



Abíčko

Časopis serveru AbcLinuxu.cz

leden 2003

Sponzorem tohoto čísla je společnost move.cz



Vychází také na CD-ROM jako příloha časopisu:

PC WORLD

Obsah

Editoriál	4
Je Linux opravdu zdarma?	5
Úvod	5
Projekt GNU	5
Linux	5
Nulová cena?	6
Závěr	6
Přehrávání DVD pod Linuxem	7
Teorie	7
Anamorfní formát	7
Regiony	7
Ochrana DVD	8
Praxe	8
Xine	8
MPlayer	9
Ogle	12
VideoLAN Client	12
Závěr	13
Diskuse	13
Moderní souborové systémy – ext3	14
Úvod a motivace	14
Módy Ext3 filesystému	14
Writeback	14
Ordered	14
Journal	15
Práce se souborovým systémem Ext3	15
Instalace	15
Nastavení	15
Ext3 jako modul	16
Souhrn	16
Kancelářský balík Hancom Office (1)	17
Úvod	17
Instalace	17
První dojmy	17
Hancom Word 6	18
Běžné funkce	19
Kompatibilita s MS Office	19
Tisk	20
Kontrola pravopisu	20
Hodnocení	20
Elektronické bankovníctví – eBanka	22
Úvod	22
Funkce	22
Plusy a mínusy	23
Závěr	24
Diskuse	25

Moderní souborové systémy – ReiserFS	26
Úvod a motivace	26
Instalace	26
Parametry zrychlující práci filesystému	27
Změna velikosti filesystému	27
Shrnutí	27
GNU Lilypond – dokončení	28
Vzhůru do příkladu	28
Střeva Lilypondu	29
Zpět k příkladu	30
Zápis not	31
Zápis textů	32
Výhody Lilypondu	33
Od kódu k tisku a dál	33
Podpůrné prostředky	34
Čeština	35
Co říci závěrem?	35
Jaderné noviny 193–196	36
Úvod	36
Pročišťování aplikačního rozhraní devfs	36
Stav ACPI	36
Bugzilla sleduje chyby v jádře	36
Problémy s moduly v řadě 2.5	37
Masivní zpomalení u SMP s 2.4.17	37
Podpora zvuku na čipsetu VIA VT8233A	38
RTLinuxFree 3.2	38
Změny v IDE	38
Reverse-Mapping verze subsystému virtuální paměti	38
XFS patche pro jádro 2.4.20-rc3	38
Vlastnost či chyba Athlonu?	39
Nový grafický nástroj na konfiguraci kernelu	39
Nastavení limitu zdrojů pro uživatele	39
Co způsobilo korupci dat v ext3 v jádře 2.4.20	39
Návrh na dynamickou správu napájení	39
NTFS 2.1.0	40

Editoriál

Vítejte u čtení časopisu Abíčko.

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

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

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

Náměty na články zasílejte do konference našich autorů: autori@abclinuxu.cz.

Sponzoring Abíčka a jiné formy reklamy si objednávejte na adrese: reklama@abclinuxu.cz.

Ostatní dotazy směrujte na adresu: literakl@abclinuxu.cz.

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

©2002 AbcLinuxu s.r.o. a autoři článků

Sazba: Ondřej Krejčík

Pro nekomerční účely smíte tento dokument jakkoliv šířit v tištěné i digitální podobě. V ostatních případech nás požádejte o svolení na adrese literakl@abclinuxu.cz.

Je Linux opravdu zdarma?

Leoš Literák

Úvod

Znáte to. Zaplatíte za softwarové vybavení počítače skoro dvacet tisíc korun. Máte-li štěstí, vše poběží, jak má. Nemáte-li a operační systém se začne sekát nebo textový editor po dopsání poslední stránky kompletně zničí formátování dokumentu, co uděláte? Kdyby software byl jako boty, šli byste jej reklamovat a žádali jeho výměnu za funkční produkt. Jenže u softwaru to neplatí – přečtěte si licenční ujednání. Přišli jste po pádu o veškerou obchodní korespondenci? Zapomeňte na žalobu o náhradu škody. Přečtěte si licenci a zjistíte, že nemáte vůbec žádná práva. Jste jen otrokem či rukojmím v rukou softwarových firem.

Projekt GNU

Existuje cesta ven, k nezávislosti a svobodě. Je jí Linux a projekt GNU, jehož cílem je vytvořit svobodný software. Tady často dochází k nedorozumění – lidé si pod termínem Free Software představují nějaký freeware či obecně software, který je zdarma. Free ale není o ceně, nýbrž o vašich právech. Většina softwaru v tomto světě používá licenci GNU GPL 2, která vám sice podobně jako u komerčních firem neposkytuje žádné záruky o funkčnosti, ale na druhou stranu vám dává výrazná práva k danému dílu. A hlavně zaručuje, že vám tato práva někdo v další verzi nevezme.

Free Software je hlavně o pokroku a spolupráci. Někteří jeho ideje pohrdavě nazývají komunistickými, ale to je nesmysl. Takoví lidé je buď nepochopili, anebo se úmyslně snaží poškodit dobré jméno tohoto hnutí. Hlavní cíle vycházejí z vědecké praxe, která byla uplatňována po celou existenci lidstva. Myslíte si snad, že se za použití ohně či kola platily v pravěku licenční poplatky? Jak by svět vypadal, kdyby námořní kapitáni považovali mapy za své intelektuální vlastnictví a nikomu je neukazovali? Spolupráce a volná výměna znalostí a zkušeností akceleroval vývoj. Různé patenty jej naopak brzdí. Proto Free Software (či obdobné hnutí Open Source) pokládá takový důraz na svobodu zdrojového kódu.

Linux

V osmdesátých letech Richard Stallman založil hnutí Free Software, které započalo pracovat na projektu GNU. Jejich snahou bylo vytvořit programové prostředí, které má a vždy bude mít svobodný zdrojový kód. Vývojáři dokázali vytvořit překládač programovacího jazyka C a všechny hlavní nástroje a aplikace. Chybělo jim však jádro, s nímž by dokázali vytvořit kompletní operační systém.

Začátek devadesátých let přinesl levný hardware (PC založené na procesoru 80x86) a nástup internetu. Finský student Linus Torvalds byl znechucen omezeními operačního systému Minix a zkusil si naprogramovat vlastní operační systém pro svůj počítač s procesorem 386. Své pokusy publikoval na internetu pod licencí GPL. A stalo se nečekané – desítky vývojářů z celého světa se k němu připojily a začaly mu pomáhat vylepšovat tuto malou práci. Během krátké doby vznikl použitelný kód, který dokázal obsluhovat počítač! A to počítač, který byl cenově dostupný, ne drahé sálové počítače. Projekt GNU tak získal funkční jádro a zformoval tak svůj první kompletní operační systém.

Během několika let se komunita kolem Linuxu rozrostla z desítek vývojářů na několik set tisíc. Proč? Zvolená licence zajišťovala, že jejich příspěvek v podobě úprav zdrojového kódu nikdy nebude ztracen či zneužit v uzavřeném softwaru. První je zaručeno tím, že kdokoliv smí dělat úpravy, ale nesmí změnit licenci. Takže i kdyby se původní autor přestal věnovat vývoji, může být snadno nahrazen. A zneužití je zabráněno nakažlivostí zdrojového kódu vydaného pod licencí GPL. Začlenění třeba jediné řádky takového kódu znamená, že i veškerý ostatní kód musí být šířen pod touto licencí. Což ve svém důsledku znamená zveřejnění zdrojového kódu.

Nulová cena?

Operační systém GNU Linux je tvořen jádrem, základními knihovnami, nástroji, grafickým rozhraním a aplikacemi. Všechny tyto součásti jsou chráněné licencí GNU GPL verze 2. Ta zaručuje, že tyto prvky vždy zůstanou pod touto licencí. Z toho vyplývá, že Linux je a vždy bude zdarma.

Jak je ale možné, že některé distribuce jsou komerční a za jejich používání se platí? Tyto firmy neprodávají Linux, ale jeho balení. Vezmou totiž jednotlivé části a přeloží je do binární podoby, propojí je, otestují výsledek, napíší instalační příručku a výsledek zabalí do úhledné krabice. Navrch ještě přidají podporu, ať už emailem nebo po telefonu. Úpravy zdrojového kódu jednotlivých komponent v souladu s licencí GPL zveřejní.

Ale distribuce sama je samostatný produkt, pro který si mohou zvolit libovolnou licenci – třeba i komerční. GPL se týká nakládání se zdrojovým kódem, nikoliv používání výsledného programu, proto ji distributoři neporušují. Ve skutečnosti tyto firmy jako MandrakeSoft, Redhat, nebo SuSE výrazně přispívají vývoji Linuxu, neboť zaměstnávají spoustu vývojářů Free Softwaru. Takže zakoupením distribuce vlastně zprostředkovaně podporujete další vývoj Linuxu.

Závěr

Jak jste si mohli přečíst, operační systém Linux je opravdu šířen zdarma. Jeho licence zaručuje, že tomu tak bude navždy.

Přehrávání DVD pod Linuxem

Robert Krátký

Teorie



Přehrávání DVD na počítači má spoustu výhod. V tomto článku se podíváme na možnosti Linuxu v této oblasti. Obecně lze říci, že mezi hlavní přednosti sledování DVD na počítači patří zejména nízká cena nutného vybavení a kvalita obrazu. Širokoúhlý formát monitorů se však standardem v dohledné době asi nestane. To nás v případě širokoúhlých prezentací (naprostá většina filmových disků) odsuzuje k černým pruhům na vrchním a spodním okraji obrazovky. Nicméně vzhledem k vyššímu rozlišení monitoru oproti TV máme záruku, že nezůstaneme ochuzeni ani o poslední kousek digitální informace, který se autorům podařilo na DVD nacpat.

Anamorfní formát

Jedním z největších přínosů DVD je možnost sledovat filmy v takzvaném anamorfním formátu. Velice stručně řečeno je to takový zápis obrazu, který využívá pro ukládání video-dat 33% plochy z oněch černých pruhů, které známe ze širokoúhlých filmů. Televize s poměrem stran 16:9 (širokoúhlá) pak tato data využije k zaplnění celé obrazovky, místo aby pouze zvětšila obraz zaznamenaný mezi černými pruhy. Docílí se tím větší ostrosti (a tedy kvality) obrazu. Počítačový monitor se vzdor svému nevyhovujícímu poměru stran (4:3) zachová podobně, protože jeho vysoké rozlišení umožňuje využití plného rozlišení videa. Černé pruhy už jsou pak jen „vata“ doplněná softwarovým přehrávačem. Podrobné a názorné vysvětlení anamorfního přepisu lze nalézt na stránce: *What's an anamorphic DVD* (<http://gregl.net/videophile/anamorphic.htm>)?

Podpora multimédií pod Linuxem má své mezery hlavně díky tomu, že (stále ještě) většina výrobců hardware (a souvisejícího software) považuje Windows za svůj primární trh. V případě přehrávání DVD si však skoro není na čem stěžovat. Dokonce si troufám tvrdit, že na konci z toho linuxový uživatel vyjde lépe. Hlavním důvodem je (překvapivě) cena, kterou je třeba zaplatit za programové vybavení. V Linuxu se dostupnost free software bere jako samozřejmost, nicméně uživatelé Windows by to s hledáním (rozumně funkčních) nekomerčních alternativ k programům přehrávajícím DVD měli dost nesnadné.

Regiony

Ještě než se dostaneme k samotnému jádru věci, je třeba zmínit dvě (nepříjemné) záležitosti. Většina těch, kteří alespoň zaslechli o DVD, jistě ví, že kvůli nátlaku produkčních společností velkých filmových studií byl svět DVD rozdělen na regiony a většina disků i přehrávačů (včetně DVD-ROM mechanik) je vyráběna tak, aby tomuto požadavku vyhovovala. V praxi to znamená, že disk kódovaný pro jeden region nelze přehrát v mechanice kódované pro region jiný. Hlavním důvodem, který k zavedení podobného modelu vedl, je mnohdy výrazné zpoždění, se kterým jsou disky vydávány v různých částech světa. Jedno vyhledávání hesla "DVD regions" na Google.com odhalí, co si o tom (nejen) internetová komunita myslí. Mapa regionů je například na adrese http://www.moviesuk.fsnet.co.uk/dvd_regions.htm.

Stolní přehrávač, který regionální kódování ignoruje není nijak těžké v našich končinách za rozumný peníz sehnat. Ostatně i přehrávače výrobcem kódované lze zakoupit již upravené tak, aby přehrávaly disky s jakýmkoliv regionálním kódováním. S DVD-ROM mechanikami, o které nám teď jde především, je to však těžší. Pokud nepatříte k těm, kteří si koupili tzv. Region-Free mechaniku již před drahnou dobou, bude se vám taková shánět velmi těžko. Je tu sice možnost použít "hacknutý" firmware, který vám z vaší DVD-ROM mechaniky udělá také Region-Free, avšak spolehlivost takového "upgrade" není rozhodně stoprocentní. Záruku na bezcenný kousek kovu, který jste umrtvili nepovedeným nahráváním nelegálního firmware asi také těžko uplatníte. Z toho vyplývá, že většina z nás si asi na počítači americké DVD nepustí.

Ochrana DVD

Druhým, neméně závažným problémem je ochrana proti kopírování. Ta se vás však týká i tehdy, jsou-li vaše úmysly počestné a kopírovat se nic nechystáte. Takové ochrany jsou na DVD většinou hned dva druhy: digitální a analogová. Analogová slouží k zabránění nahrávání obsahu DVD na analogový přístroj, nejčastěji VHS a známá je pod názvem MacroVision – narazit na problém můžete v podstatě pouze, pokusíte-li se přehrávat DVD z počítače na externím TV.

S digitální ochranou je to horší. Jmenuje se CSS (Content Scramble System) a je to systém, který je použit pro „zašifrování“ informací na DVD za použití speciálních klíčů tak, aby je mohl číst pouze schválený hardware/software (čili takový, jehož výrobce/autor za použití algoritmu zaplatí). V případě přehrávání na počítači provádí rozkódování buď MPEG-2 dekodér v podobě přídatné karty nebo (častější případ) přímo program použitý k přehrávání DVD, neboli softwarový dekodér. Ačkoliv dešifrování CSS nepředstavuje velký problém a zdrojový kód programku, který dokáže CSS klíče rozluštit, má pouhých pár řádků, právní strana otázky zdaleka tak prostá není (viz zajímavá přednáška dotýkající se jak právních, tak technických aspektů CSS). Někteří autoři open-source software umožňujícího přehrávání DVD proto nezačleňují podporu CSS dekodování přímo do programu a ponechávají na uživateli rozhodnutí, zda si tuto podporu (např. ve formě „cizího“ pluginu, viz Xine) nainstaluje dodatečně.

Praxe

Po nezbytném úvodu se můžeme konečně vrhnout na to nejdůležitější. Na internetu je již dlouhou dobu v oběhu Linux-DVD-Howto, ale vzhledem k datu zpracování již není příliš aktuální. Každopádně však stojí za přečtení, pokud stále používáte kernel řady 2.2.x.

Základním předpokladem je samozřejmě fungující DVD-ROM mechanika. Většina programů ji bude hledat pod `/dev/dvd`, a proto je třeba, pokud ještě neexistuje, takový symlink vytvořit. Důležité je mít i zapisovací(!) práva k tomuto zařízení, aby fungovalo dešifrování CSS. Dále se ujistěte, že DMA je zapnuto, protože výrazně ovlivňuje výkon. Současné nastavení se dozvíte třeba pomocí příkazu **hdparm -v /dev/dvd**. Doporučuji také Xfree 4 (a to hlavně kvůli podpoře Xv + stručný popis v manuálu ke Xine), i když ani to není nutné (viz článek o framebufferu právě zde na AbcLinuxu.cz).

Zbývá zvolit program, kterému svěříme ostatní starosti. Máme docela slušný výběr, což je jen dobře, protože na různých strojích a instalacích se jednotlivé programy zhusta chovají dost odlišně a výsledek tomu odpovídá. Nejznámější programy, ze kterých můžeme vybírat (řazeno náhodně):

- Xine
- MPlayer
- Ogle
- VideoLAN Client

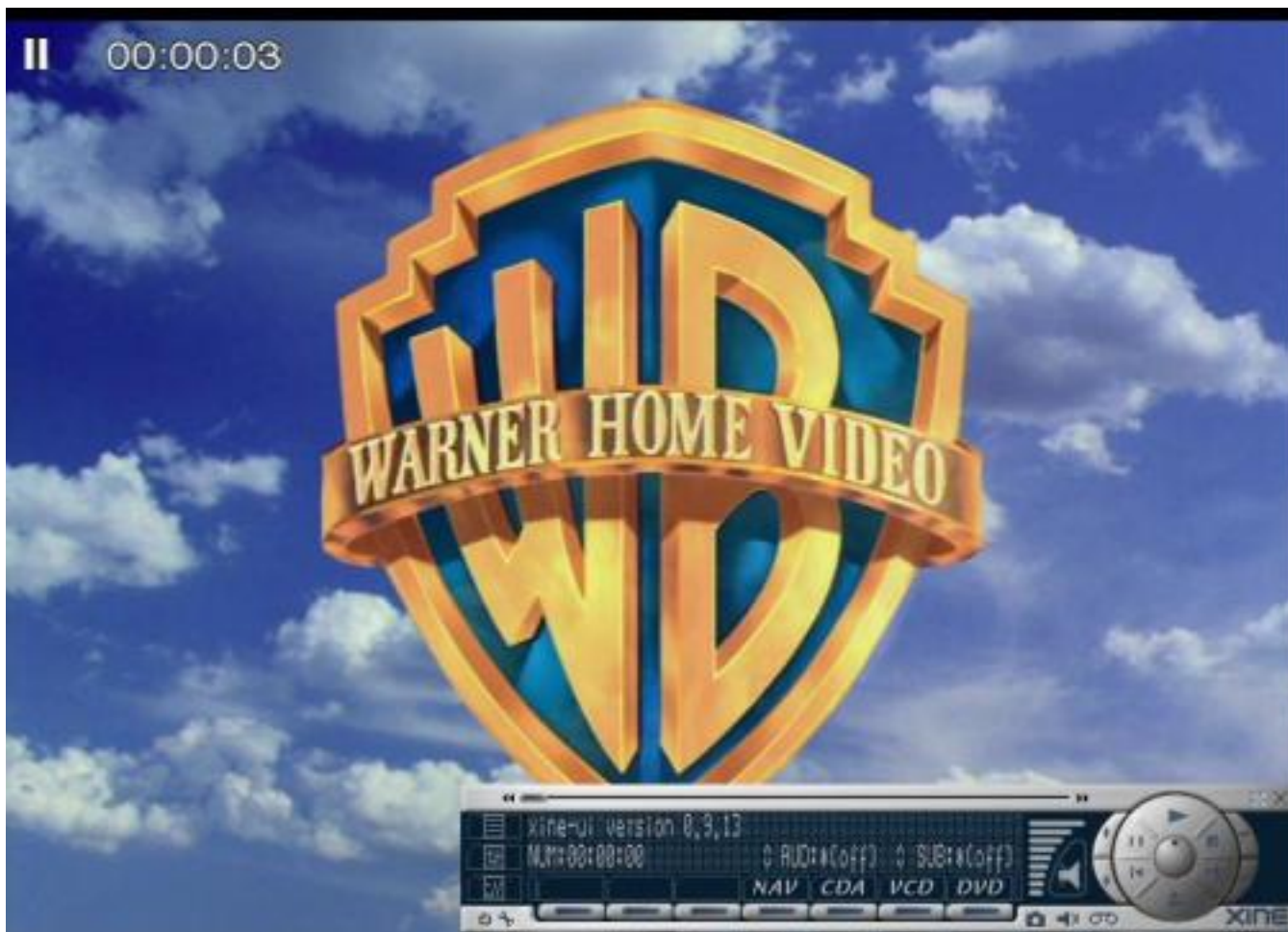
přičemž pouze Ogle je výhradně DVD přehrávač. Ostatní programy můžeme používat i k přehrávání jiných (video) formátů. Ať už se po přečtení mého hodnocení rozhodnete vyzkoušet kterýkoliv přehrávač, doporučuji používat pro výstup videa Xv a pro audio OSS emulaci ALSA – ušetříte si tím zklamání z nekvalitní reprodukce obojího.

Vzhledem k tomu, že používám Debian, který má mezi svými balíčky Xine a Ogle (MPlayer a VLC jsou také dostupné ve zkompilevané podobě), mám situaci ulehčenou, nicméně v zájmu větší objektivnosti jsem si stáhl zdrojáky všech zmiňovaných programů a zkompileval je sám. To vyžaduje trošku manévrování, protože například MPlayer doporučuje použití vlastních opatchovaných verzí knihoven `libdvcss` a `libdvread` (první DVD dešifruje, druhá čte obsah).

Xine

Kompilace a instalace Xine (<http://xine.sourceforge.net/>) není nijak neobvyklá (nehledě na to, že většina distribucí poskytuje binární balíčky) a máte-li správně nastavené locales, bude na vás Xine mluvit rovnou česky. Pouze je nutno doinstalovat plugin, který dokáže rozlousknout CSS.

Těch je více, ale nejprověřenější jsou pravděpodobně `captain_css` (je k dispozici ve dvou variantách, z nichž pouze jedna dokáže načíst DVD menu) a `dvdnav`, který menu podporuje také. Pro jejich zdárné fungování (a instalaci) je třeba mít nainstalovány již zmiňované knihovny `libdvdread` a `libdvdcss` (opět součástí většiny distribucí). Oba pluginy jsou založeny na kódu programu Ogle, který vůbec první přišel s podporou menu pod Linuxem.



Přestože jsem `captain_css`, respektive jeho děd `odnož` (s podporou menu) dříve používal, tentokrát se mi rozchodit nepodařil. `Dvdnav` naopak fungoval dobře, ale nedokázal jsem jej přinutit zobrazovat správný poměr stran, pokud nebylo okno videa maximalizováno. Nepovažuji to za moc velkou vadu na kráse, horší už je to, že zvolit titulky a jazyk lze pouze prostřednictvím menu DVD, nikoliv během přehrávání pomocí nějakého menu programu. Naopak velkou výhodou u Xine je jeho schopnost ukládat "snapshoty", a to i z DVD. Co se kvality obrazu a synchronizace se zvukem týče, není Xine co vytknout.

MPlayer

Kromě toho, že zvládá DVD, přehrává MPlayer (<http://www.mplayerhq.hu/homepage/>) snad úplně všechno, nač si vzpomenete. Poslední buildy z CVS by si měly poradit i s SVQ3 + QDM2 MOV (Quicktime 5 a výše).

Instalace MPlayeru je výjimečná pouze tím, že chcete-li používat GUI (narozdíl od Xine má MPlayer standardně pouze command line rozhraní), musíte to říct `./configure` skriptu: `--enable-gui`. Pak je ještě třeba v konfiguračním souboru (`/etc/mplayer.conf`) nastavit `gui=yes` a v neposlední řadě si nějaký ten skin opatřit, jelikož součástí MPlayeru není ani jeden. Elegantnějším řešením výběru GUI nebo CL rozhraní je používání automaticky nainstalovaného symlinku `gmplayer` pro práci s GUI.

MPlayer sice přehrává i šifrované disky "out-of-the-box", ale podpora DVD menu mu (prozatím) chybí. Existuje sice možnost spustit program s experimentálním přepínačem `-dvdnav`, který má umožnit využití knihovny `libdvdnav`, ale moje zkušenost je taková, že MPlayer spadne asi v 70% případů, což není zrovna

použitelné. Tento nedostatek je však kompenzován pohodlným menu, ve kterém lze zvolit jednotlivé tituly nebo kapitoly na disku a měnit titulky i audio (bohužel však ne za pochodu – stačí změnit jazyk titulků a přehrávání začne znovu na počátku daného titulu).



Menu MPlayeru také zpřístupňuje bohaté konfigurační možnosti (na příkazové řádce lze u tohoto programu určit, i kdy vám zazvoní u postele budík :-). Pokud budete chtít experimentovat s různými výstupními ovladači videa a audia, zkuste použít postupně příkazy **mplayer -vo help** nebo **mplayer -ao help** – to vypíše ty, které je MPlayer schopen na vašem systému využít. Běžné použití bez grafického prostředí může vypadat třeba takto (viz **mplayer --help** nebo podrobné **man mplayer**):

```
mplayer -ao oss -vo xv -dvd 2 -fs -framedrop
```

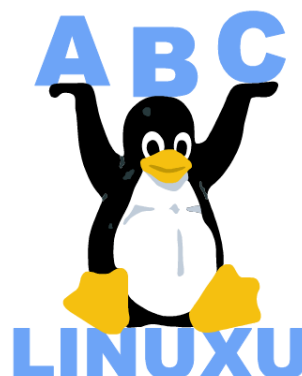
MPlayer má jeden nepříjemný zlovyk a to je dožadování se přijatelnější konfigurace systému. Může se vám proto stát, že se budete chvíli bez problému koukat na film a najednou vyskočí uprostřed video-okna dialogový box, který vám oznámí, že váš počítač je příliš pomalý, než aby soubor (DVD) mohl přehrát. Naštěstí hned také nabídne několik možných řešení, kterými by se dala tato nepříjemnost napravit, přičemž nejčastěji postačí – jak již jsem upozorňoval – buď přepnout výstup na Xv nebo nainstalovat místo původního OSS ovladače ALSA (a použít OSS emulaci).

Pravděpodobně posledním problémem, který by vám mohl znepříjemňovat život, je neschopnost (občasná – zatím netuším, proč tomu tak je pouze u některých disků) MPlayeru navazovat bez přerušení tituly na disku. Stane se tedy, že zvolíte „Přehrát DVD“ a jediné co proběhne je bliknutí černého okna a MPlayer se tváří, že už je hotovo. V takovém případě musíte v menu zvolit titul na disku, který obsahuje samotný film, což nemusí být někdy snadné, zvláště je-li disk rozvržen do nějakých 30 titulů a pouze jediný skrývá film – ostatní mohou být různé části menu, copyrightová upozornění, znělky studií apod.

MPlayer je při načítání disku o dost svižnější než Xine, kterému dekódování CSS chvíli trvá; přehrávání je plynulé, i když naopak proti Xine si všimnete trhnutí obrazu při přechodu do/z celoobrazovkového režimu.

AbcLinuxu s.r.o. nabízí tyto služby:

- Instalace
 - firewall
 - DNS
 - proxy
 - e-mail
 - samba
 - web
 - databáze
 - ftp
 - Account server
- Virtuální privátní síť (VPN)
- Zabezpečení spojení IPSEC
- Návrh a nasazení systému zálohování
- Analýza a rekonstrukce poškozených systémů
- Celopodniková řešení na bázi Linuxu a Open Source
- Odborné konzultace v oblasti Open Source a Linuxu
- AbcPalírna – pálené distribuce v centru Prahy
- Prodej reklamy na serveru AbcLinuxu.cz (35 tisíc návštěvníků měsíčně)

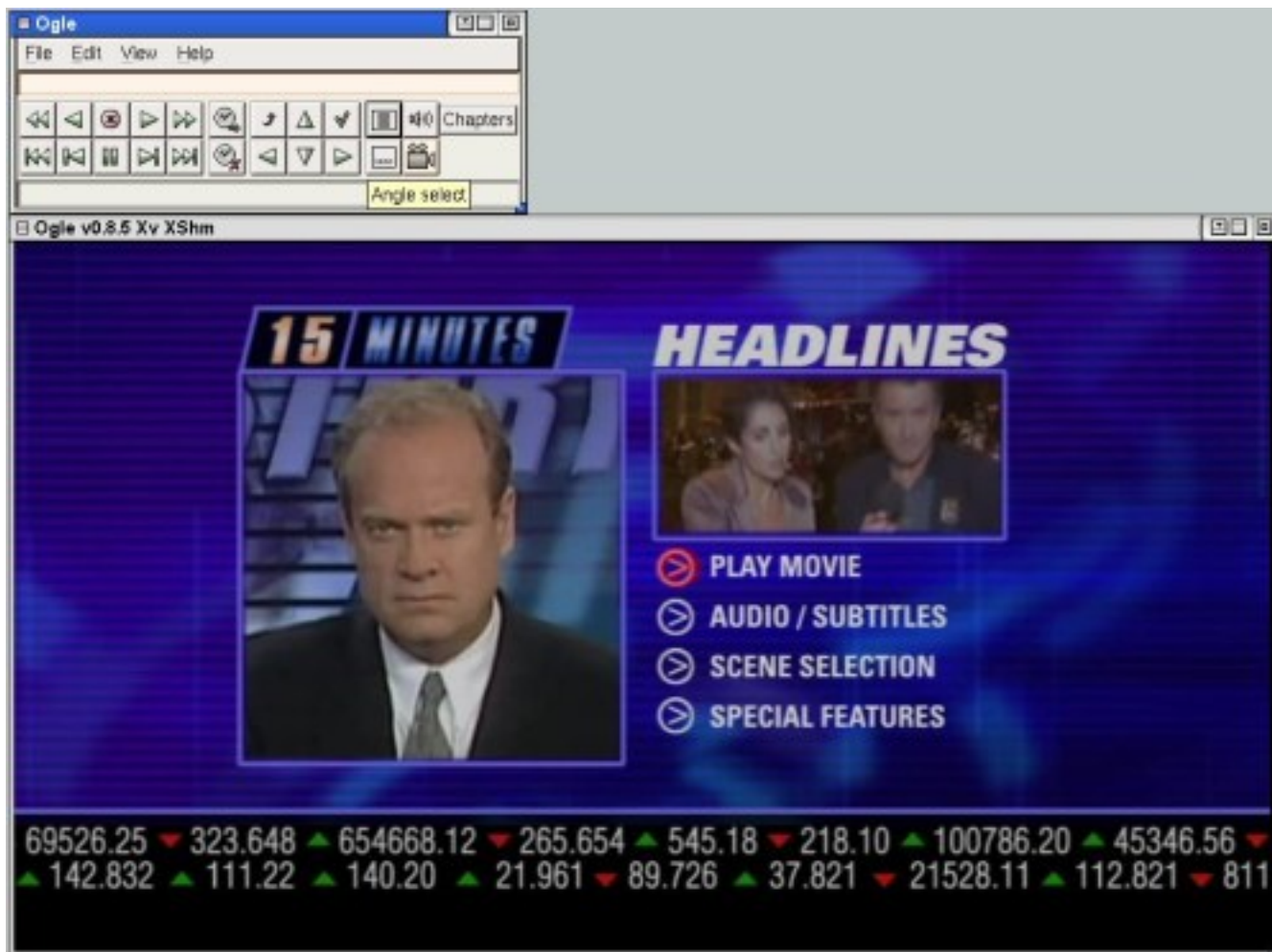


Kontakt:

e-mail: *info@abclinuxu.cz*
telefon: 721 300 742
adresa: Křemencova 10, Praha 1

Ogle

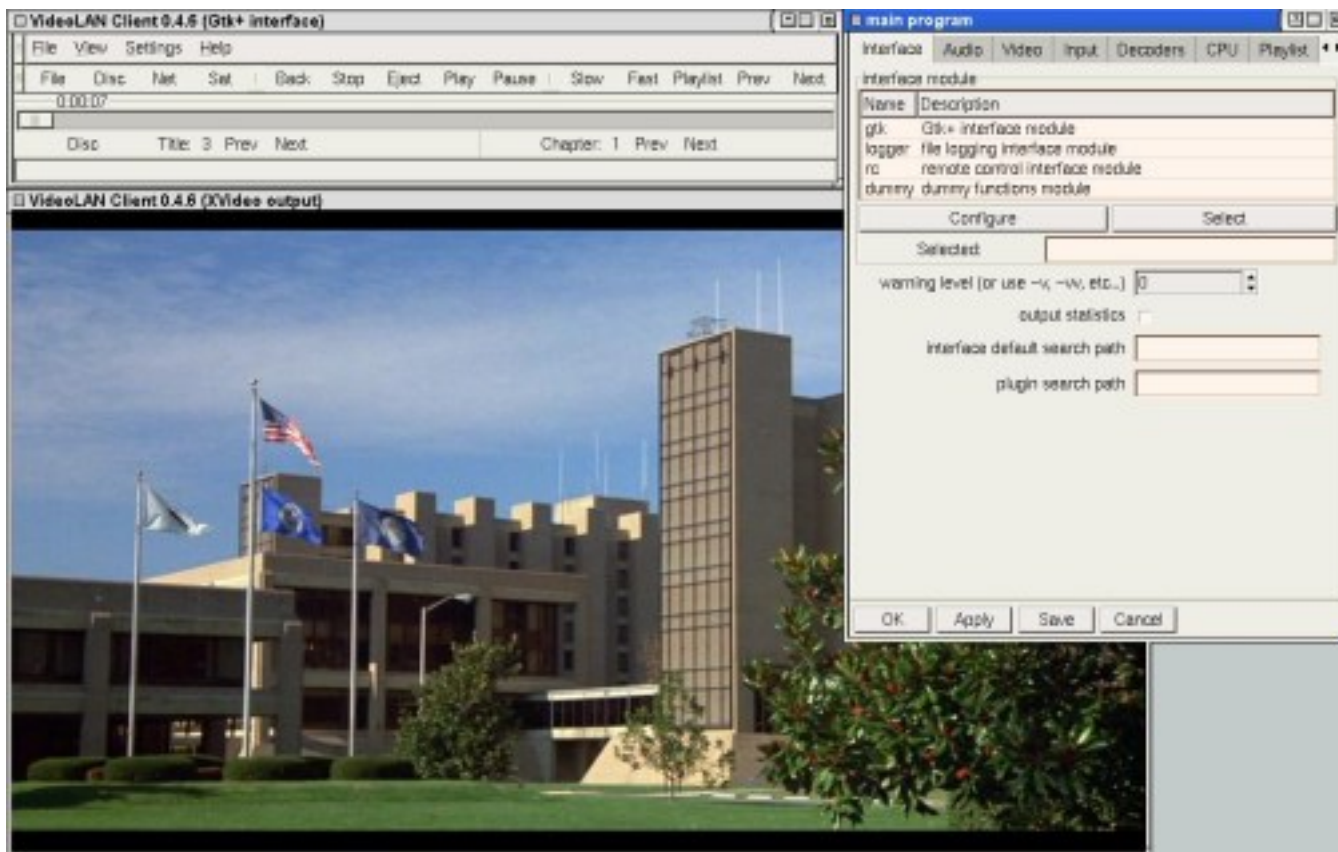
Ogle (<http://www.dtek.chalmers.se/groups/dvd/index.shtml>) opět nepředstavuje žádný tvrdý instalační oříšek. Předkompilovaný balíček bych však být vámi v tomto případě preferoval, protože ke kompilaci GUI je třeba splnit hromadu ne úplně běžných závislostí. Každopádně však instalaci GUI doporučuji. Ze všech popisovaných přehrávačů je řešené nejrozumněji – strážlivě a s ohledem na funkčnost. GUI s vámi sice nebude mluvit česky, ale je tak prosté a bez možnosti jakéhokoliv nastavení, takže na něm téměř není co překládat. Pro uživatele KDE se nabízí i možnost využít Okle, což je sympatický frontend.



Ogle kombinuje dobré vlastnosti jak Xine, tak MPlayeru a přitom jej netrápí žádná z jejich chybiček, o kterých jsem psal. Ogle sice nelze použít k přehrávání ničeho jiného než DVD, ale zato splňuje vše, co byste mohli očekávat. Přepínání zvuku i titulků je možné za běhu filmu, máme k dispozici pohodlná tlačítka pro zrychlené i zpomalené přehrávání a také nabídku s výběrem dostupných DVD menu (root, výběr kapitol, nastavení zvuku atd.). Načtení disku je stejně rychlé jako u MPlayeru. Jedinou vadou na kráse je snad nedostupnost ovládacích prvků ve fullscreen režimu (samozřejmě lze však ovládat klávesovými zkratkami, viz **man oglerc**).

VideoLAN Client

VLC (<http://www.videolan.org/vlc/>) je součástí většího balíku aplikací VideoLAN, který představuje ambiciózní řešení streaming videa po LAN (nyní již i WAN). Nás zajímá pouze Client, který může fungovat jako samostatná aplikace pro přehrávání videa. Při instalaci máme možnost zvolit si GUI dle svého vkusu a potřeb (Gnome, Qt2, gtk) – balíčky pro hlavní distribuce jsou opět k mání. Češtiny se znovu nedočkáme, což je trochu škoda, protože možnosti konfigurace pomocí GUI jsou skutečně bohaté. VLC se vyrovná Ogle ve všem kromě podpory DVD menu.



Závěr

Pokud bych bral v potaz čistě přehrávání DVD, pak je u mne Ogle mírným favoritem před VLC. MPlayer sice neztracuji, ale používám ho na všechno ostatní kromě DVD. Xine se mi zdá trošku moc nestabilní a neovládá se také zrovna pohodlně, i když to je určitě individuální.

Příště se můžeme pustit třeba do návodu, kterak vychutnat prostorový zvuk (máme-li na to hardware) a nebo rovnou do vypalování DVD videa pod Linuxem.

Diskuse

- Michal Čihař:** Já nevím, ale můj Xine má tlačítka na přepínání titulků i audio stop, takže stačí kliknout a už se to změní. Možná to záleží na skinu...
- Robert Krátký:** No jo... fakt. Díky. Vůbec mě nenapadlo, že by jiný než defaultní skin mohl mít nějaké ovladačské prvky navíc. To mě docela potěšilo – dost tuhle funkci využívám a její (domnělá) nepřítomnost v xine mi lezla na nervy. Stačilo přečíst si pořádně manuál...
- Pavouk:** Xine umožňuje bez problémů přepínat audio i titulky při přehrávání. Také má u mne nejplynulejší obraz. Na DVD se mi osvědčil a nemám žádné problémy se stabilitou. Mplayer má vyborné ovládání a skvělé titulky. Používám ho na přehrávání MPEG, DIVX apod.
- d.f.h.:** Co se týče regionu – mplayer, resp. jeho libmpdvdkit2, na regiony z vysoka kašle :-). Vyzkoušeno, mám NERegionfree mechaniku nec 5800a, při přehrávání r4 disku si xine ani neškrtně (powerdvd pod windows taky ne, jenom nabídne možnost dočasného přepnutí regionu mechaniky, což je ale samozřejmě omezené asi na čtyřikrát, pak už s tím nehnete), mplayer v pohodě přehrává a na nic se neptá. Je to asi věc libdvdcss.
- Petr Stehlik:** Xine samozřejmě taky přehrává všechny možné regiony i na neregion free mechanice – je to totiž vlastnost toho css pluginu/knihovny.

Moderní souborové systémy – ext3

Petr Novický

Úvod a motivace

V dnešním díle seriálu o souborových systémech v Linuxu se budeme zabývat filesystémem Ext3. Jedná se o filesystém s řadou výhod, ale také nedostatků. O nich bude právě řeč. Nejprve něco z jeho historie. Ext3 byl napsán Dr. Stephem Twediem jako nadstavba standardního linuxového souborového systému Linuxu (Ext2). Filesystém Ext2 je velmi rozšířený a oblíbený. Ačkoliv nemá podporu malých souborů jako ReiserFS nebo rychlost přístupu XFS, zůstává nejrozšířenější na linuxových platformách.

V dnešní době jsou některé Ext2 filesystémy opravdu velké, u nich pak trvá kontrola integrity dat v případě výpadku velmi dlouhou dobu (i několik hodin). Během doby, kdy se souborový systém kontroluje, jsou aplikace na počítači mimo provoz. Hlavním důvodem vzniku Ext3 je co nejvyšší dostupnost služeb i v případě výpadku. Nedisponuje ovšem vlastnostmi jako podpora extrémně rozsáhlých souborů, nebo mapování pomocí extents.

Jedním z nejdůležitějších cílů byla absolutní kompatibilita mezi Ext2 a Ext3 a to v obou směrech. Můžeme použít existující Ext2 filesystém, vytvořit žurnál a připojit ho jako Ext3 a dostáváme žurnálovací filesystém. Naopak pokud máme Ext3 filesystém a předtím došlo k bezpečnému odpojení tohoto souborového systému, můžeme ho připojit znovu jako Ext2 a vše funguje, jak má.

Jelikož Ext3 vychází z Ext2, má většinu jeho vlastností. Jako omezení velikosti souborového systému, velikosti souboru, velikosti bloků dat apod. Přestože některé informace se zapisují na Ext3 vícekrát než jednou, je Ext3 často rychlejší díky optimalizaci externí fragmentace.

Módy Ext3 filesystému

Pro optimalizování rychlosti a zajištění datové integrity můžeme zvolit jeden ze tří módů:

writeback

zajišťuje jen limitovanou schopnost zachování integrity dat a za určitých okolností vede ke zvýšení rychlosti operací.

ordered

je defaultním módem a garantuje konzistenci dat ve filesystému.

journal

spotřebovává více prostoru na žurnál a proto opravení souborového systému po výpadku trvá delší dobu. Někdy je však rychlejší pro určité databázové operace, NFS a synchronní operace mail serveru.

Writeback

Použitím writeback módu dostáváme nejrychlejší variantu Ext3 filesystému. Je tomu tak, protože uchovává pouze změny metadat a nečeká dále na asociované změny dat souboru předtím než změní např. velikost souboru. Protože změny dat souboru a žurnálované změny metadat filesystému jsou uskutečněny nezávisle, mohou být metadata v nekonzistentním stavu (např. obsahují ukazatele na bloky dat, na které ještě nebyla data zapsána). Použitím tohoto módu získáme pouze urychlení kontroly konzistence dat při havárii systému.

Ordered

V módu ordered uchováváme opět pouze metadata souborového systému kvůli snížení redundance mezi zápisem do filesystému a do žurnálu a je proto rychlejší. Změny metadat jsou uskutečněny, narozdíl od

módu writeback, až po změnách ve filesystému. Toto vede k lehkému snížení výkonu systému, naopak garantujeme, že při všech změnách budou metadata odpovídat datům, která jsme zapsali.

Journal

Použitím journal módu zapisujeme do žurnálu data i metadata. Každá operace se tedy musí uskutečnit dvakrát. Může proto snížit výkonnost systému, ale minimalizujeme šance ztráty nějakých dat.

Práce se souborovým systémem Ext3

Instalace

Nejprve potřebujeme zkompilovat kernel s podporou Ext3 filesystému. V jádrech řady 2.4 musíme zaškrtnout volbu `CONFIG_EXT3_FS`. Pokud používáte při konfiguraci `make menuconfig` nebo `make xconfig`, najdete tuto volbu v menu Filesystem, položka Ext3 journalling file system support. Podpora Ext3 filesystému se dá zkompilovat buď přímo do kernelu nebo jako modul. Pokud ovšem chcete používat Ext3 na kořenovém oddílu, doporučuji podporu Ext3 zakompilovat přímo do kernelu (viz dále).

Pokud máme existující Ext2 filesystém, můžeme ho lehce převést na Ext3 vytvořením žurnálu. Stačí použít následující příkaz:

```
tune2fs -j /dev/hdXX
```

kde `/dev/hdXX` je cílový diskový oddíl. K převedení dokonce nemusíme ani daný filesystém odpojovat! Důležité je také přepsání záznamu v `/etc/fstab`, kde stačí prostě přepsat `ext2` na `ext3`.

Pokud chceme vytvořit nový Ext3 filesystém, použijeme:

```
make2fs -j /dev/hdXX
```

kde `/dev/hdXX` je cílové zařízení.

Nastavení

Jelikož Ext3 vychází z Ext2, sdílí spolu vlastnost, že `e2fsck` kontroluje pravidelně filesystém i v případě, že je označen jako nepoškozený. Oddíl se kontroluje buď při každém dvacátém připojení filesystému, nebo jednou za 180 dní (toto jsou defaultní hodnoty, které lze pomocí příkazu `tune2fs` změnit dle libosti). Jelikož je tato kontrola v případě Ext3 zbytečná a pravděpodobně chceme přejít na Ext3 právě z důvodu vyhnutí se `fsck`, dá se tato vlastnost vypnout použitím příkazu:

```
tune2fs -i 0 -c 0 /dev/hdXX
```

Pokud je náš kořenový souborový systém typu Ext3 a používáme kernel s podporou Ext3, systém ho připojí právě jako Ext3. Defaultnímu chování však můžeme zabránit následujícím parametrem bootladeru LILO:

```
linux rootfstype=ext2
```

stejně tak se dají nastavit parametry při připojování filesystému (jako žurnálovací mód):

```
linux rootflags=data=journal
```

Ext3 jako modul

Pokud máme podporu filesystému Ext3 zakompilovanou pouze jako modul, je připojení kořenového adresáře poněkud těžším problémem. Nejprve musíme vytvořit ramdisk, který bude obsahovat moduly zkompilevané s kernelem. Je nutné zjistit přesnou verzi kernelu, který právě používáme. K tomu slouží příkaz:

```
uname -a
```

Dostaneme podobný výpis:

```
Linux host 2.4.19 #2 Sat Oct ...
```

Pro samotné vytvoření ramdisku použijeme následující příkaz (vytváří ramdisk pouze s moduly potřebnými pro žurnálování):

```
mkinitrd --preload jbd --preload ext3 /boot/initrd-2.4.19.img 2.4.19
```

Příkaz **mkinitrd** získáme po nainstalování například balíčku **initrd-tools** v Debianu. Nyní stačí již jen upravit konfigurační soubor bootloaderu LILO, kde přidáme řádek pod řádek **image=...**, který říká systému, že má daný ramdisk načíst (viz manuálové stránky lilo.conf).

```
initrd=/boot/initrd-2.4.19.img
```

Poté můžeme systém restartovat a zkontrolovat výpis souboru `/proc/mounts`, jestli opravdu používáme ext3 filesystém. Jelikož toto je poměrně zdlouhavý proces, doporučuji zkompilevat kernel s přímou podporou Ext3.

Souhrn

Teď si ještě shrneme ostatní výhody souborového systému Ext3, o kterých jsem se zatím nezmínil. Mezi hlavní z nich patří podpora žurnálovacího souboru na odděleném diskovém oddílu. Zajímavá je také podpora ACL (Access Control Lists) a EA (Extended Attributes), správa vadných sektorů a quot. Pomocí externího patche v kernelu dovoluje měnit velikost připojeného filesystému (ovšem pouze zvyšovat). V případě nepřipojeného souborového systému můžeme velikostí hýbat oběma směry. Pro ty, co by se o tuto problematiku dále zajímali, doporučuji GNU Parted.

Kancelářský balík Hancom Office (1)

Robert Krátký

Úvod

Hned úvodem musím předeslat následující: *Hancom Office* není zadarmo. Je to sice sada aplikací určená výhradně pro Linux, ale zdarma je smíte používat jen prvních 30 dní od instalace (a navíc pouze trošku okleštěnou "Evaluation" (zkušební) verzi). Rozhodnete-li se tento programový balík koupit, musíte počítat minimálně s 50 USD (cena při stažení z internetu). Druhou nepříjemnou záležitostí je čeština. *Hancom Office* mluví pouze anglicky (a japonsky, čínsky, korejsky a arabsky...) a ani při psaní nemáte možnost zapnout českou kontrolu pravopisu (ispell není podporován). Psát česky samozřejmě lze. Pokud je vám jedna z těchto věcí proti srsti, nemá pravděpodobně cenu pokračovat ve čtení.

Instalace

Hancom Office (dále jen "HO") je dodáván pouze v binární (HO používá své Qt i FreeType knihovny) podobě a jeho instalaci řeší `install` skript, který najdete v hlavním adresáři po rozbalení `.tgz` archívu (případně v kořenovém adresáři CD-ROM). Tento archív je k mání po bezplatném zaregistrování na stránkách firmy Hancom (<http://en.hancom.com>) a to buď jako soubor o velikosti 61 MB nebo několik 14 MB souborů (pro stahování přes pomalejší připojení k internetu). Po spuštění skriptu vás instalací provede pohodlný grafický instalátor, který vás po úspěšném dokončení vyzve k restartování X (!). Za zmínku stojí stručný návod k instalaci obsažený v souboru `readme`. Instrukce (dostí lámanou angličtinou) nabízejí řešení v případě, že instalace selže. Doslova se tam říká, že na některých linuxových systémech nemusí instalační skript pracovat správně. Nápravu má zjednat provedení příkazů:

```
su root
mount /mnt/cdrom
cd /mnt/cdrom
./install
```

Nechci být přehnaně jízlivý, ale tohle rozhodně není nazývání věcí pravými jmény. Myslím, že i zcela nezkušený uživatel by si zasloužil vědět, že instalaci je prostě třeba provádět s určitými právy – a ne tvrdit něco o zvláštních podmínkách daného systému...

První dojmy

HO se sestává ze čtyř nosných aplikací a shellu pro jejich pohodlné spouštění. Máme tu textový editor *Hancom Word* ve verzi 6 (k dispozici i samostatně), tabulkový kalkulátor *Hancom Sheet 2* (nově také samostatný), nástroj na tvorbu prezentací *Hancom Presenter 2* a konečně kreslicí program *Hancom Painter 2*.



Používáte-li běžně OpenOffice.org nebo StarOffice, okamžitě si všimnete dramatického rozdílu v době potřebné pro spuštění kteréhokoliv z programů sady HO. Na mém počítači trvá první otevření OpenOffice.org Writeru přes 20 vteřin – *Hancom Word* naběhne za necelé čtyři sekundy. I při práci je odezva všech programů celkově velmi svižná. Rovněž paměťové nároky jsou, ve srovnání se zmiňovanými oblíbenými programy, téměř zanedbatelné. A když už jsme v tom, kolikrát vám zamrzl StarOffice? Mně mockrát. Teď používám OpenOffice.org, který pro změnu občas padá. Aplikace HO jsem však nenachytil. Snažil jsem se, seč jsem mohl, ale HO nezaváhal a shodit se mi jej nepodařilo (a to jsem prováděl psí kusy...).

Uživatelské prostředí se na první pohled nijak zvlášť neodlišuje od podobného typu kancelářských aplikací, nástrojové lišty obsahují (podle zaměření programu) všechny důležité ikony. Lišty s ikonami je možné dle libosti umístit ke všem okrajům oken, pořadí a skladbu jednotlivých ikoněk však ovlivnit nelze.

Menu jsou řazena opět podle stejného vzoru jako u jiných programových balíčků, takže žádnou oblíbenou funkci byste neměli dlouho hledat. Nemilé překvapení na vás ovšem čeká v nabídce Help. Najdete tam sice slušivý About dialog a odkaz na webovou stránku Hancom, ale to nejdůležitější, totiž nápověda, tam chybí. Je sice možné spustit docela elegantní prohlížeč nápovědy, ale najdete v něm pouze obrázek, který znáte z About. Na internetových stránkách je mezi výhodami pořízení plné verze programu i položka "Elektronická nebo tištěná uživatelská příručka", což tedy pravděpodobně znamená, že do zkušební verze nebyla nápověda schválně zařazena. To trošku zamrzí, protože kvalitu návodů považuji za docela důležitý faktor pro rozhodování, zda za program skutečně platit.

Další výtku musím směřovat k nepochopitelné nekonzistenci v ovládání jednotlivých programů sady. *Hancom Word* ukrývá možnost celkového nastavení v File -> Options (položka Edit -> Properties sice existuje, ale nedělá vůbec nic...), naproti tomu v *Hancom Sheet* je podobné nastavení ukryto pod Tools -> Options. Proč? Tak to také nevím – a nechce se mi věřit, že by si toho vývojáři prostě nevšimli.

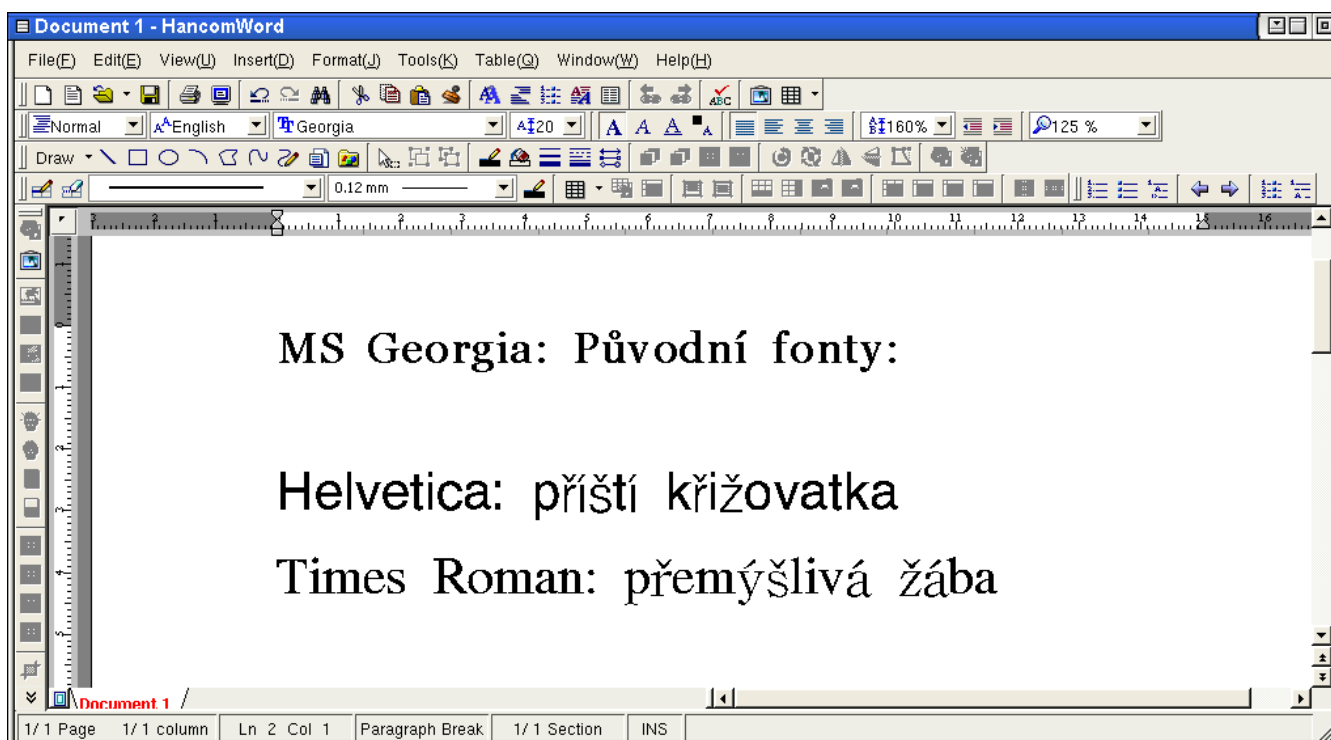
Z podobného soudku je i výběr a možnosti použití klávesových zkratk. *Hancom Word* vás možná bude chvilku mást tím, že namísto obvyklého podtržítka pro zvýraznění písmene, které je třeba stisknout pro rychlou volbu funkce (při přístupu do menu je to Alt + písmeno), nabízí toto písmeno kapitalizované v závorce za názvem funkce. Důvodem je, že ne vždy je písmeno použité pro klávesovou zkratku obsaženo v názvu funkce. Takový klávesový povel však opravdu nemůže být intuitivní a zapamatovatelný. Dejme tomu, že je to pozůstatek původního jazykového prostředí, pro které byl program napsán (korejština). Kdo mi však vysvětlí, proč ani tyto podivné klávesové kombinace nejsou jednotné ve všech aplikacích sady? Chci-li v *Hancom Word* vyvolat menu Insert(D) (Vložit), je tedy třeba stisknout **Alt+D**. V *Hancom Sheet* je to ale obvykle Insert, tudíž zkratka **Alt+I**. Je to maličko hra na schovávanou.

Poslední záležitost, na kterou jsem v této souvislosti narazil, je podivné chování dialogových oken – opět pouze v *Hancom Word*. Nejde totiž změnit jejich velikost. Ve Windows(tm) je to sice standardní chování, z X Window jsme však zvyklí na větší komfort. Při procházení adresářovou strukturou na disku v oknech typu Open jsme tedy limitováni devíti řádky, které se do implicitní velikosti okna vejdou.

Hancom Word 6

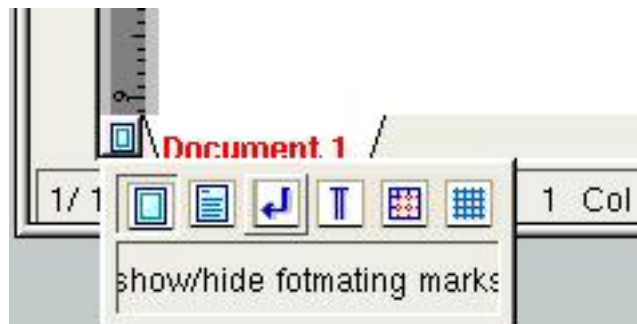
O některých zvláštěnostech uživatelského rozhraní této stěžejní aplikace jsem se již rozepsal. Sázím na to, že popisované odchylky od zbytku programů mají svůj původ v historii vzniku celé sady. Prvním, a dlouhou dobu jediným, programem totiž byl právě textový editor. Ostatní aplikace, které z balíku činí plnohodnotný kancelářský nástroj, byly doplněny až později. Autoři se teď snaží narvat nohu do rukávu. Neradi by změnili u *Wordu* ovládání, které si možná někteří uživatelé již stačili osvojit. Zároveň by však chtěli nově vytvářené programy přiblížit běžným standardům v této kategorii. Výsledkem je trošku hybrid.

Další důležité téma jsou fonty. *Hancom Office* sice disponuje vlastní základní sadou fontů (a po zaplacení dostanete další), jenže ty jsou pro zobrazení češtiny takřka k ničemu (viz *obrázek*). Jakmile však *Wordu* zpřístupníte fonty, které nemají podobné nedostatky, bude vše v pořádku. Proveďte se to vytvořením symlinků na vaše fonty do adresáře `/usr/local/HancomOffice2/font/`. Pouhé přidání cesty s vašimi fonty do souboru `.fontrc`, které je doporučeno v README, bohužel kýžený výsledek nepřinese.



Běžné funkce

Hancom Word nepostrádá žádnou z důležitých funkcí vyspělých editorů. Naopak doplňuje standardní schopnosti mnoha vítanými nápady. Preview (náhled) je prosté, ale velmi rychlé a přizpůsobivé. Nechybí snadné generování indexu a obsahu, přehledné styly odstavců, šablony dokumentů, možnost automatických oprav ani schopný kreslicí nástroj. Hodí se tzv. Multi-column break, tj. zalomení více sloupců najednou. Ocenil jsem zdánlivé maličkosti typu rozbalovací ikonky pro výběr druhu zobrazení stránek (formátovací značky, mřížka, okraje stránky, atd.) nebo možnost nastavit v procentech řádkování přímo na nástrojové liště. Jediné, co v tomto směru chybí, je absence možnosti vkládat do dokumentu pole s proměnnými.



Kompatibilita s MS Office

Ať již se nám to líbí nebo ne, kompatibilita s formáty MS Office je důležitá pro každodenní nasazení v běžném provozu. Můžete se třeba stokrát zařít, že nebudete používat těžkopádné, neúsporné a ještě ke všemu proprietární formáty fy. Microsoft, ale realita s vámi vyběhne. Dokud o tom nepřesvědčíte také své známé, klienty nebo zaměstnavatele, nezbyvá vám nic jiného než být připraven(a) na vše. *Hancom Office* o sobě tvrdí, že je v tomto směru správnou volbou. U každé z aplikací se proto podíváme, jak obstála v mém malém soukromém testu.

Pro textový editor jsem si stanovil tři jednoduchá kritéria použitelnosti. Pokud uspějí v načtení a uložení (a opětovném úspěšném načtení v microsoftích programech), budu editor považovat za dostatečně kompatibilní:

1. obyčejný text formátovaný do odstavců, běžné atributy (tučný, kurzíva), čísla stran
2. obrázek ukotvený na stránce, pevně stanovená velikost + text
3. jednoduchá tabulka v kombinaci s textem

Hned jsem zjistil, že lačka byla nasazena příliš vysoko. Přiblížím vám problémy při importu, se kterými jsem se setkal.

1. Běžný zformátovaný text nečinil *Wordu* většinou problémy. Trochu jej poplete přílišné kombinování formátovacích prvků obsažených ve stylu odstavce a těch, které jsou uplatněny přímo na text. Může se proto stát, že uvidíte nepodtržený nadpis a celý odstavec pod ním podtržen bude. Žádná velká tragédie, podobné přehmaty dělá občas MS Word také, pokud dokument otevřete v jiné verzi, než kterou jste použili pro jeho vytvoření.

Velmi mě překvapilo perfektní rozpoznání všech druhů seznamů. Jejich vlastnosti, včetně odsazení od okraje stránky a mezery mezi číslem (znaménkem) a položkou seznamu, interpretuje *Word* zcela správně (a lépe než OpenOffice.org Writer). Číslo stránek však bezchybná nejsou. Po pravdě řečeno, nejsou vůbec. Nedokázal jsem *Word* přimět ani jedinkrát k zobrazení jakéhokoliv čísla stránky v importovaném souboru. Nejdále, kam jsem se dostal, bylo ukázání celkového počtu stránek (je-li nastaveno zobrazení strana X z Y. Navíc, a to je ještě horší, *Word* nerozpozná tvrdý konec stránky. Výsledkem může být soubor, jehož délka se vám smrskne z 50 stránek na 10 a zkuste se v něm pak vyznat.

2. Také tento bod se ukázal být nad síly *Wordu*. Logo na začátku stránky ještě zvládl bezchybně, avšak dva obrázky vedle sebe už ne. Zkusil jsem dva obrázky pod sebou a mezi nimi text: obstojný výsledek. Jenže stačí text strčit do tabulky, nastavit obtékání a nestačíte se divit. Několikrát jsem dokonce skončil s popsaným obrázkem, zatímco tabulka byla na svém místě, ale prázdná. Zapomeňte na obrázkové katalogy, distribuované ve formátu MS Word.
3. Tabulky mi mírně spravily náladu. Ukázalo se, že většinou je import v pořádku. Narazil jsem pouze na dva případy, u kterých Word záhadně pozměnil šířku jednotlivých sloupců, ale nebyl jsem s to vysledovat příčinu. Trable nastanou pouze ve zmiňovaném případě „náročné“ kombinace tabulky s jinými objekty, ale také pokud tabulka přesahuje na další stránku – to se pak dějí nepředvídatelné věci.

Schválně jsem se zatím nezabýval exportem zpět do microsoftího prostředí. Nechal jsem si to jako poznámku na konec, protože podle toho, čeho jsem byl sám svědkem, si není nač stěžovat. Žádný z nešvarů vpravdě mizerného importu se exportu netýká. Je to však jen jedna (podle mého názoru ta méně důležitá) strana mince.

Tisk

Nejdřív ta špatná zpráva. V dialogovém okně Print (viz *obrázek*) je sice rolovací menu, které na první pohled vypadá, že nabídne dostupné tiskárny, ale k dispozici je pouze standardní **lp**. Chcete-li určit, která tiskárna bude pro tisk použita, nebo třeba v jakém rozlišení se bude tisknout, z programu to provést nelze. Na druhou stranu se nabízí několik velice šikovných možností nastavení. Především je to funkce automaticky vtěsnat až šestnáct stránek na jeden papír. Můžeme také tisknout odzadu, určit v procentech velikost tištěné stránky nebo nastavit pauzu po daném počtu vytištěných stran.

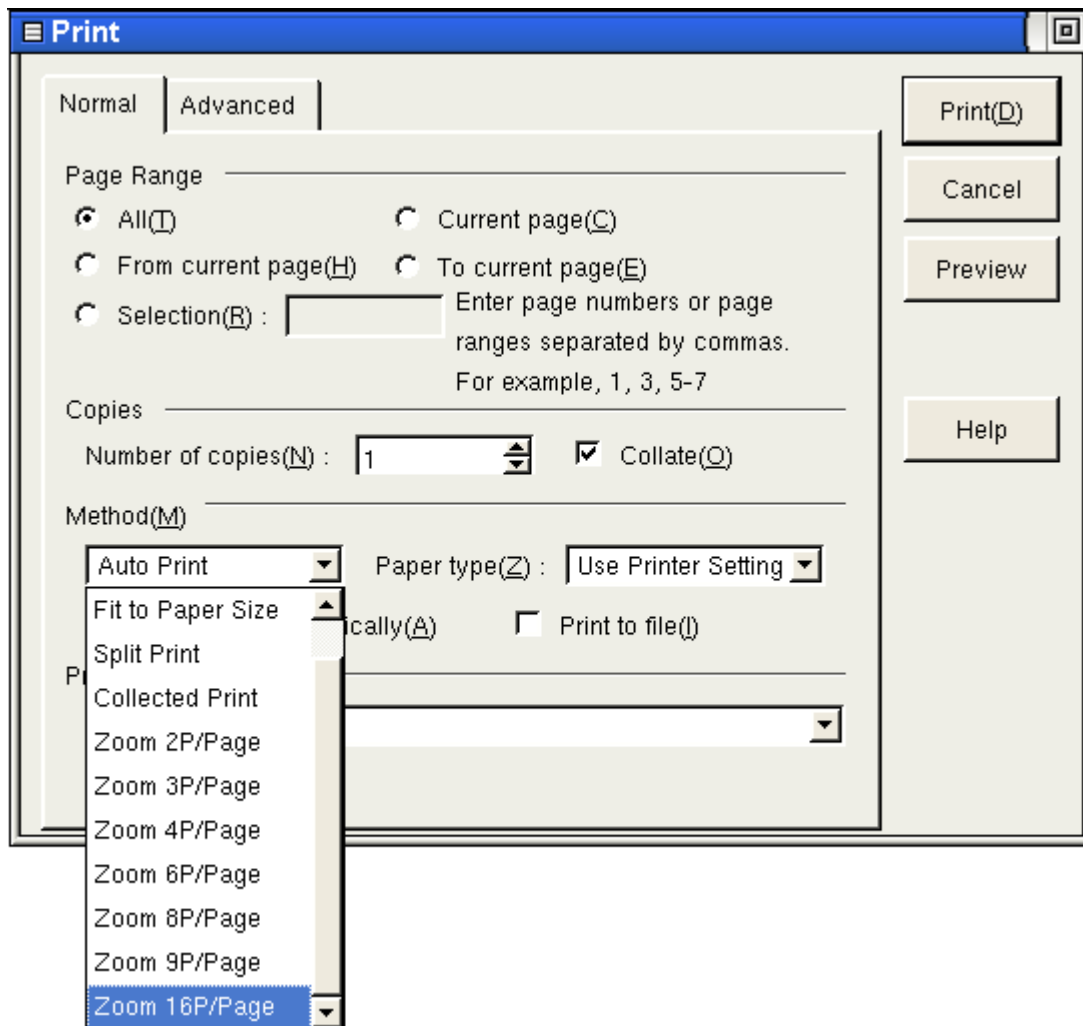
Kontrola pravopisu

Jen krátce zmíním kvalitní kontrolu pravopisu, kterou je *Word* vybaven. Čeština sice chybí, ale jinak je průběh pravopisné kontroly nejen rychlý, ale i dostatečně komfortní. Kapku mě zarazilo, že slovník spellcheckeru nezná výrazy „Hancm“ a „Linux“, ale což. Nestojíte-li o to, aby vás spellchecker informoval červenými vlnovkami o své marné snaze rozpoznat česká slova, budete muset vypnout automatickou kontrolu.

Hodnocení

Hancm Word je sympaticky vypadající editor, kterému nechybí žádná podstatná vlastnost, jež by správná aplikace tohoto druhu neměla postrádat. Nedostatky, o kterých jsem mluvil, jsou chvílemi až zarážející, nicméně v závislosti na druhu konečného nasazení/uplatnění nemusejí představovat zas tak vážný problém.

Obecně hodnotím Word jako velmi silný (a rychlý) nástroj, pomocí kterého lze snadno vytvářet i náročnější dokumenty.



Elektronické bankovníctví – eBanka

Leoš Literák


Úvod

eBanka patří k průkopníkům elektronického bankovníctví v naší republice. Jako první spustila přímé bankovníctví a dlouho neměla konkurenci. Snad i kvůli pověstnému kladnému přístupu k zákazníkům už několik let funguje pod Linuxem. Nejdříve pod Netscapem a zhruba od milníku 11 i pod Mozillou.

V tomto článku nenajdete rozhovor s představiteli banky. Tisková mluvčí paní Petroncová nejdříve požádala o zaslání otázek, následně však odmítla na ně odpovědět. Na další emaily již nereagovala.

Funkce

Z hlediska funkčnosti patří eBanka stále ke špičce. Přes internet můžete kompletně ovládat svůj účet. Pro většinu klientů může být založení účtu poslední návštěvou pobočky banky. Z pohodlí domova (či práce) můžete zadávat platby, prohlížet historii pohybů na účtu nebo si třeba opětovně vytisknout výpis z účtu.

Z ÚČTU ČÍSLO	<input type="text" value="960726001"/>	V MĚNĚ	<input type="text" value="CZK"/>
PŘEVÉST NA ÚČET	<input type="text" value="152903575"/>	KÓD BANKY	<input type="text" value="0300"/> 
ČÁSTKU	<input type="text" value="840"/>	V MĚNĚ	<input type="text" value="CZK"/> 
VARIABILNÍ SYMBOL	<input type="text" value="2261448633"/>		
KONSTANTNÍ SYMBOL	<input type="text" value="558"/> 	SPECIFICKÝ SYMBOL	<input type="text"/>
PROVÉST DNE	<input type="text" value="11.11.2002"/>	UKONČENÍ PLATNOSTI	<input type="text" value="12.11.2002"/>
ZPRÁVA PRO PŘÍJEMCE	<input type="text" value="AbcLinuxu.cz"/>		
ZPRÁVA PRO MNE	<input type="text" value="ročni poplatek za domenu AbcLinuxu.cz"/>		
ÚČEL	<input type="text" value="Nespecifikováno"/> 		
CERTIFIKAČNÍ KÓD	<input type="text"/>	<input type="button" value="Certifikuj"/>	
SPECIÁLNÍ NASTAVENÍ	<input type="checkbox"/>		
<input type="button" value="OK a nový"/>		<input type="button" value="OK"/>	<input type="button" value="Použij vzor"/>

Ke svému účtu se dostanete po zadání autentifikačního čísla. To vám vygeneruje buď elektronický klíč v podobě kalkulačky nebo jej zašle bankovní aplikace v podobě šifrované sms na váš mobil. Druhá varianta je zdarma, za první je nutné platit relativně vysoký měsíční pronájem (89 Kč). Mobilní klíč původně fungoval jen v síti Paegas, o několik let později se přidal i Eurotel a dnes je možné jej používat i v síti Českého Mobilu.

Po přihlášení máte k dispozici výběr z následujícího menu:

- Hlavní stránka
- Historie účtu
- Platby, konverze
- Inkasa
- Termínované vklady
- Platební karty
- Charge karty
- Osobní úvěr
- Osobní úvěrová linka
- Informuj mě
- Nastavení
- Nápověda
- Kontakty
- WWW stránky
- Odhlášení

Jak sami vidíte, eBanka obsahuje obrovskou spoustu funkcí. Přes web můžete nastavit prakticky cokoli a pobočku musíte navštívit jen kvůli podpisu smlouvy. Ovládání je velice pohodlné a zvyknete si na něj velice rychle. Pozitivně hodnotím možnost nastavování filtrů na pohyby na účtu či zasílání sms při splnění vámi zadané podmínky (příchozí platba vyšší než určitá částka, použití platební karty, pokles disponibilního zůstatku pod stanovenou mez, apod.). Na druhou stranu je tato služba zpoplatněna (v základním programu 3,90 Kč) a tak se budete muset rozhodnout, které informace mají pro vás tuto cenu.

Plusy a mínusy

Relativně příjemnou funkcí je i Inteligentní revolvingový termínovaný vklad, který dokáže udržovat na vašem mizerně úročeném účtě stabilní zůstatek (úrok na termínovaném vkladě není o moc lepší). Slovo Inteligentní v jeho názvu je však přehnané, jedná se o obyčejný rozhodovací proces založený na porovnání zůstatku s prahem. Pokud je v den otočky zůstatek větší, rozdíl se uloží na termínovaný vklad, v opačném případě se vyberou prostředky z termínovaného vkladu. Když mluvím s despektem o jeho inteligenci, mám na mysli to, že algoritmus vůbec nebere v úvahu inkasa či trvalé platební příkazy. Takže prah musíte nastavit dostatečně vysoko, jinak se po provedení těchto pravidelných plateb dostanete do mínusu. Podobnou službu dnes poskytuje většina bank, proto jsem (marně) čekal něco navíc.

Pro nás uživatele Linuxu je důležité, že si můžete vybrat v nastavení mezi menu v JavaScriptu a v Javě. První jmenované je naštěstí naprogramováno tak, že funguje i na jiných prohlížečích, než je Internet Explorer. To však neznamená, že můžete použít libovolný prohlížeč. Například Konqueror spadne ještě před přihlášením.

Mgr. Leoš Literák CZK

DISP. ZŮSTATEK: CZK
ČÍSLO ÚČTU:

OSOBNÍ ÚČET | HYPOTÉKA | STAVEBNÍ SPOŘENÍ | PENZIJNÍ PŘIPOJIŠTĚNÍ | ŽIVOTNÍ POJIŠTĚNÍ | PODÍLOVÉ FONDY

NASTAVENÍ UŽIVATELE

JAZYK	Čeština
FORMÁT DATA	den.měsíc.rok
FORMÁT ČÍSLA	# ##0,00
POTVRZENÍ POŽADAVKU	<input checked="" type="checkbox"/>
POUŽIT TYP MENU	Java Script
DEBETNÍ POHYBY ODLIŠOVAT	Červeným písmem
PŘEDNASTAVENÝ ÚČET	
OSLOVENÍ	Vážený pane Literáku <input type="button" value="Osobní"/>
PŘEHLEDY LISTOVAT	po 50
ZOBRAZOVAT DISPONIBILNÍ ZŮSTATEK	<input checked="" type="checkbox"/>
INFORMOVAT O NEREALIZACI TRANSAKCE	<input checked="" type="checkbox"/>
UPŘEDNOSTNIT ZASÍLÁNÍ ŠIFROVANÝCH SMS ZPRÁV	<input checked="" type="checkbox"/>
ZOBRAZOVAT PŘI PŘIHLÁŠENÍ NOVÉ ZPRÁVY VE VNITŘNÍ SCHRÁNCE	<input checked="" type="checkbox"/>
POVOLIT KONTEXTOVOU NÁPOVĚDU	<input checked="" type="checkbox"/>

27.11.2002 14:38:05

Ale ne vše je na eBance tak pozitivní, jako funkčnost jejího internetového rozhraní. Po zakoupení eBanky Českou pojišťovnou následoval prudký nárůst poplatků, takže dnes je podle testu MF Dnes eBanka druhou nejdražší bankou vůbec. Mnozí zákazníci, zvláště ti dlouholetí, si stěžují na zhoršení komunikace s bankovními úředníky. Viz některé diskuse na našem serveru. Změny k horšímu nastaly i v oblasti rychlosti načítání stránek. Není se co divit, když místo standardního popisu vzhledu skrze kaskádovité styly je použita takhle nechutně dlouhá definice:

```
<td VALIGN="TOP" BGCOLOR="#FFFFFF">  
<font face="Arial CE,Helvetica CE,Arial,Helvetica" size="-2" color="#000000">
```

Pro ilustraci: výpis posledních padesáti pohybů zabíral přes 70 KB. Odstraněním tagu FONT velikost dat klesla na polovinu. Dalším zjednodušením tagu TD jsem se dostal na 22 KB. Nechápu, proč eBanka nepodporuje kaskádové styly. Proč mám stahovat tolik duplicitních údajů? Původní implementace používala JavaScript obsahující tabulku s údaji, ze kterého se vygenerovala tabulka pohybů. Mimo úsporu při přenosu dat měla tato implementace ještě jednu výhodu: data se dala zkopírovat ze zdroje HTML dokumentu a pak uložit jako CSV soubor, který šel otevřít v libovolném tabulkovém procesoru.

Závěr

Co říci závěrem? eBanka má nejlepší elektronické bankovníctví, které po dlouhou dobu jako jediné fungovalo pod Linuxem. Na druhou stranu patří k nejdražším bankám v naší republice.

960726001: AbcLinuxu s.r.o. CZK DISP. ZŮSTATEK: 960726001
ČÍSLO ÚČTU: 960726001

FIREMNI ÚČET FIREMNI POJIŠTĚNÍ TRADE FINANCE

POHYBY NA ÚČTU

VÝBĚR Rok 2002 Upravit výběr

POŘ. ČÍSLO	DATUM ČAS	POZNÁMKA NÁZEV ÚČTU ČÍSLO ÚČTU	DATUM ODEPSÁNÍ VALUTA TYP	VARIABILNÍ SYMBOL KONSTANTNÍ SYMBOL SPECIFICKÝ SYMBOL	ČÁSTKA	POPLATEK SMĚNA ZPRÁVA
1	11.11.2002 09:29	roční poplatek za domenu AbcLinuxu.cz 152903575/0300	11.11.2002 11.11.2002 Převod	2261448633 558	-840,00	-6,90
2	07.11.2002 14:09	pronajem BBMedia za říjen Internet Billboard 224657001/2400	07.11.2002 07.11.2002 Převod	2751020029 0308	-678,40	-4,90
3	06.11.2002 23:59	Poplatek za generování výpisů 960726001	06.11.2002 01.11.2002 Generování bankovních výpisů	898		-3,90
4	02.11.2002 23:59	Poplatky 150148/2400	02.11.2002 02.11.2002 Distribuce bankovního výpisu	898		-20,00
5	31.10.2002 23:59	Úrok 10/2002 AbcLinuxu s.r.o. 960726001/2400	31.10.2002 31.10.2002 Kladný úrok	598	12,65	
6	31.10.2002 23:59	Poplatek za služby účtu 960726001	31.10.2002	898		-144,00

11.11.2002 09:30:51

Diskuse

- Libor Holub:** Ještě bych dodal, že je možné přistupovat pomocí Internetového Ek. Tato možnost stojí tuším 300 Kč/rok.
- Led Astray:** Ceny rostou, komunikace s bankou vážne. Ta se v dnešní době chová ke svým klientům už poměrně drsně a nepřátelsky. Nicméně reklamace řeší promptně. Po zrušení diskuse přímo v ebance můžete diskutovat v mail konferenci ebanka@pandora.cz.
- Petr Krčmář:** Je pravda, že s tou rychlostí by mohli něco dělat. Problémy s komunikací jsem nikdy neměl. Vždycky mi vyhověli. Řekl bych, že kvůli jednomu pitomci za přepážkou nemůžete odsoudit celou banku. Já jsem zcela spokojen.
- Jiří Svoboda:** Mozilla 1.2.1 , JRE 1.4.1_01 a jede to perfektně včetně levého JAVA menu i s diakritikou.
- Michal Kocer:** Co se týče Konqueroru, měl jsem nouzi a byl jsem nucen jej užít ve verzi 3.0.4 (v KDE 3.0.4) pro eBankování a neměl jsem nejmenších problémů.

Moderní souborové systémy – ReiserFS

Petr Novický

Úvod a motivace

V dnešním díle seriálu o souborových systémech se budeme bavit o historicky prvním linuxovém souborovém systému, který podporoval žurnálování. Jistě jste jméno ReiserFS už někdy slyšeli a v tomto díle se o něm dozvíte něco blíže. Hlavní osoba, která za tímto projektem stojí, je Hans Reiser, po kterém byl filesystém pojmenován. Ovšem vyvíjelo ho s ním samozřejmě ještě spousta dalších spolupracovníků.

Teď byste možná rádi věděli, proč ReiserFS používat. Má spoustu vlastností, které ostatní filesystémy neobsahují. Jak jsem již řekl, jedná se o žurnálovací souborový systém, tudíž jednou z výhod, které poskytuje, je zajištění konzistence dat při výpadku a také zrychlení obnovení systému oproti fsck.

ReiserFS je založen na rychlém balancovaném stromu. Je tak zrychlena práce např. s velkým množstvím položek v adresáři. Už není žádným problémem, aby adresář obsahoval třeba 100 000 souborů. ReiserFS s takto obsáhlým adresářem pracuje bez problémů.

Třetí a nejdůležitější vlastností je jeho schopnost pracovat efektivně s místem na diskovém oddílu. ReiserFS neukládá data do bloků pevných velikostí. Pokud chceme uložit několik souborů velkých například 100 B, uloží jich ReiserFS filesystém několik do jednoho bloku. Použitím tohoto souborového systému tedy šetříme nezanedbatelnou měrou prostor na našem záznamovém médiu. Stejně tak jsou do jednoho bloku ukládány konce souborů, které už nezabírají celý blok. Na druhou stranu tím však dochází ke snížení výkonosti filesystému, kvůli vyšší míře externí fragmentace.

Dalším problémem je přidání dalších dat na konec souboru. V tomto případě je na první pohled zřejmé, že se jedná o složitější problém, než např. u Ext2, kde přidání několika bytů je triviální záležitostí. Jednoduše se použije nevyužitá místa v rámci bloku, který používá soubor jen z části. Ovšem u Reiserfs musí dojít k přesouvání dat do jiných bloků, aby se tam přidávaná data vešla.

Uchovávání malých souborů se hodí hlavně v případě databází. ReiserFS byl navržen, aby umožňoval ukládat malé záznamy na disku, tudíž aby nebylo nutné používat nějaké způsoby shromažďování dat do větších souborů. Tento přístup by měl nejen zvýšit výkon, ale také snížit dobu vyvíjení aplikací tím, že programátoři nemusí vymýšlet způsoby, jak spojit několik malých bloků dat do jednoho velkého. Mohou je jednoduše na disk ukládat každý zvlášť.

Instalace

Jako první věc, kterou musíme provést, pokud chceme používat ReiserFS filesystém, je opět kompilace kernelu, který tento filesystém podporuje. Stačí povolit volbu `CONFIG_REISERFS_FS`. Pokud používáte `make menuconfig` nebo `make xconfig`, najdete ji v menu Filesystems, položka Reiserfs support. Podpora ReiserFS se dá zkompilovat přímo do kernelu nebo jako modul. Pokud budeme používat Reiserfs na kořenový svazek, je nutné zkompilovat podporu Reiserfs přímo do kernelu.

Pro vytvoření nového filesystému poté slouží příkaz:

```
mkreiserfs /dev/hdXX
```

kde `/dev/hdXX` je zařízení, kde chcete nový filesystém vytvořit. Poté již můžeme bez problémů nový diskový oddíl připojit. Nevýhodou oproti ext3 je bohužel nemožnost přetransformovat filesystém z klasického ext2 na reiserfs bez ztráty dat.

Dále je nutné přidat záznam do konfiguračního souboru `/etc/fstab`, tento záznam může vypadat následovně:

```
/dev/hdc1 /home reiserfs defaults 0 0
```

Položka `/home` je místo, kam chcete diskový oddíl připojit, tudíž si ho samozřejmě můžete změnit podle potřeby.

Parametry zrychlující práci filesystému

Jak jsem již uvedl, ReiserFS má i některé své nedostatky (práce s konci souborů), které výkon snižují. Tyto nedostatky se však dají částečně napravit pomocí správných nastavení při připojování oddílu. Pomocí volby `noatime` zrušíme záznamy filesystému o posledním přístupu ke všem položkám filesystému. Další volbou, která zrychlí práci s filesystémy je `notail`. Tím zrušíme ukládání konců souborů do jednoho bloku. Ztratíme tím sice nějaké místo na disku, ale dojde k zrychlení práce filesystému. Tedy např. kořenový filesystém znovu připojíme se správnými parametry pomocí příkazu:

```
mount / -o remount, notail
```

Opravený záznam v souboru `/etc/fstab` je následující:

```
/dev/hdc1 /home reiserfs noatime,notail 0 0
```

Změna velikosti filesystému

Užitečnou vlastností ReiserFS filesystému je možnost změny jeho velikosti. Příkaz `resize_reiserfs` slouží právě ke změně velikosti diskového oddílu. Ke zvýšení velikosti ho dokonce nemusíme ani odpojovat. Pokud souborový systém zvětšujeme, je nutné, aby bylo samozřejmě na disku volné místo (např. zrušením následujícího oddílu pomocí příkazu `cfdisk`). Když chceme zmenšit filesystém, musíme nejprve použít `resize_reiserfs` a poté teprve `cfdisk` (pozor aby jste ho nesnižovali na menší velikost, než kterou na něm data zabírají!).

Například kdybychom chtěli zmenšit velikost filesystému na diskovém oddílu `/dev/hda2` připojeném na adresář `/mnt`, postupovali bychom následovně:

```
umount /mnt
resize_reiserfs -s -1G /dev/hda2
mount /dev/hda2 /mnt
```

Bližší informace získáte z manuálové stránky `resize_reiserfs(8)`.

Shrnutí

K dalším výhodám souborového systému ReiserFS patří podpora žurnálovacího souboru na odděleném diskovém oddílu a dynamické alokování inodů. Jak jsem se již zmínil také možnost online zvětšit filesystém. Zajímavá je podpora špatných bloků a pomocí externího patche i možnost `quot`.

GNU Lilypond – dokončení

Tomáš Kopeček

Vzhůru do příkladu

Vítám vás u druhého pokračování seriálu o programu na sázení notových partů GNU Lilypond. Pokud vás minulý díl zaujal, tak teď přijde křest ohněm. Ze začátku to bude celkem nevinné a pak se ponoříme do hlubin překladače, abychom se na závěr vynořili i s nějakými těmi znalostmi.

Minule se vám (doufám) podařilo nainstalovat Lilypond a ukázali jsme si velmi, velmi jednoduchý příklad. Jenže jedna nota asi hudbu nedělá, takže zkusíme napsat kousek skladby pro klavír:

```
\version "1.6.3"
\include "paper16.ly"
\include "deutsch.ly"

% \header {
%   title = "Imagine"
%   composer = "J. Lennon"
%   tagline = ""
% }

right = \notes {
\property Staff.TimeSignature \override #'style = #'C
\time 4/4
\clef bass
<e4 g><e g><e g><e g h> |
<f a c'><f a c'><f a c'><f16 a> ais h8
}

left = \notes {
\clef bass
a,8 a a, a a, a a | d a d a d a4.
}

\score {
  \simultaneous {
    \context Staff = upper \right
    \context Staff = lower \left
  }
}
```

```
\paper {}  
\midi {}  
}
```



Uf, tady je toho najednou nějak moc. Nebojte se, zde si ukážeme základy práce a zbytek už je jen omáčka kolem. První tag je `\version`. Je celkem intuitivně zřejmé, k čemu se používá. Během vývoje se přece jen občas změnila syntaxe jazyka a aby skript `convert-ly` mohl vše správně převést do nejnovější verze, tak je třeba udržovat tuto informaci.

`\include` opravdu dělá to, co by od něj člověk očekával. Vkládá tedy soubor nalezený v cestě. První soubor, tedy `paper16.ly` nastavuje všechny potřebné hodnoty podle velikosti použitého fontu – 16 bodů. Druhý soubor slouží k načtení německých názvů not. Přece jen jsme asi zvyklejší na `h`, než na anglické `b`.

Dalším tagem je `\header`. Zde vidíme poprvé to, co bude následovat téměř všude a to jsou složené závorky. Podobně jako třeba v `TeXu` označují nějakou uzavřenou skupinu. Uvnitř `\header` je možno nastavit vše, co se bude objevovat kolem notového partu, tedy nadpisy, autory, nástroje, apod. Jak vidíte, je zapoznámkován, to jen abyste viděli, že jednořádkové poznámky jsou stejné jako v `TeXu`. Víceřádková poznámka vypadá takto `%{ ... %}`.

A teď se blížíme k hudbě, ovšem pozvolna a to přes makra. Makra v pojetí Lilypondu nejsou zdaleka tak mocná jako `TeXová`. Nemají žádné parametry, takže slouží spíše k ukládání opakujícího se kódu a především k rozumnému členění zdrojového textu. Nadefinujeme tedy makro `\right`, které bude obsahovat noty hrané pravou rukou (Ve zdrojovém textu to lomítko opravdu nechybí, pokud by tam bylo, tak by se Lilypond domníval, že již bylo definováno, pokusil by se jej rozvinout a už by vám spílal, jak jste ho asi zapomněli nadefinovat.) `\notes` říká, že vše co bude následovat se má interpretovat jako notový záznam (jiné možnosti jsou `\chords` a `\lyrics`, které budeme probírat později).

Nejdříve přeskočíme jeden řádek a nastavíme celý takt pomocí `\time 4/4`. Tento zlomek může být libovolný. Pokud pro danou kombinaci existuje nějaká značka (čtyřčtvrtový, dvoupůlový takt apod.) bude vykreslena. V opačném případě se zobrazí zlomek. Toto chování ovlivňuje vlastnost `style`. A tímto se vracíme k předchozímu řádku. Mocným nástrojem Lilypondu jsou právě vlastnosti jednotlivých objektů.

Střeva Lilypondu

Nyní od příkladu lehce odbočíme a osvětlíme si způsob, jak vlastně Lilypond pracuje. Vypíšeme si jednotlivé fáze a poté si je popíšeme:

- Inicializace
- Syntaktická analýza
- Interpretace
- *Prebreaking* – předlomení
- *Preprocessing* – příprava tiskových objektů
- *Break calculation* – definitivní výpočet zlomů
- *Breaking* – vlastní lámání
- *Outputting* – převod výstupu z interního formátu do požadovaného výstupního formátu

Nejdříve se načtou všechny inicializační soubory a aktivuje se scheme (derivát LISPU). Pak proběhne fáze syntaktické analýzy. Kdo neví, co taková slova znamenají? Někdo se hlásí? No dobře. Pro naše účely povíme velmi zjednodušeně, že (v případě Lilypondu) jde především o kontrolu, zda je vstupní soubor zapsán správně a neobsahuje kupříkladu neuzavřené závorky, překlepy v příkazech a podobně. Pokud jsou všechna vstupní data zadána správně, dojde následně k interpretaci těchto dat. V této fázi se vytváří logický popis hudebního partu v pořadí v jakém by noty byly hrány živými hráči. Z takto získaného popisu lze již stanovit pravidla pro sazbu.

Během syntaktické analýzy se již shromažďují základní údaje, jako například délky a výšky not (nebudeme přece soubor zbytečně číst dvakrát). Vytvoří se jakési základní objekty. Hlavními takto vzniklými objekty jsou `Score` a `Music`. První z nich popisuje celistvý úsek skladby a odpovídá obsahu příkazu `\score`. Druhý je pak seznamem všech hudebních událostí v dané části.

Interpretační fáze pak pracuje s jedním objektem `Score` a postupně předává hudební události patřičným interpretačním kontextům. Ty slouží k tomu, aby z hudebních událostí sestavily odpovídající grafické objekty (GROB). V programu je to zařízeno kolekcí objektů `Music.iterators`. Ty procházejí seznam `Music` a předávají získaná data objektům `Translator`. Výsledkem této části je vlastně hierarchická síť GROBů. Tyto objekty jsou navzájem propojeny a každý z nich má definovány elastické vazby na své okolí.

To zní hrozně, takže jednodušeji. Třeba osminová nota je definována hlavičkou. K hlavičce se připojí nožička (vazba musí být pevná) a k nožičce praporek opět pevnou vazbou. Tato nota je umístěna v osnově a osnova je navázána na účarí. Celá nota pak má vlastnosti určující, kolik místa minimálně potřebuje a naopak jak nejdál od ní může být nějaký další objekt, třeba pomlka. Technicky je to vlastně soustava rovnic s mnoha neznámými. Podaří-li se tuto soustavu vypočítat máme vyhráno.

A to už se dostáváme k fázi *Prebreaking* (něco jako předlámání). V místech, kde by mělo dojít ke zlomu, se doplní klíče a svislé čáry a pokračuje se svižně dál. Během *Preprocessingu* se dopočítají všechny údaje potřebné k přesnému stanovení jednotlivých zlomů. Nyní je možno vypočítat všechny pozice sloupců na které se později umístí jednotlivé GROBy. Tato fáze se nazývá *Break calculation* (výpočet zlomů). Následuje *Breaking* (lámání). Nalámání na řádky přineslo nové změny do vztahů mezi jednotlivými GROBy. Tyto vztahy se tedy znovu upraví. Typickým případem takové změny je přerušení ligatury zlomem. Musí se tedy vytvořit místo jedné ligatury dva nové objekty, každý na jiném řádku. A už jsme na konci. Máme vše vypočítáno a můžeme se pustit do tisku. Tuto fázi pro pořádek nazýváme *Outputting* (výstup).

Snad jste se v tom všem neztratili a všimli jste si interpretačních kontextů. Tyto kontexty nejsou nic ukrytého hluboko v Lilypondu a dají se přímo ovlivňovat a vytvářet. Základním kontextem je `Score`. Ten obsahuje v našem krátkém příkladu dva kontexty `Staff` (notová osnova). Jiným podobným typem kontextu je třeba `Staffgroup`, který může obsahovat více `Staffů`. I uvnitř kontextu `Staff` existují kontexty a to `Voice` a `Thread`. První označuje hlas a druhý konkrétní linku not. Nebudeme se jimi zde zabývat, ale je dobré vědět, že existují. Používají se třeba pro změnu tvaru hlaviček not pro rozeznání hlasů ve stejné osnově. Základní kontexty se vytvářejí automaticky, takže když napíšete jen několik not, automaticky se vytvoří kontext `Score` obsahující `Staff`, který zpracuje vámi zadané noty. Uměle vytvořit kontext však není problém a v příkladu je tento postup také použit za pomoci příkazu `\context = ...`

Zpět k příkladu

A konečně se můžeme vrátit k příkladu. Radši zopakuji řádek, kde se nyní nacházíme:

```
\property Staff.TimeSignature \override #'style = #'C
```

Řekli jsme si, že každý kontext má mnoho vlastností. Tyto vlastnosti se dají měnit pomocí příkazu `\property`. K vlastnosti se dostaneme podobně jako leckde jinde: `kontext.vlastnost`. Následuje metoda, jakou chceme tuto vlastnost změnit. Na výběr jsou tyto čtyři: `=`, `\override`, `\set` a `\revert`. První se používá pro nastavení vlastností celého kontextu. Zbývající tři nastavují vlastnosti GROBů uvnitř kontextu a mají tedy navíc další parametr – jméno vlastnosti GROBu. Rozdíly mezi `\set` a `\override` ponecháme pokročilejším a spojíme se s tím, že si většinou vystačíme právě s `\override`. `\revert` podle očekávání mění booleovská nastavení do opačné polohy (např. směr nožiček).

Poslední upozornění se týká znaku #. Ten slouží k přepnutí ze syntaxe Lilypondu do syntaxe Scheme. Všechny objekty jsou totiž popsány ve Scheme a musíme k nim tak přistupovat. Apostrof v Scheme značí, že bude následovat textový řetězec, tedy název vlastnosti. A co to tedy celé dělá? Říkáme, že se má vykreslit znak 'C', nikoliv zlomek 4/4. Tolik textu kvůli takové prkotině? No to jsem vám jen chtěl osvětlit, jak to všechno pracuje.

Pak následuje celý blok not. Zápis not je opravdu jednoduchý. První část je název noty (norma pro názvy not, lze načíst třeba takto `\include "deutsch.ly"`), nebo `r` jako pomlka (rest) dále řekneme, ve které oktávě nota leží. Apostrofy a čárky odpovídají patřičným oktávám. Za tím vším se může objevit případné prodloužení noty pomocí teček. Mezi notami jsem ukázal několik základních značek. Kulaté závorky označují začátek a konec ligatury. Hranaté příkazují zalomení trámců na daném místě.

Následuje `\paper`. Tag obsahuje všechna technická nastavení. Většinou stačí jeho uvedení s prázdným tělem. Pokud by tu totiž nebyl vůbec, Lilypond by nevytvořil žádný tiskový výstup. Na co je dobré netisknout? Třeba si chceme jen poslechnout hudbu. Kromě tisku vytváří Lilypond i zvukový výstup ve formě MIDI souboru. Nečekejte nic světoborného, je to dobré především na akustickou kontrolu. Druhý důvod proč netisknout může být syntaktická kontrola zapsaného textu. Zpracování souboru bez tisku je totiž výrazně rychlejší (nepočítají se žádné pozice).

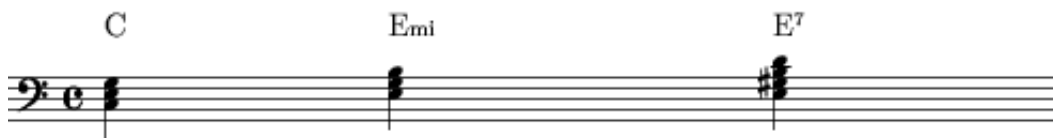
Další je `\midi`. Ten nám říká, že se má generovat i zvukový soubor. Je to podobné jako s `\paper`. Uvnitř se dá nastavit například tempo, nebo výchozí hudební nástroj.

Zápis not

Nebudeme se zabývat gregoriánským chorálem, ale hudbou popovou. Pro naše účely ji definuji jako jednoduchou (zpravidla jednohlasou) hudbu s akordovými značkami a slovy.

Napsat notovou linku už pro nás není problém. Zbývají nám tedy akordové značky a slova. Co naplat, vzhůru do nich. Pro zadání akordů můžeme použít dva módy. Bez dlouhého povídání si je předvedeme:

```
akordy = \ chords { c4 e:m e:7 }
% akordy = \ notes { <c4 e g><e g b><e gis b d'>}
\score {
  \simultaneous {
    \context ChordNames \akordy
    \context Staff {\clef bass \akordy }
  }
}
```



Prvním módem je `\chords`. Je rozhodně používanější variantou. Zápis akordů je v něm podobný jako zápis not. První část je jméno noty, následuje volitelně délka. Pokud se jedná o něco jiného než běžný durový kvintakord, tak následuje `:` a za ní všechny možné modifikace. Základní je písmeno `m` označující mollový akord. Je to parametr patřící mezi tzv. pojmenované parametry. Jiné jsou např. `aug`, `sus`, `dim`. Druhou možností jsou přímo číselné hodnoty. Chceme-li přidat septimu, napíšeme prostě `7`. Pro sníženou tercii, tedy ekvivalent parametru `m` napíšeme `3-`. Jednotlivé parametry se spojují tečkou. Pro obraty akordů použijeme například zápis `c/g`. Odstranění tónu z akordu se provádí takto: `c3`. Poslední variantou je přidání basového tónu: `c/+g`.

Druhou možností je ona zapoznámkováná. Vidíme, že je zde zápis obsahově ekvivalentní s prvním, nicméně provedený ve standardním translatoru `\notes`. Tyto zápisy jsou ekvivalentní nejen co do obsahu ale

i v použití. Lze je zcela volně zaměnit. Při této příležitosti si povšimněte, jak je vytvořen objekt Score. Vidíme, že při použití kontextu ChordNames se tisknou názvy not a v kontextu Staff se tiskne osnova s patřičnými notami a to obé za použití stejného vstupu (`\akordy`). `\simultaneous`, jak jste si jistě domysleli, působí souběžný tisk obou kontextů nad sebou. Pokud by byly uvedeny bez tohoto zapouzdření, byly by vytištěny za sebou.

To by asi tak stačilo jako lehký úvod do akordových značek. Na závěr jen poznamenám, že názvy akordů jsou různé a mezi předvolenými se přepíná takto: `\property ChordNames.ChordName \override 'style = #'jazz`.

Zápis textů

```
melodie = \notes {
  \property Staff.TimeSignature \override #'style = #'C
  \property Staff.midiInstrument = "flute"
  \property Voice.automaticMelismata = ##t
  \time 4/4
  \key e \major
  \clef treble

  e'8 e' e' e' e' e' e' e' |
  {

slova = \lyrics {
  Před -- stav -- te si, před -- stav -- te si
}

\score {
  \simultaneous {
    \addlyrics
    \context Staff \melodie
    \context Lyrics \slova
  }
}
```



Ukážeme si opět jen nejpoužívanější variantu. Jednotlivé slabiky jsou v této variantě vázány na patřičné noty. Myslím, že zápis je dostatečně samovysvětlující, takže k možným volbám. `--` značí roztažitelnou pomlčku, která se umísťuje doprostřed mezi slabiky. Druhou variantou je tzv. extender `---`. Ten se táhne od první slabiky po celou dobu jejího trvání (vhodné např. u dlouhé ligatury). Volby jako velikost fontu, tloušťka a podobně jsou opět dosažitelné přes vlastnosti kontextu. Pro složitější použití opět odkážu na vzorové příklady a manuál.

Výhody Lilypondu

Takže co dělá Lilypond tak dobrým? Ne na posledním místě je kvalita fontů. Jsou vytvořeny v MetaFontu a jsou opravdu pěkné. Druhou výhodou je kvalitní návrh systému a sázeací algoritmus. Poslední je velmi těsné propojení s jazykem Scheme. Ne zrovna malá část Lilypondu je napsána v tomto jazyce (samotný překladač je v C++). Scheme se dá používat opravdu všude a dají se jím ovlivnit snad všechny vlastnosti systému. Pro zvrhlíky je dostupná i možnost přepsání celého výstupu vlastními rutinami. Většinu času jej není třeba použít. Nicméně na druhé straně se stejně jako v \TeX u občas najdou situace, které algoritmy neřeší zrovna kvalitně a tehdy nastupuje ruční ladění.

Lilypond umožňuje snad vše, co si v dané oblasti dokážete představit. Není problém pro něj sázet rytmickou linku pro bubny, stejně jako popovou písničku s textem a akordy. V módu *ancient* umí sázet i v barokním stylu (bohužel této notaci příliš nerozumím, takže nemohu posoudit jak kvalitně).

A co na závěr dnešního dílu? Společně jsme se probrodili bažinou syntaxe Lilypondu. Tímto uzavírám popis jazyka a v závěrečném díle se vrhneme na technické drobnosti, jako je zpříjemnění editace.

Od kódu k tisku a dál

Pokud jsme vše zdárně nainstalovali, napsali zdrojový kód, tak vzhůru do práce. Nejdříve probereme základní variantu. Máme vstupní soubor `example.ly`. Jak jej dostaneme do nějaké prohlížitelné podoby? **ly2dvi example.ly** převede kód za pomoci \LaTeX u do výstupního souboru `example.dvi`. Přidáme-li přepínač `-P`, dostaneme PostScriptový výstup. V případě `-p` pak dostaneme PDF.

Teď k druhé a pro mnohé asi zajímavější variantě. Protože formát souboru `.ly` neumožňuje integrovat notový zápis s textem, tak existuje i možnost, jak včlenit kód Lilypondu přímo do zdrojového textu \LaTeX u, \TeX Infu či HTML. Předvedeme si pouze variantu pro \LaTeX , druhé dvě varianty jsou totiž velmi podobné této a zájemce si konkrétní syntax rád dohledá v manuálu. Vlastní převod je navržen tak, že speciální skript `lilypond-book` projde \LaTeX ový (HTML, \TeX info) soubor a zpracuje označené úseky Lilypondem. Pak vytvoří nový soubor, kde úseky kódu nahradí vloženým obrázkem (eps, png).

```
\lilypond{...}
...
\begin{lilypond}
...
\end{lilypond}
```

První – jednodušší varianta se hodí zejména pro krátké zápisy (několik not). Jak vidíte, v \LaTeX u je připraveno také celé prostředí pro Lilypond. Má několik nepovinných parametrů:

eps

říká, že se výsledný obrázek má vložit jako encapsulated PostScript.

verbatim, smallverbatim, intertext=

kód se nejprve zkopíruje v módu `verbatim`. Za něj se vloží vše, co bylo definováno v parametru `intertext`. Na závěr se vloží vygenerovaný výstup.

filename=

mezivýstupy se budou ukládat do zadaného souboru (Pokud není parametr uveden, použije se hash textu.)

11pt, 13pt, 16pt, 20pt, 16pt

použije se uvedená sada fontů

singleline, multiline

použije se buď jediná notová osnova, nebo se v opačném případě bude lámat podle nastavení strany

Kromě těchto parametrů je definováno ještě několik dalších (`no/fragment`, `no/indent`, `noquote`, `relative`, `printfilename`), které mají celkem jasný význam.

Kód uvnitř prostředí je plnohodnotným zdrojovým textem Lilypondu. Bez problémů se zalamuje přes hranice stran, rozezná dokonce sazbu do sloupců a uzpůsobí tak své rozměry. Tato vlastnost je však omezena pouze na příkazy `\onecolumn`, `\twocolumn` a balíček `multicols`.

Pro toto prostředí jsou také definovány dvě speciální makra, které je možno podle potřeby předefinovat. Jsou to `\preLilypondExample` a `\postLilypondExample`. Tato makra se vykonávají jak jinak než na konci a na začátku prostředí (představme si je někde uvnitř `\begin{lilypond}` a `\end{lilypond}`). Většinou se ponechávají prázdná, ale lze je použít k nastavení vlastního vzhledu vložených obrázků. Budeme-li například chtít před a za příkladem udělat čáru, můžeme to udělat takto:

```
\def\preLilypondExample{\par\noindent\hrulefill\par\noindent}
```

Prostředí v dokumentu HTML vypadá takto:

```
<lilypond verbatim>
<c16 e><c e g>
</lilypond>
```

a v \TeX info pak:

```
@lilypond[20pt, eps]
\relative c' {
  c d e f
}
@end lilypond
```

Pro další podrobnosti (už jich snad k tomuto tématu moc není) odkazují na příslušnou část manuálu.

Podpůrné prostředky

K většině značkovacích jazyků existují různé pomocné editory, kterým se více či méně daří zefektivnit práci na kódu. Například pro jazyk HTML jsou jich stovky. Jazyk Lilypondu není zdaleka tak rozšířen a z toho také plyne výrazně nižší počet těchto prostředků. Asi tak tři. Prvním je grafický frontend Denemo, zmíněný již v první části tohoto seriálu. Hlavním problémem je neaktivita tohoto projektu. Je to však jediné klikací prostředí, které pro Lilypond existuje. Dle mého osobního názoru je dobré tak akorát na to, aby se uživatel naučil, jak některé konstrukce, které nakliká, vypadají v kódu Lilypondu.

Dalšími dvěma prostředky jsou módy pro emacs a ViM. Nejsa uživatelem emacsu neporeferuji příliš o první variantě. Nicméně alespoň něco. Mezi standardní požadavky patří syntax-highlighting. Další možnosti, které toto prostředí podporuje, jsou různé výpomoci při kompilaci a náhledy. Též jsou přítomny různé zkratky pro rychlejší psaní not a tagů. Většina těchto možností je přístupná z menu. Opět téměř samozřejmostí je inteligentní odsazování.

Jak je to s ViMem? Hůř. V současné distribuci je sice soubor `vimrc`, ale nenechte se zmást. Je to spíš takový pozůstatek. Obsahuje jen několik maker pro kompilaci a ta se spíš nedají použít. Raději si napište vlastní. Kde se ovšem blýská na lepší časy, je konference `lilypond-devel`. V současné době se připravuje kvalitní soubor pro ViM včetně `syntax-highlightingu`, který je už teď údajně lepší než emacsový.

Co ovšem může některým uživatelům výrazně pomoci, je možné propojení `xdvi` s příslušným editorem. Bohužel většina distribucí nabízí `teTeX`, který obsahuje upravenou verzi `xdvi` (`xdvik`), která neodpovídá současné verzi tohoto programu. Ujistěte se tedy, že máte alespoň verzi 22.36. Údajně je tato možnost též podporována prohlížečem KDVI v KDE 3.0, kde musíte povolit vlastnost "Inverse search". V případě, že

nemáte ani jeden z těchto programů, tak zde je **xdvi**. Druhým požadavkem je textový editor umožňující přijímat externí příkazy. Tím se náš výběr omezí na emacs, GViM a NEdit.

Ve zdrojovém souboru musíme zapnout generování speciálních informací takto: `#(set-point-and-click! 'line)` přesun na odpovídající řádek, nebo `#(set-point-and-click! 'line-column)` přesun na řádek a sloupec. Podle dokumentace je mód `line-column` funkční pouze v emacsu (a to ještě po aplikaci patche dodaného s Lilypondem), ale my si ukážeme jak jej použít i ve ViMu. V emacsu jsem tento mód nikdy nepoužil, takže pouze napíšu, co říká dokumentace. Do souboru `~/ .emacs` přidáte řádek `(server-start)` a proměnnou prostředí `XEDITOR` na `emacsclient --no-wait +%l:%c %f` (v případě `line` vynecháte `:%c`). Teď už můžete spustit **xdvi** a kliknout třeba na některou notu. V emacsu se otevře odpovídající zdrojový soubor a skočí se na příslušnou část kódu.

Pro NEdit proměnnou `XEDITOR` nastavíme na hodnotu `nc -noask +%l %f`. Teď tedy slíbené nastavení pro GViM. Stačí `export XEDITOR=gvim --remote +%l:norm%c| %f`. Upozornění pro zapřísažlé nepoužívače GViMu ve prospěch ViMu. ViM tuto možnost neposkytuje, protože není schopen běžet v serverovém módu. Tuto možnost má opravdu jen grafické rozhraní.

Čeština

A co proboha ta čeština? Pokud budete sázet jen notové party, tak tento problém nemusíte řešit. Pokud ovšem budete chtít vidět nějaký ten háček a čárku, tak je to slabší. Problém je v tom, že ještě stále neexistuje jednotná podoba českého \LaTeX u a druhý problém je v systému ovládání fontů samotného Lilypondu. Lilypond standardně používá Knuthovy Computer Modern fonty. `ly2dvi` umí používat `babel`, takže kódování není problém, ovšem háčky a čárky nejsou zrovna estetické.

Pro ty, co mají nainstalovanou českou variantu – Cs-fonty (součást instalace `cslatexu` a `csplainu`), stačí použít tento patch, respektive zaměnit všechny výskyty CMR fontů v souboru `scm/fonts.scm`. Patch záměrně není vztažen ke konkrétní distribuci, ale pouze k souboru `$PREFIX/share/lilypond/1.x.x/scm/font.scm`. Tak snad bude zaručena jeho použitelnost pro co největší spektrum verzí (tento soubor se už hodně dlouho nezměnil). K čemu už jsem patch neposlal, je soubor `lilypond-book`. Tedy pokud budete chtít, tak zaměňte v tomto souboru jediný výskyt slova `latex` za `cslatex`.

Tím jsme se zbavili problémů s češtinou jako takovou. Větší problém, je to, že ani Češi nejsou jednotni v tom, jak se vlastně mají označovat jednotlivé akordy. S potěšením jsem zjistil, že v tom nejsme sami. V Lilypondu tedy existuje hned několik základních módů označení akordů. První a výchozí je `banter`. Vychází ze systému navrženého jistým Haroldem Banterem a je z použitých nejbližší českým jménům. Další varianty jsou `american` a `jazz`. Ovšem i v těchto názvech se najdou výjimky, které se českým čtenářům nebudou příliš líbit. Mollové akordy se označují například jako `Am`, což je u nás méně časté označení. Zde je ovšem síla propojení Lilypondu se Scheme. Jména akordů jsou definována v tomto jazyce a případné změny, či výjimky lze snadno podchytit.

Co říci závěrem?

Projekt má za sebou již poměrně dlouhý vývoj a značně vyzrál. Výstup již může konkurovat komerčním programům a leckdy je i předčí (z vlastní zkušenosti). Budoucnost je celkem slibná. Pevné jádro vývojářů stále pokračuje v práci s neutuchajícím nadšením. Nezbyvá tedy než poděkovat oběma hlavním autorům za jejich práci a jít sázet kytičky (nebo notičky?). Těžko totiž asi v současné době najdete lepší free software tohoto typu.

Nebylo by slušné nezmínit se například o souvisejícím projektu `Mutopia`, který shromažďuje notové záznamy (většinou vytvořené právě v Lilypondu). Jedná se o skladby, na které se již nevztahuje žádný copyright, tedy především o skladby klasické. Je to vlastně takový hudební protějšek projektu `Guttenberg`.

Poznámka pro experimentátory: v CVS je dostupná vývojová řada `1.7.xx`. Pokud se rozhodnete si hrát s touto verzí, tak buďte varováni, že se syntaxe opět mění a není zaručeno, že všechny v seriálu zmiňované postupy budou fungovat bez problémů. Například přechody mezi `scheme` a Lilypondem mají již syntaxi změněnu.

Poznámka pro ostatní: pokud bojujete s množstvím chybových hlášení překladače a nevládnete příliš angličtinou smíchanou s mnoha hudebními výrazy, tak stahujte aktuální verzi, která již obsahuje můj patch (PO) s českými hláškami.

Jaderné noviny 193–196

Leoš Literák

Úvod

Následující článek shrnuje nejzajímavější témata z konference linux-kernel. Na tomto místě vývojáři v čele s Linusem Torvaldsem diskutují nad otázkami kolem vývoje a budoucnosti jádra Linuxu (kernelu). Článek je zkráceným překladem seriálu Kernel Traffic a je zveřejněn pod licencí GPL verze 2. Originál v angličtině včetně archivu najdete na adrese <http://kt.zork.net>.

Pročišťování aplikačního rozhraní devfs

Alexander Viro se rozhodl podrobit devfs dalšímu hloubkovému průzkumu. Poslal svá zjištění, z nichž vyplynulo, že

1. spousta funkcí exportovaných z devfs není vůbec používána. Nikde.
2. podstatná část ostatních funkcí je používána pouze projektem hwgraph od SGI.
3. mnohé funkce obsahují zbytečné argumenty.
4. devfs nepoužívá strukturu dev_t, ale přenáší major a minor čísla zvlášť, což vede ke zbytečně komplikovanému kódu jak u devfs tak u volajícího kódu.
5. devfs_entry je noční můra, je to struktura, která obsahuje union, jehož položkou je struktura obsahující funkce, příznaky a union z dev_t (pár major a minor číslo) a size_t. Důvodem je snaha o používání jedné položky pro obyčejné soubory, znaková i bloková zařízení. Jejich *jediným* společným bodem je sada příznaků.

Richard Gooch odpověděl, že odpoví na Alexandrovy body, až najde čas za týden či dva. Dodal, že se bojí porušení kompatibility mezi ovladači řad 2.4 a 2.5. Alexander odpověděl:

Na to už je trochu pozdě. Kompatibilita mezi 2.4 a 2.5 byla u ovladačů již narušena a devfs se neobjeví poblíž vrcholu seznamu.

Jinde se Ryan Anderson zeptal Alexandra, zda by po vyčištění, které navrhnul, byl sám ochoten používat devfs. Alexander odpověděl, že nikoliv. Theodore Y. Ts'o navrhnul úplně odstranit devfs a poznamenal, že jej stejně nepoužívá moc lidí. Okamžitě se však ozvala řádka lidí, kteří jej používají a mají rádi.

Stav ACPI

Andrew Grover ohlásil další verzi ovladačů ACPI, která je dostupná na adrese <http://sf.net/projects/acpi>. Tato verze řeší problémy se čtením informací z baterie a zrychluje jejich načítání.

Bugzilla sleduje chyby v jádře

Martin J. Bligh ohlásil, že spustil bugzillu na sledování chyb v jádře. Najdete ji na adrese <http://bugme.osdl.org>. Poprosil vývojáře, aby se zaregistrovali a vkládali zde nalezené chyby. Bugzilla je zatím určena jen pro řadu 2.5 a jejím hlavním účelem je urychlit vytvoření řady 2.6. Je třeba ještě vytvořit lepší dělení na kategorie, což se časem stane, jak bude narůstat databáze chyb.

Pete Zaitcev se přihlásil jako správce chyb pro platformu Sparc a zeptal se, kdo je administrátor bugzilly. Ozval se Khoa Huynh, který věnuje část svého času pro správu bugzilly. Jeho tým z IBM sídlí v různých oblastech a časových zónách, od Texasu po Indii. Tým poslední dva roky pracoval na odstraňování linuxových

chyb pro interní účely a nyní by chtěli pomoci udržovat tuto databázi ve svém volném čase. Nicméně dobrovolníci z komunity jsou vítáni, zvláště až naroste objem ukládaných chyb.

Jednotliví vývojáři se začali hlásit o odpovědnost. Paul Larson se zeptal, zda stačí registrovat chybu v bugzille nebo je nutné zaslat informaci i do konference. Robert Love odpověděl, že rozhodně je důležité stále oznamovat je v konferenci. Pete Zaitcev však podotkl, že je možné přidat konferenci jako CC během registrace chyby. David S. Miller napsal, že odmítá pracovat na chybách, které nejsou v Linusově stromu. Hned první přiřazená chyba se totiž týkala jen patchů od Andrewa Mortona. Eric Northup navrhnul vytvořit kategorie pro ostatní stromy, s čímž Alan Cox souhlasil. Poprosil o vytvoření produktu 2.5-ac a přiřazení všech jeho chyb.

Larry McVoy napsal, že BitKeeper používá unikátní jména pro každý changeset a nyní upravují BK/Web tak, aby používal tato unikátní jména místo revizí. Takto bude možné spojit oznámení chyby přímo s changesetem v Linusově stromu. Khoa byl nadšen z této vlastnosti a zeptal se na její formát. Larry odpověděl, že momentálně používají něco jako

```
torvalds@home.transmeta.com|ChangeSet|20021115061315|00914
```

což je hnusné. Plánují to změnit na kontrolní součet typu `dSD4ok0iGmLGDcqOTpQPFQ==`. Pak bude stačit jen otevřít následující URL:

```
http://linux.bkbits.net:8080/linux-2.5/cset@dSD4ok0iGmLGDcqOTpQPFQ==
```

a získáte vždy stejná data.

Problémy s moduly v řadě 2.5

Christoph Hellwig si postěžoval Linusi Torvaldsovi a Rusty Russellovi:

Co se to sakra děje s moduly v 2.5-CURRENT?

Rustyho superpatch všechno rozbouřil (vzpomeňte si, že jsme v zamrznutí kódu). Místo aby dělal jednu věc, tak provádí tři zcela různé věci najednou! Měli jsme použitelné jádro 2.5 a přesně teď během zamrznutí nových vlastností, kdy čekáme na testery, vše rozbijeme?

Linus odpověděl:

Zcela otevřeně, v tuto chvíli by záložní řešení znamenalo, že vůbec nic nebude začleněno. Přišlo to ještě před zmražením kernelu, ale rozhodl jsem se začlenit tento patch později kvůli své zaneprázdněnosti. A myslím si, že je hoden začlenění (je zde ještě několik patchů, o kterých uvažuji, například kprobes a posixový časovač, možná kexec).

Lidé, pro které je současná situace kolem modulů obtížná, mohou prostě do jádra zakompilovat součásti, které potřebují.

Na to reagoval Alan Cox:

To prakticky znemožní ladění ovladačů. Dále bude nemožné zkompileovat jeden kernel pro řadu různých strojů. Navíc PCMCIA pracuje jen jako modul a nyní nefunguje kvůli rozbitému mazání modulů [unload]. Na druhou stranu přepis modulů má několik hezkých vlastností a kombinované modutils vyřeší elegantně některé problémy. Nejvíce je třeba dokumentace, aby lidé mohli opravit ovladače pro tuto změnu.

Linus souhlasil, že PCMCIA je vážný problém, na který zapomněl. Ale dodal, že myslel, že Alexander Viro přesvědčil Rustyho udržovat kompatibilitu. Jenže Rusty napsal:

To ty jsi přesvědčil mne, abych neporušoval zdrojový kód žádného modulu: kdykoliv jsi odmítnul mé záplaty, sestavil jsem další makro pro kompatibilitu :-).

Masivní zpomalení u SMP s 2.4.17

Manish Lachwani ohlásil, že zpozoroval zatuhnutí systému s jádrem 2.4.17 zkompileovaným se SMP podporou. Děje se tak, když vytváří souborový systém za pomocí `mke2fs` paralelně na dvanácti discích. Andrew Morton odpověděl, že systém asi nezatuhne, jen Manish musí počkat asi půl hodiny. Algoritmus totiž není příliš dobrý.

A Theodore Ts'o doporučil nastavit proměnnou MKE2FS_SYNC na 10. Ta přinutí **mke2fs** synchronizovat po každém zápisu desítky bloků. Andrew souhlasil, že nastavení této proměnné vyřeší tento problém.

Podpora zvuku na čipsetu VIA VT8233A

Dmitry Kudryavtsev měl problém s rozběhnutím zvuku na jeho základní desce K7VTA3 s čipsetem VIA VT8233A. Jeff Garzik odpověděl, že kernel řady 2.4 tento audio čipset nepodporuje. Jeff doufá, že přidá podporu brzy, nicméně Alsa 2.4.x a v jádře 2.5.x začleněná Alsa podporuje tuto integrovanou zvukovou kartu.

RTLinuxFree 3.2

Victor Yodaiken ohlásil:

RTLinuxFree 3.2-pre1 byl vypuštěn. Tato verze předchází RTLinuxFree 3.2., ale je považována za stabilní. Obsahuje více než rok příspěvků od nás i ostatních. Najdete ji na adrese <http://www2.fsmlabs.com/3.2-free.html>. Na Pentiu 4 s jádrem 2.4.18 je čas nejhoršího scénáře zhruba 16 mikrosekund při 1 milisekundové periodické úloze. AMD K7 jsou také skutečně dobré. Na starších strojích očekávejte 30-40 mikrosekund.

O několik dní později se Pavel Machek zeptal na rozdíl mezi RTLinux a RTLinuxFree, ale nedočkal se žádné odpovědi.

Změny v IDE

Torben Mathiasen zaslal patch a vysvětlil Alanu Coxovi:

Prosím o aplikaci tohoto patche. Je to zjednodušení mého předchozího patche. Dělá následující:

- simplex detekce na úrovni disku, dynamická pro použití s hotplug.
- přinucení používání simplex pravidel při užití hdparmu.
- přinucení používání simplex pravidel při práci s BIOS časováním.
- přesunutí detekce časování IDE BIOSu do ovladače čipsetu (poskytuje funkce z knihovny)
- poskytuje funkci popsanou výše, která nastavuje předvolené hodnoty pro ladění parametrů. Navíc řeší případ, kdy uživatel požádá o použití časování IDE BIOSu.
- umožní práci BIOS IDE časování pro IDE0 i IDE1 ve stejný čas.

Andre Hedrick byl nadšen z tohoto patche.

Reverse-Mapping verze subsystému virtuální paměti

Rik van Riel ohlásil:

Toto je patnáctá verze pokusu o vytvoření robustního a flexibilního subsystému správy virtuální paměti, která čistí spoustu kódu. Patch je dostupný na adrese <http://surriel.com/patches/2.4/2.4.19-rmap15a> a <http://linuxvm.bkbits.net/>. V plánu mám backportovat zrychlení z řady 2.5 a pte-highmem.

XFS patche pro jádro 2.4.20-rc3

Keith Owens ohlásil patche pro žurnálovací souborový systém XFS vůči aktuálnímu jádru 2.4.20-rc3. V této verzi došlo k oddělení XFS od ostatních projektů jako je kdb, xattr, ACL či dmapi, proto čtěte README v každém adresáři velmi pečlivě. Otázky na problémy pokryté v README budou ignorovány.

Vlastnost či chyba Athlonu?

Dave Jones ohlásil:

Nejnovější Athlony (Model 8 stepping 1 a výše) (XP/MP a mobilní) mají zajímavý problém. Jisté bity v registru CLK_CTL jsou naprogramovány odlišně od původních modelů. Problém nastane, když se lidé snaží zasunout takový procesor do základní desky s BIOSem, který o tomto faktu nic neví. Řešením je přeprogramovat CLK_CTL na 200xxxxx místo 0x600xxxxx, jak to bylo v minulých modelech. Pánové z AMD říkají, že tato změna zlepšuje stabilitu.

Daniel Nofftz si ze své práce na Athlonu vzpomněl, že zmíněný registr je důležitý, když se procesor vrací zpět z acpi-c2 modu. V C2 je totiž procesor odpojen od systémové sběrnice a vnitřní hodiny jsou zastaveny [clocked down]. V některých případech špatná hodnota v tomto registru mohla zabránit procesoru vrátit se z C2 modu, což vedlo k zatuhnutí počítače (chyba 11 v revizi procesoru Athlonu). Dave to potvrdil a dodal, že registr obsahuje hodnoty, které indikují procesoru, jak nastavit hodiny v CPU během nízkonapěťového módu.

Nový grafický nástroj na konfiguraci kernelu

Romain Lievin upravil kgc (<http://tilp.info/perso/gkc.html>), grafické rozhraní k LinuxKernelConfu Romana Zippela tak, aby používalo knihovnu GTK+ 2.0.

Nastavení limitu zdrojů pro uživatele

Frederik Dannemare se zeptal, zda existuje způsob, jak omezit maximální využití procesoru na úrovni uživatelů. Našel starší patch od Connectiva Linuxu pro jádra 2.2. Zeptal se, zda jej někdo portoval na novější řady. Marc-Christian Petersen mu nabídl malý a pěkný patch od Karola Golaba, který naleznete na adrese <http://www.tls-technologies.com/CPU/cpu-intro.html>.

Rik van Riel napsal, že patch Connectivity byl portován a leží na adrese <http://surriel.com/patches/>. Hu Gang potvrdil, že patch funguje a Frederik napsal, že jej hned otestuje.

Martin Waitz informoval vývojáře, že jako svou diplomovou práci si zvolil implementaci kontejneru zdrojů. Ten poskytne hierarchický způsob pro nastavení limitů zdrojů. Takže pak budete moci nastavovat limity nejen pro uživatele, ale i na jakékoli úrovni, kterou si dokážete vymyslet (pro službu, pro klienta, ...). Práce bude hotova v lednu, ale Martin může zveřejnit zdrojové kódy dříve.

Co způsobilo korupci dat v ext3 v jádře 2.4.20

Andrew Morton ohlásil:

V jádře 2.4.20-pre5 byla provedena optimalizace do funkce fsync z ext3, která může snadno způsobit korupci dat během odpojování souborového oddílu. Chyba byla poprvé ohlášena dne 29. listopadu (den po vydání jádra 2.4.20), tedy tři měsíce po svém vzniku. Smolné načasování.

Týká se jen oddílů připojovaných s volbou `data=journal` a souborů pod `chattr -j`. Takže většina lidí jí není zasažena. Problém se nevyskytuje v řadě 2.5. Symptomem je, že jakákoliv data souborů zapsána během třiceti sekund před odpojením souborového systému nejsou zapsána na disk. Řešením je použít příkaz `sync` před odpojením.

Důvodem této optimalizace byla snaha o zabránění zápisu a čekání na vyrovnávací paměti inodu, když následné commity by to udělaly stejně. Tato optimalizace byla aplikována do modů `journal` i `ordered`, ale je platná jen pro mod `ordered`. V `journal` modu jsou tak data ponechána v paměti a po odpojení tiše ztracena.

Návrh na dynamickou správu napájení

Bishop Brock z IBM ohlásil:

IBM a MontaVista založili společný projekt na vývoj dynamického řízení správy napájení [dynamic power management control] pro Linux a podpory procesorů podporujících dynamické škálování napětí a frekvence. Popis najdete na adrese <http://www.research.ibm.com/arl/projects/dpm.html>. Byl vytvořen funkční prototyp tohoto frameworku pro procesory IBM PowerPC 405LP a bude brzy zveřejněn.

Alana Coxe tento materiál zaujal a požádal o vysvětlení jedné věty z dokumentu, která zmiňuje jiné přístupy řešení. Jenže tyto postupy jsou chráněny patenty. Alan chtěl vědět, co s tím bude IBM dělat. Bishop odpověděl:

To je důležitá a komplikovaná otázka. Náš kód prošel vnitřním auditem právního oddělení, nicméně jsou zde diskuse s našimi právníky o důsledcích patentu. Konečná odpověď snad přijde příští týden. Jedná se o patent US 6,298,448 a přestože jde o důležitou část našeho návrhu, nejedná se o centrální nápad, takže věřím, že návrh by měl cenu i bez něj.

Arjan van de Ven se zeptal, v jakém je tento návrh vztahu ke cross platformnímu frameworku cpufreq. Dominik Brodowski odpověděl, že podle jeho mínění se jedná o alternativu. Hollis Blanchard dodal, že tyto škálovací události má generovat raději kernel než vstup z uživatelského prostoru. Dominik však mínil, že se jedná o duplikaci cpufreq s několika přidanými vlastnostmi. Navrhnul, ať začlení návrh DPM do existujícího frameworku cpufreq. Nikdo ale nereagoval . . .

NTFS 2.1.0

Matthew J. Fanto ohlásil verzi 2.1.0(a) ovladače souborového systému NTFS pro kernel 2.4.20. Tento souborový systém interně používají Windows NT a Windows 2000. Ovladač najdete na adrese <http://linux-ntfs.sf.net>.