



## Abíčko

Časopis serveru AbcLinuxu.cz

Srpen 2005



---

Vychází také na CD-ROM jako příloha časopisu



# Editoriál

Vítejte u čtení časopisu Abíčko.

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

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

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

Náměty na články zasílejte do konference našich autorů: [autori@abclinuxu.cz](mailto:autori@abclinuxu.cz).

Sponzoring Abíčka a jiné formy reklamy si objednávejte na adrese: [reklama@abclinuxu.cz](mailto:reklama@abclinuxu.cz).

Ostatní dotazy směřujte na adresu: [literakl@abclinuxu.cz](mailto:literakl@abclinuxu.cz).

Server <http://www.abclinuxu.cz> provozuje firma Stickfish s.r.o., která poskytuje profesionální služby v oblasti Linuxu firmám i jednotlivcům. Zabývá se hlavně bezpečností, instalacemi Linuxu a konfigurační sítových služeb. Více na <http://www.stickfish.cz>.

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

Editor a sazba: Vlastimil Ott

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

<b>Editoriál</b>	<b>1</b>
<b>Obsah</b>	<b>2</b>
<b>Rozhovor: Miguel de Icaza</b>	<b>5</b>
<b>Audio v Linuxu</b>	<b>14</b>
Úvod .....	14
Software .....	14
Audacity .....	15
ReZound .....	15
Ardour .....	16
JACK .....	16
LASH .....	17
LADSPA .....	17
Sekvencery .....	17
Rosegarden .....	18
MusE .....	18
Pracovní prostředí .....	18
Základy práce .....	19
Ovládání .....	20
Test .....	21
Vestavěné syntetizéry .....	22
<b>PEAR</b>	<b>24</b>
Úvod – framework .....	24
PEAR .....	24
Balíčky .....	24
Konvence .....	25
Distribuční systém .....	25
Instalace .....	26
Instalace pomocí správce balíčků .....	26
Instalace z tarballů .....	26
Instalace v prostředí webhostingu .....	26
Závěr .....	27
Úvod – problematika .....	27
Benchmark .....	27
Benchmark_Timer .....	27
Benchmark_Iterate .....	28
Benchmark_Profiler .....	29
APD .....	29
ab, httpperf .....	31
Závěr .....	31
Autentizace .....	31
Co nabízí PEAR? .....	31
Auth .....	32
LiveUser .....	33
<b>Adresár /proc – brána do jadra</b>	<b>35</b>
Úvod .....	35

Charakteristika procfs na Linuxe .....	35
Štruktúra adresára /proc .....	35
Adresáre procesov .....	36
Súbory a adresáre kernelu .....	36
Adresár /proc/sys .....	36
Jednoduché príklady .....	37
História procfs .....	38
Záverom .....	38
<b>Sysfs – brána do jadra</b> .....	<b>39</b>
Úvod .....	39
Charakteristika Sysfs .....	39
Začiatky Sysfs .....	39
Štruktúra Sysfs .....	39
Sysfs ako nástroj .....	39
Jednoduché príklady .....	40
Administrácia .....	41
Záverom .....	41
<b>Adresářové služby</b> .....	<b>42</b>
Úvod .....	42
Adresář .....	42
Komponenty adresáře .....	42
Použití adresáře .....	42
LDAP .....	43
Adresáře v informačních systémech .....	46
Závěr .....	46
<b>Internetová rádia</b> .....	<b>47</b>
Není rádio jako rádio .....	47
Formáty .....	47
Jak rádio najít .....	47
Streamtuner .....	47
Jak rádio přehrát .....	48
Jak rádio nahrát .....	48
Streamripper .....	49
Jak rádio nahrávat a zároveň poslouchat .....	49
Co posloucháte .....	50
<b>Jaderné noviny 313–316</b> .....	<b>51</b>
Linux 2.6.12-rc4-mm2, 27 e-mailů .....	51
sdparm verze 0.92, 4 e-mailů .....	51
dmraid verze 1.0.0.rc8, 1 e-mail .....	52
Linux 2.4.31-rc1, 1 e-mail .....	52
Nová subarchitektura pod arch/cris, 2 e-mailů .....	52
Vývoj ALSA přechází na git, 16 e-mailů .....	52
Porušuje Panda GPL?, 6 e-mailů .....	53
GCC 4 neumí kompilovat jádro 2.4, 3 e-mailů .....	53
Aktualizace Průvodce vývojáře libata, 1 e-mail .....	54
Strom -mm bude k dispozici jako repozitář git, 20 e-mailů .....	54
Nové automatizované testovací skripty pro oficiální a vývojové verze jádra, 8 e-mailů .....	54
Někteří vývojáři chválí řadu w.x.y.z, 11 e-mailů .....	55
Nový skript pro výpis krátkého changelogu z gitu, 6 e-mailů .....	55
Migrace na 4K kernel stacks, 13 e-mailů .....	56

Shrnutí real-time problémů, 103 e-mailů .....	56
DevFS odstraněno z Linuxu, 42 e-mailů .....	57
Linux 2.4 nebude podporovat GCC 4, 2 e-mailů .....	58
Linuxová dokumentace, kterou lze prohlížet jen v nelinuxových prohlížečích, 10 e-mailů .....	58
<b>Zprávičky</b>	<b>59</b>

# Rozhovor: Miguel de Icaza

Filip Vrlík

---

*Otec Gnome, Mona a řady dalších projektů. Spoluzakladatel Helix Code, pozdější Ximian, dnes koupený Novellem. Jeden z nejvynalézavějších lidí 21.století podle TheTimes.*

---

Miguel de Icaza přijel do Prahy z Mexika a já [1] měl možnost s ním udělat rozhovor. Během snídaně jsem se dozvěděl spoustu šokujících odhalení a pohledů do zákulisí. No dobře, zas až tak strhující to není, ale pár zajímavých odpovědí ohledně „Mona“ vám přinášíme v prvním díle.

**\* Otázka:** Na jaké projekty se teď v poslední době zaměřuješ nejvíce? Mohl bys nám je prosím představit?

▷ **Odpověď:** Nyní nejvíce pracuji na projektu Mono, což je vývojářská platforma založená na .NET. Na této technologii je zajímavá její nezávislost na použitém jazyku. Mnoho lidí strávilo obrovské množství času nad tím, aby tato „virtual machine“ byla dost dobrá.

**\* Otázka:** Něco jako Java, ne?

▷ **Odpověď:** Je to podobné Javě, ale zde je kladen velký důraz na podporu více jazyků. Například platforma Java neumí pořádně efektivně zvládnout jednu věc, a tou je podpora C a C++. Motivací pro Mono je produktivita. V minulosti jsme vytvořili Evolution a OpenOffice.org pomocí C++. Spousta problémů, kterým programátoři čelí dodnes, vychází z toho, že C++ je nízko-úrovňový jazyk. Musíte si vytvořit vlastní správu paměti a tak dále. Také je spousta problémů s narušením paměti: uvolníte pointer dvakrát, použijete buffer, který používá někdo jiný nebo tak narušíte paměť, kterou používá někdo jiný. Hledali jsme, Ximian hledal, platformu, která by se dala použít pro tvorbu velkých aplikací, jako je Evolution, v krátkém časovém úseku nebo s méně lidmi. Tedy buď tvořit rychle, nebo vytvořit více s větším týmem. Místo abychom teď Evolution vyvíjeli dva roky, zvládli bychom to za rok.

Podívali jsme se na Javu a C# a zdálo se, že C# zvládá to samé jako Java plus ještě něco navíc. Vždy když něco děláte, ať už je to cokoliv, tak je důležitý čas. Takže jsme to udělali takhle. Jak tedy vidíte, Mono je technologie umožňující běh různých programů pro desktop. Já se nejvíce zajímám o Linux jako o desktop, v podstatě vyvíjím pouze pro desktop, ale Mono pokrývá všechny oblasti. Já stále pracuji se skupinou desktop. Teď jsme měli obrovskou konferenci v Německu, odtud jsem teď přijel. Oni nepiší přímo pro desktop, ale já jim poskytuji nástroje. Postavili jsme GTK#, což je toolkit pro tvorbu aplikací pro Linux. Tvoříme designérský nástroj. Poskytujeme lidem kvalitní vývojářské nástroje a dokumentaci. Totiž, spousta lidí si stěžuje, že otevřený software má příšernou dokumentaci. My děláme dokumentaci pro Mono, snažíme se dokumentovat všechno, takže lidé mají dobré dokumenty. Snažíme se také zaplňovat mezery v Linuxu, tvořit věci, které Linuxu chybí, aby byl úspěšným desktopem.

*(Miguelovi zvoní alarm na mobilu.)* Tak teď jsem měl vstávat. :-)  
*(Miguel se nás ptá na abclinuxu.cz a na situaci okolo Linuxu v ČR.)* Jedni z nejlepších programátorů, s jakými jsem měl možnost spolupracovat, byli z České republiky. Dělal jsem s více takovými. Já jsem z Mexika a... nebo jinak. Když se na situaci podívám z vnějšku, tak Češi jsou na open source scéně velmi aktivní, procentuálně dvakrát více přispívají a jsou velmi dobří. Možná je to díky systému vzdělání a školství.

**\* Otázka:** Díky školství, eh, to si nemyslím ;-). Ale každopádně moc díky. Tedy za ně samozřejmě.:-) Nějaké novinky okolo Mona, nějaké konkrétní úspěchy v poslední době?

▷ **Odpověď:** Vypustili jsme Mono zhruba před rokem, asi před deseti měsíci. Dostává se do chvíle, kdy dospívá v solidní kód. Mysleli jsme, že 1.0 nebude použitelná, ale byla. Nebyla ale tak vybroušená jako je dnes. Vylepšujeme celkem dost, všechny detaily v Monu. Z používání

máte dobrý pocit. To znamená, že je výkonnější. Pro výkonnost jsme toho v poslední době udělali dost. Práce, kterou jsme udělali, se bohužel neobjeví ve verzi 1.2, bude zahrnuta až v pozdější verzi. Jedná se totiž o opravdu velké změny v architektuře. No, nejde ani tak přímo o změny ale spíše o optimalizace; některé jsou velmi těžké a složité. Nebudeme tedy s to stihnout zářijový termín, jak jsme chtěli.



Lepší výkon, stabilita, to jsou přednosti, které si ale lidé neuvědomují. Poučili jsme se z dlouho běžících serverů a aplikací pro desktop jako je Beagle. Ty jsou stabilnější, než si lidé myslí. Máme tedy tři přednosti, ale lidi moc nezajímá, že je to rychlejší či stabilnější, na tom jim nezáleží. Ale přicházíme s novou GTK#, která podporuje ještě více věcí v GTK. Máme nový toolkit, jež umí wrapovat jakoukoliv GObject knihovnu napsanou v C. Můžeme ji potom přímo použít z C#. Vyvinuli jsme propojení (bindings) do Evolution, takže jej teď můžete naskriptovat pomocí C#. Řekl jsem C#, ale může to být jakýkoliv jazyk podporovaný .NET (Python, Visual Basic, JavaScript, ...). Takže můžete mít Evolution plně pod kontrolou, naskriptovat jej, používat všechny mailové vlastnosti.

**\* Otázka:** Vidím, že obecně preferuješ C#.

▷ **Odpověď:** Ano. Já používám C#, ale další v týmu dávají přednost Pythonu nebo jiným. C# je nejlehčí z nich, protože C# v podstatě nemá run-time. C# komunikuje přímo s knihovnou, odpadá tedy prodleva oproti jazykům typu Python nebo Java, které navíc potřebují run-time. Myslím si tedy, že C# je nejlehčí, ale to neznamená, že na tyto vlastnosti musíte hledět při výběru jazyka. Já vždycky říkám, že C# je prostě jazyk, který znám, a tak ho používám.

**\* Otázka:** Někteří lidé mají podivný pocit z Mona – vidí Microsoft, jeho technologie zatažené do Unixu, problémy s autorskými právy...

*(Miguel se dává houskou...)*

▷ **Odpověď:** Ne, ne, ne! Nejsou žádné problémy s autorskými právy.

\* **Otázka:** Dobře, tak tedy technologie Microsoft, jeho technologie a tak dále. Ale na druhé straně je dobré mít zadní vrátka pro případ úspěšného rozšíření .NET. Co na to říkáš?

▷ **Odpověď:** No, já právě pracuji na úspěšném rozšíření .NET. Například, během zhruba devíti měsíců od vypuštění Mona bylo vytvořeno přes 40 desktopových aplikací. Minimálně jsme tedy dosáhli toho, aby programátoři začali psát aplikace pro desktop. Toto bylo v minulosti těžké.

\* **Otázka:** Co by nám třeba přineslo přepsání aplikace jako je Gimp do Mona? Vždyť to bude jen pomalejší, ne?

▷ **Odpověď:** Dvě věci. Za prvé si myslím, že bychom neměli přepisovat software, protože pokaždé když si řeknete, že přepíšete nějaký software, tak se pouštíte do mnohaletého projektu. Gimp má kořeny někde kolem roku 1996 a už jednou přepsán byl. Tuším, že to bylo v roce 1997, kdy byl přepsán z Motif do GTK. Ti samí lidé totiž GTK napsali. Mluvíme tedy o aplikaci, která má za sebou osm let vývoje. Nevěřím, že bychom měli investovat čas do přepisování kódu.

Když bychom potřebovali rozšířit tuto aplikaci lepším programovacím jazykem, můj návrh by zněl – začleňme (embed) Mono. Mono je začlenitelné jako Python, můžete spojit aplikaci s Monem a rozšířit ji v Monu, ale nemusíte přepisovat původní kód. To by bylo na debug milionů řádků. Takže má rada zní – hotovou věc nechte být a jen ji rozšiřujte. Můžete třeba napsat nové vlastnosti v C#, ale zachováte existující kód. Myslím, že toto je cesta k úspěchu. Opravdu odrazují lidi od přepisování. Byl návrh na přepsání Evolution, ale já řekl, že to nemá smysl, použijme bindings. Kód je funkční, odladěný, používá se, už je nasazený. Nemá význam jej roky přepisovat. Dělal bychom znovu stejnou chybu, znovu jej odladovali a tak. Jsme jen lidé. Když už něco začínáte, tak to napište v Monu, a pokud už chcete rozšiřovat tak v Monu.

\* **Otázka:** Váš projekt **Gnome** [2] má nový směr; směr zvaný „just works“ [prostě funguje]. Nejsou zde žádná nastavení pro experty, pouze jednoduché konfigurace a tak. Ale někteří uživatelé nejsou spokojeni s tímto stylem a stěžují si. Vy ale chcete zůstat při tomto?

▷ **Odpověď:** To je na delší odpověď. Cílem Gnome je učinit počítače přístupnými pro normální lidi. Nejsme příliš velký tým. Nejdříve bych rád uvedl pár faktů. Chceme Gnome dělat tak, aby bylo použitelné pro běžné lidi, chceme aby bylo robustní, chceme aby lidi měli robustní desktop. Nechceme mít špatnou reputaci, jakou Windows pravděpodobně mají. No, ona je asi dobrá, na tom nezáleží, my ji mít špatnou nechceme. Ehm. Aby lidé při používání Gnome cítili, že je spolehlivé. Že prostě funguje.

Mají z používání dobrý pocit. Velmi podobně jako u **Mac OS** [3]. Myslím, že Mac OS je velice populární a stává se ještě populárnějším zejména ve Spojených Státech, je to velmi jednotný operační systém. A je velmi podobný Gnome. Nemá spousty vlastností nebo nastavení. Co je na Macu zajímavé – dají vám jen základ, a když pak chcete extra nástroje, tak si je vždy můžete odněkud stáhnout, a tak rozšířit jeho možnosti. Ale nemusíte mít všechny tyto nástroje, protože i bez nich se dá se systémem pohodlně pracovat. Je zde kladen velký důraz na jednoduchou použitelnost.

Strávili jsme například na poslední konferenci hodiny díváním se na záznamy lidí. Natočili jsme lidi při používání Gnome, pozorovali jsme jejich ruce, myš, další kamera pozorovala obrazovku a další sledovala jejich obličej. Požádali jsme: „Připojte se prosím na internet.“ Byli to lidé používající Windows většinou reagovali takto: „OK, co mám dělat?“ Našli ikonku Firefoxu, klikli na ni a dostali Firefox. A pak se zkoušeli nějak připojit. „Jak bych se jen připojil?“ A tak začali prohledávat menu. Z těchto záznamů jsme pochopili, jaké chyby lidé dělají. Poučili jsme se – viděli jsme těch záznamů patnáct, možná více. Tým, který dělal výzkumy použitelnosti, byl poslán do Evropy – do Švédska,



Dánska, byli po celých státech, v Německu. Udělali jsme všechny tyto studie, abychom pochopili, jak lidé používají počítač.

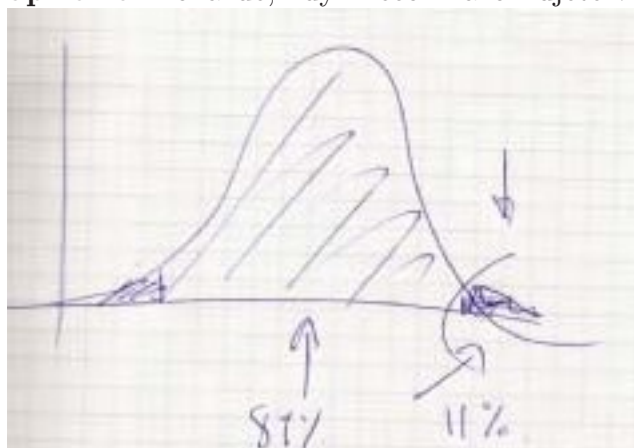
Nyní tedy máme data, která využijeme k odstranění těchto problémů a překážek z Gnome. Je hodně zajímavé sledovat tyto studie, protože můžete vidět, jaké mají lidé problémy. My, jako programátoři, různé překážky a nástrahy nevidíme. Když máte problém, víte, jak se z něho dostat.

A teď k druhé části mé rozsáhlé odpovědi, a to je přidávání nových vlastností. Vzpomínáte, jak jsem vám před chvílí říkal, že máme málo lidí? Pokaždé, když přidáte dvě různé volby, každé, když dáte možnost výběru. Takže můžete zvolit cestu A nebo B. Například jako checkbox. To znamená, že váš program se může vydat dvěma různými cestami. Může se dát jednou nebo druhou. Z toho plyne, že když přidáte nějakou vlastnost nebo rozšíříte váš program, musíte ho otestovat dvakrát. Jednou pro každou cestu. Jestliže vše půjde bez problémů, takto se setkají a pokračují jako jedna cesta. Nyní se však podívejme na nejhorší možný případ. Ten nastane tehdy, pokud se cesty již neseťkají a zůstanou oddělené. Musíte tedy testovat každou zvlášť. A teď si to představte s celým zástupem voleb, kdy se to štěpí a štěpí (!100). Testování se stává spletitějším a je zde možnost, že program nebude pracovat správně a bude zkrátka zahalen rouškou tajuplné nadpřirozenosti. Aplikace může dokonce spadnout a lidi si pak stěžují.

Myslím, že Linux nemá tolik vývojářů, kolik by měl mít. Rozhodli jsme se, že uděláme základy pořádně. Jakmile jednou máte opravdu pevné základy, můžete zvažovat, co dál. Dost toho jde zvládnout i bez přidávání spousty nastavení, protože pokud dychtíte po tisících nastavení, vždy je tu ta možnost stáhnout si to ze Sourceforge nebo Freshmeat. My ale dáme lidem fungující a standardní prostředí, chcete-li však něco navíc, řekneme vám, kde to nalézt. Zároveň jsme dělali na přidávání takových malých drobností, třešniček na dortu. Tamto bylo vaše hlavní jídlo, hlavní chod a my jsme také pracovali na dezertu, na malých věčičkách pro lepší požitek. Například nová verze GTK [4] používá pro vykreslování Cairo [5], celé to stojí na vektorových základech. Můžete přibližovat tlačítka, zase je oddalovat, se vším lze otáčet, můžete prostě dělat spousty divokých věcí. Hodně úsilí bylo také vloženo do hardwarové akcelerace v Cairo, takže vlastníte-li grafickou kartu podporující OpenGL, bude to opravdu rychlé. Pracovali jsme i na pěkných efektech, například nové verze přepínání plochy. Když se přesunete z jedné virtuální plochy na druhou, namísto klikání na náhled potáhnete vaši plochu a ona zarotuje jako krychle. Opravdu pěkné.

**\* Otázka:** Jestli tomu tedy správně rozumím, zaměřuje se Gnome na normální uživatele, nikoli experty, ale normální lidi, kteří chtějí jen používat počítač.

▷ **Odpověď:** A zde je prapříčina. Pokaždé, když něco znázorňujete v populaci...



**\* Otázka:** Gaussova křivka, použijeme Gaussova křivku (Bell curve)!

▷ **Odpověď:** Skvělé, přesně víš, o čem mluvím. Potíž je v tom, že experti jsou tito chlapi a idioti jsou tito, je jich pouze pár. Chceme tedy zaměřit naše snahy na tyto lidi, ne? Proč se nezaměřit na tyto lidi? Protože tohle je mnohem větší skupina, 89 %. Protože ostří hoši jsou tak dobří, že mohou

jít na Sourceforge a sehnat si něco lepšího nebo dokonce napsat software. Pokud těmto lidem něco chybí, nebo se jim nelíbí, tak zkrátka změní konfigurační soubor, nebo mají Gentoo a překompilují si to se svým nastavením. Proto se nezajímám o tyto lidi, ehm, ne přímo nezajímám, ale nehodlám dělat věci jen pro ně, chci dělat věci pro všechny. Dává to smysl?

★ **Otázka:** Ano.

★ **Otázka:** V kterém ohledu jsou podle tebe lepší technologie Gnome oproti KDE?

▷ **Odpověď:** Já velmi nerad srovnávám s KDE.

★ **Otázka:** Ale všichni ostatní ano...:-)

▷ **Odpověď:** Myslím, že každá skupina měla odlišná rozhodnutí. Gnome je mladší, přišli jsme téměř o rok později, bylo to asi devět měsíců. Vždy jsme byli v nevýhodě co se týče vývojářů a takových věcí. Pár věcí ale bylo naší výsadou, prospívali jsme díky licenci GTK, což je LGPL. To znamená, že kdokoli může napsat aplikaci a nemusíte platit. S Qt musíte platit, pokud chcete psát komerční software. Tato licence tedy znamenala, že společnosti jako Sun řekly, že budou používat Gnome jako desktop, protože si nemohou dovolit prodat něco zákazníkům a pak od nich očekávat, že zaplatí ještě \$2000 jiné společnosti, aby mohli psát software. Takováto LGPL licence zaujala také společnost Eazel, nevím jestli si to pamatujete, to byla společnost složená z bývalých inženýrů z Apple.

★ **Otázka:** Ti mají na svědomí Nautilus.

▷ **Odpověď:** Přesně, udělali Nautilus a potom zmizeli. Ale důležitá byla kultura, kterou přinesli, kultura použitelnosti a jednoduchosti a všech těch věcí, které jsme již diskutovali. To je něco, co donesli do Gnome, protože předtím jsme byli pouze unixovým desktopem a velká změna nastala, když jsme se rozhodli být desktopem pro uživatele. Ne pro unixové experty a programátory, ale pro uživatele. Asi v roce 2000 jsem s nimi mluvil, teď si uvědomuji... to je zajímavé, byli jsme hotoví už v 1999, ale společnost byla založena až 2000. Každopádně tuto kulturu použitelnosti musíte nacpat programátorům do hlavy, nedělají to totiž jen pro sebe. Opravdu chceme, aby Free Software uspěl, ale pro masu, ne jen pro několik z nás nebo pár přátel. Toto pojetí použitelnosti bylo s Gnome po pět let. Neměli jsme žádný hlavní směr jako ostatní. Ale zaměřili jsme se hlavně na to, abychom dali věci do pořádku a udělali to pořádně. KDE naopak, a to je můj názor, jen pokračovalo v přidávání nových vlastností. Vše bylo jen o rozšiřování vlastností místo o přemýšlení, co vlastně uživatel potřebuje. Mám takový vtíp, není to ani tak vtíp – když zapnete nějakou KDE aplikaci, nebo když se na ni dívá vývojář a najde malé místo, kde není widget (grafická komponenta, tlačítko například), tak ten prostor něčím vyplní. V tom je tedy mezi námi rozdíl a myslím, že z hlediska použitelnosti a hlediska uživatele je Gnome lepší.

A to je důvod, proč si velcí zaměstnavatelé a úředníci ve velkých městech používající Linux vybrali Gnome. Vyhráváme v testech použitelnosti v každém jednotlivém ohledu. Siemens, Porto Alegre, Rio Grande, Extramadura Andalusia, což je jeden z největších španělských zaměstnavatelů, 200000 Gnome desktopů k dnešnímu dni. Madrid a Barcelona také ohlásily, že přejdou. A to proto, že jsme se zaměřili, a pět let na tom pracovali, na porozumění problému: Je tato ikona matoucí? Může toto plést nebo zmást? Všechny tyto věci nejsou o programování, nýbrž o způsobu, jakým lidé vnímají své prostředí. Máme technologie, máme licenci, která nám pomáhá přivábit obchodní prodejce, jako například Nokia. Také máme technologie jako je Mono. C++ je velmi dobrý jazyk, ale je to jazyk pro experty. Mono dovoluje lidem, kteří nejsou experty, programátorům ve Visual Basicu, lidem ze světa Windows, těmto lidem dovoluje přejít do našeho světa. Děláme pokroky v nárůstu počtu linuxových desktopů, nesnažíme se ho dělat jen „cool“.

★ **Otázka:** Co říkáš na vyloučení Gnome ze Slackwaru?

▷ **Odpověď:** Tak trochu rozumím, proč to Patrik udělal. Je to chlápek, co dělá celou distribuci sám. A chápu proto, že si chce redukovat objem práce. A pokud používá KDE, tak ať

jej používá. Slackware je lidmi chápán jako distribuce pro experty. Neobsahuje balíčkovací systém, upgrady a takové věci. Je to smutné, ale na druhé straně velké distribuce používají Gnome - RedHat, Ubuntu, SUSE, i když SUSE používá dva, ale mělo by používat jen jeden - máme s tím ve firmě vnitřní problémy. A některé jiné používají výhradně Gnome, a mohou se tak na něj zaměřit.

★ **Otázka:** Znáš Patrika Volkerdinga osobně? Mluvil jsi s ním o Gnome?

▷ **Odpověď:** Myslím, že jsme se mohli jednou potkat na nějaké konferenci. Možná, opravdu si nevzpomínám.

Miguelův článek: [Let's make unix not suck](#) [6]

★ **Otázka:** Četl jsem tvůj esej „Let's make unix not suck“. Opravdu si myslíš, že Unix stojí za houby?

▷ **Odpověď:** V podstatě to bylo napsáno k získání pozornosti na unixové konferenci. Lidé se podle mne zaměřovali na nesprávné problémy. Nebyl to ani tak článek, bylo to napsáno za účelem ohlasu, aby o problémech začali lidé diskutovat. Spousta lidí psala do blogů, různé články...ale neporozuměli, protože povětšinou slyšeli jen informace od někoho dalšího z druhé ruky. Nechte mě tedy vysvětlit co jsem tím myslel. V Unixu jsou jistě místa či elementy, jež by bylo lepší předělat na API, než aby zůstaly souhrnem ne úplně nejlépe spolupracujících komponent. Například věci jako tiskárny, nebo úlohy typu – jak získat informace o tiskárně? Některé věci se už řeší – Bonobo může poskytovat rozhraní k Evolution, GConf vám umožňuje mít nastavení pro celý systém a D-BUS [7] na výměnu informací mezi aplikacemi. Takže pracujeme na vyřešení těchto problémů.

★ **Otázka:** Ano, ale osobně GConf rád nemám, připomíná mi to registry ve Windows.

▷ **Odpověď:** GConf je odlišný ve dvou velmi důležitých věcech. První rozdíl je ten, že windowsové registry je binární databáze. Jediná společná věc mezi GConf a windows registry je, že máte klíč pro konfigurační hodnotu a získáváte hodnoty. Ale takto to není v GConf, každá aplikace může nainstalovat soubory, jež popisují její nastavení; může taky obsahovat dokumentaci. Nejsou zde problémy s narušením dat, registry je binární databáze a jestliže se poškodí, tak je poškozeno vše, celý systém. V GConfu žádná binární databáze není.

★ **Otázka:** Pro mě je také těžké něco v té databázi upravit ručně, například textovým editorem.

▷ **Odpověď:** Ano, je to XML soubor. Správná cesta k upravování je použít tyto dva nástroje: `gconf-tool` nebo IDE editor `gconf-edit`, který spustí ten první. Věcí, ve které se také GConf a registry masivně liší, je, že registry je pouze databáze na vyhledávání. Gconf je živou databází. To znamená, že když učiníte nějakou změnu, tak každá aplikace tuto změnu zaznamená. Takže tak. Pozoroval jsi to někdy?

★ **Otázka:** Hmm, no jo.

▷ **Odpověď:** Když jakákoliv aplikace nebo nástroj pro příkazovou řádku změní nějakou věc v databázi, tak každá aplikace, která se o to zajímá, se to dozví.

★ **Otázka:** Jo, to je pravda.

▷ **Odpověď:** Tak například jedna ze zajímavých věcí – když máte dialogové okno s výběrem barvy nebo nějaké jiné nastavení, například checkbox. Podívejte – Gnome má pouze zavírací tlačítko, není zde žádné tlačítko OK, protože je to tak pro lidi jednodušší. Tlačítka OK/Zrušit jsou vynálezem z dob, kdy počítače byly velmi pomalé, udělali jste změnu, ťukli na OK a pak započal předlouhý čas výpočtů. Například jste si vybrali pozadí, klikli na OK a čekali dlouho dobu, než se celý obrázek načte. Nyní jen vyberete obrázek a je to – okamžitě se objeví. Co je na tom všem zajímavé je to, že žádný z těchto dialogů již nepotřebuje hromady kódu, protože máme napsané rutiny které aplikují změny – aplikuj X nebo aplikuj Y. Dialogové okno je jen front-end pro klíč v GConf, už nepotřebujete nic psát. Existuje dokonce i nástroj, který napojuje Glade – GUI s klíčem v GConf,

takže není potřeba psát kód. A automaticky dostanete toto chování - když uděláte změnu, tak se automaticky aplikuje a hned projeví. To je důvod, proč máme rádi GConf.

★ **Otázka:** To je hlavně pro grafické aplikace.

▷ **Odpověď:** Je to zamýšleno pro grafické aplikace a nezapomeň – GConf je vytvořen čistě na uživatelské bázi. Není to pro systém, je to pro uživatele. Další věcí na GConfu je to, že má back-end, jednotlivce to nezajímá, ale společnosti ano. Když máme tisíc strojů a já změním proxy, nebo třeba změním něco na e-mail serveru, nějaké nastavení. Změnu na e-mail serveru provedu centrálně na jednom místě a tisíce počítačů ji za běhu přijme. To je vlastnost, kterou jako jednotlivce nepotřebujete, ale velcí zaměstnavatelé ano. Novell má například pět tisíc zaměstnanců a všechny jsme je převedli na Linux, no, nyní asi 90 % společnosti. Jsme největší společnost používající Linux.

★ **Otázka:** Skvělý úspěch.

▷ **Odpověď:** Musíme přetáhnout lidi z Windows na Linux. Chtěl jsem tedy říct, že GConf má i některé vlastnosti užitečné jen pro velké společnosti. A samozřejmě ho používat nemusíte, jestli nechcete. A nemá problémy jako registry, jen má podobné GUI, ale jinak to není stejné.

★ **Otázka:** Co si myslíš o novém operačním systému Microsoftu – Longhorn? Vypadá to, že nezvládají dodělání...

▷ **Odpověď:** Myslím že to je velmi, velmi dobrá šance pro nás vytěžit z toho výhodu. Toto je pro nás ze strany Windows vlastně taková příležitost, máme něco času na převedení společnosti na Linux, taktéž vlády i jednotlivé uživatele. Je třeba udělat tři věci: musíme z něj udělat použitelný systém, je třeba ulehčit lidem jednoduchý vývoj specifických aplikací – ne aplikací, jež každý používá jako třeba prohlížeče a e-mailové klienty, ale velmi specifické věci jako účtování pro hotely nebo podpora pro správu vodního systému měst. A proto jsme přišli s Mono, potřebujeme lidem přechod z Windows na Linux usnadnit. Teď je skvělá příležitost a my ji musíme využít, takže řekněte vašim čtenářům, ať začnou psát GTK# aplikace s Mono.

★ **Otázka:** OK :)

★ **Otázka:** Dostaneme se do styku s hodně OS, který je tvůj oblíbený?

▷ **Odpověď:** Používám svůj operační systém protože je free a já jsem freesofták. Myslím si, že nejpěknější dnes je Mac OS X, kvůli opravdu skvělým vlastnostem. Myslím, že Solaris má jedny z nejlepších vývojářských nástrojů, hlavně ladící (debug) nástroje – dtrace se to myslím nazývá. Když to rozběháte, tak je Solaris v pohodě, ale je třeba vložit do toho hodně GNU věcíček. Solaris je extrémně stabilní, OS X nádherný. Linux je svoboda, kterou potřebuji, takže ho používám. Jsem jeho každodenním spotřebitelem, protože já chci Linux zdokonalit, ačkoliv ostatní operační systémy mají také výhody. Windows má zřejmě každou hračku, každou hru na světě. Já zůstávám u Linuxu, protože chci vidět tenhle trh růst.

★ **Otázka:** Jakou distribuci používáš?

▷ **Odpověď:** Používám Novell Linux Desktop, což je enterprise verze,...to znamená že máte pořád stejnou verzi a oni vám posílají bugfixy, ale neupdatují to. Mám rád pouze jednu nestabilní věc na svém stroji, nemám rád všechno nestabilní. Takže jediná nestabilní věc na mém stroji je ta, na které pracuji, a to je Mono. Nechci mít nestabilní svého správce oken nebo poštovního klienta. Zvolil jsem si to tak, používám enterprise verzi namísto cutting-edge SUSE. Používám staré verze, nejsou tak staré - asi tak 9 měsíců. Používám něco, co funguje.

★ **Otázka:** Směl bych se nyní zeptat na něco o tvém životě a osobně o tobě?

▷ **Odpověď:** Jo, jo.

★ **Otázka:** OK. Tvoje práce vypadá velmi zajímavě, ale musí to být velmi těžké. Cestuješ často kvůli práci?

▷ **Odpověď:** Nyní cestuji méně, jelikož jsem se minulý rok oženil...

★ **Otázka:** Gratuluji.

▷ **Odpověď:** Díky. Takže se nyní snažím cestovat méně, ale stále trochu cestuji. Byl jsem právě v Německu, tady v Plzni, teď v Praze, dnes ještě letím do Paříže a pak se chystám vrátit zpět domů.



★ **Otázka:** Takže většinu času trávíš doma?

▷ **Odpověď:** Ano, snažím se. Ale třeba když je čas vypuštění nové verze Mona, musím si být jist, že je připravena na vypuštění a musím pracovat mimo domov. Takže se pokouším být doma, avšak není to vždy možné. Má práce mě ale baví, cestovat a potkávat lidi, sledovat jaké věci lidé vymýšlejí s Linuxem a tak.

★ **Otázka:** OK. Stále programuješ, nebo už jen řídíš společnost a lidi?

▷ **Odpověď:** Dnes programuji už velice málo. Jádro Mono týmu má dnes 25 lidí a já to musím řídit. To znamená, že říkám lidem: „ty udělej to, ty tohle, vy udělejte tohle dřív než my tamto...“ A také se musím setkávat s lidmi, kteří používají Mono. Třeba ve finančnictví, průmyslu, softwarovém vývoji atd. Můj hlavní cíl je růst Linuxu a věnuji tomu spousty času, ale také musím zajistit dostatek peněz na zaplacení 25 lidí pracujících na plný úvazek na Monu; to samé platí pro desktop tým. Platíme tedy hackery, aby dělali na Monu. Je ostuda že neprogramuji, ale důležitější je, abych dělal tyto věci a až poté mi zbude něco málo času pro psaní kódu. Stále je třeba něco malého tu a tam dopsat, hlavně v Monu, ale není to mojí prioritou.

★ **Otázka:** Hodně cestuješ. Který stát nebo město je tvoje oblíbené, a kam by ses rád vrátil? Jak se ti líbí Praha?

▷ **Odpověď:** Určitě také do Prahy, protože jsem z ní neviděl mnoho. Rád bych zde navštívil pár jistých míst. Jedny z mých oblíbených měst jsou tyto: Barcelona, Madrid, Rio de Janeiro, Paříž, Istanbul. Také bych se rád vrátil na pár míst, jelikož jsem z nich neviděl dost, jako třeba Bejrút. Nikdy jsem třeba nebyl a moc rád bych navštívil: Káhiru, Peking a Tokio. Stále je toho hodně co dělat.

★ **Otázka:** Život je krátký, žij naplno.

▷ **Odpověď:** Přesně.

★ **Otázka:** Jaké jsou tvoje koníčky? Jak trávíš svůj volný čas?

▷ **Odpověď:** Ani vlastně nevím. Mé záliby se v průběhu času mění. Hodně jsem se naposledy zabavil čtením knih. Zrovna jsem četl o historii Libanonu a konfliktu v Palestině. Myslím, že nyní

rozumím tomuto problému velmi dobře. Také jsem začal hrát na pár hudebních nástrojů – piano, flétna, kytara. Ale jsem strašný. Je to ale teď jedna z věcí, které dělám ve svém volnu.

★ **Otázka:** Na tvých osobních stránkách jsou odkazy týkající se konfliktů v Palestině. Angažuješ se? Nebo tě to zajímá?

▷ **Odpověď:** Ano. Myslím si, že tito ubozí lidé měli... Já vlastně nevím, jak jsem se k tomu dostal. Jednou jsem s někým mluvil a ničemu z toho, co říkal, jsem nemohl uvěřit. Po našem rozhovoru jsem si stále myslel, že to není pravda, přišel jsem domů a začal si vyhledávat informace a ukázalo se, že všechno, co mi řekl, je pravda a situace je horší než jsem si myslel. Prostě jsem nikdy předtím nevěděl, jaké strašné věci se tam dějí. Myslím si, že spousta jiných lidí neví, co se děje, co je tam za problém; stejně jako předtím já. A měli by to vědět. Myslím, že palestinský problém se datuje od vytvoření Izraele v roce 1948, když vyhnali, nevím kolik miliónů Palestinců ze země.

Vyhostili tyto lidi, vyhodili ze země během války. A tito lidi utíkali sami, protože nechtěli být ve válečné oblasti. A když bylo po válce, tak je už nepustili zpátky, řekli jim: „Nemůžete se vrátit, když jste utekli do jiné země.“ A jiné země nechtěly uprchlíky. Je to jako, představ si, že byste se zítra probudili a měli byste přede dveřmi celé Rakousko a chtěli by zde pracovat. Asi byste reagovali: „Moment, tohle je moje země.“ Spousta problémů, chtělo by se mi říct většina, na Blízkém východě vychází z toho, že Libanon, Jordán, Sýrie mají početnou populaci Palestinců na svém území. Tito lidé žijí v uprchlických táborech, nemohou pracovat, nemohou práci dostat. Jenom v Libanonu je milión lidí, kteří jsou jako ve vězení. Žijí na tomto území a nemohou mimo něj pracovat nebo stavět domy. Takže postaví jeden dům a potom jej musí rozšiřovat a přistavovat další podlaží a další, další... Je to velmi nebezpečné, domy jsou křivé, chatrné a nestabilní, protože další patra jsou širší než předchozí. Tyto země zkrátka nechtějí milión pracujících lidí navíc ve své zemi a Izrael je nevezme zpět, to je jádro problému. Je to hrozné lidské utrpení. Ale co můžeš dělat?

★ **Otázka:** A poslední otázka. Filozofická otázka. Uvědomuješ si nicotnost lidského bytí při pohledu na hvězdy za letní noci?

▷ **Odpověď:** Nedívám se na hvězdy moc často :-)

★ **Otázka:** Tak to bys měl ;-)

▷ **Odpověď:** Jo, měl bych. Jsme na této planetě pouhou náhodou a přirozený přírodní výběr vytvořil naši vědomou mysl. Chopme se tedy výhody této jedinečné příležitosti, nebo bude pryč dřív než si to stačíme uvědomit. Jde jen o to, abychom z toho vytěžili co nejvíc, dokud jsme tady.

★ **Otázka:** Velmi děkuji za tvůj čas, přeji pěkný den a příjemnou cestu. Na shledanou.

## Odkazy

- [1] <http://www.abclinuxu.cz/Profile/7602>
- [2] <http://www.gnome.org/>
- [3] <http://www.apple.com/macosx/>
- [4] <http://www.gtk.org/>
- [5] <http://cairographics.org/introduction>
- [6] [http://www.xahlee.org/UnixResource\\_dir/\\_/bongo-bong.html](http://www.xahlee.org/UnixResource_dir/_/bongo-bong.html)
- [7] <http://www.freedesktop.org/Software/dbus>

\*\*\*

---

## Reklamní nabídky [www.praceabc.cz](http://www.praceabc.cz)

Pozice: Programátor C/Unix – letecký průmysl (3 volná místa) [2076]  
Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 21.07.2005

Pozice: Technická podpora pro zahraniční zákazníky [2069]  
Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 21.07.2005

# Audio v Linuxu

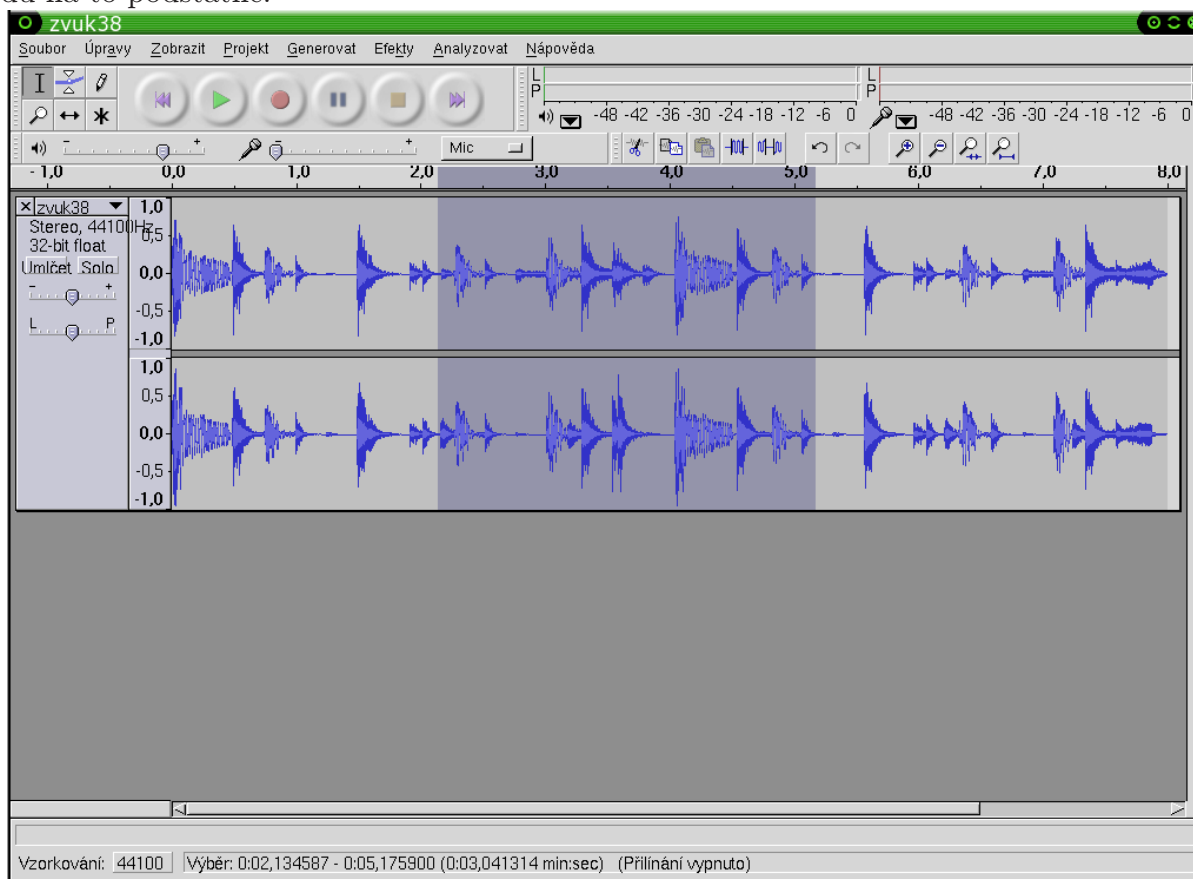
Pavel Beňak

*V tomto seriálu najdete přehled možností, které nám Linux nabízí pro práci se zvukem. Nahrávání, editace, efekty. Projekty Jack, LASH a LADSPA. Nejdříve si vyjasníme, co po Linuxu vlastně budeme chtít. Pro vytváření skladeb na PC se používají MIDI sekvencery. V tomto článku si povíme o dvou nejvýznamnějších pro GNU/Linux – Rosegarden a MusE.*

- Nahrávat zvuk.
- Upravovat nahraný zvuk.
- Efekty na zvuku v reálném čase.
- Pracovat s MIDI.
- Vydávat z počítače zvuky na základě MIDI informací (používat softwarové syntetizéry a samplery).
- Zapisovat noty (v souvislosti s MIDI, ale i samotné).

## Úvod

Možnosti jsou dobré. Pokud chcete zvuk nahrávat a nahraný zvuk poté upravovat nebo třeba provádět efekty na zvuku v reálném čase, budete spokojeni. Dobré jsou i možnosti práce s MIDI a zařízeními podporujícími MIDI. I při vytváření hudby nebo zpracování zvuku spolupracuje dohromady několik aplikací. Jestli jste někdy pracovali s programy pro Windows (jako Cakewalk Sonar nebo Steinberg Cubase), možná se ze začátku můžete cítit trochu ztraceni. Na druhou stranu je toto řešení dost flexibilní. Programátor se navíc při vytváření svého programu (třeba zvukový efekt) může soustředit opravdu na to podstatné.



Audacity

## Software

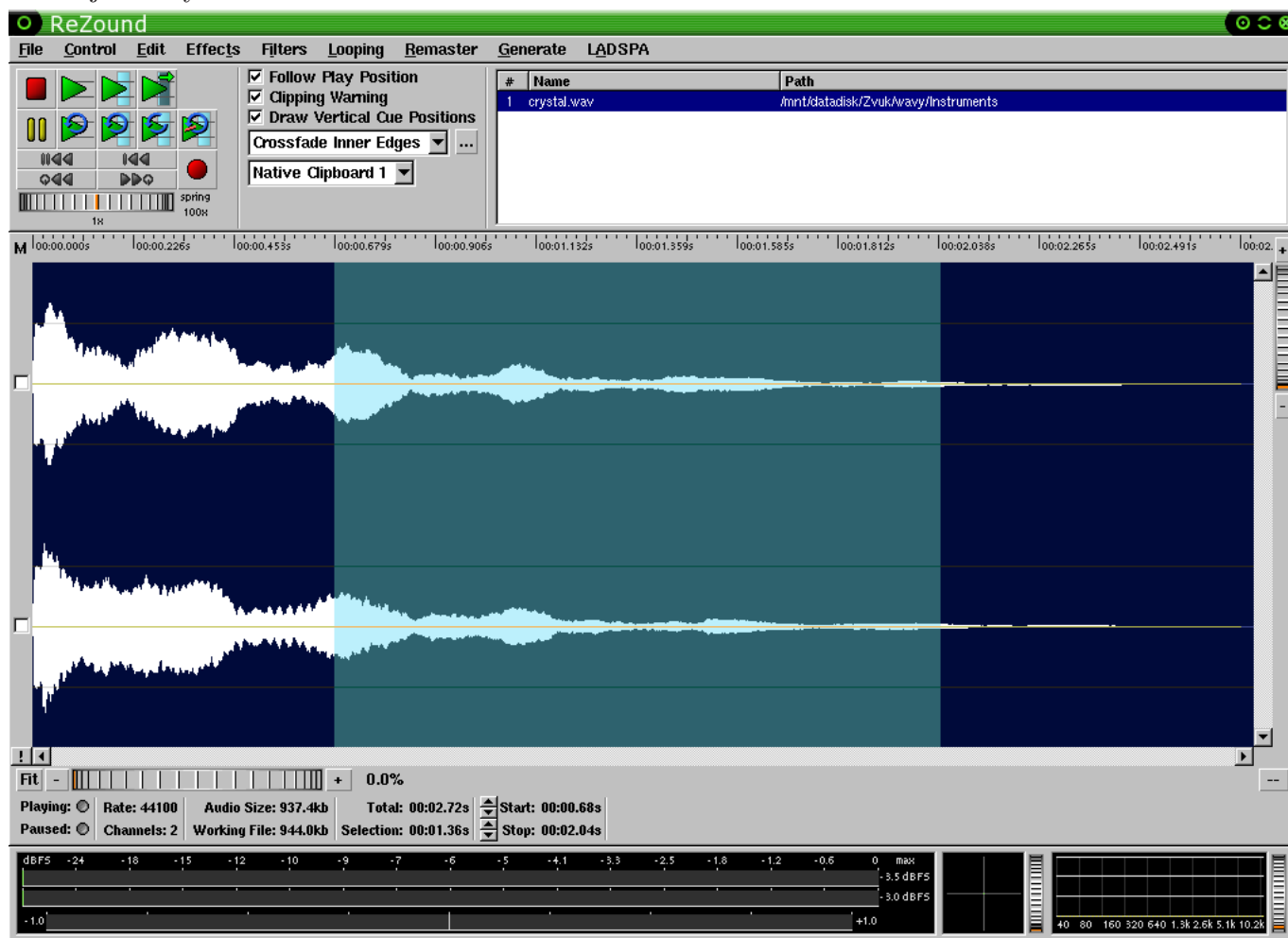
Zaměříme se zde pouze na programy, které se dají ovládat z grafického rozhraní. Pro nahrávání můžeme použít třeba programy [Audacity](#) [1], [ReZound](#) [2] nebo [Ardour](#) [3].

### Audacity

Multiplatformní Audacity vypadá na první pohled jednoduše. Přesto je to vícestopý program, který vám při nahrávání plně poslouží. Editace a pohyb i ve velkých souborech je rychlý, podporuje pluginy LADSPA a VST (pod Linuxem je to s nimi trochu složitější). Vlastní pluginy si můžete napsat ve skriptovacím jazyce Nyquist (je to jazyk odvozený od LISPu). Funkcí, která si zaslouží vyzdvihnout je (mimo jiné) odstranění šumu. Audacity je poskytován pod licencí Creative Commons. Myslím, že hlavní použití tohoto programu je pro nahrávání živých nahrávek nebo třeba audiokazet a jejich úpravu.

### ReZound

ReZound vypadá (alespoň podle mě) trochu techničtěji :-). Budí dojem, že je schopnější. Podporuje efekty LADSPA a umí spolupracovat i s audioserverem JACK. Není to vícestopý editor zvuku. Pohyb ve velkých souborech je od verze 0.10.0beta dost pomalý. Na druhou stranu tu najdete i funkce, které vám pomohou hlavně při úpravě samplů nástrojů – například zarovnání označené části zvuku tak, aby ve smyčce zněl hladce. Také různé režimy přehrávání, díky kterým si můžete otestovat, jak bude nástroj ve smyčce znít.



*ReZound*

Další zajímavou funkcí programu je možnost vypálení CD z otevřeného souboru (stopy se dělí podle tichých částí). Vzhledem k pomalému zpracovávání a pohybu ve velkých souborech se tato možnost asi moc nevyužije. Podle mailing listu se dá rychlost zvýšit snížením interního rozlišení vzorků z 32

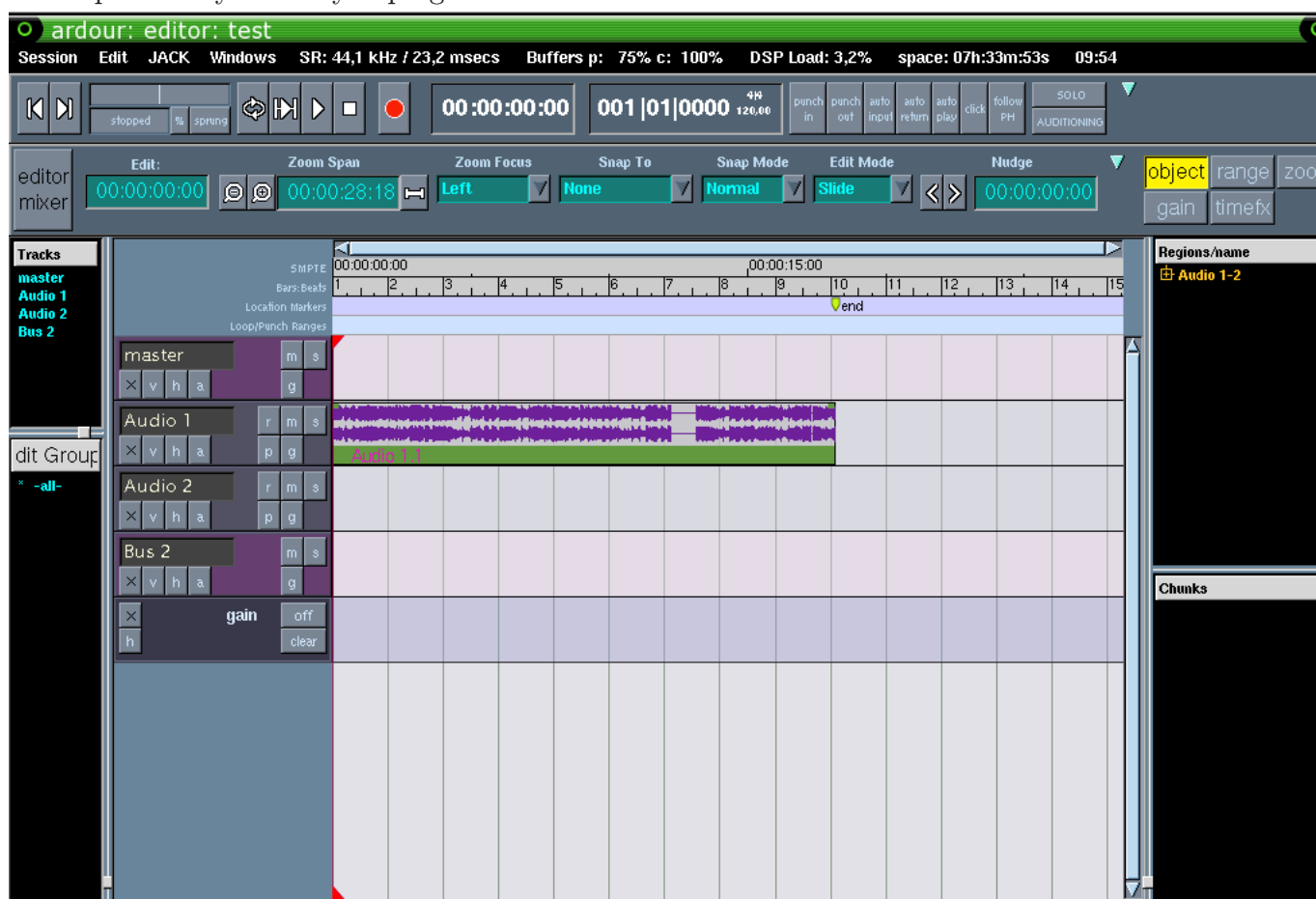


bitů na 16 bitů (k tomu je potřeba rekompile program). To s sebou samozřejmě přinese ztrátu kvality při zpracování zvuku.

Doufám, že se rychlost v dalších verzích zvýší, je to hlavní nectnost programu. Přesto je to výborný nástroj na úpravu zvukových samlpů.

## Ardour

A dostáváme se k velmi zajímavému programu s mnoha možnostmi – Ardour. Je to vícestopý nahrávací program, který spolupracuje s audioserverem JACK. Umožňuje měnit parametry zvukové stopy a použitých efektů v průběhu přehrávání. Hodnoty těchto parametrů můžete nastavit pomocí křivek, kde v každém bodě, který definujete, bude mít parametr určitou hodnotu. Je to výborný program pro nahrávání v reálném čase a pro další nedestruktivní úpravy (změny nejsou ihned zapsány do souboru). Podporuje pluginy LADSPA, jejichž parametry je možné nastavovat i pomocí MIDI kontrolerů. Program je vhodný pro složitější zpracování vícestopých nahrávek – hlavně díky možnosti měnit parametry zvukových pluginů.



Ardour – hlavní okno

## JACK

*Poznámka: Kapitalka byla opravena a rozšířena.* JACK je nízkolatenční audioserver. Umožňuje spojovat vstupy a výstupy jednotlivých aplikací nebo zvukové karty. Tak můžete zřetězit libovolné množství aplikací za sebou. Ostatní aplikace musí být spuštěny stejným uživatelem jako audioserver, jinak se k němu nedokáží připojit.

Latence je prodleva mezi požadavkem a vyřízením žádosti. Pokud je latence vysoká, dojde k výpadkům (aplikace čte vstup dat a data se někde zdržela, takže vznikne prázdné místo). Závisí jak na hardwaru, tak na jádře systému. Jack se snaží udržovat nízkou latenci použitím bufferů, latence ale velkou měrou závisí na systému. JACK potřebuje mít oprávnění pro operace v reálném čase. Ty má třeba uživatel root. Nevýhodou tohoto řešení je, že všechny klientské aplikace by se také musely

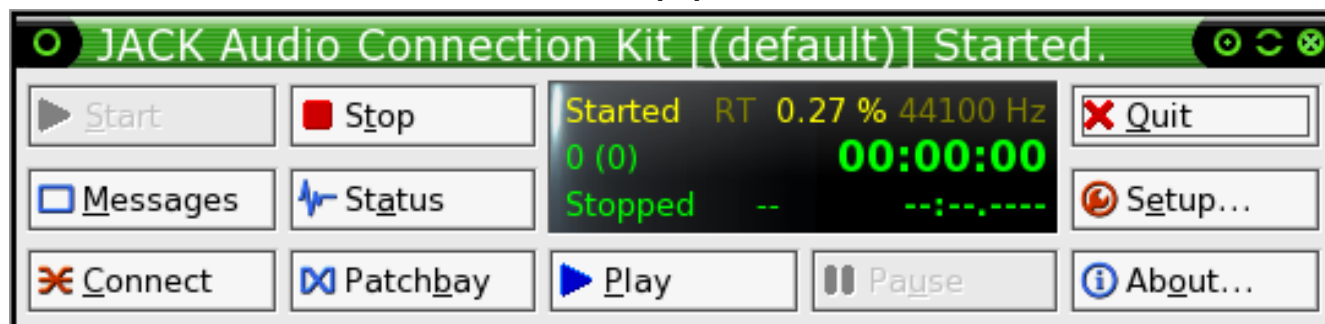
spouštět pod tímto uživatelem. Řešení latence a práv pro operace v reálném čase se liší podle verze jádra.

Pro jádra řady 2.4 jsou [nízkolatenci patche](#) [4] od Andyho Mortona. Pro získání práv pro operace v reálném čase (bez nutnosti být přihlášen jako uživatel root) pak použijete program [givertcap](#) [5]. Podle návodu na stránkách programu je třeba upravit dva řádky ve zdrojovém kódu jádra. Dál se nainstaluje knihovna [libcap](#) [6] a zkompiluje zdrojový kód programu. Je třeba vytvořit skupinu, která bude mít možnost operací v reálném čase, třeba `rtcap`. V některých distribucích je už vytvořena skupina `audio`, takže se může použít tato. Jako skupina vlastníka programu se nastaví tato skupina. Pro jádra řady 2.6 je tu [Realtime Linux Security Module](#) [7]. Pro jeho kompilaci je třeba mít jádro nakonfigurované takto:

```
CONFIG_MODULES=y
CONFIG_SECURITY=y
CONFIG_SECURITY_CAPABILITIES=m
CONFIG_SECURITY_SELINUX=y
```

Volba `CONFIG_SECURITY_CAPABILITIES` musí být nastavena jako modul. Poté stačí provést `make` a jako root pak `make install` a `depmod`. Vytvořil se modul s názvem `realtime`. Nyní můžete vytvořit skupinu, která bude mít práva pro operace v reálném čase. V některých distribucích je už vytvořena skupina `audio`. Do paměti se zavede modul s parametrem `gid`, který určuje identifikátor skupiny, například: `modprobe realtime gid=29`.

Více informací najdete v [JACK FAQ](#) [8] a na [Low Latency Resource Page](#) [9]. Jack se ovládá z příkazové řádky nebo pomocí frontendu [QJackctl](#) [10].



*QJackctl*

## LASH

Při práci na nějakém hudebním projektu máte asi otevřeno více programů a každý z nich nějak nastavený. Jejich nastavení můžete hromadně uložit pomocí [LASH](#) [11]. Projekt se dříve jmenoval LADCCA a názvy příkazů tak stále začínají. Před spuštěním programů je třeba spustit démona, ke kterému se aplikace budou připojovat. Démon potřebuje ke svému spuštění Jack. Poté můžete pracovat s ostatními aplikacemi. Pro uložení nastavení aplikací spusťte program `ladcca_control`. Ten ukládá a obnovuje stav klientských aplikací. Pro obnovení práce LADCCA démon spustí všechny aplikace a nastaví jejich parametry. Stinnou stránkou je, že LASH zatím podporuje jen málo programů. Některé nativně, pro jiné je třeba stáhnout patch.

## LADSPA

[LADSPA](#) [12] je rozhraní pro zvukové pluginy. LADSPA pluginy mohou být spuštěny v mnoha aplikacích. Můžeme je použít k efektům na uložených zvukových souborech, ale také v reálném čase. LADSPA pluginy můžeme aplikovat i na zvuk ve spojení s Jackem pomocí aplikace [Jack Rack](#) [13].

## Sekvencery

Pro práci s MIDI se používají MIDI sekvencery. Ty vysílají MIDI události do ostatních zařízení, která na ně reagují zvukem, změnou parametrů zvuku nebo nějakou jinou akcí. Abychom dostali jako odpověď na MIDI události nějaký zvuk, je potřeba syntezátor. Zvukové karty často syntezátor mají. Ukážeme si použití softwarového syntezátoru.

V oblasti sekvencerů jsou asi dva nejvýznamnější – **MusE** [14] a **Rosegarden** [15]. Oba programy jsou nejnovější verze – Rosegarden 1.0 a MusE 0.7.3. MusE prodělala oproti své původní verzi dost výrazné změny k lepšímu. Oba programy jsou napsané s využitím knihovny Qt, mají i podobné vlastnosti.

## Rosegarden

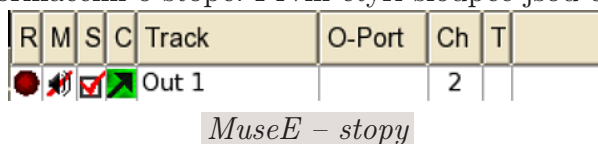
Rosegarden může spolupracovat s Jackem (není to ale podmínkou) a také podporuje rozhraní DSSI pro syntezátorové pluginy. Pokud chcete používat DSSI pluginy, budete si muset Rosegarden zkompileovat ze zdrojových kódů. Pro použití DSSI syntezátorů je potřeba mít spuštěný Jack. Editor notového partu nabízí víc, než je zvykem u sekvencerových programů. Do partu můžete vložit velké množství hudebních značek (například posun o oktávu níž nebo výš), můžete si vybrat mezi různými druhy not. Noty v partu si můžete nechat zarovnat na "hezčí" hodnoty (kvantizace) bez ovlivnění skladby. U složitějších se můžete setkat s problémy při tisku přímo z Rosegarden. Skladby můžete exportovat do formátů MIDI, Mup, MusicXML, Lilypond nebo jako Score file pro CSound.

## MusE

Řekl bych, že MusE má o něco intuitivnější ovládání než Rosegarden. Má vlastní rozhraní pro syntezátorové pluginy M.E.S.S, může mít neomezený počet vstupů a výstupů. Celá aplikace má také vlastní vnitřní routování vstupů a výstupů mezi stopami. MusE si můžete zkompileovat ze zdrojových kódů. Pro běžné použití nebo počáteční experimenty jsou asi nejlepší konfigurační parametry `--enable-fluidsynth --enable-ladcca`. Při zapnutí první volby se přeloží i softwarový syntetizér Fluidsynth, druhá volba přidá možnost použití LADSPA efektů na audio stopy.

## Pracovní prostředí

Podíváme se na to, co můžeme s MusE provádět. Jack musí být spuštěný před MusE. Po spuštění uvidíte hlavní okno a v něm jednu stopu. Nahoře je menu a panel nástrojů, po pravé straně je panel *Track Info* s informacemi a nastaveními stopy, který můžete skrýt klepnutím na tlačítko v levém dolním rohu okna. Uprostřed je seznam stop a vpravo pak okno, kde je časová osa a obdélníčky se skupinkami not. Měřítka časové osy můžete změnit tažením posuvníku v pravé dolní části okna. Podíváme se na řádek s informacemi o stopě. První čtyři sloupce jsou označeny R, M, S a C.



- *R* (ARM) – pokud je aktivní, stopa je označena pro nahrávání.
- *M* (Mute) – stopa bude ztlumena.
- *S* (Solo) – stopa bude hrát jako jediná (může jich být označeno i víc, a mohou tak hrát společně).
- *C* – typ stopy. V MusE je sedm typů stop.
  - *MIDI stopa* – obsahuje midi data.
  - *stopa bicích* – stejná jako předchozí, ale používá jiný editor.
  - *zvuková stopa* – obsahuje zvuk, zvukové soubory nebo nahraný zvuk.
  - *výstupní stopa* – vytvoří výstupní port pro Jack.
  - *vstupní stopa* – vytvoří vstupní port pro Jack.

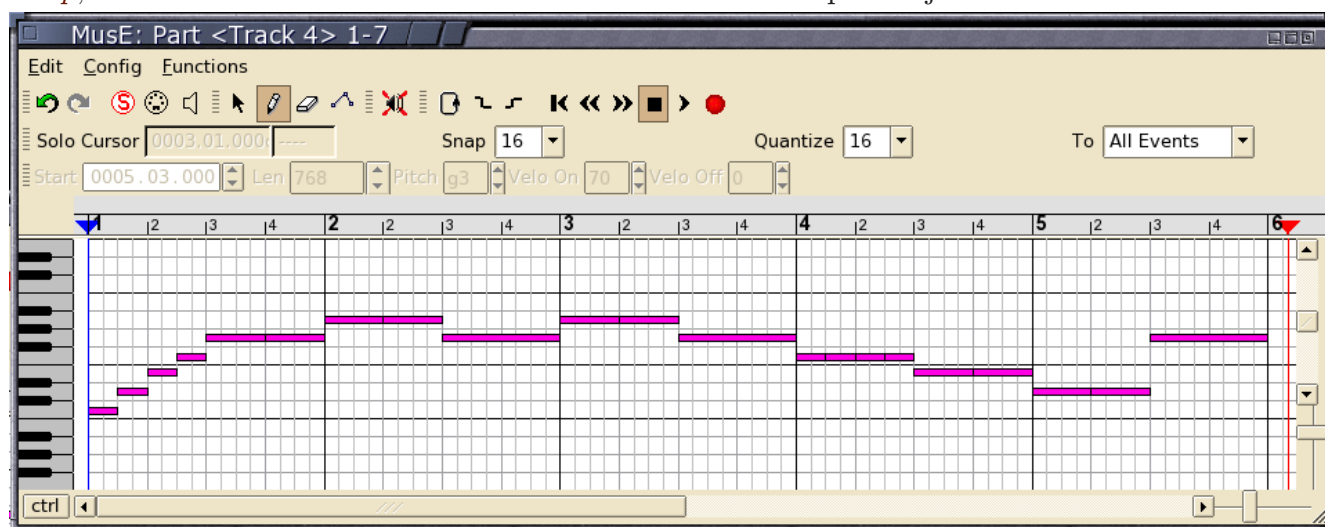
- *skupina* – slouží k seskupení několika stop, na ně pak je možné aplikovat najednou efekty nebo změnit jejich hlasitost.
- *Aux send* využijete, pokud chcete použít stejný efekt na více stop, ale na každou jinou měrou. Nemusíte na každou stopu používat stejný efekt jen s jiným nastavením mixu, ušetříte tak výkon procesoru.
- *syntezátorová stopa* vznikne vždy, když do projektu přidáte softwarový syntezátor. Je to zvuková stopa, její výstup se vnitřním routováním přeměruje do výstupní stopy.

Další sloupec (*Track*) zobrazuje jméno stopy. Tu můžete přejmenovat dvojklikem. *O-Port* je výstupní MIDI port, kam budou data odesílána. Konfiguraci portů si ukážeme o něco později. *Ch* značí číslo kanálu na vybraném portu, kanálů je 16. Kanál přenáší data pro jeden nástroj. Pokud chcete, aby hrálo víc nástrojů najednou, je potřeba každému přiřadit jiný kanál.

## Základy práce

Vytvoříme si MIDI stopu. Klepněte pravým tlačítkem myši na volné místo v seznamu stop a v menu vyberte *Add Midi Track*. Objeví se nová stopa. Na panelu nástrojů klepněte na ikonku tužky a ve stopě klepněte a táhněte. Vytvoří se „obdélníček“, který symbolizuje skupinu not. Na něj klepněte pravým tlačítkem myši a zvolte *pianoroll*. Objeví se okno pianoroll, do kterého se zadávají jednotlivé noty.

Kreslí se nástrojem tužka (klávesa **D**). Pokud tento nástroj použijete na existující notu, můžete tak změnit její velikost. Pro vymazání not slouží nástroj guma (klávesa **R**). Nástroj ukazatel (klávesa **P**) slouží k označení více not. Ty pak můžete přesouvat v čase nebo přesunout výš či níž, nebo zkopírovat a vložit na jiné místo. Kurzor se zachytává do mřížky. Její rozlišení můžete změnit pomocí roletky *Snap*, kde si zvolíte délku intervalu rozlišení. Teď můžete zapsat nějakou melodii.

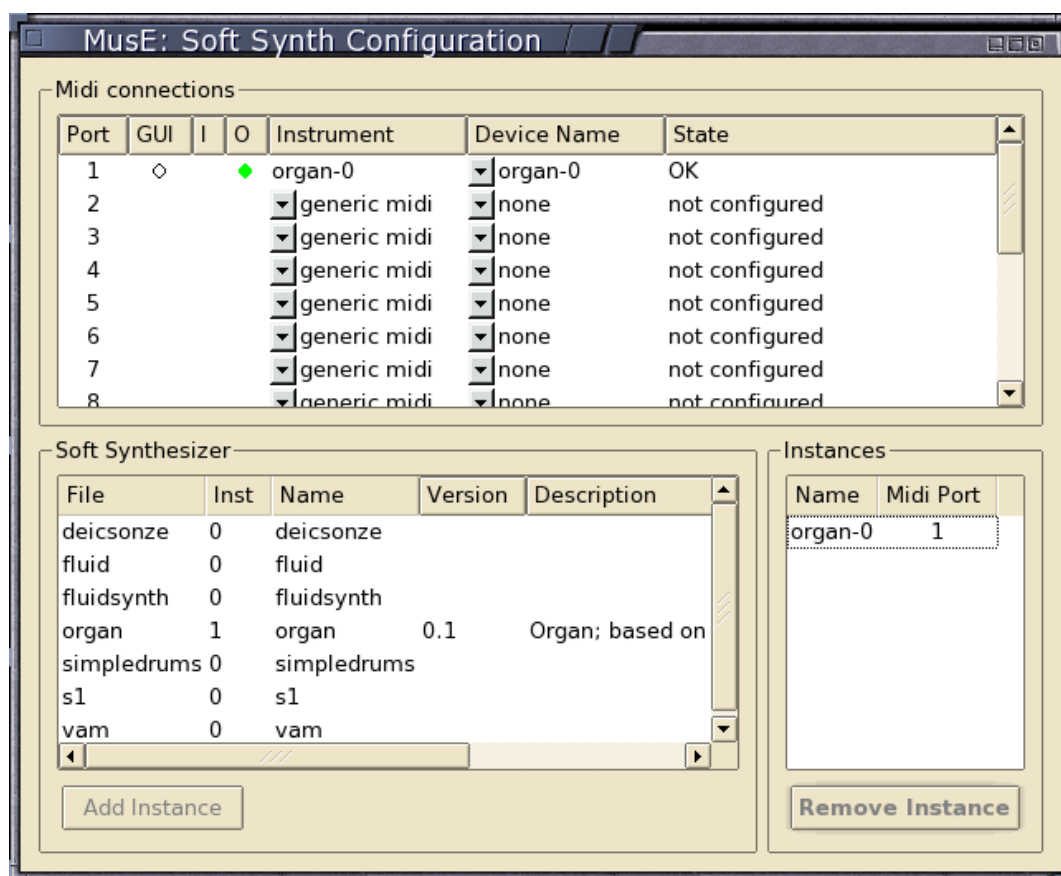


MusE – pianoroll

Každá nota má také svoji sílu stisku (velocity). Tu můžete editovat v oblasti kontrolerů, kterou zobrazíte kliknutím na tlačítko **Ctrl** v levém dolním rohu okna pianoroll. V tomto okně můžete používat buď nástroj tužka nebo nástroj čára (ve výchozím nastavení nemá přiřazenou klávesovou zkratku). Kromě velocity můžete v tomto okně editovat i hodnoty jiných kontrolerů, které lze vybrat z nabídky, objevující se po stisknutí tlačítka *Sel*. Po zapsání melodie si vytvoříme instanci softwarového syntezátoru. Můžeme použít buď jeden z interních pluginů nebo jakýkoliv softwarový syntezátor spolupracující s Jackem.

Z menu *Settings* zvolte *Midi ports/Soft synth*. Objeví se další okno. V jeho horní části jsou jednotlivé porty programu MusE svázané se syntetizéry. V dolní části vlevo je výběr pluginů, které můžete použít, vlevo pak, existující instance pluginů. V části Soft Synthesizer vyberte *organ* a klepněte na *Add instance*, plugin se objeví v části Instances. V horním okně jeho instanci přiřadíme na port 1 –

v prvním řádku vyberte v roletce *Devices* jméno tohoto pluginu (nejspíš bude organ-0). Ujistěte se, že stopa, kde je zapsaná melodie, má nastavený právě port 1. Všimněte si, že se vytvořila stopa pro zvukový výstup přidaného syntezátoru.



MusE – midi ports

Jako poslední krok je potřeba spojit výstup syntezátoru s výstupní stopou. Otevřete okno mixeru (*View – Mixer* nebo klávesa **F10**). V okně mixeru pak zvolte *View – Routing*.

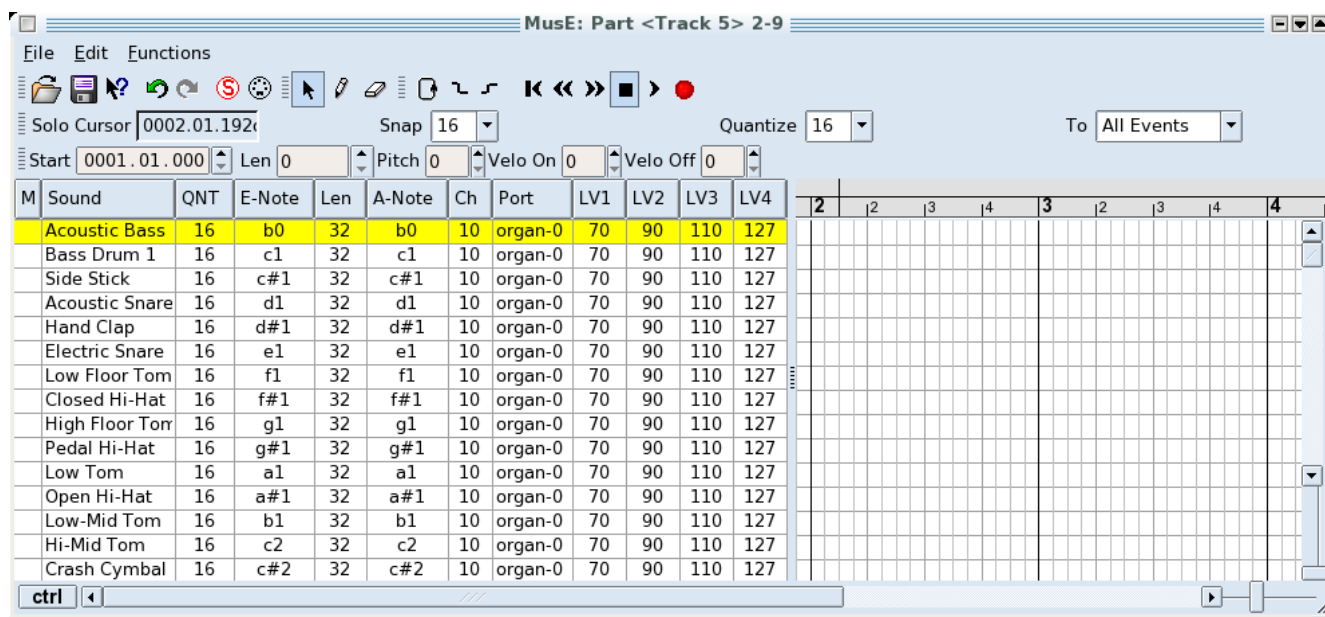
V okně *Routing* v části *Source* klepněte na organ-0, v části *Destination* na Out 1 a poté tlačítko *Connect*. V dolní části se objeví toto spojení. Nyní si už můžete přehrát, co jste vytvořili. Jestli neslyšíte žádný zvuk, podívejte se do okna *Routing*, jestli existuje spojení mezi stopou *Out 1* a výstupem vaší karty

## Ovládání

Pro editaci bicího partu se používá trochu jiný editor než pro stopu s melodií. Noty pro bicí se opět zapisují do mřížky. Vertikální pozice určuje, který zvuk bude hrát, a horizontální pozice určuje čas. Nota každé výšky má přiřazený jiný zvuk bicích. V levé části okna jsou parametry každého bicího nástroje.

- *M* – (Mute) umlčí nástroj
- *Sound* – název nástroje
- *QNT* – kvantizace, zarovnání hodnot not
- *E-note* – nota, která bude zahrána na stopě
- *Len* – délka noty
- *A-note* – nota, která se vyše do syntetizéru, když byla stisknuta klávesa určená v poli *E-note*
- *Ch* – (Channel) kanál, na kterém bude nota přehrána
- *Port* – port, na kterém bude nota přehrána
- *LV1* – při stisknutí **Ctrl+Shift**+levé tlačítko myši bude mít nota tuto hodnotu velocity

- *LV2* – při stisknutí **Shift**+levé tlačítko myši bude mít nota tuto hodnotu velocity
- *LV3* – při stisknutí **Ctrl**+levé tlačítko myši bude mít nota tuto hodnotu velocity
- *LV4* – při stisknutí levého tlačítka myši bude mít nota tuto hodnotu velocity



### MusE – drums

K čemu vlastně jsou pole A-note a E-note? Různé syntetizéry mohou mít stejné zvuky umístěné na notách jiné výšky. Aby nebylo nutné při použití jiného syntetizéru (jiné klávesy, jiný sampler, jiný program,...) měnit notový zápis, stačí vyměnit bicí mapu. Sekvencer pak bude podle bicí mapy vysílat do syntetizéru správné noty. Sekvencer podle noty určité výšky (určitého nástroje, každá výška tónu znamená jiný zvuk bicích) zapsané v partu (A-note) vyšle notu určité výšky do syntetizéru (E-note). Tato výška hraje na syntetizéru stejný zvuk. Jsou tu kvůli tomu, že různé syntetizéry mohou mít stejně znějící nástroje umístěny na jiných notách.

Za zmínku stojí také to, že pro bicí se často používá kanál 10, na kterém jsou zvuky bicích. Volbou nástroje na tomto kanálu pak měníte použitou sadu bicích.

Stopa typu *Group* vám umožní sdružit výstupy několika audiostop nebo stop s výstupem softwarového syntetizéru. Na výstupy několika stop tak můžete použít jeden efekt místo toho, abyste ho použili na každou zvlášť. Ušetříte tak výkon procesoru. Výhodou je také regulace hlasitosti všech těchto stop najednou. Výstupy audiostop můžete přeměrovat do skupiny pomocí dialogu *Routing*.

Se stopou typu *Aux* můžete na víc audiostop použít jeden efekt se stejnými nastaveními, ale s jinou měrou efektu. Efekt tak může být na každé stopě jinak hlasitý. Do této stopy se nepřivádí vstup routováním. V Mixeru (klávesa F10) a v bočním panelu nastavení stopy se u každé zvukové stopy objeví otočný regulátor pro každou stopu Aux (přímo na nastavením hlasitosti).

## Test

Vyzkoušíme si to. Otevřeme nějaký projekt, ve kterém jsou dvě nebo více zvukových stop a přidáme jednu stopu typu Aux. Na novou stopu přidáme efekt. Klepněte pravým tlačítkem myši do oblasti, kde se vkládají efekty, a vyberte třeba dozvuk (plugin freeverb). V seznamu efektů přibyla nová položka. Klepněte na ni pravým tlačítkem myši a zvolte *show gui*. Objeví se dialog s nastaveními pluginu. Nastavte *Wet Mix* na 0 (maximum efektu), aby byl dozvuk dobře slyšet. U zvukových stop, na které chcete použít tento efekt, nastavte otočným regulátorem hlasitost výstupu, který chcete do Aux stopy pustit. U jedné třeba na 0 dB u druhé na -3 dB. S hodnotami vyššími než 0 dB opatrně, mohou způsobit zkreslení zvuku (clipping).

Nyní si můžete skladbu přehrát. Podívejme se ještě na syntetizéry, které přicházejí společně s MusE. Syntetizér je zařízení (nebo program), které vytváří (nebo jen vydává) zvuk. Zvuk a jeho parametry

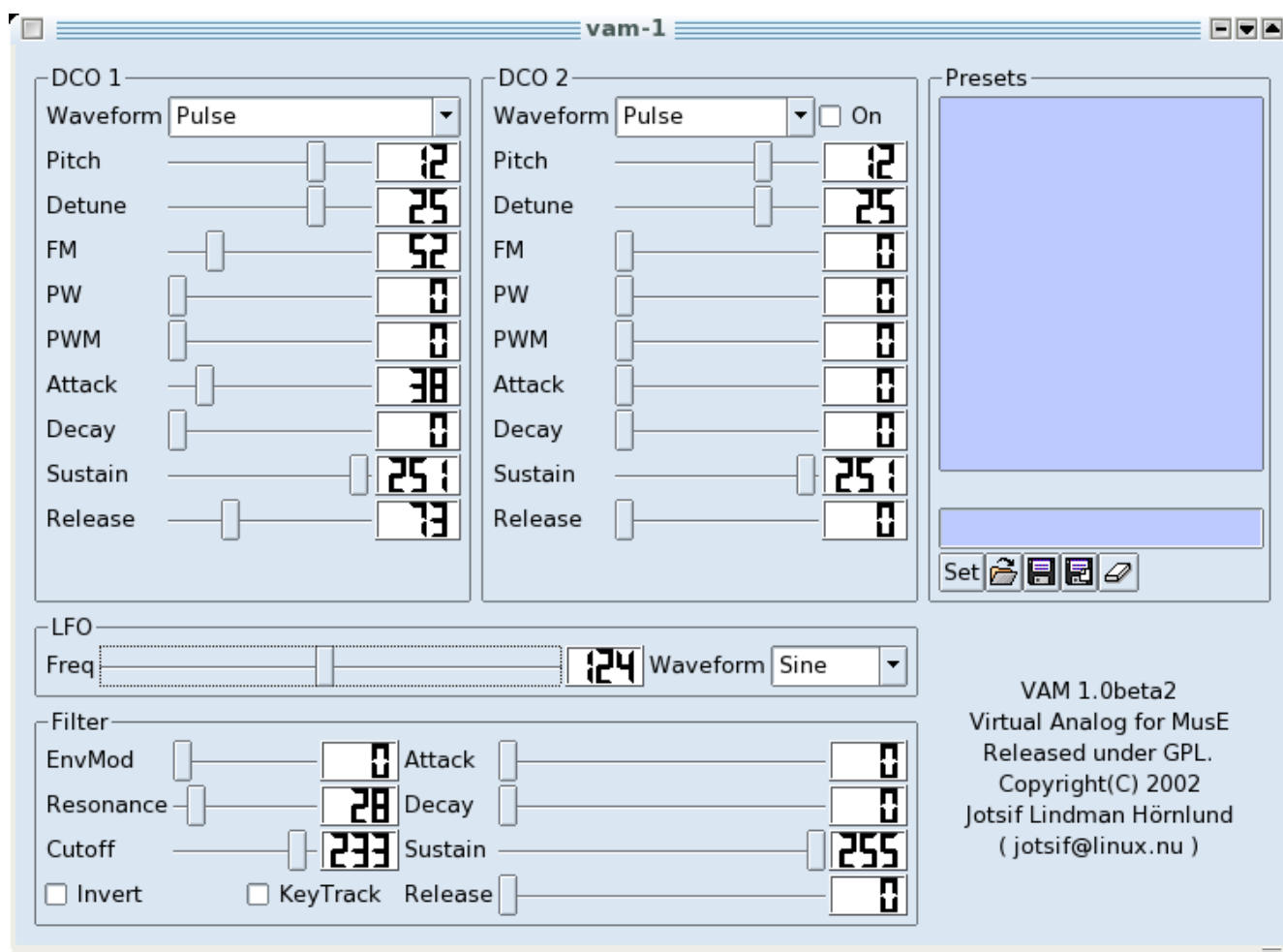
mohou být řízeny pomocí událostí MIDI, například zapnutí a vypnutí noty nebo změna parametrů zvuku. MusE má své vlastní programové rozhraní pro syntetizérové pluginy M.E.S.S.

## Vestavěné syntetizéry

- *DeicsOnze* – syntetizér s bohatou bankou zvuků. U zvuků je trochu znát, že jsou umělé. Může se vám hodit, pokud ovšem nedáte přednost třeba SoundFontům. Vypadá to, že jedna instance syntetizéru je schopná hrát zvuk pouze jednoho nástroje na všech kanálech. Je to škoda, bylo by dobré, kdyby bylo možné na každém kanále vybrat jiný nástroj. Snad se to v příští verzilepší.
- *fluid* – syntetizér, který přehrává zvuky ze SoundFontu. Umožňuje nahrát a používat pouze jednu banku. SoundFont je banka zvuků, které můžete použít společně s MIDI. Některé zvukové karty (SB AWE 32, AWE 64, SB Live!, SB Audigy) umí SoundFonty přehrávat i za pomoci syntetizéru karty. Do paměti u nich můžete zavést SoundFont příkazem `asfxload`. Na druhou stranu, takto přehraný zvuk v sobě bude mít šum ze zvukové karty (který však může být zanedbatelný).
- *fluidsynth* – slouží, stejně jako *fluid*, k používání SoundFontů. Nabízí víc možností než *fluid*. Jedna instance syntetizéru může poskytovat zvuky z více SoundFontů, zvuky z nich může rozdělit mezi jednotlivé kanály, na zvuk může aplikovat efekty dozvuk (Reverb) a Chorus, navíc má regulaci hlasitosti. Přijde mi, že je lepší interní efekty nepoužívat a využít až efekty LADSPA v syntetizéru.
- *organ* – syntetizér zvuku varhan. Zní dobře, parametry zvuku můžete upravovat. Na stránkách MusE je i [zvuková ukázka](#) [16].
- *s1* – velmi jednoduchý syntetizér bez grafického rozhraní, který generuje sinusové vlny. V praxi jej asi často nevyužijete.
- *vam* – Virtual Analog for MusE – další jednoduchý analogový syntetizér. Má grafické rozhraní, umožňuje použít dva generátory zvukových vln, každý s vlastním nastavením. Myslím, že zvuky vytvořené s jeho pomocí použijete zřídkakdy, možná jako ozvláštnění nebo jako zajímavý efekt.
- *Simpledrums* – je sampler (přehrává zvukové vzorky podle toho, jaké MIDI události přicházejí). Jak už vyplývá z názvu, jeho hlavní použití bude pro přehrávání zvuků bicích. Může používat až patnáct vzorků zvuku ve formátu .wav. Jednotlivé zvuky jsou rozmístěny od C1 po F2. Využijete ho, pokud nemáte zvuky bicích v SoundFontu, nebo pokud nepoužíváte bicí program, jako třeba Hydrogen.

Když si kompilujete MusE ze zdrojových kódů, přeloží se všechny syntetizéry, kromě *fluidsynth*, který je nutné při konfiguraci povolit přepínačem `--enable-fluidsynth`.

Na internetu se také objevují diskuse, zda je lepší používat *fluid* nebo *fluidsynth*. Názory se různí, někomu zní lépe ten první, někomu ten druhý. Použijte ten, který vám bude lépe znít. Jejich použití závisí také na tom, jak chcete zvuk zpracovávat. Syntetizéry v MusE mají jen jeden výstup. Pokud používáte *fluidsynth* a spravujete víc nástrojů, nemůžete použít na některý z nich jiný efekt, než na ostatní. Pokud budete toto vyžadovat, použijte radši jednodušší *fluid* a pro každý SoundFont vytvořte jednu instanci.



*Virtual analog for MusE*

## Odkazy

- [1] <http://audacity.sourceforge.net>
- [2] <http://rezound.sourceforge.net/>
- [3] <http://ardour.org/>
- [4] <http://www.zip.com.au/~akpm/linux/schedlat.html#downloads>
- [5] <http://www.tml.hut.fi/~tilmonen/givertcap/>
- [6] <http://ftp.kernel.org/pub/linux/libs/security/linux-privs/kernel-2.4>
- [7] <http://sourceforge.net/projects/realtime-lsm/>
- [8] <http://jackit.sourceforge.net/docs/faq.php>
- [9] <http://www.linuxdj.com/audio/lad/resourceslatency.php>
- [10] <http://qjackctl.sourceforge.net/>
- [11] <http://www.lash-audio-session-handler.org>
- [12] <http://www.ldaspa.org>
- [13] <http://arb.bash.sh/~rah/software/jack-rack/>
- [14] <http://www.muse-sequencer.org/>
- [15] <http://www.rosegardenmusic.com>
- [16] [http://www.muse-sequencer.org/music/organ\\_demo.ogg](http://www.muse-sequencer.org/music/organ_demo.ogg)

\*\*\*

## Reklamní nabídky [www.praceabc.cz](http://www.praceabc.cz)

Pozice: Programátor C/C++/Linux – i pro absolventy! [2057]  
 Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 13.07.2005

Pozice: Junior programátor C++/Unix – na zaškolení! [2014]  
 Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 13.07.2005

Pozice: Verification and Validation Engineer [2001]  
 Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 13.07.2005

Pozice: Podpora unixových aplikací (časté služební cesty do zahraničí) [1971]  
 Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 08.07.2005



# PEAR

Pavel Szalbot ml.

---

*Nebojte se, jste tu správně. Nebude řeč o ovoci, ale o frameworku pro jazyk oblíbený mezi vývojáři webových aplikací. Tím jazykem je PHP a framework se jmenuje, jak jistě správně tušíte, PEAR.*

---

## Úvod – framework

Webové aplikace se od ostatních liší v mnoha směrech. Jedním z nich je relativně časté opakování stejných problémů. Spousta webových aplikací například musí spravovat uživatelské účty, definuje způsob zobrazování a ukládání obsahu, pracuje s databázemi, zpracovává různé datové formáty jako např. XML atd. Programovací jazyky nabízejí často jen parciální řešení podobných problémů, a tak je nasnadě navrhnout vlastní knihovnu, která poskytne potřebnou funkcionalitu prostřednictvím dobře definovaného rozhraní. Knihovna může žít vlastním životem a s novou verzí zpravidla není složité upgradovat i aplikaci.

Samotný návrh knihovny ovšem není jen otázkou tvorby rozhraní a implementace příslušných algoritmů ve zvoleném jazyce. Je potřeba udržet konzistentní styl psaní kódu a vytvořit jednoduchou a úplnou dokumentaci, aby knihovnu mohli používat i další programátoři. Nakonec všichni skončíme u toho, že definujeme vlastní framework, což není vůbec špatné, ale někdo už to mohl udělat dříve a taky lépe.

## PEAR

Tento seriál se bude věnovat frameworku a distribučnímu systému knihoven a rozšíření pro aplikace psané v jazyce PHP nazvanému PEAR. Začneme seznámením s frameworkem a ukážeme si, jak jej nainstalovat a používat. Další díly budou věnovány konkrétním balíčkům, jak se aplikace a knihovny obsažené v PEARu nazývají.

**PEAR** [1] (PHP Extension and Application Repository) vznikl z diskuse na PHP Developers' Meeting v roce 2000 a od té doby urazil docela dlouhou cestu. V současnosti archiv balíčků obsahuje na 250 open source knihoven a rozšíření roztříděných tématicky do několika kategorií (např. databáze, souborový systém, HTTP, matematika, text, xml, web services apod.). Archiv doplňuje sofistikovaný distribuční systém, kvalitní dokumentace samotného frameworku, návody, mailing-listy a v neposlední řadě IRC kanál sloužící jak uživatelům, tak vývojářům nových projektů, které usilují o začlenění do repositáře.

PEAR navíc definuje čtyři zvláštní skupiny balíčků. Jednou z nich je repositář PECL (PHP Extension Community Library), který obsahuje rozšíření pro samotné PHP. PECL má vlastní webové stránky a definuje vlastní styl psaní kódu, ovšem distribuční systém včetně struktury balíčků sdílí se svojí sestrou, jak PEAR nazývá (ženský rod je zřejmě používán kvůli věnování frameworku dceři autora). Mimo PECL jsou obzvláště důležité tzv. PHP Foundation Classes (PFC), což je skupina balíčků, které standardní distribuce PHP instalují spolu s nástroji pro práci s archivem. PFC tvoří jen stabilní balíčky, které nejsou vázány na konkrétní systém a jsou díky standardizovanému API do budoucna snadno rozšiřitelné. Pokud budou distribuce PHP i nadále instalovat balíčky z PEARu, budou to ty z PFC.

## Balíčky

Ještě než začneme s frameworkem pracovat, podíváme se na proceduru, kterou prošel každý balíček, a na jeho vlastnosti, které z ní vyplynuly. Bude nám pak jasné, jaké konvence balíček dodržuje,

jaký kód budeme používat a co můžeme očekávat od jeho autora. Pokud se rozhodnete rozšířit PEAR o vlastní knihovnu, doporučuji vám přečíst si kompletní dokumentaci na webových stránkách projektu. Následující text je vhodný spíše pro uživatele (i ti by se však měli s dokumentací seznámit).

Na začátku vývoje balíčku stojí otázka, k čemu bude knihovna nebo rozšíření sloužit. Podle toho jej autor musí správně pojmenovat a zařadit do kategorie. Svůj návrh pošle do mailové konference a počká na vyjádření, zda neexistuje podobný či stejný balíček a zda vůbec bude přínosem. Nemělo by se tedy stát, že by archiv obsahoval dva balíčky, které řeší stejný problém, a tak se netrapte tím, že jste našli jen jeden, který splňuje vaše představy. Pokud se vám balíček hodí, nezapomeňte zkontrolovat jeho závislosti na ostatních (na ty vás případně upozorní i instalační program) a zjistěte, s jakou licencí je nabízen k použití. Autor se vytvořením zavazuje poskytnout podporu, opravovat nahlášené chyby a také by měl rozšiřovat kód o nové funkce, které navrhnou nebo jimi rovnou přispějí ostatní vývojáři. Může se stát, že objevíte chybu a pak byste ji určitě měli nahlásit správci balíčku. K tomu účelu slouží formulář umístěný v rámci distribučního systému.

## Konvence

Jedním z požadavků kladených na každý balíček je dodržení konvence formátování řídicích struktur, názvů proměnných, tříd atd. Zkrátka musí zdrojový kód odpovídat stylu, jehož popis je k dispozici v dokumentaci PEARu. Styl psaní kódu se docela podobá [doporučení \[2\]](#) Sunu pro programovací jazyk Java a převratné novinky v něm určitě nenajdete. Před začátkem vývoje aplikace používající balíčky z PEARu se ovšem rozhodně vyplatí zvážit, zda vlastní kód jejich použitím neznepřehledníte. Pokud se nevyhnete tomu, že budete muset provést vlastní refaktorizaci kódu balíčku, zkuste raději vytvořit wrapper nad rozhraním knihovny/rozšíření. Starší verze se totiž dají díky instalačním nástrojům poměrně snadno upgradovat a balíček mohou používat i jiné aplikace běžící na serveru.

Autor je dále povinen zveřejnit kompletní dokumentaci zdrojových kódů a to buď ve formátu Docbook XML nebo jako „prostý text“. Najít ji můžete jak na webu, tak také v adresáři, kde je PEAR nainstalován. Dokumentace, která je součástí instalace, zpravidla představuje jen návod jak balíček nainstalovat a začít používat. V distribučním systému by pro každý balíček měla být dostupná podstránka s dokumentací celého API.

Vývoj balíčku probíhá pod dohledem ostatních vývojářů, kteří také nakonec rozhodují o jeho zařazení do archivu. Jakmile je autor spokojen se současným stavem balíčku, může jej zařadit do hlasovacího procesu. Během hlasování se projekt až na výjimečné případy nesmí měnit a první verze je tak před vydáním dána na milost ostatním. Nutný počet hlasů pro přijetí je relativně malý (stačí nasbírat 5 kladných hlasů (+1 hlas získá projekt za souhlas, -1 za nesouhlas, 0 za nerozhodnutého vývojáře) za 7 dní (14 pokud se během 7 dní hlasování nezúčastní alespoň pět vývojářů)), nicméně projekt je od svých prvopočátků sledován a výsledná kvalita by měla být zřejmá už před hlasováním (pokud ovšem autor nechce riskovat).

## Distribuční systém

Distribuční systém balíčků nabízí mimo funkcí pro získávání zdrojových kódů hlavně jednotné webové rozhraní pro každý balíček, ve kterém najdete informace o jeho stavu, můžete si prohlédnout dokumentaci, zjistit informace o týmu vývojářů, kteří jej mají na starosti, nebo nahlásit nalezené chyby. Pracovat s distribučním systémem se dá pracovat buď přes webové rozhraní, nebo přes správce balíčků, který je obvykle nainstalován spolu s PHP. Obě prostředí ovšem obsahují funkce, které nenabízí to druhé (z webu např. balíčky nenainstalujete, ze shellu zase nenahlásíte nalezené chyby atd.), ale základní funkce (získání a vyhledání balíčku a informací o něm) mají společné.

## Instalace

### Instalace pomocí správce balíčků

Pokud máte nainstalováno PHP ve verzi 4.3.0 a vyšší, máte standardně k dispozici i správce balíčků, ledaže jste při překladu použili volbu `without-pear`. Jestliže správce balíčků nemáte, můžete ho nainstalovat ručně. Stačí v shellu zadat `lynx -source http://go-pear.org/ | php` a řídit se instrukcemi. Jakmile je správce nainstalován, můžete s ním začít pracovat. Seznam argumentů spouštěcího příkazu je výstupem skriptu bez parametrů: `pear`.

Zvědavé povahy si určitě všimnou, že samotný správce balíčků je PHP aplikace a dokonce je součástí balíčku se jménem PEAR. Někdo si možná prohlédl kategorii „PEAR“, někdo zase nahlédl do zdrojových kódů a zjistili jste tak, že existují i grafické podoby správce. V současnosti to jsou další dvě verze – jedna webová a druhá postavená na PHP-GTK (balíčky, které PHP-GTK používají, patří do zvláštní skupiny jako např. PFC) – obě zatím jen ve stavu betaverze. V dokumentaci o nich proto není žádná zmínka, ale vezměme webovou variantu a demonstrujme si na ní práci s CLI nástroji.

Seznam dostupných balíčků lze zobrazit příkazem: `pear remote-list`. Na terminálu uvidíte názvy všech balíčků a čísla jejich verzí. Balíček, který chceme nainstalovat, ovšem není zobrazen. Jak je to možné? Chování správce totiž ovlivňují konfigurační **proměnné** [3] a jednou z nich je i `preferred_state`, která určuje, jakou verzi (alpha, beta, devel, stable, snapshot) chceme přednostně instalovat. Určitě šikovná volba, nicméně ovlivní i seznam balíčků, který se zobrazí (vyhledávání pomocí `pear search` neovlivňuje). Dostupná verze PEAR\_Frontend\_Web je pouze beta, tudíž změním nastavení správce: `pear config-set preferred_state beta`. *Po ukončení experimentů si vraťte nastavení zpět na původní (nejspíš stable)*. Teď už balíček v seznamu vidíte a hlavně je možné jej konečně nainstalovat: `pear install PEAR_Frontend_Web`. Pracujete-li s čistým PEARem, instalátor vám ohlásí, že PEAR\_Frontend\_Web potřebuje ještě balíčky `Net_UserAgent_Detect` a `HTML_Template_IT`. Nainstalujeme tedy všechny najednou:

```
pear install PEAR_Frontend_Web NET_UserAgent_Detect HTML_Template_IT
```

Abyste mohli frontend vyzkoušet, zkopírujte soubor `WebInstaller.php`, který se nachází v adresáři `PHP_LIB_DIR/php/PEAR/` (`PHP_LIB_DIR` nahraďte adresářem, kde jste PHP řekli, že se nachází knihovny [`/usr/local/lib/`]), do www adresáře webserveru a zkuste jej zobrazit v prohlížeči. Balíček odinstalujete triviálním `pear uninstall PEAR_Frontend_Web`.

### Instalace z tarballů

Existují i balíčky, které framework používají, ale nejsou zařazeny v repositáři. Takové balíčky jsou zpravidla dostupné jako tarbally, přičemž jejich obsah je vytvořen tak, aby je správce dokázal nainstalovat. Instalace z tarballu je umožněna díky souboru `package.xml`, jehož formát je vysvětlen v dokumentaci. Při instalaci zadáte místo názvu balíčku cestu k souboru `package.xml`.

### Instalace v prostředí webhostingu

K webhostingu zpravidla nedostanete přístup k shellu a instalaci pak musíte provést ručně. Naštěstí to není nic obtížného a většina z vás by určitě přišla, jak na to. Stáhněte si zdrojové kódy vybraného balíčku a přes FTP je uploadujte např. do adresáře `/include` ve vašem webovém prostoru. Ve skriptech nezapomeňte pomocí funkce `ini_set` nastavit proměnnou `include_path` na `/include`.

Tento způsob instalace – pravděpodobně nejčastější, jaký budete provádět – s sebou nese malý zádrhel. Je třeba si ověřit závislosti a „nainstalovat“ také balíčky, na kterých vámi zvolený závisí.

Samotný balíček PEAR definuje pro PHP4 mj. často používané speciální funkce, které fungují jako destruktory.

## Závěr

Tento díl vás měl s frameworkem seznámit a poskytnout vám základy pro práci s ním. Netvrdím, že jsem vyčerpал všechna zajímavá témata, ale to jsem ani nechtěl. Výborným zdrojem nejen pro začátečníky je určitě [online dokumentace](#) [4], která není příliš rozsáhlá, ale najdete v ní všechno, co budete potřebovat. Pro budoucí přispěvatele je její znalost nutností.

## Úvod – problematika

Seznámení s jazykem PHP probíhá různě. Pro někoho může práce s ním být prvním programátorským počinem, pro jiné zase jen další kapitolou. Informace o jazyce ale obě skupiny zpravidla získávají z odborné literatury, případně z online dokumentace. Dokud vyvíjíme relativně jednoduché aplikace, všechno se zdá být v pořádku, PHP plně vyhovuje a vlastně bychom si vystačili i s tím, co PHP nabízí samo o sobě. Jenže snad každý, kdo se zabývá psaním dynamických webových aplikací, bude postaven před úkol vytvořit „něco většího“.

Najednou je třeba kontrolovat efektivitu a kvalitu kódu daleko více. Literatura adepty v tomto směru zpravidla nepoučí, přestože příklad složitějších aplikací hojně obsahuje. Jak tedy zjistit efektivitu kódu, když něco nefunguje tak, jak má? Oblíbenou možností je odhadnout, co by mohlo aplikaci brzdit (vždyť to často bývá způsobeno interakcí s databází), zkusit část kódu upravit a oslnit kolegy řešením. Nebo se trápit dále?

V tomto díle si představíme techniky, jak identifikovat „úzká hrdla“ v aplikacích psaných v PHP, a přestože je seriál věnován PEARu, zmíním se i o nástroji *ab*, který je dodáván spolu s webovým serverem Apache.

## Benchmark

Základním způsobem, jak testovat výkon různých částí kódu, je měřit čas, který byl zapotřebí k jejich vykonání. Zjistit to můžeme například voláním funkce `microtime()` před a po dokončení zvoleného úseku a následným odečtením získaných hodnot. Kód ovšem obvykle nebývá přímočarý, obsahuje různá větvení (smyčky, volání funkcí, podmínky), a dal by se tak v podstatě rozdělit na menší bloky či *sekce*, které by nás také mohly zajímat (problém by mohl být právě v nich).

Repositář PEARu obsahuje balíček s názvem Benchmark, který nám s analýzou kódu pomůže. Tvoří ho tři třídy, jmenovitě `Benchmark_Timer`, `Benchmark_Iterate` a `Benchmark_Profiler`, jejichž názvy napovídají, k čemu se dají použít.

### BenchmarkTimer

Použití třídy `Benchmark_Timer` si vysvětlíme na velmi jednoduchém příkladu.

```
<?php
require_once('Benchmark/Timer.php');

function my_sleep() {
    global $timer;
    $timer->setMarker('funkce my_sleep'); // značka
    sleep(1);
```

```

}

$timer = new Benchmark_Timer();
$timer->start(); // začínáme měřit
sleep(2);
my_sleep();
$timer->stop(); // konec měření
$timer->display();
?>

```

Na začátku jsme vytvořili objekt třídy `Benchmark_Timer` a posléze časovač spustili funkcí `start()`. Pokud chcete začít měřit hned po vytvoření objektu, předejte konstruktoru jako parametr hodnotu `TRUE`, destruktor pak měření zastaví (stejně jako funkce `stop()`) a zobrazí výsledky. Skript jsme nechali odpočinout (`sleep()`) a následně zavolali funkci `my_sleep()`. Ta používá metodu třídy `setMarker()`, která vytvoří „mezičas“ zobrazovaný ve výsledcích. Díky mezičasům můžeme měřit dobu výpočtu různých bloků. Výpis tohoto programu u mě vypadá takto (spuštěno v CLI verzi PHP):

```

-----
marker           time index           ex time           perct
-----
Start            1113649899.69254300  -                 0.00%
-----
znacka_1         1113649901.69398800  2.001445         66.64%
-----
funkce my_sleep  1113649901.69410400  0.000116         0.00%
-----
Stop             1113649902.69575700  1.001653         33.35%
-----
total           -                    3.003214         100.00%
-----

```

Jak vidíte, u každého mezičasu je uveden unixový čas průchodu, čas, který uplynul od posledního mezičasu a procentuální vyjádření náročnosti úseku. Třída bohužel neobsahuje metodu, která výsledky zobrazí seříděny podle jiného kritéria než podle času, nicméně zdrojový kód je přehledný a jednoduchý a neměli byste mít problém napsat vlastní funkci, která vám vlastní třídění umožní (použitý objektový model PHP 4 bez přístupových identifikátorů je v tomto případě výhodou).

## BenchmarkIterate

`Benchmark_Iterate` vznikla z třídy `Benchmark_Timer` děděním a najdete ji v souboru `Benchmark/Iterate.php`. Umožňuje nám oproti `Benchmark_Timer` navíc několikrát spustit a analyzovat vybranou funkci a to díky metodě `run()`. Jako parametr jí předáte počet volání, název funkce, která se má volat, a nakonec její parametry. Volání vypadá např. takto:

```
$timer->run(10, 'my_function', 3, 'parametr2', array(1, 2, 4));
```

Samozřejmě můžete volat také metody tříd. Druhý parametr pak může být buď `'objekt→funkce'` nebo `'třída::funkce'`. Výsledky zobrazíte stejně jako v rodičovské třídě funkcí `display()`.

## BenchmarkProfiler

Pro názornost si znovu ukážeme jednoduchý kus kódu:

```
<?php
require_once('Benchmark/Profiler.php');

function my_func() {
    global $timer;
    $timer->enterSection('my_func');
    sleep(1);
    $timer->leaveSection('my_func');
}

$timer = new Benchmark_Profiler(TRUE);
my_func();
?>
```

Tentokrát jsme při vytváření instance předali konstruktoru zmiňovanou hodnotu TRUE, takže jsme nemuseli volat funkce `start()`, `stop()` a `display()`, které `Benchmark_Profiler` definuje, přestože není odvozená od žádné z předchozích tříd. Funkce `enterSection($name)` a `leaveSection($name)` vytvoří v kódu pojmenovanou sekci, která je sledována, a čas v ní strávený je zobrazen ve výsledcích. Sekce mohou obsahovat další podsekcce, ty se však nesmí křížit (jako první musí být ukončena ta sekce, která byla otevřena jako poslední). Výsledky kromě časových údajů obsahují také počet iterací všech navštívených sekcí.

Výhodou balíčku `Benchmark` je především jednoduchost zmíněných tříd a možnost využití i v rámci webhostingu. Nabízené možnosti ovšem nejsou příliš široké. Chybí funkce jako zobrazení stromu volání nebo filtrování. Největší slabinou však bude omezenost použití, protože profilování aplikace vyžaduje zásahy do zdrojových kódů, což vám nemusí vždy vyhovovat. Podíváme se proto na balíček, který je součástí repozitáře PECL.

## APD

APD ([Advanced PHP Debugger](#) [5]) je rozšíření Zendu engine, které slouží k profilování aplikací napsaných v PHP. Poskytuje daleko více možností než třídy z balíčku `Benchmark` a dovolím si říct, že je to nejlepší open source nástroj svého druhu pro PHP. Pojdme si tedy říct, jak jej nainstalovat, a podívejme se, co všechno umí.

Balíček nejprve nainstalujte klasickým způsobem (`pear install apd`) a poté do `php.ini` přidejte následující řádky:

```
zend_extension = /cesta/k/apd/apd.so
apd.dumpdir = /kam/ukladat/vysledky/
```

Rozšíření tak bude nahráno do paměti a bude použitelné v každém skriptu. Do adresáře určeného volbou `apd.dumpdir` se budou ukládat výsledky profilování, a PHP (Apache) tam proto musí mít právo zapisovat. U webhostingu to obvykle není možné a vlastně nevím o žádném, který by APD měl nainstalováno, nicméně výsledky z vývojového prostředí by měly být násobkem těch z produkčního prostředí, takže informace o efektivitě jsou dobře použitelné nezávisle na výkonu stroje.

APD se používá velmi snadno. Do bodu, od kterého chcete začít profilovat, vložte volání funkce `apd_set_pprof_trace()`, čímž se APD v daném místě aktivuje. Výsledky se po ukončení skriptu

uloží do souboru `pprof.<pid>.<ext>` ve zvoleném adresáři (`apd.dumpdir`) a můžete je zobrazit programem `pprofp` (PHP skript), který byl nainstalován spolu s APD. Mimo `pprofp` je k dispozici i skript `pprof2calltree`, který převede výsledky z vybraného souboru do formátu používaného programem `KCacheGrind` [6], který určitě stojí za vyzkoušení.

Syntax skriptu `pprofp` vypadá následovně:

```
pprofp <přepínače> <soubor s profilem>
```

Volby pro třídění:

```
-a třídít dle názvů funkcí
-l třídít dle počtu volání
-r třídít dle času stráveného v tělech funkcí (real time)
-R třídít dle času stráveného v tělech funkcí (real time)
  (započítává se i čas volaných funkcí)
-s jako -r, ale dle systémového času (system time)
-S jako -R, ale dle systémového času (system time)
-u jako -r, ale dle uživatelského času (user time)
-U jako -R, ale dle uživatelského času (user time)
-v třídít dle průměrného času stráveného v tělech funkcí
-z třídít dle uživatelského + systémového času
```

Volby zobrazení:

```
-c zobrazí čas strávený ve funkcích (real time) vedle stromu volání
-m zobrazí umístění souborů
-i potlačí volání funkcí PHP
-O n určuje maximální počet zobrazených funkcí
-t zobrazí zabalený strom volání funkcí
-T zobrazí rozbalený strom volání funkcí
```

Jak vidíte, možností máme skutečně více a hlavně můžeme profilování zrušit zakomentováním jediného řádku v souboru se zdrojovým kódem, což je pružnější než používání tříd z balíčku `Benchmark`. Ukázkový výpis `pprofp` by mohl vypadat například takto (upravil jsem ho tak, aby se vešel na stránku):

```
[pavel@pandora bin]$ pprofp -r /tmp/traces/pprof.05969.12
```

```
Trace for /var/www/metalshop/client/cz/list_products.php
```

```
Total Elapsed Time = 0.45
```

```
Total System Time  = 0.03
```

```
Total User Time     = 0.20
```

	Real	User	System		secs/	cumm				
%Time	(excl/cumm)	(excl/cumm)	(excl/cumm)	Calls	call	s/call	Memory	Usage	Name	
-----										
27.2	0.12	0.12	0.00	0.00	0.00	0.00	0.00	0.00	7	0.0175
										0.0175
							0			pg_query

12.4	0.06	0.06	0.00	0.00	0.00	0.00	1	0.0563	0.0563	0	pg_connect
8.1	0.04	0.04	0.02	0.02	0.00	0.00	858	0.0000	0.0000	0	is_array
7.0	0.03	0.10	0.02	0.09	0.00	0.01	14	0.0023	0.0075	0	require_once
6.8	0.03	0.03	0.03	0.03	0.00	0.00	543	0.0001	0.0001	0	htmlspecialchars
3.4	0.02	0.02	0.01	0.01	0.00	0.00	220	0.0001	0.0001	0	array_keys
3.1	0.01	0.01	0.01	0.01	0.00	0.00	3	0.0047	0.0047	0	defined
2.6	0.01	0.01	0.01	0.01	0.00	0.00	239	0.0000	0.0000	0	explode
2.6	0.01	0.01	0.00	0.00	0.00	0.00	262	0.0000	0.0000	0	count
2.5	0.01	0.01	0.01	0.01	0.00	0.00	366	0.0000	0.0000	0	is_resource
2.0	0.01	0.01	0.01	0.01	0.00	0.00	23	0.0004	0.0004	0	preg_match
1.8	0.01	0.01	0.00	0.00	0.00	0.00	4	0.0020	0.0020	0	include_once
1.5	0.01	0.01	0.01	0.01	0.00	0.00	348	0.0000	0.0000	0	pg_fetch_assoc
1.4	0.01	0.01	0.00	0.00	0.00	0.00	9	0.0007	0.0007	0	array_merge
1.3	0.01	0.01	0.00	0.00	0.00	0.00	38	0.0001	0.0001	0	file_exists
1.2	0.01	0.32	0.00	0.21	0.00	0.03	19	0.0003	0.0167	0	include

## ab, httpperf

Přestože tyto dva nástroje nejsou ani součástí PEARu a vlastně ani přímo nesouvisí s PHP, chtěl bych se o nich alespoň krátce zmínit. Jejich primárním úkolem je měření výkonu webového serveru Apache. Získáte tak přehled o tom, kolik požadavků je server schopen vyřídit za určitý čas. Můžete také simulovat různá zatížení serveru a sledovat reakce vaší aplikace.

Ab je součástí standardní distribuce Apache, takže jej pravděpodobně najdete v adresáři, kde máte nainstalovaného httpd démona. Stejně jako httpperf dovoluje mj. zadat celkový počet požadavků, počet souběžných požadavků, poslat POST request, nastavit cookies atd. [Httpperf \[7\]](#) za ab v ničem nezaostává, naopak poskytuje více možností, jak generovat HTTP požadavky. Jejich hlubší popis je ovšem nad rámec tohoto článku.

## Závěr

Ukázali jsme si nástroje, které vám mohou pomoci s odhalením chyb, které brzdí vaše PHP aplikace. Schválně jsem se vyhnul umělým příkladům a úryvkům anonymního kódu, které by vás dost možná jen zmátly. Nejlepší způsob, jak odhalit sílu a užitečnost podobných nástrojů, je vyzkoušet si je na reálné aplikaci, které rozumíte. Určitě objevíte její slabá místa a to bylo cílem tohoto článku.

## Autentizace

Webové aplikace uživatelům zpravidla nabízejí nejrůznější služby dostupné přes přihlašovací jméno a heslo. Vzniká tak potřeba správy databáze uživatelů, která může být velmi často složitá (účty mohou být uloženy v různých databázových systémech, pravděpodobně budou různě privilegované, obsahují dodatečná data jako doba vypršení, poslední přihlášení apod.) a potřeba vytvořit autentizační systém. Stejně jako vývoj ostatních kritických komponent aplikace si systém ověřování a autorizace vyžádá spoustu času a testování (zvláště kvůli bezpečnosti) a nabízí se tak otázka, jestli by nebylo vhodné poohlédnout se po vyzkoušeném a fungujícím řešení. V PEARu naleznete několik balíčků, které autentizační systém implementují a umí toho možná více, než byste čekali. Tentokrát však buďte při používání cizího kódu ostražití více než obvykle – zabezpečení aplikace a procesům s ním spojeným byste měli dokonale rozumět a cizí kód bezhlavě nepřijímejte.



## Co nabízí PEAR?

### Auth

**Auth** [8] je základní balíček tříd poskytující autorizační a ověřovací mechanismy pro vaše skripty. Přes svoji relativní jednoduchost vám dovolí následující:

- volbu úložiště uživatelských účtů (nedefinovaná „abstraktní“ třída pro práci s úložištěm a z ní odvozené třídy pro práci s databázovými servery (přes PEAR::DB), soubory s uživatelskými účty (flat files), LDAP serverem, SOAP protokolem apod.)
- správu uživatelských účtů (vytvoření, smazání, změna hesla uživatele)
- kontrolu vypršení a nečinnosti (idle-time) relace/uživatele
- možnost používat několik relací v rámci jediné domény (v sessions známých z PHP se jako proměnné vytvářejí asociativní pole ve zdrojových kódech označené jako „session“)
- volitelné vyšší zabezpečení díky sledování IP adresy klienta a jeho HTTP klienta (pozn.: tyto vlastnosti byste měli kontrolovat jen pokud víte, co znamenají – např. IP adresa se může dynamicky měnit a uživatel tak může přijít o relaci).
- ukládat v relaci vlastní data
- často jednoduše migrovat ze starého systému ověřování uživatelů

Při práci s ním dost možná pocítíte např. absenci podpory pro různě privilegované účty, nicméně svojí jednoduchostí vám vše vynahradí tam, kde nejsou potřeba. Mohla by vám také vadit nemožnost využít HTTP autentizaci, ale tu umožní stejnojmenná třída z balíčku **Auth\_HTTP** [9], která **Auth** používá jako svoji předlohu. Příklad typického použití třídy **Auth**:

```
<?php
require_once('Auth.php');
require_once('PEAR.php');

/**
 * Zobrazí přihlašovací formulář.
 */
function displayLoginForm() {
    print("<form action=\"\" method=\"post\">\n");
    print("<fieldset>\n");
    print("<legend>Přihlašovací formulář</legend>\n");
    print("<label>Jméno:</label><input type=\"text\" name=\"user\" />\n");
    print("<label>Heslo:</label><input type=\"passwd\" name=\"passwd\" />\n");
    print("<input type=\"submit\" name=\"submit\" value=\"Přihlásit\" />\n");
    print("</fieldset>\n");
    print("</form>\n");
}

$authOptions = array(
    'sessionName' => 'testDomain',
    'allowLogin' => true,
    'postUsername' => 'user',
    'postPassword' => 'passwd',
```

```

    'advancedSecurity' => false,
    'dsn' => 'pgsql://uzivatel:heslo@zoidberg/pear',
    'table' => 'admin_users',
    'usernamecol' => 'login',
    'passwordcol' => 'pass');

$auth = new Auth('DB', $authOptions, 'displayLoginForm');
$auth->start();
if(!$auth->getAuth()) {
// nepřihlášen ...
}
?>

```

Tento jednoduchý příklad demonstruje obvyklou práci s třídou Auth. Definujeme funkci, která se postará o zobrazení přihlašovacího formuláře, následně nastavíme volby pro autentizaci a vytvoříme instanci třídy. Autentizační proces spustíme funkcí `start()`, která v případě, že uživatel není přihlášen a ze stránky je přihlášení povoleno, zobrazí přihlašovací dialog. Následné volání `getAuth()` vrací logickou hodnotu signalizující úspěšnost přihlášení. Všimněte si nastavení tabulky (`$authOptions['table']`), sloupce se jménem uživatele (`usernamecol`) a heslem (`passwordcol`). Díky této možnosti můžete svůj starý systém, který pravděpodobně funguje podobně, nahradit třídami balíčku Auth.

Nezoufejte ani v případě, že používáte jiné metody šifrování hesla – třídu pro práci s úložištěm dat můžete vždy „zdědit“ a přizpůsobit vlastním potřebám. Než se pustíte do práce s balíčkem Auth, podívejte se do [dokumentace](#) [10] pro uživatele, která vám umožní s třídou okamžitě začít pracovat. Vhod přijdou i znalosti balíčku DB [11], který je v PEARu hojně používán. Jako doplněk k Auth byl vytvořen již zmíněný `Auth_HTTP` [12], který umožňuje poskytnout HTTP autentizaci dle RFC 2617 [13].

## LiveUser

`LiveUser` [14] se na první pohled podobá balíčku Auth – umožňuje všechny akce s ním proveditelné. Proto mi původně nebylo zcela jasné, proč komunita kolem PEARu začlenění tohoto systému do frameworku povolila (přece jen existuje Auth), postupem času (a hlavně během odhalování jeho možností) se ukázalo, jak se věci mají. `LiveUser` totiž mj. obsahuje i rozšířené možnosti administrace uživatelů, o něž byl určitě zájem, a jelikož je najdete v balíčku `LiveUser_Admin`, který je určen k spolupráci s `LiveUser`...

Máte tedy v ruce nástroje, kterými uživatelům přidělujete privilegia, sdružujete je do skupin (hierarchická struktura), které můžete volitelně opatřit výchozími privilegii a co by to bylo za systém, který by neuměl z libovolného práva vyvodit jiné (např. WRITE může implikovat READ). Každé právo lze také opatřit atributem, kterému typu uživatele smí být přiřazeno, a tak vzniká rozlišení na uživatele admina a uživatele řadového. Možností administrace práv je ještě o něco více a najdete je v dokumentaci. `LiveUser_Admin` definuje základní rozhraní pro práci s právy, které rozšiřují tři kategorie tříd lišící se právě v nabízených možnostech. Jmenovitě se jedná o kategorie:

- Simple – libovolné právo lze přiřadit libovolnému uživateli.
- Medium – přidává k Simple skupiny (možnost přiřadit skupině práva a uživatele).
- Complex – všechny dokumentované vlastnosti.

Autoři vyzdvihují jednoduchost migrace mezi těmito kategoriemi díky jednotnému veřejnému rozhraní. Je nutno podotknout, že plnou funkcionalitu nabízí `LiveUser_Admin` jen pro práva uložená

v RDBMS přístupným prostřednictvím PEAR::DB nebo PEAR::MDB. Kategorie Simple je dostupná pro XML soubory.

LiveUser a LiveUser\_Admin tvoří dvojici, která by měla stačit i pro náročnější systémy. V rámci dokumentace projektu [15] je dostupný i užitečný tutoriál, který vám oproti čtení zdrojových kódů ušetří trochu času. Dokumentace na webu ovšem není zdaleka úplná a plné využití schopností balíčku vyžaduje náhled do zdrojových kódů.

## Odkazy

- [1] <http://pear.php.net>
- [2] <http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>
- [3] <http://pear.php.net/manual/en/installation.cli.php>
- [4] <http://pear.php.net/manual/>
- [5] <http://pecl.php.net/package/apd>
- [6] <http://kcachegring.sf.net>
- [7] [http://www.hpl.hp.com/personal/David\\_Mosberger/httpperf/](http://www.hpl.hp.com/personal/David_Mosberger/httpperf/)
- [8] <http://pear.php.net/package/Auth>
- [9] <http://pear.php.net/package/Auth.HTTP>
- [10] <http://pear.php.net/manual/en/package.authentication.auth.php>
- [11] <http://pear.php.net/manual/en/package.database.db.php>
- [12] <http://pear.php.net/package/Auth.HTTP>
- [13] <http://www.faqs.org/rfcs/rfc2617.html>
- [14] <http://pear.php.net/package/LiveUser>
- [15] <http://pear.limbourg.com/>

\*\*\*

---

## Reklamní nabídky [www.praceabc.cz](http://www.praceabc.cz)

Pozice: Vedoucí vývoje v nadnárodní společnosti [1885]

Obor: IT – konzultace/management Kraj: Jihomoravský Vloženo: 08.07.2005

Pozice: Verification and Validation Engineer – mobilní komunikace [2000]

Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 08.07.2005

Pozice: Administrátor [ABC00506]

Obor: IT – správa systémů Kraj: Hlavní město Praha Vloženo: 01.07.2005

Pozice: Systémový specialista – UNIX & Weblogic & Tuxedo, ref. kód: PCSUFC [ABC00459]

Obor: IT – správa systémů Kraj: Hlavní město Praha Vloženo: 30.06.2005

Pozice: C/C++ Programátor – Linux / Windows, ref.kód: PCPRRC [ABC00460]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 30.06.2005

Pozice: Architekt infrastruktury – Linux, aplikační web servery [PCAIFC]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 30.06.2005

Pozice: Unix support engineer – projekty v zahraničí, ref. kód: PCUEFC [ABC00458]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 29.06.2005

Pozice: Unix System Administrator – AIX, ref. code: PMUSFC [ABC00457]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 29.06.2005

Pozice: Systémový specialista – Unix [1869]

Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 27.06.2005

Pozice: Administrátor UNIX [1756]

Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 21.07.2005

Pozice: Programátor C++/Unix – i pro absolventy [1730]

Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 18.07.2005

Pozice: Programátor C/C++ – Windows a Unix (i pro absolventy) [2066]

Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 14.07.2005

Pozice: Architekt infrastruktury – Linux, aplikační web servery [PCAIFC]

Obor: IT – správa systémů Kraj: Hlavní město Praha Vloženo: 13.07.2005

Pozice: Configuration Engineer [1868]

Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 27.06.2005

Pozice: Vývojář Server Applications on UNIX/Linux platforms [ABC00501]

Obor: IT – vývoj systémů Kraj: Jihomoravský Vloženo: 23.06.2005

Pozice: Web programátor [ABC00375]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 14.06.2005

# Adresár /proc – brána do jadra

Matej 'Yin' Gagi

---

*Procfs je virtuálny súborový systém, pomocou ktorého môže používateľ nahliadnuť do vnútorných premenných jadra. Nájdete ho v adresári /proc. Primárne bol vyvinutý na správu procesov, ale Linux z neho urobil veľmi použiteľný nástroj na správu systému.*

---

## Úvod

Prvú implementáciu procfs napísal Tom J. Killian pre UNIX 8th Edition. Cieľom bolo vytvorenie prirodzeného kontrolného rozhrania pre procesy v systéme. Procfs mal slúžiť len programátorom.

V mnohých unixových systémoch bol `/proc` vždy využívaný na kontrolu bežiacich programov a nebol pre užívateľa nijak použiteľným. Procfs je virtuálny súborový systém. To znamená, že žiaden súbor sa nenachádza na vašom disku, ale všetko je v pamäti. Takýto súborový systém zaberá len málo operačnej pamäte a je veľmi dynamický.

## Charakteristika procfs na Linuxe

Na Linuxe nie je procfs používaný len na správu procesov. Vývojári Linuxu postupom času pridávali do štruktúry procfs mnoho rôznych informácií. Zásadným rozdielom od iných unixových systémov je, že každý súbor je textový a užívateľ dokáže prečítať jeho obsah.

Štruktúra procfs sa časom dostala do nekonzistentného stavu, pretože na začiatku neexistovali žiadne pravidlá, podľa ktorých by bol procfs organizovaný. Teraz sa procfs delí na viacero rozumne usporiadaných stromov, ktoré sú však rozhádzané v základnom strome.

## Štruktúra adresára /proc

V hlavnom adresári procfs sa nachádzajú:

- Adresáre procesov
- Adresáre a súbory jadra

```
$ cat /proc/cpuinfo
processor      : 0
vendor_id    : GenuineIntel
cpu family   : 15
model        : 2
model name   : Intel(R) Celeron(R) CPU 2.40GHz
stepping     : 9
cpu MHz      : 2424.546
cache size   : 128 KB
fdiv_bug     : no
hlt_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
```

```

cpuid level      : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic
                 sep mtrr pge mca cmov pat pse36 clflush
                 dts acpi mmx fxsr sse sse2 ss ht tm pbe
                 cid xtptr
bogomips        : 4784.12

```

### Výpis informácií o CPU.

Ukážka štruktúry adresára `/proc`. Adresáre sú vyznačené **hrubým písmom**. *Podčiarknuté kurzíva* označujú symbolické linky. Symbolická linka je súbor obsahujúci len meno iného súboru, ktorú keď otvoríte, správa sa ako súbor, na ktorý sa odkazuje.

## Adresáre procesov

Tieto adresáre spoznáte podľa toho, že ich meno je len číslo. Toto číslo je identifikátor procesu v systéme, nazývané PID (*Process IDentifier*). Štruktúra týchto adresárov je rovnaká. Obsahujú rovnako nazvané súbory:

`cwd` – Adresár, v ktorom sa práve proces nachádza.

`root` – Program môže byť uväznený v niektorom adresári, ktorý je pre neho `/`. Takýto program nemôže pristupovať k žiadnemu adresáru vyššie. `root` sa na tento adresár odkazuje. *Pozri: program chroot.*

`cmdline` – Tu sú všetky parametre programu, s ktorými bol spustený. Nultý parameter je vždy meno spusteného súboru.

`fd/` – V ňom nájdete symbolické linky, ktoré sú nazvané tiež ako čísla. Tieto linky ukazujú na všetky súbory a sieťové spojenia, ktoré má program momentálne otvorené.

Najdôležitejšie položky adresára procesu.

## Súbory a adresáre kernelu

Na iných unixových operačných systémoch sa s týmto typom položiek v `procfs` nestretnete. Je to vlastnosť Linuxu, ktorý robí `procfs` zaujímavým.

Pomocou súborov `procfs` sa exportujú užívateľského prostredia informácie z jadra. Pomocou nich môžeme zisťovať rôzne podrobnosti o systéme a niektorých druhoch hardvéru (viz predchodzí výpis, pozn. ed.). V `procfs` sa podobných súborov nachádza veľké množstvo. Ich obsah môže, ale nemusí užívateľa zaujímať. Informácie, ktoré sa v nich nachádzajú, sú vždy aktuálne, pretože sa vzťahujú priamo na premenné v pamäti.

## Adresár `/proc/sys`

Toto je pre administrátorov najzaujímavejšie miesto v `procfs`. Nachádzajú sa v ňom `sysctl` (*System Control*) súbory. Pomocou nich môžete meniť nastavenie jadra za behu a ovplyvňovať jeho správanie. K tu ukrytým súborom má užívateľ `root` práva na zápis. Zmenu nastavenia uskutočnite jednoduchým zápisom novej hodnoty do súboru, ktorý reprezentuje premennú, ktorú chcete zmeniť.

```

/proc/
|-- 1/
|-- 2/
|-- 3/
|-- 4/
|-- 18/
|-- 20/
|-- 110/
|-- 6203/
|-- 6204/
|   |-- auxv
|   |-- cmdline
|   |-- cwd -> /
|   |-- environ
|   |-- exe -> /usr/sbin/metalog
|   |-- fd/
|       |-- @ -> /dev/pts/4
|       |-- 1 -> /dev/pts/4
|       |-- 2 -> /dev/pts/4
|   .
|   .
|   |-- status
|   |-- task/
|   |-- wchan
|-- 7009/
.
.
|-- acpi/
|-- asound/
|-- buddyinfo
|-- bus/
|-- cmdline
|-- cpuinfo
.
.
|-- swaps
|-- sys/
|-- sysvipc/
|-- tty/
|-- uptime
|-- version
|-- vmstat

```

```
# echo 1 > /proc/sys/dev/cdrom/autoeject
```

### *Nastavenie automatického otvárania CD mechaniky po odpojení.*

Najdôležitejšie pre mnoho serverov je nastavenie siete. V adresári `/proc/sys/net/` sa nachádza mnoho parametrov pre rôzne sieťové protokoly.

Program `sysctl` je nástroj, ktorým si môžeme nastavovanie uľahčiť. `sysctl` vie vypísať a nastaviť všetky hodnoty súborov v adresári `/proc/sys`. Nastavenia majú rovnaký tvar ako cesta k ich súborom relatívne od `/proc/sys`, len lomítka `/` sú zamenené bodkami.

`sysctl` vie zaviesť nastavenia zo súboru `/etc/sysctl.conf`, v ktorom sa ich môže nachádzať viacero. Skoro všetky distribúcie využívajú `/etc/sysctl.conf` na konfiguráciu jadra pri zavádzaní systému.

```
# Disables packet forwarding
#net.ipv4.ip_forward = 0

# Disables IP dynaddr
#net.ipv4.ip_dynaddr = 0

# Disable ECN
#net.ipv4.tcp_ecn = 0

# Enables source route verification
net.ipv4.conf.default.rp_filter = 1

# Enable reverse path
net.ipv4.conf.all.rp_filter = 1
```

### *Ukážka časti konfiguračného súboru /etc/sysctl.conf*

Pozrite sa do `/proc/sys`, kde nájdete mnoho užitočných nastavení, a podľa uváženia ich nastavte v súbore `/etc/sysctl.conf`, aby vaša distribúcia nastavenia aplikovala vždy po reštarte.

## Jednoduché príklady

Všetky nižšie uvedené súbory sa dajú vypísať jednoduchým príkazom:

```
$ cat /proc/<súbor>
```

### `/proc/cpuinfo`

Obsahuje všetky zaujímavé informácie o vašich procesoroch. Napríklad typ procesoru, taktovaciu frekvenciu, inštrukčné sady, ...

### `/proc/version`

Absolútne všetko, čo potrebujete vedieť o verzii jadra, počítači na ktorom je spustené a prostredie v ktorom bolo skompilované.

### `/proc/sys/dev/cdrom/info`

Informácie o CDROM mechanikách. Vie vaša CD mechanika čítať DVD alebo zapisovať na DVD-R, DVD-RW? To sa niekedy z jej krytu nedozviete, z `procs` áno.

### `/proc/sys/dev/cdrom/autoclose`

### `/proc/sys/dev/cdrom/autoclose`

Tieto dva súbory obsahujú binárnu hodnotu (buď 0, alebo 1). Ak v autoclose nastavíme 1, CD mechanika sa pri pripojení (`mount`) CDčka automaticky zavrie (ak bola otvorená). Hodnota 1 v súbore `autoclose` CD mechaniku automaticky otvorí pri odpojení média (`umount`).

### `/proc/sys/net/*`

Tu sa nachádzajú nastavenia sieťovania. Medzi ne patria bezpečnostné nastavenia, ovládanie routovania a pod. Viac o týchto nastaveniach nájdete v dokumentácii zdrojových kódov jadra `Documentation/networking/ip_sysctl.txt`, alebo na [frozentux.net](http://frozentux.net) [1]. Tieto jednotlivé súbory sú len ukážkou. Ak sa budete chcieť dozvedieť niečo viac o inom súbore z `procfs`, jednoducho ho aj s celou cestou vložte do [Google](http://www.google.com) [2].

## História procfs

Ako som povedal na začiatku článku, prvú implementáciu `procfs` napísal Tom J. Killian pre UNIX 8th Edition v roku 1984. Pán Killian je autorom knihy „Processes as files“, takže je vlastne autorom celej myšlienky. V UNIX 8th Edition mal každý proces jeden súbor a `procfs` neexportoval žiadne ďalšie informácie do užívateľského prostredia. `Procfs` bol portovaný pre SVR4 Rogerom Faulknerom a Ronom Gomesom v roku 1991.

4.BSD ako prvý implementoval `procfs` s podadresármi. Solaris 2.6 prevzal všetko dobré, čo bolo v `procfs` použité v iných systémoch, a urobil z neho hlavné rozhranie na prácu s procesmi v systéme. Linux ako prvý použil ASCII formáty v súboroch `procfs`. Linux teda využíva `procfs` aj na exportovanie dát, ktoré sa procesov netýkajú, a na systémovú kontrolu.

## Záverom

Súborový systém `procfs` ponúka veľké množstvo informácií o systéme a o hardvère, ku ktorým sa užívatelia iných OS jednoducho nedostanú. Často sa môžeme stretnúť s tvrdením, že `sysfs` má vytlačiť `procfs` z jadra. To však nie je pravda. Mnoho ľudí chce `procfs` prepracovať, lenže to je zložitejšia úloha a jadrá rady 2.8 žiadne zmeny pravdepodobne neprinesú.

Tony Lawrence sa na [Developer tutorial](#) [3] vyjadril: „`/proc` by mal byť nazvaný `/kernel` a `proc` by sa nachádzal v ňom. Už je neskoro, samozrejme, ...“ V tomto prípade by sa `sysfs` taktiež nachádzal v adresári `/kernel`. `Procfs` ešte nejakú tú chvíľu v Linuxe zostane a nie je na škodu poznamovať sa s jeho výhodami a nevýhodami.

## Odkazy

[1] <http://ipsysctl-tutorial.frozentux.net/ipsysctl-tutorial.html>

[2] <http://www.google.com/linux>

[3] <http://www.developertutorials.com/tutorials/linux/Sysfs-050503/page1.html>

\*\*\*

---

## Reklamní nabídky [www.praceabc.cz](http://www.praceabc.cz)

Pozice: Linux specialista [ABC00480]

Obor: IT – správa systémů Kraj: Hlavní město Praha Vloženo: 07.06.2005

Pozice: ZÁKAZNICKÁ PODPORA – FJ, nebo NJ [ABC00469]

Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 24.05.2005

Pozice: It specialist (junior) [ABC00468]

Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 24.05.2005

Pozice: Unix system administrator [ABC00466]

Obor: IT – správa systémů Kraj: Jihomoravský Vloženo: 23.05.2005

Pozice: Perl/SQL Developer [ABC00434]

Obor: IT – vývoj systémů Kraj: Neuvědno Vloženo: 13.05.2005

# Sysfs – brána do jadra

Matej 'Yin' Gagyí

---

*Sysfs je virtuálny súborový systém podobný procfs. Nájdete ho v adresári /sys. Pomocou neho môžete kontrolovať hardvér na svojom počítači. Dnes si porozprávame o jeho charakteristikách a uvidíme niekoľko príkladov použitia.*

---

## Úvod

Ak používate jadro rady 2.6, mohli ste si vo svojom koreňovom adresári všimnúť adresár `/sys`. Je to prípojný bod pre nový virtuálny súborový systém Linuxu – Sysfs. Sysfs, ako každý virtuálny súborový systém, sa nachádza len v pamäti a na vašom pevnom disku nezaberá žiadne miesto. Jeho obsah presne reprezentuje vnútorný model zariadení v jadre Linuxu. Skoro všetky informácie o vašom hardvéri a ovládačoch sú v jeho štruktúre logicky usporiadané a niektoré i nastaviteľné, podobne ako je tomu pri `Procfs`.

## Charakteristika Sysfs

### Začiatky Sysfs

Sysfs bol vyvinutý špeciálne pre Linux a na iných unixových systémoch sa s ním asi nikdy nestretnete. Prvýkrát bol zaradený do jadra experimentálnej rady 2.5 a predpokladá sa, že v jadre zostane veľmi dlho, nakoľko rieši mnoho problémov. Hlavnou príčinou jeho vzniku bola potreba exportovať štruktúrované informácie z jadra do užívateľského prostredia. V článku [DevFS vs. udev \[1\]](#) píšem o najvýznamnejšom využití Sysfs (využíva ho utilita `udev`).

### Štruktúra Sysfs

Ak sa pozriete do vnútra adresára `/sys`, zistíte, že jeho štruktúra je veľmi hlboká a stále nachádzate ďalšie podadresáre. Často na konci celej štruktúry narazíte na symbolické linky do iných vetiev tohto stromu. Môžeme teda povedať, že nieje užívateľsky príjemná, ale rýchlo si na ňu zvyknete.

Sysfs nieje určené pre priame prezeranie užívateľom. Pomôže mu však, ak si vie napísať aspoň malý skript, ktorý za neho informácie zo štruktúry Sysfs vytiahne. To však nieje pre užívateľa veľmi atraktívne až do chvíle, kým nezačne riešiť nejaký hardvérový problém.

Sysfs presne zodpovedá rozloženiu premenných v jadre. Každý adresár predstavuje samostatný subsystém alebo modul. Súborový obsahujú skoro vždy len jednu informáciu a ich názov túto informáciu presne vystihuje.

### Sysfs ako nástroj

Konkrétnu informáciu môžete z adresára `/sys` dostať jednoduchým výpisom súboru.

```
$ cat /sys/block/fd0/removable
1
$ cat /sys/block/hda/removable
0
```



Dozvedeli sme sa, že disketová jednotka `fd0` používa vymeniteľné média, zatiaľčo disk `hda` je pevne zabudovaný. Nastaviť nejaký parameter jadra za chodu je rovnako jednoduché. Musíte však mať patričné práva na zápis, takže musíte byť prihlásený ako root.

```
# cat /sys/block/hda/queue/scheduler
noop [anticipatory] deadline cfq
# echo "cfq" > /sys/block/hda/queue/scheduler
# cat /sys/block/hda/queue/scheduler
noop anticipatory deadline [cfq]
```

Nastavili sme plánovač fronty príkazov pre pevný disk `hda`. Plánovač `cfq` je vhodný hlavne pre desktop, takže sme trochu zvýšili pružnosť, s akou sa čítanie a zápis na disk budú prevádzať. Niekedy je to o poznanie lepšie. Nastaviteľné parametre jadra spoznáte podľa toho, že daný súbor má priradené práva na zápis. **POZOR!** Nieje dobré zisťovať, na čo slúži daný parameter tým, že do neho začnete zapisovať rôzne hodnoty.

```
$ find /sys -perm +222 ! -type d ! -type l | sort
```

Ak sa o jednotlivých parametroch jadra chcete niečo dozvedieť, skúste si názov súboru (aj s cestou) [vygooglit](#) [2], alebo sa pohrabte v dokumentácii k zdrojovým kódom jadra. Pokiaľ si nieste istý, čo robíte, nič nenastavujte!

## Jednoduché príklady

Abyste sa mali od čoho odraziť, uvediem niekoľko ukážok zaujímavých súborov v `Sysfs` a dodám aj popis.

```
/sys/block/*/queue/scheduler
```

```
/sys/block/*/queue/io_sched/*
```

V súbore `scheduler` si môžete nastaviť alebo vypnúť plánovač fronty úloh pre vaše diskové zariadenia. Plánovač usporiada jednotlivé diskové operácie tak, aby ich bolo možné vykonať čo najrýchlejšie a aby jednotlivé aplikácie, ktoré pristupujú na disk, dostali svoje údaje čím skôr.

**noop**

Táto voľba vypne plánovanie. Ak nemáte spustených niekoľko procesov, ktoré pristupujú na disk, nespozorujete spomalenie.

**anticipatory**

Je rýchly plánovač vhodný pre väčšinu prostredí.

**deadline**

Ak máte na počítači spustenú databázu, ktorá je stále vyťažená, mal by vám pomôcť mierne zvýšiť jej výkon.

**cfq**

Rozdeluje čas prístupu na disk rovnomerne medzi všetky procesy. Jednotlivé požiadavky sa začínú vykonávať skôr, ale dokončujú rovnakou rýchlosťou.

```
/sys/class/*/dev
```

Tento súbtor, ak existuje pre dané zariadenie, obsahuje majoritné a minoritné číslo zariadenia. Je to užitočné, pokiaľ používate statický `/dev` a potrebujete vytvoriť zariadenie ručne.

#### `/sys/class/*/*`

Toto sú adresáre pre všetky zariadenia na počítači, aj tie virtuálne, ktoré sú vytvorené modulmi ako fuse, alebo virtuálne terminály.

#### `/sys/bus/*`

Tieto adresáre reprezentujú jednotlivé typy zberníc vo vašom počítači (PCI, IDE, USB, ...). Ich podadresáre obsahujú zariadenia pripojené na nich a tak ďalej.

Toto sú naozaj len ilustračné ukážky zo štruktúry Sysfs. Nič užitočné som neuviedol, ale zatiaľ som pomocou Sysfs riešil len problémy typu: „Rozpoznalo mi tú zvukovku? Netreba len nastaviť hlasitosť?“ A Sysfs ma ušetril hľadania spôsobu, ako to zistiť.

Druhýkrát som sa so Sysfs stretol, keď som potreboval nechať zaspáť počítač a nevedel som, ako to rýchlo urobiť. Podarilo sa mi to príkazom `echo "standby" > /sys/power/state`, ale lepšie je stiahnuť si programy, ktoré sú na takéto úkony určené.

## Administrácia

Sysfs nieje určený práve na administráciu systému, ale skôr ako zdroj informácií. Väčšie administratívne schopnosti ponúka Procfs. Sysfs ponúka len tie administratívne funkcie, ktoré sú silne späté s hardvérom a ovládačmi.

Na internete nájdete mnoho príkladov užitočného použitia Sysfs. Skúste si vyhliadnúť nejaký súbor v štruktúre `/sys` a vložte jeho celú cestu do [Googla](#) [3]. Mne sa takýmto spôsobom podarilo nájsť niekoľko užitočných návodov. Lenže ako som povedal, tento článok nepíšem ako „návod“, ale „úvod“.

## Záverom

Jedným z veľkých úspechov linuxových programátorov je nepochybne aj Sysfs. Nieje to síce vlastnosť, pre ktorú by masu obyčajných používateľov prechádzali pod Linux, ale tým, čo tak už učinili, prináša nové možnosti.

Často sa môžete stretnúť s tvrdením, že Sysfs má úplne nahradiť Procfs – a to nieje pravda. Vývojári Linuxu a rôznych nástrojov potrebovali štruktúrované informácie o hardvéri, ktoré by boli ľahko spracovateľné programami, a to im Sysfs ponúka. Dúfam, že v článku ste si našli odpovede na niektoré otázky, alebo som vám aspoň ukázal spôsob, ako sa k nim dostanete. V každom prípade, adresár `/sys` už má vo vašom systéme jasné miesto.

## Odkazy

[1] <http://www.abclinuxu.cz/clanky/show/90163>

[2] <http://www.google.com/linux/>

[3] <http://www.google.com/linux>

\*\*\*

---

## Reklamní nabídky [www.praceabc.cz](http://www.praceabc.cz)

Pozice: Perl Developer [ABC00433]

Obor: IT – vývoj systémů Kraj: Neuvedeno Vloženo: 13.05.2005

Pozice: Software Engineer (Unix Source Code) [ABC00429]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

Pozice: Unix Application Specialist (25-35000Kč) [ABC00428]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

Pozice: Senior Unix Developer [ABC00427]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

Pozice: Integration Analyst (40-50000Kč) [ABC00426]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

# Adresářové služby

Lukáš Cirkva

---

*Cílem článku je přiblížit možnosti adresářových služeb a jejich použití v informačních systémech se zaměřením na standardizovaný protokol LDAP.*

---

## Úvod

Prezentovat vlastnosti a možnosti adresářových služeb budu na produktu od firmy Sun Microsystems *Sun Java System Directory Server (JES DS)*, který znám a ve své práci používám. Každopádně níže popsané skutečnosti platí ve velké míře na většinu adresářových serverů LDAP typu, z *Open Source* projektů je třeba zmínit SW *Openldap*.

## Adresář

Adresářem je například klasický telefonní seznam, organizovaný seznam kontaktů a vizitek nebo seznam uživatelských kont v počítačových systémech. Slouží nám k organizaci a sdružování dat do skupin, aby se v nich uživatel lépe orientoval. Adresáře jsou v podstatě jakési zásobníky (kontejnery) pro ukládání dat, specializované databáze, ve kterých lze uchovat a především vyhledat velké množství informací různého charakteru. Jedná se vlastně o hierarchický seznam objektů a atributů těchto objektů, porovnatelný se strukturou adresářů a podadresářů souborového systému.

## Komponenty adresáře

Velmi často se spojují termíny adresář a adresářová služba a jsou považovány za synonyma. Naopak adresářové služby v informačních technologiích se skládají z více částí:

1. Informace uložené v adresáři.
  - identity (uživatelská jména, hesla, fotografie apod.)
  - systémové informace (schémata pro mailové služby, kalendářové služby apod.)
2. Serverový software, který tyto informace poskytuje.
3. Klientský software, který je schopen uživateli tyto informace zprostředkovat.
4. Hardware pro klientský a serverový SW.
5. Podpůrný software jako jsou operační systémy a ovladače zařízení.
6. Síťová infrastruktura, která je schopna spojit klienty se servery.
7. Bezpečnostní politika, která specifikuje oprávnění přístupu k záznamům, aktualizaci dat, apod.
8. Procedury, které adresářové služby používají pro údržbu a monitorování.
9. Software používaný pro údržbu a monitorování adresářových služeb.

Tento dokument je zaměřen na popis použití klientského a serverového SW.

## Použití adresáře

Proč využít adresář s LDAP přístupovým protokolem místo seznamu v souboru nebo relačních databázích?

1. Data uložená v adresáři lze organizovat a spravovat z jediného místa.
2. V hierarchické struktuře lze poměrně snadno rozdělovat administrativní pravomoci.
3. Konfigurace adresáře není závislá na klientském SW. Objekty a atributy lze snadno editovat. Přidáním dalších atributů k objektu není nutné měnit klienta, jen stačí například přidat další

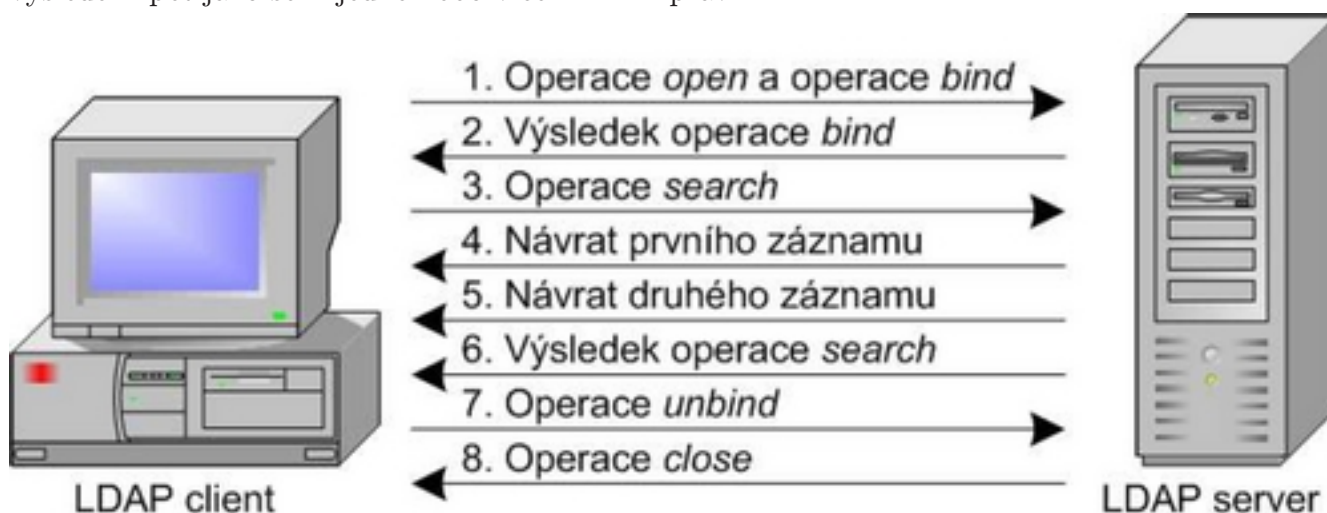
podmínku k využití přidaného atributu. V relačních databázích je poměrně obtížné změnit strukturu již navržené a realizované databáze.

4. Lze zapnout kontrolu schémat poskytující určitou ochranu proti překlepům a syntaktickým chybám.
5. Adresář je optimalizovaný na velké množství čtecích operací.
6. Adresářové služby jsou standardizovány včetně přenosového protokolu a základních schémat. Návrháři aplikací mají tedy jistotu, že budou tato schémata dodržována a nebudou pro každou aplikaci individuální jako je tomu v případě SQL tabulek.

Adresářové služby nejsou určeny k zachycení transakčních dat a neovládají práci s referenční integritou. K tomu se velmi dobře hodí klasické relační databáze využívající SQL jazyka. Nepřipadají v úvahu jakékoliv rozsáhlé transakce spojené například s vkládáním více záznamů do databáze, u kterých je nutné provádět kontrolu správných hodnot dat. Není ovšem velký problém vytvořit jakousi datovou pumpu, která by přečerpávala data mezi SQL databází a adresářovými službami.

## LDAP

*Lightweight Directory Access Protocol* je standardizovaný protokol pro přístup k adresářovým službám, definuje komunikaci mezi serverem a klientem. Zprávy obsahují také příslušná data a informace o jejich formátu. Protokol běží nad internetovými transportními protokoly jako je TCP/IP. LDAP je jednodušší alternativou k protokolu *X.500 Directory Access Protocol (DAP)*, přičemž slouží i jako brána pro přístup k tomuto protokolu. LDAP je protokol orientovaný na zprávy (message-oriented protocol). Klient vytvoří zprávu obsahující nějakou žádost a odešle ji serveru. Ten ji zpracuje a odešle výsledek zpět jako sérii jedné nebo více LDAP zpráv.



Obrázek LDAP komunikace

Příklad access logu, úspěšné operace vyhledání atributu.

```
[22/May/2005:15:41:40] conn=1 op=-1 msgId=-1 - \\  
  fd=35 slot=35 LDAP connection from 127.0.0.1 to 127.0.0.1  
[22/May/2005:15:41:40] conn=1 op=0 msgId=1 - \\  
  BIND dn="cn=Directory manager" method=128 version=3  
[22/May/2005:15:41:40] conn=1 op=0 msgId=1 - \\  
  RESULT err=0 tag=97 nentries=0 etime=0 dn="cn=directory manager"  
[22/May/2005:15:41:40] conn=1 op=1 msgId=2 - \\  
  SRCH base="dc=example,dc=cz" scope=2 filter="(uid=kuky)" attrs=ALL  
[22/May/2005:15:41:40] conn=1 op=1 msgId=2 - \\  
  RESULT err=0 tag=101 nentries=1 etime=0
```

```
[22/May/2005:15:41:40] conn=1 op=2 msgId=3 - UNBIND
[22/May/2005:15:41:40] conn=1 op=2 msgId=-1 - closing - U1
[22/May/2005:15:41:41] conn=1 op=-1 msgId=-1 - closed.
```

*Příklad LDAP komunikace – access [přístupový] log*

LDAP ve zkratce:

**Soubor čtyř modelů – informační, jmenný, funkční a bezpečnostní model.**

1. objekty a atributy – *Directory Information Tree (DIT)*
2. hierarchie objektů

- o root suffix – kořen adresáře

```
dn: dc=example,dc=cz
```

- o subsuffix

```
dn: o=organizace,dc=example,dc=cz
```

- o branchpoint – list adresáře

```
dn: uid=cirkva,ou=People,o=organizace,dc=example,dc=cz
```

3. identifikace objektů pomocí jednoznačného identifikátoru – *Distinguished Name (dn)*
4. 3 druhy schémat (atributy, třídy a namebinding)
  - o atribut – single/multiple (jeden nebo více výskytů u objektu)
  - o třída – may/must (možný nebo povinný výskyt u objektu)
  - o namebinding - definice unikátního rozlišení (moc se nevyužívá)

```
dn: cn=schema
objectclass: top
objectclass: ldapSubentry
objectclass: subschema
cn: schema
#
# aci
#
aci: (target="ldap:///cn=schema")(targetattr !="aci")\
  (version 3.0;acl "anonymous, no acis";
allow (read, search, compare) userdn = "ldap:///anyone");
#
# attribute types:
#
attributeTypes: ( 2.5.4.1 NAME 'aliasedObjectName' \
  DESC 'Standard LDAP attribute type'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE \
  X-ORIGIN 'RFC 2256' )
...
```

```
#
# object classes:
#
objectClasses: ( 2.5.6.2 NAME 'country' \\  
  DESC 'Standard LDAP objectclass'  
  SUP top MUST ( c ) MAY ( searchGuide $ description ) \\  
  X-ORIGIN 'RFC 2256' )  
...
```

*Příklad schéma – 00core.ldif*

*LDAP Data Interchange Format (LDIF)* – standardizovaný textový formát pro výměnu adresářových dat.

Objekty v adresáři jsou vzájemně odděleny prázdným řádkem a skládají se z několika částí:

1. jednoznačné identifikace – distingushed name – dn
2. seznamu tříd
3. seznamu atributů

```
dn: uid=cirkva,ou=People,o=organizace,dc=example,dc=cz
uid: cirkva
iplanet-am-modifiable-by: cn=Top-level Admin Role,dc=example,dc=cz
icsCalendar: cirkva@example.cz
mail: cirkva@example.cz
mailUserStatus: active
gn: Lukáš
sn: Cirkva
mailHost: mail.example.cz
inetUserStatus: Active
userPassword:
objectClass: userpresenceprofile
objectClass: iplanet-am-managed-person
objectClass: top
objectClass: icscalendaruser
objectClass: iplanet-am-user-service
objectClass: inetadmin
objectClass: organizationalperson
objectClass: person
objectClass: sunssoadapterperson
mailDeliveryOption: mailbox
icsExtendedUserPrefs: ceNotifyEmail=cirkva@example.cz
icsExtendedUserPrefs: ceDefaultView=weekview
icsExtendedUserPrefs: sunCalEventfilter=accepted,tentative,declined,needs-action
icsFirstDay: 1
preferredLanguage: cs
sunSSOAdapterConfigurations: default|http://?configName=sunOneCalendar&configDe
```

```
sc=SUN-ONE-CALENDAR&domain=example.cz
nsRoleDN: cn=Role,o=organizace,dc=example,dc=cz
...
```

*Příklad LDIF se systémovými i uživatelskými informacemi*

## LDAP serverový software – soubor software pro provoz LDAP serveru.

Příkazy k použití:

- search
- compare
- add, delete
- modify
- dn, bind, unbind, abadon ...

Příklady SW:

- Openldap
- Sun JES DS
- Microsoft Active Directory

## LDAP klientský software – soubor programů schopných práce s LDAP serverem.

Příklady GUI:

- gq
- ldapbrowser
- Luma
- Sun Java Console

## LDAP API – aplikační programové rozhraní pro vývoj software

### Adresáře v informačních systémech

S adresáři v informačních systémech setkáváme a budeme setkávat stále více. Všude se mluví o tom, jak je třeba hierarchické seznamy identit integrovat do jediného adresářového serveru. LDAP adresáře jsou v mnoha případech tím pravým řešením.

Příklady využití LDAP adresáře:

- Přihlašování do operačního systému – UNIX, Linux, Windows,...
- Náhrada jmenných služeb typu NIS.
- Integrace identit služeb elektronické pošty – SMTP, IMAP/POP.
- Informace pro přístup do MIS a ERP.
- Informace pro servery síťových služeb – DHCP, DNS, HTTP serverů.

## Závěr

Doufám, že čtenář při čtení toho článku získal alespoň částečný rozhled o možnostech adresářových služeb, a že v něm vzbudil zájem o další informace o této problematice. Tento text ovšem není možno brát jako kompletní manuál, k tomu byly o problematice adresářových služeb napsány obsáhlé publikace. V dalších částech tohoto seriálu bude popsána instalace, administrace i s názornými příklady. Speciální poděkování patří **Karlu Benákovi** [1], který je spoluautorem této části, a na dalších částech se podílel velmi podmětnými připomínkami.

## Odkazy

[1] <http://www.abclinuxu.cz/Profile/3014>

\*\*\*

# Internetová rádia

Robert Krátký

---

*Na internetu lze nalézt tisíce rádií všech možných žánrů a zaměření. Přečtěte si, které programy vám pomohou s poslechem a nahráváním.*

---

## Není rádio jako rádio

Pro rozdělení internetových rádií bychom mohli použít rozlišení na internetová vysílání klasických rádií a ten zbytek. Klasická rádia za svůj hlavní distribuční kanál považují rádiové vlny, internetové vysílání je doplňkem. Výhoda je zřejmá: je možné poslouchat třeba jihočeské rádio v losangelském hotelovém pokoji.

Tím však ve většině případů výhody tohoto typu internetových rádií končí. Vzhledem k tomu, že jde o živé vysílání, dostane se vám i aktuálních zpráv, ale je to také bohužel vykoupeno reklamními bloky. A rádio bez reklamy, to je jednou z hlavních předností „toho zbytku“.

Další výhodou „zbytku“ jsou relativně nízké náklady – stačí zajistit konektivitu a stroj, který lze pro vysílání použít. Pak už je potřeba jen minimum údržby, vlastně se může omezit jen na zadávání playlistů. A právě minimální náklady jsou zárukou širokého výběru stanic. Zatímco klasická rádia si většinou nemohou dovolit orientaci jen na úzký profil posluchačů, protože by je pak reklamní příjmy neuživily, plnokrevná internetová rádia bývají často zaměřena ne pouze na žánr nebo podžánr, ale nezřídka i na jedinou hudební skupinu nebo interpreta. Máte-li tedy zrovna náladu na celé odpoledne Pink Floyd, jednu desku za druhou, neměl by být velký problém takové vysílání nalézt.

## Formáty

To takzvané vysílání se dá realizovat pomocí několika vzájemně nekompatibilních formátů. Některé proprietární (Real, ASF, WMA, MP3 aj.), jiné otevřené (Ogg Vorbis). Nezáleží příliš na tom, jaká aplikace (server) se pro vysílání (streamování) používá. Důležité je, který formát posílá. Nejsnazší to na Linuxu budete mít s OGG a MP3, ale i ostatní formáty lze poslouchat, když máte nainstalované příslušné kodeky (a přehrávač, který je umí využívat – např. MPlayer nebo Xine).

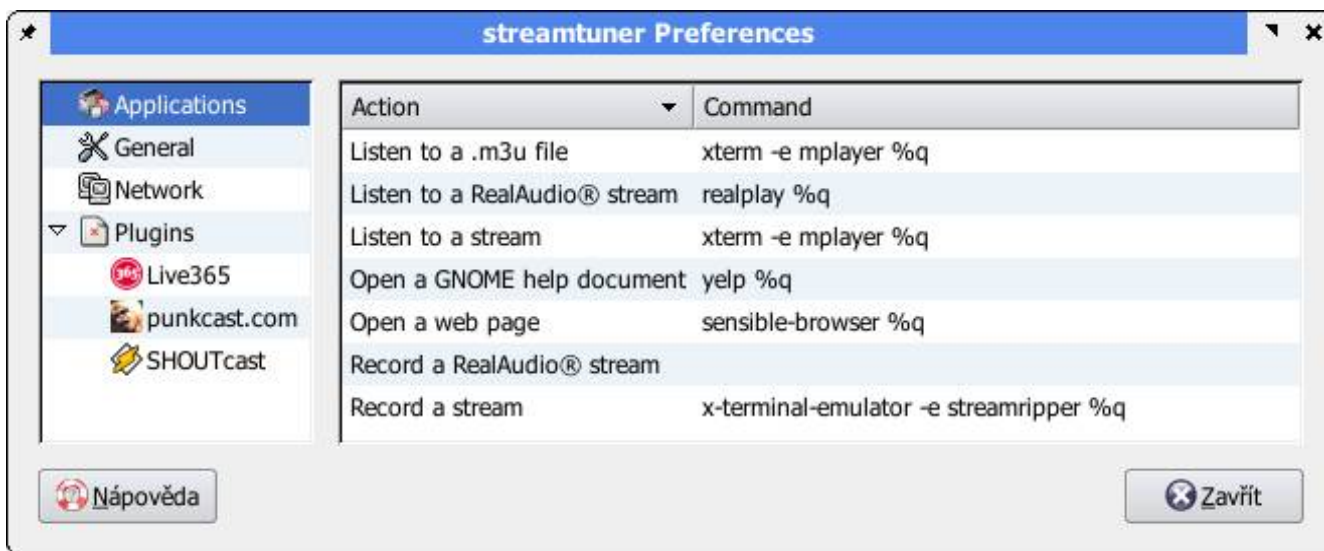
## Jak rádio najít

Jak už jsem řekl, internetových rádií jsou tisíce. Existují proto servery, které obsahují katalogy. Ale pohodlnější je vyhledávat bez reklamních bannerů v programu přímo k tomu určeném.

## Streamtuner

Program pro procházení internetových adresářů rádií. V jednoduchém a přehledném prostředí vám umožní procházet v seznámech stanic. Vybírat můžete podle žánru, právě hrané skladby, bitratu vysílání a počtu posluchačů. Streamtuner umí základní vyhledávání a je možné nastavit akce, které chcete s vybranou stanicí provést (nejčastěji asi začít přehrávat v oblíbeném přehrávači). Jde o GTK+2 program – nenechte se zmást screenshotem. A nebijte mne za používání [Gtk-Qt engine](#) [1]. Ovládání je snad velmi snadné, takže bližšího popisu snad není třeba.





streamtuner

## Jak rádio přehrát

Do toho se snad radši nebudu pouštět. . . Každý má jistě svůj oblíbený přehrávač. A v naprosté většině přehrávačů lze přehrát všechny rozumné formáty (tj. Ogg Vorbis, případně MP3). Různé přehrávače nabízejí různou úroveň komfortu při práci (tedy zábavě) se „streamovanými“ daty. Některé budou hezky zobrazovat název každé nové písničky a budou rozumně udržovat zaplněný buffer (vyrovnávací paměť), aby se při náhodném chvilkovém zahlcení linky nestalo, že rádio bude přeskakovat jako staré elpíčko. Některé ne... ale základní přehrávání by měly zvládnout i ty nejjednodušší. Ať už používáte XMMS, Beep Media Player, amaroK, MPlayer nebo třeba Xine (nebo některou z nadstaveb), vždy se přehrávání internetového streamu spustí prostě tak, že místo souboru „otevřete“ adresu, na které stanice vysílá. Případě zadejte adresu jako parametr na příkazový řádek. Většina přehrávačů akceptuje i playlisty, ve kterých bývá kromě URL i název stanice.



SHOUTcast

## Jak rádio nahrát

Konečně to opravdu zajímavé. Pomocí šikovných malých programků si můžete po internetu streamované rádio ukládat na disk. A co více, vysílání si můžete nechat rozdělit na jednotlivé songy, které budou v samostatných souborech pojmenovaných podle názvu skladby a interpreta (nebo jiných kritérií).

Když si takto uložíte pár hodin vysílání, budete mít příjemné rozptýlení do svého přenosného přehrávače nebo třeba do auta (znáte-li už vaše narippovaná cedéčka nazpaměť). Já to také využívám u opravdu zajímavých rádií, z nichž si pak vybírám budoucí úlovky pro příští návštěvu obchodu s deskami. Kdybych spoléhal na to, že si jméno skupiny a písničky poznamenuji při obyčejném poslouchání, když mě zrovna něco zaujme, neměl bych nikdy nic, protože při práci si na takové věci nikdy nevzpomenu.

## Streamripper

Tento konzolový nástroj má sice i grafické nadstavby (KStreamripper, Gstreamripper a možná další), ale jeho použití je tak snadné a funkce tak prostá, že používání nadstavby se podle mého názoru nevyplatí, ani pokud máte z příkazové řádky výrazku.

Program zvládá všechno, co byste od něj očekávali. Automaticky vytvoří adresář pro každou stanicí a do něj ukládá jednotlivé soubory pojmenované podle údajů, které rádio poskytuje. Můžete (mimo jiné) určit, jak dlouho má nahrávání trvat, případně maximální objem dat, který se má uložit. Základní použití: `streamripper http://adresa.radia.cz:8000`. Číslo na konci určuje, na kterém portu se má Streamripper připojit. Může se samozřejmě lišit. Podrobněji se zmíním jen o jedné z možností, které Streamripper nabízí...

## Jak rádio nahrávat a zároveň poslouchat

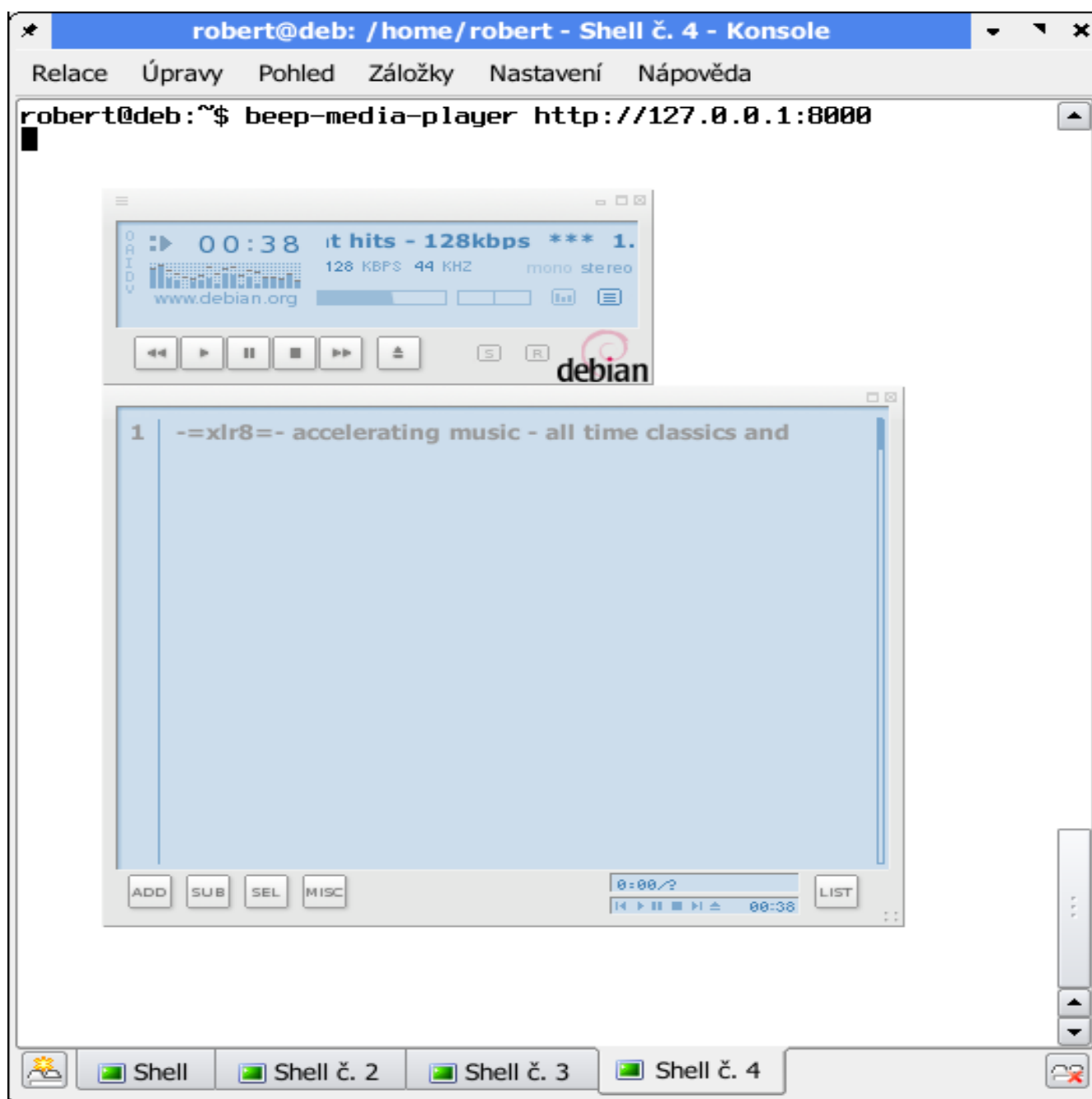
Může se stát, že ačkoliv vysílání chcete zachovat uložené na disku, rádi byste ho při nahrávání rovnou poslouchali. Při výše uvedeném použití by to nešlo, protože přijímaná data polyká Streamripper a na přehrávač hudby už nic nezbyvá. Řešením by bylo přehrávač nasměrovat na vysílající server také, ale kromě toho, že by to bylo neekonomické a neslušné vůči provozovateli stanice, nemusela by vám ani stačit šířka pásma.

Místo toho prostě Streamripperu řekneme, aby data kromě ukládání na disk také posílal dál. Streamripper pro nás vytvoří malý streamovací server a na jeho adresu potom nasměrujeme přehrávač. Může to vypadat například takto:

```
streamripper http://adresa.radia.cz:8000 -r -R 5
```

Parametrem `-r` streamripperu říkáme, že má vytvořit relay server na portu, jehož číslo jsme nezadali, a proto bude použit standardní 8000. Další parametr je nepovinný, ale ukážeme si na něm další možné využití parametru `-r`.

`-R 5` určuje, že se k portu 8000 na vytvořeném serveru může připojit jen pět klientů. Používáte-li funkci nahrávání a současného poslouchání na domácím počítači, bude vám to asi jedno, protože jediným klientem bude váš vlastní přehrávač. Ale pokud chcete parametr `-r` využít k tomu, aby Streamripper vytvořil falešné internetové rádio pro další klienty vaší sítě třeba z okolních kanceláří (aby se každý nepřipojoval zvlášť, a neplýtval tak přenesenými daty), může vás zajímat, jak počet připojujících se fanoušků korigovat. Hodit se pak může i parametr `-A`, který zařídí, že jednotlivé soubory se stopami nebudou vytvářeny.



## Co posloucháte

Jaká rádia posloucháte vy?

## Odkazy

[1] [http://www.freedesktop.org/wiki/Software\\_2fgtk\\_2dqt](http://www.freedesktop.org/wiki/Software_2fgtk_2dqt)

\*\*\*

Reklamní nabídky [www.praceabc.cz](http://www.praceabc.cz)

Pozice: UNIX Quality Specialist (25-35000) [ABC00424]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

Pozice: Analyst/Test Engineer – UNIX (25-35000) [ABC00423]

Obor: IT – správa systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

Pozice: IT Specialist (UNIX, SQL) (25-30000Kč) [ABC00422]

Obor: IT – vývoj systémů Kraj: Hlavní město Praha Vloženo: 12.05.2005

# Jaderné noviny 313–316

Robert Krátký

---

*Linux 2.6.12-rc4-mm2. sdparm verze 0.92. dmraid verze 1.0.0.rc8. Linux 2.4.31-rc1. Nová subarchitektura pod arch/cris. Vývoj ALSA přechází na git. Porušuje Panda GPL? GCC 4 neumí kompilovat jádro 2.4. Aktualizace Průvodce vývojáře libata. Strom -mm bude k dispozici jako repozitář git. Nové automatizované testovací skripty pro oficiální a vývojové verze jádra. Někteří vývojáři chválí řadu w.x.y.z. Nový skript pro výpis krátkého changelogu z gitu. Migrace na 4K kernel stacks. Shrnutí real-time problémů. DevFS odstraněno z Linuxu. Linux 2.4 nebude podporovat GCC 4. Linu-xová dokumentace, kterou lze prohlížet jen v nelinuxových prohlížečích.*

---

## Linux 2.6.12-rc4-mm2, 27 e-mailů

16. kvě – 24. kvě

Andrew Morton oznámil Linux 2.6.12-rc4-mm2:

<ftp://ftp.kernel.org/pub/linux/kernel/people/akpm/patches/2.6/2.6.12-rc4/2.6.12-rc4-mm2/> [1]

- davem založil mm-commits mailing list, aby mohli lidi kontrolovat věci, které jsou do -mm přidávány nebo z něj odebírány. Přihlaste se takto: echo subscribe mm-commits — mail majordomo@vger.kernel.org
- Aktualizace architektury x86\_64 od Andiho.
- Všechno až po (a včetně) 'spurious-interrupt-fix.patch' je plánováno k začlenění do 2.6.12. A ještě je tam pár dalších věcí.
- Další aktualizace DVB subsystému.

## sdparm verze 0.92, 4 e-mailů

20. kvě – 24. kvě

Douglas Gilbert napsal:

sdparm je utilita pro příkazovou řádku, která zjišťuje a nastavuje parametry SCSI disků (jako hdparm pro ATA disky). Obecněji řečeno jde o nastavování informací na stránkách režimů u SCSI zařízení nebo zařízení, která využívají příkazovou sadu SCSI (např. CD/DVD mechaniky (jakýkoliv přenos) a páskové SCSI mechaniky). Umí také vypsat identifikační popisky zařízení z VPD stránek.

Více informací a tarball, rpm a deb balíky zde: <http://www.torque.net/sg/sdparm.html> [2]sdparm v současné době přistupuje k více než 150 parametrům. Díky Kai Makisarovi za rady ohledně rozhraní a kontroly správnosti.

Douglas připojil seznam změn pro sdparm-0.92 a Mogens Valentin se zeptal: Pěkné! Právě jsem to stáhl a vyzkoušel na externím USB disku. Hodila by se mi jedna funkce: mohl bys přidat možnost zpomalit/roztočit USB SCSI disk? Bylo by to fajn pro externí (USB) disky. Vypadá to, že to zatím neumí. Jestli jsem to jen přehlédl, tak se omlouvám.

Douglas odpověděl:

Je toho opravdu dost ;-). Právě teď přidávám podporu pro stránky režimů podle různých transportů (protokolů). Takže sdparm se momentálně specializuje na stránky režimů (pro všechny SCSI příkazové sady) a INQUIRY informace (včetně VPD identifikační stránky zařízení).

## dmraid verze 1.0.0.rc8, 1 e-mail

23. kvě Heinz Mauelshagen napsal:

dmraid 1.0.0.rc8 je k dispozici na: <http://people.redhat.com:/heinzm/sw/dmraid/> [3] jako tarball se zdrojákem, zdrojový rpm a i386 rpm (se sdílenou a statickou binárouk dietlibc). Toto vydání přidává parametr '-ignorelocking', aby mohl být dmraid spuštěn brzy při bootu, když ještě neexistuje přístup pro čtení/zápis do /var. dmraid (Device-Mapper RAID nástroj) zjišťuje, (de)aktivuje a zobrazuje vlastnosti softwarových RAIDů (tj. ATARAID) a obsažených DOS oddílů pomocí device-mapper runtime jádra 2.6. Linux 2.6 podporuje následující typy ATARAID:

- Highpoint HPT37X
- Highpoint HPT45X
- Intel Software RAID
- LSI Logic MegaRAID
- NVidia NForce
- Promise FastTrack
- Silicon Image Medley
- VIA Software RAID

Poskytněte, prosím, informace, aby mohly být tyto formáty metadat podporovány kompletně.

## Linux 2.4.31-rc1, 1 e-mail

24. kvě

Marcelo Tosatti oznámil Linux 2.4.31-rc1:

Obsahuje malé množství jednoduchých oprav na různých místech a aktualizaci tg3.

## Nová subarchitektura pod arch/cris, 2 e-maily

26. kvě

Mikael Starvik napsal:

Chystám se založit novou subarchitekturu pod arch/cris a rád bych od vás slyšel, jak to nejlépe provést. Napadají mne tři možnosti:

1. Všechny patche jdou do LKML. Spousta nudných patchů, které většinu lidí nezajímají.
2. Odeslat ovladače správcům subsystémů ke kontrole a pak velký .tgz Andrewovi.
3. Odeslat .tgz přímo Andrewovi bez kontrol od lidí, kteří mají na starosti ovladače.

Příkláním se k možnosti č. 2. Kromě nové subarchitektury budou také patche pro obecné části cris, které přidají např. podporu SMP a provedou převod na nový systém IRQ.

Andrew Morton odpověděl, že možnost č. 2 mu zní nejlépe.

## Vývoj ALSA přechází na git, 16 e-mailů

27. kvě – 29. kvě

Jaroslav Kysela napsal:

Vytvořil jsem nový git strom pro projekt ALSA:

<rsync://rsync.kernel.org/pub/scm/linux/kernel/git/perex/alsa.git>

Tento strom má za úkol tlačit změny v ALSA k Linusovi. Jakmile bude poprvé synchronizován, začne sledovat vývoj jádra a cyklus vydávání nových verzí. Teď obsahuje všechny nejnovější patche z ALSA 1.0.9. Další repozitář vytvořím (pravděpodobně alsa-devel.git) pro testování nejčerstvějších změn v ALSA ovladačích (zaměřené na jádra -mm).

Linus Torvalds poukázal na to, že Jaroslavova konverze na git ponechává na počátku každé zprávy s commitem prázdný řádek, což mate některé webové prohlížeče git repozitářů.

Následovala krátká diskuze o tom, jestli by git nemohl nadbytečné prázdné řádky odstraňovat, ale rozhodující se ukázala být skutečnost, že logové zprávy mohou být i čistě binární a odstraňování jejich částí by způsobilo poškození dat. Nakonec Jaroslav svůj strom konvertoval znovu, ale bez prázdných rádků. Jaroslavův strom měl ještě jednu nezvyklou vlastnost – řádky „Signed-off-by“ [podepsáno] nedávaly navrch skutečného autora, což by měly. Linus k tomu vysvětlil:

Algoritmus vypadá takto:

– autor emailu, *nebo*, existuje-li, vrchní „From:“ v těle zprávy. A pravidlem je, že existující From: nikdy neodstraňujeme (ani nepřidáváme), protože autor se přeposíláním nezmění. Jinými slovy: autorství je velmi odlišné od sign-off. Sign-off se hromadí, autorství je konstantní – a proto jsou pravidla odlišná. Autorství je důležitější než sign-off, takže je nahoře, kdežto signed-off dole.

Později konvertoval Jaroslav svůj strom ještě jednou tak, aby odpovídal těmto požadavkům.

## Porušuje Panda GPL?, 6 e-mailů

30. kvě

Carlos Silva napsal: Společnost, pro kterou pracuji, se chystá koupit dva na Linuxu založené Panda Gatedefenders 8200. Pro začátek nám neřekli, ani jejich stránky to nikde nezmiňují, že jejich produkt využívá Linux. Další věc je, že nám nedají root přístup nebo žádný jiný lokální přístup k tomu zařízení – kromě HTTP stránky, kterou tam mají pro správu. Když jsem požádal o zdrojové kódy jádra, které v systému používají, řekli, že mi je nemohou dát, protože v tom je nějaký jejich proprietární kód. Måns Rullgård řekl, že firma nemá povinnost poskytnout plný přístup k zařízení. Ale připojil: Pokud používají nějaké vlastní moduly, panuje obecná shoda, že jejich zdrojové kódy zveřejňovat nemusí. Ale měli by ti dát zdrojové kódy použitého jádra společně se všemi patchi, které přidali. Carlos napsal, že to ví, ale zopakoval: Ale oni mi nechtějí dát vůbec nic. Jen prodat zařízení, nic víc :(.

Måns odpověděl: To je bohužel běžné. Měl bys začít tím, že slušně připomeneš, jaké jsou jejich povinnosti. A když neuposlechnou, přejdi na tvrdší tón a zveřejni korespondenci s nimi. Nejsem si jistý, jestli je LKML tím pravým místem, ale asi je to jediné místo, které máme. Pokud všechno selže, pošli na ně Slashdot. To pravděpodobně také nic nevyřeší, ale aspoň se budeš moci potěšit pohledem na to, jak jim jiní nadávají.

Carlos odpověděl: Už jsem je požádal slušně i tvrdě, takže to je patová situace. Giuseppe Bilotta poradil zajít na <http://www.gpl-violations.org/> [4].

## GCC 4 neumí kompilovat jádro 2.4, 3 e-maily

30. kvě

Adrian Bunk napsal: gcc 4 není jádrem 2.4 podporován pro kompilaci a já nevím o žádném důležitém důvodu, proč by kdy měl. Tento patch ten fakt dokumentuje.

Mikael Pettersson připomněl: Je to zbytečné. Jakýkoliv pokus o kompilaci vanilla jádra 2.4 pomocí gcc4 skončí kompilačními chybami.

Ale Adrian odpověděl: Bez tohoto patche ti obrazovku zaplaví varování a chyby, když se o tu kompilaci omylem pokusíš. S tímto patchem se stane totéž, ale poslední řádky budou obsahovat vysvětlení:

```
#error Sorry, your GCC is too recent for kernel 2.4
#error Sorry, váš GCC je pro jádro 2.4 příliš nový
```

## Aktualizace Průvodce vývojáře libata, 1 e-mail

30. kvě

Jeff Garzik napsal: Právě jsem prošel dokumentaci k libata, aktualizoval ji s ohledem na nejnovější háčky [hooks] a doplnil chybějící detaily jako popis zamykání/kontextu. Pohrál jsem si s auto-publikováním na lulu.com a vydal tohoto nově aktualizovaného Průvodce vývojáře libata jako krátkou knihu (92 stran): <http://www.lulu.com/content/130446> [5]. Myslím, že ta blátivá cesta na předním přebalu je příznačná :).

Průvodce je i nadále open source – stačí spustit 'make pdfdocs' v adresáři se zdrojáky jádra. PDF kniha je k dispozici zdarma z lulu.com nebo

<http://www.kernel.org/pub/linux/kernel/people/jgarzik/libata.pdf> [6].

Zdrojový kód průvodce je součástí jádra, přičemž nové dodatky jsou ve větvi 'docs' zde: <rsync://rsync.kernel.org/pub/scm/linux/kernel/git/jgarzik/libata-dev.git>

## Strom -mm bude k dispozici jako repozitář git, 20 e-mailů

01. črv - 05. črv

Andrew Morton oznámil Linux 2.6.12-rc5-mm2 a Pavel Machek se zeptal:

Uvažoval jsi o vydávání -mm pomocí gitu? Předpokládám, že kvůli způsobu tvé práce nemůžeš git normálně používat, ale i pouhé vydávání verzí pomocí gitu by bylo skvělé. (V současné době sleduji se svým stromem Linuse, git to hezky usnadňuje. Sledování -mm je však protivná ruční práce s diff, patch a ketchup...)

Matthias Urlichs odpověděl: Napsal jsem skript (vlastně to pochází z doby importování mm-do-BK), který stahuje -mm do gitu jako samostatné commity. Andrew: Mohl bys před patche, které stahuješ z gitu, vložit následující řádek:

```
GIT SHA1-vrchního-commitu URL-archívu
```

abych mohl commity označovat příslušným druhým rodičem? Pokud neděláš změny přímo na -mm, ale pouze jej (nebo je) začleňuješ, pak to funguje docela dobře.

Pavel reagoval: Výborně! Mohl bys výstup toho skriptu někam exportovat? Řekl bych, že bys pro to mohl dostat účet na kernel.org...

Matthias odpověděl, že to rád udělá, pokud mu ten účet někdo dá.

## Nové automatizované testovací skripty pro oficiální a vývojové verze jádra, 8 e-mailů

02. črv - 07. črv

Martin J. Bligh napsal: OK, konečně jsem to dotáhl do podoby, ve které s tím mohu jít ven. [http://ftp.kernel.org/pub/linux/kernel/people/mbligh/abat/regression\\_matrix.html](http://ftp.kernel.org/pub/linux/kernel/people/mbligh/abat/regression_matrix.html) [7]

V současné době zkompiluje a nabojuje jakékoliv hlavní, -mjb a -mm jádro během 15 minut od vydání. Spustí dbench, tbench, kernbench, reaim a fsx. Používám na to stroj s 4x AMD64 a 16x NUMA-Q, 4x NUMA-Q, 32x x440 (ia32) PPC64 Power 5 LPAR, PPC64 Power 4 LPAR a holý PPC64 Power 4 systém. Používané konfigurační soubory jsou odkazovány jako názvy strojů v záhlaví sloupců.

Andrew, měl bys díky tomu snáze vidět selhání při bootu -mm. Budeš-li chtít, zapracuji na nějakých automatických emailových upozorněních. Díky všem lidem z IBM, kteří pracovali na testovacím systému ABAT, na němž je to založeno – příliš mnoho, než abych je mohl vypisovat. Hlavně však díky Andymu, Adamovi a Enriquovi, kteří opravili hromady chyb a vydrželi moje nekonečné držkování o tom, že to nefunguje, jak by mělo.

Bude to umět další věci... více profilování, více testů, bude to zvládat patche atd. Nechci na kernel.org tlačit moc velké objemy (v tuhle chvíli kopíruji jen malou část vygenerovaných souborů), ale ve frontě

jsou například další zkoušky s opravnými patchi pro -mm. Také to dělá noční kompilace -bk a -git, které teď nejsou v tabulce, ale vrátím je tam brzy.

Andrew, budeš-li chtít přidat nějaké další testy, dej mi vědět... i když můžeme teoreticky dělat i testy s více stroji (např. zasítovanými), rád bych se prozatím držel těch na jednom stroji. Máme velkou hromadu testů, které chceme přidat... ale chtěl jsem ukázat aspoň ten obecný výsledkový mechanismus, aby bylo, na co se podívat.

Kliknutí na hlášku o selhání by vám mělo ukázat něco alespoň zběžně užitečného pro diagnózu problému. Kliknutí na číslo testu hned pod tím vás hodí na informace, které by měly obsahovat výkonnostní výsledky, profily atd. Později přidám grafy porovnávající výkonnost různých jader (už je mám... jen ne automatizované).

Jeff Garzik odpověděl: *Velmi pěkné. Bude to moc užitečné, díky.*

## Někteří vývojáři chválí řadu w.x.y.z, 11 e-mailů

03. črv – 07. črv Jeff Garzik napsal: *Myslím, že stabilní série jader 2.6.x.y funguje docela dobře. Pochvalme za dobře odvedenou práci tým stable@kernel.org. Řada 2.6.x.y nepochybně vyplňuje mezeru.*

Szonyi Calin nevěřil, že by mohl být strom 2.6 stabilizován pouhým přidáním několika patchů, což řada w.x.y.z zjevně dělá. Ale Szonyi uznal, že i přesto se strom zdá být stabilnější než běžný 2.6. Ale Alan Cox s Jeffovou pochvalou naprosto souhlasil: *Je natolik konzervativní, že nejenže zůstává docela stabilní (jedna chyba s oddíly za celou dobu, to je opravdu dobré), ale je i tak malý, že většina přídatných patchů, které lidi používají, lze dobře aplikovat. Dokonce i -ac se (až na makefile) daří bez potíží aplikovat, takže mi to šetří spoustu práce.*

## Nový skript pro výpis krátkého changelogu z gitu, 6 e-mailů

04. črv – 04. črv

Jeff Garzik napsal: *Připojuji skript 'git-shortlog', který jsem dal dohromady, aby napodobil skript pro vytváření krátkého changelogu [seznamu změn] v dobách BitKeeperu.*

*shortlog načte changelog ve formátu 'git-whatchanged'. Např.:*

```
git-whatchanged | git-shortlog
```

nebo

```
git-shortlog changes.txt
```

a výstupem jsou změny seřazené podle autorů:

```
autor1:
  cset 1-line popis
  cset 1-line popis
  ...
autor2:
  cset 1-line popis
  ...
...
```

Protože git rozlišuje mezi 'autorem' a 'committerem', spustil jsem

```
git-whatchanged | git-shortlog $>$ changes.txt
```



abych se podíval na autory jádra během celé historie (která je v gitu). Je to zábavné čtení, protože tohle je poprvé, co máme možnost získat obrázek o tom, kdo patche doopravdy píše, a kdo je jen commituje. Viz připojený changes.txt.bz2.

Linus Torvalds odpověděl: Díky, přidám to ke gitu a další vydání jádra už bude mít pořádný krátký changelog. Mimochodem, ukazuje to, že tvůj slučovací [merge] skript funguje špatně: z nějakého důvodu nevyplňuješ správně pole AUTHOR:

```
<jgarzik@pretzel.yyz.us>:  
Automatic merge of /spare/repo/netdev-2.6 branch r8169-fix  
Automatic merge of /spare/repo/linux-2.6/.git branch HEAD  
Automatic merge of /spare/repo/netdev-2.6 branch use-after-unmap  
Automatic merge of rsync://rsync.kernel.org/.../torvalds/linux-2.6.git branch HEAD
```

Ale „committer“ je správně. Oprav si to, prosím.

A ještě dodal:

Důvodem, proč jsi to přehlédl, je fakt, že „git-whatchanged“ obvykle slučování ignoruje. Proved

```
git-rev-list --pretty HEAD ^v2.6.12-rc5 | git-shortlog | less -S
```

abys viděl, o čem mluvím („zobraz krátký changelog všech změn od 2.6.12-rc5“).

*Poznámka překladatele: Tento díl Jaderných novin vychází s tak velkým zpožděním, protože jsem originál KT 316 – jakkoliv se mi samotnému nechtělo uvěřit – přehlédl. Prostě vyšel, ale já si toho nevšiml...*

## Migrace na 4K kernel stacks, 13 e-mailů

07. črv – 10. črv

Adrian Bunk napsal: 4Kb kernel stacks jsou budoucností pro systémy i386 a vypadá to, že problémy, které ze začátku způsobovaly, jsou již vyřešeny. Rád bych:

- dostal do dalšího -mm patch, který by bezpodmínečně zapínal 4KSTACKS
- nevyskytnou-li se chyby, tak bych do 2.6.13 nebo 2.6.14 poslal patch, který by zcela odstranil !4KSTACKS.

Jediným problémem je, že na !4KSTACKS stále závisí REISER4\_FS. Už v březnu jsem Hansovi říkal, že je potřeba to změnit. Máte představu, za jak dlouho by mohly být všechny 4K problémy v Reiser4 vyřešeny? Pokud ne, budou se možná lidi, kteří používají Reiser4, rozhodnout, jestli změnit souborový systém nebo architekturu...

Alexander Nyberg poukázal na to, že určité kombinace konfiguračních voleb nebudou bez 4KSTACKS fungovat, takže to nemůžeš jen tak odstranit bez možnosti vybrat si. Před nějakou dobou to ani nebylo složité simulovat a nevšiml jsem si žádných patchů, které by redukovaly stacks v těchto oblastech.

Trochu se diskutovalo o tom, jestli je to pořád problém, ale v konferenci se k žádnému rozhodnutí nedošlo. Na jiném místě řekl Vladimir Saveliev, že se usilovně snaží vyřešit poslední zbývající problémy, které by měl ReiserFS bez 4KSTACKS, a odhadoval, že by patche mohl mít hotové velmi rychle.

## Shrnutí real-time problémů, 103 e-mailů

07. črv – 14. črv

Paul E. McKenney napsal: Někdy během nedávného vlákna „RT patch acceptance“ [přijetí RT patchů] někdo zmínil, že by bylo fajn shrnout všechny ty různé přístupy. Následující text se o to pokouší a zároveň se snaží poskytnout rozumný rámec pro budoucí diskuzi.

Přiložil velmi dlouhý dokument [8], který byl chválen ze všech stran, protože dokázal zachovat vyrovnaný a nezaujatý pohled. Mnoho lidí přišlo s návrhy, jak dokument vylepšit, ale konverzace degenerovala na další flamewar a obviňování. Původní zpráva však stojí za přečtení každému, kdo by se v tom bludišti rád vyznal.

## DevFS odstraněno z Linuxu, 42 e-mailů

10. črv – 16. črv

Greg KH napsal: Jak každý ví [1], devfs bude z jádra brzy odstraněno. Posílám sérii patchů (dohromady 22), které to zařídí. Překvapivě bylo devfs v jádře skoro všude, proto je potřeba tolik patchů :). Pokud se některé do LKML neprobijí (kvůli omezením velikosti nebo prostě neprojdou filtrem „vkusu“), najdete je na:

../gregkh-2.6/gregkh-05-devfs/ [9]

včetně zastřešujícího souboru, pomocí kterého je můžete aplikovat. Andrew, nedávej je prosím do svého -mm stromu. Je dost pravděpodobné, že akorát způsobí tolik konfliktů, že ti to nebude za tu námahu stát :). Komentáře uvítám. Jo, a ta nejlepší část? Tohle je shrnutí diffstat: 222 souborů změněno, 112 vložení(+), 8545 výmazů(-) Je hezké taky jednou pro změnu kód z jádra odstraňovat... Díky, greg k-h [1] Co? Vy to nevíte? Nedostali jste snad připomínku[2]? Nevšimli jste si toho obrovského flamewaru skoro před rokem[3]? Žijete v jeskyni[4]?

[2] feature-removal-schedule.txt [10]

[3] <http://thread.gmane.org/gmane.linux.kernel/219278> [11]

[4] ../shop/images/products/product\_1788\_small.jpg [12] Adrian Bunk řekl: Neodstraňuj prosím ty atrapy !CONFIG\_DEVFS\_FS z devfs\_fs.kernel.h. Jsem si jistý, že někteří správci ovladačů budou chtít tyto funkce ve svém kódu zachovat, protože sdílejí ovladače mezi 2.4 a 2.6.

Ale Greg odpověděl: Všechny ovladače by měly být v hlavním stromě, tak proč by to potřebovaly? Ovladače mimo strom jsou každý za sebe...

Adrian na to řekl: Já ale mluvím o ovladačích v hlavním stromě jádra. V některých případech podporuje autor jak 2.4, tak 2.6 a dává přednost tomu, aby byla ta podpora v jediném souboru. Čas od času pošle nejnovější verzi svého ovladače Linusovi nebo Marcelovi. Když ty atrapy globálních funkcí odstraníš, přinutíš všechny správce ovladačů, kteří takhle pracují, aby je dali do svých ovladačů. Víím, že je hodně míst, kde nejsou 2.4 a 2.6 kompatibilní z dobrých důvodů. Ale je-li udržení kompatibility mezi 2.4 a 2.6 v některých oblastech tak snadné, že stačí podržet hlavičkový soubor s pár atrapami funkcí, stojí to za úvahu. A když se to všechno ponechá v jednom souboru místo rozházení po celých zdrojácích jádra, bude za pár let daleko snazší poslední výskyty vyčistit.

Christoph Hellwig k tomu řekl: DevFS volání pro 2.4 a 2.6 jsou totálně nekompatibilní. Takže nej-jednodušším způsobem podpory jak 2.4, tak 2.6 v této oblasti je: přestat podporovat devfs. Stejně bylo pořád označené jako experimentální nebo zastaralé.

Adrian byl překvapen tím, kolik se toho v DevFS změnilo během 2.5, a stáhl své připomínky. Na jiném místě poslal Greg s předmětem „[PATCH] Odstranění devfs z kódu oddílů [13]“ patche, které DevFS odstraňovaly nejen z kódu oddílů, ale odevšad. V tomto vláknu vyjádřil Armin Schindler své překvapení nad tím, že bude odstranění provedeno uprostřed stabilní řady...

Adrian mu připomněl: Podle současného vývojového modelu je 2.6 vývojovým jádrem...

A Ed Tomlinson připojil Aktuální Linusův kernel je 2.6.11.12, kde poslední .12 značí, že jde o poslední 2.6.11 s aplikovanými VIF (very important fixes = velmi důležité opravy).

Armin reagoval s tím, že si toho je vědom, ale přesto ho překvapuje, že se podobné pozdvižení děje bez zvýšení druhého čísla verze.

## Linux 2.4 nebude podporovat GCC 4, 2 e-mailů

12. črv – 13. črv

Mikael Pettersson napsal: Tahle sada patchů opravuje problémy s gcc4 v kódu „jádra“ jádra 2.4.31. Provozují 2.4 jádra kompilovaná pomocí gcc4 už několik měsíců na i386, x86\_64 a ppc32 a v současné době při srovnání s gcc34 nic nechybí. Poznámka: používejte raději čerstvý gcc-4.0.1, protože o gcc-4.0.0 se ví, že nefunguje správně. Tato sada neobsahuje opravy ovladačů, souborových systémů a architektur, které sám nepoužívám. Mám pro tento účel předběžný patch kit, ale protože se mu dostalo jen omezeného kompilačního testování, nebudu ho posílat, nebudou-li přijaty tyto hlavní patche.

Marcelo Tosatti odpověděl: Řekl bych, že už je čas začít tento druh úprav zamítat. Mohou žít mimo hlavní repozitář.

## Linuxová dokumentace, kterou lze prohlížet jen v nelinuxových prohlížečích, 10 e-mailů

14. črv – 16. črv

Nick Newcomb napsal: Pracuji v Software Revolution a říkal jsem si, že by vás mohlo zajímat, že jsme právě dokončili automatické generování dokumentace k linuxovému jádru a příbuzným subsystémům na úrovni designu. Dokumentace je složena z prolinkovaných obrázků a textových dokumentů pro všechny hlavní sybsystémy a všechna pole i funkce zdrojového kódu. Organizována je podle spletičnosti a umístění v souborovém systému. Zahrnuje linuxové jádro, správu paměti, souborové systémy, bezpečnost, šifrování, inicializaci, ovladače, architekturu a subsystémy pro komunikaci mezi procesy. Kromě toho ji nabízíme za...no, zadarmo. Myslím, že byste ji mohli využít. Pokud si to chcete prohlédnout, jděte na: <http://www.softwarerevolution.com/jeneral/open-source-docs.html> [14] Parag Warudkar řekl, že by tu dokumentaci rád používal, ale stránky jsou přístupné pouze s pluginy pro Windows. Několik ostatních lidí mělo podobné potíže a probírali možná řešení. Kyle Moffett mohl stránky prohlížet v Mac OSX a velmi se mu líbily, ale všiml si, že používají hodně obrázkových map [imagemap]. Navrhl, aby to změnili, což by jeho oblíbeným prohlížečům OmniWeb a Safari usnadnilo práci.

### Odkazy

- [1] <ftp://ftp.kernel.org/pub/linux/kernel/people/akpm/patches/2.6/2.6.12-rc4/2.6.12-rc4-mm2/>
- [2] <http://www.torque.net/sg/sdparm.html>
- [3] <http://people.redhat.com:/heinzm/sw/dmraid/>
- [4] <http://www.gpl-violations.org/>
- [5] <http://www.lulu.com/content/130446>
- [6] <http://www.kernel.org/pub/linux/kernel/people/jgarzik/libata.pdf>
- [7] [http://ftp.kernel.org/pub/linux/kernel/people/mbligh/abat/regression\\_matrix.html](http://ftp.kernel.org/pub/linux/kernel/people/mbligh/abat/regression_matrix.html)
- [8] <http://groups-beta.google.com/group/linux.kernel/msg/5ce63e8782bdd07a%20?hl=en>
- [9] <http://www.kernel.org/pub/linux/kernel/people/gregkh/gregkh-2.6/%20gregkh-05-devfs/>
- [10] <http://lxr.linux.no/source/Documentation/feature-removal-schedule.txt>
- [11] <http://thread.gmane.org/gmane.linux.kernel/219278>
- [12] [http://www.balloonplanet.com/shop/images/products/product\\_1788\\_small.jpg](http://www.balloonplanet.com/shop/images/products/product_1788_small.jpg)
- [13] <http://groups-beta.google.com/group/fa.linux.kernel/msg/%20d3d86f6fa5d1530c?hl=en>
- [14] <http://www.softwarerevolution.com/jeneral/open-source-docs.html>

\*\*\*

---

### Reklamní nabídka [www.praceabc.cz](http://www.praceabc.cz)

Pozice: Configuration Unix Engineer (abroad + Prague) (50000Kč) [ABC00421]

Obor: IT – vývoj systémů Kraj: Neuvedeno Vloženo: 12.05.2005

Pozice: správce linuxového serveru [ABC00419]

Obor: IT – správa systémů Kraj: Hlavní město Praha Vloženo: 10.05.2005

Pozice: Integration Specialist, zn. AIM [ABC00409]

Obor: IT – správa systémů Kraj: Neuvedeno Vloženo: 02.05.2005

## Zprávičky

**1.7.2005**

*Pavel Szalbot ml.*

Týden před hlasováním EP o softwarových patentech vychází na Groklaw článek vysvětlující, jak softwarové patenty ohrožují FOSS. Dozvíte se o stránce EP popisující dosavadní kroky učiněné během jednání/schvalování patentů a také si můžete přečíst řeč Richarda M. Stallmana, ve které se na patenty dívá z pohledu vývojáře.

**1.7.2005**

*Pavel Szalbot ml.*

Sun se na JavaOne zmínil o tom, že jeho plány s Java Desktop System se mění. Původní záměr prodávat systém vystřídaly úvahy o jeho uvolnění pro komunitu jako open source.

**1.7.2005**

*Pavel Szalbot ml.*

GPL už nepotřebujeme – je postavena na domnělé zranitelnosti open source softwaru a nutnosti jeho ochrany. OSS by byl úspěšnější, kdyby jeho osvojení GPL spoustě lidí neztěžovala. Rozhovor s autorem těchto myšlenek, Ericem Raymondem, si můžete přečíst na ONLamp.com.

**1.7.2005**

*Rastislav Stanik*

O hrozbe SW patentov pre malých tvorcov softwaru píše denník Právo.

**1.7.2005**

*Pavel Beníšek*

Vývojáři Gentoo linuxu založili vlastní Gentoo Documentation Project. Najdete jej na adrese [www.gentoo.org/proj/en/gdp/](http://www.gentoo.org/proj/en/gdp/). Projekt má mít zodpovědnost za oficiální dokumentaci dodávanou s Gentoo Linuxem a také za dokončení chybějících částí.

**1.7.2005**

*Robert Krátký*

LinuxInsider píše o tom, co všechno vám projde z hlediska porušování podmínek GPL. Právníčka Heather J. Meeker vysvětluje situaci a jako příklad uvádí firmu Linksys, která se porušením GPL proslavila: Open Source and the Legend of Linksys.

**2.7.2005**

*fipa*

Za 14 dní se v Liberci uskuteční již 4. ročník konference a setkání uživatelů a přátel operačního systému IBM OS/2 – Warpstock 2005 CZ. Na programu jsou ale zajímavé přednášky i mimo svět OS/2 – OpenOffice.org, Java, DocBook apod. Účast je zdarma a všichni jste srdečně zváni, stačí se zaregistrovat.

**3.7.2005**

*Milan Hejpetr*

Poštovní spořitelna od pátku asi oficiálně podporuje i jiné prohlížeče než MSIE. Doufejme, že příkladu budou následovat i některé další banky.

**4.7.2005**

*Pavel Beníšek*

Alias uvedl samostatnou linuxovou verzi svého rendereru pro program Maya pod jménem „mental ray Standalone 3.4 64-bit“. Díky dobré podpoře 64bitové platformy v linuxu profitují uživatelé zvýšeným výkonem. Podrobnější informace naleznete v tiskové zprávě.

**4.7.2005**

*Pavel Szalbot ml.*

Anandtech testoval výkon dual core procesorů několika benchmarky a jako operační systém si k tomu zvolil Linux. Jak si procesory vedly při kompresi souborů, ripování DVD, nebo hraní her?

**4.7.2005**

*Pavel Szalbot ml.*

Malá, spolehlivá, plně vybavená distribuce spustitelná z CD nebo USB klíče s minimálními nároky na operační paměť. Takový je Puppy Linux, jehož recenzi si můžete přečíst na Linux.com.

**4.7.2005**

*Pavel Szalbot ml.*

V IBM developerWorks se objevil článek představující spufs. Jedná se virtuální souborový systém, který zpřístupní jednotlivé SPU (Synergistic Processing Unit) procesoru Cell, a jenž by IBM rádo vidělo v kernelu 2.6.13.

**4.7.2005**

*Leoš Literák*

Stickfish se stal technologickým partnerem OSS Alliance a poskytl server DELL PowerApp 100 včetně konektivity pro potřeby Open Source projektů. Prvním hostovaným projektem je živá distribuce SLAX.

**4.7.2005**

*Pavel Szalbot ml.*

Pokud hledáte jednoduchý nameserver, který nemusí být ani tak rozsáhlý jako BIND (snad o to rychlejší a bezpečnější), zkuste se podívat na NSD, jenž stručně představuje článek na NewsForge.

**4.7.2005**

*klassik*

Na iDNES vyšel vcelku vtipný článek Bude internet i nadále pod kontrolou USA?

**5.7.2005**

*Pavel Szalbot ml.*

Skupina manažerů IT podniků zaslala otevřený dopis open source komunitě, ve kterém děkují za odváděnou práci, a poukazují také na změny po-

třebné k tomu, aby OSS získal ve velkých IT společnostech větší pozornost.

**5.7.2005**

*Pavel Szalbot ml.*

Jeff Johnson, správce RPM, uveřejnil reakci na článek Claudia Matsuoky – Top Ten RPM Problems, ve kterém obohatil diskusi o RPM o zajímavá témata. Informují OSNews.

**5.7.2005**

*Michal Vyskočil*

Velké společnosti, jako Alcatel, nebo Phillips říkají ne současné verzi návrhu evropského patentového zákona. Nelíbí se jim jeho komplikovanost, způsobená množstvím pozměňovacích návrhů a varují, že jeho přijetí může mít negativní dopad na zaměstnanost v Evropě. Píše The Register.

**6.7.2005**

*Jiří Stavinoha*

Byla uvolněna testovací verze nové řady publikačního systému UNITED-NUKE 4.2ms2 Beta. Kompletně přepsaná řada 4.2 méně zatěžuje databázi a je také několikanásobně rychlejší.

**6.7.2005**

*Rastislav Stanik*

Groklaw prináša článok informujúci o aktuálnom dianí okolo direktívy o SW patentoch v EÚ. Prináša tiež linku na zvukový aj video záznam debaty, ktorá sa odohrala v Európskom parlamente.

**6.7.2005**

*Ivan Vyoral*

Výzkum na Ústavu organické chemie a biochemie AV ČR je podporován výkonným linuxovým superpočítačem SGI Altix řady 3700 s procesory Intel Itanium 2. Tisková zpráva.

**6.7.2005**

*Michal Vyskočil*

The Register oznamuje výsledek hlasování o návrhu patentového zákona. Evropský parlament návrh odmítl 648 hlasy ku 14, 18 poslanců se zdrželo. Nezbývá, než dodat, že úsilí FFII a všech lidí, kteří se zapojili do kampaně, nebylo zbytečné.

**6.7.2005**

*Dominik Joe Pantůček*

Evropský parlament dnes drtivou většinou zamítl direktivu o patentovatelnosti software. Toto odmítnutí je logickou odpovědí Evropské Komisi na její nechuť k jakékoliv diskusi o této problematice.

**6.7.2005**

*petr*

Na světě máme Scribus 1.2.2, OpenSource DTP. Obsahuje hromadu oprav, aktualizací apod. Účinuje 1000 slonů!

**6.7.2005**

*Luboš Doležel*

Vyšla Opera 8.02 Technical Preview 1. Ke stažení je dostupná zde. Má obsahovat podporu BitTorrentu.

**7.7.2005**

*Pavel Szalbot ml.*

Městská správa Vídně přechází na open source, nebo alespoň dostala tuto možnost. Radní Rudi Schicker na tiskové konferenci představil distribuci Wienux, která by mohla nahradit stávající softwarové vybavení.

**7.7.2005**

*Pavel Szalbot ml.*

Opera a Trolltech se dohodly na strategickém partnerství. Opera by si pro svůj prohlížeč představovala větší podíl na mobilním trhu a Trolltech její produkt rád integruje do svého aplikačního frameworku Qtopia pro embedded-Linux. Informuje LinuxInsider.

**7.7.2005**

*Pavel Szalbot ml.*

V článku na DesktopLinux.com se dočtete o prvních krocích jednoho strojního inženýra s Linuxem po boku. Pokud zvažujete migrovat váš desktop na Linux, nebo se chcete dozvědět, jaký je Minislack, je článek právě pro vás.

**7.7.2005**

*Pavel Szalbot ml.*

Theo De Raadt, muž stojící v čele týmu vyvíjejícího OpenBSD a OpenSSH, si s Epoch Times popovídal o tom, proč vytváří svobodný software, jakého ohlasu se jeho produkty dočkaly nebo o tom, jak se vyplácí lpět na kvalitě.

**7.7.2005**

*Pavel Szalbot ml.*

Na systhread.net začal vycházet seriál, který by měl ukázat (nejen) zajímavé možnosti knihoven přibalených k správci oken Enlightenment. První díl popisuje „neuvěřitelně rychlou“ knihovnu Epeg, která umí vytvářet náhledy obrázku.

**7.7.2005**

*Pavel Beníšek*

Adobe varoval uživatele unixových operačních systémů, že Adobe (Acrobat) Reader ve verzích 5.0.9 a 5.0.10 obsahuje bezpečnostní chybu umožňující získat kontrolu na OS.

**7.7.2005**

*rajo AT platon.sk*

Na zive.cz informují o novo vzniknutom projekte Swik. Projekt Swik bude databázou open-source projektov. Prispieť k naplňaniu databázy môže každý.

**7.7.2005***Filip Korbel*

Sun začne dodávat i na náš trh silný opteronový výkon v krásném toweru pod názvem Sun Ultra 20. Ceny budou stanovené pro náš trh po 13.7.2005. V USA se základní cena pohybuje na 895,- USD. Sun Ultra 20.

**8.7.2005***Pavel Szalbot ml.*

Na [www.regexlib.com](http://www.regexlib.com) můžete najít knihovnu regulárních výrazů spolu s online aplikací pro jejich testování a několika dalšími nástroji (převážně pro .NET framework).

**10.7.2005***Jakub Krása*

Včera večer byl vydán LFS 6.1 . Obsahuje kromě jiného Linux-2.6.11.12, GCC-3.4.3 a Glibc-2.3.4.

**11.7.2005***Luboš Doležal*

Progeny, Mandriva a Turbolinux chtějí vytvořit novou distribuci založenou na Debian Sarge. Má být určena pro enterprise zákazníky.

**11.7.2005***Pavel Szalbot ml.*

Bjarne Stroustrup v rámci japonského vydání své poslední knihy uveřejnil esej zabývající se používáním C++ v minulé dekádě a odhalující pravděpodobný charakter dalších revizí oblíbeného jazyka. Chcete-li se dozvědět více, stáhněte si PDF.

**11.7.2005***Pavel Szalbot ml.*

Free Software Magazine přináší hezký článek o virtuálních počítačích, virtuálních web serverech a jejich spojitostech. Nahradí v budoucnu virtuální počítače sdílený či dedikovaný web hosting?

**11.7.2005***Lukáš Zapletal*

Červencové číslo časopisu LinuxEXPRES je na stáncích. Z obsahu: OpenOffice.org 2.0 (speciální příloha), Debian GNU/Linux 3.1, SUSE 9.3, Mandriva 2005 LE, Slax 5.0, Knoppix 3.9 nebo třeba praktický článek o diskových kvótách.

**12.7.2005***Pavel Szalbot ml.*

Čtvrtá řada PHP se rozrostla o novou verzi 4.4.0 (seznam změn). Toto vydání neobsahuje žádné novinky, opravuje pouze několik chyb, z nichž nejvýznamnější byla oprava chybné práce s pamětí, která zapříčinila změnu interního API, a proto i zvýšení prostředního čísla verze.

**12.7.2005***Martin Tesař*

Stručný přehled Gentoo komunity přináší server LWN.net. Tito „sektáři“ :) si např. jen na fo-

rums.gentoo.org vyměnili už na 2,5 miliónu příspěvků.

**12.7.2005***Pavel Szalbot ml.*

Iniciativy pro podporu používání svobodného software v Africe neutuchají. Projekty jako Go Open Source, Meraka Institute a African Institute for Mathematical Sciences připravily pro africké matematiky kurz, jenž je seznámí s FOSS pro vědecké účely pomocí distribuce Ubuntu. Více na [tectonic.co.za](http://tectonic.co.za).

**12.7.2005***Ivan Bibr*

Na organizaci dalšího ročníku olomoucké linuxové konference LinuXchage 2005 se bude podílet i společnost pro podporu free software Liberix, o. p. s.

**13.7.2005***Pavel Szalbot ml.*

Vyšel Firefox 1.0.5 opravující několik bezpečnostních chyb a zlepšující stabilitu prohlížeče. Česká verze bude jako tradičně následovat v nejbližších hodinách.

**13.7.2005***Pavel Szalbot ml.*

Claudio Fontana píše na NewsForge o svém grafickém nástroji GNU Source Installer, který by měl uživateli ulehčit instalaci aplikací ze zdrojových kódů vytvořených pomocí GNU autotools. Ve zkratce popisuje jeho dosavadní možnosti a upozorňuje na (zatím) neúplnou funkcionalitu.

**13.7.2005***Robert Krátký*

Zajímavá diskuze o tom, jestli je možné vydělávat peníze psaním open source software, probíhá na [TheServerSide.com](http://TheServerSide.com). jejím iniciátorem byl Joseph Ottinger, který upozornil na zápis v blogu Marca Fleuryho: From Volunteer Open Source to Professional Open Source.

**13.7.2005***Michal Jurosz*

Po dalším měsíci práce vyšel Pugs 6.2.8, který udává nový směr vývoje. Také díky němu je na poli Perlu 6 a Parrotu stále rušněji.

**13.7.2005***Robert Krátký*

Skupina LinuXchange vyhláší akci Call for Papers (hledání přednášejících) pro třetí ročník LinuXchange, moravské konference uživatelů svobodného a open-source softwaru. Konferenci budou mediálně podporovat portály LinuxBIZ a [abc-linuxu.cz](http://abc-linuxu.cz).

**14.7.2005***Pavel Szalbot ml.*

Jak už tomu u sourozenců bývá, novou verzi Firefoxu doplnil v zápětí Thunderbird 1.0.5. Vydání je, podobně jako u Firefoxu, zaplátové a mělo by taktéž zvýšit stabilitu aplikace.

**14.7.2005***Pavel Szalbot ml.*

Stuart Cohen, šéf Open Source Development Labs, oznámil otevření kanceláře OSDL v Lucembursku. Nově zvolený ředitel evropské pobočky, jehož úkolem bude rozšiřovat povědomí o open source v Evropě, Africe a na Blízkém východu, poznamenal, že otevření demonstruje zralost a lesk Linuxu a open source technologií ve světě. Informuje eWeek.

**14.7.2005***Pavel Szalbot ml.*

Sun plánuje uvolnit zdrojové kódy své webové technologie „single sign-on“ (jediné přihlášení) a kódy, které jej implementují, v aplikačním či webovém serveru Sunu, v rámci projektu Open SSO. Tato technologie uživatelům umožňuje snazší přihlašování do různých webových aplikací najednou a provozovatelům zase usnadňuje správu uživatelských účtů. Více o plánu Sunu se dočtete na ZDNetu.

**14.7.2005***Robert Krátký*

První verze licence, kterou by chtěla Evropská unie používat pro distribuci softwaru, který spravuje, je k dispozici k připomínkování.

**15.7.2005***Pavel Beníšek*

John Carrol se na svém blogu zamýšlí nad možnostmi, že bude Microsoft Office 2007 uvedeno i pro GNU/Linux.

**15.7.2005***petr*

Pár dní po zveřejnění zástupce ze stabilní a produkční verze DTP programu Scribus, s hrdostí oznamujeme, že nespíme na vavřínech, a světu ukazujeme to, co přináší brzká budoucnost. Vývojová řada 1.3.x.

**15.7.2005***Robert Krátký*

Slahdot i The Register informují o oznámení, ve kterém IBM potvrzuje konec podpory OS/2 (ke konci 2006) a zároveň přikládá návod k migraci na Linux.

**15.7.2005***Robert Krátký*

CNet komentuje interní email zaměstnance SCO z roku 2002, ve kterém se píše, že analýza linu-

xového jádra neodhalila žádný kód zkopírovaný z Unixu. Viz také článek na Groklaw.

**15.7.2005***Reklama*

Staňte se prvními zákazníky stanice Sun Ultra 20 Workstation ve střední Evropě. Stickfish přináší jako první na český trh nové x86-64 stroje Sun. Více informací o AMD Opteronu si můžete přečíst nebo rovnou objednat na 64bit.cz.

**15.7.2005***Jiri Popek*

Vyšla nová verze offline slovníku pDict. Novinek je hodně. Těším se na připomínky a nápady.

**18.7.2005***Vladimír Slavík*

Na této stránce jsem založil fórum o aplikaci jazyka C++ v prostředí operačních systémů odvozených od OS Unix. Mělo by sloužit k řešení problémů, výměně zkušeností a znalostí o C++ v českém jazyce.

**18.7.2005***Pavel Szalbot ml.*

Stane se příští rok rokem Linuxu? Ať už by to znamenalo např. velký přesun uživatelů k Linuxu, pravděpodobně se tak nestane. Důvody naznačuje článek na ABC News.

**18.7.2005***Pavel Szalbot ml.*

Každá větší aplikace má v rámci své třídy specifické uživatelské rozhraní a výjimkou není ani OpenOffice.org. LinuxJournal píše o tom, na co uživatele zkušenostmi z jiných balíků/editorů naráží při práci s OO.org.

**18.7.2005***Pavel Szalbot ml.*

Krátká recenze Fedora Core 4 dnes vyšla na NewsForge. Taky se vám zdá, že sudé verze distribucí Red Hatu mívají více otevřených ran než jizev? Recenze (či spíše náhled) vás možná vyvede z omylu...

**18.7.2005***Pavel Szalbot ml.*

Potřebuje open source software místo, kde by IT manažeři mohli najít informace, které je přesvědčí o jeho kvalitách, a kdo by jej měl spravovat? Úvahu na téma linuxová kampaň „Get the facts“ si přečtete na sys-con.com.

**18.7.2005***Vojtěch Hála*

Dva a půl roku vězení plus konfiskace majetku za podvody s nigerijskými spamy. Informuje BBC a Slashdot.

**18.7.2005***Leoš Literák*

Jako první na českém internetu přinášíme informaci o prodeji Email.cz společnosti Seznam. Převod domény a tím de facto prodej byl uskutečněn v pátek 15.7.2005 (viz Nic.cz). Ve čtvrtek vám server LinuxBIZ.cz přinese rozhovor s Janem Makovičkou, výkonným ředitelem Impression Media, který dříve působil jako obchodní ředitel právě v Email.cz.

**18.7.2005***Robert Krátký*

Epiacenter.com upozorňuje na podivný vývoj situace linuxových ovladačů od firmy VIA. Nepochopili, co je open source, nebo šlo jen o (nepovedenou) marketingovou věc?

**19.7.2005***Pavel Szalbot ml.*

LinuxDevCenter.com vydalo článek popisující tři iniciativy OSDL pro rozšíření používání Linuxu v různých průmyslových oblastech. Ve zkratce se dozvíte o projektech Carrier Grade Linux, Data Center Linux a Desktop Linux

**19.7.2005***Pavel Szalbot ml.*

TuxMachines se podívaly na v sobotu uvolněnou (a oficiálně zatím nepředstavenou) betaverzi budoucího Mandriva Linuxu 2006 (Beta 1). Tváří se údajně stabilně a svižně, významné novinky však čekat jako obvykle až na poslední chvíli.

**19.7.2005***Jozef Vondrák*

Zdá se, že jsou nově uvolněny verze KBasicu personal a professional edition. Jsou ke stažení zde. KBasic má být na rozdíl třeba od Gambasu multiplatformní (Linux/Mac/Windows) a na 100% kompatibilní s Visual Basicem (v.6). Koho Basic štve, může to dát najevo svým hlasováním na stránkách projektu.

**19.7.2005***Stanislav Fiala*

Vyšel nový SLAX 5.0.6 s kompletním KDE 3.4.1 a KOffice 1.4.0. Změny vzhledem k minulé verzi. Ke stažení na [slax.linux-live.org](http://slax.linux-live.org).

**20.7.2005***Pavel Szalbot ml.*

Vyšel Firefox a Thunderbird verze 1.0.6 včetně lokalizací a s opraveným API – rozšíření by teď měla fungovat.

**20.7.2005***Pavel Szalbot ml.*

KernelTrap informuje o projektu The Linux Kernel Performance Project, který jehož cílem je mapování výkonu jednotlivých verzí kernelu. Na

webu projektu jsou zatím k dispozici výsledky jader 2.6.9-2.6.12 na třech různých procesorech.

**20.7.2005***Pavel Szalbot ml.*

Jaký notebook byste doporučili pro Linux? Podobně se uživatelé často ptají a v diskusích najdou zpravidla spoustu tipů. NewsForge nicméně hlavně pro začínající uživatele a pro ty, kteří plánují první notebook, připravilo ucelený článek, který při výběru určitě poradí.

**20.7.2005***Jan Kučera*

Právě dochází ke slučování portálů OO.o.cz a OpenOffice.cz.

**20.7.2005***Patrik Sychra*

Vyšla nová verze all-in-one bezpečnostního řešení Astaro Security Linux V6. Nové vlastnosti: Transparent Firewall mode, podpora Linux Kernel 2.6, Policy-Based Routing, časové intervaly, SIP Proxy, transparentní SMTP Proxy a další. Distributorem pro ČR je Annex NET. Astaro je zdarma pro domácí uživatele!

**21.7.2005***Leoš Literák*

LinuxBIZ přináší rozhovor s Janem Makovičkou o Linuxu a prodeji email.cz seznamu.

**21.7.2005***Pavel Szalbot ml.*

KDE 3.4.1 se stalo prvním moderním desktopovým prostředím, zkompilevaným (mimo gcc i kompilátorem ze Sun Studio 10), zabaleným a plně fungujícím na platformě OpenSolaris. Na [dot.kde.org](http://dot.kde.org) se k vydání vyjádřil hlavní strůjce úspěchu, Stefan Teleman.

**21.7.2005***Pavel Szalbot ml.*

Asa Dotzler z Mozilla Foundation na ZDNetu analyzuje svůj názor na nepřipravenost Linuxu pro desktopové použití. Hovoří o čtyřech problémových bodech – migraci z jiného systému, stabilnímu API, jednoduchosti a komfortu.

**21.7.2005***Leoš Literák*

Technet píše o levném čtyřjaderném procesoru ARM11 se spotřebou maximálně 600 miliwattů. Procesor je dodáván s linuxovým SMP kernelem řady 2.6.

**21.7.2005***Pavel Szalbot ml.*

Chcete-li se seznámit s jedním z neopěvovaných unixových hrdinů, makroprocesorem m4, jinak než čtením dokumentace, můžete nahlédnout do online „magazínu“ Linux, Unix, /etc/.



**21.7.2005***gabo*

Na svete je kniha Linux pre začínajúcich užívateľov, ktorá na svojich 54 stranách poskytuje rýchlokurz základov Linuxu. Bola napísaná ako pomoc pre ľudí, ktorí majú v školách a zamestnaní linuxové servery. Bližšie informácie na dentalportal.sk.

**21.7.2005***Michal Vyskočil*

Stydí se Sun za to, že prodává Linux? Nebo jak jinak vysvětlit to, že ze své oficiální zprávy, o úspěšném prodeji 30 serverů Sun Fire V20z, po nějakém čase odstranil větu „running a standard Linux operating system“. Informuje The Register.

**21.7.2005***Standa*

Problém BPL/PLC se týká široké veřejnosti, proto prosím věnujte pozornost petici na stránkách radioamatérů. OK1VDX

**21.7.2005***Michal Vyskočil*

IBM se rozhodlo podpořit open source implementaci J2SE, projekt Harmony, píše zdnet.

**22.7.2005***Pavel Szalbot ml.*

Na ONLamp.com se objevil relativně dlouhý rozhovor s Colinem Percivalem, který prezentoval dosud nevyvrácenou práci o bezpečnostních rizicích v oblasti kryptografie ve vícevláknovém prostředí. Řeč byla samozřejmě o samotné práci, bezpečnosti, FreeBSD (Colin je členem týmu FreeBSD Security) a také o tom, jak se k problému stavěly různé IT společnosti a organizace.

**22.7.2005***Tomáš Oberhuber*

Minulý týden byla také hacknuta webová stránka SpreadFirefox.com. To firmu Mozilla, která používá bezpečnost jako hlavní prodejní atribut, taktéž neukázalo v tom nejlepším světle. PCTuning

**22.7.2005***Pavel Szalbot ml.*

Pro zručnější uživatele se na Linux.com objevil návod, jak získat opravdový poklad aneb jak dostat PC s VIA EPIA ME6000 a Linux do pirátské truhlice.

**22.7.2005***Juraj Václavík*

Pro zájemce o KDevelop je na vaclavik.dobris.net překlad tutoriálu Richarda Crooka.

**23.7.2005***Pavel Szalbot ml.*

OSNews upozornily na aktualizovanou verzi dokumentu o architektuře Cell od Nicholase Blachforda. Popisuje teď současný známý stav architek-

tury a byl doplněn i o sekci zabývající se programováním pro ni.

**25.7.2005***Leoš Literák*

Měli byste chuť jít na sraz čtenářů Abička? Kde by se letos měl konat? Přispějte do diskuse svým názorem.

**25.7.2005***Pavel Szalbot ml.*

Článek na developerWorks uživatelům přibližuje operační systém FreeBSD. Věnuje se historii v podobě 386BSD, představuje odlišnosti od linuxových systémů, systém balíčků, zmiňuje bezpečnost a dokumentaci. Vhodné čtení pro ty, kteří jej chtějí zkusit.

**25.7.2005***Pavel Szalbot ml.*

LiveLAMP je linuxová distribuce zaměřená na snadnou instalaci a provozování LAMP (Linux + Apache + MySQL + Perl/PHP/Python). Učitelům základních a středních škol má poskytnout platformu pro seznámení žáků s tvorbou webových prezentací a aplikací.

**25.7.2005***Pavel Szalbot ml.*

Detailní popis instalace a konfigurace Fedora Core 4 na server (Apache 2.0.x, Postfix, BIND9, proftpd, POP3/IMAP a Webalizer) oplývající mnoha screenshoty najdete na HowToForge.

**25.7.2005***Pavel Szalbot ml.*

Novodobé 3D hry a Linux k sobě, ať už kvůli nedostatečné podpoře grafického hardware ze strany výrobců, či chybějícím portům her, zatím příliš nepatří. Přesto se na Phoronixu můžete přesvědčit, jak si nejvýkonnější kousek nVidie, totiž GeForce 7800GTX, vede v Linuxu.

**25.7.2005***Jozef Vondrák*

Klíčenka s předinstalovaným Linuxem, na kterém lze spustit kancelářské i jiné aplikace, je celkem běžná. Pokud chcete navíc trochu bezpečí, můžete použít tuhle verzi se snímačem otisku prstu. Nakonec, pokud nechcete Linux, ale chcete pouze svá data chránit otiskem prstu, můžete zkusit toto. Stránky taktéž navozují pocit, že Linux běží přímo na klíčenkách. Ale myslím, že jde pouze o pocit.

**25.7.2005***Lukáš Petrovický*

Dnešního dne se na webu CZilla objevilo 100. lokalizované rozšíření pro programy Mozilla. Velký dík patří všem lokalizátorům. Pokud se k nim

chcete také přidat, přečtěte si články Překládáme rozšíření a Jak správně lokalizovat a napište nám.

**25.7.2005**

*Michal Čihař*

Máte přebytek energie a chcete udržovat nějaký free software? Nebo naopak víte o nějakém, který by si zasloužil správce, a nemá ho? V tom případě je tu pro vás server [www.unmaintained-free-software.org](http://www.unmaintained-free-software.org).

**25.7.2005**

*Zdeněk Štěpánek*

Pořadatelé Linux Meetingu Děčín 2005 znovu vyhláší Call for papers. Hledáme přednášky o Linuxu, OpenSource a moderních IT technologiích souvisejících s Linuxem apod. Akce je určena všem zájemcům o GNU/Linux. Začátečnickům i expertům. Podívejte se na [linuxdecin.gavanet.org](http://linuxdecin.gavanet.org)

**26.7.2005**

*Leoš Literák*

Mobil.cz píše o analýze, jak si stojí Linux v mobilech. Údajně má skoro 14 %, zatímco Windows jen 5 %. Trhu vládne Symbian se 76 %.

**26.7.2005**

*Robert Krátký*

Joachim Breitner vyzval k zapojení do iniciativy Utnubu, která se bude snažit začlenit novinky z Ubuntu (zpátky) do Debianu. Seznam balíčků z Universe Ubuntu, které nejsou v Debianu, najdete na [utnubu.alioth.debian.org](http://utnubu.alioth.debian.org).

**26.7.2005**

*Robert Krátký*

Díky Opteronu se firmě AMD podařilo ukrojit desetiprocentní koláč z trhu x86 serverů. Situaci (Intelu) hodnotí The Register.

**26.7.2005**

*Robert Krátký*

Příběh vzniku internetového portálu založeného na Linuxu popisuje Případová studie (Alliance Unichem CZ s.r.o.) – Alliance UniChem CZ s.r.o. je součástí společnosti Alliance UniChem plc, která se řadí mezi tři největší distributory léčiv v Evropě.

**26.7.2005**

*Pavel Szalbot ml.*

Redhat Magazine nabízí náhled do světa 64bitových procesorů v Linuxu. Popisuje logické důsledky a komplikace, které používání 64bitových procesorů zapříčinilo (např. poněkud zvrácenou hierarchii souborového systému některých distribucí).

**27.7.2005**

*František Dvořák*

Právě vyšla nová verze multimediálního přehrávače xine. Bylo opraveno mnoho drobných chyb v enginu i v uživatelském rozhraní. xine lze již

také zkompileovat s gcc4. Byly aktualizovány používané knihovny ffmpeg a goom, přibylo také několik modulů na dodatečné zpracování videa.

**27.7.2005**

*Pavel Szalbot ml.*

Rink Springer oznámil vytvoření plně funkčního (nefunguje jen ethernet) portu FreeBSD (5.4 a 6.0-BETA1) pro Xbox. Dostupný je také v podobě LiveCD.

**27.7.2005**

*Pavel Szalbot ml.*

Asa Dotzler ve svém blogu navazuje na zápis o nepřipravenosti Linuxu pro použití na desktopu, jenž vyšel i na ZDNetu [zprávička]. Tentokrát se věnuje výlučně problémům s instalací software souvisejícím se stabilitou API.

**27.7.2005**

*Pavel Szalbot ml.*

Jak je na tom IDE Eclipse s podporou vývoje enterprise Java aplikací? Gerard Fernandes na DevX.com poukazuje na nedostatky Eclipse při práci s J2EE technologiemi a není jich málo (omezená práce s Ant skripty, chybějící podpora pro XML, XSL, SOAP WSDL, JSP, svérázné uživatelské rozhraní). Fundovaně se tvářící autor však v závěru poněkud pochybil zmínkou o tom, že čeká na Eclipse 3.1, které slibuje částečné změny...

**27.7.2005**

*Pavel Szalbot ml.*

75 milionů kopií Firefoxu, Thunderbird se blíží k deseti milionům, Minimo (prohlížeč pro mobilní zařízení) v prealfa verzi s panelovým prohlížením, správou bookmarků a čtečkou RSS, zvýšení počtu zaměstnanců. Tak si ve zkratce vede Mozilla Foundation, jak informuje ZDNet.

**27.7.2005**

*Ondřej Surý*

Na [www.ubuntu.cz](http://www.ubuntu.cz) byly spuštěny první (beta) verze českých stránek Ubuntu Linuxu. Tímto vyzývám všechny uživatele Ubuntu Linuxu, kteří mají chuť a čas, aby se připojili k tvorbě těchto stránek. Zároveň s potěšením oznamuji, že společně s CZLUGem, začínáme projekt distribuce Ubuntu CD (Install + Live) v rámci Ubuntu ShipIt systému. Pro začátek je k dispozici 20 CD, objednávky přijímáme na [poslimi@ubuntu.cz](mailto:poslimi@ubuntu.cz).

**27.7.2005**

*Pavel Szalbot ml.*

Nainstalovat starší počítač jako shaper síťového provozu vám pomůže řada miniaturních disketových distribucí. Na FreeBSD založené PicoBSD (s shapovacím nástrojem dummynet) popisuje [linux.com](http://linux.com).

**27.7.2005***Aleš Kapica*

Polské Ministerstvo pro vědu a informatiku se rozhodlo finančně dotovat překlad OpenOffice.org, což pobouřilo část veřejnosti, které se nelíbilo, že prý jsou tímto způsobem vynakládány veřejné prostředky ve prospěch soukromé firmy. V rámci polské mutace ComputerWorldu byl zveřejněn rozhovor s ředitelem oddělení informatiky tohoto ministerstva Dariuszem Boguckim, ve kterém vysvětluje, že tento počín je součástí dlouhodobé strategie přechodu polských státních institucí na open source podle programu IDABC.

**27.7.2005***Josef Vybíral*

Dnes vyšla verze 0.42 vektorového grafického editoru Inkscape. Tato verze přináší novinky jako možnosti úprav barevných přechodů přímo na pracovní ploše, podporu klonování objektů, vylepšenou práci s textem a další. Celý seznam změn a novinek si lze přečíst v poznámkách k vydání. Tato verze je také konečně dostupná i v češtině. Zatím sice pouze rozhraní programu, ale na překladu tutoriálů se pracuje.

**27.7.2005***Filip Bartmann*

K dispozici je vývojová verze Linux from Scratch (distribuce kompletně kompilovaná ze zdrojových kódů podle návodu) s kompilátorem GCC 4.0.1 na adrese [www.linuxfromscratch.org](http://www.linuxfromscratch.org).

**28.7.2005***Boomsoft*

Vyšlo nová verze grafického prostředí KDE. KDE 3.4.2 už je na FTP. Na oficiálních stránkách zatím není informace o změnách a o vydání nové verze.

**28.7.2005***Pavel Szalbot ml.*

Vyšel Vector Linux 5.1 Standard. Hlavní novinkou na Slackwaru založeného systému je možnost správy balíčků s kontrolou závislostí pomocí slapt-get a Gslapt-gui.

**28.7.2005***Pavel Szalbot ml.*

Na ZDNetu se objevil hezký článek soustřeďující se na unixovou filosofii a připomínající její existenci v moderních systémech. Nezapomínejme, že Unix je soubor myšlenek, mnoho vývojářů a různě zaměřených variant.

**28.7.2005***Michal Čihař*

Na světě je nový Abiword 2.2.9. Změny jsou převážně pro Mac OS, ale byly opraveny i další drobnosti. Stahujte zde.

**28.7.2005***Michal Čihař*

Linus importoval kompletní vývojářský strom kernelu 2.6 do gitu. Pokud se neobjeví problémy, bude brzy následovat i 2.4.

**28.7.2005***Robert Krátký*

Na LinuxBIZ.cz vyšel rozhovor s Vlastimilem Pečínkou, který byl po dlouhou dobu hlavním vývojářem služby Email.cz. V současné době pracuje v Seznam.cz. Vlastimil Pečinka nejen o freemai-lech v Čechách.

**29.7.2005***Pavel Szalbot ml.*

Máte-li velmi rychle bootující distribuci a chcete-li porovnat rychlost s ostatními a zveřejnit úpravy, které vám pomohly, zapojte se do soutěže o nejrychleji bootující systém. Inspirací vám může být i článek, kterak optimalizovat Fedoru pro rychlejší zavedení.

**29.7.2005***Pavel Szalbot ml.*

Mezi PHP vývojáři se často traduje, že objektový model je v PHP neohrabaný, pomalý a v podstatě téměř zbytečný. Zbořit tento mýtus a ukázat jeho výhody se tentokrát pokouší článek na ONLamp.com.

**29.7.2005***Pavel Szalbot ml.*

Chovejte se dnes k vašemu systémovému administrátorovi slušně a neostýchejte se mu věnovat drobnou pozornost. Poslední pátek v červenci totiž (někteří) slaví už pošesté svůj Den systémových administrátorů ;-).

**30.7.2005***Leoš Literák*

Náš portál přechází na bugzillu. Chyby a drobná vylepšení reportujte v ní. Více v mém blogu.