

DESKTOP A JÁDRO LINUXU

Jan Kasprzak

E-MAIL: KAS@FI.MUNI.CZ

Klíčová slova: Linux, desktop, SysFS, freedesktop.org, hal, d-bus, Avahi, zeroconf, suspend

Abstrakt

Operační systém UNIX nebyl původně navrhován jako desktopový systém. Linux, který do značné míry přebírá API i další vlastnosti UNIXu, dnes ale najdeme na široké škále systémů – od mobilních telefonů přes desktopy až po superpočítače z TOP 500. V přednášce se zaměříme právě na desktopové nasazení Linuxu, a to zejména na technologie, kterými se Linux liší od jiných (i UNIXu podobných) systémů nebo které jsou jinak blízko jádru samotnému.

Abstract

The UNIX operating system has not originally been designed as a desktop system. Linux, which more or less copies the API and other properties of UNIX, can currently be found on various devices ranging from cell phones to desktops to the supercomputers of the TOP 500 list. In this paper, we will focus on Linux installed on desktops and especially on the technologies which make it different from other (even UNIX-like) systems as well as those that are somehow close to the kernel itself.

1 Desktop? Je to vůbec zajímavé?

V současné době je Linux pravděpodobně nejflexibilnější operační systém. Jádro Linuxu přeložené z jedněch a těch samých zdrojových textů, s tím stejným aplikačním rozhraním, najdeme na hardwaru od mobilních telefonů¹ až po masivně paralelní superpočítače ze seznamu TOP 500². Každá z těchto oblastí nasazení Linuxu má svá specifika: od vývoje speciálních souborových systémů pro nerotační média (například flash paměti) až po snahu o maximální paralelizaci různých částí jádra včetně úplného vyloučení zamykání³. Řada z těchto snah má nečekaný přesah z původní oblasti nasazení do jiné třídy počítačů.

Z hlediska operačního systému lze za nejzajímavější oblast nasazení považovat právě desktopové počítače. Malá zapouzdřená zařízení často zpracovávají jen jednu jednoúčelovou aplikaci, a od operačního systému kromě správy hardwaru a možná komunikace po síti vyžadují jen jedno – aby se hlavní aplikaci pokud možno co nejméně pletl do cesty. Podobně je tomu někdy i u velkých počítačů, často zpracovávajících jediný program – databázový stroj (který je sám malým operačním systémem včetně správy paměti a diskových bufferů).

Naproti tomu na desktopovém počítači typicky běží desítky až stovky nejrůznějších procesů s velmi odlišnými nároky na systém (multimediální přehrávače, webové prohlížeče, správci souborů, prohlížeče a konvertory dokumentů, atd.). Desktopové počítače také aktivně komunikují s největší škálou zařízení: od blokových zařízení, připojených přes nejrůznější sběrnice velmi odlišných rychlostí a vlastností (IDE, SATA, SCSI/SAS/FC, USB, IEEE 1394) přes hardwarově akcelerované grafické a zvukové karty, až po síťová zařízení (od Ethernetu přes PPP a WiFi až po Bluetooth), která ještě ke všemu mohou v systému dynamicky vznikat a zanikat. Linus Torvalds říká⁴:

What makes the desktop so interesting is in fact that it shows more varied usage than any other niche.

V tomto článku se zaměříme na to, co jádro Linuxu poskytuje desktopovým aplikacím navíc oproti jiným UNIXovým systémům. Dále pak na nástroje, které primárně vznikly pod Linuxem (i když mnohdy jsou přenositelné i na jiné systémy), a mají za cíl zlepšit propojení desktopového prostředí se zbytkem operačního systému.

¹Například <http://www.openmoko.org>.

²Podle <http://www.top500.org/stats/list/29/osfam> běží pod Linuxem 77.8% všech superpočítačů z aktuálního TOP 500.

³Například metoda Read-Copy-Update: <http://lse.sourceforge.net/locking/rcupdate.html>

⁴<http://lkml.org/lkml/2007/7/28/106>

2 Správa zařízení v Linuxu 2.6

2.1 Exkurze do historie

UNIX již tradičně rozlišuje dvě hlavní třídy zařízení: bloková zařízení (například disky; vyznačují se možností přímého/náhodného přístupu a obvykle bloky pevné délky) a znaková zařízení (pásky, tiskárny, terminály a podobně). Zařízení je charakterizováno svým *speciálním souborem* v adresáři `/dev`, a ten zase svým *hlavním a vedlejším číslem*:

```
$ ls -l /dev/sdb  
brw-r----- 1 root disk 8, 16 Aug 23 17:05 /dev/sdb
```

Zde jde o blokové zařízení (**b** na začátku výpisu) `/dev/sdb` s hlavním číslem 8 a vedlejším číslem 16. Hlavní číslo se používá pro adresaci ovladače (driveru) uvnitř jádra a vedlejší číslo pro rozlišení mezi zařízeními, spravovanými tímto ovladačem (zde ovladač SCSI disků).

Pamětníci jistě vzpomenu, jak fungovala správa zařízení na UNIXu v7 a systémech z něj odvozených: jádro mělo v sobě zakompilovaná dvě statická pole – tabulky `bdevsw[]` a `cdevsw[]`. V těchto polích (indexovaných hlavním číslem zařízení) měl každý ovladač uvedeno, která jeho funkce – dnes bychom objektivě řekli *metoda* – se má zavolat při pokusu o otevření zařízení, čtení, zápisu a dalších operacích.

2.2 Kde je problém?

Výše uvedený přístup je do značné míry problematický:

- Jak velké má být `dev_t`? Norma POSIX říká, že hlavní a vedlejší číslo by se mělo dohromady vejít do jednoho numerického typu. UNIX používal 8 bitů na hlavní a 8 bitů na vedlejší číslo zařízení. To ovšem ne vždy dostačuje.
- Co všechno mít v `/dev`? Tradičně UNIX měl v adresáři `/dev` speciální soubory pro většinu možných zařízení, i když na daném počítači se některé typy zařízení jistě nevyskytnou (starší distribuce Linuxu měly například poctivě vytvořené soubory `/dev/xd*` pro MFM/RLL disky, které byly naposledy ve větší míře k vidění někde na IBM PC-XT).
- Jak vlastně zařízení adresovat? Linux obvykle používal označení podle pořadí detekce: `/dev/sda` byl první SCSI disk, `/dev/sdb` druhý atd. Když pak zmizel nebo byl přidán disk někam doprostřed, označení části disků se posunulo. Naproti tomu Solaris používá jména typu `/dev/dsk/c0t7d3s0` podle topologie (číslo SCSI sběrnice, SCSI ID, SCSI LUN, číslo oblasti). Máme-li ale v systému třeba jen jeden disk a přepojíme jej na jiný řadič,

začne se najednou jmenovat jinak. Na desktopových systémech je to ještě horší: co když vezmeme IDE disk (na Linuxu dosud známý jako `/dev/hdc`), zabalíme jej do USB nebo IEEE 1394 rámečku a připojíme jej k počítači? Je to ten stejný disk a měl by se tedy jmenovat stejně?

- Dynamicky vznikající zařízení. Tato zařízení byla do jisté míry i na starších UNIXech – vzpomeňme na pseudoterminály a adresář `/dev` plný souborů jako `ttypN`, `ptypN`, `ttyqN`, atd. Jak se má jmenovat nově zapojený USB flash disk nebo Bluetooth klávesnice? Jaká má mít přístupová práva?

2.3 DevFS – slepá cesta

V Linuxu řady 2.4 se objevil virtuální souborový systém DevFS, jehož idea byla pravděpodobně převzata ze Solarisu. Ovladače samy registrovaly v DevFS speciální soubory podle toho, která zařízení zrovna viděly v systému. Systém DevFS se ovšem v reálných distribucích příliš neuchytil (s čestnou výjimkou Gentoo Linuxu), protože přinášel řadu problémů, mezi nimi tyto:

- Politika uvnitř jádra. DevFS vnucoval uživateli svoje pojmenování zařízení (u disků bylo podobné tomu na Solarisu) a nebyla možnost toto změnit.
- Okrajové stavy. DevFS s sebou nesl spoustu *race conditions* mezi otevíráním zařízení a paralelním rušením jeho ovladače z jádra.
- Přístupová práva: speciální soubor v DevFS prostě vznikl, aniž by správce systému jakkoli mohl ovlivnit, s jakými přístupovými právy se vytvoří. Přístupová práva navíc nebyla perzistentní mezi restarty systému.

2.4 První část řešení – SysFS

V Linuxu 2.6 byl DevFS prohlášen za zastaralý a skupina programátorů v čele s Gregem Kroah-Hartmannem začala pracovat na alternativě. Výsledkem byl nový model správy zařízení (všech – nejen blokových a znakových, ale také ovladačů sběrnic, síťových zařízení a dalších). Zařízení jsou zde evidována podle topologie (připojení na sběrnici), třídy (disk, vstupní zařízení, atd.) ovladačů, hlavních a vedlejších čísel a dalších atributů.

Datové struktury nového modelu jsou navíc přehledně zpřístupněny v novém virtuálním souborovém systému SysFS, obvykle připojeném na adresář `/sys`. Kromě základních atributů jako je topologie obsahuje SysFS i možnosti provádět některé akce pomocí čtení nebo zápisu hodnot z/do příslušných virtuálních souborů:

```
# echo 0 0 0 > /sys/class/scsi_host/host0/scan
# echo 1 > /sys/block/sdb/device/delete
```

```
# echo 45 > /sys/devices/platform/w83627hf.656/temp2_max
# cat /sys/class/input/input0/name
Power Button (FF)
# cat /sys/bus/pci/devices/0000:00:0f.1/vendor
0x1106
```

Správce systému tedy v SysFS má kompletní inventář hardwaru včetně typů zařízení, topologie, atributů jako výrobce, výrobní číslo a podobně. Navíc většina ovladačů byla upravena tak, že své hlavní číslo registrují dynamicky, a tedy na jednom systému může koexistovat daleko více zařízení, než by mohlo při statickém přidělování hlavních čísel centrální autoritou.

2.5 Druhá část řešení – hotplug

Jeden z problémů evidence zařízení je, že na desktopových systémech zařízení dynamicky vznikají a zanikají. Jádro přirozeně aktuální stav stále zobrazuje uvnitř SysFS, nicméně by bylo dobré nemuset pořád sledovat změny a být o nich informován až podle potřeby.

Jádro Linuxu 2.6 poskytuje dva mechanismy: starší `hotplug` funguje tak, že jádro při vzniku nebo zániku zařízení spustí externí program `/sbin/hotplug`, kterému předá informace o příslušném zařízení.

Pro větší frekvenci vzniku a zániku zařízení jádro poskytuje sofistikovanější notifikační mechanismus. Jde o frontu událostí, postavenou nad virtuálními sockety rodiny `AF_NETLINK`.

2.6 Vše pak propojí udev

Nyní už je vidět, že jádro dává správci systému nejen potřebné informace o aktuálním stavu zařízení v systému, ale i notifikační mechanismy, informující o změně. S těmito informacemi je už možné implementovat správu adresáře `/dev` v uživatelském prostoru.

Jádro ovšem zůstává zpětně kompatibilní se statickým `/dev`: nejpoužívanější zařízení mají i nadále přidělené své hlavní číslo staticky, a tak vytvoříme-li si ručně (nebo z distribuce) v `/dev` všechny potřebné soubory, může systém i nadále fungovat postaru.

Jedním ze systémů, využívajících dva výše uvedené mechanismy pro dynamickou správu adresáře `/dev`, je balík `udev`. S jeho pomocí můžeme mít adresář `/dev` buďto jako dříve přímo na disku, anebo jej přesunout na ramdisk (souborový systém `TmpFS`) a vytvářet speciální soubory dynamicky, vycházejíce z prázdného adresáře `/dev`.

Systém `udev` má dvě fáze činnosti: ta první, nazývaná *coldplug* inventarizuje stávající zařízení a vytváří jim speciální soubory v `/dev`. Ve druhé fázi pak již

démon `udev` zpracovává *hotplug* události od jádra a vytváří, resp. ruší speciální soubory nově objevených se, resp. právě odpojených zařízení.

Podstatným přínosem systému `udev` je, že politiku (pojmenování zařízení, přístupová práva a podobně) přenáší do uživatelského prostoru do svých konfigurací. Například: líbí se mi pojmenování disků ze Solarisu, ale chci, aby se myš v mém laptopu jmenovala vždy `/dev/mouse` bez ohledu na to, jakou myš použiji a do kterého USB portu na zadní straně laptopu se zrovna poslepu trefím.

Systém `udev` dokonce umožňuje i správci systému nastavit reakce na vznik zařízení typu vyvolání externího skriptu nebo notifikace přes D-Bus (viz dále).

2.7 Příklady konfigurace udev

Chování `udev` lze nastavovat pomocí sad pravidel, umístěných do souborů v adresáři `/etc/udev/rules.d`:

```
# PS/2 mys
SYSFS{protocol}=="ImExPS/2", \
    KERNEL=="mouse*", \
    SYMLINK+="input/mouse_first"
# USB mys
SYSFS{name}=="Logitech USB-PS/2 Optical Mouse", \
    KERNEL=="mouse*", \
    SYMLINK+="input/mouse_second"
# PS/2 klavesnice
SYSFS{name}=="AT Translated Set 2 keyboard", \
    SYSFS{phys}=="isa0060/serio*/input*", \
    KERNEL=="event*", \
    SYMLINK+="input/keyboard_first"
# USB klavesnice
KERNEL=="event*", \
    SYSFS{./name}=="HID 1267:0103", \
    SYSFS{bInterfaceNumber}=="00", \
    SYMLINK+="input/event31"
```

Příklad konfigurace vstupních zařízení pro víceuživatelský desktop: PS/2 klávesnice a myš jsou zpřístupněny jako `/dev/input/keyboard_first` a `mouse_first`, USB klávesnice a myš jako `event31` a `mouse_second`.

```
KERNEL=="ttyUSB*", \
    ATTRS{product}=="Papouch TMU Thermometer", \
    ATTRS{serial}=="PPQ3NTMG", \
    SYMLINK+="tmu0"
```

Zpřístupní USB teploměr daného sériového čísla (viditelný dosud jako USB sériový port) pod jménem `/dev/tmu0`.

```
KERNEL=="sd*1", \
SYSFS{model}=="G3", \
SYSFS{vendor}=="M-System", \
RUN+="/usr/local/sbin/zpravy-to-palm"
```

Zkopíruje po připojení handheldu Palm v režimu USB mass storage do jeho paměti aktuální OGG soubor s poslední zpravodajskou relací oblíbené rozhlasové stanice.

2.8 Psaní pravidel pro udev

Asi nejjednodušší cestou jak zjistit, podle čeho vlastně lze v pravidlech pro `udev` vybírat a vyhledávat naše zařízení, je zařízení připojit, podívat se pod jakým jménem je aktuálně viditelné v `/dev` a spustit následující příkaz:

```
# udevinfo -a -p 'udevinfo -q path -n /dev/mojezařizeni'
```

Upozornění: ve výše uvedeném příkazu se jedná o zpětné apostrofy.

Z vypsaných atributů pak vybrat ty, které považují za charakteristické pro dané zařízení a můj záměr: někdy mě například zařízení zajímá podle svého sériového čísla nebo identifikace výrobce (ať už je zapojeno kamkoli a přes kteroukoli sběrnici), někdy naopak podle místa kam je zapojeno nebo podle pořadí objevení v systému, a tak podobně.

Podrobnější informace o psaní pravidel pro `udev` lze najít v článku *Writing udev rules*⁵.

2.9 udev a disky

Systém `udev` má jednu specialitu pro disky: pro většinu scénářů použití má připravené symbolické linky v adresáři `/dev/disk`:

- `/dev/disk/by-id/scsi-SATA_WDC_WD2500JB-32WD-WMAEP1078446` je pojmenování podle výrobce a sériového čísla.
- `/dev/disk/by-path/pci-0000:00:0f.1-scsi-0:0:0:0` – podle topologie nadřazených sběrnic.
- `/dev/disk/by-uuid/7056dad9-8837-485d-bdbc-88225b76c678` – pojmenování diskové oblasti podle UUID souborového systému na ní. Zůstává stejné i po `dump(8)` a `restore(8)` například po havárii disku.

⁵<http://reactivated.net/writing-udev.rules.html>

3 HAL a evidence zařízení

Dalším typicky desktopovým problémem je evidence dostupných zařízení, jejich vlastností, schopností a atributů. Například program pro archivaci fotografií řeší úkol „najdi na tomto počítači připojený fotoaparát“, ať už je připojený jako USB nebo IEEE 1394 mass storage zařízení (aktuálně přimontované nebo ještě nepřipojené) nebo jako USB zařízení ovládané nějakým proprietárním protokolem (například fotoaparáty Canon).

Podobně program pro vypalování CD nebo DVD chce zjistit nejen jaké CD a DVD mechaniky v systému jsou, ale také potřebuje vědět, které z nich umí samy nahlásit, že v nich je médium, a které naopak je třeba periodicky po několika vteřinách na přítomnost média dotazovat.

A desktopové prostředí potřebuje vědět, jak jsou na tomto konkrétním laptopu pojmenovány všechny rozšiřující klávesy a jaké vysílají scan kódy.

Řešením tohoto problému pro Linux na desktopu je HAL⁶, jeden ze standardů `freedesktop.org`. Je původně psán pro Linux, ale postupně se jeho hardwarově specifická strana implementuje i pro jiné systémy, jako je FreeBSD a Solaris.

Středem systému HAL je démon `hal`. Běží na pozadí a eviduje zařízení, která v systému jsou. Mimo jiné je také napojen na mechanismus `udev`.

Dále má HAL sady skriptů pro neobvyklá zařízení (například zamykání dvířek některých méně obvyklých CD mechanik) a databázi zvláštností jednotlivých zařízení.

Výpis databáze HAL lze získat například programem `lshal(1)`. Vyhledávání je možné pomocí programů `hal-find-by-capability(1)` a `hal-find-by-property(1)`. Bližší informaci o již nalezeném zařízení lze zjistit programem `hal-get-property(1)`:

```
$ hal-find-by-capability --capability storage.cdrom
/org/freedesktop/Hal/devices/storage_model_DVDRAM_GSA_4160B
$ hal-get-property --udi \
  /org/freedesktop/Hal/devices/storage_model_DVDRAM_GSA_4160B \
  --key storage.cdrom.write_speeds
7056 5645 4234 2822 1411 706
$
```

Dalším využitím HAL je správa přístupových práv k zařízením: HAL může evidovat přihlášené uživatele a povolit například přístup ke zvukové kartě těm uživatelům, jejichž sezení je u systémové konzoly. Další možností je namapování sady zařízení na konkrétní sezení: například informace o tom, která zvuková karta přísluší uživateli pracujícímu v X11 na AGP grafické kartě, a které jinému uživateli, sedícího u monitoru od PCI grafické karty. HAL v tomto případě používá modifikaci přístupových práv přes Access Control Listy.

⁶<http://hal.freedesktop.org/>

4 Zaslání zpráv v desktopových prostředích

UNIXový model systému, kde je jádro striktně oddělené od aplikací, a kde X server je jen jedna z mnoha aplikací, přináší jisté problémy pro uživatelskou příjemnost desktopového systému. Pamětníci možná vzpomenou na podobný problém ve starších verzích grafických nadstaveb DOSu, kdy při chybě diskety byl uživatel přepnut zpět do textového režimu a zobrazena obligátní výzva

```
[A]bort, [R]etry, or [F]ail?
```

UNIX má podobný problém: jádro je „příliš daleko“ od desktopových aplikací a nemá jak jim sdělit informace typu hardwarové chyby nebo i jen vzniku nového zařízení (na což by desktopové prostředí mohlo reagovat například vytvořením nové ikony na ploše uživatele).

Iniciativa `freedesktop.org` vytvořila pro tyto účely nový standard, komunikační rozhraní D-Bus⁷, tedy Desktop Bus.

D-Bus je systém zaslání strukturovaných zpráv (ve formátu XML) mezi procesy. Obvyklé použití je v rámci jednoho systému, ale D-Bus podporuje i komunikaci nad TCP/IP. Komunikace může být buďto jeden k jednomu (běžné zaslání zpráv) nebo jeden k mnoha (publikuj/přihlas se k odběru). D-Bus může sloužit k aplikačním oznámením o změnách v systému. Tyto informace by jinak aplikace musely složité získávat periodickým dotazováním nebo by v horším případě nebyly dostupné vůbec. Dalším využitím D-Bus je objektový přístup, kdy přes D-Bus lze vytvářet určité objekty a komunikovat s nimi zasláním zpráv. Příkladem takového objektu může být třeba textový editor. Do jisté míry je technologie D-Bus podobná dalším systémům pro strukturované objektové zaslání zpráv (CORBA, COM, DCop a další)⁸.

V běžném systému obvykle najdeme dvě nezávislé instance D-Bus:

- Systémová sběrnice. Je spuštěna po startu systému a běží po celou dobu existence systému. Sem posílají zprávy aplikace, nezávislé na uživatelově sezení: například zpráva `udev` o vzniku nového zařízení nebo zpráva `HAL` o připojení dalšího svazku. Desktopová prostředí pak mohou tyto zprávy zpracovávat a například podle typu zařízení zobrazit ikonku nového svazku na ploše nebo spustit stahování fotek z fotoaparátu.
- Uživatelská sběrnice. Vzniká při startu desktopového prostředí a každé sezení má tuto sběrnici samostatnou. Slouží pro zaslání zpráv v rámci sezení (změna globálního nastavení, například typ a velikost fontu; objektové použití, například vyvolání nového okna textového editoru).

⁷<http://dbus.freedesktop.org/>

⁸Srovnání podrobněji ve FAQ: <http://dbus.freedesktop.org/doc/dbus-faq.html>

Dalšími uživateli D-Bus jsou například následující projekty:

- `libnotify`⁹ – systém zobrazování krátkodobých informativních hlášení uživateli desktopového sezení (viz též `notify-send(1)`).
- `Galago`¹⁰ – knihovna a démon pro sledování prezenze uživatele. Využívají IM a VoIP klienti pro globální sledování, je-li uživatel zrovna u počítače, je-li aktivní (nebo například zaneprázdněný VoIP hovorem), atd.

5 Víceuživatelský desktop

Jednou z vlastností, kde se pozitivně projevuje modularita UNIXových systémů, je víceuživatelský desktop. UNIX je již od základu systém víceuživatelský, s individuálními nastaveními (například jazykovými) pro každého uživatele. Proto myšlenka na rozšíření tohoto pohledu o více uživatelů, současně pracujících *v grafickém sezení* na tomtéž počítači je jen dalším přirozeným krokem v řadě.

5.1 Více grafických karet

Víceuživatelský desktop s sebou nese několik problematických oblastí. tou největší je asi vzájemná kompatibilita více grafických karet v jednom počítači. Všechny grafické karty standardu VGA totiž mapují svoje VGA registry a sdílenou paměť do adresního prostoru mezi 640 KB a 1 MB. Při startu systému je BIOSem takto nainicializována jen primární grafická karta, což je v pořádku. Při startu druhého X serveru (pro sekundární grafickou kartu) se ale může stát, že vinou ovladače sekundární grafiky nebo i samotného hardwaru začne být i druhá grafická karta mapovaná do VGA oblasti a dojde ke konfliktu. Naštěstí většina nových grafických karet již tento problém nemá, a kromě textového režimou se obejde zcela bez VGA registrů a VGA sdílené paměti.

X server X.org se tradičně chová tak, že po startu projde všechny sběrnice počítače, a zakáže všechny grafické karty kromě té, se kterou právě pracuje. Toto je pro víceuživatelské prostředí nežádoucí. Podobně je třeba vyřešit, který z několika paralelně běžících X serverů se bude starat o přepínání virtuálních konzol. Příslušné přepínače X serveru se jmenují `-isolateDevice`, `-sharevts` a `-novtswitch`. Po zjištění PCI adres jednotlivých grafických karet (použijte příkaz `lspci(8)`) může konfigurace display manageru GDM vypadat například takto:

```
[servers]
0=Prvni
```

⁹<http://trac.galago-project.org/wiki/DesktopNotifications>

¹⁰<http://www.galago-project.org/>

1=Druhy

```
[server-Prvni]
name=Prvni X server
command=/usr/bin/Xorg vt7 -layout ATiLayout \
-isolateDevice PCI:01:00:0
flexible=true
```

```
[server-Druhy]
name=Druhy X server
command=/usr/bin/Xorg vt7 -layout RivaLayout \
-isolateDevice PCI:00:19:0 \
-sharevts -novtswitch
flexible=true
```

Podobně v konfiguraci X serveru (`xorg.conf`) musí mít každá grafická karta vyznačenou svoji PCI adresu:

```
Section "Device"
    Identifier "ATI"
    Driver      "radeon"
    BusID       "PCI:1:0:0"
EndSection
```

```
Section "Device"
    Identifier "Riva"
    Driver      "nv"
    BusID       "PCI:0:19:0"
EndSection
```

5.2 Více vstupních zařízení

Současné počítače stále ještě mají vstup pro klávesnici a myš PS/2 portem. Nejjednodušší řešení vstupních zařízení pro víceuživatelský desktop tedy je mít jedno pracovní místo ovládané z PS/2 klávesnice a myši a druhé místo z USB klávesnice a myši. Jen je potřeba jednotlivá zařízení od sebe rozpoznat i v případě odlišného pořadí objevování zařízení. V kapitole o systému `udev` jsme si naznačili, jak se tohle dělá.

Linux poskytuje pro každé vstupní zařízení tzv. *event interface*, speciální soubor `/dev/input/eventN`. Každá vstupní událost (například stisk nebo uvolnění klávesy, pohyb myši) je jádrem reportovaná jako balík dat pevné délky, který lze z event interface přečíst. Grafický subsystém X.org obsahuje ovladač `evdev`,

který umí právě tyto události rozpoznávat a přeposílat X klientům. Pro jednoduchost nepoužijeme pro PS/2 klávesnici vestavěný driver X.org, ale i zde použijeme event interface. Konfigurace by v souboru `xorg.conf` by mohla vypadat nějak takto:

```
Section "InputDevice"
    Identifier "PS2Keyboard"
    Driver     "evdev"
    Option     "XkbModel" "evdev"
    Option     "Name" "AT Translated Set 2 keyboard"
#    Option     "Device" "/dev/input/event31"
EndSection
```

Ve výše uvedeném je skryta jedna neobratnost ovladače `evdev` – ovladač se snaží rozpoznávat zařízení na sběrnici sám (podobně jako to dělá `udev`). Proto zde nemáme cestu ke speciálnímu souboru, ale textové jméno (stejně je v attributech zařízení v SysFS).

Pro myši je situace ještě jednodušší: Linux poskytuje pro každou myš kromě event interface také zařízení `/dev/input/mouseN`, které používá protokol kompatibilní s myší IMPS/2. Příslušná část `xorg.conf` bude následující:

```
Section "InputDevice"
    Identifier "PS2Mouse"
    Driver     "mouse"
    Option     "Protocol" "IMPS/2"
    Option     "Device" "/dev/input/mouse_first"
    Option     "Buttons" "5"
    Option     "ZAxisMapping" "4 5"
EndSection
```

5.3 Dokončení konfigurace

Nyní zbývá už jen v `xorg.conf` definovat příslušné sekce pro monitory, spojit je spolu se sekcemi `Device` do sekcí `Display` (zde se nastavují věci jako podporovaná rozlišení a bitové hloubky) a to vše na závěr spojit se vstupními zařízeními do popisu rozložení obrazovek, tedy sekce `ServerLayout`:

```
Section "ServerLayout"
    Identifier "ATiLayout"
    Screen 0 "ATI+LGLCD" 0 0
    InputDevice "PS2Mouse" "CorePointer"
    InputDevice "PS2Keyboard" "CoreKeyboard"
EndSection
```

```
Section "ServerLayout"
    Identifier      "RivaLayout"
    Screen          0  "Riva+PhilipsLCD"  0  0
    InputDevice     "USBMouse"  "CorePointer"
    InputDevice     "USBKeyboard" "CoreKeyboard"
EndSection
```

6 Správa síťových rozhraní

Připojení k síti přináší na desktopových a zejména mobilních systémech řadu problémů, které jsme z UNIXových serverů dosud neznali. Například konfigurace sítě by měla být co nejméně automatická, nanejvýš s částečnou asistencí právě přihlášeného uživatele (ne nutně superuživatele). Uživatel by měl mít možnost vybrat si z několika slyšitelných WiFi sítí, nejlépe tak, aby zároveň viděl například i informaci o síle signálu jednotlivých dostupných sítí. Podobně konfigurace VPN, šifrovaného přístupu a podobně.

Informace o aktuálním stavu síťového připojení a jeho vlastnostech by ale měla také být k dispozici i ostatním uživatelovým aplikacím: webový prohlížeč se například může na základě typu připojení rozhodnout nestahovat obrázky, uživatel může chtít automaticky spouštět svého oblíbeného klienta pro sdílení multimediálních dat jen v určité síti, a tak podobně.

Aplikace která řeší výše uvedené problémy pod Linuxem se jmenuje Network-Manager¹¹. Jedná se o systémového démona, který přes HAL zjišťuje informace o síťových zařízeních a přes D-Bus komunikuje s uživatelským appletem a dalšími aplikacemi, které se o aktuální stav síťového připojení zajímají.

7 Uspávání a hibernace

Pro přenosné počítače a další mobilní zařízení je klíčová schopnost ušetřit napájení tím, že se počítač do jisté míry deaktivuje. Obvykle rozeznáváme dva druhy deaktivace: v prvním (*suspend*) je vypnuto vše kromě paměti RAM počítače a několika málo dalších komponent. Druhý (*hibernate*) znamená úplně vypnutý počítač. Pro pozdější obnovení stavu je u hibernace uložen obsah paměti RAM a další informace na disk počítače.

¹¹<http://www.gnome.org/projects/NetworkManager/>

7.1 Suspend

Suspend (uspání do RAM) je složitější ze dvou způsobů uspávání. Vyžaduje součinnost firmwaru počítače (ACPI) a nese s sebou další problémy, zejména při opětovné aktivaci dříve vypnutých zařízení. Asi nejproblematičtější bývají grafické karty, protože ty za normálních okolností bývají po zapnutí počítače inicializovány přímo BIOSem. Proces probouzení z RAM proto může vyžadovat některé úpravy podle typu hardwaru a chování BIOSu¹².

7.2 Hibernace

Hibernace je jednodušší. Funguje zjednodušeně řečeno tak, že se vytvoří obraz paměti systému, uloží na disk (do odkládacího prostoru), a při obnovení se naboootuje docela obyčejné jádro, které běžným způsobem nainicializuje svá zařízení, ale pak místo startu uživatelského prostoru načte a obnoví obraz paměti z odkládacího prostoru.

7.3 Balík pm-utils

Aktuální vývoj ve správě hibernace a suspendu je balík `pm-utils`¹³. Sestává se z hlavního skriptu `pm-action`, volaného pomocí symlinku jako `pm-suspend` nebo `pm-hibernate`. Tento skript spouští jednotlivé úlohy, které je třeba provést před usmáním počítače (pro hibernaci například aktivace stejného jádra v boot loaderu, jako je právě běžící jádro, atd.). A podobně má i skripty pro obnovení provozu po probuzení.

Program `pm-action` je volán obvykle z HAL na pokyn některého desktopového appletu (`gnome-power-manager` nebo `kpowersave`).

8 Sdílení zvukového subsystému

Problematika zvukového hardwaru je z hlediska jádra dostatečně zvládnuta, jádro zpřístupňuje vlastnosti zvukového hardwaru do uživatelského prostoru. Problém ovšem je v přístupu více aplikací ke zvukové kartě. Ne všechny zvukové karty podporují hardwarové mixování více zvukových proudů, a tak se může stát, že pokud zrovna přehráváte video, nikdo se nedovolá na vašeho VoIP klienta, protože neuslyšíte vyzvánění.

Situaci se snaží řešit systém PulseAudio¹⁴, dříve známý pod názvem Polyp-Audio. Jde o přenesení zprávy zvukového zařízení do samostatného démona a úpravu aplikací (aplikačních knihoven) tak, aby místo přímého přístupu k zařízení

¹²<http://people.freedesktop.org/~hughsient/quirk/quirk-suspend-try.html>

¹³<http://en.opensuse.org/Pm-utils>

¹⁴<http://www.pulseaudio.org/>

komunikovaly s tímto démonem. Podobný přístup známe z projektů Esound a aRts (oba jsou dnes již nevyvíjené).

PulseAudio je reimplementací aplikačního rozhraní Esound s tím, že obslužný démon má modulární architekturu a daleko více možností. Kromě Esound protokolu poskytuje i nativní protokol včetně práce nad TCP/IP. Takže například je možno výstup zvukové aplikace posílat po síti na jiný počítač.

PulseAudio řeší věci jako softwarové mixování, ale také nastavení hlasitosti: pro každý proud zvukových dat se eviduje název (daný aplikací nebo právě přehrávanými daty) a k názvu se eviduje naposledy použitá hlasitost. Takže můžete mít jinou hlasitost pro VoIP klienta a jinou pro přehrávač hudby, aniž byste museli měnit nasavení hlasitosti pokaždé, když dotelefonujete.

Systém PulseAudio dokonce umí definovat akce pro jednotlivé proudy zvukových dat, takže například je možno ztlumit přehrávač hudby při příchodu VoIP hovoru a podobně. Systém umožňuje funkci i bez trvale spuštěného démona tak, že instanci démona spouští ta aplikace, která poprvé v rámci sezení chce používat zvukové zařízení. Uživatel navíc může existující proudy dat přesměřovat na různá zařízení nebo i po síti.

PulseAudio je asi nejvíc přenositelnou komponentou ze všech v tomto textu zmiňovaných. Funguje kromě Linuxu například i na Solarisu, FreeBSD a dokonce existuje i nativní (ne Cygwin) port pro Windows.

9 Informace o zdrojích v síti

Jednou z často vyžadovaných vlastností desktopových systémů je, aby takovéto systémy fungovaly bez centrální evidence nebo centrálních serverů. Aby se dokázaly domluvit dva počítače jen na základě toho, že jsou na stejné síti.

Jedním z protokolů pro takovéto bezkonfigurační (zeroconf) komunikace je multicast-DNS (mDNS) a DNS service discovery (DNS-SD). Oba tyto protokoly jsou na desktopech pod Linuxem implementovány balíkem Avahi¹⁵.

Pomocí Avahi mohou aplikace vidět okolní počítače jménem i bez DNS a dokonce i bez centrální autoritou přidělené IP adresy. Avahi ale také poskytuje informace o službách na daných počítačích běžících, takže aplikace mohou přes Avahi znát podobné aplikace na jiných strojích v síti. Avahi komunikuje s aplikacemi přes D-Bus, aplikace mohou publikovat své vlastní zdroje nebo se dotazovat na cizí zdroje. Například VoIP klient Ekiga může tímto způsobem vidět připojená SIP nebo H.323 zařízení ve stejné síti. Podobně lze vidět například seznam SSH serverů nebo veřejných datových archívů.

Zajímavým využitím Avahi je malá aplikace `gnome-user-share`. Umožňuje uživatelům velmi jednoduchým způsobem vystavovat některé soubory pro stažení

¹⁵<http://avahi.org/>

na jiné počítače: uživatel soubory umístí do svého adresáře `~/Public`, a spustí tuto aplikaci. `gnome-user-share` pak má na starosti spuštění kopie Apache s minimální konfigurací na některém neprivilegovaném portu a publikování informací o nově vzniklém zdroji dat přes Avahi. Ostatní uživatelé mohou ihned vidět nový adresář, ve kterém jim jsou nabízena data. Samotné `gnome-user-share` je pěkný příklad modularity UNIXu: aplikace má asi 80 KB včetně několika katalogů zpráv. Většina činností je realizována externími nástroji: Apache a Avahi.

System Avahi běží kromě Linuxu i pod *BSD systémy a Solarisem.

10 Je Linux připravený na desktop?

Cílem tohoto příspěvku bylo ukázat, kam dospěla v posledních letech podpora desktopových prostředí ze strany jádra Linuxu a některých dalších systémových programů. Přestože v některých oblastech jako je dostupnost specifikací hardwaru grafických karet nebo bezdrátových síťových karet situace stále není optimální, postupně i zde dochází ke zlepšení.

Na druhou stranu v mnoha jiných aspektech přináší modularita UNIXu spolu s orientací části vývojářů Linuxu na desktopové systémy již mnohé zajímavé výsledky.