s odkazem na Digital Creations. Na nátlak uživatelů od tohoto požadavku počátkem tohoto roku upustila, a nyní je Zope oficiálně pod licenci GPL.

Kromě vlastního Zope existuje množství uživatelských nadstaveb, jako jsou databázové adaptéry (tedy rozhraní) pro MySQL, Postgres, Oracle atd., a Digital Creations pokračuje ve vývoji vytvářením dalších aplikačních systémů nad Zope. Tyto systémy (například ZTables pro komunikaci s .dbf soubory a další databázové úkoly) se už prodávají, ale cílem Digital Creations je po určité době převést i tyto doplňky na Open Source — a začít jiné nadstavbové systémy.

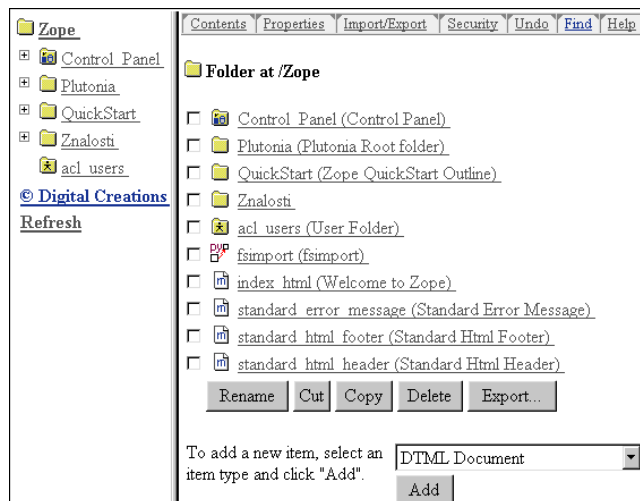
### Instalace

Instalace je poměrně jednoduchá, a proto jen naznačím, co vás čeká. Nejprve musíte zjistit, zda máte na počítači Python. Pokud ne, nainstalujte jej, nebo si stáhněte distribuci Zope s integrovaným Pythonem. (Jestli si chcete vyzkoušet Zope na Windows ještě před tím, než jej pustíte na váš linuxový server, stáhněte si z (1) binární distribuci Zope pro Windows. Tato distribuce obsahuje i Python, a instalace je triviální. Distribuce obsahuje i jednoduchý, ale postačující webserver, takže po spuštění Zope stačí jen nasměrovat váš webserver na příslušný port a adresář na *localhost*). Jakmile máte Python, můžete nainstalovat Zope, a připravit jej na spolupráci s vaším webserverem. Tady se bude postup lišit podle toho, jaký webserver používáte. Pro Apache stačí nastavit příslušné RewriteRule, přesně jak je popsáno v dokumentaci Zope. Samozřejmě musíte mít v Apachi zakompilovaný modul `mod_rewrite` (já jsem neměl :-)) a pak už stačí poslat signál `-HUP` procesu `httpd`, a nasměrovat prohlížeč na `http://ip_adresa_serveru/Zope`.

Pokud se vám nepodaří instalace napoprvé, bude pes pravděpodobně zakopán v příkazu `RewriteRule`. Naneštěstí modul `mod_rewrite` je velice mocný a složitý, takže spíš než metodu pokusů a omylů doporučuji nahlédnout do dokumentace. Ta je díky Bohu perfektní a obsahuje i odkaz na hromadu příkladů. Když budete šikovni, budete mít instalaci hotovou nejpozději za hodinu.

### Popis rozhraní

Na rozdíl od komerčních produktů, jako je ColdFusion, nemá Zope klientskou část. Místo toho se veškerá komunikace dělá přes webové rozhraní. Na jedné straně to znamená plnou platformní nezávislost (Zope server je na Linuxu, ale ovládáte jej z Windows nebo jiného OS), na druhé straně to rozhodně není tak rychlé a pohodlné. Už jen proto, že web prostě nepodporuje akce „drag and drop“. Komunikace přes web je také o něco pomalejší než přímá manipulace na lokálním stroji, která se používá u komerčních řešení.



Základní jednotkou při práci se Zope je Folder (adresář, složka). Existují obvyčejné foldery a speciální foldery. Jsou dva druhy speciálních folderů: Control Panel (jeden pro celý systém), a `acl_users` (může, ale nemusí být v každém folderu). Na obrázku je základní vzhled Zope, včetně speciálních folderů, folderu QuickStart (rychlý úvod do Zope),



folderů které jsem vytvořil v rámci pokusů se Zope (Plutonia a Znalosti), standardní hlavičky a patičky, standardní chybové zprávy, a modulu fsimport, který popíšu níže.

Všechny prvky vašeho webu se ukládají do obyčejných folderů, a to jako objekty DTML document, DTML method, Folder, Image, File, Session, Principia Draft, External Method, Mail Host, Z Gadfly Database, Z SQL Method, Z Search Interface. Kromě toho můžete krátké úseky HTML kódu ukládat do Properties. Některé tyto prvky popíšu níže.

### Folder

Základní stavební prvek Zope. Připomíná běžný adresář filesystému, ale má podstatně propracovanější systém práv, který si promítnete klepnutím na záložku Security. Folder má 29 práv (linuxový filesystém má tři: read, write, execute). V každém folderu můžete definovat libovolný počet uživatelů, nebo skupin uživatelů (takzvané role), a nastavit nejen co může která skupina uživatelů dělat, ale také která práva jsou zděděna z nadřazeného folderu. Implicitně se práva dědí směrem dolů. Díky této možností lze z kořenového folderu udělat několik folderů, které odpovídají například divizím firmy, a z nich vytvořit další foldery podle oddělení jednotlivých divizí. Některým oddělením přitom můžete dovolit vytváření vlastních folderů. Jinými slovy můžete delegovat pravomoci (a odpovědnost) na uživatele. Každý uživatel je identifikován jménem a heslem.

Vzhledem k tomu, že k ovládní Zope můžete použít libovolný browser, nic nebrání tomu, aby do vašeho webu přispívali lidé geograficky libovolně vzdálení.

### DTML Document

je v podstatě HTML dokument s vloženými DTML příkazy. Což není definice kruhem :- ) DTML je jazyk s podobnými možnostmi jako má PHP3, do HTML se vkládá jako poznámka. Můžete provádět běžné operace s proměnnými, vkládat úseky z jiných DTML dokumentů, používat SQL a podobně. Kompletní popis jazyka je nad rámec tohoto úvodního článku (a pravděpodobně i nad rámec jakýchkoliv následujících článků :- ), ale jazyk je vcelku dobře zdokumentován přímo na domovských stránkách Zope.

DTML příkazy jsou velice podobné jazyku Python, a v případě potřeby můžete do DTML dokumentů vkládat přímo příkazy Pythonu.

Každý DTML dokument má pět práv, která můžete nastavit, včetně přiřazení k rolím.

### Structured Text

Báječný vynález Zope je Structured Text, který umožňuje zapsat běžné formátování HTML uživatelsky přítulným způsobem. Například místo `<b> tučný text </b>` napíšete prostě `*tučný text*`. Místo `<i> skloněný text </i>` stačí jen `**skloněný text**`. Celkem je k dispozici okolo deseti ekvivalentů různých HTML příkazů. Díky jednoduchému zápisu se vám podaří přesvědčit i sekretářku, aby přispívala do vašeho webu. Stačí když ukážete, o kolik sekund je rychlejší zmáčknout dvě hvězdičky, než například v MS Wordu pracně myší jezdit na ikonu tučného písma. Díky systému práv se nemůže stát, že by uživatel nechtěně porušil celkovou strukturu webu (tedy, pod podmínkou že máte ta práva správně nastavená :- )



### Z SQL Method

umožňuje práci s SQL, a to tak důkladně, že si popis nechám na další pokračování.

### fsimport

Přepisovat váš statický web by bylo protivné a zdlouhavé. Museli byste do Zope zadat všechny HTML stránky a všechny obrázky. Proto se bude určitě hodit pythonovská externí metoda (čti: doplněk do Zope), která udělá automatický import adresáře do Zope. Utilita se jmenuje `fsimport.py`, a musíte si ji stáhnout zvlášť z webu Zope a umístit na server do adresáře `/Extensions`. Pak ji ještě musíte přidat do Zope, jako External Method (napíšete „fsimport“ do všech kolonek). Nejlépe do kořenového adresáře, protože díky dědičnosti bude viditelná i ve folderech nižších úrovní.

Předpokládáme, že váš statický web se nachází ve filesystému na `/home/ja/mujweb/`, a že máte v Zope vytvořený folder `/uzivatele/katka`. Dále předpokládáme, že Zope se nachází na `http://www.kdesi.cz/Zope`. Naměřte váš browser na:

```
http://www.kdesi.cz/Zope/uzivatele/katka/\
fsimport?fsdir=/home/ja/mujweb/
```

a Zope se postará o zbytek. To jest, buď provede import a zobrazí stručnou hlášku, nebo na vás vyplivne deset rádků o chybě. Pokud nastane chyba, je problém v tom, že musíte mít práva world searchable po celé cestě až k importovanému webu.

Utilitku `fsimport` budete asi používat i při dalších úpravách. Určitě bude rychlejší navrhnout nově přidávané stránky v dobrém vizuálním HTML editoru, a pak vtáhnout do stávající struktury.

### Undo, Sessions, Upload

Zope si pamatuje vše, co mu uděláte a vše můžete vrátit zpět. Protože je to víceuživatelský systém, pamatuje si i pachatele. Příklad řádku v žurnálu:

```
/ppm/nabidky/thumb/manage_addImage by prudek\
on Mar 15 at 22:56:02
```

Můžete vrátit jednu z akcí, více akcí, a dokonce nespojitou množinu akcí.

Když spustíte Session, můžete dělat úpravy webu, které se objevují přímo na webu, ale jen vám. Zbytek světa vidí původní verzi. Jakmile jste spokojeni, provedete aplikaci změny do „živého“ systému.

Hotové stránky můžete do Zope přidat výše popsanou utilitou `fsimport.py`, což ovšem vyžaduje přístup k filesystému (třeba i zprostředkovaný přes ftp). Jednotlivé soubory můžete přímo uploadovat přes browser (čili odkudkoliv) klepnutím na záložku Upload. Pozor, nefunguje v MSIE 3.0x, zřejmě kvůli špatné implementaci JavaScriptu. Řečeno reklamním heslem Paegasu: O důvod víc proč nebýt s nimi :- )

### WebDAV

je protokol vyvinutý původně Microsoftem. Je to systém pro inteligentní komunikaci mezi tvůrcem obsahu a webovým serverem. Podporuje víceuživatelský přístup, drag and

drop, klepnutí na pravé tlačítko pro zobrazení vlastností. Stručně řečeno je o hodně lepší než ftp. Taková řešení sice už existují pro komerční webové servery, ale jsou proprietární. Protože WebDAV je volně použitelný, jedná se o standard, který pravděpodobně převáží. Zajímavé je, že Zope se zřejmě stal prvním funkčním serverem normy WebDAV. Tedy aspoň podle informací na domovské stránce, protože oficiální release ještě není. Zájemci ale mohou stáhnout pracovní verzi z CVS.

Informace o WebDAV najdete v RFC 2518 a na (2). Dobrý článek o WebDAV a Zope je na (3).

### Zope zevnitř

Zope je napsáno v jazyku Python. Říká se, že nemusíte být blázen abyste mohli používat počítače, ale rozhodně to není na škodu. Totéž se dá říct o používání Zope a znalosti Pythonu. I když se dá Zope používat, aniž byste uměli Python, v současnosti se vám bude Python hodit v mnoha případech, protože existuje mnoho úkolů, které bez kousku programu v Pythonu prostě neuděláte. Naštěstí je to kvalitní, objektivně orientovaný jazyk, který se dá poměrně rychle zvládnout.

### Další dokumentace

Zope se vyvíjí natolik rychle, že dokumentace je trochu pozadu za programováním. Dá se nicméně říct, že podstatné věci jsou zdokumentované, a vše je k nalezení na domovské stránce. FAQ teprve vzniká, ale pracovní verze není k zahození; najdete ji na (4).

Stávající dokumentace je bohužel trochu podivně formátovaná, a nepřehledná. Nedá se například srovnat s HTML dokumentací k serveru Apache. Přesto je plně použitelná. Dalším perfektním zdrojem informací je emailová konference (5), která má v současné době přes třicet zpráv denně.

Autor je zaměstnancem společnosti Sonet Online s.r.o. (6). ■

- 1 Zope  
<http://www.zope.org>
- 2 WebDAV  
<http://www.ics.uci.edu/pub/ietf/webdav/>
- 3 Článek o WebDAV a Zope  
<http://webserver.cpg.com/wa/4.4/>
- 4 Zope FAQ  
<http://zdp.zope.org>
- 5 Konference o Zope  
<mailto:zope@zope.org>
- 6 Sonet Online, s.r.o.  
<http://www.sol.cz>

## Linux Journal — marec

Ing. Róbert Dobozy, 9. dubna 1999



Po dlhom čase začíname znova v Linuxových novinách s prehľadom zaujímavostí a novinek v Linux Journale. Rozhodol som sa poňať to tak, aby z toho mali niečo aj tí, ktorí nemajú možnosť tento časopis čítať. Pre lepšiu orientáciu v časopise budem v zátvorkách uvádzať názov a číslo strany.

Päťdesiatdeviate číslo Linux Journalu sa

nesie v znamení internacionalizácie a novovznikajúcich trhov. David Cinege v článku „The Linux Router Project“ (The Cutting Edge, 14) popisuje, ako už z názvu vyplýva, vlastnosti, používanie, výhody a spôsob fungovania distribúcie Linuxu s názvom Linux Router Project (1). Pre tých čo ešte nevedia, je to jednodisketová distribúcia Linuxu určená hlavne pre prácu v sieti.

V tretej časti článku „Creating a Web Based BBS“ (At the Forge, 22) nás R. M. Lerner oboznámi s tým ako sa dá pomocou MySQL a Perlu implementovať jednoduché WWW diskusné fórum s vyhľadávaním. Zdrojové texty môžete nájsť tu (2).

Nosným článkom tohto čísla je asi „Alphabet Soup: The Internationalization of Linux, Part 1“ (Feature, 30) od pána Turnbulla. Môžete tu nájsť popis problematiky, ktorá súvisí s internacionalizáciou a lokalizáciou rôznych častí Linuxu. Teda poviem vám, tí Číňania a Japonci to majú OMNOHO komplikovanejšie ako my. Online verziu oboch častí článku môžete nájsť aj na WWW stránkach Linux Journalu (3), (4).

Vedeli ste o tom, že Linux vie riadiť auto? Že nie? No do nedávna som to nevedel ani ja, ale M. Bertozzi, A. Broggi a A. Fascioli z Parmskej univerzity ma v článku „Autonomous Vehicles“ (Feature, 40) o tom presvedčili. Vraj nám na to stačí Pentium MMX a 32 MB pamäte. Takto riadené auto dokázalo prejsť viac ako 2000 km po talianskych cestách, pričom dosahovalo priemernú rýchlosť od 80,2 až po 89,8 km/h a maximálnu rýchlosť až 123 km/h. Viac informácií, ako aj video z cesty môžete nájsť na ARGOvej domovskej stránke (5).

Po prečítaní „Internationalizing Messages in Linux Programs“ (Feature, 46) od P de Maura som konečne pochopil, ako sa používa GNU gettext a ako môžem vytvoriť program, ktorý bude vedieť komunikovať v rôznych jazykoch. Ak nevíete čo si máte dnes ráno na seba obliecť, tak si skúste obliecť počítač. Ano počítač. Dr. S. Mann nás v „Mediated Reality: University of Toronto RWM Project“ (Feature, 50) oboznámi o tzv. *WearCompe*, ktorý umožňuje vytvoriť si tzv. personálnu realitu resp. sprostredkovanú realitu. Implementácia má podobu okuliarov, ktoré sprostredkujú a upravujú to čo cez nich vidieť. Tým umožňujú odfiltrovať nechcené obrazy (napr. reklamu). *WearComp* sa dá pripojiť k Internetu 2Mbps rádiovou linkou a tak umožňuje zdieľať virtuálnu realitu s inými osobami. Ako obyvateľ bývaleho „východného bloku“ ma mierne zarazila naznačená možnosť monitorovanie a riadenia toho, čo daná osoba vidí (Veľký brat prichádza?), ale myšlienka je to rozhodne zaujímavá a určite by našla uplatnenie aj v bežnom živote. Viac informácií môžete nájsť na domovskej stránke *Wearcompu* (6).

V článku „Polyglot Emacs 20.4“ od J. Babcocka sa dozvieme, že Emacs s Mule rozšírením je naozaj nevšedný nástroj aj na písanie viacjazyčných textov a pre preklad napr. z japončiny do angličtiny. Pokiaľ sa zaujímate o bezpečnosť počítačov, tak vás iste zaujmú D. Corcoran, D. Sims a B. Hillhouse v „Smart Cards and Biometrics: Your Key to PKI“. Prečítate si tu o tom ako sa dajú inteligentné karty a biometria použiť na autentizáciu používateľov. Používa sa na to inteligentná karta s čítačom odtlačkov prstov. Dokonca existujú aj modely, ktoré vedia zistiť či je daný odtlačok z ozajstného prstu (nie zo silikónového odtlačku) alebo dokonca či je daný prst živý alebo nie. Pre Linux existuje projekt s názvom *MUSCLE* (7), ktorý sa snaží vytvoriť manažér pre čítačky smart kariet.

Ďalej si tu môžete prečítať interview s Chrisom Brownom z Learning Tree International, ktoré robí linuxové kur-



zy (Interview, 72), prehľad dvoch linuxových aplikácií, ktoré sa používajú v programe Medzinárodnej vesmírnej stanice (Forum, 74) na riadenie automatického vesmírneho modulu ATV. Samozrejme tu nechýbajú ani recenzie kníh a prehľady nových produktov ako aj iné drobné články a kopece reklám. Niektoré články z tohto čísla môžete nájsť online na stránkach linuxjournal.com (8). ■

- 1 Linux Router Project  
<http://www.linuxrouter.org>
- 2 Zdrojové texty  
<ftp://ftp.ssc.com/pub/lj/listings/issue57/3193.tgz>
- 3 Online verze článku  
<ftp://ftp.ssc.com/pub/lj/listings/issue58/3252.tgz>
- 4 Online verze článku  
<ftp://ftp.ssc.com/pub/lj/listings/issue59/3296.tgz>
- 5 ARGO  
<http://millemgia.ce.unipr.it/ARGO/>
- 6 WearComp  
<http://wearcam.org/>
- 7 Muscle  
<http://www.linuxnet.com/smartcard/>
- 8 Linux Journal 59  
<http://www.linuxjournal.com/issue59/>

## Linux Journal — apríl

Ing. Róbert Dobozy, 9. dubna 1999



Aktuálne, šesťdesiate číslo sa zameriava na network computing. Ak sa chcete dozvedieť čo je to Frame Buffer, na čo to je dobré a aké sú jeho výhody a nevýhody, tak si nezapodnite prečítať článok J.Pranevicha s názvom „Linux 2.2 and the Frame Buffer Console“ (Kernel Korner, 12).

Je Perl váš obľúbený programovací nástroj? Ak áno, tak nevynechajte popis ako písať efektívne CGI skripty s mod\_perl a Apache. Napísal ho R. M. Lerner a má názov „Writing Modules for mod\_perl“ (At the Forge, 18). O apačovskom module mod\_perl sa určite dozviete viac na jeho domovských stránkach (1) alebo (2).

Kto z vás nepočul o slávnom COAST laboratóriu (3)? Práve o ňom píše S. Nystrom v „Security Research Laboratory and Education Center“ (The Cutting Edge, 28). Z laboratória momentálne vzniká centrum s názvom Center for Education and Research in Information Assurance and Security — CERIAS (4). Toto všetko vedie profesor Gene Spafford (áno TEN Gene Spafford) a prebieha tam niekoľko zaujímavých projektov. Napr. Intrusion Detection autonomous agents — AAFID (5), už v podstate zastavený a skomercializovaný Tripwire (6), MUSCLE — smart karty a biometria v linuxovom prostredí (7), databáza bezpečnostných problémov a iné.

Linuxoví začiatkovníci sa v článku (8) od V. Veselovkeho (Linux Apprentice, 36) dočítajú zaujímavé informácie o koexistencii Linuxu a Windowsov na jednom disku a o použití programu FIPS. Na ďalších stranách robí Ch. Curley recenziu zaujímavého produktu na zálohovanie, nie len Linuxu, s názvom Arkeia od firmy Knox Software (Review, 38). Vyzerá to robustne a fungujúco, takže ja si jeho free verziu asi vyskúšam. Ak máte záujem, tak to rastie na domovských stránkach firmy Knox (9). Pokiaľ vás zaujímajú nevšedne vyzerajúce ale za to perfektne fungujúce počítače, prečítajte si recenziu Corelovského (10) NetWinderu (11) (Featu-

re, 42) od M. Gagného. Tento sexy vyzerajúci počítač (je na obálke tohto čísla) o veľkosti prerasteného walkmana má v sebe 275MHz StrongARM, 64MB pamäte a až 6GB disk.

Ak vám už váš 100Mbitový Ethernet nestačí a nemáte na ATM, tak by ste mali zvážiť FlowNet (12). Popisujú ho E. Gat a M. Ciholas v článku nazvanom „FlowNet: An Inexpensive High-Performance Network“ (Feature, 46). Ide o spínanú sieť na princípe ATM, ktorá vie spracovávať Ethernetové rámce a beží rýchlosťou až 500Mbps. Uvidíme do akej miery a či sa im podarí uviesť túto technológiu do komerčného použitia. Sieťové počítače — NC sú (alebo boli?) hit. B. Vincent hovorí v „Using Linux with Network Computers“ (Feature, 50) o svojej skúsenosti so sieťovým počítačom (samozrejme bežiacim pod Linuxom) od firmy Neoware Network Computers.

Vedeli ste o tom, že v AWKu môžete písať skripty, ktoré komunikujú po sieti? Presvedčí vás o tom J. Kahrs v „Network Administration with AWK“. Potrebujete k tomu GNU AWK 3.1. Ako príklad je tu finger na 7 riadkov alebo skript na čítanie WWW stránok na 8 riadkov. Samozrejme nemá to sieťovú funkčnosť Perlu alebo Céčka, ale na jednoduché a jednúčelové skripty sa to celkom hodí.

Tých čo sa snažia pod Linuxom robiť 3D animácie určite zaujme popis programu Blender od B.Crowdera (Forum, 70). Je to profesionálne vyzerajúce 3D modelovacie a renderovacie prostredie. Stiahnuť si to môžete z oficiálnych stránok Blenderu (13), prípadne si môžete aspoň pozrieť animácie vytvorené týmto programom (14), (15).

V článku „Cood Ol' sed“ (Take Command, 84) H. de Vreught poukazuje na niektoré základné vlastnosti programu sed a na možnosti ich využitia.

Ďalej si v tomto čísle môžete prečítať interview s Johnom Ousterhoutom, autorom Tcl/Tk (Interview, 74), recenziu prostredia CDE od Xi Graphics (Review, 58), recenziu knihy Linux for Dummies — Quick Reference, 2nd Edition (Review, 60), recenziu OpenGL 3D nadstavby do Matlabu s názvom Conix 3-D Explorer (Review, 62) a recenziu knihy Perl Cookbook vydavateľstva O'Reilly (Review, 78).

(Ak toto číta niekto z ComputerPressu, tak túto knihu by som rád videl aj na našom trhu v českom alebo slovenskom preklade). Niektoré články z tohto čísla, ako aj druhú časť Alphabet Soup a nejaké ďalšie články môžete nájsť online na stránkach linuxjournal.com (16). ■

- 1 Perl.Apache.org  
<http://perl.apache.org>
- 2 ModPerl  
<http://www.modperl.com>
- 3 COAST  
<http://www.cs.purdue.edu/coast/>
- 4 CERIAS  
<http://www.cerias.purdue.edu/>
- 5 AAFID  
<http://www.cs.purdue.edu/coast/projects/autonomous-agents.html>
- 6 Tripwire  
<http://www.tripwiresecurity.com/>
- 7 MUSCLE  
<http://www.linuxnet.com/>
- 8 Linux Dual Boot  
Windows/Linux Dual Boot
- 9 Knox  
<http://www.knox-software.com/>
- 10 Corel  
<http://www.corelcomputer.com/>





11	Netwinder	<a href="http://www.netwinder.org/">http://www.netwinder.org/</a>
12	FlowNet	<a href="http://www.flownet.com/">http://www.flownet.com/</a>
13	Blender	<a href="http://www.blender.nl/">http://www.blender.nl/</a>
14	Animace	<a href="http://www.unm.edu/~kholland/blender.html">http://www.unm.edu/~kholland/blender.html</a>
15	Animace	<a href="http://members.xoom.com/blender/">http://members.xoom.com/blender/</a>
16	Linux Journal 60	<a href="http://www.linuxjournal.com/issue60/">http://www.linuxjournal.com/issue60/</a>

## Báječný svět jádra v 2.2 (pokračování)

přeložil Nathan L. Cutler, 8. března 1999

### Zálohovat? Já?

Subsystém obsluhy páskových a jiných zálohovacích zařízení se oproti verzi 2.0 změnil jen málo. Za zmínku stojí především přidání nových ovladačů (tedy podpory dříve nepodporovaných zařízení) a podstatné zlepšení fungování u těch zálohovacích zařízení, která se připojují k radiční disketových mechanik (včetně IOMEGA DITTO).

Jako řešení zálohování dat nabývají přepisovatelné CD-ROM mechaniky v poslední době na významu a Linux 2.2 je podporuje (buď přímo v jádře nebo prostřednictvím emulátoru SCSI). Jsou zde však určité „nedodělky“ (podrobnosti viz dříve uvedené pojednání o CD-ROM mechanikách).

### Pákové ovladače, myši a vstupní zařízení

Joysticky nacházejí v 2.2 lepší podporu, včetně řady nových ovladačů a podpory joysticků s přehnanými počty tlačítek. Máte-li joystick, pod Linuxem 2.2 ho s největší pravděpodobností rozchodíte.

Podpora myši v 2.2 nevykazuje oproti verzi 2.0 podstatné rozdíly. (Jako u verze 2.0 jsou v podpoře myši určité nesrovnalosti, které budou muset být řešeny v budoucnu. Z větší části jsou myši ovládány mimo jádro prostřednictvím demona. Ze zcela zřejmých důvodů však některé ovladači myši záměrně emulují microsoftí standardní myš. Bylo by hezké, kdyby se tato otázka definitivně vyřešila, buď tak nebo onak.) Všimněte si, že — ačkoliv to není výhradně otázka kernelu — novější verze XFree86, nejpopulárnější GUI pod Linuxem, podporují myši se „spinning wheel extension“ od Microsoftu. (Většina linuxových aplikací však není schopna tuto novou feature využít — zatím.) Dále Linux 2.2 podporuje několik dalších vstupních zařízení, včetně některých digitizer pads. Pokud vaše zařízení emuluje myš, což je standardní chování, pak je již podporuje Linux 2.2 a dokonce i 2.0.

### Video

Snad nepřekvapivější a technicky nejpokročilejší nový prvek jádra verze 2.2 je ovladač konzole — tzv. *framebuffer* (nazýván zkratkou „fbcon“).

Dříve linuxové jádro (u strojů na bázi procesorů Intel) komunikovalo s video zařízeními pouze v textovém módu. Grafická podpora byla zajištěna dvěma dalšími systémy:

knihovnou svgalib pro grafiku na konzoli a specializovaným X serverem pro „okenní“ grafiku. Tento provizorní systém často vyžadoval opakování konfiguračních údajů a každá jeho součást podporovala pouze omezenou podmnožinu tisíců běžně dostupných a používaných video zařízení.

Vzhledem k novosti této feature není v tuto chvíli jasné, zda opravdu posune dosavadní „status quo“ založený na dualitě. Bohužel může trvat až rok po release verze 2.2 než nový systém bude podporovat ty karty a technologie, jejichž funkčnost pod Linuxem dnes považujeme za samozřejmou. Dle mého je to krok správným směrem, ale zdržím se konečného posudku, dokud nebude jasné, do jaké míry hodlají Linux a ostatní vývojáři tuto feature prosadit.

Jako vedlejší účinek této nové feature, kernel nyní obsahuje primitivní podporu multiheadingu (více grafických adaptérů v jedné mašině) u některých typů zařízení. V současné době je tato podpora omezoována na určité výstupy v textovém módu, ale je rozumné očekávání, že tato velmi nová funkce linuxového kernelu „dozraje“ během vývojových cyklů 2.2.x a 2.3.x.

Sem patří i zmínka o tom, že s verzí 2.2 je možné odstranit podporu „virtuálních“ terminálů z kernelu. To umožňuje ušetřit ještě více paměti tam, kde je jí nedostatek.

Ač si to majitelé běžných desktopů stěžují umějí představit, Linux nyní běží ještě lépe na systémech bez jakéhokoliv videa. Vedle možnosti přihlášení po sériových a síťových linkách, což obsahovaly i Linux 2.0 a předchozí verze jádra, nyní je možné přeměrovat veškerá kernelová hlášení (dříve poslaná přímo na konzoli před inicializací jakéhokoliv hardwaru) na sériové zařízení.

### Sítě: Ethernet, ISDN a modemy

V této oblasti valné zkušenosti nemám; již několik let používám stejné síťové karty ve všech svých mašinách. Nemusíte však být Alanem Coxem, abyste poznali, že se s Linuxem 2.2 prudce zvýšil počet podporovaných zařízení Ethernet a ISDN. Podporu rovněž našla, jak mi bylo řečeno, i novější řešení jako např. kabelové modemy.

Jediné, na co si mohu stěžovat v této oblasti, je přetrvávající absence podpory tzv. Winmodemů. Ne že bych z toho vinil Linux — výroba modemů, které jsou z 80 % software, je špatně v každém případě — ale idealista ve mně doufá, že jednoho dne budou tito malí d'áblici podporováni stejně jako jejich používatelnější bratrance.

### Radiovní amatéři jsou také lidé

Již v dobách před verzí 2.0 byl Linux jedním z mála desktopových OS, které v sobě zahrnovaly přímou podporu použití amatérského radia s počítači. (Ve skutečnosti moc nerozumím tomu, co to obnáší, ale zdá se být populárnější mimo USA.) Linux 2.2 přináší velké množství zlepšení tohoto subsystému, včetně schopnosti vytvořit síť několika strojů jen pomocí radiových vln, ale není to moje parketa.

Uvítám veškeré náměty a příspěvky do této kategorie.

### Světové filesystemy

Linux 2.2 má široké spektrum nových filesystemů a typů partition k zajištění větší interkonektivity. Krom toho došlo k aktualizaci mnohých Linuxem podporovaných filesystemů (včetně těch, které zde vůbec neuvádím) a byl vypracován nový systém dcache pro značně zvýšenou výkonnost.



(Příslušná aktualizace byla dokonce podmínkou zahrnutí všech dotčených ovladačů do verze 2.2.)

Fanoušci Microsoftu jistě uvítají skutečnost, že Linux už umí číst (a možná i psát) partišny ve formátu NTFS (Windows NT) a FAT32 (Windows 98 a pozdější mutace Windows 95). Linux 2.2 zároveň umí i microsoftův systém Joliet umožňující dlouhé názvy souborů na CD-ROMech. V neposlední řadě Linux rovněž umí pracovat s novým typem extended partition — „vynálezem“ Microsoftu. Ve vývoji jsou ovladači pro čtení a psaní komprimovaných disků typu Microsoft a Stacker. Tyto ovladače však zatím nejsou součástí jádra.

Za účelem zlepšení konektivity s Macy byl zapracován ovladač HFS pro čtení a psaní disků v Macintoshích. HFS+ a starší filesystemy na Macích nejsou zatím podporovány. Kernel nově umí číst partition tabulky Macintoshe, což umožňuje mountovat disky Mac SCSI bez pomoci speciálního SW.

Je to sice smutné, ale uživatelé OS/2 stále nebudou moci psát na své partišny ve formátu HPFS. Ovladač HPFS byl obohacen o podporu nového systému dcache, ale celková „generální oprava“, ve kterou někteří doufali, se bohužel nedostavila.

Pokud ještě jsou nějací uživatelé Amigy, budou potěšeni zprávou o tom, že ovladač FFS prodělal několik menších změn oproti verzi 2.0. Toto může být obzvláště užitečné v případě, že nová generace Amig na bázi PPC bude používat stejný diskový formát.

V oblasti konektivity s jinými UNIXy udělal Linux 2.2 velké pokroky. Stále je součástí Linuxu 2.2 filesystem UFS, který se používá v systémech odvozených od BSD, včetně Solarisu a free verzí BSD. Linux 2.2 nyní umí číst formáty partition tabulek provozované FreeBSD, SunOS a Solarisem. Pro UNIXy podobné SysV Linux 2.2 nabízí poněkud vylepšenou verzi SysV FS. Linux 2.2 zároveň umí číst disky RiscOS od Acorna. V neposlední řadě Linux 2.2 v sobě obsahuje aktualizovanou verzi stále populárního filesystemu Minix, který nalézá uplatnění u menších disků a disket ve většině UNIXů. Při tolika různých vzájemně nekompatibilních formátech se člověk nestačí divit, že se někomu podařilo dotáhnout vůbec nějakou práci.

Další novinkou je odstranění podpory extended disků (tento formát se provozoval s o mnoho staršími verzemi Linuxu) a její nahrazení filesystemem second extended, neboli ext2. (Tento krok by neměl nikomu vadit, neboť ext2 je daleko lepší než jeho předchůdce.) V zájmu zlepšení podpory initial ramdisků byl vyvinut filesystem romfs s naprosto minimálními paměťovými nároky.

Byť to není přímo filesystemem, součástí Linuxu 2.2 je možnost transparentně spojit několik disků do jednoho filesystemu. V současné době je tato podpora k dispozici v módech RAID 0, 1, 4 a 5, jakožto i v jednoduchém lineárním módu.

## Podpora sítí II: Pod kapotou

Co se týče protokolů došlo k mnoha věcem, kterým prostě dostatečně nerozumím. Objevily se IPv6 (Internet protokol další generace) a SPX (rozšíření IPX). Došlo rovněž ke zlepšení DDP, což je nejrozšířenější protokol u sítí AppleTalk. Navíc, jak se dalo čekat, byla zlepšena podpora stávajících protokolů. Kéž bych měl potřebu něco z toho používat...

U méně nákladných řešení se toho moc nezměnilo. PPP, SLIP, CSLIP a PLIP jsou i nadále k dispozici. Asi jsou věci,

kteří žádné zlepšení nepotřebují. (Každý z těchto ovladačů však byl tak či onak aktualizován.)

Zde však seznam nekončí. Linux 2.2 obsahuje výborné nové řešení podpory sítí (networking core), nové řešení tunnelingu, úplně nový systém — s názvem ipchains — pro firewall a routování, možnost omezovat tok dat a celou řadu dalších věcí. Je to prostě úžasné, kéž bych byl schopen to všechno sledovat.

Protokoly pro sdílení souborů a tiskáren byly rovněž zlepšeny a značně rozšířeny. V podpoře SMB (protokol umožňující přístup na sdílené filesystemy pod Windows) byly provedeny opravy chyb a jiné aktualizace. V případě, že jste fanoušci NetWaru (o čemž lze s úspěchem pochybovat), bude pro vás dobrá zpráva, že Linux 2.2 obsahuje v této oblasti celou řadu zlepšení, včetně dvou druhů dlouhých názvů NCP. NFS byl také zlepšen na úrovni serveru i klienta. Navíc ti šprtí u CMU provádějí mravenčí práce na vývoji nového distribuovaného síťového filesystemu pod názvem Coda. Tento filesystem podporuje velký počet vysoce žádaných features včetně disconnectovaných operací pro laptopy, velmi propracovaného systému cache a zlepšené bezpečnosti.

Souvisejícím vylepšením v Linuxu 2.2 je ovladač, který umožní sdílení a vzdálené mountování celých diskových image po síti.

## Cože?! Evropa je obydlená?

Linux 2.0 je překvapivě velmi internacionální OS s podporou mezinárodních klávesnic a podobně. Linux 2.2 k tomuto základu dodává schopnost použití některých codepage UNICODE pro konverzi názvů souborů do native formátu Linuxu. (V současné době jedinými filesystemy, které tyto codepage UNICODE používají jsou VFAT a Joliet (rozšíření ISO 9660 pro CD-ROMy), oba od Microsoftu.)

## A v neposlední řadě

Kromě výše uvedených Linux 2.2 přináší řadu dalších ovladačů a novot, které sice nezapadají do žádné kategorie, ale stojí za povšimnutí:

Ovladač loopback, umožňující mountování a manipulaci s diskovými image jako kdyby byly fyzické disky, byl zlepšen hned několika způsoby. Z těchto vylepšení uživatelé bezpochyby ocení zvýšenou podporu šifrování a mountování zafixovaných pevných disků a diskových image.

Navíc je zde podpora initial ramdisků, které přináší větší pružnost při bootování Linuxu. Initial ramdisk např. umožňuje v distribuci mít kernel bez jakýchkoliv zabudovaných ovladačů a později natáhnout ovladače podle konkrétní potřeby. (Dříve musel ovladač bootovacího média nutně být v kernelu, aby bylo možno ovladače dalších zařízení načíst z pevného disku.) Tuto feature ocení zvláště ti, kdo mají zařízení Plug-and-Play, která jsou nepřístupná, dokud se nespustí příslušná user-mode utilita.

Linux 2.2 také disponuje ovladačem, umožňujícím přístup do paměti CMOS na desce, což může být užitečné v některých aplikacích. (Bohužel, podobný ovladač pro BIOS flash RAM se do verze 2.2 nedostal, takže k aktualizaci BIOSu bude i nadále nutně restartovat z diskety s MS-DOSem.)

Linux v minulosti používal metodu natahování a odstranění ovladačů, nazvanou kerneld, který byl napůl uživatelský a napůl kernelový. Tato metoda byla dobrá, ale málo



výkonná. Proto kernel`d` z verze 2.2 vypadl a byl nahrazen novým 100% kernelovým řešením s názvem `kmod`. ■

## Nové grafické karty a Xka

Robert Vojta, 13. března 1999

V dnešní uspěchané době se grafické karty vyvíjejí neuvěřitelným tempem a programátoři kteří mají něco společného s XFree86 nestíhají vyvíjet nové a nové ovladače. Je to způsobeno vytížeností programátorů XFree86 a také tím, že většina firem zabývajících se výrobou karet uvolňuje specifikaci dosti pozdě. Proto je tu pro nás chudáky, co máme nové železo, návod na to, jak zprovoznit Xka pod operačním systémem Linux.

### Co je potřeba?

Co k tomu potřebujete? První věc je nové jádro (1), a to minimálně verze 2.1.112, novější XFree86 (2) (doporučuji 3.3.3.1) a poslední věc je vaše grafická karta, která musí být kompatibilní se standardem VESA, a to minimálně verze 2.0. Takže teď víme, co k tomu potřebujeme, a můžeme se pustit do konfigurace vašeho stroje.

### Konfigurace jádra

Jádro rozbálíme do adresáře `/usr/src` a vytvoříme symbolický link `/usr/src/linux`, který bude ukazovat na naše nově rozbalené jádro. Přejdeme do adresáře `/usr/src/linux` a spustíme konfiguraci jádra a to pomocí příkazu `make menuconfig`. Jádro si nastavte dle vlastních představ a až to budete mít hotové, musíte ještě provést následující věci:

- v menu Code Maturity Level zaškrtněte položku Enable the prompt for development and/or incomplete drivers.
- v menu Console Drivers zaškrtněte následující položky
  - VGA Text console
  - Video selection support
  - Support for frame buffer devices
  - VESA VGA Graphics console
  - Advanced low level drivers
  - Mono, 2bpp, 4bpp, 8bpp, 16bpp, 24bpp a 32bpp packed pixel drivers
  - VGA Character/attributes support

Nyní můžeme opustit konfigurační program a pustit se do dalšího kroku.

### Kompilace a instalace jádra

Nyní máme nastavené nové jádro a můžeme se pustit do kompilace. Nové jádro vytvoříme sledem následujících příkazů: `make dep`, `make clean` a `make bzImage`. Pokud všechny kroky prošly, vytvoříme ještě moduly, a to následujícím způsobem: `make modules`, `make modules_install`. Nyní máme zkompilevané jádro a nainstalované moduly. Naše nové jádro se nachází

v adresáři `/usr/src/linux/arch/i386/boot` a jmenuje se `bzImage`. Nové jádro přepokopírujte do adresáře `/boot` a přejmenujte na nějaký inteligentní název, třeba `bzImage-2.2.2-ac5` (v následujících částech textu budu předpokládat, že ho máte pojmenová jako tento příklad). Dále do souboru `/etc/lilo.conf` vložíme následující řádky:

```
image=/boot/bzImage-2.2.2-ac5
label=linux-2.2.2-ac5
root=/dev/hda2
read-only
vga=794
```

Samozřejmě že řádek `root=/dev/hda2` bude závislý na tom, kde máte nainstalován root filesystem vašeho Linuxu a položku VGA si nastavíte na jednu z hodnot, které naleznete v následující tabulce:

Barvy	640x480	800x600	1024x768	1280x1024	1600x1200
256	769	771	773	775	796
32768	784	787	790	793	797
65536	785	788	791	794	798
16.8M	786	789	792	795	799

Toto rozlišení a barevná hloubka se bude používat pro textovou konzoli a Xka zároveň. Pokud uvedete do položky VGA hodnotu ASK nebude vám umožněno spouštět Xka. Nyní máme upraven konfigurační soubor `lilo` a pokud chceme, aby systém akceptoval změny, musíme spustit `/sbin/lilo`. Právě teď je váš systém připraven na restart, zkuste to a můžete se kochat malým logem tučňáka při startu systému.

### Jak upgradovat na XFree86 3.3.3.1

Z archivu XFree (3) si stáhněte binárky posledního XFree a ze souboru `RELNOTES` můžete vyčíst, jak provést upgrade nebo novou instalaci XFree. Až provedete upgrade, ujistěte se, že máte vytvořen link z `/usr/X11R6/bin/X` na `/usr/X11R6/bin/XF86_FBDev` a pokud ne, proveďte následující příkaz:

```
ln -s /usr/X11R6/bin/XF86_FBDev /usr/X11R6/bin/X
```

### Jak nastavit zařízení /dev/fb0?

Kernel s podporou framebufferu je instalován, ale ještě zbývá zprovoznit zařízení `/dev/fb0`, které umožňuje X serveru zapisovat přímo do paměti grafického adaptéru. Vytvoříme znakové zařízení s hlavním (major) číslem 29 a vedlejším (minor) číslem 0. Dá se to pomocí příkazu `mknod /dev/fb0 c 29 0`. Pro zpětnou kompatibilitu je dobré vytvořit symbolický link `/dev/fb0current`, který ukazuje na `/dev/fb0` (`ln -s /dev/fb0 /dev/fb0current`). Pokud máte v systému více různých grafických adaptérů (a monitorů), musíte instalovat další znakové zařízení. Vytvoříme ho stejným způsobem jako pro první kartu, ale s tím rozdílem, že vedlejší (minor) číslo bude 32, u třetí karty 64 atd.

### Jak nastavit XF86Config?

Poslední krok je nastavit konfigurační soubor `Xek`, aby akceptoval nový framebuffer server. Tento soubor se povětš



nou nalézá v adresáři /etc/X11 a jmenuje se XF86Config. Pro nás jsou kritické tři sekce toho souboru, a to: Monitor, Device a Screen. Provedte nastavení podle následujících řádků:

```
Section "Monitor"
  Identifier "My Monitor"
EndSection

Section "Device"
  Identifier "My Video Card"
EndSection

Section "Screen"
  Driver      "FBDev"
  Device      "My Video Card"
  Monitor     "My Monitor"
  Subsection "Display"
    Depth     16
    Modes     "default"
  EndSubsection
EndSection
```

Jedno upozornění: v části Display (nalézá se v části Screen) se musí barevná hloubka shodovat s barevnou hloubkou, kterou máte nastavenou v konfiguračním souboru lilo (vga=). Pokud to nebude souhlasit, X server se nespustí. Nesnažte se také měnit část Modes na nějaké rozlišení, protože rozlišení v kterém se budou pouštět Xka je nastaveno na stejném místě jako barevná hloubka (tedy v /etc/lilo.conf).

## Závěr

Doufám, že většině lidí pomohl tento článek vyřešit problém s jejich novou grafickou kartou a právě spokojeně používají Linux a jeho Xka. Pokud mi čas dovolí, budu aktualizovat nově vzniklou stránku o tomto problému na mém stroji (4). ■

```
1 České zrcadlo zdrojových textů jádra
  ftp://ftp.cz.kernel.org/pub/linux/kernel/
2 XFree
  http://www.xfree86.org/
3 XFree
  ftp://ftp.xfree86.org/pub/XFree86/3.3.3.1/binaries/
4 Devil
  http://devil.mendelu.cz/vesafb
```

## Co je to HTTP a HTML?

Marek Sezemský, 20. března 1999

Toto měl být původně článek o programování CGI z jazyka C. Vzhledem k tomu, že jsem „kratší seznámení s HTTP a HTML“ poněkud přetáhl, rozhodl jsem se k radikálnímu zásahu a jeden článek rozdělil na dva kusy. Tohle můžete brát jako zjednodušený popis toho, jak je možné, že Netscape (pro zatvrzelé Lynx) zobrazuje webové stránky plné grafiky (už ne Lynx :- ) a odkazů.

Samo sebou, že po dočtení se můžete pustit i do druhé části, a to do popisu knihovny cgihtml pro zpracování formulářů na Internetu (prostě programování CGI).

Jen tak mimochodem: nekladu si za cíl popisovat vše

úplně technicky a přesně, možná někde „odlínou“, takže se prosím, vážení odborníci, moc nezlobte.

## Co je HTTP?

HTTP (HyperText Transfer Protocol) je protokol (podobně jako FTP a další), který se používá na Internetu snad nejčastěji. Pomocí něj putují data od webového serveru k vám domů. Jeho činnost je následující: po navázání spojení k serveru (např. www.linux.cz) klient (Netscape, Lynx) odešle tzv. požadavek (request), který vypadá takto:

```
GET /index.html HTTP/1.1
Connection: close
User-agent: Lynx/2.8
<prázdný řádek>
```

Jestli si chcete vyzkoušet chování web serveru, zkuste si tohle napsat do telnetu, který spustíte jako telnet www.někde.cz 80, což vám otevře spojení na server přes port 80 (neboli http). Ale zpátky k tématu: První řádek je povinný. Udává metodu přenosu (jedno z GET/POST/HEAD: tyto metody jsou z hlediska CGI podrobně rozebrány v [Popis knihovny cgihtml, aneb jak na CGI z C](#)), cestu k souboru (/index.html, /, /archiv, atd.) a verzi protokolu HTTP (zde 1.1). Další řádky jsou nepovinné, ale uvedl jsem asi ty nejfrekventovanější. Pokud má Connection hodnotu Keep-alive, server po vyřízení požadavku (tj. odeslání souboru neuzavírá spojení, ale čeká na další požadavek. Při hodnotě close se spojení uzavře. Takže keep-alive vám šetří čas a tak trochu i peníze. Pomocí User-agent klient identifikuje sám sebe. Další možné parametry jsou např. Last-Modified — datum modifikace souboru (pro účely cache), přijímané typy dat (Accept-Content), a další. Zde platí, že co server nezná, ignoruje. Poslední prázdný řádek je povinný, jinak by server nepoznal, kde klient zakončí své přání. Po ukončení požadavku (oním prázdným řádkem) server odešle odpověď (reply). Ta má tento tvar:

```
HTTP/1.1 200 OK
Content-length: 120
Content-type: text/html
<prázdný řádek>
<...vlastní data (soubor)...>
```

První řádek vrací verzi HTTP na serveru (zde opět 1.1), kód chyby resp. úspěchu (200) a textový popis ke kódu (zde „OK“). Číselných kódů je mnoho, ale dají se rozlišit pomocí stovek: kódy 1xx jsou informativní, 2xx značí úspěch, 3xx je přesměrování, 4xx chyba ze strany klienta a 5xx chyba serveru. Content-length je délka odeslaných dat. Počítá se bez délky hlavičky (takže po prázdném řádku — prostě pouze data). Content-type je typ dat. text/html je HTML soubor, text/plain je čistý ASCII text. image/gif a image/jpeg jsou obrázky. Těchto typů je opravdová spousta a v různých situacích lze použít i masku: takže image/\* znamená „všechny obrázky“. Důležité je i to, že klient nerozlišuje příponu u souborů, takže pokud si řeknete např. „http://www.sezi.cz/cgi-bin/counter“, můžete si natáhnout obrázek (počítadlo), ale i jinou stránku apod. Nyní se podíváme, co se skrývá za typem text/html.

## Co je HTML?

HTML je HyperText Markup Language, neboli hypertexto-



vý jazyk značek (tagů). Je to formát souborů používaných převážně na Internetu. Do takového souboru můžete vložit odkaz na jiný soubor, měnit formátování, vkládat obrázky a spoustu dalších věcí. Tyto věci provádíte pomocí tzv. značek (tagů). Většina těchto tagů je párová (budu používat anglický tag místo české značky, připadá mi to lepší). Na jazyce HTML je krásné to, že to co neznáte (nezná prohlížeč), prostě ignoruje. Takže text v dokumentu je vždy zobrazen, ale pokud máte starší verzi prohlížeče, přijdete např. o speciální formátování, efekty nebo různé blbinky se skripty (JavaScript). Na škodu je to tehdy, když máte někde překlep (místo `<form>` třeba `<dorm>`) a HTML stránka se vám chová poněkud divně. Nejkratší HTML dokument vypadá takhle:

```
<html>
<head> <title> Titulek stránky </title> </head>
<body bgcolor="Barva pozadí" text="Barva textu" \
  link="Barva odkazů">
...text dokumentu...
</body>
</html>
```

Jako barva se uvádí anglický název (white, black), nebo hexadecimální vyjádření #RRGGBB. Např. modrá je tedy #0000FF. Mezi další tagy patří nadpisy:

```
<h1> text nadpisu </h1>
<h2> text nadpisu </h2>
...
<h5> text nadpisu </h5>
<h6> text nadpisu </h6>
```

Styl `<h1>` je největší (má tučné velké fonty), a `<h6>` nejmenší (opravdu malinký). Odsazení nového řádku se provádí tagem `<br>`, nový odstavec vytvoříme pomocí `<p>`. Odkaz na jiný soubor (dokument) vložíme pomocí:

```
<a href="http://adresa.cz/soubory/soubor.html" \
  >Podtržený text odkazu</a>
```

Adresu u href tvoří tzv. URL (tuším Universal Resource Locator), což je jakoby adresa dokumentu. Je tvořena protokolem (http://, ftp://), serverem adresa.cz a souborem /soubory/soubor.html. Jednoduché jako facka. Podobně jako odkaz vložíme i obrázek. Všimněte si parametru alt: jeho hodnota je v úvozovkách, aby bylo jasné, že vícelslovný text není další parametr. To platí v celém HTML. Kde jsou mezery, tam taky úvozovky.

```
<img src=http://adresa.cz/obrazek.jpg \
  width=sirka height=vyška alt="Textový
popisek obrázku">
```

Šířka a výška jsou v pixelech. Jako obrázky na webu se používají (takřka jedinně) GIF a JPEG. Chystá se i nový PNG, ale bohužel se přes své kvality zatím moc nerozšířil. Onen alt obsahuje text, který je „pod obrázkem“ — tedy pokud se jej např. nepodaří načíst. S grafikou buďte velice střídmi: foto autora ve formátu A4 je sice věc hezká, ale nikdo nemusí čekat půl hodiny na 100 kilo, aby viděl ten váš kšiš\*\* ; -). Pokud už něco používáte, nezapomeňte na to, že web je tady hlavně pro informace, pak pro Linux, a naposled pro modelky :-).

Seznamy vytvoříte snadno (když místo `<ul>` použijete `<ol>`, dostanete číselný seznam):

```
<ul>
  <li>Položka 1
  <li>Položka 2
  <li>Položka 3
</ul>
```

## Formuláře

Nejdůležitější v HTML, pokud se díváme z pohledu CGI, jsou formuláře. Formulář začíná párovou značkou `<form>`. Její parametry jsou action, který obsahuje URL na CGI program, který bude zpracovávat data z prvků formuláře. Dále je to položka method, která udává způsob dopravení dat k CGI. Zde se používá get nebo post. Výhody a nevýhody obou si popíšeme ve článku [Popis knihovny cgihtml, aneb jak na CGI z C](#). Pro vstupní prvky formuláře (např. textový řádek, prepínač, či jednoduché menu) se používá značka `<input>`. Jejím parametrem jsou type, name a popř. value. Parametr type udává typ vstupu (text / heslo / tlačítko / checkbox / radiobox). name je jméno pro hodnotu, na které se odkazujeme z CGI. Do value můžete zadat počáteční hodnotu vstupu. Vše objasní příklad (z kterého určitě mnohé další pochytíte). Jako první si rozebereme jednoduchý dotazník: od uživatele (surfaře Internetem) si zjistíme nějaké informace (jméno, věk, pohlaví...) a případně zapíšeme na serveru do souboru. Zdrojový formulář by vypadal asi tak, jako na výpisu [Příklad formuláře](#). (nezapomeňte, že se to celé musí vložit někde do html souboru, tohle by nemělo být jako samostatný soubor (už pro tu estetiku :-)).

Tímto bychom měli pojmy jako HTTP a HTML rozebrané. Pokud se chcete tomuto tématu věnovat i nadále, doporučuji studium RFC dokumentů. Pro HTTP/1.0 a HTML 2.0 (poslední verze jsou 1.1(1.2?) a 4.0) jsem nějaká (značně obstarožní) RFC našel u sebe v archiváliích, takže se můžete kochat: (1) a (2).

Mimochodem: Na mé homepage jsem umístil příklady k tomuto článku spolu s testovacími CGI programy, takže pokud se chcete kouknout, jak vypadá takový (3), dle libosti. ■

```
1 RFC-1945-HTTP
  http://www.sps-jia.cz/~sezi/rfc-1945-http.txt
2 RFC-1866-HTML
  http://www.sps-jia.cz/~sezi/rfc-1866-html.txt
3 formulář
  http://www.sps-jia.cz/~sezi/form1.html
```

## Popis knihovny cgihtml, aneb jak na CGI z C

Marek Sezemský, 23. březen 1999

### Pár slov úvodem

Jestli chcete zvládat (a ovládat) CGI na Internetu, a to z jazyka C, tak to čtete ten správný článek (ve správných novinách ; -). Ale proč C a ne Perl, který se pro CGI používá dosti hojně? Proti Perlu (ani jiným jazykům) rozhodně nic nemám, zcela jistě jsou velice výkonné a použitelné — zvláště Perl, ale v C už nějaký ten měsíc dělám, a tak je pro mne snazší se naučit používat nějakou tu knihovnu, než se učit Perl (i přes jeho jednoduchost ; -). Navíc používám C



```

<form action=/cgi-bin/test.cgi method=post>

<!-- URL v action může být i relativní odkaz (zde od hlavního adresáře na
serveru. (Jo a tohle je html komentář (na to byste sami nepřišli, že jo? ;-))
-->

<dl>
  <dt> Jméno
  <dd> <input type=text name=jmeno value="Pepa Novák">

  <dt> Věk
  <dd> <input type=text name=vek value=21>

  <dt> Pohlaví
  <dd>
    <input type=checkbox name=pohlavi value=muz> Muž <br>
    <input type=checkbox name=pohlavi value=zena checked> Žena <br>

  <dt> Zájmy
  <dd>
    <input type=checkbox name=zajmy value=sport checked> Sport <br>
    <input type=checkbox name=zajmy value=tv> Televize <br>
    <input type=checkbox name=zajmy value=sex checked> Sex <br>
    <input type=checkbox name=zajmy value=pc> Počítače <br>

  <dt> Vlastním Internet
  <dd>
    <select size=1 name=inet>
    <option value=domov>Doma
    <option value=skola>Ve škole
    <option value=prace selected>V práci
    </select>

  <dt> Poznámky
  <dd>
    <!-- U textarea je přednastavený text mezi párovou značkou -->
    <textarea name=pozn cols=20 rows=10>Komentář, připomínky, atd.</textarea>

  <dt>
  <dd>
    <input type=submit value=" odeslat ">
    <input type=reset value=" vymazat ">
</dl>
</form>

```

Výpis č. 1: Příklad formuláře

se mi zdá takové „univerzálnější“ (ted' se asi Perláci smějí, ale je to můj skromný názor a nikdo mi ho nevěme). Takže bychom za sebou měli otázky jazyka a jdeme dál.

### CGI: O co jde?

O vzniku CGI bych moc rád psal ono „v dávných dobách Internetu“, ale bohužel nemohu, neboť bohužel ani netuším, jak to tehdy vypadalo. Ale jedno vím jistě: CGI znamená Common Gateway Interface. Pokud bych se pokusil o překlad, vzniklo by Společné Bránové Rozhraní. Bohužel, tohle nic moc neříká, takže to vysvětlím: CGI je rozhraní, pomocí kterého Web server spouští aplikace a jejich výstup posílá přes Internet klientovi. Tímto výstupem může být GIF(JPEG) obrázek v případě počítačové reklamy, HTML soubor pokud se jedná například o výsledek hledání.

Prostě výstupem může být cokoli. Napsal jsem rozhraní: to z toho důvodu, že se nejedná o žádný nový typ spustitelných souborů, nebo snad o něco, co se kompiluje k serveru (což u MS IIS ani nejde ;-)). CGI programem se rozumí jakákoliv aplikace, schopná zpracovat vstup z formuláře, a posleze něco kloudného poslat na výstup klientovi. Ony parametry se CGI programu předávají pomocí hodnot proměnných prostředí (environmentu), které nastaví server. Prostě pomocí toho, kde máte PATH, HOME, MAIL a další (pro nechápavé: v shellu zkuste set a uvidíte). Takže server nastaví několik hodnot tohoto prostředí, do kterého mimo jiné vloží i parametry z CGI, spustí program a jeho výstup poté odešle klientovi. Zpravidla je tento výstup odeslán až po ukončení CGI programu, lze jej ale odesílat i postupně. Ale to vše si osvětlíme později. Takže ty proměnné prostředí. Víme že je server nastaví. Ale které? To samé jsem si říkal při psaní předchozí věty (opravdu). Řešením je tako-



vý malý testovací program. Jmenuje se `test.cgi.c`, takže po přeložení u GNU C s parametrem `-o` máme `test.cgi`.

```
#include <stdio.h>

int main(int argc, char **argv, char **env)
{
    int i = 0;

    printf("Content-type: text/plain\n\n");

    while ( env[i] != NULL )
        printf("%s\n", env[i++]);

    return (0);
}
```

Výpis č. 2: Program `test.cgi.c`

Jelikož se jedná o C program, stačí ho přeložit a poté zkopírovat tak, aby ho webserver „viděl“. Takže pokud jste root, bude to asi (u RedHatů) `/home/httpd/`, pokud nemáte to štěstí být root, ale máte to štěstí a jste uživatel Linuxu, zkuste ve vašem domovském adresáři něco jako `public_html`, nebo `html`. Poté stačí spustit Lynx nebo Netscape (popřípadě něco z KDE) a napsat URL našeho souborku. Takže u mě (já == root ; -)) je uložen jako `/net/httpd/html/test.cgi` a z Lynxe `http://www.sezi.cz/test.cgi`. Není to klasické uložení v `/cgi-bin`, ale nechce se mi to psát tak dlouze.

U programu si všimnete prvního vypisovaného řádku. Pokud jste četli [Co je to HTTP a HTML?](#), tak jistě víte, že `Content-type` označuje co za data to leze ke klientovi. `text/plain` značí textový soubor, pokud bychom uvedli `text/html`, jde o HTML soubor. Odesílat můžete klidně i obrázek nebo video — to záleží na vás. Nenechte se zmýlit dvojicí nových řádků — po ukončení hlavičky dokumentu MUSÍ následovat prázdný řádek, jinak klient nepozná, kdy končí hlavička a začínají data.

Zatím pouze CGI spouštíme, ale to jistě nestačí. Bude potřeba mu nějak předat ony parametry, se kterými se bude pracovat. A to pomocí formuláře z HTML stránky. Ve formulářích jsou vstupní pole (prvky), která mají svá jména (u `<input>` parametr `name`) a hodnoty (`value`). Ty se do CGI předají jako rozšíření URL. Takže např. program `test.cgi` na `www.sezi.cz` s parametrem `jmeno`, který má hodnotou `Marek` se spustí jako `http://www.sezi.cz/test.cgi?jmeno=Marek`. Pokud bychom měli ještě příjmení s hodnotou `Sezemsky`, pak by URL vypadalo takhle:

```
http://www.sezi.cz/test.cgi?jmeno=Marek\
&prijmeni=Sezemsky
```

Pokud ale uživatel do formuláře zadá jiné znaky než abecedu a čísla (např. české háčky a čárky), parametry jsou klientem konvertovány do podoby `%XX`, kde `XX` je hexadecimální kód znaku (dvouznakově — `%01` a ne `%1`). Tento styl předávání parametrů (přímo v URL) se nazývá GET (u `<form>` parametr `method`). Má několik nedostatků: maximální délka vstupu je něco kolem 240 bajtů (neručím za správnost) a takovýto požadavek se objeví v logu serveru, což je nevhodné např. pro předávání hesel, čísel kreditních karet apod. Druhou metodou je POST. Ten odstraňuje oba nedostatky: velikost vstupu je (teoreticky) neomezená (přenesl jsem trochu přes 3MB a vše bylo v pořádku)

a v logu se zobrazuje pouze URL na CGI. Asi se ptáte, jak je to možné? Velice snadno. Jak víte, klient při požadavku o stránku (obrázek, ...) odešle v první řádce text GET, který následuje URL dokumentu rozšířen o parametry pro CGI program. U metody POST odesílá `POST ; -`, a poté opět URL. Zakončí to verzí HTTP. Jako další řádek přidá `Content-length` a délku požadavku, který odešle za prázdným řádkem. Pro názornost:

```
Klient: POST /test.cgi HTTP/1.1 Content-length: 10
name=marek
```

```
Server: HTTP/1.0 200 OK (zde začíná výstup z CGI):
Content-type: text/plain
```

```
data odeslaná CGI programem
```

```
a pro GET: Klient: GET /test.cgi?name=marek HTTP/1.1
```

```
Server: HTTP/1.0 200 OK (zde začíná výstup z CGI):
Content-type: text/plain
```

```
data odeslaná CGI programem
```

Takže metody přenosu máme za sebou a teď se vrátíme k proměnným prostředí. Jako hlavní nás bude zajímat `REQUEST_METHOD`. Obsahuje název způsobu předání parametrů: GET nebo POST. Další je `QUERY_STRING`. Zde jsou v případě GETu umístěny parametry pro CGI (jmeno=Marek&prijmeni=Sezemsky). Pokud použijete POST, je `QUERY_STRING` prázdný a parametry jsou na `stdin` (takže použijete `gets()`). Zajímavé jsou i `REMOTE_HOST` a `REMOTE_ADDR`, které obsahují jméno (`linux.sezi.cz`) a IP adresu (`10.0.0.1`) počítače, odkud byl CGI vyvolán (klient). To se dá využít v případě logování záznamů o přístupech. Toliko ty proměnné, které používám. Pokud si spustíte moje testovací CGI (samozřejmě přes web a ne z shellu), spatříte všechny hodnoty, které jsou definovány a které tedy můžete použít. Ale pozor na to, že starší servery, popřípadě servery od Microsoftů nemusí podporovat vše.

## Knihovna `cgihtml`

Takže už víme, jak se CGI spouští, jak mu server předává parametry a různé informace, ale pořád nevíme, jak vydolovat nějaké v programu použitelné hodnoty. Máme dvě možnosti: buď si vytvoříte (ano, je to na vás) novou knihovnu, která bude obsluhovat vše od načtení vstupních parametrů z `env[]`, přes kontrolu vstupu, až po upload souborů (ano, i to lze přes CGI vytvořit). Už asi tušíte, že tohle nebude cesta, po které se vydáme. Takže zkuste navštívit stránku E.E.Kima, na které si obstaráte knihovnu (1). Ve formátu `.tar.gz` má pouze pár desítek kB, a to včetně dokumentace a příkladů. Lze spustit jak pod Linuxem (takže Unixem), tak i pod Windows. Během tří měsíců se mi velice osvědčila a dobře se s ní píšou i ty nej-CGI aplikace, mezi které patří i onen upload. Samozřejmostí je schopnost načíst data z HTML formulářů, a to jak přes GET, tak i přes POST, dále obsahuje i pár funkcí obsluhujících HTML, a další. Rozhodně inteligentní vlastností je ta, že pokud je CGI program využívající tuto knihovnu spuštěn z terminálu (tj. klasický shell, X-Win, MC, ...) požádá si o parametry. Po jejich zadání program prostě jede dál a neví, jestli jede přes web, nebo shell. Dobře se s tím testuje a ladí, neboť webserver při chybě v CGI (`core dump` a pod.) zobrazí pouze nic neříkající `5xx Server error`.

Práce s knihovnou `cgihtml` je velice jednoduchá a naučíte se ji ovládat během hodinky. Knihovna obsahuje několik částí:

- `cgi-lib`: obstarává vše kolem CGI



- `html-lib`: vše kolem HTML (nemusíte používat)
- `cgi-llist`: vše kolem seznamu CGI parametrů (nemusíte používat)
- `string-lib`: řetězce (také není potřeba)

Ze začátku bude asi nejlepší nějaký ukázkový příklad, který si popíšeme:

```
#include <stdio.h>
#include "cgihtml/cgi-lib.h"
#include "cgihtml/html-lib.h"

int main()
{
    llist cgi_data;
    int status;

    html_header();
    html_begin("Pokusna stranka");

    switch ( status = read_cgi_input(&cgi_data) ) {
    case -1:
        printf("Chyba v CGI vstupu<br>\n");
        break;

    case 0:
        printf("Bez parametru<br>\n");
        break;

    default:
        printf("Pocet parametru: %d.<br>\n", status);
        break;
    }

    html_end();
    list_clear(&cgi_data);

    return (0);
}
```

Výpis č. 3: Program `priklad1.cgi.c`

Pár řádek pro neznalé: Knihovnu přeložíte pomocí příkazu `make` v adresáři knihovny (např. `cghtml-1.69`). Ten zkompileje jednotlivé `.c` soubory a vytvoří statickou knihovnu `cghtml.a` (něco kolem 50kB), kterou slinkujete s Vaším CGI programem. Takže pokud knihovna je v podadresáři `cghtml`, pro přeložení našeho příkladu stačí provést:

```
$ make -C cghtml      # Zkompletuje knihovnu
$ gcc -o prik1ad1.cgi prik1ad1.cgi.c\
  cghtml/cghtml.a
```

Pro další programy stačí samozřejmě uvést pouze druhý řádek. Pokud vás děsí velikost výsledného `.cgi` která koluje kolem 40kB, stačí soubor „odstripovat“ pomocí `strip prik1ad1.cgi`, což soubor podstatně zestíhlí na nějakých 13kB, a to je skutečné nic (nestripujte knihovnu). Takže první kompilaci máme za sebou a můžeme program spustit. Činnost programu je opravdu velice minimální: pouze si zjistí, zda byl spuštěn s parametry nebo ne, a vypíše na to svůj názor. Při spouštění z webového klienta se jeví úplně normálně, ale zkuste jej spustit z shellu: při volání `f-ce read_cgi_input()` se ohlásí dříve zmiňovaná výhoda `cghtml` knihovny: vstup můžete zadat ručně! Ale dost chvály a pojďme si jej rozebrat řádek po řádku.



Immediately at the beginning is the inclusion of header files. Each system and two of `cghtml`. The `main()` function uses two variables: `cgi_data`, which is a list of CGI parameters and `status` — a return value from `read_cgi_input()`. Watch out for the `switch` construct, which allows you to branch the CGI execution clearly and simply according to `status` in the case you know the possible number of parameters coming into CGI. `read_cgi_input()` returns `-1` in the case of a read error (e.g. incorrectly coded parameters), `0` if no parameter was specified and in all other cases the number of parameters. As the first one, it calls `html_header()` and `html_begin()`. The first one sends the HTTP header („Content-type: text/html“). The second one starts the HTML document with the title (which is a parameter). The so-called „closing“ are `html_end()` and `list_clear()`, which ends the HTML document and releases dynamically allocated memory inside `cgi_data`. It is not a disadvantage if these two functions are defined as `exit` functions (viz `atexit(3)`).

For access to CGI parameters, use the `cgi_val()` function, which has two parameters: the name of the CGI (`cgi_data`) and the name of the parameter to be retrieved („prijmeni“). The function returns the text form of the parameter: thus for the call `cgi_val(cgi_data, "prijmeni")` the value returned by the function was „Sezemsky“. Be careful also about the fact that in `read_cgi_input()`, and in `list_clear()` the parameters are pointers and in `cgi_val()` and in other functions they are not. Another useful function is `cgi_val_multi()`, which returns the value (char \*\*) which has the same name. I recommend you to use it somewhere in the program to define a macro like `CGI`, which will simplify access to CGI values.

```
#define CGI(Name)  cgi_val(cgi_data, Name)
```

This is of course not a complete description of the library, but for basic use it is rich enough. If you are interested in the library, you should read the whole manual, which is attached to the library.

Now we will add the example with the form from the article [Co je to HTTP a HTML?](#). For the summary: parameters „jmeno“, „vek“, „pohlavi“, „zajmy“, „inet“ a „pozn“. With these parameters we will not do anything for the example, but thanks to the C language it is easy to put it into the source code....

### Složitější kousky: upload

How to do? Very easy. In the form you insert the tag

```
<input type=file name=soubor>
```

Which in the client looks like a text line and a button for file search. After pressing the button, the user selects a file and sends the CGI script. The `cghtml` library simply puts the file into the address defined in `Makefile` (usually it is `/tmp`), and its name is put into the CGI variable. Thus the full path is „`/tmp`+`cgi_val(cgi_data, „soubor“)`“. Then you can move, delete, or display the file wherever you want.

### Složitější kousky: postupné zobrazování výstupu

Example: on your web page you have an installed search engine, but its response is not the fastest, so the user may think that there is an error (I know, it is a bit stupid, but nothing else came to my mind). Thus the search engine should output the header, so the user can see that something is happening and then the search results page (if it is a long one).



```
#include <stdio.h>
#include "cgihtml/cgi-lib.h"
#include "cgihtml/html-lib.h"

#define CGI(Name)  cgi_val(cgi_data, Name)

int main(void)
{
    llist cgi_data;

    html_header();
    html_begin("Výstup z formuláře");

    switch ( read_cgi_input() ) {
        case -1:
            h1("Chybný vstup");
            break;

        case 0:
            h1("Nelze spouštět CGI bez parametrů");
            break;

        /* jmeno, vek, pohlavi, zajmy, inet, pozn */
        case 6:
            printf("<h1>Formulář
            pro %s</h1>", CGI("jmeno"));
            /* .... zpracovávání vstupních údajů ... */
            break;

        default:
            h1("Zkuste <a href=form1.html>
            správný formulář</a>");
            break;
    }

    return (0);
}
```

Výpis č. 4: Příklad formuláře ještě jednou

vě o desetiny sekund, tak se nic neděje, ale pokud prohlédáte např 10GB :~0, nebo máte HODNĚ pomalé vyhledávání, může to být i minuta 8~0). CGI by se mělo jmenovat jako `nph-jmeno.cgi`. Tohle nejspíš záleží na serveru, ale u Apache je to takto. Tyto soubory pak server interpretuje postupně. U Lynxe to asi nevidíte, ale na Mozille je to vidět (používám shellácký script, aby bylo vidět, že nepotřebujete nutně nějakou knihovnu).

Script si uložte třeba jako `nph-search.cgi` a zkuste si jej.

Tímto bych zakončil tuto rychlokuchařku pro CGI. Byl to skutečně jen skromný text, a zdaleka neobsahuje vše (zvláště o knihovně `cghtml`, bezpečnost), ale pro základní používání a seznámení to snad bohatě stačí. Pokud máte nějaké dotazy, popřípadě připomínky, nebo máte nějaké problémy, napište mi na (2). Snad vám pomůžu. ■

```
1 cghtml
  http://www.eekim.com/software/cgihtml/
2 sezi@sps-jia.cz
  mailto:sezi@sps-jia.cz
```

```
#!/bin/sh

echo "HTTP/1.0 200 OK"
echo "Content-type: text/plain"
echo

echo "Šnek.cz: velice pomalý vyhledávač"
echo "byl zadán dotaz na slovo: linux"
echo
echo "Odkaz #1: http://www.linux.org/"
sleep 2
echo "Odkaz #2: http://www.linux.cz/"
sleep 2
echo "Odkaz #3: http://www.linux.sk/"
sleep 2
echo
echo "Nalezeny 3 odkazy"
echo "(C) Marek Pomalý, 1999\
  Optimalizováno na rychlost."
```

Výpis č. 5: `nph-search.cgi`

## Rychlo-školička pro Makefile

Marek Sezemský, 27. března 1999

Když jsem v Linuxu začínal psát vlastní programy v céčku, pro kompilaci stačil jeden řádek v shellu. Jak jsem začal zpřehledňovat (a komplikovat) do více souborů, bohatě stačil script, který natvrdo vše zkompiloval a slinkoval. Bohužel je to funkční, ale pomalé a nepřehledné řešení. Pro kompilace malých i obrovských (jádro, apache, sendmail) programů/knihoven se prostě používá pouze a jedině Makefile.

### Jak to pracuje?

Makefile je sám o sobě textový soubor, ve kterém jsou popsány závislosti různých souborů na jiných souborech. Program, který s ním umí zacházet se jmenuje „make“. Ten při svém spuštění hledá v aktuálním adresáři soubor `Makefile`, `makefile` a nebo `GNUmakefile`. Pokud jej nalezne, zkontroluje data souborů a podle závislostí v něm definovaných provede akci, která přísluší té které závislosti. Čtete dál, pochopíte.

### Formát souboru Makefile

Uvedu příklad pro vysvětlenou: píšete program který pracuje s Internetem. Krom programu samotného potřebujete knihovnu pro síť a uživatelské rozhraní. Program je v `main.c`, síť v `net.c` a komunikaci s uživatelem obstará `gui.c`. Existují také halvičkové soubory `net.h` a `gui.h`. Program používá funkce z `net.c` a `gui.c`, takže si inkluduje i hlavičky. Makefile pro tento projekt vypadá asi takto:

```
# Program (C) Sezi, 1999
#

program: main.o net.o gui.o
        $(CC) -o $@ main.o net.o gui.o

main.o: main.c net.h gui.h
net.o: net.c net.h
gui.o: gui.c gui.h
```



Vytvořte si tento Makefile a výše zmíněné soubory. Mohou být prázdné, ale main.c musí obsahovat f-ci main() (klidně i prázdnou).

Ale teď již na Makefile: první dva řádky jsou komentáře (hojně používejte). Třetí řádek popisuje to hlavní: na čem je závislý 'program': soubory main.o net.o gui.o. Následující řádek je příkaz, který se provede pokud bude některý soubor novějšího data než 'program'. \$(CC) znamená 'makro s jménem CC' a \$@ se rozvine jako text před dvojtečkou (takže 'program'). Tyto (a další) makra si rozebereme později, prozatím stačí vědět, že \$(CC) je jméno kompilátoru. POZOR: Příkaz v Makefile musí začínat tabulátorem! Pokud použijete mezery, make to nezbaští a pošle vás někam. Další řádky definují závislosti jednotlivých .o souborů. Takže pokud máte vytvořeny soubory včetně Makefile, zkuste spustit „make“. Pravděpodobně uvidíte tyto řádky:

```
$ make
cc -c main.c -o main.o
cc -c net.c -o net.o
cc -c gui.c -o gui.o
cc -o program main.o net.o gui.o
$
```

Jelikož make nenalezl žádný z souborů .o, prostě postupoval po závislostech a zkompiloval soubory uvedené na posledních 3 řádkách. Poté vykonal řádku 'program'. Pokud zkusíte změnit datum u gui.c („touch gui.c“) a opět spustíte make, zkompiluje se pouze gui a program se celý slinkuje.

```
$ touch gui.c
$ make
cc -o gui.o gui.c
cc -o program main.o net.o gui.o
$
```

Nyní si zkuste touch na nějaký hlavičkový soubor (např. net.h) a poté opět make. Nyní se již přeloží net.c i main.c, neboť ten je na net.h závislý též. Nyní ručně vymažte „program“ plus všechny .o soubory a zkuste „make net.o“. Přeloží se pouze net.o a vše ostatní se nechá na pokoji. Make si totiž při spuštění bez parametrů vezme první závislost a tu zpracovává. Parametrem lze ale zadat, kterou část Makefile bude zpracovávat (net.o). Ale v Makefile nemusí být uvedeny pouze soubory: zkuste si na konec našeho Makefile připsat následující řádky (nezapomeňte na tabulátor),

clean:

```
Tab rm -f *.o program
```

a spustit „make clean“. Vymažou se soubory vzniklé kompilací. Takže už víme, že pokud za dvojtečkou není uvedeno žádný soubor, příkaz se provede vždy.

Nyní zpět k makrům. Makra se definují jako jméno = hodnota a odkazy na ně jsou jako \$(jméno). Takové základní a univerzální jsou CC (kompilátor) a CFLAGS (volby ke kompilátoru). Pokud si např. na začátku Makefile nastavíte CC na gcc, tak se místo kompilátoru CC použije GCC. Stejně tak CFLAGS. To samozřejmě i pro řádky, kde toto makro není uvedeno. Ale pokud se rozhodnete a tato makra předefinujete, tak se v našem příkladu v sekci „program“ nebude brát v potaz makro CFLAGS, protože není uvedeno v příkazu.

Dále fungují i příkazy include (vlození jiného souboru), ifdef ... else ... endif a ifeq ... else

... endif. První je na test definice makra, druhý porovná výrazy. Koho to zajímá, necht' pohlédne do Makefile u jádra, kde jsou obě podmínky a spousta dalšího názorně používány. Nevysvětluji to proto, jelikož málokdo něco z toho používá (kromě Linuse). Ale jejich použití je skutečně jednoduché.

### Parametry programu make

Ačkoliv se to nezdá, program make má mnoho parametrů. Jejich použití je vysvětleno v manuálu (pod make(1)). Píše se tam sice něco o rozsáhlejší dokumentaci, ale popis parametrů v man-u stačí bohatě. Abych nezdržoval, popíšu jen ty důležitější.

-C <dir>: před načítáním Makefile se přepne do adresáře <dir>. Pokud je zadáno více parametrů (např. „-C /-C etc“ je to totéž jako „-C /etc“). Používá se při rekurzivním volání make.

-f <file>: použije soubor <file> místo vyhledávaného Makefile.

-j <num>: počet souběžných procesů — (nejen) pro SMP.

-n: nevykonává příkazy, pouze je vypisuje.

-s: tiché (silent) vykonávání (nevypisuje příkazy).

Makefile se samozřejmě nemusí používat pouze pro programy v C/C++, stejně dobře zvládá i ostatní programovací jazyky a další funkce, ve kterých se kontroluje závislost souborů na časech modifikace, je to opravdu skvělá pomůcka.

Doufám, že vám tento článek pomohl při pronikání do programátorských zákoutí Linuxu. Pokud máte zájem, po vstřebání těchto informací si přečtete další článek, tentokrát o automatizaci tvorby Makefile (v Linových novinách ze srpna a září). ■

### Zasmáli jsme se!

Pavel Janík ml., 1. dubna 1998

I v tomto čísle jsme nezapomněli na naši nejčtenější rubriku :-). Tato je poněkud jiná — je kompletně naplněna příspěvky čtenářů všech možných i nemožných konferencí a hlavně (což mne nejvíce těší) od vás, čtenářů Linuxových novin. Ale začněme, jistě vás nezajímají výlevy mé duše, ale rubrika samotná...

Vrhneme se tedy nejdříve do světa signatur. Michal Šafránek pravděpodobně ještě žádnou dívku nenašel. Proč si to myslím? Inu když má někdo v signatuře následující text, není se čemu divit.

Hledám dívku, která dokáže nakonfigurovat sendmail a není jenom .jpg

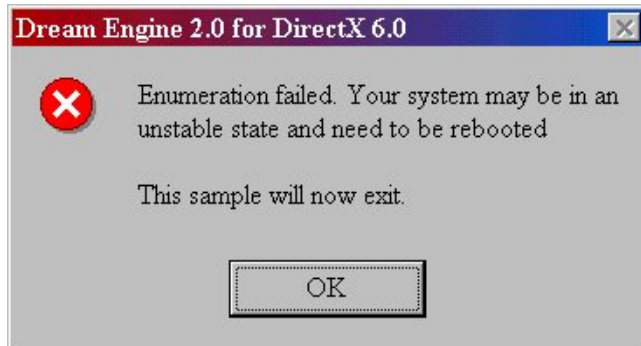
(Pozn. redakce: omlouváme se těm dívkám, kterým jsem ukřivdil. Pokud se tak některá slečna cítí, prosím napište mi: -))

Windows jsou věčné téma...

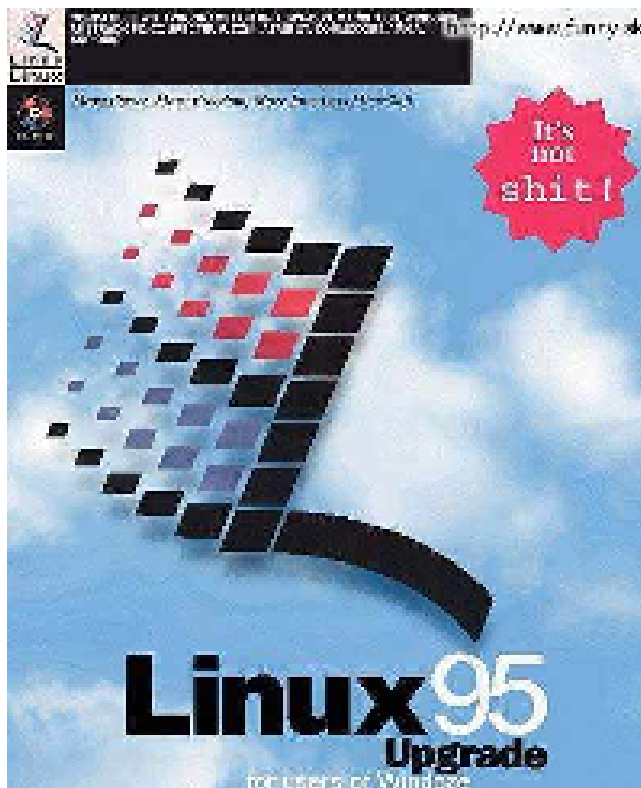
Vzhledem k drobným technickým problémům se distribuce nového systému Windows 2000 odkládá na první kvartál roku 1901.



Customer: "I've just installed Windows 98..."  
 Tech Support: "And?" Customer: "The  
 computer stopped working." Tech Support:  
 "You already said that..."



Je libo nový operační systém?



My operating system unders`1 long filena`1,  
 and yours?

Stano Meduna si všiml zajímavého threadu v konferenci  
 redhat-announce:

```
> > What is the newest version of RedHat
> > Linux? is it 5.2?
> > When will the next version come?
>
> This newsgroup is for *ANNOUNCEMENTS*,
> not questions!
>
```

Remember, he just announced that people are  
 \*Stupid\*.

Když jsem si kompiloval novou verzi balíku gettext, troš-  
 ku mne zarazila následující hláška:

WARNING: Warnings can be ignored. :-)

No nelekli byste se taky?

V konferenci linux-kernel se také živě diskutovalo (a stále  
 ještě diskutuje) na téma GNU a Linux resp. GNU/Linux  
 apod. Jistě to znáte... V jedné z odpovědí se také objevil  
 následující příspěvek týkající se programu /bin/false:

```
#!/bin/sh
```

```
usage="Usage: $0 [OPTION]...
Exit unsuccessfully.
```

```
--help      display this help and exit
--version   output version information and exit
```

Report bugs to sh-utils-bugs@gnu.ai.mit.edu"

```
case $# in
1 )
  case "z${1}" in
z--help )
echo "$usage"; exit 0 ;;
z--version )
echo "false (GNU sh-utils) 1.16"; exit 0 ;;
* ) ;;
  esac
;;
* ) ;;
esac
exit 1
```

It caught me as a surprise to see that true and  
 false are such a bloats. And even such a simple  
 program as false has a bug: try  
 false --help --version  
 Not that this is a serious bug ;)

A co k tomu poznamenal Linus?



A zero-length /bin/true does not work reliably, and never has.  
 However, making it do anything but a simple "exit 0" is horrible. Anybody who really thinks /bin/true should report a version number and a help string (or even a copyright notice) needs to get his head examined.  
 Yes, the FSF coding style says it should do it. But you should never let rules overrule common sense — if you do, you end up doing stupid things (this is especially true of the FSF codingstyle — see the kernel Documentation/CodingStyle about some other issues).

A nakonec jsem si nechal velice vtipný (dámý prominou) příspěvek z konference linux-kernel. Začalo to nevinně...

With all the recent press about Linux, it's a good thing Linus' name isn't Peter.

Hans Peter Anvin si nebyl jistý, jak to autor myslel a tak se ptá na důvod:

And what's wrong with being named Peter?

A Richard B. Johnson odpovídá:

If linus was peter, Linux would be called penix? (just guessing) ;)

If the telephone is such a great idea, why are there so few books explaining how to use it?

Podobná odpověď:

There are more books about Windows than about Linux.

Autor měl signatuře napsáno:

Python: executable pseudo-code; Perl: executable line noise

Jiná odpověď:

Python is just toooooo easy for most tasks. You really don't need any books. If you were asked to write a book on it you would be rather stuck, other than repeating the excellent texts already out.

Jedna z veselejších odpovědí:

A Dead Python is still more useful than a Perl.

Thread pokračuje:

A perl script is like a pet alligator. It's cute when it's small, but it keeps growing... :)

Zatím poslední (zajímavější) příspěvek:

"Python is the first scripting language good enough to criticise". :-)

Doufám, že se alespoň někdo pousmál, pokud dojdou nějaké další pěkné příspěvky, dám vědět. ■

## Is Python dying?

Věroš Kaplan, 31. března 1999

I v newsové skupině *comp.lang.python* je občas veselo. V poslední době se objevil dotaz „Is Python dying?“. Autor píše, že má rád Python, leč nepotkal moc knih o Pythonu, přestože existují knihy o Perlu, Javě, C, C++ a dokonce i Tcl/Tk. Pravděpodobně nejpodstatnější dotaz z článku je:

If Python is such a great language, why are there only 2 books (versus over 60 for Perl) and so few people talking about it?

Odpovědi na sebe nenechaly dloho čekat a některé jsou docela zajímavé: (rozhodně nechci zvednout flamewar o tom, co je lepší)

Python grew up in Academia, not on UseNet. Perl requires more books (this sounds silly, but... YOU try learning by reading someone else's scripts :-)  
 There is a wider range of introductory documentation for Python that is freely available.

Další odpověď na dotaz o knihách zněla:



## Linuxové noviny a jejich šíření

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

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

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

Registrované známky použité v tomto textu jsou majetkem jejich vlastníků.

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

Linuxové noviny můžete najít na akademické síti TEN-34 CZ (3), na síti IBM Global Network na adrese (4), na serveru Gymnázia Vídeňská v Brně (5) na serveru časopisu Netáčik (6), který je připojen do slovenského SIXu, případně na serveru Mathew (7).

Linuxové noviny jsou k dispozici také ve formátu HTML na adrese (8) nebo také na slovenském serveru (9). ■

1 České sdružení uživatelů operačního systému Linux

<http://www.linux.cz/czlug>

2 Adresa redakce

<mailto:noviny@linux.cz>

3 Linuxové noviny na síti TEN 34-CZ

<ftp://ftp.fi.muni.cz/pub/linux/local/noviny>

4 Linuxové noviny na síti IBM Global Network

<ftp://ftp.inet.cz/pub/People/Pavel.Janik/noviny>

5 Linuxové noviny na komerční síti CESNET

<http://www.gvid.cz/linux/noviny/>

6 Slovenské zrcadlo Linuxových novin

<ftp://netacik.sk/pub/linux/cz-noviny>

7 Linuxové noviny — Mathew

<http://www.mathew.sk/noviny>

8 Linuxové noviny ve formátu HTML

<http://www.linux.cz/noviny>

9 Linuxové noviny ve formátu HTML na Slovensku

<http://noviny.demax.sk/>



**Šéfredaktor:** Pavel Janík ml.

<mailto:Pavel.Janik@linux.cz>

**sazba:** Ondřej Koala Vácha

<mailto:koala@informatics.muni.cz>

**jazykové korekce:** Bohumil Chalupa

<mailto:bochal@met.mff.cuni.cz>

**překlady:** Hanuš Adler

<mailto:had@pdas.cz>

**převod do HTML:** Pavel Juran

<mailto:xjuran@cs.felk.cvut.cz>

