



Abíčko

Časopis serveru abclinuxu.cz

Listopad 2006



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: info@abclinuxu.cz. Sponzoring Abíčka a jiné formy reklamy si objednávejte na adrese: info@stickfish.cz. Ostatní dotazy směřujte na adresu: info@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>.

©2006 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	2
Obsah	3
Distribuční novinky – 3	6
Letem světem	6
Velké ryby	6
Gentoo	6
openSUSE	6
Debian	6
KUbuntu Edgy Eft Beta	6
Tichá voda	6
Berry Linux 0.74	6
Kurumin Linux 6.1 „Light“	6
Cross Linux From Scratch	7
Distribuční rada	7
Rozhovor: Michal Švec, SUSE	7
Distribuční novinky – 4	9
Letem světem	9
Velké ryby	9
openSUSE 10.2 alpha5	9
Mandriva 2007	9
Tichá voda	9
Slax	9
Slackintosh	9
SabayonLinux	9
Parsix	9
Distribuční rada	9
Rozhovor: Tomáš Matějček, Slax	10
Distribuční novinky – 5	12
Letem světem	12
Velké ryby	12
PC-BSD	12
Tichá voda	12
Litrix	12
Elive	12
Puppy Linux	12
DeLi	12
FreeSBIE	12
Distribuční rada	12
Distribuční novinky – 6	15
Letem světem	15
Velké ryby	15
Ubuntu, Kubuntu, Edubuntu 6.10 Edgy Eft	15
Fedora Core 6	15
Tichá voda	15
B2D	15
Astaro Security Gateway	15
Fedora Core 6 „Live-Spin“	15
VectorLinux	15
Distribuční rada	15
Fedora Core 6	18
Instalace	18
První start	19
Fedora jako pracovní stanice	19
GNOME	19
Ostatní programy	19
Multimédia	20
Fedora jako server	20
Vývojové programy	20
Závěr	20
GPS a komunikační protokol NMEA – 3 (dekódování dat)	22
Dekódování přijatých dat	22
GSA, aktivní satelity a DOP (Dilution Of Precision)	22
RMC (Recommended Minimum Navigation Information)	22

GSV (Satellites in View)	24
GGA – zeměpisná délka a šířka, geodetická výška, čas určení souřadnic	24
Souřadné systémy	24
Shrnutí	25
Jazyky a překladače – 5 (syntaxe 3)	26
Bison	26
Další možnosti	27
Konflikty	28
Antlr	28
Závěr	29
Jazyky a překladače – 6 (sémantika a typy 1)	31
Princip překladače	31
Sémantická analýza	31
Typové systémy	32
Statické typování	32
Dynamické typování	33
Závěr	34
Jazyky a překladače – 7 (sémantika a typy 2)	35
Metoda versus typ – aneb dynamický a statický přístup	35
Reflexe	35
Kolekce	36
Eval, exec, apply a jiné	37
Ostatní	38
Rozšíření jazyka	38
Univerzální třída	38
Závěr	38
Kubuntu 6.10 Edgy Eft – Instalace ve dvojím stylu	40
Ubuntu a spol.	40
Novinky ve verzi 6.10	40
Nová instalace nebo upgrade?	40
Instalační média	40
Instalace systému z LiveCD: nejschůdnější cesta pro nováčky	41
Volby úvodní obrazovky	41
Trable v grafickém prostředí LiveCD	41
Příprava instalace Kubuntu	42
Příprava oddílů pevného disku	43
Instalace GRUB: automaticky do MBR	43
Vlastní instalace	43
Alternativní instalace systému: OEM pro obchodníky a administrátory	43
Instalace v textovém režimu	43
Instalace zavaděče GRUB	45
Závěrem	45
PIM pro GNU/Linux – 5 (Haystack)	47
Váš smysl informací	47
Koncepce programu	47
Přehled funkcí	47
Výhody a nevýhody	48
Závěr	48
PIM pro GNU/Linux – 6 (Mulberry)	50
K čemu další program?	50
IMAP nadě vše	50
Přehled funkcí	50
Výhody a nevýhody	51
Závěr	52
Ako na preklad slobodných programov	53
Úvod	53
Pár dôležitých pojmov na úvod	53
Ako si uľahčiť prácu?	53
Sumarizácia	55
Preklad pomocou nástrojov GNU gettext	55
Národné prostredie	55
Tvorba po súboru	56
Prekladanie po súboru	56
Špeciality prekladu	57

Ostatné druhy prekladov	58
Pár tipov a rád na záver	58
Lokalizačné projekty a tímy	58
VLC – 6 (správce vysílání)	60
Knihovny LIVE555 Streaming Media	60
Protokoly RTP, SDP a RTSP	60
Parametry modulu rtp	60
Správce vysílání – VideoLAN Manager	61
Parametry zdrojů typu broadcast	62
Použití plánovače	62
Video na přání	63
Nastavení vzhledu XDM	64
Úvod	64
Pozadí	64
Přihlašovací okno	65
Závěr	65
SuperTux – svobodný Super Mario	66
Grafika, hudba	66
Budoucnost: Milestone 2	66
Hráčské levely a level-sety	67
Instalace	67
Instalace na Debianu	67
Instalace pro SUSE Linux	67
Gentoo	67
Instalace pro Windows	67
Jaderné noviny – 13. 9. 2006	68
Aktuální verze jádra: 2.6.17.13	68
Citáty týdne: Andrew Morton a tým -stable	68
MMIO bariéry	68
Pokus o vzkříšení kvalifikací vLinuxu	70
Jaderné noviny – 20. 9. 2006	72
Aktuální verze jádra: 2.6.18	72
Sledovací infrastruktury	72
Další implementace kontejnerů	73
nopage() a nopfn()	74
Jaderné noviny – 27. 9. 2006	76
Aktuální verze jádra: 2.6.18	76
Citáty týdne: Linus Torvalds, Bruce Perens	76
Začíná vývojový cyklus 2.6.19	76
Změny ovladačového API v 2.6.19	77
Postoj vývojářů jádra k GPLv3	78
Jaderné noviny – 4. 10. 2006	80
Aktuální verze jádra: 2.6.18	80
Citáty týdne: Alan Cox, Linus Torvalds, Andrew Morton	80
Další věci pro 2.6.19	80
Změny API: zpracovávající přerušení a vektorované I/O	81
Poslední Wireless Extensions?	82
Prezentace a fotografie ze setkání Kernel Netconf 2006	83
Uspání a probuzení	83
Kvůli varováním utíkají chyby	83
Jaderné noviny – 11. 10. 2006	85
Aktuální verze jádra: 2.6.19-rc1	85
Citát týdne: David Woodhouse	85
Odstranění populate(), nopfn() a nopage()	85
Video4Linux2 API: úvod	86
Budoucnost vývoje ReiserFS	87
Zprávičky	88

Distribuční novinky – 3

Co nového chystají nebo již nabídly velké distribuce Gentoo, Debian, openSUSE a Fedora Core. Představení malých distribucí Berry Linux, Kurumin Linux a Cross Linux From Scratch. Jak v Debianu a Ubuntu instalovat samostatné balíčky. Rozhovor o openSUSE 10.2: Michal Švec.

Marek Stopka

Letem světem

Byla vydána stabilní verze distribuce Mandriva 2007, o jejíž RC verzi se již psalo v [Distribučních novinkách – 1](#) [1]. Po více než roce očekávání a usilovné práce Patricka Volkerdinga je k dispozici také nový Slackware 11, o němž se dočtete v [Distribučních novinkách – 2](#) [2]. Stejně jako u Ubuntu byl „init“ nahrazen i u Fedory modernějším systémem, který se stará o nabootování a start distribuce. Nové řešení je založeno na rozhraní DBUS. Přečíst o tomto významném kroku si můžete na webu fedoraproject.org [3].

Velké ryby

Gentoo

Vývojáři distribuce Gentoo [založili](#) [4] nový projekt, který si klade za cíl vytvořit další stupeň následující za stage 3. Jedná se o ebuildy instalující určitou sadu aplikací, které slouží k nějakému účelu, a ve většině nasazeních bývají instalovány spolu. V současné době je k dispozici ebuild LAMP server, který obsahuje Apache/MySQL/PHP. Zatím roste tento projekt mimo oficiální portage strom, a tak musíte k využití použít portage overlay <http://overlays.gentoo.org/proj/seeds/> [5]. Postup přidávání overlay najdete v [Distribučních novinkách 1](#) [6].

openSUSE

Stejně jako do Mandrivy, i do distribuce openSUSE bylo doplněno grafické rozhraní pro konfiguraci XGL/Compiz 3D grafických efektů. Jmenuje se „Compiz Settings Manager“ a je to nová aplikace pro rozhraní Gnome. Instalovat si jej můžete do openSUSE 10.1 například prostřednictvím konfiguračního programu YaST. Jak instalaci provést, si můžete přečíst na webu linux.wordpress.com [7], kde jsou doplněny i screenshoty. Tento program může být nainstalován i do Ubuntu a dalších distribucí.

Debian

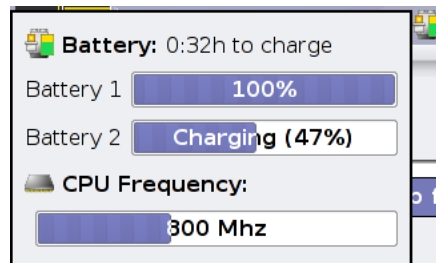
Vývojářům distribuce Debian byl Mikem Connorem z Mozilla Corporation [zaslán](#) [8] bugreport, který si

stěžuje na nedodržování ochranné známky „firefox [9]“. Takže s největší pravděpodobností bude balíček „firefox [10]“ odstraněn a zbyde pouze balíček „mozilla-firefox [11]“.

Bug má číslo #354622 a zní: *Firefox (the name) is equally protected and controlled by the same trademark policy and legal requirements as the Firefox logo. You're free to use any other name for the browser bits, but calling the browser Firefox requires the same approvals as are required for using the logo and other artwork.*

KUbuntu Edgy Eft Beta

Betaverze Kubuntu 6.10 (Edgy Eft) je nyní k dispozici pro stahování a testování. Přichází s množstvím nových vlastností. Bylo předěláno uživatelské rozhraní, za což vděčíme odborníkům na ergonomii z týmu KDE. Byl přidán applet na správu napájení (power management) používající HAL a podporující hibernaci a zobrazování stavu baterie. Dále pak došlo k přidání ikonky Oxygen a předělání bootsplash a přihlašovací obrazovky. Nové verze hlavních aplikací: [KDE 3.5.4](#) [12], [Konversation](#) [13] 1.0, [Amarok 1.4.3](#) [14]. [Amarok](#) [15] v nové verzi obsahuje patch, který automaticky zvládne nainstalovat podporu pro MP3.



Tichá voda

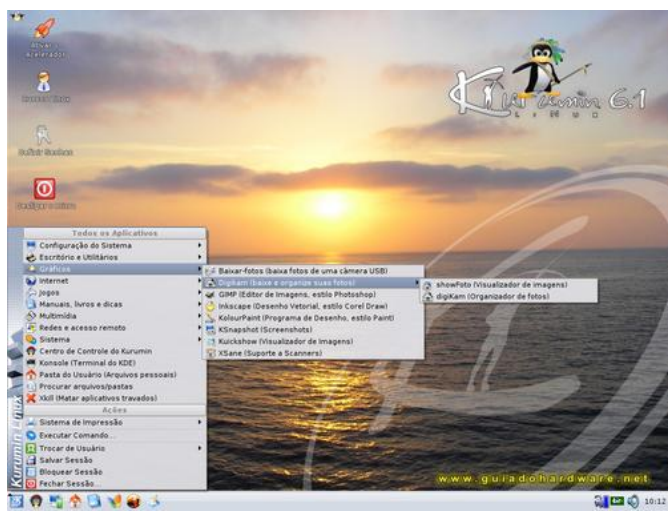
Berry Linux 0.74

Yuichiro Nakada vydal novou verzi své na Fedoře založené a na desktop orientované distribuce a LiveCD, která obsahuje podporu pro japonštinu a angličtinu. Bylo aktualizováno množství balíčků, včetně [Mozilly Firefox](#) [16] a [Thunderbird](#) [17]. Distribuce bě-

ží na novém kernelu 2.6.17.11 s podporou SMP. Zahrnuty byly také patche na bootplash. Jak je ze screenshotu patrné, distribuce se snaží podobat operačnímu systému Microsoft Windows, a tak je pro webový prohlížeč použita ikona podobná té, kterou používá Microsoft Internet Explorer.

Kurumin Linux 6.1 Light

Na světě je nová verze brazilské distribuce **Kurumin** [18] (v portugalštině), která začala jako LiveCD o velikosti 200 MB v roce 2003. Avšak i přes svou velikost obsahuje **OpenOffice.org** [19], Javu a spoustu dalších populárních programů. Jak název této verze napovídá, distribuci bylo odlehčeno od větších balíčků, a tak je její velikost 182 MB. Distribuce je tedy připravena k instalaci na libovolné mini-CD či na USB mass-storage (flashdisk) zařízení s kapacitou 256 MB. Odstraněna byla třeba podpora pro scannery a 3D akceleraci. Nově byla distribuce také optimalizována na rychlost startu a spotřebu operační paměti.



Kurumin Linux 6.1

Cross Linux From Scratch

K dispozici je verze 1.0.0 [20] „distribuce“ Cross Linux From Scratch. V podstatě se jedná o knihu, která vás provede postupem, jak napříč platformami kompilovat vlastní na zdrojových kódech založenou distribuci. Základní postup je založen na Glibc 2.4, GCC [21] 4.1.1 a binutils 2.17. Je k dispozici pro architektury x86, x86-64, SPARC, PowerPC, PPC64, MIPS, MIPS64, a Alpha.

Distribuční rada

Před nedávnem vyšla nová verze **Operry** [22]. Jste uživatelem na Debianu založené distribuce a stáhli jste si instalační .deb balíček, ale nevíte, jak jej nain-

stalovat? Řešení nebude tak těžké, jak se může zdát. Použijeme k tomu program **dpkg** s parametrem **-i**.

Čili, za předpokladu že se debianní balíček bude jmenovat **xyz.deb**, pak k jeho instalaci spustíme příkaz **dpkg -i xyz.deb**. Tato metoda instalace za nás však automaticky nebude řešit závislosti, a tak pokud na konci napíše informaci, že je potřeba doinstalovat nějaké závislosti, spustíte **apt-get -f install**, který potřebné závislosti doinstaluje (za předpokladu, že jsou v **repozitářích** [23] umístěné). Při použití uvedeného příkazu dávejte pozor; občas se snaží nějaké balíčky odinstalovat, a tak se jeho činnost musí hlídat.

Rozhovor: Michal Švec, SUSE

*** Otázka:** Jak dopadl Summer of Code pro openSUSE? Kolik projektů bylo přijato, kolik jich bylo dokončeno a jak byly významné?

▷ Odpověď: Přijato bylo 7 projektů, z toho bylo dokončeno 6, což si myslím není tak špatné na to, že to byla první účast openSUSE na Summer of Code. Z dokončených projektů najdete v openSUSE 10.2 například KDE applet pro instalaci balíčků a možná i další, jako například Gtk+ rozhraní pro YaST. Ostatní projekty se týkaly převážně Build Service (služba pro tvorbu balíčků pro různé distribuce) nebo správy workflow, nástroje SWAMP.

*** Otázka:** Novell silně tlačí na XGL, dokonce zastřešuje vývoj nového správce oken Compiz. Jak pokračují práce v této oblasti? Myslíte si, že tato funkčnost je skutečně užitečná, není to jen pozlátka pro hráčky?

▷ Odpověď: Práce na Xgl a Compiz úspěšně pokračují, před několika dny byla například vydána první oficiální verze **compiz (0.2)**. Pokračuje jak rozšiřování a stabilizace podpory grafických karet, tak i práce na nových vlastnostech a pluginech (nově například podpora D-Bus nebo plugin Plane, alternativní přepínání desktopů).

Co se týče praktické použitelnosti Xgl, hodně to závisí na vkusu a preferencích konkrétního uživatele. Sice se to na první pohled může jevit jako nefunkční pozlátka, nicméně nesmíme zapomínat, že právě takovéto věci mohou přilákat uživatele, kterým strohé, funkční prostředí nevyhovuje nebo se jim prostě nelíbí a mají radši něco zajímavého, co jejich každodenní pracovní prostředí upoutá a oživí.

*** Otázka:** Kterou novinku z připravované verze 10.2 máte nejraději? Která je podle vás nejdůležitější?

▷ **Odpověď:** Jedna ze zajímavých novinek v openSUSE 10.2 bude například lepší podpora systémů s více procesory nebo jádry, což bude užitečné zejména u notebooků, kde bude možné vypnout druhé jádro (případně jakékoliv další), a tím dosáhnout výrazné úspory energie, nicméně i u serverů může být zajímavé vypnout 8 z 16 procesorů.

Mezi další novinky patří například podpora pro clusterový souborový systém OCFS2, nástroj pam-config pro konfiguraci PAM (autorizace uživatelů) z příkazové řádky, nové menu v KDE, rychlejší instalace nebo také nově definované výběry balíčků. ■

Odkazy

- [1] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-1>
- [2] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-2>
- [3] <http://fedoraproject.org/wiki/FCNewInit>
- [4] <http://www.abclinuxu.cz/zpravicky/projekt-gentoo-seeds-a-problemy-s-nim>
- [5] <http://overlays.gentoo.org/proj/seeds/>
- [6] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-1>
- [7] <http://linux.wordpress.com/2006/09/24/suse-101-new-compiz-manager-themes/>
- [8] <http://www.abclinuxu.cz/zpravicky/firefox-neni-podle-debianu-uplne-svobodny>
- [9] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>
- [10] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>
- [11] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>
- [12] <http://www.abclinuxu.cz/zpravicky/kde-3.5.4-oficialne-ohlaseno>
- [13] <http://www.abclinuxu.cz/software/internet/im/konversation>
- [14] <http://www.abclinuxu.cz/zpravicky/vysel-amarok-1.4.3>
- [15] <http://www.abclinuxu.cz/software/multimedia/audio/prehravace/amarok>
- [16] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>
- [17] <http://www.abclinuxu.cz/software/internet/posta/mozilla-thunderbird>
- [18] <http://www.kuruminlinux.com.br/>
- [19] <http://www.abclinuxu.cz/software/kancelar/baliky/openoffice.org>
- [20] <http://cross-lfs.org/view/1.0.0/>
- [21] <http://www.abclinuxu.cz/software/programovani/kompilatory/gcc>
- [22] <http://www.abclinuxu.cz/software/internet/www/opera>
- [23] <http://www.abclinuxu.cz/slovník/repozitar>

Distribuční novinky – 4

Vyšla další testovací verze openSUSE 10.2 a finální verze Mandriva 2007. Z malých distribucí zmíníme nový SLAX, Slackintosh, Sabayon a Parsix. Jak do Ubuntu přiinstalovat podporu MP3 a dalších proprietárních formátů: EasyUbuntu. Rozhovor: Tomáš Matějček o SLAX 5.1.8.

Marek Stopka

Letem světem

Vyšla nová verze živé distribuce z českých luhů a hájů: [Slax 5.1.8](#) [1]. Rozhovor s jejím tvůrcem Tomášem Matějčkem najdete na konci článku. Vývojový tým SUSE se činí, a tak je k dispozici další – již pátá – testovací verze openSUSE 10.2 s kódovým označením Basilisk Lizard. Pokud snad máte v oblibě starší počítače Apple (popřípadě jiné PPC počítače) a distribuci Slackware, pak by vaší pozornosti nemělo ujít vydání distribuce Slackintosh 11.0.

Velké ryby

openSUSE 10.2 alpha5

Nové openSUSE je zatím k dispozici jen pro architektury i386 a x86-64 a to na DVD médiu. Systém běží na novém kernelu 2.6.18 a obsahuje Glibc i Python ve verzi 2.5. Instalátor nově podporuje LVM svazky, takže můžete pohodlně instalovat na více disků naráz.

Byla aktualizována obě hlavní desktopová prostředí. GNOME [2] na verzi 2.16.1 a KDE [3] na build ze SVN [4]. Pokud by jej snad chtěl někdo používat, upozorním na nefunkčnost aplikace kdesu. Chyba se projevuje například při pokusu o spuštění yast2; problém lze obejít pomocí použití su a spuštění požadované aplikace z rootovského terminálu.

Mandriva 2007

Vyšla také již stabilní verze Mandrivy 2007 [5] a jako hlavní novinka je uváděno použití AIGLX a XGL akcelerovaného kompozitního desktopu. O 1. RC verzi Mandrivy 2007 [6] se psalo v [Distribučních novinách 1](#) [7].

Tichá voda

Slax

Slax [8] není jen spustitelné CD s Linuxem, ale také sada skriptů, které vám usnadní budování vlastního LiveCD se vším, co potřebujete. O těchto skriptech pojednává 4. díl seriálu od [Michala Vyskočila](#) [9]: Li-

veCD – 4 ([linux live](#), před spuštěním systému) [10]. Distribuce je založena na Slackware a může být rozšířena pomocí modulů obsahujících spoustu programů. V současné době jich je na webu projektu 1294. Moduly jsou roztříděny do kategorií podle určení. Mohou být „připáleny“ na CD, a být tak přístupné při každém startu distribuce, nebo můžete použít příkaz `uselivemod` a nahrát modul za běhu v nabootovaném systému. Další velice zajímavou funkcí je „webconfig“. Umožňuje uložit na webový server autora heslem chráněnou konfiguraci a v případě dalšího bootování si ji pohodlně stáhnout a nabootovat do již zkonfigurovaného prostředí.

Slackintosh

Jak již jsem v úvodu řekl, pro milovníky PowerPC procesorů vyšla upravená verze distribuce Slackware, která se jmenuje [Slackintosh](#) [11]. Běží na kernelu Linux 2.6.17.13 a obsahuje KDE [12] 3.5.4. S vydáním nové verze také končí podpora pro starší verzi 10.2, a tak je doporučeno upgradovat na verzi novou. Při upgradu se také projevuje bug v Qt [13]. Obejít se dá tak, že Qt odinstalujete, provedete upgrade distribuce a pak Qt nainstalujete zpět.

SabayonLinux

Rychlý vývojový cyklus distribuce SabayonLinux [14] pokračuje s čerstvou verzí 3.1. Jedná se o LiveDVD založené na distribuci Gentoo, které promění libovolný x86 počítač na výkonný desktop za méně než 5 minut. Distribuce může být také instalována na disk. Obsahuje spoustu desktopových prostředí a správců oken (KDE [15], GNOME [16], Xfce [17], Fluxbox [18]) a aplikací pro desktopové použití (KOffice [19], OpenOffice.org [20], FreeNX, Amarok [21], Kaffeine [22], ...). Novou vlastností je možnost nastavit pevně požadované rozlišení. Provádí se parametrem `res`, například `res=1024x768`. Dále aktualizace ovladačů ATI a výměna kompozitního manageru Compiz [23] Quinnstorm za Beryl [24], což je v podstatě totéž jen pod jiným jménem.

Parsix

Alan Baghumian oznámil novou testovací verzi své distribuce **Parsix** [25]. Spousta balíčků byla aktualizována z debianního testing repozitáře. Byl nasazen kernel 2.6.18 se sadou patchů a přídatných ovladačů. Distribuce podporuje zápis i čtení z NTFS oddílů pomocí ovladače ntfs3g. Plánované oficiální vydání je 15. listopadu tohoto roku, takže testujte a reportujte bugy, ať ve finální verzi nejsou.

Distribuční rada

Zrovna jste si čerstvě nainstalovali Ubuntu a zjistili jste, že si nemůžete přehrát své oblíbené skladby, které máte ve formátu MP3? Pak je tato rada určena právě vám. Z licenčních důvodů nejsou v původní instalaci Ubuntu (stejně jako u spousty dalších distribucí) obsaženy balíčky s kodeky pro přehrávání některých formátů. Proto vznikl program easyUbuntu, který vám pomůže si velice jednoduchým způsobem nainstalovat kodeky, přehrávač Fla-

she, Javu atd. Program sídlí na webové stránce easyubuntu.freecontrib.org [26]. Odtud si můžete [stáhnout](#) [27] poměrně jednoduchý program, který nainstalujete pomocí postupu uvedeného v [předchozím díle](#) [28]. Tedy příkazem, jehož výstup by měl vypadat přibližně následovně:

```
sudo dpkg -i easyubuntu.deb
Password:
Instaluji balík easyubuntu.
(Čtu databázi ... nyní je nainstalováno \\  
152276 souborů a adresářů.)
Rozbaluji easyubuntu (z easyubuntu.deb) ...
Nastavuji balík easyubuntu \\  
(3.0.22-1cafuego1) ...
```



Parsix

Nyní již jen stačí do terminálu napsat `easyubuntu`. Pokud pracujete jako běžný uživatel, ještě na vás vyskočí výzva k zadání hesla. Pak si stačí vybrat, co budete potřebovat, pomocí zaškrtačkových políček. Jakmile tak učiníte, stiskněte tlačítko OK a balíčky se nainstalují.

Rozhovor: Tomáš Matějček, Slax

★ **Otázka:** Jak jsi daleko s připravovanou verzí SLAXu 5.1.8? Kdy se na ni můžeme těšit?

▷ **Odpověď:** Měla by být hotová :-). V tuto chvíli se uploadují soubory na server. Pokud během dneška a zítřka [pozn. redakce: rozhovor proběhl 8. října] nenarazím na žádnou chybu, měl by SLAX 5.1.8 vyjít příští týden.

★ **Otázka:** Šestková řada chystá několik zajímavých novinek, mohl bys nám ty nejzajímavější představit?

▷ **Odpověď:** S radostí. Nejvíc důležitá (a přitom pro uživatele skoro neviditelná) změna je náhrada `unionfs` filesystému za něco lepšího. Aufs od jednoho japonského vývojáře se zdá být tou pravou alternativou. Nabízí stabilnější běh a mnoho zajímavých vylepšení, mimo jiné podporu `mmap`. Další chystaná novinka je změna kompresního algoritmu v `squashfs` filesystému – místo standardního `gzipu` bude pou-

žita LZMA komprese. Ta slibuje až o 30 % větší úsporu místa, což už koncoví uživatelé pocítí. Do SLAXu by se konečně mohly dostat ovladače tiskáren a podpora tisku obecně. Chystají se ještě některé další novinky, náznakem se o nich dá dočíst na slax.org/todo.php [29].

★ **Otázka:** Kolik lidí se na vývoji SLAXu podílí? Jsou jen z Česka nebo i ze zahraničí?

▷ **Odpověď:** Slax vyvíjím já sám. Ale nikdy by Slax nebyl tím, čím je dnes, bez tisíců lidí z celého světa, kteří Slax používají a hlásí objevené chyby.

★ **Otázka:** Mít kouzelnou hůlku, kterou část jádra či Linuxu bys změnil či dokončil?

▷ **Odpověď:** Kdyby to šlo jen tak kouzlem, rád bych se zbavil problémů s `squashfs+unionfs+SMP`. Kombinace těchto dvou filesystémů spolu s víceprocesorovým počítačem dělá potíže (asi někde na úrovni zamykání souborů). Je to důvod, proč je Slax 5 v tuto chvíli k dispozici pouze s non-SMP jádrem (na počítačích s více procesory využije jen jeden procesor). Aufs se v této kombinaci chová stejně jako `unionfs`, nicméně jsem v kontaktu s jeho vývojářem, tak snad se mu podaří něco vymyslet :-).

■

Odkazy

- [1] <http://www.abclinuxu.cz/zpravicky/slax-5.1.8>
- [2] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>
- [3] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [4] <http://www.abclinuxu.cz/software/programovani/sprava-verzi/subversion>
- [5] <http://www.abclinuxu.cz/zpravicky/mandrivalinux-2007-je-venku>
- [6] <http://www.abclinuxu.cz/zpravicky/mandriva-linux-2007-rc1-mona>
- [7] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-1>
- [8] <http://www.slax.cz/>
- [9] <http://www.abclinuxu.cz/Profile/5004>
- [10] <http://www.abclinuxu.cz/clanky/system/livecd-4-linux-live-pred-spustenim-systemu>
- [11] <http://workaround.ch/index.html>
- [12] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [13] <http://www.abclinuxu.cz/software/programovani/knihovny/qt>
- [14] <http://www.sabayonlinux.org/>
- [15] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [16] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>
- [17] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/xfce>
- [18] <http://www.abclinuxu.cz/software/pracovni-prostredi/spravci-oken/fluxbox>
- [19] <http://www.abclinuxu.cz/software/kancelar/baliky/koffice>
- [20] <http://www.abclinuxu.cz/software/kancelar/baliky/openoffice.org>
- [21] <http://www.abclinuxu.cz/software/multimedia/audio/prehravace/amarok>
- [22] <http://www.abclinuxu.cz/software/multimedia/video/prehravace/kaffeine>
- [23] <http://www.abclinuxu.cz/software/pracovni-prostredi/spravci-oken/compiz>
- [24] <http://www.abclinuxu.cz/software/pracovni-prostredi/spravci-oken/beryl>
- [25] <http://www.parsix.org/html/index.php>
- [26] <http://easyubuntu.freecontrib.org/>
- [27] <http://easyubuntu.cafuego.net/pool/main/easyubuntu/easyubuntu.deb>
- [28] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-3>
- [29] <http://slax.org/todo.php>

Distribuční novinky – 5

Projekt PC-BSD odkoupen. Nové verze distribucí Litrix, Elive, Puppy Linux, DeLi a BSD liveCD FreeSBIE. Jak v Gentoo aktualizovat programy z CVS, SVN a Darcs.

Marek Stopka

Letem světem

Vyšla nová verze Litrix 6.10, což je LiveCD v portugalské. Pokud chcete vyzkoušet [Enlightenment](#) [1], pak je pro vás k dispozici testovací verze distribuce Elive. Projekt openSUSE má aktualizované sady instalačních CD a DVD s kódovým názvem „remastered“. Datum vydání stabilní verze distribuce Fedora Core bylo opět posunuto [2], tentokrát o několik dní. Může za to sada o víkendů nalezených bugů. Vyšla také nová verze distribuce DeLi. Vyšla první stabilní verze firewallového systému pfSense založeného na FreeBSD.

Velké ryby

PC-BSD

Projekt PC-BSD, jehož cílem je tvorba uživatelsky přívětivé verze FreeBSD zaměřené na desktop, minulý týden oznámil, že byl odkoupen kalifornskou společností iXsystems. Pokud tuto společnost neznáte, tak ji představím krátkou citací z její [webové stránky](#) [3]:

iXsystems je vedoucí společností na trhu výkonných výpočetních clusterů, blade serverů a úložných řešení. iXsystems dodává servery s FreeBSD, NetBSD, OpenBSD a Linuxem napříč průmyslem. Spousta uživatelů této distribuce se obává, že se z ní stane distribuce komerční. Zakladatel projektu však rychle řekl, že je příliš brzy vynášet nějaké závěry. Na webových stránkách se v prohlášení objevilo:

Nebudeme „close-sourcovat“, rušit „free“ verzi PC-BSD, nebo dokonce „komercializovat“. PC-BSD bude dostupné na www.pcbbsd.org jako vždy.

Tichá voda

Litrix

Vyšla nová verze Litrix 6.10, což je brazilská distribuce a LiveCD založené na Gentoo. Distribuce je plně lokalizovaná do brazilské portugalské. Obsahuje nový konfigurační panel, nové rozhraní pro konfiguraci ADSL připojení, přibyla také aplikace pro P2P stahování Limewire. V menu GRUBu [4] je

také doplněn memtest, neboli aplikace k testování operační paměti. Součástí je také nové [Wine](#) [5] pro spouštění windows aplikací. [Stahovat](#) [6] můžete ze SourceForge.

Elive

Elive je na Debianu založená distribuce a LiveCD, která obsahuje desktop [Enlightenment](#) [7] ve verzi 16 pro slabší počítače a ve verzi 17 jako CVS build k vyzkoušení. Doslova promění váš starší počítač v plnohodnotné pracovní prostředí pro běžnou kancelářskou práci. Dále obsahuje program Elpanel pro snadnou konfiguraci celého systému. Instalátor nově obsahuje automatický, manuální a pokročilý režim. Opraveno bylo také mnoho nahlášených bugů. [Stahovat](#) [8] můžete na stránkách projektu a zrcadlech.

Puppy Linux

Vyšla nová verze distribuce Puppy Linux 2.11, kterou jsem recenzoval ve [třetím dílu distribučních novinek](#) [9]. Jak možná víte, tak minulá verze obsahovala souborový systém komprimovaný pomocí algoritmu LZMA, který měl bohužel jako vedlejší důsledek až dvojnásobnou dobu startu všech aplikací. Sice to moc nevádí na nových počítačích, ale na starších, kde má Puppy Linux také své místo, to bylo velice citelné, a tak se nová verze vrátila zpět k GZIP. Dále bylo opraveno mnoho chyb. K dispozici jsou dva ISO obrazy, z nichž jeden je obohacen o některé X.org ovladače. Oba si můžete [stáhnout](#) [10] z FTP serveru.

DeLi

DeLi je distribuce určená k použití na starém hardwaru, o čemž svědčí přítomnost podpory pro rozhraní PS/2 a EISA. Cílovou skupinou této distribuce jsou tak počítače s procesorem i386 až Pentium I osazené 8 – 32 MB operační paměti. V její nové verzi je přidán program [deliget](#), který slouží ke stahování a instalování distribučních balíčků. Mezi novými balíčky je také [AbiWord](#) [11] a [XMMS](#) [12]. ISO obraz o velikosti 125 MB si můžete [stáhnout](#) [13] z Berlios.

FreeSBIE

Vyšla nová PRE-BETA verze LiveCD distribuce FreeSBIE 2.0, což je FreeBSD s desktopovým prostředím Xfce [14]. V nové verzi je odstraněn správce Fluxbox [15], který byl přidán teprve v minulé verzi. Nově také přináší možnost zadávat „cheatcodes“ před bootem, což není nic jiného než předávání parametrů jádru.

Distribuční rada

Používáte distribuci Gentoo a aby to nebylo málo, tak máte rádi up-to-date svn/cvs/darcs programy? Pak pro vás mám radu, která vám ulehčí jejich aktualizaci. Za tip děkuji Davidovi Watzkemu [16]. Budete potřebovat jen balíčky *app-portage/portage-utils* a *sys-apps/gawk*, které nainstalujete příkazem

```
emerge -avu app-portage/portage-utils \
sys-apps/gawk
```

Aktualizaci pak provedete pomocí příkazu:

```
emerge -av $( qlist -IC 9999 '\-svn$' )
```

These are the packages that would be merged, in order:

Calculating dependencies... done!

```
[ebuild R ] app-misc/youtube-dl-9999 9 kB [1]
```

```
[ebuild R ] games-fps/quake3-9999 USE="opengl -dedicated -teamarena" 0 kB
```

```
[ebuild R ] media-gfx/graphviz-9999 USE="X cairo -minimal -static -tcltk" 0 kB [1]
```

```
[ebuild R ] media-libs/imlib2-9999 USE="X bzip2 gif jpeg mp3 nls png tiff
zlib -doc (-mmx)" 0 kB
```

```
[ebuild R ] media-sound/amarok-9999 USE="aac kde opengl -arts -debug -ifp
-ipod -mysql -njb -noamazon -postgres (-real) -visualization -xinerama -xmms"
0 kB [1]
```

```
[ebuild R ] media-video/mplayer-9999-r7 USE="X aac alsa cdparanoia
custom-cflags dga dts dvd dvdread encode fbcon gif gtk jpeg mad nls nvidia
opengl png sdl theora truetype vorbis x264 xv xvid (-3dfx) (-3dnow)
(-3dnowext) -aalib (-altivec) -arts -bidi -bl -cpudetection -debug -directfb
-doc -dv -dvb -dvdnav -esd -ggi -i8x0 -ipv6 -jack -joystick -libcaca -lirc
-live -lzo -matrox (-mmx) (-mmxext) -musepack -nas -oss (-real) -rtc -samba
(-sse) (-sse2) (-svga) -tga -v4l -v4l2 (-win32codecs) -xanim -xinerama -xmms
-xvmc" 0 kB [3]
```

```
[ebuild R ] net-p2p/linuxdcpp-9999 USE="-debug" 0 kB [1]
```

```
[ebuild R ] net-p2p/mlndonkey-9999-r1 USE="-allnetworks -batch -doc -gd
-gtk -guionly" 0 kB [1]
```

```
[ebuild R ] x11-themes/gtk-engines-qt-9999 USE="-arts -debug -xinerama" 0 kB [1]
```

Pokud chcete mít každé ráno čerstvé noční sestavení vašich oblíbených programů, pak umístěte tento kód do cronu pomocí příkazu `crontab` na libovolný čas, který vám vyhovuje, avšak bez parametru `-av`:

```
emerge $( qlist -IC 9999 'svn$' )
```

■

Odkazy

[1] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/enlightenment>

[2] <http://www.abclinuxu.cz/zpravicky/vydani-fedora-core-6-znovu-opozdono>

[3] <http://www.ixsystems.com/>

[4] <http://www.abclinuxu.cz/software/system/boot/grub>

[5] <http://www.abclinuxu.cz/software/system/emulatory/wine>

[6] http://www.litrixlinux.org/site1/index.php?option=com_weblinks&catid=18&Itemid=23

[7] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/enlightenment>

[8] <ftp://ftp.nluug.nl/pub/os/Linux/distr/live/isos/>

[9] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-2>

- [10] <ftp://ftp.nluug.nl/mirror/ibiblio/distributions/puppylinux/>
- [11] <http://www.abclinuxu.cz/software/kancelar/textove-procesory/abiword>
- [12] <http://www.abclinuxu.cz/software/multimedia/audio/prehravace/xmms>
- [13] <ftp://download.berlios.de/pub/delilinux/iso/>
- [14] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/xfce>
- [15] <http://www.abclinuxu.cz/software/pracovni-prostredi/spravci-oken/fluxbox>
- [16] <http://www.abclinuxu.cz/Profile/8258>

Distribuční novinky – 6

Nové verze oblíbených velkých distribucí Fedora Core 6 a Ubuntu 6.10 (Edgy Eft). Fedora také vydala několik různých live CD/DVD. Kromě toho se chystají nové verze VectorLinux a Astaro Security Gateway. Jak nainstalovat balíčky pro RedHat v Debianu.

Marek Stopka

Letem světem

Vydání distra Fedora Core 6 bylo [posunuto](#) [1] z 16. října na 24. října. Tentokrát kvůli chybám v instalátoru, X serveru a některým závislostem na architekturách x86_64 a PPC. Byl uvolněn již 2. release candidate distribuce Damn Small Linux – můžete tedy testovat. K dispozici je nová vývojová verze distribuce 64Studio, která je založena na 64bitovém Debianu. Obsahuje programy pro tvorbu digitálního obsahu a nese číslo 0.9.5.

Velké ryby

Ubuntu, Kubuntu, Edubuntu 6.10 Edgy Eft

Vývojáři distribuce Ubuntu [uvolnili](#) [2] verzi 6.10 – Edgy Eft [3]. K dispozici jsou instalovatelné live CD, verze pro servery, alternativní textový instalátor a

průvodce aktualizací pro uživatele stávající stabilní verze. Finální verze vyšla podle plánu 26. října a bude podporována po dobu 18 měsíců. Součástí je také funkce skrývání „nepotřebných“ adresářů v kořenovém adresáři souborového systému, která se setkává s rozporuplnými reakcemi uživatelů. Více se dočtete v blogu [Kubuntu Edgy – skrývání podadresářů v adresáři](#) / [4]. Jaké další novinky tato distribuce obsahuje, to se dozvíte v [Distribučních novinkách 2](#) [5]. [Stahovat](#) [6] můžete z webu Ubuntu.

Fedora Core 6

Po všech odkladech byla 24. října [vydána](#) [7] stabilní verze distribuce Fedora Core. O novinkách v nové verzi se dočtete v článku [Fedora Core 6](#) [8]. [Stáhnout](#) [9] novou verzi můžete třeba z ftp zrcadel.

Tichá voda

B2D

Distrowatch informuje [10], že vývojáři tchaj-wanské distribuce B2D vydali novou pre-stable verzi. Systém je založen na Debian GNU/Linuxu. Změnilo se například pojmenování X serveru (xserver-xorg), dále byl zahrnut balíček pro podporu NTFS ntfs-3g z Ubuntu, byly aktualizovány překlady KDE [11] do tradiční čínštiny, přibyla utilita firestarter (grafická konfigurace firewallu) a podpora pro čínské znaky v KPDF [12]. Stahovat [13] můžete z ftp serveru tvůrců.

Astaro Security Gateway

Společnost Astaro [14], výrobce na Linuxu založených firewallů, bran (gateway) a integrovaných bezpečnostních řešení, vydala veřejnou betaverzi distribuce Astaro Security Gateway 7.0. K dispozici jsou jak ISO obrazy, tak obrazy pro VMware [15]. Nabízí VPN, OSPF, FTP proxy, funkce pro kontrolu komunikačních a p2p sítí a spoustu dalších vlastností. Vše se konfiguruje pomocí povedeného webového rozhraní, které je zobrazeno na screenshotech. Stáhnout [16] si ji můžete z ftp serveru. Zdroj: Distrowatch [17].

Fedora Core 6 Live-Spin

Spolu s běžnou verzí Fedory vyšla i sada live CD a DVD pro architektury i386 a x86_64. CD jsou dělena do tří skupin: první obsahuje GNOME [18], druhé KDE [19] a třetí je určeno pro server, takže obsahuje Apache [20], MySQL [21], PHP a další programy s konfiguračními aplikacemi. K dispozici je také DVD, které obsahuje GNOME [22], KDE [23], Xfce [24] a spoustu programů z repozitáře „extras“. Torrenty jsou k dispozici [25] na webu fedoraunity.com [26]. Zdroj: Distrowatch [27].

VectorLinux

Byl vydán první RC (release candidate) distribuce VectorLinux 5.8, která vychází ze Slackware. O verzi 5 si můžete přečíst v článku Vector Linux 5.0 SOHO [28] od Vlastimila Otta [29]. Nově obsahuje ntfs-3g pro podporu čtení i zápisu z/na diskové oddíly NTFS. Kromě toho poslední verze CUPS, AbiWord [30], Gnumeric [31], Xfce [32] a ovladačů. Stahovat [33] můžete z ftp serveru. Zdroj: Distrowatch [34].

Distribuční rada

Je váš oblíbený program k dispozici jen jako balíček pro jinou distribuci a vy používáte systém založený na Debianu? Pak by se vám mohl líbit program alien. Slouží k jednoduchému převodu .rpm (Red Hat) nebo .tgz (Slackware) balíčků na .deb. Program nainstalujete příkazem `apt-get install alien`. Je velice jednoduchý a pro základní použití stačí jediný argument – jméno převáděného balíčku. Třeba takto:

```
alien eagle-lin-eng-4.16r1-1.i586.rpm
Warning: Skipping conversion of scripts in \
package eagle-lin-eng: postinst perm
Warning: Use the --scripts parameter to \
include the scripts.
eagle-lin-eng_4.16r1-2_i386.deb generated
```

Vytvořený balíček pak už nainstalujete pomocí programu `dpkg` s parametrem `-i` (Distribuční novinky – 3 [35]). Výstup vypadá následovně:

```
# dpkg -i eagle-lin-eng_4.16r1-2_i386.deb
Instaluji balík eagle-lin-eng.
(Čtu databázi ... nyní je nainstalováno \
154008 souborů a~adresářů.)
Rozbaluji eagle-lin-eng \
(z eagle-lin-eng_4.16r1-2_i386.deb) ...
Nastavuji balík eagle-lin-eng (4.16r1-2) ...
```

Převod pomocí tohoto programu nemusí být vždy úplně čistý, a mohou vám tak chybět nějaké závislosti, které budete muset vyřešit ručně.

Odkazy

- [1] <http://www.abclinuxu.cz/zpravicky/vydani-fc6-opet-posunuto>
- [2] <http://www.abclinuxu.cz/zpravicky/vydano-ubuntu-6.10>
- [3] <http://www.abclinuxu.cz/clanky/recenze/kubuntu-6.10-edgy-eft-instalace-ve-dvojim-stylu>
- [4] <http://www.abclinuxu.cz/blog/coder/2006/10/23/154988>
- [5] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-2>
- [6] <http://www.ubuntu.com/products/GetUbuntu/download?action=show&redirect=download>
- [7] <http://www.abclinuxu.cz/zpravicky/fedora-core-6-je-tady>
- [8] <http://www.abclinuxu.cz/clanky/recenze/fedora-core-6>
- [9] <http://sunsite.mff.cuni.cz/pub/fedora/6>
- [10] <http://distrowatch.com/?newsid=03792#0>
- [11] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [12] <http://www.abclinuxu.cz/software/soubory/pdf-postscript/kpdf>
- [13] <ftp://ftp3.tnc.edu.tw/KNOPPIX/health/2006/B2DpureKGB20061019.iso>

- [14] <http://astaro.de/>
- [15] <http://www.abclinuxu.cz/software/system/virtualizace/vmware-server>
- [16] ftp://ftp.astaro.de/pub/ASL/v7.0/iso_i386/
- [17] <http://distrowatch.com/?newsid=03793>
- [18] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>
- [19] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [20] <http://www.abclinuxu.cz/software/server/http/apache>
- [21] <http://www.abclinuxu.cz/software/server/database/mysql>
- [22] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>
- [23] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [24] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/xfce>
- [25] <http://torrent.fedoraunity.org/unity/>
- [26] <http://www.fedoraunity.org>
- [27] <http://distrowatch.com/?newsid=03800>
- [28] <http://www.abclinuxu.cz/clanky/recenze/vector-linux-5.0-soho>
- [29] <http://www.abclinuxu.cz/Profile/2105>
- [30] <http://www.abclinuxu.cz/software/kancelar/textove-procesory/abiword>
- [31] <http://www.abclinuxu.cz/software/kancelar/tabulkove-kalkulatory/gnumeric>
- [32] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/xfce>
- [33] <http://ftp-osl.osuosl.org/pub/vectorlinux/veclinux-current/iso-release/>
- [34] <http://distrowatch.com/?newsid=03802>
- [35] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-3>

Fedora Core 6

Blíží se vydání nové Fedora Core, tentokrát už s číslem 6, takže je na čase vyzkoušet novinky poslední testovací verze před vydáním stabilní. Přečtěte si o instalaci a prvních dojmech z používání.

Filip Bartmann

Instalace

Nejprve si ukážeme, jaké novinky nám přináší instalátor Fedory. Hned nás nejspíše zaujme nové grafické téma nazvané Fedora DNA. Protože se jedná o testovací verzi, tak jsme na to hned v prvním kroku upozorněni. Následuje výběr jazyka, samozřejmě si vybereme češtinu. Klávesnici si můžeme vybrat jak qwerty, tak qwertz. Pokud si pozorněji prohlédneme okno instalátoru, tak zjistíme, že z levého kraje zmizely podrobnější instrukce k instalaci.

Dále na nás čeká ten nejdůležitější krok, a to rozdělení disku na oddíly. Zde přibyl tlačítko **Pokročilé nastavení ukládání dat**, které zobrazí dialogové okno s možností **Přidat iSCSI target** a **Zakázat zařízení dmraid**. Jinak můžeme nechat instalátor vymazat všechny linuxové oddíly, celý disk, použít jenom volné místo anebo si oddíly nadefinovat sami. Pokud si oddíly vytváříme sami a nastavíme velikost swapu na hodnotu menší, než je velikost RAM paměti v počítači, tak se vypíše varování, abychom to napravili. Jako užitečné rovněž hodnotím, že před definitivním ukončením rozdělování disku se zobrazí seznam formátovaných oddílů.

Zařízení	Místo připojení/ RAID/svazek	Typ	Formátovat	Velikost (MB)	Začátek	Konec
/dev/hda7		Linux native		23839	21565	24603
/dev/hda8		Linux swap		478	24604	24664
/dev/hda9	/	ext3	✓	23839	24665	27703
/dev/hda10		swap	✓	478	27704	27764
Volné		Nevyužito		68378	27765	36481

Následuje nastavení zavaděče systému, kterým je u Fedory tradičně GRUB [1]. Zde se již delší dobu nic nezměnilo. Můžeme si vybrat pouze, jestli chceme zavaděč instalovat nebo ne. Dále **Přidat/Upravit/Odstranit** položky zavaděče a **Použít heslo pro zavaděč sys-**

tému, aby nikdo jiný mimo administrátora systému nemohl nastavení GRUBu měnit. Hodilo by se ještě zaheslování jednotlivých položek. Pokud zaškrtneme **Nastavit pokročilé volby zavaděče**, budeme si v následujícím kroku moci vybrat, kam chceme GRUB nainstalovat (do hlavního zaváděcího sektoru nebo do **Prvního sektoru zaváděcího oddílu**). Také můžeme použít **LBA32** a předat jádru nějaké další parametry. Pokud se častěji přepínáme do konzole, tak zde můžeme připsat parametry **vga=**.

Po kliknutí na **Další** se dostaneme ke konfiguraci síťové karty, kde narazíme na další novinku instalátoru – na možnost nastavení IPv6 adresy. Následuje výběr časové zóny, která je, pokud jsme si na začátku instalace vybrali češtinu, správně nastavena. Pak stačí určit, jestli jsou hodiny BIOSu nastavené na UTC, což je, v případě, že máme na počítači nainstalovaný jenom Linux, doporučováno. Ještě nastavení hesla pro superuživatele roota a již se řítíme do finále instalace – výběru balíčků.

Nejprve si můžeme vybrat pouze základní okruh balíčků **Kancelář a produktivita**, **Vývoj softwaru** a **Web server**. Hned pod tím se nachází vylepšení instalátoru a to možnost přidání dalších repozitářů se sw. Implicitně je zde úložiště **Fedora Extras**. V případě, že nám základní okruh balíčků nevyhovuje, tak si vybereme **Upravit teď** a budeme si moci balíčky, které chceme nainstalovat, vybrat podrobněji. To je nut-

ností v případě, že chcete používat grafické prostředí KDE [2], protože, ve výchozím stavu se nainstaluje pouze GNOME [3]. Pro podrobnější nastavení balíčků slouží tlačítko **Volitelné balíčky**.



Po zkontrolování závislostí se zobrazí poslední doplňující informace o instalaci a stisknutím **Další** začíná formátování oddílů a instalace vybraných balíčků. Tato část závisí pouze na rychlosti našeho počítače a počtu vybraných balíčků. Pokud instalace proběhla úspěšně, můžeme počítač restartovat stisknutím tlačítka **Restartovat**.

První start

Po restartu se zobrazí grafické bootování systému a Průvodce prvním nastavením, kde nás čekají poslední kroky, před startem. Potvrdíme zde obligátní licenční smlouvu, upravíme nastavení firewallu, SELinux, datumu a času (včetně použití NTP). Rozhodně nezapomeneme vytvořit obyčejný uživatelský účet, pod kterým budeme pracovat, a nakonec si ještě zkontrolujeme, jestli funguje zvuková karta.



Fedora jako pracovní stanice

Pro běžnou kancelářskou práci máme v základní instalaci k dispozici dvě nejrozšířenější grafická prostředí KDE 3.5.4 a GNOME v nejnovější verzi 2.16.

GNOME

Mezi horkou novinku jistě patří zařazení správce oken Compiz, který společně s akcelerovaným X serverem umožňuje mít plochy umístěné na krychli, opravdovou průhlednost oken, třepání oken při přesouvání a další. Toto můžeme zapnout v menu **Systém/Nastavení/Další nastavení/Desktop Effects**. Rovněž je k dispozici nový editor menu, který je již skutečně použitelný.

V **Nautilu** [4] byl upraven dialog pro změny oprávnění, kde byla přidána také položka, která zobrazuje kontext SELinuxu. Což v Nautilu si můžeme zapnout i jako sloupec při prohlížení souborů. Další vylepšení se týká přesouvání oken mezi plochami, protože stačí chytit danou aplikaci na pruhu úloh nebo v náhledu ploch a přesunout ji na požadovanou plochu.

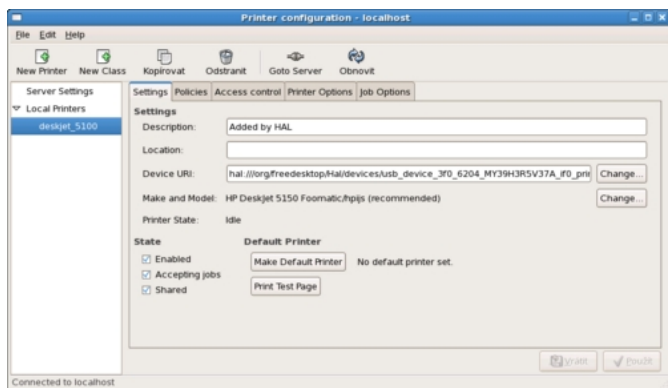
Mezi další novinky v tomto prostředí patří program **baobab** pro grafické zobrazení obsazeného místa na disku. Správce napájení také nabízí rozšířené možnosti. Do dialogu **Nastavení zvuku** přibyla záložka pro konfiguraci zvukových zařízení. Pro jednodušší prohledávání počítače byl přidán **Deskbar Applet**, který lze propojit i s **Beagle**m, čímž se jeho možnosti značně rozšiřují. Toto grafické prostředí je postaveno na knihovně **glib 2.12.3** a **GTK+ [5] 2.10.3**.

Co se týče programů specifických pro Fedoru, tak i zde se udály nějaké změny. Úplného přepracování se dočkal dialog pro nastavení tiskáren, který je založený na nejnovějším systému **CUPS** – nyní využívá více z jeho možností. Přibyl program **setrouble-shoot** pro pohodlnější řešení problémů s SELinuxem. V této testovací verzi ale ještě nefunguje dobře a po otevření logu se zasekne. Další nový program slouží k práci s virtualizačním systémem **Xen** – jmenuje se **Virtual Machine Manager**. S jeho pomocí lze nyní jednoduše vytvářet nebo rušit virtuální systémy. Co se týče výchozího správce balíčků, tak ten stále nebyl od poslední verze ve **FC5** opraven, aby při stahování a instalaci nevypadal zatuhnutě. Také byla přidána podpora pro šifrování swap a nekořenových oddílů.

Ostatní programy

Co se týče ostatních programů ve Fedoře, tak musím nejprve zmínit KDE, které je zde obsažené v nejnovější verzi 3.5.4 – žádné převratné novinky oproti

FC5 nepřináší. Dokonce ani podporu pro Compiz, což je škoda. Ale je to dáno tím, že hlavní desktopové prostředí je GNOME a tam se soustředí hlavní zájem vývojářů Fedory. Jako základ pro KDE byla použita knihovna Qt [6] 3.3.6. Nachází se zde také Qt4 a to ve verzi 4.2.0-rc1. OpenOffice.org [7] nese číslo verze 2.0.4, KOffice [8] 1.5.2. Firefox [9] je bohužel pouze ve verzi 1.5, i když díky tomu, že Firefox 2.0 již spěje do finále, tak by zde mohl být ten. Grafické prostředí je zajišťováno pomocí X.org ve verzi 7.1.



Multimédia

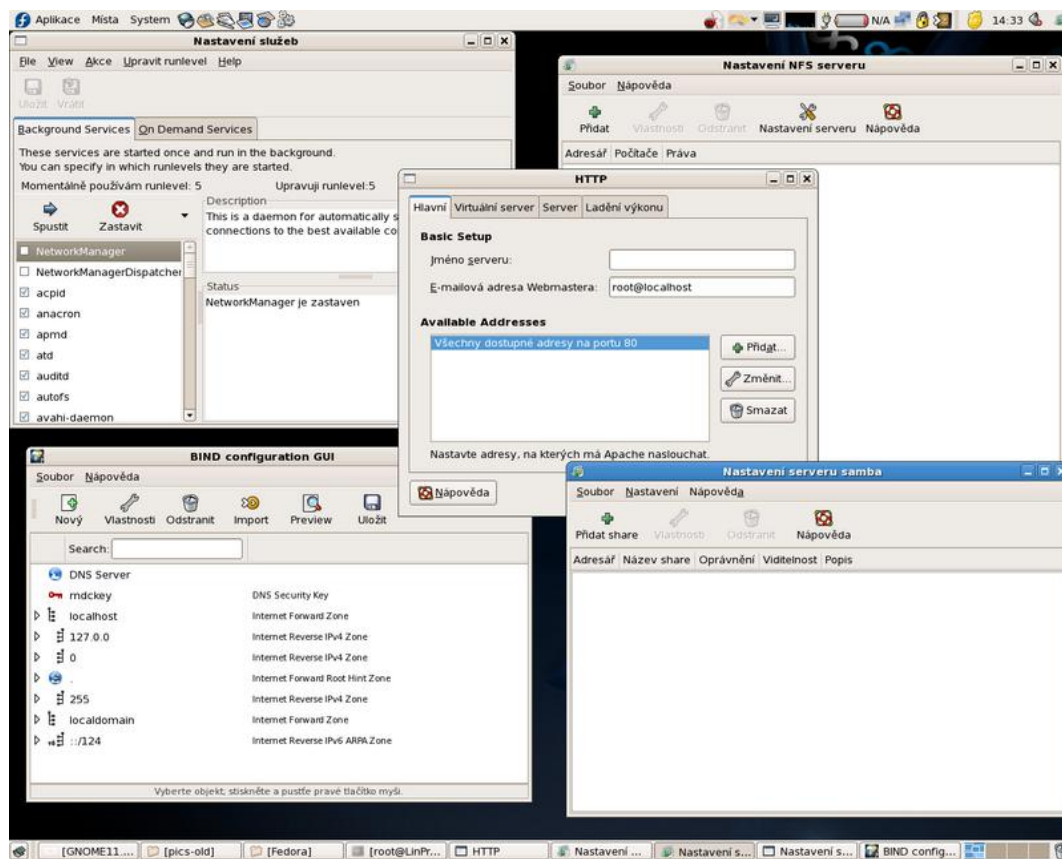
Pro přehrávání multimédií je k dispozici Totem [10] a Rhythmbox [11], které se instalují jako výchozí.

Jako další se dá doinstalovat XMMS [12] nebo Amarok [13]. Nejdůležitější linuxové multimediální programy jako MPlayer [14] nebo Xine [15] však scházejí a je nutné je doinstalovat s pomocí dalších repozitářů. Pro nahrávání obrazovky jsou k dispozici dva programy – to Istanbul a Byzanz. Kvůli patentovým omezením chybí podpora pro oblíbený formát MP3.

Média CD a DVD můžeme vypalovat pomocí programů GnomeBaker [16], Graveman či přímo v Nautilu v GNOME. Pro KDE je k dispozici tradiční K3b [17].

Fedora jako server

Pokud byste chtěli použít Fedoru jako server, tak je k dispozici Apache [18] 2.2.3, PHP 5.1.6 a MySQL [19] 5.0.22. Samozřejmě nechybí ani grafické konfigurační nástroje, které zajistí základní správu serveru. Pokud preferujete PostgreSQL [20], tak ten je zde ve verzi 8.1.4. Přepravu mailů ve výchozí konfiguraci zajišťuje Sendmail 8.13.8. Jako alternativu lze použít mnohem lépe konfigurovatelný Postfix [21] 2.3.3 nebo Exim 4.63. Správa LDAP je zajišťována pomocí OpenLDAP 2.3.27. Virtualizaci zajišťuje Xen [22] 3.0.2.



Plocha v FC6

Vývojové programy

Pro vývoj je k dispozici Eclipse [23] 3.2.0 společně s pluginy pro programování v C/C++ a přístup k CVS [24]/SVN [25]. Neškodilo by přidání pluginu [26] pro vývoj pravidel SELinuxu. Mimo Eclipse nechybí samozřejmě ani KDevelop [27] pro programování v nejběžnějších jazycích. Programování orientované na web zastane Quanta [28].

O úroveň níže se nachází kompilátory. Pro jazyky C [29]/C++ je to samozřejmě gcc 4.1.1. Javu zajišťuje gcj. Samozřejmostí je také Python [30] 2.4.3 a Perl [31]. Z exotičtějších jazyků můžeme nalézt třeba Ruby.

Závěr

Jak tedy vypadá závěrečné zhodnocení nové verze Fedory 6? Vyplatí se aktualizace, pokud již máme nainstalovanou Fedoru 5? Rozhodně si myslím, že ano, protože novinek je více než dost. Namátkou třeba WM Compiz nebo přidání programu LinEAK pro zprovoznění multimediálních klávesnic, vylepšená práce se SELinuxem a nové GNOME.

Pokud používáte jinou linuxovou distribuci a vyhovuje vám, tak asi není důvod měnit, protože Fedora, na rozdíl od jiných distribucí, nenabízí nic, co by jinde nebylo (až na zabezpečení SELinuxem).

■

Odkazy

- [1] <http://www.abclinuxu.cz/software/system/boot/grub>
- [2] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [3] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>
- [4] <http://www.abclinuxu.cz/software/soubory/sprava/nautilus>
- [5] <http://www.abclinuxu.cz/software/programovani/knihovny/gtkp>
- [6] <http://www.abclinuxu.cz/software/programovani/knihovny/qt>
- [7] <http://www.abclinuxu.cz/software/kancelar/baliky/openoffice.org>
- [8] <http://www.abclinuxu.cz/software/kancelar/baliky/koffice>
- [9] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>
- [10] <http://www.abclinuxu.cz/software/multimedia/video/prehravace/totem>
- [11] <http://www.abclinuxu.cz/software/multimedia/audio/prehravace/rhythmbox>
- [12] <http://www.abclinuxu.cz/software/multimedia/audio/prehravace/xmms>
- [13] <http://www.abclinuxu.cz/software/multimedia/audio/prehravace/amarok>
- [14] <http://www.abclinuxu.cz/software/multimedia/video/prehravace/mplayer>
- [15] <http://www.abclinuxu.cz/software/multimedia/video/prehravace/xine>
- [16] <http://www.abclinuxu.cz/software/soubory/vypalovani/gnomebaker>
- [17] <http://www.abclinuxu.cz/software/soubory/vypalovani/k3b>
- [18] <http://www.abclinuxu.cz/software/server/http/apache>
- [19] <http://www.abclinuxu.cz/software/server/databaze/mysql>
- [20] <http://www.abclinuxu.cz/software/server/databaze/postgresql>
- [21] <http://www.abclinuxu.cz/software/server/posta/postfix>
- [22] <http://www.abclinuxu.cz/software/system/virtualizace/xen>
- [23] <http://www.abclinuxu.cz/software/programovani/ide/eclipse>
- [24] <http://www.abclinuxu.cz/software/programovani/sprava-verzi/cvs>
- [25] <http://www.abclinuxu.cz/software/programovani/sprava-verzi/subversion>
- [26] <http://oss.tresys.com/projects/slide/wiki/WikiStart>
- [27] <http://www.abclinuxu.cz/software/programovani/ide/kdevelop>
- [28] <http://www.abclinuxu.cz/software/programovani/ide/quanta-plus>
- [29] <http://www.abclinuxu.cz/software/programovani/jazyky/c>
- [30] <http://www.abclinuxu.cz/software/programovani/jazyky/python>
- [31] <http://www.abclinuxu.cz/software/programovani/jazyky/perl>

GPS a komunikační protokol NMEA – 3 (dekódování dat)

Všimněme si blíže významu jednotlivých vět, které získáváme z GPS přijímače. Popisovaný přístroj má ve svých parametrech uvedeno, že komunikace počítačem probíhá prostřednictvím protokolu NMEA0183. Protokol vytvořila asociace NMEA (National Marine Electronics Association).

Jan Martinek

Dekódování přijatých dat

Asociace NMEA na své stránce [1] uvádí, že komunikační standard je dokument podléhající copyrightu a lze jej pouze zakoupit od asociace NMEA. Cena představuje řádově stovky dolarů. Ostatní zdroje na internetu prý nejsou autorizované a mohou představovat porušení copyrightu. NMEA dále uvádí, že obsah rozličných stránek s touto tematikou mnohdy obsahuje zastaralé informace.

Pod klíčovým slovem NMEA 0183 lze na internetu skutečně nalézt řadu odkazů a jejich obsah se jeví být konzistentním. Na základě popisů od různých autorů lze získat smysluplné informace z GPS přístroje, ale bohužel nemohu posoudit, zda se jedná o korektní popis standardu NMEA 0183. Dokument popisující standard jsem nezakoupil a i kdybych tak učinil, nesměl bych rozšiřovat informace v něm obsažené. Pokusím se však shrnout poznatky, které jsou dostupné na internetu. Aby nedošlo k nedorozumění, nebudu již nadále komunikační protokol označovat NMEA 0183.

Protokol, jímž komunikuje přijímač Navilock 202U, má přenosovou rychlost (baud rate) 4800, počet datových bitů je 8, přičemž sedmý bit (MSB) je vždy nulový. Počet stop bitů je jeden nebo více, parita není žádná. Navzájem spolu komunikuje vždy jeden mluvčí (talker) a jeden nebo více posluchačů (listeners). Veškerá data jsou posílána ve formě vět (sentences). Jsou dovoleny pouze tisknutelné ASCII znaky plus znaky konce řádku, tedy <CR> a <LF> (0x0d, 0x0a hexadecimálně). Každá věta začíná znakem \$ (dolar) a končí sekvencí <CR><LF>. Existují tři základní druhy vět:

- věty ze strany mluvčího (talker sentences)
- proprietární věty (proprietary sentences)
- dotazovací věty (query sentences)

Obecný formát vět ze strany mluvčího je

```
$ttsss,d1,d2,...<CR><LF>
```

První dvě písmena, která následují po znaku dolar, jsou označena `tt` a představují identifikátor mluvčího (talker identifier). Další tři písmena (`sss`) jsou identifikátor věty (sentence identifier). Následují datové položky oddělené čárkami (znak `,`). Po nich následuje nepovinný kontrolní součet. Věta je ukončena znaky <CR><LF>. Význam jednotlivých datových položek je jednoznačně definován pro konkrétní typ věty (ten je určen identifikátorem `sss`). Jestliže určitá datová položka není k dispozici, zůstane datové pole prázdné, ale čárky oddělující datová pole zůstávají (bez mezery). Kontrolní součet začíná znakem hvězdička (`,*`) a za ní jsou dvě hexadecimální číslice představující logickou operaci XOR (exclusive OR) ze všech znaků mezi `,$` a `,*`. Samotný dolar a hvězdička se do kontrolního součtu nezapočítávají. Každá věta může obsahovat nejvýše 80 znaků plus `,$` a <CR><LF>, celkem tedy 83 bajtů.

Věty proprietární umožňují výrobcům nadefinovat vlastní větu. Tyto věty začínají sekvencí `,$P`, pak následuje třípísmenný identifikátor výrobce, a dále následují jednotlivé datové položky v souladu s přáním výrobce. Obecný formát věty musí být zachován.

Dotazovací věty představují způsob, kterým může posluchač požádat mluvčího o zaslání konkrétní věty. Obecný formát je `$tt11Q,sss<CR><LF>`.

První dva znaky (`tt`) za znakem dolar jsou identifikátorem toho, kdo podává žádost. Následující dva znaky (`11`) označují dotazovaného – tedy toho, komu je žádost posílána. Pátým znakem je vždy písmeno `,Q`, které označuje, že se jedná o dotazovací typ věty. Následuje třípísmenná datová položka (`sss`) určující, o jaký typ věty se žádá. Příkladem dotazovací věty může být následující sekvence:

```
$CCGPQ,GGA<CR><LF>
```


V této větě písmena **CC** označují počítač, který žádá přístroj **GP** (tedy GPS přijímač), aby zasílal věty typu **GGA**. Po této dotazovací větě by měl GPS přijímač zasílat každou sekundu větu typu **GGA**, dokud nedostane povel k zasílání jiného typu věty. Dvou-písmenných identifikátorů existuje mnoho, pro nás je však v této chvíli nejdůležitější, že pro GPS přijímače se používá identifikátor **GP**. Existuje nepřehledné množství různých vět, avšak přístroj NAVI-LOCK 202U (a řada jiných) používá pouze čtyři, které jsou uvedeny v následujících tabulkách.

Příklad:

```
$GPGSA,A,3,29,26,22,09,07,05,\
04,,,,,,,,,1.7,1.0,1.4*30
```

RMC (Recommended Minimum Navigation Information)

Minimální doporučená informace pro navigaci

Příklad:

```
$GPRMC,170138.615,A,4912.2525,N,\
01635.0378,E,0.04,16.43,280705,,*32
```

GSA, aktivní satelity a DOP (Dilution Of Precision)

#	formát	příklad	komentář
1	c	A	Přepínání mezi N-rozměrnými módy (A – automatické, M – manuální)
2	d	3	Počet dimenzí N (1=?, 2=2D, 3=3D)
3	dd	29	ID prvního satelitu použitelného pro výpočet
4	dd	26	ID druhého satelitu použitelného pro výpočet
5	dd	22	ID třetího satelitu použitelného pro výpočet
6	dd	09	ID čtvrtého satelitu použitelného pro výpočet
7	dd	07	ID pátého satelitu použitelného pro výpočet
8	dd	05	ID šestého satelitu použitelného pro výpočet
9	dd	04	ID sedmého satelitu použitelného pro výpočet
10	dd	N.A.	ID osmého satelitu použitelného pro výpočet
11	dd	N.A.	ID devátého satelitu použitelného pro výpočet
12	dd	N.A.	ID desátého satelitu použitelného pro výpočet
13	dd	N.A.	ID jedenáctého satelitu použitelného pro výpočet
14	dd	N.A.	ID dvanáctého satelitu použitelného pro výpočet
15	d.d	1.7	PDOP (Position Dilution Of Precision) v metrech
16	d.d	1.0	HDOP (Horizontal Dilution Of Precision) v metrech
17	d.d	1.4	VDOP (Vertical Dilution Of Precision) v metrech
18	*xx	30	Kontrolní součet

#	formát	příklad	komentář
1	hhmmss.sss	170138.615	Čas (UTC)
2	c	A	Status (A=OK, V=varování)
3	ddmm.mmmm	4912.2525	Zeměpisná šířka
4	c	N	Indikátor sever/jih (N=sever, S=jih)
5	ddmm.mmmm	01635.0378	Zeměpisná délka
6	c	E	Indikátor východ/západ (E=východ, W=západ)
7	d.d	0.04	Vodorovná rychlost (Speed Over Ground, v uzlech)
8	d.d	16.43	Kurz pohybu ve stupních
9	ddmmyy	280705	Datum ddmmyy
10	d.d	N.A.	Magnetická deklinace ve stupních
11	c	N.A.	Indikátor východ/západ (E=východ, W=západ)
12	*xx	32	Kontrolní součet

GSV (Satellites in View)

Informace o družicích Množství údajů závisí na počtu viditelných družic. Jedna věta může obsahovat nejvýše 80 znaků, což vystačí pouze k uložení dat týkajících se nejvýše čtyř družic. Informace proto bývá rozdělena do několika dílčích vět. Příklad (trojice vět):

```
$GPGSV,3,1,11,09,84,297,41,05,48,256,45,07,\ \
38,059,41,26,22,178,41*74
$GPGSV,3,2,11,24,13,063,00,14,12,324,00,30,\ \
12,251,00,22,12,286,38*78
$GPGSV,3,3,11,29,10,173,35,04,09,105,30,18,\ \
06,254,00*46
```

Poznámka: Příklad v tabulce se vztahuje pouze k první větě.

#	formát	příklad	komentář
1	d	3	Celkový počet vět (číslují se od 1)
2	d	1	Číslo aktuální věty (taktéž se čísluje od 1)
3	dd	11	Počet viditelných družic
4	dd	09	Identifikační číslo družice
5	dd	84	Úhlová výška, kde se daná družice nachází
6	ddd	297	Azimut, kde se daná družice nachází
7	dd	41	Odstup signálu od šumu (SNR – Signal to Noise Ratio). Je-li tento údaj roven nule, nelze daný satelit využít k výpočtu polohy. Nejčastěji proto, že je zastíněn.
...	Podle počtu viditelných družic mohou následovat další čtveřice údajů (4-7)
n	*xx	74	Kontrolní součet

GGA – zeměpisná délka a šířka, geodetická výška, čas určení souřadnic

Příklad:

```
$GPGGA,170139.615,4912.2526,N,01635.0378,E,1,\ \
07,1.0,357.5,M,43.5,M,0.0,0000*7D
```

Souřadné systémy

GPS přijímač potřebuje znát přesnou pozici všech družic, jejichž signál používá k výpočtu vlastní polohy. Každá družice proto vysílá informace o své pozici ve formě souřadnic. Tyto souřadnice jsou vztaženy k souřadnému systému nazývanému ECEF WGS-84 (Earth Centered Earth Fixed, World Geodetic System 1984). Jedná se o pravoúhlý souřadný systém, který je definován na základě elipsoidu, který byl v roce 1984 stanoven referenčním. Jeho tvar je

zvolen tak, aby co nejlépe vystihoval tvar Země. Do geometrického středu tohoto elipsoidu je umístěn počátek souřadné soustavy – bod [0,0,0]. Osa z je totožná s osou rotace, osa x prochází průsečíkem rovníku a Greenwichského poledníku (nultého). Osa y je zvolena tak, aby systém x, y, z byl pravotočivý.

Veškeré výpočty probíhají právě v této souřadné soustavě a výsledná pozice přijímače je taktéž určena v systému ECEF. Výsledek se následně přepočítá do běžnějších zeměpisných souřadnic (zeměpisná délka a šířka) vzhledem k referenčnímu elipsoidu WGS-84. Třetí souřadnicí je výška, ale nikoli nadmořská, nýbrž vzdálenost od povrchu referenčního elipsoidu. Některé GPS přijímače však znají rozdíl mezi povrchem referenčního elipsoidu a místní střední výškou mořské hladiny. Je potřeba si to ale ověřit. Plyne z toho zásadní poznatek – výška, kterou udává GPS přijímač, nemusí souhlasit s výškovými kótami na mapě. V našich podmínkách bývá rozdíl několik desítek metrů a pro danou oblast se příliš nemění, takže stačí pouze přičíst nebo odečíst konstantu. Udávat výšku vzhledem k hladině moře je značně nepraktické a doufejme, že tento způsob co nejdříve skončí v propadlšti dějin. Až do této chvíle je vše relativně přehledné. Uživatel dostane obvyklé zeměpisné souřadnice a výšku.

Problém však nastává s takzvanými lokálními souřadnými systémy, které vznikly v historii a bohužel se drží do dnešní doby. Například v České republice se vyskytují mapová díla nejen v systému WGS-84. Používá se také S-JTSK (souřadnicový systém jednotné trigonometrické sítě katastrální) a S42 (vojenský souřadnicový systém 1942). Systém S42 se používá v zemích bývalé Varšavské smlouvy. Kartografii nerozumím, ale zdá se, že vzájemné převody mezi těmito soustavami jsou obestřeny tajemstvím. Informací je na internetu velmi poskrovnu, jsou nejasné a mnoho odkazů je neplatných. Běžní uživatelé jsou spokojeni, protože existuje řada konverzních programů – s koncovkou .exe a bez zdrojového kódu. Pevně doufám, že se najde sdílný člověk, který rozumí pojmem jako Křovákovo zobrazení, Gauss-Krügerovo zobrazení, Besselův elipsoid, Krassovského elipsoid a poskytne algoritmus, který si pak každý může naprogramovat ve svém oblíbeném programovacím jazyce.

Turistické mapy KČT (Klub českých turistů), které se běžně prodávají v České republice, mívají vyznačenu čtvercovou síť o straně 1 kilometr. Vysvětlivka praví, že souřadnice jsou zavedeny v systému S-

1942 a že prodejce GPS přijímače jistě rád poradí, jak údaje do tohoto systému přepočítat. Pokusím se shrnout, co je potřeba provést, jestliže chceme vědět, kde na mapě se momentálně nacházíme. GPS přijímač, který popisuji v tomto článku, interně provedl výpočty v kartézské soustavě ECEF WGS84. Následně je převedl do úhlových souřadnic (šířka, délka) a výšky.

To je od něj hezké, ale bohužel potřebujeme souřadnice v původní soustavě. Bude nutné je přepočítat. Není to složité, stačí ponásobit nějaké siny a kosiny, přičemž do hry vstoupí parametry elipsy WGS84 (délka hlavní poloosy a excentricita). Protože systém S42 používá elipsoid Krassovského, bude potřeba kartézské souřadnice přepočítat do jiných kartézských souřadnic vztahujících se k tomuto elipsoidu. Souřadnice se navzájem liší a převod se provádí maticovým vztahem, který obsahuje sedm parametrů, které definují převod. Vzniklé souřadnice mu-

síme dále převést na zeměpisné (a výšku), což analyticky nelze provést. Existuje však numerický postup, který poměrně rychle konverguje a po dvou či třech iteracích máme výsledek. Výšku si můžeme někam zapsat – s tou jsme již hotovi. Jenže zbývající dvě souřadnice představují bod na povrchu elipsoidu a musíme zjistit, jak se zobrazí na mapě.

V systému S42 se používá zobrazení Gauss-Krügerovo. Jde o válcové zobrazení a k jeho výpočtu je nutné (mimo jiné) vypočítat délku oblouku na elipse, což představuje vyčíslení eliptického integrálu. Toto se obvykle provádí rozvojem do nekonečné řady. Až je všechna práce hotova, můžeme konečně píchnout prstem do mapy a říct „Tady jsem.“ Zmíněné výpočty jistě budou hračkou i pro toho, kdo bloudí na pokraji smrti hladem a mrazem. Raději se nebudu zmiňovat o obráceném postupu, kdy chceme zjistit, jaké „GPS souřadnice“ má nějaké místo na mapě.

#	formát	příklad	komentář
1	hhmmss.sss	170139.615	Čas (UTC), pro který platí údaje o vypočtené pozici
2	ddmm.mmmm	4912.2526	Zeměpisná šířka
3	c	N	Indikátor severní/jižní šířka (N=sever, S=jih)
4	dddmm.mmmm	01635.0378	Zeměpisná délka
5	c	E	Indikátor východní/západní délky (E=východ, W=západ)
6	d	1	Indikátor kvality: 0 – nebylo možno určit pozici; 1 – pozice úspěšně určena; 2 – pozice úspěšně určena (diferenční GPS)
7	dd	07	Počet viditelných satelitů 00 – 12
8	d.d	1.0	Vliv rozestavení družic na určení polohy HDOP (<i>Horizontal Dilution of precision</i>)
9	d.d	357.5	Výška antény nad elipsoidem
10	c	M	Jednotka pro předchozí údaj (č.9) (M=metr)
11	d.d	43.5	Geoidal separation, rozdíl mezi WGS-84 zemským elipsoidem a střední úrovní moře (geoid). Znaménko mínus znamená, že střední úroveň země je pod elipsoidem.
12	c	M	Jednotka vzdálenosti pro předchozí položku (č.11) (M=metr)
13	d.d	0.0	Stáří poslední aktualizace DGPS. Údaj je uváděn v sekundách. Jestliže údaj chybí, nepoužívá se DGPS.
14	dddd	0000	Identifikační číslo referenční stanice pro DGPS (0000 – 1023)
15	*xx	7D	Kontrolní součet

Shrnutí

GPS je globální systém, který funguje všude na světě, je velmi dobře zdokumentovaný a je zdarma přístupný každému. S tím ovšem velmi kontrastuje chaos panující při reálném použití, protože vývoj probíhá zcela živelně. Mnoho firem vydělává na lidské neschopnosti se domluvit, ujednotit standard a přestat používat zastaralé konvence. Tatáž práce se

provádí neustále znovu a znovu. Chybí volně dostupné mapy, zdrojové kódy k programům, dokumentace k protokolům a datovým formátům a vše se hemží utajováním, licencemi a copyrighty. Naštěstí jsou však i tací, kteří si výsledky své práce nenechávají pro sebe a kterým tímto děkuji.

Poděkování. Tento článek vznikl za podpory grantu GAČR 103/06/1711.

■

Odkazy

[1] <http://www.nmea.org/pub/0183/index.html>

Jazyky a překladače – 5 (syntaxe 3)

Implementace syntaktického analyzátoru není příliš snadná záležitost a konstrukce parsovací tabulky jakbysmet. Proto si představíme dva programy pro generování syntaktických analyzátorů.

Michal Vyskočil

V předcházejícím díle [1] jsme se seznámili se syntaktickou analýzou a s pojmy LR a LL parsery. Dnešní díl bude nadupaný kódem až po kou... koncovou kapitolu, proto asi nepotěším milovníky všeho formálního, jak tomu bylo v minulém díle.

Bison

Stejně jako `flex` je GNU náhrada staršího POSIXového `lex` u, je i `bison` náhradou programu `yacc`. Pro vysvětlení hackerského humoru – `bison` je druh jako a `yacc` znamená *yet another compiler compiler*. Jedná se o generátor LR syntaktických analyzátorů a ve spojitosti s programem `flex` tvoří část (frontendu) překladače.

V minulém díle jsem ukazoval, že překladač jazyka C [2] nepřeloží konstrukci

```
printf("%s\n", n) if (n == 10);
```

Navíc jsem uvedl část gramatiky ve formátu EBNF, která se této části jazyka týká. Napíšeme si tedy jednoduchý analyzátor pro konstrukci `if` v jazyce C. Ovšem je nutné si uvědomit, že syntaxe jazyka C je velice složitá a náš příklad ukazuje pouze malou část (a to ještě ne zcela správně). To je také důvod, proč se v každém příkladu na `yacc/bison`, co jsem viděl, ukazuje gramatika Pascalu. Základem je tedy definiční soubor `cselect.y`, který obsahuje definici gramatiky.

```
%{
#include<stdio.h>
%}

%token IF ELSE
%token IDENTIFIER CONSTANT STRING_LITERAL
%token EQ_OP

%start compound_statement

%%
```

První část obsahuje potřebné definice jazyka C. Dále tu vidíme seznam jednotlivých tokenů (terminálních

symbolů gramatiky). Část `start` označuje startovací (nonterminální) symbol gramatiky. Navíc můžeme označit aritmetické operátory slovem *left* – například `%left "+"` (případně `right` pro ty s pravou asociativitou). Dva znaky procento ukončují danou část a objevuje se definice gramatiky.

```
assignment_statement
    : IDENTIFIER '=' expression
    ;
selection_statement
    : IF '(' expression ')' statement
    | IF '(' expression ')' statement ELSE \\  

    statement
    ;
expression
    : IDENTIFIER
    | CONSTANT
    | STRING_LITERAL
    | '(' expression ')'
    | expression EQ_OP expression
    ;
statement
    : compound_statement
    | selection_statement
    | assignment_statement
    ;
statement_list
    : statement
    | statement_list ';' statement
    ;
compound_statement
    : '{' '}'
    | '{' statement_list '}'
    ;
%%
```

Jak vidíte, i tato nepatrná část jazyka C potřebuje celkem 6 terminálních a 6 nonterminálních symbolů. Princip je zřejmý. Startovacím symbolem je `compound_statement`, který může být buďto prázdný, nebo obsahovat seznam příkazů `statement_list`. Se-

znam může obsahovat jeden nebo více příkazů, které jsou odděleny středníkem. A tak se pokračuje dále.

```
int main()
{
    yyparse();
}

int yyerror(char* s)
{
    fprintf(stderr, "%s\n", s);
    return 0;
}

int yylex()
{
    return 0;
}
```

Nakonec kód obsahuje C kód. Jak vidíme, tak hlavní parsovací funkce se nazývá `yyparse`. Navíc vidíme funkci `yyerror`, která slouží pro obsluhu chyb v průběhu parsování a potřebnou funkci lexikálního analyzátoru `yylex`. Zde tedy máme pouze syntaktický analyzátor, ale chybí nám lexikální. Využijeme nám známý program `lex` a napíšeme definici lexikálního analyzátoru.

Protože na ní není vcelku nic zajímavého, uvedu sem pouze odkazy. Lexikální analyzátor `cselect.l` [3], zvýrazněná syntaxe `cselect.l.html` [4], syntaktický analyzátor `cselect.y` [5] a zvýrazněná syntaxe `cselect.y.html` [6]. Případně je k dispozici archiv i s Makefile – `cselect.tar.gz` [7]. Pokud se podíváte do definičního souboru pro parser, zjistíte, že je funkce `yylex` zakomentovaná a místo ní je v kódu

```
#include "lex.yy.c"
```

kteří zařídí vložení kódu lexikálního analyzátoru generovaného programem `lex`. Překlad toho celého provedeme takto:

```
yacc -d cselect.y
lex cselect.l
gcc -o cselect y.tab.c -fl1
```

Prvním příkazem vygenerujeme hlavičkový soubor `y.tab.h`, ve kterém jsou uvedeny makra pro jednotlivé tokeny. Navíc vygenerujeme zdrojový kód analyzátoru `y.tab.c`. Varování o shift/reduce konfliktech jsou způsobena nejednoznačnostmi v definici gramatiky. Při spuštění programu můžeme zadávat vstup:

```
$ ./cselect
{ a = b }
```

```
^D
$ ./cselect
{ if ( a == b ) { s = 1; gid = 0 }}
^D
$ ./cselect
a = b
parse error
$ ./cselect
{ a = b; }
parse error
```

Jak vidíme, parser skutečně provádí kontrolu podle zadané gramatiky. Například v ní není povoleno, aby poslední příkaz v seznamu končil středníkem. Rovněž není žádné zotavení se z chyb (což není nikterak jednoduchá věc a pro podrobnosti konzultujte dokumentaci), takže náš zárodek překladače je ekvivalentem jedné úpravy `gcc`, která vrací `true`, pokud je program dobře napsán a `false`, pokud špatně. Z reálně používaných programů dostávám stejně užitečná hlášení z MSIE – *Na stránce se vyskytla chyba*.

Další možnosti

Zatím náš program nepředával žádné hodnoty mezi skenerem a parserem. K tomu slouží proměnná `yyval`, takže můžeme do definice skeneru napsat

```
[0-9]+ {yyval = atoi(yytext); return CONSTANT;}
```

Jenže `yyval` je implicitně typu `int`. Naštěstí bison dovoluje definovat vlastní typ proměnné, takže do definice parseru přidáme

```
%union {
    int integer;
    char* string;
}
...
%token <integer> CONSTANT
%token <string> IDENTIFIER
```

a definice parseru se změní na

```
[0-9]+ {yyval.integer = atoi(yytext); \
    return CONSTANT;}
```

Při průchodu stromem potřebujeme při nalezení pravidla udělat nějakou akci. Například:

```
expression
: IDENTIFIER {printf("nalezen identifikátor\\
    %s\n", $1);}
;
```

Pokud vás zarazí `$1`, pak vězte, že jde o makro, které představuje odkaz na zásobník hodnot, které

analyzátor spravuje. Například aritmetické operace se dají psát takto:

```
expression:
: expression '+' expression { $$ = $1 + $3; }
```

Konflikty

Gramatiky nebývají jednoznačné a nástroje, jako je `yacc`, je odhalují. V našem případě se jedná mj. o problém volné klauzule `else`, kterou `trpí` mnoho jazyků. Pokud zapíšeme kód

```
if (a== b) if (b > c) do_anything else do_anything;
```

Překladač neví, ke kterému z obou `if` patří klauzule `else`. Při zpracování totiž překladač může provést *reduce* a uzavřít pouze první `if`, anebo *shift* pokračovat ve čtení dalšího tokenu a zpracovat `if-else`. Implicitně `yacc` provádí přesuny tak dlouho, dokud nezíská nejdelší vstupní řetězec, takže klauzule `else` se přiřadí druhému `if`, což je stejné chování, jaké programátoři v C (Pascalu a podobných jazycích) očekávají. Pro jednoznačný zápis musí programátor napsat `|`. Ovšem `bison` umožňuje přiřazovat některým větvím gramatiky prioritu. Například v jazyce `Python` [8] tato situace nemůže nastat, protože tam jsou bloky přesně definovány odsazením.

Antlr

Program s více než podivným názvem (znamená *Another toolkit for language recognizer*) způsobil menší pozdvižení v oblasti překladačů. Do té doby se všeobecně uznávalo, že není možné efektivně implementovat $LL(n > 1)$ gramatiku (pro připomenutí, gramatiku, která potřebuje pro rozhodování více než jeden token). Ostatně, bez ohledu na snahy Niclause Wirtha dávali vývojáři přednost LR parserům, jejichž implementace byly považovány za efektivnější. Tento program, napsaný v Javě, mívání vývojářů změnil. Podotýkám, že název, který končí na LR, mnoho vývojářů má, ale nemá to nic společného s LR parserem, program skutečně generuje $LL(n)$ gramatiku. Nejsme tedy omezeni na C, nebo C++, jako je tomu v předchozím případě.

Mezi jeho další výhody patří to, že generuje kód pro analýzu rekurzivním sestupem, který je čitelnější než výstup LR analýzy `bisonu`. Také dokáže generovat překladač pro čtyři prostředí – Java, v níž je napsán, C#, C++ a Python. Jeho Public Domain Licence navíc nikterak nebrání integraci do dalších produktů, jako je třeba `Intelli Idea` (viz [seznam](#) [9]). Případně dokáže vygenerovat kód pro průchod syntaktickým stromem, obsahuje podporu i pro sémantické akce

(ve skutečnosti jí trochu obsahuje i `bison`). Ukážeme si kousek analyzátoru (popravdě se jedná o upravený příklad ze stránek `antlr.org`):

```
{
    import java.io.*;

    import antlr.CommonAST;
    import antlr.DumpASTVisitor;
}

class P extends Parser;

startRule
:   o:Constant
    {System.out.println("Constant: \\"
        "+o.getText());}
|   i:ID
    {System.out.println("ID: "+i.getText());}
;

{
    import antlr.*;
}

class L extends Lexer;

// one-or-more letters followed by a newline

Constant
: '1'..'9' ( Digit )* NEWLINE
| '0' ( 'x' | 'X' ) ( 'a'..'f' | 'A'..'F' \\
    | Digit )+ NEWLINE
;

ID: ( Char ) ( Char | Digit | '_' )+ NEWLINE
;

NEWLINE
: '\\r' '\\n' // DOS
| '\\n' // UNIX
;

protected Digit: '0'..'9' ;
protected Char: 'a'..'z' | 'A'..'Z' ;
```

Jak vidíme, je možné prakticky kdekoliv v kódu psát kód v Javě. Ve výše zmíněném příkladu definujeme třídu pro parser i scanner v jednom souboru (na rozdíl od kombinace `flex/bison`). Dalším rozdílem je to, že je překladač objektový, takže máme třídu `P`

pro parser a L pro lexikální analyzátor. Ovšem takový kód sám od sebe nedělá nic, je nutné objekty inicializovat a zavolat.

```
import java.io.*;

class Main {
    public static void main(String[] args) {
        try {
            // vytvoreni scanneru
            L lexer = new L(new \\
                DataInputStream(System.in));
            // vytvoreni parseru
            P parser = new P(lexer);
            // zavolani startovaci podminky
            parser.startRule();
        } catch(Exception e) {
            System.err.println("exception: "+e);
        }
    }
}
```

Po instalaci balíčku jsem ze skriptu `/usr/bin/runantlr` zjistil potřebnou `CLASSPATH` tak, abych mohl analyzátor přeložit.

```
$ export CLASSPATH=$CLASSPATH:\\
    /usr/share/java/antlr.jar
$ java antlr.Tool grammar.g
ANTLR Parser Generator  \\
    Version 2.7.6 (20060511)  1989-2005
$ javac *.java
```

A používáme:

```
$ java Main
120
Constant: 120
$ java Main
identifikator
ID: identifikator
$ java Main
0112
```

```
exception: line 1:2: unexpected char: '1'
```

Závěr

V tomto díle jsme si představili dva nástroje pro tvorbu syntaktických analyzátorů. Je třeba si uvědomit, že oba dva generují parsery pro podmnožinu bezkontextových gramatik a proto, pokud vyžadujeme plnou sílu gramatiky, musíme využít jiné nástroje založené na jiných algoritmech. Přestože je kombinace yacc/bison součástí prakticky libovolného unixového systému, myslím, že program antlr stojí za pozornost. Existuje ovšem daleko větší řádka překladačů překladačů (jak se tyto nástroje nazývají).

- [Essence \[10\]](#) – Generátor LR parserů pro Scheme
- [GOLD Parsing System \[11\]](#) – Generátor LALR gramatik, který se pyšní největším počtem podporovaných cílových jazyků. Vedle C, Javy nebo Pythonu umí generovat kód pro x86 assembler nebo Visual Basic.NET.
- [javaCC \[12\]](#) – Další z řady LL(n) generátorů pro Javu.
- [SableCC \[13\]](#) – Generátor LALR parserů pro Javu (umí i C, C++, OCAML, Python a C#).
- [SmaCC \[14\]](#) – Generátor LR a LALR parserů pro Smalltalk.
- [cl-yacc \[15\]](#) – LR generátor pro Common Lisp.
- [SPARK \[16\]](#) – Implementace Earlyho parsovacího algoritmu v Pythonu.
- [CYK Parser c C++ \[17\]](#) – Implementace algoritmu CYK v C++

Tímto vynecháme syntaxi a posuneme se dále. Velkou otázkou příštího dílu bude – *co to vlastně znamená?* V terminologii počítačových vědců *sémantika*. A s tím související otázku typů. Navíc si konečně ukážeme schéma překladače.

■

Odkazy

- [1] <http://www.abclinuxu.cz/clanky/programovani/jazyky-a-prekladace-4-syntaxe-2>
- [2] <http://www.abclinuxu.cz/software/programovani/jazyky/c>
- [3] <http://www.abclinuxu.cz/data/vyskocil/jazyky-a-prekladace-5-cselect.l>
- [4] <http://www.abclinuxu.cz/data/vyskocil/jazyky-a-prekladace-5-cselect.l.html>
- [5] <http://www.abclinuxu.cz/data/vyskocil/jazyky-a-prekladace-5-cselect.y>
- [6] <http://www.abclinuxu.cz/data/vyskocil/jazyky-a-prekladace-5-cselect.y.html>
- [7] <http://www.abclinuxu.cz/data/vyskocil/jazyky-a-prekladace-5-cselect.tar.gz>
- [8] <http://www.abclinuxu.cz/software/programovani/jazyky/python>
- [9] <http://www.antlr.org/showcase/list>
- [10] <http://www.informatik.uni-freiburg.de/proglang/software/essence/>
- [11] <http://www.devincook.com/goldparser/>

- [12] <http://javacc.dev.java.net/>
- [13] <http://sablecc.org/>
- [14] <http://www.refactory.com/Software/SmaCC/>
- [15] <http://www.pps.jussieu.fr/~jch/software/cl-yacc/>
- [16] <http://pages.cpsc.ucalgary.ca/~aycock/spark/>
- [17] <http://www.cs.umbc.edu/~squire/download/cykp.cpp>

Jazyky a překladače – 6 (sémantika a typy 1)

Sémantika je trochu tajuplný pojem, přesto je při překladu velice důležitá. Syntaxe, kterou jsme se zabývali do minulého dílu, definovala, jak bude program vypadat. Sémantika určuje, co to vlastně znamená. S ní úzce souvisí i problematika typů. Na startovní čáru se nám tak postavily jazyky Smalltalk, Python, Javascript, Pascal, trocha Basicu, něco v PHP nebo PL/SQL a v shellu a v neposlední řadě C++ a Java.

Michal Vyskočil

Varování: tento článek není o Javě, přesto, že se někomu může zdát, že je v něm až příliš kódu v Javě. Klidně si za ni dosadte třeba C++ a s malými odchylkami (rozhraní, reflexe) bude platit to stejné. Stejně tak není o Pythonu – ten jsem zvolil jako zástupce dynamických OOP jazyků. Přestože jsou pokročilejší jazyky (například Smalltalk), z Pythonu toho umím více.

Princip překladače

Překladač je program, který (velice zjednodušeně řečeno) postupně prochází kód v nějakém programovacím jazyce a z něj generuje spustitelný soubor (nebo bytecode). Samotný překlad se sestává z několika fází a třetí je:

Sémantická analýza

Ovšem jak lexikální, tak ani syntaktická analýza není vše. Následující kód je z hlediska scanneru, tak i parseru naprosto v pořádku:

```
" program ve Smalltalku "
n := 10.
s := 'Hello, world!'.
[n = 10] if:
  [ Transcript show: n, s; cr. ]
```

No, nejen z hlediska scanneru, ale i z hlediska většiny lidí, kteří ve Smalltalku neprogramují (tímto děkuji Pavlu Křivánkovi za to, že mě donutil si nainstalovat Squeak a ten příklad si taky spustit a opravit :-). Řekněme to takhle: syntaxe je správně, ovšem program stejně nejde spustit. Při pokusu o jeho spuštění ve Squeaku nám virtuální stroj nabídne řešení problému, že selektor zprávy `#if` není znám.

Po opravení všech tří chyb, které jsem napáchal, získáme zdrojový kód, který je platný nejen po syntaktické, ale i sémantické stránce.

```
Unknown selector, please
confirm, correct, or cancel
if:
insertContentsOf:
a2f:
isFulfilledByAncestorsOf:
installedReleaseOf:
ifTrue:
isRevisionOf:
b6f:
installedVersionOf:
isKindOf:
id2:
cancel
```

```
" program ve Smalltalku "
n := 10.
s := 'Hello, world!'.
(n = 10) ifTrue:
  [ Transcript show: s; cr. ]
```

Formálně rozlišujeme tři druhy sémantik:

- Denotační sémantika popisuje sémantiku programovacího jazyka pomocí funkcí. Ty přiřazují sémantické hodnoty syntaktickým zápisům.
- Axiomatická sémantika popisuje účinek, který má provedení při různých podmínkách. Tedy takové, které jsou platné před provedením příkazu (vstupní) nebo po jeho provedení (výstupní).
- Operační sémantika ukazuje, jak je libovolný program vykonán na stroji, jehož činnost je definována (například Turingův stroj). Sémantika potom určuje, jak tento stroj program zpracovává.

Zanecháme teoretické informatiky a formálních definic a přejdeme na příklad. Následující operace mají (v podstatě) stejnou sémantiku a to, že provedou konkatenaci řetězců *a* a *b*, přičemž výsledek uloží do proměnné *c* – `concat.tar.gz` [1].

```

/*C*/
strcat(c, a);
strcat(c, b);
//C++
std::string c = a + b;
okno + "titulek";
#shell
c=$a$b
#PHP, Perl
$c = $a . $b;
--Oracle PL/SQL
c := a || b;
#Python
c = a + b
"Smalltalk"
c := a, b.
;Common Lisp
(setq c (concatenate 'string a b))

```

Jak vidíme, tak jazyky se docela dost liší v syntaxi zápisu operace, která má nakonec (prakticky) stejný výsledek. V C nejsou řetězce, a tak se problém řeší dvojitým voláním funkce `strcat()`. V shellu stačí napsat dvě proměnné vedle sebe a jejich obsah se pouze expanduje. V PHP, Perl a PL/SQL mají pro konkatenaci speciální operátor, což je pravděpodobně nejlepší způsob. V Pythonu a C++ se používá operátor `+`. V obou jazycích se jedná o volání speciální metody, které provádí vlastní sčítání. Tento způsob ovšem nezachovává komutativnost operátoru `+`. Protože pro řetězce neplatí vztah `a + b = b + a`, který platí pro čísla (zanedbáme omezení vyjádření čísel v počítačích), takže z matematického hlediska je lepší používat jiný znak. Smalltalk se také vydal cestou speciálního operátoru (znalci vědí, že `,` není operátor, ale selektor zprávy a podobně jako v Pythonu se nakonec jedná o volání příslušné metody). A nakonec implementace v Lispu se podobá jazyku C, kdy se pro konkatenaci volá funkce (pro větší podobnost by stačilo definovat funkci

```

char* concatenate(const char* ch1, const \
char* ch2)

```

Mezi příklady se nám objevil jeden, který má sémantiku odlišnou. Je to proslavený přetížený operátor `+`, který nastavuje titulek okna v jednom nejmenovaném toolkitu, což je častý argument odpůrců přetížení operátorů. Tato konstrukce má odlišnou sémantiku a takové použití operátorů není správné. Překladače obvykle nemají možnost rozpoznat a odmítnout uživatelsky definovaný operátor se špatnou sémantikou (výjimkou je třeba Prolog), proto je dů-

ležitě, aby programátoři svoje nadšení pro operátory krotili a používali je pouze na vhodných místech. Nicméně ani to nepovažuji za argument proti přetížení operátorů, protože pokud by dotyčný programátor zvolil pro nastavení titulku metodu `add()`, byl by to potom argument proti metodám? Anebo spíš ukázka špatně zvoleného jména.

Typové systémy

Pojem typ proměnné je v každém programovacím jazyce velice důležitý. Když jsem se učil programovat v Basicu, proměnné a funkce se znakem `$` na konci byly řetězce (moje oblíbená funkce `INPUT$`) a ostatní byly čísla. V některých implementacích (třeba Atari Basicu) nebylo více konstrukcí – jako procedury, které nahrazoval příkaz `GOSUB` (a `GOTO` pro skutečné programátory). Když jsem potom přišel k Pascalu, musel jsem se naučit deklarovat typy tímto způsobem:

```

var
    i, j, k : integer;
    errMsg : string[20];

```

Což způsobovalo mojí oblíbenou chybovou hlášku *tajp mišmaš* (*type mismatch*). V té době jsem nerozuměl, proč Pascal nedokáže přiřadit integer z typu `real` a proč to prostě neořízne, jako to fungovalo v Basicu. Ovšem záhy jsem poznal, že různé jazyky řeší problematiku typů různými způsoby.

Statické typování

Statické typování velice dobře známe z jazyků jako je Pascal, C, C++, Java nebo třeba Haskell. Při programování v těchto jazycích musíte explicitně deklarovat typy proměnných. Příkladem je

```

//staticke typovani v jazyce Java
NejakaTrida foo = new NejakaTrida();

```

Deklarujeme proměnnou typu `NejakaTrida`, kterou ihned inicializujeme konstruktorem patřičné třídy. V těchto jazycích je tedy typová informace explicitní a nezávislá na proměnné. Při deklaracích hlaviček funkcí/metod je tedy nutné uvádět typy, který musí parametry splňovat. Tohoto faktu velice silně využívá C++ při tzv. přetížení funkcí, kdy má jedna funkce několik implementací, které se vybírají na základě datových typů parametrů. C++ přineslo celou řádku nových datových typů, které mají usnadnit právě psaní přetížených funkcí.

```

void spam(unsigned int i);
void spam(signed int i);

```

Při překladu statického jazyka dojde ke kontrole sémantiky (a s tím související typové kontrole) jazyka, což znamená, že jsou typy do programu (až na některé výjimky) *zadržovány* natvrdo. Na druhou stranu mnoho jazyků nabízí prostředky k obcházení typového systému, protože jsou situace, kdy je statické typování omezující. Jazyk C nabízí ukazatel `void*`, C++ má navíc šablony. V Javě pro tento účel používáme typ `Object` a od verze 1.5 také šablony.

Ovšem ani statické typování není všemocné a existuje mnoho věcí, které v době kompilace vůbec ošetřit nejde. Oblíbené přetečení zásobníku v C je jenom hezkým důkazem toho, že bez runtime kontrol nejde zabezpečit přístup do paměti. Ostatně oblíbené výjimky jazyka Java (a jiných) nejsou nic jiného, než běhové kontroly.

Dynamické typování

Dynamické typování je odlišným způsobem. Přestože bude mnoho programátorů, kteří používají staticky typované jazyky, tvrdit opak, dynamické typování neznamena slabé, nebo dokonce žádné typování. Naopak, mnoho dynamicky typovaných jazyků je silně typovaných. Odlišnosti oproti statickému jsou dvě.

1. Typy se explicitně nedeklarují, jsou uloženy v runtime jazyka.
2. S tím souvisí, že se při překladu dynamických jazyků neprovádí typová kontrola. Typy se vyhodnocují v době *běhu* programu a ne v době překladu.

Pokud nevěříte, tak následuje ukázka:

```
// Java
Integer i = new Integer(15);
```

```
# Python
i = 15
type(i)
<type int>
```

```
; Common Lisp
(type-of 15/2)
RATIO
```

Takže zatímco ve staticky typovaných jazycích je nutné proměnné deklarovat a typovost uvádět explicitně (a třeba v Javě dokonce dvakrát), u dynamických tomu tak není. Proměnné jsou reference na interní reprezentaci objektu a její nedílnou součástí je i typ proměnné. To znamená, že z hlediska

jednoznačnosti typových informací jsou na tom jak statické, tak dynamické jazyky naprosto stejně (vynecháme problematiku slabě typovaných jazyků jako třeba shellu). Hlavním rozdílem je tedy místo uložení typové informace, zda je v runtime, nebo ve zdrojovém kódu. Jenom poukáži na častou výhradu, která se v této souvislosti objevuje. Následující kód

```
Object x = (Object) new Foo();
```

nedává u dynamických jazyků vůbec smysl. Prostě se vytvoří instance třídy `Foo` a pokud podporuje potřebné metody (rozumí daným zprávám), není žádný důvod přemýšlet nad něčím takovým, jako je přetypování. Ostatně, dynamické jazyky operátory přetypování nenabízejí. Tedy pokud za ně nepovažujeme konverzní funkce, třeba pro převod z čísla na řetězec.

Jedná se o velice podstatné změny programátorova myšlení, a i proto jsou dynamicky typované jazyky často odmítány. Při překladu dynamického jazyka prakticky nedochází k sémantické kontrole, ta se dělá až za běhu programu. Tedy v té době, kdy se vyhodnocují typy. Hezký příklad jednoho ze základních principů používaných v dynamických (OOP) jazycích nabízí Java (ano, opravdu Java) – viz její definice metody `System.out.println`:

```
public void println(Object x)
```

Tato metoda dokáže vytisknout libovolný objekt. Nevěříte?

```
public class PrintableHW {
    public PrintableHW() {}
    public String toString() {
        return "Hello, world!";
    }
};
....
//uplne jina class a jina metoda :-D
System.out.println(new PrintableHW());
```

A výstupem je `Hello, world!`. Toto je klasický případ situace, kdy nás absolutně nezajímá typ daného objektu. Zajímá nás, zda podporuje metodu `toString()`, kterou zavoláme.

Co se to tu vlastně stalo? Za prvé jsme trochu ošídili typový systém Javy (protože je `PrintableHW` potomkem třídy `Object`, obešla se akce bez explicitního přetypování) a díky reimplementaci metody `toString()` třídy `Object` se zavolala naše metoda a ne ta implicitní. Kód potenciálně složité metody `toString` se nám potom rozpadá na něco jako

```
public void println(Object x) {
    "vytiskni_retezec"(x.toString());
    "vytiskni_retezec"(EOL);
}
```

Ovšem, jak jsem už napsal. Java není dynamicky typovaný jazyk (přestože obsahuje některé jejich rysy) a podobné vlastnosti nabízí pouze při přetěžování metod třídy `Object`. V okamžiku, kdy si řekneme, že by se nám podobná vymoženost hodila i u jiných metod, musíme začít dědit, anebo vytvořit nějaké rozhraní, které musí implementovat všechny třídy, které chceme podobným způsobem používat.

Takže hlavním rozdílem mezi staticky a dynamicky typovanými jazyky je okamžik, kdy se zjišťuje, zda daný objekt podporuje danou metodu. Ve staticky typovaných jazycích kontrola probíhá ve fázi překladač, v dynamicky typovaných v době běhu. Dynamický styl programování bývá také nazýván jako *duck typing*, kde se k objektům přistupuje pomocí kachního testu:

If it walks like a duck and quacks like a duck, it must be a duck.

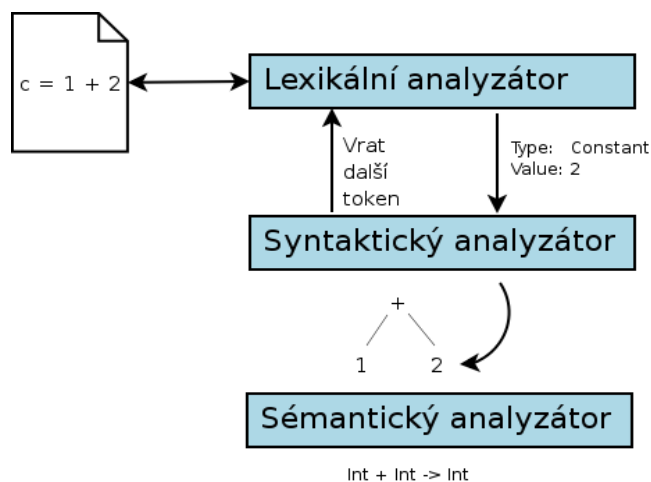
Pokud to chodí jako kachna a kváká jako kachna, potom to musí být kachna.

Což přesně vystihuje rozdíl. Staticky typovaný jazyk se zeptá – *je daný objekt typu kachna? Pokud ano, zavolej metodu `quack()`*. Dynamický jazyk funguje odlišně – *má daný objekt metodu `quack()`? Pokud ano, zavolej ji*. Je zřejmé, že není žádný problém podstrčit dynamickému jazyku například syn-

tezátor, který shodou okolností obsahuje metodu `quack()` :-). Podobná granularita až na úrovni jednotlivých metod je sice možná i ve staticky typovaných jazycích, ale nárůst počtu abstraktních tříd či rozhraní bude enormní.

Závěr

Příště si provedeme podrobnější porovnání statických a dynamických jazyků, protože se jedná o téma, které je velice zajímavé. Dojde i na šablony, přičemž zjistíme, že jde pouze o další import dynamičnosti do statických jazyků. V tomto díle nám zbývá jenom ukázat slíbený obrázek prvních tří fází překladač:



Na závěr bych chtěl poděkovat Pavlu Křivánkovi za článek [Statická vs. dynamická typová kontrola \[2\]](#), který mě inspiroval k sepsání tohoto (dvoj)dílu o statických a dynamických typových systémech. ■

Odkazy

- [1] <http://www.abclinuxu.cz/data/vyskocil/jazyky-a-prekladace-6-concat.tar.gz>
 [2] <http://www.root.cz/clanky/staticka-dynamicka-typova-kontrola/>

Jazyky a překladače – 7 (sémantika a typy 2)

V minulém díle jsme nakousli problematiku statických a dynamických typových systémů. V dnešním díle si ukážeme podrobnější srovnání možností obou přístupů. Na stranu staticky typovaných se postaví C++ a Java, na druhou stranu Python.

Michal Vyskočil

Metoda versus typ – aneb dynamický a statický přístup

V dynamických jazycích kontrola probíhá tak, že se za běhu zjistí, zda daný objekt umí požadovanou metodu (rozumí dané zprávě). Pokud ano, pak se zavolá. Pokud ne, vyhodí se výjimka o nepodporované metodě. To, co je důležité, je, že v době překladu (v podstatě jen kontrola syntaxe a převod do bytecode) se toto nezjistí, protože vyhodnocování probíhá až v čase běhu. To je velká změna od psaní ve statických jazycích, kde se tyto věci odhalí v době překladu (popravdě, programátoři většinou spoléhají na to, že toto všechno ošetří 100% překladač). A také velká výtka programátorů ve staticky typovaných jazycích, kteří často berou úspěšnou kompilaci jako důkaz, že je program správně napsaný. U dynamických jazyků je nutné psát testy, které projdou všechny větve a kód spustí, protože vlastní překlad prakticky žádnou kontrolu neprovádí. Toto chování hezky ilustrují tyto dvě ukázky kódu:

```
public interface LPrint {
    public String toString();
}
public class cls1 implements LPrint {
    public cls1() {}
    public String toString() {
        return "cls1";
    }
    public String toLongString() {
        return "### instance of cls1 ###";
    }
}
public class cls2 implements LPrint {
    public cls2() {}
    public String toString() {
        return "cls2";
    }
}
public void longPrint(LPrint x) {
    System.out.println(x.toString());
}
```

```
class cls1:
    def __init__(self): pass
    def __repr__(self):
        return "cls1"
    def toLongString(self):
        return "### instance of cls1 ###"
class cls2:
    def __init__(self): pass
    def __repr__(self):
        return "cls2"

def longPrint(x): x.longPrint()
```

Kód v Javě se ani nepřeloží. Překladač zjistí, že typ `cls2` neimplementuje metodu `longPrint()` a odmítne pokračovat. V Pythonu dojde k chybě až v době běhu a interpret vyhodí výjimku `AttributeError`. Překladač v Pythonu pouze zkontroluje tu trochu syntaktických pravidel a vygeneruje bytecode – dokud se kód nespustí, chyba se neprojeví.

Reflexe

Ovšem dynamické jazyky se tak snadno nevzdávají. Následující kód

```
def longPrint(x):
    try:
        x.longPrint()
    catch (AttributeError e):
        x.__repr__()
```

ukazuje sílu toho, že je možné za běhu testovat existenci metody a podle toho reagovat. Od tohoto okamžiku je naše tisková funkce stejně univerzální, jako standardní `print`. Nojo, řekne si javista, jaký je problém napsat něco jako

```
void longPrint(Object x, Boolean long) {
    if (long) {
        LPrint foo = (LPrint) x;
        System.out.println(foo.longPrint());
    }
    else {
        System.out.println(x);
    }
}
```



```

    }
}

```

a za cenu parametru navíc získat stejnou funkčnost? Jistě tušíte, že takové řešení postrádá jistou eleganci a obecnost. Naštěstí Java, díky tomu, že běží v runtime, obsahuje některé dynamické prvky (například operátor `instanceof`), takže je možné pomocí `java.lang.reflect` [1] získat potřebné informace. V případě třeba jazyka C++ bychom byli nahraní (pokud bychom si takový runtime nenapsali sami). Tedy možné to je, ale například autor knihy *Hardcore Java* [2] tvrdí:

Reflection is one of the least understood aspects of Java, but also one of the most powerful.

Reflexe je jedna z nejméně pochopených částí Javy, ale také jedna z nejsilnějších.

No a aby toho nebylo málo, něco nelichotivého (nejen) o reflexi se dozvíte i od *Slavy Pestova* [3], autora *jEditu*. Ten měl tu "čest" nahlédnout do zdrojů JVM. Přesto je dobré podotknout, že se pánové od Sunu (ostatní implementace neznám) v posledních verzích docela zlepšili a výkonový propad již není tak markantní. Navíc technologie *JavaBean* je na existenci reflexe postavená, což je další důvod, proč na zlepšení (žel bohu ne API, ale „jenom“ výkonu) reflexe pracovat.

A takto ji řeší dynamické jazyky, kde se k obsahu objektu dostaneme přeci jen snadněji (a navíc neřešíme věci jako rozhraní a podobně). A v těchto jazycích je používání reflexe a introspekce daleko častější než v Javě.

```

// Javascript
for (i in foo) {
}

```

```

# Python
foo.__dict__

```

```

" Smalltalk "
foo class allSelectors

```

Kolekce

Obrovská síla dynamických jazyků se projevuje při používání kolekcí. Dejme tomu, že máme seznam čísel. Čísla jsou typu `int` a my chceme spočítat (například) jejich aritmetický průměr.

```

float amean(intList* lst) {

    unsigned int sum = 0;

```

```

    for (intList::const_iterator i = \
        lst->begin();
        i != lst->end(); i++) {
        sum += *i;
    }
    if (lst->size() != 0) {
        return sum/lst->size();
    }
    else {
        return 0;
    }
}

```

Příklad je v C++ a třída `intList` je seznam typů `int` (ano, znám standardní knihovnu i šablony). Ovšem, jak to tak bývá, je nutné funkci rozšířit o podporu dalších typů jako třeba `float` a `double`. Díky možnosti přetěžování v C++ (v Javě také) můžeme napsat funkci `float amean(floatList* lst)`, nebo `float amean(doubleList* lst)`.

A co čert nechtěl, z nějakých důvodů, které jsou mimo nás (změna legislativy, šéfovy nálady, ...), se do algoritmu pro výpočet průměru zavádí změna. Nu a tak musíme přepsat jednu až *n* funkcí a v žádné neudělat chybu. C++ a Java od verze 1.5 nabízí generické datové typy. Seznam může být obecně libovolného typu, podstatné je, zda daný typ nabízí operátor `+` (tady je situace v Javě malinko složitější, ta podporuje pouze metody).

```

template <class T>
float amean(list<T>*> lst) {

    unsigned int sum = 0;
    for (typename list<T>::const_iterator \
        i = lst->begin();
        i != lst->end(); i++) {
        sum += *i;
    }
    // zbytek
}

```

Nyní máme univerzální funkci, která pracuje s libovolnými daty, která se dají sčítat a dělit číslem. Šablony tedy nejsou nic jiného než způsob, jak do statických jazyků přinést prvky těch dynamických. Sémantická kontrola v tom případě nekontroluje typy, ale to, zda existují příslušné operátory (metody). Hezkým důsledkem zavedení šablon je, že se typová kontrola neoslabil. Naopak, použitím šablon se v Javě (i v C++) typový systém naopak posiluje, protože před tím se používal typ `Object` (v C++

by se používal `void*`). To také ukazuje, že dynamické typování automaticky neznamená slabé typování. Naopak, většina dynamických jazyků je silně typovaná (rozhodně silněji než Java do verze 1.4 se svými `Object`y v kolekcích).

```
def amean(lst):
    if len(lst) != 0:
        return reduce(lambda x, y: x + y, \
            lst)/len(lst)
    else:
        return 0
```

To je kód s podobnou funkčností jako ten nahoře. Ovšem s jednou velkou výhodou. Kolekce se nemusí skládat ze stejných, ale různých typů, protože jazyku je to zcela jedno. Důležitá je existence příslušných metod a ne typ proměnné v seznamu. Nebo dokonce typ samotného seznamu. Tento kód v Pythonu je schopný vypočítat průměr z čehokoliv, co lze procházet iterátory (a čeho lze získat délku). V C++ by bylo nutné napsat kód poněkud jinak (stejně jak pracuje standardní knihovna – viz třeba `<algorithm>`).

```
template <class L>
    float amean(L begin, L end) {

        unsigned int sum = 0;
        unsigned int cnt = 0;
        while(begin != end) {
            sum += *i++;
            cnt ++;
        }
        // zbytek

    }

//zavolani
amean(lst.begin(), lst.end());
```

Taková funkce ovšem trpí tím, že mimo iterátorů z objektu nic víc nemůžeme získat. Já to vyřešil tak, že si délku seznamu spočítám v cyklu. Ovšem toto je v C++ považováno spíše za pokročilou černou magii (stejně jako reflexe v Javě) a podle mých zkušeností málokdo takové věci používá a spíše se tomu většina programátorů snaží (vědomě či nevědomky) vyhýbat. Opět musím upozornit, že v dynamicky typovaných jazycích jsou podobné konstrukce běžné a snadno napsatelné.

To, co se mi osobně na dynamických jazycích líbí nejvíce, je úspornost zápisu. Jádro celého kódu je díky funkcionálnímu `reduce` kratší než definice cyklu

`for` v C++, přičemž je ten kód ve výsledku daleko obecnější. Oproti C++ verzi s iterátory ten kód není tak zatemněný (ovšem jenom pro lidi, kteří znají lambda funkce). A to je Python poměrně ukecaný jazyk, stejný příklad třeba v Ruby by mohl být ještě o něco kratší.

Guido Van Rossum se o tomto zmiňuje v jednom rozhovoru [4] (český překlad naleznete na stránce [Jana Švece](#) [5]). Konkrétně v části [Productivity and Finger Typing](#) [6] (Produktivita a ruční psaní kódu [7]) říká:

This is all very informal, but I heard someone say a good programmer can reasonably maintain about 20,000 lines of code. Whether that is 20,000 lines of assembler, C, or some high-level language doesn't matter. It's still 20,000 lines. If your language requires fewer lines to express the same ideas, you can spend more time on stuff that otherwise would go beyond those 20,000 lines.

Kdysi jsem slyšel, že dobrý programátor dokáže spravovat 20 000 řádek kódu. Je jedno, jestli 20 000 řádek assembleru, C nebo nějakého vysokoúrovňového jazyka. Stále jde o 20 000 řádek. Pokud váš jazyk vyžaduje méně řádek k vyjádření téhož nápadu, můžete strávit více času na částech, které by se jinak do oněch 20 000 řádek nevešly.

Guido evidentně neslyšel o některých českých géniích, kteří píšou tisíce řádků skvěle komentovaného, kvalitního a otestovaného kódu v C++ (nebo PHP) denně už po několik let :-D

Eval, exec, apply a jiné

Díky tomu, že u dynamických jazyků je runtime tvořen i jejich interpreterem, je možné používat konstrukce, které vidíme v nadpise. Fakt, že můžeme za běhu konstruovat kód a ten potom nechat provést, je něco, co nám kompilované jazyky nenabízí. Nejlepším příkladem takového přístupu je Lisp, který byl prvním jazykem, který tyto konstrukce zavedl.

```
(setf a nil)
(push 2 a)
(push 1 a)
(push '+ a)
(print a)
(+ 1 2)
(print (eval a))
3
```

V Lispu není rozdíl mezi daty a kódem, takže dynamická konstrukce kódu je pouze plnění seznamu.

Generátory kódu pro syntakticky složitější jazyky nejsou tak jednoduché. Na této úrovni je Lisp podobný assembleru, kde není mezi daty a kódem rozdíl. Minimálně to platí pro počítače s von Neumannovou architekturou (a pomineme-li všechny read-only a non-execute příznaky, které beztak slouží jenom k emulaci Harwarské architektury), jejíž zástupce vám pravděpodobně leží na stole.

V jazyce Python není obsažena konstrukce

```
switch-case
```

Ovšem v něm také není potřeba (pro zajímavost existuje [Python Enhancement Proposal 3103](#) [8], ve němž se navrhuje tato konstrukce do Pythonu 3000), protože se dá nahradit za kombinaci slovníku a konstrukcí `exec`. Je pravda, že v jiných jazycích (Lisp, Smalltalk) je možné si chybějící příkazy doprogramovat sám, takže se neexistencí nejrůznějších konstrukcí nemusíme vůbec trápit.

```
case = {
    'a' : 'print "a"',
    'b' : 'print "b"',
    'c' : 'print "c"'
}
cond = 'b'
try:
    exec(case[cond])
except (KeyError):
    print "default";
```

Výhoda dynamického provádění kódu je zřejmá. Je možné za běhu sestavovat kód přesně na míru požadavkům, což kompilované jazyky (snad s výjimkou assembleru) neumožňují. Respektive umožňují, ale za cenu přibalení překladače a linkeru k programu a také za cenu starostí s IPC nebo sdílenou pamětí. Nevýhoda je ta, že se dynamické jazyky nedají kompilovat. Respektive dají... až na všechny výskyty podobných konstrukcí, které musí překládat interpret. Ale protože je u těchto jazyků fáze kompilace a spuštění stejná, není to tak markantní. Jiné jazyky, jako je třeba Java, bohužel v tomto ohledu *geniálně* zkombinovaly nevýhody kompilovaných (musí mít speciální překladač) a interpretovaných (musí mít runtime, není to nativní kód) jazyků.

Ostatní

Nakonec bych chtěl ukázat pár věcí, které se mi zdály jako dost zajímavé ukázky možností dynamických jazyků a které staticky typované neumožňují.

Rozšíření jazyka

Programovací jazyky, které mají jednoduchou syntaxi, jsou nesmírně rozšiřitelné. Všechny základní konstrukce v nich jsou totiž realizovány prostředky samotného jazyka, nikoliv syntaxí. Díky tomu je můžeme používat například ke zkoumání nových konstrukcí, které bychom jinak realizovali dost obtížně. Například podmínka `if` je v Lispu pouze makro.

```
(defmacro if (test then-form &optional \
  else-form)
  '(cond (,test ,then-form)
    (t ,else-form)))
```

Takže je možné vytvořit třeba takovou věc jako více-stavové `if` (fuzzy-if) nebo cyklus `for` s větví `else` (což nějaký jazyk opravdu má).

Univerzální třída

S využitím speciální metody `__getattr__` je možné napsat třídu, jejíž instance *umí* všechny metody.

```
class C:
    def handlerFunctionClosure(self,name):
        def handlerFunction(*args,**kwargs):
            print "call c.%(s)%(s, %s)" % \
                (name,args,kwargs)
            return handlerFunction
        return __getattr__(self,name):
        return self.handlerFunctionClosure(name)
```

A příklad volání:

```
<<<c.f1()
call c.f1((), {})
<<<c.f1('a')
call c.f2(('a',), {})
<<<c.f1(val='a')
call c.f3((), {'val': 'a'})
```

Tímto způsobem se velice snadno naprogramují různé dekorátory, proxy, nebo jim podobné návrhové vzory.

Závěr

Srovnání statických a dynamických jazyků je především o srovnání dvou různých přístupů při tvorbě programů. Možná i proto vyvolává mezi programátory tolik vášní. Vyměnit například Pascal za C++ nebo Javu není tak složité. Ale vyměnit ji za odlišný způsob myšlení, který klade nároky na zcela jiné věci, to je asi největší problém. Každý má tendence chválit to svoje, ovšem dynamické jazyky jsou mnohem flexibilnější než ty statické. Na druhou stranu

jsou pomalejší a navíc i ty nejtriviálnější kontroly dělají až za běhu programu. Statické poskytují jistý komfort typového systému, kdy musíme překladači přesně deklarovat všechny typy, ale za to máme částečnou jistotu, že je program správně. Skutečně dynamické jazyky s jednoduchou syntaxí (Lisp, Smalltalk nebo Forth) dokonce umožňují programovat sa-

motné jazykové konstrukce (`if`, `for`, ...) přímo prostředky samotného jazyka. Díky tomu je možné každý z těchto jazyků rozšířit, například pro výzkum nových konstrukcí, což ve většině *normálních* jazyků není možné. Příští díl se bude zabývat tématem operátorů a jejich vyhodnocováním, což s překladači také souvisí. ■

Odkazy

- [1] <http://java.sun.com/j2se/1.5.0/docs/api/java/lang/reflect/package-summary.html>
- [2] <http://today.java.net/today/2004/05/05/HardcoreJava.pdf>
- [3] http://www.jroller.com/page/slava?entry=the_hotspot_source_code_is
- [4] <http://www.artima.com/intv/guido.html>
- [5] <http://www.py.cz/Honza>
- [6] <http://www.artima.com/intv/speed.html>
- [7] <http://www.py.cz/RozhovorCast3>
- [8] <http://www.python.org/dev/peps/pep-3103/>

Kubuntu 6.10 Edgy Eft – Instalace ve dvojím stylu

Článek kriticky rozebírá dva druhy instalace Kubuntu 6.10: 32bitové verze, totiž instalaci z běžícího LiveCD určenou pro nováčky a začínající uživatele; a instalaci z alternativního CD-ROM v tzv. OEM-režimu, která je zajímavou možností např. pro obchodníky s počítači, kteří chtějí nabízet předinstalovaný operační systém.

Martin Tesař

Článek není návodem ani dokumentací k instalaci!

Ubuntu a spol.

Linuxová distribuce Ubuntu a její varianty (Kubuntu, Xubuntu, Edubuntu) patří dva roky od vydání první ostré verze k těm nejpobulárnějším. Koncepčně vychází z prověřeného a rozšířeného Debianu a cílem je být „Linuxem pro lidi“, tedy naprosto svobodným operačním systémem, který „just works“ (prostě funguje).

Ubuntu 6.10 Edgy Eft vyšla v říjnu 2006 jako v pořadí pátá ostrá verze, pouhé čtyři měsíce po verzi 6.06, která zavedla jako novinku dlouhodobou podporu (Long Time Support). Ta se striktně vzato nynější verze 6.10 netýká; nebylo by ostatně ani logické, aby uživatelé vyžadující dlouhodobou podporu svůj systém hned po čtyřech měsících měnili. Přesto lze z verze 6.06 jak Ubuntu, tak Kubuntu upgradovat na verzi 6.10.

Novinky ve verzi 6.10

Ještě před rozbozem vlastní instalace se aspoň stručně zmiňme o některých novinkách, které nová verze Ubuntu slibuje, a jejichž podrobnější rozbor je spíše námětem na samostatný článek. Zejména uživatele předchozích verzí by měl překvapit alespoň o něco málo rychlejší start: tradiční *sys-v-init* je nahrazen novějším systémem *upstart*, fungujícím na základě událostí.

Hodně se také diskutuje o 3D efektech pro desktop. Přítomný Xorg 7.1 sice již obsahuje rozšíření AIGLX, které umožní 3D desktop na některých grafických čipech ATI a Intel, ale uživatel musí samostatně ještě nainstalovat a nastavit kompozitní manažer *com-piz* [1], stejně jako vyžaduje ruční úpravy konfiguračního souboru */etc/X11/xorg.conf*. Také XGL, rozšíření z produkce firmy Novell, je třeba přidat do systému z repozitáře Universe.

Vůbec celá záležitost vyžaduje ještě tolik administrátorských zásahů, že 3D, ač jistě z hlediska příznivců „eye-candy“ žádoucí, patří zatím stále ještě do experimentální oblasti, i v „Linuxu pro lidi“, což platí i o třetí velké novince, virtualizátoru Xen [2]. Pozitivní zjištění však je, že tyto novinky již existují a nyní se může intenzivně pracovat na jejich dolaďování.

Nevynechme ještě povinnou zmínku o mainstreamu: v Ubuntu kromě Xorg 7.1 a kernelu 2.6.17 je přítomno tip top nové GNOME [3] 2.16 a v Kubuntu pak neméně ještě teplé KDE [4] 3.5.5. Rovněž najdete OpenOffice.org [5] 2.0.4 a GIMP [6] 2.2.13 a vývojáři jsou hrdí na to, že zachytili i nový Firefox [7] 2.0 těsně po jeho oficiálním vydání [8] (ten ovšem ve výchozí instalaci Kubuntu chybí, jako browser tam slouží čím dál lepší Konqueror [9]).

Vylepšeny byly i další detaily, např. teď ovládá WLAN šifrovací standard WPA pomocí Network-Manager. Za zmínku stojí také vylepšený nástroj pro update zdrojů a balíčků integrovaný v Synapticu [10] – podrobněji se na programové vybavení a konfiguraci desktopu pokusím podívat v příštím článku.

Nová instalace nebo upgrade?

Řada uživatelů volí při vydání nové verze kompletní re-instalaci systému, ačkoliv vývojáři věnují nemalou péči tomu, aby byl zajištěn hladký a bezproblémový upgrade. Zejména pokud jste nezaváděli do své distribuce příliš hluboké individuální úpravy, můžete se před re-instalací pokusit o upgrade přes síťové připojení. Je však záhodno dodržet doporučený postup, a protože se ve vysoké míře objevují zklamané hlasy po rozsypaných upgradech, stručně si tu správný postup upgrade z Kubuntu 6.06 vyjmenujme:

Především je třeba nahradit v konfiguračním souboru */etc/apt/sources.list* u všech zdrojů starší „dapper“ novou variantou „edgy“. Potom v terminálu postupně zadat tyto příkazy:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install kubuntu-desktop \
python-qt3 python-kde3 ubuntu-minimal
```

A nakonec restartovat počítač. Počítejte se stahováním až několika set MB; pokud jste si již stáhli instalační CD, vložte je do mechaniky a před výše uvedenými příkazy zadejte ještě `apt-cdrom`.

Instalační média

Pokud ještě Kubuntu nainstalované nemáte anebo se přece jen rozhodnete k čisté nové instalaci, můžete k tomu použít několik druhů instalačních médií. Vedle stále asi ještě nejčastěji instalované verze pro 32bitovou architekturu, které se budeme v následujícím věnovat, můžete v případě vlastnictví patřičného hardware sáhnout po verzi 64-bit nebo Mac (PowerPC).

Novinkou je tentokrát, že ISO obraz jediného instalačního CD-ROM je možné stáhnout z celé řady [mirrorů](#) [11] ve dvou variantách nazvaných *Desktop* a *Alternate*. Varianta Desktop funguje zároveň jako LiveCD, které v souladu s moderním trendem umožňuje spustit instalaci z plně běžícího operačního systému. Lze si tak předem nejen Ubuntu prohlédnout, ale také ještě před vlastní instalací zjistit, jak se bude Ubuntu snášet s vaším hardwarem, a při instalaci využít přístup k Internetu pro případné hledání řešení problémů.

Alternativní instalační CD-ROM je pak určeno pro méně obvyklé způsoby instalací, např. pro PC s malou RAM, pro instalaci na virtuální rozšířené disky (LVB), pro upgrade předchozí verze Ubuntu. Startuje v textovém režimu a nabízí výběr řady variant. Pro zkušenější ale hlavně pro nezkušené je důležitá informace, že pouze tato alternativní verze vám umožní instalovat [zavaděč](#) [12] jinam než do [MBR](#) [13] (o tom podrobněji níže v článku). Novinkou mezi instalačními metodami, kterou umožňuje toto alternativní CD, je instalační režim OEM.

Rozhodl jsem se, vzhledem k deklarované snaze vývojářů Ubuntu, vyzkoušet tentokrát oba dva způsoby instalace a podívat se na ně kritickým pohledem. Hlavně z hlediska efektního působení a snadnosti i pro začátečníka; to se týká instalace z LiveCD. Pro instalaci z alternativního CD jsem zvolil režim OEM.

Instalace systému z LiveCD: nejschůdnější cesta pro nováčky

Volby úvodní obrazovky

Kubuntu v nejnovější verzi desktopové varianty nabízí po bootování CD startovací obrazovku instalátoru a zároveň LiveCD ([screenshoty Kubuntu](#) [14]). Tmavomodrá nabídka pod logem Kubuntu je nejprve v angličtině, ale již zde můžete přepnout do češtiny či slovenštiny pomocí klávesy F2, případně lze také zvolit potřebné rozložení klávesnice. Hned po startu se začíná odpočítávat 30 sekund, takže pokud se nedotknete žádné klávesy, spustí se... instalátor? Kdepak, zavede se LiveCD Kubuntu a to přímo, po několika minutách, do plně funkčního prostředí KDE.

Zastavme se však ještě u úvodní obrazovky. Předem je třeba poznamenat, že zvolený jazyk na první obrazovce se netýká lokalizace prostředí KDE na LiveCD. Potěší však, že navzdory nápisům v angličtině funguje česká klávesnice i se všemi nabodeníčky (a tzv. mrtvými klávesami) a že je snad již definitivně opuštěna praxe volit jako výchozí písma do KDE ta, která neobsahují zrovna českou (a slovenskou) znakovou sadu. Můžete tedy jak psát, tak číst pohodlně česky.

K výběru jazyka a klávesnice na úvodní obrazovce snad ještě dvě malé poznámky: nabídka obsahuje opravdu řadu jazyků, a to v původním pojmenování (např. „Čeština“), aby se tak maximálně ulehčil výběr těm, kteří žádný jiný jazyk než mateřský neovládají. Ne každý si ale všimne, že písma v jazycích píšících zprava doleva, tedy obráceně než my, jsou psána špatným směrem. Názvy rozložení klávesnic naopak zůstávají stále anglicky, takže pro našince bude relevantní volba „Czechia“, případně „Slovakia“. Jinými slovy, počáteční zamýšlený efekt je ještě trochu nedoleštěný.

Kromě položky menu zvoucí ke startu Kubuntu a instalaci (z tohoto CD nelze instalovat bez spuštění grafického prostředí), můžete ještě vyzkoušet v případě problémů s grafikou start v *Safe Mode* a nechat otestovat CD či paměť. CD také nabízí nouzový přechod přímo do [bootování](#) [15] z prvního pevného disku, což pomůže těm úplně zmateným nováčkům k návratu do jejich obvyklého systému, kdyby si instalaci přece jen rozmysleli.

Další možnosti, které můžete zvolit již před startem, jsou volba rozlišení a počtu barev obrazovky, což se týká zejména staršího hardware, zvláštní volbou je pak start LiveCD s nástroji pro zrakově postižené,

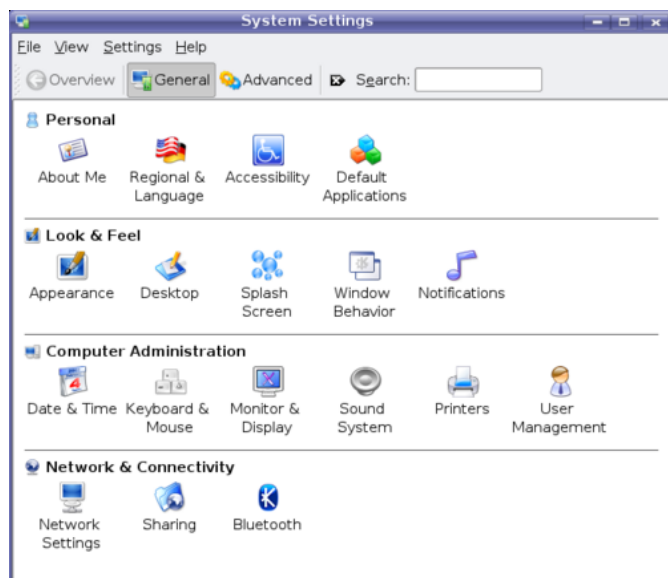
s extra kontrastním nastavením nebo s lupou, či modifikátory klávesnice pro pohybově handicapované.

Trable v grafickém prostředí LiveCD

Samotný start probíhá v souladu s trendem neděsit laické uživatele kryptickými výpisy, které mu stejně nic neřeknou: namísto výpisů detekce hardware a zavádění modulů zírání tak na splashové logo Kubuntu v temně modré a akci prozrazující vodorovný sloupek postupně měnící barvu.

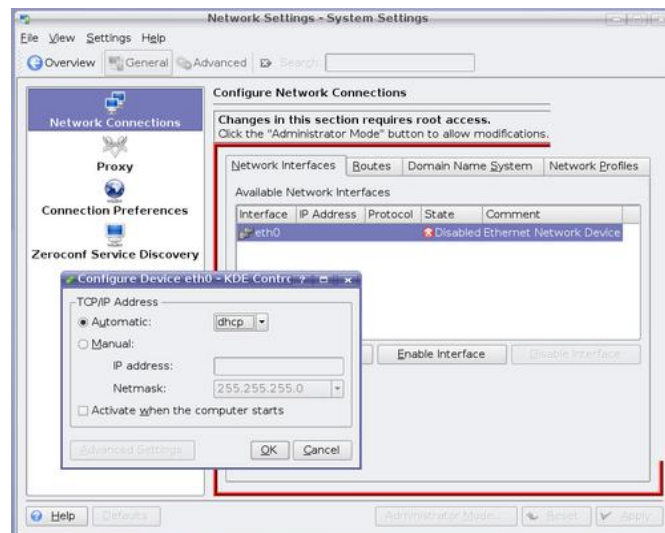
Po několika minutách (podle rychlosti systému) uvidí uživatel plochu KDE, kde jsou pouze dvě ikony: *Examples* a *Install*. Jak bylo již zmíněno, můžete KDE hned používat. Nebýt poněkud nemilého překvapení, hlavně pro uživatele se zkušenostmi z jiných LiveCD: navzdory dostupnému DHCP připojení není konfigurován přístup k síti.

První pokusy dostat se přes browser Konqueror „na Internet“ tak skončí fiaskem. Nezkušený nováček si v tomto případě bez čtení dokumentace patrně příliš neporadí – ten zkušenější se jistě pokusí zapátrat v rozbalovací nabídce KDE, kde nalezne to právě v položce *System Settings*, otvírající okno s výmluvnými ikonkami různých nastavení systému – mezi nimi *Network Settings*. Tam už lze nejprve konfigurovat detekované síťové zařízení, např. `eth0` a potom je aktivovat. Pokud nemáte připojení s DHCP, musíte samozřejmě znát patřičné údaje předem. Až pak si můžete užívat Internetu na LiveCD.



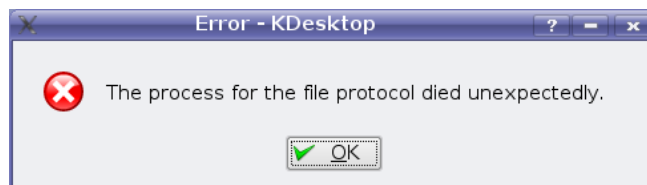
Je sice pravda, že k instalaci z CD-ROM prakticky připojení k internetu nepotřebujete, ovšem pokud připojení nemáte, těžko se vám povede nainstalovat během instalace lokalizační balíčky českého jazyka (jediné instalační CD pochopitelně nemůže obsahovat všechny jazykové soubory).

Za úvahu také stojí, že instalace z LiveCD bývá volena mimo jiné z toho důvodu, že je možné v jejím průběhu vyhledat přes Internet pomoc v diskusních fórech nebo online dokumentaci. K absenci automatické konfigurace síťového připojení sice mohly vést např. bezpečnostní obavy, ale protože alternativní instalační CD, jak ještě uvidíme, automatické připojení pomocí DHCP detekuje a ihned zřídí, působí to u LiveCD spíše jako opomenutí.



Bude-li už chtít pracovat připojení k Internetu a nenaleznete v KDE nabídce nějaký program, na který jste byli zvyklí, zkuste ho jednoduše spustit z příkazové řádky, třeba tam je, jak jsem se mohl přesvědčit sám, když jsem potřeboval *KNetAttach* [16] (program pro přístup k webovým adresářům).

A ještě maličkost, která může nezkušené i zkušenější vystrašit: při startu grafického prostředí vyběhne KDE-dialog s hlášením: *Error-KDesktop: The process for the file protocol died unexpectedly*. Ten se sice dá prostě odkliknout tlačítkem OK – ale moc nadšení to při prvním a dosud hladkém průběhu startu nepřidá.

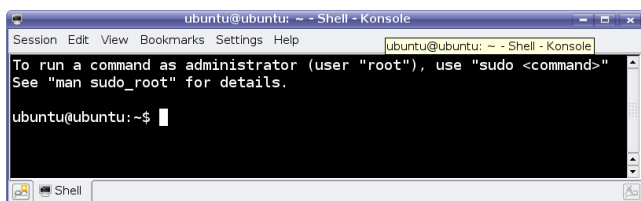


Příprava instalace Kubuntu

Tu spustíte kliknutím na ikonku pojmenovanou *Install*. Navzdory anglickojazyčnému prostředí se instalátor spustí český a konečně tak zúročí výběr jazyka v úvodní obrazovce. Ale i když jste nastavení jazyka a klávesnice budoucí instalace opomněli, můžete to nyní napravit, protože instalátor opět začíná malým dotazníkem na jazyk, časové pásmo a po-

žadované rozložení klávesnice. Také budete vyzváni k zápisu jména, uživatelského jména a volbě hesla.

Důrazná rada zejména pro začátečníky: nenechávejte kolonku hesla prázdnou a dobře si heslo zapamatujte. Kubuntu totiž startuje po instalaci do grafického režimu s přihlašovacím dialogem, který se vás na uživatelské jméno a heslo zeptá – a protože možnost přihlašování bez hesla lze nastavit až z běžícího systému, nebudete se moci bez správného hesla přihlásit.

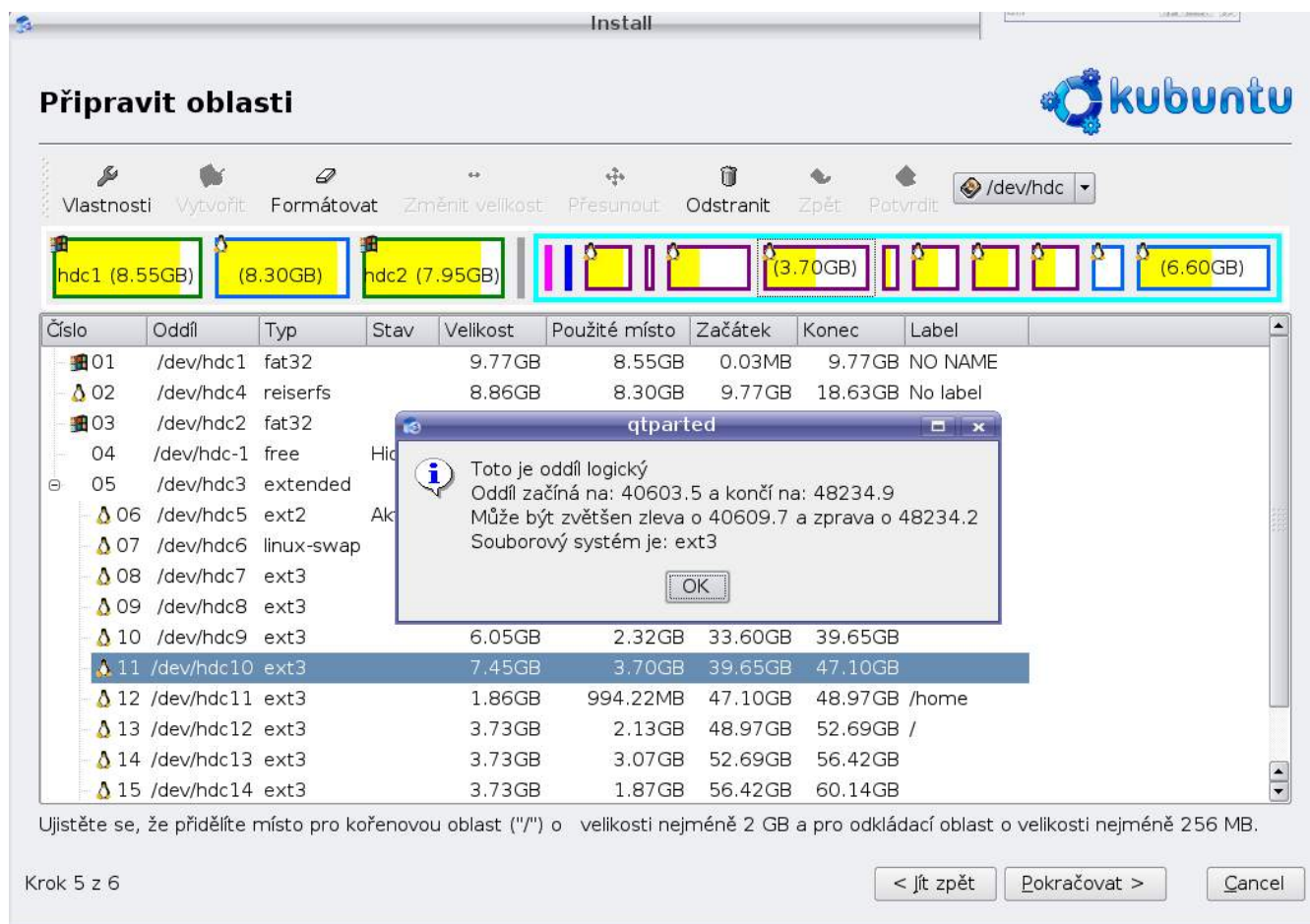


Rovněž, díky specialitě distribuce Ubuntu & co., není založen účet uživatele `root` [17], takže se ne-

můžete prostě přihlásit jako `root` z vedlejší konzole. Jednoduché řešení přihlášení přímo z distribuce neexistuje a je nutno řešit pokročilejšími metodami, např. pomocí jiného LiveCD znovu nastavit a v režimu `chroot` založit buď konto nového uživatele anebo zřítit konto `root` – což jsou všechno metody poněkud nestandardní a pro začátečníka obvykle nedostupné.

Příprava oddílů pevného disku

Pak už následuje klasická instalační procedura: volba disku, na který hodláte instalovat, s patřičnými předvolbami (celý, volná část) a také s možností ručního rozdělení a úpravy pevného disku. Pomocí dialogu s programem `qparted` [18] můžete připravit oblasti disku, rozdělit je, smazat nebo spojit, dokonce i zmenšit nějaký oddíl s Windows, pakliže víte, co děláte.



Příprava oddílů pevného disku

Na rozdíl třeba od SUSE, jehož instalátor poskytuje poměrně vyčerpávající informace, jsou ještě nezkušení administrátoři Kubuntu odkázáni na dokumentaci mimo instalátor. To se bohužel projeví např. v kroku 5, kdy vás musí napadnout pomocí kontextového menu (stisk levého tlačítka myši) a po-

ložky **Vlastnosti** patřičný vybraný oddíl zformátovat, neboť k tomu nebudete zvlášť vyzváni a v příštím kroku pak marně přemýšlet, proč vás nechce instalátor pustit dál, ačkoliv jste dělali přesně to, co po vás chtěl. Naštěstí se lze celou procedurou pohybovat i nazpět, takže zvědavější povahy snad přijdou na

správný postup důkladnou prohlídkou kontextového menu.

Je třeba také věnovat velkou pozornost tomu, který oddíl jste označili k následnému formátování, zvláště pokud jich máte na disku víc. Budete sice ještě varováni, že oddíl bude smazán a data na něm zničena, ale potom už neexistuje cesta zpět. Právě tento kritický moment instalace, při němž velice snadno můžete přijít o data nebo i systémové oddíly jiných operačních systémů na témže disku, by si zasloužil precizněji a méně stručně formulovaného průvodce.

Instalace GRUB: automaticky do MBR

V závěrečném přehledu kroků, které se instalátor chystá provést, byste také neměli přehlédnout, že instalátor vám nainstaluje také GRUB [19], a to do prvního diskového oddílu, do MBR. Možnost tento postup změnit při instalaci z tohoto CD není, a proto s tím počítejte a informujte se předem, co tato skutečnost znamená.

Instalátor sice provede detekci ostatních přítomných systémů včetně Windows a pokusí se konfigurační soubor GRUBu (`/boot/grub/menu.lst`) korektně nastavit, přece jen ale může dojít k problému. Totiž tehdy, pokud instalujete na disk poněkud dál, nebo na druhý disk, a váš `BIOS` [20] nedosáhne za cylindr 1024. Pro takové konstelace je vhodnější zvláštní diskový oddíl jako `/boot` společný pro všechny přítomné distribuce a ruční přesunutí obrazů kernelu do tohoto oddílu. Takovou netriviální instalaci si pak dobře rozmyslete a zvažte, zda není vhodnější sáhnout raději po alternativním instalačním CD.

Vlastní instalace

Potom už instalátor vyzve ke spuštění vlastní instalace, kterou nelze nijak přerušit. Počítat můžete přibližně s půlhodinou, přičemž budete průběžně stručně informováni o tom, co se právě děje. Ti, kdo chtějí vědět víc, mohou sledovat průběh na konzoli přístupné přes klávesovou zkratku `Alt+F4` (nazpět do KDE `Alt+F7`). Jinak můžete během instalace normálně dál pracovat v grafickém prostředí KDE, psát, surfovat – omezení budete pouze výkonem vašeho PC.

Jak již bylo zmíněno, je vhodné mít již před instalací konfigurované a funkční připojení k Internetu, neboť instalátor projeví v patřičném okamžiku touhu stahovat potřebné jazykové balíčky z `repozitářů` [21]. Zde vás také může potkat nemilé zdržení, jsou-li tyto zdroje momentálně nedostupné – *time out* je

nastaven dosti dlouhý a instalace se může jakoby zadržnout.

Po ukončení instalace pak zbývá restart: můžete si vybrat, zda chcete restartovat hned, anebo ještě pokračovat v práci v live distribuci a restartovat, až se vám to bude hodit.

Alternativní instalace systému: OEM pro obchodníky a administrátory

Kubuntu lze, kromě cesty instalace z „desktop“-CD, instalovat také alternativně, z CD označovaného jako „alternate“. To se doporučuje stáhnout zejména v případech, kdy chcete instalovat na slabší stroj (instalace probíhá v textovém režimu) a hlavně mít na průběh instalace a její konfiguraci větší vliv. Tedy např. na to, zda a kam se nainstaluje zavaděč GRUB.

Zajímavě vypadá varianta nazvaná v počátečním menu instalátoru *OEM-instalace*. Pod ní se skrývá běžná instalace s tím, že bude nejdříve zřízen jakýsi dočasný uživatel „oem“ se zvoleným heslem, který bude po dokončené instalaci spuštěním připraveného skriptu automaticky odstraněn. Po restartu se pak systém nového uživatele zeptá na všechny nové údaje, jazyk, klávesnici, časové pásmo (to bohužel zadat nelze) jméno a heslo a namísto uživatele „oem“ (jehož konto bylo beze zbytku vymazáno) bude zřízeno nové konto nového uživatele, který je nyní jediným pánem a vládcem systému.

Instalace v textovém režimu

Nevýhoda instalace v textovém režimu je, že během ní nemáte k dispozici grafické nástroje pro práci s Internetem, i když na rozdíl od LiveCD je třeba DHCP připojení detekováno a nastaveno automaticky, takže i během instalace jste online.

Ruční nastavení disku, přípojného bodu i formátování oddílu je oproti variantě z LiveCD poněkud přehledněji komentováno, a člověk se tak i při komplikovanější konfiguraci instalace cítí pevněji v kramflecích: už jen tím, že si vše nastavujete opravdu sami a že kroky za sebou následují všechny logicky. Na rozdíl od grafické instalace nelze sice zvolit možnost neformátovat nalezený swapový oddíl, ale to je trivialita.

Je však třeba mít více trpělivosti, protože čas od času není vůbec navenek znát, že se vůbec něco při instalaci děje, takže jen šumící CD vyvrací dojem, že došlo k nějaké havárii (pokud samozřejmě nesledujete akce v konzoli `Alt+F4`). To se hodí zejména

někdy kolem 85 % hotové instalace, kdy instalátor provádí v pozadí pokusy stahovat české jazykové balíčky z českých repozitářů: ty mohou být dočasně nedostupné a instalace se pak zastaví, aniž byste věděli proč, vlastní instalátor prostě jen tak čeká.

Při instalaci, která i v textovém režimu probíhá převážně v češtině (kromě nevýznamných hlášení, že právě probíhá „preparing, configuring“ atd.), budete vlastně až na jejím konci upozorněni, že instalační CD neobsahuje plnou podporu pro zvolený jazyk (tedy češtinu). Toto upozornění však stejně jako nabídka automaticky potřebné soubory z internetu doplnit a nainstalovat zůstala pouze v angličtině. Na totéž si vzpomínám z verze 5.10 [22] a musím opakovat, že zejména tato část instalátoru by lokalizována být určitě měla.

Instalace zavaděče GRUB

Ke konci instalace můžete ovlivnit umístění zavaděče GRUB. V nápovědě je uváděna možnost instalovat GRUB na různá média spolu s názornými příklady (pro disketu příklad chybí). Máte také na rozdíl od instalátoru z LiveCD možnost pokračovat bez zavaděče, přičemž se vám dostane bohatě upozornění, důrazného varování ("kritická chyba!") a nakonec rady, jak zapsat vlastnoručně parametry do konfiguračního souboru GRUB, abyste neskončili s nestartujícím počítačem.

Zde ještě následuje upozornění na již výše popsany charakter OEM-instalace – avšak pouze v angličtině: Při prvním restartu se musíte přihlásit jako uživatel „oem“ se zvoleným heslem. Ve spuštěném systému pak můžete doinstalovat jako administrátor další programy a provádět systémová nastavení.

Po požadovaných úpravách grafického prostředí a instalaci nových programů pak vyvoláte v terminálu (Konsole) příkazem `oem-config-prepare` skript, který systém připraví na přihlášení nového uživatele hned po příštím restartu. Je třeba mít na paměti, že uživatelská nastavení a vytvořené soubory v `/home` budou po přihlášení nového uživatele vymazány, pamatujte, že uživatel „oem“ je pouze dočasný!

I při prvním přihlášení nového uživatele se bohužel opakuje nám již z LiveCD známé chybové hlášení (*The process for the file protocol died unexpect-*

edly), naštěstí jen při tom prvním. První přihlášení také trvá poněkud déle než další běžné starty – na to asi bude vhodné případného zákazníka připravit.

```
oem@ubuntu:~$ sudo oem-config-prepare
Password:
Adding system startup for /etc/init.d/oem-config ...
/etc/rc2.d/S12oem-config -> ../init.d/oem-config
/etc/rc3.d/S12oem-config -> ../init.d/oem-config
/etc/rc4.d/S12oem-config -> ../init.d/oem-config
/etc/rc5.d/S12oem-config -> ../init.d/oem-config
oem-config will run the next time the system boots.
oem@ubuntu:~$
```

Závěrem

Instalační procedura nového Kubuntu chce jistě nabídkou alternativ oslovit co možná nejvíce druhů uživatelů. Nápad jistě dobrý, ale na jeho doladění je ještě třeba pracovat.

Jako první citelný nedostatek se jeví, že LiveCD, určené především začátečníkům, nedokáže automaticky detekovat připojení k síti přes DHCP, což bývá dnes u LiveCD zavedeným standardem. Rovněž bych si přál namísto automatické instalace GRUBu do MBR alespoň volbu pro pokročilý přece jen GRUB neinstalovat – nemusel bych pak GRUB obnovovat z jiné linuxové distribuce.

Drobné lokalizační nedostatky během instalace jsou vesměs překonatelné, ale možná by se tým (K)Ubuntu mohl poučit od SUSE, jak to vypadá, když se to udělá profesionálně. Totéž by se dalo říci o informacích instalátoru o právě prováděných úkonech, zejména při rozdělování a formátování disku. Instalace v režimu OEM je vynikající nápad hodný převzetí i do ostatních distribucí. Škoda jen, že první start do KDE (a každý do LiveCD) kazí chybová hláška, která nemá zjevnou příčinu. Shrnuto a podtrženo, při instalaci Kubuntu 6.10 jsem nepotkal žádné závažné nedostatky a alespoň při instalaci vše fungovalo, jak jsem očekával. Z tohoto hlediska je Kubuntu dobrá volba i pro začátečníka. O to více zamrzí vesměs kosmetické vady, které nakonec celkový dojem mohou poněkud pokazit.

Poznámka: Článek kriticky hodnotí proces instalace a není tedy úplným návodem k instalaci Kubuntu, tím méně může nahradit dokumentaci. ■

Odkazy

- [1] <http://www.abclinuxu.cz/software/pracovni-prostredi/spravci-oken/compiz>
- [2] <http://www.abclinuxu.cz/software/system/virtualizace/xen>
- [3] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>

- [4] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [5] <http://www.abclinuxu.cz/software/kancelar/baliky/openoffice.org>
- [6] <http://www.abclinuxu.cz/software/grafika/bitmapy/gimp>
- [7] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>
- [8] <http://www.abclinuxu.cz/zpravicky/firefox-2-oficialne-vydan>
- [9] <http://www.abclinuxu.cz/software/internet/www/konqueror>
- [10] <http://www.abclinuxu.cz/software/system/balicky/synaptic>
- [11] <http://www.abclinuxu.cz/slovník/zrcadlo>
- [12] <http://www.abclinuxu.cz/slovník/zavadec>
- [13] <http://www.abclinuxu.cz/slovník/mbr>
- [14] <http://shots.osdir.com/index.php?distro=451>
- [15] <http://www.abclinuxu.cz/slovník/boot>
- [16] <http://www.abclinuxu.cz/software/nastroje/knetattach>
- [17] <http://www.abclinuxu.cz/slovník/root>
- [18] <http://www.abclinuxu.cz/software/system/disk/qtparted>
- [19] <http://www.abclinuxu.cz/software/system/boot/grub>
- [20] <http://www.abclinuxu.cz/slovník/bios>
- [21] <http://www.abclinuxu.cz/slovník/repozitar>
- [22] <http://www.abclinuxu.cz/clanky/recenze/ubuntu-5.10-breezy-badger>

PIM pro GNU/Linux – 5 (Haystack)

Další díl seriálu o PIM aplikacích se nese opět v duchu pohledu do daleké budoucnosti. Právě tam totiž míří projekt Haystack, který nemá za zády nikoho menšího, než světově proslulou univerzitu MIT. A slibuje podobnou revoluci v PIM oblasti, o jakou se pokouší Chandler.

Lukáš Jelínek

Váš smysl informací

Haystack je představován jako projekt, který má za cíl umožnit lidem, aby spravovali své informace způsobem, který pro ně má největší smysl. Má odbourat bariéry, které staví vývojáři aplikací tím, že předdefinovávají druhy informací a způsob manipulace s nimi.

Celý projekt Haystack se skládá z několika menších podprojektů, řešících součásti celého systému. Prvním z nich je univerzální informační klient (*Haystack client*). Má umožnit správu osobních informací zcela individuálním způsobem, aby uživatel rychle našel přesně to, co hledá. Z hlediska PIM je důležité právě toto, nicméně se stejně zmíním i o dalších komponentách projektu.

Druhou součástí je *Relo*. Zaměřuje se na rychlou orientaci uživatele ve velkých objemech informací, speciálně při vývoji aplikačních programů, a využívá vztahy mezi objekty a redukci naráz poskytovaných informací.

Dalším subprojektem je *Piggy Bank*, což je sémantický webový prohlížeč (realizovaný formou pluginu do *Firefoxu* [1]). Extrahuje informace z *www* stránek (které musí samozřejmě dodržovat sémantická pravidla podle doporučení W3C) a vytváří z nich RDF bázi. K informacím pak poskytuje přístup přes rozhraní usnadňující práci. Poslední součást projektu má název *Re:Search*. Jedná se opět o plugin do *Firefoxu* a slouží ke zefektivnění vyhledávání. Principem je opakované využití dřívějších výsledků vyhledávání a jejich začlenění do výsledků nových.

Technická poznámka: Haystack client existuje ve dvou verzích. Původní, údajně plně funkční, bohužel není přístupná (i když se všechno tváří, že by měla být) a nepodařilo se mi v laboratoři CSAIL MIT najít nikoho, kdo by to byl schopen vyřešit. Lze stáhnout pouze novější, refaktorovanou verzi, která je ovšem pouze velice omezeně funkční a často padá. Následující text se proto z většiny opírá o informace po-

skytované na webu projektu a o nepřiliš důkladnou analýzu kódu jednotlivých pluginů.

Koncepce programu

Klient je vybudován jako aplikace na platformě *Eclipse* [2]. Jednotlivé součásti (moduly) jsou implementovány jako pluginy Eclipse. Datový model využívá RDF, což poskytuje výhodu snadné rozšiřitelnosti existujících modulů a přidávání nových. Uživatelské rozhraní je společné všem modulům a zajišťuje konzistentní ovládání celého programu. Důležité také je, že se nerozlišují informace podle druhu (např. poštovní zprávy, webové stránky, dokumenty, obrázky apod.).

Zcela zásadní je takový přístup, že pro každý objekt (ať už je to kus textu, obrázek, zvuk atd.) je vždy k dispozici celé spektrum operací, které mají pro tento objekt smysl. Například u jakéhokoli textu lze kdykoli zkontrolovat pravopis, u obrázku provádět různé manipulace a úpravy (bez ohledu na to, zda se tento obrázek nachází na disku, v poštovní zprávě nebo třeba v úkolu) apod. Kde je to jen trochu možné, lze k objektům přidávat vlastnosti, vytvářet mezi nimi vazby atd. Samozřejmostí je sjednocení e-mailové a IM komunikace.

Ovládání je založeno na přetahování (drag & drop), kontextových nabídkách (přes pravé tlačítko myši) a jednoduše řešené navigaci. Různé druhy informací mají své rozbalovatelné a zavíratelné bloky. Další bloky obsahují např. doporučené operace, prostředky pro organizaci informací, anotace, kolekce, oblíbené položky, startovní body (základní nabídka běžných činností) a různé další věci.

Každý informační objekt je identifikován svým URI (URN uvnitř, resp. URL vně aplikace), který lze například uložit někde mimo a později ho použít při hledání informace. Podobně můžeme každý objekt vyexportovat do RDF formátu.

Přehled funkcí

Vzhledem k tomu, že jde více o zkoušení koncepce než o aplikaci k ostrému nasazení, nemá příliš smysl řešit detaily jednotlivých funkcí. Proto to vezmeme spíš přehledově:

- *pošta* – poštovní klient integrovaný v aplikaci s celým spektrem poskytovaných činností. Současná implementace zvládne SMTP, POP3 a IMAP, navíc je klient připraven také pro IM komunikaci.
- *kontakty* – správa kontaktů je řešena v kontextu celé koncepce aplikace. Znamená to, že se u kontaktů pracuje např. s rolmi uživatelů ve vzájemné komunikaci, nebo že se zobrazuje prezenční informace (důležitá pro IM). Haystack podporuje výměnu informací pomocí vCard a CSV dat. O podpoře LDAP apod. mi není nic známo, pravděpodobně není k dispozici.
- *plánování* – je důležitou součástí klienta Haystack. Kromě klasických událostí a úkolů, které najdeme prakticky ve všech PIM aplikacích, zde figurují také specifické věci – například připomínání důležitých informací (nezapomenout si přečíst zprávu, prohlédnout obrázek, někomu odpovědět apod.) nebo naopak připojování objektů k úkolům.
- *kanály* – nejen RSS, ale i některé jiné kanály (např. počasí) jsou rovněž integrální složkou aplikace. I na informace odtud se vztahují rozsáhlé operační možnosti programu.
- *digitální knihovna* – v uložených informacích všeho druhu lze vyhledávat jak obvyklými metodami, tak pomocí dynamických hierarchií (postupného přesňování dotazu).

Výhody a nevýhody

I když se jedná o poměrně specifickou záležitost, stejně uvedu výhody a nevýhody. Berte to prosím jen jako velice orientační, příští verze může leccos změnit. Tedy výhody:

- *Přenositelnost*. Založení na konceptu RDF dává dobré předpoklady k přenositelnosti bez větších překážek. Také samotná aplikace, díky využití Eclipse, lze spouštět všude tam, kde je Eclipse k dispozici.
- *Modularita*. Výše uvedené výhody se projevují i tady. Realizace v podobě eclipsovských pluginů je již ve své podstatě modulární, totéž se týká

i dat, která jdou do celkového modelu snadno začlenit.

- *Koncepce programu*. Podobně jako Chandler, i tento projekt si klade za cíl "převrátit přístup naruby". Ve výsledku může podstatně zefektivnit práci, i když je to samozřejmě hodně subjektivní a někomu mohou víc vyhovovat klasické programy.
- *Přehledné GUI*. Uživatelské rozhraní působí velice přehledně a srozumitelně. Mimochodem je znát, že někteří tvůrci jiných aplikací se zde pravděpodobně inspirovali (nebo přinejmenším došli k témuž) a styl GUI použitý v Haystacku se objevuje i v jiných programech.

Nevýhody tu jsou samozřejmě také:

- *Praktická nepoužitelnost*. Jak jsem již uvedl v technické poznámce, s Haystackem je to vcelku podivné. Stará verze není veřejně dostupná a nová skoro nefunguje. Za této situace nelze Haystack používat ani experimentálně.
- *Pomalost a vysoké nároky*. Jako všechny programy založené na Eclipse, i Haystack je náročný na paměť i procesor. A v současné podobě je skutečně velice pomalý.
- *Synchronizace chybí*. Možnost synchronizace, ať už s přenosnými přístroji nebo s nějakým serverem zřejmě zcela chybí. Pokud by tento stav přetrvával, byl by to docela problém.
- *Bez groupwarové podpory*. Stejný problém. Haystack umí jen něco málo ze základů skupinové práce, ale kooperace prostřednictvím groupwarových serverů zřejmě obecně podporována není.

Závěr

Je škoda, že tak nadějný projekt nedává více možností k vyzkoušení a případnému lepšímu začlenění do praxe. Celé to na mě působí tak, že lidé, kteří na Haystack klientu pracovali, zřejmě odešli a projekt stagnuje. Nasvědčuje tomu i komunikace (resp. „nekomunikace“) ze strany CSAIL MIT, a také to, že web projektu byl naposledy aktualizován před dvěma lety. Pokud by se práce opět rozběhly, mohli bychom se dočkat velice slušného PIM řešení. Případně by mohl někdo převzít buď celý projekt nebo aspoň jeho základ (a vybudovat novou aplikaci).

Kdo navštíví stránky projektu, kromě popisu celé koncepce tam najde také fiktivní příběh jedné uživatelky pracující s Haystack klientem, doplněný od-

povídajícími screenshoty programu. O vlastnostech aplikace to řekne víc než sebedelší text.

Toto sice měl být poslední díl, ale nakonec je vše jinak. Ze zprávičky jsem se mezitím dozvěděl, že byl oživen vývoj PIM programu Mulberry. I když nejde

o svobodný (a dokonce ani open-source) program, díky svým vlastnostem by určitě mohl být pro některé uživatele zajímavý. Proto mu bude příští díl věnován.

■

Odkazy

[1] <http://www.abclinuxu.cz/software/internet/www/mozilla-firefox>

[2] <http://www.abclinuxu.cz/software/programovani/ide/eclipse>

PIM pro GNU/Linux – 6 (Mulberry)

I když se jedná o proprietární program, má mezi PIM aplikacemi na GNU/Linuxu své místo. Tím spíš, že po nějaké době útlumu (způsobeném bankrotem firmy, která ho vyvíjela) byl tento projekt oživen a již vzniklo několik nových verzí.

Lukáš Jelínek

K čemu další program?

Někomu by se mohlo zdát, že představovat další PIM program, a k tomu ještě proprietární, je zcela zbytečné. Tento názor nesdílím, Mulberry má – kromě toho, že se jedná o dlouhodobě existující aplikaci – řadu funkcí, kterými jiné programy nedisponují. Navíc patří mezi produkty, které jsou vyvíjeny pro více platforem současně (zde kromě GNU/Linuxu také pro MS Windows a Mac OS X), proto najde uplatnění v heterogenním prostředí.

Historie programu začíná v roce 1995, kdy Cyrus Daboo zahájil jeho vývoj (tehdy jen jako poštovního klienta) během svého působení na University of Cambridge. Po svém odchodu z univerzity ve vývoji pokračoval (hlavně proto, že o program měla univerzita zájem) a v srpnu 1996 byla dokončena první verze pro Mac OS. V následujících letech vznikaly (v rámci firmy Cyrusoft International) další verze s přibývajícím funkcemi. První linuxová verze byla vydána v roce 2001. Verze 4 (znamenající přerod v plnohodnotnou PIM aplikaci) vznikla v květnu '05.

Narozdíl od některých jiných podobných aplikací není program Mulberry založen ani na rozhraní Qt [1], ani na GTK+ [2], a už vůbec není propojen s nějakým desktopovým prostředím. To je na jedné straně výhoda – nejsou problémy se závislostmi – ale i nevýhoda, protože uživatelské rozhraní se dá považovat za nevhledné a částečně problematické (příliš velké fonty bez možnosti zmenšení apod). Působí to dojmem, že linuxová verze je tu „jen tak mimochodem, aby se neřeklo“.

Instalační balík lze stáhnout ze stránek projektu. Jde o běžný tarball komprimovaný jako GZIP, obsahující jednak spustitelný soubor aplikace, a dále pak adresář určený k nakopírování do domovského adresáře uživatele nebo na některé ze systémových míst (např. `/usr/lib/mulberry`). V tomto adresáři jsou pluginy (v podobě dynamických knihoven) pro různé funkce, dále XML soubory s texty, a konečně také dvě ikony (stejný obrázek ve dvou formátech).

Zastavme se nejdřív u pluginů. Zajišťují především bezpečnostní funkce (autentizační metody, SSL, S/MIME apod.), konverze dat a kontrolu pravopisu (založenou na [Aspell](#) [3] a [ispell](#) [4]). Rozhraní pluginů však není veřejně dostupné, proto ani nejsou k dispozici jiné pluginy, než ty obsažené v základním balíku. Škoda.

Trochu jiná je situace u souborů s textovými řetězci. Protože jsou to běžné XML soubory, není problém si je upravit – například přeložit do jiného jazyka. Sice jsem nikde žádné takové překlady nenašel (nejspíš kvůli licenci, která obecně nedovoluje žádné změny), ale technicky tomu nic nebrání.

IMAP nade vše

Program Mulberry je výrazně zaměřen na práci prostřednictvím protokolu IMAP (byť POP3 a lokální úložiště zvládá také). S tím souvisí i to, že podporuje – poměrně málo rozšířené – protokoly ACAP a IMSP. Základem je on-line práce s IMAP, ale lze pracovat i off-line. Pro filtraci pošty lze používat technologii Sieve.

Co se týká kontaktů, kromě místních adresářů lze používat i vzdálené. A to jak LDAP, tak již zmíněné ACAP a IMSP. Místních i vzdálených adresářů může být libovolný počet. Ovšem pozor, i základní místní adresář se musí vytvořit ručně, program si sám žádný nevytváří.

Podobná situace platí i u kalendářů. Opět může být libovolný počet místních i vzdálených, pro komunikaci se používají protokoly WebDAV a CalDAV (případně obyčejný HTTP protokol – pouhé kopírování dat bez synchronizace). Kalendářová data jsou uložena ve formátu iCalendar. Opět se ale musí kalendáře vytvářet ručně, což může být matoucí (zobrazí se časový přehled, ale data do něj vložit nejdou).

Přehled funkcí

Jako vždy, i nyní bude následovat přehled toho, co program všechno umí. Něco jsem již naznačil, tady to bude trochu úplnější.

- *pošta* – Jak již bylo řečeno, klient umí IMAP, POP3 a lokální schránku. Ovšem jeho schopnosti jsou mnohem větší. Má velmi zajímavé možnosti v nakládání se zprávami (zobrazování smazaných zpráv, přesměrování a odmítnutí, volby ohledně ukládání do složek, použití DSN, třídění a filtrace, pokročilé použití MIME apod.). Málokterý poštovní klient má tolik funkcí, z těch fungujících na GNU/Linuxu nejspíš žádný, alespoň ne v základní podobě.
- *kontakty* – U kontaktů se ukládá (v porovnání s jinými podobnými programy) relativně málo informací – v podstatě pouze základní údaje a uživatelské poznámky. Na druhou stranu může být adresářů více, jak lokálních, tak vzdálených. Množina funkcí není tak rozsáhlá jako u pošty, ale všechno potřebné je k dispozici.
- *kalendáře* – Na rozdíl od kontaktů, tady se naopak ukládá velmi mnoho informací. Nastavování parametrů událostí a úkolů je poměrně složité a nepřehledné, vyřadit se mohou cestovatelé zvyklí pracovat současně s různými časovými zónami (ovšem na letní čas se jaksí pozapomnělo). Pohledy na časové úseky (den až rok) jsou široce nastavitelné. Týmová práce je podpořena rozesíláním a zpracováním pozvánek účastníkům (iMIP).
- *konfigurace* – Používání protokolů IMSP a ACAP umožňuje ukládat na server nejen kontaktní informace, ale také nastavení programu. Uživatel si tak může uložit veškerá svá nastavení aplikace Mulberry (samozřejmě s výjimkou nastavení serveru, kam se má připojit) a mít je k dispozici vždy, když s programem pracuje – ať už sedí u kteréhokoli počítače.

Výhody a nevýhody

U nyní popisovaného programu to bude poměrně vyhraněné. Málo vlastností je u něj průměrných, posuďte sami. Nejprve tedy výhody:

- *Přenositelnost*. Mulberry je k dispozici pro tři časté platformy (bývala dříve i verze pro Solaris, ale ta už se někam vytratila). Ovšem na všech třech platformách se program chová stejně a používá stejná data – kombinace různých systémů není problém.
- *Stabilita*. Základy programu jsou dlouhodobě vyzkoušeny a aplikace netrpí žádnými dětskými nemocemi. Dá se říci, že když si člověk přečte příručku, může program bez problémů používat a nebudou ho čekat takové záludnosti, jako je nepredikovatelné chování a záhadné pády.
- *Kvalitní GUI*. Uživatelské rozhraní je logicky navrženo a příjemně se používá. Kdyby tak netrpělo nízkou kvalitou použitých widgetů, byla práce s ním efektivnější než u kteréhokoli jiného podobného programu.
- *Velký rozsah funkcí*. Již jsem zmínil, jak obrovskou škálu funkcí zde najdeme, zejména v oblasti e-mailové komunikace. Prakticky na co si člověk vzpomene, to tam najde.
- *Široká konfigurovatelnost*. Skoro všechno se dá všelijak nastavovat a parametrizovat. A tím, že se dá nastavení navíc uložit na server, se to ještě o řád zlepšuje.
- *Nenáročnost*. Založení programu jen na knihovnách X (s vyloučením náročných toolkitů) se příznivě projevuje na výkonnostních nárocích. Z dosud popisovaných PIM aplikací běhá jednoznačně nejrychleji a s nejmenší spotřebou paměti. Ani na pomalých strojích s menší pamětí nebude mít problém s hladkým během.

Bohužel, o nevýhodách jsem zatím moc nemluvil, o to horší to bude teď:

- *Problémy s českými znaky*. Jediné místo, kde čeština bez problémů funguje, jsou příchozí zprávy. Ve všech ostatních případech je to tragédie. Pravděpodobně to souvisí s tím, že systém používá kódování UTF-8 a Mulberry se s tím nedokáže vyrovnat. České znaky se mi tam nepodařilo dostat, ať jsem se snažil jakkoli.
- *Ošklivé GUI*. Uživatelské rozhraní je sice dobře navrženo (a pod Windows v podstatě i hezky vypadá), ale jeho vzhled v GNU/Linuxu je přímo otřesný. Na vině jsou hlavně obrovské fonty, které není jak zmenšit a které často přetékají mimo prostor pro ně určený. Taktéž některé barvy (rovněž neměnitelné) působí velice neesteticky. Tyto neduhy by možná byly řešitelné ručním nastavením X resources, ale kdo se s tím má lopotit...
- *Na přenosná zařízení zapomeňte*. Podpora PDA, telefonů apod. je – stručně řečeno – nulová. Na tak propracovaný program je to trochu překvapivé a je to rozhodně škoda.
- *Groupware s otazníkem*. Mulberry si zakládá na protokolech IMSP a ACAP. Není divu, když autorem nejrozšířenějších serverů (pokud lze vů-

bec o nějakém rozšíření hovořit) pro tyto protokoly je autor aplikace Mulberry.

Jinak je nutno se spolehnout pouze na protokoly jako LDAP a CalDAV, žádnou specifickou podporu pro groupwarové servery zde nenajdeme.

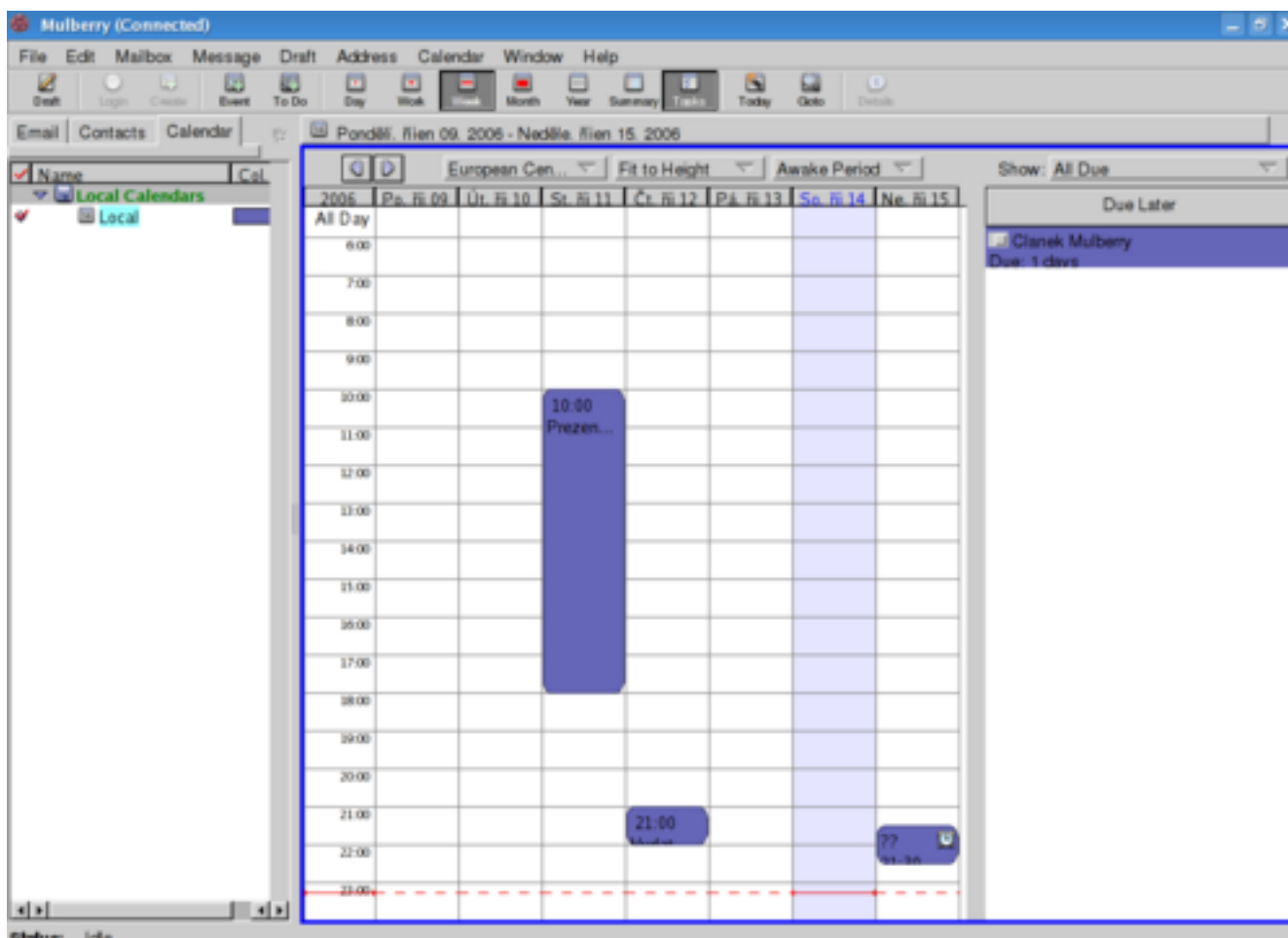
- *Proprietární program.* Původně to byl komerční placený produkt. Nyní už je sice zdarma, ale stále je proprietární. To by ještě tolik nevadilo, kdyby bylo aspoň dostupné rozhraní pro pluginy. V tuto chvíli není možné přidávat do programu mnohé funkce (třeba pro přenosná zařízení), které by se velice hodily.

Závěr

Rád bych prohlásil, že jsem našel vynikající program. Dokonce bych to v tomto případě téměř mohl udě-

lat. Ale bohužel nemohu, aplikace bez korektního zacházení s národními znaky je pro běžnou práci absolutně nepoužitelná. Bude-li vyřešeno toto, pak bude Mulberry na dobré cestě stát se na GNU/Linuxu dobrou alternativou k programům jako Evolution a Kontakt. Zvláště, pokud by bylo zveřejněno rozhraní pro pluginy nebo nejlépe změněna licence na některou svobodnou.

Seriál o PIM aplikacích je u konce. Doufám, že nebyl pouze nudným výčtem několika málo programů, ale že někomu pomohl porovnat možnosti jednotlivých aplikací a vybrat si tu nejvhodnější. Případně, že ukázal i některé směry, kterými se PIM řešení mohou ubírat nebo dokonce ubírají.



Odkazy

- [1] <http://www.abclinuxu.cz/software/programovani/knihovny/qt>
- [2] <http://www.abclinuxu.cz/software/programovani/knihovny/gtkp>
- [3] <http://www.abclinuxu.cz/software/kancelar/pravopis/aspell>
- [4] <http://www.abclinuxu.cz/software/kancelar/pravopis/ispell>

Ako na preklad slobodných programov

Máte vo svojom okruhu ľudí, ktorí by radi používali Linux, ale programy, ktoré používajú, nie sú preložené do slovenčiny alebo češtiny? Chceli ste niekedy prispieť do nejakého slobodného programu, ale neviete programovať? V nasledujúcom dvojdielnom seriáli sa oboznámite so spôsobom prekladania slobodných programov.

Stanislav Valasek

Ciele:

1. Prilákať nových prekladateľov a čo najjednoduchšie popísať postup prekladu slobodného softvéru.
2. Na praktickom príklade ukázať preklad konkrétneho programu.

Úvod

Mnohé slobodné programy používajú na prekladanie rozhrania nástroje GNU `gettext` [1], ktoré celý preklad zjednodušujú na preklad hlášok v textových súboroch (s príponami `.po/.pot`). Tieto súbory sa dajú prekladať v ľubovoľnom textovom editore.

Prácu prekladateľa však môžu veľmi uľahčiť grafické programy, ktoré zjednodušujú samotný preklad a výsledok dokážu jednoducho uložiť do výsledného binárneho súboru (s príponou `.mo`). Tento je už priamo využiteľný prekladaným programom (stačí preklad prekopírovať do príslušného adresára).

Rozhranie slobodných programov, ako sú napríklad dialógy alebo textové hlášky v príkazovom riadku, je najčastejšie v anglickom jazyku. Ak nie, je angličtina veľmi často prvý jazyk, do ktorého je daný program preložený. Ak chcete začať s prekladmi, odporúčam veľmi dobrú znalosť anglického jazyka.

Pár dôležitých pojmov na úvod

- *Internationalisation* – *internacionalizácia* / *i18n* – označuje proces toho, že je program schopný podporovať viacero jazykov. To znamená, že program nie je viazaný na konkrétny jazyk.
- *Localization* – *lokalizácia* / *l10n* – označuje proces toho, že takto pripravenému programu dodáme potrebné informácie na to, aby bol schopný komunikovať v natívnom jazyku, a zohľadňoval kultúrne zvyklosti danej krajiny a jazyka. Toto je možné viacerými spôsobmi. Najčastej-

šie používaným z nich je nižšie popísaný GNU `gettext`.

Ak to stručne zhrnieme, tak internacionalizáciu riešia programátori a lokalizáciu prekladatelia. Je nutné si uvedomiť, že internacionalizácia programu zďaleka neznamená iba preloženie textov v jeho rozhraní. Niektoré textové hlášky sú napísané priamo v skriptoch a iných častiach programu, ktoré nie sú priamo preložiteľné pomocou nástrojov GNU `gettext`. Skoro všetky programy majú možnosti a voľby, ako napríklad Y/N (Yes/No), ktoré tiež môžu byť preložené. Preložené môžu byť aj manuály a dokumentácia, ktorá je k programu priložená, čo je často samostatnou úlohou. Všeobecne sa dá povedať, že lokalizácia rieši nasledovné oblasti:

- *Znaky a znakové sady* – najpoužívanejšia znaková sada v anglicky hovoriacich krajinách je ASCII. Znaková sada ISO 8859-1 pokrýva väčšinu hlavných európskych jazykov, okrem strednej a východnej Európy. Znaková sada ISO 8859-2 obsahuje znaky zo stredoslovanských jazykov, takže aj slovenčinu a češtinu. Naše jazyky obsahuje i znaková sada UTF-8.
- *Meny* – pri narábaní a zobrazovaní peňažných jednotiek sa musí korektne zobrazovať i správna mena danej krajiny.
- *Dátumy* – ich zápis sa môže líšiť podľa krajiny, napríklad 24. 12. 2006 sa v USA zapíše ako 12/25/06 a v Austrálii ako 25/12/06.
- *Čísla* – reprezentácia čísla sa tiež líši podľa krajiny
 - 12,345.67 – Británia
 - 12.345,67 – Nemecko
 - 1,2345.67 – Ázia
- Netreba zabúdať ani na systém jednotiek, napríklad metrický v porovnaní s imperiálnymi jednotkami.

- *Textové správy* – najjasnejšia oblasť lokalizácie.

Ako si uľahčiť prácu?

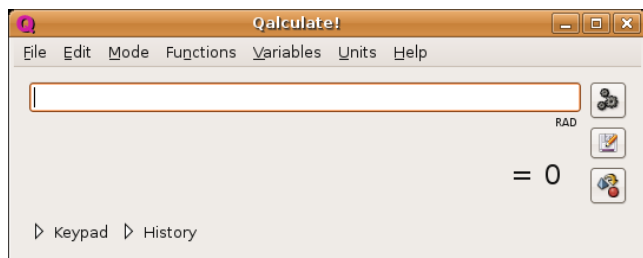
Prácu prekladateľa dokážu výrazne zjednodušiť grafické nástroje umožňujúce pohodlnejšiu prácu s .po, .pot a prípadne aj .mo súbormi. Medzi pravdepodobne najznámejšie programy patria nasledujúce:

- **GTranslator** [2] – po editor pre GNOME prostredie.
- **KBabel** [3] – po editor pre KDE prostredie.
- **POEdit** [4] – multiplatformový po editor.
- Emacs po mód – pre pokročilejších prekladateľov.

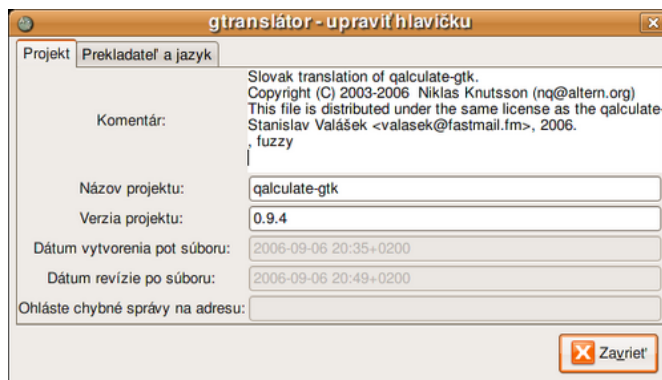
Ako ukážkový program som vybral kalkulačku **Qalculate!** [5]. Program môžete stiahnuť priamo, ale je veľká šanca, že už bude súčasťou Vašej distribúcie. V mojej aktuálnej distribúcii Ubuntu Dapper už bola distribuovaná verzia 0.9-2-1. Inštalácia bola tým pádom relatívne jednoduchá:

```
sudo aptitude install qalculate-gtk
```

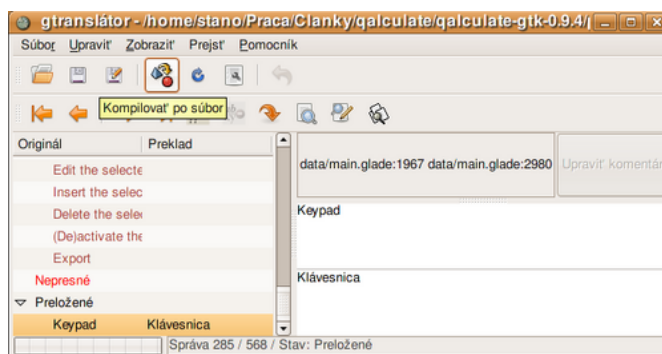
(Pre rpm balík: `rpm -U qalculate-gtk.rpm`). Po prvom spustení program vyzerá nasledovne:



Napriek tomu, že mám nastavené slovenské lokálne prostredie, program komunikuje po anglicky. Najprv sa pozrieme, aké jazyky sú v aktuálnej verzii programu podporované. Zo [stránky projektu](#) [6] stiahneme zdrojovú verziu programu a zistíme, že v adresári `qalculate-gtk-0.9.4/po` existuje lokalizácia pre švédčinu – `sv.po` a nórčinu – `nl.po`. Prázdna jazyková šablóna `sk.po` sa generuje z .pot súboru, ktorý získame buď priamo z distribúcie, alebo môžeme požiadať autora projektu o vygenerovanie aktuálnej verzie .pot súboru. Ďalej predpokladám, že súbor `sk.po` máme, takže sa poďme pustiť do reálnej prekladateľskej práce. V ďalšom texte bude paralelne ukázaný postup najčastejšie vykonávaných činností v po editoroch GTranslator a KBabel. Pri prvom otvorení prázdnej jazykovej šablóny nás GTranslator vyzve na doplnenie dát o preklade, ktoré doplníme nasledovne:



Aby sme vyskúšali, ako „hladko“ funguje prekladanie a test výsledku, preložíme si texty priamo v úvodnom dialógu: „Keypad“, „History“ a text tooltipu „Convert units in result“. Následne vygenerujeme výslednú – binárnu verziu prekladu. Preklad jedného z textov a generovanie binárneho súboru je znázornené tu:

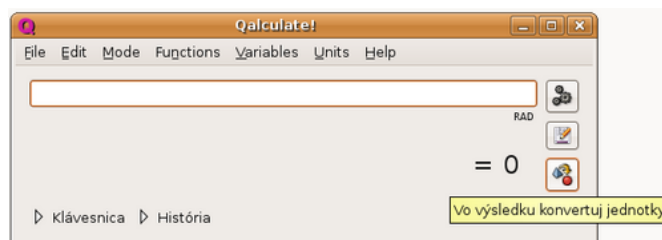


A čo s vygenerovaným súborom `qalculate-gtk.mo`? Stačí ho nakopírovať do adresára pre binárne verzie správ. Na Debian GNU/Linux systémoch je to adresár `/usr/share/locale/xx/LC_MESSAGES/`, kde `xx` je kód jazyka, pre slovenčinu „sk“ a češtinu „cs“. To do cieľme príkazmi:

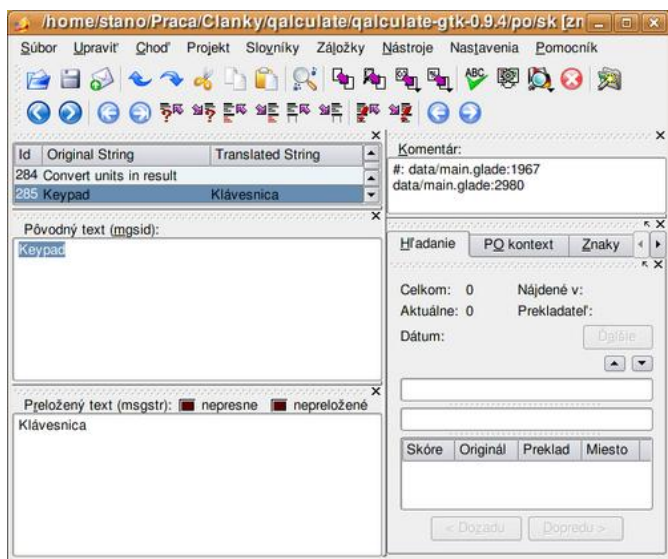
```
sudo cp qalculate-gtk.mo \\  
/usr/share/locale/sk/LC_MESSAGES/  
#kópia do adresára pre slovenské preklady  
  
qalculate-gtk
```

A výsledok je nasledujúci (obrázok).

Je veľmi vhodné preklad overovať, či už priebežne, ak si nie sme istí správnym preložením konkrétnej hlášky, alebo až na konci prekladu, kde je overenie výsledného prekladu nutnosťou.



Preklad jednej správy v programe KBabel je znázornený na obrazovke nižšie:



Nezistil som ako sa v programe KBabel generuje výsledný binárny súbor; mám podozrenie, že to nepodporuje. Binárny preklad však jednoducho získame zo súboru `sk.po` pomocou príkazu:

```
msgfmt -v -o ./qalculate-gtk.mo ./sk.po
```

Sumarizácia

Na záver krátka rekapitulácia jednotlivých krokov, ktoré budeme musieť spraviť, aby sme preložili konkrétny program a sprístupnili preklad ostatným používateľom:

1. *Oznámte svoj záujem o preklad pôvodnému prekladateľovi alebo lokalizačnému tímu.*
2. *Stiahnite a nainštalujte si potrebné nástroje, minimálne GNU gettext (ktorý je často už súčasťou distribúcie) a prípadne niektorý z vyššie spomínaných po editorov.*
3. *Získajte najnovšiu verziu anglického po súboru pre prekladaný jazyk, alebo si stiahnite .pot šablónu. Ak chcete aktualizovať preklad, stiahnite po súbor pre váš jazyk. Nastavte korektné údaje o vás, prekladateľskom tíme, cieľovom jazyku a požadovanom kódovaní. Najčastejšie používané sú UTF-8 alebo ISO 8859-2.*
4. *Preložte, čo ide...*
5. *Spätne skontrolujte preklad.*
6. *Otestujte váš preklad – vytvorením .mo súboru, jeho skopírovaním do adresára `LC_MESSAGES` pre váš jazyk a spustením aplikácie.*
7. *Zašlite preklad späť buď pôvodnému prekladateľovi, alebo priamo potvrdením zmien do ver-*

zovacieho nástroja – najčastejšie [CVS](#) [7] alebo [Subversion](#) [8]. Pokiaľ sa zmeny nedostanú do vašej distribúcie, môžete pokojne používať vami preloženú verziu programu.

8. *Nezabudnite na pravidelné aktualizácie prekladu po vydaní každej novej verzie programu. Najčastejšie sú .po súbory priebežne aktualizované tak, ako programátori pridávajú alebo menia texty a hlášky programu. Nevýhodou je, že niekedy nie je možnosť inštalácie a otestovania takto vytvoreného prekladu.*

Preklad pomocou nástrojov GNU gettext

Ak nemáme alebo nechceme používať na preklad grafické programy, môžeme prekladať aj pomocou textového editora a nástrojov [GNU gettext](#) [9].

Národné prostredie

Pred samotným prekladom ešte pár slov k nastaveniu národného prostredia. V distribúciách založených na Debian GNU/Linux zistíme aktuálne nastavené národné prostredie príkazom `locale`:

```
stano@nb-valasek:~$ locale
LANG=sk_SK.UTF-8
LANGUAGE=sk_SK.UTF-8
LC_CTYPE="sk_SK.UTF-8"
LC_NUMERIC="sk_SK.UTF-8"
LC_TIME="sk_SK.UTF-8"
LC_COLLATE="sk_SK.UTF-8"
LC_MONETARY="sk_SK.UTF-8"
LC_MESSAGES="sk_SK.UTF-8"
LC_PAPER="sk_SK.UTF-8"
LC_NAME="sk_SK.UTF-8"
LC_ADDRESS="sk_SK.UTF-8"
LC_TELEPHONE="sk_SK.UTF-8"
LC_MEASUREMENT="sk_SK.UTF-8"
LC_IDENTIFICATION="sk_SK.UTF-8"
LC_ALL=
```

Nastavený je teda jazyk `sk` a krajina `SK` a kódová stránka na UTF-8, čo je štandardná kódová stránka v Ubuntu systéme. Podrobnejší popis pre Debian GNU/Linux je možné nájsť na stránke [Debian reference – 9.7 Localization \(l10n\)](#) [10].

Ak chceme spustiť program v inom, ako aktuálne nastavenom jazyku, spustíme ho nasledovne:

```
LANG=en1 qalculate-gtk
```

Tento príkaz napríklad spustí program `qalculate-gtk` v nórčine. Prípadne príkaz `LANG=C program` spustí

program v originálnom/anglickom preklade. Takto môžeme súčasne spustiť rovnaký program v rôznych jazykoch.

Rovnako je možné celé prostredie priamo nastaviť do rôznych jazykov. Napríklad príkazom

```
LANGUAGE="sk_SK:sk:cs_CS:cs:de:en_GB:en"
```

v súbore `/etc/environment` spôsobíme, že každý spúšaný program skúsi nájsť slovenčinu, ak ju nenájde, pokračuje druhým, tretím jazykom... Adresár na výsledné preklady sme už spomínali vyššie, takže:

- máme textovú podobu správ pripravených na preloženie,
- vieme postup, ako si preklad priebežne overovať.

Tvorba po súbore

Už hotové preklady nemusia obsahovať všetky textové reťazce, ktoré sú v aktuálnej verzii programu. Preto je lepšie začať vygenerovaním prázdnej jazykovej verzie z pot šablóny príkazom `msginit`. Program `Qalculate` ale neobsahuje pot šablónu, takže si ju vygenerujeme príkazom

```
xgettext -f ./po/POTFILES.in \\  
-o ./po/messages.po
```

Následne je možné použiť príkaz `msginit` alebo premenovať `messages.po` na `sk.po`. Program `msginit` by mal uľahčiť prácu a automaticky nastaviť hlavičky .po súboru podľa vášho prostredia.

Prekladanie po súbore

Keď toto všetko vieme a máme `sk.po` súbor, podme zaktualizovať hlavičky po súbore tak, aby sme sa mohli vrhnúť na samotný preklad:

```
# SOME DESCRIPTIVE TITLE.  
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT \\  
HOLDER  
# This file is distributed under the same \\  
license as the PACKAGE package.  
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.  
#  
#, fuzzy  
msgid ""  
msgstr ""  
"Project-Id-Version: PACKAGE VERSION\n"
```

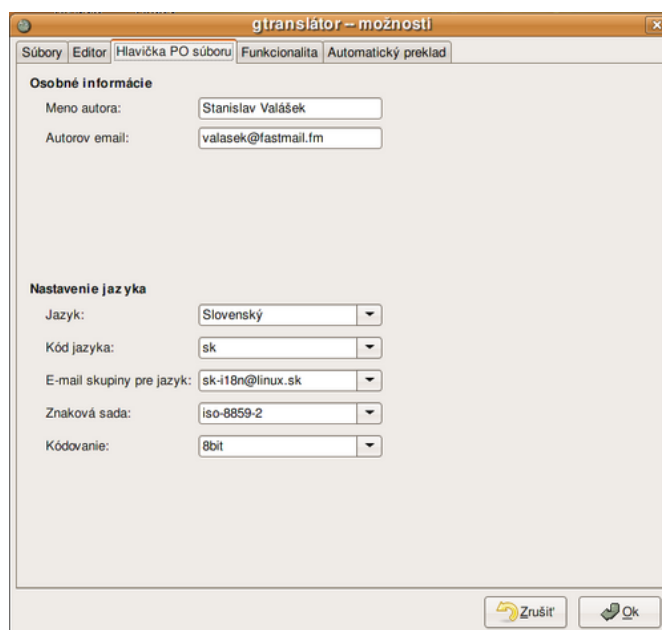
```
"Report-Msgid-Bugs-To: \n"  
"POT-Creation-Date: 2006-08-06 17:43-0400\n"  
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
```

```
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"  
"Language-Team: LANGUAGE <LL@li.org>\n"  
"MIME-Version: 1.0\n"  
"Content-Type: text/plain; charset=CHARSET\n"  
"Content-Transfer-Encoding: 8bit\n"
```

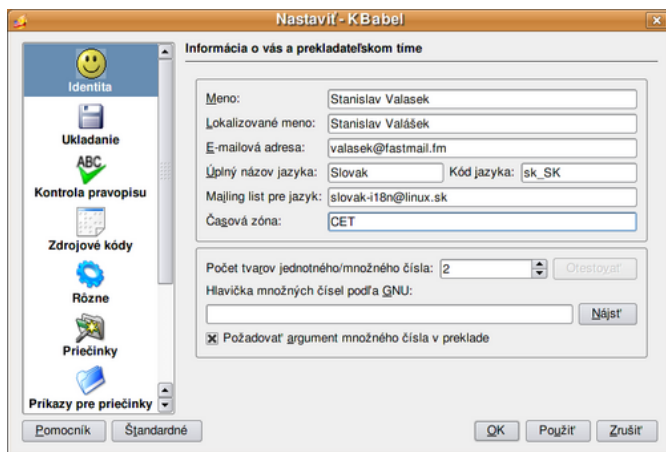
Kompletný preklad hlavičiek:

```
# Translation of qalculate-gtk.po to Slovak  
# This file is distributed under the same \\  
license as the Qalculate package.  
# Copyright (C) 2003-2006 Niklas Knutsson \\  
(nq@altern.org).  
# Stanislav Valasek <valasek@fastmail.fm>, \\  
2006.  
#  
msgid ""  
msgstr ""  
"Project-Id-Version: qalculate-gtk\n"  
"Report-Msgid-Bugs-To: \n"  
"POT-Creation-Date: 2006-08-06 17:43-0400\n"  
"PO-Revision-Date: 2006.05.08 13:33\n"  
"Last-Translator: Stanislav Valasek \\  
<valasek@fastmail.fm>\n"  
"Language-Team: sk <sk-i18n@linux.sk>\n"  
"MIME-Version: 1.0\n"  
"Content-Type: text/plain; charset=UTF-8\n"  
"Content-Transfer-Encoding: 8bit\n"
```

Obsah týchto hlavičiek dokážu programy [GTranslator](#) [11] i [KBabel](#) [12] veľmi pohodlne upraviť podľa vami zadaných dát. V `GTranslator` ich zadávame v nastaveniach:



`KBabel` nás k nastaveniu vyzve pri prvom spustení a je možná ich neskoršia úprava.



Pri uložení sa hlavička po súbore automaticky aktualizuje zadanými údajmi o prekladateľovi a pri každej aktualizácii prekladu v programe GTranslator alebo KBabel sa automaticky aktualizuje dátum poslednej modifikácie. Ešte pred prekladom pár slov k formátu .po súbore, ktorý obsahuje samotné texty v nasledujúcej forme:

```
#: panel/panel_config.c:1050
#, fuzzy
msgid "Color to use:"
msgstr "Použité farby:"
```

Prvý riadok obsahuje názov zdrojového súboru a riadok, kde sa nachádza prekladaný text. Druhý riadok nie je povinný a môže obsahovať komentáre alebo direktívy, ako napríklad direktívu fuzzy alebo c-format. Direktíva pridáva prekladateľ alebo sú pridané automaticky pri generovaní .po súboru. Direktíva fuzzy znamená, že preklad textu nie je istý a treba ho overiť. Takto označené hodnoty sa pri štandardnom generovaní mo súboru nedostanú do binárnej podoby prekladu. Tretí riadok obsahuje pôvodný a štvrtý riadok preložený text. Úplný popis formátu je na stránkach [GNU gettext – The Format of PO Files](#) [13].

Aby sme vyskúšali ako „hladko“ funguje prekladanie a test výsledku, preložíme si texty priamo v úvodnom dialógu: „Keypad“, „History“ a text tooltipu „Convert units in result“. Tieto texty sa nachádzajú v `sk.po` súbore a tu je ich doplnenie o preklad:

```
#: data/main.glade:1886
msgid "Convert units in result"
msgstr "Vo výsledku konvertuj jednotky"

#: data/main.glade:1967 data/main.glade:2980
msgid "Keypad"
msgstr "Klávesnica"
```

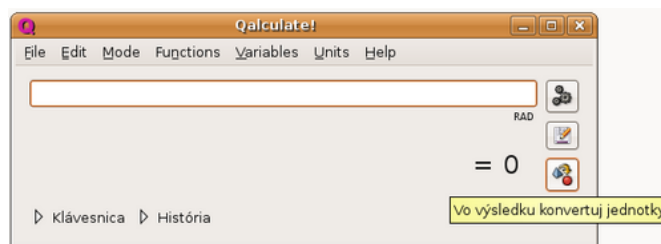
```
#: data/main.glade:2006 data/main.glade:3035
msgid "History"
msgstr "História"
```

Poznámka – na prekladanie používajte textové editory, ktoré dokážu správne určiť a uložiť kódovanie editovaného súboru. Toto spĺňajú najčastejšie používané editory ako: [gedit](#) [14], [Kate](#) [15], [Emacs](#) [16], [vim](#) [17] a mnohé ďalšie.

Následne spustíme už známe príkazy na vygenerovanie prekladu a spustíme program.

```
msgfmt -v -o ./qalculate-gtk.mo ./sk.po
3 preložené správy, 566 nepreložených správ.
stano@nb-valasek:/qalculate-gtk-0.9.4/po$ \
sudo cp qalculate-gtk.mo /usr/share/locale/\
sk/LC_MESSAGES/
stano@nb-valasek:/qalculate-gtk-0.9.4/po$ \
qalculate-gtk
```

A výsledok:



Špeciality prekladu

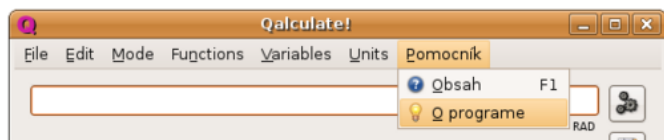
Počas prekladania môžete naraziť na špeciálne formátovanie `msgstr` reťazcov. V dialógovej aplikácii sa stretnete s tzv. akceleračnými klávesami, ktoré určujú, pomocou ktorého znaku v reťazci sa dá daná funkcia zvoliť klávesovou skratkou ALT+klávesa. Je nutné si dávať pozor, aby dva reťazce v jednom menu nemali rovnaké akceleračné klávesy. Malý príklad aj s chybou – rovnaké akceleračné klávesy.

```
#: data/main.glade:1608
msgid "_Help"
msgstr "_Pomocník"
```

```
#: data/main.glade:1617
msgid "_Contents"
msgstr "_Obsah"
```

```
#: data/main.glade:1639
msgid "_About"
msgstr "_O programe"
```

A výsledok:



Na nasledujúcom texte je možné ukázať dve veci:

```
#: data/main.glade:2341
msgid "x<sup>2</sup>"
msgstr "x<sup>2</sup>"
```

1. Texty neobsahujú vždy iba samotný text, ale niekedy i technické informácie. V tomto prípade ide o HTML značky ``, ale môže ísť napríklad i o formátovacie značky programovacieho jazyka, v C++ napríklad `%s` alebo `%d`. Tieto značky sa bezo zmeny prenášajú do preloženého textu. Je dobré oboznámiť sa s najčastejšie používanými značkami, aby ste vedeli, čo robia, a ako ich vhodne začleniť do prekladateľného textu. Tieto špeciálne znaky sú čiastočne popísané v [GNOME I10n-guide – Gotchas](#) [18].

2. Zároveň vidíme, že niektoré reťazce nie je treba prekladať a sú rovnaké ako v anglickej verzii. Tu nám prácu veľmi zjednoduší príkaz `msgen`, ktorý automaticky prekopíruje z `.po` alebo `.pot` súboru všetky anglické reťazce do preložených. Po spustení príkazu `msgen sk.po -o sk-new.po` dostaneme súbor `sk-new.po`, v ktorom máme všetky reťazce z pôvodného prekladu prekopírované do nového prekladu. Nevýhodou je, že stratíme informáciu o tom, koľko reťazcov ešte nie je preložených.

Urýchlenie a zjednodušenie prekladu je možné dosiahnuť tým, že zo všetkých našich `.po` súborov extrahujeme preložené texty do slovníka, ktorý následne používame na prvotný preklad ešte nepreloženého súboru a až potom preklad dokončujeme. Viac informácií je na stránke [GNU gettext – Usage of Translation Compendia](#) [19]. Niektoré prekladateľské tímy, ako napríklad [GNOME](#) [20] a [KDE](#) [21] majú takéto slovníky už pripravené a jednotliví prekladatelia ich len dopĺňajú.

Ostatné druhy prekladov

Okrem vyššie spomenutého postupu prekladu programu pomocou nástrojov GNU gettext sú niektoré slobodné programy prekladané iným spôsobom. Typickým príkladom je preklad balíkov Mozilla suite pomocou špeciálneho programu [Mozilla translator](#) [22], alebo špeciálny postup [prekladu balíka OpenOffice.org](#) [23], kde prekladatelia síce na preklad pou-

žívajú `.po` súbory, ale sú špeciálne generované zo zdrojových kódov.

Samostatnou kapitolou je preklad dokumentácie, kde sa spôsob prekladu líši podľa toho, o akú dokumentáciu ide.

Pár tipov a rád na záver

- Doporučujem začať prekladom malej samostatnej aplikácie, ktorú často používate, a dokážete priamo skontrolovať preklad.
- Ideálne sa prekladajú aplikácie, ktoré dennodenne používate a môžete po sebe drobné chyby v prekladoch postupne odladiť.
- Spravte toľko prekladov, koľko ich dokážete udržiavať v aktuálnom stave. Toto obzvlášť platí pre preklad napríklad webových stránok, ktoré rýchlo zastarávajú.
- Nie je anglický text zrozumiteľný? Napíšte chybovú hlášku.
- Nemeňte riadky s `msgid`.
- Ak chcete spustiť program v inom ako aktuálne nastavenom jazyku, spustite ho nasledovne:

```
LANG=nl qalculate-gtk
```

Tento príkaz napríklad spustí program `qalculate-gtk` v nórčine. Prípadne príkaz `LANG=C program` spustí program v originálnom/anglickom preklade. Takto si prípadne môžete súčasne spustiť rovnaký program v rôznych jazykoch.

- Prípadne je možné si celé prostredie priamo nastaviť do rôznych jazykov. Napríklad príkazom

```
LANGUAGE="sk_SK:sk:cs_CS:cs:de:en_GB:en
```

v súbore `/etc/environment` spôsobíte, že každý spúšťaný program skúsi najst slovenčinu, ak ju nenájde, pokračuje druhým, tretím jazykom...

- Na zobrazenie rozdielov medzi dvoma po súbormi sú vhodné programy: `podiff` [24] alebo grafický program `Meld` [25] alebo `KDiff3` [26].

Lokalizačné projekty a tímy

- Preklad základných balíkov Linuxu:
 - [Preklad GNU balíkov](#) [27]
 - [OpenOffice I10n](#) [28] – lokalizácia OpenOffice balíkov
 - [GNOME translation projekt](#) [29] – lokalizácia desktopového prostredia GNOME

- KDE localisation [30] – lokalizácia desktopového prostredia KDE
- Mozilla l10n [31] – lokalizácia Mozilla balíkov
- Preklad jednotlivých distribúcií:
 - Mandriva [32]
 - Fedora [33]
 - Ubuntu [34]
 - Debian GNU/Linux [35]

Odkazy

- [1] <http://www.abclinuxu.cz/software/nastroje/lokalizace/gettext>
- [2] <http://www.abclinuxu.cz/software/nastroje/lokalizace/gtranslator>
- [3] <http://www.abclinuxu.cz/software/nastroje/lokalizace/kbabel>
- [4] <http://www.abclinuxu.cz/software/nastroje/lokalizace/poedit>
- [5] <http://www.abclinuxu.cz/software/nastroje/kalkulatory/qalculate>
- [6] <http://qalculate.sourceforge.net/downloads.html>
- [7] <http://www.abclinuxu.cz/software/programovani/sprava-verzi/cvs>
- [8] <http://www.abclinuxu.cz/software/programovani/sprava-verzi/subversion>
- [9] <http://www.abclinuxu.cz/software/nastroje/lokalizace/gettext>
- [10] <http://qref.sourceforge.net/quick/ch-tune.en.html#s-l10n>
- [11] <http://www.abclinuxu.cz/software/nastroje/lokalizace/gtranslator>
- [12] <http://www.abclinuxu.cz/software/nastroje/lokalizace/kbabel>
- [13] http://www.gnu.org/software/gettext/manual/html_mono/gettext.html#SEC9
- [14] <http://www.abclinuxu.cz/software/nastroje/textove-editory/gedit>
- [15] <http://www.abclinuxu.cz/software/nastroje/textove-editory/kate>
- [16] <http://www.abclinuxu.cz/software/nastroje/textove-editory/emacs>
- [17] <http://www.abclinuxu.cz/software/nastroje/textove-editory/-g-vim>
- [18] <http://developer.gnome.org/projects/gtp/l10n-guide/#gotchas>
- [19] http://www.gnu.org/software/gettext/manual/html_mono/gettext.html#SEC54
- [20] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/gnome>
- [21] <http://www.abclinuxu.cz/software/pracovni-prostredi/desktop/kde>
- [22] <http://www.mozillatranslator.org/>
- [23] <http://l10n.openoffice.org/>
- [24] <http://www.mail-archive.com/ubuntu-translators@lists.ubuntu.com/msg00022.html>
- [25] <http://www.abclinuxu.cz/software/soubory/porovnani/meld>
- [26] <http://www.abclinuxu.cz/software/soubory/porovnani/kdiff3>
- [27] <http://www.iro.umontreal.ca/translation/>
- [28] <http://l10n.openoffice.org/>
- [29] <http://developer.gnome.org/projects/gtp/>
- [30] <http://i18n.kde.org/>
- [31] <http://www.mozilla.org/projects/l10n/>
- [32] <http://www1.mandrivalinux.com/l10n/links.php3>
- [33] <http://fedoraproject.org/wiki/L10N>
- [34] <https://launchpad.net/rosetta>
- [35] <http://www.debian.org/international/>

VLC – 6 (správce vysílání)

Poslední díl seriálu o přehrávači VLC media player se zaměřuje napoužívání jeho součásti VLM – správce vysílání, který umožňuje streamovat více multimediálních zdrojů zároveň a nabízet službu Video-On-Demand, video na přání.

Jiří Poláček

Knihovny LIVE555 Streaming Media

Pro potřeby dnešního dílu o VLC jsem musel přehrávač zkompilovat znovu, abych získal podporu pro modul *rtp* a následně též služby VOD (Video On Demand – video na přání). Potřebné knihovny, pokud nejsou přímo dostupné z repozitářů naší distribuce, lze získat z webu live555.com [1]; kompilační volby VLC pak doplníme o parametry `enable-livedotcom` a `with-livedotcom-tree=/usr/lib/live` (uvést cesty ke knihovnám silně doporučuji, zdá se, že bez této volby překladač knihovny nenajde, nicméně dál svou práci dokončí a uživatel se pak diví, proč mu program nefunguje). Konkrétně jsem tedy instaloval s následujícími volbami:

```
./configure --enable-dvdrad --enable-dvdradnav\
--enable-dvb --enable-dvbpsi --enable-real\
--enable-aa --enable-ncurses --enable-skins2\
--enable-v4l --enable-xosd --enable-realrtsp\
--enable-livedotcom\
--with-livedotcomtree=/usr/lib/live
```

Poznámka k nadcházející verzi 0.9.0: kromě toho, že se můžeme těšit na opravy chyb, pár vylepšení a počestěné grafické rozhraní, mění se také názvy výše uvedených parametrů na

`enable-live555` a `with-live555-tree`.

Protokoly RTP, SDP a RTSP

Modul *rtp* nabízí směrové (*unicast*) a vícesměrové (*multicast*) vysílání po síti, tj. dělá v podstatě to samé, co modul `standard` za použití protokolu UDP, používá k tomu ovšem pro tyto účely určený protokol RTP (viz RFC 3550 [2]) s jeho vymoženostmi.

Tou hlavní vymožeností, která je v modulu *rtp* dostupná, je vystavování souboru ve formátu SDP (Session Description Protocol, viz RFC 4566 [3]) popisující parametry vysílání. Může vypadat například následovně:

```
v=0
o=- 106166909550 2 IN IP4 127.0.0.1
s=Nazev vysilaneho proudu
```

```
i=Doplňující popisec vysilaneho proudu
u=http://informace-o-filmu.com/
e=reziser@informace-o-filmu.com
t=0 0
a=tool:vlc 0.8.5
c=IN IP4 192.168.0.26
m=video 1232 RTP/AVP 96
a=rtpmap:96 MP4V-ES/90000
a=fmtp:96 profile-level-id=3; \
config=0000012000068400670c4810a0518f;
m=audio 1230 RTP/AVP 14
a=rtpmap:14 MPA/90000
b=AS:128
```

Z položek lze snadno odhadnout, že soubor SDP obsahuje informace o tom, odkud a kam se vysílá (připomínám, že RTP je vystavěno nad UDP), jaké složky proud obsahuje a případně nějaké doplňující popisky. Naladění takto vysílaného multimediálního proudu znamená zadání adresy k souboru SDP uživatelem, zbytek již obstará přehrávač.

Aby těch protokolů nebylo málo, je nyní zapotřebí zmínit se také o RTSP (Real Time Streaming Protocol, viz RFC 2326 [4]), který se používá pro ovládní přenosu multimediálních dat – typicky právě v souvislosti s protokolem RTP. Plná implementace RTSP v důsledku umožňuje přímo v klientském programu vzdáleně pouštět a pozastavovat vysílaný zdroj, skákat z místa na místo apod. jako by byl přehráván z lokálního souboru (samozřejmě za předpokladu, že to povaha vysílaného zdroje umožňuje). V modulu *rtp* však bohužel lze za pomoci RTSP vysílaný zdroj pouze naladit, video na přání je tak nutno nabízet jiným způsobem (viz dále). Pro lepší orientaci v uvedených protokolech doporučuji nahlédnout do Wikipedie: [\[RTP \[5\]\]](#), [\[SDP \[6\]\]](#) a [\[RTSP \[7\]\]](#).

Parametry modulu rtp

O vystavení souboru SDP se stará parametr příznačně pojmenovaný `sdp` a může nabývat obecně čtyř různých typů hodnot:

- `file://<cesta k lokálnímu souboru>`
– soubor SDP se uloží na místní disk
- `http://<webová adresa>` – SDP bude přístupné prostřednictvím vestavěného webového serveru přehrávače
- `rtsp://<internetová adresa>` – principiálně totéž, co předchozí varianta; SDP bude přístupné via protokol RTSP
- `sap` – soubor bude ohlašován pomocí speciální multicastové adresy, viz čtvrtý díl seriálu [8].

Povinným parametrem modulu `rtp` je `dst` pro určení cílové adresy vysílání. Podle dokumentace je výjimkou použití protokolu RTSP pro vystavení souboru SDP – v takovém případě se jako cíl dosadí adresa klienta, který vznesl požadavek, toto chování se mi však nepodařilo ověřit. Lépe to ozřejmí příklady:

```
vlc pisnicka.mp3 --sout \\  
'#rtp{dst=239.255.0.101, \\  
sdp=rtsp://192.168.0.100:8080/pisnicka.sdp}'
```

Uvedeným způsobem vysílanou písničku na multicastovou adresu naladíme následovně:

```
vlc rtsp://192.168.0.100:8080/pisnicka.sdp
```

Následující příklad distribuující SDP přes oznamování SAP ukazuje další parametry modulu `rtp`:

```
vlc video.avi --sout '#rtp{port-video=1430, \\  
port-audio=1230, dst=239.255.0.101, sdp=sap, \\  
name="Název vysílaného proudu", \\  
description="Doplňující popisek \\  
k vysílanému proudu", \\  
url="http://informace-o-filmu.com/", \\  
email="reziser@informace-o-filmu.com"}'
```

Parametry `port-video` a `port-audio` definují sudá čísla portů pro první stopy příslušné složky – případně další zvukové či videostopy budou vysílány na následujících sudých portech; liché porty jsou rezervovány pro ovládání vysílání dané složky (například právě protokolem RTSP). Konečně s pomocí parametrů `name`, `description`, `url` a `email` můžeme vysílaný multimediální zdroj popsat, význam má zejména zadání jména zdroje, které se následně objeví v seznamu skladeb oznamovaných prostřednictvím SAP.

Správce vysílání – VideoLAN Manager

VLM neboli *VideoLAN Manager* je součástí přehrávače VLC, s jejíž pomocí je možné zároveň streamovat více multimediálních zdrojů pomocí jediné

instance VLC a nabídnout službu videa na přání. Správce se ovládá pomocí *telnetového rozhraní* (viz první díl [9]), případně lze použít i webové rozhraní a částečně též klasické grafické rozhraní. Používání VLM poněkud nabourává doposud zažitě zvyky ovládání přehrávače, pojďme si proto nejdříve osvětlit princip. S pomocí VLM definujeme ovládáme tři různé druhy elementů:

- `broadcast` – zdroj, který vysíláme podobně jako televizní či rozhlasové vysílání; pouze administrátor může vysílání spouštět a zastavovat
- `vod` – video na přání, tj. administrátor definuje nabídku zdrojů a klient si pak sám spouští vysílání
- `schedule` – pomocný element, který umožňuje spouštět a zastavovat zdroje definované jako `broadcast` podle zadaného časového rozvrhu

Elementy lze vytvářet a mazat, definovat i zobrazovat jejich nastavení a samozřejmě ovládat. Ukažme si pro začátek, jak s pomocí VLM streamovat totéž, co následujícím příkazem:

```
vlc video.avi --sout \\  
'#standard{access=http, mux=ts, dst=:8087}'
```

V následujících příkazech budeme pro ovládání VLM používat telnetové rozhraní, které si zpřístupníme pomocí následujících dvou příkazů:

```
vlc -I telnet &  
telnet localhost 4212
```

Výchozí heslo je `admin`, pro nastavení telnetového rozhraní viz `vlc -p telnet --advanced`. Po přihlášení můžeme začít pracovat s VLM:

```
new PrvniVysilani broadcast  
setup PrvniVysilani enabled  
setup PrvniVysilani input video.avi  
setup PrvniVysilani output #standard{access=\\  
http, mux=ts, dst=:8087}  
control PrvniVysilani play
```

Za příkazem následuje námi vymyšlené unikátní jméno elementu a upřesňující parametry. Příkaz `new` vytváří nové elementy uvedeného typu (`broadcast`, `vod`, `schedule`), volitelně zde mohou následovat parametry, jež dodatečně upravujeme příkazem `setup`. Kromě parametru `play`, který spouští definovaný zdroj, jsou u příkazu `control` přípustné ještě rozkazy `stop`, `pause` a `seek` s procentuálním vyjádřením, kam se přesunout v rámci aktuálně přehrávaného zdroje.

Další, v příkladu nepoužité, příkazy jsou `del` pro zrušení elementu, `show` pro výpis aktuální konfigurace a dvojice `save & load` pro uložení a nahrání aktuální konfigurace VLM. Zde je vhodné poznamenat, že konfiguraci VLM lze ze souboru nahrát již při startu přehrávače za pomoci parametru `vlm-conf`. Konečně výpis všech příkazů lze získat po zadání `help`.

Parametry zdrojů typu broadcast

Vraťme se k příkazu `setup`, který si zaslouží podrobnější rozbor parametrů. Začneme těmi, které se týkají *médií*, tj. elementů `broadcast` a (částečně) `vod`:

- *input* – definice vstupního zdroje, který se má vysílat; vícenásobné použití tohoto parametru je přístupné – vytváří se tak vlastně seznam skladeb nebo chcete-li playlist
- *output* – definice síťového vysílání, tj. přesně totéž, co hodnota parametru `sout`; pro video na přání se nepoužívá
- *enabled / disabled* – povolení či zakázání elementu; zakázané médium prostě a jednoduše nemůže být přehráváno
- *loop / unloop* – povolení či zakázání přehrávání položek v seznamu skladeb stále dokola
- *option* – slouží k dodefinování libovolného parametru (budeme-li například do vysílaného videa vkládat logo, zadáme cestu k obrázku s pomocí `setup Název option logo-file=logo.png`)

Příklad včetně výpisu příkazu `show` a vymazání vytvořeného elementu:

```
new ukazky broadcast enabled
setup ukazky loop
setup ukazky input dvdsimple:// input \
  trailer.mov
setup option dvd=/dev/dvdrecorder
setup ukazky output #std{access=udp,mux=ts,\
  dst=@239.255.0.101}
control ukazky play
show ukazky
show
  ukazky
    type : broadcast
    enabled : yes
    loop : yes
    inputs
      dvdsimple://
```

```
trailer.mov
output : #std{access=udp,mux=ts,\
  dst=@239.255.0.101}
options
  dvd=/dev/dvdrecorder
instances
  default : playing
control ukazky stop
del ukazky
```

Použití plánovače

U elementu `schedule` lze nastavovat parametry týkající se časového rozvrhu událostí:

- *append* – definuje příkaz, který se má v nastavenou dobu provést; může se jednat o jakýkoliv příkaz VLM, obvykle se spouští nebo zastavuje přehrávání nějakého zdroje

- *date* – stanovuje datum a čas ve tvaru

```
(rok)/(měsíc)/(den)-(hodina):\
  (minuta):(sekunda)
```

definující, kdy se mají příkazy definované pomocí `append` provést; alternativně lze uvést klíčové slůvko `now` pro okamžité spuštění

- *period* – volitelné nastavení prodlevy, po kterou se bude čekat na opětovné spuštění příkazů; prodleva se opět zadává ve tvaru počtu

```
(let)/(měsíců)/(dní)-(hodin):\
  (minut):(sekund)
```

Pozor však na to, že měsíc se počítá jako třicet dní a rok jako dvanáct měsíců

- *repeat* – volitelné nastavení počtu repríz, pokud je nastavena prodleva; bez této volby s nastavením prodlevy by spuštění příkazů probíhalo donekonečna

Příklad plného využití všech možností plánovače:

```
new film broadcast enabled
setup film input film.avi
setup film output #std{access=http,mux=ts,\
  dst=192.168.0.100:8087}
new pust_film schedule enabled
setup pust_film date 2006/12/1-12:00:00
setup pust_film period 0/0/1-0:0:0
setup pust_film repeat 6
setup pust_film append control film play
```

Film bude spuštěn poprvé 1. prosince v poledne a pak ve stejnou dobu po šest následujících dní. Zájemci,

kteří chtějí využít VLC ke streamování a nahrávání pořadů digitálního televizního vysílání, se mohou inspirovat tímto [konfiguračním souborem VLM](#) [10].

Video na přání

Konečně se dostáváme ke slibovanému videu na přání. Z povahy zdroje se jedná o směrové vysílání k uživateli, který si toto vysílání vyžádal a který si jej také řídí za pomoci protokolu RTSP. Uelementů vod nelze zapnout opakování pomocí volby `loop`, nspecifikuje se výstup a nelze použít příkaz `control`; zřízení videa na přání ve VLM se nám tak redukuje na:

```
new na_prani vod enabled input film.avi
```

Adresu, na které bude VLC naslouchat požadavkům na videa na přání, je nutno zadat již při startu přehrávače pomocí parametru `rtsp-host`:

```
vlc -I telnet --rtsp-host 192.168.0.100:8554
```

Ve výchozím nastavení se naslouchá na všech rozhraních (lze zadat jako 0.0.0.0) na portu 554.

Volitelně lze také omezit počet uživatelů, kteří mohou využívat služby videa na přání, a to parametrem `rtsp-throttle-users`. Video si lze spustit zadáním adresy, která se skládá z adresy serveru a cesty odpovídající názvu elementu `vod`, v našem případě tedy takto:

```
vlc rtsp://192.168.0.100:8554/na_prani
```

A to je vlastně vše. Seriál o VLC končí, vývoj tohoto přehrávače a jeho možnosti však ne. Pro další inspiraci i oboustrannou podporu zamiřte na webové stránky projektu [VideoLan](#) [11], [uživatelské fórum](#) [12] a [Wiki](#) [13].

■

Odkazy

- [1] <http://live555.com/>
- [2] <http://www.ietf.org/rfc/rfc3550.txt>
- [3] <http://www.ietf.org/rfc/rfc4566.txt>
- [4] <http://www.ietf.org/rfc/rfc2326.txt>
- [5] http://cs.wikipedia.org/wiki/Real-time_Transport_Protocol
- [6] http://en.wikipedia.org/wiki/Session_Description_Protocol
- [7] <http://en.wikipedia.org/wiki/RTSP>
- [8] <http://www.abclinuxu.cz/clanky/multimedia/vlc-4-zaciname-streamovat>
- [9] <http://www.abclinuxu.cz/clanky/multimedia/videolan-client-1-instalace-a-ovladani>
- [10] <http://www.abclinuxu.cz/data/polacek/vlc-6-televize.vlm>
- [11] <http://www.videolan.org/>
- [12] <http://forum.videolan.org/>
- [13] <http://wiki.videolan.org/>

Nastavení vzhledu XDM

XDM je rychlý a nenáročný správce přihlášení dodávaný přímo v rámci systémů XFree86 a X.org. Přečtěte si, jak jej nakonfigurovat, aby se pohodlněji ovládal a lépe vypadal.

Petr Břeň

Původně vyšlo v [blogu](#) [1]. Jako článek vychází se souhlasem autora.

Úvod

Odjakživa, tedy od přechodu na GNU/Linux, jsem se přihlašoval do textové konzole; ovšem poslední dobou jsem zřejmě jaksí zpohodlněl. Uvědomil jsem si, že nemá cenu, abych téměř vždy hned po přihlášení psal do shellu příkaz `startx`, a rozhodl jsem se přejít ke grafickým správcům přihlášení. Po krátké úvaze a shlédnutí diskuzí [výběr window manageru a xdm](#) [2] a [xdm a window manager, nebo aspon shutdown](#) [3] jsem se rozhodl, že pokud používám [Fluxbox](#) [4] a nepotřebuji v přihlašovací okně tlačítka na volbu správce oken ani vypnutí či dokonce restart počítače, je logické, abych měl [XDM](#) [5].

Mnoho uživatelů prý používá [KDM](#) [6] nebo [GDM](#) [7] jen kvůli estetice. Nepotřebují žádná tlačítka, jde jim jen o to, že výchozí přihlašovací okno XDM je hnusné a je obtížné na tom něco vylepšit. Já jsem se rozhodl zůstat u XDM, protože spuštění KDM nebo GDM by trvalo o dost déle, než potom spuštění Fluxboxu. Celkově mi přišlo absurdní mít správce přihlášení rozsáhlejší než správce oken. Na vzhledu XDM jde dost věcí ovlivnit, není to taková hrůza.

XDM (X Window Display Manager) existuje už od roku 1988. Dlouhá léta je standardní součástí okenního systému X11 a určitě jej najdete ve všech hlavních současných distribucích GNU/Linuxu i v dalších svobodných unixových systémech. V dokumentaci vaší distribuce je jistě popsáno, jak XDM nainstalovat a jak docílit jeho spuštění. V Debianu jej nainstalujeme příkazem `apt-get install xdm` a sám se nám přidá do init skriptů.

Pozadí

Nejprve uděláme něco s barvou pozadí, poněvadž ta je ve výchozí konfiguraci XDM nejodpornější – stejná jako u samostatného X serveru, jinde jsem takovou neviděl.

Docílíme toho editací souboru `/etc/X11/xdm/Xsetup`, který obsahuje příkazy spuštěné s právy roota před zobrazením přihlašovacího okna. Pro jednoduchost

si tam vložíme jedinou řádku, jejímž obsahem bude příkaz ke změně pozadí `xsetroot`. Nahlédneme do jeho manuálové stránky a mj. zjistíme, že k nastavení jednobarevného pozadí jej spouštíme s parametrem `-solid jmeno_barvy`. Seznam podporovaných barev bychom měli nalézt v souboru `/etc/X11/rgb.txt`, ovšem za sebe musím poznamenat, že u mě to ty barvy jaksí zaokrouhuje (jako `darkblue` i `steelblue` na `blue` apod.). Stylově a rychle to uděláme třeba takto:

```
# echo "xsetroot -solid blue" >> \\  
/etc/X11/xdm/Xsetup
```



Chceme-li na pozadí něco složitějšího, třeba obrázek, použijeme místo `xsetroot` například `wmsetbg` či `fbsetbg`. Jak, to zjistíme v jejich manuálových stránkách. Příklad:

```
# echo "fbsetbg -C /home/user/pozadi.jpg" >> \\  
/etc/X11/xdm/Xsetup
```

Takto nastavené pozadí vydrží i po přihlášení, tedy dokud/pokud si pozadí plochy (správně `root window`, *kořenové okno*, odtud `xsetroot`) nepřebere pod svoji kontrolu správce oken.

Kromě příkazu na změnu pozadí (kořenového okna) můžeme z tohoto souboru spouštět i prakticky cokoliv dalšího. Napadá mě jedině `xclock` s nějakou pěknou `-geometry`. Někdo si tam může dát třeba [conky](#) [8], ale já osobně to považuji na těch pár vteřin za zcela zbytečné. Tímto způsobem lze na při-

hlašovací obrazovku umístit i tlačítka na vypínání, reboot apod. Použít k tomu můžeme mj. `xdialog` [9]. Je to popsáno v jednom z návodu odkazovaných na konci článku.



Přihlašovací okno

Změny vzhledu samotného přihlašovacího okna pak najdeme v souboru `/etc/X11/xdm/Xresources`. Na začátku toho souboru vidíme nastavení klávesových zkratk, do kterého doporučuji zatím nezasahovat, protože klávesovými zkratkami se zde chápou i tak základní věci, že šipka vlevo znamená pohyb kursoru vlevo; já jsem si na tom nic neměnil. Zajímavé jsou následující řádky:

```
xlogin*greeting: Vítejte na CLIENTHOST
# uvítání, titulek přihlašovacího okna.
# (proměnná CLIENTHOST
# slouží k zobrazení jména počítače.)

xlogin*namePrompt: Jméno:
# titulek políčka pro přihlašovací jméno

xlogin*passwdPrompt: Heslo:
# titulek políčka pro heslo

xlogin*fail: Přihlášení selhalo
# text, který se vypíše při chybném
```

```
# pokusu o přihlášení
```

Dále vřele doporučuji změnit fonty:

```
xlogin*greetFont: \
-**-bold-**-sans-**-100-**-150-**-
xlogin*font: \
-**-bold-**-sans-**-100-**-150-**-
xlogin*promptFont: \
-**-bold-**-sans-**-100-**-150-**-
login*failFont: \
-**-bold-**-sans-**-100-**-150-**-
```

Tohle jsem si tam dal já po pár okamžicích hraní si s `xlsfonts`, vy určitě vymyslíte něco hezčího. Je nutné podotknout, že jsem nepřišel na to, zda a jak XDM podporuje antialiasing, takže pokud na to ani vy nepřijdete, vybírejte si fonty, které vypadají hezky bez antialiasingu. V těchto místech možná někdo má logo své distribuce nebo si tam bude chtít něco svého vložit. Podporovány jsou jen obrázky ve formátu XPM, pokud vím. Já tam mám:

```
xlogin*logoFileName: \
/usr/share/X11/xdm/pixmaps/debian.xpm
```

A na konec něco uděláme s barvami:

```
xlogin*greetColor: darkred
```

Já jsem si změnil jen barvu textu uvítání, ale jak jistě vidíte u sebe, je toho tam víc, včetně věcí s barvou přímo nesouvisejících, jako šířka rámečku přihlašovacího okna, jeho stínu atd. (Tam jsem nevyspekuloval nic estetičtějšího než výchozí hodnoty.)

Závěr

Jde toho změnit ještě víc, viz dokumentace, na kterou odkazuji. Doufám, že jsem tímto příspěvkem aspoň někoho přiměl zvážit, zda pro něj není XDM vyhovující, i když si to třeba předtím nemyslel. Na druhou stranu je fakt, že kdo nutně potřebuje už v přihlašovacím dialogu průhlednost a vyhlazené fonty, tomu asi XDM vyhovovat nebude.

■

Odkazy

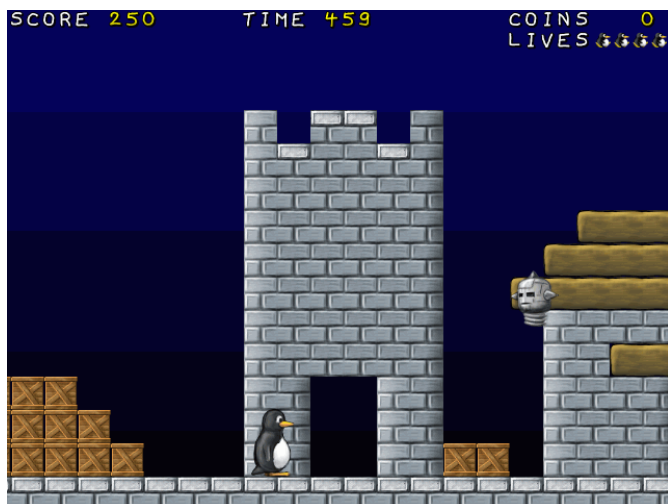
- [1] <http://www.abclinuxu.cz/blog/bhyblog/2006/10/22/154808>
- [2] <http://www.abclinuxu.cz/forum/show/30889>
- [3] <http://www.abclinuxu.cz/forum/show/65727>
- [4] <http://www.abclinuxu.cz/software/pracovni-prostredi/spravci-oken/fluxbox>
- [5] <http://www.abclinuxu.cz/software/system/spravce-prihlaseni/xdm>
- [6] <http://www.abclinuxu.cz/software/system/spravce-prihlaseni/kdm>
- [7] <http://www.abclinuxu.cz/software/system/spravce-prihlaseni/kdm>
- [8] <http://www.abclinuxu.cz/software/system/monitorovani/conky>
- [9] <http://xdialog.dyns.net/>

SuperTux – svobodný Super Mario

Bylo příjemné mrazivé odpoledne. Tux a jeho nádherná Penny si užívali pikniku na ledových pláních Antarktidy. Náhle něco vyskočilo z křoví, zablesklo se a Tux usnul. Když vstal, Penny byla pryč. Na místě, kde ležela, našel Tux pouze dopis od svého úhlavního nepřítele Noloka: „Unesl jsem tvou nádhernou Penny do své pevnosti. Cestu k ní hlídají houfy mých nohsledů. Nemáš šanci získat Penny zpět.“ Tux se z dálky podívá na pevnost a zde začíná jeho dobrodružství.

Luboš Turek

Tak zní příběh jedné z nejhranějších her pro Linux. Málodky se stane, že by open-source předělávka hry překonala originál. Občas se ovšem pošťestí a nejen já jsem se u SuperTuxe bavil daleko více, než při hraní Super Maria od Nintenda. A co víc, hra má český překlad.



Jak jsem již naznačil, [SuperTux \[1\]](#) je silně inspirován sérií Super Mario od Nintenda. Jedná se o dvourozměrnou „jump and run“ [skoč a běž] hru.

Hraní je nenáročné a zvládne ho každý. Úkolem (jak už to u plošinovek bývá) je projít/proskákat/proplézt se každou úrovní od začátku do konce. Pohyb tučňáka se ovládá šipkami plus jednou klávesou pro běh resp. střílení. S těmito informacemi můžete již začít hrát.

Začínáte jako malinký tuxík, pro kterého jedinou možností, jak zneškodnit nestvůru, je skočit jí na hřbet. Raději to ale nezkoušejte u těch špičatých. Za malého tuxíka nebudete hrát věčně. Až najdete bonus v podobě sněhové koule (někdo tomu říká i vejce), stane se tučňák stane větším a odolnějším – může si dovolit jeden střet s nepřítelem, což ho opět zmenší. Druhý bonus je ohnivá květina, po jejímž pozření můžete házet malé ohnivé kuličky,

což je způsob, jak zcela bez problému projít většinu levelů. Další bonus, který jistě potěší, je hvězdička, která vám pomůže stát se na chvíli nezničitelným a naopak zničit každého, o koho zavádíte. Posledním bonusem je malé tuxátko, které vám přičte život. Dále se vyplatí sbírat mince; každých 100 vám také přičte život.

Krom 26 kol na Antarktidě obsahuje hra dva bonusové ostrovy. Kvalita těchto kol ale není zdaleka na takové úrovni. Není nic nepříjemnějšího, než se zaseknout někde, kde nemáte už ani možnost sebevraždy. Jediné, co vám pak zbývá, je počkat si, než vyprší časový limit, nebo zaklínat programátory (nefunguje).

Grafika, hudba

Grafika hry je zdařená, působí jednoduše – nebudete mít dojem, jak tomu u svobodných her často bývá, že je vytvořena taktikou pejska a kočky. Co se týče zvuků a hudby, jsem také spokojen. Hudba je svěží a skvěle přispívá k ledové atmosféře hry.

Budoucnost: Milestone 2

Po dobytí Nolkova hradu vás ale čeká nemilé překvapení. Penny nenajdete. Abyste ji konečně získali, musíte navštívit jiné (prozatím neimplementované) světy. V Milestone 2 to bude les, ve kterém se můžete těšit na větší svobodu při průchodu mapou. Na rozdíl od Antarktidy budete mít většinou více možností, kudy jít. Začínáte v řídkém, listnatém lese a postupně se dostanete do hustého, strašidelného lesa, který připomíná Sherwood nebo park před hlavním nádražím v Praze. Na jeho konci budete muset zdolat „Bosse“, kterým bude Ghost – na rozdíl od Antarktidy, kde jím byl Yeti. V Milestone 2 bude i spousta dalších vylepšení:



Dveře (doors)

Přemístí vás na jiné místo na mapě. Do dveří vejdete stisknutím šipky nahoru.

Přepínače (switches)

Jsou dva druhy: páčky a panely. Přepínač může zpřístupnit zablokovanou cestu, aktivovat platformu, vypnout vítr apod. Většina přepínačů je velice blízko objektu, který ovlivňují. Objekt je navíc často označen značkou stejné barvy, jakou má přepínač.

Platformy (platforms)

Jedná se o létací, plovací, posouvající a jinak různě se pohybující objekty.

Trampolíny (Trampolines)

Trampolína je objekt, který vám pomůže se pohybovat po krajině. S pomocí trampolíny jste schopni vyskočit do neuvěřitelné výšky. Výskok se ještě stupňuje, pokud při dopadu držíte šipku dolu.

Informace (info blocks)

Pokud do nich při výskoku vrazíte hlavou, zobrazí hlášku.

Dále se můžete těšit na spoustu nových monster v podobě stromů, pavouků, raket apod. Schopnosti hlavního hrdiny nebudou zahálet a také budou vylep-

šeny. Bude umět salto, při němž sice vyskočí vysoko, avšak pohyb dopředu bude pomalejší, což oceníte v místech, kde je méně prostoru pro běh.

Také přibude možnost dělat takzvaný „Butt Stomp“ (dalo by se to přeložit jako zadkodup...), kterým zabijete nestvůry, na které normální skok nestačí. Vývojáři jsou velice otevření nápadům od fanoušků. Pokud si myslíte, že to má cenu, pokuste se zapojit!

Hráčské levely a level-sety

Nezoufejte – fakt, že jste *ST* dohráli, ještě neznamená nutnou odinstalaci. Součástí hry je i zabudovaný editor vysoké kvality. Jelikož se SuperTux těší veliké popularitě, na internetu se vyskytuje spousta zajímavých levelů a level-setů. Zkuste například web supertux.info [2].

Instalace

Pro Linux x86 existuje automatický [instalátor](#) [3] vytvořený za pomoci [Autopackage](#) [4]. Touto cestou jsem šel i já; je to rychlé a bezproblémové. Spousta distribucí již SuperTux obsahuje:

Instalace na Debianu

Jako superuživatel stačí napsat:

```
apt-get install supertux
```

Instalace pro SUSE Linux

Pro instalaci SuperTuxe můžete použít Yast:

```
yast2 -i supertux
```

Gentoo

I Gentoo obsahuje SuperTuxe. Instalujte klasicky přes emerge: `emerge supertux`. Pokud přece jen dáte přednost překladu zdrojů [5], nepamenejte, že hra vyžaduje knihovny [SDL](#) [6]. Doporučuji využít OpenGL akceleraci. Na mém artefaktu to zlepšilo poměr FPS z 18 na 70+

Instalace pro Windows

Pro Windows existuje [grafický instalátor](#) [7].

Odkazy

[1] <http://www.abclinuxu.cz/software/hry/arkady/supertux>

[2] <http://www.supertux.info>

[3] http://supertux.berlios.de/wiki/index.php/Download/Installation#Generic_Linux_x86_Installer

[4] <http://www.abclinuxu.cz/software/system/balicky/autopackage>

[5] http://supertux.berlios.de/wiki/index.php/Download/Installation#Any_system.2FSource_Code

[6] <http://www.abclinuxu.cz/software/programovani/knihovny/sdl>

[7] http://supertux.berlios.de/wiki/index.php/Download/Installation#Windows_Installer

Jaderné noviny – 13. 9. 2006

Aktuální verze jádra: 2.6.17.13. Citáty týdne: Andrew Morton a tým -stable. MMIO bariéry. Pokus o vzkříšení kvalifikací v Linuxu.

Robert Krátký

Aktuální verze jádra: 2.6.17.13

Aktuální verze jádra je 2.6.17.13, vydaná 8. září, několik minut po nepovedené 2.6.17.12 [1]. Do těchto vydání bylo zařazeno docela dost důležitých oprav, i když žádná z nich není vedena jako ◊CVE [2].

Adrian Bunk vydal 2.6.16.29-rc1 [3] a 2.6.16.29-rc2 [4] s další sadou oprav.

Aktuální předverze je 2.6.18-rc7, vydaná [5] 13. září. Ok, ok, já vím. Myslel jsem, že -rc6 bude poslední, ale mám prostě lepší pocit, když vydáme ještě -rc7, i když změny jsou jen malé.

Finální verze by už měla vyjít velmi brzy. Aktuální verze -mm stromu je 2.6.18-rc6-mm2 [6]. Mezi změny patří úpravy USB API, velký patch pro x86-64 (včetně podpory ochrany stacku), ACL pro tmpfs a patch, který na některých systémech možná změní pořadí číslování PCI zařízení. V -mm je teď 1915 patchů – nejmíce, kolik kdy bylo.

Citáty týdne: Andrew Morton a tým -stable

Cesta k 2.6.19-rc1 bude drsná – čeká tu nezvykle hodně práce a mezi subsystemovými stromy jsou nezvyklé (i když stále malé) přesahy, které bude nutné vyřešit. Proto předpokládám, že nám -rc1 zabere delší dobu než obvyklé dva týdny.

– Andrew Morton [7]

Velmi se omlouváme za chyby, které se připlety do verze .12 – viníky už jsme vykopli.

– The -stable team [8]

MMIO bariéry

Paul Mackerras nedávno [nahlásil](#) [9] závažnou chybu. Ethernetový ovladač tg3 pracuje, podobně jako mnoho dalších síťových ovladačů, se sadou popisovačů bufferů uložených v paměti hostitelského systému. Tyto popisovače popisují buffery, které jsou k dispozici pro přichozí síťové pakety. Když přijde paket, rozhraní vezme další popisovač, nacpe tam data a řekne procesoru, že je paket k dispozici. Hlášená

chyba se projevuje takto: procesor pozmění datovou strukturu popisovače, pak provede zápis do MMIO (memory-mapped I/O) registru, aby zařízení řekl, že má začít s I/O. Zařízení však tento MMIO zápis dostane dříve, než data zapsaná do hlavní paměti dorazí ke svému cíli, takže pracuje se starými daty. V takovém případě je správný průběh přinejmenším nepravděpodobný.

Chyby vyplývající z přehození pořadí paměťových operací mohou být zrádné a těžko odhalitelné. Vývojář by na kód mohl zírat hodiny, aniž by si uvědomil, že to, co se ve skutečnosti děje hluboko v hardwaru systému, neodpovídá tomu, jak je napsaný kód. Nesprávné chování může být nepravidelné a nemusí být možné ho reprodukovat.

Řešením tohoto druhu problémů bývá obvyčejně umístění nějaké paměťové bariéry v situacích, kde na pořadí operací záleží.

Pro autory ovladačů je možná nejnámějším druhem bariéry následující klasické pravidlo: MMIO zápisy do I/O paměti na PCI sběrnici nelze považovat za dokončené, dokud nebude z daného adresního rozsahu provedeno čtení. Takže ovladače vypadají často tak, že i když je u mnoha registrů nastavena hodnota popisující I/O operaci, před konečným zápisem musí být provedeno čtení, které nastaví "OK" bit. Bez toho čtení, které funguje jako MMIO bariéra, by mohlo zařízení začít pracovat se starými hodnotami a nadělat škody.

Chyba v tg3 však znázorňuje trochu odlišný problém: neexistuje žádné zaručené řazení zápisů do běžné paměti a MMIO rozsahu. A proto se Paul ptal: neměl by být MMIO zápis předdefinován tak, aby byl přísně zařazen s ohledem na předchozí zápisy do běžné paměti? Na mnoha architekturách (včetně i386) se o řazení hezky postará hardware, ale u jiných (Paul pracuje na PowerPC64) žádné záruky nejsou. Přidání bariér do MMIO zápisových operací (`iowrite32()`, `writel()` atd.) na příslušných platformách by mohlo odstranit množství potenciálních chyb.

Linusovi se to nezamlouvalo [10], protože to považuje za příliš náročné. Paměťové bariéry mohou procesor na dlouhou dobu zdržet, takže je lepší se jim vyhnout, nejsou-li opravdu nutné. Proto je podle Linuse vhodnější chtít po programátorech, aby bariérové operace používali v konkrétních případech.

Tento přístup však také není bezproblémový. Například proto, že jádro v současné době neimplementuje bariéru určenou k udržení řazení běžných a MMIO paměťových operací. Máme sice `mmiowb()`, ale ta má za účel zajistit řazení pouze MMIO operací. Proto Linus zmínil možnost vytvoření nových bariér s názvy jako `mem_to_io_barrier()`, které by se postaraly o požadovaný pořádek.

Jinou možností by bylo předefinování MMIO operací tak, aby obsahovaly bariéru předtím, než dojde na MMIO přístup. To by sice opravilo chybu v `tg3` bez zbytečné režie, ale jen za cenu odstranění bariéry, která je teď umístěna až za operací. Takové řešení by preferoval [11] Paul:

Řekl bych, že nejlepší by v tuto chvíli bylo přesunout `sync` ve `writeX()` před ukládání, jak navrhuješ, a přidat „eieio“ před natáhnutí v `readX()`. To by sice znamenalo, že spoléháme na autory ovladačů, aby dali `mmiowb()` mezi `writeX()` a `spin_unlock`, ale aspoň je to zdokumentované.

Takový přístup se však nelíbil [12] Davidu Millerovi (a dalším):

Autoři ovladačů ty paměťové bariéry nebudou používat správně. Možná myslíš, že ano, protože to bude „dokumentované“, ale to nic nemění na skutečnosti, že přestože autoři ovladačů dobře zvládnou jednoduchá rozhraní, tak tyto paměťové bariéry jsou složité koncepty, které bude polovina z nich mít špatně.

David by byl raději, kdyby věci fungovaly správně i při té jednoduché variantě – i za cenu větší režie. A jak zmiňovali ostatní, vždycky se dají implementovat nebariérové verze základních MMIO funkcí pro vývojáře, kterým záleží na výkonu, a kteří (si myslí, že) vědí, co dělají.

Největší starosti by možná byly s návrhem, který představil Paul: zavolání `mmiowb()` mezi poslední zapisovací MMIO operací a voláním `spin_unlock()`. Spinlocky procesorům (nebo procesům v preemptivním prostředí) zabírají v pomíchání operací na jediném zařízení. Ale spinlock žije v paměti, takže je možné, že by odemkací operace uspěla (a umožnila tak dalšímu procesu přistupovat k MMIO oblasti) před dokončením zápisů předchozího procesu.

Proto se volá `mmiowb()` – ale vypadá to jako věc, na kterou by autoři ovladačů často zapomněli.

Alternativu [13] navrhl Alan Cox. Jde o vytvoření nové dvojice spinlock operací:

`spin_lock_io()` a `spin_unlock_io()`. Byly by výslovně definovány k ochraně operací s MMIO oblastmi a obsahovaly by potřebné bariéry. Kdyby se podařilo naučit ovladače zařízení, aby tyto zamykací operace používaly (a autory ovladačů je mnohdy snadné něco naučit – stačí jim dát pivo, když se něco povede), nemusely by pamatovat na vkládání bariér.

Pár problémů je však i s tímto řešením. Už existuje množství variací operace `spin_lock()`; přidání další možnosti by počet zamykacích volání výrazně navýšilo. Kód, který volá funkce, zatímco drží zámky, si musí být už teď vědom zamykacích potřeb volaných funkcí – a to by bylo také daleko komplikovanější. Linus by se proto takovému přístupu raději vyhnul [14] a vyžadoval používání explicitních bariér.

A ještě další přístup – který bude nakonec možná zaveden – je předefinování a rozšíření sady pomocných MMIO funkcí. Podle popisu [15] Benjamin Herrens Schmidta by stávající funkce (`writel()` atd.) byly plně řazené – i kdyby je to mělo zpomalit. Všechny ovladače, které je používají, by i nadále fungovaly – a některé by se díky tomu dočkaly opravení neobvyklých skrytých chyb.

Pro většinu ovladačů by takové funkce byly postačující – paměťové bariéry kolem MMIO operací by zpravidla výkon neovlivnily. Pro situace, kde by bariéry nebyly nutné nebo by byly ke škodě, by byly definovány nové pomůcky s názvy jako `_writel()` nebo `_iowrite32()`. Tyto funkce by zajistily, že budou MMIO operace periferním zařízením vnímány v pořadí určeném procesorem, ale bez dalších záruk. Při použití těchto primitiv by byl programátor zodpovědný za umístění bariér (v případech, ve kterých je řazení MMIO a běžných paměťových operací podstatné).

A nakonec byla navržena sada funkcí s názvy jako `_raw_writel()` – pro programátory, kteří by chtěli balancovat na ostří nože. Tyto funkce by neposkytovaly žádné záruky řazení a nezajímaly by je otázky typu byte swapping. Jsou jen o krůček výše než přímé zadávání I/O operací v assembleru. Benjaminův návrh také znovu oživuje nápad na vytvoření nové sady paměťových bariér pro specifické situace. Takže `io_to_io_barrier()` by zajistila řazení MMIO operací; byla by užitečná ve spojení s „raw“ opera-

cemi popisovanými výše. Další bariéry by se různými způsoby zabývaly řazením MMIO a běžných paměťových operací. Viz Benjaminův email, kde je celý seznam.

Objevilo se dost návrhů na změny, ale žádný vážný odpor. Takže to nakonec možná bude fungovat právě tak – i když v budoucnu lze očekávat další diskuze. Protože se jedná o jednu z nejošemetnějších oblastí programování jádra na současném hardwaru, nebudou snadná a konečná řešení rychle k mání.

Pokus o vzkříšení kvalifikací vLinuxu

V roce 1998, než bylo jádro 2.1 uzavřeno novým funkcím, došlo k začlenění funkce kvalifikací [capabilities]. Kvalifikace rozdělují moc účtu roota na sadu práv, z nichž každé může být uděleno nebo odebráno nezávisle na ostatních. Například proces, který potřebuje mít možnost vázat se na privilegované číslo portu, by mohl tuto schopnost získat, aniž by mu zároveň bylo umožněno přebít práva souborů, zabíjet další procesy nebo překračovat limity zdrojů. Zastánci kvalifikací si již dlouho představují svět, kde není účet roota a všechny úlohy mají právě takovou minimální úroveň oprávnění, kterou potřebují k vykonání své práce. Takový systém by měl být údajně bezpečnější.

Svět je plný linuxových distribucí, z nichž mnohé jsou orientovány na vyšší úroveň bezpečnosti. Ale pokud vím, tak nikdo zatím nesestavil úspěšnou distribuci založenou na kvalifikacích. Pro tento nedostatek implementací je více důvodů, včetně toho, že se nikomu nepodařilo vymyslet, jak spravovat systém s několika desítkami dalších bezpečnostních bitů připojených ke každému spustitelnému souboru. A neměla by se přehlížet ani skutečnost, že od dob jádra 2.1.x až do dneška nebyla jediná verze, kde by kvalifikace opravdu správně fungovaly.

Částečně za to může nekompletní implementace: nebyl začleněn žádný patch, který by k souborům kvalifikační masky připojoval. A jádro také nikdy správně neimplementovalo dědičnost kvalifikací: co se s kvalifikací stane, když proces spustí nový program. Vlastně je to tak, že je dědičnost kvalifikací už nějakou dobu úplně vypnuta, a bez dědičnosti nemůže model kvalifikací fungovat. Takže se využívání kvalifikací omezovalo na velmi malý počet programů, které byly naprogramovány tak, aby se zbavily kvalifikací, které nepotřebují.

David Madore se tento stav rozhodl změnit [sadou patchů \[16\]](#), které mají podporu kvalifikací opravit.

Patch dělá několik věcí, z nichž první je rozšíření sady kvalifikací z 32 na 64 bitů. Stávající jádra mají definovaných 31 kvalifikací, takže není tak těžké si představit, že by v budoucnu bylo potřeba více. Citelně by to bylo znát, kdyby se někdo odhodlal rozdělit na menší práva kvalifikaci `CAP_SYS_ADMIN`, pod kterou teď může spadat vše. Patch některé z nových bitů používá v rámci sady „běžných kvalifikací“, které by normálně měly mít všechny procesy. Patří k nim možnost používat `fork()` nebo `exec()`, schopnost otevírat soubory a zapisovat do nich, používání `ptrace()` a možnost navýšení práv spuštěním `setuid` programu. Je to myšleno tak, že vyžaduje-li to bezpečnost, mohou tyto procesy zmíněné kvalifikace zahodit, a tím ztížit zneužití svých zranitelností.

Jádrem patche je však implementace dědičnosti priorit. Pochopení této části vyžaduje malý úvod. Ačkoliv lze mluvit o kvalifikacích procesu, každý linuxový proces má tři samostatné masky kvalifikací. Sada *povolonych* obsahuje všechny kvalifikace, které proces smí mít. Nelze je však používat, pokud nejsou v *účinné* sadě – podmnožině povolené sady. A konečně má každý proces také sadu *dědičnou*, která drží seznam kvalifikací (opět podmnožinu povolené sady), jež lze předat programům spuštěným pomocí `exec()`. Procesy mohou účinné a dědičné sady kdykoliv měnit (v rámci povolené sady), ale povolenou sadu rozšířit nelze.

V systému založeném na kvalifikacích mají spustitelné soubory také sadu tří masek kvalifikací. Tyto masky mají stejné názvy jako masky procesů a funkce je téměř stejná. Zděděná maska souboru však omezí kvalifikace, které lze zdědit z jakéhokoliv jiného procesu. Davidův patch také obsahuje kód od Sergeje Hallyna, který přidává podporu masek kvalifikací do vrstvy souborových systémů. Když proces spustí nový spustitelný soubor, jsou masky nakombinovány následovně:

- $P'_p \leftarrow (P_i \cap F_i) \cup (F_p \cap bnd)$
- $P'_e \leftarrow (P_i \cap P_e \cap F_i) \cup (F_p \cap F_e \cap bnd)$
- $P'_i \leftarrow P'_p$

Tyto rovnice jsou převzaty přímo z [Davidovy stránky o „nových kvalifikacích“ \[17\]](#), kde je mnohem více podrobností o celé této práci. Převedené do běžného jazyka to vypadá asi takto:

- Povolené [permitted] kvalifikace pro nový spustitelný soubor (P'_p) jsou průnikem dědičné sady procesu před voláním `exec()` (P_i) a dědičné sady souboru (F_i). Pak je přidána povolená sada

ze souboru (F_p), ale ne předtím, než bude omezena celosystémovou sadou vymezující kvalifikace [18].

- Účinné [effective] kvalifikace (P'_e) budou stejné jako dědičné – kromě toho, že kvalifikace, které nejsou u stávajícího procesu aktivní nebo nejsou v účinné sadě souboru, budou odmaskovány.
- Dědičné [inheritable] kvalifikace (P'_i) budou stejné jako povolené kvalifikace.

Tato pravidla se z větší části shodují s tím, jaké chování se očekává od systému založeného na kvalifikacích. Kvalifikace jsou na takových systémech přiznávány programům, nikoliv uživatelům. Běžné bity práv pak mohou hrát roli v tom, které programy smí určití uživatelé spouštět.

Davidův patch se však od představ o systému založeném na kvalifikacích liší v jednom důležitém ohledu: jak zachází s programy, které nemají žádné sady kvalifikací definovány. Na většině systémů se to bude týkat téměř všech spustitelných souborů. Podle pravidel by s takovými programy mělo být nakládáno jako by měly prázdnou dědičnou sadu, což by – podle výše uvedených pravidel – způsobilo, že by byly spouštěny bez jakýchkoliv kvalifikací. Davidův patch však nechává tyto programy spouštět se stejnými kvalifikacemi, které měl proces předtím – i když

v případě setuid bitů by samozřejmě mohlo dojít ke změně. Tato interpretace porušuje klasický kvalifikační model. Výhodou je, že díky tomu na současných systémech funguje.

Ted T'so **namítal** [19], že takový kompromis zásadně oslabuje bezpečnost kvalifikačního modelu. Navrhl, aby šlo o konfigurovatelné chování; každý souborový systém by měl parametr popisující, jak by mělo být s kvalifikacemi nakládáno v případě, že by soubory neměly nastavené masky. Součástí této změny by mohla být i sada výchozích kvalifikací pro nové soubory.

Další připomínky byly poměrně předvídatelné: proč se trápit s kvalifikacemi, když máme SELinux, který umí to samé a ještě více. Ve skutečnosti dělá SELinux něco vzdáleně podobného, ale dost nepřímo; připojuje k souborům nálepky, pak k nálepkám přiřazuje kvalifikace pomocí mechanismu pravidel. Každý, kdo měl to štěstí spatřit tu veselou hlášku při bootu Fedory („váš souborový systém musí být přenalepkován, čekejte prosím, bude to trvat velmi dlouho“), ví, že udržování správně synchronizovaných souborů a nálepek je složitá záležitost. Není žádný důvod se domnívat, že by udržování masek kvalifikací ve správném stavu mělo být snazší. To samo o sobě by mohlo do budoucna omezovat možnosti využití kvalifikací. ■

Odkazy

- [1] <http://lwn.net/Articles/198925/>
- [2] <http://www.abclinuxu.cz/slovník/cve>
- [3] <http://lwn.net/Articles/198688/>
- [4] <http://lwn.net/Articles/199041/>
- [5] <http://lwn.net/Articles/199326/>
- [6] <http://lwn.net/Articles/199187/>
- [7] <http://lwn.net/Articles/198975/>
- [8] <http://lwn.net/Articles/198927/>
- [9] <http://lwn.net/Articles/198989/>
- [10] <http://lwn.net/Articles/198991/>
- [11] <http://lwn.net/Articles/198992/>
- [12] <http://lwn.net/Articles/198995/>
- [13] <http://lwn.net/Articles/198996/>
- [14] <http://lwn.net/Articles/198998/>
- [15] <http://lwn.net/Articles/199106/>
- [16] <http://lwn.net/Articles/198977/>
- [17] <http://www.madore.org/~david/linux/newcaps/>
- [18] <http://lwn.net/1999/1202/kernel.php3>
- [19] <http://lwn.net/Articles/199007/>

Jaderné noviny – 20. 9. 2006

Aktuální verze jádra: 2.6.18. Sledovací infrastruktury. Další implementace kontejnerů. `nopage()` a `nopfn()`.

Robert Krátký

Aktuální verze jádra: 2.6.18

Aktuální verze jádra je 2.6.18, vydaná [1] 19. září. Určitě si přečtete oznámení; vypadá to, že v něm jsou i věci, které nepocházejí přímo z gitu. Nová verze přináší spoustu novinek, včetně odvozování priorit [2], nové obecné IRQ vrstvy [3], nového časového subsystému [4], validátoru zámků [5], SMPnice [6], úprav virtuální paměti, velké aktualizace SATA [7], odstranění DevFS [8] a mnoha dalších. Vizte stránku LinuxChanges na KernelNewbies [9], kde je mnohem podrobnější seznam, stránku o změnách API v 2.6 na LWN [10] s informacemi o změnách interních programovacích rozhraní nebo dlouhý changelog [11]. Aktuální verze -mm stromu je 2.6.18-rc7-mm1 [12]. Andrew k vydání poznamenal:

Trvalo snad deset hodin, než bylo možné tuhle hromadu kódu jakž takž zkompilovat, aby se na x86, x86_64 a PPC dobelhala k loginu. Nadšenci mohou testovat.

Také připomněl, že toto jádro nebude fungovat na distribucích se starší verzí `udev` kvůli změnám v ovladačové jádře. Mluvilo se o tom v srpnu (nová jádra a staré distribuce [13]).

Další změny: „pravděpodobně chybná“ změna na API `kmap()`, které se má starat o otázky koherence, nový parametr `GFP_THISNODE` pro alokaci paměti, odstranění pochybného ovladače HDAPS [14] bez uvedení důvodu (i když stojí za zmínku, že jeden z posledních patchů [15] pro 2.6.18 dal jasně najevo, že anonymní příspěvky nemohou být přijaty), SLIM [16] a množství oprav.

Pro uživatele 2.6.16: Adrian Bunk 13. září vydal 2.6.16.29 [17] s několika opravami.

Aktuální 2.4 předverze je 2.4.34-pre3, vydaná [18] 19. září. Hlavní změnou je začlenění gcc 4.0 patchů [19].

Sledovací infrastruktury

Někdy věci prostě nejdou podle plánu. Mathieu Desnoyers je aktuálním správcem Linux Trace Toolkitu [20], balíku pro sledování jaderných událostí, který – i přes početnou skupinu uživatelů – zůstává už

mnoho let mimo jádro. Nedávno vydal novou verzi LTT [21] s následujícím úvodem:

Podle Christophovy rady z letošního léta předkládám menší patch, který by mělo být snazší zkontrolovat, takže snad budou všichni spokojeni.

Výsledkem byla diskuze se stovkami komentářů [22], z nichž mnohé lze považovat za neslušné i podle měřítek `LKML` [23]. LTT zjevně odhalil citlivé místo – zvláště překvapivé to je s ohledem na skutečnost, že oblasti sporu jsou minimální.

Čas od času se lidi ptali, jestli jádro vůbec nějaké sledování potřebuje. Tato otázka se zdá být vyřešena (s kladným výsledkem); neshody nyní panují o tom, jestli má být sledování statické nebo dynamické. Statické sledování vkládá do kódu explicitní sledovací body (vypadají jako volání funkcí); sledovací systém pak tyto sledovací body může dle libosti za běhu zapínat nebo vypínat. V dynamickém systému jsou sledovací body do běžícího systému vkládány většinou ve formě přerušovacích instrukcí.

Jádro už má dynamické sledování: KProbes [24]; LTT však používá (především) statický model. Největší otázkou (alespoň na povrchu) tedy je, jestli jádro kromě stávajícího dynamického mechanismu potřebuje ještě balík pro statické sledování. Diskuze se točí kolem několika bodů:

- Režie, část 1: pokud zrovna není sledování využito (normální stav na většině systémů), mají dynamické sledovací body zjevně menší režii: vůbec neexistují. Přes všechno úsilí věnované tomu, aby byly statické sledovací body rychlé, když nejsou využívány, tak tam pořád jsou a vždy budou mít (malou) režii.
- Režie, část 2: pokud sledování využíváno je, bývají statické sledovací body rychlejší. Přerušovací mechanismus, který (v současné implementaci) využívají KProbes, si vezme přibližně desetkrát více cyklů procesoru než statický sledovací bod. Existují projekty (především `djprobes`), které mohou tuto režii výrazně snížit. V rámci diskuze poslal Ingo Molnar sérii patchů, které snižují režii KProbes zhruba na polovinu.

Proč se v takovém případě starat o režii? Sledování se mnohdy používá ke zkoumání častých událostí, takže větší počet sledovacích bodů může věci znatelně zpomalit. Jde však hlavně o to, že těžkopádné sledovací body mohou změnit načasování událostí, což vede k obávaným "heisenbugs", které zmizí v okamžiku, kdy je vývojář začne hledat.

- Režie správy: někteří vývojáři mají obavy, že by přidání statických sledovacích bodů zkomplikovalo správu kódu. Sledovací kódy samy o sobě kód zahušťují a navíc musí fungovat provždy. Každý z nich by si šlo představit jako malé systémové volání, které se po umístění nesmí měnit. Vývojáři se také obávají, že by v budoucnu mohly přicházet požadavky na přidávání dalších bodů.

Na druhou stranu, dynamické sledovací body znamenají jiný druh režie: každý, kdo potřebuje určitou skupinu bodů, se musí starat o jejich správu. Při změnách jádra se budou muset měnit i body, aby tyto změny odrážely. Udržovat dynamické sledovací body aktuální může být velmi složité a namáhavé. Nástroje jako [System-Tap](#) [25] v tomto ohledu pomáhají, ale zdaleka se v současnosti nejedná o kompletní řešení. Naproti tomu statické sledovací body umístěné v jádře budou fungovat i při změnách kódu.

- Pružnost: dynamické sledovací body lze umístit kdykoliv a kamkoliv, kdežto statické vyžadují přinejmenším editaci kódu, rekompilaci a restart. Dynamické sledovací body mohou také snáze podporovat filtrování událostí za běhu. Statické jsou zase lepší při přístupu k lokálním proměnným.
- Podpora architektury: KProbes v současné době nejsou na všech architekturách implementovány, takže nejsou dostupné všem uživatelům Linuxu. Statické sledovací body obvykle nevyžadují tolik figlů specifických pro jednotlivé architektury, takže je snazší je univerzálně podporovat. Mluvílo se však také o tom, že začleněním statických sledovacích bodů by správci architektury ztratili motivaci ke zprovozňování KProbes.

Dá se pochopit, že si člověk při pročítání diskuze začne zoufat. Kupodivu je však množství neshod nižší, než by se zdálo. Účastníci se skoro shodují v tom, že existuje uplatnění pro *obojí* – statické i dynamické sledovací body. Statické sledování událostí pomůže

mnoha lidem porozumět tomu, co se v systému děje – nejen vývojářům jádra, ale i vývojářům uživatelských aplikací a systémovým administrátorům. Chtít po těchto všech, aby zjišťovali, kam vložit například sledovací bod pro hlášení změn scheduleru v konkrétním jádře by bylo mnohem těžší.

Klíčové je však to, že hodnota statického bodu není ani tak v jeho statickém umístění, ale v tom, že ukazuje, kde by měl sledovací bod být. Proto bylo navrženo řešení, které by mělo být přijatelné pro všechny – vložit místo sledovacích bodů „značky“. Tyto značky, které by byly v jiné části obrazu jádra, jsou pouhými záložkami ukazujícími, kam by měly být v případě potřeby vloženy dynamické sledovací body. Mathieu k tomu účelu poslal [jednoduchý značkovací patch](#) [26]; rychle se dostal pod palbu kvůli implementačním otázkám, ale jen málo lidí bylo vyloženo proti takovému nápadu.

Takže značky možná budou tím krokem vpřed. Pokud by mohl být kód LTT přepracován podle konceptu značek, mohlo by to otevřít dveře diskuzi o tom, co ještě je potřeba udělat, než bude kód začleněn (je tu několik problémů, které byly až doposud zastíněny stávající debatou). Po odpovídajícím zvažování by do jádra mohla být přidána pečlivě vybraná sada značek/sledovacích bodů, což by všem umožnilo se napojit a monitorovat události. Až opadnou vášně, mohlo by z toho být řešení, se kterým budou souhlasit skoro všichni.

Další implementace kontejnerů

Kontejnery se už přibližně rok těší zvýšené pozornosti vývojářů. [Koncept kontejnerů](#) [27] nabízí mnohé z předností plné paravirtualizace, ale za mnohem nižší cenu, což umožňuje spuštění více virtuálních strojů na jediném hostiteli. Jediným problémem je všechny přesvědčit o tom, co to vlastně ten kontejner je. Nedávná [kontejnerová sada patchů](#) [28] od Rohita Setha je dalším pokusem, jak tomuto konceptu dát tvar.

Mnohé přístupy ke kontejnerům se točí kolem stromu procesů – jeden proces se uzavře do kontejneru a stane se v něm „init“ procesem; kontejner je pak zaplněn potomky původního procesu. Rohitův patch část této funkčnosti zachovává – když proces zavolá `fork()`, bude potomek náležet stejnému kontejneru jako předek (je-li nějaký). Mechanismus je však ještě trochu pružnější. Do kontejneru mohou být kdykoliv přidány – nebo z něj odebrány – libovolné procesy.

Takové změny se provádí prostřednictvím rozhraní `configs` [29]. Je-li `configs` připojen do `/config`, může administrátor systému s kontejnery pracovat v `/config/containers`. Nový kontejner se založí vytvořením nového adresáře; kontejnery jsou tedy identifikovány pomocí jednoduché jmenné struktury. Adresář kontejneru obsahuje několik souborů:

- `addtask`: zapsání ID procesu do tohoto souboru způsobí přidání příslušného procesu do kontejneru. Procesy, které už patří nějakému kontejneru, nemohou být do nového kontejneru přidány jen tak; nejprve musí být odstraněny ze starého kontejneru.
- `rmtask`: odstranit proces lze zapsáním ID do tohoto souboru.
- `page_limit`: maximální počet aktivních stránek paměti, které může kontejner použít.

Pak ještě několik informačních souborů pro získávání statistik o provozu kontejneru. Paměťový limit funguje pomocí přidání kontejnerového ukazatele do každé struktury `mm_struct` a `address_space` v systému. Jak dochází k uvolňování nebo zabírání stránek, je aktualizováno i počítadlo kontejneru. Pokud kontejner svůj limit přesáhne, pustí se do práce samostatný proces (pracovní fronta), který uvolňuje stránky patřící kontejneru.

Je-li limit překročen o hodně, budou procesy v rámci kontejneru na okamžik pozdrženy (když se pokusí přidat stránky), aby likvidátor nezůstal pozadu.

Rohitovy kontejnery se tedy starají o ovládání souhrnného využití zdrojů. V tomto smyslu připomínají *cifřšpióny zdrojů* [30] – i když žádný kód z toho patche nevyužívají. Tyto kontejnery také postrádají jednu funkci, kterou lze nalézt ve většině jiných implementací: nějaký způsob kontroly jmenných prostorů. Procesy umístěné do jednoho z těchto kontejnerů i nadále vidí celý systém – a budou k němu mít přístup.

Takže jde jen o částečné řešení problému – alespoň z tohoto pohledu. Funkce pro kontrolu jmenného prostoru by patrně mohly být přidány později, i když by bylo zajímavé vidět, jak by taková kontrola spolupracovala s možností kdykoliv přidávat nebo odebírat procesy.

`nopage()` a `nopfn()`

`nopage()` je operace s adresními prostory, která se má starat o velké chyby stránek v daném adresním rozsahu. Pro adresní prostory se zálohou v souboru

rech existuje obecná `nopage()` metoda, která zajistí načtení potřebných stránek do paměti. Ovladače zařízení také občas poskytují `nopage()` coby součást své implementace `mmap()`. V případě ovladačů je chyba stránky většinou řešena nalezením `structpage` odpovídající bufferu mapovanému do paměti a předáním výsledku zpátky jádru.

`nopage()` může signalizovat dvě chyby:

`NOPAGE_SIGBUS` u skutečně špatných adres a

`NOPAGE_OOM` u situací, při kterých OOM (nedostatek paměti) způsobil selhání pokusu o řešení chyby. Chybí však možnost dát najevo, že byla `nopage()` přerušena signálem a operace by se měla zkusit znovu.

Taková situace většinou nenastává, protože pokud musí `nopage()` čekat, většinou je to provedeno tak, aby operaci nešlo přerušit. Benjamin Herrenschmidt však na tento problém narazil a navrhl [31] malou změnu, která by přidala novou hodnotu

`NOPAGE_RETRY`. Reakce by byla předvídatelná – opětovné spuštění operace po zpracování signálu.

Ukázalo se, že Google *podobný patch* [32] má pro svou interní potřebu, i když důvody použití jsou jiné. V případě Google slouží patch k odstranění problému s výkonem. Patch nebyl navržen k začlenění kvůli možným potížím s DoS [33] a také proto, že jej jeho autor považuje spíše za hack.

Je možné, že nakonec bude patch v nějaké podobě začleněn, ale nejprve je nutné na něm zapracovat. Oba patche ukazují, že k vracení `NOPAGE_RETRY` je více důvodů, takže by dávalo smysl je zpřístupnit vyšší úrovni zpracovávající chyb stránek.

To by umožnilo odstranění některých potenciálních problémů s efektivitou, ale i nadále je tu hrozba možného DoS útoku.

Při té příležitosti se mluvilo o dlouho známém omezení `nopage()`, které spočívá v tom, že může řešit jen situace, kdy má příslušná fyzická paměť odpovídající `struct page`. Tyto struktury existují u hlavní paměti, ale ne pokud je paměť například na periferním zařízení a mapovaná do PCI I/O paměťové oblasti. Některé architektury také provádějí velmi podivné věci se speciální pamětí a vícenásobnými pohledy na tu samou paměť. V takových případech musí ovladače paměť explicitně namapovat do uživatelského prostoru pomocí `remap_pfn_range()` místo použití `nopage()`.

Jes Sorensen má už nějakou dobu připravený [patch \[34\]](#), který přidává další operaci s adresním rozsahem: `nopfn()`. Je volána v reakci na chyby stránek, není-li k dispozici žádná operace `nopage()`; má za úkol vrátit fyzickou adresu (jako číslo rámce stránky) stránky, což chybu napraví. Adresa bude uložena přímo do tabulky stránek procesu, přičemž není potřeba žádná `struct page` a neprovádí se počítání referencí. Jes napsal [ovladač speciální paměti pro IA-64 \[35\]](#), který demonstruje využití této operace.

V minulosti nebyla tato věc příliš populární – [Linus je proti \[36\]](#), stejně jako další. Někteří to považují za komplikování subsystému virtuální paměti; ti by byli raději, kdyby se v kódu používala `remap_pfn_range()` nebo se podle potřeby vytvořily speciální `page` struktury. Je však několik situací, ve kterých prý `nopfn()` funguje lépe, a tlak na začlenění neustupuje. Bude tedy zajímavé sledovat, jestli se dostane do 2.6.19 nebo ne.

Odkazy

- [1] <http://lwn.net/Articles/200149/>
- [2] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-5.-4.-2006#odvozovani-priorit-v-jadre>
- [3] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-24.-5.-2006#jn-nova-obecna-irq-vrstva>
- [4] <http://lwn.net/Articles/120850/>
- [5] <http://lwn.net/Articles/185666/>
- [6] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-7.-6.-2006#smpnice>
- [7] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-17.-5.-2006#jn-velke-zmeny-v-sata>
- [8] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-21.-6.-2006#proc-odstranit-devfs>
- [9] <http://kernelnewbies.org/LinuxChanges>
- [10] <http://lwn.net/Articles/2.6-kernel-api/>
- [11] <http://kernel.org/pub/linux/kernel/v2.6/ChangeLog-2.6.18>
- [12] <http://lwn.net/Articles/200017/>
- [13] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-2.-8.-2006#nova-jadra-a-stare-distribuce>
- [14] <http://lwn.net/Articles/194729/>
- [15] <http://lwn.net/Articles/200222/>
- [16] <http://lwn.net/Articles/160157/>
- [17] <http://lwn.net/Articles/199546/>
- [18] <http://lwn.net/Articles/200057/>
- [19] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-23.-8.-2006#stara-jadra-nove-kompilatory>
- [20] <http://ltt.polymtl.ca>
- [21] <http://lwn.net/Articles/200060/>
- [22] <http://thread.gmane.org/gmane.linux.kernel.tracing/1651>
- [23] <http://www.abclinuxu.cz/slovník/lkml>
- [24] <http://lwn.net/Articles/132196/>
- [25] <http://sourceware.org/systemtap/>
- [26] <http://lwn.net/Articles/200016/>
- [27] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-12.-4.-2006#kontejnery-a-odlehlena-virtualizace>
- [28] <http://lwn.net/Articles/199643/>
- [29] <http://lwn.net/Articles/148180/>
- [30] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-30.-8.-2006#zdroje-ma-dati-dal>
- [31] <http://lwn.net/Articles/199642/>
- [32] <http://lwn.net/Articles/200215/>
- [33] <http://lwn.net/Articles/200216/>
- [34] <http://lwn.net/Articles/200156/>
- [35] <http://lwn.net/Articles/200199/>
- [36] <http://www.gelato.unsw.edu.au/archives/linux-ia64/0603/17557.html>

Jaderné noviny – 27. 9. 2006

Aktuální verze jádra: 2.6.18. Citáty týdne: Linus Torvalds, Bruce Perens. Začíná vývojový cyklus 2.6.19. Změny ovladačového API v 2.6.19. Postoj vývojářů jádra k GPLv3.

Robert Krátký

Aktuální verze jádra: 2.6.18

Aktuální verze jádra je i nadále 2.6.18. Období pro začleňování novinek do 2.6.19 už začalo a v okamžiku, kdy vzniká tento text, je už v hlavním git repozitáři přes 2000 patchů – shrnutí najdete v přehledu níže. Aktuální verze -mm stromu je [2.6.18-mm1](#) [1]. Mezi nedávné změny patří ovladač xpad dance pad, nová verze bezpečnostního modulu SLIM a množství oprav. -mm se rychle zmenšuje, protože patche putují do hlavního stromu.

Adrian Bunk vydal [první předverzi 2.6.16.30](#) [2]. Kromě běžných oprav přidává tento patch také několik nových ovladačů, což vyprovokovalo dotazy ohledně kritérií pro přijímání patchů pro dlouhodobě spravovaný strom 2.6.16. Vypadá to, že do budoucna bude 2.6.16 trochu otevřenější k novému kódu než strom -stable.

Citáty týdne: Linus Torvalds, Bruce Perens

Nezaručuji, že změním názor pokaždé. Ale mohu slíbit, že pokud mi většina lidí, kterým důvěřuji, řekne, že jsem pitomec, alespoň zběžně se nad tím zamyslím.

[Sbor: „Linusi, jsi pitomec.“]

– [Linus Torvalds](#) [3]

Nakonec si budeme muset přiznat, že Linux je 15 let staré jádro, a že jak půjde vývoj dál, něco jej nahradí. Nedokážu říci, co to bude, ale myslím, že nejlepším způsobem, jak k tomu přivést lidi, je použít GPL 3.

– [Bruce Perens](#) [4]

Začíná vývojový cyklus 2.6.19

Záplava čekajících patchů byla vpuštěna do hlavního stromu. Těch 2000 patchů, které byly dosud začleňovány, bude pravděpodobně na počet ještě přebito těmi zbývajícími. Následuje seznam věcí, které si už do jádra cestu našly. Začínáme změnami viditelnými pro uživatele:

- Souborový systém OCFS2 už není označen jako „experimentální“.
- Aktualizace InfiniBand, včetně lepší podpory RDMA a ovladačů pro několik nových adaptérů.
- Podpora směrování podle pravidel v IPv6 [policy routing] – a mechanismus pro vytváření více IPv6 směrovacích tabulek pro tato pravidla.
- Sada patchů s ovladači [paralelního ATA](#) [5].
- Sada patchů s [labeled networking](#) [6] [značené síťování], která implementuje [Commercial IP Security Option](#) [7].
- Podpora architektury Atmel AVR32.
- Super-H podpora pro Titan, SH7710VoIPGW a I-O DATA Landisk desky.
- Velké aktualizace architektury PowerPC a S/390. S/390 má teď – kromě jiného – podporu [KProbes](#) [8].
- Nové ovladače pro externí flash na deskách AT-[STK1000](#), sběrnice TI [OMAP1/2 i2c](#), zařízení ESI Miditerminal 4140, RAID řadiče [Areca](#), SCSI řadiče [SuperTrak EX8350/8300/16350/16300](#), síťová rozhraní [QLogic QLA3xxx](#), ethernetové adaptéry [IBM eHEA](#) a ethernetový řadič na deskách [Cirrus Logic ep93xx](#). Začleněn byl také kontroverzní ovladač [aic94xx](#), který napsal původně [Luben Tuikov](#) a od té doby byl upraven mnoha dalšími.

Změny viditelné vývojáři jádra:

- Hodnota `CHECKSUM_HW` byla v síťovacím subsystému dlouho využívána pro podporu hardwarové kontroly součty. Nyní je nahrazena hodnotou `CHECKSUM_PARTIAL` (určena pro odchozí pakety, u nichž musí práci dokončit hardware) a `CHECKSUM_COMPLETE` (pro příchozí pakety, které už byly hardwarem kompletně zkontrolovány).
- Množství změn v oblasti správy paměti, včetně sledování nečistých stránek v mapování sdílené paměti, volitelnosti zón DMA32 a HIGHMEM a mechanismus pro sledování rozsahů pa-

měti (a mezer mezi nimi) nezávislý na architektuře.

- Makra `pud_page()` a `pgd_page()` teď vrací ukazatel `struct page` místo virtuální jaderné adresy. Kód, který potřebuje původní funkci, by měl používat `pud_page_vaddr()` nebo `pgd_page_vaddr()`.
- Množství změn v ovladačovém API, včetně paralelního hledání zařízení a vylepšení procesu uspání/probuzení.
- Nyní je k dispozici oznamovací řetěz pro situace, kdy je nedostatek paměti; smyslem je připravit funkce, které budou v případě potřeby schopné uvolnit nějakou paměť.
- Trochu se změnila sémantika API `kmap()`: na architekturách s komplikovanými problémy spojitosti paměti by `kmap()` a `kunmap()` měly spravovat spojitost mapovaných stránek, což by odstranilo nutnost explicitního vyprázdňování stránek z keše.
- PCI vrstva teď podporuje PCI Express Advanced Error Reporting [pokročilé hlášení chyb].
- Struktura `inode` doznala několika změn, které měly za cíl ji zmenšit.
- Byla přidána operace s adresním prostorem `no_pfn()` [9].

Pokud vás zajímá, co ještě bude pravděpodobně začleněno, podívejte se na [dokument](#) [10] od Andrew Mortona. Zajímavá je například další sada paměťových patchů (přičemž pokračuje debata o tom, jestli dává smysl mít `ZONE_DMA` volitelné), přepracování kódu protokolu síťového času, sada patchů s [vektorovaným AIO](#) [11] (možná), dlouhá řada vylepšení NFS, [eCryptfs](#) [12] (i když někteří jsou proti), různá vylepšení mapovače zařízení a RAID a množství změn obecné IRQ vrstvy. Kromě toho Andrew plánuje začlenit pár patchů týkajících se kontejnerů: virtualizace jmenných prostorů `utsname` a `IPC`. Řekl k tomu:

Samo o sobě to nedává smysl, takže hraji na důvěru – zakládá se to na předpokladu, že Linux bude jednou mít kompletní virtualizaci jmenných prostorů s dostatečným pokrytím na to, aby to bylo uživatelskému prostoru k něčemu platné.

Normálně bych všechny takové funkce prostě podržel v -mm, dokud by nebyly něčím užitečné. Ale v tomto případě by šlo o spoustu příliš dlouho dr-

žených patchů. Takže začnu podobné menší kousky přesouvat do hlavního stromu.

Jednou z věcí, které *nebudou* začleněny, je Reiser4, který stále vyžaduje různé opravy. Vypadá to, že bude muset počkat na další vývojový cyklus.

Změny ovladačového API v 2.6.19

Linuxové ovladačové jádro se i nadále rychle vyvíjí. [Sada patchů pro 2.6.19](#) [13] v tomto procesu pokračuje několika vylepšeními – a také změnami API. Tentokrát to však vypadá, že změny spíše přidávají, takže by nemělo docházet k problémům se stávajícími ovladači.

Citlivou otázkou je doba trvání bootu Linuxu – není moc uživatelů, kteří by si přáli, aby jejich systémy nabíhaly pomaleji. Během bootu se děje mnoho věcí a existuje také mnoho způsobů, jak proces urychlit. Většina příležitostí k urychlení je na straně uživatelského prostoru, ale jádru může dlouho trvat hledání zařízení. Každé musí být nalezeno, inicializováno a připojeno k systému; součástí takové procedury může být čekání na boot periferních zařízení, natažení firmwarů nebo dokonce fyzické akce jako roztočení disků. V důsledku toho stráví jádro spoustu času nečinností – čekáním, až si své odbude samo zařízení.

Jako jeden ze zjevných způsobů urychlení tohoto procesu se jeví paralelní hledání zařízení. Jádro by tak mohlo v okamžiku čekání na odpověď jednoho zařízení nastavovat jiné. Zároveň by se naplno využily víceprocesorové systémy.

Ovladačové jádro v 2.6.19 bude mít konečně možnost provozu v takovém režimu. První změnou je přidání parametru (`multithread_probe`) do struktury `device_driver`.

Bude-li mít ovladač tento parametr nastaven, přesune se v okamžiku hledání vlastní nastavování daného zařízení do samostatného vlákna, které bude moci běžet souběžně s ostatními. Na konci inicializačního procesu jádro počká na ukončení všech nedokončených vláken a pak teprve připojí kořenový oddíl a spustí uživatelský prostor.

Na jednoprocessorových systémech vede tato změna k relativně malému snížení doby bootu. Ovladače obvykle procesor během procesu hledání nepustí, takže k paralelnímu běhu není moc příležitostí – ani ve chvílích, kdy musí jádro chvíli počkat. Na víceprocesorových systémech však může být účinek o dost zřetelnější – každý procesor může zařízení hledat pa-

ralelně s ostatními. Tato změna bude tedy nejvíce užitečná na velkých systémech s mnoha připojenými zařízeními.

Přesněji: bude užitečná, až bude zapnuta; tato funkce je totiž v současné době označena jako „experimentální“ a je s ní spojena řada varování. I když je povolena, týká se pouze PCI zařízení. Ne všechny ovladače byly napsány s ohledem na paralelní hledání, takže například nemají správné zamykání. Problémy mohou nastat i s odběrem elektřiny – současné zapnutí příliš mnoha zařízení může způsobit přetížení – pokud odběr přesáhne možnosti zdroje; výsledný požár by mohl boot dost výrazně zpomalit. Pořadí číslování zařízení bude pravděpodobně méně deterministické. A tak dále. Každopádně by však tato funkce měla časem vést ke zrychlení bootu – především u systémů, kde je dobře známý a stálý hardware (například u embedded aplikací).

Z jiného soudku: API, které se stará o uspávání a probouzení, bylo trošku zaplněno. Mechanismus tříd má teď své vlastní háčky v `struct class`:

```
int (*suspend)(struct device *dev, \
    pm_message_t state);
int (*resume)(struct device *dev);
```

Nová metoda `suspend()` je volána poměrně brzy v rámci uspávacího procesu a měla by se postarat o úkoly týkající se tříd. Mezi ně může patřit utišení zařízení a zastavení zpracovávání na vyšší úrovni. Metoda `resume()` je volána ke konci probouzení a měla by dokončit přípravu zařízení v třídě k provozu.

Většinu `suspend/resume` [uspávání/probouzení] však stále zajišťuje sběrniceový subsystém. Ta část API dostala tři nové metody `struct bus_type`:

```
int (*suspend_prepare)(struct device *dev, \
    pm_message_t state);
int (*suspend_late)(struct device *dev, \
    pm_message_t state);
int (*resume_early)(struct device *dev);
```

Všechny tyto metody jen přidávají další místa, na která se kód sběrnice může zaháknout a provést, co potřebuje provést. Takže `suspend_prepare()` je volána brzy, dokud je systém pořád ještě v provozním stavu. Metoda `suspend()` se oproti předchozím jádrům neliší: je volána po zmrazení úloh a může spát, je-li to nutné. Nová metoda `suspend_late()` je naopak volána velmi pozdě, když už jsou vypnuta přerušení a běží jediný procesor. Při probouzení je

`resume_early()` také volána při ještě vypnutých přerušeních a SMP a stará metoda `resume()` je volána později. PCI subsystém tyto nové funkce zpřístupňuje prostřednictvím tří nových metod ve struktuře `pci_driver`:

```
int (*suspend_prepare)(struct pci_dev *dev, \
    pm_message_t state);
int (*suspend_late)(struct pci_dev *dev, \
    pm_message_t state);
int (*resume_early)(struct pci_dev *dev);
```

V hlavním jádře zatím nejsou žádné ovladače, které by tyto nové metody používaly. A nakonec: subsystém tříd pokračuje v migraci k definitivnímu odstranění struktury `class_device`.

Kvůli tomu se v `struct class` objevily další dvě metody:

```
int (*dev_uevent)(struct device *dev, \
    char **envp, int num_envp,
    char *buffer, int buffer_size);
void (*dev_release)(struct device *dev);
```

Poskytují podobné funkce jako metody `uevent()` a `release` v `struct class_device`. V rámci této migrace byly také přidány dvě nové pomocné funkce:

```
int device_create_bin_file(struct device *dev,
    struct bin_attribute *attr);
void device_remove_bin_file(struct device *dev,
    struct bin_attribute *attr);
```

Tyto metody ovladačům umožní v `sysfs` vytvářet binární atributy aniž by musely pracovat přímo s kódem `sysfs`.

Následující obsah je ©KernelTrap

Postoj vývojářů jádra k GPLv3

25. zář, James Bottomley [14] poslal do LKML text pojmenovaný „Nebezpečí a problémy GPLv3“, který sestavilo deset neaktivnějších vývojářů jádra. Text začíná zkoumáním podílu GPLv2 na úspěchu linuxového jádra a pak pokračuje poukázáním na některé potenciální nedostatky chystané GPLv3. Konkrétně se mluví o DRM ustanoveních: **Přestože považujeme používání DRM mediálními společnostmi při pokusech o ovládání obsahu v zařízeních uživatelů za velmi znepokojující, brání nám víra v základní svobody definované v části 3 v přijetí licence, která by obsahovala omezení koncových uživatelů.**

Ustanovení o dodatečných omezeních: Část o dodatečných omezeních v současné verzi návrhu dělá z GPLv3 guláš možných omezení, která se budou tvůrcům distribucí velmi těžko právně postihovat.

A ustanovení o patentech: Ze současného návrhu to vypadá, že by mohlo být potenciálně ohroženo celé patentové portfolio firmy jen kvůli umístění GPLv3 programu na jejich webovou stránku.

Dokument uzavírá:

Tři klíčové stížnosti popsané v části 5 jsou pro nás samostatně i dohromady dostatečným důvodem k odmítnutí současného návrhu licence. Rovněž však podotýkáme, že bez těchto nepřijatelných ustanovení představuje současný návrh přinejlepším jen velmi malou přidanou hodnotu oproti ověřené GPLv2. Následná diskuze obsahuje mnohá objasnění od Linuse Torvaldse. Když se objevil názor, že si měl výslovně ponechat právo na úpravu licence celého jádra, poukázal na to, že to lépe funguje při sou-

časném modelu, kdy nemá pevnou kontrolu nikdo: Nezapomínejte, že perfektní je nepřítelem dobrého. Když chceš věci, které jsou „teoreticky“ perfektní, většinou dopadnou „v praxi“ hrozně. Takže když nemáme žádného šéfa, mohlo by to *teoreticky* působit problémy. Ale *v praxi* je to o dost lepší, než mít někoho, s kým je potřeba si dělat starosti.

Také zdůraznil, že linuxové jádro není projekt FSF: Vždycky jsem to říkal velmi jasně: Linux je *open source*. Nikdy nešlo o projekt FSF a vždycky bylo hlavní dávat zdrojový kód zpět a udržovat ho otevřený – nic jiného.

A ještě upřesnil: Celé to přejmenovávání na „Open Source“ proběhlo *právě* proto, že se lidi chtěli od FSF distancovat. Skutečnost, že FSF a její stoupenici odmítli akceptovat název „Open Source“ a dál nazývají Linux „Free Software“, to není naše chyba.

■

Odkazy

- [1] <http://lwn.net/Articles/200745/>
- [2] <http://lwn.net/Articles/200698/>
- [3] <http://lwn.net/Articles/201440/>
- [4] <http://lwn.net/Articles/200656/>
- [5] <http://lwn.net/Articles/198344/>
- [6] <http://netlabel.sourceforge.net/>
- [7] http://sourceforge.net/docman/display_doc.php?docid=34650&group_id=174379
- [8] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-20.-9.-2006#sledovaci-infrastruktury>
- [9] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-20.-9.-2006#nopage-a-nopfn>
- [10] <http://lwn.net/Articles/200298/>
- [11] <http://lwn.net/Articles/170954/>
- [12] <http://lwn.net/Articles/156921/>
- [13] <http://lwn.net/Articles/201100/>
- [14] <http://kerneltrap.org/node/7160>

Jaderné noviny – 4. 10. 2006

Aktuální verze jádra: 2.6.18. Citáty týdne: Alan Cox, Linus Torvalds, Andrew Morton. Další věci pro 2.6.19. Změny API: zpracovávající přerušování a vektorované I/O. Poslední Wireless Extensions? Prezentace a fotografie ze setkání Kernel Netconf 2006. Uspání a probuzení. Kvůli varováním utíkají chyby.

Robert Krátký

Aktuální verze jádra: 2.6.18

Aktuální verze jádra 2.6 je i nadále 2.6.18; do hlavního repozitáře stále proudí patche pro připravovanou verzi 2.6.19-rc1.

Aktuální verze -mm stromu je 2.6.18-mm3 [1]. Mezi nedávné změny patří patch, který odstraňuje spoustu zbytečných varování kompilátoru, podpora swapového souboru pro softwarové uspání a **subsystém kevent** [2]. Aktuální předverze řady 2.4 je 2.4.34-pre4, vydaná [3] 2. října. Tentokrát bylo začleněno jen několik oprav; vypadá to, že se blíží finální stabilizační fáze 2.4.34.

Citáty týdne: Alan Cox, Linus Torvalds, Andrew Morton

Vyhrazuji si právo se jednou pokusit žalovat lidi, kteří můj kód cpou do Tivo. Třeba prohrají, ale toho práva se nevzdávám.

– Alan Cox [4]

Mexičani mají chupacabru, my máme Al Vira. Když ho zaslechnete řvát, modlete se, že se chystá rozcupovat kód někoho jiného. Utíkat nemá cenu.

– Linus Torvalds [5]

Vypadá to, že celé tohle jaderné snažení je neko-nečné spiknutí proti chudince mému malému Vaio.

– Andrew Morton [6]

Další věci pro 2.6.19

Přívalem patchů do hlavního repozitáře pokračuje vysokým tempem; od shrnutí z minulého týdne [7] bylo začleněno dalších několik tisíc patchů. Následuje seznam toho nejdůležitějšího – nejprve uživatelské věci:

- Konečně byl začleněn **clusterový souborový systém GFS2** [8]; obsahuje svou vlastní implementaci distribuovaného správce zámků.
- Nové ovladače: USB porty MCS7840 USB. USB řadiče ELAN U132, USB-na-PCMCIA adaptéry ELAN Uxxx, vibrační zařízení Playstation 2 Tran-

ce, Super-I/O čip VIA VT1211, monitory teploty procesoru AMD K8, tunery Philips TDA 10086 a TDA826x, USB mosty DiBcom založené na DiB0700, tunery Hauppauge Nova-T 500, adaptéry TI Flash Media PCI74xx a PCI76xx, komunikační koprocesory QUICC Engine a HP Quicksilver AGP GART.

- Byl vylepšen kód NFS serveru, včetně možnosti provádět I/O přes TCP spojení v mnohem větších blocích.
- Začleněn šifrovací souborový systém **eCryptfs** [9].
- IPSec kód získal podporu režimu **Bound End-to-End Tunnel (BEET)** [10].
- Ovladač USB hraček [gadgets], který připojí gadget rozhraní k ALSA MIDI subsystému. Účelem je umožnit systému vystupovat jako MIDI streamovací zařízení připojené k USB.
- Souborový systém tmpfs nyní může používat POSIXové ACL.
- Je-li do `/proc/sys/kernel/core_pattern` zapsán řetězec `|program`, budou všechny výpisy posílány přes rouru `programu`, místo aby byly zapisovány na disk.
- Byly začleněny některé z prvních kontejnerových patchů, včetně samostatných jmenných prostorů pro utsname informace a objekty SYSV IPC.
- Byl odstraněn bezpečnostní modul BSD secure level.
- Subsystém „floppy tape“ byl označen pro vyřazení v 2.6.20; nikdo ho nespravuje, nemá pravděpodobně žádné aktivní uživatele a jeho ukládací kapacita 1,6 GB je v současné době už trochu kuriózní. Pokud na tomto médiu někdo opravdu měl důležitá data, měl je už překopírovat na něco novějšího.

Změny viditelné pro vývojáře jádra:

- Bylo začleněno SRCU – verze read-copy-update, která umožňuje blokování na straně čtení.
- Mnohem lepší podpora uspávání a probouzení pro USB vrstvu.
- Byla přidána nová sada funkcí, které USB ovladačům umožňují rychle zjistit směr a režim přenosu koncového bodu.
- Poněkud okleštěná verze Wireless Extensions 21. Většina původních funkcí byla odstraněna, přičemž se spoléhá na to, že budou WE brzy nahrazena něčím jiným.
- Velké množství anotací, které utilitě `sparse` umožňují odhalovat big/little endian chyby.
- Několik Video4Linux ovladačů bylo převedeno na V4L2 API.
- Pole `flags` v `struct request` bylo rozděleno na dvě nová pole: `cmd_type` a `cmd_flags`. První obsahuje hodnotu, která popisuje typ požadavku (požadavek souborového systému, správa napájení atd.), kdežto v druhém jsou parametry, které upravují způsob, jak příkaz funguje (čtení/zápis, bariéry atd.).
- Vrstva blokových zařízení může být při konfiguraci jádra úplně vypnuta; může se hodit u některých embedded zařízení.
- Jádro teď má obecný boolean typ nazývaný *bool*; nahrazuje několik podomácku spíchnutých boolean typů z různých částí jádra.
- Nová funkce pro alokaci kopie bloku paměti:

```
void *kmemdup(const void *src, \
              size_t len, gfp_t gfp);
```

Na mnoha místech byly sekvence alokuj-pakopíruj upraveny tak, aby používaly `kmemdup()`.

- Byl začleněn patch s [infrastrukturou pro sledování latence](#) [11].
- Byly odstraněny metody `readv()` a `writev()` ze struktury `file_operations`; nahradily je `aio_readv()` a `aio_writev()` (jejichž prototypy byly změněny). Vizte dále.

V tuto chvíli je ještě možné začleňovat další patche, takže se do 2.6.19 mohou dostat další významné změny.

Změny API: zpracovávачe přerušení a vektorované I/O

Za normálních okolností by vydání 2.6.19-rc1 signalizovalo zpomalení vývoje a hledání chyb. Tentokrát to však bude možná vypadat jinak, protože mezi -rc1 a -rc2 dojde pravděpodobně k začlenění velké (změněno téměř 1100 souborů) změny API. Důvody jsou následující: patch, který zasáhne tolik souborů, se zcela jistě srazí i mnoha změnami, které teď proudí do hlavního stromu. Pozdržení patche do doby, až současný příliv utichne, by mělo všem zjednodušit práci. Takže o co jde? Zpracovávачe přerušení v současné době vypadají takto:

```
irqreturn_t handler(int irq, void *data, \
                    struct pt_regs *regs);
```

Struktura `regs` obsahuje stav registrů procesoru v okamžiku přerušení. Tento stav je předáván každému zpracovávачi přerušení, ale skoro nikdy se nepoužívá; z pohledu většiny zpracovávачů přerušení je stav registrů před přerušením jen hrstkou náhodných bitů. Předávání ukazatele však má podle Davida Howellse svou reži:

Ukazatel registrů se používá jen na několika místech, ale jeho předávání nás může stát jak místo ve stacku, tak nadbytečný kód. Na architektuře FRV má odstranění parametru `regs` ze všech `genirq` funkcí za následek zrychlení o 20 procent (tj. od konce `timer_interrupt()` do konce `do_IRQ()`).

David tedy sestavil [patch](#) [12], který ze zpracovávачů přerušení parametr `regs` odstraňuje. Pokud nějaký kód registry skutečně potřebuje – zjevně jde pouze o zpracovávач přerušení časovače – může ukazatel získat zavoláním nové funkce `get_irq_regs()`. A protože tento krok vyžaduje opravu každého zpracovávачe přerušení v systému – a těch je v hlavním jádře hodně – jde o velký patch, který mění hodně souborů.

Patch se právě objevil, takže podle normálních pravidel by už bylo trochu pozdě pro začlenění do 2.6.19. Během této verze by tedy seděl v -mm a do jádra by šel v 2.6.20. Ale [Andrew Morton](#) k tomu řekl [13]:

Myslím, že je to dobrá změna. Ale nechce se mi takového bumbříčka spravovat dva měsíce mimo strom! Pokud to chceme, měli bychom to tam prostě vrazit a zatnout zuby.

Nikdo proti této změně zjevně neprotestuje, i když Linus [vzpomenul](#) [14] krutý úděl lidí, kteří spravují ovladače mimo hlavní jádro. Všechno ukazuje na

brzké začlenění, které bude možná doprovázet speciální definovaný symbol, aby mohli správci nezačleněného kódu napsat kód, který by fungoval s oběma prototypy.

Odjinud: Strukturu `file_operations` lze nalézt v útrokách snad každého subsystému, jenž provádí I/O. Zatímco ovladače znakových zařízení vytvářejí strukturu `file_operations` přímo, většina ostatních částí systému (souborové systémy, síťové protokoly a ovladače, ovladače blokových zařízení) je schovává na vyšší úrovni. Dva ze členů struktury jsou:

```
ssize_t (*aio_read) (struct kiocb *iocb, \
    char __user *buf,
                    size_t len, loff_t pos);
ssize_t (*aio_write) (struct kiocb *iocb, \
    const char __user *buf,
                    size_t len, loff_t pos);
```

Tyto metody implementují asynchronní čtení a zápisy – operace, které mohou být dokončeny po té, co se původní volání vrátí do uživatelského prostředí. Dlouhodobě známým nedostatkem implementace asynchronního I/O v Linuxu je absence vektorovaných operací; každé AIO volání může pracovat jen s jediným bufferem. Jádro 2.6.19 tento nedostatek smaže za cenu změny dvou výše uvedených prototypů:

```
ssize_t (*aio_read) (struct kiocb *iocb, \
    const struct iovec *iov,
                    unsigned long niov, loff_t pos);
ssize_t (*aio_write) (struct kiocb *iocb, \
    const struct iovec *iov,
                    unsigned long niov, loff_t pos);
```

Jediný buffer byl nahrazen polem struktur `iovec`:

```
struct iovec
{
    void __user *iov_base;
    __kernel_size_t iov_len;
};
```

Jednobufferová volání jsou teď obalena jednou jedinou strukturou `iovec` a předána novým, vektorovaným verzím AIO operací. Všechn kód, který poskytuje `aio_read()` a `aio_write()`, bude muset být aktualizován s ohledem na nové API – a na možnost, že se po něm možná bude chtít provádění vektorovaných operací.

Změny jdou však ještě dál, protože byly odstraněny `file_operations` metody `readv()` a `writev()`. Příslušná systémová volání jsou teď implementována

voláním `aio_read()` a `aio_write()`. Převedení starých metod `readv()` a `writev()` není příliš obtížné, protože není nutné, aby byly

`aio_read()` a `aio_write()`

asynchronní (vlastně jim bude v tomto případě předáno „synchronní KIOCB“, což značí, že je operaci nutné provést synchronně). Ve většině případů jde prostě o přebrání nového prototypu a v případě potřeby také nalezení ukazatele

`struct file` v `iocb->ki_filp`.

(Vizte [článek z minulého února](#) [15], kde jsou o této změně podrobnější informace.)

Poslední Wireless Extensions?

„Wireless extensions“ [rozšíření pro bezdrátové sítě] je API založené na `ioctl()`, které uživatelskému prostoru umožňuje ovládat parametry specifické pro bezdrátová síťová rozhraní – ESSID, šifrovací hesla, kanály atd. API dlouho spravoval Jean Tourrilhes; posledních několik verzí jádra mělo verzi 20. V tuto chvíli už byla do pre-2.6.19 začleněna verze 21, ale přinejmenším některé části možná zase půjdou pryč.

Potíž je v tom, že verze 21 obsahuje tak velké změny, že by s ní již starší verze nástrojů nefungovaly. Konkrétně jde o způsob formátování ESSID předávaného jádru. Konfigurace, které se mohly k určité síti připojit s verzí 20, to už s verzí 21 nezvládnou. Je možné to obejít přidáním mezery do řetězce ESSID, ale o tom nebude většina uživatelů vědět, a i kdyby, tak na to přijdou až po upgradu jádra, když zjistí, že už se nemohou připojit k síti.

Od chvíle, kdy se o tomto problému začalo mluvit, dali mnozí vývojáři (včetně [Linuse](#) [16]) najevo, že je taková změna API nepřijatelná. Pro změnu samozřejmě existují pádné důvody – způsob, jakým se s těmito řetězci v protokolech zachází, se během času změnil. Ale správným řešením by bylo přidat novou `ioctl()`, která by se o nový formát řetězce postarala; starší verze by byla podporována navždy. Kdyby to bylo provedeno takto, změna formátu by nevadila.

To se zdá jako dobré řešení, ale má to malý háček. Jean už tento problém patrně delší dobu předvídal. Aby minimalizoval škody, vydával už asi půl roku Wireless Tools [nástroje], které formátu verze 21 rozumí. Dost distribucí už také tyto nástroje přebralo a zařadilo mezi své balíčky; jde například o Slackware 11 a Mandriva 2007. Pokud tyto nástroje přijdou do styku s Wireless Extensions verze vyšší než

20, budou očekávat nový formát ESSID; kdyby tedy byly změny ve verzi 21 odstraněny, přestanou tyto nástroje fungovat.

Takže Wireless Extensions 21 způsobí na nějakých systémech problémy bez ohledu na to, jestli bude změna formátování ESSID ponechána nebo ne. Jediným způsobem, jak v tuto chvíli předejít chybám na nasazených systémech, je podržet Wireless Extensions na verzi 20. Bezdrátová rozšíření už asi nebudou dále rozšiřována.

Pokud to dopadne takhle, bude to krátkodobě znamenat určité nepříjemnosti, protože se do API nebudou moci dostat potřebná vylepšení. Dlouhodobě je však stejně v plánu Wireless Extensions nahradit; proto je vyvíjeno nové netlink API nazývané nl80211. Toto API je však úzce svázáno se stackem [Devicescape](#) [17] 802.11, kterému však cesta k stavu vhodnému pro začlenění trvá o dost déle, než se očekávalo. Linuxové bezdrátové API tak možná bude nějakou dobu na mrtvém bodě.

Prezentace a fotografie ze setkání Kernel Netconf 2006

David Miller vystavil [prezentace a fotky](#) [18] z konference linuxových vývojářů síťování. Pokud vás zajímají hardcore podrobnosti, najdete jich tam spoustu.

Následující obsah je ©KernelTrap

Uspání a probuzení

28. zář, [originál](#) [19] Nedávná diskuze na [LKML](#) [20] se věnovala aktuálnímu stavu uspávání a probouzení [suspend a resume] v linuxovém jádru. Nigel Cunningham reagoval na patch pro uswsusp: **Pánové! Proč nechcete pochopit, že všechny tyto pokusy o uswsusp jsou čiré bláznovství?**

A pokračoval zopakováním svého přesvědčení, že důležité otázky uspávání mají být řešeny v jádře, a přesun do uživatelského prostoru ničemu nepomůže. Andrew Morton se připojil: **Navrhuji trápit mozkové závitky prací na tom, aby byla současná implementace uspávání hezky funkční, stabilní a rychlá. V tuto chvíli se kód suspend-to-disk dlouze zabývá podivnými-ale-asi-zbytečnými věcmi, místo aby zapisoval paměť na disk. Nemám tušení, co to všechn**

ten čas dělá, ale vsadil bych se, že to nikomu není moc k užítku ;-) Dále se diskutovalo o tom, proč je tedy uspávání tak pomalé, a jak vyřešit základní problémy, které brání spolehlivému a stabilnímu fungování. Nigel hlásil: **Jedním z největších problémů, se kterými se pravidelně potýkám, jsou chyby ovladačů.**

Ačkoliv základní kód funguje, mnoho ovladačů pořád není schopných se nechat úspěšně uspat a probudit. Vývojáři se shodli, že tyto chyby by měly být sledovány prostřednictvím bugzilly a někdo by se jimi měl zabývat. Andrew Morton odpověděl: **Vím o tom... brzy se mi snad podaří tomu věnovat více času. Prozatím se prosím postarejte o to, aby ty problémy byly v bugzille – důležité je vědět, kdo je nahlásil.**

Kvůli varováním utíkají chyby

3. říj, [originál](#) [21] Jeff Garzik podotkl, že novější verze GCC jsou víc a víc upovídáné: **Množství varování při kompilaci jádra v poslední době stoupl natolik, že se v nich ztrácejí chyby a i jinak znepríjemňují život.**

Založil nový strom „gccbug“, ve kterém začal utiňovat falešná varování (po ověření, že jsou skutečně falešná). **Audit už odhalil několik menších chyb, což podporuje moje tvrzení, že přílišné množství varování ukrývá chyby.**

Daniel Walker připomněl dřívější [zprávu](#) [22] o vytvoření makra, které se o plané popluchy stará, a shrnul tehdejší diskuzi: Nejde o to, že by se kód po utišení varování nedostalo řádné kontroly. Ale budoucí změny by mohly způsobit problémy, které by byly také skryty. Nemyslím si však, že by to byl velký nedostatek, zvláště když je možné při konfiguraci makro vypnout. Andrew Mortonovi se makro líbilo: **Je pravda, že ta falešná varování jsou spíše na škodu, a opravdové problémy mohou být přehlédnuty. Takže má určitě cenu se pokoušet nalézt způsob, jak vývojářům a testerům ukázat ta skutečně důležitá varování. Nevýhodou je, že to trochu zaneřádí kód a vznikne menší riziko skrytí pravých „use-uninitialised“ chyb. Ale myslím, že přednosti převažují nad nedostatky.**

Odkazy

[1] <http://lwn.net/Articles/202595/>

[2] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-23.-8.-2006#kevents-a-nova-api>

[3] <http://lwn.net/Articles/202443/>

[4] <http://lwn.net/Articles/202872/>

- [5] <http://lwn.net/Articles/202873/>
- [6] <http://lwn.net/Articles/202874/>
- [7] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-27.-9.-2006#zacina-vyvojovy-cyklus-2.6.19>
- [8] <http://sources.redhat.com/cluster/>
- [9] <http://lwn.net/Articles/156921/>
- [10] <http://www.ietf.org/internet-drafts/draft-nikander-esp-beet-mode-06.txt>
- [11] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-30.-8.-2006#api-pro-urceni-limitu-latence>
- [12] <http://userweb.kernel.org/~akpm/irq-maintain-regs-pointer-globally-rather-than-passing-to-irq-handlers.patch>
- [13] <http://lwn.net/Articles/202453/>
- [14] <http://lwn.net/Articles/202455/>
- [15] <http://lwn.net/Articles/170954/>
- [16] <http://lwn.net/Articles/202839/>
- [17] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jn-reportaz-wireless-networking-summit-2006#devicescape>
- [18] <http://vger.kernel.org/netconf2006.html>
- [19] <http://kerneltrap.org/node/7176>
- [20] <http://www.abclinuxu.cz/slovník/lkml>
- [21] <http://kerneltrap.org/node/7193>
- [22] <http://lkml.org/lkml/2006/5/11/50>

Jaderné noviny – 11. 10. 2006

Aktuální verze jádra: 2.6.19-rc1. Citát týdne: David Woodhouse. Odstranění `populate()`, `nopfn()` a `nopage()`. Video4Linux2 API: úvod. Budoucnost vývoje ReiserFS.

Robert Krátký

Aktuální verze jádra: 2.6.19-rc1

Aktuální předverze je 2.6.19-rc1, vydaná [1] 4. října. Shrnutí změn najdete v článcích [Začíná vývojový cyklus 2.6.19](#) [2] a [Další věci pro 2.6.19](#) [3] z posledních dvou týdnů. To hlavní: [ovladače pro paralelní ATA](#) [4], [labeled networking](#) [značené síťování] [5] pro IPsec a CIPSO a několik nových architektur, množství nových ovladačů, [clusterový souborový systém GFS2](#) [6], [eCryptfs](#) [7] a spousta interních změn.

Dlouhý changelog [8] obsahuje podrobnosti – ale protože se jedná o téměř 5000 patchů od více než 600 lidí, je lepší mít na to hodně volného času. [Krátký changelog](#) [9] je trochu stručnější, ale přesto dost dlouhý.

V tuto chvíli by už měly být do hlavního repozitáře začleňovány pouze opravy. Hodně patchů opravy opravdu jsou, ale Linus přidal i několik významných změn. Podle očekávání tak byla začleněna [změna prototypu zpracovávající přerušování](#) [10], což ovlivnilo celý strom. Konečně byl odstraněn zasloužilý, leč dlouho nepoužívaný soubor `<linux/config.h>`. Přidán byl také vývojový souborový systém ext4, který obsahuje [mnohá vylepšení](#) [11], včetně podpory rozsahů a 48bitových čísel bloků. Vizte [soubor s dokumentací ext4](#) [12], pokud si s ext4 chcete hrát (a máte dobré zálohy).

Aktuální verze -mm stromu je 2.6.19-rc1-mm1 [13]. Mezi nedávné změny patří přidání ext4 (který se hned přesunul do hlavního jádra), pokračující práce na mechanismu „swap token“, obecná implementace `log2()` a patch s dynamickým „tikem“ (vizte také [API pro určení limitu latence](#) [14]).

Citát týdne: David Woodhouse

Spravování ovladačů mimo hlavní jádro je i během těch světlejších chviliek bezostyšné sebemrškačství. Ale opravdu je nám to jedno – kdybychom jim neztěžovali život takhle, akorát by si začali píchat špendlíky pod nehty, aby si ten nedostatek bolesti vynahradili. Když se nad tím trochu zamyslete, tak jim vlastně pomáháme – takhle si aspoň neublíží.

– David Woodhouse [15]

Odstranění `populate()`, `nopfn()` a `nopage()`

VMA operace `nopfn()` byla přidána do 2.6.19-rc1; vizte článek z minulého měsíce: `nopage()` a `nopfn()` [16]. Ukazuje se však, že `nopfn()` bude možná z těch rozšíření API, které v jádře vydrží jen rekordně krátkou dobu; Nick Piggin poslal [sadu patchů](#) [17], které přinášejí výrazné změny v zacházení s výpadky stránek na nejnižších úrovních.

Struktura `vm_operations_struct` v 2.6.19-rc1 definuje tři metody, které se o nízkourovňové stránkování starají:

```
struct page *(*nopage)(struct vm_area_struct \
    *area,
                        unsigned long address, int *type);
unsigned long (*nopfn)(struct vm_area_struct \
    *area,
                        unsigned long address);
int (*populate)(struct vm_area_struct *area, \
    unsigned long address,
                unsigned long len, pgprot_t prot,
                unsigned long pgoff, int nonblock);
```

Obyčejně jsou výpadky stránek řešeny prostřednictvím `nopfn()` (existuje-li) nebo `nopage()`. Tyto fce by měly vzít danou `address` a přiřadit ji ke stránce ve fyzické paměti. U oblastí virtuální paměti (VMA), které mají zálohu v souborech, reaguje vrstva souborového systému na volání `nopage()` alokováním stránky paměti a načtením příslušného obsahu ze zálohy. Stránka je pak předána zpět jádru, aby byla začleněna do tabulky stránek. Ovladače zařízení, které `nopage()` implementují, obvykle jen přeloží `address` do příslušného ukazatele bufferu v paměti, který je mapován do uživatelského prostředí.

`nopfn()` i `nopage()` předpokládají, že je mapování mezi adresami virtuální paměti a offsetem v rámci VMA lineární – proto je jako parametr poskytována pouze adresa. Jádro však podporuje také [nelineární mapování](#) [18]; v takovém případě může aplikace změnit VMA na komplexní okno do jiných částí zá-

lohového souboru. `nopfn()` a `nopage()` tato mapování nezvládají, protože nemají potřebné informace. Místo toho musí každá záloha, která podporuje nelineární mapování, poskytnout metodu `populate()`, která má parametry jak pro `address` virtuální paměti, tak přiřazený offset (`pgoff`) do zálohového zařízení.

Na scénu vstupuje Nick, který pracoval na záluďném problému v jedné z tradičně nejzáluďnějších částí jádra: kódu, který se stará o zkracování souborů. V některých případech mohla být stránka, která byla odstraňována na základě volání `truncate()`, zároveň zpracována pomocí `nopage()`, což vedlo ke zmatkům ve správě paměti. Když Nick přemýšlel nad zamykacími pravidly pro tyto operace, uvědomil si, že musí existovat lepší způsob. Výsledkem je nová VMA operace nazývaná `fault()`:

```
struct fault_data {
    struct vm_area_struct *vma;
    unsigned long address;
    pgoff_t pgoff;
    unsigned int flags;

    int type;
};

struct page *(*fault)(struct vm_area_struct \\
    *vma,
    struct fault_data *fdata);
```

Tato metoda má nahradit

`nopfn()`, `nopage()` a `populate()`.

Dojde-li k výpadku stránky, naplní jádro strukturu `fault_data` potřebnými informacemi: `address` přiřazená k výpadku v uživatelském prostoru, odpovídající offset `pgoff` a dva příznaky, které značí, jestli k výpadku došlo při zápisu, nebo jde o nelineární mapování.

Funkce `fault()` by měla najít stránku, která může uspokojit požadavek na offset `pgoff`; obvykle nebude `address` vůbec potřebovat. Funkce může buď vrátit přiřazenou `struct page`, nebo nastavit záznam v tabulce stránek přímo (pomocí něčeho jako `vm_insert_page()`) a vrátit `NULL`. Ať tak nebo tak, pole `type` by mělo být nastaveno podle druhu výpadku (major nebo minor). Pokud není možné výpadek zpracovat, měl by být do `type` vložen příslušný chybový kód.

Nickův patch se hned zbavuje metod

`nopfn()` a `populate()`. V současné době se `nopfn()` používá na jediném místě a starší API `populate()` nebylo nikdy mimo hlavní jádro příliš rozšířeno. Funkce `install_page()` je také určena k brzkému odstranění. Metoda `nopage()` je však často používána v ovladačích zařízení – uvnitř i vně jádra. Proto byla označena jako zastaralá a její odstranění je naplánováno na říjen příštího roku. Objevily se návrhy na dřívější odstranění `nopage()` (dejme tomu za půl roku), ale žádné definitivní rozhodnutí nepadlo.

Až na tyto drobnosti to vypadá, že má tato změna všeobecnou podporu. Pro 2.6.19 jsou patche přeci jen trochu moc čerstvé, takže se v hlavním jádře pravděpodobně objeví nejdříve v 2.6.20. Ale v tu chvíli budou muset správci a autoři ovladačů provést aktualizaci svého kódu.

Video4Linux2 API: úvod

Jonathan Corbet: nedávno jsem psal ovladač pro kameru – konkrétně tu, která bude dodávána v systému [One Laptop Per Child](#) [19]. Ovladač používá jaderné API, které je pro podobné účely určené: Video4Linux2 API. Při psaní tohoto ovladače jsem zjistil něco vskutku šokujícího – toto API není zrovna dobře dokumentováno. I když uživatelská strana má [dokumentaci docela dobrou](#) [20]. Ve snaze o nápravu této situace bude v následujících měsících LWN vydávat sérii článků, které popisují, jak pro rozhraní V4L2 psát ovladače.

V4L2 má dlouhou historii – první jiskřička se v očích Bill Dirkse rozsvítila v srpnu roku 1998. Vývoj pokračoval dlouhé roky, až bylo V4L2 API konečně zařazeno do hlavního jádra v listopadu 2002 při vydání 2.5.46 [21]. Dodnes však novější API mnohé ovladače nepodporují; konverze pořád probíhá. Mezitím se V4L2 vyvíjí, přičemž do 2.6.18 šly docela zásadní změny. Aplikací, které by fungovaly s V4L2, moc není. V4L2 je navrženo pro podporu nejruznějších zařízení, z nichž jen některé jsou skutečně „video“:

- *Rozhraní pro zachytávání videa* bere video data z tuneru nebo kamery. Pro mnohé bude zachytávání videa hlavním uplatněním V4L2. Protože mám v této oblasti nejvíce zkušeností, budou se tyto články na zachytávací API zaměřovat trochu více, i když to není vše, co V4L2 nabízí.
- *Rozhraní pro výstup videa* aplikacím umožňuje ovládat zařízení, která umí poskytovat video obrázky – například formou televizního signálu – mimo počítač.

- *Overlay video rozhraní* je variací zachytávacího rozhraní, které se stará o přímé zobrazení video dat ze zachytávacího zařízení. Video data jdou přímo ze zachytávacího zařízení na displej, aniž by procházela přes procesor systému.
- *VBI rozhraní* poskytují přístup k datům přeneseným během blanking intervalu. Jsou dvě: „raw“ a „sliced“ – liší se úrovní hardwarového zpracování VBI dat.
- *Rádiové rozhraní* poskytuje přístup k audio proudům z AM a FM tunerů.

Jsou možné i jiné druhy zařízení. V4L2 API má názvy zařízení „codec“ a „effect“, která obě provádějí transformace proudů video dat. Tyto oblasti však ještě nebyly kompletně specifikovány – a o to méně implementovány. Ve starším V4L1 API také existují rozhraní „teletext“ a „radio data system“; nebyly však přesunuty do V4L2 a nevypadá to, že by se to někdy v dohledné době chystalo.

Video zařízení se od mnohých jiných liší tím, jak hodně je lze konfigurovat. V důsledku toho implementuje většina ovladače V4L2 kód, který aplikacím umožní objevit možnosti daného zařízení a nakonfigurovat jej tak, aby fungovalo podle představ. V4L2 API definuje několik desítek zpětných volání pro konfigurační parametry jako frekvence tuneru, okna a ořezávání, počet snímků, video komprimace, video formáty atd. Značná část tohoto seriálu se bude věnovat právě konfiguračnímu procesu.

Pak je tu ještě ten drobný zádrhel, že je nutno provádět efektivní I/O v dané kvalitě videa. V4L2 API definuje tři různé způsoby přenosu video dat mezi uživatelským rozhraním a zařízením, z nichž některé jsou dosti komplexní. O video I/O a vrstvě video-buf, která je připravena pro provádění běžných úkolů, budou samostatné články.

Následující obsah je ©KernelTrap

Budoucnost vývoje ReiserFS

12. říj, *Protože byl Hans Reiser* [22], zakladatel firmy Namesys, nedávno zatčen jako hlavní podezřelý v případě zmizení své manželky, zabývala se krátká diskuze na [LKML](#) [23] budoucností ReiserFS. Alan Cox poukázal na to, že *reiserfs je psán týmem lidí v Namesys a v případě reiserfs3 také lidmi ze SuSE i odjinud*. Alexander Lyamin, který je na stránkách Namesys uveden jako „hostmaster a sysadmin“, připojil, že tým je *v tuto chvíli dost otřesen a nervózní*.

Potvrdil, že ReiserFS 3.6 je v udržovacím režimu, a pak pokračoval o Reiser4: *Díky Andrew Mortonovi pořád provádíme revize. Vydáváme patche, opravujeme problémy a vůbec se snažíme vše pročistit*.

Takový je plán na dalšího alespoň půl roku. Ohledně budoucnosti poznamenal, že závisí na výsledku soudu: *Pokud to dopadne tak, jak doufáme, bude to v pořádku. Pokud to dopadne špatně, pak to bude složité. Pokusili bychom se jmenovat prostředníka, který by spravoval obchody Namesys*.

■

Odkazy

- [1] <http://lwn.net/Articles/203042/>
- [2] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-27.-9.-2006#zacina-vyvojovy-cyklus-2.6.19>
- [3] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-4.-10.-2006#dalsi-veci-pro-2.6.19>
- [4] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-6.-9.-2006#ovladace-pro-parallelni-ide>
- [5] <http://netlabel.sourceforge.net/>
- [6] <http://sources.redhat.com/cluster/>
- [7] <http://lwn.net/Articles/156921/>
- [8] <http://kernel.org/pub/linux/kernel/v2.6/testing/ChangeLog-2.6.19-rc1>
- [9] <http://lwn.net/Articles/203912/>
- [10] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-4.-10.-2006>
- [11] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-14.-6.-2006#ext3-pro-velke-souborove-systemy>
- [12] <http://lwn.net/Articles/203915/>
- [13] <http://lwn.net/Articles/203683/>
- [14] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-30.-8.-2006#api-pro-urceni-limitu-latence>
- [15] <http://lwn.net/Articles/203732/>
- [16] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-20.-9.-2006#nopage-a-nopfn>
- [17] <http://lwn.net/Articles/203689/>
- [18] <http://lwn.net/Articles/24468/>
- [19] [/blog/atom/2006/9/27/151679](http://blog.atom/2006/9/27/151679)
- [20] <http://v4l2spec.bytesex.org/spec/book1.htm>
- [21] <http://lwn.net/Articles/14568/>
- [22] <http://kerneltrap.org/node/7211>
- [23] <http://www.abclinuxu.cz/slovník/lkml>

Zprávičky

Mandriva Linux 2007 je venku

MandrivaLinux 2007 je venku. Podle MandrivaUser zatím jen pro i586, verze pro x86_64 bude co nevidět.

Ondřej Konečný 1.10.2006

Studie Pravda o správě Linuxu k připomínkování

OSS Alliance získala oprávnění na překlad a další distribuci studie (EMA – Get the Truth on Linux Management), která se zabývá skutečným porovnáním nákladů na správu systémů Linux a MS Windows. Překlad studie je nyní k dispozici k připomínkování.

Filip Molčan 2.10.2006

Křišťálová Lupa 2006: hlasování

Pět nejčastěji nominovaných projektů, služeb, osobností a společností českého internetu v každé kategorii soutěže Křišťálová Lupa 2006 postupuje do finálního veřejného hlasování, které potrvá až do 31. října 2006. Kategorie, hlasování.

Tomas Pelka 2.10.2006

Rozumíme IPv6 adresám

IPv6 adresy se od IPv4 adres značně odlišují a některé principy (např. multicast) fungují trochu jinak než obdoby ve starším protokolu. Proto na EnterpriseNetworkingPlanet připravili článek, který vám vše okolo nových adres a funkcí vysvětlí.

Luboš Doležel 2.10.2006

Závažná bezpečnostní chyba ve Firefoxu?

Skupina hackerů tvrdí, že objevila závažnou chybu v podpoře JavaScriptu ve Firefoxu. Chybu předvedli ve své prezentaci, která prý obsahuje dostatek informací pro reprodukci tohoto problému. Vývojáři z Mozilly se v současnosti snaží chybu najít a opravit.

Luboš Doležel 2.10.2006

Songbird – hudební přehrávač na jádře Mozilly

Songbird je software postavený na jádře Mozilly, který funguje jako multiplatformní hudební přehrávač a integruje se s různými on-line službami (např. Last.fm). Podobně jako Firefox je možné ho obohatit o další funkce pomocí rozšíření. Před časem už vyšla testovací verze 0.2.

Luboš Doležel 2.10.2006

Seriál o protokolu LDAP

Na serveru Linuxjournal.com se objevily dva články o LDAPu. První z nich popisuje historii protokolu a druhý popisuje instalaci Fedora Directory Serveru.

Jiří „Geo“ Lužnický 2.10.2006

Rhythmbox 0.9.6

Hudební přehrávač Rhythmbox dospěl do verze 0.9.6. Došlo k mnoha drobným opravám a vylepšením celého programu. Lépe si rozumí s DBUS. Podpora vypalování a přehrávání AudioCD byla přesunuta do pluginů.

Jiří „Geo“ Lužnický 2.10.2006

Letáková kampaň proti DRM

Krátký popis toho, jak to vypadalo při rozdávání letáků proti DRM u jednoho z obchodů Apple Store, naleznete v blogu jednoho z účastníků.

Jiří „Geo“ Lužnický 2.10.2006

Zkušenosti s Diskless Remote Boot in Linux

Frank Tuzi na Linux.com píše o svých zkušenostech s Diskless Remote Boot in Linux (DRBL). Jak název napovídá, DRBL je řešení tenkých klientů založené na bezdiskových stanicích, které bootují Linux ze sítě. Takové řešení tedy můžete vyzkoušet bez zásahu do již nainstalovaných systémů.

Luboš Doležel 2.10.2006

Slackware Linux 11

Po několika RC verzích je na světě Slackware Linux 11. Jako výchozí verze jádra je zvolena 2.4.33.3. Jako alternativa je k dispozici verze 2.6.17.13 a 2.6.18. Dále distribuce obsahuje například X.Org 6.9.0, GCC 3.4.6 a Apache 1.3.37. Na mirrorrech by se měl její ISO obraz objevit během několika dní. Zatím je k dispozici přes BitTorrent.

Jiří „Geo“ Lužnický 3.10.2006

Výchozím FS pro SUSE Linux už nebude ReiserFS

Výchozím souborovým systémem pro SUSE Linux byl dosud ReiserFS, od verze 10.2 se jím však stane ext3. Důvodem je nízký výkon ReiserFS při používání ACL a xattrs a malé množství vývojářů, kteří nyní kód ReiserFS udržují.

Luboš Doležel 3.10.2006

Programujeme s knihovnou OpenSSL

Třetí díl seriálu o programování s OpenSSL od IBM developerWorks se zabývá zpracováváním certifikátů. V předchozích dílech (1. a 2.) máte příležitost se naučit používat základní funkce této knihovny.

Luboš Doležel 3.10.2006

Vyšla hra World of Padman

Včera vyšla souhrnná verze komixově vypadající modifikace hry Quake 3 s názvem World of Padman. Pokud tuto hru ještě neznáte, podívejte se na nějaké obrázky. K dispozici je instalátor pro Linux a Windows.

Luboš Doležel 3.10.2006

Pohár Niké a open source

Pěkný příklad nasazení open source technologií můžete vidět na lode.knihajzd.info. Jedná se o sponzorské stránky věnované Mistrovství České Republiky v off-shore jachtingu. Opensource technologie jsou zde použity pro vizualizaci realtime dat pocházejících z cca. 30 jachet momentálně závodících někde na Jadranu.

Ivo Příkryl 3.10.2006

Scribus 1.3.3.4

S potěšením oznamujeme vydání Scribusu 1.3.3.4, což je multiplatformní DTP opensource software. Jedná se o opravnou verzi řady 1.3.3.x. K dispozici jsou instalační balíčky pro

Suse, Gentoo, Mandriva, Debian a MS Windows. Přičemž stále pracujeme na přepracované ultimátní verzi 1.4.

Petr Vaněk 4.10.2006

KDE 3.5.5 a nový vzhled kde.org

3. října se v Gentoo portage objevilo KDE 3.5.5. Na stránkách kde.org oficiální oznámení zatím není, ale můžete se alespoň podívat na jejich nový vzhled.

Mimi MIONS 4.10.2006

Qt 4.2

Softwarová společnost Trolltech, známá svým C++ frameworkem Qt, vydala jeho novou verzi 4.2. Mezi nejpodstatnější změny v nové verzi, kterou Trolltech označil za nejvýznamnější od vydání 4.0, patří nový Graphics View framework (vykreslování milionů 2D objektů s transformacemi v reálném čase), bližší integrace s Gnome (styl Cleanlooks, podpora pro glib eventloop = možnost Qt pluginů pro GTK+ aplikace a naopak), integrace s desktope obecně (system-tray, otevírání URL, pořadí tlačítek v dialogích), podpora pro Dbus a také možnost jednoduchého stylování widgetů pomocí CSS. Všechny hlavní změny s ilustracemi a povídáním najdete na doc.trolltech.com.

Boris Dušek 4.10.2006

Kódové označení FC6

Nuže, příznivci Fedory, pojdme si zahlasovat o nové kódové označení Fedory Core 6!

Pavel Šefránek 4.10.2006

Co bude následovat po Sysvinit

Už celé roky je na Linuxu používán spouštěcí init démon pocházející z Unix System V. Čím dál tím častěji se objevují projekty, které se snaží o jeho moderní náhradu, uvedme např. inittng nebo Upstart. Na Linux.com si můžete přečíst o úskalích jejich vývoje a nových nápadech.

Luboš Doležel 4.10.2006

Super Grub Disk – záchranné CD

Linux.com představuje záchranné CD nazvané Super Grub Disk. Jak už název napovídá, soustředí se na obnovu bootovacího sektoru GRUBu. Mimo to však umí obnovit i záznam pro LILO a Windows, nebo dokáže skrývat diskové oddíly.

Luboš Doležel 4.10.2006

Přepisujeme kód z PHP4 do PHP5

IBM developerWorks vám poradí, jak přepracovat váš PHP4 kód tak, aby využíval nových vlastností PHP5. Toto může znatelně zlepšit stabilitu a udržitelnost kódu, a to především tam, kde lze použít objektové programování.

Luboš Doležel 4.10.2006

Gnome 2.16.1

Vyšla nová verze ve stabilní větve desktopového prostředí GNOME. Více se o vydání verze 2.16.1 a změnách oproti 2.16.0 dozvíte na Gnomedesktop.org.

Jiří „Geo“ Lužnický 4.10.2006

NuSphere vydalo phpDock

NuSphere zahájilo distribuci runtime prostředí phpDock, kombinujícího webový server, webový prohlížeč a interpret php

do jednoho balíku. phpDock umožňuje použít nejrozšířenější webové technologie (PHP, HTML, AJAX) v desktopových aplikacích.

BoodOk 5.10.2006

Slamd64 11

3. října 2006 vyšla nová verze (č. 11) Slamd64 – 64bitového Slackware Linuxu. K novinkám patří gcc 3.4.6, KDE 3.5.4, Mozilla Firefox 1.5.0.7, oprava řady chyb a další. Více o projektu na www.slamd64.com. K dispozici jsou i ISO image: mirrors.

penguin666 5.10.2006

Google prohledává zdrojáky

Google uvedl novou stránku google.com/codesearch. Je možné využít bohatší sadu příkazů, než je tomu u běžného vyhledávače Googlu. Téměř ihned po té kdosi našel kód z Winzipu, který je odpovědný za generování seriových čísel. Zdroj digg.com.

elviin 5.10.2006

Jak na úkolník v konzoli

Nepamatujete si pořádně, co máte všechno za úkoly, ale různé PIM aplikace se vám zdají zbytečně těžkotonážní? Server Linux.com ukazuje použití skriptů, které na jednoduchý „úkolník“ stačí.

Jiří „Geo“ Lužnický 5.10.2006

FreeBSD 6.2-beta2

Vyšla druhá betaverze FreeBSD 6.2. Podle oznámení bylo opraveno několik chyb, ale na odstranění pár větších se zatím pracuje. Stahovat lze z mnoha mirrorů.

Jiří „Geo“ Lužnický 5.10.2006

FSF Europe připravuje web o DRM

Free Software Foundation Europe připravuje web drm.info, který má veřejnost informovat o nebezpečích Digital Restrictions Managementu. Hlavním heslem tohoto webu bude: „nevěřme vašim zařízením“.

Luboš Doležel 5.10.2006

Webový prohlížeč Dillo shání sponzory

Webový prohlížeč Dillo (jeho přednostmi jsou velikost a rychlost, nikoliv počet funkcí) shání sponzory. Pokud se najdou, přibude podpora pro CSS, JavaScript a hezčí text.

Jiří „Geo“ Lužnický 5.10.2006

Podíl Firefoxu stále roste

Ačkoliv mnoho skeptiků tvrdí, že se růst Firefoxu brzo zastaví, zatím se tak nestalo. Od srpna do září se podíl Firefoxu zvýšil o 0,7 % a globálně tak podle kalifornské společnosti Net Applications dosáhl 12,5 %.

Luboš Doležel 5.10.2006

Vládní podpora ODF ve Francii

Dokumenty vydávané francouzskou vládou by měly být dostupné ve formátu ODF a zároveň bylo navrženo, aby Francie požádala své evropské partnery o obdobné jednání. Také se doporučuje, aby v EU vznikl orgán, který by měl za cíl zajistit technologickou nezávislost. Další informace o návrzích Bernarda Carayona naleznete na InfoWorld.com.

Luboš Doležel

5.10.2006

The Corruptibles

Na stránkách Electronic Frontier Foundation (EFF) je ke stažení krátký animovaný film The Corruptibles upozorňující na největší současná nebezpečí DRM (odpůrci zkratku vykládají jako Digital Restrictions Management).

Michal Křenek

6.10.2006

Automatická konverze FLACu s MP3FS

Podpora bezztrátového kodeku OGG FLAC v hardware/software není příliš dobrá, a proto vznikl virtuální filesystém MP3FS. Tento filesystém založený na FUSE transparentně konvertuje OGG FLAC soubory v připojeném adresáři do formátu MP3. Podrobnosti o něm a jeho používání naleznete na Linux.com.

Luboš Doležel

6.10.2006

Recenze notebooku Lenovo ThinkPad T60p

Hardware in Review představuje notebook Lenovo ThinkPad T60p. Nejedná se o klasickou recenzi na hardware, v článku je testována i funkčnost tohoto notebooku pod linuxovými distribucemi, a to včetně softwarového nástroje ThinkVantage. Právě ThinkVantage a ovladače od ATI se zdají být velkou slabinou tohoto výrobku.

Luboš Doležel

6.10.2006

Porušování licence aplikace Konsole

Programy eKonsole a qonsole porušují licenci GNU GPL u známé aplikace Konsole. Tyto programy jsou v zásadě porty pro mobilní zařízení a byly šířeny přes fórum MotorolaFans.com. Jak se autorovi Konsole daří řešit tento problém, popisuje Linux.com.

Luboš Doležel

6.10.2006

Firefox 2 RC2 ke stažení

Mozilla Firefox 2 Release Candidate 2 byl uvolněn ke stažení. Více informací v poznámkách k vydání.

Jindřich Pozlovský

7.10.2006

V Debianu bude IceWeasel místo Firefoxu

Problém s použitím jména Firefox v Debianu byl vyřešen. Namísto přejmenování pomohou existujícímu projektu GNU IceWeasel, aby bylo možno brzo vydat první verzi.

Luboš Doležel

7.10.2006

Nový vzhled grafů v OpenOffice.org

Vývoj nové komponenty pro grafy v OpenOffice.org [zprávička] pokračoval. V článku na Linux.com se můžete podívat na informace a screenshoty této komponenty. Nová verze se bohužel objeví až v OpenOffice.org 2.6, o kterém není zatím jasné, kdy bude vydáno.

Luboš Doležel

7.10.2006

Srovnání ovladačů pro ATI R300

Phoronix.com porovnává proprietární a svobodný ovladač pro ATI R300. Zatímco svobodný ovladač podporuje rozšíření GLX_EXT_texture_from_pixmap, proprietární ovladač má vyšší výkon ve hrách a funguje s ním TV-out.

Luboš Doležel

7.10.2006

Ivan Bíbr končí v QCM

Hlavní postava české verze linuxové distribuce MandrivaLinux a jedna z hlavních postav společnosti QCM – Ivan Bíbr – podle informací portálu ITBiz.cz ukončila pracovní poměr u QCM. Fungování komunitního serveru Mandrivalinux.cz by jeho odchod neměl nijak ohrozit.

Robert Krátký

9.10.2006

Vydání Fedora Core 6 opožděno

Vydání Fedora Core 6 se zpozdí. Původně měla tato verze vyjít už 6. října, vydání se však z důvodu neopravených chyb posouvá na 17. říjen.

Luboš Doležel

9.10.2006

Lepší využití SMP systémů s xjobs

Xjobs je konzolový nástroj se kterým můžete lépe využít možnosti SMP systémů – umožňuje snadno paralelizovat provádění úloh. Přesnější popis funkce s ukázkami je na Linux.com.

Luboš Doležel

9.10.2006

Microsoft podpořil SCO ve sporech proti IBM

Dřívější zvěsti ohledně role Microsoftu ve sporech SCO proti linuxovým společnostem se ukázaly být pravdivé. Podle různých článků, které se nyní objevují na Internetu, Microsoft finančně podpořil SCO, ale tak, aby to nebylo na první pohled jasné – skrze společnost BayStar. Na Groklaw.net je celé dění podrobně popsáno.

Luboš Doležel

9.10.2006

Podcast s Timothee Bessetem na téma her

Zástupci LinuxGames.com pohovořili s Timothee „TTimo“ Bessetem, linuxovým vývojářem z ID Software. Tématem rozhovoru, který si můžete stáhnout ve zvukové podobě, byly samozřejmě hry pro Linux (Quake 3, ET: Quake Wars, Defcon) a lidé kolem nich.

Luboš Doležel

9.10.2006

Tony Mobily o středně velkých OSS projektech

Tony Mobily v blogu na FreeSoftwareMagazine.com píše o problémech středně velkých svobodných softwarových projektů – tedy programů, které už mají dostatek uživatelů, ale ještě nedosáhly na ty velké projekty, které si mohou užívat výhod finanční podpory. Takové projekty jsou často následkem hádek odsouzeny k zániku.

Luboš Doležel

9.10.2006

Automatická replikace dat mezi MySQL servery

Vysoké dostupnosti MySQL serveru se dá dosáhnout replikací dat mezi dvěma servery („master-master replication“), kdy druhý server je schopen převzít obsluhu požadavků v případě výpadku prvního, a to bez ztráty dat. HowtoForge vás navede správným směrem k takové konfiguraci.

Luboš Doležel

9.10.2006

Slax 5.1.8

Vyšel Slax 5.1.8. Nová verze této mini live distribuce obsahuje KDE 3.5.4, opravy ze Slackware 11 a plnou podporu NTFS. Zajímavostí je možnost naboťovat z CD, které si pak následně tahá data z USB disku. Kompletní seznam změn, a stahovat lze ze slax.org.

Jiří „Geo“ Lužnický

9.10.2006

Jak je to s podporou tiskáren v Linuxu

Na serveru Linuxjournal.com vyšel článek popisující (spíše v obecné rovině) instalaci a používání tiskárny v Linuxu. Přestože vše fungovalo velmi dobře, má autor článku pocit, že Linux je co se týče podpory výrobců poněkud stranou.

Jiří „Geo“ Lužnický

9.10.2006

larch 2 – vytvořte si vlastní Arch Linux Live-CD

Vyšla nová verze larch-2 – konstrukční sady skriptů larch určené k vytvoření vlastního Arch Linux Live-CD na míru. Používá již nový mkinitcpio systém (využívající udev), podporuje různé profily konfigurace, umožňuje použít GRUB bootloader, obsahuje nový instalátor 'larchin' a umožňuje ukládat sezení na boot médium. Podobně jako Slax Linux Live skripty využívá unionfs (lze zapisovat na read-only médium) a squashfs (vynikající kompresní poměr).

Michal Křenek

10.10.2006

Setkání CZOSUG: Xen Bootcamp

V sobotu 21. října se na půdě ČVUT v Praze uskuteční celodenní setkání v rámci Czech Opensolaris User Group na téma Xen, jeho nasazení na různých platformách, užívání a nejen to. Bližší informace o sérii přednášek a ukázek můžete najít v publikovaném e-mailu.

Milan Jurik

10.10.2006

Soutěž programátorů C++

Společnost Seznam.cz pořádá 4. listopadu soutěž programátorů C++ pod Linuxem. Klání je omezeno na 15 účastníků, kteří se mohou pokoušet registrovat od 21. října a rozdat si to o notebooky v cenách od 20 do 50 tisíc korun.

Jiří Poláček

10.10.2006

N.E.R.O. – souboj počítačové inteligence

Linux.com představuje hru N.E.R.O. Jedná se o bojovou hru, ve které proti sobě bojují dvě umělé inteligence (AI) – pokud si vytrénujete chytrější umělou inteligenci, vítězství máte v kapse.

Luboš Doležel

10.10.2006

Elektronicky podepisujeme dokumenty OpenOffice.org

Věděli jste, že v OpenOffice.org můžete dokumenty elektronicky podepisovat? Pokud ne, přečtěte si návod od Linux.com na zřízení certifikátu od CACert a následně podepsání vašeho dokumentu.

Luboš Doležel

10.10.2006

Konzole Nintendo Wii možná poběží na Linuxu

Nová herní konzole Nintendo Wii (dříve Nintendo Revolution) měla prý běžet na Linuxu. Podle informací se zdálo, že na této platformě bude možno spouštět pouze digitálně podepsané programy, což by ze hry vyrazovalo většinu současných projektů pro Linux – AKTUALIZACE: celá informace o nasazení Linuxu na Wii se ukázala být pouhým výmyslem. Projekt Wii-Linux.com si každopádně běh Linuxu na této platformě klade za cíl.

Luboš Doležel

10.10.2006

Zprávy o vývoji nového Flash přehrávače

Na ZDNet.com se píše o stavu vývoje nového Flash pluginu do prohlížečů na Linuxu. Zatím je ještě hodně nestabilní, ale první betaverze by měla vyjít během dvou týdnů – do té doby se můžete podívat na screenshot.

Luboš Doležel

10.10.2006

Jak vypadá Firefox 2.0 RC2

Pro ty, kteří nesledují vývoj Firefoxu a ještě nevědí o vlastnostech, které nás čekají ve verzi 2.0, je na ArsTechnica krátké seznámení se zatím poslední předverzí RC2. V článku je navíc ukázána změna vzhledu mezi Beta 2 a RC2.

Luboš Doležel

10.10.2006

O WLAN Marvell čipu ve 100-dolarovém notebooku

Některým lidem, kteří se inspirovali názorem Theo de Raadta, se nelíbí fakt, že ve stodolarovém notebooku má být použit značně proprietární čip Marvell 88W8388 pro bezdrátové sítě. Jim Gettys odůvodňuje tuto volbu tím, že tento čip je jediný vhodný pro daný účel.

Luboš Doležel

10.10.2006

KDE 3.5.5

Dnes se na oficiálním mirroru KDE objevila nová verze 3.5.5. Na spoustě mirrorů ještě zdrojové kódy nejsou, nebo jsou přetížené – stahovat lze třeba z ftp.tiscali.nl. Dostupný je i seznam změn. Oficiální oznámení na KDE.org ještě není.

David Watzke

11.10.2006

Hans Reiser obvinený z vraždy

Známy tvorca filesystemu ReiserFS Hans Reiser bol včera (10.10.2006) zatknutý pre podozrenie z vraždy svojej manželky, ktorá je nezvestná od 3. septembra tohto roku. Viac na Slashdote.

Michal Žila

11.10.2006

Podvodné maily České spořitelny

Dnes večer se objevily ve schránkách mnoha uživatelů podvodné zprávy vybízející klienty České spořitelny k aktivaci nového bezpečnostního systému v rámci služeb internetového bankovníctví Servis24. Jedná se o HTML stránku s odkazem na server v Austrálii, kterou samozřejmě banka nespravuje. Pokud jste ji dostali také, informace v ní uvedené prosím ignorujte a nevkládejte do formulářů své přihlašovací údaje.

Na infolince služby Servis24 již o problému vědí. Podvodné stránky se nacházejí na adrese <http://202.157.132.58:9070/index.htm>. V době psaní zprávičky ještě prohlížeč Firefox podvod nerozeznal. Thunderbird si však byl jistý hned od počátku.

Viktor Fuglík

11.10.2006

Jaký software se používá v Hollywoodu

Jedním z důvodů, proč nVIDIA soustředí relativně mnoho práce do vývoje ovladačů pro Linux, je jeho četnost použití ve filmových studiích. Linux se tam dostal natolik daleko, že se stává naprosto běžnou záležitostí. Hollywoodský vývojář Robin Rowe připravil stránky LinuxMovies.org, kde se můžete podívat, jaký svobodný a komerční software je na Linuxu v Hollywoodu používán.

Luboš Doležel

11.10.2006

FreeDesktop.org Portland 1.0

FreeDesktop.org vydalo první verzi rozhraní a nástrojů Portland 1.0. Účelem Portlandu je umožnění snadné integrace aplikací s konfigurací svobodného desktopu, kterou se uživatel rozhodl používat. Více informací naleznete ve Wiki.

Luboš Doležel

11.10.2006

Přidáváme pluginy do DokuWiki

Příkladem toho, že nejen klasické aplikace mohou být obohacovány o pluginy, je webová aplikace DokuWiki. Na Linux.com se píše o tom, jak snadno může být tato Wiki rozšiřována o další funkce.

Luboš Doležel

11.10.2006

Přecházíme z Unixu na Linux v 7 krocích

Přes všechny podobnosti mezi Unixem a Linuxem se najdou rozdíly, které mohou váš přechod na Linux značně zkomplikovat. Pokud nevíte, co vás čeká, podívejte se na článek na SearchOpenSource.com, který vás přechodem provede v sedmi krocích.

Luboš Doležel

11.10.2006

Eudora bude open source

Firma Qualcomm oznámila, že upouští od současného vývojového modelu mailového klienta Eudora a že začíná pracovat na open-source verzi, která vyjde v první polovině r. 2007 a bude založená na Thunderbirdu. Děje se tak proto, že ačkoliv jde o výdělečnou záležitost, vývoj takového SW se neslučuje se zaměřením firmy (bezdrátové komunikace). Opensourcování udrží Eudoru při životě a Qualcommu ulehčí. Více na NewsForge.

Daniel Kvasnička ml.

12.10.2006

Linuxalt 2006 se blíží

Již za méně jak jeden měsíc se bude konat konference LinuxAlt 2006 v Brně. Na webové stránce www.linuxalt.cz je nyní k dispozici registrace a seznam přednášek, které budou prezentovány odborníky ve svém oboru. Na samotném programu se intenzivně pracuje a bude zveřejněn co nejdříve.

Michal Vyskočil

12.10.2006

Vývoj Reiser4 není ohrožen

Zatčení Hanse Reiser4 vyvolalo velké obavy o budoucnost vývoje Reiser4. Podle Linux.com není vývoj Reiser4 ohrožen, bude prý vynaloženo maximální úsilí, aby ruští a ukrajinští vývojáři Namesysu byli i nadále placeni a pokračovali v práci na tomto souborovém systému.

Luboš Doležel

12.10.2006

Oficiální oznámení o vydání KDE 3.5.5

Po včerejším vydání KDE 3.5.5 se na stránkách kde.org objevilo oficiální oznámení o vydání. Krom seznamu změn v něm najdeme například i seznam distribucí, ve kterých ho můžeme v nejbližší době očekávat.

Jiří „Geo“ Lužnický

12.10.2006

Začínáme s GIMPem

Free Software Magazine vás provede začátky používání grafického editoru GIMP, konkrétně se naučíte tvorbu banneru

pro vaše webové stránky. Nepřehlédněte ani seriál GIMP v příkladech zde na AbcLinuxu.

Luboš Doležel

12.10.2006

Správa verzí na Linuxu

IBM developerWorks pomocí příkladů ukazuje použití několika hlavních systémů správy verzí na Linuxu – CVS, Subversion, Arch a GIT. Mimoto v článku naleznete srovnání centralizovaných a distribuovaných repositářů.

Luboš Doležel

12.10.2006

Virtual Iron 3.0, konkurence pro VMware

Virtual Iron se rozhodl konkurovat virtualizačnímu software VMware uvolněním verze 3.0 svého produktu. Virtual Iron je založen na Xenu a pro svou funkčnost vyžaduje procesor Intel nebo AMD s hardwarovou podporou virtualizace.

Luboš Doležel

12.10.2006

Jak na udev

Článek zabývající se správou adresáře /dev, vyšel na serveru Networking Planet. Nejdříve nám připomene historii správy zařízení pomocí skriptu makedev, a pseudo souborový systém devfs. Dále se pak věnuje používání programu Udev. Tím se pak detailněji zabývá druhý díl článku. Viz také DevFS vs. udev.

Jiří „Geo“ Lužnický

12.10.2006

Zase jedna perla

Před časem tu proběhly asi dva odkazy na stránky jakési americké fanatičky pomlouvající vše včetně Linuxu. A tady je nový výtvar na Shelley The Republican.

Petr Vyhnal

12.10.2006

Sunbird a Lightning 0.3

Vyšla verze 0.3 organizérů Sunbird a Lightning (totéž jako Sunbird, ale integrované do Thunderbirdu). V nové verzi je předěláno uživatelské rozhraní a dále byly například opraveny chyby, které souvisely s lokalizacemi.

Jiří „Geo“ Lužnický

12.10.2006

Google Docs & Spreadsheets je online!

Google před několika dny spustil Google Docs & Spreadsheets? on-line aplikaci na tvorbu, publikaci a sdílení textových a tabulkových dokumentů. K použití je potřeba Google konto, mezi důležitými vlastnostmi je pak kooperace více uživatelů, možnost veřejného přístupu k dokumentu a export do Wordu, RTF, HTML, ODF i PDF. Více na TechCrunch.

Daniel Kvasnička ml.

13.10.2006

AbcHosting: platby i v slovenských korunách

Novinka pro slovenské klienty služby AbcHosting: Protože chceme být lepším partnerem a dodavatelem, jsme potěšeni, že vám můžeme nabídnout možnost úhrady ve slovenských korunách (byl zřízen účet u slovenské Tatra banky, a.s.).

Reklama

13.10.2006

Wine 0.9.23

Dnes vyšla nová verze Wine 0.9.23. Nepřináší velké změny, došlo k rozsáhlému přepsání printf výrazů pro kompatibilitu s Win64 architekturou, byla přidána podpora dynamických

jednotek pro MacOSX a opraveny některé chyby MSI. Podstatnou změnou pro testery je v implementace shaderů pomocí GLSL (není standardně aktivní, stále se využívá ARB), která nyní funguje s novými beta ovladači nVidia (bohužel už nechodí s verzemi staršími).

Mirek

13.10.2006

OpenOffice.org 2.0.4

Vyšla finální verze OpenOffice.org 2.0.4. Obsahuje vylepšený export do PDF, podporuje přímý export do LaTeXu, Calc a Impress jsou rozšířeny o nové funkce plus další novinky a opravy.

Luboš Doležel

13.10.2006

SUSE Linux 10.1 Remastered

Pokud hodláte instalovat distribuci SUSE Linux 10.1, sáhněte raději po „Remastered“ verzi. Ušetříte si tím čas, který byste strávili stahováním updatů. K dispozici je oznámení o vydání.

Jiří „Geo“ Lužnický

13.10.2006

Negativní zkušenosti s Cedegou

Na webu přibývá stížností na kvalitu Cedegy – ta totiž není taková jako tvrdí TransGaming. O negativních zkušenostech jednoho uživatele s Cedegou verze 5.2.6 se píše na linuxgamers.net.

Luboš Doležel

13.10.2006

O operačním systému Nexenta

Víte, co získáte kombinací OpenSolarisu, nástrojů GNU a Ubuntu? Správnou odpovědí je Nexenta. O vlastnostech tohoto neobvyklého systému se dočtete na Linux.com, nebo ho můžete přímo vyzkoušet.

Luboš Doležel

13.10.2006

Libye kupuje 1,2 milionu laptopů s Linuxem

Libye hodlá zakoupit 1,2 milionu levných notebooků s Linuxem. Stává se tak pátou zemí, která o tento výrobek projevila vážný zájem. Celkově ji to vyjde na 250 milionů dolarů.

Luboš Doležel

13.10.2006

Test Mandrivy 2007 PowerPack

Na SoftwareInReview.com otestovali novou Mandrivu 2007 edice PowerPack. Můžete se dočíst o nových funkcích, přidaných komerčních programech, ale i chybách, které se během testování projeví.

Luboš Doležel

13.10.2006

Tipy a triky při používání .htaccess

Soubor s názvem .htaccess je mocným nástrojem pro konfiguraci některých možností webu ze strany Apache. Jeho psaní není nijak zvlášť složité – pokud jeho použití ještě neznáte, podívejte se na dvoudílný článek o něm.

Luboš Doležel

13.10.2006

Ovladač nVidia 1.0-9626 BETA

nVidia pokračuje ve vydávání BETA ovladačů, včera byla uvolněna verze 1.0-9626. Stahujte verzi pro x86 a x86-64. Podrobnosti o změnách v této verzi nejsou známy, minimálně se však zdá, že ovladač funguje i bez odstranění podpory I2C.

Luboš Doležel

14.10.2006

Nové fórum uživatelů Fedora Core

Na českých stránkách Fedory bylo zprovozněno nové fórum pro uživatele distribuce. Výhodou by do budoucna měla být možnost zakládat více tematických fór, již současnou výhodou je větší přehlednost a možnosti fóra.

Jakub Talaš

14.10.2006

KDE slaví 10 let

Projekt K Desktop Environment (KDE) oslavil 10 let své existence. Slavnostní setkání proběhlo na Technické akademii Esslingen v Ostfildernu blízko německého Stuttgartu.

Luboš Doležel

14.10.2006

Studie EU: Ekonomický dopad open-source

Portál ITBIZ.cz ve spolupráci s organizací OSS Alliance zveřejnil závěrečnou zprávu studie ekonomického dopadu open-source na inovaci a konkurenceschopnost, kterou si nechala vypracovat Evropská komise.

Filip Molčan

16.10.2006

Mark Shuttleworth prvním patronem KDE

Mark Shuttleworth se stal prvním „patronem“ KDE. Patronem se může stát každá osoba nebo organizace, která chce KDE dlouhodobě (finančně) podporovat (viz stránka KDE e.V. o Supporting Members).

Robert Krátký

16.10.2006

Linphone 1.5.0

Vyšla nová verze 1.5.0 SIP VoIP klienta Linphone. Mezi novinky patří podpora MPEG4 a Theora videa a lepší výběr kodeků podle dostupného pásma.

petr_p

16.10.2006

Používáme cURL

cURL („client for URLs“) je konzolový nástroj pro stahování a uploadování souborů přes HTTP, FTP a další protokoly. Pokud se s ním naučíte dobře pracovat, bude vám dobrým pomocníkem nejen ve skriptech. Linux.com vás provede základy jeho používání.

Luboš Doležel

16.10.2006

Diskuze o nápadech pro Mozillu Firefox 3

Mozilla připravila stránku, kde probíhají diskuze o nových funkcích v Mozille Firefox 3. Stránka je umístěna ve Wiki, takže svými nápady může přispět opravdu každý.

Luboš Doležel

16.10.2006

Výsledky hlasování ohledně Debianu

V posledním kole všeobecného hlasování okolo budoucna Debianu se dohodlo, že vedoucí projektu zůstane na svém místě. Dále se odhlasovalo, že Debian Etch pravděpodobně vyjde dříve. Detaily naleznete na LWN.net.

Luboš Doležel

16.10.2006

O rozvoji OSS a svobodných formátů v Massachusetts

LinuxInsider píše o americkém státě Massachusetts, který se proslavil svým postupem ohledně používání open source a svobodných formátů. V článku se píše o odezvě a problémech s tímto projektem, který byl zahájen již v roce 2003.

Luboš Doležel

16.10.2006

Centrum.cz umožní telefonovat přes Skype

Poté, co byla ohlášena spolupráce Atlasu a ICQ, se Centrum dohodlo na spolupráci se Skype. To ještě více nahrává spekulacím, že Seznam bude nabízet ke svým účtům otevřený Jabber/Jingle.

mrzout

16.10.2006

Jaký význam má licence EUPL

InfoWorld píše o licenci European Union Public License, která sice není zatím hotová, ale už je k dispozici návrh. V článku se dočtete o postavení této licence mezi ostatními běžně užívanými open source licencemi.

Luboš Doležel

16.10.2006

Předvedeno budoucí jádro Gimp

Na festivalu Píksel 06 byla předvedena knihovna GEGL (Generic Graphical Library), která by měla v budoucnu nahradit současné jádro grafického editoru Gimp. Knihovna byla pro použití v GIMPu navržena už v roce 1999, ale zůstala v ústraní až do léta 2005. ?vind Kol?s se rozhodl tuto knihovnu oživit a předvedl její možnosti na serveru Linux.com

Jiří „Geo“ Lužnický

16.10.2006

Článek o Linuxu v ABC

V příštím čísle časopisu ABC vyjde článek o Linuxu s názvem „Fenomén Linux“, slibuje to úvodník příštího čísla na poslední straně ABC 21.

Belik

16.10.2006

CZOSUG BootCamp – Xen

Tuto sobotu pořádá CZOSUG akci BootCamp, která proběhne od 10 do 18 hodin na ČVUT na Karlově Náměstí. Tématem přednášek bude virtualizační systém Xen.

Jiří „Geo“ Lužnický

16.10.2006

Lobbisté v dopisu pro EK očerňují FLOSS

Iniciativa za volbu software (Initiative for Software Choice), jejíž členem je i Microsoft, používá lobbisty k tomu, aby Evropskou komisi upozornila na to, co považuje za rizika open source, např. ekonomické dopady. Dopis, který se podařilo získat, však napříkald nezohledňuje fakt, že více než polovina FLOSS vývojářů je za svou práci placena. Více na TheInquirer.net.

Luboš Doležel

16.10.2006

Firefox 2.0 RC3

Dňa 16.10.2006 vyšla ďalšia verzia oblúbeného prehliadača Firefox 2.0 RC3. Poznámky k tomuto buildu.

Ivan

17.10.2006

KOffice 1.6

Kancelářský balík KOffice 1.6 (oznámení na dot.kde.org) obsahuje především spoustu vylepšení v aplikacích Kexi, Krita a KPlato. Verze 1.6 je poslední (hlavní) verzí před 2.0, která už bude založena na Qt4.

Robert Krátký

17.10.2006

Buffer overflow v ovladačích od NVidie

V binárních ovladačích od NVidie se nachází vzdáleně (přes X server) zneužitelná chyba, viz rapid7.com. Vypadá to, že exploit na ní existuje. NVidia byla upozorněna již 7. července.

Michal Kašpar

17.10.2006

Perl 6 a Pugs 6.2.13

Perl 6 Wiki a Perl6 FAQ vám spolu s novou verzí Pugse (6.2.13, r14402) připomenou, že Perl 6 žije a je nutné s ním stále počítat. Pugs shell a provázání dokumentace s částí 18 tisíc testů [1], [2] a se stavem implementace [3], [4] jsou nejnovější vychytávky v tomto zajímavém projektu. A kdy bude beta šestky? No přece na vánoce. Všichni okolo to už vědí od července... 2001 :-).

Michal Jurosz

17.10.2006

Vývoj výkonu ovladačů ATI za poslední půlrok

Je tomu půl roku, co ATI vydala grafické ovladače s podporou karet řady Radeon X1000. Phoronix.com pozorně sledoval vývoj a nyní na grafech ukazuje, jak se výkon těchto ovladačů během uplynulých šest měsíců měnil.

Luboš Doležel

17.10.2006

Sedm kroků pro vyšší zabezpečení Linuxu

ZDNet Australia popisuje Linux jako systém, který je ve srovnání s WinNT a Novell Open Enterprise Server stabilnější, avšak jeho zabezpečování je prý komplikovanější. Proto uvádí sedm základních kroků, které by měl administrátor dodržovat pro tu nejvyšší bezpečnost.

Luboš Doležel

17.10.2006

Recenze několika zvukových editorů

Linux Journal představuje několik nejdůležitějších zvukových editorů na Linuxu: Ardour, Common Music a Csound. Přirovnává je k programům známým z Windows, jako je např. Cubase nebo SoundForge.

Luboš Doležel

17.10.2006

Ext4 byl začleněn do -mm jádra

Souborový systém ext4 se blíží do stabilního jádra. LinuxWatch shrnuje nové možnosti tohoto souborového systému, který je s ext3 kompatibilní podobným způsobem, jakým je ext2 kompatibilní s ext3 – připojení pod nižší verzí funguje za cenu ztráty nových funkcí. Ext4 můžete už nyní vyzkoušet v jádře 2.6.19rc1-mm1.

Luboš Doležel

17.10.2006

Vydávání nových verzí je otrava

Máte rádi psaní softwaru, baví vás hledání chyb a čtení bugreportů? Johna Goerzena taky, ale nebaví ho vydávání nových verzí. V jeho blogu najdete několik kroků, které musí stále opakovat. Jako ideální řešení by viděl napsání programu, který by je dělal místo vývojáře.

Jiří „Geo“ Lužnický

17.10.2006

Google o svém instalačním systému Slack

Zástupce Google na setkání australské Linux User Group prozradil, jakým způsobem spravují své velké množství serverů. Nepoužívají žádný z klasických balíčkovacích systémů, ale svůj vlastní nazvaný Slack. Slack byl uvolněn jako open source už před rokem, ale tentokrát to bylo poprvé, kdy o

něm někdo z Google veřejně mluvil. V článku na Computerworld.com.au je popsáno, jak Slack funguje.

Luboš Doležel 17.10.2006

Vydání Fedora Core 6 znovu opožděno

Vydání distribuce Fedora Core 6 se znovu opozdí, tentokrát o několik dnů. Příčinou jsou závažné neopravené chyby, které byly během víkendu odhaleny. Jedná se o problémy s multi-lib knihovnamí, balíčkem fedora-release a chybnými soubory překladů.

Luboš Doležel 17.10.2006

Linuxová VoIP ústředna na Mac Mini

Německá společnost 4S Newcom začne brzo nabízet IP po-
bočkovou telefonní ústřednu, která je založena na Mac Mini s iPod Shuffle jako bootovacím zařízením pro Linux. Více o tomto nezvyklém projektu píše LinuxDevices.com.

Luboš Doležel 18.10.2006

LSB Developer Network – obdoba MSDN

Free Standards Group vytváří projekt LSB Developer Network, který má představovat obdobu MSDN. Tím by měly odpadnout stížnosti na roztříštěnost dokumentace, která je zapotřebí na tvorbu přenosných aplikací na Linuxu.

Luboš Doležel 18.10.2006

Příkazy pro začínající správce serverů

ReallyLinux přináší výčet těch nejdůležitějších příkazů na Linuxu, které se hodí při správě Linuxu na serveru. V tabulce je vedle každého příkazu i jeho detailní popis, takže tuto stránku využijí především začátečníci.

Luboš Doležel 18.10.2006

SGL už není pod ochranou proti věřitelům

Společnost SGL v květnu tohoto roku požádala o ochranu před věřiteli. SGL se podařilo zbavit svého dlouholetého dluhu, propustilo 12 % zaměstnanců a získalo 115 milionů dolarů na zlepšení stavu firmy. Proto bylo toto opatření odvoláno.

Luboš Doležel 18.10.2006

MySQL AB přichází s MySQL Enterprise

MySQL AB oznámilo novou komerční podporu ve formě předplatného, kterou nazvali MySQL Enterprise. Součástí této nabídky je mj. Database Administrator (DBA), který správce automaticky upozorní na vznikající problém a navrhne jak jej zmírnit. Více o DBA a MySQL Enterprise naleznete na internetnews.com.

Luboš Doležel 18.10.2006

Recenze distribuce Slackware 11

Joe Brockmeier otestoval nedávno vydaný Slackware 11 ve VMware i na skutečném hardware a o svých zkušenostech napsal na Linux.com. V článku si stěžuje na drobné chyby a zатуhnutí při pokusu o instalaci na svůj notebook, nicméně oceňuje zahrnutí podpory některých problematických formátů (MP3 a QuickTime).

Luboš Doležel 18.10.2006

Vydání FC6 opět posunuto

Už to začíná být trapné psát, ale vydání Fedora Core 6 bylo opět posunuto. Z 19. (zítra) na 24. 10. Mimo jiných důvodů

(bugů) je tu stále problém s Anacondou, spouštěním X, poškozenými závislostmi na x86_64 a ppc(64) atd. Doufejme, že je to poslední odklad. Původní datum vydání mělo být 6. 10.

Milan Bažant 18.10.2006

Jak psát správně v OO Writeru

Na to, aby člověk napsal pěkně vypadající text, nemusí být TeXista. Stačí dodržovat pár základních postupů a pak to lze i zvládnout i s Writerem z OpenOffice.org.

Jiří „Geo“ Lužnický 18.10.2006

NetBeans Java Profiler, flash tutoriál

Říká se, že Java Profiler v IDE NetBeans je více než dobrý. Pokud nevíte, k čemu vlastně slouží, nebo vás zajímá ukázka toho, jak se s ním pracuje, tak můžete shlédnout tuto flashovou ukázkou.

Jiří „Geo“ Lužnický 18.10.2006

Gaim 2.0.0beta4

Vývojáři IM klienta Gaim právě uvolnili další betaverzi v poradí – Gaim 2.0.0 beta4. Stáhnout si ju můžete z serverov sourceforge, no na oficiálně oznámenie si budeme ešte musiet počkat.

Andrej Krivulčík 18.10.2006

OpenOffice 2.0.4 – sk v ArchLinuxu

18.10.2006 nám vyšiel v Archlinuxe vo verzii 2.0.4 OpenOffice.org. Od dnes máte k dispozícii aj slovensky interface k tejto verzii. CS(Z) interface je zatiaľ len k verzii 2.0.3, ale neinštalujte do 2.0.4, je to nekompatibilné.

Ivan 19.10.2006

Adobe Flash Player 9 Beta

Tak už i na Linuxu můžeme okusit dobrodružství Adobe Flash Playeru 9 (release notes). Zatím sice beta, ale funguje mi výtečně. Více informací. A máme i offline přehrávatko.

Petr Vaněk 19.10.2006

Práce se soubory v OpenOffice.org Basic

Další díl seriálu o skriptovacím jazyce OpenOffice.org Basic vyšel na Linux.com. Tentokrát se naučíte zapisovat data do souborů.

Luboš Doležel 19.10.2006

GNOME 2.17.1

Vyšlo GNOME 2.17.1. V této verzi najdete nejen opravy, ale i nové funkce, překlady a dokumentaci. Jedná se o první vydání na cestě k verzi 2.18, která by měla být hotová v březnu 2007.

Luboš Doležel 19.10.2006

Páteční rozhovor s vedoucím Linux Game Publishing

Michael Simms, který stojí za známou firmou Linux Game Publishing a e-shopem Tux Games, bude hostem dalšího podcastu od LinuxGames.com. Do pátečního večera máte šanci pokládat své dotazy.

Luboš Doležel 19.10.2006

NVidia opravuje bezpečnostní chybu ovladačů

NVidia vydala prohlášení ohledně vzdáleně zneužitelné chyby ve svých grafických ovladačích. Současně s tím byly uvolněny stabilní verze ovladačů 1.0-8776, které tuto chybu opravují.

Luboš Doležel 19.10.2006

Oslava k vydání finální verze Firefoxu 2.0

Blíží se vydání finální verze prohlížeče Mozilla Firefox 2.0. Pokud patříte mezi příznivce tohoto produktu, pak byste to měli oslavit. Doporučené datum pro uspořádání party je 27. – 29. října, tedy příští víkend.

Jiří „Geo“ Lužnický 19.10.2006

Čím tvořit screenshoty

Při rozšiřování sekce Software na AbcLinuxu možná budete chtít přidat nějaký pěkný screenshot. Čím ho vytvořit se dozvíte v článku na serveru Linux.com, který popisuje několik programů k tomu určených.

Jiří „Geo“ Lužnický 19.10.2006

RC verze Ubuntu Edgy Eft, start vývoje Feisty Fawn

Vývojáři Ubuntu usoudili, že stav Edgy Eft je vhodný pro testování koncovými uživateli, a tak vydali RC verzi, která má za úkol týden před vydáním finální verze opravit poslední chyby. Současně Mark Shuttleworth startuje vývoj verze 7.04 (Feisty Fawn – divoká srna).

Jiří „Geo“ Lužnický 19.10.2006

Jabbin – fork Psi s podporou VoIP

Linux.com představuje program Jabbin. Jabbin je fork Jabber klienta Psi, který se však více zaměřuje na podporu VoIP. Autor článku vyzkoušel program na několika serverech a narazil na pády během pokusu o hovor s uživateli Google Talk. Základní funkce klienta však byly bez pádů.

Luboš Doležel 19.10.2006

Učebnice ABC/Linuxu

Po několika měsících a mnoha aktualizacích vychází aktualizovaná pdf verze online knihy Učebnice GNU/Linuxu s novým názvem Učebnice ABC/Linuxu. Stahovat můžete v sekci GNU/Linux CZ dokumentace na aboutme.ic.cz nebo přímo z Abička (PDF).

Pavel Vlasák 20.10.2006

4L – LightScribe zapisovač

Po dlouhém čekání využijí majitelé vypalovaček s technologií LightScribe tuto funkci i na Linuxu. Společnost LaCie, která je významným podporovatelem projektu K3b, uvolnila program LaCie LightScribe Labeler for Linux (4L). Bohužel se nejedná o open source software, ale jen o freeware.

Luboš Doležel 20.10.2006

Songbird 0.2

Multiplatformní kombinovaný hudební přehrávač a webový prohlížeč Songbird dospěl do verze 0.2. Rozhodně se nejedná o stabilní verzi, tato je určena jen pro vývojáře a testery. Pokud tento software, který je založen na Gecko a GStreameru, ještě neznáte, podívejte se na obrázky.

Luboš Doležel 20.10.2006

NVidia NVPerfKit 2.1

NVidia vydala nástroj NVPerfKit 2.1. NVPerfKit slouží k analýze a ladění výkonu OpenGL programů a podle NVidie vám tak pomůže zvýšit výkon vašeho software až o 35 %.

Luboš Doležel 20.10.2006

Písma v TeXu pro začátečníky

Uživatelé TeXu jistě velmi dobře znají písmo Computer Modern Roman. Začátečnickům však často zůstávají skryty další fonty v jejich systému. Proto na Linux.com vyšel článek, který se tématem testování písem v TeXu zabývá.

Luboš Doležel 20.10.2006

Znáte zvíře od každého písmenka abecedy?

Mnoho lidí si láme hlavu nad systémem pojmenování jednotlivých verzí Ubuntu. Ačkoliv se na první pohled současný model může zdát absolutně náhodný, není tomu tak. První písmena názvů jdou podle abecedy, jedná se o dvě slova začínající na stejné písmeno, a druhé slovo je zvíře. Znáte však od každého písmenka abecedy alespoň jedno zvíře?

Jiří „Geo“ Lužnický 20.10.2006

Open source nástroje pro práci s MySQL db

Program phpMyAdmin zná snad každý, kdo kdy něco kutil s MySQL databází. Byla by však chyba považovat ho za jediný open-source nástroj v této oblasti. Představení několika zajímavých programů nejen pro monitorování stavu databáze přináší server OnLamp.

Jiří „Geo“ Lužnický 20.10.2006

Ovladač pro čtečku otisků prstů na notebooku HP

Pokud máte notebook HP Compaq nx6125 nebo podobný (nx6325, nc6320, atd.), jistě vás bude zajímat, že už se začal vytvářet ovladač pro čtečku otisků prstů v těchto notebookech.

Martin 21.10.2006

Beryl 0.1.1

Vyšla nová verze 0.1.1 OpenGL kompozitního správce oken Beryl. Bylo opraveno velké množství chyb a přidány některé nové efekty a funkce. Zároveň byla do hlavního SVN repozitáře začleněna větev „flame“, takže nyní si všichni milovníci eye-candy mohou užívat krásného efektu shození okna při jeho uzavření. Byl také spuštěn nový design webových stránek a webový theme repozitář.

Michal Křenek 22.10.2006

MPlayer 1.0rc1

Vyšel první Release Candidate skvělého přehrávače MPlayer verze 1.0. Novinek je opět velká spousta. Určitě stojí za stáhnutí nebo aktualizaci.

Alois Nešpor 23.10.2006

Firefox 2.0 je tady!

Na ftp.mozilla.org je k dispozici verze 2.0 známého prohlížeče Mozilla Firefox. Využijte příležitosti a stahujte a instalujte ještě před oficiálním vydáním.

Ladislav Jancik 23.10.2006

Používáme program gpart

Gpart je nástroj, který se vám může hodit v nouzi – dokáže na disku hledat různé souborové systémy při ztrátě tabulky oddílů. Návod na serveru Linux.com vás naučí s tímto programem pracovat.

Luboš Doležel

23.10.2006

Jak na Xen 3

Zaujal vás virtualizační nástroj Xen 3 (například jste jej viděli na sobotním Bootcampu), ale nevíte jak ho nainstalovat? Pokud používáte Debian (nejlépe Sarge), pak by to pro vás pomoci velmi detailního návodu neměl být problém. Popsána je jak instalace hostitelského stroje, tak jednotlivých virtuálních strojů.

Jiří „Geo“ Lužnický

23.10.2006

Plánování projektů s GnoTime

GnoTime (GNOME Time Tracker) je poměrně jednoduchý program, který vám usnadní časové plánování projektů a úkolů. Joe Barr ve svém článku na Linux.com popsal své zkušenosti s používáním tohoto software.

Luboš Doležel

23.10.2006

Jak na tunely skrz internet

Pokud jsou pro vás vzdálená Xka a tunelování portů, z například firemní sítě skrze internet černá magie, pak si přečtěte článek na IBM Developerworks. Poradí vám s nastavením jak klientské (a to i pokud jste na klientovi nuceni používat Windows), tak serverové části.

Jiří „Geo“ Lužnický

23.10.2006

Phoronix testoval ovladač karet Intel 965 Express

Phoronix.com pokračuje ve svém maratónu testů grafických ovladačů. Tentokrát otestovali grafický ovladač pro čipsety Intel 965 Express. Výkon není příliš dobrý, ale hlavní kvalitou tohoto čipu má být nízká spotřeba, dobrá cena a stabilita.

Luboš Doležel

23.10.2006

Java SE 6 (Mustang) se blíží

Vydání Javy SE 6, která je známa pod označením „Mustang“, se blíží. Po několika betaverzích se dá očekávat uvolnění finální verze už v nejbližších hodinách či dnech. Informit.com vám ukáže novinky této verze, mezi něž patří lepší integrace s prostředím, přístup k parametrům síťových rozhraní a mnoho dalších zajímavých funkcí.

Luboš Doležel

23.10.2006

Fedora Core 6 je tady!

Po mnohých odkladech je Fedora Core 6 konečně připravena. Její finální verze se na nás těší již 24.10.06 v 14:00 GMT. Zároveň s tím byly aktualizovány i screenshoty a videa ve FedoraTour a samozřejmě o tom vyšel i nový Fedora Weekly News Issue č. 63. Zdá se, že finální Fedoru 6 již nic nezastaví!

Milan Bažant

23.10.2006

Recenze Fedora Core 6 na Phoronix

Phoronix přináší první pohled na finální verzi distribuce Fedora Core 6, která vyšla po několika odkladech 23. 10.

Robert Krátký

24.10.2006

Java jako open source v roce 2007

Sun Microsystems plánuje uvolnění velkých částí Javy jako open source do konce tohoto roku, plně open source projektem se stane v roce 2007. O detailech této události píše News.com.

Luboš Doležel

24.10.2006

Red Hat dokončil přechod města Chicago na Linux

Ukázkou toho, že ani v USA se neváhá s nasazováním open source, je město Chicago. Red Hat oznámil úspěšné dokončení přechodu tamních serverů na Linux. Tyto servery obsluhují požadavky, jako je zřizování řídičských průkazů, online žádosti o práci nebo kontroly restaurací. Bližší informace přináší sys-con.com.

Luboš Doležel

24.10.2006

V Pákistánu vznikla nadace za podporu FLOSS

India eNews píše o sousedním Pákistánu, který je často kritizován za vysokou míru softwarového pirátství. Článek se zaměřuje na nadaci Free and Open Source Software Foundation of Pakistan (FOSS-FP) fungující od roku 2005. Cílem FOSS-FP je informovat veřejnost o alternativách pirátského software a posílit zkušenosti studentů s Linuxem.

Luboš Doležel

24.10.2006

Snadnější sledování logů z více strojů

Sledování logů patří mezi povinnosti každého dobrého správce. Při větším počtu spravovaných serverů to však může být poněkud otravné, ale je možné si to usnadnit tím, že umožníme sledování všech logů z jednoho místa.

Jiří „Geo“ Lužnický

24.10.2006

První pohled na GAIM 2.0

Minulý týden byla vydána čtvrtá betaverze IM komunikátoru GAIM. Článek od Linux.com popisuje nové funkce a změny v tomto programu, nebylo zapomenuto ani na nové textové rozhraní. Autor článku nebyl spokojen se snahou o zjednodušení přihlašovacího dialogu, ani s neexistencí podpory Jingle.

Luboš Doležel

24.10.2006

Rychlost Qt 4 oproti Cairo

Zack Rusin (vývojář Qt a KDE) ve svém blogu zveřejnil benchmark rychlosti grafického engine Qt 4 oproti Cairo. Grafický engine Qt 4 je 5 – 7 rychlejší než Cairo. Při použití nativního OpenGL backendu (místo XRender) jsou rychlostní rozdíly ještě mnohonásobně větší. Využití OpenGL k renderování 2D grafiky se v Qt 4 tedy opravdu vyplatí, Cairo je oproti tomu při použití OpenGL backendu (Glitz) téměř stejně pomalé jako Cairo využívající XRender.

Michal Křenek

24.10.2006

Firefox 2 oficiálně vydán

Mozilla konečně oficiálně vydala druhou verzi prohlížeče Firefox, který se dal již dříve stáhnout na FTP serverech. S novou verzí přišel i redesign stránek mozilla.com. Tisková zpráva czilla.cz.

stribny

24.10.2006

OS engine Hry Age of Kings

Příznivce Open Source a hry Age of Empires II: Age of Kings jistě potěší zpráva, že se již nějakou dobu vyvíjí OS engine této hry. Jeho jméno je Prophet (podle jména původního engine, který se jmenuje Genie) a je postaven na OpenGL a SDL. Jeho portace na Linux (ač není primárním cílem) by tedy neměla být problematická.

Vít Pelčák

25.10.2006

Byl dokončen překlad ubuntuguide.org

Překlad neoficiálního průvodce Ubuntu Dapper Drake do češtiny byl dokončen. Česká verze byla rozšířena o podrobnější návod na instalaci proprietárních ovladačů nVidia a Ati.

Vít Pelčák

25.10.2006

FC6: 10 000 stažení během 5 hodin

Blog na CNET News informuje, že Fedora Core 6 dosáhla během pěti hodin 10 000 stažení. To odpovídá jednomu stažení každých 1,8 sekundy.

Luboš Doležel

25.10.2006

SGL se soudí s ATI kvůli porušení patentu

SGL se pomalu dostává z problémů a hned toho využívá. SGL se soudí s ATI kvůli porušení patentu na „zobrazovací systém, který používá rasterizaci a framebuffer pracující v číslech s plovoucí desetinnou čárkou“.

Luboš Doležel

25.10.2006

Spouštění externích příkazů ve VIMu

Mezi funkce editoru VIM patří spouštění externích příkazů, což má zajímavé možnosti použití. Článek od Linux.com vás naučí využít tuto funkci ke zpracování textu: vložení výstupu programu do editoru nebo předání aktuálního dokumentu externímu programu.

Luboš Doležel

25.10.2006

Tux Paint 0.9.16

Po téměř roční prodlevě je venku nová verze oblíbeného kreslicího programu pro děti Tux Paint 0.9.16. Obsahuje novou funkci prezentace a má nové typy štětců.

Luboš Doležel

25.10.2006

Setkání uživatelů *Ubuntu

Jelikož zítra bude vydán Edgy Eft, rozhodli jsme se při této příležitosti udělat setkání uživatelů *Ubuntu na jabber konferenci (server: chat.linjab.net ; místnost: ubuntu). Návod na připojení: Konferenční místnost na Jabberu. Oficiální začátek 26.10.2006 od 18:00, a...speciálně pro závisláky ;-) taky dnes (25.10.) od půlnoci...

kAja Ziegler

25.10.2006

Dokončení 14 projektů zlepšujících FreeBSD

Na stránkách FreeBSD se objevilo oznámení o úspěšném dokončení všech čtrnácti projektů, které vznikly v rámci letošního Summer Of Code. Jedná se například o opravy IPv6 stacku, implementaci podpory AutoFS, nebo umožnění instalace FreeBSD při bootu ze sítě.

Jiří „Geo“ Lužnický

25.10.2006

Shuttleworth: Free software musí být krásný

Pokud chceme, aby svět začal brát F/OSS vážně, musíme ho dělat krásný. Nejde o vnitřní krásu, eleganci nebo ideologickou čistotu ? prostě vizuálně atraktivní software. Tak nějak by se dal shrnout obsah aktuálního zápisu v blogu Marka Shuttlewortha: „Pretty“ is a feature.

Daniel Kvasnička ml.

25.10.2006

Xen 3.0.3

Xen 3.0.3 je tu. Nová verze má nový plánovač využití CPU a lepší podporu běhu nemodifikovaných OS. Nově obsahuje podporu virtuálních disků s dynamickou alokací místa a copy-on-write. Další novinky a informace naleznete na OSDDir.com.

Luboš Doležel

25.10.2006

Multiplatformní editor videa Jahshaka 2.0

Je to sice už pár týdnů, co vyšla nová stabilní verze video editoru Jahshaka, ale pokud patříte mezi příznivce zpracování videa v Linuxu či jiném operačním systému, určitě byste měli zkusit možnosti tohoto open source programu.

Martin

25.10.2006

Beryl: začleněn efekt BeamUp

Do hlavního SVN repozitáře programu Beryl byla začleněna větev efektu „BeamUp“. Že nevíte, o co se jedná? Fanoušci seriálu StarTrek jistě otevřou láhev nejlepšího vína (zoufalci jako já otevřou krabičák). Jedná se o efekt teleportování z výše uvedeného seriálu.

Zelená potvora

26.10.2006

Vydáno Ubuntu 6.10

Dnes bylo vydáno nové Ubuntu, Kubuntu i Edubuntu 6.10 s označením „Edgy Eft“. Obsahuje moderní vzhled, aktuální verze aplikací a s nimi nové, často experimentální vlastnosti. Vydání bude podporováno po dobu 18 měsíců. Oficiální oznámení (v angličtině), překlad.

Martin „mhb“ Böhm

26.10.2006

O novinkách v Samba 4.0

Příští verze Samby 4.0 přinese spoustu nových funkcí a opět tím sníží potřebu komerčních řešení. Bude mít lepší podporu Active Directory, Kerberosu a nový Virtual File System. Více se dozvíte v článku na SearchOpenSource.com.

Luboš Doležel

26.10.2006

Oracle adoptuje Red Hat Linux

Oracle oznámil, že bude pro Unbreakable Linux dávat stejnou podporu, jako dává pro databáze a ostatní software. Unbreakable Linux je založený na Red Hat Linuxu, ale ceny za podporu od Oracle by měly být nižší než v případě RH.

Luboš Doležel

26.10.2006

NFSv4 v Linuxu považováno za stabilní

Podpora NFSv4 v Linuxu, která byla zařazena v roce 2005, prošla důkladným testováním a měla by být stabilní. Spolehlivost dokládá dostupnost NFSv4 v enterprise distribucích Red Hat Enterprise Linux a SUSE Linux Enterprise. O události informuje TechnologyNewsDaily.com.

Luboš Doležel

26.10.2006

Alan Cox o bezpečnosti open source software

Alan Cox při konferenci na LinuxWorld upozornil na to, že otevřenost zdrojového kódu nemusí automaticky znamenat bezpečnost software. Řekl, že mnoho open source projektů má k bezpečnosti daleko. Další úryvky z jeho projevu naleznete na ZDNet.co.uk.

Luboš Doležal 26.10.2006

V Británii vzniklo národní open source středisko

V britském Birminghamu vzniklo národní open source středisko. Středisko bude hostitelem konferencí a seminářů a bude zdrojem studií o využívání open source ve Velké Británii. Pokud chcete vědět více, čtete článek od ZDNet UK.

Luboš Doležal 26.10.2006

Novell oznamuje PC s předinstalovaným Linuxem

Novell se dohodl se čtyřmi výrobci na distribuci počítačů s předinstalovaným Linuxem. Jedná se o výrobce ETegro Technologies, MAXDATA, Transtec a R Cubed Technologies. Každý z nich bude prodávat desktopy a notebooky s předinstalovaným SUSE Linux Enterprise Desktop 10.

Luboš Doležal 26.10.2006

Přechod na Linux v Mnichově v plném proudu

Heise online informuje o průběhu přechodu na Linux v Mnichově. Uživatelé nemají s novým systémem žádné zásadní problémy, snad s výjimkou toho, že si několik dnů museli na nové prostředí zvykat. Pokud bude vše probíhat podle plánu, na většinu z 14 000 PC bude do dvou let běžet Linux.

Luboš Doležal 26.10.2006

Veřejná betaverze hry Serious Sam 2

Vývoj hry Serious Sam 2 pro Linux pokročil a společnost Croteam se rozhodla vydat veřejnou betaverzi hry. Je vyžadována grafická karta s Pixel Shaders 2.0. Problémy se hrou, která se zatím chová značně nestabilně a trpí nízkým výkonem, oznamujte ve fóru.

Luboš Doležal 26.10.2006

Ubuntu 6.10 obsahuje oficiální Firefox 2

Ubuntu 6.10 je první linuxová distribuce, která obsahuje oficiální plnou a podporovanou verzi Firefoxu 2. Ubuntu se tak odlišuje od Debianu, který obsahuje významné úpravy Firefoxu, a musí proto používat jiné označení – IceWeasel. Totéž platí pro Thunderbird/IceDove.

Petr Tomeš 26.10.2006

Zotero – zajímavé rozšíření pro Firefox

Zotero je rozšíření pro Mozillu Firefox, které se bude hodit lidem, kteří tráví mnoho času hledáním zajímavostí na webu. Obsahuje důmyslný systém záložek s poznámkami uživatele a umí se integrovat s weby některých amerických knihoven. Detailnější rozbor funkcí naleznete na Linux.com.

Luboš Doležal 27.10.2006

Recenze Ubuntu Edgy

Joe Brockmeier otestoval Ubuntu Edgy a pro Linux.com napsal článek se svými zkušenostmi. Zmiňuje se o praktickém nástroji Apport, který by měl v případě pádu aplikace umož-

nit odeslat chybové hlášení. Nelíbí se mu, že distribuce není tak „edgy“, jak si představoval – chybí mu některé nejnovější aplikace.

Luboš Doležal 27.10.2006

Úvod do Apache Derby, MyFaces a Facelets

Článek od IBM developerWorks uvádí čtenáře do vývoje webových aplikací v Javě pomocí Apache Derby, MyFaces a Facelets. Nejprve popisuje k čemu které z nich slouží, aby následně předvedl, jak se s nimi dá vyvinout software pro online rezervaci letenek.

Luboš Doležal 27.10.2006

Záznam povídání o iptables

Na stránkách Audiovizuálního Centra SH se objevila další z přednášek SUTu, tentokrát o tom, jak nastavit firewall pomocí iptables.

Jiří „Geo“ Lužnický 27.10.2006

Upstart je ve stavu, kdy může nahradit sysvinit

Autor programu Upstart, náhrady sysvinitu, ho označil za připravený k používání. Technické detaily a informace o tom, jak s Upstartem spolupracovat, naleznete v autorově blogu.

Jiří „Geo“ Lužnický 27.10.2006

Frozen Bubble 2 je na světě

Právě vyšla druhá, notně vylepšená verze jedné z nejoblíbenějších linuxových her? Frozen Bubble. Nabízí mnoho nových možností, například hru na lokální síti nebo přes Internet či trénink multiplayeru. Může se pochlubit remixovanou hudbou, vylepšenou grafikou a překladem do mnoha jazyků včetně češtiny. Ke stažení jsou jak zdrojové kódy, tak i několik binárních balíčků.

Vítězslav Šmíd 27.10.2006

Vyšlo Wine 0.9.24

Vyšlo Wine 0.9.24. Mezi novinky patří podpora pro více monitorů (přes Xineramu), oprava spousty memory leaků a dalších různých chyb.

David Watzke 27.10.2006

Sdílení skeneru přes síť

Jednou ze zajímavých možností SANE (Scanner Access Now Easy) je možnost sdílení vašeho skeneru přes síť. Pokud vás tato funkce zaujala, čtete jednoduchý návod od Linux.com na zprovoznění této služby.

Luboš Doležal 27.10.2006

Jak umí MythTV přeskokovat TV reklamy

MythTV obsahuje jednu zajímavou funkci – dokáže automaticky detekovat reklamní spoty ve vysílání a přeskočit je. Na pvrwire.com se dozvíte jak detekce funguje a uvidíte ji v akci.

Luboš Doležal 29.10.2006

Brasero 0.5.0 je na světě

Ve středu 25. října vyšla nová stabilní verze vypalovacího programu pro GNOME Brasero (dříve bonfire) – 0.5.0. Oproti verzím 0.4.x přibyl například test integrity dat, podpora libburn backendu nebo možnost přidávat pauzy individuálně mezi jednotlivé stopy audio CD. Stahovat můžete z sourceforge.net/projects/bonfire.

Martin Sourada

29.10.2006

NetBeans 5.5 je venku

Okolo půlnoci byla vydána finální verze programu NetBeans 5.5. Stahovat můžete přímo ze stránek netbeans.org, kterým se dostalo nového vzhledu.

Jakub

30.10.2006

DVD verze Ubuntu Edgy Eft

Minulý týden vyšla nová verze Ubuntu Edgy Eft, nicméně v nabídce byla ke stažení pouze CD verze. Pokud vám více vyhovuje verze DVD, která byla vydána zcela v tichosti, stahujte z jednoho z mirrorů: ČR, USA, Finsko, Nizozemí, Nizozemí, Nizozemí, UK, Švédsko.

Kovář David

30.10.2006

Mark Shuttleworth o Firefox vs. IceWeasel

Mark Shuttleworth se na svých stránkách snaží horkým hlavám vysvětlit postoj Mozilly i Debianu v případě Firefox vs. IceWeasel.

Robert Krátký

30.10.2006

Proč Opera nebude open-source?

Článek na novozélandském Computerworldu vysvětluje, proč se Opeře nechce do světa open source. V zásadě jde o to, že téměř nikdo nemá u webových prohlížečů zájem o placenou podporu, která však v open-source business modelech hraje velkou roli.

Daniel Kvasnička ml.

30.10.2006

Databáze kontaktů bez GUI

Článek ze seriálu CLI Magic na Linux.com pojednává o správě kontaktů z konzole. Naučíte se udržovat databázi kontaktů bez použití GUI nebo příliš komplikovaného software, jakým je např. LDAP.

Luboš Doležel

30.10.2006

Konfigurace služeb na Ubuntu 6.10 Server

Ubuntu 6.10 Server je distribuce, na které můžete provozovat server s Apache, Bind nebo podobnými známými službami. HowtoForge vás provede konfigurací takového víceúčelného serveru, a to jen v sedmi krátkých stranách plných ukázek.

Luboš Doležel

30.10.2006

HP: open source může být zdrojem zisku

Mnoho lidí si pod open source představí software zdarma nebo vývojáře bez jakéhokoliv zisku. Bdale Garbee z Hewlett-Packard je přesvědčen, že výdělky z poskytování open source řešení mohou být vyšší než v případě proprietárních řešení. Více píše australský ZDNet.

Luboš Doležel

30.10.2006

OpenSUSE 10.2 Beta 1

OpenSUSE dospělo k verzi 10.2 Beta 1 (oznámení). Distribuce je poměrně hodně populární, a tak i na betaverzi vyšla recenze, ve které se dozvíte jaké problémy můžete očekávat při betatestování.

Luboš Doležel

30.10.2006

Amarok 1.4.4

Nový Amarok 1.4.4 přináší novinky v podpoře synchronizace s iPodem, nové režimy prolínání skladeb a spoustu dalších změn a oprav. Všimněte si také integrace s hudebním obchodem Magnatune.com. Stahujte balíčky pro své distribuce.

Luboš Doležel

30.10.2006

Export dat z OpenOffice.org Base

Base je jméno méně propracované části OpenOffice.org, která se zaměřuje na databáze. SearchOpenSource.com vás provede exportem dat z tohoto software, který bohužel není tak snadný, jak by mohl být.

Luboš Doležel

31.10.2006

Protest FSF proti směrnicím Brazílské vlády

Free Software Foundation – Latin America (FSFLA) protestuje proti předpisům brazilské vlády, které nutí některé občany k používání nesvobodného software při platbě daní. Více informací o problému a kampani naleznete na Linux.com.

Luboš Doležel

31.10.2006

Recenze Krita 1.6

Krita je grafický editor z balíku KOffice, což z něj ovšem nedělá druhořadou aplikaci – pozvolna se stává alternativou populárního programu GIMP. Linux.com otestoval Kritu 1.6, za zmínku stojí i článek na AbcLinuxu.

Luboš Doležel

31.10.2006

K3b

Vývoj K3b se blíží k verzi 1.0. Dne 6.9. vyšla verze K3b 1.0 Preview 2. Takže se máme na co těšit.

Saljack

31.10.2006

Základní úkony v OpenOffice.org Calc

Článek v blogu na Linux Journal vás naučí některé základní úkony v tabulkovém procesoru OpenOffice.org Calc. Jedná se o filtrování, řazení, skrývání buněk a další funkce.

Luboš Doležel

31.10.2006

GRASS GIS 6.2.0

Právě byla vydána nová stabilní verze nejrozšířenějšího Open Source geografického informačního systému GRASS. Kromě oprav chyb bylo přidáno několik nových modulů a hlavně zcela přepracováno grafické uživatelské rozhraní.

Jáchym Čepický

31.10.2006

■