



Abíčko

Časopis serveru AbcLinuxu.cz

květen 2002

Obsah

Editoriál	3
Nastavení síťové karty	4
Debian	4
Mandrake	6
Jak na duální boot s Windows 2000	8
Úvod	8
Zjištění startovacího oddílu	8
Zaváděč jádra	8
Uložení obrazu zaváděče	10
Dokončení	10
Diskuse	10
Ovládáme mobily značky Siemens	11
Instalace	11
Zkouška spojení	11
Základní operace	12
SMS	12
Uložení kontaktu na disk	12
Měníme LOGO operátora	13
Měníme šetříč obrazovky	13
Měníme vyzvánění	13
Záloha	13
Práce s PHONEBOOK	13
Synchronizace času	14
Download	14
Jaderné noviny	15
Stav podpory procesoru 386	15
Nesouhlas některých vývojářů s BitKeeperem	15
Loggování událostí	15
Linus má prázdniny	15
Nový ovladač NTFS	16
Diskuse nad SSSCA	16
Maximální velikost oddílu	16
Status RFC 2385 pod Linuxem	16
DMCA dopadá na vývojáře kernelu	16
Vymýcení velkého kernelového zámku	17
Quota na žurnálovacích souborových systémech	17
Statistiky BitKeepera	17
Nové fbdev API	17
Debata nad licencí kernelu	17

Editoriál

Vítejte u čtení časopisu Abíčko.

Abíčko vychází jako měsíční příloha serveru *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 *AbcLinuxu.cz* (založen na jaře 1999 pod názvem Linux Hardware) 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 *AbcLinuxu.cz* najdete rozsáhlou databázi hardwaru (návodů na instalaci pod Linuxem), velice aktivní diskusní fórum, podrobné návody a tutoriály, recenze, archiv ovladačů, informace o linuxovém jádře (včetně populárních Jaderných novin) i rozcestník po ostatních linuxových serverech. Další služby jsou v přípravě (včetně služeb pro firmy).

Máte-li námět na článek, zašlete jej do konference našich autorů: *autori@abclinuxu.cz*. Máte-li zájem o sponzoring Abíčka nebo jinou formu reklamy, kontaktujte nás na adrese *reklama@abclinuxu.cz*. Ostatní dotazy směřujte na adresu *literakl@abclinuxu.cz*.

©2002 Leoš Literák a autoři článků
Sazba: Ondřej Krejčík

Pro nekomerční účely smíte tento dokument jakkoliv šířit v tištěné i digitální podobě. V ostatních případech nás požádejte o svolení na adrese *literakl@abclinuxu.cz*.

Nastavení síťové karty

CIJOML

Tento článek se zaměřuje na distribuci Debian. Soubory, které je nutné změnit, jestliže používáte jinou distribuci, uvedu na konci článku. V ostatních distribucích jsou soubory, o kterých zde budu mluvit uloženy na jiných místech, ale mají stejný obsah. Pro bližší informace konzultujte manuálové stránky vaší distribuce. K článku je též připojen návod pro Mandrake, který laskavě poskytl pan David Horký.

Debian

Budeme předpokládat, že máme od našeho distributora jádro s podporou naší síťové karty, popřípadě, že jsme si jádro přeložili sami, a že tato karta byla nalezena. Potom by výpis jádra mohl vypadat třeba takto:

```
cs: cb_alloc(bus 2): vendor 0x115d, device 0x0003
PCI: Enabling device 02:00.0 (0000 -> 0003)
PCI: Setting latency timer of device 02:00.0 to 64
eth0: Xircom cardbus revision 3 at irq 11
PCI: Enabling device 02:00.1 (0000 -> 0003)
ttyS04 at port 0x4080 (irq = 11) is a 16550A
xircom cardcs: IO port probe 0x0c00-0x0cff: clean.
cs: IO port probe 0x0800-0x08ff: clean.
cs: IO port probe 0x0100-0x04ff: excluding 0x280-0x287 0x378-0x37f 0x4d0-0x4d7
cs: IO port probe 0x0a00-0x0aff: clean.
spurious 8259A interrupt: IRQ7.
bus adaptor found, registering as eth0, using irq 11
```

Výpis jádra zjistíte použitím příkazu **dmesg**. Pro nás je důležitý řádek obsahující **eth0**. Ten specifikuje, že karta je od výrobce Xircom, a že jí byla kernelem přidělena adresa **eth0**. Tímto se budeme řídit při nastavování sítě, protože IP adresa se samozřejmě přiděluje přímo danému adaptéru.

Nyní přistoupíme k nastavení vaší síťové karty tak, aby byla nakonfigurována okamžitě při bootu – bez našeho zásahu. Toto nastavení se samozřejmě liší v každé síti, proto požádejte vašeho správce sítě o informaci, jak mají být dané části nastaveny. Nyní popíši nastavení pro statické přidělení IP adresy, které má u UNIX boxů větší smysl než nastavení přes DHCP server. Pakliže se ve vaší síti přidělují IP adresy skrze DHCP server, změňte řádek `iface eth0` na `iface eth0 inet dhcp` a zbývající řádky odmažte.

```
vi /etc/network/interfaces

# The loopback interface
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet static
address 192.168.0.72
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
bgateway 192.168.0.2
```

Ještě je potřeba nastavit DNS resolving – zajišťuje převod IP adresy v číselné formě do lidsky zapamatovatelné formy. V tomto případě – první server vám zajistí resolving skrze lokální server a druhý v případě, že DNS server ve vaší společnosti vypadne:

```
nameserver 192.168.0.2
nameserver 217.77.161.130
```

Nyní je bezpečnější počítač rebootovat. Vyhnete se tak problémům, kdy nadálku zrebootujete počítač a on naběhne, ale se špatným nastavením sítě a vy musíte z tepla domova do práce a tuto prkotinu přenastavit. Věřte mi, že to není nic příjemného. Přesvědčíte se po znovunabootování za pomoci příkazu **ifconfig** a měli byste vidět něco takového:

```
eth0      Link encap:Ethernet HWaddr 00:10:A4:9D:23:33
inet addr:192.168.0.72 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:10444 (10.1 KiB)
Interrupt:11 Base address:0x4000
lo        Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:68 errors:0 dropped:0 overruns:0 frame:0
TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:4712 (4.6 KiB) TX bytes:4712 (4.6 KiB)
```

Občas se ovšem dostaneme do situace, kdy potřebujeme síť překonfigurovat v ten daný okamžik a to ne natrvalo. Toto se provádí přímo z příkazové řádky příkazem **ifconfig** a **route**. Nyní si to předvedeme. Předpokládejme stejnou konfiguraci sítě. Postupujme takto:

```
ifconfig eth0 192.168.0.72 netmask 255.255.255.0
route add default gw 192.168.0.2
```

Jak vidíte, není to nic složitého. Broadcast si systém nastaví inteligentně sám. Všechny položky odpovídají variantě uvedené výše. Toto nastavení vám poskytne váš správce sítě.

Ještě by se vám mohla hodit informace o dalších souborech v systému, které mají co do činění s nastavením sítě. Jedná se o soubory, o kterých se za celý svůj život nemusíte ani dozvědět, jelikož nejsou nutné, ale když o nich víte, mohou vám velice usnadnit život. Jsou to tyto soubory:

- `/etc/hosts`
- `/etc/hosts.allow`
- `/etc/hosts.deny`

Z nich je asi nejdůležitější `/etc/hosts`, ve kterém si můžeme k dané IP adrese přiřadit jméno, jaké sami chceme. Postup je takovýto. Jako první se uvádí IP adresa, potom oddělovač – mezera nebo tabulátor, a následně jméno. Minimálně vám doporučuji udělat tuto editaci:

```
127.0.0.1      localhost
```

Nedoporučuji vám do tohoto souboru přidávat vše – vzniknul by tak nepříjemný zmatek. Pro více domén použijte raději vhodně nakonfigurovaný DNS server. O tom však až v některém z dalších článků. Tato volba se občas dá použít, pakliže nechcete vypisovat celé dns jméno. `/etc/hosts` to udělají za vás. Stačí v případě *AbcLinuxu.cz* zadat:

```
212.47.8.34   abc
```

Poté stačí do vašeho browseru zadat jen `http://abc` a bez problémů se připojíte k *AbcLinuxu.cz*.

Další soubory se již skoro nevyužívají. Je dobré je zde ovšem alespoň pro informaci zmínit. Dříve se používaly k omezení přístupu strojů z definovaných IP adres do vašeho systému. Problém je, že vždy dovolí kontaktovat port na vašem počítači a až poté probíhá vyhodnocení povolení přístupu k vašim službám. To při určitých podmínkách může útočníkovi umožnit minimálně DDOS útok na váš systém. Proto raději používejte firewallovací nástroje přímo v jádře – staré **ipchains** a nové, vynikající **iptables**. Nastavení se provádí takto pro soubor `/etc/hosts.allow`:

```
ALL: .domena.cz EXCEPT terminalserver.domena.cz
```

Toto nastavení provede to, že zakáže přístup všem strojům z domény *domena.cz* kromě serveru *terminalserver.domena.cz*. Pro bližší nastavení použijte manuálové stránky ve vaší distribuci.

Dalším souborem je `/etc/hosts.deny`, který je obdobou `hosts.allow`, ale funguje opačně. Zde si dejte pozor na to, abyste měli nastaveno ještě navíc `ALL: PARANOID`, což nastavuje vyšší zabezpečení a mělo by být ve všech systémech.

Mandrake

Následující návod uveřejňujeme s laskavým svolením pana Horkého z firmy QCM.

Nastavení síťové konfigurace v Mandraku je velice jednoduché. Běžní uživatelé asi zvolí univerzální konfigurační nástroj Mandrake Control Center. Z oddílu Síť & Internet zvolíme ikonu Připojení. Otevře se nám okno s přehledem konfiguračních parametrů. Pomocí Průvodce, případně Expertního režimu nastavíme žádané parametry – IP adresu, subnet, defaultní cestu.

Zkušenější uživatelé si asi vyberou metodu přímé editace konfiguračních souborů. V Mandraku jsou nejdůležitější informace o síti uloženy v adresáři `/etc/sysconfig`. Zvláště nás bude zajímat podadresář `network-scripts`, kde v souboru `ifcfg-eth0` jsou uloženy konfigurační parametry ethernetového rozhraní `eth0`.

Příklad nastavení `ifcfg-eth0`, kdy se IP adresa získává pomocí dhcp protokolu:

```
DEVICE=eth0
BOOTPROTO=dhcp
NETMASK=255.255.255.0
ONBOOT=yes
```

Příklad nastavení s pevnou IP adresou:

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.1.1
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
NETMASK=255.255.255.0
ONBOOT=yes
```

Defaultní cestu nadefinujete v souboru `/etc/sysconfig/network`. Pokud máte více rozhraní, můžete zadefinovat, i které bude pro ni použito. Zde je příklad:

```
NETWORKING=yes
FORWARD_IPV4=false
HOSTNAME=brio.mandrake.cz
DOMAINNAME=mandrake.cz
GATEWAY=192.168.1.50
GATEWAYDEV=eth0
```

V souboru `/etc/hosts` definujete převod IP adres a jmen, nastavení name serverů je standardně uloženo v souboru `/etc/resolv.conf`.

Je nutné podotknout že pro nastavování sítě, stejně tak jako ostatní globální změny systému, je nutné znát heslo `roota`.

David Horký, <http://www.mandrake.cz>

Jak na duální boot s Windows 2000

Leoš Literák

Jste nuceni občas používat W2K? Už vás nebaví bootovat Linux z diskety? Chcete přidat Linux do startovací nabídky Windows 2000?

Úvod

I já jsem byl nucen řešit v nové práci tento problém. Disk už byl rozdělen a okna nainstalována, takže jsem neměl moc možností na výběr. V tomto článku vám ukážu řešení, které mi funguje. Text je psán tak, aby pomohl i začátečníkovi, guru ať tyto pasáže přeskochí a mně promine. A pro pořádek uvádím, že jsem tento návod našel někde na internetu.

Zjištění startovacího oddílu

Prvním krokem je nabootování Linuxu. Pravděpodobně z předem připravené diskety. Pokud jste na ni zapomněli, můžete zkusit nabootovat z CD s vaší distribucí, ale návod je mimo rozsah textu. Přihlašte se jako root. Nyní musíte zjistit, který diskový oddíl je nastaven jako startovací. Pokud si to pamatujete z instalace, je to nejlepší. Jinak budete muset trochu hledat. Není to ale nic složitého. Příkaz:

```
[root@localhost root]# mount
/dev/hda6 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda7 on /mnt/transfer type vfat (rw,uid=500)
none on /dev/shm type tmpfs (rw)
```

vám vypíše všechny připojené oddíly. Pokud by zde byl některý oddíl připojen jako /boot, zapsali byste si jeho jméno. V mém případě jsem takový oddíl nevytvořil, proto je pro mě startovacím oddílem kořenový adresář /. Ten je na prvním řádku výpisu a jmenuje se /dev/hda6.

Zaváděč jádra

Dalším krokem je instalace linuxového zaváděcího kódu. Pravděpodobně tento krok provedla už instalační procedura vaší distribuce, takže tuto část může většina z vás vynechat. Pokud se teprve chystáte instalovat Linux na váš počítač, zvolte při konfiguraci zaváděče operačního systému (bootloaderu) umístění do oddílu, nikoliv do MBR (master boot record).

Instalace lila není nic složitého, ale je to přeci jen riskantní krok. Pokud se spletete, můžete poškodit startovací tabulky a operační systémy budete bootovat už jen z diskety. Proto buďte velmi opatrní a tuto část použijte opravdu jen tehdy, když nic jiného nebude fungovat. S největší pravděpodobností následující část je pro vás zbytečná, takže pokud k tomu nemáte dobrý důvod, přeskochte ji.

Prvním krokem je vytvořit nebo upravit soubor `/etc/lilo.conf`. Měl by vypadat zhruba takto, místo `/dev/hda6` použijte váš startovací oddíl.

```
prompt
timeout=50
default=linux
boot=/dev/hda6
map=/boot/map
install=/boot/boot.b
message=/boot/message
lba32

image=/boot/vmlinuz-2.4.7-10
    label=linux
    initrd=/boot/initrd-2.4.7-10.img
    read-only
    root=/dev/hda6
```

Také nezapomeňte upravit i ostatní cesty, přesněji soubor s jádrem, který se bude pravděpodobně lišit. Jeho jméno zjistíte takto:

```
[root@localhost root]# ls /boot
boot.b
chain.b
initrd-2.4.7-10.img
kernel.h
kernel.h-2.4.7
message
module-info
module-info-2.4.7-10
os2_d.b
System.map
System.map-2.4.7-10
vmlinuz
vmlinuz-2.4.7-10
```

Ještě jedno upozornění: řádek `label=linux` a následující jsou odsazeny tabulátorem, nikoliv mezerami! Podrobnou nápovědu získáte spuštěním příkazu:

```
man 5 lilo.conf
```

Posledním krokem je spuštění příkazu

```
[root@localhost root]# lilo
Added Linux
```

Na tento krok nesmíte zapomenout!

Uložení obrazu zaváděče

Nyní je třeba uložit binární obraz zaváděče linuxového jádra do souboru. Použijeme na to standardní příkaz **dd**, v příkladě si nahradíte název oddílu vaším startovacím oddílem:

```
dd if=/dev/hda6 of=/root/bootsect.lnx bs=512 count=1
```

Tímto příkazem se uložilo prvních 512 bytů z oddílu `/dev/hda6` do souboru `/root/bootsect.lnx`. Při použití tohoto příkazu buďte opatrní, pokud byste přehodili hodnoty parametrů `if` a `of`, mohli byste si poničit data ve vašem Linuxu.

Nyní je nutné vygenerovaný soubor přenést do Windows nebo na disketu. Kdyby Microsoft neházel vývojářům Linuxu klacky pod nohy, nebylo by nic jednoduššího, než soubor přímo zkopírovat na disk C:. Bohužel podpora zápisu na souborový systém NTFS skončila v prachu dějin, takže musíme použít disketu. Případně jej zkopírovat na disk se souborovým systémem VFAT. Pokud se teprve chystáte instalovat váš počítač, doporučuji vám vytvořit si malý oddíl s tímto souborovým systémem a používat jej pro přenos dat mezi W2K a Linuxem.

Zkopírování dat na disketu není žádný velký problém. Pravděpodobně vám bude stačit následující série příkazů:

```
[root@localhost root]# mount -t msdos /dev/fd0 /mnt/floppy/
[root@localhost root]# cp /root/bootsect.lnx /mnt/floppy/
```

Dokončení

Po té, co jste přenesli zmíněný soubor na médium, které je přístupné na čtení pro W2K, můžete přebootovat do Windows. Zbytek instalace už je hračka. Nejdříve soubor `bootsect.lnx` zkopírujte na disk C:. Potom v ASCII editoru otevřete soubor `C:\boot.ini` a přidejte na jeho konec následující řádek:

```
C:\BOOTSECT.LNX='Linux'
```

Pak rebootujte váš počítač a během startu se vám objeví menu se dvěma položkami. Doufám, že budete častěji volit tu druhou :-)

Diskuse

- CIJOML:** Pokud budete instalovat nové jádro, je třeba vygenerovat soubor `bootsect.lnx` znovu.
- MaB:** Zatím jsem to vždycky dělal obráceně, tzn. uložit zaváděč windows do souboru a bootovat přes lilo. Tím odpadá i generování souboru po každé změně jádra.
- Libor:** No mám na počítači W2000 a Linux (MDK 8.2) a oba systémy bootují v pohodě přes lilo, které je v MBR. Původně jsem měl instalován W2000 a poté jsem nainstaloval Linux na druhý oddíl na HD a při instalaci zvolil lilo v MBR a jede to bez nejmenších problémů.

Ovládáme mobily značky Siemens

Stanislav Musil

Jelikož jsem se stal šťastným majitelem mobilního telefonu Siemens S45, rozhodl jsem se hned využít přiloženého datového kabelu. Protože s mobilním telefonem dostanete software pouze pro Windows, tak jsem hned začal pátrat po linuxové alternativě. A kamarád mě odkázal právě na následující prográmeček **scmxx** (<http://www.hendrik-sattler.de/scmxx/>) s tím, že je opravdu dobrý.

Instalace

Nejprve si do pomocného adresáře rozbalíme archiv `scmxx-0.x.x.tar.bz2` (měl jsem k dispozici verzi 0.5.15). Potom začneme s kompilací. Vystačíte si se standardní trojkombinací **configure-make-make install**. Autor ještě doporučuje před touto trojkombinací použít příkazy **autoconf** a **autoheader**. Ale je to jen doporučení, i bez těchto dvou příkazů to půjde kompilovat. Při použití **configure** máte možnost dopředu nastavit standardní port, na kterém program bude telefon hledat a jeho standardní přenosovou rychlost. Doporučuji

```
./configure --with-device=/dev/ttyS1
```

což odpovídá tomu, že telefon budete připojovat přes sériový kabel na COM2. Mimoto program používá proměnou `SCMXX_TTY` k výběru komunikačního portu. Potom již stačí zadat:

```
make
make install
```

Program se standardně nainstaluje do adresáře `/usr/local/bin/`.

Zkouška spojení

Pokud jste při instalaci již nastavili výchozí komunikační port, tak vám bude stačit napsat `scmxx --info` a měl by se Vám objevit výpis základních informací o vašem telefonu, např.

```
Accessing device /dev/ttyS1
OK, device is present.
Found SIEMENS mobile phone
Vendor:          SIEMENS
Model:           S45
Revision:        21
Serial (Phone):  xxxxxxxxxxxxxxxxxxxx
Serial (SIM):     xxxxxxxxxxxxxxxxxxxx
SIM-ID:          xxxxxxxxxxxxxxxxxxxx
Operator:        PAEGAS-CZ
SMS Server:      +420603052000
Charset:         GSM
Battery:         50%
Signal/BER:      -87 dBm/unknown
```

```
Time: 02/04/12,17:16:50
Avail. Slots: bmp: 0-2, mid: 0-4, vcf: 0-500, vcs: 1-50
Phonebooks: FD, SM, ON, ME, LD, MC, RC, OW, MS, CD, BL, RD, CS
```

Veškerá komunikace probíhá pomocí **AT** příkazů. Podrobnější informace včetně posílaných a přijímaných **AT** příkazů získáte parametrem `--verbose`, zkráceně `-v`. Detailní popis parametrů pro jednotlivé podporované telefony Siemens najdete na adrese <http://www.hendrik-sattler.de/scmxx/phones.html>.

Základní operace

Práce s programem je velice jednoduchá a intuitivní. Celý příkaz se skládá ze tří hlavních částí. První se týká nastavení spojení s telefonem (volba portu a rychlosti). Pokud jste správné hodnoty zadali už při kompilaci, nemusíte zde nic zadávat. V druhé části si zvolíte, zda chcete:

data odstranit	<code>--remove</code>
poslat do nebo přes telefon	<code>--send</code>
dostat z telefonu do počítače	<code>--get</code>

Do poslední části se píše, s čím chcete pracovat:

obrázky	<code>--bitmap</code>
zvonění	<code>--midi</code>
kalendář	<code>--vcal</code>
záznam v adresáři	<code>--vcf</code>
seznamy	<code>--pbook</code>
krátké textové zprávy	<code>--sms</code>

SMS

Parametry

<code>--sms-text text</code>	pošle text jako SMS zprávu
<code>--sms-nr číslo</code>	číslo, na které bude SMS odeslána
<code>--direct</code>	zpráva se odešle bez uložení do telefonu
<code>--flash</code>	zpráva se na cílovém telefonu zobrazí jen na displeji (pokud to podporuje)

Příklad:

```
scmxx --sms-nr 0777777777 --sms-text moje textovka
```

Pokud chcete vložit nový řádek do textovky, použijte `\n` a pokud chcete použít zpětné lomítko tak `\\`.

Uložení kontaktu na disk

```
scmxx --get --vcf=1 --file=~ /test.vcf
```

Uloží kontakt číslo 1 do souboru `test.vcf` v domovském adresáři uživatele ve standardním vizitkovém VCF formátu.

Měníme LOGO operátora

```
scmxx --send --bitmap=0 --file=soubor.bmp
scmxx -sB0 -f soubor.bmp
cat soubor.bmp | scmxx -sB
```

Jedná se o ekvivaletní příkazy, vyberte si jeden z nich.

barva: 2/16/256/šedá, rozlišení od 97x26 až po 101x43 pro S35i (větší velikosti budou ořezány)

Měníme šetřič obrazovky

```
scmxx --send --bitmap=2 --file=soubor.bmp
scmxx -sB2 -f soubor.bmp
```

Měníme vyzvánění

```
scmxx --send --midi=0 --file=soubor.midi
scmxx -sM0 -f soubor.mid
cat soubor.mid | scmxx -sM
```

formát: standardní midi formát 0. Midi 0 odpovídá číslu melodie 43 v mobilním telefonu

Záloha

Jenom změníme `send` na `get`.

```
scmxx --get --bitmap=0 --file=soubor.bmp
```

Právě jsme zazálohovali logo operátora.

Práce s PHONEBOOK

```
scmxx --get --pbook=XX --file=/home/uzivatel/zaznamy.txt
```

Tabulka hodnot parametru `--pbook`, pokud chcete výpis jen na obrazovku

```
scmxx --get --pbook=XX
```

FD	pevná volání – telefonní seznam (vyžaduje SIM PIN2)
SM	telefonní seznam na SIM kartě (záleží na typu) [118-250]
ON	vlastní telefonní čísla v SM+ME [3]
ME	telefonní seznam přímo v paměti telefonu [50]
LD	naposledy volaná čísla [10]

MC	ztracená volání [10]
RC	přijatá volání [10]
OW	stejně jako ON
MS	stejně jako MC
CD	stejně jako RC
BL	blacklist
RD	VIP čísla (CS s ! na konci)
CS	setříděná čísla (SM+ME+FD)

Program bere **sm** nebo **SM** nebo **Sm** nebo **sM** jako stejné, tzn. že není citlivý na velikost písmen.

Synchronizace času

V tomto případě se synchronizuje čas na mobilu s vaším časem v počítači.

```
scmxx --set-time
OK, device is present.
Found SIEMENS mobile phone
Time was synchronized
```

Download

Homepage programu je <http://www.hendrik-sattler.de/scmxx/>

Jaderné noviny

Leoš Literák

Stav podpory procesoru 386

Linus diskutoval s vývojáři nad procesorem 386, když H. Peter Anvin navrhnul zrušit podporu tohoto procesoru v jádře. Důvodů, proč to udělat, je povícero: CMPXCHG, BSWAP a zvláště WP. Alan Cox však poznamenal, že problémy týkající se těchto instrukcí se objevují jen v neobvyklých situacích. Například `cmpxchg8` se projevuje jen při direct rendering, který bude na procesoru 386 stejně poněkud pomalý :-). Takže toto není důvod pro odstranění podpory 386. Podpora SMP na těchto procesorech je jiný problém, naštěstí ale nevidí žádné šilence ji používající.

Linus odpověděl: Protože jediná osoba, která mě napadá, že by byla tak šílená, aby vyzkoušela SMP na 386, jsi ty, spadl mi kámen ze srdce :-). Ne, není to dostatečný důvod zrušit podporu 386. Kvůli chybějící instrukci WP kernel prostě nedokáže správně řídit threaded memory management. A nikdy to umět nebude. Používání 386 do síťových serverů je v pořádku, prostě z nich jen nedělejte CPU farmy (chtělo by to docela velkou farmu 386-ek, aby se vyrovnaly **jedinému** Pentiu II :-).

Nesouhlas některých vývojářů s BitKeeperem

Už delší dobu kolem komunity tiše vře spor nad licencí BitKeeperu. Dlouho se skrýval v soukromých emailech, až dnes probublal na povrch – do konference. Open Source klub z univerzity v Ohio zaslal Linusovi petici, aby přestal obhajovat BitKeeper. Podle nich je jádro Linuxu symbol hnutí Open Source a proto se nesluší, aby se oficiálně propagoval komerční software na správu verzí.

Andrew Morton s jejich názory souhlasil a žádal, aby jejich (Linusova) volba neomezila jeho možnost přispívat Linuxu. Randy Dunlap naprosto souhlasil, ale Eric W. Biederman navrhnul, aby protestující proti BitKeeperu místo sepisování petic raději vytvořili Open Source náhradu. K němu se připojil Rik van Riel, který dodal, že volba mezi ne zcela volným softwarem a úplně neužitečným volným softwarem je snadná. Podobně i Alan Cox napsal, že Linus nikoho nenutí BitKeeper používat a že by měli raději napsat vlastní systém pod licencí, která se jím líbí.

Diskuse se stupňovala, když zakročil Linus: Chlapi, zklidněte se. Nikoho nenutím používat BitKeeper. Ulehčuje mi život při práci s některými lidmi (obvykle správci subsystému), ale mnozí vývojáři jej nepoužívají a vůbec je to nezpomaluje. Například Al Viro zaznamenal jedinou změnu: lepší Changelog obsahuje jeho emailové komentáře.

Logování událostí

Larry Kessler oznámil dostupnost knihovny na logování událostí podle návrhu normy POSIX standard 1003.25. Knihovna je k dispozici na adrese <http://evlog.sourceforge.net/>. Určitou kontroverzní částí byla politika, podle níž se při zaplnění bufferu mazaly příchozí požadavky místo nejstarších. Dominik Kubla to považoval za bezpečnostní riziko, díky němuž mohl útočník zatajit své akce po zaplnění bufferu. Larry však vysvětlil, že při běžné konfiguraci bude démon stíhat číst události a navíc, do bufferu se ukládají jen události z jádra či modulů. Důvodem pro tuto politiku je snaha zachytit původní chybu, která by mohla být jinak přepsána chybami následnými.

Linus má prázdniny

Linus oznámil verzi 2.5.7 a dodal, že si bere dvoutýdenní dovolenou, během které nebude číst emaily. Problémy máme diskutovat v konferenci linux-kernel, spravovat patche mají Dave Jones a Jeff Garzik.

Nový ovladač NTFS

Anton Altaparmakov oznámil dostupnost nové verze 2.0 NTFS ovladače pro Linux (pouze read-only). Je určen pro jádro 2.5.7 a zašle jej Linusovi, jakmile se vrátí z dovolené. Ovladač byl intenzivně otestován a přežil všechny testy. Je plně kompatibilní s preemptible patchem a SMP. Měl by fungovat jak na big endian, tak na little endian architekturách, tak i na 32bitových a 64bitových platformách.

Nový ovladač je jasně rychlejší než původní (v mých testech o 20%), zatěžuje méně procesor a je ve všech ohledech lepší.

Diskuse nad SSSCA

Paul G. Allen napsal: To je velmi zlé. Pokud jej podepíší, bude se vztahovat na všechny software. Herman Oosthuysen dodal: řešení je zřejmé – pokračovat v cestě načrtnuté Richardem Stallmanem "Distribuuje všechny software ve zdrojovém kódu, žádné binárky". Tímto způsobem je autor chráněn zákonem o svobodě slova. Spousta lidí považovala Richarda za paranoidního, jenže on byl spíše prorokem.

Itai Nahshon napsal, že by mnohem horší bylo, kdyby někdo dostal patent na technologii na ochranu proti nelegálnímu kopírování. Alan napsal: Microsoft jej má. Herman soudil, že nebudeme muset platit Microsoftu poplatky, neboť zákon říká, že ochrana proti kopírování musí být Open Source. Jenže Rik van Riel jej zmrazil konstatováním, že dostupnost zdrojového kódu neznamena právo jej používat, pokud jiná společnost vlastní patent na technologii. Ovšem Herman reagoval s tím, že pokud tě zákon nutí něco používat, není možné z toho vybírat poplatky za použití. Máme dost precedentů.

Patent Microsoftu najdete na adrese <http://cryptome.org/ms-drm-os.htm>.

Maximální velikost oddílu

Michal Jaegermann měl problémy s vytvářením diskového oddílu o velikosti kolem 2 TB a tak se zeptal na maximální velikost oddílu pro ext2 či ext3. Andreas Dilger odpověděl, že limitem pro řady 2.2 a 2.4 jsou 2 TB, což pochází z $2^{32} * 512$ B na sektor. Používání LVM ani MD tento problém nepřekoná.

Status RFC 2385 pod Linuxem

David Schwartz se zeptal, zda někdo pracuje na ochraně BGP sessions pomocí MD5 podpisů. David by totiž rád používal bezpečné přenosy pro Zebru, takže by rád přispěl patchem s touto vlastností. Alan Cox odpověděl, že mezitím bylo prokázáno, že MD5 algoritmus je zranitelný vůči útokům pomocí koliznímu vyhledávání (Dobb) [collision search attacks]. David S. Miller nebyl tak taktní: Není žádný důvod nepodporovat toto smetí v uživatelském prostoru. Pokud potřebujete šifrování, použijte ipsec. Alan dodal, že ani v uživatelském prostoru není MD5 nejlepší volbou. Lidé přes kryptografii používají SHA a vědí proč.

Ovšem David Schwartz si stál na svém: Není žádný důvod, který by diskvalifikoval MD5 pro tuto určitou aplikaci. SHA by zvětšila každý paket a tím snížila efektivní MTU. Otázkou je, zda chceme, aby Linux dokázal spolupracovat s autentifikačním schématem Cisca. To ale nebyl solidní důvod pro Davida Millera, který napsal, že nenechá zaneřádit síťový subsystém Linuxu kvůli omylům Cisca.

Pak se debata obrátila konstruktivnějším směrem. David S. Miller odmítl zvyšovat komplexnost TCP (už tak velkou) a Alan navrhnul směřovat [route] BGP odpad skrze netlink tap zařízení. A David dodal, že jiným řešením by mohl být modul do netfilteru. Alan odpověděl, že Davidovo řešení by bylo rychlejší. Po chvíli se David ozval, že Alanovo řešení je lepší, neboť zajišťuje nulový overhead a je možné nastavit routování BGP pouze skrze zařízení připojené k TAP.

DMCA dopadá na vývojáře kernelu

David Rees nebyl schopen najít Changelog pro jádro 2.2.20. Alan Cox odpověděl, že pro občany jiných států než USA je tento dokument k dispozici na adrese <http://www.thefreeworld.net>. Rasmus Bag Hansen se podíval nad tím, jak je možné, že patch je v USA legální, zatímco jeho popis nikoliv. Mike Fedyk na to měl vysvětlení: v podstatě jde o to, že politici patchi nerozumí, zatímco shrnutí dokáží přečíst. A navíc v mnoha případech oprava bezpečnostní díry neukazuje způsob, jakým by se dala zneužít, zatímco bezpečnostní zpráva ano.

Vymýcení velkého kernelového zámku

Během pokusů o nahrazení velkého kernelového zámku BKL [big kernel Lock] méně drastickými způsoby zamykání, Dave Hansen zaslal patch a dodal, že tento patch přesouvá `disassociate_ctty` nahoru a uvolní BKL, jakmile je hotov. Zeptal se, zda je to čisté řešení, nebo některé z těch `exit_*`() metod potřebují tty. Tento patch redukoval čas držení BKL v metodě `do_exit()` stokrát. Původně byl čas 200 mikrosekund, nyní je 1,5 mikrosekundy. Nicméně, tyto hodnoty byly počítány na NUMA-Q stroji, takže reprezentují nejhorší případ.

Linus Torvalds odpověděl, že by preferoval přesunutí BKL do funkcí, které jej skutečně potřebují a odstranění ho z funkcí `do_exit()`. Je to obzvláště vhodné z toho důvodu, že si nejsem jist, zda `sem_exit()` jej vůbec ještě potřebuje. Pokud ne, prostě ho odstraňme odtud.

`disassociate_tty` je stejný případ, v budoucnu stejně opravíme tty vrstvu, tak udělejme z BKL interní věc. Dave souhlasil a zaslal nový patch.

Quota na žurnálovacích souborových systémech

Ken Brownfield velmi toužil po žurnálování na jeho 2 TB disku, ale potřeboval podporu `quot` a nebyl si jist, zda budou fungovat s `ext3` nebo `ReiserFS`. Luigi Genoni odpověděl, že používá `ReiserFS` a `quota tool 3.04` na `slackware-current` a nemá žádné problémy (jádro 2.4.18). A Andreas Dilger dodal, že pokud budeš používat jádra od RedHatu nebo jiné `-ac` kernely, získáš podporu 32bitových UID pro `quotu`.

Statistiky BitKeepera

Larry McVoy ohlásil nějaké statistiky ohledně BitKeeperu a jeho používání. Za dva měsíce se objevilo 3177 `changesetů` (patchů), obsahující 55000 změn v 11832 souborech. Interní data BitKeeperu už dosáhla 7 MB a rostou tempem 3 MB za měsíc, což je problém. Trpí zvláště výkonnost `update` (přesněji testy integrity). Nová verze BitKeeperu obsahuje konfigurační volbu pro zjednodušení těchto testů.

Nové `fbdev` API

James Simmons zaslal patch, který začíná nahrazovat starý kód používáním nového API frame bufferu. Základním cílem je odstranit tuny redundantního kódu a vytvořit mnohem jednodušší rozhraní. Kód je separován od konzole a bude mít výrazný vliv na `embedded` zařízení.

Debata nad licencí kernelu

Andrea Arcangeli změnil kód funkce `vmalloc` tak, aby se exportovala jako `EXPORT_SYMBOL` a ne `EXPORT_SYMBOL_GPL`. Tím ale vytočil Alana Coxe, který odpověděl, že autoři kódu jej umístili pod GPL a Andrea nemá žádné právo toto změnit. Tím porušuje licenci, porušuje digital rights management system (5 let vězení v USA) a co hůře – ignoruje základní morální práva autorů.

Andrea pochyboval o tom, zda takový export označuje, že kód je pod licencí GPL, na což Alan reagoval s tím, že každý jeho kus kódu je uvolněn pouze pod licencí GPL. Kdokoliv používá můj kód v ne-GPL kódu, dělá tak na základě vlastního právního rozhodnutí, zda se jedná o odvozenou práci či nikoliv. Kdežto Andreova úprava mění licenci je jiná záležitost. Jedná se o prostou krádež.

Andrea se ohradil s tím, že pokud taková triviální změna (funkce může být nahrazena inline verzí) je nezákonná, vytvoří fork kernelu a reimplementuje všechny takovéto funkce.

Do hádky vstoupil razantně Linus: OK, všichni jste na omylu. Odstranění `_GPL()` je v tomto případě v pořádku, ale ne z důvodů zde uvedených, nýbrž protože i Ingo souhlasil, že ovladače nemají mít ponětí o vnitřnostech VM. Fakt, že kód byl přeportován z 2.5 a `_GPL` tam zůstalo, je prostě chyba, neboť jsem nedostal updaty od Inga.

Ingo Molnar souhlasil a dodal, že metodu exportoval jako `_GPL` z čistě technických důvodů, neboť viděl ne-GPL ovladače ji používat a považoval ji za soukromou metodu. Nicméně, z principu je autorem Linus, takže toto je konec příběhu.

Ne tak pro Alana: takže Linus má právo měnit licenci příspěvků ostatních? Měli bychom si to ujasnit. Tento kód je přece zkopírován z bttv a napsal jej on a DaveM. Do jádra jej začlenil Gerd s GPL exportem. Situaci objasnil až Gerd Knorr. Jediný důvod, proč použil GPL, byla snaha zachovat stejnou politiku, jakou měla tato funkce v řadě 2.5. Jinak by se lidé naštváli, kdyby ji začali používat a později zjistili, že je v řadě 2.5 exportována podstatně přísněji. Rád zašle patch odstraňující `_GPL`, jakmile se tak stane i v řadě 2.5. Mimochodem, ten kód v 2.5 není cut & paste z bttv.

Tigran navrhnul přejmenovat `EXPORT_SYMBOL_GPL` na `EXPORT_SYMBOL_INTERNAL`, aby v budoucnu byl už podle názvu metody jasný její účel. Linus s ním souhlasil, neboť toto je podle něj jediný účel `_GPL` exportu. Alan se zeptal Keitha Owense, zda to nenaruší chod modutils, neboť stabilita 2.4 je nedotknutelná. Keith odpověděl, že taková změna je triviální a zpětně kompatibilní, ale provede ji, až se všichni shodnou. Do té doby, ať jej netahají do svých hádek.

Debata tím však neskončila. Alan, Tigran a Ingo diskutovali na téma GPL a odvozených prací a zda chtějí povolit binárním ovladačům přístup k vnitřním funkcím (šířenými pod GPL).

Tento článek vychází ze seriálu Kernel Traffic (<http://kt.zork.net>) a je zveřejněn pod licencí GPL verze 2.