



## Abíčko

Časopis serveru [abclinuxu.cz](http://abclinuxu.cz)

**Prosinec 2007**

---

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), rozcestník po ostatních linuxových serverech nebo velmi oblíbené blogy. Každý registrovaný uživatel si jej může založit a psát si do něj poznámky nejen o Linuxu.

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

Náměty na články zasílejte do konference našich autorů: [info@abclinuxu.cz](mailto:info@abclinuxu.cz). Sponzoring Abíčka a jiné formy reklamy si objednávejte na adrese: [info@stickfish.cz](mailto:info@stickfish.cz). Ostatní dotazy směřujte na adresu: [info@abclinuxu.cz](mailto:info@abclinuxu.cz).

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

©2007 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>Distribuční novinky – 48/2007</b>	<b>5</b>
Letem světem .....	5
Google zahajuje soutěž pro středoškoláky .....	5
Nigerijská firma žaluje One Laptop Per Child .....	5
KDE 4 se odkládá na leden 2008 .....	6
Úspěch OpenDocument Formatu roste .....	6
TrueBSD 2.0 RC1 .....	6
Elive 1.0.2 .....	7
Ubuntu 8.04 Alpha 1 Hardy Heron .....	7
Distribuční rada: změna obnovování repozitářů v openSUSE .....	8
<b>Distribuční novinky – 49/2007</b>	<b>9</b>
Letem světem .....	9
BusyBox žaluje Verizon kvůli porušování GPL .....	9
Sun bude oceňovat OSS vývojáře .....	9
Jak se staví Seagate k podpoře Linuxu .....	9
Poslední dění okolo OOXML .....	10
Trinity Rescue Kit 3.3 Beta .....	10
openSUSE 11.0 Alpha 0 .....	10
Frugalware Linux 0.8 Pre 1 .....	11
Geubuntu 7.10 .....	11
Distribuční rada: obnovujeme práva souborů na RPM systémech .....	12
<b>Distribuční novinky – 50/2007</b>	<b>13</b>
Letem světem .....	13
Knol: Wikipedie dle gusta Googlu .....	13
VCS Bazaar 1.0 je na světě .....	13
Nizozemí směřem k OSS a otevřeným standardům .....	13
eeeXubuntu – Ubuntu pro Eee PC .....	14
Parsix GNU/Linux 1.0r0 Test 1 .....	14
CRUX 2.4 RC1 .....	14
Mandriva Linux 2008 Spring Alpha 1 Barlia .....	15
Distribuční rada: zrychlujeme správce balíčků (openSUSE) .....	15
<b>Distribuční novinky – 51/2007</b>	<b>16</b>
Letem světem .....	16
Vývojáři Samby dostanou dokumentaci od Microsoftu .....	16
Norsko volí otevřené formáty .....	16
FSF Europe podporuje další prošetřování Microsoftu .....	16
SimplyMEPIS 7.0 .....	17
VectorLinux 5.9 .....	18
NetBSD 4.0 .....	18
Distribuční rada: monitorování uživatelů v reálném čase .....	19
<b>Svatá trojice vs. balíčkovací systém</b>	<b>21</b>
Kdy se tedy kompilace programu vyplatí? .....	21
Když je kompilace tak mocný nástroj, proč ji tedy nepoužívat? .....	21
Co jsou to knihovny a k čemu vlastně jsou? .....	22
Jak se liší vývojářská verze knihovny od běžné .....	22

Závěr .....	22
<b>Lokální útoky na operační systém linuxového typu</b>	<b>23</b>
Úvod .....	23
Exploitování programu .....	23
Lokální útoky – přetečení paměti .....	24
Paměť .....	24
Přetečení paměti .....	25
Přetečení zásobníku .....	26
Přetečení v segmentech halda a bss .....	27
Lokální útoky – formátovací řetězce .....	27
Formátovací řetězce a funkce printf() .....	27
Závěr .....	29
Literatura .....	29
<b>System vo firemnej sieti – VLSM</b>	<b>31</b>
Historia podsietí .....	31
CIDR bloky .....	31
VLMS .....	32
Príklad .....	32
Riešenie .....	33
Zaver .....	33
<b>Jaderné noviny – Video4Linux2 – 7 (ovládání)</b>	<b>34</b>
<b>Jaderné noviny – 44 a 45/2007</b>	<b>37</b>
Testování suspendu a hibernace .....	37
Citát: comp.os.linux větší než alt.sex .....	37
Swap přes NFS .....	37
Citát: Přetypování na void není příliš populární .....	38
Oprava chybných optimalizací překladače .....	38
2.6.24-rc2, „Spící město“ .....	38
Citát: Synchronizujte, než to zkusíte .....	39
Squashfs míří do hlavní řady jádra .....	39
Citát: Plánuješ žít věčně? .....	39
Subsystem distribuovaného úložiště dokončen .....	39
Citát: Navrhněte rozumné zamykací schéma .....	39
Bezpečnostní cíle AppArmor .....	40
Citát: Dobré pro vývoj, špatné pro obchod .....	40
Stabilní větve 2.6 .....	40
<b>Jaderné noviny – 46 a 47/2007</b>	<b>41</b>
Opravy plánovače .....	41
Distribuovaný síťový souborový systém Ceph .....	41
Zapisovatelný mmap pro FUSE .....	42
2.6.24-rc3, malé opravy a pročištění .....	42
Git ve Windows .....	42
CFS – backporty verze 24 .....	43
Modulární IO plánovače .....	43
Citáty .....	43
Andrew Morton: Nijak oslňující výkon .....	44
Theodore T'so: Paměti relativně zlevňují .....	44
David Miller: Vývoj ledovcovým tempem .....	44

Linus Torvalds: Zdarma dvě kopie Linuxu navíc .....	44
Theo de Raadt: Překvapivě svobodná licence .....	44
Alan Cox: Smysluplné výchozí hodnoty .....	44
David Miller: Kvalita hlášení o chybách .....	44
<b>Jaderné noviny – 7. 11. 2007</b> .....	<b>45</b>
Aktuální verze jádra: 2.6.24-rc2 .....	45
Citáty týdne: Ingo Molnár, Rusty Russell, David Miller .....	45
Správa paměti grafických procesorů .....	45
ID procesů ve světě s více jmennými prostory .....	47
<b>Jaderné noviny – 14. 11. 2007</b> .....	<b>49</b>
Aktuální verze jádra: 2.6.24-rc2 .....	49
Stále aktuální -mm strom .....	49
Citáty týdne: Ingo Molnár, James Bottomley .....	49
Různá témata týkající se kvality jádra .....	49
Lepší proměnné pro jednotlivé procesory .....	52
<b>Jaderné noviny – 20. 11. 2007</b> .....	<b>54</b>
Aktuální verze jádra: 2.6.24-rc3 .....	54
Citáty týdne: Greg Kroah-Hartman, Linus Torvalds .....	54
Systémové volání sys_indirect() pro nepřímé volání systémových volání .....	54
Jmenné prostory PID v jádře 2.6.24 .....	56
API pro uživatelský prostor .....	57
Interní API .....	57
Závěr .....	58
<b>Jaderné noviny – 28. 11. 2007</b> .....	<b>59</b>
Aktuální verze jádra: 2.6.24-rc3 .....	59
Citát týdne: Donnie Berkholz .....	59
Přiškrcení množství exportovaných symbolů .....	59
kmemcheck – kontrola neinicializované paměti .....	61
Aktualizace systémových volání .....	61
indirect() .....	61
timerfd() .....	62
hijack() .....	62
<b>Jaderné noviny – 5. 12. 2007</b> .....	<b>64</b>
Aktuální verze jádra: 2.6.24-rc4 .....	64
Citáty týdne: James Morris, Alan Cox .....	64
SEEK_HOLE nebo FIEMAP? .....	64
Návrat síťových kanálů .....	66
<b>Zprávičky</b> .....	<b>74</b>

## Distribuční novinky – 48/2007

Google zahajuje soutěž pro středoškoláky. Nigerijská firma žaluje One Laptop Per Child. KDE 4 se odkládá na leden 2008. Úspěch OpenDocument Format roste. TrueBSD 2.0 RC1. Elive 1.0.2. Ubuntu 8.04 Alpha 1 Hardy Heron. Distribuční rada: změna obnovování repozitářů v openSUSE.

Luboš Doležel

### Letem světem

Služba [Launchpad Personal Package Archive](#) <sup>(1)</sup>, která částečně napodobuje openSUSE Build Service, byla po několika měsících betatestování [vedena do provozu](#) <sup>(2)</sup>.

Vyšel [SUSE Linux Enterprise Real Time 10](#) <sup>(3)</sup> – distribuce s odezvou v reálném čase. Muslimové se dočkali nové varianty Ubuntu právě pro ně: [Ubuntu Muslim Edition 7.10](#) <sup>(4)</sup>. [Linux Caixa Mágica 12 RC1](#) <sup>(5)</sup> je portugalská linuxová distribuce – dříve založená na openSUSE, nyní na Mandrivě.

[IPCop 1.4.18](#) <sup>(6)</sup> opravuje podporu NTP, byl aktualizován software a uživatelské rozhraní. [StartCom Enterprise Linux 4.0.5](#) <sup>(7)</sup> přináší poslední novinky z Red Hat Enterprise Linuxu 4. [Shift Linux 0.5](#) <sup>(8)</sup> je živé CD Debianu, které je k dispozici v několika edicích podle desktopového prostředí.

[Damn Small Linux 4.1 RC3](#) <sup>(9)</sup> obsahuje nový textový editor, aktualizuje pciutils a opravuje různé chyby. [Pioneer Linux 3.0.2](#) <sup>(10)</sup> je nová verze distribuce založené na Kubuntu.

[Turbolinux 11 Server](#) <sup>(11)</sup> je japonská placená distribuce pro enterprise odvětví – nabízí dlouhodobou podporu, Zend Framework, bezpečnostní systém Tomoyo Linux, usnadnění migrace z Windows nebo Solarisu a další zajímavosti. Vyšlo [FreeBSD 6.3-RC1](#) <sup>(12)</sup>, po kterém bude následovat další release candidate a pak přijde konečná verze, pokud se neobjeví závažné problémy.

[ASPLinux 12](#) <sup>(13)</sup> je ruská distribuce založená na Fedoře – záměrem distribuce je, aby vše fungovalo hned po instalaci (zahrnuje proprietární ovladače, problémové kodeky atp.).

### Google zahajuje soutěž pro středoškoláky

Středoškolští studenti ve věku od 13 let mají příležitost [zúčastnit se nové soutěže](#) <sup>(14)</sup> Google nazvané [The Google Highly Open Participation Contest](#) <sup>(15)</sup>. Cílem je, aby i další lidé měli šanci naučit se něco o vývoji svobodného softwaru – nejde jen o programování, ale i o psaní dokumentace, provádění průzkumu mezi uživateli atd. Účastníci získají certifikát a tričko, pokud dokončí alespoň jeden úkol, za dokončení každých tří úkolů dostanou 100 dolarů, maximálně může každý získat celkem 500 dolarů. Tři vybraní se navíc budou účastnit slavnostního vyhlášení v sídle Google v Mountain View. Highly Open Participation Contest se účastní celkem deset open source projektů – seznam zahrnuje jména jako Apache Software Foundation, Drupal, GNOME, Mono, Python a jiné. Soutěž byla zahájena 27. listopadu, volit úkoly je možné do 22. ledna a práce bude ukončena 4. února.

### Nigerijská firma žaluje One Laptop Per Child

Nigerijská společnost Lagos Analysis Corp. [zažalovala](#) <sup>(16)</sup> projekt [One Laptop Per Child](#) <sup>(17)</sup> kvůli porušování patentů. OLPC prý okopírovalo jejich „inovativní“ klávesnice a navíc se dopustilo zpětného inženýrství jejich ovladače klávesnice – už při pohledu na [klávesnici této firmy](#) <sup>(18)</sup> se nabízí otázka, proč by někdo měl takové věci zapotřebí. Lagos chce, aby OLPC nahradilo způsobené škody, přestalo vyrábět a distribuovat své laptopy a přestalo používat údajně nelegální ovladač klávesnice. Firma tvrdí, že OLPC si za tímto

účelem zakoupilo dvě klávesnice značky KONYIN v americké a nigerijské verzi. Na vzhled klávesnic OLPC se pro porovnání můžete podívat na [Wiki](#)<sup>(19)</sup>. Najít znění porušovaného patentu je však prakticky nemožné a navíc se nejedná o žádnou převratnou invenci. Je symbolické, že Ade Oyegbola, zakladatel firmy Lagos, byl v roce 1990 usvědčen z bankovních podvodů a byl za ně na několik let poslán do vězení.

Server Boston.com [mezitím uvedl informace](#)<sup>(20)</sup> o prodejnosti notebooků OLPC. Prodejnost je nižší, než se původně očekávalo, ale Peru nyní uzavřelo smlouvu na nákup 260 000 výrobků a mexický miliardář Carlos Slim zakoupil 50 000 počítačů pro svou zemi. Od začátku programu Give One Get One (zaplať dva, jeden daruj) bylo veřejností zakoupeno 190 000 laptopů – někteří lidé si dokonce nenechají ani jeden notebook a darují oba dva. Ti, kteří si přístroj ponechají, dostanou od T-Mobile roční přístup k Internetu přes bezdrátové hotspoty zdarma. Souběžně s Give One Get One běží projekt Give Many, který přesvědčuje charitativní spolky, aby za přístroje pro rozvojové země zaplatily.

## KDE 4 se odkládá na leden 2008

Očekávané KDE 4 mělo původně vyjít už tento měsíc, ale ti, kteří vyzkoušeli poslední vývojové verze, jistě potvrdí, že software nebyl ve stavu vhodném pro vydání. Většina potřebné práce je už hotová, ale důležité komponenty prostředí (◊ [Plasma](#)<sup>(21)</sup> a [Konqueror](#)) na tom tak dobře nejsou. Zakroužkujte si tedy v kalendáři [nové plánované datum vydání](#)<sup>(22)</sup>: 11. leden 2008. KDE žádá všechny, kteří by mohli s prací pomoci, aby tak učinili – seznam současných zásadních nedostatků je na [stránkách Techbase](#)<sup>(23)</sup>. Připomeňme, že na 17. až 19. leden 2008 je [plánována oslava vydání](#)<sup>(24)</sup> v prostorách Google v Mountain View. Projev [Aarona Seiga](#)<sup>(25)</sup> bude odtamtud živě přenášen přes Internet.

## Úspěch OpenDocument Formatu roste

[Úspěch OpenDocument Formatu roste](#)<sup>(26)</sup>, stále více zemí z něj dělá vládní standard pro výměnu dokumentů. Po Jihoafrické republice je tu Nizozemí, které začne podporovat ODF od příštího dubna. Dále i Jižní Korea, kde Úřad pro technologii a standardy před několika měsíci schválil ODF jako národní standard, avšak toto rozhodnutí není pro ostatní vládní úřady závazné. Důležitou roli hraje také Francie, kde půl milionů státních úředníků přechází na [OpenOffice.org](#). Méně radikální kroky podniká Belgie – tam se instaluje ODF plugin pro MS Office. V Rusku projevují o ODF zájem dobrovolníci: specifikace ODF je dostupná pouze v anglickém jazyce, což některým lidem přišlo omezující, a tak [byl zhotoven neoficiální překlad](#)<sup>(27)</sup> specifikace do ruštiny. Na závěr drobná zpráva z oblasti boje s OOXML: Andy Updegrave [začal psát](#)<sup>(28)</sup> e-book ODF vs. OOXML: Válka slov.

## TrueBSD 2.0 RC1

TrueBSD je živý systém založený na FreeBSD – v čele vývoje stojí jediný programátor, dále mu několik lidí pomáhá s webovými stránkami a různými drobnostmi. Od [této verze](#)<sup>(29)</sup> se jedná o živé DVD namísto CD, což znamená zahrnutí velkého množství programů ([KDE](#), [GNOME](#), [Compiz](#), [Amarok](#), [OpenOffice.org](#) a další). Dále probíhá přechod na UTF-8 (v systémové konzoli je však i nadále používáno obyčejné kódování), systém je založen na FreeBSD řady 7, používá se modulární X.Org (který je automaticky nakonfigurován) a je podporován Flash.



## Elive 1.0.2

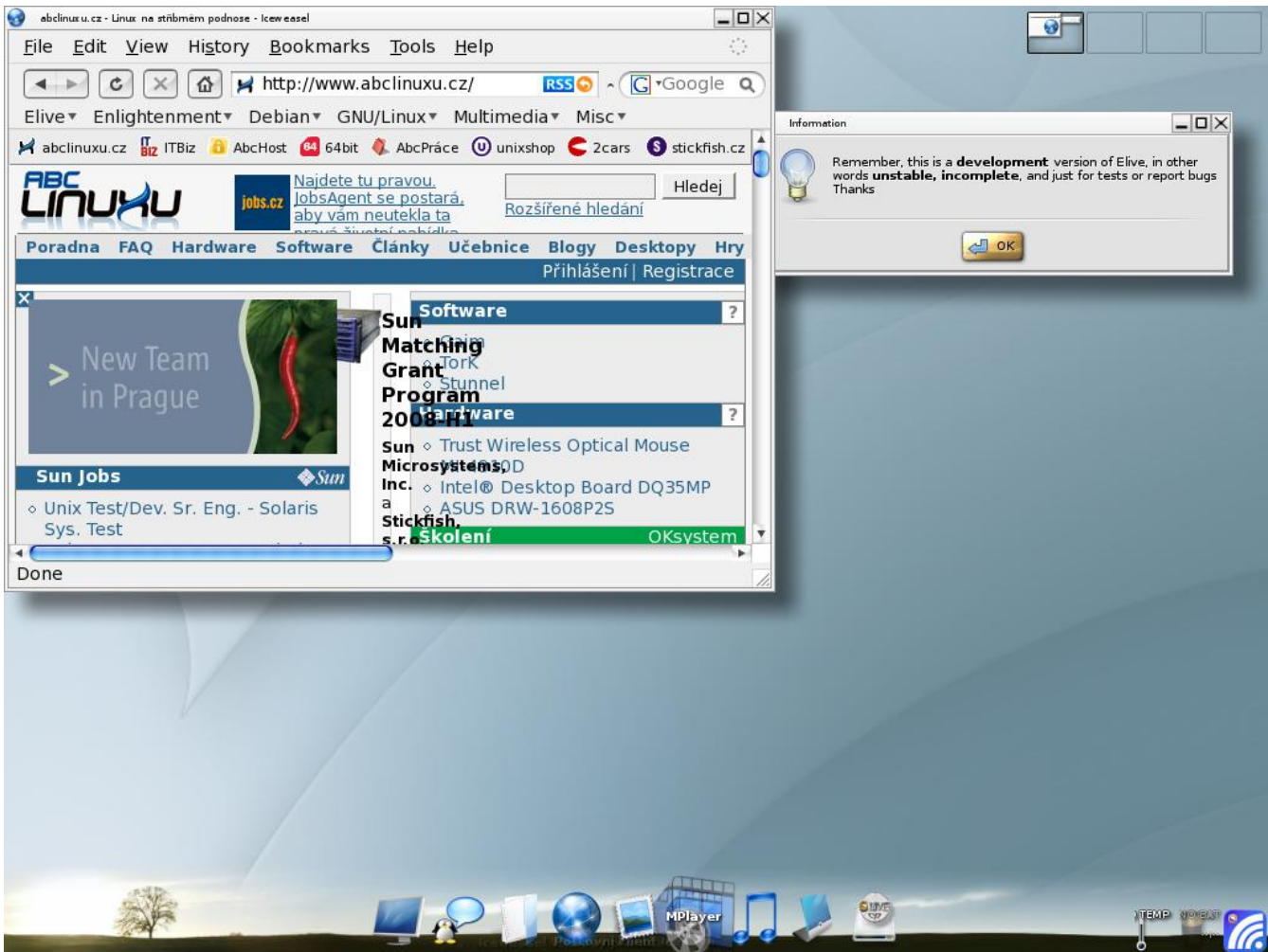
Elive je ukázkové živé CD s prostředím Enlightenment. Velkou novinkou verze 1.0.2<sup>(30)</sup> je zařazení Enlightenment 17. Zvukový systém zvládá více souběžných zvuků a uvnitř uživatelského jména a hesla je možné použít znak dolaru (aniž by následoval problém s přihlášením). Nyní je možné nainstalovat dodatečné správce oken a volit mezi nimi při přihlašování, přičemž si do systému můžete nainstalovat i jiné dodatečné programy. Živý systém je možné bootovat z externích zařízení, ale tato funkce ještě vyžaduje testování.

## Ubuntu 8.04 Alpha 1 Hardy Heron

Na světě je<sup>(31)</sup> první vývojová verze Ubuntu 8.04<sup>(32)</sup> – zatím se začínají rýsovat nové funkce a v některých oblastech dosud nepadla rozhodnutí. Hlavní novinkou v této vývojové verzi (Tribe 1) je pravděpodobně X.org 7.3, které mj. přináší lepší automatickou detekci nastavení v situaci bez konfiguračního souboru. Dále došlo k velkému začlenění balíčků z Debianu – v tomto vydání tak najdete mnoho aktualizovaných a úplně nových balíčků. Při testování si dejte pozor na známé problémy: vinou špatně nastavené priority Bluetooth pluginu nebude v Totemu fungovat přehrávání většiny souborů, dále se objevuje nemožnost přepínat virtuální terminály po přihlášení a také chyba, kvůli které se při instalaci nezapíše volba klávesnice.

Tým Kubuntu nelení<sup>(33)</sup> a slibuje, že začne v množství nabízených funkcí dohánět Ubuntu. Do Kubuntu se dostane KDE Bluetooth, Kicker s lepší podporou Compizu, KNetworkManager bude podporovat PPP a PPTP, Kaffeine automaticky nainstaluje FFmpeg kodeky a ve výchozím nastavení bude povolen zápis na NTFS. Kromě testovacích verzí Ubuntu a Kubuntu vyšla i testovací verze Edubuntu, JeOS, Xubuntu, Gobuntu a Ubuntu Studio.





Elive 1.0.2

## Distribuční rada: změna obnovování repozitářů v openSUSE

Správce balíčků YaST některé uživatele obtěžuje častým obnovováním dat repozitáře – naučíme se změnit časový interval, po kterém k obnovení může dojít, či jej úplně zakázat. Pokud chceme automatické obnovování zakázat/povolit, nastavíme to v YaST modulu Software Repositories nebo použijeme následující příkazy:

```
# zákaz:
zypper mr --disable-autorefresh "označení repozitáře"

#povolení:
zypper mr -a "označení repozitáře"
```

Pokud chceme změnit počet minut, po kterých má dojít k obnovení, upravíme soubor `/etc/zypp/zypp.conf` a nastavíme tam tento údaj:

```
repo.refresh.delay = 20 # příklad: dvacet minut
```

■

## Distribuční novinky – 49/2007

BusyBox žaluje Verizon kvůli porušování GPL. Sun bude oceňovat OSS vývojáře. Jak se staví Seagate k podpoře Linuxu. Poslední dění okolo OOXML. Trinity Rescue Kit 3.3 Beta. openSUSE 11.0 Alpha 0. Frugalware Linux 0.8 Pre 1. Geubuntu 7.10. Distribuční rada: obnovujeme práva souborů na RPM systémech.

Luboš Doležel

### Letem světem

SystemRescueCd 0.4.2<sup>(34)</sup> aktualizuje některé součásti a obnovuje podporu pro smbfs. Vyšlo FreeBSD 7.0 Beta 4<sup>(35)</sup> – konečnému vydání by měly předcházet ještě dva release candidate. Vine Linux<sup>(36)</sup> 4.2 RC1 je japonská komunitní distribuce pro desktopy a servery. NetBSD 4.0 RC5<sup>(37)</sup> opravuje několik odhalených chyb. Berry Linux 0.87<sup>(38)</sup> je živé CD založené na Fedoře 8. Big Linux 4.0 Beta 3<sup>(39)</sup> je brazilská distribuce založená na Ubuntu. Skolelinux 3.0r1<sup>(40)</sup> je aktualizace větve Debianu určené pro použití v oblasti vzdělávání. SimplyMEPIS 7.0 RC2<sup>(41)</sup> zahrnuje aktualizace z Debianu a různé opravy použitého jádra. Astaro Security Gateway 7.1<sup>(42)</sup> přidává nové funkce: nativní režim Active Directory SSO, inteligentní cachování, streamování audio/video a další funkce. EnGarde Secure Linux 3.0.18<sup>(43)</sup> přidává nový webový nástroj pro sledování systému (teplota, větráky, informace ze S.M.A.R.T.) a jiné novinky. VectorLinux 5.9 RC3<sup>(44)</sup> je téměř připravený pro vydání: nabízí nový vzhled, nástroje pro snadnou konfiguraci a zahrnuje také TuxOnIce (pro uspávání systému). dyne:bolic 2.5.2<sup>(45)</sup> opravuje detekci USB a jiné hardwarové problémy. Vyšel také Damn Small Linux 4.1<sup>(46)</sup> s několika inovacemi. CentOS 5.1<sup>(47)</sup> obsahuje novinky z Red Hat Enterprise Linuxu 5.1<sup>(48)</sup>.

### BusyBox žaluje Verizon kvůli porušování GPL

BusyBox<sup>(49)</sup> spustil třetí kolo žalob proti organizacím, které používají nebo šíří BusyBox v rozporu s licencí [GNU GPLv2](#)<sup>(50)</sup>. Software Freedom Law Center zastupující vývojáře BusyBoxu [tentokrát žaluje](#)<sup>(51)</sup> americkou telekomunikační společnost Verizon Communications, Inc. Ta zákazníkům služby FiOS dodává bezdrátové routery, ve kterých je přítomen BusyBox – těmto lidem však neposkytuje zdrojový kód, a dokonce tak nečiní ani poté, co je SFLC kontaktovalo. Cílem žaloby je opět náprava situace a nahrazení způsobených škod.

### Sun bude oceňovat OSS vývojáře

Sun ohlásil projekt<sup>(52)</sup> [Open Source Community Innovation Awards Program](#)<sup>(53)</sup>, prostřednictvím kterého budou finančně odměňováni open source vývojáři, kteří přispějí do vybraných projektů Sunu – konkrétně jde o GlassFish, NetBeans, OpenJDK<sup>(54)</sup>, OpenOffice.org, OpenSolaris a OpenSPARC. Celková výše odměn by za rok měla dosáhnout alespoň jednoho milionu dolarů. V polovině ledna bude projekt zahájen a budou oznámeny také podrobnosti – každý z uvedených open source projektů si stanoví vlastní pravidla a vlastní systém posuzování. Jonathan Schwartz, prezident Sunu, se nechal slyšet, že se nemůže dočkat výsledků. První výherci budou oznámeni v srpnu 2008.

### Jak se staví Seagate k podpoře Linuxu

Výrobce disků Seagate uvedl na trh řadu disků Free Agent, u kterých se projevuje defekt<sup>(55)</sup>: pokud se disk uvede do úsporného režimu, po probuzení se připojí v režimu USB 1. Pokud pomineme, že disk je

předformátován na NTFS, Seagate se k podpoře jiných systémů než Windows postavil velmi negativně. Seagate řeší na Windows problém skrze dodávaný software, ale namísto toho, aby uznal chybu, tvrdí, že není záměrem, aby disky byly kompatibilní s Linuxem (a Macem). Podařilo se najít [způsoby](#) <sup>(56)</sup>, jak problém obejít, ale přístup Seagate, který ani neuzná vadu svého výrobku, je poněkud zklamáním.

## Poslední dění okolo OOXML

Microsoft [vycouval](#) <sup>(57)</sup> z jednoho ze svých slibů, díky němuž na svou stranu získal některé státy, které hlasovaly pro schválení OOXML jako ISO standardu. Šlo o to, že by OOXML byl po schválení spravováno přímo organizací ISO – nový plán Microsoftu situaci mění a počítá s tím, že ISO maximálně vydá erratu a skutečnou údržbu a vývoj dalších verzí bude provádět Ecma (tedy Microsoft). ISO by tak ani nemělo příležitost zařadit, aby bylo OOXML kompatibilní s existujícími ISO standardy. Některé tato otočka nepřekvapila, jiní to označují za zradu. Jisté však je, že do 11. prosince musí být ISO nahlášen seznam zástupců z jednotlivých zemí, kteří se budou účastnit jednání, kde se řeší komentáře proti OOXML. Iniciativa <NO>OOXML připravila [seznam](#) <sup>(58)</sup>, kam se mohou zadávat informace o tom, kdo jednotlivé země zastupuje a zda se náhodou nejedná o lidi z firmy Microsoft.

## Trinity Rescue Kit 3.3 Beta

Trinity Rescue Kit <sup>(59)</sup> je bootovatelná distribuce Linuxu (založená na Mandrivě), která se zaměřuje na záchranné operace s Linuxem a Windows. Dokáže resetovat hesla, klonovat diskové oddíly a má vlastní nástroje pro záchranu smazaných souborů. Dále umí klonovat instalace Windows přes síť, hledat viry, zapisovat na NTFS, upravovat oddíly a ještě více. [Nová verze 3.3 Beta](#) <sup>(60)</sup> přináší zajímavou schopnost: nabootovat přes síť systém z již běžícího systému, a to bez zásahů do konfigurace DHCP serveru (v oznámení je popsán princip fungování). Dále byly opraveny chyby zahrnující zatuhnutí systému při mountallfs vinou NTFS ovladače v jádře, problém při bootování z USB a chyba v souvislosti s antivirem F-Prot.

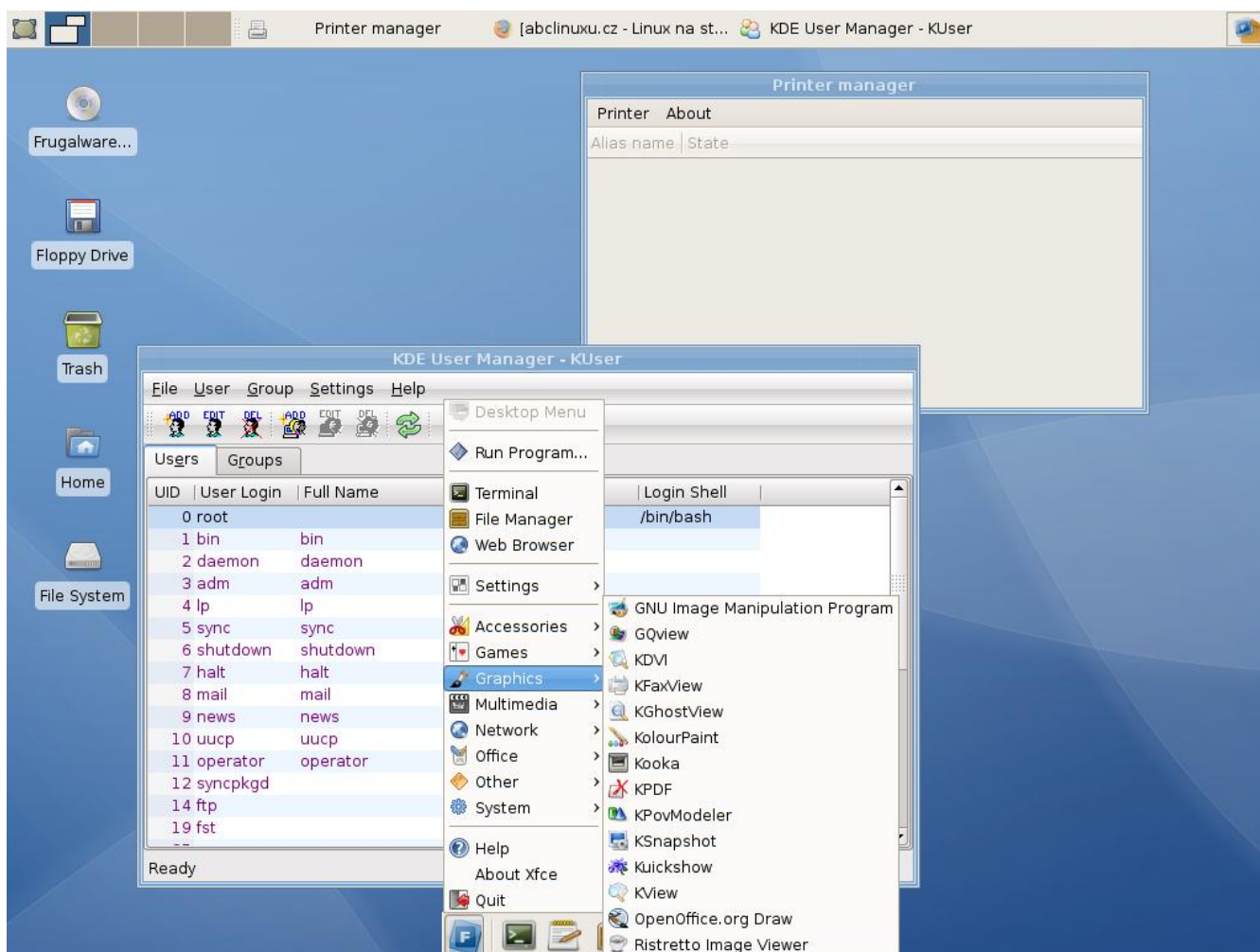


## openSUSE 11.0 Alpha 0

openSUSE přistoupilo k vydání jakési před-alfy s označením **Alpha 0** <sup>(61)</sup>, která obsahuje snapshot současného Factory. Od vydání openSUSE 10.3 bylo přidáno/aktualizováno 2187 balíčků, mezi nimiž je i doposud nevydaný software (například GCC 4.3.0) – vydání je tedy vhodné pro odvážné povahy. Systém běží na Release Candidate 4 jádra 2.6.24 a X.org 7.3. Jako prostředí nabízí KDE 3.5.8 nebo KDE 4.0 RC1, mezi ostatní vyzdihované novinky patří CUPS 1.3.4, Yast2 2.16 či GIMP 2.4. Při instalaci počítejte s tím, že se mohou vyskytnout menší potíže, a přečtěte si [seznam nejotravnějších známých chyb](#) <sup>(62)</sup>.

## Frugalware Linux 0.8 Pre 1

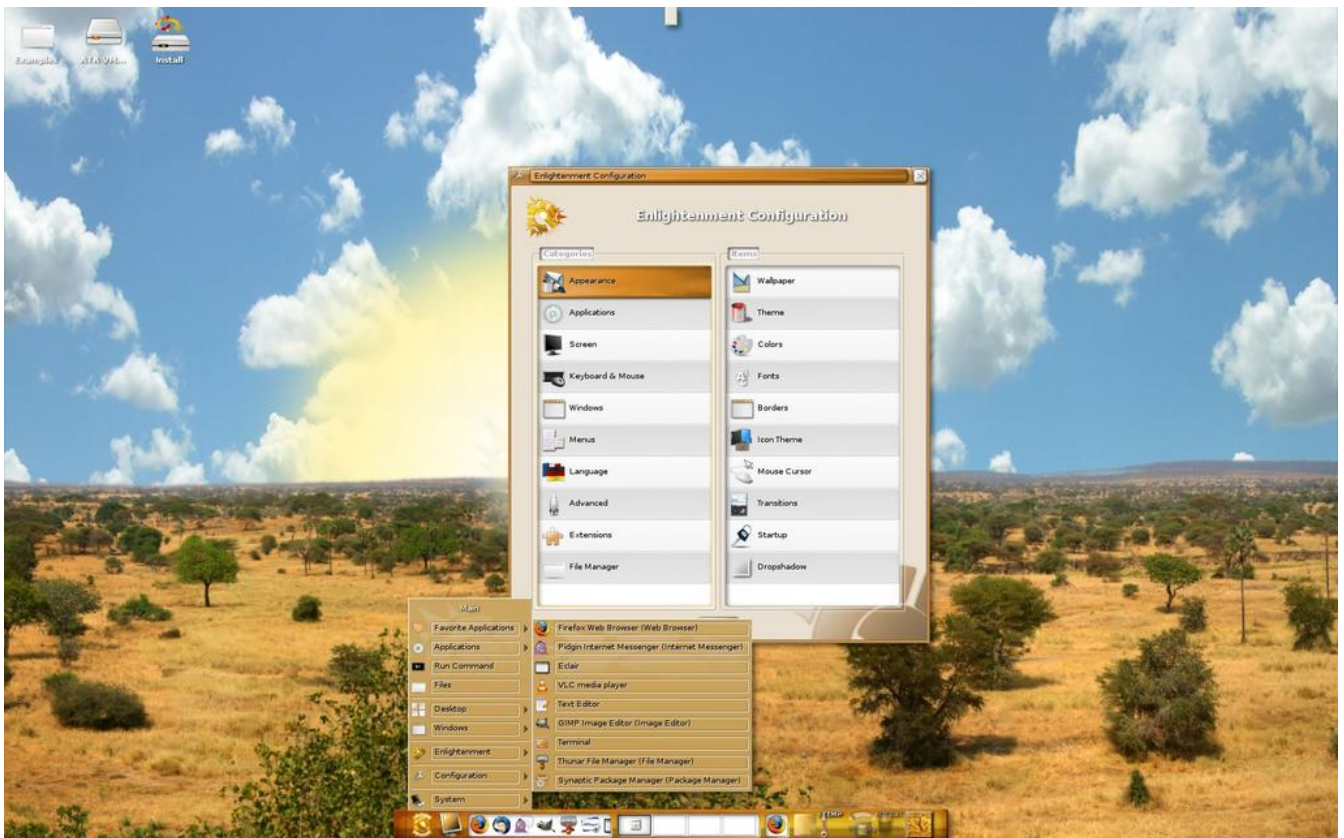
**Frugalware Linux** <sup>(63)</sup> je nezávislá (původem maďarská) distribuce navržená pro středně pokročilé uživatele. V jednoduchosti návrhu se snaží následovat Slackware a ke správě balíčků je používán Pacman z Arch Linuxu, i když balíčky Frugalwaru mají příponu fpm. Frugalware má dvě větve – current (aktualizováno denně) a stable (aktualizováno jednou za 6 měsíců) – a podporuje architektury i686 a x86\_64. Pracuje se i na portu pro PowerPC. **Frugalware Linux 0.8 Pre 1** <sup>(64)</sup> přechází na Linux 2.6.23 (s vlastními opravami), používá GCC 4.2.2, nahrazuje Beryl pomocí **Compiz Fusion** a kromě 600 aktualizovaných balíčků obsahuje i 100 balíčků úplně nových. Do systému přibyl Frugalware Update Notifier, který se nachází v liště a informuje uživatele o dostupných aktualizacích, a instalátor nyní podporuje WPA.



## Geubuntu 7.10

**Geubuntu** <sup>(65)</sup> je další z derivátů Ubuntu – tento nabízí prostředí **Enlightenment** obohacené o software z ostatních prostředí (především **GNOME**). Samotný Enlightenment totiž nepovažují za kompletní prostředí

– věří, že se to v budoucnosti bude zlepšovat a doplňků z GNOME bude ubývat. Do té doby hodlají pracovat na lepší integraci. Samotná distribuce vycházela ve dvou edicích<sup>(66)</sup> – Sunshine a Moonlight, které se lišily pouze výchozím tématem desktopu. Od verze 7.10 jsou tyto edice sjednoceny do jednoho ISO obrazu.



## Distribuční rada: obnovujeme práva souborů na RPM systémech

Stačí jeden nerozvážný příkaz a můžeme si poškodit oprávnění důležitých systémových souborů. V takové situaci může systém přestat fungovat nebo mohou vznikat bezpečnostní problémy. Na systémech, kde se používají RPM balíčky, není hned nezbytně nutné systém přeinštalovat – v takové situaci nám může pomoci jediný cyklus:

```
for p in $(rpm -qa); do
    rpm --setperms "$p"
    rpm --setugids "$p"
done
```

## Distribuční novinky – 50/2007

Knol: Wikipedie dle gusta Googlu. VCS Bazaar 1.0 je na světě. Nizozemí směrem k OSS a otevřeným standardům. eeeXubuntu – Ubuntu pro Eee PC. Parsix GNU/Linux 1.0r0 Test 1. CRUX 2.4 RC1. Mandriva Linux 2008 Spring Alpha 1 Barlia. Distribuční rada: zrychlujeme správce balíčků (openSUSE).

Luboš Doležel

### Letem světem

**Frenzy** <sup>(67)</sup> – toolkit pro síťové a systémové administrátory založený na FreeBSD 6.2 – vyšel ve **verzi Beta 3** <sup>(68)</sup>. Vývoj Damn Small Linuxu se přesunul do další řady, a tak je tu **DSL 4.2 RC1** <sup>(69)</sup>. GoblinX přišel s novou edicí své distribuce – ta se nazývá „Micro“ a zabírá jen necelých 100 MB, stahujte **verzi 2.6 RC1** <sup>(70)</sup>. Vývojáři živého CD Elive neodpočívají ani před Vánocemi a nabízejí vývojové vydání **Elive 1.2.3** <sup>(71)</sup>, které by mělo být stabilnější než předchozí verze. FreeNAS 0.686 Beta 3 je pravděpodobně poslední betou před označením systému za stabilní – přidává kontrolu souborového systému během startu, zařazuje podporu pro sdílení AFP plus jiné změny a vylepšení. Z Brazílie vzešel desktopový **Litrix Linux 7.12** <sup>(72)</sup> založený na Gentoo Linuxu. Španělé rovněž nezaháleli: **LliureX 7.11** <sup>(73)</sup> (živé DVD založené na Edubuntu) zlepšuje podporu hardwaru, rozšiřuje počet jazykových modulů (dostupná je i angličtina) a má robustnější automatickou konfiguraci.

### Knol: Wikipedie dle gusta Googlu

Google začal tento týden uzavřené skupině lidí **nabízet službu nazývanou Knol** <sup>(74)</sup>, přičemž „knol“ je překládáno jako jednotka vědění. Svým systémem připomíná Wikipedii, ale více zde zdůrazňuje podíl jedince na práci a jeho znalosti – vedle článku je sloupeček s krátkým resumé autora. Zatímco na Wikipedii může každý vytvořit článek na nové téma a za předpokladu, že jsou plněna všechna pravidla jej může kdokoliv rozšiřovat, Knol bude webovou encyklopedií, kterou píše lidé s ověřenými znalostmi v dané oblasti. Pokud chcete na článku něco změnit, můžete, ale vaše úpravy projdou kontrolou původního autora. Ke každé stránce je možné psát otázky, ale také ji můžete hodnotit či diskutovat nad jejím obsahem. Zatímco na Wikipedii platí, že by nemělo existovat více článků na stejné téma, zde to asi možné bude, ale hodnocení uživatelů se projeví na umístění článku mezi výsledky vyhledávání. Aby byli autoři motivováni ke sdílení svých znalostí, mohou si zvolit, že chtějí, aby jim Google dával část příjmů z reklam na dané stránce. Nabízí se otázka, do jaké míry bude Knol zneužitelný k podsouvání zkreslených informací veřejnosti a zda dokáže konkurovat tak zavedenému projektu, jakým Wikipedie jistě je.

### VCS Bazaar 1.0 je na světě

**Bazaar 1.0** <sup>(75)</sup> je první velká verze nového systému pro správu verzí – podobné programy se často používají při vývoji programů, kde probíhá spolupráce mezi více vývojáři, v situacích, kde je třeba mít možnost vracet se k předchozím verzím souborů, nebo tam, kde je žádoucí, aby veřejnost měla k aktuálním zdrojovým kódům snadný přístup. Na rozdíl od většiny obdobných systémů je Bazaar distribuovaný – dle Canonicalu, který za vývojem Bazaar stojí, je to výhoda v situaci, kdy se jedná o open source či outsourcovaný vývoj s programátory roztroušenými po celém světě. Tento model svým principem usnadňuje tvorbu nezávislých větví produktu. Bazaar je dle slov firmy snadno použitelný v konzoli i v **GUI** <sup>(76)</sup> a je také dále rozšiřitelný – sám si s sebou nese přes 20 pluginů v Pythonu (ve kterém je celý Bazaar napsán). K nasazení softwaru není třeba dedikovaný server, ale postačí webový server s FTP serverem. Licencován je pod **GNU GPL** <sup>(77)</sup> nebo proprietární licencí. Běží na systémech, kde je k dispozici Python 2.4 nebo vyšší.

## Nizozemí směrem k OSS a otevřeným standardům

Nizozemský parlament [schválil plán](#) <sup>(78)</sup> pro přechod na otevřené standardy a open source software. Přechod bude vypadat tak, že úřady budou muset v budoucnu odůvodňovat, proč chtějí na určitém místě i nadále používat proprietární operační systém či jiný proprietární software – pro část úřadů to začne platit už od příštího dubna a všechny úřady se budou muset podřídit od prosince 2008. Zároveň mají za povinnost samy přijít s plánem pro tento přechod. V červnu 2008 představí Nizozemský institut pro standardy návrh, který bude obsahovat výčet přípustných otevřených standardů (přijatelné jsou jen ty, které nejsou zatíženy poplatky za patenty). Už nyní však má svou pozici jistou [ODF](#) <sup>(79)</sup>. Dále byly připraveny plány pro fungování tendrů, nákupů a použití OSS na ministerstvech. Nizozemí svou náklonnost k open source ukázalo už před čtyřmi lety plánem s označením [OSOSS](#) <sup>(80)</sup> – ten dosáhl použití otevřených standardů na 60 % správních míst v zemi a použití open source v 47 % „situací“.

## eeeXubuntu – Ubuntu pro Eee PC

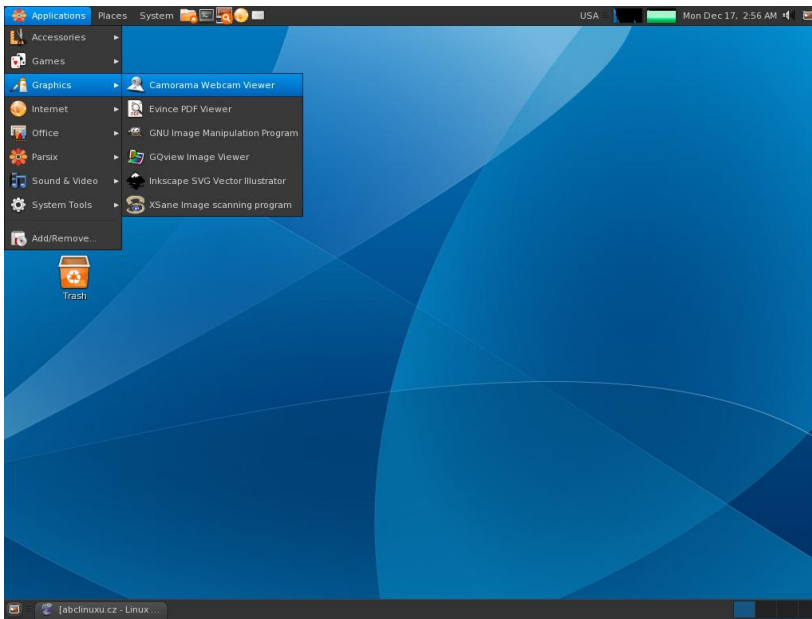
[eeeXubuntu](#) <sup>(81)</sup> je nový fork distribuce [Ubuntu](#) <sup>(82)</sup>. Není to však další z forků, které vznikly jen pouze kvůli jinému výchozímu prostředí (či jen jinému výchozímu stylu prostředí), ale varianta, která se zaměřuje na podporu levných notebooku [ASUS Eee PC](#) <sup>(83)</sup>. Jak už název napovídá, hlavním prostředím distribuce je [XFCE](#) – to je nakonfigurované pro běh při nízkém rozlišení. Distribuce obsahuje ovladač madwifi se speciálními patchi pro toto zařízení, dále upravenou verzi [ALSA](#) <sup>(84)</sup> pro funkčnost mikrofону a změněnou hodnotu swappiness, aby integrovaný solid state disk (SSD) nebyl pod takovou zátěží. eeeXubuntu vychází v podobě živého instalovatelného CD – hlavním nedostatkem zůstává instalátor Ubiquity, který se obtížně vejde na obrazovku výrobku.

## Parsix GNU/Linux 1.0r0 Test 1

[Parsix GNU/Linux](#) <sup>(85)</sup> je živá instalovatelná distribuce založená na KANOTIXu (tedy i na Debianu). Výhradně se zaměřuje na prostředí [GNOME](#), kde používá elegantní styl. Zajímavostí distribuce je podpora perské klávesnice a perských fontů a zahrnutí vícejazyčného slovníku [xFarDic](#) <sup>(86)</sup>. Na pomoc uživatelům je připraven IRC kanál, oficiální diskuzní fórum a také fórum pro francouzsky a italsky hovořící zájemce. [Poslední testovací verze](#) <sup>(87)</sup> distribuce nese kódové označení „Ramon“ – vylepšuje skripty, které zajišťují startování a vypínání živého systému, používá SquashFS s kompresí LZMA a běží na jádře 2.6.23.1 s [TuxOnIce](#) <sup>(88)</sup>, lwlwifi a jinými patchi. Ke konfiguraci grafického serveru je využíván grafický nástroj přeportovaný z Ubuntu. Hlavní programy systému jsou GNOME 2.20.2, [OpenOffice.org](#) 2.2.1, [IceWeasel](#) 2.0.0.8, [GIMP](#) 2.4.1, [Pidgin](#) 2.2.2 a přehrávač [Exaile!](#) 0.2.10. Ačkoliv se jedná o testovací verzi, neměly by se vyskytovat žádné zásadní potíže, nicméně testování veřejností je třeba. Stahovat můžete pro architektury i386 a amd64.

## CRUX 2.4 RC1

[CRUX](#) <sup>(89)</sup> je odlehčená distribuce pro i686 se zaměřením na zkušené uživatele Linuxu. Důraz je kladen na přímočarost – balíčky jsou uloženy v tar.gz souborech, init skripty jsou ve stylu BSD a výchozí nabídka aplikací je malá. Druhým cílem je používání nových funkcí Linuxu, nástrojů a knihoven. CRUX poskytuje také systém portů pro jednoduchou instalaci a aktualizaci aplikací. Repozitářů s porty je celá řada, můžete se podívat na [jejich přehled](#) <sup>(90)</sup>. Aktuální [předverze 2.4 RC1](#) <sup>(91)</sup> používá GCC 4.2.2, binutils 2.18 a glibc 2.6.1. Přechází na X.org 7.3, nyní už bez podpory Xprint. CRUX nově používá initramfs, díky čemuž už není třeba zadávat `root=...` – umístění mechaniky je detekováno automaticky. CD je možné odpojit za běhu systému a instalace může kompletně probíhat na dálku – stačí nabootovat CD, nastavit rootovské heslo a síť a spustit sshd.



Parsix GNU/Linux 1.0r0 Test 1

## Mandriva Linux 2008 Spring Alpha 1 Barlia

V připravované Mandrivě dojde k několika velkým změnám. [Poslední vývojová verze přechází](#) <sup>(92)</sup> na označování disků pomocí UUID, což mimo jiné umožní, aby stejný disk byl připojován vždy na stejné místo nezávisle na okolnostech (tedy např. na pořadí připojení). V repozitářích (ale ne v instalačních obrazech) najdete zatím poslední předverzi KDE 4.0 RC2 a v systému je naplno používán X.org 7.3 (včetně X.org serveru 1.4). V Mandrivě je také nasazen zvukový server [PulseAudio](#) <sup>(93)</sup> (ten se objevil např. v poslední Fedoře) – PulseAudio není jen další zbytečný software do systému navíc, ale nabízí i [zajímavé funkce](#) <sup>(94)</sup>. Pokud bude dodržen [vývojový plán](#) <sup>(95)</sup>, konečná verze distribuce bude na světě 2. dubna 2008.

## Distribuční rada: zrychlujeme správce balíčků (openSUSE)

Nejen zvědaví uživatelé provádějí mnoho instalací a odinstalací balíčků. Na rozdíl od jiných systémů nám na Linuxu tak akutně nehrozí „zašpinění“ systému, ale může dojít k fragmentaci. Na distribuci openSUSE máme dvě databáze týkající se správy balíčků, u kterých fragmentace hrozí. První z nich je databáze ZYPP – jedná se o sqlite databázi, a tak pročištění provedeme následovně:

```
sqlite3 /var/cache/zypp/zypp.db vacuum
```

Jinou možností je nechat cache kompletně zregenerovat:

```
rm /var/cache/zypp/zypp.db
zypper refresh
```

Druhou databází je databáze RPM, u které můžeme použít

```
rpm --rebuilddb
```

Neočekávejte velké zázraky – téměř každý souborový systém je sám o sobě náchylný k fragmentaci. V Linuxu je defragmentačních nástrojů méně, protože souborové systémy jsou lépe navrženy, ale při velkém zaplnění disku se práce zpomalí i tak. Pro nápravu můžeme překopírovat systém na jiný disk, smazat obsah původního souborového systému a zase nakopírovat data zpět nebo použít nástroje, které některé souborové systémy nabízejí (např. u XFS je to `xfs_fsr`).



## Distribuční novinky – 51/2007

Vývojáři Samby dostanou dokumentaci od Microsoftu. Norsko volí otevřené formáty. FSF Europe podporuje další prošetřování Microsoftu. SimplyMEPIS 7.0. VectorLinux 5.9. NetBSD 4.0. Distribuční rada: monitorování uživatelů v reálném čase.

Luboš Doležel

### Letem světem

Firma Xterasys – jedna z těch, které SFLC zažalovalo kvůli porušování licence u BusyBoxu – [přistoupila na mimosoudní vyrovnání](#) <sup>(96)</sup>. [Ubuntu 8.04 Alpha 2](#) <sup>(97)</sup> ukazuje aktuální stav vývoje distribuce, odvážlivci mohou pokračovat v testování. [CentOS 4.6](#) <sup>(98)</sup> odráží změny Red Hat Enterprise Linuxu 4.6. [Damn Small Linux 4.2](#) <sup>(99)</sup> nahrazuje některé aplikace, lépe zvládá konfiguraci grafických stylů a nabízí jiná vylepšení. [Musix GNU+Linux 1.0R3 Test 3](#) <sup>(100)</sup> řeší chyby s konzistencí souborového systému a aktualizuje (nebo rozšiřuje) softwarovou výbavu. [GoblinX 2.6 RC1 „Mini“](#) <sup>(101)</sup> především aktualizuje balíčky v systému. [ZenEdu Live Christmas Edition](#) <sup>(102)</sup> je vánoční snapshot Zenwalku s úpravami pro vzdělávání. [CRUX 2.4](#) <sup>(103)</sup> přechází na X.org 7.3 a používá initransfs. [sidux 2007-04.5](#) <sup>(104)</sup> představuje další verzi živého CD postaveného na Debianu. Vyšel [Linpus Linux 9.4 „Lite“](#) <sup>(105)</sup> vhodný pro levné počítače jako [ClassMate PC](#) <sup>(106)</sup> nebo Samsung UMPC. [Sabayon Linux 3.5 Beta 1](#) <sup>(107)</sup> má nové téma, logo, nového správce balíčků, X.org 7.3, podporu UnionFS, AUFS a ext4 a spoustu jiných věcí. [FreeNAS 0.686](#) <sup>(108)</sup> odstraňuje poslední nalezené nedostatky. [SAM Linux 2008 RC1](#) <sup>(109)</sup> přechází na GNOME Office, používá hlavní menu z Linux Mint a ještě více. Cizojazyčné novinky zastupují nizozemský [Nonux 4.3](#) <sup>(110)</sup> a japonský [Vine Linux 4.2](#) <sup>(111)</sup>.

### Vývojáři Samby dostanou dokumentaci od Microsoftu

Organizace [Protocol Freedom Information Foundation](#) <sup>(112)</sup> [podepsala](#) <sup>(113)</sup> se společností Microsoft smlouvu na získání dokumentace Windows potřebné pro interoperabilitu s jinými produkty. Tuto věc umožnilo rozhodnutí Evropské unie z roku 2004, ale také jednorázový poplatek 10 000 Euro. Poněkud nuceně vypadá [prohlášení Microsoftu](#) <sup>(114)</sup>, ve kterém stojí, že firma je „potěšena“ tímto obchodem. Dokumentace je nyní dostupná všem open source vývojářům, kteří podepíší obligátní [NDA](#) <sup>(115)</sup> – dá se očekávat, že radost přinese hlavně vývojářům Samby. Dokumentace mj. obsahuje výčet použitých patentů – Samba si je však licencovat nebude a bude se jim snažit vyhýbat. Jeden z vývojářů projektu (Jeremy Allison) řekl: *„Dávají nám veškerou dokumentaci potřebnou k tomu, aby vše fungovalo. Pro nedokonalost už nebudeme mít žádné omluvy... pokud něco nemáme, nebudeme moci říct, že to není naše chyba, že nevíme, jak to udělat.“* Protože vývoj doposud probíhal zdlouhavým zpětným inženýrstvím síťové komunikace, můžeme očekávat, že práce na podpoře protokolu Server Message Block (SMB) 2.0 z Windows Vista bude znatelně rychlejší. Důležité je, že Microsoft je povinen udržovat dokumentaci aktuální a bez chyb, takže vývoj je zajištěn i do budoucna.

### Norsko volí otevřené formáty

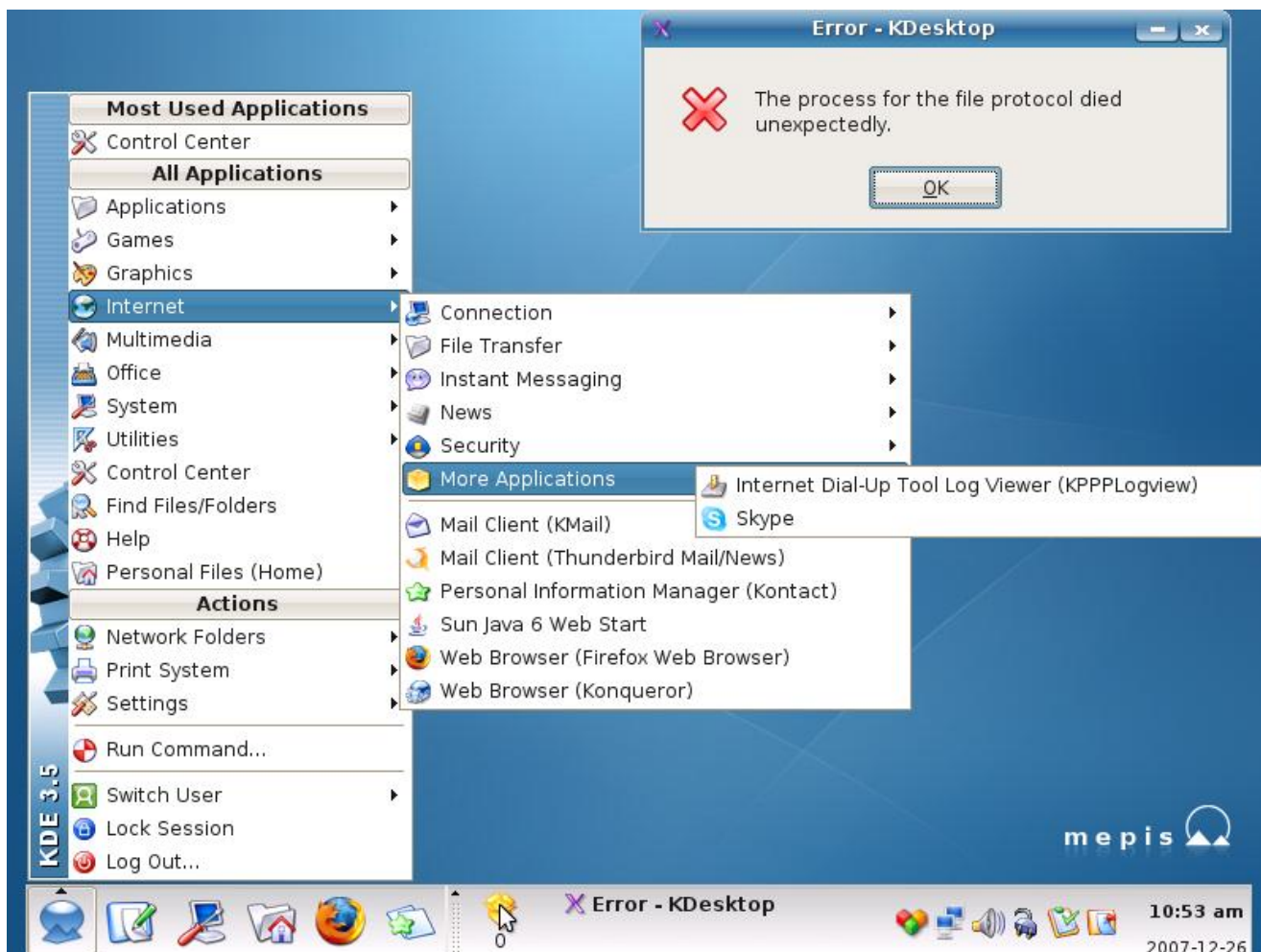
Otevřené formáty se dostávají na výsluní v neustále větším množství zemí světa. Poslední zemí [se stalo Norsko](#) <sup>(116)</sup>, kde byly pro použití vládními úřady zvoleny tři formáty: HTML pro informace na webu, PDF pro dokumenty, kde je nutné zachovat rozvržení, a [ODF](#) <sup>(117)</sup> pro dokumenty, které je třeba dále upravovat. Úřady mohou publikovat informace i v jiných formátech, ale vždy také v jednom z této trojice. Zákon vstoupí v platnost 1. ledna 2009 a v roce 2014 musejí být všechny starší dokumenty převedeny do zmiňovaných otevřených formátů. Podaří se podobné věci dosáhnout i v České republice?

## FSF Europe podporuje další prošetřování Microsoftu

Opera Software se rozhodla podat k Evropské komisi stížnost na integraci Internet Exploreru do Windows, přičemž firmě se nelíbí ani svérázný výklad platných webových standardů v tomto prohlížeči. Ačkoliv má celá věc se svobodným softwarem společného jen málo, FSF Europe se rozhodlo podpořit postup této firmy. V [prohlášení](#) <sup>(118)</sup> se píše, že Microsoft by měl plně implementovat otevřené standardy tak, jak činí svobodné prohlížeče jako [Mozilla Firefox](#) nebo [Konqueror](#). FSFE tento postoj zaujalo kvůli tomu, že právě tyto „poctivé“ prohlížeče mají špatné výchozí postavení na trhu a jsou nuceny učit se simulovat nespécifikované a chybné chování Internet Exploreru. Protože se nejedná o nijak snadný úkol, tyto prohlížeče pak v očích věci neznalých uživatelů vypadají ve srovnání s IE jako podřadné.

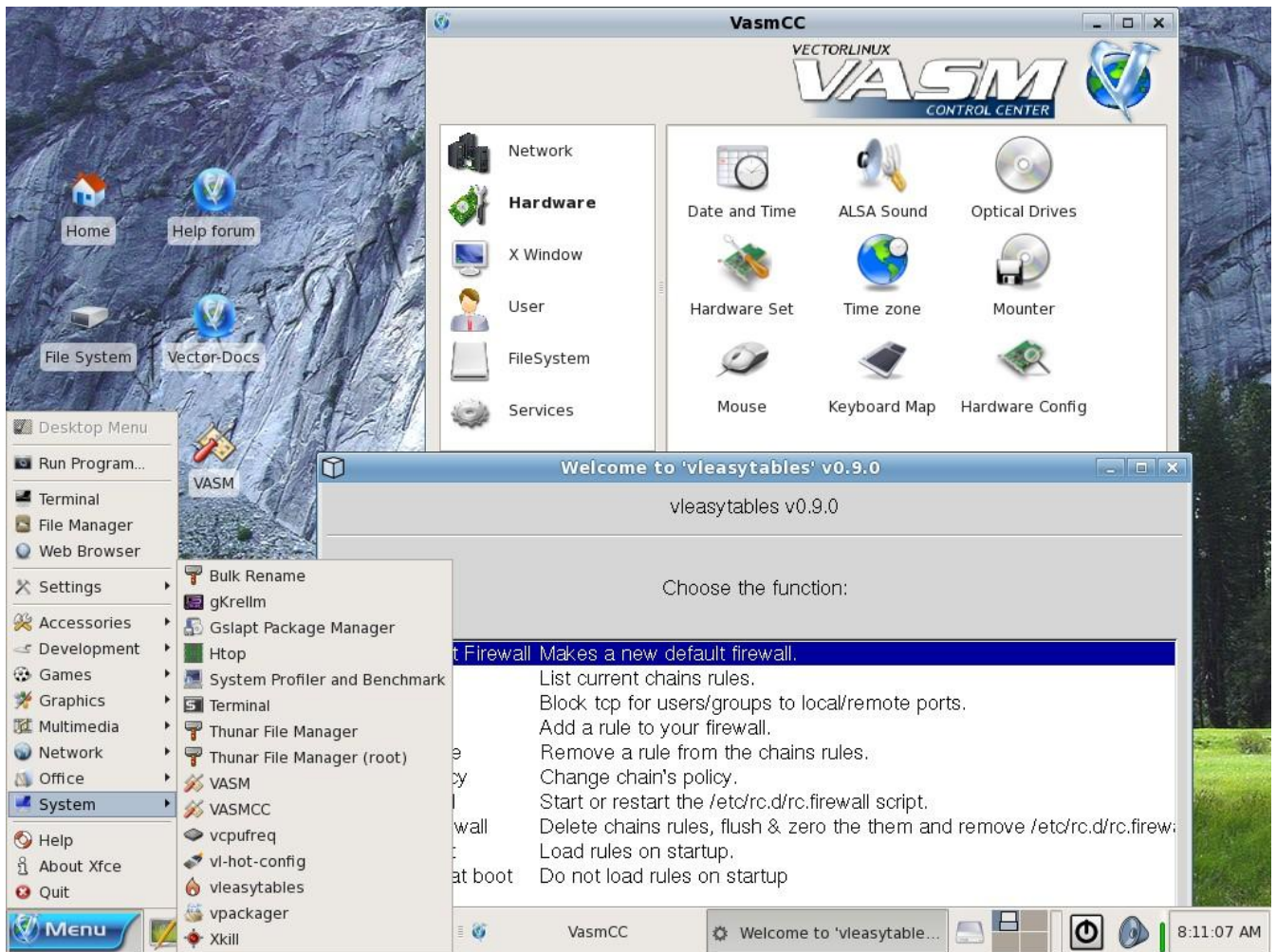
## SimplyMEPIS 7.0

[SimplyMEPIS](#) <sup>(119)</sup> je uživatelsky přívětivá distribuce, jež stojí na balíčcích Debianu a Ubuntu. Krátce před Vánoci vyšla [verze 7.0](#) <sup>(120)</sup> v 32-bitové a 64-bitové variantě, která používá nové desktopové téma a detailní uživatelský manuál – obojí je dílem komunity. Jednou za měsíc vyjde ISO obraz s aktualizacemi a někdy v dubnu nebo květnu bude zahájena práce na synchronizaci s Debian Lenny. Podívejme se však na novinky aktuálního vydání: SimplyMEPIS 7.0 nabízí jádro 2.6.22.14 s bezpečnostními opravami, X.org 7.1, prostředí [KDE 3.5.8](#), [OpenOffice.org 2.3.0](#) a [Mozilla Firefox 2.0.0.11](#) spolu s [Mozilla Thunderbirdem 2.0.0.6](#). Systém zvládá zápis na NTFS, obsahuje ovladače madwifi-ng a také wpa-supPLICANT. Proprietární grafické ovladače zastupují NVIDIA 100.14.19, 1.0.9639 a 1.0.7185 a AMD fgLrx 8.43.2. Mezi ostatní výbavu pak spadá [Amarok 1.4.7](#), Sun Java 6 nebo [digiKam 0.9.2](#).



## VectorLinux 5.9

VectorLinux se zaměřuje na stabilitu, jednoduchost a výkon systému – vydání 5.9 final GOLD edition <sup>(121)</sup> nabízí celý seznam funkčních pluginů pro prohlížeče – Flash, Java, Real Media, PDF, Quick Time a další. Jako prostředí používá Xfce 4.4.2, Fluxbox a Jwm, nabízí prohlížeče SeaMonkey 1.1.7, Mozilla Firefox 2.0.0.11 a Opera 9.5.0 beta1, přičemž další programy a prostředí jsou vždy k dispozici v repozitářích. Konfiguraci systému lze provádět pomocí nástrojů Vasm Control Center, Vwifi či Gslapt. Pokud vám schází nějaký balíček, je k dispozici upravená verze nástroje cruxports.



## NetBSD 4.0

Přichází NetBSD 4.0 <sup>(122)</sup> s celou řadou důležitých změn. V oblasti síťování přibyla podpora IPv6 v API pro tunely gre(4) a zařízení tun(4). Dále přibyl NDIS-wrapper pro používání ovladačů z Windows a byly zahrnuty nástroje wpa\_supplicant a wpa\_cli. NetBSD nově podporuje UDF na blokových zařízeních a optických médiích (zatím jen ke čtení) a byl přidán souborový systém tmpfs, jehož cílem je nahradit mfs, neboť tmpfs je výkonnější a paměťově efektivnější. Subsystem zvukových karet je bohatší o ovladače pro Intel ICH8, nForce 3/4 a modemy AC'97, z OpenBSD byla přejata podpora Intelligent Platform Management Interface (IPMI) 1.5 a 2.0 a také výčet zahrnutých ovladačů síťových karet je delší o přírůstky v podobě nForce, Marvell Yukon 2 a ovladačů bezdrátových karet jako Intel PRO/Wireless 3945ABG nebo Ralink RT2500.

Správa výkonu si nyní rozumí s procesory AMD K7 a K8 (technologie AMD PowerNow a Cool'n'Quiet) a v bohatém seznamu nových ovladačů pro úložiště nalezneme mj. ahcisata pro řadiče kompatibilní s AHCI 1.0/1.1. Jádro nyní odpovídá specifikaci Multiboot, což znamená zdatně vyšší podporu NetBSD ze strany

GRUBu. Z FreeBSD byl importován FireWire stack, dále byla vyvinuta podpora Bluetooth a dále se pracuje na 32bitové emulaci Linuxu na platformě amd64. Ostatních změn je opravdu nepřeborné množství, proto čtěte [oznámení o vydání](#) <sup>(123)</sup>.

## Distribuční rada: monitorování uživatelů v reálném čase

`whowatch` <sup>(124)</sup> je konzolový nástroj, se kterým můžete snadno a v reálném čase monitorovat aktivitu uživatelů v systému. Můžete si prohlížet, odkud (a jak) je uživatel připojen nebo jaké má spuštěné procesy – s tím souvisí i možnost procesy ukončovat. Hned po spuštění programu se nám ukáže výpis přihlášených uživatelů a při stisku enteru se ukáže strom procesů vybraného uživatele.

```
4 users: (0 local, 0 telnet, 4 ssh, 0 other)          load: 0.16, 0.15, 0.06
(sshd)      root    pts/0  89.31.42.18    mc
(sshd)      root    pts/6  89.31.42.18    mc
(sshd)      root    pts/7  89.31.42.18    -bash
(sshd)      root    pts/9  89.31.42.18    whowatch
```

```
ENTER:proc_tree t:init_tree i:idle/cmd c:cmd s:sysinfo q:quit
```

```
4 users: (0 local, 0 telnet, 4 ssh, 0 other)          load: 0.03, 0.11, 0.05
(sshd)      root    pts/6  89.31.42.18
22455      -  -bash
23892      -  - mc
23894      -  - bash -rcfile .bashrc
23898      -  - nano
```

```
ENTER:users c:cmd t:init_tree d:details o:owner s:sysinfo ^I:INT ^K:KILL l:siglist
```

K seznamu všech procesů v systému v programu není daleko – stačí stisknout `I` a program přepne výpis. Ke každému procesu lze zobrazit tabulku s detaily (využití paměti, vlastník, adresář apod.) klávesou `D`. Pokud se vám proces znelíbí, použijte `Ctrl-I` pro zaslání signálu SIGINT nebo `Ctrl-K` pro signál SIGKILL. Kromě podrobností o uživateli umí `whowatch` ukázat i stručné informace o systému.

```
START: Fri Oct 26 15:00:06 2007
EXE: /usr/sbin/apache2
ROOT: /
CWD: /

STATUS:
Uid:    0      0      0      0
Gid:    0      0      0      0
FDSize: 32
Groups:
VmPeak: 21568 kB
VmSize: 21568 kB
VmLck:  0 kB
```

<- -> [a]up, [z]down

```
BOOT TIME: Fri Oct 26 14:59:31 2007
CPU: 0.0% user 0.0% sys 0.0% nice 100.0% idle
MEMORY:
MemTotal:      255908 kB
MemFree:       115300 kB
Buffers:       41380 kB
Cached:        57360 kB
SwapCached:    0 kB
Active:        76964 kB
Inactive:      51260 kB
HighTotal:     0 kB
HighFree:      0 kB
LowTotal:      255908 kB
```

<- -> [a]up, [z]down

■

## Svatá trojice vs. balíčkovací systém

V diskuzi se občas objevují dotazy na to, jak si zkompilovat nějaký program. Většinou takové otázky pokládají začátečníci, kteří se o sestavení nějakého programu pokusili, protože jim určitý software chyběl. A první, co našli na stránkách hledaného programu, byla právě „svatá trojice“. Přitom chtěného cíle mohli dosáhnout i mnohem méně trnitou cestou.

Petr Štrajt

Pokud zjistíte, že vám chybí nějaký program, který potřebujete ke své práci, *zkuste jej nejprve vyhledat pomocí balíčkovacího systému své distribuce*, v naprosté většině případů potřebný software naleznete a nainstalujete během několika minut pomocí jednoho či dvou příkazů.

Může se stát, že na první pokus neuspějete (potřebný [balíček](#)<sup>(125)</sup> není v databázi balíčkovacího systému nalezen) – nejčastěji k tomu dochází u různých kodeků sloužících k přehrávání multimediálního obsahu.

Tyto programy bývají šířeny pod nějakou omezující licenci, a proto je nebylo možno přidat do standardních [repozitářů](#)<sup>(126)</sup>. Podrobné informace naleznete v dokumentaci vaší distribuce.

### Kdy se tedy kompilace programu vyplatí?

Běžně jsou programy dostupné v balíčkovacích systémech většiny distribucí kompilovány s nějakým průměrným nastavením, které nevyužívá všechny možnosti programu. Pokud máte speciální požadavek, je pro vás sestavení ze zdrojového kódu řešením. Pak skutečně může přijít ke slovu „svatá trojice“, což je ve skutečnosti posloupnost tří příkazů, s jejichž pomocí lze většinu zdrojových kódů, které využívají [GNU Autotools](#)<sup>(127)</sup>, připravit ke kompilaci, zkompilovat a nainstalovat:

```
./configure
make
make install
```

Ti z vás, kteří mají chuť experimentovat, mohou zkusit své štěstí třeba při kompilaci webového prohlížeče [Links](#). Je to relativně malý program a jeho sestavení by na průměrném stroji nemělo trvat dlouho.

Tenhle šikovný program, který je v základní konfiguraci textový, ale dokáže pracovat i v grafickém rozhraní, které naleznete ve většině distribucí. Chcete-li se dozvědět více, navštivte domácí stránku prohlížeče na adrese [links.twibright.com](http://links.twibright.com)<sup>(128)</sup>. [Informace potřebné ke kompilaci programu](#)<sup>(129)</sup>.

### Když je kompilace tak mocný nástroj, proč ji tedy nepoužívat?

- Instalace balíčku trvá cca. minutu. Kompilace většího programu může trvat několik hodin i den. Podle toho, na jak výkonné sestavě kompilujete.
- Instalujete-li program pomocí balíčkovacího systému, tak vám stačí znát název programu, příkaz pro jeho instalaci, aktualizaci a případně odstranění.
- Naproti tomu, kompilace programu není zrovna triviální, vyžaduje poměrně podrobnou znalost daného systému a často můžete narazit na různé problémy.

Nejčastěji první pokusy o kompilaci programu končí hláškou skriptu [./configure](#)<sup>(130)</sup>, že chybí nějaká knihovna nebo její vývojová verze.

## Co jsou to knihovny a k čemu vlastně jsou?

Knihovna je část programu, která obsahuje funkce potřebné pro výkon nějaké činnosti, například vypsání textu na obrazovku, jak si ukážeme dál.

```
#include <stdio.h>
main() { printf("Ahoj svete\n"); }
```

Účelem programu je vypsání na obrazovku nápis `Ahoj svete`. Samotné vypsání požadovaného textu na obrazovku zajišťuje funkce `printf`.

Výhodou funkce je to, že programátor se zabývá pouze tím, co udělat, například vypsání textu na obrazovku, a neřeší, jak to udělat, to má na starosti funkce.

## Jak se liší vývojářská verze knihovny od běžné

Vývojářské verze knihoven obsahují hlavičkové soubory (obyčejně s příponou `.h`). Pokud máte nainstalovány tyto knihovny, tak můžete na svém systému kompilovat programy, které ke svému běhu využívají tyto knihovny.

Vývojářské verze knihoven nebývají součástí běžné instalace a právě jejich absence bývá nejčastější příčinou problémů s kompilací.

Další nevýhodou ručně kompilovaných programů je, že si sami musíte ohlídat, zda ve vašem programu nebyla odhalena nějaká bezpečnostní chyba a případně se postarat o její odstranění.

Častým argumentem pro ruční kompilaci nějakého programu bývá, že chcete mít novější verzi nějakého softwaru, než je ta, která je obsažena ve vaší distribuci. Tuto situaci lze řešit vhodným nastavením zdrojů softwaru. Podrobnější informace o tom, jak to řešit v Debianu, naleznete [v manuálu](#) <sup>(131)</sup>.

## Závěr

O správě programů v Linuxu by toho šlo napsat ještě hodně, ale to už přenechám povolanejším. Pokud se mezi těmi, kdo tento zápis četli, našel alespoň jeden člověk, který se vyhne chybám, které jsem nedávno také udělal, pak splnil svůj účel.

■

# Lokální útoky na operační systém linuxového typu

Článek se zabývá otázkou lokální bezpečnosti, tedy takovou, která je realizována lokálně, nikoliv prostřednictvím sítě. Operační systém Linux byl vybrán z důvodu jeho rostoucí obliby, ale především jeho celkové vyšší transparentnosti a otevřenosti.

Tomáš Pelka

## Úvod

Bezpečnost je velmi komplexní záležitost. Zjednodušeně by se dala představit jako skládanka, u níž je bezpodmínečně nutné zajistit správné složení do výsledného celku, ve kterém nebudou „díry“.

Linux má už ve své podstatě v tomto ohledu dvě protichůdné vlastnosti. Ta kladná vychází z toho, že do něj přispívá velké množství lidí, a tudíž se velmi rychle rozvíjí. Se zmíněnou vlastností koresponduje velmi rychlá oprava chyb. Velké množství přispívatelů ale vytváří stinnou stranu problematiky. Chyby mohou být opraveny, ale také nevědomky do operačního systému zaneseny.

Vyvstává potřeba organizace, která by celý proces kontrolovala. Dojde-li v operačním systému k implementační chybě, je velmi složité ji později napravit. I když původní Unix nebyl přímo navržen s ohledem na bezpečnost, lze dnes unixové systémy, včetně Linuxu, zabezpečit poměrně dobře. Díky otevřenosti zdrojových kódů jádra a programů lze poměrně snadno prozkoumávat výhody a slabiny programů na té nejnižší úrovni a díky licenci [GNU GPL](#) <sup>(132)</sup> či jakékoliv jiné OSS licenci tyto slabiny opravit nebo možnost jejich výskytu omezit. Otevřenost kódu zaručuje, že před námi nikdo nemůže nic skrývat. Navíc otevřenost nám umožňuje Linux vylepšovat a upravovat [1].

Vše však má svoji odvrácenou stranu. Nevýhoda filozofie tkví v tom, že stejně jako uživatelé mají přístup k otevřeným zdrojovým kódům také potenciální útočníci, kteří mohou tyto kódy zneužít ve svůj prospěch.

Používání bezpečných programů je jen část úkolu. Neméně důležité je služby také správně nakonfigurovat.

Pokusme se hackera identifikovat a lépe popsat. Hacker je člověk či raději programátor, který vnímá programování jako jistou formu uměleckého vyjádření a počítač jako umělecký nástroj. Hackeři jsou poháněni snahou po sebezdokonalení a řešení neobvyklých problémů novými cestami, touhou rozebírat a chápat. Tyto hodnoty poháněné vědomostmi se nazývají hackerskou etikou: uznání logiky jako formy umění, podpora svobodného toku informací překonávající všechny hranice a omezení s prostým cílem dosáhnout lepšího chápání okolního světa. Tato etika vychází ze stejné filozofie jako Pythagorejci ve starém Řecku, kteří měli podobnou etiku a subkulturu navzdory neexistenci počítačů [2].

Jak rozpoznat dobrého hackera, který přináší nový, kreativní pohled na technické vymoženosti, od toho špatného, zloděje čísel kreditních karet? Pro špatného hackera byl vymyšlen termín cracker, který popisuje špatného hackera a odlišuje jej od toho dobrého. Bohužel tento výraz není obecně často užíván. A tedy pojmy hacker a cracker v uších obecné veřejnosti splývají [2].

## Exploitování programu

Exploitování je základem hackingu. Program není nic jiného než komplexní sada pravidel sledující určitý tok činností, která počítači říkají, co má dělat. Exploitování programu znamená přinutit programy vykonávat to, co chce útočník, dokonce i když je program navržen tak, aby tomu zabránil. Program může dělat jen to, k čemu je navržen. Bohužel to, co je napsáno se nemusí shodovat s tím, co programátor zamýšlel. Bezpečnostní díry jsou ve skutečnosti vady nebo přehlédnutí v návrhu programu nebo v prostředí, ve kterém program pracuje [1].



Často opakovaná programátorská chyba se nazývá *off-by-one*. Název napovídá, že jde o chybu, při níž se programátor splete o jedničku. Následující příklad vše vysvětlí. Představme si, že máme postavit plot o délce 20 m. Po dvou metrech bude sloupek. Kolik sloupků budeme potřebovat? Samozřejmě, bez dlouhého přemýšlení je odpověď 10. Ale to je špatně. Ve skutečnosti je jich potřeba 11. Někdy se tato chyba nazývá také *fencepost error*. Takovéto chyby často bývají nepostřehnuty, protože programy nejsou standardně testovány na všechny možné vstupní varianty. Jejich efekt se mnohdy neprojeví, ale když už se projeví, může dojít k lavinovému efektu a krom logiky programu může být narušena i bezpečnostní stránka [1].

Jeden nedávný omyl (cca před třemi lety) se přihodil i v OpenSSH. Byla tam nalezena chyba *off-by-one* v kódu pro alokaci kanálu. Tato chyba byla po objevení hodně exploitována. Kód obsahoval tuto podmínku `if`:

```
if (id < 0 || id > chanelles_alloc)
```

a správně měl kód vypadat takto:

```
if (id < 0 || id >= chanelles_alloc)
```

Toto je jen jeden příklad z mnoha. Existují ovšem chyby, jejichž exploitování není tak jednoduché. Mezi takové netriviální, obecné exploitovací techniky patří chyby přetečení paměti (*buffer-overflow*) a chyby ve formátovacím řetězci. S těmito chybami je nakonec možné zcela převzít vládu nad průběhem vykonávání programu. Nejběžnější technikou je propašování části škodlivého kódu a jeho spuštění. Takovým technikám se říká *execution of arbitrary code* (spouštění libovolného kódu). V následujících kapitolách bude tato chyba často skloňována.

## Lokální útoky – přetečení paměti

V nadcházející části se budeme věnovat především chybně napsanému kódu. Chybně napsaný kód je velmi nepříjemná záležitost. Způsobuje časté havárie, ztráty dat a zbytečné úniky výpočetního výkonu. Jakkoliv může být takový kód při normální práci nepříjemný, zvykli jsme si na něj a počítáme s ním. Často zálohujeme, stahujeme aktualizace a bezpečnostní záplaty. Skutečný problém ale nastává, když je chyba v programu potenciálně zneužitelná útočníkem. Chybný kód představuje vstupní bránu pro řadu útoků, ať už lokálních nebo vzdálených. Existuje několik tříd programátorských chyb, které mohou způsobit vážné trhliny v bezpečnosti služeb nebo celého systému.

Největší problém nastává u programů, které mají nastaven `suid` respektive `sgit` bit. Takové programy by měly být programovány co nejpečlivěji, protože program běží s jinými právy, než s jakými byl původně spuštěn. Uživatel využívající takový program může kvůli chybě získat práva, která by vůbec získat neměl. U tohoto druhu programů je třeba dbát na to, aby byla práva odkládána ve správném čase a hlavně korektně. Nejúčinnější ochranou je dobrá a rychlá informovanost. Jen tak je možné provést odpovídající kroky vedoucí k nápravě.

### Paměť

Dříve, než budou vysvětleny konkrétní programátorské chyby, které jsou využitelné k exploitování, je třeba říci pár slov o paměti, její segmentaci a deklaraci.

Paměť je dočasné úložiště, nic víc než bajty označené tzv. adresou. Bajty na jednotlivých adresách lze číst nebo zapisovat.

Procesor má vlastní paměť, která je relativně malá. Tyto části paměti se nazývají registry. Existují speciální registry, které udržují informace o vlastním spouštěném programu. Jeden z nejdůležitějších je EIP (*extended instruction pointer*). Jak název napovídá, jedná se o ukazatel ukazující na právě prováděnou instrukci.

Dalšími důležitými registry pro vykonávání programu jsou EBP (extended base pointer) a ESP (extended stack pointer) [2]. Tyto informace jsou platné pouze pro procesory Pentium a kompatibilní.

Důležitou věcí v souvislosti s deklarací paměti je pořadí bajtů v čtyřbajtovém slovu na architektuře x86. Toto řazení je známé jako little endian, které říká, že nejméně významný bajt je první. Znamená to, že bajty slova, jsou uloženy v převráceném pořadí. Hexadecimální hodnota `0x12345678` v kódování little endian je v paměti uložena jako `0x78 0x56 0x34 0x12`.

Občas se stane, že pole alokovaných bajtů není využito úplně. Za alokovanými bajty budou nevyužité znaky. V tom případě se použije nula (bitová nula nebo také znak `null`) pro ukončení řetězce. Ukončování řetězců nulovou hodnotou zvyšuje efektivitu a umožňuje funkcím s řetězcem lépe pracovat.

Paměť programu je rozdělena na pět segmentů: `text`, `data`, `bss` (block started by symbol, segment neinicializovaných dat), `heap` (halda) a `stack` (zásobník). Každý segment reprezentuje speciální část paměti, která je vymezena pro určité účely [3].

Segment `text` je někdy označován jako code. Jde o místo, kde se nacházejí instrukce strojového jazyka. Tato vlastnost však platí jen pro procesory vycházející z Von Neumannovy architektury, kde není oddělena paměť na datovou a instrukční. Naproti tomu procesory s Harvardskou architekturou mají zmiňované paměti odděleny, tudíž tento bezpečnostní problém odpadá. Navíc novější procesory obsahují ve své instrukční sadě příznak, který tyto paměti odlišuje.

`Data` a `bss` se využívají k ukládání globálních a statických proměnných.

Segment `heap` (halda) se používá pro ostatní programové proměnné. Celá tato paměť je řízena alokačními a de-alkačními algoritmy, které rezervují paměťové místo pro další použití.

Segment `stack` (zásobník) má také proměnnou velikost a používá se jako dočasné úložiště pro kontext volání funkcí. Segmenty `heap` i `stack` jsou dynamické a rostou proti sobě.

## Přetečení paměti

Programovací jazyk C nepatří mezi jazyky silně typované. Odpovědnost za datovou integritu je ponechána na programátorovi. Kdybychom tuto odpovědnost nechali na překladači, výsledný kód by byl neúměrně velký. Optimalizace a efektivita výsledného kódu je cenou za zvýšenou pečlivost programátora. Kontrola programátora také zvyšuje náchylnost na přetečení paměti. Jakmile je proměnná v alokované paměti, neexistuje žádný mechanismus, který pohlídá, že se proměnná opravdu do alokované paměti vejde. Popsané skutečnosti jsou hlavním rozdílem oproti jazykům opravdu silně typovaným jako například C++ či Java. Máme-li například následující kus kódu:

```
void overflow_function(char *str){
    char buffer[20];
    strcpy(buffer, str);
}

int main() {
    char big_string[128];
    int i;
    for(i = 0; i < 128; i++){
        big_string[i] = 'A';
    }
    overflow_function(big_string);
    exit(0);
}
```

Výsledkem po kompilaci a spuštění je:

```
user@host:~$ gcc overflow.c -o overflow
user@host:~$ ./overflow
Neoprávněný přístup do paměti (SIGSEGV)
```

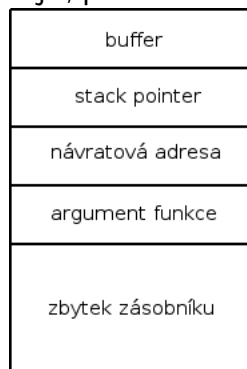
Ve funkci byl vytvořen buffer o velikosti 20 znaků, to je námi očekávaná hodnota. Do tohoto bufferu však vložíme 128 znaků. Z důvodu přetečení paměti program selže. Tato situace nahrává hackerovi. Programátor nepředpokládal, že někdo bude jako vstupní parametr funkce zadávat tolik znaků. Program, který jinak funguje naprosto normálně, najednou "spadne". Tato skutečnost programátora jen utvrdí v použití vstupního filtru, validátoru dat.

### Přetečení zásobníku

Vraťme se ještě k předchozímu programu. Když se zavolá funkce `overflow_function`, vytvoří se na stacku rámeček zásobníku. Když se funkce poprvé zavolá, vypadá rámeček podobně jako na obrázku 1.1.

Když se funkce pokusí zapsat 128 bajtů dat do 20 bajtového bufferu, dojde k zapsání 20 bajtů do bufferu a zbylých 108 bajtů přepíše návratovou adresu, ukazatel na rámeček a argument funkce. Když podprogram skončí, pokusí se pomocí záznamu o návratové adrese skočit zpět do hlavní funkce `main()`, ale tato adresa je přepsána nesmyslnou hodnotou, v našem případě samými `A`, což je hexadecimálně `0x41`. Skočí tedy na adresu `0x41414141`, která je neplatná. V každém případě program skončí. Tento jev je nazýván přetečení zásobníku (stack-based overflow).

K přetečení může dojít i v jiném segmentu paměti. Ale přetečení zásobníku je z hlediska využití nejzajímavější, protože můžeme přepsat návratovou adresu a tím změnit tok vykonávání programu. Kdyby totiž byla



návratová adresa jiná než `0x41414141` a byla by to platná adresa nějakého kódu, který eventuálně podstrčí útočník, znamenalo by to vykonávání právě podstrčeného kódu. Tato metoda se nazývá tzv. infekce kódu (bytecode infection). Bytecode je kus chytře podstrčeného assemblerovského kódu, který je vložen do bufferu. Takový kód má několik omezení. Musí být samostatný a nesmí obsahovat speciální znaky v instrukcích, protože by měl vypadat jako data v bufferu.

Jeden z nejpoužívanějších kódů je tzv. shellkód, který spouští příkazový interpret shell. Podaří-li se přelstít nějaký suid root program (takový program, který vlastní root a má přidělen suid bit) tak, aby se s jeho právy spustil shell, má útočník vyhráno a získává plnou kontrolu nad strojem a systémem.

1.1: Rámeček zásobníku

Vycházíme-li ze situace, že máme k dispozici takto postižený program, postačí jen vhodně vygenerovat data, která předáme programu. Data budou obsahovat shellkód, který po skončení programu přepíše návratovou adresu a umožní tak převzít kontrolu nad strojem jako s právy uživatele root. Aby byl shellkód spustitelný, je žádoucí předem znát jeho adresu, což může být u dynamického zásobníku poněkud problematické. Pro správnou funkci je nutné přepsat pouze čtyři bajty návratové adresy. Pokud tato podmínka není splněna, program je ukončen. Pro řešení této situace existují dvě známé techniky.

První je nazvána NOP sled. NOP je instrukce, která nedělá vůbec nic. Použijeme ji ve formě velkého pole NOP, které je ukončeno shellkódem. EIP bude neustále inkrementován, až narazí na shellkód.

Druhou technikou je zaplnění konce bufferu mnoha po sobě jdoucími návratovými adresami. Vytvořený buffer bude vypadat:

```
NOP sled    shellkód    opakující se návratová adresa
```

U obou technik je ale nutné znát alespoň přibližné umístění bufferu v paměti, aby bylo možné uhodnout návratovou adresu. Jednou z možností je aproximování umístění v paměti na základě ukazatele na zásobník (registr ESP). Odečtením offsetu od ESP získáme relativní adresu libovolné proměnné. Jelikož je prvním

prvkem na zásobníku buffer, který je přepsán, správná adresa by měla být ukazatel na zásobník, takže offset by měl být blízko 0.

### Přetečení v segmentech halda a bss

Nejen u zásobníku může dojít k přetečení. Stejný problém se týká i haldy a segmentu bss. Tento typ útoku není tak standardizovaný jako přetečení zásobníku, ale to neznamená, že není stejně účinný. Jelikož v tomto případě neexistuje návratová adresa, která by mohla být přepisována, závisí tyto útoky na přepisování některých důležitých proměnných, jež se nacházejí v paměti za bufferem. Pokud je například za bufferem uložen seznam přístupových práv nebo autentizační informace, může přetečení bufferu znamenat získání plných přístupových práv (v nejhorším případě). V jiných případech se může jednat o ukazatel na funkce obsažené v těle programu. Přetečením bufferu a přepsáním těchto adres můžeme spustit shellkód. Protože exploitování v segmentech halda a bss je mnohem více závislé na rozložení paměti v programu, není tak triviální nalézt zranitelná místa programu.

Jeden poměrně chytrý způsob, jak pomocí tohoto druhu přetečení získat superuživatelská práva, ukazují následující řádky. Využijeme jednoduchého programu pro zápis do souboru. Podmínky: nastavení suid bitu, vlastnictví roota a samozřejmě náchylnost k tomuto druhu přetečení. Napaden bude velmi důležitý soubor `/etc/passwd`. Právě pomocí tohoto přetečení bude vytvořen nový uživatelský účet. Prostým editováním `/etc/passwd`. Tento účet bude bez hesla a bude superuživatelský (uid bude rovno 0). Odpovídající řádka v `/etc/passwd` by mohla vypadat následovně.

```
r00t::0:0:hacker:/root:/bin/bash
```

Vkládaný řetězec je nutné drobně upravit. Je důležité, aby data měla správnou délku, která způsobí správné přetečení. Jinými slovy, musí přesně padnout do masky rámce zásobníku.

## Lokální útoky – formátovací řetězce

Podobně jako u metod využívajících přetečení paměti je cílem této metody změna toku programu. Exploity tohoto druhu závisí taktéž na programových chybách a na rozdíl od přetečení paměti nemusejí mít přímý dopad na bezpečnost. Ve srovnání s exploitačními technikami postavenými na přetečení paměti jsou techniky formátování řetězce poměrně snadno odstranitelné [2]. Ovšem za předpokladu, že se o těchto chybách ví.

### Formátovací řetězce a funkce printf()

Formátovací řetězce nacházejí uplatnění ve formátovacích funkcích, např. ve funkci `printf()`. Příkladem může být následující kus kódu.

```
printf("Vypiseme neco na standardni vystup:%d",some_var);
```

"Vypiseme neco na standardni vystup:%d" je formátovací řetězec. Funkce jej vytiskne, ale nejprve provede speciální operaci, projde jej a otestuje na výskyt formátovacích parametrů. V příkladu je formátovacím parametrem `%d`. Tento parametr zapříčiní vypsání argumentu `some_var`, tedy hodnoty proměnné jako celé číslo v desítkové soustavě. Většina parametrů získává hodnotu proměnné předáním hodnoty. Existují i výjimky, které očekávají ukazatel. Jedná se především o řetězcové proměnné a data ve formě bajtů. Speciální činnost funkce zmíněná na začátku kapitoly proběhne vždy, když funkce najde ve formátovacím řetězci formátovací parametr. Každý formátovací parametr očekává samostatnou proměnnou. Funkce má proměnný počet parametrů. Důležitá je také skutečnost, že argumenty funkce jsou uloženy na zásobníku v opačném pořadí [4]. Rámec zásobníku je vyobrazen na obrázku 1.3.

V případě, že se budeme zabývat zobrazováním čísel, je jasné, že číslo bude fyzicky uloženo stále stejně ve formátu dvojkového doplňku. To, co ovlivňujeme formátovacím parametrem, je jeho reprezentace.

vrchol zásobníku
adresa formátovacího řetězce
první hodnota
adresa první hodnoty
druhá hodnota
adresa druhé hodnoty
.
.
dno zásobníku

Vraťme se nyní k počtu parametrů. Bylo řečeno, že je třeba uvést tolik hodnot, kolik je formátovacích parametrů. Co však nastane v případě, že jeden nebo více parametrů chybí? Funkce jednoduše vytiskne hodnotu zásobníku na odpovídající pozici. Právě této vlastnosti je při útocích tohoto typu využito. Zjednodušeně řečeno měníme počet argumentů předávaných nebo očekávaných formátovací funkcí.

Někdy se stane důsledkem nepozornosti, že programátor namísto `printf("%s",string)` zapíše pouze `printf(string)`. Funkčně je vše v pořádku, formátovací funkce vypíše všechny značky řetězce dané adresou `string`. Funkce končí, když narazí na nulový bajt. Následuje jednoduchý program pro demonstraci výše zmíněné chyby.

1.3: Rámec zásobníku funkce

`printf()`

```
#include <stdlib.h>
int main(int argc,char *argv[]){
char text[1024];
static int testVal=-72;
if(argc < 2){
printf("Pouziti:%s<text k vytisknuti>\n", argv[0]);
exit(0);
}
strcpy(text, argv[1]);
//spravny zpusob vytisteni
printf("Spravne:\n");
printf("%s",text);
printf("\n");
//nespravny zpusob vytisteni
printf("Spatne:\n");
printf(text);
printf("\n");
//debug
printf("[*]testVal@0x%08x = %d0x%08x\n",&testVal,testVal,testVal);
exit(0);
}
```

Oba způsoby fungují korektně, pokud programu předložíme jakýkoliv text bez formátovacích parametrů. Změna ovšem nastane, pokud programu předáme parametr, jenž obsahuje formátovací řetězec.

```
user@host:~$ ./format text%x
Spravne:
text%x
Spatne:
textbf800828
[*]testVal@0x08049728=-720xfffffb8
```

Program vyhodnotil součást řetězce `text%x` jako další formátovací parametr, avšak bez příslušné proměnné ve formátovací funkci. Program tedy vypsál obsah paměti následující za proměnnou `testVal` na zásobníku a tuto hodnotu formátoval jako hexadecimální číslo. Postup je možné využít opakovaně k prozkoumání obsahu zásobníku. Postačí vhodně zvolit parametr. Například požadujeme-li prozkoumání 50 paměťových míst za poslední validní hodnotou, předáme parametr například ve tvaru `perl -e print "%80x."x50; [2]`.

Příklad, který byl uveden, ukazoval vypsání paměti pomocí parametru předávaného hodnotou (`%x`). Použijeme-li však parametr předávaný odkazem (`%s`), pokusí se program vypsát obsah paměti na adrese dané parametrem programu. Pokud je zvolena neplatná adresa, program havaruje. Pokud je adresa platná, bude vypsán obsah na této paměťové adrese.

Čtení obsahu rámce zásobníku není příliš zajímavé, daleko vhodnější je možnost zápisu do něj. Je použita naprosto stejná technika jako při čtení zásobníku; jedinou odlišností je použití jiného formátovacího parametru. Jedná se o parametr `%n` [4].

## Závěr

Další, velmi výhodnou možností je použití „Warning options“ překladače GCC. Jelikož nemám žádné znalosti jiných překladačů, zaměřím se pouze na GCC. Mezi tyto „Warning options“ nebo „Warning flags“ patří například `-Wall`, `-Wformat`, `-Wno-format-extra-args` nebo `-Wformat-security`. Tyto přepínače mimo jiné kontrolují formát argumentů některých funkcí (`printf()`, `scanf()`, ...) nebo varují při možnosti vzniku bezpečnostních problémů [5].

Popsané lokální útoky nebo lépe řečeno programátorské chyby lze většinou poměrně snadno odstranit. Daleko větším problémem je jejich nalezení. Díky komunitě, která se kolem Linuxu rozrostla, není tento problém neřešitelný. Je kladen poměrně velký důraz na otevřenost i v otázkách bezpečnosti, nic není skrýváno. Tato skutečnost napomáhá tomu, aby byly chyby brzo nalezeny a odstraněny. Doporučením pro předcházení problémům je častá aktualizace systému a sledování diskusních fór, konferencí a jiných informačních zdrojů s tematikou bezpečnosti. Mimo tyto základní metody je vhodné pro další ochranu před tímto druhem útoků systematicky získávat dobré programátorské návyky, pečlivé prověřování a testování napsaného kódu.

V poslední době je stále oblíbenější řešení knihovna [Libsafe](#) <sup>(133)</sup>. Obrovskou výhodou je jednoduchá instalace. Navíc není nutné měnit již zkompileované programy. Podle autorů zastaví tato knihovna drtivou většinu chyb způsobených přetečením bufferu. Libsafe chrání pouze provádění aplikací, ne samotného jádra operačního systému. Tento kód chrání jedna z alternativ Libsafe systému [OpenWall](#) <sup>(134)</sup>. Jedná se o jadernou záplatu (kernel patch). Řeší jisté bezpečnostní problémy na úrovni jádra, například vytváří tzv. nespustitelný zásobník (nonexecutable user stack area), čímž zabraňuje spouštění kódu v zásobníku. Zaměřuje se především na otázky spojené se správou paměti (nechrání haldu).

Další alternativou k Libsafe je například [StackGuard](#) <sup>(135)</sup>.

Knihovna Libsafe pracuje následovně. Namísto běžně používaných funkcí jazyka C, jako je například `gets()`, `strcpy()`, `getwd()`, `scanf()` nebo `sprintf()`, vyvolá službu Libsafe. Při tomto volání zjistí, zda-li se cíl volání nachází na zásobníku, a také jestli některá z adres nepřepíše data mimo rámec zásobníku. V případě, že jsou předcházející podmínky splněny, je operace zablokována [1].

Zmíněné funkce provádějí přesně to, co programátor požaduje. Tato vlastnost je důvodem jejich nebezpečnosti. Je dosaženo vysoké rychlosti na úkor bezpečnosti. Druhým problémem je skutečnost, že programátor není schopen zadat velikost bufferu. Nabízí se další řešení: používat bezpečné verze funkcí jako `fgets()` místo `gets()` nebo `strncpy()` namísto `strcpy()` [1].

Lokální útoky nejsou samozřejmě tvořeny jen přetečením paměti a chybami formátovacího řetězce. Dále je možné setkat se s útoky zvanými návrat do libc (*Return-to-libc attacks*), přetečením celých čísel (*Integer overflows*), chyby souběhu (*Race conditions*) atd.

## Literatura

- [1] TOXEN, Bob. *Bezpečnost v Linuxu: Prevence a odvrácení napadení systému*. Brno: Computer Press, 2003. 876 s. ISBN 80-7226-716-7.

- [2] ERICKSON, J. *HACKING – umění exploitace*. Brno: Zoner PRESS, 2005. 263 s. ISBN 8086815-21-8.
- [3] HATCH, B. – LEE, J. – KURTZ, G. *HACKING bez tajemství – Linux*. Brno: Computer Press, 2003. 644 s. ISBN 80-7226-869-4.
- [4] HEROUT, P. *Učebnice jazyka C*. České Budějovice: Kopp, 2001. 263 s. ISBN 80-85828-21-9.
- [5] Contributors to GCC. *Manuálové stránky GCC*, [poslední aktualizace 18. 4. 2007].

■

# System vo firemnej sieti – VLSM

CIDR je mechanizmus interpertovania IPv4 adriess zavedený v roku 1993 a pre smerovanie na IPv6 sieťach je samozrejmosťou. Rieši obmedzenia historického rozdeľovania podsietí do tried (A, B, C). CIDR (Classless Inter-Domain Routing) využíva techniku VLSM (Variable-Length Subnet Masking) na efektívne alokovanie podsietí v rámci väčšej siete a o tejto technike, a ako ju využiť, pojednáva nasledujúci článok.

**Michal Sabol, Ma-  
tej Gagyí**

Tento text sa zaoberá históriou a pokročilou teóriou tvorby sietí (so staticky pridelenými externými IP adresami). Popisovaný systém je vhodný pre veľmi veľké siete, ale aj menšie siete (menej ako 256 IP adries). Pre jeho pochopenie budeme potrebovať základné znalosti o TCP/IP sieťach – adresovanie na TCP/IP sieti, konfigurácia stackých IP adries na TCP/IP sieti a znalosť princípu masky podsietí. V prípade, že tieto znalosti nemáte, no ovládáte anglický jazyk, môžete využiť odkazy na konci článku.

## Historia podsietí

Ešte pred zavedením tried podsietí v roku 1981 určovalo adresovanú podsieť iba prvých 8 bitov IP adresy (prvé číslo v adrese: X.x.x.x). Zvyšných 24 bitov adresovalo konkrétne zariadenie v danej podsieti (posledné 3 čísla x.X.X.X).

Jednoduchšie povedane, súčasne mohlo existovať iba 256 sietí a každá z nich mohla mať 16 777 214 účastníkov. Táto koncepcia vyhovovala v časoch, kým existovalo iba niekoľko veľkých sietí ako ARPANET (predchodca Internetu). S príchodom LAN sietí začalo byť jasné, že 256 sietí nestačí.

Náplastou na tento problém bola zmena definície IP adries ([RFC 791](#) <sup>(136)</sup>) a zavedenie tried sietí. Triedy umožnili rôznym sieťam mať rôznu dĺžku adresy. Prvá a posledná IP adresa neboli podľa novej definície v sieti povolené (kde IP adresa pozostáva z adresy siete a druhú časť tvoria len nuly alebo jednotky). Jednotlivé triedy siete boli určené prvými bitmi v adrese.

Trieda	Zaciatok	Koniec	CIDR ekvivalent
Trieda A	0.0.0.0	127.255.255.255	/8
Trieda B	128.0.0.0	191.255.255.255	/16
Trieda C	192.0.0.0	223.255.255.255	/24
Trieda D (multicast)	224.0.0.0	239.255.255.255	/4
Trieda E (rezervovana)	240.0.0.0	255.255.255.255	/4

## CIDR bloky

CIDR (Classless Inter-Domain Routing), ako už bolo povedané v prexe, odstránil triedy podsietí. Tie totiž problémy nevyriešili na dlho. Bez tried je možné vytvoriť si sieť o takej veľkosti, akú potrebujeme, a nemrať zbytočne IP adresami.

Novinkou sú masky sietí (subnet mask). Maska siete sa pozostáva so sekvencie jedničiek a sekvencie núl, pričom celá maska má dĺžku 32 bitov. Masty siete sa zapisujú dvomi spôsobmi: buď sú reprezentované podobne ako IP adresy (napríklad 255.255.255.0), alebo počtom jednotiek v maske za lomítkom (napríklad /24). Druhý zápis sa zvyčajne pripája za IP adresu (napríklad 192.168.1.1/24).



Kazdá podsieť je jeden CIDR blok a každý CIDR blok môže byť rozdelený na viacero menších CIDR blokov. Podsiete v CIDR bloku nemusia mať rovnako dlhú masku. Táto agregácia súvislých sieťových adries umožňuje vynechať informácie o niekoľko-násobnom delení podsietí z nadradených smerovacích tabuliek.

Dĺžka sieťovej masky určuje aj počet IP adries na sieti. Na sieti musíme vyhradiť 2 IP adresy, jednu ako adresu siete (IP adresa routera) a jednu pre broadcast. Počet použiteľných IP adries teda bude  $2^n - 2$ . Napríklad na sieti 80.42.0.0/16 bude použiteľných IP adries  $2^{16} - 2 = 65534$ .

Dĺžka masky	Počet zariadení	Maska podsiete	Dĺžka masky	Počet podsietí
/30	2	255.255.255.252	/30	64
/29	6	255.255.255.248	/29	32
/28	14	255.255.255.240	/28	16
/27	30	255.255.255.224	/27	8
/26	62	255.255.255.192	/26	4
/25	126	255.255.255.128	/25	2
/24	254	255.255.255.000	/24	1

Nasledujúca tabuľka (vpravo) informuje, na koľko podsietí s rovnakou dĺžkou masky môže byť rozdelená sieť triedy C (sieť s maskou /24).

## VLSM

Na delenie pridelených adresných priestorov na podsiete existuje mechanizmus, ktorý nám pomáha šetriť drahocennými IP adresami a meniť parametre siete podľa potreby. Môžeme si odrezať približne toľko IP adries, koľko potrebujeme.

VLSM nám pri tvorbe podsietí pomôže a jeho výstupom samozrejme nie je len konfigurák na routeri, ale podľa analýzy môžeme napísať aj dokumentáciu (tú nadriadený správca siete bude isto vyžadovať). Tento článok má za cieľ vysvetliť podstatu VLSM, ale nemože byť vyčerpávajúcou učebnicou VLSM.

### Príklad

Najjednoduchšie je vysvetliť VLSM na príklade. Uvažujme teda, že naša firma má k dispozícii rozsah sieťových adries 220.32.16.0 až 220.32.16.255 (myslíme tým externé IP adresy prístupné z Internetu). Inak povedane, vlastníme sieť 220.32.16.0/24 (sieť triedy C).

Pobočka	Veľkosť siete
Pobočka	Veľkosť siete
Pobočka A	12 zariadení
Pobočka B	25 zariadení
Pobočka C	6 zariadení

Túto sieť chceme rozdeliť na podsiete pre naše 3 pobočky A, B, C. Pobočka A má priamy prístup k Internetu a pobočky B a C sú s ňou spojené pomocou WAN sériových liniek. Naše pobočky majú nasledujúce počty zariadení.

Keďže každé WAN spojenie tvorí takisto sieť s dvomi zariadeniami (routere, ktoré spája), musíme s týmito sieťami počítať pri návrhu podsietí. Vlastne celý návrh robíme kvôli WAN spojeniam, pretože nechceme po WAN linkách posielat dáta, ktoré nemusíme.

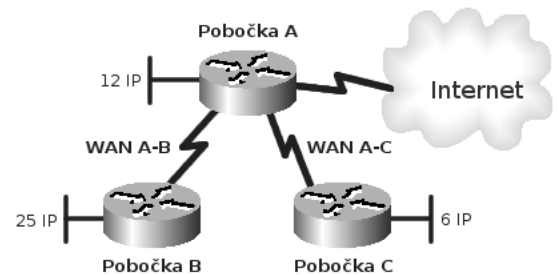
Ak by sme netvorili podsiete, každý paket by bol šírený cez celú sieť a aj cez všetky WAN linky. To nie je prípustné, hlavne na pomalých spojeniach a všade tam, kde musíme za prenesené dáta platiť. Takto nám vzniknú ďalšie 2 podsiete. Túto situáciu zobrazuje obrázok 1.

WAN spojenie	veľkosť siete
Pobočky A a B	2 zariadenia
Pobočky A a C	2 zariadenia

## Riešenie

Postupovať budeme tak, že pre každú sieť zvolíme takú dlhú masku, aby mala dostatok IP adries pre všetky zariadenia, no nealokovala viac IP adries, ako je naozaj nutné. Čiže pobočka A by dostala masku /28 (sieť bude mať maximálne 14 zariadení).

Ak by sme pobočke A dali menšiu masku, nemali by sme pre ňu dostatok IP adries. Ak by sme jej priradili väčšiu sieť, nevyužili by sme viac než polovicu IP adries siete (napríklad pri /27, ktorá pojme 30 zariadení, by sme premrhali 18 IP adries).

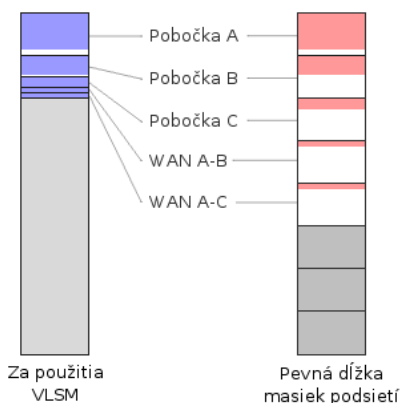


Pre jednoduchosť je lepšie začať od siete s najväčším počtom zariadení a postupovať smerom dole. Nasledujúca tabuľka je takto zotriedená. Len dodám, že prvá adresa v rozsahu alokovaných IP adries je zároveň adresa siete.

Pobočka	Zariadení	Maska	Rozsah IP adries
pobočka B	25	/27	220.32.16.0 – 22.32.16.31
pobočka A	12	/28	220.32.16.32 – 22.32.16.47
pobočka C	6	/29	220.32.16.48 – 22.32.16.55
WAN A – B	2	/30	220.32.16.56 – 22.32.16.59
WAN A – C	2	/30	220.32.16.60 – 22.32.16.63

Táto schéma je riešenie, no z ďaleka nie je progresívne. Ak by sa potreby ktorejkoľvek pobočky zväčšili a počet potrebných IP adries by prekročil počet adries alokovaných podsietou pobočky, museli by sme celý návrh prepracovať odznova. To nie je prípustné pre veľké siete, napríklad siete triedy B, s ktorými sa môžeme stretnúť u veľkých ISP.

Na tomto príklade si môžeme ukázať, ako pomocou VLSM šetríme IP adresami na sieti. Všimnite si, že v našom príklade sme alokovali 64 IP adries. Ak by sme nepoužili VLSM a sieť jednoducho rozdelili na 8 rovnako veľkých sietí, alokovali by sme 140 IP adries.



Aj keby sme chceli byť chytrí a špeciálne pre WAN spojenia by sme vytvorili 3 malé siete, alokovali by sme 104 IP adries. To je síce menší rozdiel, no takto by sme mohli byť stále chytřejší a chytřejší, až by sme sa dostali k istému ako s VLSM. Obrázok 2 ilustruje tento rozdiel v šetrení IP adresami.

## Zaver

V pokračovaní tohto článku si našu sieť rozdelíme efektívnejšie a ukážeme si dva spôsoby vizualizácie rozdelenia siete. Zaradíme tak isto aj také informácie, ktoré budú požadované v diskusii pod týmto článkom.

■

## Jaderné noviny – Video4Linux2 – 7 (ovládání)

Je známo, že uživatelé nejsou nikdy spokojeni; nestačí jim, že mohou z kamery vidět video, hned by si chtěli hrát s parametry jako jas, kontrast atd.

Robert Krátký

Po dokončení [šesté části seriálu](#) <sup>(137)</sup> už víme, jak video zařízení nastavit a jak tam a zpět přenášet snímky. Je však známo, že uživatelé nejsou nikdy spokojeni; nestačí jim, že mohou z kamery vidět video, hned by si chtěli hrát s parametry jako jas, kontrast atd. Takové úpravy mohou být prováděny ve video aplikaci, ale pokud to hardware podporuje, tak má své výhody je provádět rovnou v hardwaru. Například úprava jasu může znamenat ztrátu dynamického rozsahu, pokud se provádí až po přenosu. Naproti tomu hardwarová úprava umožňuje zachování plného rozsahu, kterého je senzor schopen. Hardwarové úpravy samozřejmě také ulehčí procesoru hostitelského systému.

Současný hardware mívá možnost nastavovat za běhu velkou spoustu parametrů. Konkrétní funkce těchto parametrů se však zařízení od zařízení výrazně liší. I tak jednoduchá úprava jako "jas" může zahrnovat přímočaré nastavení registru nebo také o dost komplexnější změnu nějaké obskurní transformační matice. Bylo by fajn co nejvíce těchto detailů před aplikací skrýt, ale má to své meze. Příliš abstraktní rozhraní by mohlo znemožnit plné využití možností hardwaru.

Ovládací rozhraní V4L2 se snaží věci co nejvíce zjednodušit a zároveň umožnit plné využití hardwaru. Začíná definicí standardních názvů ovládacích funkcí; patří mezi ně `V4L2_CID_BRIGHTNESS`, `V4L2_CID_CONTRAST`, `V4L2_CID_SATURATION` a mnoho dalších. Máme také booleovské ovládací prvky funkcí jako vyvážení bílé, horizontálního a vertikálního zrcadlení atd. Kompletní seznam předdefinovaných ID hodnot ovládacích prvků najdete ve [specifikaci V4L2 API](#) <sup>(138)</sup>. K dispozici je také možnost ovládacího prvku specifického pro ovladač, ale ty budou zjevně používány jen specializovanými aplikacemi. Soukromé ovládací prvky začínají u `V4L2_CID_PRIVATE_BASE`.

Jako obvykle poskytuje API V4L2 mechanismus, pomocí kterého může aplikace ovládací prvky snadno očíslovat. Pokud to chce provést, provede `ioctl()` volání, která se dostanou do V4L2 ovladače prostřednictvím zpětného volání `vidioc_queryctrl()`:

```
int (*vidioc_queryctrl)(struct file *file, void *private_data,
                       struct v4l2_queryctrl *qc);
```

Za normálních okolností ovladač zaplní strukturu `qc` informacemi o požadovaném ovládacím prvku nebo, pokud není ovládací prvek podporovaný, vrátí `EINVAL`. Tato struktura má několik polí:

```
struct v4l2_queryctrl
{
    __u32          id;
    enum v4l2_ctrl_type type;
    __u8          name[32];
    __s32         minimum;
    __s32         maximum;
    __s32         step;
    __s32         default_value;
    __u32         flags;
    __u32         reserved[2];
};
```

Ovládací prvek, o který se jedná, bude předán prostřednictvím `id`. Aplikace může poskytnout ID ovládacího prvku s nastaveným bitem `V4L2_CTRL_FLAG_NEXT_CTRL`; v takovém případě by měl ovladač vrátit ID ovládacího prvku následujícího po tom, který předala aplikace. `id` by však každopádně mělo být nastaveno na ID ovládacího prvku, který je skutečně popisován.

Všechna ostatní pole nastaví ovladač, aby popsal zvolený ovládací prvek. Datový typ ovládacího prvku je uveden v `type`; může to být `V4L2_CTRL_TYPE_INTEGER`, `V4L2_CTRL_TYPE_BOOLEAN`, `V4L2_CTRL_TYPE_MENU` (pro neměnnou sadu voleb) nebo `V4L2_CTRL_TYPE_BUTTON` (pro ovládací prvek, který při nastavení provede nějakou akci a ignoruje jakoukoliv předanou hodnotu). `name` popisuje ovládací prvek; může být použito v rozhraní, které aplikace prezentuje uživateli. U celočíselných ovládacích prvků popisují `minimum` a `maximum` rozsah hodnot, které prvek implementuje, a `step` udává velikost kroků daného rozsahu. `default_value` je výchozí hodnota, ale aplikovat lze pouze na ovládací prvky typu celé číslo, boolean a menu. Ovladače by měly hodnoty ovládacích prvků nastavovat na výchozí jen při inicializaci; stejně jako ostatní parametry zařízení, i tyto by měly přežít volání `open()` a `close()`. Z toho vyplývá, že `default_value` vůbec nemusí odpovídat aktuální hodnotě ovládacího prvku.

Kromě toho máme ještě sadu příznaků, které ovládací prvek dále popisují.

- `V4L2_CTRL_FLAG_DISABLED` znamená, že je ovládací prvek zakázán, takže by jej aplikace neměly brát v potaz.
- `V4L2_CTRL_FLAG_GRABBED` značí, že ovládací prvek nelze dočasně změnit – například proto, že jej převzala jiná aplikace.
- `V4L2_CTRL_FLAG_READ_ONLY` označuje ovládací prvky, kterých je možné se dotazovat, ale nelze je měnit.
- `V4L2_CTRL_FLAG_UPDATE` znamená, že úprava této hodnoty může ovlivnit jiné ovládací prvky.
- `V4L2_CTRL_FLAG_INACTIVE` značí ovládací prvek, který pro aktuální konfiguraci zařízení není relevantní.
- A `V4L2_CTRL_FLAG_SLIDER` prozrazuje, že by aplikace tento prvek měly reprezentovat posuvníkem.

Aplikace si mohou přát dotázat jen na několik ovládacích prvků, které mají naprogramovány, nebo mohou chtít očíslovat celou sadu. Ve druhém případě začnou u `V4L2_CID_BASE` a budou pokračovat až k `V4L2_CID_LASTP1`, např. pomocí `V4L2_CTRL_FLAG_NEXT_CTRL`.

U ovládacích prvků typu menu (typ `V4L2_CTRL_TYPE_MENU`) budou aplikace pravděpodobně chtít očíslovat i možné hodnoty. Příslušné zpětné volání:

```
int (*vidioc_querymenu)(struct file *file, void *private_data,
                        struct v4l2_querymenu *qm);
```

Struktura `v4l2_querymenu` vypadá takto:

```
struct v4l2_querymenu
{
    __u32    id;
    __u32    index;
    __u8     name[32];
    __u32    reserved;
};
```

Při vstupu je `id` hodnota ID daného ovládacího prvku typu menu a `index` je hodnota indexu konkrétní položky menu. Indexové hodnoty začínají nulou a pokračují až k hodnotě `maximum`, kterou vrátila `vidioc_queryctrl()`. Ovladač vyplní `name` položky menu; pole `reserved` by mělo být nastaveno na nulu. Jakmile aplikace ví o dostupných ovládacích prvcích, pustí se pravděpodobně do zjišťování a měnění jejich hodnot. Struktura používaná pro tento účel je relativně jednoduchá:

```
struct v4l2_control
```

```
{
  __u32 id;
  __s32 value;
};
```

Aby se mohla aplikace dotázat na konkrétní ovládací prvek, nastaví `id` na ID prvku a provede volání, které se do ovladače dostane jako:

```
int (*vidioc_g_ctrl)(struct file *file, void *private_data,
                    struct v4l2_control *ctrl);
```

Ovladač by měl nastavit `value` na aktuální nastavení ovládacího prvku. Měl by se také samozřejmě ujistit, že o tom konkrétním prvku ví, a případně vrátit `EINVAL`, pokud by se aplikace pokoušela dotazovat na neexistující prvek. Pokusy o dotazování na tlačítkové ovládací prvky by také měly vracet `EINVAL`. Požadavek na změnu ovládacího prvku dopadne takto:

```
int (*vidioc_s_ctrl)(struct file *file, void *private_data,
                    struct v4l2_control *ctrl);
```

Ovladač by měl ověřit `id` a ujistit se, že `value` je v mezích povoleného rozsahu. Pokud je vše v pořádku, měla by být nová hodnota nastavena v hardwaru. Nakonec stojí za zmínku, že v rámci V4L2 je podporováno samostatné [rozhraní s rozšířenými ovládacími prvky](#) <sup>(139)</sup>. Je určeno pro poměrně komplexní ovládací prvky; v praxi se využije především pro parametry k enkódování a dekodování MPEG. Rozšířené ovládací prvky mohou být seskupovány do tříd a jsou podporovány i 64bitové celočíselné hodnoty. Rozhraní je podobné tomu obyčejnému; podrobnosti najdete ve specifikaci.

■

## Jaderné noviny – 44 a 45/2007

Testování suspendu a hibernace. Citát: comp.os.linux větší než alt.sex. Swap přes NFS. Citát: Přetypování na void není příliš populární. Oprava chybných optimalizací překladače. 2.6.24-rc2, „Spící město“. Citát: Synchronizujte, než to zkusíte. Squashfs míří do hlavní řady jádra. Citát: Plánuješ žít věčně? Subsystem distribuovaného úložiště dokončen. Citát: Navrhněte rozumné zamykací schéma. Bezpečnostní cíle AppArmor. Citát: Dobré pro vývoj, špatné pro obchod. Stabilní větev 2.6.

Jirka Bourek

Následující obsah je ©KernelTrap.

### Testování suspendu a hibernace

2. listopad, originál <sup>(140)</sup>

Následující patche přidávají nové testovací možnosti pro suspend a hibernaci,

psal Rafael J. Wysocki. <sup>(141)</sup> První patch přidává možnost testovat jádro suspendu (STD), aniž by se skutečně suspendovalo, což je užitečné při pátrání po problémech s ovladači atd. Druhý modifikuje jádro hibernace tak, aby mohlo využívat ty samé možnosti (jsou o něco lepší, než existující testovací režimy hibernace, protože ty nemohou skutečně testovat globální ACPI metody.)

Testovací možnosti zavádějí nový atribut `/sys/power/pm_test_level`, který akceptuje číslo od 1 do 5, přičemž každá hodnota simuluje jinou úroveň kódu suspendu a hibernace. Rafael ho popsal: 5 – testuje zmrazení procesů; 4 – testuje zmrazení procesů a suspendování zařízení; 3 – testuje zmrazení procesů, suspendování zařízení a globální ovládací metody platformy; 2 – testuje zmrazení procesů, suspendování zařízení, globální ovládací metody platformy a zakáže nebootovací CPU; 5- testuje zmrazení procesů, suspendování zařízení, globální ovládací metody platformy, zakáže nebootovací CPU a suspenduje platformová/systémová (platform/system) zařízení.

Jestliže je suspend zahájen normálním způsobem, jádro suspendu provede všechny normální operace až do bodu stanoveného testovací úrovní. Posléze počká 5 vteřin a pokračuje resume operacemi potřebnými k přechodu systému do plně funkčního stavu.

Rafael dodal, že nastavení `pm_test_level` na 0 testovací funkce zakáže.

### Citát: comp.os.linux větší než alt.sex

Tento měsíc bylo dosaženo dalšího milníku: všimněte si polí „Nedávný provoz“ (jak v kilobytech, tak v počtu zpráv) – podle `newsstatu` je `c.o.linux` skutečně aktivnější, než `alt.sex`.

– Linus Torvalds, zpráva z 3. listopadu 1992 <sup>(142)</sup> na Linux kernel mailing list.

### Swap přes NFS

2. listopad, originál <sup>(143)</sup>

Problém se swapováním přes síť má stejný problém jako swapování obecně: potřebujete paměť, abyste mohli uvolnit paměť. Normálně se to řeší použitím `mempools`, což můžeme vidět ve vrstvě BIO,

popisoval Peter Zijlstra <sup>(144)</sup>. Swapování přes síť má ten problém, že síťový subsystem nepoužívá alokace o pevné délce, ale velice spoléhá na `kmalloc()`. Kvůli tomu jsou `mempools` nepoužitelné.

Prvních patnáct patchů nastavuje obecný framework pro rezervování paměti. Patche 16 – 23 zajišťují, aby síťový stack tento framework používal. Peter popisoval: **Dokončení síťového přenosu zahrnuje přijetí paketů, což, když není paměť, je poněkud obtížné. A dokonce i když je paměť, není zajištěné, že potřebný paket přijde v době, kdy se k té paměti dostaneme. Řešením tohoto problému najdeme ve faktu, že o síti se předpokládá, že je ztrátová. I když není paměť k přijetí paketů, síťová karta je musí zahodit. Proto to přesuneme do síťového stacku.**

Patche 24 – 26 nastavují infrastrukturu pro swapování do souborového systému místo do blokového zařízení, což je potom využíváno posledními patchi a **nakonec je NFS upraven tak, aby využíval novou infrastrukturu sítě a vm a poskytl swap přes NFS.**

Když byla zpochybnována užitečnost těchto patchů, Peter odpověděl: **Mezi firmami je o to velký zájem, proto to dělám. Typické využití jsou 1) clustery/blade clustery, kde místní disk znamená problém s náklady (údržba při selhání, teplo, atd.) a 2) virtualizace, kde přesunutí úložiště na síť zjednodušuje migraci.**

## Citát: Přetypování na void není příliš populární

**Přetypování na (void) není příliš populární zvyklost. Přišli jste na to, že to skutečně dělá něco užitečného?**

– Andrew Morton, [zpráva z 19. 10. 2007](#) <sup>(145)</sup> na Linux kernel mailing list.

## Oprava chybných optimalizací překladače

2. listopad, [originál](#) <sup>(146)</sup>

Nedávná diskuze o **chybných optimalizacích** <sup>(147)</sup> GCC vedla [Linuse Torvaldse](#) <sup>(148)</sup> k poznámce: **Ohledně gcc mám smíšené pocity.** Upřesnil, že na jednu stranu považuje gcc za skvělý překladač tvořený výbornými vývojáři, ale na druhou za starý projekt. **Během času se v něm nahromadily různé pozůstatky a pročištění je často téměř nemožné.** Dodal, že ačkoliv chyby v překladači mohou být frustrující, znepokojuje ho především to, jak někteří z vývojářů vnucují definice jazyka. **Zdá se, že si myslí, že důležitější je čistá specifikace jazyka jako právník, než řešit skutečné problémy uživatelů.**

Andrew Haley poznamenal, že existuje aktivní skupina vývojářů, která se snaží gcc vylepšit, a řekl: **Dejte nám šanci.**

Vrátil se také k původní chybné optimalizaci, která celou diskuzi začala, s tím, že oprava byla začleněna do všech otevřených větví gcc. **Patch portujeme do všech otevřených větví. Nicméně tento patch ovlivňuje pouze jeden konkrétní případ, kdy gcc zavádí data race. Jsme si jistí, že jsou další, které opravené nejsou.** K tomu dodal, že aktivně prohledávají kód, aby našli a odstranili podobné chyby.

## 2.6.24-rc2, Spící město

6. listopad, [originál](#) <sup>(149)</sup>

**Jo, nepřipomínejte mi to – mám zpoždění,**

začal Linus Torvalds **oznámení o druhém release candidate 2.6.24** <sup>(150)</sup>. **Prakticky nebylo nic zvláštního, co by zdržovalo vydání, prostě jsem jenom minulý týden zapomněl.**

Pokračoval výpisem několika ze změn:

**Není tam příliš mnoho vzrušujících věcí. Nějaké aktualizace v architekturách – MIPS, arm, blackfin, x86, sparc[64], sh, s390. Také různé aktualizace ovladačů – libata, IDE, síťování, DVB. Nějaký spad ze změn v rozsyp-sesbírej (scatter-gather)** <sup>(151)</sup>, **úklid v plánovači a oprava statistik využití CPU, které se v určitém bodě zbláznily.**

Linus poznamenal, že přestože nebyly žádné významné změny, je krátký záznam příliš dlouhý na to, aby ho do diskuze zaslal. Těm, kdo ho chtějí vidět, navrhl příkaz `git shortlog v2.6.24-rc1`, nicméně, upřímně řečeno to není žádný Lev Tolstoj. Jestliže máte problémy se spaním, můžete to zkusit vytisknout a vzít do postele: nebude to víc než pár stránek (tiskněte oboustranně a zachraňte strom), ale pochybuji, že se skutečně dostanete na konec. Spící město.

## Citát: Synchronizujte, než to zkusíte

Rád bych věděl, jestli ovladač disketových mechanik funguje pro 2 (nebo více) jednotky. Nikdo se o tom zatím nijak nevyjádřil. Proto proveďte sync, než to zkusíte (jenom pro případ...).

– Linus Torvalds, zpráva z 6. 11. 1991<sup>(152)</sup> na Linux Activists mailing list.

## Squashfs míří do hlavní řady jádra

7. listopad, originál<sup>(153)</sup>

S potěšením oznamuji další vydání Squashfs. Toto vydání je dvacáté druhé během o něco více než pěti let. Squashfs 3.3 má mnoho hezkých vylepšení, jak ve vlastním souborovém systému (větší bloky a rozptýlené – sparse – soubory), tak v nástrojích Mksquashfs a Unsquashfs,

prohlásil Phillip Lougher<sup>(154)</sup> o nejnovějším vydání komprimovaného read-only linuxového souborového systému. Poznamenal, že stále potřebuje opravit endianitu souborového systému, pak že se pokusí dostat Squashfs do hlavní řady jádra. Nové rysy v tomto nejnovějším vydání zahrnují:

1. Maximální velikost bloku byla zvětšena na 1 Mbyte a výchozí velikost bloku byla zvýšena na 128 kilobytů. To zlepšuje kompresi.
2. Nyní jsou podporovány rozptýlené soubory. Rozptýlené soubory jsou soubory, které mají rozsáhlé oblasti nealokovaných dat, které se běžně nazývají díry. Tyto soubory Squashfs detekuje a ukládá mnohem efektivněji. To zlepšuje kompresi a výkon při čtení.

## Citát: Plánuješ žít věčně?

Opravdu potřebujeme 128bitový čas? Opravdu chceš žít věčně?

– Avi Kivity, zpráva ze 7. 11. 2007<sup>(155)</sup> na Linux kernel mailing list.

## Subsystém distribuovaného úložiště dokončen

8. listopad, originál<sup>(156)</sup>

S potěšením vám oznamuji sedmé a konečné vydání subsystému distribuovaného úložiště,

prohlásil Evgenij Poljakov<sup>(157)</sup>, když dokončil věci z TODO seznamu na webových stránkách projektu.<sup>(158)</sup> Vydání nazval `squizzed black-out of the dancing back-aching hippo` s poznámkou `jasně to ukazuje na můj stav`.

Novými vlastnostmi v tomto vydání jsou podpora kontrolních součtů, rozšířená autokonfigurace pro detekci a automatické povolení kontrolních součtů, pokud jsou podporovány vzdáleným hostitelem, nové soubory v sysfs pro označení zadaného node jako čistého (synchronizován) nebo špinavého (nesynchronizován) a oprava mnoha chyb.

Evgenij vydal první verzi<sup>(159)</sup> svého subsystému distribuovaného úložiště v červenci 2007. V září vysvětlil, že to byl první krok ve větším projektu distribuovaného souborového systému<sup>(160)</sup>, který plánuje. Koncem října Andrew Morton<sup>(161)</sup> psal, že práce vypadá připravená pro začlenění<sup>(162)</sup> do -mm jádra.



## Citát: Navrhnete rozumné zamykací schéma

Budeme rádi, když navrhnete rozumné zamykací schéma schopné vypořádat se s tímhle nepořádkem.

– Al Viro <sup>(163)</sup>, zpráva z 19. 10. 2007 <sup>(164)</sup> na Linux kernel mailing list.

## Bezpečnostní cíle AppArmoru

8. listopad, originál <sup>(165)</sup>

Tento dokument si klade za cíl specifikovat bezpečnostní cíle, kterých chce dosáhnout AppArmor, aby uživatelé mohli zjistit, jestli AppArmor splní jejich požadavky, a vývojáři, jestli splňuje, co slibuje. Tento dokument *není* obecné vysvětlení způsobu fungování AppArmoru, ani to není vysvětlení, proč by někdo měl použít AppArmor místo jiného systému.

Takto začíná dokument od Crispina Cowana <sup>(166)</sup>, který následoval po nedávném návrhu <sup>(167)</sup> Arjana van de Vena zdokumentovat účel tohoto bezpečnostního modulu. Crispin pokračoval:

AppArmor má za úkol chránit systémy před útočníky, kteří využívají chyby v programech běžících v systému. Hrozba spočívá v tom, že útočník může způsobit, aby zranitelná aplikace udělala něco neočekávaného a nechtěného. AppArmor na toto nebezpečí reaguje omezením aplikace tak, aby mohla přistupovat pouze ke zdrojům, které potřebuje ke správnému běhu; efektivně aplikaci nutí k běhu s "nejmenšími právy".

Aplikace mají přístup k nejrůznějším zdrojům – souborům, meziprocesové komunikaci, síti, kvalifikacím (capabilities), mohou spouštět jiné aplikace. Účelem nejmenších práv je omezit poškození, které škodíci uživatel nebo kód mohou způsobit tím, že se zabrání přístupu ke všem zdrojům, které aplikace nepotřebuje k tomu, aby mohla pracovat. Například nastavení pro webový server může povolit přístup k webovým stránkám pouze pro čtení, což útočníkovi, který prolomil web server, zabrání měnit stránky.

## Citát: Dobré pro vývoj, špatné pro obchod

Zveřejňování chyb je dobré pro vývoj, ale špatné pro obchod.

– Rik van Riel, zpráva ze 2. 10. 2007 <sup>(168)</sup> na Linux kernel mailing list.

## Stabilní větve 2.6

9. listopad, originál <sup>(169)</sup>

Po posledním vydání jsem řekl, že si myslím, že 2.6.22.12 bude poslední v řadě 2.6.22.y. Od té doby jsem obdržel množství dalších patchů, které by bylo hezké mít ve stromě .22.y,

řekl <sup>(170)</sup> Greg KH <sup>(171)</sup>.

Takže prozatím ponechám strom 2.6.22.y otevřený a budu čas od času vydávat nové verze tak, jak budou přicházet. Dělán to jenom ze svých sobeckých důvodů, protože ho každý den používám na svém stroji s openSUSE 10.3, jehož jádro z této verze vychází :)

Greg KH a Chris Wright udržovali stabilní 2.6.x.y sadu patchů pro jádra 2.6.x a 2.6.(x-1) od března 2005. Správce stabilního jádra 2.4 Willy Tarreau spravoval patche proti větvi 2.6.20 od srpna 2007, nicméně ohlašoval, že přejde k údržbě stabilní větve 2.6.22, až s tím Greg přestane. Adrian Bunk pokračuje ve správě stabilní větve 2.6.16.

■

## Jaderné noviny – 46 a 47/2007

Opravy plánovače. Distribuovaný síťový souborový systém Ceph. Zapisovatelný mmap pro FUSE. 2.6.24-rc3, malé opravy a pročištění. Git ve Windows. CFS – backporty verze 24. Modulární IO plánovače. Citát: Andrew Morton: Nijak oslňující výkon. Theodore T'so: paměti relativně zlevňují. David Miller: Vývoj ledovcovým tempem. Linus Torvalds: Zdarma dvě kopie Linuxu navíc. Theo de Raadt: Překvapivě svobodná licence. Alan Cox: Smysluplné výchozí hodnoty. David Miller: Kvalita hlášení o chybách.

Jirka Bourek

Následující obsah je ©KernelTrap.

### Opravy plánovače

12. listopad, [originál](#) <sup>(172)</sup>

Ingo Molnár <sup>(173)</sup> zaslal Linusi Torvaldsovi <sup>(174)</sup> požadavek k začlenění nejnovějších oprav CFS <sup>(175)</sup>. ◊CFS <sup>(176)</sup>, zcela férový plánovač, byl do hlavní řady jádra začleněn <sup>(177)</sup> v červenci 2007. Poprvé byl zařazen v jádře 2.6.23 <sup>(178)</sup> vydaném v říjnu 2007. Zdá se, že plánovač se rychle stabilizuje, lze ho nalézt jen v málokterém seznamu oprav nejnovějšího kódu, který čeká na začlenění. Ingo změny shrnul:

Jsou tam dvě mezisubsystemové skupiny oprav – tři commity, které řeší opravu překladu KVM na !SMP – od Aviho mám potvrzeno, že se začlení přes git strom plánovače, protože mění centrální include soubor. Další je oprava regrese účtování času CPU [CPU time accounting] u powerpc od Paula Mackerrase.

Zbývajících 14 commitů: jedna oprava pádu (docházelo k ní přes nový souborový systém kontrolních skupin <sup>(179)</sup> [control-groups filesystem]), oprava regrese účtování zpoždění [delay-accounting], dvě opravy regrese výkonu, oprava latence, dvě malé opravy regrese plánovacího chování a sedm pročištění.

### Distribuovaný síťový souborový systém Ceph

15. listopad, [originál](#) <sup>(180)</sup>

Ceph je distribuovaný síťový souborový systém vytvořený tak, aby poskytoval excelentní výkonnost, spolehlivost a škálovatelnost s POSIXovou sémantikou. V této konferenci často vidím frustraci z toho, že chybí škálovatelný distribuovaný GPL souborový systém s dostatečně robustní replikací a obnovením po pádu, který by běžel na běžném hardwaru. Rád bych si myslel, že – s trochou péče – by Ceph mohl tuto díru zaplnit,

oznámil Sage Weil <sup>(181)</sup> v ◊LKML <sup>(182)</sup>.

Zmínil, že souborový systém byl původně vyvinut v rámci jeho závěrečné PhD práce, a pokračoval seznamem jeho vlastností – POSIX sémantika, škálovatelnost od pár po tisíce uzlů [nodes], podpora pro petabajty dat, vysoce dostupný design bez bodů selhání, n-cestná replikace dat přes několik uzlů, automatické znovuvyrovňování dat podle toho, jak jsou uzly přidávány nebo odstraňovány, a na FUSE založený klient. K tomu dodal, že pracuje na lehkém jaderném klientovi, flexibilních snapshotech, kvótách a vylepšení bezpečnosti. Nakonec porovnal Ceph <sup>(183)</sup> s jinými souborovými systémy:

Na rozdíl od clusterových souborových systémů, jako je GFS, OCFS2 a GPFS, které spoléhají na symetrický přístup všech klientů ke sdíleným blokovým zařízením, Ceph odděluje správu dat a metadat mezi nezávislé cluster serverů podobně jako Lustre. Nicméně na rozdíl od Lustre metadatum a ukládací uzly

[storage nodes] běží zcela v uživatelském prostoru a nepotřebují žádnou podporu v jádře. Ukládací uzly pro ukládání dat používají buď přímo blokové zařízení nebo velký soubor, nebo mohou použít existující souborový systém (XFS atd.) pro lokální ukládání (momentálně s oslabenou bezpečnostní sémantikou). Data souborů jsou rozdělena [striped] na velké kusy a rozprostřena mezi skladovací uzly, aby se zatížení rozdělilo mezi co nejvíce uzlů a tak se dosáhlo velké propustnosti. Když některý ze skladovacích uzlů selže, data jsou distribuovaným způsobem znovureplikována přímo ukládacími uzly (s nějakou koordinací od cluster monitoru), díky čemuž je celý systém extrémně efektivní a škálovatelný.

## Zapisovatelný mmap pro FUSE

16. listopad, [originál](#) <sup>(184)</sup>

Miklos Szeredi zaslal požadavek na komentáře k patchi nazvanému [design zapisovatelného mmap pro FUSE](#) <sup>(185)</sup>. Vysvětlil: [Mapování zapisovatelné sdílené paměti je něco, co jsem se ve <FUSE](#) <sup>(186)</sup> pokoušel implementovat věčně. Nyní to mám snad konečně vymyšlené, bude to fungovat s bash-sdíleným-mapováním a fsx-linux. Žádám o komentáře k tomu přístupu.

Pokračoval popisem patche:

`fuse_writepage()` alokuje novou dočasnou stránku s GFP\_NOFS—\_GFP\_HIGHMEM. Zkopíruje obsah originální stránky a zařadí do fronty uživatelského souborového systému, který tuto dočasnou stránku používá, požadavek na ZÁPIS. Z pohledu VM je zápis dokončen okamžitě: stránka je odstraněna z [radix stromů](#) <sup>(187)</sup> a příznaky PageDirty a PageWriteback jsou vynulovány. Per-bdi čítač zápisů není dekrementován, dokud se zápis opravdu nedokončí. [...] Při „zašpiňování“ stránky fuse čeká, dokud se nedokončí předchozí zápis, pak teprve pokračuje. Tím je zajištěno, že naráz může být použita jenom jedna dočasná stránka pro jednu stránku v cache.

## 2.6.24-rc3, malé opravy a pročištění

17. listopad, [originál](#) <sup>(188)</sup>

Linus Torvalds [oznámil třetí release candidate](#) <sup>(189)</sup> nadcházejícího jádra 2.6.24 a shrnul: Hmmm... Spousta malých oprav, nějaká pročištění a pár věcí (jako aktualizace cris), které nejsou tak docela kompletní, ale běžného uživatele nijak neovlivní a snad je bude snazší synchronizovat v budoucnu. Opravy síťových ovladačů, několik aktualizací IDE a infiniband, nějaké pozdní aktualizace cpufreq a aktualizace hwmon.

Na straně architektury jsou kromě výše zmíněné aktualizace cris nějaké aktualizace sh, arm, powerpc a mips a také jedno konečné pročištění po [sjednocení x86](#) <sup>(190)</sup>. (A myslím to skutečně vážně. Zbytek může počkat po 2.6.24, s tímhle už je konfigurace x86 skutečně téměř spojena a jak i386, tak x86\_64 jsou v konfigurátoru pouze speciální případy architektury x86.)

## Git ve Windows

19. listopad, [originál](#) <sup>(191)</sup>

Tato krásně temná čtvrtá hodina ranní je stejně dobrá příležitost jako každá jiná k tomu, abych vám nabídl páté číslo msysGit Herald, ne-tak-úplně-čtrnáctideníku, který vás informuje o snaze přinést jeden z nejlepších systémů pro správu zdrojového kódu ubohým duším, které musejí pracovat ve Windows,

psal [Johannes Schindelin](#) <sup>(192)</sup> v mailové konferenci [gitu](#). Dodal, že projekt se konečně koncentruje na snahu zajistit, aby git fungoval ve Windows, a že se konečně podařilo zprovoznit instalátor. Na [domovské stránce projektu](#) <sup>(193)</sup> se píše:

Bohužel je Git na Windows oficiálně podporován jenom za použití Cygwinu. Nicméně existuje fork (který snad teď bude brzy spojen s "oficiálním" gitem), jenž umožňuje zkompilevat git s použitím MinGW/MSys.

Je trochu složité zajistit všechno, co je pro běh potřeba (MSys, iconv, Tcl/Tk, gcc, make, zlib, regex, atd.), takže se projekt snaží dodávat jediný .zip (ve skutečnosti 7-zipem zabalený instalátor), který je možné rozbalit, a dvojklikem na `msys.bat` všechno nastavit. Pak můžete hned začít hackovat ve svém oblíbeném nástroji pro správu zdrojových kódů.

## CFS – backporty verze 24

19. listopad, [originál](#) <sup>(194)</sup>

Ingo Molnár oznámil, že 24. verze jeho zcela férového plánovače <sup>(195)</sup> je nyní k dispozici portovaná pro jádra 2.6.24-rc3, 2.6.23.8, 2.6.22.13 a 2.6.21.7. Poznamenal, že od posledního backportu došlo k významným změnám: 36 souborů změněno, 2359 vložení (+), 1082 smazání (-). To je 187 samostatných commitů od 32 autorů. 99 % těchto změn je již v upstreamu v Linusově git stromu a budou uvolněny jako část 2.6.24 (čekají 4 commity, které jsou v malém patchi 2.6.24-rc3-v24).

Také vyzdvihl několik významnějších zlepšení:

Vylepšená interaktivita díky vlastnosti „virtuální intervaly“ [virtual slices] od Petera Zijlstra. Se zvyšující se zátěží plánovač zkracuje virtuální časové intervaly [timeslices], které úlohy dostanou, takže aplikace mají stále stejnou latenci, než se dostanou k CPU. (Což funguje, dokud intervaly nedosáhnou hodnoty minimální granularity.)

CONFIG\_FAIR\_USER\_SCHED je nyní k dispozici ve všech portovaných jádrech a váhy pro jednotlivé uživatele jsou konfigurovatelné přes `/sys/kernel/uids`. Hodně bylo vylepšeno také skupinové plánování.

## Modulární IO plánovače

26. listopad, [originál](#) <sup>(196)</sup>

Adrian Bunk zaslal [patch](#) <sup>(197)</sup>, který mění linuxové IO plánovače na nemodulární volbu, jež by při kompilaci vybrala jeden IO plánovač s tím, že není žádná velká výhoda mít modulární plánovače a ani se nezdá, že by se to moc používalo.

Dodal, že odstranění volby by v každém obrazu jádra ušetřilo 2 kB. Jens Axboe moc nadšený nebyl: Naprosto nACK, modulární plánovače používám pro testování neustále. Jenom to, že ty je náhodou nepoužíváš, není důvod, aby byly odstraněny.

Když Adrian upozornil na to, že se nezdá, že by nějaké distro mělo IO plánovače k dispozici jako moduly, Jens prohlásil, že to je chyba, a rýpnul si: Je to už dlouho, co jsem naposledy považoval `.config` z `distra` za `benchmark/příručku` jakéhokoliv druhu.

Adrian pokračoval otázkou na technické důvody, proč pokračovat s podporou čtyř různých IO plánovačů, a vyjádřil obavu, že by to v jednotlivých plánovačích mohlo vést k chybám, které by nikdo nenahlásil. Jens řekl, že má v plánu vytvořit perfektní IO plánovač, ale že v současnosti různé plánovače nabízejí lepší výsledky pro různé typy zátěže.

Po tvrdé práci a testování bychom měli být schopni zbavit se předjímajícího [anticipatory] plánovače. Ten stále poráží `CFQ` <sup>(198)</sup> pro některé zátěže, ve kterých je dobrý deadline, takže hned teď to nepůjde.

Arjan van de Ven jeden důvod nabídl: Je nejmíň jeden technický důvod, proč mít víc než jeden plánovač: některé typy ukládacích zařízení (např. velké EMC stroje, stejně jako disky bez rotujících částí [solid state]) se chovají jako disky, ale nemají žádný postih při seekování; jakýkoliv čas CPU obětovaný na to, aby se zabránilo seekování, je u nich ztráta času, takže taková zařízení určitě potřebují jiný IO plánovač; takový, který bude značně odlehčen.

Jens potom potvrdil: vždycky je riziko spojené s „duplikací“, stejně jako u několika ovladačů pro stejný hardware. O tom se nehádám.

## Citáty

### Andrew Morton: Nijak oslňující výkon

Trvalo mi dobrých pět hodin, abych tohle aspoň trochu přinutil ke zkompilování. Bylo potřeba 20 - 30 patchů a několik stromů a 5 – 10 patchů jsem zahodil. To není nijak oslňující výkon.

– Andrew Morton <sup>(199)</sup>, zpráva z 6. 11. 2007 <sup>(200)</sup> na Linux kernel mailing list.

### Theodore T'so: Paměti relativně zlevňují

Paměti jsou momentálně levnější – mluvíme o možná třiceti až čtyřiceti dolarech za megabajt, pokud do stroje můžete dát SIMMy. Upradovat počítač ze dvou na čtyři mega nestojí až *tak* moc peněz.

– Theodore T'so <sup>(201)</sup>, zpráva z 15. 11. 1991 <sup>(202)</sup> na Linux Activists mailing list.

### David Miller: Vývoj ledovcovým tempem

Jasně, když vyvíjíš v takovém tempu, v jakém se pohybují ledovce, jako to dělají u Solarisu, nepřidáváš do cyklického subsystému žádné vlastnosti ani nepracuješ na vylepšování škálovatelnosti, pak to jistě může být bezchybné a nemusí se na to 6 let sahat.

– David Miller <sup>(203)</sup>, zpráva z 14. 11. 2007 <sup>(204)</sup> na Linux Kernel mailing list.

### Linus Torvalds: Zdarma dvě kopie Linuxu navíc

Jestli to chci do Linuxu přidat? Jestli někdo stránkování implementuje, dostane zdarma dvě kopie Linuxu navíc. Co říkáte na tuhle nabídku?

– Linus Torvalds <sup>(205)</sup>, zpráva ze 17. 11. 1991 <sup>(206)</sup> na Linux Activists mailing list.

### Theo de Raadt: Překvapivě svobodná licence

Firmware (pokud nebyl uložen v seeprom v zařízení) pro uticom(4). Byl někým zkompilován z kompletního zdrojového kódu, který TI zveřejnila pod překvapivě svobodnou licenci. (Která pravděpodobně není ani nijak vynutitelná, protože před ni zapomněli dát frázi o copyrightu... kde si najímají právníky? Každopádně z toho mají všichni prospěch.)

– Theo de Raadt <sup>(207)</sup>, zpráva ze 16. 11. 2007 <sup>(208)</sup> na OpenBSD Source Changes mailing list.

### Alan Cox: Smysluplné výchozí hodnoty

To je velmi arogantní úhel pohledu. Nemusím být televizní technik na to, abych mohl televizi používat. Distribuce by měly ve standardní instalaci poskytovat smysluplné výchozí hodnoty. Jádro jim k tomu už poskytuje mechanismy.

– Alan Cox <sup>(209)</sup>, zpráva ze 17. 11. 2007 <sup>(210)</sup> na Linux Kernel mailing list.

### David Miller: Kvalita hlášení o chybách

Tenhle případ bude dobrý příkladem v situaci, až příště začne další stupidní vlákno o tom, jak nikdo neřeší hlášení o chybách. Podle mě je to spíš záležitost kvality těch hlášení.

– David Miller <sup>(211)</sup>, zpráva z 15. 11. 2007 <sup>(212)</sup> na Linux Kernel mailing list.

## Jaderné noviny – 7. 11. 2007

Aktuální verze jádra: 2.6.24-rc2. Citáty týdne: Ingo Molnár, Rusty Russell, David Miller. Správa paměti grafických procesorů. ID procesů ve světě s více jmennými prostory.

Robert Krátký

### Aktuální verze jádra: 2.6.24-rc2

Aktuální předverze je (k 7. 11. 2007) 2.6.24-rc2, vydaná<sup>(213)</sup>, poněkud opožděně, 6. listopadu. Linus<sup>(214)</sup> vysvětlil: **To zdržení nezpůsobilo nic konkrétního. Prostě jsem na to minulý týden zapomněl.**

Patche začleněné od -rc1 jsou většinou opravy, ale přibylo i pár vylepšení DCCP, příznak pro potlačení varování o použití zastaralých rozhraní, asynchronní API pro upozorňování na události pro SCSI/SATA (říká systému, kdy byl vložen disk), japonský překlad dokumentu `SubmittingPatches`, API pro správu napájení ATA spojení [ATA link] a trochu další práce na [spojení x86](#)<sup>(215)</sup>. Seznam patchů najdete v [krátkém changelogu](#)<sup>(216)</sup> a podrobnosti v [dlouhém](#)<sup>(217)</sup>.

Od vydání -rc2 se zatím do hlavního git repozitáře žádné další patche nedostaly.

Starší jádra: 2.6.22.11<sup>(218)</sup> a 2.6.22.12<sup>(219)</sup> byly vydány 2. a 5. listopadu. Tyto verze obsahují řádku patchů, z nichž jeden se týká bezpečnosti – pro oba dva lidi, kteří používají souborový systém Minixu. Greg Kroah-Hartman<sup>(220)</sup> nedávno uvedl<sup>(221)</sup>, že navzdory [předchozím vyjádřením](#)<sup>(222)</sup> bude řada 2.6.22.x ještě nějakou dobu pokračovat.

2.6.16.56<sup>(223)</sup> a 2.6.16.57<sup>(224)</sup> vyšly 1. a 5. listopadu. Obsahují docela dost oprav, z nichž několik řeší bezpečnostní problémy.

### Citáty týdne: Ingo Molnár, Rusty Russell, David Miller

Už teď máme v jádře příliš mnoho nekompletních konceptů a API. Možná mám přehnané obavy, ale bojím se, že dopadneme jako v případě [kvalifikací](#)<sup>(225)</sup> nebo [sendfile](#) – příliš brzy začleněný kód, který nebyl léta dokončen, možná nebude dokončen nikdy. VMS a WNT tyhle věci podle mě vyřešily lépe – jejich API systémy zahrnují vše a jsou kompletní – a to i v okrajových případech.

– Ingo Molnar<sup>(226)</sup>

Starosti mi dělá to, že `stringbuf` byl dobrý, ale ne skvělý. A přitom na jádro vždycky pohlížím jako na baštu opravdu dobrého C kódu, který pečlivě vybrušují rozumní programátoři. V tomto případě však i ty neměřené optimalizační pokusy ukazují na nedostatek snahy zkušených jaderných hackerů.

– Rusty Russell<sup>(227)</sup>

Důvod, proč je kód v jádře tak dobrý, není to, že by se tam už dostal v perfektním stavu – je to proto, že ho pořád dokola ořezáváme, dokud se nedostane do toho perfektního stavu. Tenhle ořezávací systém nám umožňuje během týdne a půl nacpat do stromu 25 MB změn a ono to nakonec všechno funguje.

– David Miller<sup>(228)</sup>

### Správa paměti grafických procesorů

Správa video hardwaru je už dlouhou dobu slabou stránkou Linuxu (a svobodných operačních systémů obecně). X Window System schytá hodně kritiky za problémy v této oblasti, ale pravda je taková, že potíže se projevují na více místech a jádro X Window Systému nikdy práci neusnadnilo. Grafické procesory

(GPU) jsou stále výkonnější, a to až do té míry, že z jistého pohledu jde ve většině systémů o nejrychlejší procesory. Ale jaderná podpora programování GPU je pozadu. Na nápravě této situace se však intenzivně pracuje a pro začlenění do jádra byla právě představena jedna důležitá komponenta.

Bývaly časy, kde se video paměť sestávala z jediného framebufferu, ze kterého byly na displej posílány pixely; byla starost systémového procesoru dostat do toho framebufferu užitečná data. Se současnými GPU je situace komplikovanější: typický GPU může pracovat s několika různými druhy paměti:

- Video RAM (VRAM) je vysokorychlostní paměť instalovaná přímo na videokartě. Většinou ji lze vidět na systémové PCI sběrnici, ale není to pravidlo. Je dost pravděpodobné, že v této paměti bude framebuffer, ale také mnoho dalších druhů dat.
- Na mnoha levnějších systémech je „video RAM“ ve skutečnosti vyhrazená část systémové paměti. Taková RAM je dána stranou, aby ji mohl využívat GPU, a pro nic jiného ji nelze využít. I adaptéry s vlastní pamětí mohou mít vyhrazenou oblast obecné RAM.
- Video adaptéry obsahují jednoduchou jednotku pro správu paměti (graphics address remapping table, GART = přemapovací tabulka grafických adres), kterou je možné použít k mapování různých stránek systémové paměti do adresního prostoru GPU. Výsledkem je, že GPU má kdykoliv přístup k libovolné (roztrošené [scattered]) části stránek systémové RAM.

Každý z druhů video paměti má různé vlastnosti a omezení. S některými se rychleji pracuje (z pohledu CPU nebo GPU). Některé druhy VRAM nemusí být CPU schopni přímo adresovat. Paměť může, ale nemusí podporovat koherenci dat v keši [cache coherency] – rozlišení, které vyžaduje opatrné programování, aby se předešlo poškození dat a problémům s výkonem. A grafické aplikace mohou chtít pracovat s mnohem většími objemy video paměti, než lze pro GPU v určité chvíli zviditelnit.

To všechno představuje problém se správou paměti, který, i když je podobný potřebám správy systémové RAM, má svá vlastní speciální omezení. Takže grafičtí vývojáři už léta říkají, že by Linux potřeboval řádnou správu paměti, která je přístupná GPU. Ale celá léta už fungujeme bez správce paměti, což má za následek, že je správa prováděna nepěknou kombinací kódu v X serveru, jádře a často proprietárních ovladačích. Naštěstí se zdá, že už tento stav nebude trvat dlouho, a to díky modulu s mapou překladových tabulek [translation-table maps (TTM)], který z většiny napsali Thomas Hellstrom, Eric Anholt a Dave Airlie. TTM poskytuje správce paměti pro potřeby GPU a grafických klientů.

Hlavní objekt, který – z pohledu uživatelského prostoru – TTM spravuje, je „buffer objekt“. Bufferový objekt je kus paměti alokovaný aplikací a případně sdílený více aplikacemi. Obsahuje oblast paměti, se kterou může pracovat GPU. Je zaručeno, že bufferový objekt nezmizí, dokud na něj nějaká aplikace udržuje odkaz; umístění bufferu se však může měnit.

Jakmile aplikace vytvoří bufferový objekt, může jej namapovat do svého adresního prostoru. Podle toho, kde se buffer právě nachází, může mapování vyžadovat přemístění bufferu do takového druhu paměti, který může CPU adresovat (přesněji řečeno by takový přesun vynutil výpadek stránky při pokusu aplikace o přístup k namapovanému bufferu). Je samozřejmě také nutné se postarat o koherenci dat v keši.

Přijde chvíle, kdy bude potřeba, aby byl tento buffer k dispozici GPU pro nějakou operaci. TTM vrstva poskytuje speciální „validační“ `ioctl()` pro přípravu bufferů ke zpracování; validace bufferu by opět mohla zahrnovat přesun nebo nastavení mapování pomocí GART. Adresa, na které bude GPU k bufferu přistupovat, nebude známa až do validace; po validaci nebude buffer odstraněn z adresního prostoru GPU, dokud se s ním bude pracovat.

To znamená, že jádro musí vědět o dokončení práce s daným bufferem; stejně tak to musí vědět aplikace. TTM vrstva k tomu účelu poskytuje „plotové“ [fence] objekty. Plot je speciální operace, která je umístěna do FIFO grafického procesoru. Když je plot spuštěn, nastaví signál, který značí, že všechny instrukce zařazené do fronty před ním, byly spuštěny a že GPU už nebude přistupovat k žádným bufferům, které k nim byly přiřazeny. Způsob fungování signalizace závisí na GPU; může se jednat o přerušení nebo jen zapsání hodnoty na speciální místo paměti. Při spuštění signálu jsou všechny přiřazené buffery označeny,

aby bylo zřejmé, že na ně už GPU neodkazuje, a odešle se oznámení procesům v uživatelském prostoru, které to zajímá.

Na vytiženém systému může běžet množství grafických aplikací, z nichž každá se snaží grafickému procesoru cpát buffery. Nebylo by překvapení, kdyby poptávka po bufferech, které může GPU adresovat, přesáhla množství paměti, ke které má GPU přístup. Takže TTM vrstva bude muset v reakci na přichozí požadavky buffery přesouvat. U bufferů mapovaných pomocí GART může stačit odmapovat stránky z bufferů, které zrovna nejsou pro GPU operace validovány. V ostatních případech může být nutné překopírovat obsah bufferů do jiného druhu paměti, případně i pomocí GPU hardwaru. V takových situacích je nejprve nutné buffery invalidovat v tabulkách stránek všech uživatelských procesů, které je mají namapovány, aby se pojistilo, že do bufferu nebude během přesunu zapisováno. Jinými slovy, TTM tak opravdu rozšiřuje systémový kód pro správu paměti.

Další otázka je nasnadě: co se stane, když chtějí grafické aplikace používat více paměti, než může celý systém poskytnout? Stránky běžné systémové RAM, které se používají jako video paměť, jsou pevně stanoveny (a nelze je použít pro nic jiného), takže jejich počet musí být omezen. Stávající řešení je omezit počet těchto stránek na zlomek dostupné spodní paměti – 1 GB na 4GB, 32bitovém systému. Bylo by fajn mít možnost rozšířit tuto paměť zápisem nepoužívaných stránek do swapu, ale linuxová implementace swapu neumí pracovat se stránkami, které patří jádru. Vypadá to, že do budoucna se plánuje umožnit X serveru vytvořit velký virtuální rozsah pomocí `mmap()`, který by pak šlo swapovat. To však zatím implementováno není.

Další informace o kódu TTM lze nalézt v [tungstengraphics.com/mm.pdf](http://tungstengraphics.com/mm.pdf) <sup>(229)</sup>. V tuto chvíli to funguje s opatchovanou verzí ovladače i915 od Intelu, další budou doplněny. TTM bylo navrženo <sup>(230)</sup> k začlenění do jádra -mm a do hlavního v rámci 2.6.25. Do té doby půjde především o posouzení uživatelského API, které by bylo těžké po začlenění měnit. Bohužel to vypadá, že k tomuto API není moc dokumentace.

## ID procesů ve světě s více jmennými prostory

Článek o kontejnerech z minulého týdne <sup>(231)</sup> pojednával o jmenných prostorech ID procesů. Účelem těchto jmenných prostorů je určovat, které procesy uvidí proces uzavřený v kontejneru. Rozšířené používání PID pro identifikaci procesů způsobilo, že tento konkrétní patch prošel dlouhým vývojem, než byl do 2.6.24 začleněn. Vypadá to však, že se objevily potíže, které by mohly zabránit tomu, aby byla tato funkce v další verzi jádra k dispozici. Jak se stává často, i tentokrát jsou největší problémy s uživatelským API.

1. listopadu Ingo Molnár <sup>(232)</sup> poukázal na to <sup>(233)</sup>, že některé otázky, které v roce 2006 pokládal <sup>(234)</sup> Ulrich Drepper <sup>(235)</sup>, zůstaly bez odpovědi. Všechny se týkají toho, co se stane, když používaný PID unikne ze jmenného prostoru, do kterého patří. Existuje několik jaderných API týkajících se meziprocesové komunikace a synchronizace, kde by se to mohlo stát. Reálné signály obsahují informace o ID procesů, stejně jako fronty zpráv SYSV. Přejemnějším bude k tomu, aby tato rozhraní správně fungovala ve všech jmenných prostorech PID, potřeba, aby jádro provádělo překlady PID, kdykoliv PID překročí hranici jmenného prostoru.

Největší potíž však asi bude s robustním mechanismem futexů, jenž používá PID ke sledování, který proces vlastní určitý futex v daný okamžik. Jedním z klíčových konceptů u futexů je rychlý způsob jejich získávání (pokud se o futex nesoupeří), který vůbec nevyžaduje zapojení jádra. Jenže při tom rychlém získávání je také nastaveno pole PID. Není tedy možné dát jádru možnost provést kouzlení s překladem PID, aniž by to mělo negativní dopad na výkon, který byl jednou z hlavních motivací pro futexy.

Ingo, Ulrich a další, kterým tento problém dělá starosti, by byli rádi, kdyby byly jmenné prostory PID v 2.6.24 úplně vypnuty. Není však jasné, jak by takové řešení vypadalo, nebo jestli je vůbec nutné.

Ulrich by pravděpodobně preferoval <sup>(236)</sup>, aby byly částečně odstraněny rozmanité možnosti, které jádro poskytuje pro ovládání sdílení jmenných prostorů. S rozhraním, které je v 2.6.24-rc1, může proces, jenž zavolá `clone()`, vyžadovat, aby byl potomek umístěn do nového jmenného prostoru PID, ale aby ostatní jmenné prostory (například souborové systémy nebo síťování) byly sdíleny. To si, podle Ulricha, říká o potíže:



Celý ten přístup, který umožňuje zapínání a vypínání jmenných prostorů, je prostě špatný. Nechte, ať je to buď všechno nebo nic – aspoň u těch problematických jako NEWPID. Přístup ke stejnému souborovému systému, ale zároveň používání jiného jmenného prostoru PID, to prostě nebude fungovat.

Sjednocení několika voleb jmenných prostorů do jediného bitu „nový kontejner“ by pomohlo s nedostatkem klonovacích bitů, ale problémy API by to možná vůbec nevyřešilo. Dokonce i procesy s odlišnými jmennými prostory souborových systémů by mohly najít ten samý futex prostřednictvím souboru, který by byl viditelný v obou prostorech. Předávání [dokladů](#) <sup>(237)</sup> přes sokety unixových domén by to mohlo také pěkně zamotat. Zdá se, že jsou místa, kde se používají PID, na která si zatím nikdo nevzpomněl.

Další možný přístup, který v popisované debatě ani nezazněl, by bylo vytvoření globálně unikátních PID hodnot, které by fungovaly ve všech jmenných prostorech. Současná 32bitová hodnota PID by mohla být rozdělena na dvě pole, přičemž nejvýznamnější bity by značily, ve kterém jmenném prostoru je PID (obsažený v nejméně významných bitech) definován. Většinou by byla potřeba jen ta druhá část; ta by byla interpretována ve vztahu k aktuálnímu jmennému prostoru PID. Ale v případech, kdy by to dávalo smysl, by se použilo plné, unikátní PID. To by umožnilo funkcím, jako jsou futexy, fungovat ve všech jmenných prostorech PID.

I tak by s tím však byly problémy. Základní myšlenka jmenných prostorů PID je úplné schování procesů, které jsou vně aktuálního prostoru; vytvoření a používání globálních PID v této izolaci dělá díry. A je jisté, že by se v uživatelském API objevily komplikace, které by bylo těžké jen tak zaplácnout.

Pak je tu otázka, jestli jde vůbec o podstatný problém. Linus [si to nemyslí](#) <sup>(238)</sup> a poukazuje na to, že sdílení PID mezi jmennými prostory je podobné jako používání PID v zamykacích souborech sdílených po síti. Zamykání založené na PID nefunguje na souborech připojených přes NFS a rozhraní založená na PID nebudou fungovat mezi jmennými prostory PID. Linus to shrnul takto:

**Nechápu, jak tomu můžete říkat „chyba návrhu jmenných prostorů PID“, když to s těmi jmennými prostory PID nemá zjevně nic společného – naopak to má všechno společné s \*futexy\*, které bláznivě předpokládají, že jsou PID unikátní, a které na tom postavily část uživatelského rozhraní.**

Dalo by se argumentovat tím, že o konfliktu se jmennými prostory PID se vědělo už při začleňování futexů, a že se s tím dalo něco udělat už tenkrát. Ale to už teď nikomu nepomůže. A kromě toho tam jsou i jiné problémy než jen s futexy.

Jmenné prostory PID výrazně komplikují uživatelské API; redefinují základní hodnotu, která měla všeobecně chápaný význam od raných dnů unixu. Není tedy nijak divné, že se vynořily zajímavé otázky. Získávání solidních odpovědí na všetečné otázky ohledně API není tradičně silnou disciplínou vývoje Linuxu, ale třeba se to změni.

■

# Jaderné noviny – 14. 11. 2007

Aktuální verze jádra: 2.6.24-rc2. Stále aktuální -mm strom. Citáty týdne: Ingo Molnár, James Bottomley. Různá témata týkající se kvality jádra. Lepší proměnné pro jednotlivé procesory.

Robert Krátký

## Aktuální verze jádra: 2.6.24-rc2

*Aktuální předverze je i nadále 2.6.24-rc2.* Od vydání -rc si už do hlavního git repozitáře našlo cestu docela dost patchů; jde především o opravy, ale probíhá i práce na podpoře CIFS ACL a odstraňování několika zastaralých dokumentů. -rc3 by měla vyjít velmi brzy.

Aktuální verze -mm stromu je [2.6.24-rc2-mm1](#) <sup>(239)</sup> – první -mm od 2.6.23-mm1, která vyšla 11. října. Mezi nedávné změny patří aktualizace mapovače zařízení, velká aktualizace stromu ovladačů (kvůli které přestalo dost věcí fungovat), hodně aktualizací IDE, podpora dvousměrového SCSI, velká sada oprav SLUBu <sup>(240)</sup> a další patche z oblasti správy paměti [„mammary manglement“], podpora kvalifikací <sup>(241)</sup> na 64 bitech, několik vylepšení ext4 <sup>(242)</sup> a začlenění vývojového stromu PCI hotplug.

## Stále aktuální -mm strom

V rámci oznámení [2.6.24-rc2-mm1](#) <sup>(243)</sup> [Andrew Morton](#) <sup>(244)</sup> zmínil, že někteří lidé chtějí ještě aktuálnější jádro, než může nabídnout -mm, a proto vytvořil [-mm of the minute tree](#) <sup>(245)</sup>, který je aktualizován několikrát denně. **Pokusím se dávat pozor na to, aby ty patche šly aplikovat, ale na beton nebude možné je všechny zkompileovat a spustit.**

Strom bude exportován jako série patchů, takže abyste z toho dostali něco, co by bylo možné kompilovat, budete potřebovat Quilt. Dobře se bavte.

## Citáty týdne: Ingo Molnár, James Bottomley

**Jsem přesvědčen, že bychom měli o hodně kvalitnější jádro, kdyby existoval jeden centrální mailing list místo těch elitářských a rozdrobených konferencí. Každé téma by bylo globálně na očích a bylo by velmi snadné zainteresovat i další lidi z jiných subsystémů.**

– [Ingo Molnár](#) <sup>(246)</sup>

**Pokud to není nahlášeno na linux-scsi, je dost velká pravděpodobnost, že si takové zprávy o chybě nevšimneme. Skutečnost, že si někteří lidé všimnou bugů hlášených v [LKML](#) <sup>(247)</sup>, a přepošlou je do linux-scsi, je šťastná shoda náhod, na kterou by se nemělo spoléhat.**

**LKML je asi 10 – 207× frekventovanější než linux-scsi, ale je tam mnohem větší poměr odpadu k tématu. Když máme specializovaný mailing list, který sledují všichni experti, tak to v důsledku zvyšuje naši schopnost opravovat chyby.**

– [James Bottomley](#) <sup>(248)</sup>

## Různá témata týkající se kvality jádra

Diskuze o kvalitě jádra nejsou v linux-kernel žádnou novinkou. Je to téma, které se objevuje pravidelně, a to více než u jiných projektů svobodného softwaru. Velikost jádra, rychlost změn kódu a velká různorodost prostředí, kde je jádro nasazováno, jsou všechno faktory, které vedou k unikátním problémům; připočtete

skutečnost, že chyby v jádře mohou vést ke katastrofickým selháním, a máte materiál pro nekonečnou debatu.

Zatím poslední kolo začalo, když Natalie Protasevich, vývojářka ze společnosti Google, která pomáhá Andrew Mortonovi sledovat bugy, poslala [seznam několika desítek otevřených chyb](#) <sup>(249)</sup>, jež by si zasloužily další pozornost. Andrew [odpověděl](#) <sup>(250)</sup>, jak se na dění kolem těchto hlášených chyb dívá on; ve většině případů to bylo „žádná odezva vývojářů“: **Napočítal jsem asi sedm hlášení, kterými se někdo zabývá. Dvacetsedm je zatím ignorováno.**

Několik vývojářů reagovalo v tom smyslu, že je Andrew příliš přísný a ne vždy má v těchto věcech pravdu. Ať už má, nebo nemá pravdu, Andrewa tato diskuze zjevně vyprovokovala.

Své tvrzení [obhajoval](#) <sup>(251)</sup> tím, že vytáhl na světlo často opakovanou obavu o horšící se kvalitě jádra. To je podle něj záležitost, která si zaslouží okamžitou pozornost:

**Pokud se jádro \_opravdu\_ zhoršuje, pak to nebude jen tak jasné, dokud to nebude trvat několik let. A pak by už bylo práce tolik, že návrat na přijatelnou úroveň by znamenal opravdu velké úsilí. A jádru by pak také velmi dlouho trvalo, než by si napravilo pověst.**

Ale zhoršuje se opravdu kvalita jádra? Na to se odpovídá velmi těžko. Neexistuje žádný objektivní standard, podle kterého by šlo jádro posuzovat. Některé druhy problémů lze odhalit automatickým testováním, ale v jaderném prostoru se mnohé bugy projeví až při nasazení s určitými zátěžemi na konkrétní hardwarové kombinaci. Stoupající počet hlášení o chybách ještě nemusí znamenat horší kvalitu, když se zvyšuje počet uživatelů i velikost kódu.

Podobnou argumentaci použil [Ingo Molnár](#) <sup>(252)</sup>, který [poukázal na to](#) <sup>(253)</sup>, že méně hlášení o chybách nemusí značit zlepšování kvality. Místo toho by to mohlo znamenat, že testeři začínají být otráveni a vývojového procesu už se neúčastní – zhoršující se kvalita by tak vlastně mohla vést k tomu, že bude nahlášeno méně chyb. Proto Ingo připomíná, že musíme být vstřícnější k testerům, ale zároveň se i více snažit o opravdové měření kvality jádra:

**Snažil jsem se poukázat na to, že se musíme oprostít od našeho současného subjektivního přístupu k posuzování kvality a přinejmenším si uvědomit, jak z vrchu se díváme na QA. Jakmile budeme schopni změřit, jak špatně si vedeme, vývojáři jádra se přizpůsobí a napraví to. Používejme tedy více debugovacích nástrojů, jak statických, tak dynamických. Také bychom měli měřit, kolik je testerů, a musíme měřit, kolik uživatelů, kteří začali používat nějakou vlastnost hned po uvedení, s tím zase seklo, a proč.**

Obecně je pravda, že problémy, které lze změřit a vyčíslit, bývají vyřešeny rychleji a efektivněji. Klasický příklad je PowerTop, díky kterému jsou problémy se správou napájení jasně patrné. Jakmile měli vývojáři možnost vidět, kde je problém, a především, o kolik jejich opravy situaci zlepšily, tak velký počet chyb zmizel během chvíle. V tuto chvíli mají vývojáři mnoho možností, jak se snažit o zlepšení kvality jádra, ale nemají možnost zjistit, jestli jejich úsilí k něčemu vede, nebo ne. Když není k dispozici objektivní měření, jsou snahy o zlepšování jen tápáním ve tmě.

Jako příklad může posloužit diskuze o funkci "git bisect". Pokud se snažíte vyhledat příčinu regrese, ke které došlo mezi 2.6.23 a 2.6.24-rc1, dalo by se čekat, že budete muset prohlédnout několik tisíc patchů, abyste objevili ten, který potíže způsobil – což je úkol, který většina lidí považuje za poněkud děsivý. Půlení [bisection] testerovi pomůže provést binární hledání v určitém rozsahu patchů přičemž s každou rekompilací a restartem se polovina vyloučí. S pomocí "bisect" lze tedy regresi vystopovat relativně automatickým způsobem při provedení přibližně "pouze" desítky kompilací a restartů. Na konci procesu bude provinilý patch nezvratně identifikován.

Půlení funguje tak dobře, že vývojáři často po testerech požadují, aby vysledovali kořeny problému, který nahlásili. Někteří lidé to považují za způsob, který používají líní vývojáři ke shoení práce spojené s vyhledáním svých vlastních bugů na uživatele, kteří tou chybou trpí. Sestavit a otestovat desítku jader je na

testery trochu moc velký požadavek. Mark Lord například říká, že většinu bugů je poměrně snadné najít <sup>(254)</sup>, když se vývojář na kód skutečně podívá; celý proces půlení je často zbytečný.

Na druhou stranu, někteří vývojáři vidí v půlení mocný nástroj, který testerům usnadnil aktivně se na procesu podílet. David Miller <sup>(255)</sup> napsal <sup>(256)</sup>: **Podobně jako u Internetu i v tomto případě je takto strávený čas ku prospěchu věci, protože to znamená přesouvání práce na koncové body. Git bisect je ve skutečnosti skvělý příklad principu koncových bodů v akci při vývoji softwaru a QA. Je to výhra pro vývojáře i koncového uživatele, který čeká, až bude jeho bug opraven, protože ten, kdo bug nahlásil, má opravdu možnost se na tom aktivně podílet a pomoci s rychlejším vyřešením.**

Zpět k původnímu seznamu bugů: další věc, o které se mluvilo, je používání jiných konferencí kromě linux-kernel. Některé z chyb nebyly řešeny, protože nebyly hlášeny do konferencí zaměřených na daný subsystém. Jiné zase Andrew označil jako bez odpovědi, a přitom se o nich mluvilo (a někdy byly i vyřešeny) v těch oddělených konferencích. V obou případech za to může nedostatečná komunikace mezi konferencemi subsystémů a širší komunitou.

V reakci na to někteří vývojáři začali (opět) požadovat, aby se subsystémové konference používaly méně. Tvrdí, že když všichni pracujeme na stejném jádře, takže chceme všichni vědět, co se se ním děje. Debatování o subsystémech na oddělených konferencích povede k nižší kvalitě jádra. Ingo Molnár <sup>(257)</sup> to vyjádřil takto <sup>(258)</sup>:

**Nucenou izolací diskuzí ztrácíme mnohem více než získáváme tím, že do konference přijde méně mailů. Je MNOHEM snazší informace filtrovat (podle vláken, témat, lidí atd.), než je shromažďovat z více zdrojů. Znovu a znovu se ukazuje, že izolace diskuzí o jádře vede k problémům.**

Přesun diskuzí zpět do linux-kernel by však šlo prosadit jen těžko. Většina subsystémových konferencí má menší množství zpráv, přátelštější atmosféru a diskutuje se tam více k věci. Většina těch, kteří jsou členy takových konferencí, nebude mít pocit, že by jim přesun zpět do linux-kernel nějak usnadnil život. Nezbývá než doufat, že se více vývojářů přihlásí do obou a pohlídají, aby oběma směry proudily relevantní informace.

David Miller připomněl <sup>(259)</sup>, proč se některým chybám nedostává tolik pozornosti: vývojáři si musejí vybrat, kterými chybami se budou zabývat. Problémy, které postihují hodně uživatelů a které je snadné reprodukovat, daleko snadněji upoutají vývojářovu pozornost. Hlášení o chybách, které skončí na konci seznamu priorit („plívy“ [chaff]), mají tendenci tam zůstat dlouho. David však říká, že takový systém funguje poměrně dobře:

**Naštěstí je to tak, že když to ignorované hlášení není plíva, objeví se znovu (a znovu a znovu), což způsobí změnu priorit, protože nejenže už to není plíva, ale ještě je k tomu k dispozici mnohem více informací, takže je dvojnásob zajímavé se tím zabývat.**

Vzhledem k tomu, nakolik je nepravděpodobné, že by kdy bylo dost vývojářů na to, aby mohli zareagovat na každé hlášení o chybě, vězí hlavní problém ve stanovování priorit. Andrew Morton má jasnou představu o tom <sup>(260)</sup> které chyby by měly být řešeny nejdříve: regrese oproti předchozím verzím.

**Pokud bychom byli opravdu aktivní při likvidaci regresí, pak myslím, že bychom si mohli být jisti, že se kvalita jádra nezhoršuje. Pravděpodobně se bude spíš zlepšovat, protože také opravíme některé dlouhodobé chyby.**

Regresím se za poslední dva roky začalo věnovat mnohem více pozornosti. Jsou aktivněji sledovány a před vydáním nových verzí se kontrolují seznamy známých regresí. Největší potíž je podle Andrewa v tom, že regrese, které zůstanou i po vydání nové verze, často ze seznamu vypadnou. Kdyby se dával větší pozor i na tyto problémy, kvalita jádra by se časem zlepšovala.

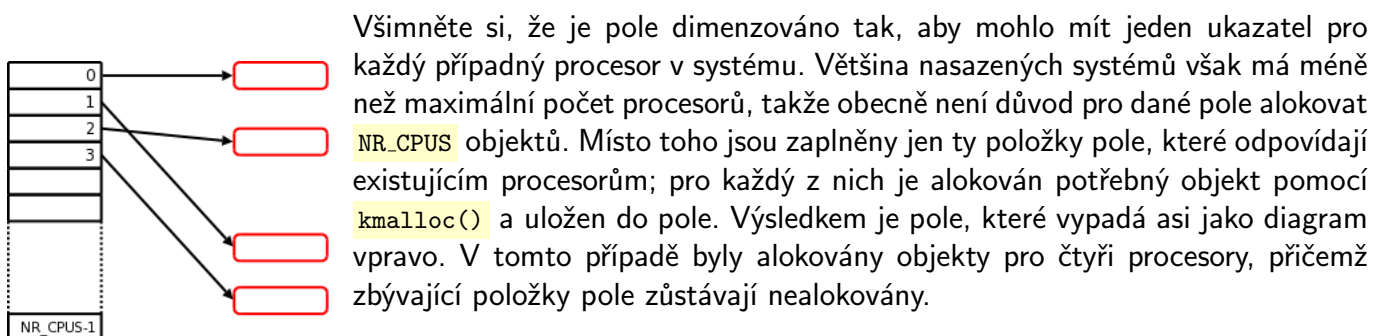
## Lepší proměnné pro jednotlivé procesory

Jednou z největších výhod víceprocesorových počítačů je skutečnost, že je hlavní paměť dostupná všem procesorům v systému. Tato schopnost sdílet data dává programátorům spoustu volnosti. Jednou z prvních věcí, kterou se tito programátoři naučí (nebo by se měli naučit), je však to, že sdílení dat mezi procesory by se mělo za každou cenu vyhýbat. Sdílení dat – obzvláště dat, která se mění – způsobuje všelijaké nepěknosti s keší a velmi snižuje výkon. Nedávno dokončená série [What every programmer should know about memory](#) <sup>(261)</sup> se těmto problémům věnuje velmi podrobně.

Během let se vývojáři jádra snažili co možná nejvíce využívat dat jednotlivých procesorů [per-CPU data], aby se minimalizovalo soupeření o paměť a související dopad na výkon. Jako jednoduchý příklad si můžete představit statistiky o diskových operacích, které udržuje bloková vrstva. Navyšování globálního čítače při každé diskové operaci by způsobilo, že by příslušný řádek keše [cache line] neustále lítal mezi procesory; operace s diskem jsou dost časté na to, aby byl dopad na výkon měřitelný. Takže si každý procesor udržuje svůj lokální čítač; nikdy tak nemusí s ostatními procesory soupeřit o jeho navýšení. Když je potřeba celkový počet, všechny čítače jednotlivých procesorů se sečtou. Vzhledem k tomu, že jsou čítače dotazovány daleko méně často než navyšovány, ukládání počtů pro jednotlivé procesory přináší výrazné výkonnostní zlepšení.

V současných jádrech je většina těchto proměnných u jednotlivých procesorů spravována pomocí pole ukazatelů. Takže například struktura `kmem_cache` (kterou implementuje [SLUB alokátor](#) <sup>(262)</sup>) obsahuje pole:

```
struct kmem_cache_cpu *cpu_slab[NR_CPUS];
```



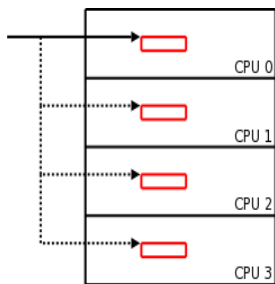
Letmý pohled na diagram však okamžitě ukazuje jeden potenciální problém tohoto schématu: každé z těchto polí pro jednotlivé procesory bude mít ke konci pravděpodobně nevyužitě místo. `NR_CPUS` je konstanta nastavovaná při konfiguraci; většina jader pro obecné využití (např. ta, která jsou dodávána s distribucemi) mívá `NR_CPUS` nastaveno tak vysoko, aby jádro fungovalo na většině nebo všech systémech, na které lze narazit. Zkrátka, je velmi pravděpodobné, že `NR_CPUS` bude o dost vyšší, než je skutečný počet přítomných procesorů.

Christoph Lameter si navíc povšiml, že problémů je ještě více; poslal tedy sérii patchů [implementujících nový alokátor pro jednotlivé procesory](#) <sup>(263)</sup>. Kromě plýtvání místem na konci polí jednotlivých procesorů řeší Christophův patch ještě následující problémy:

- Je-li jedno z těch polí obsaženo ve větší datové struktuře, může dojít k tomu, že oddělí ostatní proměnné v té struktuře, což by způsobilo zabránění více kešových linek než normálně.
- Každý procesor (většinou) využívá přesně jeden ukazatel z toho pole; tento ukazatel je v době, kdy je používán, uložen v datové keši procesoru. Řádky keše však uchovávají mnohem více než jeden ukazatel; v tomto případě jsou ve zbytku řádku skoro určitě ukazatele pro ostatní procesory. Nedostatkový prostor keše je tak plýtván na úplně zbytečná data.
- Přístup k objektu vyžaduje dvojí vyhledání ukazatele – jedno pro získání ukazatele na objekt z pole a druhé pro získání samotného objektu.

Christophovo řešení je v principu docela jednoduché: změníme všechna ta malá pole u jednotlivých procesorů na jedno velké. Při takovém schématu by byl pro každý procesor při inicializaci systému alokován

vyhrazený rozsah paměti. Ve virtuálním adresním prostoru jádra by všechny tyto rozsahy byly souvislé, takže mám-li ukazatel na oblast procesoru 0, chybí mi k oblasti kteréhokoliv dalšího procesoru přidání jen jediného ukazatele.



Když je alokován objekt procesoru, dostane každý procesor kopii získanou z jeho vlastní oblasti. Klíčové je, že offset do každé oblasti procesoru je stejný, takže adresa kteréhokoliv objektu procesoru se snadno vypočítá z adresy prvního objektu. Pole ukazatelů je tedy možno nahradit jediným ukazatelem na objekt v oblasti rezervované pro procesor 0. Výsledná organizace vypadá (s dostatečnou dávkou představitosti) asi jako diagram vpravo. Pro daný objekt existuje jen jeden ukazatel; všechny ostatní verze objektu se hledají přidáním konstantního offsetu. Rozhraní nového alokátoru je relativně přímočaré. Nová proměnná procesoru se vytvoří pomocí:

```
#include <linux/cpu_alloc.h>

void *per_cpu_var = CPU_ALLOC(type, gfp_flags);
```

Toto volání alokuje sadu proměnných udaného `type`, přičemž se pro ovládání toho, jak bude alokace provedena, použije běžné `gfp_flags`. Ukazatel na specifickou verzi proměnné některého procesoru lze získat takto:

```
void *CPU_PTR(per_cpu_var, unsigned int cpu);
void *THIS_CPU(per_cpu_var);
```

`THIS_CPU()` vrací, jak by se dalo předpokládat, ukazatel na verzi proměnné alokované pro aktuální procesor. K dispozici je makro `CPU_FREE()`, které systému vrací objekt daného procesoru. Christophův patch konvertuje všechny stávající uživatele rozhraní a pak staré rozhraní odstraňuje úplně.

Tento způsob má několik výhod. Každý přístup k proměnné některého procesoru znamená o jednu operaci s ukazateli méně. Pro všechny procesory se používá jeden ukazatel, což má za následek menší datové struktury a lepší využití řádků keše. Proměnné jednotlivých procesorů jsou v paměti seskupeny, což by mělo také vést k lepšímu využití keše. Získali jsme zpět všechnu paměť vyplývanou ve starých polích ukazatelů. Christoph také tvrdí, že tento mechanismus, díky kterému je snazší sledovat paměť jednotlivých procesorů, usnadňuje podporu výměny procesorů za chodu [CPU hotplugging].

O návrhu se zatím moc nediskutovalo. Objevilo se pár připomínek k používání VERZÁLEK pro pojmenování maker. Největší stížnost se však týká toho, že statické oblasti jednotlivých procesorů zaplácávají datový prostor jádra. Na některých architekturách je kvůli tomu jádro tak velké, že nemůže nabootovat, a na všech architekturách je to velká režié. Není jasné, jak bude tento problém řešen. Pokud se však řešení najde, je dost velká šance, že se nový kód dostane do hlavního jádra v rámci vývojového cyklu verze 2.6.25.

■

## Jaderné noviny – 20. 11. 2007

Aktuální verze jádra: 2.6.24-rc3. Citáty týdne: Greg Kroah-Hartman, Linus Torvalds. Systémové volání `sys_indirect()` pro nepřímé volání systémových volání. Jmenné prostory PID v jádře 2.6.24: API pro uživatelský prostor, interní API.

Robert Krátký

### Aktuální verze jádra: 2.6.24-rc3

Aktuální předverze je (20. 11. 2007) 2.6.24-rc3, vydaná <sup>(264)</sup> 16. listopadu. Kromě spousty oprav obsahuje podporu nových zařízení I/OAT a patch označující funkci **jmenných prostorů PID** <sup>(265)</sup> jako „experimentální“. Vizte **krátký changelog** <sup>(266)</sup> se seznamem patchů nebo **dlouhý** <sup>(267)</sup> se všemi podrobnostmi.

Od vydání -rc3 bylo zatím do hlavního git repozitáře začleněno jen několik oprav.

Aktuální vydání stabilní řady 2.6 je 2.6.23.8 ze 16. listopadu. Pár dní předtím Greg Kroah-Hartman <sup>(268)</sup> **poznámek** <sup>(269)</sup>:

Ok, trochu jsem se v poslední době se -stable loudal a neuvědomil jsem si, jak moc pozadu už jsem. Všichni posílali patche, což je skvělé, ale máme teď před sebou obrovské vydání, kde by jich bylo 114.

Aby usnadnil kontrolu kódu, rozdělil patche do několika částí, z nichž každá vyšla jako samostatná stabilní verze. Takže máme:

- 2.6.23.2 <sup>(270)</sup> (změny v jádře jádra),
- 2.6.23.3 <sup>(271)</sup> (opravy architektury),
- 2.6.23.4 <sup>(272)</sup> (síťování),
- 2.6.23.5 <sup>(273)</sup> (síťové ovladače),
- 2.6.23.6 <sup>(274)</sup> (ostatní ovladače),
- 2.6.23.7 <sup>(275)</sup> (souborové systémy) a
- 2.6.23.8 <sup>(276)</sup> (bezpečnostní opravy – ale i v ostatních najdete opravy týkající se bezpečnosti)

2.6.23.9, které obsahuje 29 patchů, je **teď připravováno** <sup>(277)</sup>.

Starší jádra: 2.6.22.13 <sup>(278)</sup> vyšlo také 16. listopadu a obsahovalo pouze bezpečnostní opravy. 2.6.22.14, ve kterém je dalších pár desítek oprav, se **teď připravuje** <sup>(279)</sup>.

Ještě starší jádra: 2.4.35.4 <sup>(280)</sup> vyšlo 17. listopadu s několika opravami. 2.4.36-pre2 <sup>(281)</sup> vyšlo s převážně těmi samými opravami.

### Citáty týdne: Greg Kroah-Hartman, Linus Torvalds

A co se týče 802.11n, tak už jsem *takhle* blízko k získání specifikací důležitého čipsetu, takže snad pro vás budeme mít zanedlouho práci. Teď kdyby si jen pospíšili ti právníci...

– Greg Kroah-Hartman <sup>(282)</sup> stupňuje napětí.

Vždycky jsem si o HIGHMEM myslel, že je to prostě nepoužitelné. Je to OK pro rozšíření 2 - 4 GB (tj. HIGHMEM4G, ne 64G) a asi je to i na hraně použitelnosti při 4 - 8 G, když si dáváte pozor.

Ale upřímně řečeno se odmítám starat o to, co se děje nad to. Pokud máte ve svém stroji 12 G (nebo, bůh chraň, ještě více) a neráčíte se obtěžovat upgradovat na 64bitový procesor, tak \*mně\* nestojí za to si kvůli tomu dělat vrásky.

– Linus Torvalds <sup>(283)</sup>

## Systémové volání `sys_indirect()` pro nepřímé volání systémových volání

Příprava uživatelských API je těžká. I když se rozhraní v době vytvoření zdá kompletní a dobře navržené, v budoucnu se často objeví nové požadavky, které může staré API jen těžko uspokojit. Unix například začal se systémovým voláním `wait()`. Přišly však komplexnější aplikace a začalo být potřeba čekat na konkrétní procesy, získat o končícím procesu více informací, čekat tak, aby nedošlo k bloku, atd. Takže kromě `wait()` teď máme ještě `waitid()`, `waitpid()` a `wait4()`. Protože staré verze systémových volání nemohou být (téměř) nikdy odstraněny, proměnlivé potřeby mají tendenci časem způsobovat přibývání nových volání.

Nedávno Ulrich Drepper<sup>(284)</sup> požadoval, aby bylo možné přidávat příznaky k systémovým voláním, která vytvářejí popisovače souborů, ale nemají parametr pro příznaky. Příkladem mohou být třeba `socket()` a `accept()`. Chování popisovačů souborů, které byly vytvořeny pomocí těchto systémových volání, je možné upravit zpětně (pomocí `fcntl()`), ale vždycky tam bude chvilka, během které bude popisovač existovat, aniž by bylo nastaveno požadované chování. Jedná-li se například o "close on exec" [zavřít při spuštění] a zároveň běží vícevláknový program, mohlo by jedno vlákno spustit nový program pomocí `exec()` ještě předtím, než by jiné stihlo nastavit příznak "close on exec". Výsledkem takového souběhu by byl děravý popisovač souborů, který by mohl znamenat bezpečnostní problém. Jediný způsob, jak tomuto konkrétnímu souběhu zabránit, je přinutit jádro, aby popisovače souborů vytvářelo už s nastavenými požadovanými příznaky.

Tradičně by tento problém řešilo přidání nového systémového volání; mohlo by se například přidat 4parametrové volání `socket4()`, které by mělo požadovaný parametr `flags`. Takové řešení však není optimální; jak bylo řečeno, vede to k neustále narůstajícímu počtu systémových volání. Bylo by tedy fajn najít jiné řešení. Ulrich si myslí, že se mu povedlo právě to, když přidal jediné systémové volání (`indirect()`), které funguje tak, že přidává dodatečné parametry k existujícím voláním.

První implementaci `sys_indirect()`<sup>(285)</sup> vytvořil Davide Libenzi v červenci. S tím kódem však Ulrich nebyl moc spokojen:

**Davideova předchozí implementace je podle mého názoru mnohem komplikovanější, než by bylo potřeba. Tento kód je, jak se můžete přesvědčit, triviální. Minulý týden jsem o tomto přístupu mluvil s Linusem a na chvilku jsme spolu dokonce v něčem souhlasili.**

Prototyp nového systémového volání vypadá takto:

```
int indirect(struct indirect_registers *regs,
            void *userparams,
            size_t paramslen,
            int flags);
```

Struktura `regs` obsahuje registry procesu, které se obvykle používají v systémových voláních; jinými slovy číslo systémového volání a jeho (normální) parametry. Dodatečné parametry, které mají být systémovému volání předány, jsou v `userparams` a jejich délka je `paramslen`. Parametr `flags` není v současné době využit; je tam kvůli případnému budoucímu rozšíření, protože není možné rozšířit `indirect()` sebou samým.

Struktura `task_struct` byla rozšířena o nové pole:

```
union indirect_params indirect_params;
```

Tato union má obsahovat pole pro každý druh parametru, který by mohl být k systémovému volání přidáván; v Ulrichově kódu vypadá takto:

```
union indirect_params {
    struct {
        int flags;
    } file_flags;
};
```



Může být tedy použita k předání parametru `flags` systémovým voláním, které se zabývají popisovači souborů.

Když je zavoláno `indirect()`, zkontroluje si, jestli je číslo požadovaného systémového volání na interním seznamu povolených. Pokud nebylo dané systémové volání označeno jako rozšiřitelné tímto způsobem, volání selže s `EINVAL`. V opačném případě jsou parametry překopírovány do struktury `task_struct` aktuálního procesu a volání je spuštěno běžným způsobem. Po dokončení volání je oblast `indirect_params` ve struktuře úlohy vynulována.

Jádro systémovému volání nedává nijak najevo, že bylo spuštěno prostřednictvím `indirect()`; jediným rozdílem v takovém případě je, že v `indirect_params` mohou být nenulové hodnoty. Takže mechanismus lze také chápat jako způsob, jak k systémovým voláním přidat parametry s výchozí hodnotou nula. Bez dalších úprav tedy není možné přidat systémovému volání parametr, u kterého by předání nuly mělo jiný význam než úplné vynechání parametru.

Pokud by se v budoucnu objevila potřeba přidat další parametr, může být velikost struktury `indirect_params` zvětšena. Za předpokladu, že si jádro udrží staré chování, když má nový parametr hodnotu nula, starší aplikace a knihovny budou fungovat jako dříve. Rozšířené systémové volání nepotřebuje (a nemůže) vědět, jestli je zvětšená struktura `indirect_params` používána, nebo ne.

Nabízí se také možnost využití tohoto mechanismu nad rámec rozšiřování systémových volání: vývojáři [sysletů](#) <sup>(286)</sup> v tom vidí způsob, jak specifikovat asynchronní chování. Aktuální patche implementující syslety jsou v podstatě jen obalová vrstva kolem systémových voláních, která říká, že jde o asynchronní volání (a co dělat s výsledky). Přidávat systémovým voláním dvě nepřímé vrstvy se nezdá jako optimální řešení, takže někteří vývojáři projeví zájem přidat místo toho informace o sysletech do `indirect()`. To je jeden ze zamýšlených účelů zatím nepoužívaného parametru `flags`.

Samozřejmě se nečeká, že by aplikace volaly `indirect()` přímo. Daleko pravděpodobnější se zdá, že by `indirect()` bylo zavrtáno hluboko v C knihovně, která by pro aplikace [exportovala přímočařejší rozhraní](#) <sup>(287)</sup>.

Zatímco některým vývojářům (včetně [Linuse](#) <sup>(288)</sup>, zjevně) se patch líbí, jiní moc nadšení nejsou. [David Miller](#) <sup>(289)</sup> si [nebral servítky](#) <sup>(290)</sup>: **Řekl bych, že tohle nepřímé systémové volání je nejošklivější rozhraní, které jsem kdy viděl navrhované pro jádro.**

Na H. Peter Anvina to také [dojem neudělalo](#) <sup>(291)</sup>:

**Myslím, že je to děsná berlička. Zase další multiplexor, kterým se v jádře tolik snažíme vyhýbat. A aby to bylo ještě horší, tak je to multiplexor, který vytváří zase další jednoúčelovou konvenci pro volání, zatímco bychom se měli snažit jaderné volací konvence co nejvíce sjednotit.**

Nebylo by tedy nijak překvapivé, kdyby se před začleněním do jádra toto rozhraní ještě vyvíjelo – je to nové a trochu zrádné API, kterému by diskuze jen prospěla. Protože však za touto prací stojí reálná potřeba, je pravděpodobné, že se `indirect()` nakonec v nějaké podobě v hlavním jádře objeví.

## Jmenné prostory PID v jádře 2.6.24

*Autoři: Pavel Emeljanov a Kir Kolyškin [Pavel Emelyanov a Kir Kolyshkin].*

Jednou z nových vlastností v jádře 2.6.24 je podpora [jmenných prostorů PID](#) <sup>(292)</sup>, kterou vyvinul tým [OpenVZ](#) <sup>(293)</sup> s pomocí IBM. Jmenný prostor PID umožňuje vytvářet sady úloh, z nichž každá vypadá – z pohledu ID procesů – jako samostatný systém. Jinými slovy, v různých jmenných prostorech můžete mít stejná ID.

Tato funkce je základním předpokladem pro migraci [kontejnerů](#) <sup>(294)</sup> mezi hostiteli; když máme jmenný prostor, můžeme jej přesunout na jiného hostitele, ale ponechat hodnoty PID – a to je nutné, protože úloze by se nemělo měnit PID. Bez této funkce by migrace s největší pravděpodobností selhala, jelikož na cílovém uzlu by mohly existovat procesy se stejnými ID, což by způsobilo konflikty při adresování úloh prostřednictvím jejich ID.

Jmenné prostory PID jsou hierarchické; při vytvoření nového jmenného prostoru PID uvidí všechny procesy v aktuálním jmenném prostoru úlohy v tom novém (tj. budou schopné je adresovat pomocí PID). Nicméně, úlohy v novém jmenném prostoru neuvidí ty v aktuálním. To znamená, že od té chvíle má každý proces více než jedno PID – jedno pro každý jmenný prostor.

## API pro uživatelský prostor

Nový jmenný prostor lze vytvořit zavoláním systémového volání `clone(2)` s nastaveným příznakem `CLONE_NEWPID`. Pak bývá vhodné změnit kořenový adresář a připojit novou instanci `/proc`<sup>(295)</sup>, aby fungovaly běžné utility, např. `ps`. Vzhledem k tomu, že předek zná PID svého potomka, je pro čekání na ukončení možné použít běžným způsobem `wait()`.

První úloha v novém jmenném prostoru bude mít PID `1`. Bude to tedy jeho `init` a všechny osiřelé úlohy se stanou jejími potomky. Na rozdíl od samostatných strojů může tento `init` skončit, což znamená ukončení celého jmenného prostoru.

Protože teď máme izolované skupiny úloh, měli bychom zařídit, aby `proc` ukazoval pouze ty PID, které jsou pro danou úlohu viditelné. Abychom toho dosáhli, musí být `procfs` připojen vícekrát – jednou pro každý jmenný prostor. Pak budou PID zobrazené v připojené instanci právě z toho jmenného prostoru, kde proběhlo připojení.

Například pokud uživatel vytvoří nový adresář `proc.2`, založí nový jmenný prostor PID a připojí tam `procfs`, uvidí PID, jak vypadají v rámci nového jmenného prostoru. Bude tam PID číslo `1`, což je `init` daného jmenného prostoru, a všechny další PID, které mohou být stejné jako PID v aktuálním jmenném prostoru, ale přitom mohou odkazovat na jinou úlohu.

Žádné další změny v uživatelském API nejsou potřeba. Úlohy mohou i nadále získávat své PID, PGID atd. s pomocí známých systémových volání. Mohou také pracovat se sezeními a skupinami. Úlohy mohou vytvářet vlákna a pracovat s `futexy`.

## Interní API

Všechna PID, které může úloha mít, jsou popsána v `struct pid`. Tato struktura obsahuje hodnotu ID, seznam úloh, které mají toto ID, referenční čítač a hashovanou položku seznamu, která je uložena v hashované tabulce pro rychlejší vyhledávání.

Pár slov o seznamech úloh. Každá úloha má v podstatě tři PID: ID procesu (PID), ID skupiny procesů (PGID) a ID sezení (SID). PGID a SID mohou být úlohami sdílené. Pokud například dvě nebo více úloh náleží do stejné skupiny, pak ID skupiny odkazuje na více než jednu úlohu.

Se jmennými prostory PID je tato struktura pružná. Každé PID může mít několik hodnot, přičemž každá bude platit v jiném jmenném prostoru. Takže úloha může mít PID 1024 v jednom jmenném prostoru a 256 v jiném. Struktura `struct pid` je tedy změněna.

Takhle vypadala `struct pid` před zavedením jmenných prostorů PID:

```
struct pid {
    atomic_t count;           /* referenční čítač */
    int nr;                   /* hodnota pid */
    struct hlist_node pid_chain; /* hash */
    struct hlist_head tasks[PIDTYPE_MAX]; /* seznam úloh */
    struct rcu_head rcu;      /* RCU pomocník */
};
```

A takhle vypadá teď:

```
struct upid {
    int nr;                   /* přesunuto ze struct pid */
};
```

```

struct pid_namespace *ns;          /* jmenný prostor, ve kterém
                                   je tato hodnota vidět */
struct hlist_node pid_chain;      /* přesunuto ze struct pid */
};

struct pid {
atomic_t count;
struct hlist_head tasks[PIDTYPE_MAX];
struct rcu_head rcu;
int level;                        /* počet upid */
struct upid numbers[0];
};

```

Jak vidíte, `struct upid` teď reprezentuje hodnotu PID. Pro převod `struct pid` na PID a naopak můžete použít sadu pomocníků, např. `task_pid_nr()`, `pid_nr_ns()`, `find_task_by_vpid()` atd.

Všechna tato volání mají v názvech informaci:

`...nr()`

Tato volání pracují s tzv. "globálními" PID. Globální PID jsou čísla, která jsou v celém systému unikátní, stejně jako byly dříve PID. Např. `pid_nr(pid)` vám řekne globální PID dané `struct pid`. Hodí se pouze v případě, že hodnota PID neopustí jádro. Například pokud nějaký kód potřebuje uložit PID a pak podle něj úlohu najít. V tomto případě je však lepší uložit přímý ukazatel na `struct pid`, protože globální PID budou používány pouze v jaderném logu.

`...vnr()`

Tito pomocníci pracují s "virtuálními" PID, tj. s ID, jak je vidí proces. Například `task_pid_vnr(tsk)` vám řekne PID úlohy, jak je daná úloha vidí (s `sys_getpid()`). Nezapomeňte, že pokud pracujete v jiném jmenném prostoru, bude vám to nejspíš k ničemu. Proto se používají při práci s aktuální úlohou, protože všechny úlohy vždy vidí své virtuální PID.

`...nr_ns()`

Tato volání pracují s PID, jak jsou viděna z určeného jmenného prostoru. Pokud chcete získat PID nějaké úlohy (například pro nahlášení do uživatelského prostoru a pozdější vyhledání dané úlohy), můžete zavolat `task_pid_nr_ns(tsk, current->nsproxy->pid_ns)`, což vrátí číslo, a pak úlohu najdete pomocí `find_task_by_pid_ns(pid, current->nsproxy->pid_ns)`. Používají se v systémových voláních, když je zdrojem PID uživatelský prostor. V takovém případě může jedna úloha odkazovat na jinou v jiném jmenném prostoru.

## Závěr

Popisované rozhraní bylo začleněno do jádra 2.6.24. Je však označeno jako „experimentální“, aby jej distributoři nenasazovali, dokud se nevyřeší zbývající problémy. V období do odstranění označení „experimentální“ se však v tomto API neočekávají žádné velké změny.

■

# Jaderné noviny – 28. 11. 2007

Aktuální verze jádra: 2.6.24-rc3. Aktuální verze jádra: 2.6.23.9.  
Citát týdne: Donnie Berkholz. Přiškrcení množství exportovaných symbolů. kmemcheck – kontrola neinicializované paměti.  
Aktualizace systémových volání: indirect(), timerfd(), hijack().

**Robert Krátký**

## Aktuální verze jádra: 2.6.24-rc3

Aktuální předverze je (k 28. 11. 2007) i nadále 2.6.24-rc3. Opravy dále velmi rychle proudí do hlavního git repozitáře; 2.6.24-rc4 musí být na spadnutí.

Aktuální verze -mm stromu je 2.6.24-rc3-mm2<sup>(296)</sup>. Mezi nedávné změny patří nové API timerfd (vizte níže), několik změn ovladačového jádra a aktualizovaná verze bezpečnostního modulu SMACK<sup>(297)</sup>. Aktuální stabilní verze řady 2.6 je 2.6.23.9, vydaná<sup>(298)</sup> 26. listopadu. V této verzi je pár desítek důležitých oprav. Starší jádra: 2.6.22.14<sup>(299)</sup> vyšlo 21. listopadu.

## Citát týdne: Donnie Berkholz

Linuxové jádro by potřebovalo, aby každou změnu doprovázela dokumentace – část patche by vždy měla mířit do adresáře [Documentation/](#).

– Donnie Berkholz<sup>(300)</sup>

## Přiškrcení množství exportovaných symbolů

Jaderný mechanismus pro nahrávání modulů neposkytuje modulům přístup ke všem částem jádra. Místo toho musí být každý symbol, který je určen pro použití natahovatelnými moduly, výslovně exportován prostřednictvím jedné z variant makra EXPORT\_SYMBOL(). Smyslem tohoto omezení je limitovat dosah modulů a poskytnout relativně dobře definované modulové API. V praxi však exportování moc omezováno není, takže jsou modulům k dispozici tisíce symbolů. Natahovatelné moduly mohou přistupovat k mnoha symbolům, jejichž užitečnost je zjevná (třeba printk() nebo kmalloc()), ale dostanou se i k obecným symbolům typu edd, tpm\_pm\_suspend(), vr41xx\_set\_irq\_trigger() nebo flexcop\_dump\_reg().

Některým vývojářům dělá přílišné exportování symbolů právem starosti. Chybně exportované symboly mohou vést autory modulů ke špatným rozhraním; například exportování sys\_open() vývojáře aktivně ponouká k otevírání souborů přímo v jádře, což téměř nikdy není dobrý nápad. Ale jakmile jsou symboly jednou exportovány, může být těžké je přestat exportovat<sup>(301)</sup>. Ačkoliv oficiální stanovisko zní, že interní jaderné API se může měnit kdykoliv, pravda je taková, že přinejmenším někteří vývojáři se zdráhají dělat problémy externím modulům, když se tomu dá vyhnout.

Jako aktuální příklad může posloužit init\_level4\_pgt – nízkoúrovňový symbol exportovaný pouze architekturou x86\_64. Aktuální -mm strom ten export odstraňuje, kvůli čemuž přestává fungovat proprietární modul NVIDIA. Andrew Morton<sup>(302)</sup> to odstranění popisuje<sup>(303)</sup> jako fikaný způsob, jak zúžit počet testerů, abychom nedostávali tolik hlášení o chybách.

Přestože mnoho vývojářů dává hlasitě najevo, že jim nezáleží na binárních modulech, i tak je dost velká pravděpodobnost, že tohle odstranění exportu (symbolu, který by opravdu neměl být globálně přístupný) kvůli zmiňovanému problému neprojde až do hlavního jádra.

Každopádně je už dlouho patrný zájem o pročištění modulárního API, i když se tomu zrovna moc lidí nevěnovalo. Čas od času někdo ukáže na některý ze snadných terčů: symbolů, které jsou exportovány jen

proto, aby bylo možné modularizovat další kousky kódu hlavního jádra. Například celá sada symbolů TCP stacku (věci jako `__tcp_put_md5sig_pool()`), které mají přesně jednoho uživatele: modul IPv6. Omezení těchto jednoúčelových exportů by mohlo výrazně zeštíhlit modulární API, aniž by bylo těžší hlavní jádro modularizovat.

Andi Kleen navrhl [patch implementující jmenný prostor pro modulové symboly](#) <sup>(304)</sup>, který by měl umožnit právě takové zeštíhlení. S tímto patchem mohou být symboly exportovány do specifických „jmenných prostorů“, které jsou dostupné pouze schváleným modulům. Termín „jmenný prostor“ v tomto případě zrovna moc nesedí; pořád bude jen jeden globální jmenný prostor, v rámci kterého budou muset být všechny exportované symboly unikátní. Tyto „jmenné prostory“ jsou spíše speciální zóny, jež obsahují symboly, které nejsou globálně přístupné. Fungují jako pouze-GPL exporty, které omezují dostupnost symbolů určité podskupině modulů.

Pro vytvoření omezeného exportu je běžná deklarace `EXPORT_SYMBOL()` změněna na:

```
EXPORT_SYMBOL_NS(namespace, symbol);
```

Kde `namespace` je název omezeného jmenného prostoru. Takže když se vrátíme k příkladu s TCP, tak Andiho patch obsahuje množství změn jako:

```
-EXPORT_SYMBOL(__tcp_put_md5sig_pool);
+EXPORT_SYMBOL_NS(tcp, __tcp_put_md5sig_pool);
```

Všimněte si, že není žádná `_GPL` verze; všechny symboly exportované do konkrétního jmenného prostoru jsou automaticky brány jako pouze-GPL.

Druhou částí rovnice je umožnění přístupu ke jmennému prostoru. K tomu slouží:

```
MODULE_NAMESPACE_ALLOW(namespace, module);
```

Taková deklarace (jež se musí objevit v modulu, který exportuje symboly do `namespace`) říká, že daný `module` může přistupovat k symbolům v určeném jmenném prostoru. Andiho patch vytváří tři jmenné prostory (`tcp`, `tcpcong` pro moduly kontrolující zahlcení a `udp`), čímž z globálního jmenného prostoru odstraňuje asi 30 symbolů.

Řada vývojářů tento patch přivítala, protože v něm vidí krok vpřed ve snaze o racionalizaci API natahovatelých modulů. Je to také bráno jako způsob, jak externím modulům zamezit používání symbolů, které by používat neměly. Kromě toho se tím snižuje počet rozhraní, která musí být udržována stabilní v situacích, kde nejsou změny přípustné (například u enterprise jader). A konečně, jmenné prostory symbolů umožňují trochu exporty organizovat a dokumentovat, kdo je má používat.

Ne všichni však souhlasí. Především [Rusty Russell](#) <sup>(305)</sup> má obavy, že patch věci zbytečně komplikuje a ztěžuje práci vývojářům externích modulů. [Rusty](#) <sup>(306)</sup>:

**Například když dáváš všechny `udp` funkce do jmenného prostoru „`udp`“, co nám to přinese? Co bude jednodušší spravovat? Všechny ty funkce začínají na „`udp_`“: dělá snad lidem potíže uhádnout, k čemu jsou určené? Pokud chceš opravdu omezit „veřejná rozhraní“, pak by bylo mnohem jednodušší přímo říct, co smějí externí moduly používat.**

**Herbert Xu to vidí podobně** <sup>(307)</sup>: **Ty symboly jsou exportovány proto, že je potřebují protokoly. Kdyby nebyly každému přístupné, bylo by těžké začít s psaním novým protokolů...**

**Takže přinejmenším s ohledem na ten síťový kód začínám spíš souhlasit s Rustym: je-li symbol využíván více než jedním modulem z jádra, je dost pravděpodobné, že budeme chtít, aby byl exportován pro všechny.**

Ačkoliv se zdá, že tyto hlasy jsou v menšině, mají docela velkou váhu. Nechce se mi tedy hádat, jestli bude patch začleněn, a pokud ano, tak v jaké podobě. Dříve nebo později se však něco v této oblasti stane, protože snaha o pročištění modulárního API jen tak nezmizí.

## kmemcheck – kontrola neinicializované paměti

Používání neinicializované paměti může vést k opravdu nepříjemným chybám. Když máte štěstí, tak jádro slítne a v tracebacku nechá rozluštitelnou stopu, která zaneřádila slab [slab poisoning] ( `0x5a5a5a5a` nebo něco podobného). Jindy se však stane nějaká zakuklenější nepravost a budete nuceni dlouho hledat jednu hloupou chybu. Nebylo by lepší, kdyby jádro prostě umělo detekovat odkazy na neinicializovanou paměť a nahlas v tu chvíli zařvat?

Patch `kmemcheck` <sup>(308)</sup>, který nedávno poslal Vegard Nossum, nabízí právě to – i když možná trochu těžkopádným způsobem. Jádro se zapnutou funkcí `kmemcheck` nebude asi vhodné pro produkční nasazení, ale mělo by dobře umět hledat paměť, která ještě nebyla nastavena na užitečnou hodnotu. `Kmemcheck` je poměrně jednoduchý patch; používá následující přístup:

- Každá alokace paměti je zachycena na úrovni alokátoru stránek. Pro každou alokaci je požadovaný řád navýšen o jeden, což velikost alokace zdvojnásobí. Dodatečné („stínové“) stránky jsou inicializovány nulami a drženy schované.
- Alokovaná paměť je vrácena volajícímu, ale bit „present“ [přítomna] je v tabulkách stránek smazán. Každý pokus o přístup k takové paměti tedy způsobí výpadek stránky.
- Jakmile dojde k výpadku, `kmemcheck` (pomocí nechutného kódu specifického pro jednotlivé architektury) zjistí přesnou adresu a velikost zkoušeného přístupu. Jde-li o zápis, budou odpovídající bajty ve stínové stránce nastaveny na `0xff` a operaci bude povoleno dokončit.
- Pro přístupy kvůli čtení jsou otestovány odpovídající bajty stínových stránek; jsou-li některé z nich nula, kód usoudí, že jde o snahu přečíst neinicializovaná data. Pak je vypsán traceback stacku, aby se přišlo na to, kde k tomu přístupu dochází.

Je zřejmé, že provozování `kmemcheck` bude mít negativní vliv na výkon. Výpadek stránky při každém přístupu do paměti slab prostě nemůže být rychlý. Zdvojnásobení velikosti každé alokace si také vezme svoje, včetně vlivu na keš, která bude muset pracovat s dvakrát tak velkým množstvím paměti. Ale když běží jádro v debugovacím režimu, tak se to dá vydržet.

Vegard poslal [ukázkový výstup](#) <sup>(309)</sup>, který ukazuje, jak systém reaguje na čtení z neinicializované paměti. Dá-li se tomu výstupu věřit, tak přístupu k neinicializované paměti není v současných jádrech nic exotického. Pokud se ukáže, že některé z odkazů, na které teď bylo upozorněno, jsou opravdu chyby, tak už se `kmemcheck` osvědčil, i kdyby si nikdy nenašel cestu do hlavního jádra.

## Aktualizace systémových volání

### indirect()

Když se minulý týden diskutovalo o navrhovaném systémovém volání `indirect()` <sup>(310)</sup>, stěžovali si někteří vývojáři, že je to opravdu nepěkné rozhraní. Od té doby se diskuze přenesla na rozhraní systémových volání obecně, ale moc nápadů, jak vylepšit `indirect()`, z toho nevzešlo.

Zatím vede alternativa, kterou navrhoval H. Peter Anvin: raději než k rozšiřování systémového volání používat `indirect()`, to je lepší prostě vytvořit nové volání s požadovanými parametry. Pak může být většinou stará implementace nahrazena jednoduchým kouskem kódu, který volá novou verzi s výchozími hodnotami u nových parametrů. Je to jednoduchý přístup, který snadno zachovává binární kompatibilitu, aniž by měl za běhu velkou režii. Protože čísel systémových volání je dostatek, mohlo by se takto postupovat ještě dlouho.

Správa zvyšujícího se počtu systémových volání však něco stojí; každé z těch systémových volání je součástí uživatelského API, které už musí fungovat navždy. `indirect()` nová systémová volání nepřidává. Za předpokladu, že budou parametry přidávány opatrně (s výchozími hodnotami nula), pak by mělo být celkem snadné se vyhnout potížím s API.

Existují však omezení toho, kolik parametrů lze k systémovým voláním přidat; na většině systémů se to pohybuje kolem šesti. Pokud chce nějaké systémové volání více parametrů, musí se pouštět do nepříjemností s nepřímými bloky. Vytváření nových systémových volání s dodatečnými parametry zvýší počet případů, ve kterých bude toto nepřímé zpracování parametrů potřeba. Takže přístup využívaný voláním `indirect()` bude nakonec stejně v nějaké podobě nasazován.

Klíčovým argumentem je však stále mechanismus `sysletů/threadletů` <sup>(311)</sup>. Možnost udělat z kteréhokoliv volání asynchronní volání je velmi lákavá, ale aby to šlo provést, jsou potřeba ještě nějaké další informace – přinejmenším místo, kam výsledek volání uložit. Asynchronní systémová volání v Linuxu jsou, z praktického hlediska, druh nepřímého volání. Navrhované rozhraní `indirect()` vypadá, že by mohlo být pro asynchronní volání dobře využito – i když přesné API ještě nebylo stanoveno.

V důsledku toho je pravděpodobné, že se `indirect()` v nějaké podobě nakonec do jádra dostane – přesto je však ještě dost času na to, aby někdo přišel s lepším nápadem.

## timerfd()

Když se tu naposled mluvilo o `timerfd()` <sup>(312)</sup>, bylo právě vypnuto v jádře 2.6.23 kvůli stížnostem na své rozhraní. Od té doby se toho moc nestalo, takže téměř určitě nebude ani v 2.6.24. Přesto se však na tomto systémovém volání pracovalo a byl představen návrh **nového API** <sup>(313)</sup>. Tato verze má tři systémová volání, z nichž první je `timerfd_create()`:

```
int timerfd_create(int clockid, int flags);
```

Parametr `clockid` systému říká, které hodiny se mají použít: `CLOCK_MONOTONIC` nebo `CLOCK_REALTIME`. Parametr `flags` byl přidán nedávno; v současné době se však nepoužívá a musí být nula. Je tam kvůli předpokladu, že někdo někdy bude chtít chování nějak upravit, takže je lepší se vyhnout potřebě nepřímé verze, dokud to jde. Návrhová hodnota z `timerfd_create()` je popisovač souboru, který je možné předat funkci `read()` nebo kterékoliv variantě `poll()`. Nejprve by však měl být časovač asi naprogramován pomocí:

```
int timerfd_settime(int fd,
                   int flags,
                   const struct itimerspec *timer,
                   struct itimerspec *old_timer);
```

`fd` je popisovač souboru získaný z `timerfd_create()`. Je-li časovač nastaven na absolutní čas, pak `flags` obsahuje `TFD_TIMER_ABSTIME`. A `timer` je doba vypršení časovače. Pokud `old_timer` není `NULL`, tak bude umístění, na které je ukazováno, nastaveno na předchozí hodnotu časovače. Hodnotu časovače je také možné získat pomocí:

```
int timerfd_gettime(int fd, struct itimerspec *timer);
```

Hodnota vrácená v `*timer` bude aktuální nastavení časovače přiřazeného k `fd`.

Tato nová verze API nebyla moc komentována, takže bude asi začleněno něco podobného. Normálně už by bylo příliš pozdě na přidání změny do 2.6.24, ale log patchů v `2.6.24-rc3-mm2` <sup>(314)</sup> říká „Asi 2.6.24?“. Takže jeden nikdy neví. Pokud to nebude začleněno brzy, skoro určitě se to dostane do 2.6.25.

## hijack()

Systémové volání `hijack()` <sup>(315)</sup> je i nadále vyvíjeno v rámci relativně tichých konferencí jaderných subsystemů. Toto volání se chová podobně jako `clone()` v tom smyslu, že vytvoří nový proces. Na rozdíl od `clone()` však `hijack()` způsobí, že nový proces sdílí zdroje s určeným třetím procesem, místo s předkem. Existuje hlavně proto, aby bylo snadné vstupovat do různých jmenných prostorů. Rozhraní `hijack()` je skoro beze změny:

```
int hijack(unsigned long clone_flags, int which, int id);
```

Zadaná hodnota `id` je interpretována podle `which`, který má tři možné hodnoty:

- `HIJACK_PID` říká, že `id` je ID procesu; nově vytvořený proces bude sdílet zdroje (včetně jmenných prostorů) s určeným procesem.
- `HIJACK_CG` říká, že `id` je otevřený popisovač souboru `tasks` v cílové **kontrolní skupině** <sup>(316)</sup>. V takovém případě jádro vyhledá proces v určené kontrolní skupině a ten je použit pro sdílení zdrojů a jmenných prostorů.
- `HIJACK_NS` je nejnovější možnost; podobně jako u `HIJACK_CG` jde o otevřený popisovač souboru označující kontrolní skupinu. V tomto případě však nový proces zdědí pouze danou kontrolní skupinu a přiřazené jmenné prostory. Tato verze je určena pro vstup do prázdné kontrolní skupiny (když tam nejsou žádné procesy, po kterých dědit).

Toto nové systémové volání vůbec nebylo představeno na linux-kernel; je dost možné, že v současné podobě tam svůj první výstup nepřežije. Příkladně bude asi požadována změna názvu (na něco trochu více popisného, co by, pokud možno, nezpůsobilo, že se uživatelé objeví na seznamu podezřelých osob výzvědných služeb). Je však zřejmé, že plná implementace **kontejnerů** <sup>(317)</sup> v Linuxu bude jednou potřebovat nějaké systémové volání `enter_container()`.

■



# Jaderné noviny – 5. 12. 2007

Aktuální verze jádra: 2.6.24-rc4. Citáty týdne: James Morris, Alan Cox. SEEK\_HOLE nebo FIEMAP? Návrat síťových kanálů.

Robert Krátký

## Aktuální verze jádra: 2.6.24-rc4

Aktuální předverze je (k 5. 12. 2007) 2.6.24-rc4, vydaná <sup>(318)</sup> 3. prosince. Linus Torvalds <sup>(319)</sup> k tomu řekl, že velikost patche je **trochu skličující**

– začleněno totiž bylo docela dost změn. Jsou to skoro všechno opravy, ale přibylo i počítání využití procesoru skupinami procesů. Vizte **krátký changelog** <sup>(320)</sup> nebo podrobnosti v **dlouhém** <sup>(321)</sup>.

Od vydání -rc4 se do hlavního repozitáře dostalo téměř 100 dalších patchů.

Aktuální verze -mm stromu je 2.6.24-rc4-mm1 <sup>(322)</sup>. Mezi nedávné změny patří poslední verze API timerfd <sup>(323)</sup>, nový paměťový řadič a reimplementovaný ovladač ramdisku.

## Citáty týdne: James Morris, Alan Cox

**Člověk by musel čekat s otevřenou pusou hodně dlouho, než by mu do ní přiletěl pečený holub.**

– James Morris <sup>(324)</sup>

**Chceš-li zjistit, co je potřeba, můžeš si představit jednoduchý uživatelský příklad – třeba systém, který tě ochrání před díly Erica S. Raymonda <sup>(325)</sup>. Nahraď matematickou analýzu a heuristiku nástrojem pro uživatelské prostředí, který vyhledá různé texty od ESR, a pak to navrhni pro něj, jestli ti to udělá radost.**

**SELinux asi zvládne všechnu tu špinavou práci sám, protože umí označit soubor jako eric.t a omezit k němu další přístup.**

– Alan Cox <sup>(326)</sup>

## SEEKHOLE nebo FIEMAP?

Rozptýlené soubory vypadají, že jsou větší než objem úložného prostoru, který je pro ně ve skutečnosti alokován. Běžný způsob vytvoření takových souborů je přesunout interní ukazatel pozice v souboru za jeho konec a pak tam zapsat nějaká data; unixové systémy tradičně nealokují diskové bloky pro část souboru za původním koncem, který byl přeskočen. Výsledkem je „díra“, neboli kus souboru, který logicky existuje, ale není nijak reprezentován na disku. Operace čtení na takové díře uspěje, ale vrácená data budou nuly. Průměrně chytré komprimační a archivační utility díry v souborech rozpoznají; tyto díry nejsou ve výsledných archívech uloženy a při obnovení souboru z archívu nebudou zaplněny.

Postup rozpoznávání děr je poměrně primitivní: v podstatě jediný způsob, jak to udělat přenositelně, je hledat bloky zaplněné nulami. Ta technika sice funguje, ale vyžaduje procházení dat, aby se získaly informace, které už nižší úroveň systému znají. Zdá se, že by měl existovat šikovnější způsob.

Přibližně před dvěma lety navrhli vývojáři Solarisu a ZFS **rozšíření lseek()** <sup>(327)</sup>, které by aplikacím umožnilo hledat díry v rozptýlených souborech efektivněji. Toto rozšíření přidává dvě nové možnosti "odkud":

- **SEEK\_HOLE** umísťuje popisovač souboru na začátek první díry, která se nachází za daným offsetem. Pro účely této operace je „díra“ definována jako oblast samých nul libovolné délky, ale od systému se nevyžaduje, aby všechny díry detekoval. Takže v praxi budou krátké rozsahy nul přeskočeny, stejně jako (s největší pravděpodobností) velké (víceblokové) rozsahy, které už byly zapsány na disk.

- `SEEK_DATA` posouvá na začátek další oblasti (po daném offsetu), která není díra.

Tato funkčnost už je nějakou chvíli součástí Solarisu; vývojáři Solarisu by však chtěli, aby se rozšířila i jinde a stalo se z ní něco více než jen rozšíření Solarisu. Josef Bacík proto nedávno navrhl **implementaci** <sup>(328)</sup> tohoto rozšíření pro Linux. Z interního hlediska přidává nového člena do struktury `file_operations` (`seek_hole_data()`), který by měl souborovým systémům umožnit novou operaci efektivně implementovat. Dalo by se namítnout, že pokud někdo chce oddělit v souboru díry od dat, může to už udělat pomocí `ioctl()` příkazu `FIBMAP`. To je sice pravda, ale `FIBMAP` není efektivní způsob, jak získávat tento druh informací – zvláště ne na souborových systémech, které podporují extenty. Volání `FIBMAP` vrátí mapovací informace o jediném bloku; mapování velkého souboru by si mohlo vyžádat miliony volání, a přitom už by souborový systém měl (opět) vědět, jak takové informace poskytnout mnohem přímočařejším způsobem. Přesto to však vypadá, že se tento patch do jádra nedostane. API není moc oblíbené, protože je považováno za ošklivé, a navíc mění sémantiku volání `lseek()`. Ale především by mohlo být zajímavé se dozvědět o reprezentaci souboru více než jen to, kde jsou díry. A ukazuje se, že už byl navržen `ioctl()` příkaz, který umí všechny tyto informace poskytnout. Jde o rozhraní `FIEMAP` <sup>(329)</sup>, které v říjnu specifikoval Andreas Dilger.

Volání `FIEMAP` bere jako parametr následující strukturu:

```
struct fiemap {
    __u64  fm_start;    /* logický offset počátečního bajtu */
    __u64  fm_length;  /* logická délka mapy (dovnitř/ven) */
    __u32  fm_flags;   /* FIEMAP_FLAG_* příznaky pro žádost (dovnitř/ven) */
    __u32  fm_extent_count; /* počet extentů v fm_extents (dovnitř/ven) */
    __u64  fm_end_offset; /* konec mapování v posl. ioctl */
    struct fiemap_extent  fm_extents[0];
};
```

Aplikace, která se chce něco dozvědět o tom, jak je soubor uložen, nastaví počáteční offset v `fm_start` a délku oblasti, o kterou se jedná, v `fm_length`. Obsahuje-li `fm_flags` `FIEMAP_FLAG_NUM_EXTENTS`, systémové volání prostě nastaví `fm_extent_count` na počet extentů použitých na uložení specifikovaného rozsahu bajtů a ukončí se. V této podobě může být `FIEMAP` použit k určení, jak moc je soubor na disku fragmentovaný. Pokud chce aplikace více informací, alokuje se dost místa pro ještě jednu strukturu `fm_extents`:

```
struct fiemap_extent {
    __u64 fe_offset; /* offset počátku extentu v bajtech */
    __u64 fe_length; /* délka extentu v bajtech */
    __u32 fe_flags; /* vrácené příznaky FIEMAP_EXTENT_* extentu */
    __u32 fe_lun; /* logické číslo zařízení extentu (začínající 0)*/
};
```

V tomto případě by `fm_extent_count` mělo být nastaveno na počet těchto struktur, než bude zavoláno `FIEMAP`. Po ukončení budou tyto struktury (tolik, kolik udává vrácená hodnota `fm_extent_count`) vyplněny informacemi o vlastních extentech souboru; `fe_offset` říká, kde (na disku) extent začíná, a `fe_length` je velikost extentu. V poli `fe_flags` se může objevit hned několik hodnot:

- `FIEMAP_EXTENT_HOLE` říká, že v této oblasti souboru nejsou žádná data – je to díra.
- `FIEMAP_EXTENT_UNWRITTEN` říká, že místo bylo alokováno na disku, ale nic do něj nebylo zapsáno. Takto označeno bude místo, které bylo prealokováno pomocí `fallocate()` <sup>(330)</sup>.
- `FIEMAP_EXTENT_UNMAPPED` naopak označuje extent, kam nějaká aplikace zapsala data, ale nebylo pro něj alokováno žádné místo na disku.
- `FIEMAP_EXTENT_DELALLOC` značí, že je prováděna opožděná alokace; tento příznak automaticky znamená i `FIEMAP_EXTENT_UNMAPPED`.

- `FIEMAP_EXTENT_SECONDARY` dává najevo, že data pro tento segment jsou umístěna někde jinde; s tímto příznakem se setkáte na souborových systémech spravovaných nějakým hierarchickým způsobem. Tento příznak také předznamenává `FIEMAP_EXTENT_UNMAPPED`.
- `FIEMAP_EXTENT_NO_DIRECT` říká, že k datům nelze přistupovat přímo – nejprve je nutné je zpracovat (například dekomprimace nebo rozšifrování).
- `FIEMAP_EXTENT_LAST` značí konečný extent souboru.
- `FIEMAP_EXTENT_EOF` říká, že požadovaný rozsah jde až za konec souboru.
- `FIEMAP_EXTENT_ERROR` značí extent, u kterého došlo k nějaké chybě; pole `fe_offset` bude v tomto případě obsahovat číslo chyby.
- `FIEMAP_EXTENT_UNKNOWN` říká, že data existují, ale neví se kde. Tento příznak by asi popisoval většinu prostoru s mými osobními soubory, i když není jasné, jak by se to jádro domáкло.

Jak je vidět, toto nové volání poskytuje opravdu bohaté informace, včetně údajů o tom, jak je soubor na disku rozdělen, alokačních strategiích, a dokonce i rozhodnutích hierarchických ukládacích enginů. Implementace už existuje pro souborový systém `ext4`<sup>(331)</sup>. Kód zatím nebyl navržen k začlenění do hlavního jádra, ale bylo by překvapivé, kdyby k tomu co nevidět nedošlo. Až to přijde, bude moci knihovna C implementovat `SEEK_HOLE` a `SEEK_DATA` v uživatelském prostoru, pokud o to bude zájem.

## Návrat síťových kanálů

Koncept síťových kanálů<sup>(332)</sup> poprvé přednesl Van Jacobson<sup>(333)</sup> téměř před dvěma lety na konferenci `linux.conf.au 2006`. Tato myšlenka slibuje zlepšení výkonu síťování přesunutím zpracovávání síťových dat co nejbližší koncovému bodu – nejlépe až do uživatelského prostoru. Kanálová schémata chtějí jádro vyšoupnout ze zpracovávání paketů a udržet to na jediném místě (na stejném procesoru). Díky tomu by se měly minimalizovat nepodařené pokusy o využití keše [cache misses], přepínání kontextů a další aktivity, které zhoršují výkon. Kanály však v reálném světě tvrdě narazily; když se začnou brát v potaz potřeby jako filtrování paketů, překlady adres a tak podobně, začne být těžké zachovat jednoduchost, na které výkon kanálů závisí. Takže o dva roky později stále neexistuje implementace kanálů, která by měla aspoň šanci na začlenění do hlavního jádra.

To však neznamená, že by se v té oblasti nic nedělo. Evgenij Poljakov [Evgeniy Polyakov], hacker, který se nikdy nenechá odradit (vizte rozhraní `kevent`<sup>(334)</sup>), dále vyvíjí patche se svou implementací kanálů; [22. verze](#)<sup>(335)</sup> vyšla 4. prosince.

Tato verze patche má dobře definovanou interní strukturu, která jadernému kódu umožňuje se na kanály napojit [hook]. Nejlépe vyvinutý režim je však ten, který pouze přenáší pakety z a do uživatelského prostředí. K tomu účelu přibýlo nové systémové volání:

```
int netchannel_control(struct unetchannel_control *ctl);
```

Kompletní obsah struktury `unetchannel_control` najdete v patchi. Ta důležitější pole jsou:

- `cmd` popisuje akci, kterou si volající kód přeje spustit. Na rozdíl od předchozích verzí patche je teď podporována jen jedna akce: `NETCHANNEL_CREATE`, která vytváří nový kanál.
- `type` je typ kanálu, který se má vytvořit. V současné době je implementován pouze typ `NETCHANNEL_COPY_USER`, který kopíruje pakety z a do uživatelského prostoru.
- `unc.data` popisuje kanál, který má být vytvořen: obsahuje zdrojové a cílové adresy a porty a číslo protokolu.

Jakmile je kanál vytvořen, přidá se do vyhledávacího stromu, který je orientován na bleskově rychlé hledání. V kódu pro příjem paketů je nový háček [hook], který v tom stromu vyhledává každý příchozí paket; pakety, které tam nejsou nalezeny, jsou normálně zpracovány jaderným síťovým stackem. Ale všechny pakety, jejichž adresy, porty a protokol jsou ve stromu objeveny, jsou odkloněny na kód kanálů, ještě než se vůbec dostanou do fronty síťového stacku.

Posledním kouskem (na přijímací straně) je jednoduchá implementace `read()`. Proces, který si přeje přijmout paket ze síťového kanálu, musí pouze přečíst příslušný popisovač souboru a další dostupný paket bude nakopírován do poskytnutého bufferu. Bylo by samozřejmě fajn se zbavit toho kopírování, ale to není vůbec lehké: paket musí být přijat, dříve než je znám jeho cíl. Existují síťové adaptéry, které umějí směřovat pakety podle informací v jejich hlavičce, ale stávající netfilter nemá potřebná vylepšení API pro ovladače, aby mohl tuto možnost využít pro provoz bez kopírování.

Podobné je to s operací `write()`, která způsobí, že bude příslušný paket nakopírován do jádra a nacpán na poměrně nízké úrovni do síťového stacku. Ani v případě zápisu to zatím nejde bez kopírování.

Evgenij však nad postupem, který by nevyužíval kopírování, zjevně přemýšlí. Pravděpodobně by použil [síťový alokátor](#)<sup>(336)</sup>. I bez toho to však vypadá, že jsou kanály velmi rychlé, když se použijí s jeho [síťovým stackem pro uživatelský prostor](#)<sup>(337)</sup>. Některé představené [výsledky benchmarků](#)<sup>(338)</sup> ukazují výrazné zlepšení oproti běžnému linuxovému síťovému stacku – třikrát více než šířka pásma při použití třetiny výkonu procesoru, když jsou přenášeny malé pakety. U větších paketů (4096 bajtů) výkonnostní zlepšení v podstatě mizí – hlavní příčinou je nejspíše režie způsobená kopírováním paketů z a do jádra.

Zlepšení výkonu u malých paketů je vítané: existuje hodně aplikací, včetně profesionálního finančnictví, které vyžadují velká množství malých přenosů. Přidání podpory provozu bez kopírování by mohlo zlepšit i výkon u velkých paketů. Hlavní zkouška však bude přidání všech funkcí, které současní uživatelé od síťování očekávají – většina zatím v implementaci kanálů chybí. V kódu už jsou háčky určené pro přidání zpracovávání jednotlivých paketů; mohly by být použity pro filtrování, překlady adres, kontrolu provozu nebo kteroukoliv jinou požadovanou funkci. Zbývá zjistit, jestli bude možné tyto háčky využít, aniž by se přišlo o výkonnostní náskok kanálů. Ale Evgenij se asi nevzdá, dokud na to nebude mít odpověď.

■

## Odkazy

### Distribuční novinky – 48/2007

- [1] <https://help.launchpad.net/PPAQuickStart>
- [2] <http://www.abclinuxu.cz/zpravicky/launchpad-personal-package-archive-v-provozu>
- [3] <http://www.abclinuxu.cz/zpravicky/suse-linux-enterprise-real-time-10>
- [4] <http://www.ubuntume.com/newspage:710release>
- [5] <http://comunidade.caixamagica.pt/index.php?article=4004&visual=2>
- [6] [http://sourceforge.net/project/shownotes.php?release\\_id=558489](http://sourceforge.net/project/shownotes.php?release_id=558489)
- [7] <http://linux.startcom.org/?app=14&rel=28>
- [8] <http://shiftlinux.net/news/item/shift-linux-05-final-released>
- [9] <http://damnsmalllinux.org/cgi-bin/forums/ikonboard.cgi?act=ST;f=36;t=19416>
- [10] [http://www.tapioneer.com/press/112707\\_Rel.htm](http://www.tapioneer.com/press/112707_Rel.htm)
- [11] <http://www.turbolinux.co.jp/cgi-bin/notify/index.cgi?date2=20071029110826&mode=syosai>
- [12] <http://lists.freebsd.org/pipermail/freebsd-stable/2007-November/038670.html>
- [13] <http://www.asplinux.ru/company/pr/2007/11-28.html>
- [14] <http://www.abclinuxu.cz/zpravicky/google-highly-open-participation-contest>
- [15] <http://code.google.com/opensource/ghop/2007-8/>
- [16] <http://www.abclinuxu.cz/zpravicky/olpc-zazalovano-kvuli-poruseni-patentu>
- [17] <http://www.abclinuxu.cz/slovník/olpc>
- [18] [http://www.lancorltd.com/images/keyboard\\_002.jpg](http://www.lancorltd.com/images/keyboard_002.jpg)
- [19] [http://wiki.laptop.org/go/OLPC\\_Keyboard\\_layouts](http://wiki.laptop.org/go/OLPC_Keyboard_layouts)
- [20] [http://www.boston.com/business/technology/articles/2007/12/01/one\\_laptop\\_per\\_child\\_orders\\_surge/](http://www.boston.com/business/technology/articles/2007/12/01/one_laptop_per_child_orders_surge/)
- [21] <http://www.abclinuxu.cz/slovník/plasma>
- [22] <http://www.abclinuxu.cz/zpravicky/kde-4-odloženo-na-leden-2008>
- [23] [http://techbase.kde.org/Schedules/KDE4/4.0\\_Release.Beta.Goals](http://techbase.kde.org/Schedules/KDE4/4.0_Release.Beta.Goals)
- [24] <http://dot.kde.org/1191409937/>
- [25] <http://www.abclinuxu.cz/kdo-je/aaron-seigo>
- [26] <http://www.abclinuxu.cz/zpravicky/odf-se-tesi-uspechu-v-dalsich-zemich-sveta>
- [27] <http://lists.oasis-open.org/archives/office/200712/msg00000.html>
- [28] <http://www.consortiuminfo.org/standardsblog/article.php?story=20071125145159900>
- [29] <http://www.truebsd.org/en/releases/2.0rc1>
- [30] <http://www.elivecd.org/Main/News/74>
- [31] <https://lists.ubuntu.com/archives/ubuntu-devel-announce/2007-December/000357.html>
- [32] <http://www.ubuntu.com/testing/hardy/alpha1>
- [33] <https://wiki.kubuntu.org/HardyHeron/Alpha1/Kubuntu>

### Distribuční novinky – 49/2007

- [34] <http://www.abclinuxu.cz/zpravicky/systemrescuecd-0.4.2>
- [35] <http://www.abclinuxu.cz/zpravicky/freebsd-7.0-beta4>
- [36] <http://beta.vinelinux.org/>
- [37] <http://mail-index.netbsd.org/netbsd-announce/2007/12/07/0000.html>
- [38] <http://yui.mine.nu/berry/changelog.php>
- [39] <http://www.biglinux.com.br/forum/viewtopic.php?t=3705>
- [40] <http://lists.debian.org/debian-devel-announce/2007/12/msg00003.html>
- [41] <http://www.mepis.org/node/14182>
- [42] [http://up2date.astaro.com/2007/12/up2date\\_and\\_iso\\_asg\\_7100.html](http://up2date.astaro.com/2007/12/up2date_and_iso_asg_7100.html)
- [43] <http://www.engardelinux.org/modules/index/releases/3.0.18.cgi>
- [44] <http://www.vectorlinux.com/article.php?sid=27>
- [45] <ftp://ftp.dyne.org/dynebolic/releases/latest/README.txt>
- [46] <http://www.abclinuxu.cz/zpravicky/damn-small-linux-4.1>
- [47] <http://www.abclinuxu.cz/zpravicky/centos-5.1>
- [48] <http://www.abclinuxu.cz/clanky/ruzne/distribucni-novinky-45-2007#red-hat-enterprise-linux-5.1>
- [49] <http://www.busybox.net>
- [50] <http://www.abclinuxu.cz/slovník/gnu-gpl>
- [51] <http://www.abclinuxu.cz/zpravicky/busybox-dalsim-zalovanym-je-verzion>
- [52] <http://www.abclinuxu.cz/zpravicky/sun-zavede-system-odmen-pro-oss-vyvojare>
- [53] <http://www.sun.com/opensource/awards>
- [54] <http://openjdk.java.net/>
- [55] <http://hardware.slashdot.org/article.pl?sid=07/12/09/0651200>
- [56] <http://alienghic.livejournal.com/382903.html>
- [57] <http://www.abclinuxu.cz/zpravicky/obrat-ooxml-by-nezustalo-pod-spravou-iso>
- [58] <http://www.nooxml.org/brm>

- [59] [http://trinityhome.org/Home/blog.php?blog\\_cat\\_id=2&b\\_node=2](http://trinityhome.org/Home/blog.php?blog_cat_id=2&b_node=2)
- [60] [http://trinityhome.org/Home/blog\\_comments.php?front\\_id=15&blog\\_id=228](http://trinityhome.org/Home/blog_comments.php?front_id=15&blog_id=228)
- [61] <http://www.abclinuxu.cz/zpravicky/opensuse-11-alfa0>
- [62] [http://en.opensuse.org/Bugs:Most\\_Annoying\\_Bugs\\_11.0\\_dev](http://en.opensuse.org/Bugs:Most_Annoying_Bugs_11.0_dev)
- [63] <http://www.frugalware.org/>
- [64] <http://www.frugalware.org/news/79>
- [65] <http://geubuntu.intilinux.com/Home.html>
- [66] <http://geubuntu.intilinux.com/Screens.html>

## Distribuční novinky – 50/2007

- [67] <http://frenzy.org.ua/>
- [68] [http://frenzy.org.ua/eforum.shtml?action=thread\\_show!11974580022803&section=001&thread=1197436216](http://frenzy.org.ua/eforum.shtml?action=thread_show!11974580022803&section=001&thread=1197436216)
- [69] <http://damnsmalllinux.org/cgi-bin/forums/ikonboard.cgi?;act=ST;f=36;t=19505>
- [70] [http://www.goblinx.com.br/en/index\\_news.htm](http://www.goblinx.com.br/en/index_news.htm)
- [71] <http://www.elivecd.org/Main/News/75>
- [72] [http://www.litrixlinux.org/site1/index.php?option=com\\_content&task=view&id=49&Itemid=27](http://www.litrixlinux.org/site1/index.php?option=com_content&task=view&id=49&Itemid=27)
- [73] <http://lliurex.net/home/es/node/2260>
- [74] <http://www.abclinuxu.cz/zpravicky/google-by-htel-konkurovat-wikipedii>
- [75] <http://www.abclinuxu.cz/zpravicky/canonical-vydal-vcs-bazaar-1.0>
- [76] <http://www.abclinuxu.cz/slovník/gui>
- [77] <http://www.abclinuxu.cz/slovník/gnu-gpl>
- [78] <http://www.heise.de/english/newsticker/news/100520>
- [79] <http://www.abclinuxu.cz/slovník/odf>
- [80] <http://www.ososs.nl/>
- [81] <http://wiki.eeuser.com/>
- [82] <http://www.abclinuxu.cz/zpravicky/eeexubuntu-upravene-ubuntu-pro-asus-eee-pc>
- [83] <http://www.abclinuxu.cz/zpravicky/eee-pc-levny-linuxovy-notebook-od-asusu>
- [84] <http://www.abclinuxu.cz/slovník/alsa>
- [85] <http://www.parsix.org/>
- [86] <http://xfardic.sf.net/>
- [87] <http://www.parsix.org/html/News-article-sid-63.html>
- [88] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-4.-7.-2007>
- [89] <http://crux.nu>
- [90] <http://crux.nu/portdb/>
- [91] <http://crux.nu/Main/ReleaseNotes2-4>
- [92] [http://wiki.mandriva.com/en/2008.1\\_Alpha\\_1](http://wiki.mandriva.com/en/2008.1_Alpha_1)
- [93] <http://www.pulseaudio.org/>
- [94] <http://www.pulseaudio.org/wiki/AboutPulseAudio#Features>
- [95] [http://wiki.mandriva.com/en/2008.1\\_Development](http://wiki.mandriva.com/en/2008.1_Development)

## Distribuční novinky – 51/2007

- [96] <http://www.abclinuxu.cz/zpravicky/spor-busybox-vs-xterasys-vyresen-mimosoudne>
- [97] <https://lists.ubuntu.com/archives/ubuntu-devel-announce/2007-December/000362.html>
- [98] <http://www.abclinuxu.cz/zpravicky/centos-4.6>
- [99] <http://www.abclinuxu.cz/zpravicky/damn-small-linux-4.2>
- [100] <http://www.musix.org.ar/wiki/index.php?title=Musix1.0r3test3-announce>
- [101] [http://www.goblinx.com.br/en/index\\_news.htm](http://www.goblinx.com.br/en/index_news.htm)
- [102] <http://support.zenwalk.org/index.php?topic=13033.0>
- [103] <http://lists.crux.nu/pipermail/crux/2007-December/008026.html>
- [104] <http://sidux.com/Article376.html>
- [105] <http://www.linpus.com/xampp/webmaster/Products/Linux9.4Lite.htm>
- [106] <http://www.abclinuxu.cz/slovník/classmate-pc>
- [107] <http://www.sabayonlinux.org/forum/viewtopic.php?f=60&t=11982>
- [108] [http://www.freenas.org/index.php?option=com\\_content&task=view&id=38&Itemid=24](http://www.freenas.org/index.php?option=com_content&task=view&id=38&Itemid=24)
- [109] <http://sam.hipsurfer.com/news.php?readmore=13>
- [110] <http://www.nnlinux.com/nieuws.html>
- [111] <http://vinelinux.org/whatsnew.html#20071225>
- [112] <http://www.protocolfreedom.org/>
- [113] <http://www.abclinuxu.cz/zpravicky/microsoft-preda-vyvojarum-samby-dokumentaci>
- [114] <http://www.reuters.com/articlePrint?articleId=USL207546720071220>
- [115] <http://www.abclinuxu.cz/slovník/nda>
- [116] <http://www.abclinuxu.cz/zpravicky/norsko-voli-otevrene-formaty-html-odf-a-pdf>
- [117] <http://www.abclinuxu.cz/slovník/odf>

- [118] <http://www.fsfeurope.org/news/2007/news-20071221-01.cs.html>
- [119] <http://www.mepis.org/>
- [120] <http://www.mepis.org/node/14187>
- [121] <http://vectorlinux.com/article.php?sid=40>
- [122] <http://www.abclinuxu.cz/zpravicky/netbsd-4.0>
- [123] <http://mail-index.netbsd.org/netbsd-announce/2007/12/19/0000.html>
- [124] <http://wizard.ae.krakow.pl/mike/>

## Svatá trojice vs. balíčkovací systém

- [125] <http://www.abclinuxu.cz/slovník/balicek>
- [126] <http://www.abclinuxu.cz/slovník/repozitar>
- [127] <http://www.abclinuxu.cz/slovník/gnu-autotools>
- [128] <http://links.twibright.com/index.cz.php>
- [129] <http://links.twibright.com/user.html>
- [130] <http://www.abclinuxu.cz/slovník/configure>
- [131] <http://www.debian.org/doc/manuals/apt-howto/ch-apt-get.cs.html>

## Lokální útoky na operační systém linuxového typu

- [132] <http://www.abclinuxu.cz/slovník/gnu-gpl>
- [133] <http://directory.fsf.org/project/libsafe/>
- [134] <http://www.openwall.com/>
- [135] <http://citeseer.ist.psu.edu/cowan98stackguard.html>

## System vo firemnej sieti – VLSM

- [136] <http://tools.ietf.org/html/rfc791>

## Jaderné noviny – Video4Linux2 – 7 (ovládání)

- [137] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-video4linux2-6b-streamovány-i-o>
- [138] <http://v4l2spec.bytesex.org/spec/>
- [139] <http://v4l2spec.bytesex.org/spec/x756.htm>

## Jaderné noviny – 44 a 45/2007

- [140] [http://kerneltrap.org/Linux/Testing\\_Suspend\\_and\\_Hibernation](http://kerneltrap.org/Linux/Testing_Suspend_and_Hibernation)
- [141] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/1/377768>
- [142] <http://kerneltrap.org/mailarchive/linux-activists/1992/11/3/15150>
- [143] [http://kerneltrap.org/Linux/Swap\\_Over\\_NFS](http://kerneltrap.org/Linux/Swap_Over_NFS)
- [144] <http://kerneltrap.org/mailarchive/linux-kernel/2007/10/30/369602>
- [145] <http://kerneltrap.org/mailarchive/linux-kernel/2007/10/19/348792>
- [146] [http://kerneltrap.org/Linux/Fixing\\_Compiler\\_Misoptimizations](http://kerneltrap.org/Linux/Fixing_Compiler_Misoptimizations)
- [147] [http://kerneltrap.org/Linux/Compiler\\_Misoptimizations](http://kerneltrap.org/Linux/Compiler_Misoptimizations)
- [148] <http://www.abclinuxu.cz/kdo-je/linus-torvalds>
- [149] [http://kerneltrap.org/Linux/2.6.24-rc2\\_Snooze\\_City](http://kerneltrap.org/Linux/2.6.24-rc2_Snooze_City)
- [150] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/7/391733>
- [151] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-31.-10.-2007#api-pro-retezeni-scatterlistu>
- [152] <http://kerneltrap.org/mailarchive/linux-activists/1991/11/6/264>
- [153] [http://kerneltrap.org/Linux/Squashfs\\_Aiming\\_For\\_Mainline\\_Kernel](http://kerneltrap.org/Linux/Squashfs_Aiming_For_Mainline_Kernel)
- [154] <http://kerneltrap.org/mailarchive/linux-fsdevel/2007/11/5/385832>
- [155] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/7/392300>
- [156] [http://kerneltrap.org/Linux/Distributed\\_Storage\\_Subsystem\\_Feature\\_Complete](http://kerneltrap.org/Linux/Distributed_Storage_Subsystem_Feature_Complete)
- [157] <http://kerneltrap.org/mailarchive/linux-fsdevel/2007/11/5/387077>
- [158] <http://tservice.net.ru/s0mbre/old/?section=projects&item=dst>
- [159] <http://kerneltrap.org/node/14029>
- [160] [http://kerneltrap.org/Linux/Debating\\_Distributed\\_Block\\_Devices](http://kerneltrap.org/Linux/Debating_Distributed_Block_Devices)
- [161] <http://www.abclinuxu.cz/kdo-je/andrew-morton>
- [162] [http://kerneltrap.org/Linux/Distributed\\_Storage\\_Subsystem\\_Headed\\_For\\_-mm](http://kerneltrap.org/Linux/Distributed_Storage_Subsystem_Headed_For_-mm)
- [163] <http://www.abclinuxu.cz/kdo-je/alexander-viro>
- [164] <http://kerneltrap.org/mailarchive/linux-fsdevel/2007/10/19/347998>
- [165] [http://kerneltrap.org/Linux/AppArmors\\_Security\\_Goals](http://kerneltrap.org/Linux/AppArmors_Security_Goals)
- [166] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/8/397703>
- [167] [http://kerneltrap.org/Linux/Documenting\\_Security\\_Module\\_Intent](http://kerneltrap.org/Linux/Documenting_Security_Module_Intent)

- [168] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/2/378248>
- [169] [http://kerneltrap.org/Linux/Stable\\_2.6\\_Branches](http://kerneltrap.org/Linux/Stable_2.6_Branches)
- [170] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/5/387032>
- [171] <http://www.abclinuxu.cz/kdo-je/greg-kroah-hartman>

## Jaderné noviny – 46 a 47/2007

- [172] [http://kerneltrap.org/Linux/Scheduler\\_Fixes](http://kerneltrap.org/Linux/Scheduler_Fixes)
- [173] <http://www.abclinuxu.cz/kdo-je/ingo-molnar>
- [174] <http://www.abclinuxu.cz/kdo-je/linus-torvalds>
- [175] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/9/400274>
- [176] <http://www.abclinuxu.cz/slovník/cfs>
- [177] <http://www.abclinuxu.cz/zpravicky/planovac-cfq-se-dostal-do-jadra>
- [178] <http://www.abclinuxu.cz/zpravicky/vyslo-dlouho-ocekavane-jadro-2.6.23>
- [179] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-31.-10.-2007#kontrolni-skupiny>
- [180] [http://kerneltrap.org/Linux/Ceph\\_Distributed\\_Network\\_File\\_System](http://kerneltrap.org/Linux/Ceph_Distributed_Network_File_System)
- [181] <http://kerneltrap.org/mailarchive/linux-fsdevel/2007/11/13/407456>
- [182] <http://www.abclinuxu.cz/slovník/lkml>
- [183] <http://ceph.sourceforge.net/>
- [184] [http://kerneltrap.org/Linux/Fuse\\_Writable\\_Mmap](http://kerneltrap.org/Linux/Fuse_Writable_Mmap)
- [185] <http://kerneltrap.org/mailarchive/linux-fsdevel/2007/11/15/417491>
- [186] <http://www.abclinuxu.cz/slovník/fuse>
- [187] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-15.-3.-2006#jn-radix-stromy>
- [188] [http://kerneltrap.org/Linux/2.6.24-rc3\\_Small\\_Fixes\\_and\\_Cleanups](http://kerneltrap.org/Linux/2.6.24-rc3_Small_Fixes_and_Cleanups)
- [189] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/17/423116>
- [190] [http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-1.-8.-2007#i386-a-x86\\_64-zase-spolu](http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-1.-8.-2007#i386-a-x86_64-zase-spolu)
- [191] [http://kerneltrap.org/Linux/Git\\_on\\_Windows](http://kerneltrap.org/Linux/Git_on_Windows)
- [192] <http://kerneltrap.org/mailarchive/git/2007/11/13/407531>
- [193] <http://code.google.com/p/msysgit/>
- [194] [http://kerneltrap.org/Linux/CFS\\_Scheduler\\_v24\\_Backports](http://kerneltrap.org/Linux/CFS_Scheduler_v24_Backports)
- [195] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/19/427817>
- [196] [http://kerneltrap.org/Linux/Modular\\_IO\\_Schedulers](http://kerneltrap.org/Linux/Modular_IO_Schedulers)
- [197] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/25/444191>
- [198] <http://www.abclinuxu.cz/slovník/cfq>
- [199] <http://www.abclinuxu.cz/kdo-je/andrew-morton>
- [200] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/6/391187>
- [201] <http://www.abclinuxu.cz/kdo-je/theodore-t-so>
- [202] [http://kerneltrap.org/Quote/Memory\\_Is\\_Getting\\_Relatively\\_Cheap](http://kerneltrap.org/Quote/Memory_Is_Getting_Relatively_Cheap)
- [203] <http://www.abclinuxu.cz/kdo-je/david-s.-miller>
- [204] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/15/415394>
- [205] <http://www.abclinuxu.cz/kdo-je/linus-torvalds>
- [206] <http://kerneltrap.org/mailarchive/linux-activists/1991/11/17/343>
- [207] <http://www.abclinuxu.cz/kdo-je/theo-de-raadt>
- [208] <http://kerneltrap.org/mailarchive/opensource-source-changes/2007/11/16/421355>
- [209] <http://www.abclinuxu.cz/kdo-je/alan-cox>
- [210] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/17/423443>
- [211] <http://www.abclinuxu.cz/kdo-je/david-s.-miller>
- [212] <http://kerneltrap.org/mailarchive/linux-kernel/2007/11/15/415232>

## Jaderné noviny – 7. 11. 2007

- [213] <http://lwn.net/Articles/257495/>
- [214] <http://www.abclinuxu.cz/kdo-je/linus-torvalds>
- [215] [http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-1.-8.-2007#i386-a-x86\\_64-zase-spolu](http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-1.-8.-2007#i386-a-x86_64-zase-spolu)
- [216] <http://lwn.net/Articles/257497/>
- [217] <http://kernel.org/pub/linux/kernel/v2.6/testing/ChangeLog-2.6.24-rc2>
- [218] <http://lwn.net/Articles/257002/>
- [219] <http://lwn.net/Articles/257237/>
- [220] <http://www.abclinuxu.cz/kdo-je/greg-kroah-hartman>
- [221] <http://lwn.net/Articles/257238/>
- [222] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-44-a-45-2007#stabilni-vetve-2.6>
- [223] <http://lwn.net/Articles/256827/>
- [224] <http://lwn.net/Articles/257367/>
- [225] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-31.-10.-2007>
- [226] <http://lwn.net/Articles/257639/>



- [227] <http://ozlabs.org/rusty/index.cgi/tech/2007-11-06.html>
- [228] [http://vger.kernel.org/davem/cgi-bin/blog.cgi/2007/11/06#rustys\\_wrong](http://vger.kernel.org/davem/cgi-bin/blog.cgi/2007/11/06#rustys_wrong)
- [229] <http://www.tungstengraphics.com/mm.pdf>
- [230] <http://lwn.net/Articles/257230/>
- [231] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-31.-10.-2007#jmenne-prostory-pro-pid>
- [232] <http://www.abclinuxu.cz/kdo-je/ingo-molnar>
- [233] <http://lwn.net/Articles/257298/>
- [234] <http://www.nabble.com/Re%3A-question%3A-pid-space-semantics.-p3409990.html>
- [235] <http://www.abclinuxu.cz/kdo-je/ulrich-drepper>
- [236] <http://lwn.net/Articles/257300/>
- [237] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-24.-10.-2007#zaznamy-o-dokladech>
- [238] <http://lwn.net/Articles/257303/>

### Jaderné noviny – 14. 11. 2007

- [239] <http://lwn.net/Articles/258522/>
- [240] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-11.-4.-2007#slub-alokator>
- [241] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-31.-10.-2007>
- [242] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-6.-7.-2006#nadesel-cas-ext4>
- [243] <http://lwn.net/Articles/258522/>
- [244] <http://www.abclinuxu.cz/kdo-je/andrew-morton>
- [245] <http://userweb.kernel.org/akpm/mmotm/>
- [246] <http://lwn.net/Articles/258599/>
- [247] <http://www.abclinuxu.cz/slovník/lkml>
- [248] <http://lwn.net/Articles/258600/>
- [249] <http://lwn.net/Articles/258553/>
- [250] <http://lwn.net/Articles/258554/>
- [251] <http://lwn.net/Articles/258555/>
- [252] <http://www.abclinuxu.cz/kdo-je/ingo-molnar>
- [253] <http://lwn.net/Articles/258557/>
- [254] <http://lwn.net/Articles/258559/>
- [255] <http://www.abclinuxu.cz/kdo-je/david-s.-miller>
- [256] <http://lwn.net/Articles/258561/>
- [257] <http://www.abclinuxu.cz/kdo-je/ingo-molnar>
- [258] <http://lwn.net/Articles/258562/>
- [259] <http://lwn.net/Articles/258563/>
- [260] <http://lwn.net/Articles/258565/>
- [261] <http://lwn.net/Articles/250967/>
- [262] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-11.-4.-2007#slub-alokator>
- [263] <http://lwn.net/Articles/258248/>

### Jaderné noviny – 20. 11. 2007

- [264] <http://lwn.net/Articles/259027/>
- [265] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-7.-11.-2007>
- [266] <http://lwn.net/Articles/259028/>
- [267] <http://kernel.org/pub/linux/kernel/v2.6/testing/ChangeLog-2.6.24-rc3>
- [268] <http://www.abclinuxu.cz/kdo-je/greg-kroah-hartman>
- [269] <http://lwn.net/Articles/258939/>
- [270] <http://lwn.net/Articles/258941/>
- [271] <http://lwn.net/Articles/258942/>
- [272] <http://lwn.net/Articles/258943/>
- [273] <http://lwn.net/Articles/258944/>
- [274] <http://lwn.net/Articles/258945/>
- [275] <http://lwn.net/Articles/258946/>
- [276] <http://lwn.net/Articles/258947/>
- [277] <http://lwn.net/Articles/259412/>
- [278] <http://lwn.net/Articles/258948/>
- [279] <http://lwn.net/Articles/259350/>
- [280] <http://lwn.net/Articles/259059/>
- [281] <http://lwn.net/Articles/259060/>
- [282] <http://lwn.net/Articles/259356/>
- [283] <http://lwn.net/Articles/259361/>
- [284] <http://www.abclinuxu.cz/kdo-je/ulrich-drepper>
- [285] <http://lwn.net/Articles/240329/>
- [286] <http://www.abclinuxu.cz/clanky/jaderné-noviny/jaderné-noviny-28.-2.-2007#threadlets>

- [287] <http://lwn.net/Articles/259345/>
- [288] <http://www.abclinuxu.cz/kdo-je/linus-torvalds>
- [289] <http://www.abclinuxu.cz/kdo-je/david-s.-miller>
- [290] <http://lwn.net/Articles/259346/>
- [291] <http://lwn.net/Articles/259347/>
- [292] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-31.-10.-2007#jmenne-prostory-pro-pid>
- [293] <http://www.abclinuxu.cz/clanky/system/opencvz-instalace-a-spusteni>
- [294] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-12.-4.-2006#kontejnery-a-odlehena-virtualizace>
- [295] <http://www.abclinuxu.cz/clanky/system/adresar-proc-brana-do-jadra>

## Jaderné noviny – 28. 11. 2007

- [296] <http://lwn.net/Articles/260127/>
- [297] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-8.-8.-2007#smack-zjednodusena-kontrola-pristupu>
- [298] <http://lwn.net/Articles/259844/>
- [299] <http://lwn.net/Articles/259552/>
- [300] <http://lwn.net/Articles/260136/>
- [301] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-12.-9.-2007#exportovane-symboly-a-interni-api>
- [302] <http://www.abclinuxu.cz/kdo-je/andrew-morton>
- [303] <http://lwn.net/Articles/260045/>
- [304] <http://lwn.net/Articles/259839/>
- [305] <http://www.abclinuxu.cz/kdo-je/rusty-russell>
- [306] <http://lwn.net/Articles/260059/>
- [307] <http://lwn.net/Articles/260061/>
- [308] <http://lwn.net/Articles/259974/>
- [309] <http://folk.uio.no/vegardno/linux/kmemcheck-20071127.txt>
- [310] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-20.-11.-2007>
- [311] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-28.-2.-2007#threadlets>
- [312] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-26.-9.-2007#nove-api-timerfd>
- [313] <http://lwn.net/Articles/259777/>
- [314] <http://lwn.net/Articles/260127/>
- [315] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-31.-10.-2007#hijack>
- [316] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-31.-10.-2007#kontrolni-skupiny>
- [317] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-20.-9.-2006#dalsi-implementace-kontejneru>

## Jaderné noviny – 5. 12. 2007

- [318] <http://lwn.net/Articles/260849/>
- [319] <http://www.abclinuxu.cz/kdo-je/linus-torvalds>
- [320] <http://lwn.net/Articles/260850/>
- [321] <http://kernel.org/pub/linux/kernel/v2.6/testing/ChangeLog-2.6.24-rc4>
- [322] <http://lwn.net/Articles/260969/>
- [323] <http://lwn.net/Articles/260172/>
- [324] <http://lwn.net/Articles/261034/>
- [325] <http://www.abclinuxu.cz/kdo-je/eric-raymond>
- [326] <http://lwn.net/Articles/261035/>
- [327] [http://blogs.sun.com/bonwick/entry/seek\\_hole\\_and\\_seek\\_data](http://blogs.sun.com/bonwick/entry/seek_hole_and_seek_data)
- [328] <http://lwn.net/Articles/260699/>
- [329] <http://lwn.net/Articles/260803/>
- [330] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-4.-7.-2007#pokracujici-pribeh-fallocate>
- [331] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-6.-7.-2006#nadesel-cas-ext4>
- [332] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-25.-10.-2006#sitove-kanaly>
- [333] <http://lwn.net/Articles/169961/>
- [334] <http://www.abclinuxu.cz/clanky/jaderne-noviny/jaderne-noviny-9.-5.-2007#navrat-kevent>
- [335] <http://lwn.net/Articles/260873/>
- [336] <http://lwn.net/Articles/195292/>
- [337] <http://tservice.net.ru/~s0mbre/old/?section=projects&item=unetstack>
- [338] [http://tservice.net.ru/~s0mbre/blog/2006/12/21#2006\\_12\\_21](http://tservice.net.ru/~s0mbre/blog/2006/12/21#2006_12_21)

## Zprávičky

### KDE 4 odloženo na leden 2008

Na dot.kde.org vyšlo oznámení vývojářů KDE, že vydání KDE 4.0 bude odloženo na leden. Původní termín byl polovina prosince. Důvodem odložení jsou problémy týkající se vizuálních prvků a Konqueroru.

[ProgDan](#)

1.12.2007

### Anjuta DevStudio 2.2.3

Vyšlo vývojové prostředí Anjuta DevStudio 2.2.3 (stabilní řada). Opravuje závažný problém s profilovacím pluginem a aktualizuje katalánský a japonský překlad.

[Luboš Doležel](#)

2.12.2007

### Urpmi Internet Automatic 3.0: vybavená Mandriva

Vyšla verze 3.0 programu Urpmi Internet Automatic. Cílem je maximálně zjednodušit nastavení Mandrivy po instalaci. Přidá internetové zdroje (Official, PLF, Cyron), umožní instalaci podpory multimédií (audio a video kodeky, knihovna pro přehrávání DVD), doplňky pro práci s grafikou a archívy, programy SMplayer, Kid3, C Audio Konverter, C Sound Konverter, Audacity, Video > Mpeg4, K9Copy, Opera (včetně č/s lokalizace), Kino, Gimp Kdialogs, Javu, Flash, Wine, několik motivů vzhledu, ikony, dekorace a styly oken, nejběžnější herní knihovny.

[Miroslav Cyroň](#)

2.12.2007

### Hana Linux: spolupráce Jižní a Severní Korey

ETNEWS uvádí, že Severní a Jižní Korea budou spolupracovat v oblasti vývoje GNU/Linuxu. K nim se připojí ještě i čínská strana a vznikne tak „Special Duties Committee“. Výsledkem spolupráce bude distribuce Hana Linux a rozvoj dalších projektů jako OpenOffice.org.

[Luboš Doležel](#)

3.12.2007

### CentOS 5.1

Vyšel CentOS 5.1 pro i386 a x86\_64. Obsahuje balíčky z Red Hat Enterprise Linuxu 5.1 ze všech variant včetně Server a Client, a to v jediném repozitáři. Tentokrát je to poprvé, co byl připraven i ISO obraz pro síťovou instalaci.

[Luboš Doležel](#)

3.12.2007

### Xfce 4.4.2

Vyšlo Xfce 4.4.2. Obsahuje opravy chyb a aktualizace překladů za posledních 8 měsíců – čtete přehled změn.

[Luboš Doležel](#)

3.12.2007

### Damn Small Linux 4.1

Vyšel Damn Small Linux 4.1. Aktualizuje pciutils na verzi 2.2.9 a Torsmo na verzi 0.18, přidává nový textový editor a různé bootovací volby, jsou lépe podporováni alternativní správci oken, usnadňuje stahování dat u MyDSL, byl odstraněn Small Window Manager a celá řada dalších změn.

[Luboš Doležel](#)

3.12.2007

### Ostravský linuxový seminář – 12. 12. na VŠB-TU

Ve středu 12. prosince od 17:00 hod. se na půdě VŠB-TU Ostrava (v posluchárně NA1 – Nová Aula) bude konat seminář z pásma OLS (Ostravských Linuxových seminářů). Podrobnosti se dozvíte na stránkách semináře. Seminář bude přenášán on-line. Odkaz na on-line stream bude umístěn na [prenosy.cesnet.cz](http://prenosy.cesnet.cz).

[Lumír Jasiok](#)

3.12.2007

### NetBeans IDE 6.0

Vyšel NetBeans IDE 6.0. Má vylepšený editor, podporu Ruby, JRuby a Ruby on Rails, snáze se aktualizuje, má integrovaný profiler, rozšířenou podporu pro webové služby a celou hromadu dalších novinek.

[Luboš Doležel](#)

3.12.2007

### SystemRescueCD 0.4.2

Vyšla nová verze výborného záchraného CD založeného na Gentoo s názvem SystemRescueCD – 0.4.2. Nová verze obsahuje jádro 2.6.23.8 s podporou Reiser4, novou verzi NTFS-3g – 1.1120, byla znovu zapracována podpora pro smbfs, v konzoli je ve výchozím nastavení vypnutý unicode, taktéž byly přidány ovladače MadWiFi, přibýly další nástroje ntfsprogs. Přidány byly bootovací volby rootpass=xxx pro nastavení hesla roota a ethx=ipaddress pro nastavení IP adresy pro všechna rozhraní.

[Tomáš Pantlík](#)

4.12.2007

### 2D výkon ovladačů NVIDIA 169.04 beta

Jednou z novinek ovladače NVIDIA 169.04 beta bylo zvýšení výkonu akcelerace RENDER. Phoronix tuto změnu otestoval pomocí render\_bench a skutečnost potvrdil: testy 2D vykreslování, jejichž dokončení dříve trvalo i několik minut, jsou nyní hotové za pár sekund.

[Luboš Doležel](#)

4.12.2007

### Adobe Flash Player 9 Update 3

Vyšel Adobe Flash Player 9 Update 3. Tato verze by měla fungovat i pod prohlížečem Opera, slibovaná verze pro x86\_64 však stále nebyla uvolněna.

[Luboš Doležel](#)

4.12.2007

### Ohlášena platforma Red Hat MRG

Red Hat dokončuje práce na svém systému s odezvou v reálném čase – ještě tento měsíc by měla vyjít beta platformy Red Hat MRG (Messaging, Real Time and Grid). Při ohlašování tohoto produktu zástupce Red Hatu sdělil, že nedávno vydaný SUSE Linux Enterprise Real Time 10 je postavený na jejich beta kódu a Novell na oplátku nepřispěl ani jediným řádkem kódu.

[Luboš Doležel](#)

4.12.2007

### Theodore T'so nastupuje do Linux Foundation

První severoamerický vývojář Linuxu Theodore T'so nastupuje do Linux Foundation jako hlavní stratég pro vývoj plat-

forem (chief platform strategist). Bude se podílet na vývoji Linux Standards Base a na dalších projektech organizace. V LF bude pouze po omezenou dobu – za dva roky se vrátí zpět k původnímu zaměstnavateli IBM.

Luboš Doležel

4.12.2007

## Mandriva Flash 2008 4 GB v prodeji

Do prodeje se dostal Mandriva Flash 2008 o velikosti 4 GB. Je to živý systém s prostředím KDE – zbylý diskový prostor flash disku můžete využít pro svá data. Obsahuje klasickou sestavu programů: Mozilla Firefox a Thunderbird, OpenOffice.org, Skype, Javu, Flash Player a další. Doporučovaná cena je 79 eur.

Luboš Doležel

4.12.2007

## AIM transport v Gmail

Na webu Google blogoscooped se objevila informace o novince, že Google zabudoval do Gmailu AIM transport pro svůj integrovaný chat – google talk. Díky tomu se lze spojit i s uživateli u nás velmi oblíbeného protokolu ICQ (patří pod společnost AOL, stejně jako protokol AIM) bez nutnosti transportu 3. strany.

Vladik

4.12.2007

## Standardizace OpenXML opět ve hře

Ve dnech 25. – 29. 2. 2008 proběhne v Ženevě zasedání zástupců národních standardizačních organizací, jehož cílem je posoudit, jak byly vyřešeny připomínky k návrhu DIS 29500 (formát Office Open XML). Během 30 dnů po skončení tohoto zasedání mohou hlasující státy přehodnotit své původní stanovisko (z 2. září 2007) ke standardizaci formátu. Podrobnosti uvádí dokument, který připravila ODF Alliance.

Lukáš Jelínek

5.12.2007

## OpenOffice.org 2.3.1

Vyšel OpenOffice.org 2.3.1. Nepřináší žádné nové funkce, pouze opravuje různé pády a další regrese poslední verze (použije s PDF exportem či pády při načítání velkých obrázků). Novinky očekávejte ve verzi 2.4, která přijde v březnu 2008.

Luboš Doležel

5.12.2007

## PDF 1.7 schválen jako ISO 32000

Populární formát PDF (ve verzi 1.7) byl schválen jako standard ISO 32000. Celá věc byla ukázkou toho, jak by měl probíhat korektní standardizační proces. Dokonce i samotný Adobe přidával za Spojené státy komentáře o nedostatcích, jakmile nějaké našel.

Luboš Doležel

5.12.2007

## openSUSE Weekly News 2

openSUSE.org dnes vydal druhé číslo svého týdeníku, ve kterém informuje o novinkách a zajímavostech okolo openSUSE. V druhém díle se například můžete dočíst, jak zrychlit správce balíčků, o Monu na MacOS, o nových efektech v KDE 4 nebo o portaci YASTu do Qt 4 (screenshot).

Michal Smrž

5.12.2007

## Likewise uvolnilo Active Dir autentizaci pro Linux

Likewise uvolnilo open source podporu autentizace vůči Active Directory. S Likewise Open je tak snadno možné použít stejné přihlašovací údaje pro stanice s Linuxem jako s Windows. Kód je pod GNU GPLv3, více píše Linux-Watch.

Luboš Doležel

5.12.2007

## Sun zavede systém odměn pro OSS vývojáře

Sun Microsystems zavede systém odměn pro open source vývojáře, kteří přispívají projektům jako OpenJDK, NetBeans, OpenOffice.org aj. Chce tak kompenzovat jejich vynaloženou práci a dále je motivovat. Podrobnosti ještě budou zveřejněny. Čtěte ComputerWorld.

Luboš Doležel

5.12.2007

## Zabezpečení RHEL 5 podle NSA

NSA zveřejnila detailní 170-stránkovou příručku o zabezpečení Red Hat Enterprise Linuxu 5. RHEL se tímto přidává k systémům od Applu, Microsoftu a Sunu, ke kterým NSA vydává návody. Mnohé rady jsou použitelné i na jiných distribucích.

Michal Schmidt

6.12.2007

## Jak zvýšit výkon XRender na fglrx

Phoronix informuje, že ATI ovladače fglrx mají skrytou volbu TexturedXrender. Ta zlepšuje 2D akceleraci a v testech to znamená nárůst výkonu o 80 %. Stačí ji přidat do xorg.conf; pravděpodobně však ještě není její vývoj dokončený a proto nebyla publikována.

Luboš Doležel

6.12.2007

## FreeBSD 7.0-BETA4

Vyšlo FreeBSD 7.0-BETA4. Bohužel se zvětšuje časový skluz vývoje – podle předpokladů by měly v prosinci ještě vyjít dvě RC verze a v lednu verze konečná, ale není moc jisté, zda to tak dopadne.

Luboš Doležel

6.12.2007

## Novell reaguje na obvinění Red Hatu o chamtivosti

Novell odpověděl na obvinění Red Hatu, že postavil SUSE Linux Enterprise Real Time 10 na jejich beta kódu, aniž by přispěl něčím nazpátek. Prohlášení říká, že toto je prostě open source a Red Hat nemá shazovat své zpoždění na někoho jiného. Navíc připomíná, že Novell do vývoje real time Linuxu pochopitelně přispívá také.

Luboš Doležel

6.12.2007

## V KDE 4 bude i jednoduché K Menu

Mnohým uživatelům se nelíbí Kickoff menu v KDE 4. Jejich prosby byly vyslyšeny, a tak bude v KDE 4 i klasické jednoduché menu. Do Release Candidate 2 se však už nedostane.

Luboš Doležel

6.12.2007

## Článek o Compiz Fusion na PCTuning.cz

Na PCTuning vyšel článek o Compiz Fusion. Článek popisuje vybrané efekty v rámci Compiz Fusion, které ilustruje

videozáznamy. Mimo jiné zmiňuje i nízké hardwarové nároky, bohatost rozšíření a rychlost vývoje.

Nicky 726

6.12.2007

## Obrat: OOXML by nezůstalo pod správou ISO

Microsoft udělal další podraz v souvislosti s formátem OOXML. Původně slíbil, že OOXML by po svém schválení bylo standardem, který by byl spravován ISO, a díky tomuto slibu získal nějaké kladné hlasy navíc. Nyní obrací se slovy, že OOXML by zůstalo pod kontrolou Ecma (a Microsoftu), přičemž ISO by pouze vydalo erratu. Čtěte Groklaw.

Luboš Doležel

7.12.2007

## openSUSE 11 alfa0

Krátce po vydání openSUSE 10.3 byla vydána testovací verze openSUSE 11 alfa0. Stahovat můžete ze [en.opensuse.org](http://en.opensuse.org). Takže testujte a hlase chybičky... Kvalitní vydání vám bude odměnou.

Marek Stopka

7.12.2007

## BusyBox: dalším žalovaným je Verizon

Vývojáři BusyBoxu se rozhodli tvrdě jít po všech, kteří s jejich softwarem nakládají v rozporu s licencí. Tentokrát je žalován Verizon, telekomunikační firma z USA. Ten zákazníkům služby FiOS poskytuje bezdrátový router, ve kterém je BusyBox, aniž by byly splněny všechny náležitosti stanovené v GNU GPLv2.

Luboš Doležel

7.12.2007

## Druhá petice za uvolnění zdroj. kódů OS/2

Na stránkách OS2 World.Com se dozvíte, které části OS/2 jsou plně vyvíjené IBM, a tudíž je možné uvolnit jejich zdrojový kód pod open source licencí. Zároveň byla odstartována druhá petice, i vy se můžete přidat!

Jiri Krc

7.12.2007

## Geubuntu 7.10

Vyšlo Geubuntu 7.10, derivát Ubuntu. Tato distribuce kombinuje prostředí GNOME a Enlightenment. Do toho všeho přidává i součásti z XFce.

Luboš Doležel

7.12.2007

## Cesta ven z potíží s videem na webu?

Mozilla a Opera Software spolupracují na přípravě a podpoře speciálního HTML tagu pro videa. Současná situace, kdy jsou pro přehrání videí na webu zapotřebí různé kodeky, nebo dokonce Flash přehrávač, je nevyhovující – např. pro obyčejné obrázky něco takového potřeba není. Je zároveň důležité, aby byla stanovena sada povolených kodeků: nyní se pracuje na podpoře Ogg Theora. Čtěte LinuxWorld.

Luboš Doležel

7.12.2007

## Kolidující uživatelé, odstávka

Během neděle dojde k řešení kolizí mezi loginy a přezdívkami uživatelů (viz upozornění). Po deváté večer bude server zhruba čtvrt hodiny nedostupný z důvodu údržby routeru

(upgrade, přepojení konektivity na firewallu Juniper J4350). Děkujeme za pochopení.

Redakce

8.12.2007

## Ruby on Rails 2.0

Framework pro webové aplikace Ruby on Rails se právě dočkal své druhé hlavní verze. Jakmile na blogu jeho původního autora odezní Slashdot effect, budeme si moci přečíst více.

Jakub Hegenbart

8.12.2007

## Papírový VIM kalendář pro rok 2008 ke stažení

Bram Moolenaar (tvůrce editoru vim) připravil pro fanoušky a uživatele editoru Vim skládací kalendář na rok 2008 ke stažení a vytisknutí na stranu papíru A4 (formát PDF). Takže pokud jste fanoušky vimu, není co řešit.

Jiří Pagáč

8.12.2007

## NASM 2.0

Na světě je nová verze oblíbeného přenositelného assembleru NASM. Přináší lepší integraci s C99, lepší podporu x86\_64, podporu SSE 3 až 5 a další změny. Číslo verze poskočilo rovnou na 2.0.

Jakub Hegenbart

9.12.2007

## fprint vs. Microsoft

fprint, knihovna pro práci se čtečkami otisků, narazila na zájímavý problém – nová revize HW od firmy Microsoft začala při každém zasunutí do USB portu kontrolovat challenge-response algoritmem (který se ukázal být 128bitovým AES), zda ovladač, který ji obsluhuje, pochází od Microsoftu. Dlužno dodat, že fprint 0.5.0 již posílá platné odpovědi :-).

Jan Kundrát

9.12.2007

## Kolize mezi uživateli odstraněny

Ne-unikátní loginy a přezdívkové uživatelské byly opraveny. Seznam uživatelů, jejichž login či přezdívka byl změněn, najdete v této tabulce.

Leoš Literák

9.12.2007

## Vyzkoušejte si openSUSE 11 alpha0 online

Marek Stopka nabízí možnost vyzkoušet si alpha verzi openSUSE 11 pomocí VNC bez nutnosti lokální instalace. K dispozici je KDE i GNOME.

Marek Stopka

9.12.2007

## Nokia: Ogg je proprietární

Za silného protestu firmy Apple schválilo před několika měsíci W3C Ogg Theora a Vorbis jako doporučené kodeky pro implementace HTML5. Nyní se zdá, že Nokia se snaží rozhodnutí zvrátit: tvrdí, že omezující DRM je důležité, a svobodné a patenty nezátížené Ogg označuje za proprietární. Čtěte Slashdot.

Luboš Doležel

10.12.2007

## BSA pouze varuje, ale open source nedoporučí

Protipirátská organizace BSA se již podruhé ocitla v komiksu. První komiks se týkal podobnosti zkratky BSA a BSE, což

je označení pro nemoc šílených krav. Autoři komiksu tentokrát kritizují situaci, kdy organizace v rámci své poslední kampaně obesílá poštou 15 tisíc náhodně vybraných českých firem. Dopis zástupce firem upozorňuje na rizika softwarového pirátství, ale zároveň nezmiňuje možnost open source. Tento postup v minulosti kritizoval například Filip Molčan, předseda OSS Alliance.

Radim Hasalík

10.12.2007

## Nový správce písem Fontmatrix

Jedním ze šlágrů nekonečných debat a profesionálním nasazení Linuxu v DTP a grafice je i absence správy písem. Možnou a celkem pádnou odpovědí je Fontmatrix. V souvislosti s ním vyšlo drobné zamýšlení nad danou problematikou. Binární balíčky Fontmatrixu pro openSUSE hledejte v oblíbeném BuildService repozitáři.

Petr Vaněk

10.12.2007

## Google Picasa 2.7 Beta

Vyšlo Google Picasa 2.7 Beta „pro Linux“. Tato verze nabízí uploadování do online Picasa alb, vylepšuje importování, lépe pracuje s formáty RAW plus jiné novinky.

Luboš Doležel

10.12.2007

## V Nizozemí se bude rozhodovat o nasazení ODF

Nizozemský parlament bude rozhodovat o povinném používání OpenDocument Formátu na vládních úřadech. Přechod by vyšel na více než 8 milionů Eur, ale vedl by ke snížení nákladů v budoucnu. Microsoft pochopitelně bojuje proti tomu a chce, aby se definice neomezovala na ODF. Čtěte ComputerWorld.

Luboš Doležel

10.12.2007

## Everex uvede ultrapřenosné PC s Linuxem

Firma Everex uvede UMPC (ultrapřenosné PC) s Linuxem. Bude trochu podobné Asus Eee PC a bude na něm běžet distribuce gOS, která se zaměřuje na používání Google Apps. Podle LinuxDevices poběží na CPU VIA 1,2 GHz s 30 GB pevným diskem a ve výbavě bude webkamera.

Luboš Doležel

10.12.2007

## OLPC představeno prezidentovi Srí Lanky

OLPC bylo představeno prezidentovi Srí Lanky. Tato země možná notebooky pro své děti zakoupí, ale nebude moci nést všechny náklady – možná bude potřebovat, aby část platil soukromý sektor. Přečtěte si také článek na DesktopLinux uvažující nad tím, jak Linux přebírá segment levných počítačů.

Luboš Doležel

11.12.2007

## VMware VI 3.5

Společnost VMware dnes oficiálně uvolnila novou produktovou řadu obsahující ESX server 3.5 a management server VCMS 2.5 spolu s VMware Consolidated Backup frameworkem v 1.1 a VMware Update Managerem. Zajímavý je iaké nový cenový model pro SMB trh.

Vladan Laxa

11.12.2007

## JBoss Developer Studio

Red Hat oznámil nový produkt JBoss Developer Studio pro vývojáře. Obsahuje aplikační platformu JBoss, vývojové nástroje založené na Eclipse a Red Hat Enterprise Linux. V ceně 99 dolarů je i jeden účet pro plný přístup k Red Hat Network.

Michal Schmidt

11.12.2007

## Recenze tří komerčních programů pro hudebníky

Linux Journal připravil recenzi tří komerčních programů pro hudebníky. Jde o programy Transcribe! (pro přepis melodií), MuSing (transformace bubnových podkladů) a energyXT2 (MIDI sequencer).

Luboš Doležel

11.12.2007

## Phoronix otestoval 7 verzí Wine

Phoronix se zaměřil na vývoj výkonnosti Wine v jednotlivých verzích. Testy ukázaly, že výkon je stabilní až na několik velice překvapivých zlomů. Více informací na phoronix.com.

Šimon Tóth

11.12.2007

## Theora a Vorbis dočasně mizí z HTML5

Kvůli k bouřlivým diskusím (nejen) v poštovní konferenci WHATWG odstranil Ian Hickson ze specifikace HTML5 zmínku o kodecích Theora a Vorbis, vizte jeho e-mail v konferenci, diff specifikace HTML5 a Slashdot.

Tomáš Znamenáček

11.12.2007

## KDE 4.0 Release Candidate 2

Vyšlo KDE 4.0 Release Candidate 2. Uživatelé jsou opět žádáni o hlášení chyb. Stahovat můžete i živé CD.

Luboš Doležel

11.12.2007

## Bude Access Linux Platform na japonských mobilech?

LinuxDevices píše o japonském úspěchu Access Linux Platform (ALP) pro mobilní telefony. Vypadá to, že výrobci NEC, Panasonic a Esteemo začnou uvažovat o výrobě 3G telefonů (pro tento trh) postavených na této platformě. ALP je navíc kompatibilní se specifikacemi, které vydala LiMo Foundation.

Luboš Doležel

11.12.2007

## Linuxová vánoční besídka v Brně

Srdečně zveme všechny na linuxovou vánoční besídku v Brně, která se koná tento pátek (14. 12. 2007) od 18.00 v hospůdce Pegas na Jiráskově (mapa). Dárky jsou dobrovolné, ale kamarády jistě potěší :-).

David Jaša

12.12.2007

## Sun uvolnil specifikace UltraSPARC T2 (Niagara 2)

Sun Microsystems vydává specifikace nového procesoru UltraSPARC T2 (označovaného jako Niagara 2) s licencí GPL.

Specifikace UlstraSPARC T1 (Niagara 1) byla od svého uvolnění v roce 2005 stažena 6500×. Informuje eWeek.

Robert Krátký

12.12.2007

## wxWidgets programy pro Asus Eee PC

LWN.net odkazuje na zápisek v blogu vývojářů wxWidgets, který pojednává o možnostech portování wx programů na Asus Eee PC.

Robert Krátký

12.12.2007

## Stav linuxového desktopu v Číně

Linux.com představuje situaci okolo oblíbenosti Linuxu na desktopu v Číně. Linux zde zažívá nárůst, a to obzvláště poté, co čínská vláda začala bojovat s nelegálním užíváním softwaru. I nadále se však 31,9 % uživatelů hned zbaví Linuxu a nahradí jej nelicencovanými Windows.

Luboš Doležel

12.12.2007

## Jak probíhá port SimCity na OLPC

LinuxWorld připravil rozhovor s Donem Hopkinsem, který pracuje na portování hry SimCity na notebooky OLPC. Dozvíte se, na co při portování narazil, jak hodnotí původní kvalitu kódu, jaké funkce byly odstraněny a jaké přidány a co se plánuje do budoucna.

Luboš Doležel

12.12.2007

## Írán využívá superpočítač s Linuxem

Amirkabirská univerzita v Íránu používá superpočítač s Linuxem údajně pro předpověď počasí a meteorologický výzkum. Počítač tvoří 216 procesorů Opteron od AMD s výpočetní kapacitou až 860 gigaflopů. Postavení počítače údajně komplikovalo obchodní embargo ze strany Spojených států, které musí dodržovat i výrobce procesorů AMD. Více na ITBIZ.cz.

Radim Hasalík

12.12.2007

## OpenSUSE Weekly News 3

openSUSE.org dnes vydal třetí číslo svého týdeníku. Krom povídání o OpenSUSE 11 a Moonlightu se tam nachází návod, jak aktivovat u freetype fontů subpixel hinting, který je deaktivován, aby neporušoval možné licence Microsoft ClearType.

Michal Smrž

12.12.2007

## EasyTAG 2.1.4

Vyšla verze 2.1.4 programu pro editaci tagů u audiosouborů EasyTAG. Kromě oprav chyb je viditelnou změnou zobrazení tagů přímo v hlavním okně se seznamem souborů (viz screenshot), dále je přidána podpora pro obrázky v Ogg a FLAC souborech a kód pro práci s FLAC byl kompletně přepsán.

Otakar

12.12.2007

## PCTuning: recenze Asus Eee PC

Na PCTuningy vyšla recenze nového subnotebooku Asus Eee PC. Subnotebook využívá upravenou verzi Xandros Linuxu.

Nicky 726

12.12.2007

## OpenOffice.org 3.0 – co nás čeká?

Na serveru Audiovizuálního centra studentů ČVUT je ke stažení přednáška Filipa Molčana z Linux Meetingu 2007 v Ústí nad Labem o projektu OpenOffice.org a o tom, co nás čeká ve verzi 3.0 tohoto kancelářského balíku.

Otakar

12.12.2007

## Co s nahlášenými chybami KDE 3?

Bram Schoenmakers píše o rozhodnutí, před kterým nyní stojí vývojáři KDE. Bugzilla KDE má přes 15 tisíc neodstraněných nahlášených chyb a přes 13 tisíc přání uživatelů – tento počet se stále zvyšuje. Bylo navrženo řešení, že všechny chyby KDE 3 budou uzavřeny, neboť jejich procházení by zabralo spoustu času a kód KDE 4 je už dosti odlišný. Můžete vyjádřit svůj (ne)souhlas v hlasování.

Luboš Doležel

12.12.2007

## Volné učební materiály pro Maemo 4.x

Nokia vydala učební materiály pro zájemce o vývoj na platformě Maemo 4.x (linuxové tablety Nokia). Materiály zahrnují přehled nástrojů a postupů při vývoji, nabízí ukázkové příklady a další užitečné informace. Více se dozvíte v oznámení.

Luboš Doležel

12.12.2007

## Movable Type je opravdový Open Source

Počínaje dnešním dnem je Movable Type (až dodnes komerční CMS / systém pro blog) uvolněn pod GPL. Více informací najdete v Community Blogu.

Daniel Kvasnička ml.

12.12.2007

## Ulteo: OpenOffice.org jako online služba

Ulteo byla dříve distribuce založená na Debianu a Ubuntu. Jenže projekt se později zaměřil na vývoj webového kancelářského systému a nyní se do fáze betatestování dostala online verze OpenOffice.org. Pro jeho použití stačí prohlížeč s podporou JavaScriptu a podporou Javy (od Sunu). Více na DesktopLinux.

Luboš Doležel

13.12.2007

## Kde byste našli Linux v mlékárenském voze?

LinuxDevices píše o zajímavém použití Linuxu. Německá firma uvádí produkt pro vozy pro převoz tekutých potravin (např. mléka). Ten je složený ze dvou ethernetem propojených linuxových jednotek. Systém používá GPS pro lokalizaci vozu a GPRS pro komunikaci s centrálou.

Luboš Doležel

13.12.2007

## Prosincové setkání CZJUG na ČVUT

V pondělí 17. 12. 2007 od 18.00 proběhne na půdě ČVUT v místnosti K1 na Karlově náměstí prosincové setkání CZJUG (CZ Java User Group). Budou se konat prezentace od Romana Staňka a Václava Pecha. Více informací na java.cz.

freshmouse

13.12.2007

## ECDLix – distribuce pro testy ECDL na open source

Společnost Certicon a.s. ve spolupráci s Coprosys Ústí spol. s r. o. připravuje linuxovou distribuci ECDLix. Bude určena pro testování ECDL. Testy lze již nyní skládat i v OpenOffice.org, Thunderbirdu či Firefoxu. Začátkem roku 2008 přibude i možnost nahradit při testech Windows za Linux a celé testování provést na open source.

Rudolf Svátek

13.12.2007

## Compiz Fusion – instalace, ladění a vliv na výkon

Na [www.pctuning.cz](http://www.pctuning.cz) vyšlo pokračování článku o Compiz Fusion. Tentokrát sa autor pozrel na nastavenia CF i jeho výkon. Názov napoveda viac: Compiz Fusion – instalace, ladění a vliv na výkon.

Ivan

13.12.2007

## O úspěchu OLPC v jedné britské rodině

BBC publikovalo článek, kde otec popisuje, jakým přínosem byl pro jeho dítě notebook OLPC. Není to však rodina z Afriky, ale z Velké Británie, které se podařilo výrobek získat. Otec byl zpočátku poněkud skeptický, ale nyní už věří, že se jedná o vynikající výukový nástroj.

Luboš Doležel

13.12.2007

## Backendy Phononu od Trolltechu jdou do KDE SVN

Trolltech oznámil, že jeho backendy multimediálního frameworku Phonon (součást budoucího Qt 4.4) budou přesunuty do SVN repozitářů projektu KDE. Přesun do veřejného repozitáře znamená rozšíření spolupráce s open source komunitou. Původně měl Phonon jen Xine backend, ale Trolltech naprogramoval GStreamer backend a další backendy pro Windows a Mac OS X.

Luboš Doležel

13.12.2007

## Valgrind 3.3

Vyšla nová verze vývojářského nástroje valgrind. Novinek je spousta, opět funguje Helgrind (nástroj pro detekci chyb při práci s vlákny), Massif byl kompletně přepsán a Callgrind nyní umí profilovat předpovědi skoků. Detaily <http://valgrind.org>.

Šimon Tóth

14.12.2007

## phpBB 3.0 Olympus

Konečně vyšla finální verze oblíbeného fóra phpBB 3.0. Přehled nových vlastností je na domovské stránce.

Ivan

14.12.2007

## Btrfs 0.9 – konkurence ZFS od Oracle

Vyšla verze 0.9 souborového systému Btrfs. Btrfs je moderní filesystem vyvíjený firmou Oracle, který svými vlastnostmi konkuruje ZFS (více viz blogpost Linuxová konkurence ZFS od Oracle – Btrfs). V nové verzi došlo k opravě mnoha chyb, vylepšení výkonu a přidání několika nových funkcí (např. podpora xattr). Na stránkách Btrfs nyní najdete i časový plán.

Btrfs však stále ještě není určeno k nasazení (interní formát se bude ještě měnit).

[\\_xm\\_](#)

14.12.2007

## Plně otevřený WebKit

Projekt WebKit zcela otevřel svou politiku pro přijímání nových příspěvatelů a recenzentů kódu [reviewers]. WebKit se tak dostává z přímé kontroly firmy Apple a padá poslední stín uzavřenosti projektu. Nová politika už přinesla první ovoce – recenzenta Holgera Freythera.

David Jaša

14.12.2007

## Canonical vydal VCS Bazaar 1.0

Canonical vydal Bazaar 1.0 – software pro správu verzí (VCS), který se odlišuje hlavně tím, že je decentralizovaný. Podle firmy je centrální umístění podobného systému vhodné jen pro proprietární a striktně řízené projekty. Uživatelům CVS a Subversion by použití softwaru mělo připadat snadné. Více píše Linux-Watch.

Luboš Doležel

14.12.2007

## Sourcefire nabízí placenou podporu pro ClamAV

Firma Sourcefire, která se už dříve stala vlastníkem open source antivirového softwaru ClamAV, začala pro tento produkt nabízet technickou podporu. Firma předplatitelům navíc dodává speciální moduly, se kterými je možné sledovat určité druhy úniků dat (čísla kreditních karet, rodná čísla apod.).

Luboš Doležel

14.12.2007

## COMDOM Antispam 1.4

COMDOM Software vydal verzi 1.4 svého produktu COMDOM Antispam pre servery. Nová verzia rozširuje výkonné jadro Bayes filtra o ďalšie funkcie CJK (Chinese, Japanese, Korean) email support, Real Time Black Lists a smtpd\_client\_restrictions. Viac v poznámkach k vydaniu.

Radovan

14.12.2007

## Wine 0.9.51

Vyšlo Wine 0.9.51. Tato verze zlepšuje podporu WinHelp, lépe zvládá japonské fonty, opravuje rpcrt4 a záznam zvuku přes ALSA a kromě jiného také umí lépe přepínat rozlišení.

Luboš Doležel

14.12.2007

## EeeXubuntu: upravené Ubuntu pro ASUS Eee PC

EeeXubuntu je upravené Ubuntu pro subnotebook ASUS Eee PC. Řeší některé specifické problémy (snížení počtu zápisů na SSD disk, nativní podpora ovladačů Wi-fi, optimalizace pro instalaci z flash paměti a vylepšení pro práci v nízkém rozlišení).

Nicky 726

14.12.2007

## Vývoj Slaxu přerušen

Tomáš Matějčík, vývojář live distribuce Slax, dnes oznámil přerušení jeho vývoje. Znamená to konec Slaxu pod jeho vedením, nebo jen prázdniny?

Vlastimil Ott

14.12.2007



**Linux 2.6.22.15 a 2.6.23.10**

Vyšel Linux 2.6.22.15 a 2.6.23.10. Obě dvě verze obsahují důležité opravy chyb. Jde např. o opravy v KVM, síťování nebo o opravu možného poškození zásobníku při použití futexů.

[Luboš Doležel](#)

15.12.2007

**Traverso DAW 0.42.0**

Vyšla verze 0.42.0 DAW programu Traverso. Nová verze podporuje čtení a zápis formátů Ogg Vorbis, WavPack, FLAC a MP3, umí změnu vzorkování „za letu“, obsahuje systém pro zálohování projektu, lepší práci s diskem a také přibyla možnost kódovat nahrávané stopy do formátů W64 a WavPack.

[Otakar](#)

15.12.2007

**Google by chtěl konkurovat Wikipedii**

Google vyvíjí publikační systém podobný Wikipedii. Publikování článků v podání Googlu má však několik zajímavých vylepšení. Jde například o zveřejnění identity autora a možnost částečného financování reklamou. Negativem je nemožnost přímé editace cizích příspěvků, články půjdou pouze komentovat.

[Nicky 726](#)

15.12.2007

**IBM: vyplňte 3min. anketu a vyhraďte MP4 přehrávač**

IBM nabízí možnost vyhrát MP4 přehrávač (a další ceny) za vyplnění 3minutové ankety – zkuste štěstí.

[Reklama](#)

15.12.2007

**SquirrelMail 1.4.13**

Vyšel SquirrelMail 1.4.13. Účet jednoho ze spolupracovníků projektu byl zneužit k „vylepšení“ předchozích verzí 1.4.11 a 1.4.12 o škodlivý kód. Přišlo se na to po zjištění, že neseděl MD5 hash zveřejněných souborů. Je tedy velmi důležité, aby uživatelé postižených verzí aktualizovali.

[Luboš Doležel](#)

15.12.2007

**ApacheMQ 5.0**

Tento týden byla uvolněna nová verze JMS brokeru Apache ActiveMQ. Opravuje celou řadu bugů a přidává pár novinek, např. podporu transformační služby, zrcadlení front zpráv, zasílání zpráv typu BLOB, integraci enterprise vzorů, webovou konzoli atd. ActiveMQ umí poměrně dost věcí a nabízí jednoduchý a účinný způsob integrace různých aplikací postavený nad vyzrálým open source produktem. Kromě Javy má ActiveMQ možnost spolupráce s C++, .NET, Springem, REST atd.

[Karel Benák](#)

16.12.2007

**TrueCrypt 5.0 s GUI i pro GNU/Linux**

Na leden roku 2008 je plánováno vydání páté verze open-source on-the-fly šifrovačeho nástroje TrueCrypt. (Krátkou zprávu o vysoké bezpečnosti TrueCryptu si můžete v češtině přečíst např. v článku předního českého kryptologa Vlastimila Klímy a Milana Zámotného.) Krátký seznam změn můžete najít již teď na domovské stránce programu. Pro uživatele

GNU/Linuxu bude asi nejzajímavější novinkou přidání grafického uživatelského rozhraní pro linuxový port TrueCryptu.

[Michal Růžička](#)

16.12.2007

**Newyorská burza začíná nasazovat Linux**

Newyorská burza dává před proprietárními Unixy přednost Linuxu. Podle The New York Times je pro burzu důležitá flexibilita a nezávislost – z těchto důvodů je ze hry vyřazen HP-UX, AIX i Solaris. Zatím byl Linux nasazen na 600 servech na platformě x86.

[Luboš Doležel](#)

17.12.2007

**CentOS 4.6**

Vyšel CentOS 4.6 (legacy řada) pro i386, x86\_64, s390, s390x a ia64. Představuje obsah Red Hat Enterprise Linuxu 4.6 s aktualizacemi vydanými do 15. prosince. Systém nabízí autofs5, OpenOffice.org 2.0 a jiné novinky.

[Luboš Doležel](#)

17.12.2007

**Mozilla umožňuje posílat peníze přímo projektům**

Mozilla Foundation začala dárčům umožňovat volbu projektu, na který chtějí přispět. Zatímco finanční příspěvky bylo organizaci možné posílat už od krátce po založení společnosti v roce 2003, nyní můžete poslat peníze konkrétním projektům, a to Bugzille, Caminu, SeaMonkey nebo Mozilla Accessibility Community. Podrobnosti na MozillaZine.

[Luboš Doležel](#)

17.12.2007

**NetBSD 4.0**

Po dlouhém čekání a znovuoddělení od hlavní vývojové větve zhruba před rokem vyšlo včera (16. 12. 2007) NetBSD 4.0. Oficiální oznámení je očekáváno na 23. 12. (máme tu i precedens). Stahovat můžete již nyní z hlavního FTP serveru nebo z některého ze zrcadel (chvilku potrvá, než se na nich čerstvé obrazy objeví).

[Dominik Joe Pantůček](#)

17.12.2007

**Sun uvolnil javový PDF renderer**

Sun Microsystems uvolnilo open source PDF renderer, který je kompletně v Javě. Je k dispozici pod licencí LGPL. Není to žádný čerstvý projekt, Sun na něm pracoval už od roku 2003.

[Luboš Doležel](#)

17.12.2007

**Swfdec 0.5.5**

Vyšel Swfdec 0.5.5. Tato verze je průlomová v tom, že je nově podporováno mnoho různých přehrávačů videí. Kromě toho byla zlepšena podpora vstupu, takže třeba na YouTube by měl fungovat proužek pro převíjení po videu.

[Luboš Doležel](#)

17.12.2007

**BusyBox vs Xterasys vyřešeno mimosoudně**

Hrozba soudních popotahování zafungovala na další firmu: Xterasys porušující GPL u BusyBoxu se licenci podřídí mimosoudně. Xterasys nyní zveřejní zdrojový kód na webových stránkách, ve firmě bude mít člověka, který zajistí, aby se

podobná věc neopakovala, a vynahradí žalující straně škody.  
[Luboš Doležel](#) 18.12.2007

## Damn Small Linux 4.2

Vyšel Damn Small Linux 4.2. Mtpaint nahradil xpaint, bylo vylepšeno všechno možné okolo nabídky grafických stylů, jejich konfigurace atd., dále došlo ke změně některých ikon, bylo opraveno několik chyb a Torsmo lépe zvládá názvy baterií.  
[Luboš Doležel](#) 18.12.2007

## Česká pošta zřejmě přejde na grafický systém

Podle předběžných informací hodlá Česká pošta na přibližně 2000 pobočkách postupně vyměnit používaný hardware. Tyto počítače by přešly na grafické uživatelské rozhraní. Vyžádaly si to změny v souvislosti s připravovaným projektem Czech-Point, který umožňuje občanům na jednom místě získat výpisy z různých rejstříků. V současné době pošta využívá SuSE Linux a vlastní aplikaci nAPOST.  
[Radim Hasalík](#) 18.12.2007

## Sun bude nabízet podporu pro OpenOffice.org

Sun Microsystems začne nabízet technickou podporu pro OpenOffice.org. Ta nebude určena přímo pro konečné uživatele, ale pro firmy, které tento produkt distribuují. Cena bude začínat na 20 dolarech za osobu a rok. Podle firmy je OO.org v současnosti používáno desítkami milionů uživatelů. Více na LinuxWorld.  
[Luboš Doležel](#) 18.12.2007

## Italské Ministerstvo ekonomiky zvolilo Red Hat

Italské Ministerstvo ekonomiky se rozhodlo nasadit Red Hat Enterprise Linux na kritických místech. Důvodem je vysoká stabilita a zvýšení výkonu. Red Hat tak údajně pohání např. tamní portál veřejné správy (i když navenek je vidět Microsoft IIS).  
[Luboš Doležel](#) 18.12.2007

## Unreal Tournament 3 server pro Linux

Vyšel Unreal Tournament 3 server pro Linux. Jedná se o samostatný instalátor, který nepotřebuje původní DVD. Klient bohužel stále nevyšel.  
[Luboš Doležel](#) 18.12.2007

## KDE4 RC2+

Již 11. prosince vyšlo KDE4 RC2. Donedávna měla i tato verze místy potíže s použitelností. Pokud jste ovšem dnes aktualizovali (v openSUSE) vývojovou verzi, mohli jste si všimnout, že verze se tiše změnila na RC2+. Tímto upgradem se přidalo mnoho funkčnosti např. do Amaroku, Plasmu, ...  
[David Kolibáč](#) 18.12.2007

## Mozilla Firefox 3 beta 2

Mozilla Corporation vydala Firefox 3 beta 2 v téměř 30 jazykových verzích včetně češtiny. Obsahuje na 900 vylepšení oproti předchozí betaverzi včetně oprav pro stabilitu, výkon,

využívání paměti, vylepšení platformy i uživatelského rozhraní. Ikony, tlačítka a styly nabídek linuxové verze nyní používají motiv GTK. Více podrobností najdete v poznámkách k vydání. Za povšimnutí stojí i nová uvítací stránka po spuštění.

[Petr Tomeš](#)

19.12.2007

## Qt 4.4.0 Technical Preview

Vyšlo Qt 4.4.0 Technical Preview. Integruje do sebe Web-Kit, Phonon, rozšířenou podporu XML, v GraphicsView mohou být widgety, má nový systém nápovědy, IPC framework, concurrency framework a vylepšenou podporu tisku.  
[Luboš Doležel](#) 19.12.2007

## MonoDevelop 1.0 beta 3

Na světě je MonoDevelop 1.0 beta 3. Dokovací okna se v této verzi lépe chovají (mohou se automaticky skrývat), lépe je podporován Mac OS X (prostředí může běžet s ovladačem GTK+ Quartz) a jsou podporovány projekty pro Moonlight.  
[Luboš Doležel](#) 19.12.2007

## Novell: tržby 22 miliónů dolarů za Linux

Ve čtvrtém fiskálním čtvrtletí roku 2007 zaznamenal Novell v oblasti linuxových produktů tržby ve výši 22 miliónů dolarů. Podle vyjádření firmy se na těchto tržbách podepsala také roční spolupráce s Microsoftem. „Zákazníci od nás chtěli, aby Linux lépe spolupracoval s Windows a oni se mohli soustředit výlučně na své podnikání,“ vysvětlil důvod komunitou kritizované spolupráce Jeff Jaffe, výkonný viceprezident a technologický ředitel společnosti Novell ve svém vyjádření pro média.

[Radim Hasalík](#)

19.12.2007

## GIMP 2.4.3

Před třemi dny vyšel GIMP 2.4.3, opraveny byly např. problémy s kódováním v názvech souborů ve WMF import pluginu, byla odstraněna některá nicneříkající varovná hlášení v JPEG a TIFF import pluginech plus drobná oprava v Makefile a další. Mezi úpravami je také aktualizace některých překladů, mj. i českého.

[Petr Vorel](#)

19.12.2007

## Flash 9.0.115.0

Flash 9.0.115.0, který vyšel už počátkem tohoto měsíce, opravuje minimálně devět závažných bezpečnostních chyb. Tyto chyby mohou vést k tomu, aby útočník mohl spustit libovolný kód pod právy prohlížeče.

[Luboš Doležel](#)

19.12.2007

## Creative Commons přichází s novým licencováním CC+

Creative Commons přichází s „licencí“ (protokolem) CC+. Jedná se o kombinaci klasické CC licence s upozorněním, že autor je ochoten dílo poskytnout ke komerčním účelům za určitých podmínek.

[Luboš Doležel](#)

19.12.2007

## V serverových mailových službách dominuje Linux

Nejpoužívanější poštovní servery běží na některé z distribucí Linuxu a obhospodařují tak přes 80 % českého a slovenského Internetu. Tabulku nejpoužívanějšího řešení vede Postfix, pak následuje QMail, Sendmail a Exim. O provedeném průzkumu se dočtete zde.

Milan

20.12.2007

## Obchodní značka Unlocked Media pro DRM-free obsah

Společnost Neuros Technology, výrobce otevřeného media-centra Neuros OSD a MPEG4 rekordéru Recorder 2, ve své tiskové zprávě představila novou obchodní značku a logo „Unlocked Media“, kterou by měla být označována otevřená média (nezatížená žádným DRM). DRM je podle společnosti nejen hrozbou vývoje otevřených produktů, ale ohrožuje i inovaci obecně. Společnost doufá, že spotřebitelé získají lepší možnost rozeznávat DRM-free média a uvědomí si, jaké omezení jejich práv DRM znamená.

\_xm\_

20.12.2007

## Alan Cox o svobodném softwaru

Alan Cox na Red Hat Magazine ve čtyřminutovém videu hovoří krátce o problémech, před kterými dnes stojí svět svobodného softwaru.

Michal Schmidt

20.12.2007

## Amarok 1.4.8

Právě byl uvolněn Amarok 1.4.8. Opět bylo opraveno mnoho závad jako například zatumnutí při aktivním dynamickém přehrávání. Čtení metadat Last.fm nyní funguje i s Xine 1.1.8 a byla vylepšena podpora posledních iPodů.

Luboš Doležel

20.12.2007

## GPG slaví 10 let existence a vychází nové verze

Technologie PGP již dříve oslavila 15 let své existence. Dnes své 10. narozeniny slaví GNU Privacy Guard, svobodná alternativa ke komerčnímu PGP (se kterým je ale na úrovni OpenPGP standardu kompatibilní). Krátký přehled historie GPG si můžete přečíst v oznámení. Zároveň s tím vychází narozeninové verze GPG „jedničkové“ i „dvojkové“ řady – GPG 1.4.8 a GPG 2.0.8 – které opravují některé chyby a přidávají nové funkce. GPG je nadále udržováno v obou vývojových větvích. GPG řady 1 neobsahuje některé nové funkce z řady 2, je ale kompaktnější a bez příliš mnoha závislostí. Snadněji se proto sestavuje a portuje na různé „malé“ platformy.

Michal Růžička

20.12.2007

## Odstávka abclinuxu – výměna disku

V průběhu noci dojde ke kratší odstávce portálu z důvodu výměny poškozeného disku. Děkujeme za pochopení.

Redakce

20.12.2007

## O svobodný Ubuntu notebook není zájem

V červenci se Mark Shuttleworth dotázal komunity, zda by měla zájem o high-end notebook poháněný Ubuntu s výhradně svobodným softwarem. Podle Phoronixu se ukázalo,

že o podobnou věc zájem není – aktivita na wiki i mailing listu je minimální.

Luboš Doležel

20.12.2007

## Provádíme zálohování dat na DV kazety

Na Linux.com vyšel článek popisující způsob, jakým můžete použít videokameru k zapisování dat na DV nebo MiniDV kazety, což se může hodit k zálohování. Samotný postup není zvláště složitý – nevýhodou tak zůstává problémová podpora hardwaru a náklady.

Luboš Doležel

20.12.2007

## Rozhovor s RMS o GPL, Flashi, patentech a svobodě

Internet.com publikovalo rozhovor s Richardem Stallmanem. Jsou v něm probírána obvyklá témata jako čtyři základní svobody, RMS se dále navází do Flashe a patentů a rozebírá princip GNU GPLv3 a postoj Linuse Torvaldse.

Luboš Doležel

20.12.2007

## Ovladače NVIDIA 169.07

Vyšly NVIDIA ovladače 169.07. Rozšiřují výčet podporovaných karet o poslední kousky, zahrnují ovladač CUDA, opravují spoustu problémů, zvyšují výkon akcelerace RENDER, experimentálně podporují 30bitovou hloubku barev a ještě více.

Luboš Doležel

20.12.2007

## Microsoft předá vývojářům Samby dokumentaci

MS předá vývojářům projektu Samba dokumentaci Windows, která jim umožní dále rozšířit funkčnost. Jedná se o výsledek práce Evropské komise, která přinutila Microsoft, aby takovou dokumentaci na požádání poskytoval. Organizace Protocol Freedom Information Foundation, která zde Sambu zastupuje, za tuto dokumentaci zaplatí 10 000 Euro a vývojáři budou muset podepsat NDA.

Luboš Doležel

20.12.2007

## Webové rozhraní Packmanu nyní také česky

Webové rozhraní repozitáře Packman je od dnešního dne dostupné i v české jazykové mutaci. Packman je znám převážně uživatelům distribucí SUSE a openSUSE.

Marek Stopka

20.12.2007

## ATI Catalyst 7.12 (aka 8.44)

Vyšla nová verze ovladačů grafických karet ATI Catalyst 7.12 (resp. 8.44 podle staršího značení). Jsou opraveny úniky paměti, které byly znatelné od přepsání kódu v ovladačích 8.41, byl odstraněn problém s 3D akcelerací na některých AGP kartách, v instalátoru přibyla podpora pro Red Flag DT 6.0 a openSUSE 10.3. Od verze 8.41 jsou toto první ovladače, které oficiálně podporují i FireGL karty. Více podrobností, včetně dalších oprav, najdete v článku na phoronix.com, příp. v poznámkách k vydání.

Jiří Hlinka

21.12.2007

## Matthew Szulik odstupuje z funkce CEO Red Hatu

Matthew Szulik odstupuje z funkce výkonného předsedy Red Hatu. Důvodem jsou zdravotní problémy jeho ženy. Jeho nástupcem se dle Linux-Watch stane Jim Whitehurst – společnost věří, že se i jemu podaří udržet růst tržeb, jaký firma zažívá.

Luboš Doležel

21.12.2007

## RadeonHD 1.1.0, xf86-video-ati 6.7.197

Pro uživatele grafických karet ATI vyšly dva open source ovladače: xf86-video-ati 6.7.197 a RadeonHD 1.1.0. První ovladač opravuje chyby v podpoře EXA, lépe zvládá notebooky a Macy a opravuje různé chyby. Nový RadeonHD zase lépe detekuje monitory, stabilizuje existující funkce, podporuje RS600 a předběžně zvládá RV670.

Luboš Doležel

21.12.2007

## PlayOnLinux – nástroje pro hraní pod Wine

Server linux-gamers.net představuje sadu nástrojů PlayOnLinux. Skládá se ze skriptů v Bashi a Pythonu a umožní vám snadno nakonfigurovat Wine pro potřeby konkrétní hry. V současnosti je podporováno přes 50 her.

Luboš Doležel

21.12.2007

## Norsko volí otevřené formáty: HTML, ODF a PDF

Norsko se rozhodlo jít vstříc otevřeným formátům. Od prvního ledna 2009 tam budou muset všechny vládní úřady používat pro publikování informací HTML, PDF (pokud je nutné zachování rozvržení) a ODF (pokud je nutná možnost editace). Interně bude možné používat i formáty jiné, ale na veřejnost by se měly dostávat pouze tyto.

Luboš Doležel

21.12.2007

## GNU Octave 3.0

Je tomu 11 let co vyšlo GNU Octave 2.0 a nyní přichází verze 3.0. Byla značně zlepšena kompatibilita s Matlabem, všechny interní proměnné byly převedeny na funkce, změnil se způsob práce s vyhledávacími cestami (search paths) a také možnosti nastavování varování.

Luboš Doležel

23.12.2007

## Kubuntu 8.04 nebude LTS

Canonical oznámil, že příští verze Kubuntu 8.04 nebude označena jako LTS (nebude mít dlouhodobou podporu). Vývojáři poté přehodnotili svůj přístup ke KDE4 a zdá se, že výchozí prostředí na lisovaných CD Kubuntu bude KDE4. Obrazy CD s KDE3 budou rovněž dostupné.

Martin „mhb“ Böhm

23.12.2007

## Konec soutěže o logo ArchLinuxu

Právě skončila soutěž o nové logo distribuce ArchLinuxu. Vítězný návrh je k vidění na této stránce.

Ondřej Konečný

23.12.2007

## iptables 1.4.0

Vyšlo iptables 1.4.0. Tato verze znatelně zlepšuje podporu IPv6 díky používání všeobecných xtables. Dále byly opraveny různé chyby.

Luboš Doležel

24.12.2007

## Server Penguin.cz ma problémy

Server Penguin.cz ma existenční problémy po zlyhaní hardveru servera. Každá pomoc s prebytočným HW požadovaných vlastností (P4, 1GB RAM, 2x 100GB) je vítaná.

Michal „feco“ Fecko

25.12.2007

## Trailer hry Penumbra: Black Plague

Na jaře tohoto roku vyšla pro Linux hra Penumbra: Overture, jež slibovala, že svůj hororový příběh rozvine v podobě pokračování. Už je jisté, že bude pouze jedno: Penumbra: Black Plague, které vyjde během prvního čtvrtletí roku 2008. Tento měsíc byl zveřejněn teaser.

Luboš Doležel

26.12.2007

## GLE 4.1.0

Vyšla nová verze grafického skriptovacího jazyku GLE 4.1.0. Oproti předchozí stabilní verzi 3.5 došlo k pročištění kódu a přidání celé řady funkcí. Novinkou je i grafické uživatelské rozhraní QGLE pro přímou editaci zdrojového kódu.

Radek

26.12.2007

## Avidemux 2.4

Vyšla nová verze programu Avidemux, a to 2.4 Final. Z nových vlastností zmiňme podporu MKV a FLV kontejnerů a podporu 3 rozhraní – CLI, GTK+ a Qt4 (které zatím není kompletní).

Goliash

26.12.2007

## Podcast na téma OOXML a GNOME

Zatímco vývojáři prostředí KDE mají k Microsoft OOXML negativní postoj, jeden vývojář GNOME se dokonce účastní procesů ECMA, což pochopitelně vede ke sporům a hádkám. Právě další vyjasnění důvodů této spolupráce se stalo tématem podcastu na Linux.com. Hlavním účelem účasti GNOME je mít potřebný vliv na standard a přístup k informacím, které podpoří vývoj svobodné implementace.

Luboš Doležel

26.12.2007

## Debian 4.0r2 Etch

Vyšel aktualizovaný Debian 4.0r2 Etch. Obsahuje několik desítek bezpečnostních oprav a také opravy chyb.

Luboš Doležel

27.12.2007

## SCO konečně vyřazeno z Nasdaq

Firma SCO je konečně vyřazena z Nasdaq, dnes vstupuje rozhodnutí v platnost. Groklaw navíc vydal shrnutí finanční situace firmy, kde autor článku humorně soudí, že Novell dostane náhrady tak maximálně v hodnotě (či podobě) nábytku.

Luboš Doležel

27.12.2007

## BitTorrent klient Deluge

Linux.com píše o relativně jednoduchém multiplatformním BitTorrent klientu Deluge. Zvládá šifrování komunikace, dá se rozšiřovat pluginy a můžete jej ovládat přes webové rozhraní. Je zhotoven pomocí PyGTK a jeho licenci je GNU GPL.

Luboš Doležel

27.12.2007

## Zenwalk Linux 5.0 Beta

Vyšel Zenwalk Linux 5.0 Beta. Zájemci mohou testovat několik novinek: byl zaveden freedesktop.org HAL, používá se X.org 7.3 (a s tím souvisí mnoho aktualizovaných ovladačů), WiFi Radar byl nahrazen pomocí wicd a součástí distribuce je firmware pro bezdrátová zařízení Intelu.

Luboš Doležel

27.12.2007

## Wine 0.9.52

Přichází Wine 0.9.52. Lépe zvládá grafické tablety, podporuje deskriptory RPC kontextů, opravuje dlouhotrvající potíže s bitovou hloubkou obrazu, přibyl shellový adresář Místa v síti a jako obvykle byly opraveny všelijaké chyby.

Luboš Doležel

28.12.2007

## Benchmark programovacích jazyků

O nejrůznějších testech výkonnosti porovnávacích software či hardware se dočteme denně. Chlapce z Debianu však napadlo benchmarkovat programovací jazyky (resp. jejich implementace). V projektu The Computer Language Benchmarks Game se tak můžete například dočíst, že se na prvním místě umístil jazyk C++ těsně následovaný klasickým C-čkem. Naopak jako poslední jsou Ruby a Prolog.

Petr Tomášek

28.12.2007

## Vietnam by měl upřednostňovat open source software

Vietnamské Ministerstvo informatiky a telekomunikací doporučilo vládním úřadům, aby při výběru softwaru byly upřednostňovány open source produkty. Podle LinuxInsider vydalo ministerstvo seznam vybraného open source softwaru. Zatím však nesdělilo, co vedlo k tomuto činu.

Luboš Doležel

28.12.2007

## CMS Drupal se dočkal komerční podpory

Open source CMS Drupal nabývá na popularitě a proto ani není překvapením, že v USA vznikla firma, která se bude zaměřovat na podporu a vylepšování tohoto produktu. Jmenuje se Acquia a už se jí podařilo získat investice ve výši 7 milionů dolarů. Rozšířenost Drupalu se dle odhadů zdvojnásobuje každý rok.

Luboš Doležel

28.12.2007

## Webový prohlížeč Netscape je oficiálně mrtvý

Webový prohlížeč Netscape je oficiálně mrtvý. V posledních letech se firma AOL pokoušela projekt vzkřísit vydáním verze 8, která uměla používat jak Gecko, tak jádro Internet Exploreru, a později verzi 9, jež nabízela integraci s různými populárními službami. Ani to jej nezachránilo a 1. února 2008

bude ukončena práce na bezpečnostních opravách.

Luboš Doležel

28.12.2007

## Článek o svázanosti OS s hardwarem

Na Digiworldu vyšel článek, který srovnává následky rozsáhlé výměny hardwaru (CPU, základní deska) pro operační systém Linux a Windows.

Nicky 726

29.12.2007

## Škrty v OOXML

Microsoft se rozhodl provést škrty ve specifikaci OOXML. Volby typu 'AutoSpaceLikeWord95', jazyk VML a některé další funkce budou označeny jako 'deprecated' a odsunuty do 'příloh' ke specifikaci, aniž by kdy spatřily světlo ISO světa. Jak kompatibilní budou dokumenty uložené v již existujícím Office 2007 s normou po změnách (bude-li schválena) není prozatím známo.

Jakub Hegenbart

29.12.2007

## Warner bude prodávat hudbu bez DRM

Warner Music Group se zařadila mezi společnosti, které pochopily nesmyslnost DRM a začne americkým zákazníkům na Amazonu nabízet hudbu v nechráněném formátu MP3. Více na BBC News.

Daniel Kvasnička ml.

29.12.2007

## icasi Bible 0.9.16 a 0.10.0

Ke stažení je nová verze icasi Bible 0.9.16 s aktualizací Hesel Jednoty bratrské pro rok 2008 a také dlouho slibovaná nová verze 0.10.0, jejíž rozhraní je postaveno na programu kdialog. Verze 0.10.0 zvládá nejen kódování iso-8859-2, ale také utf-8. V textovém režimu program komunikuje pomocí voleb spuštění a textového výstupu.

Jiří Vičík

30.12.2007

## Kalva 0.8.90

Po dlouhé době vyšla verze 0.8.90 TV programu Kalva. Jedná se o program sloužící k přehrávání a nahrávání TV signálu pomocí mplayeru/mencoderu, umožňuje časované nahrávání, nahrávání seriálů a podobně. Kalva tvoří alternativu k dnes již mrtvému xdtv. Sice zdaleka neposkytuje tolik možností, ale vynahradí to svou jednoduchostí a nenáročností.

Marek Janata

31.12.2007

## Ovladač Radeon podporuje TV-out u R500

Open source ovladač Radeon nyní podporuje TV-out u grafických karet řady R500. Najdete ji včetně detekce TV standardu v git repozitáři xf86-video-ati. Pokud máte o tuto funkci zájem, testujte a hlase nalezené chyby.

Luboš Doležel

31.12.2007

## Technologická ředitelka OLPC opouští post

Technologická ředitelka projektu OLPC Mary Lou Jepsen oznámila, že opouští projekt a bude se věnovat dalšímu (komerčnímu) rozvoji technologie displejů a jiných úsporných

součástí notebooků. S projektem však bude i nadále spolupracovat. Více na [ITworld.com](http://ITworld.com).

[Luboš Doležel](#)

31.12.2007

### Spicebird: kombinace Thunderbirdu a Lightningu

Mozilla Links v krátkosti představuje program Spicebird. Ten je postavený na Mozilla Thunderbirdu a Lightningu a vypadá to, že integruje i rozšíření SamePlace pro komunikaci přes Jabber. Výsledkem je docela zajímavé dílo – podívejte se na

prezentaci.

[Luboš Doležel](#)

31.12.2007

### Životní příběh prohlížeče Netscape Navigator

Na webu ComputerWorldu vyšel životní příběh prohlížeče Netscape Navigator, jehož vývoj byl firmou AOL nedávno ukončen. Příběh začíná v době, kdy se prohlížeč jmenoval Mosaic Netscape, pokračuje obdobím Netsite Web serveru a přes vznik Mozilly se dostává až do současnosti.

[Luboš Doležel](#)

31.12.2007

■