



Abíčko

Časopis serveru AbcLinuxu.cz

Listopad 2005



Vychází také na CD-ROM jako příloha časopisu



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 <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 <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. Novinkou posledních měsíců, která našla brzy odezvu, jsou blogy neboli internetové deníčky. Každý registrovaný uživatel si jej může založit a psát si do něj poznámky nejen o Linuxu.

V neposlední řadě chceme upozornit také na výkladový [slovník pojmů](#) a vznikající [elektronickou učebnici Linuxu](#), na níž se můžete podílet i vy!

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 směřujte na adresu: literakl@abclinuxu.cz.

Server <http://www.abclinuxu.cz> provozuje firma Stickfish 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://www.stickfish.cz>.

©2005 Stickfish s. r. o. a autoři článků

Editor a sazba: Vlastimil Ott

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

Typografické konvence

Ve výpisech zdrojových textů mohou být použity znaky `\\`. Značí přechod na nový řádek, který ovšem *není* součástí samotného zdrojového textu, byl přidán editorem z důvodu lepšího vzhledu případně nemožnosti text formátovat bez jejich použití.

Obsah

Editoriál	1
Obsah	2
Novinky v KDE 3.5	5
První spuštění	5
Vylepšený Konqueror	6
Ostatní vylepšení	6
Na co se nedostalo	8
Hodnocení	8
Jabber a ICQ: Imcom a mICQ	9
mICQ	9
Imcom	10
Jabber	11
Záver	11
Poznámky:	11
Reportáž: Blender Conference 2005, De Waag, Amsterdam	13
Pohled uživatele	13
Pohled vývojáře	13
Nejen počítačové obrazovky	14
Ark Linux	15
Instalace	15
První start	15
Programy	15
Ovládací centrum	16
Balíčky	16
Závěr	17
Kommander	18
K čemu je Kommander	18
Starý vs. nový parser	18
Hello, world!	19
Syntaktická odbočka	20
Správci rozložení	20
Rozpěrky	21
Vkládání příkazů	21
Akce	22
Ukázková aplikace	22
Proměnné	22
Klíčová slova Kommanderu	24
ScriptObject	24
Signály a sloty	24
Použití jiného skriptovacího jazyka	25
Problém se zavináčem	25
Nový parser	25
Proměnné a pole	26
Podmínky a cykly	26

Syntaktické rozdíly	27
Použití jiného skriptovacího jazyka	27
Aplikace napsané v Kommanderu	27
Static IP Configurator	27
kEchoSub	27
KDE DVD Authoring wizard	28
KQemu	28
Závěr	28
Jak se píše procesor	29
Úvod	29
Návrh obvodu	29
VHDL	29
Procesy	32
Jak psát design	34
Registry	35
Příklad	37
Kód	38
Závěr	40
Oracle: vývoj na SUSE 10.0	41
Úvod	41
Instalace serveru	41
Instalace klienta	42
TOra	42
gVim	43
Enterprise Manager	43
Slackware Linux na 386? Proč ne!	44
Entrée	44
Výběr distribuce	44
Příprava k instalaci	44
Start instalace	45
Rozdělení disku	46
Instalace	46
Výběr balíčků	46
Mám dva monitory – co s tím?	47
A co teď s tím?	47
Solartec – síť postavená na Linuxu	49
Úvod	49
Databáze	49
Poštovní systém	50
Samba	50
Zálohování	50
Hardware	51
Distribuce	51
Linux na desktopu	51
Pražská pobočka	51
Linux netradičně	52
Monitorování	52
Závěr	52
Jaderné noviny 326–330	53
Linux 2.6.13-rc7, 34 e-mailů	53

Nové testovací vydání SPUBS; diskuze o umístění v adresářové struktuře zdrojových kódů ..	53
Ovladač čidla zrychlení HDAPS; problémy s rozpoznáváním hardwaru, 36 e-mailů	54
Ovladač pro MPC8xx PCMCIA si našel cestu do jádra 2.6	54
Monitorování využití jader mezi dobrovolníky, 33 e-mailů	55
Linux 2.6.13; nová pravidla pro rychlejší vydávání nových verzí jádra, 42 e-mailů	55
Kontroverzní článek o stavu grafiky v Linuxu, 44 e-mailů	56
Podpora Marvell SATA, 21 e-mailů	56
Odstranění zastaralých funkcí; výzva správcům, aby aktualizovali ovladače, 10 e-mailů	56
Hádky vývojářů kvůli pravidlům posílání patchů, 15 e-mailů	57
Pár rad pro upgrade z 2.4 na 2.6, 5 e-mailů	58
DevFS má stále hlavu na špalku; uživatelé se brání, 32 e-mailů	58
Zpřístupnění NUMA dat uživatelskému prostoru, 7 e-mailů	59
Umístění patchů čekajících na -stable jádro, 3 e-mailů	59
Používání linux/irq.h nebo asm/irq.h v ovladačích, 5 e-mailů	60
Automatická konfigurace jádra, 42 e-mailů	60
Russellovy projekty, 7 e-mailů	60
Stěhování master.kernel.org do nového domova, 12 e-mailů	61
Linux 2.6.14-rc2, 38 e-mailů	62
uclinux portován na Blackfin CPU, 8 e-mailů	62
Vytížení Andrewa, 28 e-mailů	62
Linux 2.6.14-rc2-mm1, 42 e-mailů	63
Správa subsystému I2C, 36 e-mailů	64
Cogito 0.15.1 vydáno s důležitou opravou, 1 e-mail	64
Zprávičky	65

Novinky v KDE 3.5

Michal Vyskočil

KDE 3.5 již dospělo do stádia betaverze. Pojdte si přečíst a podívat se, co nového nabízí.



21. září 2005 vyšla první betaverze KDE 3.5. Byl jsem zvědavý na to, co je nového, proto jsem neodolal a pustil se do instalace. Díky vývojářům KDE a jejich aplikaci **Konstruk**t [1] je překlad hračkou. Ale pokud používáte SUSE nebo Kubuntu, máte už připravené **binární balíčky** [2]. Já, jsa uživatel Gentoo, žádné binární balíčky nemám a Konstruct jsem zvolil proto, abych měl nové KDE nainstalované mimo hlavní systém. Instalace totiž proběhne do adresáře `~/kde-3.5-beta1`, takže váš systém zůstane nedotčený. Kompilace KDE je velice jednoduchá:

```
tar -xjf konstrukt-unstable.tar.bz2
cd konstrukt/meta/kde
nice -n 19 make install
```

V Konstraktu máte na výběr mezi několika možnými cíli kompilace:

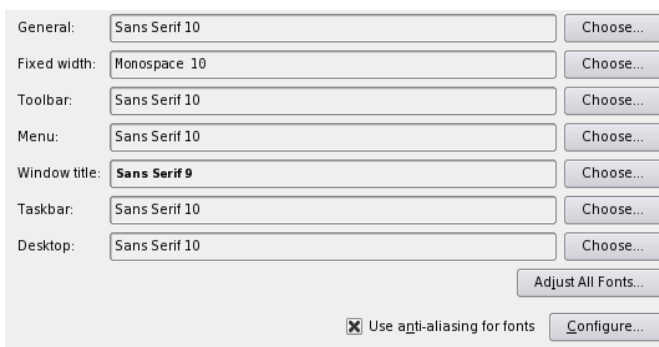
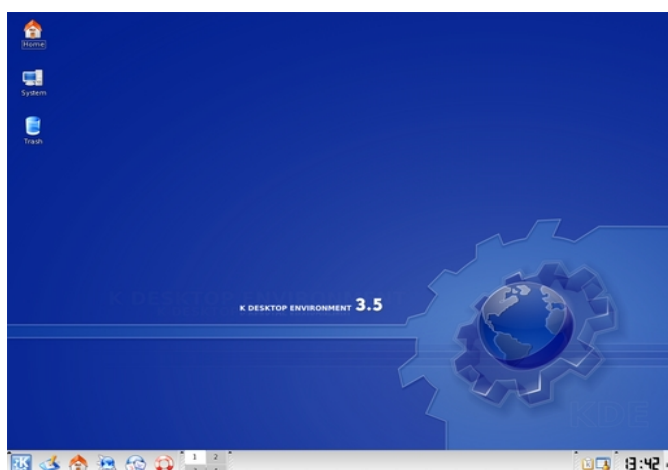
Cíl	Velikost	Popis
kde/kdebase	56MB	Desktop, prohlížeč, editory a terminál
meta/kdepim-crypto	68MB	Navíc obsahuje kdepim a kryptovací knihovny
meta/kde	172MB	Všechny balíčky „KDE 3.5 Beta“ a jejich závislosti
meta/everything	248MB	Nainstaluje všechny stabilní balíčky

Pokud chcete českou lokalizaci, potom si ještě nainstalujte `i18n/kde-i18n-cs/`. Pro kompilaci je nutné mít nainstalovány potřebné balíčky, včetně vývojových verzí. Jejich seznam naleznete na www.kde.org/info/requirements/3.5.php [3]. Ruční kontrola je velmi otravná, ale uživatelé distribucí Mandrake a SUSE mohou použít nástroj *Detektive*. Spuštěním příkazu `cd misc/detektive;make` dojde ke kontrole potřebných balíčků ve vašem systému. Pokud používáte jinou distribuci, můžete pomoci s jeho rozšířením. Po instalaci je vhodné nastavit takto systémové proměnné:

- `export QTDIR=~/.kde3.5-beta1`
- `export KDEDIR=~/.kde3.5-beta1`
- `export KDEDIRS=~/.kde3.5-beta1`
- `export LD_LIBRARY_PATH=~/.kde3.5-beta1/lib`
- `export PATH=~/.kde3.5-beta1/bin:$PATH`
- `export KDEHOME=~/.kdetest`

První spuštění

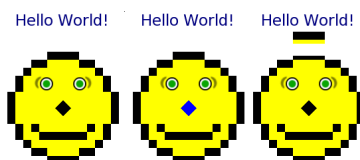
Vývojáři KDE opět změnilí úvodní obrazovku. Možná je to proto, aby byl rozdíl mezi novým a starým KDE vidět na první pohled. Ale většina distributorů ji stejně změní na svoji. Výchozí vzhled po spuštění vypadá následovně.



Výchozí písmo nezvládá diakritiku. Já používám Dejavu fonts, které nemají s nabodeníčky žádný problém a na všech dalších screenshots bude použito právě toto písmo.

Vylepšený Konqueror

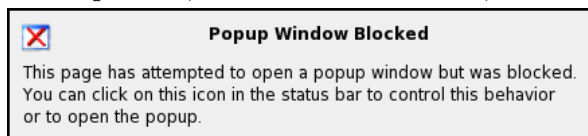
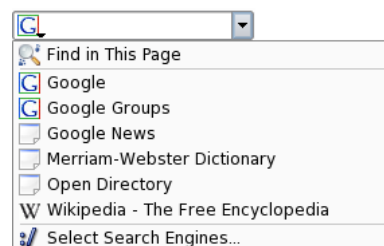
Už nějaký čas používám Konqueror jako výchozí prohlížeč, proto jsem byl hodně zvědavý, jak se povedlo implementovat vylepšení vykreslovacího jádra KHTML, které, podobně jako vývojová verze Safari, projde testem Acid2 [4]. Doufejme, že uvolnění Webkitu jako *opensource* [5], usnadní vývojářům KDE začleňovat opravy a vylepšení od vývojářů Apple. Po navštívení odkazu se mi objevil správně vyrenderovaný *ksichtík*, který dokonce reaguje na pohyb myši změnou barvy nosu. Ale při změně velikosti okna se trochu posune.



Do verze 3.5 se již dostala toužebně očekávaná funkce *AdBlock*, ale z toho, co jsem viděl, mám poněkud rozpačitý dojem. Funkce se zapíná v kontextovém menu (stejně jako v prohlížečích rodiny Mozilla), ale funguje pouze pro obrázky, není možné vypnout Flash. Další podivností je, že zablokovaný obrázek nahradí červenou značkou ala zákaz stání. Vypadá to opravdu strašně!

Tímto samozřejmě výčet vylepšení Konqueroru nekončí. Do Konqueroru byl začleněn Search Bar Plugin, což není nic jiného, než známá vyhledávací lišta z Firefoxu. Ve vyhledávací liště je už možné mít více vyhledávačů (dosud byl pouze jeden) a komu se tato vymyšlenost nelíbí, může ji snadno vypnout v menu *Nastavení/Rozšíření*.

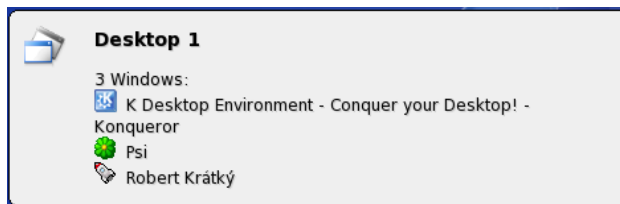
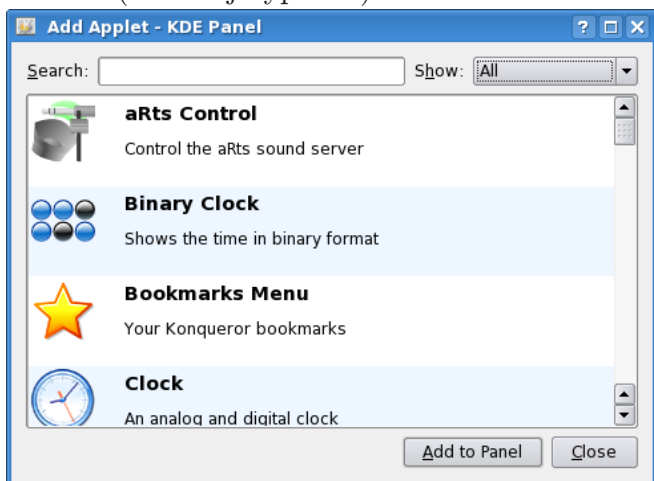
Vylepšena byla také funkce blokování pop-up (vyskakovacích) oken. Když prohlížeč zablokuje okno, tak se mimo malé ikonky objeví i tato zpráva. Projel jsem všechny testy na www.popuptest.com/ [6] a Konqueror nepustil nic. Pouze triky, které využívají Java applety, jsem nemohl otestovat. Nově byl rozšířen i prohlížeč manálových stránek, zejména index je přehlednější. Další zajímavé rozšíření Konqueroru je Metabar, boční panel, který zobrazuje definované akce, přiřazené aplikace, informace o souboru, ... A maličkost na závěr, Konqueror má nové logo.



Ostatní vylepšení

Protože nejen Konquerorem živ je člověk, je načase uvést další novinky. Panel Kicker dostal nový dialog *Add Applet* (přidat applet), který byl převzat z konkurenčního prostředí GNOME.

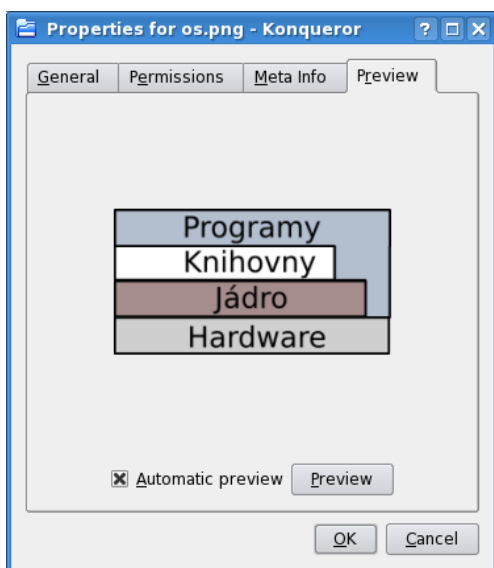
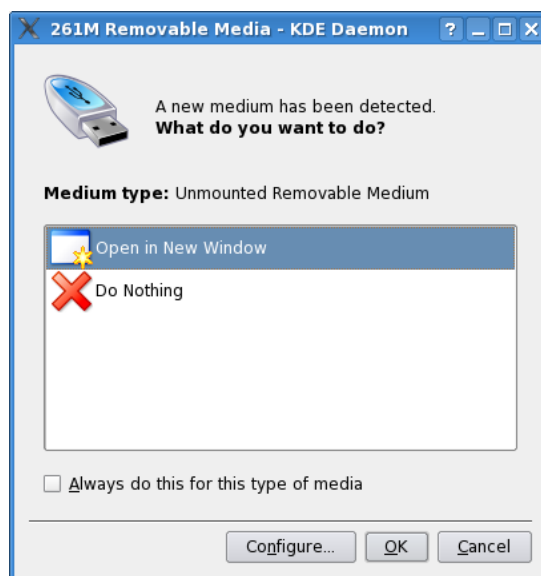
Oblíbené bubliny z KDE 3.4 byly ještě více vylepšeny, teď umí, ve spolupráci se stránkovačem, zobrazit i seznam oken z určité plochy. Bohužel jejich bublavý efekt, s nímž se zobrazují, není možné nastavit (rozuměj vypnout).



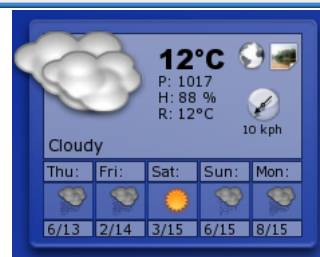
To, co jsem původně označoval za blbinku – *Keyboard Status Applet* (applet stavu klávesnice), je ve skutečnosti implementací *Keyboard Access Functional Specification* [7]. Tento applet zobrazuje stav speciálních kláves a tlačítek myši.



Dostáváme se k velmi kontroverzní funkci *Storage Media Notification* (upozornění na úložné médium). Po vzoru operačního systému Windows XP přichází i KDE s oknem, které se zobrazí po vložení některého média. Pro někoho to může být dar z nebes, pro jiné jen zbytečně otravující funkce. Odpůrce můžu ubezpečit, že se toto okno dá snadno vypnout (tlačítko *Configure* je přímo v okně), a to tak důkladně, že se mi je nepodařilo znova zapnout. Ovládací centrum se zase vrátilo ke stromovému zobrazování položek (jako výchozí nastavení), což vítám, protože režim *icon view* mi přijde nepřehledný. Souborový dialog umí i náhledy. To je podle mě zbytečná funkce, protože v KDE jsou náhledy souborů (obrázků a dokumentů) skoro všude. Snad jenom pro filmy se to může hodit, ale rychlejší je pustit si klasický přehrávač.



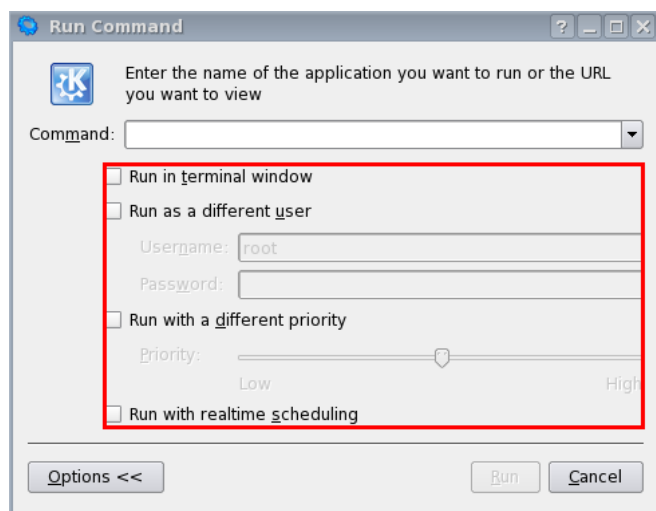
- ⊕ Appearance & Themes
- ⊕ Desktop
- ⊕ Internet & Network
- ⊕ KDE Components
- ⊕ Peripherals
- ⊕ Power Control
- ⊕ Regional & Accessibility
- ⊕ Security & Privacy
- ⊕ Sound & Multimedia
- ⊕ System Administration



Aplikace KMail má nové importní filtry: Sylpheed a The Bat!

- Import From OS X Mail
- Import Opera Emails
- Import Sylpheed Maildirs and Folder Structure
- Import Thunderbird/Mozilla Local Mails and Folder Structure
- Import The Bat! Mails and Folder Structure
- Import Outlook Express Emails
- Import Folders From Pegasus-Mail
- Import mbox Files (UNIX, Evolution)
- Import Lotus Notes Emails
- Import Plain Text Emails

Populární SuperKaramba byla nově začleněna přímo do KDE a není ji nutné instalovat zvlášť. Pokud ji ještě neznáte, tak SuperKaramba je prostředí pro spouštění interaktivních appletů, které sídlí na ploše a zobrazují rozličné, veskrze užitečné, informace.



Aplikace KSnapshot má velmi užitečnou vlastnost. Dokáže vyfotit pouze jeden určitý prvek z okna a při psaní tohoto článku jsem ji intenzivně využíval. Protože tato vlastnost nejde z pochopitelných důvodů vyfotit pomocí samotného KSnapshotu, vezměte zavděk screenshotem vytvořeným utilitkou xwd. Na obrázku vidíte vylepšené okno spuštění. Do KDE byla také začleněna nová dekorace oken Smooth Blend. Oproti všem ostatním umožňuje detailní nastavení velikosti písma v titulku, velikosti rámečku a velikosti oken. Možná časem dokáže nahradit můj oblíbený styl Quartz extra small z KDE 3.4.

Na co se nedostalo

Seznamy vylepšení jsou velmi dlouhé a není v silách jednoho člověka všechno odzkoušet, takže už jenom letecky:

- Kopete podporuje webkamery MSN a Yahoo! Ostatně seznam změn v Kopete je jeden z nejdelších, ale většina se týká zlepšené podpory právě pro MSN a Yahoo! Zlepšena práce s globální identitou.
- Balíčky z kdeedu byly masivně vylepšeny, například Kalzium. Byly začleněny nové aplikace, jako KGeography (znáte vlajku Ukrajiny, nebo Běloruska?), Kanagram a blinken.
- Panel záložek z del.icio.us [8] pro Konqueror.
- kio_file už podporuje ACL a umožňuje jejich nastavení z GUI.
- Rozpoznávání řeči v KHotKeys (vyžaduje aRts).

Hodnocení

Ač se jedná teprve o betaverzi, prostředí vypadá stabilně. Žádné pády se nekonaly. Rychlost startu nemůžu hodnotit, protože srovnání oproti prelinkovanému KDE z Gentoo, nebo oproti minimálnímu KDE z ABC Linux 2005, by pro tuto verzi dopadlo špatně. Ale odhaduji, že je srovnatelná s běžnou binární distribucí.

Velkým plusem jsou vylepšení Konqueroru, především začlenění změn od Apple do KHTML. Současná implementace Adblocku se mi nelíbí a v seznamu změn nevidím, že by na ní někdo ještě pracoval. Velmi mě potěšila schopnost KSnapshotu označit celý prvek (nebo skupinu prvků), taková blbůstka, ale moc užitečná. Přejel bych na budoucí stabilní verzi KDE bych doporučil především těm, kteří používají Konqueror a využijí jeho vylepšení.

Odkazy

- [1] <http://developer.kde.org/build/konstruk/>
- [2] <http://ftp.fi.muni.cz/pub/kde/unstable/3.5-beta1/>
- [3] <http://www.kde.org/info/requirements/3.5.php>
- [4] <http://www.webstandards.org/act/acid2/test.html>
- [5] <http://webkit.opendarwin.org/>
- [6] <http://www.popuptest.com/>
- [7] <http://accessibility.freestandards.org/a11yspecs/kbd/AccessX-func-spec.html>
- [8] <http://del.icio.us>

Jabber a ICQ: Imcom a mICQ

Matej 'Yin' Gagyí

Akého IM klienta používate na komunikovanie so svojimi známymi? V dnešnej dobe väčšina ľudí používa niektorého zo zástupcov grafických aplikácií. Prečo by sa teda niekto zaujímal o textové alternatívy? Vždy môžu nastať komplikácie a X prestanú fungovať. Alebo chcete klienta pustiť na školskom serveri v screene. Alebo vás jednoducho otravujú okienka navyše, tak ako mňa. V každom prípade dôvod sa nájde.

CenterICQ [1] je známa alternatíva. V konzole vykresľuje prehľadné okná pomocou knižnice ncurses a podporuje rad protokolov. Práca s ním je jednoduchá a intuitívna. To sú dôvody, prečo o ňom netreba rozprávať.

mICQ

mICQ je ICQ klient s príkazovým riadkom. Je napísaný v C-čku a má podporu pre skriptovací jazyk TCL. Podporuje SSL, UTF-8, ICQ protokol verzie 8 a SOCKS5 proxy.

Komunikácia pomocou mICQ sa výrazne líši od komunikácie, na ktorú ste zvyknutí. Každý úkon musíte vykonávať pomocou príkazov. Posielanie správy, vyhľadávanie používateľa, spravovanie kontaktov, to všetko sa môže zdať zo začiatku neuveriteľne ťažké. Keď sa však jednotlivé príkazy naučíte, príde vám to jednoduché.

Správy, tak ako prichádzajú a odchádzajú, zostávajú zobrazené v termináli. Jednotlivé kontakty nemajú priradené vlastné okná, na ich rozlíšenie si musíte všimnúť výpis mICQ.

```
yin@yin ~ $ micq
  _/ \_
 / \ / \
 >--o--<
 \_ / \_
  \m/CQ
mICQ (Matt's ICQ clone) version 0.5.0.4 (2005-06-04 22:27:23 UTC)
(c) 1998-2000 Matthew D. Smith, (c) 2001-2005 Ruediger Kuhlmann.
Released under version 2 of the GNU General Public License (GPL v2).
This program was made without any help from Mirabilis or their consent.
No reverse engineering or decompilation of any Mirabilis code took place to make this program.

Warning: Locale setting C is no real locale and probably not what you want.
This console uses encoding US-ASCII.
No translation requested. You live in nowhere-land, eh?
Opening v8 connection to login.icq.com:5190 for 320670287...
Opening peer-to-peer connection at localhost:0... ok.
Opening scripting FIFO at /home/yin/.micq/remote-control... ok.
Redirect to server 205.188.9.244:5190...
19:14:39 online
Local: Group contacts-icq8-320670287 (#0)
Server: Group Top-Level Contact 123456789 (#19)
Server: Group Buddies Contact 987654321 (#3)
Server: Group ? Contact 741852963/741852963 (#30351) <intimate>
Local: Group work Contact 741852963/Miso (#0)
Server: Group ? Contact 369258147/369258147 (#21257) <intimate>
Local: Group work Contact 369258147/Mirec (#0)
Local: Group Friends Contact 321654987/guru (#14125) <normal>
Local: Group (none) Contact 987654321/looser (#26456) <normal>
Differences in 1 contact groups, altogether 6 contacts, 0 ignored, 0 hidden, 0 intimate, 2 local.
19:14:42 320670287 Your status is online.
=====
Looser      (online)      [Kopete 0.10]
=====
work
=====
Mirec      (not available) [libicq2000]
Miso      (not available) [libicq2000]
=====
Friends
=====
Buddies
=====
Top-Level
=====
mICQ> █
```

mICQ nie je pre pohodlných ľudí a možno ani pre tých aktívnejších. mICQ je síce naozaj nepohodlné riešenie, ale má aj jedinečné výhody. Pri vyťukávaní príkazov a rozoznávaní kontaktov v jednom okne terminálu si precvičíte pamäť, orientáciu, strojopis a vaši kamaráti vás začnú brať naozaj vážne. Ak chcete vedieť viac, pozrite sa na [domovskú stránku mICQ \[2\]](#).

Pre úplnosť uvediem niekoľko základných príkazov:

help

Vypíše príkazy, ktoré mICQ pozná.

e, ee, eg, eeg

Vypíše používateľov, ktorí nie sú offline. **e** je stručný výpis, **ee** je široký výpis. Koncovka **-g** zaradí kontakty do skupín.

w, ww, wg, wwg

Vypíše všetky kontakty. Platia rovnaké princípy výpisu, ako pri **e**.

msg *kontakt správa*

Pošle správu kontaktu.

a

Pošle správu kontaktu, ktorému ste posielali správu naposledy.

r

Pošle správu kontaktu, ktorý vám naposledy napísal.

h *kontakt alebo history kontakt*

Vypíše logy rozhovorov s uvedeným kontaktom.

f *kontakt, finger kontakt*

Zistí o kontakte jeho osobné informácie uvedené na serveri.

auth *kontakt*

Povolí používateľovi za kontaktom vidieť váš stav (online, away, ...).

Imcom

Imcom je klient pre otvorený IM protokol Jabber. Je podobný mICQ, ale je písaný v Pythone (aj jeho moduly sú písané v Pythone). Podporuje SSL a mnoho ďalších funkcií, ktoré Jabber ponúka.

Pre Imcom platia podobné pravidlá ako pre mICQ. Práca s ním je ťažká, aj keď pracovať s Jabberom pod grafickým klientom sa mi zdá ťažšie. Stačí si zapamätať niekoľko príkazov a všetko je hneď poruke; nie v dialógu, ktorý otvoríte z menu, cez nejaký nástrojový panel. Len stlačíte 3–4 klávesy a je to.

Neprijemnou skutočnosťou je to, že Imcom je veľmi slabo vyvíjaný. Dokonca na jeho oficiálnej stránke sa nachádza niečo úplne iné, ako by sa malo. Na druhej strane je príjemné, že v máji vyšla po dlhom čase nová verzia – 1.34 a zdrojové kódy sa objavili na adrese <http://nafai.dyndns.org/files/> [3]. Na tomto mieste by sa mali v budúcnosti objavovať aj ďalšie verzie Imcomu.

Znovu si dovoľím uviesť niekoľko príkazov:

/help

Vypíše všetky príkazy Imcomu, aj so stručným vysvetlením.

e

Vypíše všetky kontakty, ktoré nie sú offline.

w

Vypíše všechny kontakty.

`/msg kontakt správa`

Pošle správu kontaktu.

`// správa`

Pošle správu kontaktu, kterému ste posielali správu naposledy.

`/r správa`

Pošle správu kontaktu, ktorý vám naposledy napísal.

`/log kontakt`

Vypíše logy rozhovorov s uvedeným kontaktom. Pozor, Imcom rozlišuje, či zadávate JID alebo nick.

`/info kontakt`

Zistí o kontakte jeho osobné informácie uvedené na serveri.

`/auth kontakt`

Povolí používateľovi za kontaktom, vidieť váš stav (online, away, ...).

`/agents`

Vypíše zoznam služieb a transportov na vašom Jabber serveri.

`/agenthelp URL služby`

Začne registráciu na uvedenú službu vášho Jabber servera.

Jabber

Dovolím si v skratke predstaviť Jabber. Jabber je, ako som už povedal, open source IM protokol vyvíjaný v [Jabber Foundation](#) [4]. Na rozdiel od protokolu ICQ nie je ovládaný jednou komerčnou firmou a jeho smerovanie riadi komunita.

Jabber je založený na XML jazyku a je mimoriadne rozšíriteľný. Cez Jabber môžete ľahko a jednoducho pristupovať aj do sietí ICQ, MSN, Yahoo!, IRC a ďalších. Okrem toho ponúka Jabber množstvo ďalších rozšírení.

Google si vybral Jabber ako základ vlastnej IM siete a zostáva s ním ďalej kompatibilný. O Google Talk ste sa už mali možnosť dozvedieť zo [správičky](#) [5].

Záver

Dva nenáročné IM klienty Imcom a mICQ boli predstavené. Dúfam, že sa vám ich služby zídu nielen vo chvíľach, keď si nič iné nebudete môcť dovoliť. Mne v každom prípade učarovali a komunikovanie pomocou niečoho iného si už ani nedokážem predstaviť. Oba uvedené programy môžete rozšíriť pomocou modulov. Týchto modulov je na internete veľmi málo. Ak bude záujem, napíšem článok o písaní modulov pre Imcom alebo potom pre mICQ. A možno napíšem aj ukážkové moduly podľa prianí čitateľov.

Poznámky:

- Posielanie viacriadkovej správy – Ak nenapíšete správu, ktorú chcete poslať, objaví sa výzva na písanie viacriadkovej správy. Keď správu dopíšete, na prázdny riadok vložte bodku `␣`. Ak chcete správu zrušiť, na prázdny riadok vložte mriežku `#`.
- Kontakt – V Imcome kontakt môže byť JID (Jabber ID) používateľa alebo nick v rosteri. V mICQ je kontakt ICQ číslo (ICU) alebo nick v zozname kontaktov.

```
yin@yin ~ $ imcom
IMCom version 1.33 final (less bugfixes) is released under the BSD license.
See LICENSE for details
Written by Casey Crabb (JID: airog@floobin.cx)
Logged on
Roster has been downloaded
ic.marulienko/centericq changed status to xa (som doma
) at 13:55:56
ICQ Service/registered changed status to online (online) at 13:55:58
pflow.miso.icq changed status to xa (online) at 13:55:58
pflow.shareman.icq changed status to online (online) at 13:55:58
upjs.schmotzer.icq changed status to xa (online) at 13:55:58
pflow.renkaf.icq changed status to online (online) at 13:55:58
pflow.fefe.icq changed status to online (online) at 13:55:58
pflow.skipi.icq changed status to online (online) at 13:55:58
def.schrapnel.icq changed status to online (online) at 13:55:58
pflow.gabi.icq changed status to xa (online) at 13:55:58
abc.robert.icq changed status to online (online) at 13:55:58
pflow.martin_grafcik.icq changed status to online (online) at 13:55:58
wiki.liso.icq changed status to online (online) at 13:55:58
pflow.mato_tomaga.icq changed status to online (online) at 13:55:58
pflow.jejka.icq changed status to online (online) at 13:55:58
pflow.gabec.icq changed status to online (online) at 13:55:58
pflow.miki_.icq changed status to online (online) at 13:55:58
pflow.dreamer.icq changed status to away (online) at 13:55:58
pflow.rjasko.icq changed status to online (online) at 13:55:58
pflow.gabca.icq changed status to online (online) at 13:55:58
pflow.mirec.icq changed status to online (online) at 13:55:58
def.random.icq changed status to online (online) at 13:55:59
upjs.schmotzer.icq is unavailable : Disconnected at 13:56:38
IMCom> █
```

Odkazy

- [1] <http://www.abclinuxu.cz/clanky/recenze/centericq-komunikacni-centrala>
- [2] <http://micq.org/>
- [3] <http://nafai.dyndns.org/files/>
- [4] <http://www.jabber.org/>
- [5] <http://www.abclinuxu.cz/zpravicky/googletalk>

Reportáž: Blender Conference 2005, De Waag, Amsterdam

Vítězslav Válka

Poslední dobou pozoruji, že „OS“ přestává být operačním systémem a začíná být zkratkou pro Open Source. Přesvědčila mě o tom konference o Blenderu, který je – jak jinak – Open Source také, přestože jeho cesta byla poněkud trnitá.

Již čtvrtý rok se v Nizozemí, tedy zemi, kde se Blender zrodil, konalo setkání vývojářů a uživatelů této aplikace. Použití Blenderu je zakotveno v jakémsi mnohostranném nástroji nejen pro tvorbu virtuální reality, ale i dalších vazeb na realitu samotnou. Pokud se však vrátíme k samotné konferenci, ta by se dala vnímat ze dvou hlavních pohledů.

Pohled uživatele

Díky několika přednáškám současně probíhajícím v prostoru historické budovy De Waag (kde kdysi kreslil i Rembrandt) přímo v centru kanály propleteného Amsterdamu, jste mohli i jako nezaspěčený snadno pochopit, že Blender neúprosně rychle roste a přestává být oním podivným a nespolehlivým „Open Source“. Ale každým dnem mnohem více profesionálním nástrojem nejen pro rendering statických vizualizací, ale i propojovacím článkem několika zdánlivě nesouvisejících disciplín. Během příštích let tak podle mého názoru začne směle ohrožovat upevněné pozice dnes zavedených aplikací jako je Maya nebo 3D Studio.



Pohled vývojáře

Samotný Ton Rosendaal, duchovní otec celé aplikace, vedl hned několik přednášek. Jedna z nich, která mě zaujala nejvíce, byla o budoucnosti Blender Foundation a Blenderu jako učivu na školách. Přestože se řešila spousta konkrétních dotazů, proč a jak například certifikovat učitele, nebo jak sdružovat profesionální autory prostřednictvím úzce profilovaných webových stránek, klíčová byla následující otázka: „Co by měla Blender Foundation v budoucnu představovat? Komu by měla sloužit a jaká by měla být její náplň?“ Odpověď však určitě není jednoduchá. Pozitivní však rozhodně byl můj subjektivní dojem, že Blender jako takový je na vzestupu. Nadace získala nejen spoustu peněz díky prodeji v Online shopu, ale například i díky několika grantům EU.

Podle výše uvedeného soudím, že je dobrá doba pro jakýkoliv perspektivní Open Source, protože díky současné podpoře těchto projektů ze strany státu, EU nebo jiné instituce může být zajištěn provoz takovéto zastřešující organizace, což je samozřejmě klíčové pro řízení nejen tohoto projektu, ale kterékoliv jiné organizace.

Jak asi víte, tak v současné době je jediným fulltime zaměstnancem Blender Foundation právě Ton Rosendaal, dále pak několik externistů jako je např. Angela, která má na starosti oblast managementu nadace. Když k tomu nakonec doplníme, že je tu Blender již více než deset let, a jeho otevřený vývoj jde s každou verzí raketově kupředu, je to jednoznačně záruka toho, že nyní mluvíme o přicházející revoluci na poli 3D modelingu, minimálně v oblasti ovlivnění cenotvorby a marketshare v oblasti takových aplikací.

Zajímavostí bylo například i to, že na konferenci měl přednášku perspektivní český programátor pan Jiří Hnídek, který se aktivně podílí na vývoji tzv. Verse Blenderu. Ten má za cíl integrovat do

aplikace protokol pro síťovou spolupráci více uživatelů na jednom projektu a jeho demo předvedl p. Hnídek přímo na konferenci.

Nejen počítačové obrazovky

Za nejzajímavější ze tří dnů bych označil sobotu, kdy byla poprvé představena upoutávka na připravovaný první Open Source celovečerní film (na kterém už několik měsíců pracuje v Amsterdamu tým těch nejlepších „blenderistů“) s názvem Elephant's Dream (tedy Sen slona) a vyprávějícím o životě ve stroji. A současně se konala společná večeře v nedaleké čínské restauraci, kde jste se v pohodě mohli u jednoho stolu bavit s učitelem žijícím v Holandsku, ale pocházejícím z Anglie, Němcem, Španělem nebo Italem. V neděli byla konference příjemně uzavřena vyhlášením tzv. Susanne Award, tedy nejlepší práce (tentokrát animace) v Blenderu v posledním roce.



Pokud bych tedy měl celou konferenci shrnout, byl jsem velmi spokojen s jejím obsahem i celkovou organizací. Rozsah i zaměření přednášek bylo opravdu různorodé, a tak jste si mohli udělat o možnostech Blenderu docela slušný obrázek i v případě, že byste na konferenci přišli jako úplný nováček.

Během dne jste navíc měli kdykoliv možnost doplnit tekutiny nebo potraviny. Byl připraven přístup k několika 17" Powerbookům, jednomu 24" PC s Ubuntu Linuxem a jednomu s Windows. Vše vybavené připojením k internetu a speciální testovací verzí Blenderu určenou právě pro konferenci. Přestože jsem tedy neuvažoval, že se na tuto akci budu vracet, tak díky úžasné atmosféře a relativně blízkému Holandsku, resp. Amsterdamu, který sám o sobě je nezapomenutelný, o tom dnes uvažuji.



Ark Linux

Filip Bartmann

Ark Linux je uživatelsky přívětivá distribuce, která je určena hlavně pro začátečníky. Recenze popisuje novou verzi 2005.2-rc3, jež obsahuje jádro 2.6.14-rc2 a betaverzi KDE 3.5.

Na webových stránkách distribuce Ark Linux máme ke stažení pět ISO obrazů: základní CD, CD s extra softwarem, CD s vývojářskými programy, CD s programy pro server a nakonec ještě CD s dalšími jazyky. Pro základní instalaci ale postačí první obraz, který však neobsahuje češtinu. Pokud chcete mít kompletně lokalizovaný systém, je potřeba si rovněž stáhnout 300 MB CD s dalšími jazyky. Na prvním CD se nenacházejí žádné balíčky, které by umožnily kompilaci SW ze zdrojových kódů, k tomu potřebujete CD č. 3. Pokud ale s Linuxem teprve začínáte, tak vám první CD stačí.

Distribuce Ark Linux je postavena na knihovně glibc 2.3.5 a na nejnovějším jádře 2.6.14-rc2, protože se jedná o vývojovou verzi 2005 rc3. Server X.Org se nachází ve verzi 6.8.2. V protikladu k těmto novinkám působí zjištění, že distribuce používá místo dnes již obvyklého udev ještě DevFS.

Instalace

Po vypálení ISO obrazu můžeme spustit instalaci, která je orientovaná hlavně na začátečníky, a pokročilým uživatelům toho nenabídne mnoho k nastavení. Nejprve se zobrazí dialogové okno s výběrem jazyka, rozložení klávesnice a časovou zónou. V dalším dialogovém okně si jenom vybereme typ instalace: plně automatickou nebo pokročilou.

Ani pokročilá instalace toho ale moc k nastavení nabídně: budeme moci pouze rozdělit disk programem `qtparted`. Samotná instalace proběhne celkem svižně, nemusíte při ní ani odcházet od počítače – i proto, že vám instalátor nabídně hru Tetris. Pokud se vám hra zalíbí a nestihnete jí dohrát do konce instalace, není to problém. Instalátor zobrazí okno s upozorněním, že instalace je dokončena a restart nechává na vás. Opravdu zajímavý nápad.

První start

Po restartu počítače na vás čeká obvyklý linuxový zavadač GRUB. Systém v grafice naboootuje do grafického prostředí KDE v nejnovější verzi 3.5. Toto prostředí již lokalizované je i při použití pouze prvního CD.

Přihlašovací obrazovka se ale nezobrazí, automaticky je bez hesla přihlášen uživatel `arklinux`. Začátečnickům to může pomoci, ale pokud se již trochu v Linuxu vyznáte a nejedná se o instalaci na domácím počítači, doporučuji vám v Ovládacím centru KDE toto zakázat. Volba se nachází v oddílu *Správce přihlášení*, na kartě *Pohodlí* a jmenuje se *Povolit automatické přihlášení*. Rovněž povýšení práv na superuživatele je možné bez zadání hesla, což lze nechat povolené pouze v domácím prostředí. Jinak v Ovládacím centru distribuce zvolte *User Accounts* a v zobrazeném dialogovém okně na záložce *User Info* klikněte na *Set password* a heslo nastavte. Na závěr je ještě třeba kliknout na účet uživatele `arklinux` a zadat zde heslo, pod kterým se chcete přihlašovat k systému.

Po prvním naboootování se zobrazí průvodce, který vám umožní si KDE nastavit. V něm vidíme, že se v distribuci nachází i vzhled Baghira, což je vlastně imitace vzhledu MacOS X. Takže pokud máte rádi nakousnuté jablko, můžete jej vyzkoušet :-).

Programy

Po rozbalení hlavního menu zjistíte, že mezi nainstalovanými programy nechybí ani OpenOffice.org v nejnovější verzi 2.0. K prohlížení internetu si však musíte vystačit s Konquerorem, Firefox byste

museli doinstalovat. Jinak balíčky z prvního CD na obvyklou práci stačí. Pokud pro instalaci systému použijete pouze první CD, tak budete také muset oželeť Evolution a používat pouze KMail. K instant messagingu lze využít program Kopete. Pro vypalování CD je zde výborný program K3b. Po jeho spuštění ale následuje zjištění, že nejsou nainstalované programy pro vypalování DVD; alespoň ty by na základním CD být ještě mohly. Pokud si je nemáte možnost doinstalovat, musíte si vystačit s vypalováním CD. Pro přehrávání multimédií je k dispozici MPlayer (verze 20051004) ve spojení s frontendem KMPPlayer. Nechybí ani skvělý zvukový server Jack, obyčejní uživatelé však spíše využijí aRts z KDE. Na počítačích s Wi-Fi hardwarem najde možná využití i nainstalovaný ndiswrapper.

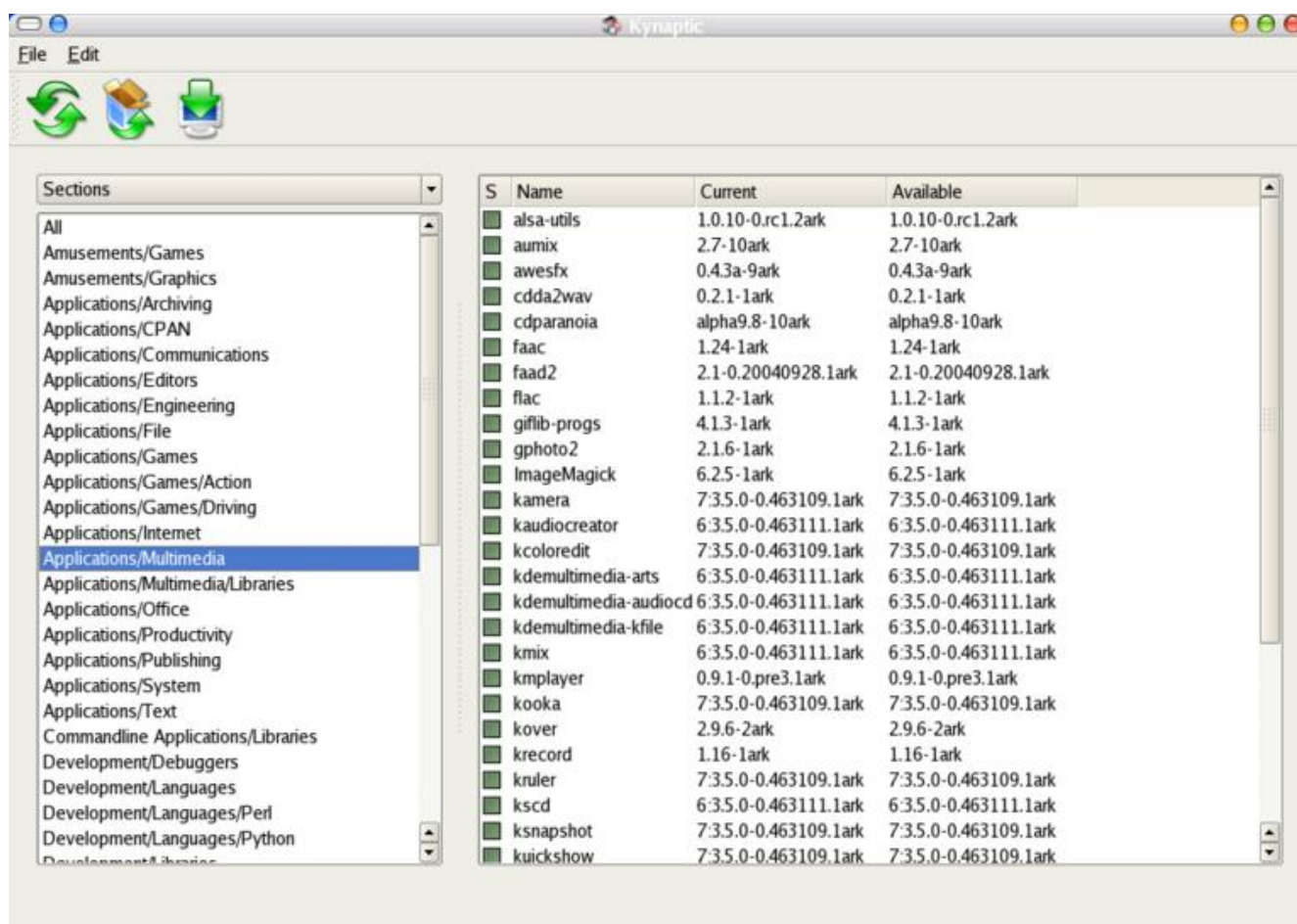


Ovládací centrum

Pod *System Settings* se ukrývá ovládací centrum celé distribuce, které nabízí většinou možnosti ovládacího centra KDE vyvedené v pěkné grafice. Můžete si tak v grafice nastavit celý počítač. Pokud si však budete chtít nastavit síť, zjistíte, že dialogové okno pouze nabídne výběr typu připojení a poté zobrazí dostupná síťová zařízení. Po náhlednutí do originální dokumentace jsem však zjistil, že lze síť nastavit i podrobněji – k tomu musíte na daném zařízení provést dvojklik; tlačítko *Pokročilejší nastavení* by neškodilo.

Balíčky

Jako program pro správu balíčků je zvolen Kynaptic, což je Synaptic z Debianu pro KDE. Balíčky jsou však ve formátu RPM, takže vlastně spojují to nejlepší z obojího. Zde jsem ale narazil na problém: nikde se v grafice nedají nastavit repozitáře s balíčky, takže pokud chcete použít jiná úložiště balíčku než ta výchozí, editaci konfiguračního souboru se nevyhnete.



Závěr

Celkově na mně distribuce zanechala dobrý dojem. I přes to, že se jedná o vývojovou verzi, tak je stabilní. Jediný problém, který jsem zaznamenal, se týkal ovládacího centra KDE, ve kterém se po zadání superuživatelského hesla místo příslušné karty zobrazila pouze úvodní zpráva centra. To je ale problém prostředí KDE, se kterým jsem se již setkal i u předchozí vývojové verze před řadou 3.4. Ve stabilní verzi to již bylo opraveno.

Na samotný závěr testování jsem se setkal ještě s jedním nedostatkem. Ve chvíli, kdy jsem chtěl použít disketovou mechaniku, jsem zjistil, že nebyla systémem rozpoznána (neexistovala v souboru `/etc/fstab`), takže začátečník v Linuxu by jí nezprovoznil, doufám že je to chyba pouze vývojové verze a ve stabilní se již neobjeví.

Pokud vám tyto chyby nevadí, můžete distribuci vyzkoušet, jinak počkejte na stabilní vydání, které by mělo být hotové brzy.

Kommander

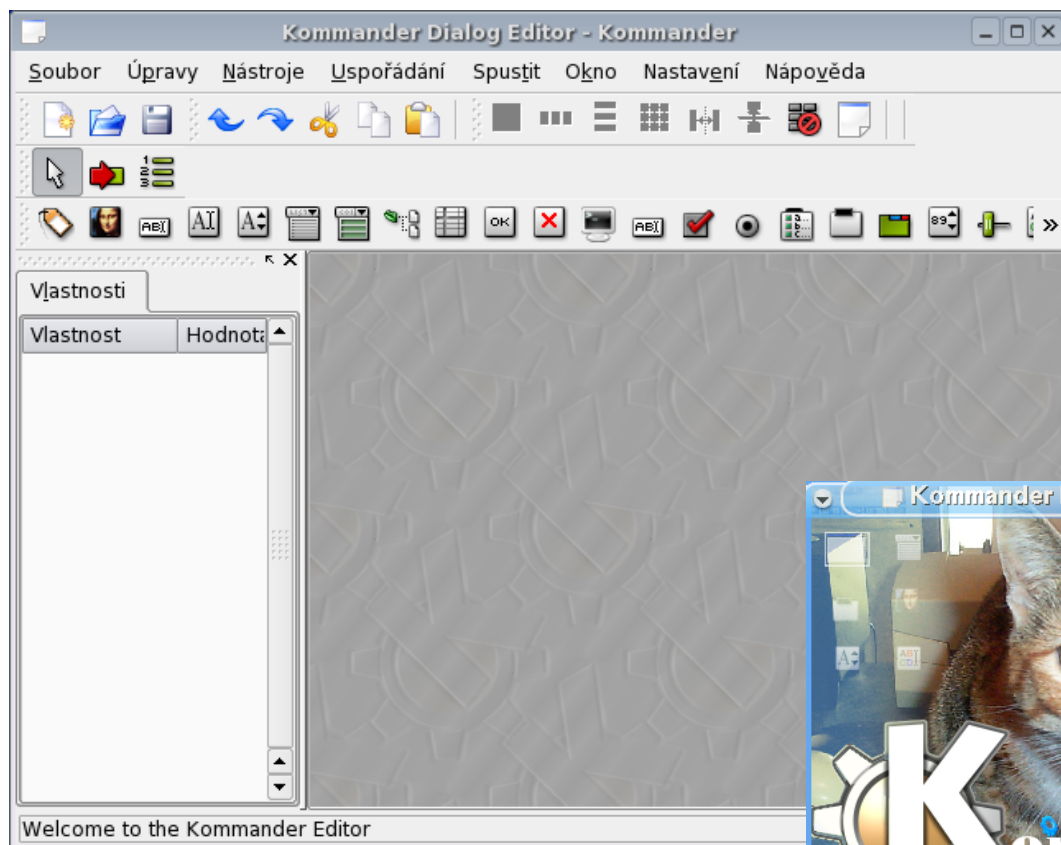
Michal Vyskočil

Kommander je nástroj pro psaní jednoduchých aplikací s grafickým uživatelským rozhraním. Seznámíme a „Hello World“. Více si rozebereme výhody a nevýhody starého parseru. Zmíním i možnost používat jiný skriptovací jazyk než shell. Nový parser rozhodně zlepšil použitelnost Kommanderu; změny, výhody, nevýhody. Plus příklady aplikací.

K čemu je Kommander

V unixových operačních systémech si můžeme pro každou netriviální, a opakující se činnost napsat skript. Pokud je určen pouze pro naši potřebu, není problém. Když ho mají používat odborníci (správci, programátoři, ...), stačí napsat dokumentaci, nebo manuálové stránky. Jenže pokud má sloužit i běžným uživatelům, nastává problém, protože mají ti obvykle strach z příkazové řádky a dokumentaci nikdy nečtou.

A vývojáři KDE přišli s řešením. Kommander je přímo určen pro psaní jednoduchých GUI aplikací. Díky technologii DCOP (viz [KDE: tipy a triky – III \(DCOP a KDialog\)](#) [1]), kterou Kommander vnitřně používá, je možné ze skriptu ovládat grafické prvky – widgety. Také nám odpadají problémy s binární kompatibilitou knihoven, od toho nás odstiňuje Kommander. Celý projekt je uložen v jednom SGML souboru, takže jeho instalace je velmi jednoduchá. Týká se to i binárních souborů, například obrázků, které jsou uloženy v souboru v textové podobě. Parser Kommanderu funguje jako makrojazyk a bohužel má i spoustu velmi nepříjemných omezení.



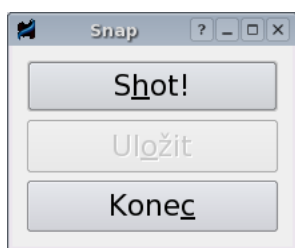
Starý vs. nový parser

Od Kommanderu verze 1.2, který je součástí KDE od verze 3.4 je k dispozici nový parser, který se velmi liší od toho starého. V dalším textu budu mluvit o tom starém. Nový parser je relativní novinka a většina aplikací na kde-apps.org [2] používá ten starší, nemluvě o tom, že je použitelný až od KDE verze 3.4. Nový parser bude tématem dalších dílů.

Dost už bylo textu, je na čase uvést nějaký obrázek. Kommander se sestává ze dvou programů. Kommander Executor (`kmdr-executor`), který provádí skripty uložené v souborech `kmdr`. Druhým je Kommander Editor (`kmdr-editor`), vývojový nástroj, v němž se navrhují dialogy a píše kód.

Při spuštění se můžeme podívat na obrázek kočky, ale moc dlouho ji nevidíte, protože Editor startuje velmi rychle. Pokud jste někdy psali projekt pro KDE, nebo Qt, bude vám jeho prostředí připadat velmi známé. Je to totiž upravený (a zjednodušený) Qt Designer od Trolltechu. Soubor se skriptem je prakticky totožný, s `ui` souborem, který generuje Designer.

Hello, world!



Přeskočme teorii a přejdeme rovnou k praxi. Od dob K&R je tradicí, že prvním příkladem je aplikace, která vypisuje *Hello, world!*. Já se rozhodl tuto tradici porušit, a to ze dvou důvodů. Jednak je tento příklad nevhodný pro předvedení Kommanderu. Za druhé považuji čtenáře abclinuxu.cz za tak inteligentní, aby pochopili i složitější příklad: aplikaci pro snímání oken z obrazovky – `kommander-1-snap.kmdr` [3].

Všichni jistě znáte způsob, jak vyfotit v X Window okno (viz [Snímání obrazovky aneb vyfoťte si okno](#) [4]). Jenže příkaz `xwd` ukládá do stejnojmenného formátu. Pokud chceme rozumný, všeobecně používaný formát PNG, celý proces se nám roztáhne na takovou kolonu příkazů:

```
xwd | xwdtopnm | pnmtopng > foo.png
```

Geekové žasnou, ale běžného uživatele tím asi moc nepřesvědčíte. Proto tento kód přiřadíme tlačítku *Shot!*. V kontextovém menu tlačítka zvolíme *Edit Kommander Text* a do okna tu kolonu napíšeme.

Jak prosté! Teď máme jednoduchou grafickou aplikaci, která vyfotí aktuální okno a uloží do souboru `~/foo.png`. Akce je přiřazena tlačítku jménem `BtnShot`, typu `ExecButton`. Tohle řešení není moc dobré, uživatel by měl mít možnost si vybrat jméno souboru a jeho umístění. Kód je nutné trochu modifikovat:

```
@setGlobal(IMG,@exec(tempfile))
xwd | xwdtopnm | pnmtopng > @global(IMG)
@BtnSave.setEnabled(true)
```

Co jsme teď udělali? Za prvé neukládáme obrázek do domácího adresáře, ale do souboru, který vytvoří příkaz `tempfile` v adresáři `/tmp`. Jméno nově vytvořeného souboru je důležité, a protože s ním budeme pracovat i v dalších částech kódu, uložíme si jej příkazem `@SetGlobal()` do globální proměnné `IMG`. Ve druhém řádku již přesměrováváme výstup do proměnné `IMG`, jejíž hodnotu vrací `@global()`. No a v posledním příkazu povolíme tlačítko `BtnSave`, protože už máme obrázek, takže můžeme ukládat.

Syntaktická odbočka

Jak jste si všimli, tak příkazy Kommanderu začínají znakem `@`. Stejným znakem začínají nejen příkazy, ale i jména grafických objektů a dokonce i klíčová slova. Alespoň ve starém parseru to tak je. Tlačítku `BtnSave` potom přiřadíme tento kód.

```
cp @global(IMG) @Input.savefile(~, *.png,"Uložit screenshot")
```

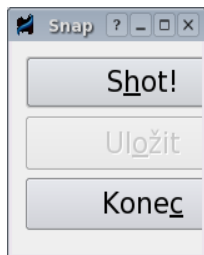
Ten zkopíruje soubor `img` na to místo, které vrátí dialog `@Input.savefile()`, Prvním parametrem je výchozí adresář, druhým filtr souborů a posledním je titulek dialogu. Kód tlačítka `BtnExit`, které je typu `CloseButton` už uvedu jen pro pořádek.

```
rm -f @global(IMG)
```

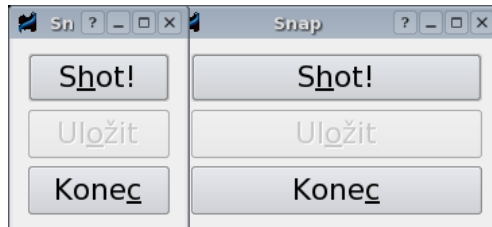
Tím jsme získali jednoduchou aplikaci na tvorbu screenshotů obrazovky. Její největší nevýhodou je špatná správa souborů v `/tmp`, které se ne vždy smažou. Každý obslužný událostní skript běží nezávisle na ostatních, takže příkaz `trap` nám zde nepomůže. Ale ilustrační aplikaci to můžeme odpustit.

Správci rozložení

Není nic smutnějšího, než když widgety nedokáží reagovat na změnu velikosti okna a aplikace potom dopadne nějak takto:



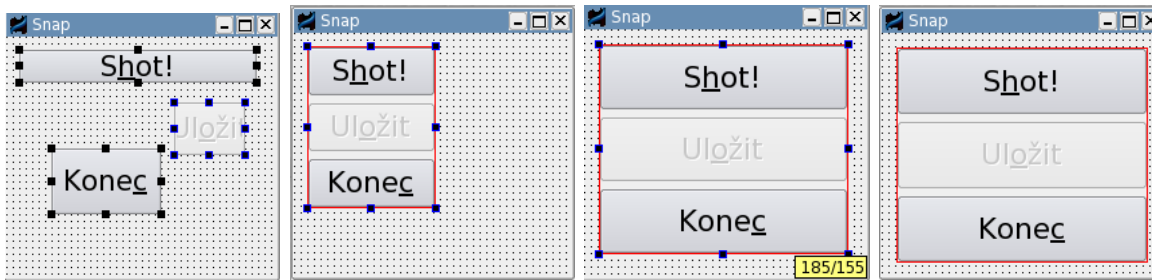
Snap se pochopitelně chová správně:



Kdysi dávno jsem měl tu čest vidět aplikaci ve Visual Basicu, kde se o rozložení widgetů v okně staral přímo programátor. Ještě dnes si vzpomínám na ta kvanta kódu, která musela být napsána a taky na to, že žádosti o změny uživatelského rozhraní byly předem odmítnuty. Nevím, zda dotyčný jenom nevěděl, anebo to skutečně nešlo. Každý lepší toolkit obsahuje svého správce rozložení, který se o rozložení widgetů stará automaticky. Kommander používá stejný toolkit jako KDE, a proto i správce rozložení jsou ty, které používá Qt.

- vertikální rozložení
- horizontální rozložení
- rozložení do mřížky
- vertikální oddělovač mezi dva widgety
- horizontální oddělovač mezi dva widgety

V našem příkladě vidíme, že jsou všechna tlačítka pod sebou, takže je můžeme libovolně naskládat na formulář, označit, dát vertikální rozložení a ony se srovnají, jak potřebujeme (jenom změníme jejich velikost na požadovanou). Nakonec ještě vybereme layout pro celý formulář (v tomto případě je to úplně jedno, který vybereme) a máme pěkně fungující formulář, který reaguje na změny velikosti okna a bez jakéhokoliv řádku kódu.



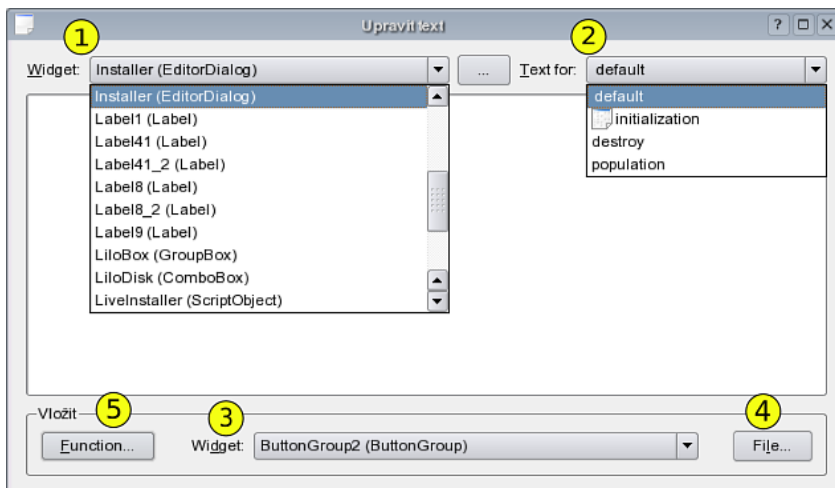
Chování widgetů se dá ještě detailněji nastavit pomocí vlastnosti *Size Policy*.

Rozpěrky

V mnoha případech nechceme, aby byl jeden prvek těsně u druhého. Proto potřebujete nějakým způsobem správci rozložení říci, jak se má chovat. Toto zabezpečují *spacers* (překládám jako rozpěrky). Může se stát, že budeme chtít, aby byla mezi tlačítkem *Konec* a zbytkem mezera. Takže přidáme rozpěrku mezi *Konec* a *Uložit* a mezera mezi nimi je na světě.

Podobně jako mají widgety *Size Policy*, jsou všechny rozpěrky nastavitelné.

Vkládání příkazů



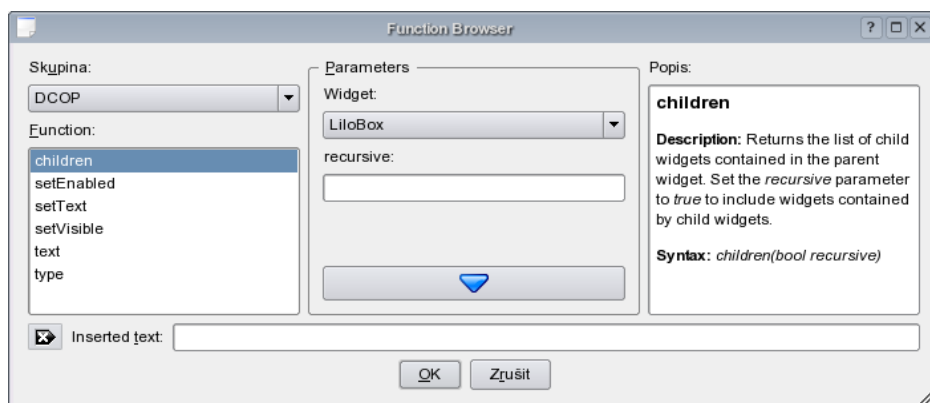
Skript v Kommanderu je běžný textový SGML soubor. Můžeme jej tedy editovat ve svém oblíbeném editoru (například ve vimu), Většina lidí ale asi využije spíše prostředí Kommander Editoru. Příkaz *Edit Kommander Text* nás zavede do editačního okna, které jsme viděli v předchozím dílu (*Kommander – 1 (Skriptované GUI)* [5]). Teď si je popíšeme

1. Zde vybereme widget, pro který budeme psát obslužný skript. Pokud nám nevyhovuje rozbalovací seznam, tak vedlejší tlačítko zobrazí widgety ve formě stromu.
2. Seznam *Text for* vybírá akci, při které dojde ke spuštění skriptu. Prvek *Installer* má vyplněnou akci `initialization`. To znamená, že daný kód se provede při inicializaci okna formuláře.
3. Tento prvek by se měl jmenovat spíše *Insert widget*, protože slouží ke vložení jména widgetu do textu.
4. *File* zobrazí dialog, v němž vybereme soubor, jehož obsah se vloží do našeho skriptu
5. Pravděpodobně nejdůležitější tlačítko (především pro začátečníky) je *Function...*, které zobrazí seznam všech funkcí a příkazů Kommanderu. V něm si každý může najít potřebné funkce, včetně jejich popisu.

Toto okno je velmi praktické, protože vám umožňuje snadno proniknout do hlubin Kommanderu a jeho funkcí. Příkazy jsou rozděleny do sedmi skupin:

- *Array* – funkce pro práci s poli.
- *DCOP* – což jsou funkce pro obsluhu widgetů. Každý widget má skupinu funkcí, kterým rozumí. Viz KDE: tipy a triky – III (DCOP a KDialog).
- *File* – funkce pro práci se soubory.
- *Input* – zobrazuje vstupní dialogy (v minulém díle jsme viděli příkaz `savefile`).

- *Kommander* – klíčová slova a příkazy Kommanderu – (`@if`, `@exec`, ...).
- *Message* – zobrazuje okna se zprávami.
- *String* – funkce pro práci s řetězci.



Ve sloupci *Parameters* vidíme parametry dané funkce a poslední sloupec obsahuje krátký popis funkce a její syntaxe. Tlačítko s velkou modrou šipkou pak vloží text do spodního textového pole a stiskem *OK* dojde k vložení funkce do kódu. Vzhledem k této skutečnosti je zbytečné psát referenční příručku ke Kommanderu, vše potřebné máte v okně *Functions Browser*. Ale pokud byste na ní trvali, tak ji naleznete na docs.kde.org [6]. Dále raději píšu o věcech, které takto nezjistíte – a některé z nich mi daly pořádně zabrat, než jsem na ně přišel.

Akce

Každý widget má množinu akcí, které může vykonat. Nejčastější akce jsou `default` a `population`. Nejčastěji používanou je právě `default`, která třeba u tlačítka značí jeho stisknutí. Formuláře navíc mají `initialization` a `destroy`, které se provádí při startu a ukončení aplikace. Akce `default` se spouští při zavolání prvku. Nejjednodušším příkladem je rozbalovací seznam `ComboBox1`, který v této akci má napsáno `@widgetText`. Tím dojde k tomu, že budou následující ekvivalentní.

```
@ComboBox1
@ComboBox1.text
```

Samozřejmě může být v oné akci napsán jakýkoliv kód, například transformace textu na něco jiného. Check Boxy (přepínače) a Radio buttony (skupinové přepínače) nemají akci `default`. Namísto toho mají `checked`, `semichcked` a `unchecked`. Kód se spouští, pokud je tlačítko vybráno, napůl vybráno (pokud je třetí stav povolen) nebo nevybráno. Díky tomu nemusíme stav tlačítek zjišťovat, stačí potřebný kód napsat do odpovídajících sekcí.

Ukázková aplikace

Původně jsem chtěl pro demonstraci vlastností jazyka Kommanderu použít některou z existujících aplikací, ale nakonec jsem si ji musel sám napsat. Nedělá sice nic užitečného, ale je jednoduchá a přizpůsobena tomuto seriálu. Můžete si ji [stáhnout](#) [7] a vyzkoušet.

Proměnné

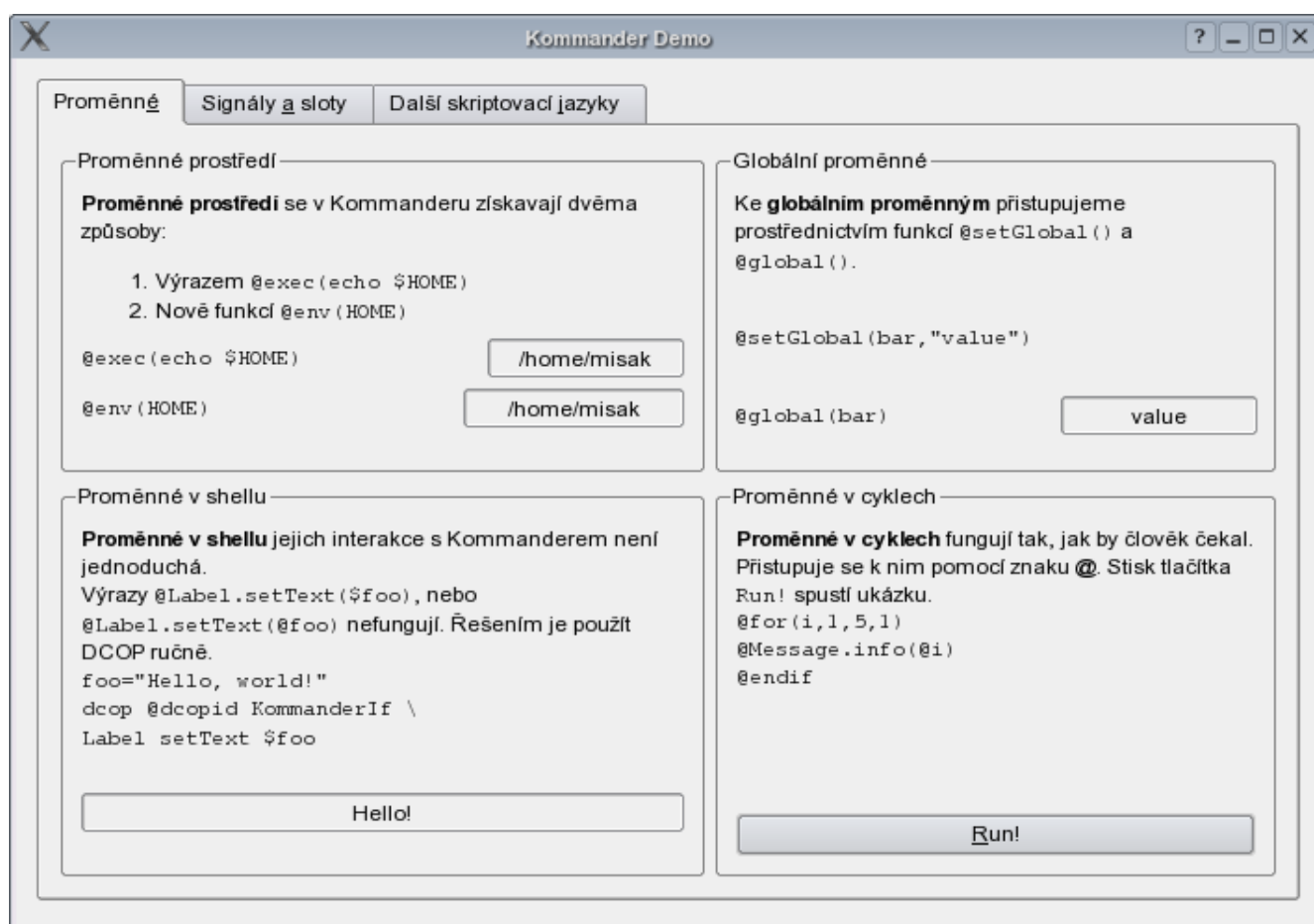
Pokud, stejně jako já, považujete proměnné za základ všech (tedy alespoň procedurálních) programovacích jazyků, pak asi zažijete menší šok. Práce s proměnnými je v Kommanderu docela zamotaná a co víc, dokumentace (které není mnoho) o tomto tématu mlčí. Takže mi nezbyvalo nic jiného, než se ponořit do čtení zdrojových kódů aplikací napsaných v Kommanderu. Tak jsem objevil čtyři způsoby, jak Kommander pracuje s proměnnými:

1. *Proměnné prostředí* – dříve se k nim přistupovalo poněkud krkolomně pomocí příkazu

`@exec(echo $FOO)`. Dnes již Kommander obsahuje funkci `@env(FOO)`.

2. *Proměnné v shellu* – protože obslužné rutiny běží v shellu, je používání proměnných úplně stejné. Uvozují se znakem \$.
3. *Globální proměnné* – způsob, který je hojně využívám v instalátoru, a znáte jej i z minulého dílu. Klíčem jsou dvě funkce `@setGlobal(foo, hodnota)` a `@global(foo)`. Jak už jsem říkal, každý obslužný skript běží samostatně, a proto proměnné, které potřebujeme i jinde v programu, musíme převést na globální.
4. *Proměnné v cyklech* – jazyk Kommanderu obsahuje i svoje speciální cykly a v nich se používají proměnné. K těmto proměnným se přistupuje pomocí znaku @.

```
@for(i,1,5,1) @Message.info(@i) @endif
```



Na tomto by snad nebylo nic tak těžkého, kdyby... Naprosto neřešitelným problémem pro mě bylo, jak použít proměnnou shellu jako parametr některé z funkcí. Vysvětlím na malém příkladě. Mám proměnnou `foo` a chci její obsah uložit do widgetu `label`. Kód, který by asi každý napsal nějak takto:

```
foo="Hello, world!"
@label.setText($foo) # nic se nestane
@label.setText(@foo} # chyba, @foo není definován, nefunguje.
```

Nakonec se mi podařilo najít *řešení* v podobě výrazu:

```
foo="Hello, world!" dcop @dcopid KommanderIf setText label $foo
```

Tedy řešení. Po tomto zjištění jsem se pokorně vrátil ke globálním proměnným. Zůstala jedna výhoda. Alespoň teď vidíme, jak přesně starý parser Kommanderu pracuje.


```
@Widget.názevFunkce(Obsah) #se převede na volání
dcop @dcopid Kommander If název Funkce Widget Obsah
```

Pokud neznáte DCOP (Desktop COmmunication Protocol), potom vezte, že se jedná o rozhraní, kterým spolu mohou programy (nejčastěji v KDE) komunikovat. Prvním parametrem na příkazovém řádku je identifikační číslo běžícího skriptu. Dále následuje jméno rozhraní, jméno funkce. Potom následují parametry pro funkci `setText`, a to jméno widgetu a text, který se má vložit. Více o DCOP naleznete v článku KDE: tipy a triky – III (DCOP a KDialog).

Klíčová slova Kommanderu

Přiznám se, že konceptu klíčových slov Kommanderu příliš nerozumím. Jistě, je dobré mít příkazy jako `@dcopid`, který vrací identifikaci běžícího okna. A nebo příkazy pro práci s globálními proměnnými a podobně. Ovšem smysl příkazů `@if`, `@for` a `@while` mi uniká. Přesto jsou součástí jazyka. Jenže jejich implementace je více než podivná! První nevýhodou je, že *není možné* tyto příkazy zanořovat! Kód

```
@if ( ... )
    @if ( ... )
    @endif
@endif
```

skončí chybovým hlášením `Unknow special: endif!` No a věc, která mě dostala ještě více, je nepřítomnost klíčového slova `@else`. Kvůli tomu je tato podmínka prakticky nepoužitelná.

Dalším aspektem, který snižuje použitelnost výrazu `@if` jsou neexistující konvence pro návratové hodnoty. Pravdivost příkazu `@if` je definována jako nenulové číslo, anebo neprázdný řetězec. Jenže taková funkce `@String.find` vrací při neúspěchu -1. Funkce `@String.compare` vrací při rovnosti řetězců 0, pokud je řetězec menší, tak -1, jinak 1. Což je implementace vhodná pro řazení řetězců, ale ne pro porovnávání rovnosti řetězců v podmínce. Pokud máte možnost, vyhněte se těmto funkcím, protože je evidentní, že je to celé šité horkou jehlou!

ScriptObject

Velkým rozdílem oproti klasickým skriptům je, že každý obslužný skript běží odděleně. U větších objektů potřebujeme udělat dekompozici a opakující se (nebo logické) části dát jinam. Řešením je widget `ScriptObject`. Tam si můžeme vložit potřebný kód a zavoláním `@ScriptObject` jej vykonáme. Tyto skripty trpí dvěma nevýhodami. Jednak netuším, zda vůbec podporují parametry (nikde jsem to nenašel). Dalším problémem je, že ač nevizuální, dělají problémy při používání správce rozložení. Na druhou stranu jsou potřebné při používání signálů a slotů (více dále). Pokud chceme sofistikovanější funkce, nezbyde nám nic jiného než použít klasický shellový způsob

```
'./cesta/k/souboru/knihovna.sh'
```

Tím bohužel přijdeme o výhodu Kommanderu, že je vše v jednom skriptu.

Signály a sloty

Signály a sloty je způsob, kterým spolu widgety z knihovny Qt mohou komunikovat (více viz [Začínáme KProgramovat – I](#) [8]). A protože používáme Qt widgety, používáme i signály a sloty. V Editoru propojujeme patřičné signály se sloty v okně *Edit Connections*.

Tento koncept se mi bez příkladu špatně vysvětluje. V demo aplikaci máte záložku *Signály a sloty*, kde máte příklad. Máme rozbalovací seznam (ComboBox) jménem `Partitions`, který obsahuje seznam oddílů. Ten naplníme při startu příkazem

```
@Partitions.insertItems(@exec(df | awk '/^\s*\s*/dev/ {print substr($1,6)}' | sort),0)
```

Tento kód nám naplní seznam všech oddílů bez úvodního `/dev/`. Nyní chceme zobrazit informace o celkové kapacitě, volném a použitém místě. V této chvíli využijeme `ScriptObject` z předchozí části, protože ten bude zjišťovat velikosti a vypisovat je.

```
@Space.setText(Kapacita @exec(df --block-size 1K -h /dev/@Partitions.text \\
| awk '/dev/ {print $2}')B)
```

Zjištění jiných hodnot než celkové velikosti je už analogické. Stačí změnit parametr příkazu `print`. Tento kód se bude provádět při startu, po naplnění seznamu. A dále po změně widgetu `Partitions`. K tomu je nutné správně propojit patřičný signál a slot, jak ukazuje tento obrázek.

Odesílatelem je widget `Partitions`, který vyšle signál `widgetTextChanged(const QString&)(textWidgetuZměněn)`. Příjemcem je objekt `Size`, respektive jeho slot `execute` (proved' se). Tím dojde při výběru jiného disku k aktualizaci hodnot.

Použití jiného skriptovacího jazyka

Při psaní kódu se nemusíte omezovat (ne)možnostmi shellu. Kommander nabízí snadnou integraci s jiným programovacím jazykem. V dokumentaci jsem nenašel, které jsou podporované. Osobně jsem zkusil Perl, Python a Ruby, ale předpokládám, že bude stačit, aby interpret dokázal načítat vstup ze standardního jazyka. Ostatně můžete vyzkoušet svůj oblíbený interpret (třeba interpret jazyka C?) a o výsledku poreferovat v diskuzi. Pro Pythonisty mám dobrou zprávu, Kommander zachovává odsazení, takže jeho používání nic nebrání.

```
@execBegin(python)
import os
def hello():
    os.system('kdialog --msgbox "Hello from Python!"')
hello()
@execEnd
```

Jak vidíte, kód v Pythonu je uzavřen mezi `@execBegin(python)` a `@execEnd`. Uvnitř už můžete používat libovolné konstrukce Pythonu, případně Kommanderu. Podle dokumentace není doporučeno psát tento kód v událostních skriptech jednotlivých widgetů kvůli možným problémům s návratovými hodnotami. Kommander rovněž neklade žádná omezení na počet použitých jazyků.

Problém se zavináčem

V Perlu i Ruby má zavináč syntaktický význam. Proto je nutné tyto výrazy uvodit ještě jedním zavináčem, aby je parser snadno poznal od svých výrazů.

```
@execBegin(perl)
open(MTAB, "/etc/mtab");
@@foo = <MTAB>;
foreach (@@foo) {
    print $_;
}
@execEnd
```

Nový parser

Od Kommanderu verze 1.2 (KDE 3.4 [9]) existuje nový parser Kommanderu, který vytváří [Michal Rudolf \[10\]](#) (z Polska). Zatímco starý parser je skupina maker, ten nový je už plnohodnotný jazyk, který podporuje následující:

- lokální a globální proměnné a asociativní pole
- výrazy
- práce s řetězci
- podmínky a cykly
- přímá manipulace s widgety

Protože není nový parser zpětně kompatibilní se starým (naštěstí), je nutné jej explicitně zapnout. To můžeme udělat dvěma způsoby. Buďto napíšeme `#!kommander` na první řádek skriptu. Nebo můžeme nastavit vlastnost celého formuláře `useInternalParser` na `true`.

Budu postupně rozebírat body z minulého dílu ([Kommander – 2 \(Starý parser\) \[11\]](#)), kde jsem Kommander kritizoval.

Proměnné a pole

Práce s proměnnými je nyní standardní, jako v jakémkoliv jiném jazyce. Proměnné jsou ve výchozím nastavení lokální. Globální proměnné jsou ty, které mají prefix `_`.

```
local    = 1
_global = 1
```

Jazyk Kommanderu je dynamicky typovaný a procedurální. Pouze někde narazíte na „objektový“ kód `widget.method(...)`. Ale toto je pouze zvolený způsob syntaxe, protože je to ekvivalentní se zápisem `method(widget, ...)`. Nemáme žádnou možnost vytvářet vlastní třídy, takže se skutečně jedná o procedurální jazyk.

Základními typy jsou celá čísla, reálná čísla a řetězce. Docela zamrzí skutečnost, že Kommander nepodporuje složitější typy, jako struktury, výčty, seznamy a podobně. Podporovány jsou všechny klasické aritmetické, logické a relační operátory.

```
+ - * / % (mod)  && (and) || (or) ! (not) < > <= >= == != (<>)
```

Jazyk podporuje jak céčkové operátory, tak jejich protějšky z Pascalu (v závorce). Spokojení tedy mohou být příznivci obou jazyků. Podporována jsou asociativní, jednorozměrná, pole. Opět se vytváří automaticky a není je třeba deklarovat.

```
foo[0]=1
foo[2]="1"
bar["a"]="c"
bar["c"]=1
```

Podmínky a cykly

Dále moje kritika směřovala k implementaci příkazů pro větvení. Ty se v nové verzi již chovají standardně. Je možné je libovolně zanořovat. Navíc patří řetězce mezi standardní typy jazyka, tudíž se porovnávají operátory `==`, nebo `!=`. Jejich syntaxe je následující:

```
if condition then code
```

```
[ elseif condition then code ]
[ else code ]
endif

while condition do code end

for variable=start_value to stop_value [ step step_value ] do code end

foreach variable in array do code end
```

Syntaktické rozdíly

Největším rozdílem je vynechání znaku @, jak to ilustruje následující příkaz:

```
@label.setText("Starý parser")
label.setText("Nový parser")
```

U většiny funkcí byl zrušen jejich rádobý objektový přístup. Funkce dostaly nové prefixy.

- @Array → array_
- @File → file_
- @Input → input_
- @File → file_
- @Message → message_
- @String → str_

Jedinou výjimkou jsou DCOP funkce pro ovládání widgetů, kde je daný přístup vhodný. Funkce Kommanderu pouze ztratily svůj prefix @ (`@env()` → `env()`) a rovněž jejich počet byl redukován.

Použití jiného skriptovacího jazyka

Starý parser nativně používal služeb shellu. To už nyní neplatí, takže je nutné shellový kód spouštět explicitně příkazem `exec()`. Nepovinný je druhý parametr, v němž se má daný příkaz spustit.

Pro delší úseky kódu tento způsob není vhodný. Bohužel klasické `execBegin() ... execEnd` v novém parseru nefunguje. Jediná možnost je použít služeb starého parseru, což se vylučuje s nastavením `useInternalParser` na true. Pokud potřebujeme využívat služeb jiného jazyka, nezbývá nám, než používat starý parser a ten nový explicitně zapínat jen tam, kde je potřeba.

Aplikace napsané v Kommanderu

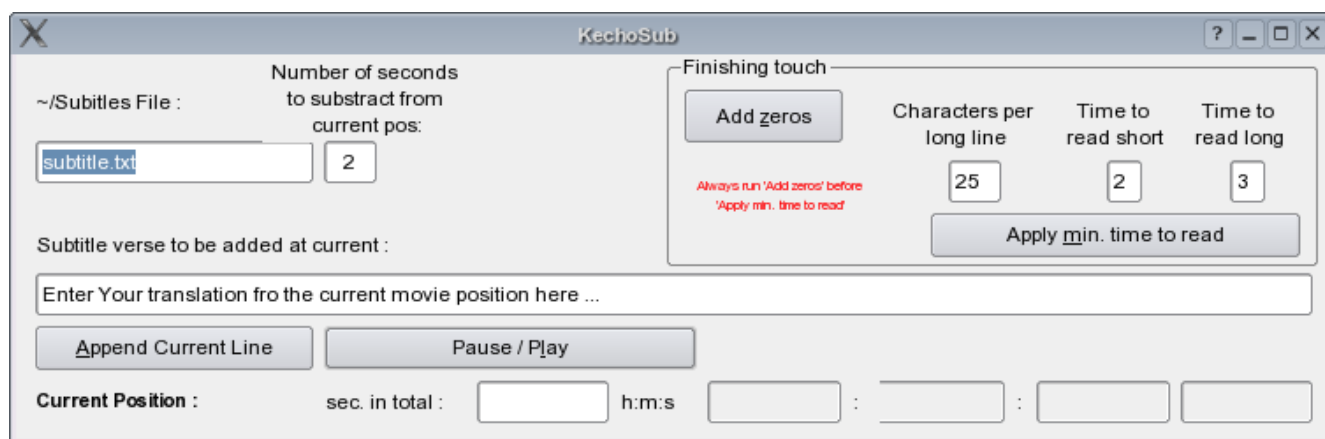
Zatím jsme se bavili o Kommanderu pouze teoreticky. Ale je k dispozici už řada hotových programků. Nejvíce jich je na kde-apps.org [12]. Vyberu pár zajímavých, bohužel většina z nich používá starý parser.

Static IP Configurator

Což byla první aplikace [13] v Kommanderu, kterou jsem viděl. Jejím autorem je Tomáš Matějčíček a je součástí jeho live distribuce Slax.

kEchoSub

Program [14] na tvorbu titulků k filmům.



KDE DVD Authoring wizard

Vytváří [15] DVD z jednoho, nebo několika mpeg-2 souborů. Umí tvorbu playlistů, vytvářet DVD menu, vytvořit ISO DVD, ... Kontroluje si svoje závislosti.

KQemu

GUI rozhraní [16] pro Qemu. A samozřejmě mnoho dalších aplikací. Je zajímavé, že nejvíce jich je z oblasti kódování videa, ripování DVD, ...

Závěr

Kommander je bezesporu velmi zajímavý koncept a dá se předpokládat, že v budoucnu jeho užívání vzroste. Představuje snadný a rychlý způsob, jak vytvářet jednoduché utility s grafickým uživatelským rozhraním. Všechny nelogičnosti a nevýhody starého parseru odstraňuje ten nový, čímž zvyšuje použitelnost jazyka. Bohužel se jedná o nový jazyk s překotným vývojem, který trpí velkým nedostatkem dokumentace. Což tento seriál alespoň v češtině trochu napravit.

Odkazy

- [1] <http://www.abclinuxu.cz/clanky/tipy/kde-tipy-a-triky-iii-dcop-a-kdialog>
- [2] <http://kde-apps.org>
- [3] <http://www.abclinuxu.cz/data/vyskocil/kommander-1-snap.kmdr>
- [4] <http://www.abclinuxu.cz/clanky/tipy/snimani-obrazovky-aneb-vyfotte-si-okno>
- [5] <http://www.abclinuxu.cz/clanky/programovani/kommander-1-skriptovane-gui>
- [6] <http://docs.kde.org/stable/en/kdewebdev/kommander/commands.html>
- [7] <http://www.abclinuxu.cz/data/vyskocil/kommander-2-demo.kmdr>
- [8] <http://www.abclinuxu.cz/clanky/programovani/zaciname-kprogramovat-i>
- [9] <http://www.abclinuxu.cz/clanky/recenze/novinky-v-kde-3.4>
- [10] <http://quanta.sourceforge.net/main2.php?newsfile=mrudolf>
- [11] <http://www.abclinuxu.cz/clanky/programovani/kommander-2-stary-parser>
- [12] <http://www.kde-apps.org>
- [13] <http://www.kde-apps.org/content/show.php?content=25666>
- [14] <http://www.kde-apps.org/content/show.php?content=23107>
- [15] <http://www.kde-apps.org/content/show.php?content=27528>
- [16] <http://www.kde-apps.org/content/show.php?content=19407>

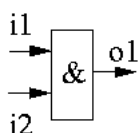
Jak se píše procesor

Luděk Crha

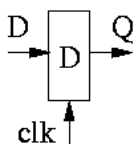
Počítač má dnes skoro každý. Většina lidí také ví, že jeho základním prvkem je procesor. Víte ale, jak se takový procesor dá vytvořit? Zamysleli jste se někdy nad tím, čím je dána maximální frekvence procesoru? Je možné si navrhnout a vyzkoušet vlastní procesor "doma"? Na tyto a podobné otázky odpoví tento článek. Další příkaz VHDL, navrhne první složitější obvod a vysvětlíme, co je simulace a syntéza.

Úvod

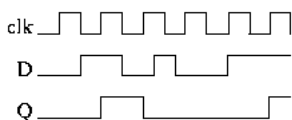
Nejdříve uvedeme základní pojmy. Z pohledu návrhu je procesor číslicový obvod. V číslicovém obvodu je třeba definovat dvě základní úrovně signálu – logická '0' a logická '1'. Číslicový obvod se skládá z hradel a propojení mezi nimi. Hradlo je elementární prvek, který realizuje základní funkci. Zde je příklad realizace logické funkce AND.



Hradlo na obrázku má dva vstupy a jeden výstup. Pokud je na obou vstupech '1', na výstupu hradla bude '1'. Při jakékoliv jiné kombinaci vstupů bude na výstupu '0'. My budeme používat hradla realizující logické funkce AND, OR, NOT a XOR.



S těmito hradly již můžeme vytvořit libovolnou logickou funkci. Něco nám ale pořád chybí – a to je paměťový prvek, který by pomohl vnést do našeho obvodu časovou závislost. Použijeme hradlo typu D, které si dokáže pamatovat hodnotu.



V okamžiku náběžné hrany (přechod z '0' do '1') hodinového signálu clk se na výstup Q zapíše hodnota vstupního signálu D, jinak zůstává hodnota na výstupu nezměněna. Na následujícím obrázku je znázorněn příklad časového průběhu, který demonstruje toto chování.

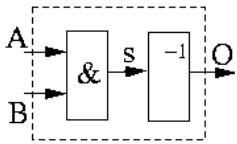
Hradlo D tedy funguje velmi jednoduše, ale, jak si ukážeme dále, spolu s pomocí logických hradel nám umožní navrhnout celý procesor.

Návrh obvodu

Teď, když máme základní stavební prvky, můžeme začít navrhovat obvod. Ale jak? Existují návrhové systémy, kde graficky skládáme obvod z hradel a ta propojujeme. Pro složitější obvody je však tento přístup takřka vyloučen. Naštěstí existují jazyky, které nám umožní textově popsat strukturu obvodu. Jejich syntaxe je velmi podobná programovacím jazykům. V této podobnosti je ale skryto jedno velké nebezpečí – nemusíte si ani uvědomit, že popisujete hardware a nepíšete program. My budeme používat jazyk VHDL (VHSIC Hardware Description Language), který je u nás nejrozšířenějším jazykem pro návrh.

VHDL

Základní prvky ve VHDL jsou entita a architektura. V deklaraci entity definujeme vstupy a výstupy obvodu, pomocí architektury tvoříme vlastní obvod. Dalším důležitým pojmem je signál, základní propojovací prvek. Signál si můžeme představit jako vodič, kterým spojujeme hradla. Vstupy a výstupy definované v entitě jsou také signály. Základem pro vytváření obvodu je příkaz přiřazení `<=`, který signálu na levé straně přiřadí hodnotu na pravé straně. Všechny uvedené pojmy si ukážeme na jednoduchém příkladu, realizaci obvodu na následujícím obrázku:



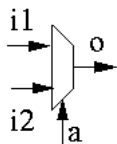
```
-- dve minuska jsou komentar
ENTITY jednoduchy_obvod IS -- entitu nazveme jednoduchy_obvod
PORT (                       -- klicove slovo PORT
  A, B: IN std_logic,       -- dva vstupy, A a B (typ std_logic
                             -- znamena bezny vodici)
  O: out std_logic          -- jeden vystup
);
END jednoduchy_obvod;      -- konec deklarace entity

ARCHITECTURE moje_architektura OF jednoduchy_obvod IS
                             -- tvorime obsah entity
  signal s: std_logic;       -- vytvorim si pomocny signal
BEGIN                       -- zacatek tela architektury
  s <= A and B;              -- v signalu s bude logicky soucin
                             -- signalu A a B
  O <= not s;                -- na vystupu O bude negace s;
END moje_architektura      -- konec architektury
```

V deklaraci entity definujeme vstupy obvodu **A** a **B** a výstup **O**. Tím říkáme, jak bude vypadat obvod z vnějšku. Architekturu (tedy vnitřek obvodu) přiřadíme k entitě a v deklarační části si vytvoříme pomocné signály, v našem případě jediný signál **s**. V těle architektury (za klíčovým slovem **begin**) již píšeme schéma obvodu. V signálu **s** chceme mít logický součin signálů **A** a **B**. To implikuje vytvoření hradla **AND**. Na výstup **O** přiřazujeme negaci signálu **s**, což vede na vytvoření hradla **NOT** mezi signál **s** a výstup **O**.

Syntaxi příkazu **when** si ukážeme na následujícím příkladu:

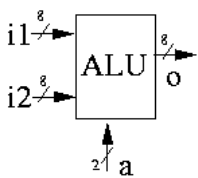
```
O <= I1 when A = '0' else I2;
```



Tento zápis znamená: do signálu **O** přiřad' signál **I1**, pokud je hodnota signálu **A** '0', jinak do signálu **O** přiřad' signál **I2**. Důležité je uvědomit si, že tímto příkazem neříkáme, co chceme vytvořit, ale jak se má výsledný obvod chovat. Z tohoto zápisu dostaneme zřejmě **multiplexor**, obvod, který funguje jako přepínač. Tento konkrétní multiplexor přepíná vstupy **I1** nebo **I2** na výstup **O** podle jednobitové adresy **A**.

Nyní můžeme pomalu začít skládat náš procesor. Uvažujme osmibitovou architekturu, šířka datových sběrnic tedy bude osm bitů. Pracovat s jednotlivými vodiči (signál **std_logic**) by bylo zdlouhavé, využijeme proto typ signálu **std_logic_vector**, který vzniká složením jednotlivých signálů **std_logic**. Při deklaraci signálu **std_logic_vector** určujeme pořadí bitů – např. (7 downto 0) znamená osmi-bitový signál s nejméně významným bitem vpravo, s bity číslovanými od 0 do 7. Konstanty typu **std_logic_vector** se zapisují v uvozovkách (např. "1010").

Jako první prvek vytvoříme **ALU** – aritmeticko-logickou jednotku. V procesoru zajišťuje realizaci aritmetických a logických funkcí.



Do naší ALU vedou dva osmibitové datové vstupy I1 a I2, výsledek operace je přiveden na výstup O. Operaci (zvolme např. AND, OR, NOT, +) určujeme dvoubitovým signálem A (dva bity jsou dostatečné k adresování právě jedné ze čtyř operací).

```
library IEEE;
use IEEE.std_logic_1164.all;
use ieee.std_logic_arith.all;           -- vloženi potrebných knih.
use ieee.std_logic_unsigned.all;

ENTITY ALU IS
PORT (
    I1, I2: IN std_logic_vector(7 downto 0); -- dva 8bitové datové vstupy
    A: in std_logic_vector(1 downto 0);      -- 2bitová adresa operace
    O: out std_logic_vector(7 downto 0)      -- jeden 8bitový výstup
);
END ALU;

ARCHITECTURE behavioral OF ALU IS          -- tvoříme obsah entity ALU
BEGIN
O <= I1 AND I2 when A = "00" else         -- na výstupu logický součin
    I1 OR I2 when A = "01" else          -- logický součet
    not I1 when A = "10" else            -- negace I1
    I1 + I2;                             -- scitání
END behavioral;
```

V deklaraci entity definujeme vstupy a výstupy ALU. Tělo architektury se skládá z jediného příkazu `when`, který určuje, jaká funkce se má objevit na výstupu při různých kombinacích bitů adresy. Fyzická realizace této funkce by odpovídala čtyřem obvodům realizujícím dané funkce (AND, OR, NOT, +) a čtyřvstupového osmibitového multiplexoru, který vybírá výsledek jednoho z obvodů na výstup dle hodnoty signálu A. Co dál s VHDL kódem? V úvahu připadají dvě činnosti

- `simulace` nebo
- `syntéza`

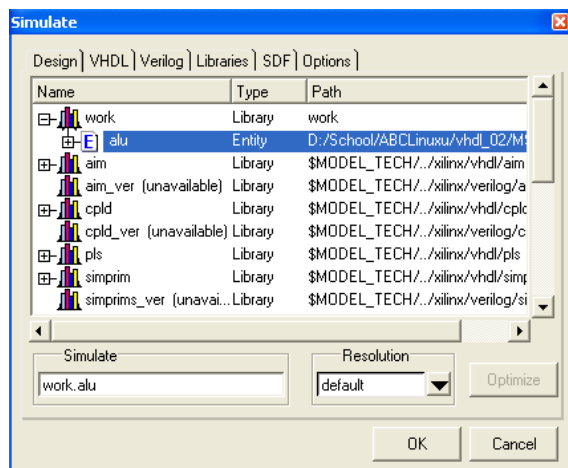
Simulace je ověření činnosti obvodu na počítači – program (simulátor) napodobuje činnost obvodu a v podobě časového průběhu signálů je možné sledovat funkci obvodu. Syntéza je převod VHDL kódu do formátu vhodného pro fyzickou realizaci obvodu.

Nyní si řekněme něco více o simulaci. Odsimulovat kód si můžete zkusit sami na domácím počítači, např. pomocí volně šiřitelného software ModelSim XE Starter, který si můžete stáhnout ze stránek firmy Xilinx [1]. Program vyžaduje jednorázovou registraci. Po stáhnutí a nainstalování je ještě třeba požádat o registrační klíč přes internetový formulář.

Po spuštění programu je nejdříve potřeba vytvořit nový projekt (*File* → *New* → *Project*). Poté si buď vytvoříte nový soubor pro zápis VHDL kódu (např. pravé tlačítko myši v prostoru *Workspace* → *Add to Project* → *New File*), nebo přidáte do projektu již existující soubor (pravé tlačítko → *Add to Project* → *Existing File*). V našem případě bude soubor obsahovat kód ALU uvedený výše. Příkazem *Compile* → *Compile All* zkompilujete zdrojové soubory do knihovny work. Nyní je možné

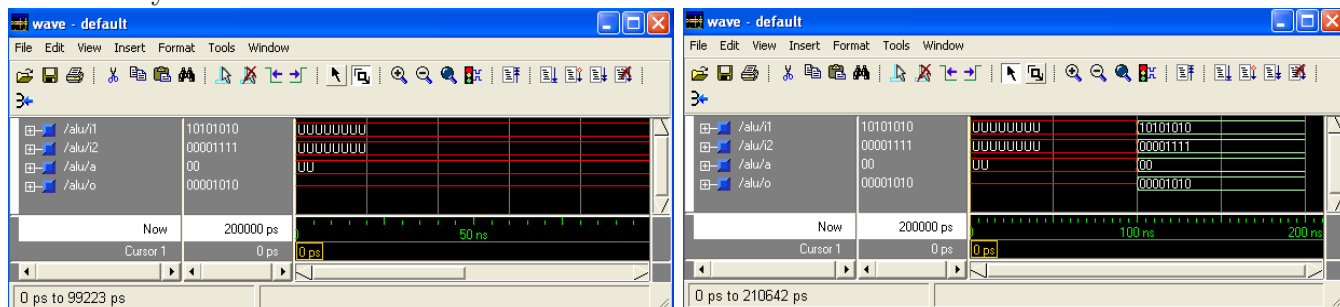
spustit režim simulace pomocí *Simulate* → *Simulate*. Vybereme entitu ALU z knihovny Work (je třeba rozvinout záložku *work*).

V hlavním okně ModelSimu se nyní objevila nová záložka *sim*, pomocí které můžeme nastavovat entitu, se kterou chceme pracovat (v našem případě je to pouze ALU). Z menu *View* nyní můžeme



vyvolávat okna potřebná pro simulaci, nejdůležitější jsou okna *Signals* a *Wave*. *Wave* zobrazí časový průběh signálů (prozatím je prázdné). V okně *Signals* vidíme všechny signály použité v entitě (včetně portů), které můžeme přidat do okna *Wave* (*Add* → *Wave* → *Signals in Region*). Jsme sice v režimu simulace, ale zatím jsme nic neodsimulovali. Simulaci musíme nechat běžet po určitou dobu – to můžeme udělat např. v hlavním okně ModelSimu v příkazovém řádku pomocí příkazu `run` s parametrem času, po který má simulace běžet. Pokud chceme např. simulovat obvod po dobu 100 nanosekund, zadáme příkaz `run 100 ns`.

Pokud jste celý postup provedli na příkladu naší ALU, objevily se vám v okně *Wave* na časové ose červené čáry.



Tím nám simulátor říká, že hodnoty signálů nejsou definované. Musíme proto na vstupy obvodu nastavit úroveň signálů – v okně *Signals* vybereme signál, do kterého chceme vnutit hodnotu a v *Edit* → *Force* → *Value* zadáme binárně hodnotu. Pokud např. zadáme hodnoty I1 10101010, I2 00001111, A 00 (funkce logického součinu) a znovu spustíme simulaci (`run 100 ns`), na výstupu O se nám objeví hodnota 00001010 (logický součin vstupů I1 a I2). Pokud je vidět pouze konec simulace, je dobré zobrazit celý průběh v okně *Wave* – *View* → *Zoom* → *Zoom Full*.

Nyní si můžete hrát s různým nastavováním signálů a testování funkce ALU. Postup přímého zadávání hodnot signálů je velmi zdlouhavý, v praxi se používá spíše metoda nadřazené entity zvané **testbench**, o kterém budeme hovořit v některém z dalších dílů.

Obvod máme odsimulovaný, dalším krokem v průběhu návrhu je syntéza. Pro syntézu již potřebujete další hardware, běžně se používají programovatelná hradlová pole CPLD nebo FPGA. Dostupné jsou i hotové kity, které řeší napájení programovatelného obvodu a jeho připojení k PC a periferiím. Pokud zvolíte obvod CPLD nebo FPGA firmy Xilinx, můžete si k nim zdarma stáhnout nástroj pro syntézu ISE WebPack. S tímto vybavením si můžete sami obvod nahrát a používat.

Důležité při psaní VHDL kódu je uvědomit si, co vlastně děláme. Nepíšeme program, ale určujeme funkci obvodu. Z toho také vyplývá použití programovatelných polí – hodí se pro paralelní nebo časově kritické aplikace, pro implementaci složitých algoritmů je naopak lepší využít procesor.

Procesy

Prozatím lze všechny příkazy, které jsme si ukazovali, zařadit mezi tzv. *souběžné* příkazy – všechny se provádí najednou bez ohledu na pořadí. V obvodové realizaci to ukazuje na paralelní obvody. Pomocí *procesu* můžeme použít příkazy *sekvenční*, kde naopak pořadí příkazů hraje důležitou roli. Proces

sám o sobě je jeden souběžný příkaz. Sekvenční zpracování příkazů v procesu se řídí následujícími pravidly:

1. Pokud jednomu signálu přiřazujeme různé hodnoty, předchozí příkazy jsou přepsány posledním z nich.
2. Pokud signál využíváme jako vstup i výstup, jeho hodnota se mění až po ukončení procesu

Proces můžeme také chápat jako myšlenkovou přípravu na vytvoření signálů. Nejlepší bude ukázat si činnost procesu na příkladu. Předpokládejme dvě různé architektury k entitě `example_1` (deklaraci entity neuvádím), která má pouze jeden výstupní signál `O`:

```
ARCHITECTURE arch_1 OF example_1 IS
BEGIN
  O <= '1';
  O <= '0';
END arch_1;
```

```
ARCHITECTURE arch_2 OF example_1 IS
BEGIN
  p_1: process
  begin
    O <= '1';
    O <= '0';
  end process;
END arch_2;
```

V prvním případě se snažíme do výstupního signálu `O` napojit současně dvě hodnoty – log. '1' a log. '0'. To by mohlo mít v reálném obvodu katastrofální následky. Pokud například uvažujeme logiku, kde log. '1' znamená připojení na +5V a log. '0' připojení na zem, zapisujeme tímto kódem zkrat. Ve druhém případě se ale příkazy zpracovávají sekvenčně. Druhý přiřazovací příkaz „přebije“ první a na výstupu `O` bude '0';

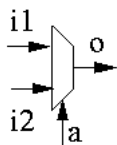
V procesu není možné použít souběžný příkaz `when`, místo něj můžeme aplikovat sekvenční příkaz `if`. Uvažujme příklad z minulého dílu, kde jsme vytvořili multiplexor pomocí příkazu

```
O <= I1 when A = '0' else I2
```

V procesu by vypadal zápis tohoto multiplexoru následovně:

```
p_1: process (I1, I2, A)
begin
  if A = '0' then
    O <= I1;
  else
    O <= I2;
  end if;
end process;
```

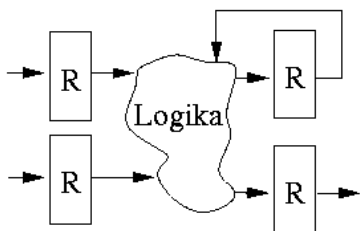
Zaměřme se nyní více na syntaxi procesu. Před klíčovým slovem `process` je návěští, v našem případě `p_1`. To je důležité pro případnou identifikaci procesu v průběhu simulace nebo syntézy. Za klíčovým slovem `process` je tzv. *sensitivity list*. V něm říkáme simulátoru, kdy má proces spustit a obnovit hodnoty výstupních signálů. Tento výpočet provede simulátor právě při změně libovolného signálu obsaženého v sensitivity listu. V praxi to znamená, že do sensitivity listu je třeba zapsat všechny signály, které do procesu vstupují. V našem případě jsou to signály `I1`, `I2` a `A`. Vezměme nyní následující proces:



```
p_1: process (I1, I2, A)
begin
  O <= I2;
  if A = '0' then
    O <= I1;
  end if;
end process;
```

Tento zápis je zcela identický s předchozím procesem. Nejdříve do výstupního signálu `O` přiřazujeme signál `I2`, ale v případě, že je hodnota signálu `A` '0', dostane se ke slovu přiřazovací příkaz `O <= I1`. Výsledek je tedy opět stejný multiplexor.

Jak psát design

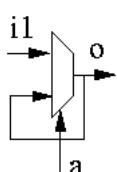


Nyní uděláme malou odbočku k tomu, jakým způsobem psát design. Budeme hovořit o synchronním designu, tzn. v obvodu se vyskytuje jediný hodinový signál `clk`. Dnešním standardem je jasně rozdělit design na registry (paměťové prvky, realizovatelné např. klopným obvodem D) a kombinační logiku (prvky AND, OR, NOT...) bez cyklů.

Registry jsou všechny taktovány na stejnou (např. náběžnou) hranu `clk`. V okamžiku této náběžné hrany se přepíše signál ze vstupu registru na výstup. Následuje cesta signálu přes kombinační logiku, než dorazí na vstup dalšího registru. Jakmile všechny signály projdou přes logiku a ustálí se, může přijít další náběžná hrana hodin a celý proces se opakuje.

Z toho vyplývá i maximální frekvence hodin. Jejich perioda musí být minimálně taková, aby se signál ustálil i na nejdelší cestě v celém obvodu. Pokud tedy taktujeme např. procesor na 1600 MHz, znamená to, že nejdelší cestou (mezi dvěma registry spojenými pouze logikou) v jeho designu projde signál za méně než 0,625 nanosekundy. Podívejme se nyní na následující proces:

```
p_1: process (I1, I2, A)
begin
  if A = '0' then
    O <= I1;
  end if;
end process;
```



Pokud má signál `A` hodnotu '0', bude na výstupu `I1`. Co ale když bude `A` '1'? To potom znamená zachovat na výstupu aktuální hodnotu. A zde se dostáváme k jádru problému. K zachování hodnoty potřebujeme paměťový prvek. V tomto konkrétním případě bude tvořen zpětnou vazbou, např. takto (obrázek).

A právě tato zpětná vazba (latch) není přípustná. Proč je vlastně zpětná vazba v logice problém? Např. může zmást syntezátor při počítání délky cesty v obvodu. Obvod potom může fungovat, ale také nemusí (což většinou nastane v nejméně vhodné dobu).

Jaký z toho plyne závěr? V příkazu `if` (pokud proces netvoří registr, ukážeme si příště) je třeba vždy pokrýt všechny stavy vstupních signálů pro výstupní signály. Jinými slovy, každý výstupní signál musí mít jasně definovanou hodnotu pouze v závislosti na vstupních signálech. Podobný problém může nastat i u souběžného signálu `when` – tam je také vhodné vždy uvádět klauzuli `else`.

Registry

Všechny naše dosavadní obvody měly jeden společný rys. Neobsahovaly žádný paměťový prvek, takže hodnoty na výstupu byly závislé pouze na vstupních signálech. Takovéto obvody nazýváme *kombinační*. Skládají se pouze z logických členů (AND, OR, NOT, ...) a neobsahují zpětné vazby. Naproti tomu obvody, které obsahují paměťové prvky, se nazývají *sekvenční*. Jejich výstup není určen pouze vstupními signály, ale i vnitřním stavem obvodu (stavem jednotlivých paměťových prvků).

Jak jsme již naznačili minule, jako paměťový prvek budeme využívat pouze registr (tzn. nebudeme tvořit zpětné vazby v logice). Podívejme se, jak můžeme registr vytvořit pomocí VHDL. Využijeme k tomu entitu `example_2` podobnou entitě z minulého dílu s tím rozdílem, že obsahuje navíc dva vstupní signály – `RESET` a `CLK`.

```
ENTITY example_2 IS
PORT (
  CLK, RESET: in std_logic;
  I1, I2: in std_logic;
  O: out std_logic
);
END example_2;

ARCHITECTURE arch_3 OF example_2 IS
BEGIN
  p_1: process (RESET, CLK)
  begin

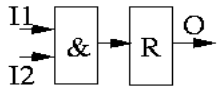
    if (RESET = '1') then -- asynchronni reset
      O <= '0';
    elsif (CLK'event and CLK = '1') then -- reakce na vz. hranu
      O <= I1 and I2;
    end if;

  end process;
END arch_3;
```

Do procesu jsme vložili dvě sekce. Jednu reagující na reset a jednu na vzestupnou hranu hodin. VHDL zápis konkrétně znamená:

- Pokud je hodnota signálu `RESET '1'`, spustí první sekci příkazů (to odpovídá činnosti asynchronního resetu v registru – nastavení implicitních hodnot po startu systému).
- V opačném případě (`elsif`) spustí druhou sekci příkazů, ale pouze v případě, že došlo ke změně signálu `CLK` a tento signál má hodnotu `'1'` (odpovídá náběžné hraně). Jinak řečeno, přiřadí

do výstupního signálu určenou logickou funkcí, ale pouze v okamžiku náběžné hrany hodin. Proto tento zápis vloží na konec cesty každého výstupního signálu registr.



Náš příklad bude tedy vypadat takto (obrázek). Podmíněný příkaz tvořící registr (`if RESET... elsif CLK...`) je dobré brát pokud možno dogmaticky. Dnešní programovatelné obvody důrazně dělí obvod na logiku a registry a každý zásah do této části VHDL kódu může tuto strukturu narušit.

Programovatelné obvody také mají dedikované hodinové rozvody, proto, pokud si nejsme jisti, je dobré používat v celém obvodu pouze jeden signál `CLK`. Nyní známe takřka všechno, co je třeba k napsání procesoru ve VHDL. Navrhněme proto další entitu – sadu registrů. Zvolme jednoduchou variantu sady čtyř osmibitových registrů, kde jedinou funkcí bude zápis do registrů a čtení z nich.

```

ENTITY register_set IS
PORT (
  CLK, RESET: in std_logic;
  I: in std_logic_vector(7 downto 0);
  O: out std_logic_vector(7 downto 0);
  ADDR: in std_logic_vector(1 downto 0);

  WR: in std_logic
);
END register_set;

ARCHITECTURE behavioral OF register_set IS
signal R1, R2, R3, R4: std_logic_vector(7 downto 0);
BEGIN
  p_1: process (RESET, CLK)
  begin

    if (RESET = '1') then -- asynchronni reset
      R1 <= (others => '0');
      R2 <= (others => '0');
      R3 <= (others => '0');
      R4 <= (others => '0');
    elsif (CLK'event and CLK = '1') then -- reakce na vz. hranu
      if WR = '1' then
        if A = "00" then
          R1 <= I;
          elsif A = "01" then
            R2 <= I;
            elsif A = "10" then
              R3 <= I;
            else

```

```

R4 <= I;
    end if;
    end if;
end if;
end process;

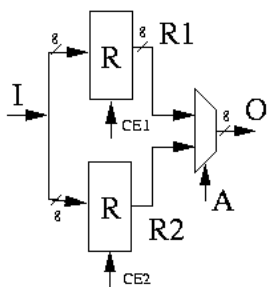
O <= R1 when A = "00" else
    R2 when A = "01" else
    R3 when A = "10" else
    R4;

END behavioral;

```

V entitě definujeme vstupní a výstupní signály. **CLK** a **RESET** jsou důležité pro synchronizaci registrů. **I** a **O** jsou vstupní a výstupní datové sběrnice, **A** vybírá jeden ze čtyř registrů. Pomocí signálu **WR** zapisujeme do jednotlivých registrů. Signál pro čtení není třeba, na výstupní sběrnici je vždy hodnota aktivního registru.

V deklarační části architektury si definujeme čtyři pomocné osmibitové signály, pomocí kterých vytvoříme vlastní registry. Hlavní část těla architektury tvoří proces, který implikuje vytvoření registrů a realizuje zápis. Kód nám říká, že v případě aktivního signálu **WR** má být přiřazena hodnota sběrnice

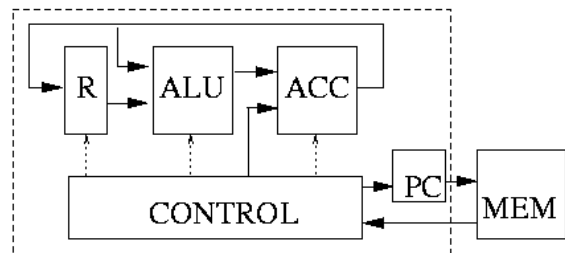


I do jednoho ze signálů **R1** – **R4**. Toto přiřazení se má provést pouze v okamžiku náběžné hrany signálu **CLK**. Pro syntezátor to tedy znamená vytvoření čtyř osmibitových registrů se vstupem **I**, výstupem **R1** – **R4** a povolovacím signálem (chip enable, **CEx**) tvořeným ze signálů **WR** a **A**.

Výstup **O** je tvořen pomocí signálu **WHEN**. V tomto případě to bude multiplexor, který vybírá výstup jednoho z registrů v závislosti na adrese **A**. Na obrázku je znázorněno zjednodušené schéma pro 2 registry.

Příklad

Pro naše ukázkové účely bude stačit úplně jednoduchá architektura procesoru, kterou vidíte na následujícím obrázku. Sběrnice jsou označeny plnou čarou, řídicí signály jsou tečkované.



Vlastní procesor je ohraničen čárkovanou čarou. Jeho jediný výstup je adresa pro paměť, ze které čte instrukce. Tato adresa je uložena v registru **PC** (*Program Counter*). Nejdůležitějším registrem je *Akumulátor* (**ACC**), do kterého se zapisují výsledky operací **ALU**, a do kterého také můžeme zapsat přímou hodnotu pomocí instrukce. Z akumulátoru také můžeme zapisovat hodnoty do registrů v registrovém poli **R**. Jeden vstup **ALU** je vždy akumulátor, druhý je jeden z registrů.

Control nastavuje řídicí signály pro ostatní prvky procesoru. Abychom mohli tento prvek navrhnout, musíme znát formát instrukce. Pro příklad zvolme velmi jednoduchý formát, kde instrukce bude mít 16 bitů, přičemž horních 8 bitů bude případný přímý operand a spodních 8 bitů budou zakódované řídicí signály:

DIRECT OPERAND								RSRVD	reg	alu	op				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Op určuje operaci. Zvolme 4 možné operace:

- 00 bude zápis do PC (změna adresy pro čtení z paměti, takže instrukce skoku).
- 01 pro provedení operace ALU.
- 10 pro zápis přímého operandu do akumulátoru.
- 11 pro zápis z akumulátoru do registru.

Alu označuje dva bity, které určují, jaká operace ALU se má provést. Dva bity *reg* vybírají registr, se kterým se má provést daná operace. Další dva bity jsou prozatím nevyužité. Následuje osmibitový přímý operand.

Kód

Podívejme se tedy konečně na VHDL kód, který popisuje tento procesor. Protože entity ALU a pole registrů jsme již navrhli v předchozích dílech, vložíme je do naší entity jako tzv. komponenty. Tento přístup nám umožňuje hierarchicky skládat obvod z dílčích entit. Komponentu nejdříve musíme nadeklarovat v deklarační části architektury a v těle architektury potom provést napojení této komponenty na naše signály.

```
library IEEE;
use IEEE.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

ENTITY main IS
PORT (
    CLK    : in  std_logic;
    RESET  : in  std_logic;
    INST   : in  std_logic_vector(15 downto 0);
    ADDR   : out std_logic_vector(7 downto 0)
);
END main;

ARCHITECTURE behavioral OF main IS

component ALU IS PORT (
    I1, I2: IN  std_logic_vector(7 downto 0);
    A: in  std_logic_vector(1 downto 0);
    O: out std_logic_vector(7 downto 0));
END component;

component register_set IS PORT (
    CLK, RESET: in  std_logic;
    I: in  std_logic_vector(7 downto 0);
    O: out std_logic_vector(7 downto 0);
    ADDR: in  std_logic_vector(1 downto 0);
    WR: in std_logic);
END component;
```

```
signal acc, alu_acc, reg_alu: std_logic_vector(7 downto 0);
signal pc: std_logic_vector(7 downto 0);
signal op_pc, op_alu, op_reg, op_acc: std_logic;

BEGIN
  p_pc: process (RESET, CLK)
  begin
    if (RESET = '1') then
      pc <= (others => '0');
    elsif (CLK'event and CLK = '1') then
      if op_pc = '1' then
        pc <= INST(15 downto 8);
      else
        pc <= pc + '1';
      end if;
    end if;
  end process;

  p_acc: process (RESET, CLK)
  begin
    if (RESET = '1') then
      acc <= (others => '0');
    elsif (CLK'event and CLK = '1') then
      if op_acc = '1' then
        acc <= INST(15 downto 8);
      elsif op_alu = '1' then
        acc <= alu_acc;
      end if;
    end if;
  end process;

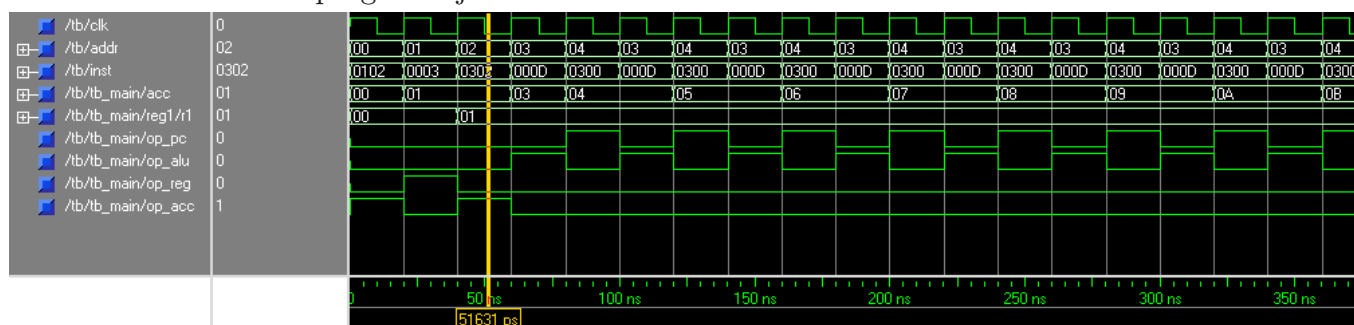
  op_pc <= '1' when INST(1 downto 0) = "00" else '0';
  op_alu <= '1' when INST(1 downto 0) = "01" else '0';
  op_acc <= '1' when INST(1 downto 0) = "10" else '0';
  op_reg <= '1' when INST(1 downto 0) = "11" else '0';

  reg1: register_set port map (CLK => clk, RESET => reset, I => acc, \
    0 => reg_alu, ADDR => INST(5 downto 4), WR => op_reg);

  alu1: alu port map (I1 => acc, I2 => reg_alu, A => INST(3 downto 2), \
    0 => alu_acc);
  ADDR <= pc;
END behavioral;
```


Vstupy naší entity jsou hodinový vstup, reset a data z paměti (instrukce). Jediným výstupem je adresa paměti. Po deklaraci komponent a signálů následuje vlastní kód, který tvoří obvod. V něm jsou dva procesy reagující na hodinový signál, které tvoří dva registry našeho obvodu – PC a ACC. Co a zda do těchto registrů zapisovat, se řídí signály `op...`, které jsou tvořeny pomocí instrukce `when` z dvou bitů instrukce určujících operaci procesoru. Následuje již pouze propojení komponent s registry a s ALU. K jejich řízení využíváme jednotlivé bity instrukce (např. operace ALU je dána bity 2 a 3). Spojení mezi těmito komponentami navzájem a spojení s dalšími prvky procesoru zajišťují interní signály (např. z ALU do akumulátoru je to signál `alu_acc`).

Jako příklad funkce jsem zvolil jednoduchý program, který uloží do akumulátoru hodnotu 1, tuto hodnotu zapíše do registru R1 (přímý zápis do tohoto registru není možný), uloží do akumulátoru hodnotu 3, inkrementuje tuto hodnotu a poté provede skok opět na tuto instrukci inkrementace. Simulace běhu tohoto programu je zde:



Vícebitové hodnoty jsou zobrazeny v hexadecimálním formátu. Pokud chcete vidět přesnou funkci tohoto programu, je třeba převést hodnoty instrukce do binárního formátu a porovnat s formátem instrukce.

V signálu `Addr` je zřejmé, že *Program Counter* na adrese 04 narazil na instrukci skoku na adresu 03, čímž se vytvořila nekonečná smyčka. V registru `Acc` (akumulátor) můžeme vidět inkrementující se hodnotu. První operace je zápis do akumulátoru (aktivní signál `op_acc`), poté zápis akumulátoru do registru (signál `op_reg`) a znovu zápis do akumulátoru. Potom již jen inkrementujeme akumulátor v nekonečné smyčce.

Procesor, který byl zvolen pro ilustraci, má řadu nedostatků, např. neumožňuje podmíněné skoky, formát instrukce plýtvá pamětí atd. Řešení těchto nedostatků však může posloužit jako cvičení zvládnutému čtenáři.

Závěr

Tímto náš úvod do VHDL končí. Cílem bylo ukázat, že hardware je možné popsat pomocí jazyka. Pokud ve vás alespoň trošku vzbudil zájem o problematiku návrhu obvodů, potom splnil svůj účel. Všem děkuji za pozornost!

Odkazy

[1] http://www.xilinx.com/ise/verification/mxe_details.html

Oracle: vývoj na SUSE 10.0

Petr Vaněk

Desátá verze SUSE je na světě poměrně krátkou dobu, tudíž s ní zatím veškeré dostupné verze Oracle databáze nepočítají. Naštěstí jsou ale starší SUSE Novellem a Oracle podporovány, takže se dá desítka přiohnout a „ohákovat“.

Úvod

S níže popsáním postupem jsem strávil krušné chvílky, takže rád ostatním od problému pomůžu.

V následujícím textu předpokládám, že na je na počítači nainstalován (Open)SUSE 10.0 s alespoň minimálním X Window systémem a C/C++ vývojovými nástroji, a že se chystáme instalovat 10g release 2. Dále předpokládám znalost základních termínů Oracle.

Instalace serveru

Samotný DB server není potřeba, pokud existuje možnost připojení na nějaký síťový. Já si jej ale pro jistotu stejně instaluji. Nejprve stáhneme a nainstalujeme tzv. **orarun** [1] balík (běhové prostředí Oracle) ze staršího SUSE Linux Enterprise Serveru. Jako root instalujeme pomocí

```
rpm -ivh --nodeps orarun-1.8-109.15.i586.rpm
```

a stále jako root zůstáváme. Přes YaST nainstalujeme **libaio** a **libaio-devel** balíky (asynchronní IO, hlavně nesplést s **libao**). YaST zde vypíše pestrobarevné varování o nesplněných závislostech s **pdksh** a **xshared**, ale to nevadí, klidně je ignorujeme, protože to je pozůstatek SLESu.

Orarun nám mimo jiné vytvořil nového uživatele **oracle** a skupinu **oinstall**, pod kterými vše poběží. Uživatel **oracle** je ale vypnutý, takže musíme buď proklikat YaST (*Bezpečnost a uživatelé – Správce uživatelů*), anebo editujeme login shell v **/etc/passwd** a nastavíme heslo. V souboru **/etc/profile.d/oracle.sh** editujeme proměnnou **ORACLE_HOME** na **\$ORACLE_BASE/product/10gR2** a zvolíme **ORACLE_SID**.

Dále spustíme **/usr/sbin/rcoracle start**, což nastaví parametry linuxového jádra na Oraclu příjemné. Také zde ignorujeme varovná hlášení o nenastavené proměnné **ORACLE_HOME** – nastaví se posléze. Pak opustíme roota a přihlásíme se do X uživatele **oracle**. Z webu Oracle stažený zip balík (**10201_database_linux32.zip**) rozbalíme a pustíme se do nejjednoduššího hacku v dějinách databází – v souboru **database/install/oraparam.ini** do seznamu **Certified Versions** připíšeme SuSE-10 (a to i přesto, že třeba používáme OpenSUSE). Tuto editaci lze obejít parametrem **-ignore prerequisites** při instalaci, ale to jsem se jednak dozvěděl až později, a druhak je to takové bez šťávy.

A teď už nic nebrání tomu, abychom spustili **runInstaller**, který nás provede celou instalací (ano, je to v grafice, protože máme na instalaci málo času a nervů). Manuálně musíme provést kontrolu „Product-Specific Prerequisites Checks?“, protože instalátor opět nenajde SUSE 10. V závěru ještě musíme pod rootem spustit skripty **/opt/oracle/orainventory/orainstRoot.sh** a **/opt/oracle/product/10gR2/root.sh**, k čemuž nás ovšem vyzve dialogové okénko. Nyní by už měl být server dostupný včetně nastavených **tnsnames** a **listeneru**, o čemž se přesvědčíme např. programem **tnsping**.

Pokud chceme spouštět Oracle automaticky při startu, editujeme v souboru **/etc/sysconfig/oracle** proměnnou **START_ORACLE_DB** na „yes“, to samé s **listenerem** apod. Můžeme také použít dobrodiní YaSTu (*Systém – Editor souborů /etc/sysconfig – Productivity – Databases – Oracle*). Na svém

pracovním stroji mám nastavení následující: V `/etc/sysconfig/oracle` mám start Oracle a příslušenství povolen, přičemž službu nespouštím v žádném runlevelu, anžto je to potvora nenažraná. Lokální DB server si spouštím podle potřeby přes `/etc/init.d/oracle start`.

Instalace klienta

Opět pod uživatelem `oracle` rozbalíme stažený balík (`10201_client_linux32.zip`), ve vzniklém instalačním adresáři provedeme výše popsanou úpravu `client/install/oraparam.ini`, čímž obojdeme kontrolu podporovaných verzí operačního systému, a v X spustíme instalátor. Protože instalujeme vývojářské prostředí, nezvolíme tzv. instantního klienta (tj. minimální běhové prostředí klientských aplikací), ale vybereme software pro DBA nebo vývojáře.

Do `.profile` si přidáme Oracle proměnné, abychom mohli spouštět patřičné programy, a třeba se dostali k `tnsnames`.

```
# oracle vecirek
export ORACLE_HOME=/opt/oracle/product/10gR2
export PATH=$PATH:$ORACLE_HOME/bin
```

Oracle také neregistruje své knihovny v `ld.so.conf`. Proto si v `/etc/ld.so.conf.d/` vytvořím soubor `oracle.conf`, který bude automaticky vložen při načtení linkeru. Do něj napíšu

```
/opt/oracle/product/10gR2/lib
```

a zavolám `ldconfig`.

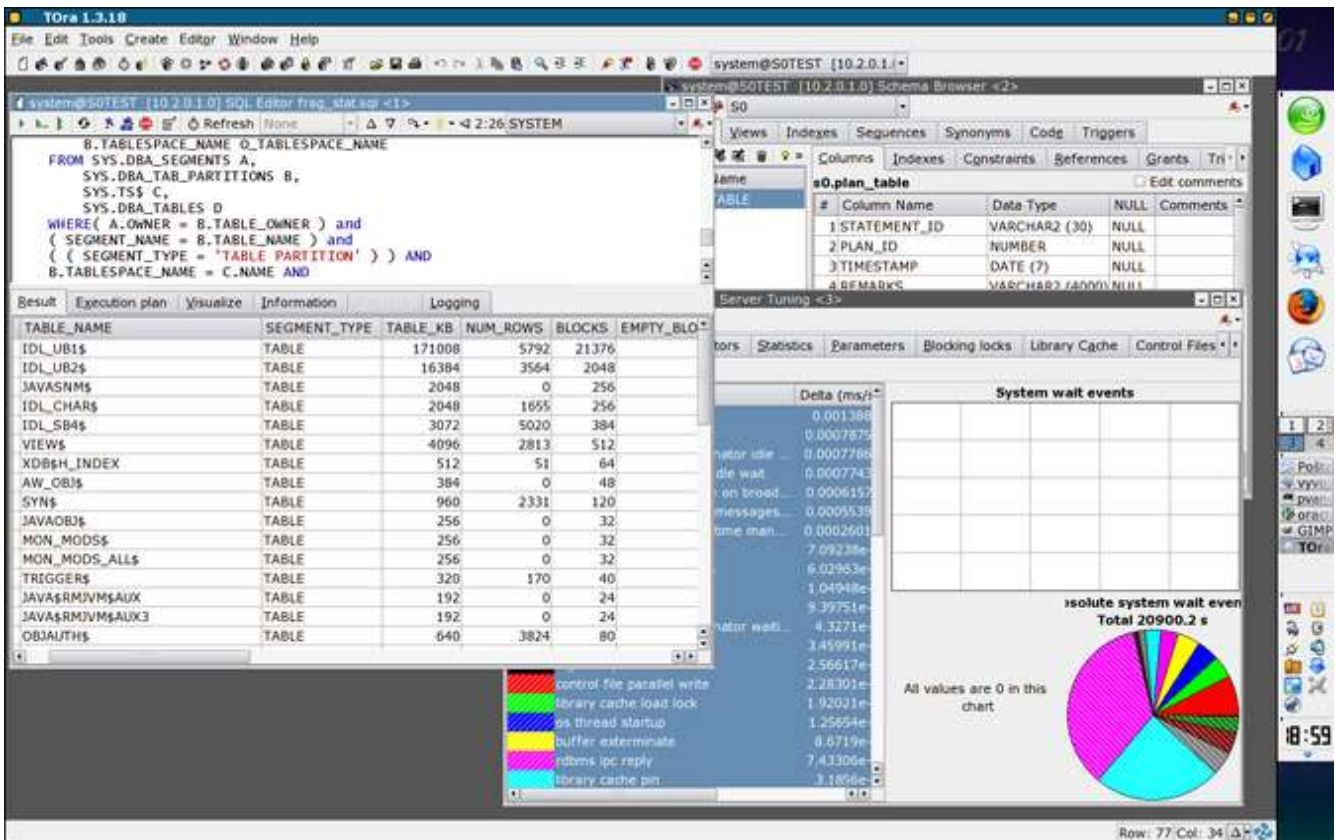
Tora

`sqlplus` je určitě šikovný nástroj, ale my, jako správní pojídači koláčů, raději sáheme po něčem, co je pro vývoj šikovnější, co nám například pomůže s laděním SQL nebo monitoringem databáze. V linuxovém světě nemáme na výběr, musí nám vyhovovat Tora [2] a Tora pro SUSE v RPM neexistuje a Tora naneštěstí používá *hodně nestandardní* způsob překladau ze zdrojových kódů...

Protože si nechceme zahnojit systém a protože nechceme číst dlouhé *manuály balení do RPM* [3], použijeme kompilaci do domácího adresáře vývojáře, toho času do mého. V systému musíme mít pochopitelně nainstalovány tzv. devel RPM balíky Qt `qt3-devel` a o něco méně pochopitelně `qt3-devel-tools`. Stažený balík (`tora-1.3.18.tar.gz`) rozbalíme a vezmeme tyto pečlivě odladěné configure parametry. Protože SUSE a protože Tora.

```
./configure \\  
  --prefix=/home/pvanek/tora-bin\<\  
  --without-kde\<\  
  --with-qt-dir=/usr/lib/qt3\<\  
  --with-qt-includes=/usr/lib/qt3/include\<\  
  --with-qt-libraries=/usr/lib/qt3/lib\<\  
  --with-static-oracle
```

Autor nemá nic proti KDE, sám jej používá, ale nechtělo se mu instalovat kde-devel balíky. Bez parametru `--with-static-oracle` nebude dostupná Oracle komponenta (to je ten SUSE důvod). Ostatní cesty ke knihovnám a hlavičkovým souborům si na koleně vyráběný configure skript sám nenajde.



gVim

Psaní větších bloků PL/SQL příkazů je pro někoho pohodlnější v tomto editoru. Existuje do něj řada modulů, které spolupracují s databázovým serverem. Já si vybral [oracle.vim](#) [4], ze kterého v podstatě používám prosté spuštění obsahu souboru jako sqlplus skriptů, a nasazuji tím editované package apod. (Pikantní je zejména to, že gVim používám pouze ve Windows.)

Jádrem konfigurace je úprava procedury `SelectDatabase()`, kde lze definovat často používaná připojení v poměrně srozumitelné if podmínce.

Enterprise Manager

EM je správcovský nástroj spouštěný jako webová aplikace na vlastním web serveru. Před spuštěním musíme nastavit identifikátor databáze, kterou chceme pomocí EM prohlížet, např.:

```
export ORACLE_SID=s0test
```

a jako uživatel oracle zavoláme `/opt/oracle/product/10gR2/bin/emctl start dbconsole`. Výpisy spuštění EM nám řeknou, že je vše dostupné na adrese <http://pvanek.clance.local:1158/em/console>. Pozor – některé funkce EM nelze provozovat v Konqueroru, takže se spuštění Firefoxu nevyhne.

Odkazy

- [1] <http://ftp.novell.com/partners/oracle/sles-9/>
- [2] <http://tora.sf.net>
- [3] <http://www.abclinuxu.cz/clanky/ruzne/abcserialy#balicrpm>
- [4] http://www.vim.org/scripts/script.php?script_id=141

Slackware Linux na 386? Proč ne!

Lukáš Rýdlo

Kterak rozchodit GNU/Linux na stroji, který už má svá nejlepší léta za sebou...

Entrée

Leží vám doma nevyužitá tříosmšestka a sentiment vám ji nedovolil vyhodit? Nebo se chce někdo z vašich známých zbavit tohoto muzejního kousku? Nainstalujte na něj Linux! Ne, opravdu nejsem linuxový fanatik ani jsem se nezbláznil. Také je mi jasné, že takovýto stroj je nejlepší použít jako „gameboy“ s DOSem a Dunou, GP nebo podobnými aplikacemi... Proč tedy instalovat Linux na tříosmšestku? Protože si spousta lidí myslí, že tak starý počítač se nedá rozumně využít a není to pravda.

Zkusíme si nainstalovat jednu ze starších distribucí GNU/Linuxu. To nám umožní lépe proniknout do tohoto systému, jelikož v této verzi ještě nebudou žádná klikátka a také se nebudeme muset bát, že něco poškodíme. Vždyť ten stroj jsme přeci už dávno nepoužili a hry se dají hrát i v DosBoxu. V tomto článku popíši plnohodnotnou instalaci stanice, tedy nikoliv terminálu, jak by asi většina očekávala. Je to proto, abychom si ukázali, co všechno mohlo běžet na stroji s procesorem AMD 386 DX/DXL-40, diskem o velikosti 250 MB a s RAM 4 (později 8) MB. Aby toho nebylo málo, ukážeme si, že na X-Window nám opravdu stačí monochromatická grafická karta s monitorem zářícím pouze jednou barvou.

Výběr distribuce

Je lepší Debian nebo Slackware? Fedora nebo Mandrake? Flamewar necháme dnes stranou a sáhneme po Slackware ve verzi 3.3. Proč? Protože existuje spousta mirrorů, ze kterých si ho můžeme stáhnout, a hlavně – instalační adresář má pouhých 511 MB, z čehož potřebujeme dokonce jen 164 MB. Také je to velmi průhledný systém s téměř prázdným adresářem `/etc`, což nám vyhovuje. Kdo chce, může samozřejmě odzkoušet nějakou starší verzi Debiana nebo třeba RedHatu, který je ale poměrně nenažraný, co se týče požadavků na velikost harddisku. Z čistě teoretického hlediska bychom mohli použít i některou z novějších verzí Linuxu. Bohužel procesoru 386 chybí určité poměrně důležité instrukce, takže je nepravděpodobné, že bez kompilace s optimalizací pro tuto řadu procesorů něco zmůžeme (při kompilaci gcc se přidává parametr `-march=i386`).

Pokud jste hrdým vlastníkem čtyřosmšestky, pak neváhejte zkusit nainstalovat něco novějšího, protože 486 je na tom už lépe. Za zmínku ještě stojí disketové distribuce. Volil jsem mezi Floppixem, AlfaLinuxem a TinyLinuxem. Žádná z nich ale nevyhovovala, protože Floppix nemá podporu harddisků, a zbylé dvě distribuce jsou optimalizované pro procesor 486. Navíc naše upgradovaná kalkulačka zvládne přece mnohem víc!

Příprava k instalaci

Nejprve je třeba stáhnout instalačky. Stačí adresáře `slakware-3.3/slakware`, `slakware-3.3/boot-dsks.144` a `slakware-3.3/rootdsks`, všehovšudy skoro 164 MB. Není třeba stahovat celý kořenový adresář, protože například alternativní instalační obrazy nebo zdrojové kódy nás pravděpodobně nebudou zajímat. Odkud stahovat, poradí strýček Google. Lze zkusit například jeden server našich rakouských kolegů: `ftp.slackware.at` [1]. Nyní si rozmyslíme způsob instalace. Na výběr je z několika variant:

1. Ortodoxní disketová

Usedneme mezi tříosmšestku a stroj, na kterém jsou instalačky. Instalačky budeme kopírovat na diskety a instalovat z disket. To je velmi nepraktické, protože během instalace nejspíš „odvažíme“ pár disket a mechanika to bude taky ještě dlouho rozdýchávat. Navíc budeme potřebovat buď dvě disketové mechaniky nebo velkou RAM.

2. Ortodoxní z disku

Vezmeme nějaký další disk, na kterém vytvoříme DOSový (nebo ext2) oddíl, nakopírujeme na něj instalačky a připojíme ho do tříosmšestky jako druhý disk. Nabootujeme z diskety a instalujeme z druhého disku na první. To je velmi snadné řešení, ale potřebujeme další disk s velikostí minimálně 122 MB pro plnou instalaci. Pokud se chcete ochudit o některé aplikace, můžete umazat příslušné adresáře (e* je emacs, t* TeX, x* XFree atd.).

3. Ortodoxní z CD

Stačí mít v tříosmině CD mechaniku a je to skoro stejné, jako instalace z disku. Nevím jak BIOS vašeho obstarožního mazlíčka, ale ten můj si s CDromkou neumí popovídat. Každopádně by to bylo asi nejlepší řešení.

4. Ulejšácká

Tato varianta spočívá v tom, že se zapojí disk tříosmšestky jako první do stroje s instalačkami a celá instalace proběhne na něm. Z CD nebo z disku. Pokud se rozhodnete pro tuto variantu a zvolíte disk, uvědomte si, že jádro, které budete bootovat k instalaci, nezvládá současné souborové systémy. Budete tedy potřebovat instalačky na DOSovém svazku. Ext2 svazek nelze použít, protože soudobé distribuce na něm používají nové „fičury“, které jádro 2.0.30 ve Slackware 3.3 neumí přečíst. No jo, to je ten pokrok ;-).

Start instalace

Nejprve se posadíme k počítači s instalačkama. Vezmeme dvě diskety. První bude bootovací a v prostředí Linuxu (v adresáři s instalačkama) ji vytvoříme příkazem

```
dd if=bootdsk144/bare.i of=/dev/fd0
```

Tím jsme přenesli instalační obraz ze souboru `bare.i` na disketu v první disketové mechanice. Pokud používáte z nepochopitelných důvodů systém strýčka Billa, pak hledejte program `rawrite.exe`. Druhou disketu naplníme obrazem instalačního svazku. Máme na výběr. Soubor `rootdsk/color.gz` obsahuje textový instalátor, ale v barvičkách a s menu. To je dobrá volba, pokud máte barevný monitor. Pokud máte černobílý, nevidíte modré podsvícení zvolené položky menu, což je škoda. Musíte tedy zvolit čistě textovou instalaci – soubor `text.gz`. Oba soubory jsou zabalené gzipem a my je příkazem `gunzip soubor` rozbálíme. Děláme to proto, že náš drobeček má malou RAMku a ušetříme si tak problémy s nedostatkem paměti a vypnutým ramdiskem. Příslušný rozbalený soubor přeneseme na druhou disketu:

```
dd if=rootdsk/color of=/dev/fd0
```

Podotýkám, že disketa nesmí být přimountovaná, když používáte `dd` (pro jistotu tedy můžete použít `umount /dev/fd0`). Nyní vložíme do tříosmšestky první disketu a spustíme ji. Počkáme, až zobrazí výzvu `boot:`. Nyní záleží na velikosti RAM. Se čtyřmi MB se nepodaří vytvořit ramdisk, takže zadáme

```
mount root=/dev/fd0 ramdisk=0
```

čímž přimountujeme disketu v první mechanice jako kořenový svazek a zakážeme ramdisk. Pokud naše mašinka oplývá dostatečnou pamětí, stačí spustit normální instalaci s ramdiskem stiskem enteru. Za chvíli se objeví žádost o další disketu (*VFS: Insert root floppy disk to be loaded into ramdisk and*

press *ENTER*). Může být poněkud matoucí, že hláška „to be loaded into ramdisk“ se objeví i při zakázaném ramdisku, ale to je normální. Vyměníme tedy bootovací disketu za disketu s instalačním obrazem color nebo text, dle volby. Drobeček bude chvíli chroustat novou disketu a pak napíše `slackware login:`. Ano ano, instalace je připravena.

Rozdělení disku

Přihlásíme se jako root bez hesla. Dalším krokem bude příprava disku. Spuštěním `fdisk /dev/hda` změním rozdělení prvního harddisku na primárním IDE řadiči. Zadáním `p` se vypíše aktuální rozdělení disku. Postupně stiskem `d` a čísla oddílu vymažeme všechny existující oddíly (patrně bude jen jeden, pak nemusíme zadávat číslo oddílu). Nyní pro změnu stiskem `n` vytvoříme nový oddíl. Zadáme primary partition – písmeno `p` – a `1` jako první oddíl. Můj disk má 250 MB, tak jsem zvolil 16 MB na swap – odkládací prostor Linuxu. Proto jsem pro první partition nastavil *First cylinder* na 1 a *Last cylinder* na +234MB. Opět jsem stiskl `n`, pak `p` a `2`. Vytvářím tak druhý oddíl – to bude swap. Jeho *First cylinder* nechám stiskem enteru na standardní hodnotě a do hodnoty *Last cylinder* opíši druhé číslo z nabízeného intervalu.

Teď zbývá sdělit systému, co je který oddíl. Stisknu tedy `t` a zadám `1` na dotaz k číslu partition. Tento oddíl má být ext2, takže bude mít *Hex code* rovno 83. Následuje určení typu druhého oddílu: `t`, pak `2` a jako *Hex code* bude 82, což je Linux swap. Znovu si vypíši stiskem `p` rozdělení disku. Teď se mi líbí, proto se můžu pochválit a stisknout `w` – uložit. Pokud by se mi nelíbilo, stisknu `q` a ukončím `fdisk` beze změn.

Smutné je, že spuštěný shell nemá historii, a proto budu muset příkaz `fdisk` znovu vyťukat, pokud chci něco opravit. Mimochodem, kurzorové šipky nemanipulují s historií, ale s kurzorem, který pak vesele běhá po obrazovce. Bohužel si shell každý takový stisk pamatuje a pokud zadám např. *šipka* a `cd`, hledá se nikoliv příkaz `cd`, ale *šipka cd*, který samozřejmě nenalezne. Konec teoretizování. Teď si ručně zaktivujeme swap. Stačí zadat příkaz `mkswap /dev/hda2`, který z druhého oddílu udělá swap, a `swapon /dev/hda2`, který swap zinicilizuje. To je nutné, pokud máme malou paměť, ale tak už to u tříosmšestek bývá. Navíc se tím nic nepokazí.

Instalace

Teď stačí spustit instalaci příkazem `setup`. Nastavení klávesnice přeskočíme, protože české země Slackware 3.3 nezná a na francouzské klávesnici se blbě píše. Dále se vybírá swap (v grafické-color instalaci tedy začínáme 4. položkou). Volíme *yes*, protože Slackware je šikovný a určitě správně poznal, který oddíl to má být. Dále se dovídáme důležitou informaci, že pokud je již náš swap aktivní, nesmíme instalátoru povolit jeho opětovnou inicializaci. Proto na zdvořilou otázku *Do you want setup to use mkswap...* zdvořile odpovíme *no* (opět v instalaci z obrazu color je třeba odpovědět *no* na další dotaz ke spuštění `swapon`).

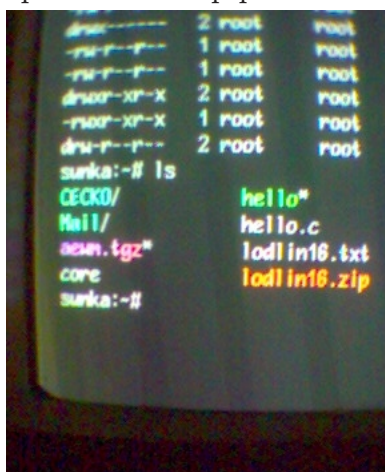
Následuje volba *root Linux partition*. Naším kořenovým svazkem bude `/dev/hda1`, který jsme před chvílí vytvořili. Instalátor by rád věděl, zda má oddíl nechat beze změny nebo formátovat. Zvolíme formátovat a v následujícím dotazu si vybereme 4096 inodů (budeme moci mít víc souborů, zjednodušeně vzato). Pokud máme připojený disk s nelinexovou partition, bude chtít instalátor vědět, zda ji má zahrnout do `/etc/fstab`, tedy zda na ni budeme v budoucnu chtít přistupovat z Linuxu. No proč ne, že?

A pokračujeme výběrem instalačního media. Já zvolím instalaci z harddisku a v následujících dvou dotazech odpovím `/dev/hdb1` a `/`. Je to proto, že mám připojený disk s jedinou partition a instalačkami na prvním IDE řadiči jako slave (`/dev/hdx` je harddisk, místo x je a, b, c nebo d podle posloupnosti 1. ide master, slave, 2. ide, master, slave; následuje číslo partition). Navíc adresáře s balíčky (ap, e1, a další) jsou přímo v kořenovém svazku disku, tedy proto `/`. Kdo může válčit z CD-Romky, ať to zkusí. Nejhorší varianta jsou diskety.

Výběr balíčků

Pokročíme k výběru balíčků. Vybíráme podle velikosti disku. Já jsem na svůj 250 MB neinstalovat akorát zdrojáky jádra, Emacs (Vim, co je lepší ;-)), FAQ, a X server development kit a vystačil jsem. Pravda, z jednotlivých sad jsem si velmi vybíral, a tak přišel zkrátka i `apache` a `sendmail` nebo třeba dokumentace k X. Při instalaci z color-obrazu jsem volil jako „prompting mode menu“. Vybírají se tak jednotlivé skupiny balíčků. Při akutním nedostatku místa na disku lze výběr provést důkladněji přes volbu `expert`, kdy se vybírá po balíčcích.

V závěru instalace nás čeká ještě pár úkonů. Prvně přeskočíme instalaci jádra, protože jsme při instalaci balíčků prozíravě vybrali jádro ze skupiny A (Base). Přeskočíme i tvorbu bootdisku a vrhneme se na konfiguraci modemu, myši, cdromky a výběr fontu. Doporučuji `t.fnt` – vypadá opravdu stylově. Teď to nejdůležitější – instalace zavaděče systému – `LILO`. V instalaci s barevným menu stačí dát `begin`, odentrovat rozšiřující volby a vybrat instalaci do MBR a bez čekání. Pak přidáme linuxový oddíl zadáním `/dev/hda1` a názvu – třeba Linux. Pak potvrdíme `install` a je to. Jak triviální. Následuje konfigurace sítě. Já jsem se chystal propojit tohoto drobečka přes paralelní laplink, a tak jsem mu dal jméno `sunka`, IP adresu 192.168.1.4, doménu `doma` a masku 255.255.255.0. Již jsme v cílové rovině, a tak vybereme timezone Europe/Prague, zapíšeme `fstab` a dozvíme se, že instalace je kompletní. Opouštíme setup přes `exit` a vypínáme stroj příkazem `shutdown -h now`.



Mám dva monitory – co s tím?

Jsa hrdým vlastníkem dvou bezprizorních grafických karet (Hercules a Trident) a monitorů, připojil jsem ke své tříosměstce obojí. Pokud si to chcete zkusit také, neváhejte. Problém je jediný. Jádro nedokáže využít druhý monitor, takže shell na něm nepojede, jak jsem původně doufal. Zbývá tedy varianta práce v X. Bohužel XFree86 ve verzi 3.3.1 ještě nemá Xineramu – podporu dvou monitorů. Co tedy můžeme udělat? Buď se spokojíme s tím, že druhý monitor bude jen pasivní zobrazovací prvek (na objekty na něm nebudeme moci kliknout), nebo zkompilujeme XFree ve verzi 4.0 s podporou Xineramy a optimalizací pro procesory 386. Druhá varianta je sebevražda, proto jsem se rozhodl pro první. V `/etc/X11/XF86Config` odkomentujeme další sekci `Device`, `Monitor` a `Screen`, popřípadě zduplikujeme původní a změním jména. Vlastníci černooranžových, černozeleňých či černobílých monitorů budou hledat sekci `Screen` s položkou `Driver "mono"`. V nainstalovaném `XF86Config` stačí jen chvíli hledat. Mimochodem, máte-li pouze černobílý displej, zakomentujte všechny sekce `Screen` s driverem jiným než „mono“, jinak X nenastartuje...

Po startu X vidíme obraz na obou monitorech, ale jak se na ten druhý dostat? Spustíme `xterm` a zadáme `export DISPLAY=:0.1`, což způsobí, že následně spuštěné (ručně z tohoto `xtermu`) X-aplikace se objeví na druhém monitoru. Aplikace typu kalkulačka tam ale nemají smysl, protože jak bylo zmíněno, nelze na ně kliknout...

A co teď s tím?

Teď znovu nastartujeme. Root má stále nepřidělené heslo, a tak jestli to trápí naše administrátorské svědomí, nastavíme ho příkazem `passwd`. V dalším kroku se pustíme do nastavení spojení přes paralelní laplink s druhým strojem, který se jmenuje *bedna*. Přidáme tedy do `/etc/hosts` dva řádky:

```
192.168.1.1 bedna bedna.doma
192.168.1.4 sunka sunka.doma
```

V `/etc/rc.d/rc.modules` zakomentujeme řádek s modulem `lp`, který by obsluhoval tiskárnu a naopak odkomentujeme modul `plip`. Připíšeme ještě `ifconfig plip1 sunka pointopoint bedna up` a `route add -host bedna plip1`. Kdo má – jako já – na „bedně“ síťovou kartu, kterou je napojena k dalším strojům ze sítě 192.168.1.0, nezapomene připsat také

```
route add -net 192.168.1.0 gw bedna plip1
```

To bychom měli síť na jedné straně. Na druhém stroji se musíme zbavit všech modulů pro obsluhu paralelního portu a pak provést:

```
insmod parport
insmod parport_pc io=0x378 irq=7
insmod plip
ifconfig plip0 bedna pointopoint sunka up
```

Někdy může být nutné přidat `route add -host sunka plip0`, ale soudobé distribuce by to měly nastavit samy. Nyní se můžeme zkusit připojit se na bednu. SSH není k dispozici, použijeme `telnet`. Můžeme spustit X server. Prvně je třeba editovat `/etc/X11/XF86Config`, ve kterém musí být odkomentována (zbavena úvodního znaku `#`) alespoň jedna sekce `Device`, `Monitor` a `Screen`. Výběrem sekcí s označením `generic` nic nezkazíme. Podstatné je, aby byly zvoleny na sebe navazující sekce, tedy položky `Device` a `Monitor` v sekci `Screen` musí nést stejnou hodnotu jako je název příslušných odkomentovaných sekcí. Po uložení konfigurace a spuštění `startx` naběhne `fvwm95`. K tomu je zapotřebí alespoň 8 MB RAM. Míň nemá smysl. Pokud se divíte, že není vidět tlačítko „Start“, je to proto, že obrazovka má nastavené větší virtuální rozměry. K liště se lze dostat pouhým stáhnutím myši ke spodnímu okraji.

Asi si teď říkáte, že `telnet` jede, X server taky, ale k čemu je to všechno dobré? Chtělo by to nějaké rozumnější aplikace. Nuže lze použít Vim ve verzi 4.6, gcc ve verzi 2.7.2.2, gdb 4.16, několikero her pro shell v adresáři `/usr/share/games`, BASH je ve verzi 1.14.7 i s historií (narozdíl od shellu při instalaci), mc 4.0, sc – tabulkový editor v shellu. Extrémisté mohou odzkoušet TeX (verze 3.14159). Bohužel v X toho moc nenaděláme. Kromě standardní sady aplikací typu kalkulačka nebo hodiny tu není moc co k vidění. Za zmínku ale stojí webový prohlížeč Arena. Má vskutku svébytné ovládání a vzhled. Ideální hračka pro studenta informatiky a jiné příznivce IT...

Odkazy

[1] <ftp://ftp.slackware.at/>

Solartec – síť postavená na Linuxu

Petr Bravenec

Případová studie mapující nasazení linuxových síťových služeb ve firmě Solartec. Linux zajišťuje provoz souborových, aplikačních, databázových a poštovních serverů. Uživatelské desktopy s Windows komunikují pomocí Samby, u výrobní linky slouží Lynx.

Úvod

Když jsem napsal první verzi tohoto textu, byl jsem zděšen jeho suchopárností a sebechválou. Poprosil jsem tedy některé kolegy, aby si tento text přečetli a doplnili jej o některé jedovatosti, kterými obvykle moji práci komentují. Nepodařilo se. Abych vám čtení textu aspoň trochu ulehčil, sebral jsem zrcadlo, papír a tužku a nyní máte (asi) první příležitost číst na abclinuxu článek s ilustracemi.

Z technických důvodů v tomto textu nejsou, naleznete je v [on-line verzi \[1\]](#) článku, pozn. ed.)

Firma [Solartec \[2\]](#) je jediný výrobce solárních článků v České republice. Velká část aktivit firmy je však orientovaná na problematiku související s fotovoltaikou a s obnovitelnými zdroji energie vůbec. V posledních letech se celý obor dynamicky rozvíjí a spolu s tím narůstají jak nároky na síťové služby, tak možnosti přístupu k různé technice.

Když jsem před čtyřmi lety do firmy nastoupil, byl nejsložitějším výpočetním systémem MS Excel s databází naměřených solárních článků – týdenní statistické přehledy dělal kolega ručně a to, co by se v databázi řešilo několika SQL dotazy, trvalo pár hodin.

Svou kariéru jsem začínal na počítačích [EC1033 \[3\]](#) (nebyl na tom unix), potom SMEP s krátkým obdobím na PC a FoxPro (ve srovnání se SMEPem bylo AT jen hračka) a od roku 1993 jsem pracoval především na operačních systémech unixového typu. Windows jsem na svém desktopu smazal v roce 1994 a o rok později jsem nasazoval Linux poprvé v podnikovém prostředí jako souborový server pro přibližně stovku uživatelů. Zkušeností a kuráže pro vystavění podnikové sítě na Linuxu jsem měl tedy dost a o tom správném operačním systému na servery jsem neměl nejmenší pochyby.

Jediný počítač s Linuxem byl ze začátku firewall, který byl součástí dodávky internetového připojení. I když je to dnes beznadějně zastaralé PC (Pentium 75 MHz, 32 MB RAM a 800 MB disk), slouží po doplnění dalšími síťovými kartami, VPN a velkým množstvím pravidel do iptables dodnes. Postupně přibývaly servery s databází, web serverem pro intranet, poštou a s dalšími službami a dnes je jenom v serverovně sedm počítačů s Linuxem.

Na uživatelských počítačích MS Excel s odpovídajícím okýnkovým podkladem pochopitelně zůstal. Uživatelé si na svá Windows nenechají sáhnout a vyhrožují, že když jim na to sáhnou, sáhnou oni na mě. Spokojený je Microsoft, spokojení jsou uživatelé, nespokojený je jen můj plastický chirurg, který mi s optimistickým výrazem našeptává: „...ale klidně jim na to sáhněte, nějak to spíchnem...“

Databáze

Z mého pohledu je pro chod každé firmy jedním z nejdůležitějších systémů databáze. Z dřívějšíka jsem měl zkušenosti s databázovými stroji CA Ingres a Oracle. Z open source databází jsem měl nejvíce zkušeností s databází PostgreSQL, i když jsem nikdy neměl možnost nasadit tuto databázi v podnikovém prostředí. Během uplynulých čtyř let se databáze ukázala jako velmi výkonná a mimořádně spolehlivá zvláště ve srovnání s Ingresem, i když toto srovnání není zcela fér (s Ingresem jsem pracoval zhruba před deseti lety a během té doby se vývoj určitě nezastavil). Pro naše využití by (nejspíš) naprosto nepřípadala v úvahu databáze MySQL, především kvůli nemožnosti řádně definovat vztahy mezi tabulkami a z toho vyplývající nespolehlivosti a výrazně nižšímu výkonu. V databázi je dnes

přes dvě stě tabulek (jistě ne všechny se používají). Největší tabulka má téměř devět miliónů záznamů, další největší v pořadí dva a půl miliónu. Další tabulky jsou podstatně menší, řádově desítky až stovky tisíc vět, většinou však jen pár stovek. Databázový stroj je relativně hodně zaměstnaný – běžící výrobní linka dovede krmit databázi několika stovkami záznamů za minutu.

V databázi jsou uložena data o výrobě, docházkový systém, internetové stránky, ceníky, internetový obchod a data různých projektů z oblasti fotovoltaiky ([Sledování vlivu znečištění na fotovoltaické panely](#) [4], [Fotovoltaická fasáda Corinthia Panorama hotel](#) [5]).

Až na výjimky jsou databázové aplikace vytvořené v PHP a tvoří součást intranetu. Skripty pro pořizování dat jsou z velké většiny relativně jednoduché a velká část aplikační logiky je přenesena na databázi. Část nejpoužívanějších skriptů je kvůli pohodlí při zadávání dat optimalizovaná pro textový prohlížeč Lynx. Nenáročnost textového prohlížeče umožňuje využít na výrobní lince i jinde pro zadávání dat stroje, které mají pomalu víc ventilátorů, než bogomipsů – počítače 386 s připojenými sériovými terminály (samozřejmě s Linuxem).

Poštovní systém

Dlouhá léta jsem používal na poštovních serverech sendmail. Kdo někdy zkusil dělat nějaké změny do konfiguračního souboru sendmailu, jistě mi dá za pravdu, že konfigurovat sendmail se podobá spíše práci kryptoanalytika a v některých vyhraněných případech se pro sendmail charakteristická změň interpunkčních znamének musí napsat i do konfiguračního předpisu pro preprocesor m4. Takže i když jsem investoval do sendmailu veliké množství času a energie a dokonce jsem držitelem certifikovaného glejtu o úspěšném absolvování týdenního kursu s hodnocením „prospěl na výbornou“ (kurs nebyl dostatečně náročný), rozhodl jsem se, že na novém působišti se naučím pracovat raději s novým poštovním serverem.

V roli MTA tedy vystupuje Postfix. Ve srovnání se sendmailem se konfiguruje velmi snadno, jeho modulární koncepce dává naději na vyšší bezpečnost a v neposlední řadě se Postfix relativně snadno doplňuje o různé další moduly (deliver agenty, antiviry a podobně). Uživatelé samotní mají poštu uloženou na serveru v IMAP složkách (Cyrus IMAP server). Celé to mírně komplikuje pouze MS Outlook, který ani v poslední verzi nedovede pracovat s IMAP složkami na uspokojivé úrovni – tvrdošíjně skladuje odesílanou poštu do lokálních složek a jeho způsob ukládání off-line obsahu zcela znemožňuje používání cestovních profilů. Uživatelé jsou nuceni buď čekat dlouhé minuty při každém odhlášení/přihlášení, nebo musejí mít MS Outlook nastavený ručně na každém počítači, ke kterému se přihlašují. Na MS Outlooku je znát, že je to svět sám pro sebe, a že by mu opravdu svědčilo spíše prostředí s MS Exchange serverem, normální poštovní protokoly tak docela nezvládá.

Součástí pošty je i ochrana proti nežádoucímu obsahu. Na serveru neběží žádný antivirový program – takřka všechny nebezpečné přílohy dovede odfiltrovat procmail sanitizer. O redukci nevyžádané pošty se přímo na serveru stará Spamassassin – žádná pošta typu „Four inches in two weeks!!! Impress your girl!!!“ už nemusí snižovat sebevědomí pánů v naší síti.

Samba

Protože většina uživatelských PC běží v prostředí Windows, je pro uložení uživatelských dat nejvhodnějším řešením Samba. Ta slouží jako primární řadič domény - na serveru jsou uloženy všechny uživatelské údaje. Samba slouží i jako společný tiskový server pro všechny tiskárny ve firmě. I když tiskárny jsou rozházené různě po republice, uživatelé je mohou najít vždy na jednom společném místě. Uložení uživatelských dat a účtů na jednom místě usnadňuje zálohování a jemnější rozdělení přístupových práv mi pomohlo eliminovat různá sdílení, která jinak měli uživatelé ve zvyku vytvářet na svých discích.

Zálohování

Při mé nedávné návštěvě v bývalém působišti mi ukazoval kolega zařízení se spoustou pásek – elektronický trpajzlík na povel systému sebral potřebnou pásku a po krátké poradě s ostatními pidizameštnaci zálohovacího zařízení běžel vrazit pásku do nejbližší volné mechaniky. Bylo by krásné mít do sítě připojené podobné zařízení, ale celá legrace by stála více, než všechny servery dohromady. Takže je zálohování o cosi levnější – nejjednodušším řešením bylo vyčlenit pro zálohy jeden počítač s dostatečnou diskovou kapacitou. Zálohy probíhají bezobslužně každou noc protokolem rsync. Počítač je umístěn odděleně od ostatních, aby při případné návštěvě zlodějů, vody, ohně či jiné nepřijemnosti nepřišla firma o veškerá data.

Hardware

Veškerý hardware je vybírán z toho, co se dá celkem běžně koupit v obchodech s počítači – v případě problémů je tak možné sehnat potřebné náhradní díly velmi rychle. Dobrá polovina serverů je tvořena počítači, které už se díky svému výkonu nehodily k nikomu na stůl, ale ještě mohou posloužit třeba jako DNS nebo FTP server. U těchto služeb není výkon rozhodující, ale z bezpečnostních důvodů je vhodné oddělit tyto procesy od vnitřní sítě a od firewallu. Nejčerstvější přírůstek do rodiny serverů je PC s dvoukernelovým AMD Athlon64 X2.

Distribuce

Z dřívějšíka jsem byl zvyklý na SVR4 a v době, kdy jsem s Linuxem začínal, měl k SVR nejbližší RedHat. Bohužel podpora volně dostupných distribucí pomalu končí a vývoj nastupující Fedory mi připadá poněkud překotný a nekontinuální.

Nechal jsem se přesvědčit, že na čtyřiašedesátibitový počítač je nejlepší distribucí Gentoo. Tento systém mě při instalaci značně potrápil, ale po zprovoznění se mi zdá Gentoo přímočařejší a trochu jednodušší na údržbu. Dovedl jsem se v systému zorientovat mnohem rychleji – do administrace mi nezasahují žádné klikací konfiguratory, protože jsem si žádné nenainstaloval, a konfigurační soubory jsou přesně tam, kde bych je hledal. Celé nastavení je optimalizované pro můj nejoblíbenější konfigurační nástroj – editor vim.

(Přečetl jsem si, jak tady chválím Gentoo – přitom Gentoo na mě zanechalo ve skutečnosti spíše dojem nenainstalovatelného systému se špatnou dokumentací, s velkým množstvím chyb a se svérázným systémem na jejich odhalování a odstraňování – problémy, se kterými se potýkám, jsou dávno zařazeny v kategorii „vyřešeno“. Systém se většinu času pohybuje mezi dvěma stavy - nefunguje nebo se překládá, případně ještě přesněji nefunguje a zároveň se překládá. No co už, nainstaloval jsem to, tak bojuji.)

Linux na desktopu

Na několika místech se používá Linux i na desktopu. Běžní kancelářníci, jak už jsem uvedl, Linux nechtějí, ale na výrobní lince, kde nejnáročnější činnost sestává z vyplňování intranetových formulářů, prostě dostali na stůl předpotopní 386 s prohlížečem Lynx. Nejpokročilejší kancelářskou technikou na lince je staré Pentium s 32 MB RAM – Linux dovoluje, aby aplikace běžely na serveru. Uživatelé tak mohou i na takovém stroji používat internet, poštu a OpenOffice.org.

Veškerou produkci výrobní linky je třeba překontrolovat a změřit. V tomto případě slouží opět Linux. K počítači s Linuxem je připojený jednoúčelový přístroj s jednočipem AVR, samotná aplikace běží v prostředí X11. Procesor AVR samozřejmě není natolik výkonný, aby mohl utáhnout celý linuxový cirkus – Linux byl ale využitý při vývoji aplikace pro jednočip (překladač gcc).

Pražská pobočka

I když má Solartec sídlo ve valašském křemíkovém údolí – v Rožnově pod Radhoštěm – má svou pobočku i v Praze. Pobočka je připojena k mateřské firmě přes VPN, všechno obstarává samozřejmě Linux. Práce na serverech v pražské pobočce se liší od práce na rožnovských serverech pouze v jednom bodě – při případném restartu některého serveru trnu, jestli nebude třeba server chtít překontrolovat disky ručně. Naštěstí i na dálku lze odmontovat většinu disků a v readonly režimu překontrolovat čas od času i disk s běžícím systémem. Správa Linuxu na dálku je natolik jednoduchá a spolehlivá, že jsem se dokázal vyhýbat návštěvě pražské pobočky téměř rok. Kancelář jsem poprvé navštívil, až když bylo třeba posílit pražské pracoviště o další kus hardware.

Linux netradičně

Různé projekty v oblasti obnovitelných zdrojů energie předpokládají, že zařízení budou monitorovaná přes internet. Připojovat k maličké fotovoltaické elektrárně PC není příliš produktivní, monitorovací zařízení (PC) by spotřebovalo veškerou vyprodukovanou energii. V současné době experimentujeme s jednodeskovými počítači s procesory PXA250 – operační systém těchto zařízení je samozřejmě opět Linux.

Monitorování

Činnost většiny serverů je sledovaná periodicky spouštěnými jednoduchými skripty, které hlídají například zaplnění disků, zatížení procesorů či funkčnost některých subsystémů. Případné problémy se okamžitě hlásí zasláním SMS na mobilní telefon. Monitorováním jsem byl nadšen až do čtrnáctidenní dovolené, na jejímž začátku se na serveru zacyklil šílený bludný proces a server každou hodinu poslal zprávu o tom, že procesor je v plamenech. Což o to – mobil se dá vypnout. Ale když jsem telefon po pár dnech zapnul, musel jsem na tu záplavu plamenných SMS zavolat chorvatské hasiče.

Závěr

Linux v Solartecu na desktopech běžně nenajdete. Zkušenosti ale ukazují, že tam, kde se uživatelé nedovedou bránit a nemají příliš specifické nároky, je Linux provozovatelný bez problémů i na desktopu. Nasazení na serverech je samozřejmě vyzkoušené i v mnoha jiných firmách a u nás ani nic jiného neběží. Linux je spolehlivý, výkonný a nenáročný na údržbu – velkou část údržby (updaty, restarty procesů a podobně) provádím v noci nebo o víkendech, připojený k síti obvykle jen přes GPRS kartu.

Odkazy

- [1] <http://www.abclinuxu.cz/clanky/ruzne/solartec-sit-postavena-na-linuxu>
- [2] <http://www.solartec.cz/>
- [3] <http://compmus9.valuehost.ru/histussr/1-33.htm>
- [4] <http://server.solartec.cz/clanek.php?CLANEK=82>
- [5] <http://panorama.solartec.cz/>

Jaderné noviny 326–330

Robert Krátký

Linux 2.6.13-rc7. Nové testovací vydání SPUFS; diskuze o umístění v adresářové struktuře zdrojových kódů. Ovladač čidla zrychlení HDAPS; problémy s rozpoznáváním hardwaru. Ovladač pro MPC8xx PCMCIA si našel cestu do jádra 2.6. Monitorování využití jader mezi dobrovolníky. Linux 2.6.13; nová pravidla pro rychlejší vydávání nových verzí jádra. Kontroverzní článek o stavu grafiky v Linuxu. Podpora Marvell SATA. Odstranění zastaralých funkcí; výzva správcům, aby aktualizovali ovladače. Hádky vývojářů kvůli pravidlům posílání patchů. Pár rad pro upgrade z 2.4 na 2.6. DevFS má stále hlavu na špalku; uživatelé se brání. Zpřístupnění NUMA dat uživatelskému prostoru. Umístění patchů čekajících na -stable jádro. Používání linux/irq.h nebo asm/irq.h v ovladačích. Automatická konfigurace jádra. Russellovy projekty. Stěhování master.kernel.org do nového domova. Linux 2.6.14-rc2. uclinux portován na Blackfin CPU. Vytížení Andrewa. Linux 2.6.14-rc2-mm1. Správa subsystému I2C. Cogito 0.15.1 vydáno s důležitou opravou.

Linux 2.6.13-rc7, 34 e-mailů

23. srp – 29. srp

Linus Torvalds oznámil Linux 2.6.13-rc7:

Už jsem chtěl opravdu vydat 2.6.13, ale zatímco jsem čekal na vyjasnění dalších problémů, proběhlo dost změn, takže myslím, že bude lepší nechat ještě -rc7. Většina změn v -rc7 je triviálních, buď jednořádkové opravy nebo se týkají nějakého konkrétního ovladače či neobvyklé konfigurace. Krátký seznam změn (připojen) by měl docela dobře osvětlit, o co šlo.

Nové testovací vydání SPUFS; diskuze o umístění v adresářové struktuře zdrojových kódů

25. srp – 29. srp

Arnd Bergmann napsal: Jde o pracovní verzi souborového systému SPU. Od minulé verze bylo přidáno hodně funkcí a opraveno mnoho chyb. Máme už podporu pro read() souboru /run, aby bylo možné spouštět na SPU kód. Bylo přidáno několik souborů, které poskytují přístup k více hardwarovým funkcím, hlavně kanálům pro hlášení signálů [signal notification channels].

Ale především máme funkční ukládání a obnovování SPE, které napsal Mark Mutter. Výsledkem bude jaderný plánovač [scheduler] pro SPU. A protože to souborový systém dost nafouklo, rozdělují jej teď na více menších patchů.

Během následujících týdnů provedeme ve zdrojácích rozsáhlejší změny, takže tato verze je pravděpodobně poslední, která je binárně kompatibilní s předchozími. Máte-li specifické požadavky, které spufs v současné podobě nespĺňuje, je teď vhodná doba nám o nich dát vědět.

Pekka Enberg odpověděl: Jsem zmaten. Kód se liší na různých architekturách a provádí I/O na zařízení. Proč to chcete dávat do fs/ a ne drivers/?

Arnd vysvětlil: Nikdy jsem o tom nepřemýšlel jako o ovladači zařízení, ale spíš jako o rozšíření pro architekturu – proto byl kód původně v arch/ppc64/kernel. Vzhledem k tomu, že pracuje s VFS, je teď ve fs/spufs. Jinak mi ale na umístění nezáleží. Já bych navrhoval následující místa (řazeno od nejlepšího).

1. arch/powerpc/platforms/cell

2. arch/powerpc/spe
3. fs/spufs
4. drivers/spe

1) By bylo místo, kde bych chtěl stejně mít nízkoúrovňový kód (arch/ppc64/kernel/spu_base.c), takže dává smysl tam mít vše, co spravuji.

2) By mohlo být lepší, budeme-li mít v budoucnu v arch/powerpc více druhů platform, které by používaly SPE. Např. pokud chceme podržet kód pro Playstation odděleně od obecného Cell.

Pekka odpověděl: Nemám nic proti 1, 2 a 4. Samozřejmě byste mohli abstrahovat části spufs, které slouží obecným účelům (kdybychom například měli podobná rozšíření architektury, která by mohla sdílet kód), a dát je do 3, ale to by asi způsobilo zbytečné komplikace.

Ovladač čidla zrychlení HDAPS; problémy s rozpoznáváním hardwaru, 36 e-mailů

26. srp – 30. srp

Robert Love napsal: Pracuji na ovladači pro IBM Hard Drive Active Protection System (HDAPS), který poskytuje čidlo zrychlení ve dvou osách a různá další data. Ten hardware najdete v nových notebookech IBM ThinkPad.

Brian Gerst se zeptal: Existuje nějaký způsob, jak zjistit, jestli je zařízení v systému (PCI, ACPI atd.), aniž by se zkoušely porty?

Robert odpověděl: O žádném nevíme. Je to zařízení pro zastaralou platformu. Takže bohužel žádná probe() rutina.

Arjan van de Ven navrhl: DMI určitě...

Robert na to řekl, že rád přijme patche a Alan Cox napsal: Myslím, že by to před začleněním mělo být opraveno.

Robert odpověděl: Ujasněme si to. Existuje init rutina, která kontroluje přítomnost zařízení. Jen není k dispozici rychlý, jednoduchý a neagresivní test. Ovladač by se však určitě nenatáhl na notebooku bez potřebného hardwaru.

Dave Jones poznamenal: Zkoušení I/O portů kvůli hardware, když tam zařízení není, může být osudné. Co se stane, když budu mít na I/O portu 0x1600 něco úplně jiného? (A tak projdu tvým testem request_region()). accelerometer_init() pak začne vesele zkoušet porty a provádět všelijakou další černou magii. A poblíž řekl Jeff Garzik o DMI: Protože je takový test možný, jde určitě o věc, kterou je třeba před začleněním napravit.

Robert odpověděl: Zaprvé, já přeci Linuse nežádám, aby to začlenil. Všichni by se měli z hluboka nadechnout. Zadruhé, nevíme, jestli bude řešení pomocí DMI fungovat. Vyzkouším to.

Ovladač pro MPC8xx PCMCIA si našel cestu do jádra 2.6

29. srp – 1. zář

Marcelo Tosatti napsal: Posílám ovladač 8xx PCMCIA, který původně napsal Magnus Damm, s několika vylepšeními a opravami. Ovladač byl začleněn do 2.4, ale nikdy nebyl portován na 2.6. Prosím o kontrolu, uvítám komentáře (včetně estetických záležitostí).

Magnus Damm reagoval: Je fajn, že byl ovladač portován na 2.6. Tenkrát jsem ho napsal pro pcmcia-cs, ale po čase se dostal do 2.4. Děkuji všem, kdo přidali kód a opravy. Nevím moc dobře, jak funguje současná linuxová PCMCIA vrstva, a už se nezabývám PowerPC, takže samotný port komentovat nemohu.

Monitorování využití jader mezi dobrovolníky, 33 e-mailů

29. srp – 1. zář

Andrea Arcangeli napsal: Během Kernel Summitu někdo začal mluvit o tom, že není jasné, jak moc jsou jádra rc/pre/git testována před vydáním finální verze. Připravil jsem tedy server, který automaticky sleduje, jak moc jsou jednotlivá jádra testována. Budou to samozřejmě velmi hrubé odhady a ani to nemůže být věrohodné, ale možná to bude užitečné. Pokud to bude k ničemu, tak se nic neděje, protože jsem s tím strávil jen chvíli ;). Podrobnosti najdete na stránkách projektu: <http://klive.cpushare.com/> [1]

Kompletní zdrojáky (včetně serveru) jsou zde:

<http://klive.cpushare.com/downloads/klive-0.0.tar.bz2> [2]

Spuštění klienta: `wget http://klive.cpushare.com/klive.tac`

Pak při každém bootu (třeba v `/etc/init.d/boot.local`): `twistd -oy klive.tac` Teoreticky bychom se klienta mohli zbavit úplně a udělat z toho konfigurační volbu do jádra. Ale nemám zdání, jestli půjde o užitečný projekt, takže se mi tomu nechce v tuhle chvíli věnovat příliš času.

Chvíli se diskutovalo o tom, jestli by to lidé neřadili do kategorie „spyware“. A Alan Cox k tomu v jednu chvíli řekl, že by bylo nutné počítat se sbíráním údajů jen od lidí, kteří by to výslovně dovolili:

Asi by to snížilo objem, ale zároveň zvýšilo kvalitu. Zvláště pokud by id z cookie mohlo být vloženo do bugzilly, a hlášení o hardware by bylo ve strojově zpracovatelném tvaru (takže by pak bylo možné ptát se např. takto: „spolehlivost s nVidia IDE“).

A Sven Ladegast souhlasil: Změna zdrojových kódů jádra, díky které by byla automagicky odesílána data někam po síti, nemůže být ve výchozí konfiguraci povolena.

A Alan dodal: Nepotřebujeme osobní údaje (email, jméno, ip, adresa atd.). Ale co by s tím šlo dělat, to by bylo opravdu zajímavé.

Linux 2.6.13; nová pravidla pro rychlejší vydávání nových verzí jádra, 42 e-mailů

28. srp – 3. zář

Linus Torvalds oznámil Linux 2.6.13: Tady to je. Nejhorší část 2.6.13 bude asi fakt, že jsme pro přidělování nepřidělených zdrojů začali využívat obecný kód pro nastavení PCI sběrnice. To odhalilo docela dost nepříjemných detailů, ale také by to mělo znamenat, že hodně notebooků by mělo dokázat objevit PCI zařízení za mosty, které BIOS nenastavil.

Změny oproti -rc7 jsou malé, připojuji celý shortlog a diffstat. Co se nových pořádků týče: většinu příštího týdne budu pryč, ale obecně bychom se odted' měli snažit provést všechny velké změny (začlenění nového kódu) během prvních dvou týdnů po vydání verze. Pak přejdeme do klidového režimu, a pokud budete mít něco, co se nestihlo kvalifikovat, bude to muset počkat na 2.6.14. Měli bychom tak docílit kratší doby mezi verzemi, takže i věci, které prošvihly termín, nebudou muset čekat příliš dlouho na další.

Jesper Juhl napsal, že seznam změn [ChangeLog] obsahuje pouze patche od -rc7, a zeptal se, jestli by Linus nemohl poslat i kompletní 2.6.12 - 2.6.13. Linus odpověděl:

Stalo se. (No, chvíli bude trvat, než se objeví na mirrorech). Je to 2,3MB logů (i shortlog má přes 5000 řádků a 201kB, jestli vás to zajímá). Ten jsem na server nedával a konference mi nedovolí ho poslat sem. Ale uživatelé gitu mohou provést

```
git-rev-list --pretty=short v2.6.12..v2.6.13 -- git-shortlog
```

což ho vygeneruje. Pokud nechcete vidět nelineární začleňování, použijte „-no-merges“.

Jerome Pinot navrhl: Spuštěním gitu Linusově stromě

```
$ git-whatchanged v2.6.12..v2.6.13 --pretty=full
```

dostanete tohle: <ftp://ngc891.blogdns.net/pub/linux/misc/ChangeLog-2.6.12-2.6.13.txt> [3]

Ale pozor, má to 3,7MB.

A Linus, neúnavný apoštol gitu, odpověděl:

Daleko hezčí je provést

```
git log --no-merges v2.6.12..v2.6.13
```

což vrátí mnohem čitelnější výsledek. `git-whatchanged` se hodí, když chcete vidět soubory, které byly změněny (hlavně s parametrem „-p“, který zobrazí celý diff), nebo když chcete výstup omezit na určitý subsystém („`git-whatchanged drivers/usb`“). Ale když chcete jen vidět log, použijte „`git log`“. Parametr „`--pretty=full`“ ukáže informace i o člověku, který patch posílal (a s „`git log`“ to jde také), ale o to asi většina lidí nestojí. Ve skutečnosti je většinou lepší použít „`--pretty=short`“, což zobrazí pouze první řádky („`head-line`“ – řádek s předmětem emailu s patchem) zprávy. Kromě toho můžete výstup „`git log`“ poslat rourou na „`git-shortlog`“. Tak dostanete formát shortlogu (tj. pouze `head-line` a seříděné podle autorů).

Kontroverzní článek o stavu grafiky v Linuxu, 44 e-mailů

30. srp – 2. zář

Jon Smirl řekl: Napsal jsem článek, který zkoumá současný stav linuxové grafiky, a navrhuje možnou cestu dál. Jde o dlouhý text obsahující hodně technických podrobností, který by měl sloužit jako průvodce pro budoucí vývojáře. Pokud se podrobné části netýkají vaší oblasti práce, přeskočte je. <http://www.freedesktop.org/~jonsmirl/graphics.html> [4]

Mezi tématy jsou současný X server, `framebuffer`, `Xgl`, grafické ovladače, podpora více uživatelů, používání GPU a nový design serveru. Doufám, že vám článek pomůže s doplněním chybějících znalostí a budete si moct utvořit představu o oblasti grafiky. Text byl kontrolován, ale pokud i přesto obsahuje chyby, dejte mi prosím vědět. Názory na obsah také vítám.

Během diskuze, kterou zpráva vyvolala, zrušil Daniel Stone, administrátor serveru `freedesktop.org`, Jonův účet. Jon na to řekl, že Daniel se rozhodl můj článek o stavu X serveru cenzurovat. Jeho chabá výmluva je, že jsem přestal pracovat na jádře `Xegl`. Zdá se, že tisíce řádek kódu a spravování `fd.o` nehrají roli.

Daniel napsal v soukromém emailu: Provedl jsem několik čistek neaktivních účtů a projektů. Nejsi zdaleka první ani poslední. Už jsem takhle neuklízěl dlouho, protože nebyl čas. Měl jsem teď příležitost se na `fd.o` zase podívat, a když jsem si uvědomil, že tvůj účet nedělá nic jiného, než že hostí HTML stránku, tak mi to stačilo, abych začal s úklidem. A právě teď pracuji na mnoha dalších dřímajících přispěvatelích.

Po chvíli neveřejné diskuze si Daniel své rozhodnutí zjevně rozmyslel a Jonův účet a článek obnovil.

Podpora Marvell SATA, 21 e-mailů

30. srp – 7. zář

Brett Russ napsal: Tohle je první veřejná verze mého libata kompatibilního nízkourovňového ovladače pro rodinu Marvell SATA. V současné době běží úspěšně v režimu PIO na čipu 6081. Na podpoře EDMA se pracuje a měla by být brzy hotova.

Jeffu Garzikovi se kód líbil a chtěl ho dostat do oficiálního jádra co nejdříve. Brett se nevzpouzel a poslal aktualizovaný patch s korektním záznamem pro changelog a hlavičkou `Signed-Off-By`. Christoph Hellwig měl připomínky technického rázu, ale Jeff prohlásil, že chce patch i přesto protlačit. Christoph ho obvinil z přílišného autoritářství. Jeff reagoval:

Dokud nepřiložíš ruku k dílu při správě 2.4.x, jsi pouze brzda bez technických argumentů. Jestli se tak chceš chovat, dej se na politiku. Já mám na práci důležitější věci. Brett poslal další aktualizovanou verzi patche (a Bogdan Costescu napsal: *Díky! Na tento ovladač jsem čekal*). Jeff patch aplikoval a technická diskuze pokračovala.

Odstranění zastaralých funkcí; výzva správcům, aby aktualizovali ovladače, 10 e-mailů

31. srp – 7. zář

Russell King napsal: V souladu se souborem feature-removal.txt se chystám na brzké odstranění následujících funkcí:

- register_serial
- unregister_serial
- uart_register_port
- uart_unregister_port

Některé ovladače však tyto funkce stále používají:

```
drivers/char/mwave/mwavedd.c: return register_serial(&serial);
drivers/char/mwave/mwavedd.c: unregister_serial(pDrvData->sLine);
drivers/misc/ibmasm/uart.c: sp->serial_line = register_serial(&serial);
drivers/misc/ibmasm/uart.c: unregister_serial(sp->serial_line);
drivers/net/ioc3-eth.c: register_serial(&req);
drivers/net/ioc3-eth.c: register_serial(&req);
drivers/serial/serial_txx9.c: line = uart_register_port(&serial_txx9_reg, &port);
drivers/serial/serial_txx9.c: uart_unregister_port(&serial_txx9_reg, line);
```

Ty ovladače by skutečně měly být během pár dní opraveny, aby nepřestaly fungovat. Tímto tedy žádám správce zmíněných ovladačů, ať projeví trochu ochoty a své ovladače aktualizují. Bohužel některé z těch ovladačů neobsahují emailové adresy správců, ani nejsou uvedeny v souboru MAINTAINERS (mwavedd a serial_txx9). Tohle je poslední varování, pak budou funkce odstraněny. Kromě toho jsou tu následující ovladače, které deklarují funkce stejného jména. Jejich správci by měli zjistit, proč tomu tak je, a pokud to bude možné, odstranit je.

```
drivers/serial/crisv10.c:register_serial(struct serial_struct *req)
drivers/serial/crisv10.c:void unregister_serial(int line)
```

Alan Cox s Russellem vyřešil ovladač mwave a Max Asbock poslal patch opravující ibmasm.

Hádky vývojářů kvůli pravidlům posílání patchů, 15 e-mailů

8. zář

Russell King napsal: Koukám, že DaveM převzal správcovství sériového subsystému. V budoucnu tedy všechny patche prosím posílejte jemu, díky. (Od úterka jsou všechny sériové ovladače v ARM nefunkční.) Díval bych se na to možná jinak, kdyby mě alespoň ze slušnosti zařadil do CC: poslaného patche (o kterém už jsem řekl akpm, že by jej měl odmítnout). Díky. To máme další subsystém, se kterým si už nemusím dělat hlavu.

Alan Cox poznamenal: Pokud je to tak, nezapomeň prosím poslat Linusovi patch opravující soubor MAINTAINERS.

Linus Torvalds se do toho vložil: Pánové, přestaňte blbnout. Rusellovi už jsem poslal soukromý email. A Alane, nevidím žádný důvod, proč bys měl ještě přilévat olej do ohně. Občas se stane chyba, ale vztekáním ji nenapravíte. Místo toho je potřeba říct, že ten a ten je idiot, protože něco pokazil, a přimět ho, aby chybu zase napravit. Nemusíte to provést uhlazeně a nadávání nevádí. Takže místo „Už si s váma nechci hrát, protože Davem udělal chybu,“ radši řekněte něco jako „Davem je zas*aný, natvrdlý pitomec, provedl tohle a je to chyba, protože ...“ Vidíte? V obou případech si ulevíte.

V druhém případě bude navíc vypadat hloupě ten, kdo chybu udělal. Ale v prvním případě jste to vy, kdo vypadá hloupě.

Pár rad pro upgrade z 2.4 na 2.6, 5 e-mailů

8. zář – 9. zář

Weber Ress měl vést tým techniků při upgradu jader z 2.4 na 2.6 na mnoha serverech. Požádal o rady a Michael Thonke připomněl:

Google je tvůj nejlepší přítel a první zdroj informací.

Ještě přidal odkaz na článek [William von Hagen](#) [5] o tomto tématu. Jesper Juhl napsal:

Provádím upgrady mnoha kernelů, takže ti povím trochu o tom, jak to dělám a co doporučuji. Pak si s těmi informacemi nalož dle libosti :) Úplně nejdříve se přesvědč, jestli jsou všechny základní utility/nástroje aktualizované na verze, které budou fungovat s novým jádrem. Když stáhneš zdrojové kódy jádra 2.6.13, rozbal je a podívej se do souboru Documentation/Changes. Najdeš tam seznam nástrojů a utilit s uvedenými minimálními verzemi, které jsou vyžadovány pro správnou funkci s daným jádrem.

Až budeš mít aktualizované základní utility, musíš zkontrolovat, jestli budou s novým jádrem také fungovat všechny další důležité programy, které v systému máš. Jakmile si budeš jistý, že je všechno aktualizované tak, aby to fungovalo s novým jádrem, můžeš 2.6.13 zkompileovat a nainstalovat. Není nutné odstraňovat stávající jádro, 2.6.13 můžeš nainstalovat vedle starého a zkusmo s ním nabootovat. Kdyby nenaběhlo, můžeš se vrátit ke starému. Pro svou distribuci pravděpodobně najdeš v dokumentaci, která verze je „připravena na 2.6“. Já používám Slackware a Slackware 10.1 je kompletně připraven na 2.6, takže na systému se Slackware 10.1 je vše snadné. Stačí nainstalovat nové jádro místo 2.4 a je to – všechny nástroje si s tím poradí.

DevFS má stále hlavu na špalku; uživatelé se brání, 32 e-mailů

9. zář – 14. zář

Greg KH, kterému nebylo umožněno odstranit DevFS před vydáním verze 2.6.12, poslal stejné patche oproti 2.6.13; doufal, že tentokrát už se na ně dostane: Pokud by někdo opravdu moc chtěl devfs v jádře, poslal jsem patch, který to zařídí v 300 řádcích – ndevfs. Chcete-li jej používat, najdete ho v archívech (je snadné ho spravovat mimo strom jádra, protože potřebuje jen 3 háčky [hooks] do hlavního jádra).

Mike Bell odpověděl, že NDevFS je designově vadné. Vytváří pro zařízení zase další názvosloví. A co je horší, nefungují kvůli tomu zařízení jako ALSA a vstupní subsystém, jejichž umístění jsou natvrdo určená v knihovnách. Pokud nebude sysfs dostávat atributy, ze kterých půjdou odvodit skutečné názvy, nebude to fungovat.

Greg připustil, že NDevFS není pěkné řešení, ale jen alternativa. A dodal:

Kromě toho to vůbec nenabízím k začlenění do jádra, ale pouze jako důkaz pro ty, kteří tvrdí, že není možné snadno spravovat sadu patchů typu devfs mimo jádro. Jinde řekl David Lang, že při odstraňování DevFS je potřeba být opatrný, protože hodně systémů by mohlo přestat fungovat. A Greg se zeptal: Ok, jak dlouho bych měl čekat?

A David napsal: Kdyby byla ve verzi 2.6.13 odstraněna konfigurační volba, pak by podle mě měl kód zůstat do 2.6.15 nebo 2.6.16 (pokud budou nové verze každé dva měsíce, muselo by to být .16). Zvláště když se tolik lidí bojí řady 2.6, je nutné počkat alespoň jeden celý vývojový cyklus, spíš dva (a možná více, budou-li krátké) a potom dát pryč zbytek kódu v další verzi.

Nezapomínej, že distribuce nebalí všechna jádra. Přeskakují verze a dokud je nevyzkouší, nebudou vyřešeny všechny problémy. K tomu připočti, že je dost lidí zmatených číslováním verzí, takže protože je 13 liché číslo, považují 2.6.13 za testovací jádro a 2.6.14 za stabilní – na .13 se tedy nepodívají. Tohle všechno za předpokladu, že již byly vyřešeny problémy, které měli lidé se sysfs – nedokázalo nahradit funkce používané s devfs.

Vlákno skončilo bez jasného rozhodnutí o tom, jak dlouho ještě bude DevFS v jádře.

Zpřístupnění NUMA dat uživatelskému prostoru, 7 e-mailů

10. zář

Andi Kleen napsal: Právě jsem si všiml, že byl začleněn ten hnusný kód `/proc/*/numa_maps` od SGI. Několikrát jsem proti němu argumentoval a záměrně jsem podobnou funkčnost nepřidal při psaní kódu NUMA metod, protože nejde o dobrý nápad.

- Hodně nepěkného kódu.
- V této chvíli v podstatě jen debuggovací hack.
- Dává uživatelskému prostředí k dispozici hodně interních informací z jádra a o paměti (např. kolik lidí má namapovanou stránku), které by neměly být takto dostupné.
- Formát je velmi komplikovaný a je téměř nulová šance, že se v uživ. prostředí najde nějaký bezchybný způsob zpracování dat.
- Neexistuje žádné uplatnění, které by to potřebovalo (byl jeden teoretický případ, kde se to mohlo hodit, ale byla pro něj navržena lepší řešení).

Mohl by být ten patch odstraněn, prosím?

Andrew Morton řekl, že do fronty zařadil odstranění patche. Christoph Lameter měl ale pocit, že patch není beznadějný, a nechápal, proč by měl být vrácen. Andrew odpověděl: Je-li užitečný vývojářům aplikací, pak fajn. Je-li však užitečný jen pro vývojáře jádra, pak je argument slabší. V té oblasti však stále probíhá dost vývoje, takže pořád existuje důvod takovou monitorovací funkci v hlavním jádře mít.

Christoph reagoval: Pořád se mi nedaří pochopit, jak mohou lidi přijmout argumentaci, která říká: *Uživatelé nemají právo vědět, které uzly [nodes] operační systém alokoval pro proces, a také nemají vědět, jak se na oblastech paměti projeví práce s pamětí.*

Vývojáři aplikací pak musí hádat, jaký vliv má práce s pamětí na alokaci paměti. Při debuggování alokování paměti si dnes ti chudáci musí představit, co operační systém provádí. Prostřednictvím jiných záznamů v `/proc` vidí celkové množství paměti alokované na uzlu a pak na základě toho hádají, která aplikace si ji zabrala. Pak své aplikace upraví a zkusí to znovu.

V současné době mám za to, že by bylo lepší ponechat `/proc/<pid>/numa_stats` místo používání `smaps`, protože formát `smaps` je trochu ukecaný a bylo by těžké z něj vyčíst rozdělení alokace. Pokud použijeme `smaps`, bude asi potřeba nějaký nástroj, který by údaje zpracoval a prezentoval. `numa_stats` lze použít rovnou.

Přinejmenším potřebujeme mít možnost vidět, co se děje. Samozřejmě bychom byli radši, kdyby bylo možné měnit pravidla, která ovládají alokaci paměti. Argument, který říká, že vrstva to není schopná řešit, je pochopitelně pravdivý, protože pokusy o napravení těchto záležitostí byly zablokovány.

Andrew se začínal přiklánět k uznání této argumentace a ponechání patche, ale debata se nedobrala žádného jasného rozhodnutí.

Umístění patchů čekajících na -stable jádro, 3 e-maily

13. zář – 14. zář

Jean Delvare se zeptal: Je někde možné vidět patche navržené pro jádro -stable? Jsou emaily posílané do `stable@kernel` někde archivovány? Řekl bych, že by to bylo vhodné. Já bych například uvítal začlenění jednoho patche do 2.6.13.2, ale nechci hlásit již známý problém.

Michal Piotrowski poslal odkaz na [zkrácený log fronty pro -stable \[6\]](#) a Jean odpověděl:

Přesně to jsem potřeboval, už to mám v záložkách. Díky.

Používání linux/irq.h nebo asm/irq.h v ovladačích, 5 e-mailů

9. zář – 15. zář

Alexander Viro napsal: Často se objevují chyby v přenositelnosti kódu, když se někdo rozhodne začlenit do ovladače linux/irq.h místo asm/irq.h. Skoro vždy je to špatný postup a např. na ARM to způsobuje okamžité problémy. Takže udělám následující:

- Zkontroluji všechna stávající začlenění linux/irq.h; např. v arch/x86_64 byla až na dva případy všechna 100 % zbytečná, jedno mělo být asm/irq.h a druhé asm/irq.h + asm/hw_irq.h. Jediným oprávněným uživatelem linux/irq.h na amd64 je asm/hardirq.h. Situace na dalších místech v arch/* je podobná – většina začlenění není vůbec potřeba.
- Odstráním nesmyslná začlenění, architekturu po architektuře – u všech, které jsou v hlavním stromě. Přehodím ta, která by měla začleňovat asm/irq.h.
- Stávající obsah linux/irq.h dám do asm-generic/hardirq.h (ve skutečnosti je to přesně tohle – deklarace pro hardirq kód, relevantní na mnoha, ale ne všech architekturách).
- Přehodím zbývající uživatele linux/irq.h na asm-generic/hardirq.h (opět u architektur, které jsou v hlavním jádře).
- Nahradím obsah linux/irq.h tímto: `#warning` a `#include <asm-generic/hardirq.h>`.
- Po 2.6.14 zruším linux/irq.h úplně.

Námítky? Tahle varianta nechává lidem, kteří spravují kód mimo hlavní strom, na provedení oprav čas až do 2.6.15 – a to je opravdu více než dost. . .

Ale Geert Uytterhoeven navrhl: Nebylo by logičtější udělat preferovaný include z linux/irq.h? Většinou jsou verze v linux/* upřednostňovány před asm/*.

Matthew Wilcox odpověděl: Není téměř žádný důvod pro to, aby `<*/irq.h>` vůbec existovalo. Skoro všechny ovladače ve skutečnosti chtějí `<linux/interrupt.h>`.

Jenže Christoph Hellwig poukázal na to, že `<linux/interrupt.h>` neobsahuje `<asm/irq.h>` a některé architektury tam mají důležité prototypy.

Vlákno skončilo bez jasného rozhodnutí, jestli je lepší dávat přednost linux/irq.h nebo asm/irq.h.

Automatická konfigurace jádra, 42 e-mailů

14. zář – 17. zář

Ahmad Reza Cheraghi napsal: Napsal jsem systém pro vytváření souboru .config na základě hardwaru v systému. Bylo by fajn, kdybyste mi řekli váš názor.

úvod: <http://www.energyparty.de/ahmad/readme> [7]

jaderný patch: www.energyparty.de/ahmad/autoconfig_0.1_patch.tgz [8]

zdrojové kódy: http://www.energyparty.de/ahmad/autoconfig_0.1.tgz [9]

Hodně lidem se nápad líbil a Hua Zhong řekl: Vypadá to, že existuje trend odrazovat běžné uživatele od používání jader z kernel.org. Ale s tím já většinou nesouhlasím.

Roman Zippel se však stavěl skepticky k tomu, že by měl Ahmad (nebo kdokoliv jiný) dost energie na dokončení projektu. Měl pocit, že se jedná o obrovské množství práce, ale Ahmad a další věřili, že by šlo hodně obtíží obejít. Ahmad poblíž napsal: Chtěl bych říci, jak jsem na to přišel. Když jsem se poprvé bezúspěšně snažil nakonfigurovat jádro, pomyslel jsem si, proč nejde něco takového zařídit automaticky. Takže nejprve jsem to chtěl dělat např. pro ty, kdo neví co znamená volba „Enable loadable module support“, a jestli ji mají povolit nebo ne. Teď už vím, že všechny věci automaticky dělat nelze (například souborové systémy nebo protokoly). A to je důvod pro systém. Aby mohl každý (tím myslím experty) přispět návrhem nejlepší odpovědi, a tak by mohly být i volby, včetně té zmíněné, rozhodnuty automaticky. Takže když bude někdo chtít použít nové jádro, neskončí zklamaný a nebude litovat, že Linux (nové jádro) vyzkoušel.

Russellovy projekty, 7 e-mailů

15. zář

Alexander Viro poslal patch pro ovladač EPCA driver a Russella Kinga vložil do CC. Russell odpověděl: Díky za kopii, ale mě nezajímá žádný sériový ovladač, který nepoužívá hlavní sériové rozhraní. Nemám v úmyslu fungovat jako „člověk, který musí zkontrolovat každou změnu jen proto, že ovladač obsahuje slovo serial“ – to není role, kterou jsem se rozhodl zastávat.

Alexander reagoval: Hele, když vidím, jak vehementně si stěžuješ, když *nejsi* v CC... radši to hraji na jistotu, abych neskončil jako správce sériového subsystému ;-). OK, takže od jakých věcí chceš kopie? Já tedy tipuji, že to bude arch/arm/*, include/asm-arm/*, drivers/serial/*, include/linux/serial*. No, a všechny změny tty rozhraní, kdybys se někdy do něčeho takového pustil... Něco chybí nebo přebývá?

Russell odpověděl: V podstatě je to:

arch/arm/*	
drivers/*/arm	
drivers/mfd/*	(to platí teď, ale jestli bude i v budoucnu, to záleží na tom, co se tam objeví)
drivers/mmc/*	
drivers/serial*	(i když z toho jen ovladače využívající serial_core – bohužel to vypadá, že tam byly umístěny i ovladače, které serial_core nepoužívají)
include/asm-arm/*	
include/linux/8250*	
include/linux/serial*	
fs/adfs/*	

Ale jsem i autorem některých ovladačů a u těch bych také samozřejmě ocenil CC.

Stěhování master.kernel.org do nového domova, 12 e-mailů

16. zář – 18. zář

H. Peter Anvin hlásil: master.kernel.org bude offline počínaje dneškem krátce po 15:00 PDT/22:00 UTC, 16. září 2005; kvůli přesunu do Open Source Lab na oregonské státní univerzitě. Díky tomu budeme mít daleko lepší připojení a spolehlivější řešení záloh – kromě přístupu ke skutečnému NOC (síťové operační středisko). Velký dík Javieru Hendersonovi, který se nabídl, že stroj přiveze soukromým letadlem, aby se snížila doba offline. Předpokládáme, že bude zase online zítra kolem 12:00 PDT/19:00 UTC, v sobotu, 17. září. Nová IP adresa stroje bude 140.211.167.34.

Po přesunu napsal Michael Marineau:

Mimochodem, jste-li někdo zvědavý, tak Scott Kveton vystavil fotografie [uvítacího výboru](#) [10] z dnešního rána :-). Na jiném místě poznamenal Linus Torvalds: No, tak už je zpátky online a funkční, jen to vypadá, že zrcadlení na veřejné stroje kernel.org možná zase nebylo zapnuto.

A Kees Cook řekl: Kouslo se to, ale asi před hodinou jsem vše opravil. Právě byl dokončen první aktualizací cyklus a teď pracuje na kompletní denní synchronizaci, která bude chvíli trvat. Pak se zase naběhne na pravidelné aktualizací cykly.

Linux 2.6.14-rc2, 38 e-mailů

19. zář – 21. zář

Na mezinárodní den „Mluvte jako piráti“ [11] oznámil Linus Torvalds Linux-2.6.14-rc2:

Ahoj suchozemský krysy! Tady máte Linux-2.6.14-rc2. Nic moc extra, vy prašiví psi, ale byly začleněny změny v ALSA, LSM, audit a watchdog, které chyběly v -rc1. Také nová várka věcí od Andrewa. Ale kolem a kolem je to celkem rozumné – jak je vidět z připojeného zkráceného logu. Arrr!

Martin J. Bligh ohlásil: SCSI je v několika případech nefunkční kvůli absence patche, který příkládám. A to znamená, že některé z testovacích strojů nemohly nabootovat. Jamesi, je nějaká šance dostat to do hlavního jádra?

James Bottomley odpověděl: Už tam je... bohužel těsně po 2.6.14-rc2, ale když použiješ poslední verzi z git, měl by tam být. A Martin byl spokojen.

uclinux portován na Blackfin CPU, 8 e-mailů

20. zář – 27. zář

Luke Yang z firmy Analog Devices napsal:

Portovali jsme uclinux na náš Blackfin procesor. Kód je teď aktualizovaný pro 2.6.13. Patch pošlu do konference.

Zároveň se zeptal: Víím, že se blíží jádro 2.6.14. Mohl by být náš patch zařazen do 2.6.13?

Deepak Saxena odpověděl: Ne. 2.6.13 je pro nové funkce uzavřeno, stejně jako 2.6.14 (pokud by nešlo o extra zvláštní případ, o kterém by Linus myslel, že je natolik důležitý, aby ho nechal ještě proklouznout). V tuhle chvíli je nejlepší poslat patche ke kontrole, provést požadované změny a být připraven poslat patch oproti 2.6.14 během prvního týdne po vydání, aby mohl být začleněn do 2.6.15.

Luke tedy neváhal s patchem a Jesper Juhl navrhl pár technických věcí a rad k stylu.

Vytížení Andrewa, 28 e-mailů

20. zář – 26. zář

Eric W. Biederman poslal Linusi Torvaldsovi několik oprav týkajících se restartu a Pavel Machek řekl, že správný postup by byl to nejprve poslat Andrew Mortonovi, aby je začlenil do -mm stromu. Linus napsal:

Jedné věci se skutečně bojím. Že bude Andrew v jednu chvíli tam, kde já byl před pár lety – přepracovaný a vystresovaný díky hromadám patchů. Sice napsal nebo upravil spousty různých nástrojů pro sledování patchů a začleňování pomocí gitu snad práci usnadňuje, ale i tak mi to dělá starosti. Kdyby Andrew vyhořel, zasáhlo by nás to velmi citelně. Přemýšlím, jak tyhle problémy zmírnit. Vyhovuje mi -mm v roli pokusného králíka, takže cesta přes Andrewa je v tomhle směru výborná, ale má to své stinné stránky. Andrew?

Andrew poznamenal: Patche jako takové moc času nezabírají. Hodně času ale vezmou patche, které nefungují – jeden špatný patch může zabrat hodiny a způsobit, že si o jeho původci začnu myslet ošklivé věci.

A doplnil: Zvládám to dobře. Množství patchů nepředstavuje problém z hlediska prosté manipulace. V současné době je největší potíž s nedostatečnou kontrolou patchů. V této oblasti budu muset přitvrdit. Relativně málo vývojářů je ochotných provádět podrobné kontroly velkých patchů, takže by bylo fajn na to lidi trochu nasměrovat. Velmi oceňuji Christophovu práci, díky. Říkáme to pořád, ale množství patchů skutečně jednoho dne opadne. Vlastně už u 2.6.15 se toho moc neděje.

Chyby jsou velký problém. Vydání -mm trvá nejméně 4 hodiny a jediná chyba může způsobit odsunutí na další den a pak musím začít znovu. Několikrát mi zabralo pár dní jen to, abych dal dohromady strom, který na čtyřech architekturách nabootuje a na sedmi půjde zkompileovat. S chybami teď trávím více a více času. Máme stovky chyb, které lidi nahlásili, o kterých příslušní vývojáři vědí, a se

kterými NIC NEDĚLAJÍ. „Nemohu to reprodukovat“ není adekvátní odpověď, když jsou tu ochotní testeři, kteří mohou během odhalování problému pomoci. Jsou stovky počítačů, o kterých víme, že nefungovaly, a vývojáři si k opravování takových věcí prostě musí vyhradit trochu času.

Strom -mm filtruje velké množství chyb, aby se nedostaly do hlavního jádra – odhadoval bych, že proklouzne přibližně 10 procent. I když těch 90 procent jsou ty lehké věci - často jen kompilační chyby. Rád bych -mm vydával častěji a také bych byl rád, kdyby nemělo reputaci divokého a špatného kódu. Obojí by šlo, kdyby si autoři patchů dali více záležet na testování.

V jednu chvíli napsal Eric: Pro lidi, kteří jako já pracují na částech jádra bez správců, je to zvlášť problematické, protože často není k mání žádný prostředník, kterému bych patche mohl posílat.

A Andrew napsal: Jo. A soubor MAINTAINERS není zrovna kompletní. Já většinou vím, do koho šťouchnout, a když není po ruce zřejmý adresát, mám 26000 patchů, které mohu grepnout, abych zjistil, kdo by o daném kódu mohl něco vědět. S nízkourovňovým x86 je docela potíž, protože má hodně kuchtíků, ale žádného jasného šéfkuchaře.

Jednotliví správci odpovídají každý jinak rychle, liší se pečlivostí i procentem opuštěných patchů. Zmatky nastávají také ve chvíli, kdy pošlu kopii zprávy o patchi správci. Mám pak patch Linusovi poslat já nebo chci, aby to udělal správce? Má-li správce strom v -mm, tak patch automaticky nechám být, jakmile ho správce začlení, takže v takovém případě je vše jasné.

Když správce řekne „díky, začleněno“, ale nemá strom v -mm, pak většinou patch držím donekonečna, až se konečně objeví u Linuse. Nebo na něj nakonec zapomenu a začlením ho stejně ;). Když správce patch pouze schválí, pošlu ho Linusovi. Když správce patch neschválí, tak ho buď vypustím nebo označím, že nemá být začleněn a podržím ho, abych si pamatoval, že máme chybu, kterou je potřeba opravit. Má-li správce strom v -mm a patch nezačlení, ponechám si ho, abych jej mohl občas správci poslat, dokud nepřijde nějaká konečná odpověď.

Alexander Nyberg připojil: A navíc je bugme.osdl.org hodně nevyužitý (světlou výjimkou je ACPI) a Andrew se tam hodně angažoval. Na spoustu hlášení o chybách v té konferenci odpovídá pouze Andrew. Myslím, že by se mu mělo dostat daleko více pomoci, než v současné době dostává.

Andrew odpověděl:

Ano, jaderný bugmeister je funkce úplně oddělená od honění patchů. Je to něco, co by mohl a měl dělat někdo další. Uvažuji o tom, že vyvinu postupy, zjistím, co funguje, a pak se poohlédnu po jiném chudákovi, kterému bych to přenechal. Nebudu tvrdit, že se mi to v současné době daří. Možná proto, že tomu nevěnuji dost času, abych mohl ukázat, co funguje a co ne.

Neřekl bych, že bugmeister je na plný úvazek, ale na velkou část ano. Vyžaduje to někoho, kdo se nebude ostýchat, a kdo dobře rozumí jádru z hlediska kódu, procesů (heh) a lidí. Musí mít schopnost udržovat si celkový přehled o tom, kde jsme, které chyby jsou vážné, a které ne. Schopnost výstižně tento přehled sdělit všem ostatním. Schopnost říct Linusovi ”nemůžeš jádro vydat, dokud nebudou opraveny chyby A, B a C”. Dovednosti a intuici potřebnou k tomu, aby rozlišil, který patch je ojedinělá záležitost a lze ignorovat, pokud se neobjeví znovu atp. Je to jedna z těch prací, která zabere všechny čas, který jí můžete věnovat. Vývoj jádra je dnes profesionálnější než rádi předstíráme a vývojáři ocení, když to pro nás bude někdo dělat. Je to ale dost nuda.

Linux 2.6.14-rc2-mm1, 42 e-mailů

19. zář – 28. zář

Andrew Morton oznámil Linux 2.6.14-rc2-mm1:

<ftp://ftp.kernel.org/pub/linux/kernel/people/akpm/patches/2.6/2.6.14-rc2/2.6.14-rc2-mm1/> [12]

- Přidal jsem git strom ‘git-sas.patch’: Ovladač a podpora SAS od Lubena Tuikova.
- Různé další věci – nic zásadního.

Správa subsystému I2C, 36 e-mailů

22. zář

Greg KH poslal několik patchů pro USB a PCI v 2.6. Mezi nimi byl i jeden, který **odstraňuje mé jméno z funkce správce I2C. Jean to teď vše zvládá lépe než já.**

Patch jako správce I2C stanovil Jeana Delvarea.

Cogito 0.15.1 vydáno s důležitou opravou, 1 e-mail

23. zář

Petr Baudis oznámil Cogito 0.15.1: **Oznamuji vydání verze 0.15.1 Cogita, uživatelsky přívětivého rozhraní pro Linusův GIT. Cogito najdete v: <http://www.kernel.org/pub/software/scm/cogito/> [13] až bude zrcadlení kernel.org zase chvilku fungovat ;-).**

Posílám kopii do této konference, protože vydání obsahuje opravu chyby, která mohla způsobovat nepěknou ztrátu dat. A pravděpodobně se to týká všech uživatelů Cogita (byla tam od verze cogito-0.11.2). Pokud jste měli nějaké lokální necommitované změny a začlenili jste nové věci (buď pomocí cg-update nebo cg-merge), v některých případech cogito potichu zlikvidovalo vaše lokální změny. Způsobovalo to nesmyslné volání git-checkout-cache, na které upozornil Linus.

Odkazy

- [1] <http://klive.cpushare.com/>
- [2] <http://klive.cpushare.com/downloads/klive-0.0.tar.bz2>
- [3] <ftp://ngc891.blogdns.net/pub/linux/misc/ChangeLog-2.6.12-2.6.13.txt>
- [4] <http://www.freedesktop.org/~jonsmirl/graphics.html>
- [5] <http://linuxdevices.com/articles/AT3855888078.html>
- [6] <http://www.kernel.org/git/?p=linux/kernel/git/chrisw/stable-queue.git;a=shortlog>
- [7] <http://www.energyparty.de/ahmad/readme>
- [8] http://www.energyparty.de/ahmad/autoconfig_0_1_patch.tgz
- [9] http://www.energyparty.de/ahmad/autoconfig_0_1.tgz
- [10] <http://osuosl.org/photos/kernel/view>
- [11] <http://www.talklikeapirate.com/piratehome.html>
- [12] <ftp://ftp.kernel.org/pub/linux/kernel/people/akpm/patches/2.6/2.6.14-rc2/2.6.14-rc2-mm1/>
- [13] <http://www.kernel.org/pub/software/scm/cogito/>

Zprávičky

1.10.2005

Daniel Kvasnička ml.

Na serveru Linux-Watch vyšel článek vyjadřující názor, že na světě je „zatraceně mnoho linuxových distribucí“ a že světu Linuxu by prospělo, kdyby se více konsolidoval.

1.10.2005

Tomáš Kopečný

Máte rádi mmorpgy s nádhernou 3D grafikou? A ještě k tomu pod GPL2? Potom vyzkoušejte 3D fantasy hru PlaneShift!

1.10.2005

Alois Nešpor

Vznikl nový projekt kccmp, neboli Kernel Configuration Comparison, díky kterému můžete v grafickém rozhraní porovnat dva .config soubory s uloženým nastavením linuxového jádra. Vhodné zejména pro notorické kompilátory linuxového jádra a jiné hračky.

2.10.2005

Robert Krátký

Přehrávač MOC (Music On Console) má ve vývojové verzi 2.4.0-alpha1 konfigurovatelné rozložení prvků v okně. Recenze v článku Textové hudební přehrávače – II.

2.10.2005

Jozef

Bližšie informácie o pripravovanom Stretnutie priateľov GNU/Linuxu v Žiline su na tomto webe. Pre tých, čo už posielali registračný mail, a neobdržali odpoveď, prosím nech sa manuálne pridajú do zoznamu, ktorý je na uvedenom webe.

3.10.2005

Robert Krátký

Krusader, správce souborů pro KDE, nabízí v první betaverzi řady 1.70 mnohá vylepšení. V seznamu změn a nových funkcí najdete i instrukce k instalaci.

3.10.2005

Robert Krátký

Opera se chystá předběhnout Firefox a urvat si druhé místo mezi webovými prohlížeči pro sebe. O plánech, strategii a procentech píše The Register: Opera tunes up to overtake Firefox.

3.10.2005

Robert Krátký

RIAA (Recording Industry Association of America) je v žalobě obviňována mimo jiné z podvodu a porušení soukromí. Žalobu podala žena, kterou RIAA dříve žalovala za sdílení souborů. Více The Inquirer: RIAA sued by victim a Slashdot: Another Victim Countersues RIAA Under RICO Act.

3.10.2005

Robert Krátký

Porušujete zákon přehráváním DVD v GNU/Linuxu? Možná to nebude tak strašné. Přečtěte si na LXer, co si o tom myslí soudce Buford I. Motion: I want my DVD, your honor.

3.10.2005

Robert Krátký

V novém vydání Linux Gazette (#119) mluví Jimmy O'Regan o výsledcích programátorského léta financovaného firmou Google: After the Summer of Code.

3.10.2005

Robert Krátký

Matt Asay ve svém blogu komentuje studii (PDF) o tom, jak vznikají a fungují komunity vývojářů kolem úspěšných open source projektů: Analyst: (More on) Defining successful open source communities.

3.10.2005

tomasini

Na oficiálních stránkách hry wesnoth.org najdete novou připravovanou verzi 1.0rc1. Podle statistiky překladů patří ten náš český k již hotovým. Přeji příjemné chvíle při hraní a testování. A tvůrcům hry k brzkému vydání finální verze. Výborná práce, jen tak dále!

3.10.2005

Robert Krátký

V rozhovoru na Business Week mluví Linus Torvalds o Linuxu na desktopu, patentech, obchodní známce „Linux“ a mobilních aplikacích: Torvalds' Baby Comes of Age.

3.10.2005

Lukáš Polívka

Láká vás profesionální herní kariéra? Znamená pro vás svobodný software hodně? Třeba se prosadíte ve hře Warow, FPS postavené na silně modifikovaném Quake ? engine s důrazem na trick-jumpy, která si získala značnou pozornost např. na německé progamingové scéně. Dostupné jsou binárky pro GNU/Linux i MS Windows.

4.10.2005

Robert Krátký

dot.kde.org informuje o vydání živého CD (založeno na PCLinuxOS) s přehrávačem amaroK. CD obsahuje i písničky vydané pod svobodnou licenci Creative Commons od společnosti Magnatune.

4.10.2005

Robert Krátký

Byl vydán AbiWord 2.4. Obsahuje množství nových funkcí (kontrola pravopisu, obtékání obrázků, rovnice, podpora OpenDocument) a samozřejmě opravy chyb. Poznámky k vydání se screenshoty, informuje Footnotes (gnomedesktop.org).

4.10.2005*Robert Krátký*

Společnost Round Two/Flock (autor rozšíření TabBrowser) vydala čtyři aktualizace svých rozšíření pro Firefox: FlashGot, Bandwidth Tester, SwitchProxy a ExtensionsMirror.nl. Viz článek [Rozšiřte si Firefox!](#)

4.10.2005*Robert Krátký*

Linus Torvalds v LKML prohlásil, že hardwarové specifikace jsou k ničemu, a kód napsaný na jejich základě bývá často mizerný. Sys-Con komentuje (a cituje) debatu, kterou tyto výroky rozpoutaly na mnoha místech webu.

4.10.2005*Robert Krátký*

SUSE 10.0 ještě nevychladla a OSDir.com už přináší sérii screenshotů z verze 10.1 Alpha 1. SUSE Linux 10.1 Alpha 1 screenshots.

4.10.2005*Robert Krátký*

Další v sérii článků o CLI na Linux.com představuje běžným uživatelům utilitu logrotate, která pomáhá s udržováním pořádku ve /var/log.

4.10.2005*Robert Krátký*

Mark Shuttleworth, zakladatel distribuce Ubuntu, sestavil svůj vlastní seznam často kladených dotazů o Ubuntu. Vysvětluje v něm mimo jiné, proč raději nedává peníze Debianu, a kde se vzaly názvy jako Warty Warthog a Breezy Badger.

4.10.2005*Petr Vaněk*

Před nedávnem slibovaná nová vývojová verze DTP programu Scribus je na světě.

4.10.2005*Stanislav Valasek*

Ucelený přehled instalace pluginů do Mozilla Firefoxu v Ubuntu je v Ubuntu fóře.

5.10.2005*Robert Krátký*

Paul Murphy tvrdí ve svém blogu na ZDNet.com, že GPL rozvoji Linuxu spíše překáží než pomáhá: Linux: GPL impedes more than it helps.

5.10.2005*Robert Krátký*

Linspire chystá firemní verzi své distribuce: Professional. Kevin Carmony, výkonný ředitel Linspire, se nechává slyšet, že díky instalaci softwaru a správě systému pomocí Click'n'Run bude Linspire Pro vážným konkurentem firemním desktopům od Red Hatu a Novellu.

5.10.2005*Robert Krátký*

Kristian Rietveld odkazuje na několik zajímavých ukávek nových funkcí, které jsou chystány pro GTK+ 2.10. Mezi jinými i tato kouzla s widgety.

5.10.2005*Pavel Szalbot ml.*

Článek na ONLamp.com vám představí nástroj cvs2svn, který umožňuje snadnou migraci projektů spravovaných CVS do Subversion.

5.10.2005*Pavel Szalbot ml.*

Jak hodnotíte svůj přehled v historii vývoje unixových systémů? Slyšeli jste už názvy Minix, XENIX, HPBSD, či Ultrix (a spoustu dalších) a pořádně nevíte, co znamenají? Zkuste se podívat na časovou osu a návaznosti v unixové historii.

5.10.2005*Pavel Szalbot ml.*

LinuxPlanet se na konferenci Women In Games International zeptal několika řečníků na to, kolik open source kódu (zpravidla knihoven) se používá při vývoji her a kolik nástrojů při něm pomáhá a tento počín nechal vzniknout článek o open source ve hrách.

5.10.2005*Robert Krátký*

Anyterm je emulátor terminálu, se kterým můžete pracovat ve webovém prohlížeči (JavaScript + modul pro Apache). Verzi 1.1.6 je možné zkompilovat pro embedded zařízení. Na my.anyterm.org je možné Anyterm používat dokonce bez instalaci na svůj systém.

5.10.2005*Jozef Henzl*

Právě byl oficiálně spuštěn www.linuxvbrne.org, na kterém budete moci najít informace o konaných akcích okolo GNU/Linuxu v Brně. Mimo jiné i článek o 0. neoficiální akci konané 23.9. Předem však prosíme o trpělivost s výpadky způsobené technickými problémy hostingu.

5.10.2005*Robert Krátký*

Jeremy Garcia z LinuxQuestions.org popisuje, jak pomocí CVS a SVN stahovat vývojářské verze zdrojových kódů některých známých open source projektů: Downloading Source Trees 101.

5.10.2005*Pavel Szalbot ml.*

Apache Software Foundation oznámila vydání Beehive 1.0 (informuje ZDNet). Beehive tvoří třívrstvý (NetUI, Controls a Web Service Metadata) objektový model na J2EE a Struts, který usnadňuje práci s těmito technologiemi.

6.10.2005*Leoš Literák*

Dlouhý rozhovor s Ondřejem Procházkou, vedoucím vývoje portálu Seznam, přináší LinuxBIZ. Hlavním tématem je Open Source z pohledu manažera.

6.10.2005*Pavel Szalbot ml.*

Lidem vyvíjejícím GNOME zřejmě příliš nezáleželo na rychlosti startu jejich správce oken a tak se do jeho analýzy jeden z nich pustil. Připravil dokument o tom, jak zkrátit a analyzovat dobu spouštění GNOME, ovšem některé z objevených metod si na cestu do zdrojových kódů budou muset počkat.

6.10.2005*Pavel Szalbot ml.*

Prodej očekávaného linuxového tabletu Nokia 770 byl odložen na konec letošního roku. Na LinuxDevices.com se spekuluje o tom, že tvrzení zdrojů z Nokie o potřebě doladit softwarové prostředí ukazuje na možný krach jednání Nokie s Operou. Projekt Minimo, zainteresovaný Mozilla browser pro mobilní zařízení a navíc údajně podporovaný Nokii, nedávno oznámil agresivní plán vývoje vrcholící 22. prosince...

6.10.2005*Pavel Szalbot ml.*

Tento měsíc by měla vyjít nová verze GIMPu 2.4. Ochutnat některé z řady novinek, např. zajímavý „foreground extractor“, můžete v článku o prvním pohledu na GIMP 2.4.

6.10.2005*Martin Šebek*

Vypadá to, že oblíbený komiks Kernel Panic znovu ožívá. Již druhou středu v řadě vyšlo nové pokračování. Dočkáme se přechodu na Nifty Doorways?

6.10.2005*fipa*

SUSE LINUX 10.0 Eval je ke stažení na FTP (seznam mirrorů). Upozornil portal.suse.cz.

6.10.2005*Peter Kotrčka*

Vyšel RC Ubuntu GNU/Linuxu, čo je snáď posledné vydanie pred budúcotýždňovou finálnou verzioou. Sťahujte 5.10-rc-install-i386.iso.

6.10.2005*Pavel Szalbot ml.*

PostgreSQL se blíží k 8.1, ale je třeba vydávat i opravy pro stávající hlavní verzi a tak je tu PostgreSQL 8.0.4. Mezi novinkami je spousta oprav z nichž zajímavější mohou být např. opravená práce s pamětí, oprava příkazu VACUUM, úprava kódu tsearch2 a v neposlední řadě i vylepšení ne-linuxových odrůd.

6.10.2005*Michal Vyskočil*

Renaud Deraison, autor programu Nessus ohlásil, že verze 3 nebude uvolněna pod GPL.

7.10.2005*Pavel Szalbot ml.*

Jestli patříte k nadšencům, kteří už včera nainstalovali OpenSUSE, pak jste si možná všimli (pokud je tedy potřebujete), že systému chybí aplikace jako DVD přehrávač, MP3 přehrávač, Flash player, Real player atp. Mohl by se vám hodit tento návod popisující doinstalování OpenSUSE na plnohodnotný desktop.

7.10.2005*Pavel Szalbot ml.*

Milovníkům tahových strategií se naskytla příležitost vyzkoušet finální verzi hry The Battle for Wesnoth. Více o hře se dozvíte v poznámkách k vydání.

7.10.2005*Pavel Szalbot ml.*

Na světě je další verze ndiswrapperu, na něž jsou odkázáni majitelé některých bezdrátových karet. Novinky jsou zatím zahaleny tajemstvím (pravděpodobně mj. podpora dalších karty/ovladačů).

8.10.2005*Alois Nešpor*

Jakým směrem se ubírá MPlayer, je vidět v Changelogu. Vývojáři zapracovali na grafickém rozhraní, které se dočkalo portace na GTK+ 2.0, je opraveno dost chyb a přibyla podpora dalších kodeků (jak pro mplayer, tak mencoder), dokončena byla česká dokumentace, také přibyla podpora hintingu pro TrueType fonty. Vše a ještě více bude již ve verzi 1.0pre8.

10.10.2005*Pavel Szalbot ml.*

Zdá se, že se rychlost GNOME přestává líbit i vývojářům. Na GNOME summitu totiž jeden z nich vystoupil s prezentací nazvanou „Making GNOME Fast“, která rázně a jednoduše vysvětluje, kde jsou chyby.

10.10.2005*Pavel Szalbot ml.*

Virtualizační nástroj Xen zraje. S procesory podporujícími virtualizaci (technologie Intel VT či AMD SVM) odpadá stará bolest s nutností modifikace operačního systému, která však Xen zvyšuje oproti ostatním. Článek shrnující stav Xenu k verzi 3.0 najdete v Linux Magazine.

10.10.2005*Pavel Szalbot ml.*

Zajímavý projekt nazvaný „Unix pro začínajícího mága“ vás knižní (PDF) formou seznámí se

základy unixových systému podobně, jak jste se možná kdysi spřátelili s DOSem.

10.10.2005

Pavel Szalbot ml.

Ředitel MySQL AB po početném množství rozčarovaných mailů obhajuje partnerství s nevalně proslulou SCO. Podle jeho slov se zdá, že MySQL AB nikoho neodsuzuje a pokusí se SCO vyléčit, jelikož k tomu forma jejich partnerství vytváří možnosti.

11.10.2005

Pavel Szalbot ml.

Jen u slov o optimalizaci GNOME nezůstane. Ale spoň to potvrzují plány GNOME zahrát si i embedded ligu a s tím spojená potřeba přidat podporu pro nové ovládací prvky (stylus), optimalizovat spotřebu paměti a výkon, či dva noví členové kolegia vyvíjející pro embedded systémy, jak informuje ZDNet.

11.10.2005

Pavel Szalbot ml.

Webové systémy správy obsahu donedávna degradovala absence nákupního košíku a tím možnost vytvoření vlastního eshopu. Minimálně dva oblíbené CMS si však našly cestu i internetovému obchodu a jak si jejich řešení vedou, si můžete přečíst na NewsForge.

11.10.2005

Pavel Szalbot ml.

Zapomeňte na Sun a Google a přivítejte Open Document Fellowship. Jedná se o nově vzniklou organizaci, která se zaměřuje na objasňování úlohy otevřených formátů v architekturách zaměřených na služby (v podstatě při výměně dat mezi systémy). Více píše MadPenguin.

11.10.2005

Jiri Pech

V rámci konference LinuXchange proběhne též sekce nazvaná Linux ve školství a státní správě. Více zde.

11.10.2005

Pavel Szalbot ml.

Na DesktopLinux můžete zjistit, jestli pro vás bude další Mozilla Thunderbird 1.5 vhodným mailovým klientem. Autor přes spoustu nevýhod a nepřipravenost na „enterprise“ nasazení vyzdvihl např. rychlost jakou jsou opravovány chyby a programovány nové funkce.

11.10.2005

Leoš Literák

Open Source Alliance zve na seminář zabývající se linuxovými řešeními na procesorech IBM Power. Akce se koná 17. 10. a budou na ni přednášet

přední světoví odborníci na danou problematiku, vstup je zdarma.

12.10.2005

Pavel Szalbot ml.

Zend, firma vyvíjející jádro populárního PHP, vydala verzi PHP pro software Oracle, kterou nabízí zdarma (mimo volitelnou technickou podporu). Informuje ZDNet.

12.10.2005

Pavel Szalbot ml.

Na světě je Boo 0.7, nový staticky typovaný programovací jazyk syntakticky inspirovaný Pythonem s důrazem na rozšiřitelnost jazyka a kompilátoru. Kolik programovacích jazyků ovládáš, tolikrát jsi počítačem;-)!

12.10.2005

Pavel Szalbot ml.

Linux.com recenzuje poslední verzi „rozmarného jezevce“ Ubuntu 5.10. Líbila se detekce hardwaru včetně WiFi karty, Smeg – editor menu pro GNOME či KDE v distribuci Kubuntu, která je také zmíněna. Stále bohužel chybí instalace typu „vývojář“...

12.10.2005

Michal Čihař

Jak se dalo po změně licence u Nessusu čekat, objevil se jeho fork, který bude nadále pod GPL. Jmenuje se Porz-Wahn a hledá další členy týmu.

12.10.2005

Jose

Xara uverejňuje svoj produkt Xara X pod GPL. Zatiaľ toho moc uvoľnená verzia pre Linux/Mac nevie, funguje otváranie súborov .xar a prehliadanie vrstiev.

12.10.2005

Leoš Literák

Na LinuxBIZ najdete studii nasazení Linuxu ve firmě Baťa. Desítky prodejen jedou kompletně pod GNU/Linuxem.

12.10.2005

Robert Krátký

V rámci veletrhu INVEX se bude konat i 7. ročník odborné konference Linux Hall – IT bez hranic a tematická konference o problematice bezpečnosti – Security Invex 2005. Více viz LinuxBIZ Linux Hall a Security Invex 2005.

13.10.2005

Pavel Szalbot ml.

Kdo si dobře vzpomíná, dnes jsme se měli dočkat vydání OpenOffice.org 2.0. Jelikož bylo uvolnění finální verze několikrát odloženo, nikoho asi nepřekvapí, že se tak stalo i tentokrát. Ovšem s (dalším) příslibem, že do 2 týdnů OO.org 2.0 dostaneme.

13.10.2005*Robert Krátký*

Ve verzi 1.0.0 má program pro zálohování DVD – k9copy – experimentální podporu pro zachování původních menu a umí vytvářet ISO obrazy.

13.10.2005*Rudolf Adamkovič*

Dnes vyšla nová verzia distribúcie Ubuntu Linux 5.10 s kódovým označením Breezy Badger. Inštaláčn  CD/DVD pre platformu x86, AMD64  i PowerPC n jdete na ubuntu.com.

13.10.2005*Pavel Szalbot ml.*

ArsTechnica vydala  l nek popisuj c  t i editory nejr zn j  ch zvukov ch z znam  – Ardour, Audacity a ovl d n m Emacs napodobuj c  SND. V posledn   asti se dozv te, co z sk te pou z v n m FOSS editor .

13.10.2005*Michal  ihař*

Co v echno bude um t nov y Gaim 2.0.0 prozradil jeden z jeho vyvoj ř . Krom  za len n  mnoha oprav a v sledk  Summer of Code bude za len n   projekt gaim-vv p r d vaj c  podporu pro zvuk a video a mo n  se do k ame i podpory pro Google Talk.

13.10.2005*Jiř  Hlinka*

ATI vydala po dvou m s c ch novou verzi propriet rn ch ovlada   grafick ch karet (pro Radeony 8500 a  X850). Novou verzi 8.18.6 je mo n  st hnout ze str nek ATI. Seznam zm n najdete na rage3d.com.

14.10.2005*Jiř  Stavinoha*

Byla uvoln na nov , kompletn  p epsan  řada publika n ho syst mu UNITED-NUKE 4.2ms2. Z dlouh ho v y tu zm n stoj  zm nit nap . n kolikan sobn  v    rychlost zpracov n , podporu řablon  i podporu pro Magic SEO URL plugin.

14.10.2005*Pavel Szalbot ml.*

NewsForge popisuje plugin Off-The-Record pro Gaim, kter y jej obohacuje o zaj mav  a pro IM nov  bezpečnostn  koncepty. Umo ňuje podepisovat ob ma   astn ky rozhovoru z roveň – t t t strana nedok že ur it, kdo je autorem a d le – řifrovac  kl  e generuje plugin p ed ka d m rozhovorem a neukl d  je na disk.

14.10.2005*Pavel Szalbot ml.*

OpenSolaris si narozd l od sv ho p edka hled  m sto i v mobiln m trhu a vznik  tak komunita OpenSolaris Laptop Community, je  si klade za c l zp r stupnit OpenSolaris majitel m laptop 

a vytvořit v n m podporu pro nejr zn j   hardware v nich pou z van y. V voj ř i Sunu navíc pracuj  na implementaci r zn ch technik pro spr vu spotřeby. Informuj  OSNews.

14.10.2005*Pavel Szalbot ml.*

Pon kolik t  vych z   l nek o vznikaj c m fenom nu Ruby on Rails, tentokr t v ak vych zej c  z řady t ch, co osl ňuj  jednoduchost  v p r kladech, jeliko  d v   ten ř m n hlednout do „tajemstv “ frameworku. Je tak snad hodnotn j   pro zb hl  v voj ře.

14.10.2005*Pavel Szalbot ml.*

Ubuntu Breezy Badger byl portov n na architektury IA64, SPARC (pouze UltraSPARC) a HPPA (od 1.1) a otevir  se tak komunit m, je  n kter  distribuce p ehli ej .

14.10.2005*Robert Kr tk y*

Je t  p ed vyd n m KDE 3.5 v  la pravd podobn  j  posledn  opravn  verze řady 3.4 (KDE 3.4.3). Ozn men , seznam zm n.

14.10.2005*Michal Medveck y*

Ji  z tra a poz t i prob hne p t y ro n k konference OpenWeekend. Kdo nem žete do Prahy, m žete sledovat on-line stream. V ce informac  na www.openweekend.cz.

16.10.2005*Robert Kr tk y*

Pou z v te KDE, ale nesed  v m emul tor termin lu Konsole? Vyzkoušejte YaKuaake. Zaj mav  je, že vyu z v  plochu obrazovky jakoby po vzoru programu screen. P v  v  la verze 2.7.2.

16.10.2005*Robert Kr tk y*

Max Bruning srovn v  na opensolaris.org t i hlavn  subsystemy kernel  Solarisu, Linuxu a FreeBSD: schedulery, spr vu pam ti a architekturu souborov ch syst m .

16.10.2005*Robert Kr tk y*

N kolik rad a tip  pro konfiguraci hardwaru a softwaru (urpmi zdroje, p ehr v n  DVD atp.) v distribuci Mandriva 2006 najdete na Linux tips for free.

16.10.2005*Robert Kr tk y*

OpenOffice.org 2.0 Release Candidate 3 je dal m krokem k fin ln  verzi 2.0.  esk  RPM a DEB bal čky jsou j  na ftp.linux.cz.

16.10.2005*Pavel Szalbot ml.*

LinuxDevices.com upozorňují na 10 článků a tutoriálů, které IBM zveřejnilo ve svých developerWorks. Zabývají se zajímavými technickými tématy se vztahem k opensource a Linuxu.

16.10.2005*Lukáš Polívka*

Nedávno spuštěný server Ebuild Exchange má sloužit uživatelům distribuce Gentoo k pohodlnému sdílení vlastních ebuildů (včetně potřebných patchů a eclasses), které si ještě neproklesly cestu do oficiálního Portage tree.

17.10.2005*Pavel Szalbot ml.*

Na OSNews se objevil tutoriál pro připojení mobilního telefonu s GPRS přes infraport k Linuxu. Autor v příštích týdnech slibuje obdobný článek pro USB či Bluetooth připojení.

17.10.2005*Pavel Szalbot ml.*

KDE.News si na aKademy popovídaly se dvěma vývojáři z Nokie o vyvíjeném internetovém prohlížeči, který je postaven na KHTML. Řeč byla o úskalích a výhodách KHTML a KJS, či o spolupráci na vývoji mobilního prohlížeče s KDE komunitou.

17.10.2005*Pavel Szalbot ml.*

Ubuntu se podařilo zazářit na scéně linuxových distribucí tak rychle, jako ještě nikomu před ním a jeho hvězda zatím nepohasíná. Jaká tajemství stojí za jeho úspěchem?

17.10.2005*Robert Krátký*

Po více než roce vyšla nová stabilní verze textového editoru Vim (6.4). Neobsahuje žádné nové funkce - ty budou až v připravované verzi 7.

17.10.2005*Robert Krátký*

Whitedust přináší rozhovor s autorem skenovacího nástroje Nmap – Fyodorem. Mluví se v něm o hackerech, open source, knihách a tak dále.

17.10.2005*Pavel Szalbot ml.*

Kouzlení v příkazové řádce (seriál CLI Magic na Linux.com) pokračuje nástrojem pro monitorování síťových připojení, který vás upozorní na vámi nepovolené, s názvem Trojan Scan.

17.10.2005*Michal Křenek*

Nedávno vyšly nové ntfsprogs 1.12.1, které již (od verze 1.12.0) podporují prostřednictvím příkazu 'ntfsmount' (jenž využívá FUSE) téměř kompletní zápis na NTFS (kromě prepisování existujících souborů, které již fungovalo dříve, přibyla podpora vytváření a mazání nových souborů a adresářů).

tujících souborů, které již fungovalo dříve, přibyla podpora vytváření a mazání nových souborů a adresářů).

18.10.2005*Pavel Szalbot ml.*

Jednou ze překážek pro rozšíření Ruby bývala absence kvalitního IDE (pomineme-li Emacs, Vim, či aplikace pro Mac) a tak vzniklo rozšíření pro Eclipse nazvané RDT (Ruby Development Tools). Seznámí vás s ním IBM developerWorks.

18.10.2005*Pavel Szalbot ml.*

Blížící se vypuštění prvního komerčního produktu kalifornské XenSource, která dohlíží na projekt Xen, zřejmě zatřese s trhem virtualizačních nástrojů a zvláště jeho králem VMware. Tomu je vyčítána neúměrná cenová politika, která brzdí rozšíření virtualizace v serverech. Více v článku na crn.com.

18.10.2005*Pavel Szalbot ml.*

Open Clip Art Library se dočkala verze 0.18 a obsahuje nyní přes 6900 public domain vektorových obrázků (SVG) od více než 500 autorů.

18.10.2005*Mgr. Ladislav 'R@ndom' Gagyi*

V piatok 14.9.2005 bola releasnutá nová verzia distribúcie Slackware. Takže po niekoľkých mesiacoch čakania je tu 10.2. V novej verzii nájdete nové KDE, XFce, glibc a konečne je integrovaný do systému Firefox a Thunderbird. Changelog.

18.10.2005*Aldagautr*

Po několika odkladech [zprávička] a RC [zprávička] tu máme konečně ostrou verzi OpenOffice.org 2.0. Ke stažení na ftp.linux.cz. UPDATE: oficiálního uvedení si všimly i novinky.cz.

18.10.2005*Jozef Henzl*

V Brně se v pátek 21.10. v 18 hodin bude konat 1. oficiální setkání příznivců Linuxu. Místem konání bude opět restaurace Rubín v Žabovřeskách. Zveme všechny zájemce.

19.10.2005*Pavel Szalbot ml.*

Free Standards Group oznámila projekt Linux Standard Base Desktop Project. Projekt by měl urychlit rozšíření Linuxu na desktopech poskytnutím standardního runtime prostředí pro vývojáře aplikací. Více na NewsForge.

19.10.2005*Pavel Szalbot ml.*

Když shaníte aplikace pro GNOME, kam se díváte? Klasicky můžete zvolit Freshmeat, Sourceforge a jim podobné, ale už rok a půl můžete hle-

dat i na GnomeFiles.org, které právě slaví tisíce přidáním aplikaci. Informují OSNews.

19.10.2005*Pavel Szalbot ml.*

Na ONLamp.com nedávno vyšel článek o webserveru (instalace a nastavení) tthttpd, který oproti populárnímu Apache klade důraz spíše na minimalizaci náročnosti, rychlost a jednoduchost. Kdo by čekal, že ztratí spoustu funkcí, možná bude mile překvapen možností vytvořit virtuální hostování, podporou HTTP/1.1, CGI či IPv6.

19.10.2005*Robert Krátký*

S mírným zpožděním vychází druhá betaverze desktopového prostředí KDE 3.5. Oznámení, info stránka, Construct.

19.10.2005*Robert Krátký*

Jimmy Wales, spoluzakladatel Wikipedie, přiznává, že nekvalitní záznamy představují velký problém projektu. Larry Sanger, který už na projektu nepracuje, tvrdí, že hlavní potíž je v až přílišném antielitismu.

19.10.2005*Robert Krátký*

Kvůli sporům o obchodní značku se Google ve Velké Británii vzdal názvu Gmail (i když současným uživatelům adresy zůstanou) a přejmenoval službu na Googlemail. The Register škodolibě komentuje, že Google přišel o svůj G-bod...

19.10.2005*Matej 'Yin' Gagyi*

Dnes (19. október 2005) približne o 18:30 prekonala prehliadač Mozilla Firefox neuveriteľných, okruhlych 100 000 000 stiahnutí od vydania verzie 1.0.

20.10.2005*Radek Vokál*

Právě vyšla nová verze síťového snifferu ethereal. Jako s každou novou verzí, i ve verzi 0.10.13 je opraveno několik závažných bezpečnostních chyb.

20.10.2005*Pavel Szalbot ml.*

Čtete na webech rádi hlášku o optimalizaci pro vybraný prohlížeč a rozlišení? A přijde vám málo dávat známými ikonkami najevo, že na svých webech dodržujete standardy W3C? Zkuste svůj web obohatit ikonkou „Too Cool for IE“.

20.10.2005*Pavel Szalbot ml.*

V prosinci bychom se dle pozměněného plánu vývoje X.Org měli dočkat nové verze X Window Systemu. X11R6.9 a X11R7.0 budou mít identický zdrojový kód, budou se však lišit organizační kódou a způsobem instalace. Včera vyšla první RC

verze, která mimo jiné dokumentuje to, co nás v nových „Xkách“ čeká.

20.10.2005*Pavel Szalbot ml.*

Nainstalovali jste proxy server Squid a pochopili základy a teď byste rádi nakonfigurovali některé z pokročilejších funkcí jako ACL, prokazování identity skrz (authentication) proxy či blokování stránek? Zkuste třeba článek na Linux.com, který by mohl poskytnout potřebné znalosti.

20.10.2005*Pavel Szalbot ml.*

Sun uvolnil zdrojové kódy další své technologie. Tentokrát se jedná o Jini, která umožňuje vývojářům psát síťové služby implementované softwarově či hardwarově v Javě. Jako malé překvapení může zapůsobit zvolená Apache licence. Více na ZDNetu.

20.10.2005*Robert Krátký*

Online volební systém v Estonsku běží na Linuxu. Poprvé by měl být zásadněji prověřen při parlamentních volbách v roce 2007: Estonia first to allow online voting nationwide.

20.10.2005*Daniel Kvasnička ml.*

Simon Phipps, šéf open-source u Sunu, poukazuje na dohled jako na přehlíženou oblast open-source světa s tím, že jsme si moc zvykli na to, že každý je „the good guy“.

20.10.2005*Robert Krátký*

Gnomedesktop.org hlásí vydání tabulkového procesoru Gnumeric 1.6. Nová verze (a stabilní řada) přináší mimo jiné vylepšené grafy.

20.10.2005*Robert Krátký*

VMware začal zdarma poskytovat tzv. přehrávač (VMware Player) virtuálních systémů vytvořených placenou verzí produktu. Pomocí přehrávače je možné virtuální systém spustit nebo se připojit k nasdílenému.

20.10.2005*ivan*

V Linux PC Shopu v centru Prahy jsou ke koupi nová trička abclinuxu.cz.

20.10.2005*Rastislav Stanik*

OpenOffice.org 2.0 je na svete. Sťahujte o dušu spasenú.

20.10.2005*Martin Landa*

Webové stránky České skupiny uživatelů geografického informačního systému GRASS (Geogra-

phic Resources Analysis Support System) naleznete nově na grass.fsv.cvut.cz.

21.10.2005

Pavel Szalbot ml.

Boje o značku nemíjejí ani komunitu kolem Debianu. Konkrétně se diskutuje o tom, jestli jeho jméno může použít DCC Alliance. V podstatě by to nemusel být problém – členové aliance zaměstnávají vývojáře Debianu, kód (z práce na Linux Standard Base) se vrací do Debianu. Někteří vývojáři se však bojí ztráty značky Debian. Více na Linux-watch.com.

21.10.2005

Robert Krátký

Slashdot informuje, že id Software vydal vydal linuxového klienta i server pro hru Quake 4. FAQ.

21.10.2005

Robert Krátký

Švédský expert na vyhledávače pojmenoval svého syna Google. The Inquirer na tom vidí pozitivní alespoň to, že takové dítě půjde vždycky snadno najít...

21.10.2005

Robert Krátký

Podle recenze na Capnkirby.com je Mandriva 2006.0 pěkná, ale (stejně jako dřívější .0 verze) nepříliš stabilní.

21.10.2005

Radek Vokál

Pravidelně každý měsíc vychází Red Hat magazín. V jeho dvanáctém čísle se dočtete, jak se zapojit do vývoje distribuce Fedora Core, včetně informací o projektech psaných v jazyce Python a projektech v Javě.

21.10.2005

Robert Krátký

Ve výtažku z rozhovoru vydaném v časopise LinuxFormat 73 mluví Eben Moglen, právník FSF, o možných problémech při přechodu na GPL v3 a představuje svůj názor na sadu licencí Creative Commons.

21.10.2005

Robert Krátký

Stephen Shankland komentuje jednu z připravovaných nových funkcí editoru Vim 7: kontrola překlepů. Brace yourself: Spell checking in vi?

21.10.2005

Vaclav Hanzl

Florian Mueller (NoSoftwarePatents.com) má reálnou šanci získat prestižní cenu Evropan roku. Chcete-li pomoci v boji proti softwarovým patentům, hlasujte dle návodu zde.

21.10.2005

Stanislav Petr

Opera vydala novou testovací verzi svého prohlížeče – 9.0tp1. Stahovat můžete na snapshot.opera.com.

21.10.2005

Robert Krátký

Co dělat, když apt-get selže (nesplněné závislosti, konflikty mezi balíčky atd.), vám poradí článek na Linux.com: What do to when apt-get fails.

21.10.2005

Robert Krátký

Došlo ke spojení portálů (a úsilí) LinuxBIZ a LinuxWorld – nový název zní LinuxBizWorld (adresy linuxbiz.cz, linuxworld.cz, lbw.cz a linuxbizworld.cz všechny směřují na LinuxBizWorld). Na stránkách naleznete aktuality, komentáře a případové studie týkající se nasazování linuxových a open/free software řešení. Oficiální představení (staro)nového projektu proběhne v úterý 25.10. v rámci LinuxHall.

21.10.2005

Robert Krátký

Čím je Firefox Mozilla, tím je jdoplnit, OpenOffice.org. Šlo by vzít jen to nejdůležitější třeba z OO.o Writeru a vytvořit rychlejší, menší a jednodušší textový procesor? Podobně jako byl Firefox vypreparován z Mozilla Suite... Zamýšlí se nad tím Jono Bacon na O'Reilly Net.

22.10.2005

Daniel Kvasnička ml.

Světlo světa spatřil Flock, nový internetový prohlížeč založený na Mozilla Firefoxu. Je k dispozici pro Linux, Mac OS X a Windows.

22.10.2005

Robert Krátký

Verze Xorg 6.9 RC1 vyšla před pár dny a obsahuje novou 2D akcelerační architekturu EXA od Zacka Rusina. EXA by měla zvýšit výkon xcompmgr. 6.9 RC1 je od čtvrtka v Debian Experimental.

22.10.2005

Robert Krátký

V čem je OpenOffice.org lepší než StarOffice 8 od Sunu? Například od pracovního prostředí (KDE, GNOME) přebírá vzhled. OpenOffice.org 2.0 Has Edge over Its StarOffice 8 Cousin.

22.10.2005

Robert Krátký

MadPenguin.org recenzuje distribuci SUSE 10.0. Článek pokrývá vše od instalace, před desktop (KDE i GNOME) a Beagle až po FreeNX a YaST.

22.10.2005

Michal Křenek

Již od začátku října můžete stahovat amatérský (avšak zpracováním téměř profesionální) fan-film

Star Wreck: In the Pirkinning. Jedná se o parodii na Star Trek a Babylon 5, na které tým několika Finů pracoval celých 7 let! Efekty ve filmu jsou přímo úchvatné, mnohdy lepší než v původních seriálech. Celý film je šířen pod Creative Commons licencí a můžete jej stahovat ve formátu XviD (554MB). Ke stažení jsou i české titulky. Žádný fanoušek Star Treku či Babylonu 5 (a nejen fanoušek) si tento film nemůže nechat ujít ;-).

24.10.2005*Pavel Szalbot ml.*

Po 12 letech vývoje se Wine dostane do fáze, ve které ještě nebyl – bude vydána beta verze. O historii projektu a tvrdé práci, která za ním stojí, si můžete přečíst na NewsForge.

24.10.2005*Pavel Szalbot ml.*

Jeden z mnoha průvodců začátečníka i pokročilého uživatele Unixem nazvaný Unix for Advanced Users je k dispozici na webu univerzity v Indianě.

24.10.2005*Pavel Szalbot ml.*

Vývojový tým Ubuntu hrdě oznámil první verzi Ubuntu 5.10 Server, zvláštní edice Ubuntu přizpůsobená pro použití na serverech. Obsahuje oblíbené servery a jejich doplňky (Apache, MySQL, LDAP, PHP, Bind, Samba), defaultně neinstaluje X Window System a umožňuje 400M instalaci.

24.10.2005*Pavel Szalbot ml.*

Základní seznámení s tiskovým systémem CUPS (Common UNIX Printing System) spolu s důvody, proč byste jej mohli upřednostnit před LPD, si můžete přečíst v LinuxJournalu.

24.10.2005*Leoš Literák*

Populární databáze MySQL vyšla ve verzi 5.0. Její uživatelé už nemusí závidět svým kolegům uložené procedury, triggerry, pohledy a další základní funkce.

24.10.2005*Adam Rambousek*

Masarykova univerzita pořádá 1. a 2. února třetí ročník konference o elektronické podpoře výuky, SCO 2006. Hlavními tématy konference jsou současné trendy v e-learningu, standardy a jejich význam pro e-learning, LMS a jejich využití, zkušenosti s e-learningem a další. Uvítáme přednášky i praktické workshopy. Další informace, přihlášky a diskuse na webových stránkách konference.

24.10.2005*Robert Krátký*

Jak používat vývojovou verzi Firefoxu 1.5 v Ubuntu 5.10 (Breezy Badger) se dozvíte z krátkého HOWTO na LXer.com.

24.10.2005*Robert Krátký*

Philip Langdale napsal na Gnomedesktop.org o situaci, kterou se snaží řešit vývojáři prohlížeče Galeon (málo času, pomalý vývoj): The future of Galeon.

25.10.2005*Pavel Szalbot ml.*

Po nedávném článku o stavu wikipedie [zprávička] se na Registeru objevilo další zamyšlení nad Wikipedií, tentokrát částečně postavené na reakcích na minulý článek.

25.10.2005*Pavel Szalbot ml.*

Mozilla zmrazila vývojový strom Firefoxu 1.5 v rámci příprav na vydání prvního kandidáta, který by se měl objevit tento pátek. Objevilo se však několik chyb (bezpečnostních a komponenty Update), které se pro vydání opravují.

25.10.2005*Pavel Szalbot ml.*

Firmy používající PostgreSQL nedávno sponzorovaly vývoj systému, aby v blížícím se vydání získaly vytoužené funkce (konkrétně vylepšení GiST indexů). Naskytla se tak reálná cesta k žádaného vylepšení fulltextového modulu tsearch2, který by mohlo podpořit HP. Více o vývoji PostgreSQL najdete na NewsForge.

25.10.2005*Pavel Szalbot ml.*

Measurement Factory a Infoblox zveřejnily výsledky průzkumu více než 1.3 miliónů internetových DNS serverů. Výsledky potvrzují obavy z nízké úrovně zabezpečení.

25.10.2005*Robert Krátký*

O různých přístupech k designu kernelů operačních systémů je článek na systhread. Popisuje monolitické, mikro a modulární/hybridní kernely: Different Kernel Designs Overview.

25.10.2005*Martin Grombiřík*

Projekt Wine dospěl po 12 letech vývoje vydáním verze 0.9 do stadia betaverze. Oznámil to dnes ve 14:47 středoevropského času koordinátor vývoje Alexandre Julliard. Autoři doufají, že přestože i tato verze může obsahovat chyby, většina programů půjde minimálně nainstaloovat a něco užitečného bude provádět. Stahovat je možno z ibib-

lio.org a SourceForge.net (a v brzku snad i z příslušných zrcadel).

26.10.2005

Pavel Szalbot ml.

Kterak začít spravovat svoje mailové schránky textovým nástrojem Mutt spolu s klientem fetchmail a filtrem procmail si můžete přečíst v článku na Linux.com. Dozvíte se jak prvotně nakonfigurovat zmíněné nástroje.

26.10.2005

Pavel Szalbot ml.

Poté, co výrobce skeneru sítí Nessus opustil pro poslední verzi GPL, došlo k očekávanému vytvoření nové větve, která s GPL pokračuje. Co provázelo vznik projektu GNessUs, shrnuje NewsForge.

26.10.2005

Robert Krátký

Java Desktop System od Sunu bude dostupný pro více linuxových distribucí. Součástí JDS je prohlížeč, StarOffice a Java virtual machine. Sun gives dekstop Linux another push.

26.10.2005

Robert Krátký

Na světě je CrossOver Office 5 – upravená verze Wine, kterou prodává firma Codeweavers. Obsahuje – mimo jiné – základní podporu pro provoz MS Office 2003 a přepracovaného správce oken. Announcing CrossOver Office Version 5.0.

26.10.2005

Robert Krátký

Vyšla verze 2.1 oblíbeného systému pro správu obsahu (CMS) – Plone – pro aplikační server Zope. Plone Foundation Releases Plone 2.1.

26.10.2005

Robert Krátký

Sličná jepice (alias Dapper Drake) by měla vyjít v dubnu 2006. Nová verze (K)Ubuntu má být především skvěle vyladěná pro bezproblémové použití na desktopu. Bude to také první verze, která bude oficiálně podporována (desktop 3 roky a server 5 let) i po vydání dalších. Mark Shuttleworth o Dapper.

26.10.2005

Tomas

Dnes je v rádiu Leonardo na programu v 22:00 v Magazínu Nula-jednička téma Proč zvolit operační systém Linux?

26.10.2005

SiLK

Ve dnech 12.-13. listopadu se v budově ČVUT na Karlově náměstí bude konat pátý ročník semináře InstallFest. Věnuje se GNU/Linuxu a je určený začátečníkům a mírně pokročilým. Přijedou zajímaví hosté, novinkou jsou workshopy

a program je vskutku bohatý. Posuďte sami na <http://www.installfest.cz>.

27.10.2005

Pavel Szalbot ml.

Matthew Szulik, CEO RedHatu, se ve středu distancoval od dřívějších domněnek o tom, že by velké společnosti jako RedHat měly kontrolovat, co open source komunita vytváří. Důležité je podle něj zůstat věrný GPL. Kdo nepodporuje software bez patentů či porušuje GPL, brzdí inovaci. Více v článku „RedHat CEO decries open source pretenders“.

27.10.2005

Pavel Szalbot ml.

Když přijde řeč na digitalizaci knih, začnou se rozvíjet dva příběhy – jeden je o open source a druhý o Googlu. Neziskové organizace chtějí digitalizovat veškeré vědění a bojí se, aby neskončilo v privátním sektoru. Google naopak lituje toho, že se záslužný čin stává bojem dvou projektů se společným cílem. O digitalizaci knih se dočtete na ZDNetu.

27.10.2005

Pavel Szalbot ml.

Na OSNews se objevil článek zabývající se SSH přihlašováním bez hesla pomocí DSA klíče, jeho správným nastavením a výhodami či riziky s tímto procesem spojenými.

27.10.2005

Robert Krátký

Jak vytáhnout, seřadit a zobrazit (v grafu) data z textu (například emailových zpráv) pomocí rutiny AWK vám poradí článek na LinuxJournal. Okay, sed experts, have at it.

27.10.2005

Petr Tomeš

Tým Ubuntu CZ nabízí bezplatné zaslání CD s Ubuntu 5.10. K dispozici je 620 CD (500 ks pro architekturu x86, 100 ks pro amd64 a 20 ks pro powerpc), celkem 30 kg :).

27.10.2005

Robert Krátký

Podle nedávné studie Evans Data Corp. se IT administrátoři čím dál více bojí bezpečnostních problémů v proprietárních databázových produktech a na open source databáze nahlížíjí jako na možné řešení. Za poslední půlrok se nasazování open source databází zvýšilo o 20 procent. Are open source databases more secure?

27.10.2005

Robert Krátký

V prvním díle série článků o notačním softwaru pro Linux se Dave Phillips na LinuxJournal věnuje systému abc a dvěma grafickým nadstavbám.

28.10.2005*Jose*

Študenti Univerzity Tomáše Bati ve Zlíně si od tohoto roku mohou vybrat šablónu pro psaní diplomové práce. A to vo formátech MS Word, OpenOffice.org a LaTeX. Šablóny sú naformátované tak, aby sa všetky práce pomocou nich vytvorené, čo najviac podobali.

28.10.2005*Josef Vybíral*

Včera byla vydána nová opravná verze grafického editoru GIMP (v2.2.9). Mezi opravami je například i ulehčení uzavírání selekce tvořené inteligentními nůžkami. Seznam změn (cz).

28.10.2005*Robert Krátký*

Po dvou odkladech kvůli chybám, z nichž některé se nakonec ukázaly být plané popluchy, vyšlo ve čtvrtek jádro 2.6.14. Oznámení v LKML.

29.10.2005*Pavel Szalbot ml.*

Je libo databázový systém Oracle zdarma? V ořezané verzi to bude brzy možné. Oracle by měl v nejbližší době oznámit beta verzi Oracle 10g Express Edition limitovanou na 1 CPU, 4GB diskového prostoru a 1GB RAM. Zřejmě reakce na sílící tlak od „low-end“ open source databází.

29.10.2005*Bohumír Zámečník*

Víte, že kolem Slunce obíhá planetka jménem 9885 Linux? Dokonce má i kolegy jménem 9965 GNU, 9793 Torvalds, 9882 Stallman, nebo také 7796 Jára Cimrman. Ale nebojte se, spadnout na Zem se nejspíš v dohledné době nechystá.

29.10.2005*Luboš Doležel*

Po dlouhém čekání je tu hra America's Army 2.5.0 pro Linux.

30.10.2005*Jan Grmela*

Podle zpráv z fóra instantního klienta SIM, jehož vývoj se jako fork CVS verze 0.9.4 přesunul na Berlios, je k dispozici již téměř funkční QT4 verze, kterou je možno získat přes Subversion.

31.10.2005*Pavel Szalbot ml.*

CosmoPOD nabízí vzdálený KDE desktop přes NX zdarma. Více o službě nebo o tom, proč autoři zvolili KDE, najdete na dot.kde.org.

31.10.2005*Pavel Szalbot ml.*

V úvodním díle seriálu Modern Memory Management ONLamp.com rozebírá způsoby správy paměti na moderních unixových operačních systémech.

31.10.2005*Pavel Szalbot ml.*

BeleniX je první live CD postavené na OpenSolaris, které bootuje do X Window System s XFce. Vyvinula ho indická pobočka Sunu za účelem popularizace systému v oblasti. OSDir nabízí sérii screenshotů.

31.10.2005*Robert Krátký*

Jak si nová distribuce poradí se starším notebookem (nebo notebook s distribucí)? V prvním díle série článků si můžete přečíst o instalaci Fedora Core 4: New Linux with an Old Laptop: Fedora Core 4.

31.10.2005*Robert Krátký*

Steven J. Vaughan-Nichols píše na eWeek o tom, že ačkoliv dnes už si nikdo nemyslí, že by SCO nějak poškozovalo důvěryhodnost Linuxu, jen tak se ho nezbavíme. Soud s IBM je naplánován na únor 2007...SCO Isn't Going Away Anytime Soon.

31.10.2005*Robert Krátký*

MadPenguin recenzuje knihu Moving to Linux. A nejvíce se mu líbí, že bere téma s humorem. Na rozdíl od jiných knih-průvodců pro přechod k Linuxu se také neštítí příkazové řádky. Fighting FUD With Humor.

31.10.2005*Robert Krátký*

Co dělat, když má systém bolístku? V rozsáhlém článku (Linux Magazine) představuje a vysvětluje Roderick W. Smith nejružnější způsoby záchrany. The Linux Emergency Room.

31.10.2005*Robert Krátký*

Podle článku na eWeek by měla být GPL verze 3.0 představena již počátkem roku 2006. Eben Moglen popisuje, jak bude probíhat připomínkování. Datum vydání bude upřesněno příští měsíc.

31.10.2005*Robert Krátký*

Jestli má ještě Slackware smysl se ptát komentářů na Linux-Watch (nebo už nastal čas distribucí, ve kterých je editace konfiguračních souborů na obtíž?).