



## Abíčko

Časopis serveru AbcLinuxu.cz

Březen 2006



---

Vychází také na CD-ROM jako příloha časopisu



# Editoriál

Vítejte u čtení časopisu Abíčko.

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

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

Na <http://www.abclinuxu.cz> najdete rozsáhlou databázi návodů na zprovoznění hardwaru pod Linuxem, velice aktivní diskusní fórum, podrobné návody a tutoriály, recenze, archiv ovladačů, informace o linuxovém jádře (včetně populárních Jaderných novin) i rozcestník po ostatních linuxových serverech. Novinkou posledních měsíců, která našla brzy odezvu, jsou blogy neboli internetové deníčky. Každý registrovaný uživatel si jej může založit a psát si do něj poznámky nejen o Linuxu.

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

Náměty na články zasílejte do konference našich autorů: [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 konfigurací síťových služeb. Více na <http://www.stickfish.cz>.

©2006 Stickfish s. r. o. a autoři článků

Editor a sazba: Vlastimil Ott

Pro nekomerční účely smíte tento dokument jakkoliv šířit v tištěné i digitální podobě. V ostatních případech nás požádejte o svolení na adrese [info@abclinuxu.cz](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>Dinosaurius žere Žábu</b>	<b>4</b>
Požadavky a instalace	4
Jak se používá	4
Cena a podpora	6
Závěrem	7
<b>Hibernace</b>	<b>8</b>
Úvod	8
Začínáme:	8
Nastavenie jadra	8
Konfigurácia	9
Nastavenie hibernácie	9
<b>GNU GPL v3</b>	<b>11</b>
DRM	11
<b>Gentoo Linux – majstrovská inštalácia</b>	<b>13</b>
Základné kroky inštalácie	13
Portage	13
/etc/make.conf	14
/etc/env.d/*	14
/etc/init.d/*	14
/etc/conf.d/*	14
USE flagy	14
Záver	15
<b>Gentoo Linux – partície</b>	<b>16</b>
Portage tree	16
Bootovacia partícia	16
Loopback device	16
Swap? Už nikdy viac!	17
Záver	17
<b>Gentoo Linux – optimalizácie</b>	<b>18</b>
CFLAGS, CXXFLAGS	18
Ako funguje GCC	18
Optimalizácia	18
Inštrukčné sady	19
LDFLAGS	19
<b>Gentoo Linux – rýchlosť kompilácie (ccache a distcc)</b>	<b>21</b>
ccache	21
distcc	21
Záver	22
<b>Real-time modifikace Linuxu</b>	<b>23</b>
1. ÚVOD	23
1.1 Co je to Linux	23

1.2 Probírané linuxové real-time modifikace	23
1.2.1 RTLinux	23
1.2.2 RTAI/fusion	23
1.2.3 Realtime preempt patch	23
2. RTLINUX	24
2.1 Úvod	24
2.2 POSIX	24
2.2.1 Základní pojmy	24
2.2.2 Plánování	25
2.2.3 Posixio	25
2.2.4 Signály	25
2.3 Shrnutí	25
3. RTAI	26
3.1 Úvod	26
3.1.1 Úvod do RTAI/fusion	26
3.2 Časovače a přerušení	26
3.3 Základní koncepty	27
3.4 Software	27
3.5 Podporované architektury	27
4. REALTIME PREEMPT MODIFIKACE JÁDRA	27
4.1 Souběžnost v jednoprocessorovém jádře	28
4.1.1 Kritické úseky a jejich správa	28
4.2 Souběžnost na SMP jádře a správa kritických úseků	28
4.2.1 Ochrana před jinými vlákny	29
4.3 Analýza prodlev preemptivního jádra	29
4.3.1 Prodlevy přerušení	29
4.3.2 Prodlevy závislé na úlohách	29
4.3.3 Shrnutí	29
4.4 Plně preemptivní real-time jádro	30
4.4.1 Vylepšení real-time zpracování úloh	30
4.4.2 Nezávislost kritických úseků	30
4.4.3 Nahrazování Spinlocku	30
4.4.4 Shrnutí	30
4.4.5 Vlastnosti Real-Time jádra	31
Real-time mutex	31
4.5.1 Dědičnost priorit	31
4.6 Benchmarky	31
4.6.1 Interbench	31
4.7 Shrnutí	33
5. APLIKACE	33
5.1 Embedded systémy a jiné aplikace pro řízení	34
5.2 Multimediální aplikace – JACK	34
6. Závěr	34
<b>Zprávičky</b>	<b>36</b>

# Dinosaurus žere Žábu

Petr Vaněk

---

*Člověk, který vyvíjí v PL/SQL, nemá na Linuxu na různých ustláno. Sice má k dispozici celou řadu kvalitních textových editorů, ale velice nedostatečně jsou zastoupeny nástroje DBA, popřípadě DB vývojáře. Co vybrat?*

---

Sqlplus, TOra, modul do Eclipse nebo modul do vimu? Sqlplus je skvělý na spouštění hotových skriptů, modul do vimu jen a pouze volá sqlplus, stejně tak, jako je možné třeba z kate. Zmlsaný jedinec navíc vyžaduje zobrazenou hierarchii objektů, pomocné nástroje správy, debugger apod.

Zatím je na UN\*Xových strojích možno provozovat program [TOra](#) [1], anebo něco z hromádky mrtvých projektů, jež lze nalézt např. na Sourceforge, pokud alespoň jejich vývojáři vydali nějaké soubory. Suma sumárum: žádná sláva.

TOru se snažím používat, ale má pro mne jednu zásadní nevýhodu. Je nepřehledná. Je tak nepřehledná, že její spřízněnost s [Toad](#) [2]em, ve světě Oracle nejpopulárnějším žabákem, je očividná. On vlastně i původní autor pracuje pro Quest Software. Čtenář už jistě pochopil, že nerad používám obojí software. Oba mají natolik přeplácené rozhraní, které se navíc mění verzi od verze, že v menším rozlišení už v nich není místo na vlastní zdrojový kód – kdo je nezná, ať si spustí kdevelop s implicitním nastavením.

Netuším, jestli si stávající situaci uvědomili v Oracle, ale v podstatě je mi to jedno, každým pádem je k dispozici nový nástroj [Raptor](#) [3]. Tento je zatím v plenkách, tedy ve fázi veřejného testování. O pár postřehů se s vámi podělím.

## Požadavky a instalace

Fungující X Window System a Sun JDK 1.5. Toť vše. Tedy žádný Oracle klient, ani Instant client. Nejsou zapotřebí ani administrátorská práva k vlastní instalaci Raptoru, protože se tento zatím jenom rozbálí kamkoli na disk a spouští se rovnou z tohoto adresáře. Hodně příjemný a pro testy užitečný přístup.

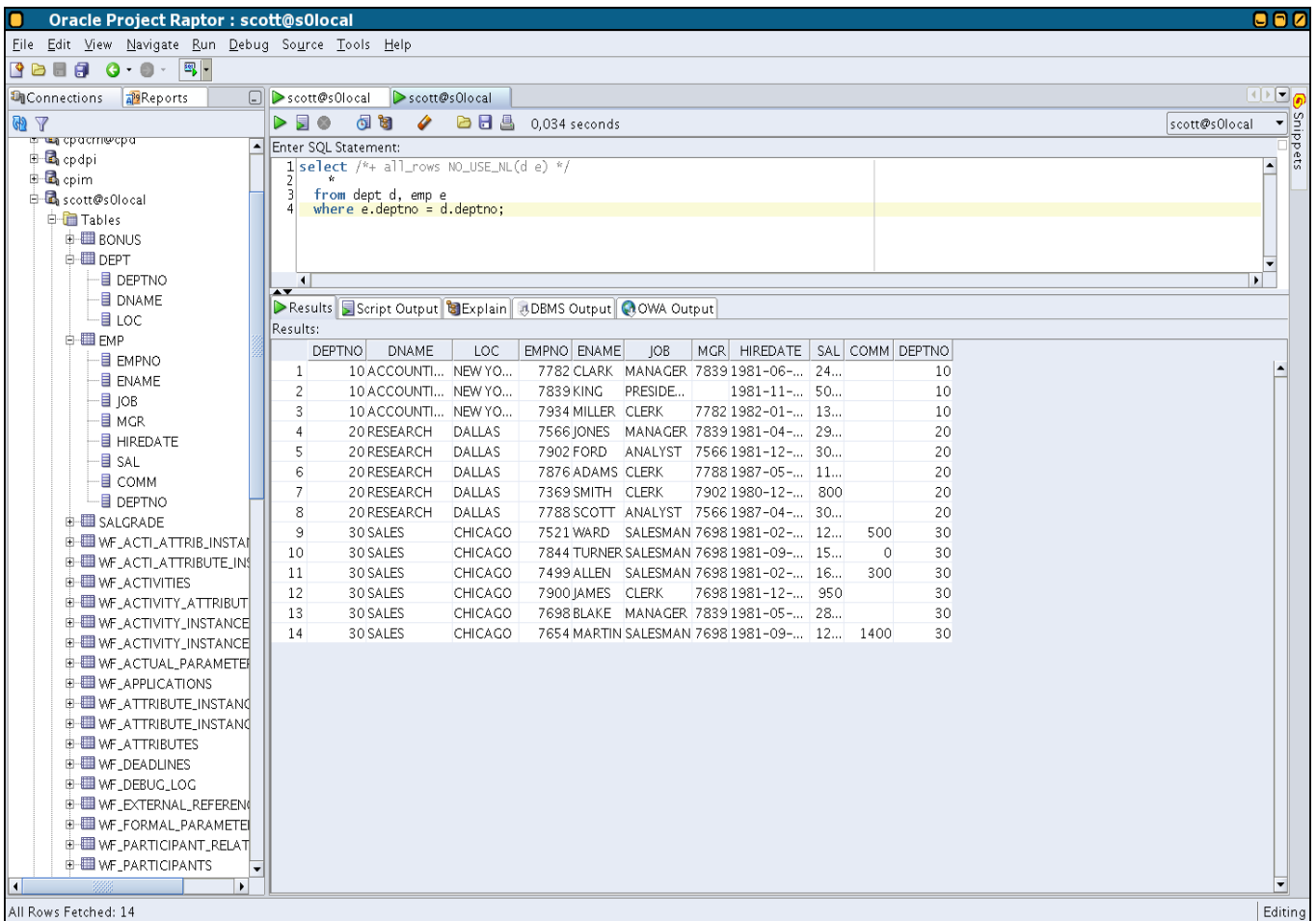
Co se podpory Oracle serverů týče, lze se připojit pouze na 9 a vyšší. Respektive s 8 Raptor běží také, ale nelze využít všechny jeho vlastnosti (nefunkční sestavy, explain plan atd.) Protože je program implementován v Javě, lze použít téměř kdekoli. Mám jej vyzkoušený na SUSE a Gentoo Linuxu a MS Windows XP.

## Jak se používá

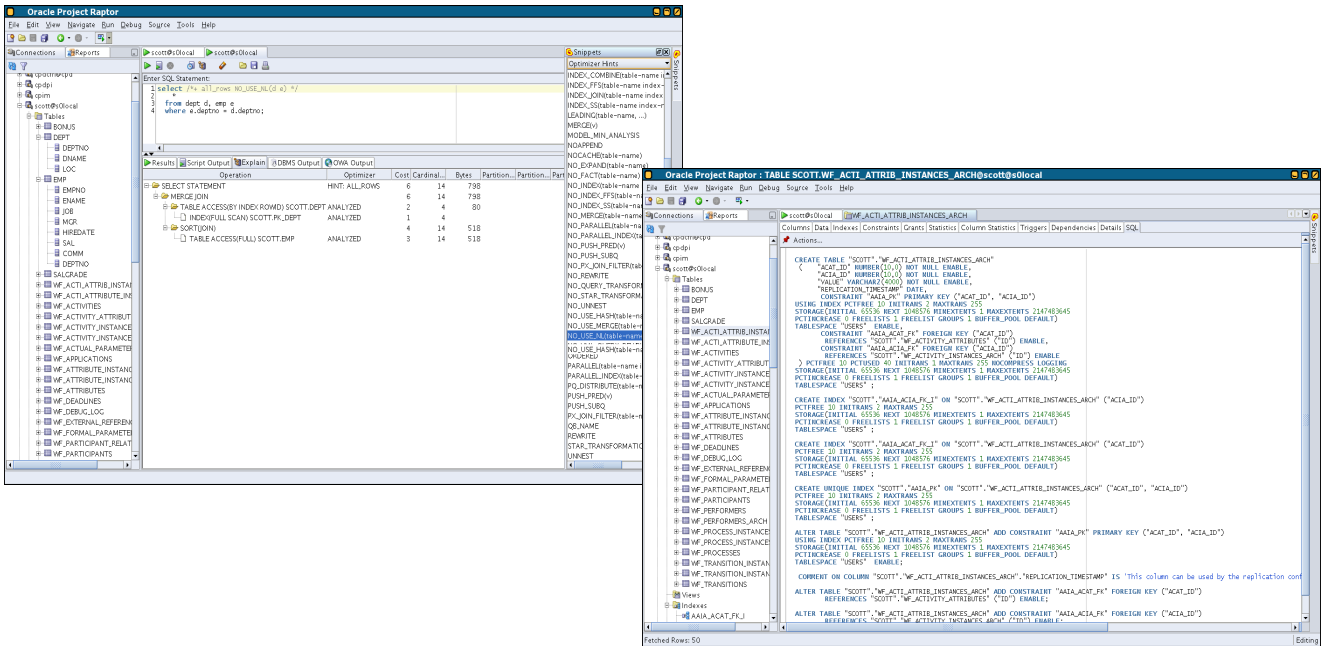
První pohled prozradí, že Raptor vypadá tak, jak jsme u databázových IDE zvyklí. Strom objektů, SQL editor, tabulka výsledků nebo jiných výstupů. Ovšemže je tu něco jinak.

První a nejdůležitější poznámka se týká toho, že Raptor pro každé uživatelem nadefinované připojení k DB používá jen jedinou session. Proto i když mám otevřená dvě okna, v jednom už se provádí nějaký skript a v druhém spustím třeba i jednoduchý select, tak tento musí počkat. Stejně se sice chovají i Toad nebo SQL Navigator, ale u nich je možné otevřít nové spojení ručně.

Strom objektů se aktualizuje příjemně rychle, dokonce i např. v případě stovek synonym. Na druhou stranu se v něm nezobrazují databázové linky a adresáře (ALL\_DB\_LINKS, ALL\_DIRECTORIES). Ty je třeba spravovat manuálně.



Textový editor SQL je velmi klasický. Neobsahuje žádné převratné editační funkce, ale také není zcela bez různých vychytávek. Pokročilé funkce jsou překvapivě uvedeny v menu Source, nikoli Edit. Funkce odsazování bloků, hromadné (od)komentování jsou přítomny, takže mi nejvíce chybí možnost přípravy kódu pro použití v EXECUTE IMMEDIATE, t.j. chytré obalení vyznačeného textu apostrofy, což je sice děsivá technika, ale občas smrtelně užitečná.

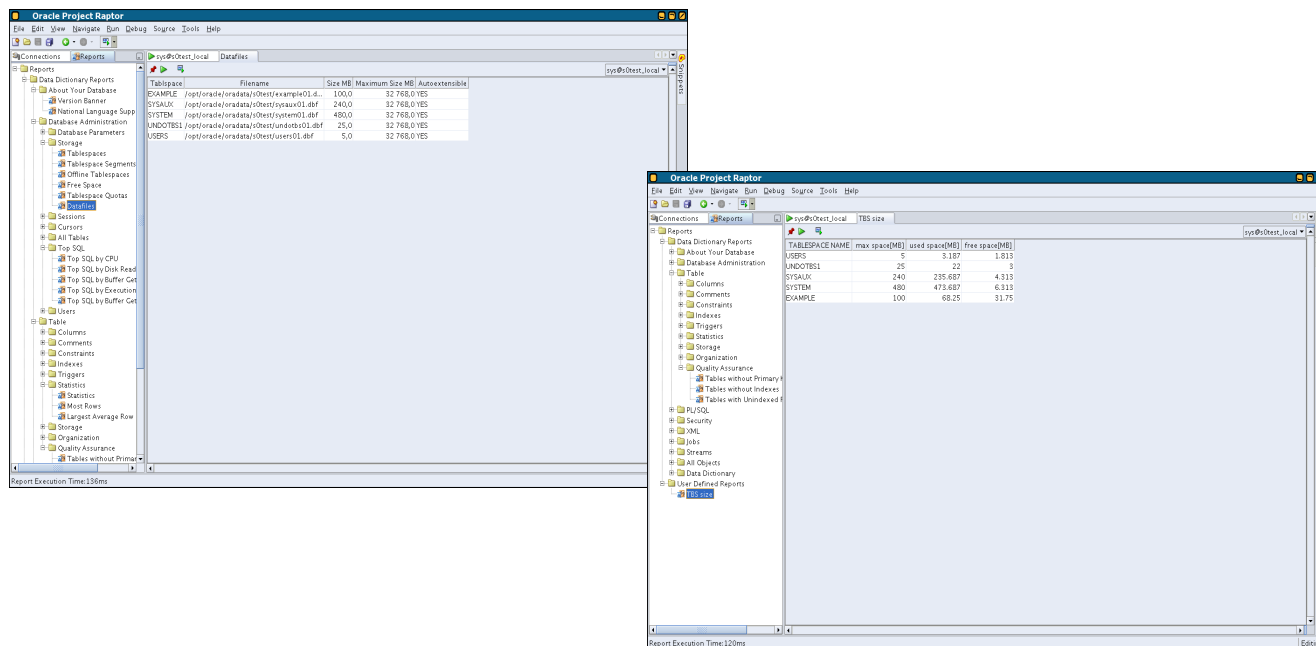


Z tabulky výsledků je možný export dat do CSV, XML, SQL insertů, Oracle Loaderu a čistého textu. Celkem pochopitelně chybí export do jakéhokoli tabulkového procesoru, což je sice škoda,

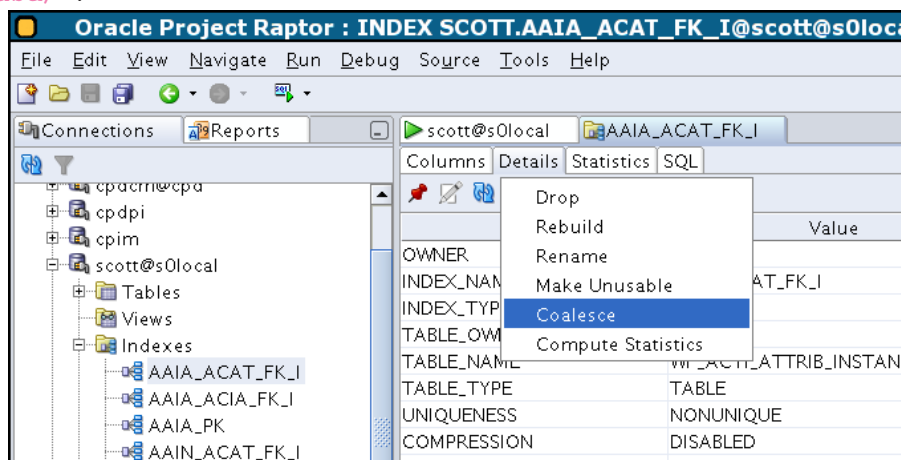
ale rozumím tomu, že je implementace, za podmínky už zmiňované platformní nezávislosti, téměř nemožná.

Nástroj Explain plan je velice přehledný. Používá decentní barvy a zvýraznění, která neruší, ale podporují orientaci. Ve srovnání s PL/SQL Developerem (černobílý) a SQL Navigátorem (circusově barevný) jednoznačně vítězí. Na druhou stranu neobsahuje všechny možné výstupy, které lze dostat z formátovaného DBMS\_XPLAN.DISPLAY().

Fragmenty kódu, tzv. snippets, v sobě obsahují i stručnou nápovědu. Osobně tyhle věci nepoužívám, ale rychlý seznam všech hintů optimalizátoru se mi opravdu hodí. Zatím ale chybí možnost vkládání vlastních fragmentů bez externí úpravy konfiguračního souboru. Debugger se zatím stále chová velice podivně a je nestabilní.



Už v prvním spuštění je předdefinována řada sestav. Od seznamu inicializačních parametrů, přes výpisy aktuálních připojení, aktuální SQL podle různých kritérií až po zobrazení práv a rolí. Příznějme si, že to jsou většinou jednoduché selekty do DBA\_% nebo USER\_% pohledů, které má každý, kdo s DB pracuje už někde napsané, ale spustit sestavu je často rychlejší, než hledat soubor vsql10.2.sql někde v ~/sql/dba/\*.



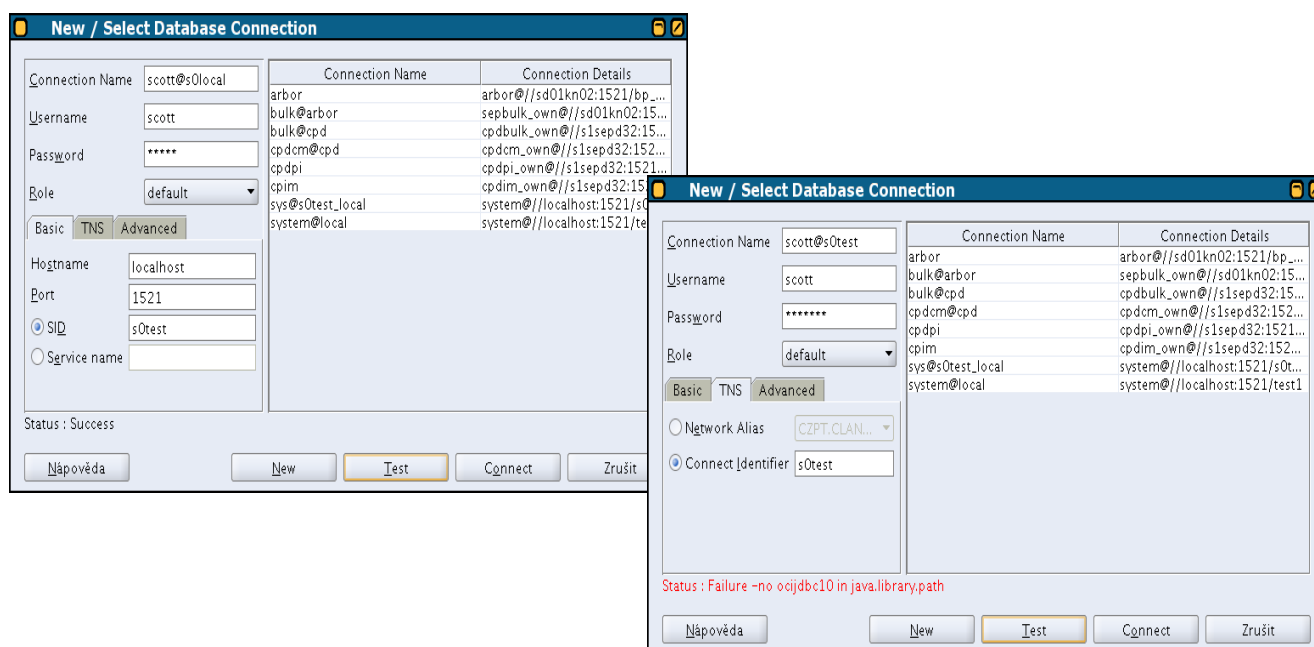
Nic také nebrání tomu, aby si uživatel zmíněné hotové selekty nastavil jako své nové sestavy. Má je pak uspořádány ve zvolené stromové struktuře.

## Cena a podpora

Za Raptor není a nebude třeba [platit](#) [4]. Podpora pro uživatele, kteří nemají licencovaný některý z Oracle produktů, bude jen a pouze v OTN diskusním fóru. Tyto diskuse však nejsou z nejkvalitnějších, respektive Raptor fórum je ještě stále poměrně věcné a obsahuje hodnotné příspěvky, což se nejspíš změní po uvolnění finální verze – stačí prolistovat ostatní OTN fóra. Pro uživatele s licenci podporu zajistí Oracle Support.

## Závěrem

Jak vyjádřit své nadšení, aniž bych byl nařčen z podjatosti? Opravdu to vypadá, jako kdyby se mi vývojáři Raptoru trefili do noty. Jednoduchý a snadno použitelný nástroj, který není zatížen megalomanskými vylepšeními. Raptor zkrátka podkopává zkušenost, že v Oracle neumí napsat nic jiného než databázový server (uživatelé Designeru a dalších lahůdek budou jistě souhlasit). Raptor mi žábu z disku sežral. P.S.: Škoda jen, že se finální verze bude jmenovat Oracle SQL Developer. Třikrát škoda.



## Odkazy

- [1] <http://sourceforge.net/projects/tora>
- [2] <http://www.toadsoft.com/>
- [3] [http://www.oracle.com/technology/products/database/project\\_raptor/index.html](http://www.oracle.com/technology/products/database/project_raptor/index.html)
- [4] [http://www.oracle.com/technology/products/database/project\\_raptor/pricing\\_faq.html](http://www.oracle.com/technology/products/database/project_raptor/pricing_faq.html)

\*\*\*



# Hibernace

Tibor Piňo

---

*Jak uspat notebook vám poradí návod, který obsahuje popis nastavení jádra, ACPI a dalších nástrojů nutných k úspěšnému zprovoznění hibernace.*

---

## Úvod

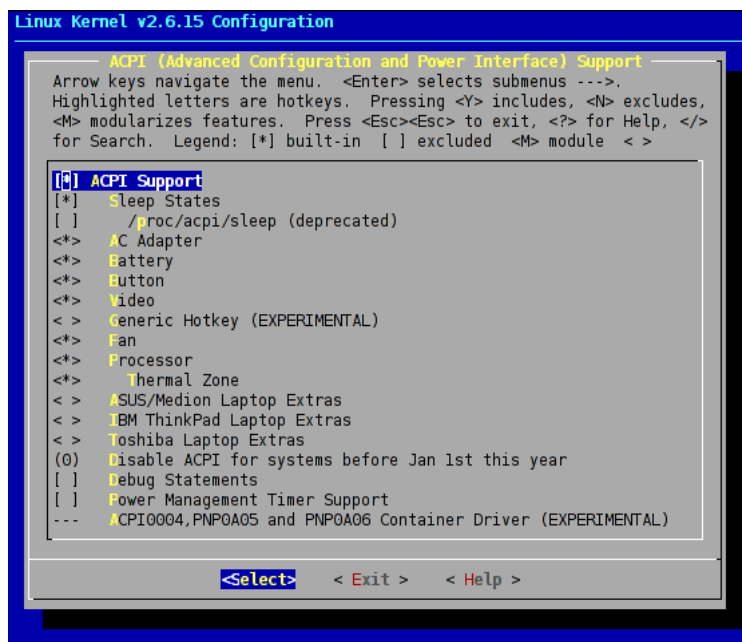
Hibernácia v Linuxe je oblasť ešte nie úplne prebádaná, potýkajúca sa s množstvom problémov. Medzi najväčšie úskalia patria grafické karty, ktoré nie vždy najlepšie spolupracujú. Základným kameňom pre nás bude ACPI (Advanced Configuration and Power Interface) a software suspend zo stránky ([www.suspend2.net](http://www.suspend2.net) [1]). ACPI bolo vyvinuté firmami ako HP, Intel, Microsoft, Phoenix a Toshiba. V súčasnosti ide de facto o štandard v oblasti riadenia spotreby. V Linuxe má ACPI slušnú podporu, tak prečo to nevyužiť. K software suspend dodám len toľko, že zvyčajne funguje lepšie, ako software suspend zahrnutý priamo vo vanilla jadre. V našom článku si samozrejme ukážame niekoľko trikov a tipov, ako sa horeuvedeným problémom vyhnúť. Tak, a už stačilo suchej teórie, môžeme sa pustiť do práce. Ingrediencie: vanilla kernel 2.6.15 [2], patch suspend pre jadro 2.6.15 [3], hibernate skript [4].

## Začíname:

1. V prvom kroku si rozbalíme zdrojové kódy jadra do adresára `/usr/src/linux`.
2. Rozbalíme si stiahnutý suspend patch napr. do `/usr/src/patch/suspend`.
3. Následne sa presunieme do adresára `/usr/src/linux`.
4. Príkazom `/usr/src/patch/suspend/apply` aplikujeme patch na jadro.
5. V prípade, že by sme z akýchkoľvek dôvodov niekedy chceli patch odstrániť, použijeme taký istý postup s príkazom `/usr/src/patch/suspend/unapply`.

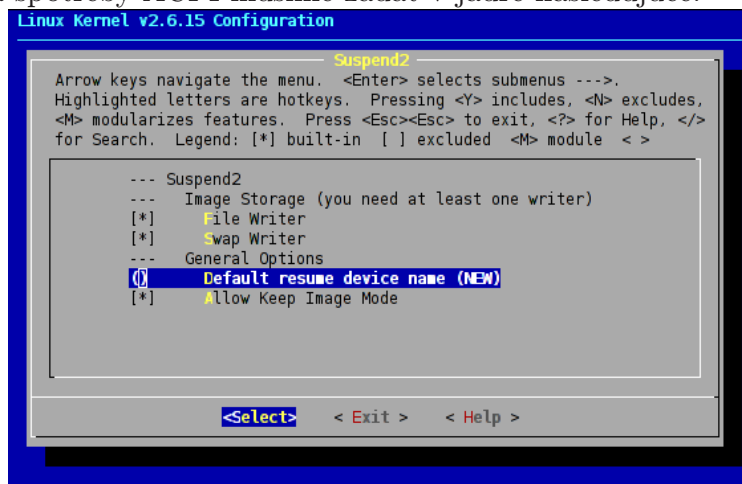
## Nastavenie jadra

Po úspešnom „opatchovaní“ jadra musíme v ďalšom kroku postupne povoliť v jadre všetky potrebné položky, ako je náš suspend, riadenie spotreby ACPI... Pre povolenie suspend musíme zadať v jadre nasledujúce:



*Power management support → Suspend2*

Pre povolenie riadenia spotreby ACPI musíme zadať v jadre nasledujúce:



Power management support → ACPI support

Ako môžeme vidieť, je značnou výhodou, že ACPI má preddefinovanú extra podporu pre vybrané značky notebookov. Tak, a teraz stačí len jadro skompilovať a úspešne z neho naboťovať.

## Konfigurácia

V ďalšom kroku sa budeme venovať nainštalovaniu hibernate scriptu, ktorý sa bude starať o uspávanie nášho miláčika a samotným nastaveniam hibernácie. Rozbalíme balík s hibernate scriptom, presunieme sa do adresára, kde sme súbory rozbalili a inštaláciu spustíme príkazom `./install.sh`, čím sa nám nainštalujú potrebné veci. Nás bude hlavne zaujímať:

- `/usr/local/sbin/hibernate` (skript, ktorý uspáva notebook)
- `/etc/hibernate/hibernate.conf` (konfiguračný súbor, ktorým riadime hibernáciu)

## Nastavenie hibernácie

Teraz by sme mali mať všetko pripravené k úspešnej hibernácii a môžeme sa pustiť do jej nastavovania.

Celé to funguje nasledovne. Beží nám démon `acpid` (musíme ho mať, samozrejme, nainštalovaný), ktorý sa stará o obsuhu udalostí. Napríklad, ak zavrieme notebook, démon `acpid` obsluží túto udalosť a zavolá napr. náš hibernate script. Nastavenie obsluhy týchto udalostí vykonávame v adresári `/etc/acpi/events/` tým spôsobom, že doň vkladáme skripty, v ktorých udávame, pre akú udalosť má démon spúšťať konkrétne programy.

Ak by sme si chceli sami odskúšať, aké ACPI udalosti generuje náš notebook, musíme vypnúť démona `acpid` a následne príkazom `cat /proc/acpi/event` môžeme po stlačení príslušnej klávesy alebo kombinácie kláves na obrazovke vidieť, aké udalosti generuje náš notebook a obslužiť ich tak, ako sme písali vyššie.

U mňa to funguje nasledovne. V adresári `/etc/acpi/events` mám vytvorený súbor `lid` [5], ktorý zavolá script hibernate po zatvorení notebooku. Taktiež si môžeme zvoliť vlastnú kombináciu pre uspávanie notebooku, ktorú zistíme horeuvedeným spôsobom.

Teraz sa budeme venovať nastaveniam samotnej hibernácie. Všetky zmeny budeme vykonávať v súbore `/etc/hibernate/hibernate.conf` [6]. Štruktúra tohto súboru je veľmi intuitívna a zvládne ju aj začiatočník, ja budem opisovať problémy, s ktorými som sa stretol. V prvom kroku musíme nastaviť spôsob uspávania, a to je napr. `suspend to ram` alebo `suspend to disk`. My si zvolíme `suspend to ram`.

PowerdownMethod 3

Rozdiel medzi `suspend to ram` a `suspend to disk` je v tom, že pri `suspend to disk` sa uloží obsah pamäte na disk a notebooku sa úplne vypne (opätovné spustenie je všeobecne o niečo pomalšie ako

pri *suspend to ram*), zatiaľ čo pri *suspend to ram* dáta ostávajú v pamäti a notebook spotrebúvava určité množstvo energie.

Často sa mi stávalo, že pri zatvorení notebooku som v ňom nechal zastrčený USB flash disk, takže ho treba „umountnúť“, aby som ho po zobudení notebooku mohol opäť korektne pripojiť.

```
Unmount /media/usbdisk
```

Pri uspaní je dobré vypnúť sieť a taktiež používam dosť často pcmcia kartu, ktorú takisto odpájam.

```
DownInterfaces eth0
UpInterfaces auto
EjectCards yes
```

Mojím najväčším problémom a musím podotknúť, že nielen mojím, bolo rozchodenie grafickej karty po otvorení notebooku. Vždy sa mi notebook prebudil do čiernej obrazovky a problémom bola práve grafická karta. Vyriešil som to programom [video\\_post](#) [7], ktorý by mal dokázať opäť rozchodiť vašu grafickú kartu. Primárne je určený pre grafické karty od ATI, ale funguje aj na iných typoch (osobne odskúšané). Video\_post treba rozbaľiť, skompilovať, vložiť niekde do systému (najlepšie tam, kde majú naň dosah systémové premenné) a vložiť nasledujúce riadky do nášho súboru [/etc/hibernate/hibernate.conf](#):

```
OnResume 30 /root/emu/video_post
OnResume 20 chvt 1
OnResume 10 chvt 7
```

Samozrejme že cestu [/root/emu/video\\_post](#) nahradíte cestou ku skompilovanému [video\\_post](#). Nieкто nemusí mať s prebudením svojej grafickej karty problém, takže video\_post vôbec nepotrebuje a do [/etc/hibernate/hibernate.conf](#) stačí zadať nasledujúce riadky:

```
OnResume 20 chvt 1
OnResume 10 chvt 7
```

Tento článok si nekladie za cieľ byť vyčerpávacím návodom. Snažil som sa byť čo najvšeobecnejší a úspech závisí od konkrétneho hardvéru. Určite vám môže byť nápomocná vynikajúca webová stránka [www.linux-laptop.net](#) [8], na ktorej nájdete množstvo príkladov konfigurácii k vášmu notebooku na rôznych distribúciách Linuxu.

## Odkazy

- [1] <http://www.suspend2.net>
- [2] <http://www.kernel.org>
- [3] <http://www.suspend2.net/downloads/>
- [4] <http://www.suspend2.net/downloads/>
- [5] <http://www.abclinuxu.cz/data/pino/lid>
- [6] <http://www.abclinuxu.cz/data/pino/hibernate.conf>
- [7] [http://bugme.osdl.org/show\\_bug.cgi?id=3670#c7](http://bugme.osdl.org/show_bug.cgi?id=3670#c7)
- [8] <http://www.linux-laptop.net>

\*\*\*

# GNU GPL v3

Marek 'marx' Grác

---

*Na svete je prvý návrh licencie GNU/GPLv3 z dielne Free Software Foundation. Od verejnej prezentácie v Cambridge už ubehlo niekoľko týždňov a tak je snáď správny čas na zhrnutie rozličných postojov, nálad a reakcií. Keďže zmien je neúrekom, tak sa pozrieme najmä na tie najproblematickejšie časti.*

---

Dnes už priam klasická licencia GNU/GPLv2 (ďalej len GPLv2, GPLv3) prišla na svet v júni 1991 a keďže sa vývoj nezastavil, tak je potrebné reagovať na nové podnety. O tom, ako dlho ešte vydrží, je zbytočné polemizovať, pretože väčšina vývojárov nebude nútená prejsť na novú verziu. Dá sa však očakávať, že najmä programy spod patronátu FSF prejdú na novú verziu okamžite. Pre väčšinu drobných programov však nová GPLv3 prináša len kozmetické výhody, a tak sa budeme stretávať so starou licenciou v oboch variantoch (s/bez „2 or any later“) ešte dosť dlho.

Udialo sa viacero zmien, ktoré sú hodné zmienky a aj tak sa určite k niektorým nedostaneme. Nová verzia mala niekoľko hlavných a niekoľko vedľajších cieľov. Jedným z tých vedľajších bolo skrátenie textu licencie. To sa viditeľne nepodarilo, pretože GPLv3 má už viac než 4500 slov (porovnaj s GPLv2: 3000, Apache License: 1400, BSD: 222 slov). Tentokrát by však mal mať dokument vyššiu právnu váhu, pretože na ňom spolupracoval Eben Moglen [1], profesor práva a histórie práva, ktorý je odborníkom na *duševné vlastníctvo*. Pravdepodobne aj vďaka nemu znie GPLv3 viac právnický. Niektoré časti sú podľa mňa (neprávnik) pre bežných smrteľníkov nejasné a ich výklad bude závisieť najmä na kvalite právnika.

No dosť bolo omáčky a prejdime si zaujímavé pasáže. Hneď prvou, veľmi dôležitou, zmenou je presné špecifikovanie kompletného zdrojového kódu. Ten musí obsahovať všetok zdrojový kód potrebný k tomu, aby ste mohli pochopiť, vytvárať modifikácie, inštalovať a spúšťať program. A čo je veľmi dôležité, tak musí obsahovať aj všetky šifrovacie a autentizačné kódy potrebné k tomu, aby sa kód (napr. vami modifikovaný) správal rovnako, až na zmeny spôsobené vašimi úpravami, a to za každých okolností. Táto úprava je vyslovene zameraná proti spoločnostiam ako TiVo [2], ktoré síce dodávajú zdrojový kód, ale na svojom zariadení umožňujú spustiť len taký kód, ktorý je podpísaný. *Za každých okolností* však obsahuje viacero zrád. Niektoré distribúcie pri „bezpečnom“ nastavení povoľujú používať len tie moduly kernelu, ktoré sú podpísané. To však znamená, že pri GPLv3 by museli dať k dispozícii svoj privátny kľúč a celé podpisovanie by stratilo zmysel. Druhou výhradou je, že pri zmene či použití hardvéru nie je možné zaručiť rovnakú funkčnosť. Napríklad, nič nám nebráni spraviť počítač, ktorý spúšťa len podpísané programy. Existencia takého počítača však zabraňuje aplikácii GPLv3 pre všetky (nielen tento) počítače, pretože nedokážeme zaručiť všetky podmienky z GPLv3.

## DRM

Mediálne veľmi zaujímavou pasážou je zákaz DRM. Toto jasne negatívne stanovisko sa stretlo s vyslovene kladnými i zápornými reakciami. Aj keď ide skôr o vyslovenie názoru, než o praktické dopady, pretože bez podpisovania aplikácii nemá zmysel vytvárať DRM aplikácie pod GPL. Jedným z užitočných dopadov je, že po odstránení DRM z GPL aplikácie neporušíte DMCA v USA (príp. obdobný zákon inde). Druhým, a zrejme o čosi vážnejším, dopadom by bolo, že žiadne zariadenie so softvérom pod GPL by nemohlo prejsť certifikačným procesom, pretože by nešlo zaručiť, že sa na ňom použije len certifikovaný softvér (napr. fiškálny modul v registračnej pokladnici alebo systém pre elektronické voľby). Takéto obmedzovanie sa však nepáči viacerým skupinám. A to rovnako ľuďom technicko/pragmatickým (Linus Torvalds) ako aj bojovníkom za slobodu (Debian GNU/Linux). Podľa odborníkov z Debianu vytvára takéto pravidlo precedens a hrozí, že v ďalšej verzii sa zakáže aj použitie na iné účely. Títo ľudia to myslia so slobodou skutočne vážne, pretože pred časom označili licenciu

GNU/FDL ako neslobodnú a preto vo *free* verzii Debianu nenarazíte na takúto licenciu. Podobný osud by mohol stihnúť aj programy pod GPLv3.

Zaujímavé, aj keď nie až také drastické zmeny nastali aj pri šírení aplikácií v binárnej podobe. Zdrojový kód musí byť stále dostupný minimálne na požiadanie. Jednou z možností, ako dodávať zdrojový kód, je dať ponuku platnú aspoň tri roky, že dodáte kompletný zdrojový kód. Za túto prácu si môžete účtovať desaťnásobok vašich vyčísliteľných nákladov (t.j. súčasný stav krát desať). Ak sa budeme trochu snažiť a dodržíme literu licencie, tak kompletný zdrojový kód musí obsahovať všetko potrebné pre spustenie nášho programu (okrem kernelu, glibc a pár ďalších súčastí). A ak používame zopár knižníc a tie používajú opäť zopár knižníc, tak nie je jednoduché povedať, čo všetko vlastne musíme dodať. Takéto skúmanie nepochybne stojí čas, ktorý sa dá jednoducho previesť na peniaze a pri troche šikovnosti sa dostaneme na štvor či päť miestne sumy (samozrejme v dolároch, či eurách), čo nás síce neuchráni pred povinnosťou zverejniť zdrojový kód, ale dokáže výrazne sťažiť prístup k nemu (najmä v spojení s nasledujúcou časťou).

Dosť netradičnou novinkou je generálna amnestia pre tých, čo porušia GPL, ale do 60 dní sa postarajú o nápravu. Má ísť o preventívne opatrenie, aby sa zbytočne nezaťažovali súdy tými, ktorí porušili GPL nechtiac. Harald Welte z [www.gpl-violations.org](http://www.gpl-violations.org) [3] sa vyjadril, že každý obhajca tvrdí, že porušenie nebolo zámerné. Takáto amnestia zníži vymáhateľnosť GPL, pretože potom nebude možné, aby súd použil predbežné opatrenia.

Obávanou novinkou mala byť možnosť sprístupnenia zdrojových kódov aj pre aplikácie, ktoré nemá používateľ priamo nainštalované. V tomto prípade nakoniec prevládlo miernejšie riešenie, ktoré umožňuje vyžadovať, aby program obsahoval funkcionálnu potrebnú na stiahnutie kompletného zdrojového kódu. Keďže tento postoj je odlišný od toho, že môžeme program slobodne modifikovať, tak sa autori rozhodli, že na spôsob americkej ústavy zavedú systém dodatkov. Existuje len obmedzené množstvo týchto dodatkov a zameriavajú sa aj na problémy s ochrannými známkami alebo využitia pôvodného mena projektu. Práve ony umožňujú jednosmernú kompatibilitu s Apache License a teda využívanie projektov ako Apache, Tomcat, alebo SpamAssassin v aplikáciách šírených pod GPLv3 (príp. vytvorenie vlastnej vetvy širenej pod GPLv3). Na druhej strane, vďaka nim vzniká hneď niekoľko licencií GPLv3, ktoré môžu spôsobovať problémy s kompatibilitou licencií jednotlivých free programov.

Jednou z oblastí, s ktorou by sa mala GPLv3 vysporiadať, sú softvérové patenty. Ak distribuujete dielo pokryté GPLv3, potom dávate celosvetový súhlas na bezplatné použitie v tomto a v každom odvodenom diele, ak zostane zachovaná licencia. Na obranu používateľov slúži presun zodpovednosti na toho, kto vám poskytol aplikáciu za predpokladu, že to spravil vedome. Keďže každý právnik bude tvrdiť, že použitie patentu, na ktoré spoločnosť nemá práva, bolo neúmyselné, tak stále neexistuje ochrana koncových používateľov pred patentovými spormi.

Na záver sa ešte treba zmieniť o tom, že hoci každý má možnosť sa vyjadriť k novému zneniu GPLv3, tak na záver neprebehne žiadne verejné hlasovanie, dokonca ho nebude schvalovať ani vedenie Free Software Foundation. Rozhodnutie o finálnej verzii bude len v rukách Richarda Stallmana, ktorý sa musí riadiť len duchom licencie, pretože žiadne iné podmienky nie sú kladené na ďalšiu verziu. Prvý návrh novej verzie skutočne rieši niektoré existujúce problémy. Samozrejme, že ich riešenie otvára priestor pre nové problémy, ktoré doteraz nebolo potrebné riešiť. Bohužiaľ sa však stále nájdu pasáže, na ktorých nejednoznačnosti sa nič nezmenilo.

## Odkazy

- [1] [http://en.wikipedia.org/wiki/Eben\\_Moglen](http://en.wikipedia.org/wiki/Eben_Moglen)
- [2] <http://www.tivo.com>
- [3] <http://www.gpl-violations.org>

\*\*\*



# Gentoo Linux – majstrovská inštalácia

Matej 'Yin' Gagi

---

*Toto nie je inštaláčna príručka Gentoo! Tipy a triky, ktoré v tomto článku nájdete, môžete použiť na urýchlenie aj iných distribúcií. Dnes si ale urobíme malý úvod do Gentoo.*

---

Distribúcia Gentoo Linux priniesla do Linuxu BSD porty a tým zanechala v histórii Linuxu hlbokú stopu. Gentoo a jeho filozofia naozaj stoja za skúšku.

Mnohé z trikov, ktoré užívatelia Gentoo využívajú, by ocenili aj užívatelia iných distribúcií. Vedomosti, ktoré som pri snahe vytvoriť čo najrýchlejší operačný systém získal, som sa rozhodol zapísať a publikovať, aby z nich mohlo profitovať viac ľudí.

Riešenia, ktoré budem v nasledujúcich dieloch seriálu opisovať, je najjednoduchšie aplikovať v distribúcii Gentoo. Napriek tomu, že som mnohé z nich na iných distribúciách neskúšal, budem ich opisovať tak, aby bol postup na iných distribúciách jasný každému.

## Základné kroky inštalácie

Inštalácia Gentoo väčšinou nie je triviálna záležitosť. V niektorých prípadoch môže zabrať aj celý týždeň. Ale to sa nemôže stať, ak poznáte a rozumiete celému jej priebehu. Takže bude dobré si vyjasniť jej princípy.

Filozofia Gentoo je založená na možnosti slobodného výberu. V Gentoo sa musíte často rozhodovať, čo chcete a čo nie. Preto si treba dopredu rozmyslieť, koľko času ste ochotný svojmu systému venovať. Za dostatočnú pozornosť vás Gentoo odmení rýchlosťou a komfortom, za jej nedostatok len frustráciou.

Prvýkrát ste postavený pred vážne rozhodnutie už pri sťahovaní inštaláčného, alebo skôr „štartovacieho“ Live CD. Existuje viacero ISO obrazov, pre rôzne platformy a pre rôzne druhy inštalácie. Rozhodovanie si môžete zjednosušiť tým, že si stiahnete „univerzálne CD pre x86“, na ktorom máte všetko potrebné na inštaláciu a funguje aj na 386-kách. Po stiahnutí, napálení a naboťovaní z Live CD vás čakajú nasledovné kroky:

1. konfigurácia siete
2. delenie disku
3. pripojenie partícií

Tu prichádza ďalšie rozhodovanie: výber začiatočného **stage**. Existujú tri **stage**:

1. V prvom (**stage1**) necháte skompilovať základne nástroje na kompiláciu a dostanete sa do fázy **stage2**.
2. Ak začínate od **stage2**, kompilujete už len nutné systémové nástroje.
3. **Stage3** je už celkom použiteľný systém, treba ešte skompilovať ďalšie systémové nástroje, tu sa znovu rozhodujete ktoré.

Po výbere **stagu** vás čakajú tieto kroky:

1. Skopírovanie balíku s vybraným **stage** na disk.
2. Nastavenie systému pre inštaláciu.
3. Kroky v **stage1**, **stage2** a **stage3**.
4. Kompilácia kernelu (existuje automatizovaný generátor konfiguračného súboru kernelu).
5. Príprava bootovacieho manažéra (◊[LILO](#) [1], alebo ◊[Grub](#) [2]).
6. Reštart do novej distribúcie a doinštalovávanie ďalších programov (X.org a pod.).

## Portage

Srdcom distribúcie Gentoo je systém Portage. Portage je správca balíčkov, ktorý na povel automaticky stiahne, skompiluje a nainštaluje (prípadne odinštaluje) určený program. Portage vie inštalovať a vytvárať aj binárne balíky, ale jeho primárnym poslaním je automatizovaná kompilácia.

Informácie o jednotlivých balíkoch sú uchovávané v súboroch zvaných ebuildy. Tie sa predvolene nachádzajú v adresári `/usr/portage`, rozumne usporiadané v kategóriách. V uvedenom adresári sa nachádzajú aj ďalšie súbory, napríklad stiahnuté zdrojové kódy v podadresári `distfiles`.

Všetky ebuildy usporiadané v kategóriách sa nazývajú *Portage tree* (Portage strom). Gentoo aktualizuje oficiálny Portage strom denne. Svoju lokálnu kópiu môžete synchronizovať s oficiálnym stromom a potom aktualizovať staré programy. Robte to približne raz za mesiac.

## /etc/make.conf

Hlavným konfiguračným súborom Portage je `/etc/make.conf`. Tu sa nachádzajú všetky parametre kompilácie (`USE flag`, parametre kompilácie a podobne).

## /etc/env.d/\*

Adresár `/etc/env.d/` obsahuje niekoľko súborov nazvaných `XXabc` (XX – číslo, abc – názov balíku). Každý z týchto súborov je určený na konfiguráciu jedného balíku. Nastavenia v adresári `/etc/env.d/` premieta do pravých konfiguračných súborov program `env-update`.

Syntax týchto súborov je pre všetky rovnaká. To vytvára istú abstraktnú vrstvu na konfiguráciu niektorých dôležitých programov v systéme. Cez tieto súbory môžete nastavovať zväčša užívateľské programy a programy na správu systému. Nie systémových démonov a iné služby.

## /etc/init.d/\*

Tu nájdete všetky systémové služby, ktoré máte v Gentoo nainštalované, od nastavení času a sieťoviek, cez grafický login (`xdm` a pod.) až po Apache, firewall a routovanie. Spustenie a vypnutie služby je jednoduché:

```
/etc/init.d/služba start
/etc/init.d/služba stop
```

Do procesu bootovania môžete služby pridávať a potom ich z neho odoberať programom `rc-update`:

```
rc-update add služba default
rc-update del služba default
```

## /etc/conf.d/\*

Podobne ako v adresári `/etc/env.d/`, aj tu sa nachádzajú súbory s nastaveniami. Tentokrát sa ich mená zhodujú so súbormi v adresári `/etc/init.d/`. Jednotlivé súbory obsahujú nastavenia pre tú službu, ktorej názov sa zhoduje s názvom dotyčného súboru s nastaveniami. Je to celkom jednoduché a praktické. Všetky súbory, až na nastavenia sieťových kariet, majú rovnakú, jednoduchú syntax. Znovu je to užitočná abstraktná vrstva nad pravou konfiguráciou systému. Tentokrát ale nie je potrebný žiaden nástroj na premietanie nastavení do konfiguračných súborov.

## USE flagy

`USE` je názov premennej prostredia, ktorá v Gentoo určuje, ktoré programy a funkcie bude systém podporovať, a ktoré nie. Hodnota tejto premennej sa skladá zo slov oddelených medzerou. Každé slovo vyjadruje jednu vlastnosť alebo funkciu.

Portage nastaví a skompiluje každý program tak, aby vyhovoval nastaveniam v `USE` flagoch. Ak nechcete používať `oss`, ale `alsa` (zvukový systém), všetky tie programy, ktoré sa takto dajú nastaviť budú používať zvukový systém `alsa`: `USE="alsa -oss"`.

`USE` flagy sa dajú nastaviť v súbore `/etc/make.conf`. Najlepšie ale je, že `USE` flagy sa dajú nastaviť osobitne pre každý balík pri jeho kompilácii a nemusíte sa uspokojiť s jednotným nastavením celého systému:

```
$ USE="dodatočné use flagy pre inštalovaný balík" emerge balík
```

Na výber je množstvo `USE` `flag`ov, treba si len vybrať. Všetky popisy nájdete vo svojom Gentoo v súboroch `/usr/portage/profile/use.desc` a `/usr/portage/profile/use.local.desc`.

## Záver

Krátke zoznámenie zo základnými princípmi distribúcie Gentoo Linux máme za sebou. Teraz aspoň viete, o čom to asi je. Nabudúce sa už pustíme do skutočnej optimalizácie systému a na svoje si príde každý pokročilý užívateľ Linuxu, aj keď nepoužíva distribúciu Gentoo.

## Odkazy

[1] <http://www.abclinuxu.cz/slovník/lilo>

[2] <http://www.abclinuxu.cz/slovník/grub>

\*\*\*



# Gentoo Linux – partície

Matej 'Yin' Gagi

---

*Správne rozdelenie partícií môže nielen zvýšiť bezpečnosť dát, ale aj zrýchliť počítač a usporiť diskový priestor! Čím je na partícii viac súborov, tým je pomalšia. Súborový systém sa stáva zložitejší a operačný systém má stále viac problémov s fragmentáciou dát a adresárovou štruktúrou.*

---

## Portage tree

V Gentoo sa problémy s veľkými partíciami prejavujú hlavne na systéme Portage, ktorý je už aj tak veľmi pomalý. Riešenie som našiel na Gentoo-wiki: Umiestnite adresár `/usr/portage` na samostatnú partíciu. Odporúčam nastaviť veľkosť na minimálne 500MB (bez podadresára `distfiles` stačí menej ako 800MB). Adresár `/usr/portage/distfiles` zvykne byť veľký rádovo stovky megabytov, až niekoľko gigabytov. Preto by mal byť umiestnený na inú partíciu, najlepšie na tú, kde máte veľa dát (hudba, filmy, databázy, CVS repozitáre a pod.).

Portage o umiestnení adresára `distfiles` nebude vedieť automaticky, ale môžete mu o ňom povedať. Editujte súbor `/etc/make.conf` a doplňte (prípadne zmente) riadok s premennou `DISTDIR`:

```
DISTDIR="/novy/adresar/distfiles"
```

Alebo pripojte svoj nový adresár `distfiles` na cestu `/usr/portage/distfiles` pomocou príkazu `mount` s parametrom `--bind`. Robte to až potom ako pripojíte partíciu s Portage tree, inak prekryje `distfiles`:

```
# mount /dev/hdXY /usr/portage
# mount --bind /novy/adresar/distfiles/ /usr/portage/distfiles
```

## Bootovacia partícia

Dobré je vytvoriť aj samostatnú partíciu pre adresár `/boot`, ktorú systém nebude pripájať automaticky pri štarte. Ak z nejakého dôvodu dôjde v systéme k poruche a táto partícia sa nepoškodí. Kernel, konfigurácia GRUBu a podobne, sa vám pri obnove systému môžu hodiť. Navyše GRUB je na konfiguračnom súbore na disku závislý.

Odporúčam použiť overený súborový systém `ext2`, alebo `ext3` o veľkosti niekoľko MB až 100MB. Táto partícia by nemala byť pripájaná automaticky pri štarte systému. To sa dá docieľiť tým, že mu do súboru `/etc/fstab` pridáme parameter `noauto`:

```
/dev/hdXY    /boot    ext3    defaults,noauto    0 0
```

## Loopback device

*Loopback device* (nemyslí sa lokálne sieťové rozhranie) je blokové pseudo zariadenie, cez ktoré môžete zo súboru urobiť partíciu. Je to čisto softvérová záležitosť, takže peňaženku potrebovať nebudete. Najprv musíme vytvoriť dostatočne veľký súbor na našom disku:

```
dd if=/dev/zero of=/náš/súbor bs=1M count=veľkosť súboru v MB
```

Program `dd` skopíruje *nuly* zo špeciálneho súboru `/dev/zero` do nášho výstupného súboru. Parameter `bs=1M` hovorí programu, aký veľký má byť *jeden block* a parameter `count=...`, koľko blokov skopírovať. Parametre `of`, `bs` a `count` upravte podľa potreby.

Pomocou programu `dd` môžeme vytvoriť aj takzvaný *sparse súbor*, ktorý pri vytvorení nezaberie na disku takmer žiadne miesto. Pri používaní sa tento súbor bude postupne zväčšovať podľa potreby:

```
dd if=/dev/zero of=/náš/súbor bs=1M skip=veľkosť_súboru_v_MB count=0
```

V súbore musíme vytvoriť súborový systém. Pre `ext2` musí byť súbor veľký aspoň 60 kB, pre `ext3` aspoň 2 MB a pre `reiserfs` aspoň 33 MB: `# mkfs.ext2 /náš/súbor` alebo `# mkfs.ext3 /náš/súbor` alebo `# mkfs.reiserfs -f /náš/súbor`. Súbor so súborovým systémom už môžeme pripojiť. Najjednoduchší spôsob je použiť príkaz `mount` a ten sa postará o správne spustenie programu `losetup`: `# mount /náš/súbor /miesto/pripojenia -o loop`.

Loopback device môže byť aj šifrovaný. Vytvorenie a sprevádzkovanie takéhoto bezpečného úložiska dát je zložité a mimo rámec tohto článku. Ak v diskusii niekto prejaví záujem, napíšem aj osobitný článok o bezpečnosti.

## Swap? Už nikdy viac!

Mnoho miesta na pevnom disku zaberá partícia na *swap*. Swap je miesto, kam operačný systém ukladá dáta z operačnej pamäte, keď sa v nej minie miesto. Málokedy sa však stáva, že operačný systém swap použije a v tomto prípade je to len vyhodnený diskový priestor.

Namiesto samostatnej partície môžeme použiť loopback device, s ktorým sa dá narábať pohodlnejšie, ako so samostatnou partíciou. Súbor, do ktorého swap umiestnime, sa musí nachádzať na niektorej z partícií, ktoré sa pripoja automaticky pri štarte. Inak sa swap pri štarte nepripojí.

```
# dd if=/dev/zero of=/swap bs=1M count=256
# mkswap
```

Ale pozor! Pre swap sa neodporúča použiť sparse súbor. Kernel musí do swapu zapisovať dáta veľmi rýchlo. Alokácia miesta pre sparse súbor by mohla spôsobiť také spomalenie systému, že by sa vám nemuselo podariť ho ani vypnúť. Mne osobne sa nepodarilo sparse súbor ani pripojiť ako swap.

Swap v súbore pripojíme ešte o čosi jednoduchšie ako súborový systém: `# swapon /swap`

Záznam o swape môžeme pridať aj do súboru `/etc/fstab`. Gentoo sa snaží swap pripojiť dvakrát. Prvýkrát na začiatku štartovania systému a druhý krát po pripojení všetkých súborových systémov. To znamená, že súbor so swapom môžete umiestniť na ktorýkoľvek súborový systém (ktorúkoľvek partíciu). Záznam v súbore `/etc/fstab`:

```
/swap swap swap defaults 0 0
```

## Záver

Na záver by som ešte rád podotkol, že som nespomenul mnoho ďalších zaujímavých riešení. Napríklad [šifrovaný loopback device](#) [1], na ktorý môžete odkladať veľmi dôležité dáta, alebo [Linux Volume Manager](#) [2] – správca dynamických partícií.

## Odkazy

[1] <http://www.ibiblio.org/pub/Linux/.../Loopback-Encrypted-Filesystem-HOWTO.html>

[2] <http://www.gentoo.org/doc/en/lvm2.xml>

\*\*\*

# Gentoo Linux – optimalizácie

Matej 'Yin' Gagy

---

*Viete prečo je lepšie skompilovať si programy vlastnoručne alebo nechať Portage, aby to urobilo za vás? Pretože binárne balíky nie sú optimalizované pre váš procesor a sú badateľne pomalšie. Dnes si porozprávame o tom, ako zo systému vyťažiť maximum.*

---

## CFLAGS, CXXFLAGS

Premennú prostredia `CFLAGS` používa `gcc` pri kompilácii zdrojových kódov v jazyku C. V Gentoo túto premennú nastavujeme v súbore `/etc/make.conf` a Portage ju pred kompiláciou balíčka prečíta a nastaví. Premenná `CXXFLAGS` funguje rovnako, ale pre jazyk C++. Ak nemáte problém s diskovým priestorom, ani s časom potrebným na kompiláciu, môžete bezpečne použiť nasledujúcu hodnotu `CFLAGS` a `CXXFLAGS`:

```
CFLAGS="-O3 -fomit-frame-pointer -pipe"
CXXFLAGS="$(CFLAGS)"
```

Výsledný program však nebude ušitý na mieru vášmu procesoru a bude využívať iba inštrukčnú sadu procesora i386 (samozrejme ak vlastníte procesor založený na tejto architektúre).

## Ako funguje GCC

Najprv by sme si mali povedať pár slov o tom, ako pracuje GCC. Ako už možno viete, GCC obsahuje viac kompilátorov naraz – kompilátory pre jazyky C, C++, Objective C, Fortran a Javu, ale aj ďalšie. Kompilátor GCC sa skladá z dvoch častí – frontend a backend.

Každý jazyk má vlastný frontend. Frontend načíta (preparsuje) a spracuje samotné zdrojové kódy. Výstupom frontendu je reprezentácia zdrojového kódu v jazyku RTL (Register Transfer Language). Dá sa povedať, že RTL je abstraktný štruktúrovaný assembler. Je to veľmi všeobecný a zložitý jazyk zároveň, preto nemá zmysel ho používať pre bežné programovanie.

Strom RTL sa pri kompilácii nachádza v pamäti len v binárnej forme. V binárnej forme vstupuje do backendu kompilátora GCC, ktorý je všeobecný pre celý kompilátor. Backend generuje z RTL stromu kód assemblera pre určený procesor. Backend tiež vykonáva optimalizácie nad RTL stromom a aj pri generovaní assemblera.

Backend vie vykonať niekoľko desiatok rôznych optimalizácií a nie všetky sú vhodné v každej situácii. Preto sú rôzne druhy optimalizácií roztriedené do skupín, aby ich bolo možné jednoducho zapínať – o chvíľku si o nich povieme. Zostali vám nezodpovedané otázky? Ak vás zaujíma, ako vyzerá RTL, skúste skompilovať jednoduchý program s parametrom `-dr`: `gcc -dr zvyšné parametre`.

Ak chcete vedieť, ako prebiehajú rôzne optimalizácie v backende kompilátora GCC, prečítajte si článok [The GNU Instruction Scheduler](#) [1] z roku 1989 (podstata GCC sa nezmenila).

## Optimalizácia

Najdôležitejší je parameter optimalizácie `-Ox`. Určuje, ktorú skupinu optimalizácií chceme pri kompilovaní použiť. Písmeno `x` treba nahradiť číslom, alebo písmenom želanej skupiny. Popis jednotlivých skupín optimalizácií:

**-00**

Neoptimalizuj. Kompilátor GCC nevykoná žiadne optimalizácie, kompilácia je najrýchlejšia a výsledný program je najpomalší.

**-01**

Optimalizuj. Kompilátor vykoná optimalizácie, ktoré zmenšia a zrýchlia výsledný program a pritom príliš nepredĺžia čas kompilácie a jej nároky na pamäť.

**-02**

Optimalizuj ešte viac. Kompilátor zapne všetky optimalizácie, ktoré urýchlia program a pritom priveľmi nezväčšia jeho veľkosť. Kompilácia je pomalšia a vyžaduje viac pamäte. Táto voľba sa používa najčastejšie.

**-03**

Optimalizuj čo najviac. Toto je najnáročnejšia optimalizácia. Výsledný program je najväčší, ale aj najrýchlejší. Ak máte dobrý procesor (500MHz a viac) a potrpíte si na rýchlosť behu aplikácií, je to pre vás ideálna voľba.

**-Os**

Optimalizuj pre veľkosť. Zapne všetky optimalizácie parametru **-O2**, ktoré nezväčšia veľkosť výsledného programu a pridá ďalšie optimalizácie pre veľkosť výsledku. Rýchlosť a nároky na pamäť sú podobné, ako pri **-O2**.

Ak špecifikujete viac parametrov **-Ox** naraz, použije sa iba posledný.

Parameter **-fomit-frame-pointer** znemožní debugovanie (používateľ to nepotrebuje) a zmenší veľkosť binárky. Parameter **-pipe** urýchli kompiláciu, pretože **gcc** nebude medzivýsledky ukladať na disk, ale ponechá ich v pamäti.

## Inštrukčné sady

Každý typ procesora má vlastnú inštrukčnú sadu – inštrukcia je jeden úkon, ktorý procesor dokáže vykonať. V assembleri nepoužívame príkazy, ale priamo inštrukcie procesora. Z toho vyplýva, že program v assembleri je závislý na procesore, pre ktorý je určený a nedá sa spustiť na inom.

Našťastie nové procesory poznajú všetky inštrukcie starších procesorov a inštrukčnú sadu len rozširujú o ďalšie inštrukcie. Túto vlastnosť majú procesory z radu Intel x86 – procesory bežných PC (i386, i486, Pentium atď.).

Ako som už povedal, kompilátor GCC generuje z RTL kódu kód assembleru a potrebuje teda vedieť, pre ktorý procesor má assembler vygenerovať. Na platforme x86 generuje predvolene pre procesor i386 a ten je možné použiť na ktoromkoľvek vyššom procesore.

Parameter **-march=architektúra** určuje inštrukčnú sadu procesora, na ktorom má program fungovať. Ak chceme, aby program fungoval aj na starších procesoroch, ale chceme ho optimalizovať pre svoj procesor, môžeme použiť jeden z parametrov:

- **-mcpu=architektúra** (gcc verzie 3.3 a nižšie)
- **-mtune=architektúra** (gcc verzie 3.4 a vyššie)

(Verziu gcc zistíme pomocou príkazu: **gcc -v**.)

Za slovo *architektúra* dosadíte označenie procesora, pre ktorý kompilujete. Označenia všetkých procesorov nájdete v manuálovej stránke GCC. Ak kompilujete pre procesor typu x86, hľadajte sekciu *Intel 386 and AMD x86-64 Options* (jednoduchšie je nechať vyhľadať na manuálovej stránke reťazec *athlon*).

## LDFLAGS

LDFLAGS používá linker (program, který spája části zdrojových kódů do spustitelného souboru, alebo knižnice). Bezpečná hodnota premennej LDFLAGS je: `LDFLAGS="-Wl,-O1"`

Ak chcete dostať zo systému maximum a ste ochotný trochu riskovať:

```
LDFLAGS="-Wl,-O1 -Wl,--sort-common -z -combreloc -Wl, \\  
--enable-new-dtags -Wl,--relax"
```

V niektorých prípadoch sa pri kompilácii `glibc` môžu vyskytnúť problémy. Je lepšie nepoužívať v hodnote `LDFLAGS` parameter `-Wl,--relax`:

```
LDFLAGS="-Wl,-O1 -Wl,--sort-common -z -combreloc -Wl,--enable-new-dtags"
```

Problémy pre kompiláciu robia aj *OpenOffice.org* a *GStreamer*. Tu sa odporúča radikálne riešenie – nepoužívať žiadnu hodnotu `LDFLAGS`: `LDFLAGS=""`

V týchto špeciálnych prípadoch si môžete usporiť námahu s editovaním souboru `/etc/make.conf` a premenné určiť priamo z príkazového riadku:

```
# LDFLAGS="" emerge gstreamer  
# LDFLAGS="" emerge openoffice
```

## Odkazy

[1] <http://www.student.informatik.tu-darmstadt.de/~misar/gcc/scheduler.html>

\*\*\*

# Gentoo Linux – rýchlosť kompilácie (ccache a distcc)

Matej 'Yin' Gagi

---

*Ako dlho sa na vašom počítači kompiluje nová verzia prehliadača Mozilla Firefox, prehrávača MPlayer alebo celé KDE, či Gnome? Pár minút? Tak to už asi poznáte programy ccache a distcc.*

---

## ccache

`ccache` dokáže zrýchliť opakované kompilovanie 5 až 10krát. Je výtvorom Andrewa Tridgella z projektu Samba, ktorý sa inšpiroval ideou projektu `compiler-cache` [1].

Pred každým spustením `gcc` sa `ccache` pozrie do svojej databázy a pokúsi sa v nej nájsť skompilovaný súbor. Ak výsledok kompilácie nenájde, spustí `gcc` a jeho výsledok vloží do databázy. Pri vývoji aplikácií sa medzi dvoma verziami aplikácie často mení iba zopár súborov a ostatné zostávajú nedotknuté. Pri kompilovaní novej verzie už raz skompilovaného programu má `ccache` veľkú šancu usporiť čas potrebný pre kompiláciu už raz skompilovaných súborov.

Portage dokáže `ccache` používať automaticky. Stačí ho len nainštalovať: `# emerge ccache` a pridať do súboru `/etc/make.conf` do premennej `FEATURES`: `FEATURES="ccache"`. Ak chcete `ccache` používať mimo Portage (v inej distribúcii alebo pri programovaní), musíte dať cestu k binárkam `ccache` do premennej prostredia `PATH` (pred cestou ku `gcc`): `# export PATH=/usr/lib/ccache/bin:$PATH`. Databáza `ccache` bude rásť a rásť, preto je dobré stanoviť jej nejaký limit: `# ccache -M 1G`. Používatelia Gentoo si tento údaj môžu nastaviť aj v súbore `/etc/make.conf`: `CCACHE_SIZE="1G"`.

## distcc

Z dielne Samby pochádza aj program `distcc`, tentokrát je jeho autorom Martin Pool. `Distcc` distribuuje kompiláciu medzi viacerými počítačmi. To sa oplatí hlavne, pokiaľ chcete inštalovať Gentoo na pomalý počítač a máte k dispozícii jeden, alebo viac rýchlejších strojov.

Program `distcc` najprv spracuje vstupný zdrojový súbor preprocesorom (`gcc -E`). Vznikne tak zdrojový súbor, ktorý obsahuje všetky hlavičkové súbory, ktoré sú potrebné na jeho skompilovanie. Program `distcc` tento súbor pošle na niektorý zo vzdialených počítačov, kde sa spustí kompilácia. Skompilovaný súbor s príponou `.o` (*object*), sa vráti späť na klientsky počítač a `distcc` sa ukončí.

Aby tento postup kompiláciu urýchlil, musí byť kompilovaných viacero zdrojových súborov naraz. Program `make` podporuje paralelnú kompiláciu. Počet súčasne spustených kompilácií sa nastavuje parametrom `-jN` (`N` – počet paralelných kompilácií). Používatelia Gentoo si tento parameter môžu nastaviť v súbore `/etc/make.conf`: `MAKEOPTS="-j2"`. S počtom paralelných kompilácií sa musíte trochu pohrať. Zčať môžete s hodnotou rovnou počtu procesorov na kompilovanie plus jeden (napríklad ak kompilujete na dvoch dvoj-procesorových počítačoch nastavte `MAKEOPTS="-j5"`). Tento počet vám pomôže doladiť grafický monitor `distccmon-gui`.

`Distcc` treba samozrejme nainštalovať, v Gentoo jednoducho: `# emerge distcc`. Portage vie používať `distcc` automaticky, len ho musíte pridať do súboru `/etc/make.conf` do premennej `FEATURES`: `FEATURES="distcc"`. Ak nepoužívate Gentoo, musíte si binárky `distcc` pridať do premennej `PATH` ešte pred cestou ku `gcc`. Ale pozor, ak používate aj `ccache`, najprv musí ísť cesta k `ccache`, potom k `distcc` a až potom všetko ostatné:

```
# export PATH=/usr/lib/ccache/bin:/usr/lib/distcc/bin:$PATH
```

Na vzdialených počítačoch musíte buď spustiť démona `distccd`, nakonfigurovať `initd` server alebo sprístupniť počítač cez `ssh`. Odporúčané je spustiť démona `distccd`:



```
# distccd --allow 192.168.1.0/24
```

Použivatelia Gentoo budú mať po inštalácii k dispozícii rc skript:

```
# rc-update add distccd default
# /etc/init.d/distccd start
```

Parameter `--allow` je veľmi dôležitý. Určuje, ktoré IP adresy budú mať prístup k `distccd` serveru. Použivatelia Gentoo si tento a ďalšie parametre `distcc` môžu nastaviť v `/etc/conf.d/distccd`.

Nakoniec na klientskom počítači nastavte zoznam a limity pre vzdialené servery. To sa robí pomocou premennej prostredia `DISTCC_HOSTS`:

```
# export DISTCC_HOSTS="localhost/2 192.168.1.1/2 superserver.com/8"
```

V Gentoo použijete konfiguračný nástroj `distcc-config`:

```
# distcc-config --set-hosts "localhost/2 192.168.1.1/2 superserver.com/8"
# env-update
# source /etc/profile
```

## Záver

Ešte musím upozorniť, že `ccache` je citlivý na to, či používate `distcc` alebo nie. Ak ste prvú verziu kompilovali distribuovane a pri kompilácii novej verzie už nemáte k dispozícii viac počítačov a `distcc` vymažete za súboru `/etc/make.conf`, `ccache` si bude myslieť, že ste zmenili verziu `gcc` a medzivýsledky kompilácie vo svojej databáze jednoducho nenájde.

Ak hrozí, že nebudete môcť kompilovať na viacerých počítačoch pomocou `distcc`, je lepšie `distcc` do súboru `/etc/make.conf` nedávať a spúšťať distribuované kompilovanie nasledovne:

```
# FEATURES="distcc" emerge parametre
```

## Odkazy

[1] <http://www.eriky.de/compilercache/>

\*\*\*

# Real-time modifikace Linuxu

Vít Pelčák

---

*Prozkoumání možností modifikací operačního systému GNU/Linux pro práci v reálném čase. Zaměřeno především na plánování úloh reálného času v běžném Linuxu. Analýza vlivů několika variant preemptivního jádra na časové odezvy procesů reálného času. Srovnání hard real-time linuxových modifikací (RTLinux, RTAI). Analýza vlivů několika variant preemptivního jádra na časové odezvy procesů reálného času. Srovnání hard real-time linuxových modifikací (RTLinux, RTAI). Praktická aplikace.*

---

## 1. ÚVOD

Cílem seriálu je prozkoumání možností modifikací operačního systému Linux pro práci v reálném čase, srovnání hard real-time variant Linuxu a porovnání vlivů variant preemptivního jádra na časovou odezvu procesů reálného času.

### 1.1 Co je to Linux

Operační systém GNU/Linux je svobodná implementace Unixu vyvíjená Linusem Torvaldsem a týmem programátorů a hackerů mající za cíl vytvořit operační systém podle norem POSIX a Single UNIX Specification. Linux má všechny vlastnosti, které se od moderního plnohodnotného unixového operačního systému očekávají. Je také šířen zdarma pod licencí GNU/GPL a jsou u něj k dispozici zdrojové kódy.

### 1.2 Probírané linuxové real-time modifikace

#### 1.2.1 RTLinux

RTLinux je komerční linuxová real-time modifikace, která kromě úpravy jádra přináší také sadu vlastních nástrojů a modulů, které dodatečně rozšiřují možnosti jádra. Snaží se co nejvíc přiblížit POSIX 1003.1 Real-Time standard specifikaci.

#### 1.2.2 RTAI/fusion

RTAI/fusion je v podstatě přídatný modul, který využívá služeb HAL (Hardware Abstraction Layer), což je abstraktní vrstva nad hardwarem. Snaží se o přesunutí běhu procesů z prostoru jádra do uživatelského prostoru, a taky dokáže tyto nelinuxové systémy emulovat. Tím lze zjednodušit přechod aplikací z jiných real-timeových operačních systémů na systémy založené na Linuxu. RTAI/fusion je volně ke stažení na adrese <http://www.rtai.org/> [1].

#### 1.2.3 Realtime preempt patch

Je to úprava zdrojových souborů Linuxu, kterou vyvíjí Ingo Molnar, snažící se o řešení vysokých prodlev linuxového jádra a mimo jiné do jádra přináší několik variant preemptivity. Tato se projevila jako natolik užitečná, že některé její části jsou přejímány i do samotného linuxového jádra. Tato úprava je volně ke stažení na internetové adrese <http://people.redhat.com/mingo/realtime-preempt/> [2].



## 2. RTLinux

### 2.1 Úvod

RTLinux je hard real-time varianta Linuxu, která umožňuje ovládání robotů, data pořizujících systémů, zařízení citlivých na čas, výrobních dílen a dalších nástrojů.

1. verze RTLinuxu byla navržena pro provoz na levnějších a méně výkonných počítačích založených na x86. Poskytovala však pouze spartánské API a programovací prostředí.
2. verze RTLinuxu byla zcela přepsána. Podporuje symetrický multiprocessing, pracuje na větším rozsahu systémů a má spoustu rozšíření pro zjednodušení použití.

RTLinux nabízí možnost běhu speciálních real-time úloh a obslužných rutin přerušení (interrupt handlers) na stejném PC jako standardní Linux. Tyto úlohy a obslužné rutiny se vykonávají podle potřeby bez ohledu na to, co Linux právě dělá. V nejhorším případě je čas mezi okamžikem, kdy je procesorem detekováno přerušení, a okamžikem, kdy začne pracovat rutina obsluhující toto přerušení, minimálně 15 mikrosekund. Periodické úlohy v RTLinuxu na stejném hardware běží během 25 mikrosekund od naplánovaného okamžiku. Tyto časy jsou omezeny hardwarem, a jak se hardware vyvíjí, tak se zlepšuje i výkon RTLinuxu. Standardní Linux má skvělý průměrný výkon a může dokonce poskytovat řádově milisekundovou přesnost plánování úloh použitím POSIXových soft real-time schopností. Standardní Linux není navržen tak, aby poskytoval přesnost pod milisekundu a spolehlivé časovací záruky. RTLinux druhé verze je strukturován jako soubor volitelných komponentů doplňujících jádro. Jádro povoluje instalaci obslužných rutin přerušení s velmi nízkou prodlevou (Low Latency). Tento hlavní komponent byl rozšířen o podporu SMP (Symetric MultiProcessing) a zároveň je zjednodušen odebráním některých vlastností, které mohou být poskytovány jiným způsobem. Hlavní funkčnost RTLinuxu spočívá v kolekci modulů poskytujících volitelné služby a úrovně abstrakce. Tyto moduly zahrnují:

1. rtl sched – plánovač priorit, který podporuje „jako POSIXové“ rozhraní a původní v1 RTLinux API.
2. rtl time – který řídí hodinový signál procesoru a exportuje abstraktní rozhraní pro připojení obslužných rutin na hodinový signál.
3. rtl posixio – podporuje POSIX styl read/write/open rozhraní pro ovladače zařízení.
4. rtl fifo – propojuje RT úlohy a obslužné rutiny přerušení k linuxovým procesům přes vrstvu zařízení, takže linuxové procesy mohou číst a zapisovat do RT komponent.
5. semaphore – dává RT úlohám blokovací semaforey (synchronizační objekty, které slouží k synchronizaci přístupu ke sdíleným prostředkům). Podpora POSIXových mutexů bude taky k dispozici.
6. mbuffer – poskytuje sdílenou paměť mezi RT komponenty a linuxovými procesy (Yodaiken, V., Barabanov, M.).

Klíčovým cílem návrhu RTLinuxu je transparentní, modulární a rozšiřitelný systém. Transparentnost znamená, že zde nejsou žádné neotevratelné černé skříňky a cena každé operace by měla být stanovitelná. Modularita znamená, že je možné vynechat nepotřebnou funkci a ušetřit tak výkon. Základní RTLinux systém podporuje pouze vysokorychlostní obsluhu přerušení. Rozšiřitelnost znamená, že programátoři by měli být schopni přidat moduly a upravit si systém podle vlastních požadavků.

### 2.2 POSIX

#### 2.2.1 Základní pojmy

◊POSIX [3] 1003.1 Real-Time standard je pro hard real-time stejně nedosažitelný jako nutný. Nedosažitelný je kvůli přímé implementaci například POSIXového souborového systému. Má požadavky,

které real-time systém nemůže poskytnout. Nutný je proto, že nabízí jediné široce používané rozhraní pro real-time, zařizuje propojení a přesun software mezi real-time a ne-real-time POSIXovými systémy.

RTLinux v2 se přibližuje POSIX real-time specifikacím následováním modelu pro POSIXový jednoprocessorový/minimální real-time systém s některými rozšířeními pro SMP. POSIXový návrhový standard definuje „Minimální real-time systémový profil“ (PSE51), který je zamýšlený pro hard real-time systémy jako RTLinux. V sekci „Odůvodnění“ v POSIX AEP dokumentu je definováno, že „vláknový model POSIX.1c (se všemi zapnutými volbami, ale bez souborového systému) nejlépe odrážel současnou praxi v jistých real-time embedded oblastech. Namísto podpory plného souborového systému je pro jádra takové velikosti považována za dostatečnou pouze základní podpora I/O zařízení (read, write, open, close, control).“

## 2.2.2 Plánování

Plánovací modul ve verzi 2 považuje plánovač a soubor real-time úloh za jeden POSIX proces s úlohami odpovídajícími POSIXovým vláknům. Na SMP systému (clusteru) můžeme mít několik paralelně běžících plánovačů a každý vypadá jako POSIX proces.

## 2.2.3 Posixio

RTLinux 2.0 taky přináší posixio modul, jenž poskytuje standardní rozhraní read/write/open vstupně-výstupních operací pro ovladače s POSIX stylem. Problémem pro podporu POSIXového souborového API je, že otevírání v POSIXovém souborovém systému není vnitřně v reálném čase. Otevření souboru totiž může vyžadovat neohrazené přechody prostorů jmen (unbounded traversal of the namespace), následování symbolických odkazů, rozlišování adresářů a křížení přípojných bodů. Autoři POSIX standardu naštěstí povolují velmi omezenou verzi pro otevírání. Jediné názvy cest podporované RTLinuxem jsou ve tvaru `/dev/name`. Jediný podporovaný mód je read/write (Yodaiken, V., Barabanov, M.).

## 2.2.4 Signály

Jedna oblast POSIX standardu, která ještě nebyla implementována, je požadavek asynchronních signálů. „Signální služby jsou základní mechanismus na POSIX standardu založených systémů a jsou potřeba pro manipulaci s událostmi a řešení problémů.“ Současný záměr je udělat ze signálů volitelnou součást. Užitečnost obecného signálního mechanismu v hard real-time prostředí není vůbec jasná. Účelem takového mechanismu je přerušit tok řízení vlákna a vnutit jej do rutiny obsluhující chyby a události. Real-time úlohy by měly provádět jenom jednoduché operace. Ovládání signály se tedy zdá být nesmyslné. S událostmi může být manipulováno rutinami obsluhujícími události – rutiny obsluhující hardwarová přerušení nebo funkce softwarových událostí. Pokud může událost ukončit dlouho běžící operaci, rutina obsluhující události by měla pozastavit úlohu a možná i zavolat plánovač. A co třeba chyby v plovoucí čárce? Pravidlo RTLinuxu číslo jedna praví, že ne-real-time služby by neměly být poskytovány real-time komponentami. Pokud real-time úloha používá plovoucí čárku, měla by raději explicitně kontrolovat chyby, než dovolit nejhorší případ přerušení v plovoucí čárce. Takže real-time úlohy jsou považovány za komponentu zajišťující kompatibilitu a ne za hlavní komponentu. Na druhou stranu RTLinux nabízí mnoho signálových schopností v jiných formách.

## 2.3 Shrnutí

RTLinux je testovaný a ověřený hard real-time, POSIX operační systém, který používá embedded Linux jako aplikační platformu. RTCore real-time jádro, které je srdcem RTLinuxu, poskytuje celistvost, rychlou odezvu, malé odchylky od plánování (scheduling jitter) a hladký přístup k Linuxu.

*Charakteristiky RTLinuxu:*

- *Hard real-time výkon* – žádná překvapení nebo tichá selhání, jak se konfigurace systému mění s nárůstem zátěže. Krajní lhůty jsou vždy dodrženy.

- *POSIX API*– 1003.13 PSE51 pro real-time a plný přístup do Linuxu prone-real-time programy.
- *Oddělený design* – dual-kernel technologie zvedá výkon zabráněním ne-real-time procesům v překážení si s real-time procesy. Design RTLinuxu podporuje modularitu a opakované využití kódu, zatímco podává výkon omezený pouze hardwarem.
- *RTLinux* – kompletní vývojový systém s testovaným jádrem a souborem nástrojů, plně zapouzdřitelný, přes síť bootovatelný souborový systém. Jako obvykle jsou obsaženy veškeré věci, které jsou v Linuxu obvyklé jako např. plná podpora TCP/IP, X-Window System GUI a vývojové nástroje.

## 3. RTAI

### 3.1 Úvod

RTAI znamená Real Time Application Interface (Aplikační rozhraní v reálném čase). Nejedná se o real-time operační systém, jako například VXworks nebo QNX. Je založen na linuxovém jádře a poskytuje možnost udělat jej plně preemptivním. Linux trpí nedostatkem podpory reálného času. K zajištění správnosti časování je nutné provést v jádře několik změn, např. v zacházení s přerušeními nebo s metodami plánování. Takto můžeme získat real-time platformu s nízkou prodlevou a plnicí náročné požadavky na předvídatelnost v plném ne-real-time prostředí Linuxu (přístup k TCP/IP, grafické zobrazení a okenní systémy, souborové a databázové systémy atd.). RTAI nabízí linuxovému jádru stejné služby přidáním vlastností real-time operačního systému. Skládá se hlavně z rozvrhovače přerušení (interrupt dispatcher): RTAI hlavně odchyťává přerušení periférií a v případě nutnosti je přesměrovává do Linuxu. Není to ale přímo modifikace jádra, nýbrž přidává k jádru vlastní modul, který využívá konceptu HAL (Hardware Abstraction Layer) k získání informací z Linuxu a k zachytávání některých nezbytných funkcí. To vede k jednoduché adaptaci v linuxovém jádře, jednoduchý RTAI port z verze na verzi Linuxu a jednodušší použití jiných operačních systémů namísto RTAI. Když se neobjevuje žádná real-time aktivita, RTAI považuje Linux za úlohu běžící na pozadí.

#### 3.1.1 Úvod do RTAI/fusion

Fusion je pokračující vývojová větev projektu RTAI, která se zaměřuje na vyplnění mezery mezi tradičním přístupem přes jádro pro dosažení omezeného chvění v nejhorších případech a dlouholeté úsilí o snížení průměrné prodlevy původního *vanilla* [4] jádra Linuxu.

Prvním cílem RTAI je odsunout aplikace vyžadující real-time záruky z prostoru jádra (kernel-space), kde byly při starém způsobu přístupu s využitím pomocného jádra uzamčeny. Namísto toho jim umožňuje využít běžný linuxový programovací model v uživatelském prostoru (user-space), zatímco bude stále garantováno omezení nejhoršího zpoždění. RTAI/fusion se zaměřuje na doplnění podpory low latency pro velmi náročné real-time aplikace, které vyžadují aby maximální chvění zůstalo bez výjimky v rozsahu pár desetin mikrosekund za jakýchkoliv okolností a musí být přitom provozuschopné v uživatelském prostoru (user-space) (Dietrich, S. T., Walker, D.).

Druhý hlavní cíl RTAI/fusion je zjednodušení přechodu aplikací běžících v tradičních RTOS (jako např. VxWorks, pSOS+, VRTX a podobně) na systémy založené na Linuxu. To je možné díky poskytnutí efektivních emulací API nad RTOS abstrakční vrstvou. Jelikož je velká většina těchto systémů implementována podobným způsobem, je možné vymodelovat obecný set základních vlastností a později je podle libosti upravit pro napodobení různých API.

### 3.2 Časovače a přerušení

Dobrá správa časování a přerušení znamená pro systémy reálného času skutečnou výzvu. Ale jak mohu do PC časovat? Co to je přerušení, a jak jej mohu řídit? Následující text se bude vztahovat k Intel x86 architektuře.

UP (jednoprocessorová architektura) poskytuje specifický čip, který řeší problém generování přesných časových zpoždění pod softwarovou kontrolou. Je označen 8254 a je to programovatelný intervalový

časovač/čítač, který může být brán jako sada čtyř I/O portů. Tři jsou nezávislé 16bitové čítače a čtvrtý je kontrolní registr pro programování módů.

RTAI poskytuje dva módy, periodický (mode 2 z 8254) a jednorázový časovač (mode 0). V jednorázovém módu jsou hodiny přeprogramovány s každým přerušením. Naproti tomu v periodickém módu je časovač naprogramován jenom na začátku a potom periodicky generuje přerušování. Je na programátorovi, který mód si pro daný úkol vybere. Periodický mód je o hodně efektivnější, pokud se má vykonávat jedna nebo mnoho úloh (ale se společnou periodou), které pravidelně vzorkují, zatímco jednorázový mód je flexibilnější, protože dovoluje časovat několik úloh bez společného dělitele časových úseků nebo spuštění některou vnější událostí. V jednorázovém módu ve RTAI je čas měřen na základě hodin procesoru a ne na čipu 8254, který je použit jenom ke generování jednorázového přerušování. To dovoluje přeprogramovat čítač pouze dvěma I/O instrukcemi, což trvá přibližně 3 ms.

### 3.3 Základní koncepty

RTAI/fusion definuje dva real-time módy operací pro linuxové úlohy, které ovládá:

- Primární mód garantuje velmi nízké prodlevy přenecháním kontroly real-time úlohy plánovači (co-scheduler), jehož operace nemohou být zpožděny žádnou pravidelnou aktivitou Linuxu, včetně maskování přerušování. V tomto módu se úloha jeví jako uspaná v TASK\_INTERRUPTIBLE stavu, ale ve skutečnosti běží ve svém původním MMU kontextu pod kontrolou co-scheduleru. Úloha vstupuje do tohoto módu po inicializaci a pokaždé volá blokování nativní RTAI služby. Úloha tento mód opouští, aby vstoupila do sekundárního módu, jenom když vykonává standardní linuxové systémové volání. Typická plánovací zpoždění v tomto módu jsou v současné době v nejhorším případě asi 50 mikrosekund při velkém zatížení na středně výkonném x86 hardware. Některá obvyklá linuxová systémová volání mohou být obsloužena prostými doplňky RTAI. Například systémové volání na nosleep je propojeno do vysoce přesného RTAI časovače, jenž má vyšší přesnost než běžný systémový časovač a má velmi nízkou plánovací prodlevu. Úloha stále běží v primárním módu (nebo se do něj přepne), když je taková nahrazená služba zavolána (Dietrich, S. T., Walker, D.).
- Sekundární mód stále garantuje nízká zpoždění, nicméně v nejhorším případě jsou tyto hodnoty vyšší. Tento mód odkládá přerušování, která jsou zpracována Linuxem, tak dlouho dokud real-time úloha běží, aby nedošlo ke zdržení vykonávané úlohy. Poslední jmenovaná technika velmi zlepšuje předvídatelnost úloh běžících v sekundárním módu bez toho, aby nějak ovlivnily prodlevu přerušování pro ostatní úlohy operující v primárním módu.

Linuxové úlohy řízené RTAI/fusion jsou transparentně a automaticky přepínány mezi oběma módy podle úrovně real-time služby, o kterou žádají. Real-time priority jsou trvale udržovány přes tyto přesuny, takže real-time úloha řízená jádrem Linuxu by měla mít podle potřeby stále zajištěnou vyšší prioritu než ostatní úlohy, které jsou spravovány spoluplánovačem.

### 3.4 Software

RTAI/fusion je postaveno na vrstvě Adeos pro upřednostnění zpracování hardwarového přerušování a další implementaci způsobu spolupráce mezi real-time rozšířením a linuxovým jádrem.

Celý základ kódu RTAI/fusion se naprosto odchýlil od předcházející RTAI architektury a implementace, jelikož je založen na projektu Xenomai, který byl nedávno s RTAI sloučen. Důvodem je také konflikt s projektem RTLinux, jehož vývojáři tvrdí, že původní systém mají patentován.

### 3.5 Podporované architektury

RTAI/fusion v současné době běží pouze na architektuře x86 (UP a SMP), ale je také nově k dispozici port na platformu PPC (zatím jenom UP). Dodatečně vytvořil HYADES projekt RTOS jádro pro architekturu ia64 a plánuje se integrace tohoto portu, jakmile bude vydán.



## 4. REALTIME PREEMPT MODIFIKACE JÁDRA

V nových jádrech řady 2.6 se objevilo mnoho vylepšení reagujících na požadavky embedded systémů, včetně nového plánovače a preemptivního jádra.

Od jader této řady se toho, na základě pokroků v technologii plánování, očekávalo hodně.

Nový plánovač sice dosahoval dobrých výkonů, ale odezva byla trvale znehodnocována prodlevami, když úlohy přistupovaly ke sdíleným systémovým datům uvnitř kritického úseku.

Počáteční měření ukázala, že jádra řady 2.6 v preemptivním výkonu zaostávala za řadou 2.4. Taký ve zpracování audia překonávala jádra 2.4 některé rané verze 2.6.

### 4.1 Souběžnost v jednoprocessorovém jádře

#### 4.1.1 Kritické úseky a jejich správa

Během operací prováděných jádrem jsou sdílená data, porty a jiné hardwarové registry předmětem současného přístupu mnoha vláken a rutinami obsluhujícími přerušení.

Kritický úsek je část chráněného kódu, se kterým může operovat pouze jedna úloha, aby se zajistila integrita sdílených dat. Každý proces běžící v kritickém úseku musí tento opustit. Teprve pak je povolen přístup jinému procesu.

Souběžné přístupy do sdílených objektů jsou seřazovány podle hranic kritických úseků, a tak je za všech okolností zajištěna stabilita sdílených dat a operací s hardwarovými registry. V jádře existují dva základní typy kritických úseků:

1. Kritické úseky chrání vlákna před současnými přístupy jinými vlákny (preempt disable).
2. Kritické úseky chrání vlákna před současnými přístupy přerušeními (interrupt disable).

První a nejjednodušší druh kritického úseku potlačuje přepínání úloh pokud vlákno provádí operace uvnitř kritického úseku. Tato vlastnost zajišťuje, že vlákno může tento úsek opustit před tím, než je povolen přístup jinému vláknu.

Před tím, než vlákno přistoupí k datům sdíleným s rutinou obsluhující přerušení, jsou přerušení vypnuta a znovu jsou zapnuta až po dokončení aktualizace těchto dat. To chrání data před souběžným přístupem rutinou obsluhující přerušení poté, co vykonávající vlákno vstoupilo do kritického úseku. Pro základní správu kritických úseků v jednoprocessorovém prostředí jsou mechanismy preempt disable a interrupt disable postačující. V jednoprocessorovém prostředí na CPU vykonává operace pouze jedno vlákno, a to tak dlouho, dokud neopustí CPU samo, nebo je k tomu donuceno výjimkou procesoru jako třeba přerušením.

### 4.2 Souběžnost na SMP jádře a správa kritických úseků

Správa souběžnosti na SMP (symetric multiprocessing) jádře je výrazně komplexnější záležitost než je tomu u jednoprocessorového jádra, protože pouhé vypnutí preemptivity a přerušení už na ochranu dat sdílených procesory nestačí. To proto, že úlohy běžící na jiných procesorech nebudou vědět, jestli jsou tyto volby na CPU vypnuty, a mohou tak vstoupit do stejného kritického úseku zároveň. Z tohoto důvodu je nutné fyzicky alokovat sdílené zámky, které řídí přístup k jednotlivým kritickým úsekům. Tyto zámky, které chrání přístup do kritických úseků, se nazývají spinlock. Jsou alokovány v paměti a identifikují kritické úseky asociované se specifickými sdílenými daty. Tato architektura odděluje uzamykání pro nezávislá data. To dovoluje, aby mnoho vláken najednou mohlo pracovat v mnoha kritických úsecích jádra. Když úloha získá spinlock, je jí umožněn vstup do kritického úseku a je také zabráněno přepínání úloh v procesoru, dokud neproběhne spin unlock. Během držení spinlocku mohou být vypnuta přerušení (Gerum, P.).

## 4.2.1 Ochrana před jinými vlákny

Ochrana před vlákny a přerušováními je, až na několik rozdílů, obdobná jako u jednoprocessorové implementace.

Pokud vlákno v jádře SMP systému narazí na spinlock, vyvstane situace, kdy bude úloha usilovat o uzamčení a přístup do kritického úseku, který již bude blokován jinou úlohou. Vlákno usilující o přístup bude aktivně čekat (točit se na místě – spin) na uvolnění uzamčení (lock). Odtud plyne anglický název „spinlock“.

Pokud tedy bude vlákno na daném CPU aktivně čekat, nebude to CPU provádět žádnou práci. SMP implementace má vztah k plně preemptivnímu real-time jádru, protože jakýkoliv SMP kód je v principu plně preemptivní.

## 4.3 Analýza prodlev preemptivního jádra

Prodlevy preemptivity linuxového jádra řady 2.6 jsou složeny ze dvou hlavních prvků:

1. Prodlevy způsobené zpracováním přerušování.
2. Prodlevy způsobené zpracováním kritických úseků.

### 4.3.1 Prodlevy přerušování

Zpracováním přerušování může být vlákno pozdrženo tak dlouho, dokud se tato operace nedokončí. Také procesy se po tuto dobu nemohou přepínat. Prodlevy způsobené zpracováním přerušování mají 3 podskupiny:

1. Úseky s vypnutým přerušováním. Tyto mají na prodlevu jen minimální vliv.
2. Přednost zpracování přerušování před úlohami. To má velký vliv na prodlevu přepínání procesů obzvláště pokud dojde k velkému I/O zatížení.
3. softIRQ\* zpracování také velmi ovlivňuje prodlevu preemptivity. Zvláště softIRQ zpracování vztahující se k sítím je specifický zdroj výrazného zatížení preemptivní prodlevy.

\* Rutina obsluhující přerušování by měla být vykonána co nejrychleji a zároveň musí vykonat mnoho práce. Kvůli tomu se zpracování přerušování dělí na dvě části, tzv. poloviny. Samotná rutina obsluhující přerušování tvoří horní polovinu. Ta se spustí hned po přijetí přerušování a představuje jenom práci, která je časově kritická, jako je např. poslání potvrzení o přijetí přerušování nebo reset hardware. To, co se může vykonat později, je zahrnuto v dolní polovině. Ta proběhne později, když jsou všechna přerušování povolena.

### 4.3.2 Prodlevy závislé na úlohách

- Linuxové jádro během operace v kritickém úseku vypíná preemptivitu. Proto může být úloha s vysokou prioritou blokována jakoukoliv úlohou s nižší prioritou, která pracuje v kritickém úseku, a musí čekat dokud tato úloha daný kritický úsek neopustí.
- V SMP jádře může aktivita přerušování prodloužit čas potřebný k získání spinlocku. Přerušování prodlužují dobu, po kterou vlákno drží spinlock, a tak brání jeho předání jinému vláknu.
- Swapování může způsobit obrácení priorit, protože procesy s vysokou prioritou mohou být swapováním blokovány.
- Na jednoprocessorovém jádře je uzamčení jakéhokoliv kritického úseku ekvivalentní uzamčení všech úseků (Gerum, P.).

### 4.3.3 Shrnutí

V linuxovém jádře je přes 10000 nezávislých kritických úseků. Jejich identifikace je časově náročná a vyžaduje důkladné testování a analýzu. Omezování nejdelších kritických úseků je náchylné na chyby.

Podstatné zvýšení výkonu zpracování přerušování týkající se prodlevy preemptivity není dost dobře možné. Je nerozumné optimalizovat nepreemptivní kritické úseky jeden po druhém pro zlepšení preemptivity úloh.

Pro výrazné zvýšení preemptivního výkonu linuxového jádra je potřeba prozkoumat původní principy návrhu, jelikož se týkají zpracování přerušování a správy kritických úseků.

## 4.4 Plně preemptivní real-time jádro

U přerušovacího real-time subsystému je požadováno, aby vykonávání přerušování bylo upřednostnitelné (preemptible), aby dodržovalo priority, aby tyto priority přerušování byly alokovatelné na úlohové úrovni (tasklevel) v prostoru priorit a musí být možné povýšit prioritu real-time úloh nad zpracování přerušování.

### 4.4.1 Vylepšení real-time zpracování úloh

Nepředvídatelná zpoždění způsobená zpracováním přerušování jsou vyřešena přesunutím zpracování přerušování do kontextu k tomu vyhrazených úloh. Těmto úlohám lze pak přiřadit libovolné priority. Pouhým přesunutím softIRQ přerušovací aktivity a hardIRQ zpracování signálů do prostoru úloh (task space) nemůže být dosaženo žádných velkých zlepšení prodlev preemptivity na úlohové úrovni (tasklevel). K tomu je nutné snížit celkovou velikost a počet nepreemptivních kritických úseků. Principy návrhu plně preemptivního real-time linuxového jádra jsou založeny na rozšíření konceptů zpracování a zamykání, které již byly použity u SMP jádra (Gerum, P.).

### 4.4.2 Nezávislost kritických úseků

Kritické úseky jsou označeny unikátními jmény, a tak mohou v SMP za jistých podmínek CPU souběžně používat různé nezávislé kritické úseky. Tyto úseky mohou být chráněny alternativním uzamykacím mechanismem známým pod jménem mutex. To zajistí integritu dat a zároveň rozšíří spinlock operace v některých klíčových aspektech:

1. Pokud se narazí na uzamčený úsek, úlohy jsou pozastaveny a zařazeny do fronty v pořadí podle své priority. Po odemčení úseku jsou úlohy opět aktivovány.
2. Po obdržení mutex jádro asociuje uzamčený kritický úsek s úlohou uložením identity úlohy do zámku.
3. Kritické úseky mohou být ochráněny bez vypnutí upřednostňování (preemption).
4. Na úlohy operující v kritických úsecích je možno aplikovat dědičnost priorit.

### 4.4.3 Nahrazování Spinlocku

Pokud úlohy operují mimo kritické úseky, je linuxové jádro řady 2.6 už preemptivní. Nahrazení nepreemptivních spinlocků mutexy dovolí upřednostňování úloh uvnitř kritických úseků s real-time charakteristikami:

- Vlastnictví vláknem je v real-time jádře asociováno s každým uzamčeným mutexem.
- RT mutex poskytuje detekci zablokování a dědičnost priorit.
- O všech zámcích se předpokládá, že jsou preemptivní, včetně zámků deklarovaných v kódu ovladačů. To vylučuje možnost neefektivní implementace uzamykání, která může do systému zavést nedetekovanou prodlevu upřednostnění (preemptivity).
- Pouze malá kontrolovatelná množina zámků zůstane nepreemptivní.

### 4.4.4 Shrnutí

Odkládání zpracování přerušování do prostoru úloh a náhrada nepreemptivní implementace spinlocku mutexem více zatěžují CPU, ale to je cena za přizpůsobení preemptivního plánování úloh a přeměnu

univerzálního linuxového jádra ve vysoce preemptivní, protože plánovač zde může dát přednost kterémukoliv vlákně, i když se toto nachází přímo v kritickém úseku.

## 4.4.5 Vlastnosti Real-Time jádra

Návrh plně preemptivního linuxového jádra je založen na funkčním rozšíření nepreemptivních uzamykacích vlastností SMP jádra. Následně uvedený datový typ RT-Mutex poskytuje základ kompletní preemptivity, dědičnost priorit a detekci blokování. Robustností implementace SMP v linuxovém jádře, které již garantuje časovou nezávislost a správný chod v prostředí souběžných operací, se výrazně zjednodušuje real-time problém.

### Real-time mutex

Real-time mutex na jednoprocessorovém jádře umožňuje běh mnoha vláken v mnoha různých kritických úsecích.

#### 4.5.1 Dědičnost priorit

Aby byla zredukováána zpoždění způsobená aplikacím s vysokou prioritou, které se pokoušejí přistupovat ke kritickým úsekům a které již byly uzamčeny úlohou s nižší prioritou, byl původní real-time mutex vylepšen o dědění priorit.

Dědičnost priorit zvyšuje prioritu úlohy, která byla během práce v kritickém úseku předběhnuta na nejvyšší úroveň ze všech úloh čekajících na přístup do téhož kritického úseku.

Tranzitivní dědičnost priorit dovoluje vlákně s nejvyšší prioritou uplatnit prioritu přes více úrovní závislostí kritických úseků.

## 4.6 Benchmarky

Realtime preempt – patch vytvořený Ingo Molnarem nabízí 4 úrovně preemptivity:

1. No Forced Preempt – využívá pouze základní preemptivní linuxový model (toto je vhodné pro servery, kde není tolik třeba rychlá odezva ale především vysoká propustnost).
2. Voluntary Preemption – zde bylo přidáno několik explicitních preemptivních bodů. To redukuje prodlevu za cenu mírně nižšího výkonu.
3. Low-Latency – jádro je, kromě kódu pracujícího v kritickém úseku, plně preemptivní (vhodné pro desktop s rychlou odezvou).
4. Real-Time – od předchozí varianty se liší tím, že díky využití mutexu je preemptivní i kód provádějící operace v kritickém úseku. Je to na výkon nejnáročnější varianta a je zaměřena na Real-Time systémy.

Pro testování byla zkompileována 4 jádra verze 2.6.14. Pro každou úroveň preemptivity jedno. Tato jádra byla zkompileována ze stejnými parametry vyjma již zmíněné úrovně preemptivity.

### 4.6.1 Interbench

Na testování prodlev byl použit pro tento účel vyvinutý program Interbench (volně ke stažení na <http://members.optusnet.com.au/ckolivas/interbench/> [5]), který simuluje různé druhy zátěže v reálném čase a měří prodlevu při jejich vykonávání. Testování bylo provedeno v runlevelu 1, kdy běžely pouze nutné procesy, a počítač tak nebyl zatěžován dalšími úlohami, jež by znehodnocovaly výsledky testů.

Pro testování v režimu reálného času byl program spuštěn s parametrem `-r`. Během této sady testů se prováděl i Hack test, který byl ale natolik náročný, že způsobil pád programu. Aby se tento test neprováděl, byl vypnut parametrem `-x Hack`. Výsledný příkaz, kterým byl program spuštěn, tedy byl `./interbench -r -x Hack`. Typy zátěží simulované programem interbench:



**Write:**

Opakovaný streamovaný zápis souboru o velikosti fyzické paměti nadisk.

**Read:**

Opakované čtení souboru o velikosti fyzické paměti z disku.

**Compile:**

Simulace náročné kompilace na čtyřprocesorovém počítači prováděním testu Read, Write a zatížením CPU souběžně.

**Memload:**

Simulace velkého zatížení paměti a swapu opakovaným přístupem ke 110 % fyzické paměti, přesunování swapu do paměti a zpět a uvolňováním paměti.

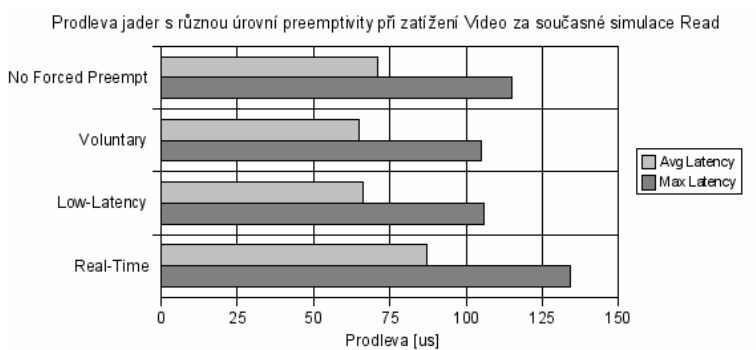
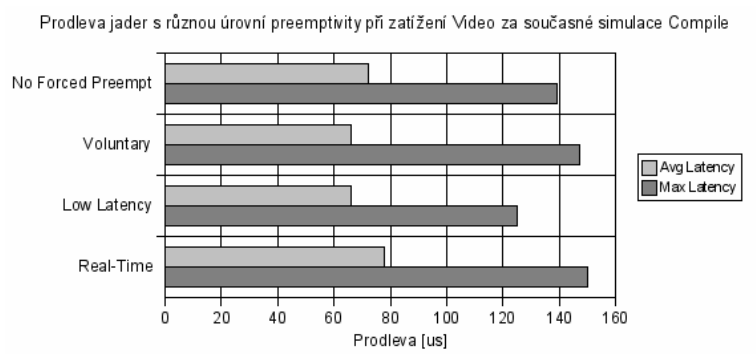
**Audio:**

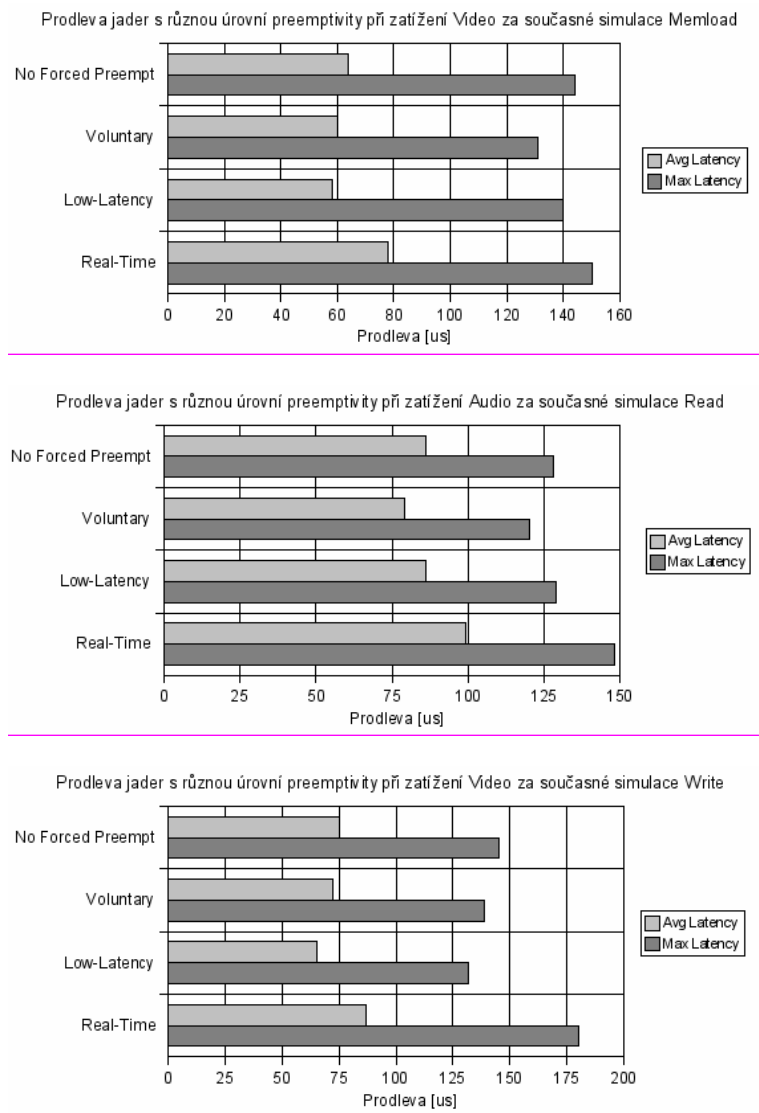
Audio je simulováno jako vlákno, které se pokouší běžet v 50 ms intervalech a pak vyžaduje 5 % výkonu CPU.

**Video:**

Video je simulováno jako vlákno, které se pokouší využívat CPU 60krát za sekundu a využívá při tom 40 % výkonu CPU.

Byla testována schopnost všech 4 variant jádra reagovat na požadavek o vykonávání úlohy za současného běhu jiné zátěže. Pro real-time benchmark se hodí zátěž zpracováním multimédií, protože se dá dobře nasimulovat a testovat. Třeba zatížení při softwarovém renderingu by v real-time oblasti nedávalo příliš smysl a neotestovalo by požadované vlastnosti jádra.





Z testů je patrné, že průměrná prodleva (světle šedé hodnoty) i maximální prodleva (tmavě šedé hodnoty) Real-Time varianty je překvapivě nejvyšší. Dá se to vysvětlit tak, že jak již bylo zmíněno, úpravy jádra jsou postupně přejímány do původního kódu jádra. Tím jsou výrazně omezeny jeho dřívější nedostatky. Zároveň se u této varianty projevila vyšší náročnost na výkon, což se negativně projevilo na výsledné prodlevě. Na druhou stranu by Real-Time varianta měla lépe garantovat splnění úlohy běžící v real-time režimu bez ohledu na to, jak je CPU zatíženo, a měla by být lépe předvídatelná.

## 4.7 Shrnutí

Vývoj této úpravy linuxového jádra dosáhl takové úrovně, že může konkurovat předcházejícím dvěma řešením a to jak nekomerčnímu RTAI, tak komerčnímu řešení RTLinux.

Navíc byla nedávno přidána vylepšená implementace časovačů jádra (Ktimers). To umožňuje real-time úlohám vzájemně spolupracovat s velmi nízkou prodlevou s vysoce výkonnými časovacími zařízeními s jemným rozlišením. Ty umožňují implementaci řídicích smyček v reálném čase s nebyvalou přesností. Také je umožněno použití RT mutexu v uživatelském prostoru přes standardní kernel API. Také úlohy v uživatelském prostoru jsou schopny využít efektivní dědičnosti priorit, detekce uváznutí a robustních protokolů pro jejich odblokování.

## 5. APLIKACE

Z výše zmíněných informací je zřejmé, že takto koncipované řídicí systémy najdou uplatnění v různých aplikacích. Jsou to zejména zapouzdřená (embedded) zařízení a taky zpracování zvuku a videa. Mimo

to parametry real-time verze Linuxu dovolují používat tento operační systém jako prostředek pro modelování a simulaci různých jevů.

## 5.1 Embedded systémy a jiné aplikace pro řízení

Real-time modifikace Linuxu je možno využít pro ovládání různých elektronických zařízení a jako Embedded zařízení. Embedded systémy jsou takové systémy, které jsou vestavěné do různých zařízení. Mezi vlastnosti, které může Linux nabídnout těmto kategoriím, patří:

- Podpora pro různé síťové, zvukové a grafické karty, různé typy sběrnic jako např. ISA, PCMCIA, PCI, AGP, PCI-E atd. Taktéž podporuje různé porty pro datovou komunikaci jako je USB, Bluetooth, LPT, IEEE 1394/ atd.
- Podpora různých platforem jako je AMD, Intel, Motorola, PPC, Alpha, Sparc, Crusoe atd. Je použitelný na uniprocessorových počítačích, ale také nabízí podporu pro SMP.
- Plně podporuje protokol TCP/IP, tedy je možné na něm provozovat WWW servery, firewally a další aplikace vyžadující tento protokol.
- Významnou vlastností pro embedded systémy je modularita, tzn. je možné za běhu připojovat moduly do jádra a tím i rozšiřovat možnosti jádra. Tato vlastnost také zajišťuje přístup téměř ke všem zařízením připojeným k systému.
- Existuje mnoho nástrojů a možnost výběru z několika programovacích jazyků, které byly na platformu Linux portovány jako např. Assembler, Perl, Python a Fortran.
- Běžně je Linux vybaven překladačem jazyka GNU C (obsahuje kompletní překladač pro C, C++ a objektové C).
- minimální nároky pro systém jsou:
  - procesor i386 (běží v 32bitovém chráněném režimu)
  - 4 MB paměti
  - zařízení pro zavedení systému – jakékoliv zařízení, které umí obsloužit BIOS případně flash paměť
  - minimální systém je možné umístit na 1.44 MB disketu i s dostatkem nástrojů pro formátování a inicializaci souborového systému na pevném disku a k přenesení systému na pevný disk.
- Jádro samotné sice není plně preemptivní, ale výše zmíněné varianty úprav linuxového jádra tento problém velmi úspěšně řeší.
- Systém je volně šířený pod licencí [GPL](#) [6] – neplatí se žádné licenční poplatky a je tedy možné jej volně stáhnout z internetu.
- Systém Linux je vysoce stabilní a prověřený operační systém.

## 5.2 Multimediální aplikace – JACK

JACK je audio server s nízkou prodlevou běžící na Linuxu a jiných systémech vyhovujícím normě POSIX. Může propojit několik různých audio aplikací a taky jim umožnit vzájemné sdílení audio dat. Tyto aplikace (klienty) mohou běžet ve vlastních procesech jako normální aplikace nebo jako plugin uvnitř serveru JACK. JACK byl od základu navržen pro profesionální práci s audiem a jeho design se zaměřuje na dvě klíčové oblasti: synchronní běh všech klientů a operace s nízkou prodlevou.

## 6. Závěr

Všechny tři varianty podávají obdobný výkon. Je jenom na tvůrci systému, které variantě dá přednost (pouze realtime preempt ještě není úplně dokončená a vyladěná úprava). Nejzásadnější rozdíly jsou ve způsobu implementace real-time úpravy a ceně produktu.

RTLinux je komerční řešení a všechny real-time úlohy běží jako moduly v real-time jádře a používají jeho API. Proto se pro něj jinak programují aplikace. Naproti tomu pro Realtime preempt se úlohy programují jako pro původní Linux, protože tyto volají standardní linuxové API.

RTLinux a RTAI se navíc, na rozdíl od realtime preempt, který se stále ještě vyvíjí, už nasazují na real-time zařízeních.

Linux samotný má řadu výhod. Mezi nejvýznamnější patří:

- *Cena*  
Linux je šířen pod licencí GNU/GPL a je zdarma.
- *Dostupnost*  
Software je možné stáhnout z internetu a nebo si jej nechat nainstalovat a nakonfigurovat organizací, která se zabývá jeho distribucí a podporou pro uživatele.
- *Modularita*  
Za běhu je možné připojovat další moduly do jádra a tím rozšiřovat možnosti systému.
- *Optimalizovatelnost*  
Systém je možné zkompilovat pouze s požadovanými vlastnostmi na daný hardware, a tak efektivně zmenšit jeho velikost a zvýšit výkon.
- *Otevřenost*  
Linux je šířen jako tzv. open source, což znamená, že je dodáván se zdrojovými kódy. Do nich je možno nahlédnout a případně je i upravit podle potřeby.
- *Přenositelnost*  
Systém je možné přenést i na jiné hardwarové platformy s různými typy procesorů.
- *Stabilita*  
Linux je stabilní a prověřený operační systém, který dokáže běžet i na podprůměrném hardwaru.
- *Velikost*  
Minimální systém s potřebnými nástroji je možné nahrát na standardní disketu o velikosti 1.44 MB.

Celá oblast se nadále rozvíjí a výše zmíněné systémy jsou dále vyvíjeny a zdokonalovány. Samozřejmě také narůstá počet aplikací, kde je možné využít takto koncipované real-time operační systémy.

## Odkazy

- [1] <http://www.rtai.org/>
- [2] <http://people.redhat.com/mingo/realtime-preempt/>
- [3] <http://www.abclinuxu.cz/slovník/posix>
- [4] <http://www.abclinuxu.cz/slovník/vanilla>
- [5] <http://members.optusnet.com.au/ckolivas/interbench/>
- [6] <http://www.abclinuxu.cz/slovník/gpl>

\*\*\*

## Zprávičky

**1.2.2006**

*David Watzke*

Vyšlo KDE 3.5.1. Neobsahuje nové funkce, ale jen opravuje chyby z verze 3.5 (viz seznam změn). Zpráva o vydání, stažení.

**1.2.2006**

*Daniel Kvasnička ml.*

ArsTechnica informuje, že Google popřel informace o údajné Ubuntu-based distribuci nazvané Goobuntu (zprávička). Mluvčí Googlu uvedla, že firma Ubuntu používá, ale pouze pro interní účely a jeho distribuci neplánuje.

**1.2.2006**

*Daniel Kvasnička ml.*

Vyšla první ostrá verze nástupce Mozilla Suite – SeaMonkey 1.0. Důležitými novinkami jsou např. podpora SVG, možnost měnit pořadí panelů, autoscroll a další. Download (česká verze bude během týdne), poznámky k vydání.

**1.2.2006**

*Daniel Kvasnička ml.*

Nokia oznámila vydání nové verze Pythonu pro sérii S60. Důležitou změnou je uvolnění kompletního zdrojového kódu a to pod Apache licenci v2.0 a licencí Pythonu. Informuje OSDiR.

**1.2.2006**

*Nikola Ciprich*

Správcové Wikipedie zjistili velké množství nepravdivě zadaných nebo upravených údajů ve z IP adres vyhrazených americkému kongresu a senátu. Na základě toho byl celému rozsahu adres zakázán přístup. Více news.com.com.

**2.2.2006**

*Jiří Culka*

Poměrně nedávno projekt GNU začal vyvíjet svou bodnou implementaci populární technologie Macromedia Flash 7 – Gnash. Pokud chcete vědět, jak tento přehrávač zprovoznit jako plugin ve Firefoxu, pomůže vám návod na PCBurn.com. Popisuje sice postup na Mandrivě, ale s menšími úpravami ho lze použít i na jiných distribucích.

**2.2.2006**

*Jiří Culka*

Právě vyšel Firefox verze 1.5.0.1. Jedná se pouze o opravu chyb a vylepšení stability, uživatelům verze 1.5 je doporučován upgrade. Stahovat můžete z Mozilla.org.

**2.2.2006**

*Daniel Kvasnička ml.*

Pokud vás zajímá, jak v praxi vypadá cenzura, kterou Google provozuje na čínském trhu, následujte odkaz na jistý blog na Lupě.

**2.2.2006**

*Daniel Kvasnička ml.*

Před týdnem Novell poodhalil roušku připravovaného Novell Linux Desktopu 10, který by měl vyjít až společně s další enterprise verzí SuSE Linuxu. Informuje a podrobnější informace přináší PCWorld.com.

**2.2.2006**

*Daniel Kvasnička ml.*

Vyšla první beta kancelářského balíku KOffice 1.5. Mezi nejdůležitější změny patří např. nastavení formátu OpenDocument jako výchozího či nově přidané aplikace Kexi a KPlato. Balíky jsou k dispozici pro SuSE, Kubuntu a Debian Sid a Sarge. Oznamení, changelog.

**2.2.2006**

*Ivan Bibr*

Na katedře geoinformatiky PřF UP Olomouc proběhne v týdnu od 20. února osvětová akce nazvaná Týden open source. Vstup je zdarma (nutná registrace účastníků), přednášky jsou určeny především studentům a akademickým pracovníkům. Pořádáno ve spolupráci s Liberix, o.p.s.

**3.2.2006**

*Richard Chudoba*

ABCLinuxu má nyní už víc jak 10000 registrovaných uživatelů. První uživatelé se před lety registrovali ještě v hardware sekci na Penguinu.

**3.2.2006**

*Daniel Kvasnička ml.*

Článek na Linux-Watch se zamýšlí nad společností Sun a jejími plány na dvojí licencování Solarisu (CDDL & GPL3). Poukazuje také na důsledky, které by v této souvislosti mělo rozhodnutí Linuse Torvaldse také použít GPL3.

**3.2.2006**

*Daniel Kvasnička ml.*

Tým OpenOffice.org vyhlásil 1. února soutěž, jejímž cílem je zvětšit objem dostupné dokumentace pro uvedený balík a tak k němu otevřít dveře více lidem. Do soutěže je možno přihlašovat články o portování OOo na jiné platformy či popisy různých nových vlastností. Vítězové budou odměněni \$750.

**3.2.2006**

*Daniel Kvasnička ml.*

Na LinuxDevices.com se můžete podívat, jak vypadá vojenské vozidlo bez posádky řízené Linuxem. Jde o průzkumné, hlídkové a tažné vozidlo R-Gator a v jeho útrobách bije BlueCat Linux od LynuxWorks

**3.2.2006**

*David Watzke*

Vyšla nová verze Wine (0.9.7), která jako vždy opravuje různé chyby (viz zprávu o vydání).



**5.2.2006***Pavel 'lingeeek' Szalbot*

Celkem nepozorovaně vyšla nová verze IDE NetBeans 5.0. Nově obsahuje GUI builder Matisse, přepracována byla podpora CVS, nabízí podporu SAS 8.2, Weblogic9 či JBoss 4 a vylepšení se nevyhnula ani samotnému editoru.

**5.2.2006***Martin Tesař*

Po 18 měsících vyšla nová verze záchranného live CD SystemRescueCd, mezi novinky patří kernel 2.6.15.1 a podpora Reiser4 (ChangeLog).

**5.2.2006***Leoš Literák*

V pondělí 6. 2. kolem desáté večer dojde k několika hodinovému výpadku abclinuxu z důvodu stěhování serveru do house centra AbcHostingu. Děkujeme za pochopení, omlouváme se za případné potíže.

**6.2.2006***Daniel Kvasnička ml.*

Steven J. Vaughan-Nichols se na webu DesktopLinux.com zamýšlí nad tím, proč Adobe Photoshop suverénně vede v anketě Novellu o nejtoužebněji očekávanou aplikaci pro Linux.

**6.2.2006***Daniel Kvasnička ml.*

Také se honíte za detailními informacemi ohledně připravovaného Internet Exploreru 7? Na webu <http://www.ie7.com/> zcela jistě naleznete všechny informace potřebné k objektivnímu posouzení tohoto internetového prohlížeče.

**6.2.2006***Daniel Kvasnička ml.*

Eclipse Foundation přijala návrh firmy IBM na zapracování AJAX Toolkit Frameworku do své platformy pro tvorbu webů (Web Tools Platform). Na své si přijdou hlavně příznivci produktů Dojo, OpenRico a Zimbra. Informuje NewsForge.

**6.2.2006***Lukáš Zapletal*

Únorové číslo LinuxEXPRES věnované palmům a PDA zařízením v Linuxu je na stáncích. Časopis je dostupný také jako PDF v rámci elektronického předplatného, stáhněte si ukázkou.

**6.2.2006***Daniel Kvasnička ml.*

Tvůrci prohlížeče Opera ve své tiskové zprávě ohlásili podporu protokolu BitTorrent v příštích verzích svého produktu a to ve formě integrovaného vyhledávání a stahování torrentů. Vyzkoušet tyto vlastnosti budeme moci v testovací Opeře 9 TP2, která má podle CNETu vyjít již zítra.

**7.2.2006***Daniel Kvasnička ml.*

PyDev (Python IDE fungující jako plug-in v Eclipse) dospěl. Včera vyšla verze 1.0 a dnes ráno bugfix v1.0.1. Vylepšení se týkají v první řadě debuggeru, managementu projeků, utility PyLint a dalších. Nově také vznikl projekt PyDev Extensions, který doplňuje IDE o další funkce.

**7.2.2006***Robert Krátký*

Ben Goodger, vedoucí vývojář Firefoxu, napsal o historii Mozilly a Firefoxu. Popisuje své zaměstnání v Netscape, vztah Netscapu a projektu Mozilla, komunikační problémy v Netscapu a redesign UI. Where Did Firefox Come From?

**7.2.2006***Daniel Kvasnička ml.*

Autor článku na OSNews nahlíží na bezpečnost OS z hlediska běžného uživatele, pro kterého jsou fotky z Johnnyho prvního školního dne mnohem důležitější, než hromádka systémových konfiguračních souborů a podotýká, že v tomto ohledu jsou na tom Windows a UNIXové systémy stejně.

**7.2.2006***Daniel Kvasnička ml.*

Inspirován těmito návrhy na budoucí koncept GNOME, uveřejnil Jon „Despleda“ Manning (vývojář GNOME) svoje vlastní. Jednou z jeho myšlenek je úplně vypustit princip ukládání dat na plochu.

**7.2.2006***Aleš Kapica*

Tak a Opera 9.0 TP2 je venku. Stáhnout můžete přes <http://snapshot.opera.com/unix/>.

**7.2.2006***fipa*

OSS Alliance ve své tiskové zprávě upozorňuje na nejasnosti panující kolem formátu Microsoft Office Open XML.

**7.2.2006***Robert Krátký*

Steven J. Vaughan-Nichols na Linux-Watch představuje vyhledávač kódu pro programátory – Krugle. Programmers get their own search engine.

**8.2.2006***Robert Krátký*

Byli oznámeni finalisté soutěže Extend Firefox. Podívejte se, která z těch nejlepších rozšíření pro Firefox vám ještě chybí. Vizte také článek Rozšířte si Firefox!

**8.2.2006***Daniel Kvasnička ml.*

O'Reilly nám na svém portálu věnovaném internetové telefonii představuje open-source VoIP řešení nazvané OpenZoep. Engine je navržený tak,

aby byl začlenitelný do co nejvíce druhů aplikací, třeba i do internetového prohlížeče.

**8.2.2006***Daniel Kvasnička ml.*

Server LinuxP2P.com přináší rozhovor s R. M. Stallmanem na téma peer-to-peer sítí. Řeč je o intelektuálním vlastnictví, DRM, iTunes a dalších.

**8.2.2006***Daniel Kvasnička ml.*

Článek na Tectonicu cituje a komentuje průzkum Gartnera, podle kterého největší konkurencí pro Novell na poli malých a středních podniků nebude Microsoft, ale Apple.

**8.2.2006***Robert Krátký*

Správce sbírek Tellico 1.1 přidává v nové verzi – mimo jiné – podporu pro export seznamů (knih) přímo do dokumentů OpenOffice.org a možnost automaticky aktualizovat záznamy podle zdrojů z internetu.

**9.2.2006***Daniel Kvasnička ml.*

Vyšla další z vývojových verzí programu The GIMP a to 2.3.7. Zdrojový kód je k dispozici na ftp.gimp.org, detailní seznam změn na developer.gimp.org a v případě, že se chystáte tuto verzi zkusit, neměly by vašim očím uniknout poznámky k vývojovým verzím.

**9.2.2006***Daniel Kvasnička ml.*

Vývojářský tým kolem OpenOffice.org chce po Sunu, aby se vzdal intelektuálního vlastnictví ve prospěch neziskové a nezávislé nadace. Současné vztahy Sunu a IBM totiž způsobují, že ačkoliv IBM v rámci produktu Workplace vyvíjí svůj vlastní fork, do OpenOffice.org nepřispívá. Informuje vnunet.com.

**9.2.2006***Daniel Kvasnička ml.*

Na Linux-Watch si můžete přečíst další ze série článků o kernelu a GPL3. Dozvíte se, že Linus Torvalds neodmítl GPL3 definitivně a také to, že hlavním problémem pro něj není stať týkající se DRM.

**9.2.2006***Daniel Kvasnička ml.*

Consortiuminfo.org reaguje na zde avízovaný článek o OpenOffice.org, Sunu a IBM. Autor blogu si došel pro informace přímo k aktérům sporu a kritizuje vnunet.com za nepřesné informace.

**9.2.2006***Martin Tesař*

Byl vydán Damn Small Linux 2.2; nová verze této miniaturní live distribuce (50 MB) se kromě

řady updatů vrací kvůli podpoře staršího hardwaru k jádru 2.4.26 (changelog).

**9.2.2006***Michal Křenek*

Poté, co Novell zveřejnil zdrojové kódy svých úprav, byl Xgl zařazen do X.Org CVS stromu. Nyní David Reveman (po předvedení schopností Xgl na konferenci XDevConf 2006) uvolnil i zdrojové kódy kombinovaného window a kompozitního manageru Compiz. To, co Xgl již nyní dokáže, je vskutku úchvatné. Pokud byste si chtěli Xgl vyzkoušet na vlastní kůži, můžete využít stránek opensuse.org/Xgl. Plně akcelerovaný X server je konečně tu!

**9.2.2006***Jindřich Pozlovský*

Na webu projektu SkyOS se objevila informace, že bylo na tento systém portováno Mono 1.1.13.

**9.2.2006***Mirek*

Vyšla nová verze binárních ovladačů pro grafické karty ATI. Řeší problémy s kernely řady 2.6.15 a problém s více jak 20ti řádky režimů monitoru v xorg.conf, který způsoboval zaseknutí systému při startu.

**10.2.2006***Robert Krátký*

Podle Warrena Woodforda, zakladatele distribuce MEPIS, možná dojde ke změně „mateřské“ distribuce. V budoucnu bude MEPIS pravděpodobně brát balíčky z Ubuntu místo Debianu.

**10.2.2006***Vencour*

Dle článku na RegHw možná bude ve Francii legální používat pro osobní potřeby sítě typu p2p. Francouzský parlament čeká diskuse, zda budou tito uživatelé platit dobrovolnou daň nebo přírážku 5 . Anthony G. byl žalován SPCP (Francouzským svazem nahrávacích společností) za sdílení 1875 souborů v síti Kazaa během roku 2004. Oblastní soud v Paříži odmítl souhlasit s argumenty SPCP; pro následující dva roky se soudci rozhodli tyto činy kvalifikovat jako „soukromé kopírování“.

**10.2.2006***Daniel Kvasnička ml.*

Na Linux.com si můžete přečíst recenzi na wiki-book s názvem Blender 3D: Noob to Pro. Jak už název napovídá, jde o knihu, která by vám měla pomoci stát se Blender profesionálem a překonat neobvyklosti tohoto 3D modelovacího a animačního programu.

**10.2.2006***Daniel Kvasnička ml.*

V blogu na PCWorldu si můžete přečíst zamyšlení nad tím, zda bude Songbird (přehrávač založený na enginu Firefoxu, jehož první betaverze vyšla před pár dny - zatím jen pro Windows) kvalitní konkurencí pro iTunes.

**10.2.2006***Daniel Kvasnička ml.*

Vyšla poslední verze prostředí GNOME z řady 2.12 – 2.12.3. Změn je mnoho a to ve všech třech kategoriích: platform, desktop a bindings. Zdrojové kódy jsou zde: platform, desktop a bindings.

**10.2.2006***Pavel 'lingeek' Szalbot*

Zůstat v obraze ve světě Javy vám může pomoci portál Java Posse. Bruce Eckel na jeho adresu poznamenal, že jeho komunita sdílí myšlenku nadřazenosti Javy, nicméně nabízí cenný náhled do oblastí, které se jen těžce sledují (např. nekonečné zprávy JSR).

**10.2.2006***Woodpecker*

Společnost Apple se rozhodla odměnit největší přispěvatele do Open Source projektu WebKit. Dvanáct nejaktivnějších dostane za odměnu jejich nejnovější přenosný počítač MacBook Pro a pět z nich dostane též pozvánku na letošní vývojářskou konferenci WWDC 2006.

**11.2.2006***Lukáš Jelínek*

Ve dnech 14. – 16.2.2006 se v Norimberku koná veletrh Embedded World 2006. Protože velká část bude letos věnována použití GNU/Linuxu v embedded řešeních, zájemcům vřele doporučuji veletrh navštívit.

**13.2.2006***David Jaša*

Měsíc se s měsícem sešel a v pátek se koná už páté setkání LVB (příznivců Linuxu v Brně), opět v restauraci Onyx.

**13.2.2006***Pavel 'lingeek' Szalbot*

V rámci mailing listu projektu Serendipity se objevila informace o tom, že hesla k účtům na BerliOS byla nějakou dobu veřejně dostupná. Objevitel na tuto skutečnost upozornil, nicméně kromě nápravy se BerliOS k incidentu nijak nevyjádřil (a dokonce posléze zrušil účet inkriminovaného vývojáře).

**13.2.2006***Daniel Kvasnička ml.*

Pokud by vás zajímalo, jak implementovat v Pythonu rozhodovací strom, v Python Devcentru na

ONLamp.com je k dispozici rozsáhlý článek na toto téma.

**13.2.2006***Daniel Kvasnička ml.*

V Ubuntu Wiki jsou k dispozici tzv. Test Plans – doporučené postupy testování připravované verze Ubuntu (Dapper Drake), které by měly pomoci zkvalitnit zpětnou vazbu od uživatelů a následně odstranit co nejvíce chyb.

**13.2.2006***Daniel Kvasnička ml.*

Intel poodhalil roušku svého připravovaného čtyřjádrového procesoru s názvem Clovertown, který by se měl ke koncovým zákazníkům dostat na začátku r. 2007. Více o tomto i dalších plánovaných procesorech Intelu v článku na CNET News.com.

**13.2.2006***Jáchym Čepický*

4.2.2006 proběhla v americkém Chicagu zakládající schůze nově vytvořené Open Source Geospatial Foundation. Mezi členy prozatímního výboru jsou Arnulf Christl (Mapbender, Německo), Chris Holmes (GeoServer, USA), Gary Lang (MapGuide, USA), Markus Neteler (GRASS, Itálie), Frank Warnmerdam (GDAL/OGR, Kanada). Setkání sponzorovala firma Autodesk.

**13.2.2006***Pavel 'lingeek' Szalbot*

Look&feel javovského Swingu ala Gtk nepatřil k nejzdařilejším. Před časem OSNews informovaly o tom, že nadcházející verze jazyka přinese nativní Gtk prvky a ...předběžný výsledek (Mustang je stále ve vývoji) si můžete prohlédnout.

**13.2.2006***Trained.Monkey*

Vyšla první beta minidistribuce Minimax. Obsahuje kernel 2.6.15, mc, links, centericq, instalátor pro Arch linux a spoustu dalších utilit. Livecd má pouze 32 MB. Detaily na kotek.net/minimax.

**13.2.2006***Jan Grmela*

Všichni, kteří jste až do teď byli nuceni snášet nekvalitní zvuk notebookových integrovaných karet, jásejte! Počínaje ALSA verze 1.0.11-rc2 (changelog, podporované karty) byla přidána podpora pro Creative Audigy 2 ZS Notebook. Momentálně funguje 7.1 výstup, na záznamu a digitálním výstupu se pracuje.

**14.2.2006***Daniel Kvasnička ml.*

Tvůrci MySQL v pondělí oznámili, že se jim v rámci posledního kola získávání financí podařilo sehnat 18,5 milionu dolarů. Stalo se tak díky



Intelu, SAPu, Red Hatu a dalším. MySQL plánuje použít finance pro vývoj dalšího produktu a rozšíření svých aktivit na trhu. Informuje Networkworld.com.

**14.2.2006***Daniel Kvasnička ml.*

Na Debian Administration najdete podrobný článek, který vás provede instalací a konfigurací frameworku Ruby on Rails v kombinaci s Apache serverem a vašim oblíbených databázovým systémem. Návod je pro Debian, ale nebude problém ho využít i v jiné distribuci.

**14.2.2006***Daniel Kvasnička ml.*

Na IT Managers Journal je k přečtení zajímavý článek o konvencích a slušném a hrubém chování v open-source kultuře: Recognizing rudeness in the open source culture

**14.2.2006***Michal Čihař*

Pěkný návod na zálohování notebooků (a vůbec počítačů, které nejsou neustále zapnuté) vyšel dnes na Debian Administration.

**14.2.2006***Jiří Hlinka*

Dnes vyšla nová verze 1.70.0 souborového manažeru Krusader. Oproti poslední betě, která vyšla před třemi měsíci, je v ostré verzi hlavně spousta opravených chyb.

**15.2.2006***Richard Chudoba*

Potřebujete občas v KMailu odstranit z emailu velkou přílohu, aniž byste museli smazat celý email? V bugzille projektu KDE se objevil workaround, který to řeší.

**15.2.2006***Daniel Kvasnička ml.*

Daniel Robbins, zakladatel Gentoo Linuxu, odešel po pouhých 8 měsících od Microsoftu, kde pracoval v Linux and Open Source Software Lab. Podle jeho slov se tak stalo proto, že neměl příležitost uplatnit veškeré svoje technické schopnosti. Více na ZDNetu.

**15.2.2006***Daniel Kvasnička ml.*

Google spolupracuje s firmou CodeWeavers Inc. na portaci aplikace Picasa na Linux. Jak je možno vyčíst z názvu druhé jmenované firmy, nepůjde o úplně nativní aplikaci, ale o kombinaci Wine a windowsové verze programu. Pokud bude Picasa úspěšná, další Googlí aplikace budou následovat. Více na DesktopLinux.com.

**15.2.2006***Daniel Kvasnička ml.*

Sun Microsystems splnil svůj slib a uvolnil specifikace procesoru UltraSparc T1 „Niagara“ jako open-source, aby usnadnil portaci Linuxu a BSD UNIXů na tento procesor. Použita byla licence GNU GPL v2. Informuje CNET News.com

**15.2.2006***Leoš Literák*

Dnes večer dojde k několikaminutové odstávce abíčka z důvodu upgradu RAM. Poslední dobou více swapovalo než vám zaslalo stránky. Děkujeme za pochopení.

**16.2.2006***Martin Tesař*

FTP server Opery už obsahuje finální verzi 8.52, zatím ještě bez oficiálního vyhlášení.

**16.2.2006***Daniel Kvasnička ml.*

Na NewsForge si můžete přečíst recenzi PC-BSD, jehož v1.0 je na spadnutí. Systém je založen na FreeBSD a určen spíše pro běžného uživatele. Autor se zaměřuje na instalaci, správu software i HW kompatibilitu.

**16.2.2006***Daniel Kvasnička ml.*

The Register informuje, že FSF je přesvědčena o Torvaldově nepochopení stati týkající se DRM v připravované GNU GPL v3. NewsForge na základě tohoto článku otce Linuxu kontaktoval, aby se ho zeptal na jeho současný pohled na věc. Odpověď byla celkem ostrá.

**16.2.2006***Katarína Machálková*

Pomozte vytvářet program dalších setkání Czech OpenSolaris Users Group. Jaká témata přednášek a seminářů vás zajímají? Vyjádřete se v anketě a budete zařazení do losování o ceny, které proběhne na prvním OpenSolaris bootcampu v Praze.

**16.2.2006***Daniel Kvasnička ml.*

Mark Shuttleworth, člověk stojící za Ubuntu Linuxem, už je viditelně unaven ze stále se opakujících otázek na téma Google & Ubuntu & Goobuntu a tak ve svém vlastním blogu píše: Absolutely no truth to the rumour...

**17.2.2006***Katarína Machálková*

Všichni příznivci operačního systému OpenSolaris jsou zváni na setkání CZOSUGu, které se uskuteční v úterý 21.února v 18:00 v pražské pobočce Sun Microsystems. Tématem bude tentokrát přednáška Martina Červeného na téma OpenSolaris v roce 2005 a jeho vývoj do budoucna, jakož i diskuze k programu připravova-

ného bootcampu na téma OS „Installation and Device Drivers Development“.

**17.2.2006***Daniel Kvasnička ml.*

Na SearchOpenSource.com si můžete přečíst rozhovor s Johnem H. Terpstra, spoluzakladatelem projektu Samba. Řeč je hlavně o schopnostech open-source světa pochopit potřebu uživatele a o faktu, že firmy chtějí od Linuxu a open-source víc, než jen snížení TCO.

**17.2.2006***Daniel Kvasnička ml.*

NewsForge představuje několik nepostradatelných rozšíření pro Firefox a Thunderbird. Je mezi nimi např. RSS čtečka Sage pro Firefox či GnuPG rozšíření Enigmail pro Thunderbird.

**17.2.2006***Daniel Kvasnička ml.*

Dobrou zprávu pro příznivce AJAXu, dnes velmi populární webové programovací techniky, přináší LinuxInsider. Informuje o tom, že firmy jako IBM, Novell, Oracle, Borland či Red Hat budou v rámci iniciativy Open AJAX přispívat kódem do různých open-source nástrojů pro práci s touto technologií a budou jejich používání propagovat.

**17.2.2006***Filip Korbel*

Přidali jsme 2U server SuperMicro SC822i do nabídky AbcHostingu. Rozšířit stávají 4 disky po 80GB je možné až na 6 disků. 1GB RAM DDR a Athlon 2500+ zajišťují dostatek výkonu i diskového prostoru.

**17.2.2006***David Watzke*

Vyšla nová testovací verze kancelářského balíku OpenOffice.org (2.0.2 RC1). Stahovat můžete také z oficiálního webu.

**17.2.2006***Honza*

Davyd Madeley přináší souhrn novinek, které bude obsahovat GNOME 2.14, jehož betaverze právě vyšla. Finální verze je plánována na 15. března.

**17.2.2006***Marek Nožka*

14. února vyšla nová stabilní verze perfektního e-mailového klienta Sylpheed. Sylpheed vyniká svou jednoduchostí, rychlostí a dobrou spoluprací s externími programy. Nová verze podstatně rozšiřuje funkčnost o vyhledávání v podsložkách a funkcí rychlého hledání. Příjemná je i tray ikona.

**17.2.2006***Mirek*

Vyšla nová verze „0.4“ distribuce Elive založené na Debianu. Změn je opravdu moc, byla zaktuali-

zována naprostá většina balíků. CD obsahuje dva nezávislé režimy pro liveCD a instalaci, do liveCD je možné nainstalovat jakýkoliv balík přímo pomocí apt-get nebo synaptic. Za pár hodin by měla být k dispozici přes torrenty.

**18.2.2006***Jan Grmela*

Ve středu se na hlavních stránkách Wine objevila informace o nové verzi 0.9.8. Tato verze mimo jiné zahrnuje lepší podporu webových prohlížečů, spoustu opravených chyb v Direct3D a kostru WordPadu.

**19.2.2006***Věroš Kaplan*

Deskovou hru s tématem softwarových patentů přinesl jako víkendový strip UserFriendly.org.

**19.2.2006***Petr Tomeš*

Ubuntu Flight CD 4 je připraveno. Jde o čtvrté z řady milníkových obrazů CD, které budou vydány během vývojového cyklu pro Dappera. Seznam důležitých změn tohoto vydání v rámci celé distribuce je dostupný na: [http://wiki.ubuntu.cz/Dapper\\_Flight\\_4](http://wiki.ubuntu.cz/Dapper_Flight_4).

**20.2.2006***David Majda*

CZilla vyhlašuje soutěž v zasílání zpráviček ze světa Mozilly, které jsou zobrazovány na titulní stránce jejího webu. Zprávičky zasílejte na adresu [soutez-zpravicky@czilla.cz](mailto:soutez-zpravicky@czilla.cz) budou průběžně zveřejňovány a tři nejaktivnější zprávičkáři budou odměněni. Podrobnosti a pravidla najdete na stránce o soutěži.

**20.2.2006***azurIt*

Vyšla verzia 1.2 programu SAT (Simple scAnning Tool). Ide o jednoduchý a rýchly sieťový skener, ktorý sa zameriava na identifikáciu vzdialených zariadení a služieb (napríklad na základe banerov). Changelog. Download.

**20.2.2006***Daniel Kvasnička ml.*

Nedlouho po uvolnění specifikací procesoru UltraSparc T1 „Niagara“ (zprávička se Dave Millerovi (vedoucí týmu programátorů starajících se v Sunu o Linux na Sparcu) povedlo na tomto procesoru zprovoznit Linux. Informuje ZDNet.

**20.2.2006***Daniel Kvasnička ml.*

Opravdová revoluce, ta, jejíž výsledky zakořenění a zůstávají, se děje potichu. Stejně tak je to s open-source modelem vývoje software a jednou to pochopí i Microsoft. Steven J. Vaughan-Nichols

z Linux-Watch to říká na rovinu: Open Source – the one, true way to develop software.

**20.2.2006**

*Daniel Kvasnička ml.*

GNOME a KDE soutěží o korunu krále linuxového desktopu už od r. 1997. Spolupráce na zdrojových kódech nebyla ničím neobvyklým, avšak propagaci free desktopu dělal každý tým po svém. To se však mění: KDE and GNOME collaborating on free desktop promotion (NewsForge).

**20.2.2006**

*Jan Kundrát*

Siemens/BenQ uvolnil svoji aplikaci Mobile Phone Manager pro Linux. Program je napsaný v Jave, jmenuje se XMPM a ke stažení se nabízejí i některé GPL komponenty. Roste na stránce o modelu C75, dále viz thread v Gentoo fóru či info na MobilMania.cz.

**21.2.2006**

*Daniel Kvasnička ml.*

Práce na implementaci prohlížení zpráv v panelech do poštovního klienta Thunderbird jsou u ledu. Dobrovolný vývojář mající v současné době práci na starosti nemá od konce ledna čas se jí věnovat a pokud se nenajde jiný, ve verzi 2.0 tuto vlastnost nenajdeme. Informuje ZDNet.

**21.2.2006**

*Daniel Kvasnička ml.*

Na stránkách jisté indické hostingové firmy je k přečtení několik postřehů k optimalizaci nastavení Apache serveru tak, aby byl jeho výkon co nejvyšší.

**21.2.2006**

*Adam Rambousek*

Adoptování tučňáků či jiných „opensource“ zvířat samozřejmě není jen český zvyk. Můžete se podívat, jak vypadá tučňák Mandriva.

**21.2.2006**

*Daniel Kvasnička ml.*

NewsForge informuje, že open-source VoIP program WengoPhone je teď možné stáhnout a používat jako rozšíření ve Firefoxu. Zajímavý je fakt, že ceny pro volání např. na pevné linky v ČR jsou nižší, než u Skype.

**21.2.2006**

*Daniel Kvasnička ml.*

Free Software Magazine uveřejnil převelice zajímavý rozhovor s neméně zajímavým člověkem: Markem Shuttleworthem, zakladatelem (nejen) Ubuntu Linuxu. Dočtete se například, že Mark byl první jihoafričan ve vesmíru a že se současným stavem Ubuntu stále ještě není úplně spokojený.

**22.2.2006**

*fipa*

Organizace OSS Alliance na svých stránkách zveřejnila velice zajímavou diplomovou práci, která se zabývá možnostmi nasazení OSS/FS ve veřejné správě ČR.

**22.2.2006**

*Daniel Kvasnička ml.*

OSDir cituje zprávu z Google Groups jistého Ezry Zygmuntowicze, který odhaluje své plány na live CD zaměřené na práci s Ruby/Rails a zve všechny zájemce, aby mu pomohli – ať už prakticky či jen svými nápady. Live CD ponese název Rubuntu a je tedy nad slunce jasné, která distribuce poslouží jako základ.

**22.2.2006**

*Daniel Kvasnička ml.*

OSN doporučila svým členům používat open-source software a to zejména v oblastech zdravotnictví, školství a mezinárodního obchodu. Toto doporučení tlumočil komisař OSN Dominique Ourdrago na II. mezinárodní open-source konferenci ve Španělsku. Informuje TMCnet.

**22.2.2006**

*Daniel Kvasnička ml.*

Už vás nebaví tahat se s myší z jednoho webu na druhý ve snaze vymáčkout z Internetu co nejvíce informací? Pak vás bude zajímat YubNub – portál, na kterém můžete s pomocí jednoduchých příkazů použít mnoho různých webových služeb k dosažení požadovaného výsledku. Představuje NewsForge.

**22.2.2006**

*Michal Jurosz*

Vyšel Parrot 0.4.2 „GPW“. Vývoj stále pokračuje: JIT je občas rychlejší než gcc, přibyly pojmenované parametry, bylo opraveno mnoho chyb a přibyly i nové maličkosti.

**23.2.2006**

*Martin Malec*

Security-portal, který se minulý měsíc potýkal s vážnými problémy, opět funguje, především však od včerejška má nový, doufejme že již stabilní hosting. S nadějí vzhlížíme ke světlejším zítřkům:-)

**23.2.2006**

*Daniel Kvasnička ml.*

Samsung Electronics ohlásil otevření zdrojového kódu ovladače pro flash paměť OneNAND, aby zjednodušil používání paměti v jiných produktech. Informuje DigiTimes.com.

**23.2.2006**

*Daniel Kvasnička ml.*

Tvůrci ReactOS (open-source OS snažící se o naprostou binární kompatibilitu s MS Windows) shrnují dění kolem projektu za posledních pár



měsíců a odpovídají i na nejdiskutovanější otázku o projektu: There is clearly NO code which is directly copy-pasted from MS kernel code.

**23.2.2006***Daniel Kvasnička ml.*

Pokud vás zajímá, jak pokračují práce na portaci Linuxu na Intel Mac, můžete si prohlédnout obrazy Knoppixu běžícího na novém iMacu.

**23.2.2006***Lukáš Zapletal*

Úplný přehled o možnostech připojení mobilních telefonů k Linuxu podává článek Aničky Bernáthové, který se zabývá zejména přístroji prodávanými v ČR.

**23.2.2006***Daniel Kvasnička ml.*

S klony Ubuntu Linuxu se v poslední době roztrhl pytel. Přibližně před týdnem byl ohlášen projekt Ebuntu: Ubuntu & Enlightenment DR17.

**24.2.2006***fipa*

Na serveru LINUXBIZWORLD.cz dnes vyšel rozhovor s hlavní postavou Mandriva Linuxu u nás, Ivanem Bíbrem.

**24.2.2006***Daniel Kvasnička ml.*

Rob Braun (vývojář jádra OpenDarwinu) tvrdí, že s příchodem Intel Macu umírají open-source aktivity firmy Apple. Děje se tak údajně proto, že Apple má strach z napomáhání „OSx86 piracy“. Apple nejdříve neuvolnil nástroje potřebné ke kompilaci Darwinu a teď to vypadá, že už nebude uvolňovat ani kód k xnu kernelu Mac OS X. Informuje Slashdot.

**24.2.2006***Filip Korbel*

AbcHosting naskladnil další servery. Nyní je k dispozici více jak 8 serverů a nově také Dell PowerEdge 350 s 512 MB RAM, CPU P3 850 MHz a dvěma HD za 1290,- Kč měsíčně. Doplňili jsme také dva kusy Sun Netra X1 také za 1290,- Kč.

**24.2.2006***Daniel Kvasnička ml.*

Linuxové jádro čeká nový DHCP klient. Hlavní myšlenkou konceptu, který má na svědomí vývojář kernelu Stefan Rompf, je automatické hledání nového připojení po odpojení od DHCP serveru. Odpůrci myšlenky tvrdí, že nový klient není potřeba. Více na Internetnews.com.

**24.2.2006***Daniel Kvasnička ml.*

Patentové šílenství pokračuje. Jisté kalifornské webdesignové firmě byl přiznán patent na „rich media presentation“ pokrývající prakticky každou implementaci moderních technologií jako je

Flash, AJAX, Java a další. Informuje Informati-onWeek.

**24.2.2006***Jan Grmela*

Po téměř nekonečných letech neaktivity a několikátého znovuoživení projektu byla včera večer vypuštěna release candidate verze 0.9.4 instant messaging klienta SIM. Prozatím není k dispozici changelog, můžete si však stáhnout tarball zdrojových kódů.

**24.2.2006***Peter Konecny*

Môžu mať vo svojej podstate niečo spoločné moderný operačný systém a prastarý ekosystém Zeme? Na LinuxOS.sk vyšla pekná glosa Linux, les a evolučné myšlienky, kde autor porovnáva neporovnateľné.

**24.2.2006***Adam Rambousek*

V Mandriva Linuxu bude integrována služba Mindawn – jediná služba pro online distribuci hudby, která podporuje Linux, a jedna z mála, která nabízí hudbu ve formátech Ogg Vorbis a FLAC. Podrobnosti se dočtete v tomto článku.

**25.2.2006***Martin*

Na zive.sk vyšel velice odvážný článek definující pro koho je vhodný GNU/Linux. Autor sebevědomě shrnuje vstup do linuxového světa a jeho výhody a nevýhody. Konečně celá pravda: Pre koho je určený Linux?

**25.2.2006***hajma*

První beta živého CD Mandriva Linux One 2006.0 je k dispozici ke stažení a testování.

**27.2.2006***Daniel Kvasnička ml.*

Firma PalmSource nedávno uvolnila detailnější informace o novém Palm OS. Kódové označení je ACCESS Linux Platform (ALP) a jde zatím pouze o prototyp v ranném stádiu vývoje. Více informací a screenshoty na Brighthand.com.

**27.2.2006***Daniel Kvasnička ml.*

Tom Adelstein na LXeru představuje firmu JBoss jako učebnicový příklad úspěšného open-source businessu a přeje si, aby více lidí kolem Linuxu tento model chápalo: JBoss is not Linux.

**27.2.2006***Daniel Kvasnička ml.*

Pokud vlastníte starší počítač a nevíte co na něj, relativně rozsáhlý článek na Linux.com představující pár distribucí vhodných pro starší stroje by vám mohl pomoci.

**27.2.2006**

*Daniel Kvasnička ml.*

Firma IBM by ráda v rámci svého dalšího tahu proti nechvalně známé SCO donutila Microsoft, aby odtajnil interní dokumenty týkající se jeho soupeření s Linuxem a navíc zpřístupnil úplné záznamy dohod mezi MS a SCO od června 2002. Informuje LinuxInsider.

**27.2.2006**

*Jan Kunderát*

Gentoo Linux 2006.0 je venku.

**28.2.2006**

*Daniel Kvasnička ml.*

SearchOpenSource.com nabízí poutavý rozhovor s britským open-source vývojářem a spoluautorem knihy Linux Desktop Hacks. Mluví o překážkách, které stojí lidem v cestě za Linuxem a zaměřuje se na tu nejtěžší: iracionalitu.

**28.2.2006**

*Daniel Kvasnička ml.*

Pokud byste rádi na svém Apache serveru hostovali web střežený dialogem žádajícím vás o jméno a heslo, můžete se inspirovat přehledným tutoriálem v blogu All About Linux.

**28.2.2006**

*Daniel Kvasnička ml.*

Mohutný článek ve Free Software Magazine má za cíl vás podrobně seznámit s X Window Systemem a to jak z teoretického, tak z praktického hlediska.

**28.2.2006**

*Daniel Kvasnička ml.*

Na OSDDir.com je již možné shlédnout screenshoty ze včera avízovaného Gentoo 2006.0.

**28.2.2006**

*Daniel Kvasnička ml.*

Ač se to některým nezdá, je mnoho věcí, které se můžou IT manažeři naučit od vývojářů software. Patří mezi ně metodologie vytváření dokumentace, SCM a další užitečné věci. Více v článku na ITManagersJournal.com.