

LINUX

Praktické návody

Linux NET-3, použití Linuxu na sítích

Terry Dawson, VK2KTJ, terry@perf.no.itg.telstra.com.au v1.2, 20. srpna 1997

Operační systém Linux má podporu jádra pro síť napsánu téměř od základu nově. Výkon implementace TCP/IP dělá z posledních verzí jader plnohodnotnou konkurenci i pro nejlepší srovnatelné systémy. Tento dokument si klade za cíl popsat instalaci a konfiguraci síťového softwaru a příslušných nástrojů v Linuxu.

1 Změny oproti předchozí verzi

Navíc:

- Odkaz na dokument PLIP – díky Claes IP NAT
- Úpravy/aktualizace
- Mnoho oprav od Alessandra Rubiniho – díky!
- Aktualizovaná e-mailová adresa Larryho Stefaniho – díky Larry
- Opravené umístění síťových nástrojů *ftp.linux.uk.org* – díky Rone
- Opravený nesprávný příkaz pro směrování – díky Johne
- Více nefunkčních směrovacích příkazů! – díky Jean-Pierre
- Adresy IPv6 jsou 16bitové, ne 32bitové, aha – díky Ezezi

Ještě schází:

- Přidat omezovač provozu (shaper)
- Popsat nový směrovací algoritmus
- Přidat volby IPv6 pro kompilaci jádra

- Popsat údaje /proc/sys/net/*
- Zařízení WanRouter

2 Úvod

Původní NET-FAQ napsal Matt Welsh a já, abychom ještě před spuštěním projektu Linux Documentation Project odpověděli na často pokládané dotazy o sítích pod Linuxem. Pokrývali jsme zde rané vývojové verze síťového jádra Linuxu. Po NET-FAQ následovalo NET-2-HOWTO a jednalo se o jeden z prvních dokumentů celého projektu LDP. Nyní byla používána verze 2 a později i 3 síťového softwaru jádra Linuxu. Tento dokument bezprostředně navazuje a vztahuje se již jen k verzi 3.

Předchozí verze dokumentu byly poměrně velké, protože takový byl i rozsah tématu. Proto byly vyčleněny některé další dokumenty, které se zabývají specifickými síťovými otázkami. Tento dokument se na ně na příslušných místech odkazuje a pokrývá oblasti, které ostatní dokumenty ještě nepopsaly.

2.1 Ohlasy

Vždy vítám ohlasy a obzvláště cenné příspěvky. Vše mi prosím odesílejte na můj *e-mail* <<mailto:terry@perf.no.itg.telstra.com.au>>*.

3 Jak používat tento dokument

Formát tohoto dokumentu se od starších verzí liší. Přeskupil jsem části tak, aby bylo možné přeskočit informativní části na začátku, pak následují obecné předpoklady, které je nutné zvládnout, a potom teprve přichází specializované části.

Čtete obecné části

Tyto části se vztahují ke každému nebo téměř každému postupu popisovanému později, takže je pro vás velmi důležité, abyste je pochopili.

Berte v úvahu vaši síť

Měli byste vědět, jak je (nebo má být) navržena vaše síť a jaký typ hardwaru a technologií využívá.

* Poznámka korektora: Současným správcem NET-3 HOWTO je Alessandro Rubini.

Přečtěte si části, zaměřené na vámi požadované technologie

Jakmile víte, co chcete, můžete nalézt jakoukoliv komponentu. Tyto části obsahují pouze podrobnosti k určitým technologiím.

Proveďte konfigurační práci

Měli byste se pokusit nakonfigurovat vaši síť a pečlivě se zaměřit na jakékoliv problémy, které vyvstanou.

Podle potřeby vyhledejte další pomoc

Jestliže se objeví problémy, se kterými vám tento dokument nepomůže, přečtěte si v příslušné části, kde je možné nalézt pomoc nebo nahlásit chyby.

Bavte se!

Používání sítí je zábava, tak si to užijte.

4 Obecné informace o používání Linuxu na sítích

4.1 Stručná historie vývoje podpory sítí v jádře Linuxu

Vývoj jádra se zcela novou implementací protokolů TCP/IP, která měla být alespoň tak dobrá, jako již existující implementace, nebyl vůbec snadný. Rozhodnutí o nevyužití existující implementace bylo provedeno v čase, kdy vznikla určitá obava z přetížení existující implementace díky určitým omezením práv ze strany U.S.L. Tehdy také bylo dost elánu k takové práci, mělo se to provést jinak a snad i lépe než kdy jindy.

Prvním dobrovolníkem, který vedl vývoj síťového kódu jádra, byl Ross Biro <biro@yggdrasil.com>. Ross vytvořil jednoduchou a nekompletní, ale většinou využitelnou implementační sadu rutin, které byly doplněny ethernetovým ovladačem pro síťovou kartu WD-8003. To stačilo, aby se softwarem experimentovalo dostatečně mnoho lidí, a aby se s tím někteří lidé dokonce připojili k živému Internetu. Takže na vývoj podpory sítě vznikl v linuxové komunitě určitý tlak, ale tomuto tlaku (někdy nevybíravému) podlehl Ross a vzdal se pozice vedoucího vývoje. Jeho snaha však odstartovala práci dobrým směrem a naznačila, že vývoj je možný.

Orest Zborowski <obz@Kodak.COM> vytvořil pro jádro Linuxu původní BSD socketové programové rozhraní. To byl velký skok, umožňující mnoha existujícím síťovým aplikacím úpravu pro Linux bez vážnějších potíží.

Někdy v této době vyvinul Laurence Culhane <loz@holmes.demon.co.uk> první ovladače pro Linux, podporující protokol SLIP. Takto mohli s novým síťovým softwarem experimentovat i lidé, kteří neměli přístup k ethernetovým sítím. Opět to někteří zkusili i na Internetu. Tak mnozí lidé nabyli dojmu, že je možné Linux na sítích využívat a počet experimentátorů se síťovým softwarem výrazně stoupl.

Jedním z lidí, kteří na problému vytvoření síťové podpory pracovali, byl Fred van Kempen <waltje@uwalt.nl.mugnet.org>. V období určité nejistoty po Rossově odstoupení z pozice vedoucího vývoje Fred obětoval svůj čas a námahu a tuto roli přijal. Fred měl v určitých směrech s Linuxem jisté ambice a také zde podpořil další pokrok. Vytvořil série síťového kódu, nazývané kód „NET-2“ jádra (kód „NET“ byl Rossův), které byli mnozí lidé schopni velmi dobře využívat. Do vývojové agendy Fred formálně vložil mnoho inovací, jako je dynamické rozhraní zařízení, podpora protokolu Amateur Radio AX.25 a více modulárně navržená implementace síťové podpory. Fredův kód NET-2 využívalo velké množství nadšenců, jejichž počet se zvyšoval s tím, jak se šířilo poznání o dobré funkčnosti. Síťový software byl v této době tvořen mnoha záplatami (patch) standardního kódu jádra a nebyl začleněn v jeho normální šířené verzi. Aby to všechno dobře fungovalo, byly zde dokumenty NET-FAQ a následný NET-2-HOWTO. Fred se soustředil na vývoj vylepšení standardních síťových implementací a to bralo hodně času. Komunita uživatelů začala být netrpělivá, i když zde byla spolehlivost, která uspokojila 80 % uživatelů. Stejně jako u Rosse začal stoupat tlak na Freda jako hlavního vývojáře.

Alan Cox <iialan@www.uk.linux.org> navrhl řešení situace. Předpokládal, že by si vzal Fredův kód NET-2 a odladil by jej, aby byl spolehlivý a stabilní a uspokojil nedočkavou uživatelskou základnu. Sejmul by tak z Freda tlak a Fred by mohl nadále pokračovat ve své práci. Alan nezůstal jen u návrhů a jeho první verze linuxového síťového kódu se nazývala „Net-2D“ (debugged - odladěný). Kód fungoval v mnoha běžných konfiguracích spolehlivě a uživatelská základna byla spokojená. Alan měl vhodné nápady a um, takže nepopíratelně přispěl do projektu i do řešení problému, kterým směrem se má ubírat vývoj kódu NET-2. Byly zde dva proudy, jeden zastával názor, že „nejdříve to musí fungovat a pak to musí fungovat lépe“, zatímco druhý chtěl, „aby to hned fungovalo lépe“. Linus nakonec nabídl podporu Alanovým vývojovým snahám a začlenil jeho kód do standardní distribuce kódu jádra. Tak se dostal do nevýhodné pozice Fred. Jakékoliv pokračování vývoje by postrádalo velkou uživatelskou základnu, která by kód aktivně využívala a testovala. Fred sice ještě v práci chvíli pokračoval, ale pak odpadl a na jeho pozici vedoucího vývoje linuxového síťového jádra se dostal Alan.

Donald Becker <becker@cesdis.gsfc.nasa.gov> brzy rozvinul svůj talent pro aspekty nižší úrovně použití sítí a vytvořil velký balík ethernetových ovladačů. Téměř všechny, které se nyní v jádrech používají, pochází od Donalda. Významně zde přispěli i mnozí jiní lidé, ale Donaldova práce je plodná, a proto si zaslouží speciální zmínku.

Alan určitou dobu vybrušoval kód NET-2-Debugged, přičemž pracoval na některých záležitostech, které zůstaly nedořešeny v seznamu „TODO“ (zbývá dokončit). V době, kdy byl zdrojový text jádra Linuxu 1.3.* teprve v plenkách, se síťový kód jádra přesunul do verze NET-3, na které jsou založeny i současné verze. Alan pracoval na mnoha různých aspektech síťového kódu a přitom mu pomáhali mnozí další talentovaní lidé ze síťové části komunity uživatelů Linuxu, takže kód rostl ve všech směrech. Alan vytvořil dynamická síťová zařízení a první standardní implementace AX.25 a IPX. Dále pokračoval ve vyiplávání kódu, pomalu jej přeorganizoval a vylepšil do dnešní podoby.

Podpora PPP byla přidána Michaellem Callahanem <callahan@maths.ox.ac.uk> a Alem Longyearem <longyear@netcom.com>. Tohle byl také jeden z faktorů, který ovlivnil zvýšení počtu uživatelů, kteří Linux aktivně využívali pro sítě.

Jonathon Naylor <jsn@cs.nott.ac.uk> přispěl značným vylepšením Alanova kódu AX.25, kde přidal podporu protokolů NetRom a Rose. Podpora AX.25/NetRom/Rose je sama o sobě dost významná, protože kromě Linuxu nemá standardně tuto podporu žádný jiný operační systém.

K vývoji linuxového síťového softwaru značnou měrou samozřejmě přispěly i stovky dalších lidí. S některými se setkáte v částech, vztahujících se k technologiím, další lidé přispěli moduly, ovladači, opravami chyb, návrhy, hlášeními o testech a morální podporou. Všichni si mohou říci, že se zúčastnili a přispěli tím, čím mohli. Síťový kód linuxového jádra je skvělým příkladem výsledku linuxového stylu anarchistického vývoje, který přitom stále není ukončen.

4.2 Kde získat další informace o sítích pod Linuxem

Míst, ze kterých můžete získat dobré informace o použití sítí pod Linuxem, je spousta.

Alan Cox, současný správce síťového kódu linuxového jádra, má stránku WWW, která obsahuje výtah ze současného a nového vývoje síťové podpory Linuxu: <http://www.uk.linux.org/NetNews.html>.

Dalším místem je kniha od Olafa Kircha, nazvaná **Network Administrators Guide**. Jedná se o práci pod *Linux Documentation Project* <http://sunsite.unc.edu/LDP/> a můžete si ji interaktivně přečíst na adrese <http://sunsite.unc.edu/LDP/LDP/nag/nag.html>

nebo ji můžete přes ftp získat v různých formátech z archívu ftp LDP `ftp://sun-site.unc.edu/pub/Linux/docs/LDP/network-guide/`. Olafova kniha je srozumitelná a nabízí dobrý vyšší standard síťové konfigurace pod Linuxem.

V linuxové hierarchii news je skupina, věnovaná sítím a příbuzným záležitostem. Jedná se o *comp.os.linux.networking* <**news:comp.os.linux.networking**>.

Existuje poštovní konference, ve které se můžete ptát na záležitosti kolem sítí pod Linuxem. Přihlásíte se odesláním zprávy:

```
To: majordomo@vger.rutgers.edu
Subject: anything at all
Zpráva: subscribe linux-net
```

Na různých sítích IRC jsou často kanály **#linux**, na kterých jsou lidé schopni odpovídat na dotazy kolem sítí pod Linuxem.

Při popisu jakéhokoliv problému prosím nezapomeňte na veškeré nutné podrobnosti. Zejména se jedná o verzi vámi používaného softwar – hlavně jádra, nástrojů, jako jsou *pppd* nebo *dip* - a o přesnou povahu problému. To znamená přesné zaznamenání chybových hlášení a předtím použitých příkazů.

4.3 Kde získat některé informace o sítích, které nejsou specifické pro Linux

Jestliže vám jde o některé základní obecné informace k použití sítí TCP/IP, doporučuji vám tyto dokumenty:

TCP/IP introduction (úvod)

Tento dokument existuje v *textové verzi* <`ftp://athos.rutgers.edu/runet/tcp-ip-intro.doc`> a v *postscriptové verzi* <`ftp://athos.rutgers.edu/runet/tcp-ip-intro.ps`>.

Jestliže na dané téma potřebujete ještě podrobnější informace, potom vřele doporučuji:

```
"Internetworking with TCP/IP"
od Douglase E. Comeru
ISBN 0-13-474321-0
Prentice Hall publications.
```

Jestliže se chcete naučit psát síťové aplikace pro prostředí, kompatibilní s Unixem, pak vám doporučuji:

"Unix Network Programming"
 od W. Richarda Stevens
 ISBN 0-13-949876-1
 Prentice Hall publications.

Nahlédněte také do newsové skupiny *comp.protocols.tcp-ip* <news:comp.protocols.tcp-ip>.

Důležitým zdrojem konkrétních technických informací k Internetu a protokolům TCP/IP jsou RFC. RFC je zkratka pro „Request For Comment“ a jedná se o standardní typ dokumentace k protokolům Internetu. RFC je možné nalézt na mnoha místech. Buď se jedná o ftp-servery nebo o WWW, kdy je možné prohledávat databázi RFC na základě klíčových slov.

Jedním ze zdrojů RFC je: *Nexorská RFC-databáze*

<<http://pubweb.nexor.co.uk/public/rfc/index/rfc.html>>.

5 Informace k běžné konfiguraci sítě

Následující části budete potřebovat znát a pochopit ještě předtím, než se skutečně dostanete ke konfiguraci vaší sítě. Jedná se o základní principy, platné nezávisle na povaze použité sítě.

5.1 Co potřebuji na začátku?

Před vytvořením nebo konfigurací sítě budete potřebovat pár věcí. Nejdůležitější jsou:

5.1.1 Aktuální zdrojový text jádra

Protože jádro, které nyní používáte, nemusí mít podporu pro zamýšlené síťové typy nebo karty, budete pravděpodobně potřebovat zdrojový text jádra, aby pak bylo možné jádro překompilovat s patřičnými volbami.

Nejnovější zdrojový text jádra je vždy možné získat na: *ftp.kernel.org* nebo

<<ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/v2.0>>.

Normálně bývá zdrojový text jádra rozbalen do adresáře `/usr/src/linux`. Informace k aplikaci úprav (patch) a sestavení jádra naleznete v dokumentu k jádru <[Kernel-HOWTO.html](#)>. Informace ke konfiguraci modulů jádra naleznete v dokumentu o modulech <[Module-HOWTO.html](#)>.

Pokud není řečeno něco jiného, doporučuji zůstat u standardních verzí jádra (mají druhou číslici ve verzi sudou). Vývojové verze jader (ty s druhou číslicí ve verzi lichou) mohou mít strukturální nebo jiné změny, které mohou při práci s jiným softwarem ve vašem systému způsobovat problémy. Jestliže si nejste jisti, že byste takovýto druh problémů zvládli (kromě dalších potenciálních softwarových chyb), tyto verze nepoužívejte.

5.1.2 Aktuální síťové nástroje

Síťové nástroje jsou programy, které používáte ke konfiguraci linuxových síťových zařízení. Tyto nástroje umožňují například přiřazení adres zařízením a konfigurování směrovacích cest.

Většina současných distribucí Linuxu je dodávána se síťovými nástroji, takže pokud instalujete z distribuce a ještě jste nenainstalovali síťové nástroje, učinite tak.

Jestliže neinstalujete z distribuce, musíte nástroje získat a zkompilovat sami. To ale není obtížné.

Síťové nástroje jsou nyní spravovány Berndem Eckenfelsem a jsou k dispozici na adrese [<ftp://ftp.inika.de/pub/comp/Linux/networking/NetTools/>](ftp://ftp.inika.de/pub/comp/Linux/networking/NetTools/) a jejich mirror najdete na adrese [<ftp://ftp.uk.linux.org/pub/linux/Networking/base/>](ftp://ftp.uk.linux.org/pub/linux/Networking/base/).

Vyberte verzi, která se nejlépe hodí k vámi používanému jádru. Při instalaci se řiďte příloženými instrukcemi.

Při instalaci a konfiguraci verze, aktuální v době psaní, postupujte následovně:

```
#
# cd /usr/src
# tar xvfz net-tools-1.33.tar.gz
# cd net-tools-1.33
# make config
# make
# make install
#
```

Navíc, pokud předpokládáte konfiguraci firewallu nebo využití funkce IP-masquerade, budete potřebovat příkaz `ipfwadm`. Jeho poslední verzi je možné získat na [<ftp://ftp.xos.nl/pub/linux/ipfwadm>](ftp://ftp.xos.nl/pub/linux/ipfwadm). Opět jsou zde k dispozici různé verze. Vyberte vždy tu verzi, která nejvíce vyhovuje vašemu jádru.

Při instalaci a konfiguraci verze, aktuální v době psaní, postupujte následovně:

```
#
# cd /usr/src
```

```
# tar xvfz ipfwadm-2.3.0.tar.gz
# cd ipfwadm-2.3.0
# make
# make install
#
```

5.1.3 Programy síťových aplikací

Programy síťových aplikací jsou programy, jako je `telnet` a `ftp` a jejich příslušné programy pro server. Distribuci těch nejběžnějších nyní spravuje David Holland <dholland@hcs.harvard.edu>. Získat je můžete na adrese <<ftp://ftp.uk.linux.org/pub/linux/Networking/base/>>.

Při instalaci a konfiguraci verze, aktuální v době psaní, postupujte následovně:

```
#
# cd /usr/src
# tar xvfz /pub/net/NetKit-B-0.08.tar.gz
# cd NetKit-B-0.08
# more README
# vi MCONFIG
# make
# make install
#
```

5.1.4 Adresy

IP-adresy se skládají ze čtyř bajtů. Je zvykem je zapisovat v desítkové podobě, oddělené tečkou. Každý bajt je převeden na desítkové číslo (0-255), nuly na začátku se nepíší a čísla se oddělují tečkou „.“. IP-adresu má každé rozhraní hostitele nebo routeru. Za určitých okolností je sice povoleno na jednom stroji u všech rozhraní použít stejnou IP-adresu, ale obvykle má každé rozhraní svoji vlastní.

IP-sítě jsou souvislé řady IP-adres. Všechny adresy v rámci sítě mají určitou část čísel adresy společnou. Tato společná část se nazývá „síťová část“ adresy. Zbývající čísla jsou nazývána „hostitelská část“ adresy. Počet bitů, sdílených všemi adresami sítě, se nazývá síťová maska a tato maska také určuje, které adresy do sítě patří a které ne. Vezměme si například:

Adresa hostitele	192.168.110.23
Síťová maska	255.255.255.0
Síťová část	192.168.110.
Hostitelská část	.23
Síťová adresa	192.168.110.0
Vysílací adresa	192.168.110.255

Každá adresa, která odpovídá síťové masce, ukazuje na síť, do které náleží. Síťová adresa má proto mezi adresami na síti nejnižší číslo a svoji hostitelskou část má nulovou.

Vysílací adresa je speciální adresa, na kterou slyší všichni hostitelé sítě (kromě své vlastní). Na tuto adresu se odesílají datagramy, které má obdržet každý hostitel sítě. Odesílají se sem určité typy dat, jako směrovací informace a různá upozornění. Existují dvě běžné vysílací adresy. Nejčastěji se používá nejvyšší možná adresa sítě. V předchozím příkladu by to byla 192.168.110.255. Někdy se jako vysílací používá síťová adresa. V praxi se nic nemění, ale všichni hostitelé sítě musí mít hlavně nakonfigurovanou stejnou vysílací adresu.

Z administrativních důvodů byly již v dobách vývoje protokolu IP vyčleněny určité skupiny adres do sítí, které byly seskupeny do tříd. Tyto třídy nabízí různý rozsah sítí, které je možné alokovat. Rozsahy jsou následující:

Třída sítě	Síťová maska	Síťové adresy
A	255.0.0.0	0.0.0.0 - 127.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
Multicast	240.0.0.0	224.0.0.0 - 239.255.255.255

Využívání adres závisí na tom, co děláte. Můžete využít i kombinace těchto tříd.

Instalace linuxového stroje na již existující IP-síť

Jestliže se chystáte linuxový stroj instalovat na již existující IP-síť, musíte kontaktovat osobu, která ji spravuje, a zeptat se na následující informace:

- IP-adresa hostitele
- Adresa IP-sítě
- Vysílací IP-adresa
- Síťová IP-mask

- Adresa routeru
- Adresa DNS

S takto získanými informacemi pak nakonfigurujete vaše linuxové síťové zařízení. Nemůžete si je vymyslet a pak očekávat, že by fungovaly.

Vytvoření zcela nové sítě, která se nebude připojovat k Internetu

Jestliže vytváříte soukromou síť a neočekáváte, že se kdy budete připojovat k Internetu, můžete si adresy vybrat jaké chcete. Z důvodů jednotnosti (a také bezpečnostních) jsou zde však pro tyto účely rezervovány určité adresy IP-sítě. Specifikace podle RFC1597 je následující:

REZERVOVANÉ ALOKACE SOUKROMÝCH SÍTÍ

Třída sítě	Síťová maska	Síťové adresy
A	255.0.0.0	10.0.0.0 – 10.255.255.255
B	255.255.0.0	172.16.0.0 – 172.31.255.255
C	255.255.255.0	192.168.0.0 – 192.168.255.255

Nejprve se rozhodnete, jak má být vaše síť velká. Potom vyberete tolik adres, kolik bude třeba.

5.2 Kam mám vložit konfigurační příkazy?

K procedurám zavádění systému Linuxu existují různé přístupy. Po zavedení jádra se vždy spouští program „init“. Tento program pak čte svůj konfigurační soubor `/etc/inittab` a začne proces zavádění. `init` má pár odrůd a to jsou také největší rozdíly mezi jednotlivými distribucemi nebo stroji.

`/etc/inittab` obvykle obsahuje přibližně takovýto údaj:

```
si::sysinit:/etc/init.d/boot
```

Tento řádek určuje název skriptu příkazového interpretu, který spravuje zaváděcí sekvenci. Tento soubor je obdobou souboru `AUTOEXEC.BAT` v operačním systému MS-DOS.

Jsou zde obvykle i další skripty, volané zaváděcím skriptem, přičemž síť se často konfiguruje právě jedním z nich.

Jako průvodce vaším systémem může sloužit následující tabulka:

Distribuce	Konfigurace rozhraní a směrovacích cest	Inicializace
Debian	<code>/etc/init.d/network</code>	<code>/etc/init.d/netbase</code> <code>/etc/init.d/netstd_init</code> <code>/etc/init.d/netstd_nfs</code> <code>/etc/init.d/netstd_misc</code>
Slackware	<code>/etc/rc.d/rc.inet1</code>	<code>/etc/rc.d/rc.inet2</code>
RedHat	<code>/etc/sysconfig/network-scripts/ifup-<ifname></code>	<code>/etc/rc.d/init.d/network</code>

Většina současných distribucí zahrnuje program, který umožňuje konfiguraci mnoha běžných síťových rozhraní. Jestliže jednu z nich máte, před pokusem o ruční konfiguraci se podívejte, jestli bude dělat to, co požadujete.

Distribuce	Konfigurační program sítě
RedHat	<code>/sbin/netcfg</code>
Slackware	<code>/sbin/netconfig</code>

5.3 Vytvoření vašich síťových rozhraní

V mnoha unixových operačních systémech se síťová zařízení objevují v adresáři `/dev`. V Linuxu tomu tak není. V Linuxu jsou síťová zařízení vytvářena dynamicky softwarem, takže nevyžadují přítomnost souborů zařízení.

Ve většině případů je síťové zařízení automaticky vytvořeno ovladačem zařízení při inicializaci a nalezení hardwaru. Například ovladač ethernetového zařízení vytvoří rozhraní `eth[0..n]` postupně po nalezení ethernetového hardwaru. První nalezená ethernetová karta se stane `eth0`, druhá `eth1` atd.

V některých případech, například *slip* a *ppp*, jsou síťová zařízení vytvořena zásahem některého uživatelského programu. Použije se stejné postupné číslování zařízení, ale zařízení nejsou automaticky vytvářena v době zavádění systému. Důvod je v tom, že na rozdíl od ethernetových zařízení se počet aktivních zařízení *slip* a *ppp* za běhu počítače mění. Tyto případy si později ještě podrobněji popíšeme.

5.4 Konfigurace síťového rozhraní

Jakmile máte všechny programy, které potřebujete, a informace o adrese a síti, můžete přistoupit ke konfiguraci vašich síťových rozhraní. Když mluvíme o konfiguraci síťového rozhraní,

mluvíme o procesu přiřazení příslušných adres síťovému zařízení a o nastavení příslušných hodnot pro další nastavitelné parametry síťového zařízení. Nejčastějším programem, který se k tomuto účelu používá, je příkaz `ifconfig`.

Většinou příkaz použijete v následující podobě:

```
# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
```

V tomto případě konfiguruji ethernetové rozhraní „eth0“ s IP-adresou „192.168.0.1“ a síťovou maskou „255.255.255.0“. Dodatek „up“ za příkazem sděluje rozhraní, že se má zaktivovat.

Jádro předpokládá při konfiguraci rozhraní určité implicitní hodnoty. Například pro rozhraní můžete určit síťové adresy a vysílací adresy, ale když je neurčíte (viz můj příklad), jádro je odvodí z dodané síťové masky. Když nedodáte masku, jádro ještě může využít třídu sítě nakonfigurované IP-adresy. V mém případě by jádro předpokládalo na rozhraní konfiguraci sítě třídy C a u rozhraní by nastavilo síťovou adresu „192.168.0.0“ a vysílací adresu „192.168.0.255“.

Příkaz `ifconfig` má mnoho dalších voleb. Nejdůležitější jsou:

up	Tato volba aktivuje rozhraní.
down	Tato volba deaktivuje rozhraní.
[- arp]	Tato volba u rozhraní nastavuje nebo ruší využití protokolu ARP.
[- allmulti]	Tato volba nastavuje nebo ruší příjem všech hardwarových multicastových paketů. Hardwarový multicast umožňuje skupinám hostitelů obdržet pakety, adresované na speciální místa určení. Tohle může být důležité v případě využití aplikací, jako jsou videokonference, které normálně nepoužíváte.
mtu N	Tento parametr umožňuje nastavit MTU tohoto zařízení.
netmask addr	Tento parametr umožňuje nastavit síťovou masku sítě, do které toto zařízení patří.
irq addr	Tento parametr je funkční pouze na určitých typech hardwaru a umožňuje nastavení IRQ tohoto zařízení.
[- broadcast [addr]]	Tento parametr umožňuje nastavit přijetí datagramů, určených na vysílací adresu, nebo jejich příjem zrušit.
[- pointopoint [addr]]	Tento parametr umožňuje nastavení adresy stroje na vzdáleném konci vazby typu point to point, jako je <i>slip</i> nebo <i>ppp</i> .

hw <typ> <adresa> Tento parametr umožňuje nastavení hardwarových adres určitých typů síťových zařízení. Není vždy nutný pro Ethernet, ale hodí se k jiným typům sítě, jako je AX.25.

Příkaz `ifconfig` můžete použít na jakémkoliv síťovém rozhraní. Některé uživatelské programy, jako je `pppd` a `dip`, automaticky konfigurují síťová zařízení ve chvíli, kdy je vytváří, takže ruční využití `ifconfig` není nutné.

5.5 Konfigurace resolveru

Resolver je částí standardní linuxové knihovny. Jeho základní funkcí je poskytnutí služeb převodu „lidských“ názvů, jako je „*ftp.funet.fi*“, na strojové IP-adresy, jako je **128.214.248.6**.

5.5.1 Co je v názvu?

Se vzhledem jmen hostitelů Internetu jste se již asi seznámili, ale jak se odvozují, to už asi chápete méně. Názvy lokací tvoří hierarchii, to znamená, že jejich struktura se podobá stromu. „*Doména*“ je rodina nebo skupina názvů. Tu můžeme rozdělit na „*subdomény*“. „*Doménou nejvyšší úrovně*“ rozumíme doménu, která není subdoménou. Domény nejvyšší úrovně jsou specifikovány v RFC-920. Příklady nejběžnějších domén nejvyšší úrovně:

- COM** komerční organizace
- EDU** vzdělávací organizace
- GOV** vládní organizace
- MIL** vojenské organizace
- ORG** jiné organizace

Domény nejvyšší úrovně jednotlivých zemí

dvou písmenný kód, reprezentující určitou zemi

Každá z těchto domén nejvyšší úrovně má subdomény. Domény nejvyšší úrovně, rozdělené podle zemí, jsou dále rozděleny na subdomény podle domén **com**, **edu**, **gov**, **mil** a **org**. Například komerční a vládní organizace v Austrálii by měly **com.au** a **gov.au**. Z historických důvodů je většina domén nejvyšší úrovně, které nejsou podle zemí, určena organizacím v USA, i když v Spojené státy mají také vlastní kód země - „*us*“.

Další úroveň rozdělení většinou reprezentuje název organizace. Další subdomény se liší podle povahy. Často je vyjádřením oddělení organizace, ale může to být i jakékoli jiné kritérium, které dává síťovým administrátorům organizace smysl.

Část názvu, která je nejvíce vlevo, je vždy jedinečný název hostitelského stroje - „*název hostitele*“. Část napravo je pak „*název domény*“ a celý název je *plně kvalifikované doménové jméno*.

Když jako příklad využiji svého hostitele pro e-mail, plně kvalifikované doménové jméno je „**perf.no.itg.telstra.com.au**“. To znamená, že název hostitele je „**perf**“ a název domény je „**no.itg.telstra.com.au**“. Doménové jméno je odvozeno z domény nejvyšší úrovně, což je v tomto případě moje země - Austrálie. Moje adresa pak náleží komerční organizaci, takže jako doménu následující úrovně máme „**com**“. Název společnosti je (byl) „**telstra**“ a naše vnitřní pojmenovávací struktura je založena na organizační struktuře. V mém případě můj stroj náleží do Information Technology Group, sekce Network Operations.

5.5.2 Jaké informace budete potřebovat

Budete potřebovat vědět, do které domény náleží název vašeho hostitele. Tuto překladovou službu poskytuje namerresolver provedením žádosti na DNS-server, takže budete muset znát IP-adresu lokálního nameserveru, který můžete využívat.

Editovat musíte tři soubory a nyní je postupně všechny probereme.

5.5.3 /etc/resolv.conf

Jedná se o hlavní konfigurační soubor resolveru. Jeho formát je velmi jednoduchý. Jedná se o textový soubor s jedním klíčovým slovem na každém řádku. Většinou se zde používají tři klíčová slova:

domain	Toto klíčové slovo určuje název lokální domény.
search	Toto klíčové slovo určuje seznam dalších domén, kde je možné hledat název hostitele.
nameserver	Toto klíčové slovo může být použito vícekrát a určuje IP-adresu DNS-serveru, na který se můžete obrátit při zjišťování názvů.

Ukázkový /etc/resolv.conf:

```
domain maths.wu.edu.au
search maths.wu.edu.au wu.edu.au
nameserver 192.168.10.1
nameserver 192.168.12.1
```

Tento příklad určuje, že implicitní název domény, připojované za nepřesné názvy (názvy hostitelů bez domény), je **maths.wu.edu.au** nebo (v případě neúspěchu) přímo **wu.edu.au**. Jsou zde dva nameservery, přičemž při zjišťování názvu je možné využít oba.

5.5.4 /etc/host.conf

Zde se konfigurují některé položky, ovlivňující chování kódu resolveru. Formát tohoto souboru je podrobně popsán na manuálové stránce „**resolv+**“. Téměř za všech okolností by vám měl fungovat následující příklad:

```
order hosts,bind
multi on
```

Tato konfigurace sdělí resolveru, aby před pokusem o žádost na nameserver kontroloval soubor `/etc/hosts` a aby podle tohoto souboru vracel všechny platné adresy.

5.5.5 /etc/hosts

Sem se vkládají názvy a IP-adresy lokálních hostitelů. Jestliže sem vložíte hostitele, nemusíte při zjišťování jeho IP-adresy obtěžovat DNS-server. Nevýhodou je to, že tento soubor musíte sami ručně aktualizovat (dochází-li ke změnám). V dobře spravovaném systému se zde objevují pouze údaje pro zpětnovazebné rozhraní loopback a názvy lokálních hostitelů.

```
# /etc/hosts
127.0.0.1 localhost loopback
192.168.0.1 this.host.name
```

Můžete použít i více než jeden název hostitele na řádek, viz první údaj v příkladu.

5.6 Konfigurace zpětnovazebného rozhraní

Rozhraní „**loopback**“ je speciálním typem rozhraní, které umožňuje provést připojení na sebe samého. K tomu existují různé důvody, například testování síťového softwaru bez zasahování do sítě. Většinou se pro loopback přidává IP-adresa „**127.0.0.1**“. Takže pokud zkusíte telnet na **127.0.0.1**, vždy se dostanete na lokálního hostitele.

Konfigurace rozhraní loopback je jednoduchá a určitě ji zvládnete.

```
# ifconfig lo 127.0.0.1
# route add -host 127.0.0.1 lo
```

O příkazu `route` si více povíme v následující části.

5.7 Směrování

Směrování je velké téma. Dalo by se o něm napsat opravdu hodně. Většina z vás má na směrování velmi jednoduché požadavky, někteří ale ne. Já se budu zabývat pouze základy. Jestliže se zajímáte o podrobnější informace, doporučuji vám odkazy ze začátku tohoto dokumentu.

Začneme definicí. Co je IP-směrování? Takto to chápu já: IP-směrování je proces, kterým hostitel s více síťovými přípojeními rozhodne, kam odeslat obdržené IP-datagramy.

K tomu by se hodil příklad. Představte si typický router, mohl by mít linku PPP, mimo Internet několik ethernetových segmentů pro pracovní stanice a další linku PPP do jiného úřadu. Když router na kterémkoliv ze svých síťových přípojení obdrží datagram, bude směrování mechanismem, rozhodujícím, na které rozhraní se datagram dále odešle. Směrování potřebují i jednodušší hostitelé. Všechny internetové hostitele mají alespoň dvě síťová zařízení - jedním je výše popsané rozhraní loopback a druhým je rozhraní použité pro komunikaci se zbytkem sítě - snad Ethernet, PPP nebo sériové rozhraní SLIP.

Dobrá, tak jak tedy směrování funguje? Každý hostitel má speciální seznam směrovacích pravidel, nazývaný směrovací tabulka. Tabulka obsahuje řádky většinou s alespoň třemi políčky - adresa určení, název rozhraní, kam se má datagram směrovat a třetí je volitelná IP-adresa dalšího stroje, který má datagram provést přes další krok na jeho cestě sítě. V Linuxu je možné tabulku zobrazit následujícím příkazem:

```
# cat /proc/net/route
```

nebo použitím jednoho z následujících příkazů:

```
# /sbin/route -n
# /bin/netstat -r
```

Směrovací proces je velice jednoduchý: přichází datagram je obdržen, adresa určení (pro koho to je) je porovnávána s údaji v tabulce. Vybere se údaj tabulky, který adrese nejlépe odpovídá, a na takto získané rozhraní je datagram propuštěn. Jestliže je zaplněno pole brány, datagram je předán přes specifické rozhraní, jinak se předpokládá, že adresa určení je dosažitelná uvedeným rozhraním.

Pro manipulaci s touto tabulkou se používá speciální příkaz. Tento příkaz bere argumenty příkazového řádku a převádí je na volání systému jádra, kde se žádá o přidání, smazání nebo úpravu údajů směrovací tabulky. Příkaz se nazývá „route“.

Jednoduchý příklad. Představte si ethernetovou síť. Bylo vám řečeno, že se jedná o síť třídy C s adresou **198.168.1.0**. Pro vaše využití vám byla přidělena IP-adresa **192.168.1.10** a řekli vám, že **192.168.1.1** je router, připojený na Internet.

Prvním krokem je konfigurace rozhraní (viz výše) - k tomu využijete příkaz, podobný tomuto:

```
# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

Nyní potřebujete údaj přidat do směrovací tabulky, aby jádro vědělo, že datagramy pro všechny hostitele s adresou, odpovídající **192.168.1.***, by měly být odeslány na ethernetové zařízení. Použili byste přibližně takovýto příkaz:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

Povšimněte si využití argumentu „**-net**“, který sděluje směrovacímu programu, že tento údaj je síťový směr. Vaše další volba je zde směr „**-host**“, což je směr, specifický pro jednu IP-adresu.

Tento směr vám umožní vytvořit IP-spojení se všemi hostiteli na vašem ethernetovém segmentu. Ale co pak se všemi IP-hostiteli, kteří na vašem ethernetovém segmentu nejsou?

Nastavení směrů pro všechna možná místa určení na síti by bylo zhola nemožné, takže se zde využívá určitý trik. Jedná se o směr „**default**“, který odpovídá všem možným místům určení, ale pokud se nalezne i jiný směr, je použit místo **default**. Filozofií je zde umožnit prohlásit „a všechno ostatní pošlete sem“. V tomto případě by tedy údaj vypadal takto:

```
# route add default gw 192.168.1.1 eth0
```

Argument „**gw**“ sděluje směrovacímu příkazu, že následujícím argumentem je IP-adresa (nebo název) brány nebo routeru, kterému budou všechny odpovídající datagramy předány pro další směrování.

Takže vaše kompletní konfigurace by vypadala takto:

```
# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
# route add default gw 192.168.1.1 eth0
```

Jestliže se blíže podíváte na vaše síťové soubory „**rc**“, zjistíte, že alespoň jeden z nich vypadá dost podobně. Tato konfigurace je velmi běžná.

Nyní se zaměříme na trochu komplikovanější směrovací konfiguraci. Představme si konfiguraci dříve zmiňovaného routeru, podporujícího linku PPP na Internet a segmenty LAN pro pracovní stanice na úřadě. Představme si, že router má tři ethernetové segmenty a jednu linku PPP. Naše směrovací konfigurace by vypadala takto:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
# route add -net 192.168.2.0 netmask 255.255.255.0 eth1
# route add -net 192.168.3.0 netmask 255.255.255.0 eth2
# route add default ppp0
```

Každá pracovní stanice by využívala jednodušší formu (prezentovanou výše), pouze router potřebuje určit všechny síťové směry odděleně, protože směrovací mechanismus **default** u pracovních stanic by všechny odchytil a nechal na routeru, aby je rozdělil. Možná se divíte, zde implicitní směr neurčuje „gw“. Důvod je jednoduchý - protokoly sériových linek (PPP a SLIP) mají na svých sítích pouze dva hostitele - jeden na každém konci. Určit hostitele z druhého konce linky jako bránu je zbytečné, protože nic jiného stejně nezbyvá. U těchto typů síťových připojení tedy není nutné určit bránu. Jiné typy sítí, jako je Ethernet, arcnet nebo token ring, určení brány vyžadují, protože tyto sítě v sobě podporují velké množství hostitelů.

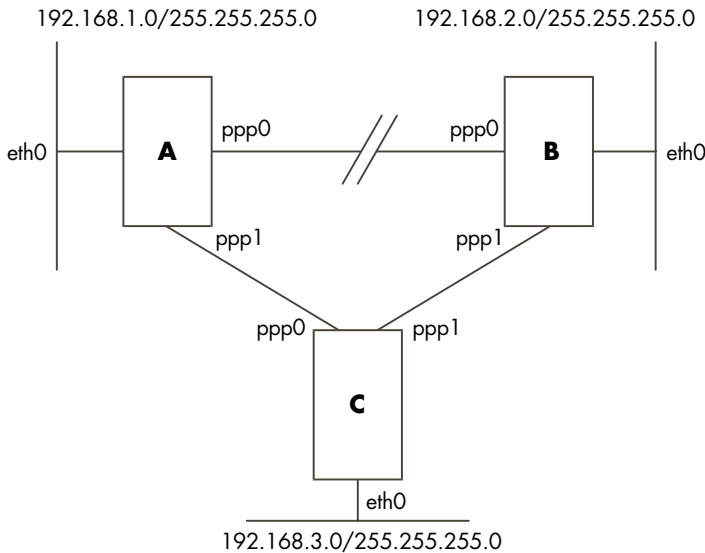
5.7.1 Takže co dělá program routed?

Výše popsaná směrovací konfigurace se nejlépe hodí pro jednodušší sítě, kde je většinou jen jedna možná cesta k cíli. Když je síť složitější, věci se hodně zkomplikují - našťestí se to většiny z vás netýká a nemusíte se tím zabývat.

Velkým problémem „ručního směrování“ nebo „statického směrování“ (jak bylo popsáno) je situace, kdy selže stroj nebo linka a jediným způsobem směrování jinou cestou (existuje-li vůbec taková) je osobní zásah a provedení příslušných příkazů. Tohle je samozřejmě hloupé, pomalé, nepraktické a riskantní. Tohle je možné provádět i automaticky - existuje mnoho postupů, souhrnně nazývaných „dynamické směrovací protokoly“.

Možná jste o těch běžnějších slyšeli - nejznámější jsou RIP a OSPF. RIP je velmi běžný na malých sítích, jako jsou sítě v malých až středních společnostech nebo v budovách. OSPF je modernější a schopnější zvládnout konfigurace velkých sítí. Je také vhodnější v případě, že existuje velké množství možností cesty sítí. Známé implementace těchto protokolů jsou: „routed“ - RIP a „gated“ - RIP, OSPF a další. Program „routed“ se normálně dodává ve vaší distribuci Linuxu nebo je obsažen v balíku „NetKit“, viz výše.

Příklad kdy a jak můžete využít dynamický směrovací protokol vypadá následovně:



Máme tři routery A, B a C. Každý podporuje jeden ethernetový segment s IP-sítí třídy C (síťová maska 255.255.255.0). Každý router má také linku PPP ke zbylým routerům. Síť je ve tvaru trojúhelníku.

Mělo by být jasné, že směrovací tabulka na routeru A by měla vypadat takto:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
# route add -net 192.168.2.0 netmask 255.255.255.0 ppp0
# route add -net 192.168.3.0 netmask 255.255.255.0 ppp1
```

Tohle by mělo spolehlivě fungovat až do chvíle, kdy by selhala linka mezi routerem A a B. V takovém případě by hostitelé na ethernetovém segmentu A nemohli dosáhnout hostitele na ethernetovém segmentu B, protože jejich datagramy by byly směrovány na linku PPP routeru A, která je nefunkční. Mohou ale stále komunikovat s hostiteli na segmentu C a ti zase s hostiteli na segmentu B, protože linka mezi B a C je funkční.

Když ale A komunikuje s C a C s B, proč by A nemohlo poslat svoje datagramy do B přes C? To je zhruba druh problémů, které řeší dynamické směrovací protokoly jako RIP. Jestliže všechny routery A, B a C měly puštěný směrovací démon, pak jsou jejich směrovací tabulky při poruše některé linky automaticky upraveny, aby odrážely nový stav sítě. Konfigurace takové sítě je snadná - na každém routeru potřebujete provést pouze dvě věci. V našem případě pro router A:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
# /usr/sbin/routed
```

Směrovací démon „*routed*“ při spuštění automaticky nalezne všechny aktivní síťové porty a odesílá a přijímá zprávy na každé síťové zařízení, aby bylo možné určit a aktualizovat směrovací tabulku na hostiteli.

Tohle bylo velice stručně vysvětlení dynamického směrování a jeho použití. Jestliže potřebujete více informací, měli byste se vrátit na začátek dokumentu a pročíst si odkazy.

Důležité postřehy, vztahující se k dynamickému směrování, jsou:

1. Démona pro dynamické směrování potřebujete na stroji s Linuxem spouštět pouze v případě, že máte na výběr více možných směrů na místo určení.
2. Tento démon automaticky upravuje vaši směrovací tabulku, čímž odráží změny v síti.
3. RIP je vhodný pro malé až středně velké sítě.

5.8 Konfigurace vašich síťových serverů a služeb

Síťové servery a služby jsou ty programy, které umožňují vzdálenému uživateli, aby se stal uživatelem vašeho linuxového stroje. Serverové programy naslouchají na síťových portech. Síťové porty jsou významem adresace určité služby na určitém hostiteli. Podle nich server pozná rozdíl mezi přicházejícím telnetovým a ftp-připojením. Vzdálený uživatel vytvoří síťové připojení k vašemu stroji a serverovému programu. Program síťového démona, který na tomto portu naslouchá, připojení přijme a provede. Existují dva způsoby, jakými mohou síťoví démoni fungovat. Oba jsou běžně využívány v praxi:

standalone (samostatný)

Síťový démon naslouchá na vyznačeném síťovém portu a jakmile je provedeno přicházející připojení, spravuje samotné síťové spojení, aby poskytovalo svoje služby.

slave (podřízený) **pod `inetd` serverem**

Server `inetd` je síťový démon, který se specializuje na přichází síťová spojení. Má konfigurační soubor, kde se říká, který program má být spuštěn po navázání spojení. Na protokoly TCP nebo UDP je možné nakonfigurovat jakýkoliv servisní port. Porty jsou popsány v dalším souboru, ke kterému se brzy dostaneme.

Musíme nakonfigurovat dva důležité soubory. Jedná se o `/etc/services` (přiřazuje názvy číslům portů) a o `/etc/inetd.conf`, který je konfiguračním souborem síťového démona `inetd`.

5.8.1 /etc/services

Tento soubor je jednoduchou databází, která přiřazuje „lidské“ názvy strojovým portům služeb. Jeho formát je jednoduchý. Soubor je textový, přičemž každý řádek reprezentuje údaj databáze. Každý údaj se skládá ze tří polí, oddělených libovolným množstvím prázdného místa (tabulátory nebo mezery). Pole jsou:

```
jméno          port/protokol   přezdívky      # komentář
```

jméno Jednoslovný název, který reprezentuje popisovanou službu.

port/protokol Toto pole je rozděleno na dvě:

port číslo, které určuje číslo portu, na kterém bude pojmenovaná služba k dispozici. Většina běžných služeb má přiřazena standardní čísla. Jsou popsána v **RFC-1340**.

protokol Toto pole je možné nastavit na **tcp** nebo **udp**.

Důležité je, že údaj **18/tcp** se značně liší od **18/udp** a není zde žádný technický důvod, proč by stejná služba měla existovat na obou. Většinou převládá obecný smysl a údaj pro oba se objeví pouze v případě, že je určita služba také pro oba k dispozici.

přezdívky Další názvy, které je možné použít při odkazu na údaje této služby.

Jakýkoliv text, který se na řádku objeví za znakem „#“, je ignorován a posuzován jako komentář.

Ukázkový soubor /etc/services

Všechny současné distribuce Linuxu nabízí důležitý soubor `/etc/services`. Jen pro případ, že byste vytvářeli stroj úplně od základů, sem umístíte kopii souboru `/etc/services`, který je v distribuci *Debian* <<http://www.debian.org/>>.

```
# /etc/services:
# $Id: services,v 1.3 1996/05/06 21:42:37 tobias Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single
# well-known
# port number for both TCP and UDP; hence, most entries here have
# two entries
```



```
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1340, "Assigned Numbers" (July 1992). Not all
# ports are included, only the more common ones.

tcpmux      1/tcp # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp  sink null
discard     9/udp  sink null
sysstat     11/tcp users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
qotd        17/tcp quote
msp         18/tcp # message send protocol
msp         18/udp # message send protocol
chargen     19/tcp ttytst source
chargen     19/udp ttytst source
ftp-data    20/tcp
ftp         21/tcp
ssh         22/tcp # SSH Remote Login Protocol
ssh         22/udp # SSH Remote Login Protocol
telnet      23/tcp

# 24 - private
smtp        25/tcp mail

# 26 - unassigned
time        37/tcp timserver
time        37/udp timserver
rlp         39/udp resource # resource location
nameserver  42/tcp name # IEN 116
whois       43/tcp nickname
re-mail-ck  50/tcp # Remote Mail Checking Protocol
re-mail-ck  50/udp # Remote Mail Checking Protocol
domain      53/tcp nameserver # name-domain server
domain      3/udp nameserver
mtp         57/tcp # deprecated
bootps     67/tcp # BOOTP server
```

```
bootps      67/udp
bootpc      68/tcp # BOOTP client
bootpc      68/udp
tftp        69/udp
gopher      70/tcp # Internet Gopher
gopher      70/udp
rje         77/tcp netrjs
finger      79/tcp
www         80/tcp http # WorldWideWeb HTTP
www         80/udp # HyperText Transfer Protocol
link        87/tcp ttylink
kerberos    88/tcp kerberos5 krb5 # Kerberos v5
kerberos    88/udp kerberos5 krb5 # Kerberos v5
supdup      95/tcp
# 100 - reserved
hostnames   101/tcp hostname # usually from sri-nic
iso-tsap    102/tcp tsap # part of ISODE.
csnet-ns    105/tcp cso-ns # also used by CSO name server
csnet-ns    105/udp cso-ns
rtelnet     107/tcp # Remote Telnet
rtelnet     107/udp
pop-2       109/tcp postoffice # POP version 2
pop-2       109/udp
pop-3       110/tcp # POP version 3
pop-3       110/udp
sunrpc      111/tcp portmapper # RPC 4.0 portmapper TCP
sunrpc      111/udp portmapper # RPC 4.0 portmapper UDP
auth        113/tcp authentication tap ident
sftp        115/tcp
uucp-path   117/tcp
nntp        119/tcp readnews untp # USENET News Transfer Protocol
ntp         123/tcp
ntp         123/udp # Network Time Protocol
netbios-ns  137/tcp # NETBIOS Name Service
netbios-ns  137/udp
netbios-dgm 138/tcp # NETBIOS Datagram Service
netbios-dgm 138/udp
netbios-ssn 139/tcp # NETBIOS session service
```

```
netbios-ssn 139/udp
imap2       143/tcp # Interim Mail Access Proto v2
imap2       143/udp
snmp        161/udp # Simple Net Mgmt Proto
snmp-trap   162/udp snmptrap # Traps for SNMP
cmip-man    163/tcp # ISO mgmt over IP (CMOT)
cmip-man    163/udp
cmip-agent  164/tcp
cmip-agent  164/udp
xdmcp       177/tcp # X Display Mgr. Control Proto
xdmcp       177/udp
nextstep    178/tcp NeXTStep NextStep # NeXTStep window
nextstep    178/udp NeXTStep NextStep # server
bgp         179/tcp # Border Gateway Proto.
bgp         179/udp
prospero    191/tcp # Cliff Neuman's Prospero
prospero    191/udp
irc         194/tcp # Internet Relay Chat
irc         194/udp
smux        199/tcp # SNMP Unix Multiplexer
smux        199/udp
at-rtmp     201/tcp # AppleTalk routing
at-rtmp     201/udp
at-nbp      202/tcp # AppleTalk name binding
at-nbp      202/udp
at-echo     204/tcp # AppleTalk echo
at-echo     204/udp
at-zis      206/tcp # AppleTalk zone information
at-zis      206/udp
z3950       210/tcp wais # NISO Z39.50 database
z3950       210/udp wais
ipx         213/tcp # IPX
ipx         213/udp
imap3       220/tcp # Interactive Mail Access
imap3       220/udp # Protocol v3
ulistserv   372/tcp # UNIX Listserv
ulistserv   372/udp
#
```

```
# UNIX specific services
#
exec          512/tcp
biff         512/udp comsat
login        513/tcp
who          513/udp whod
shell        514/tcp cmd # no passwords used
syslog       514/udp
printer      515/tcp spooler # line printer spooler
talk         517/udp
ntalk        518/udp
route        520/udp router routed # RIP
timed        525/udp timeserver
tempo        526/tcp newdate
courier      530/tcp rpc
conference   531/tcp chat
netnews      532/tcp readnews
netwall      533/udp # -for emergency broadcasts
uucp         540/tcp uucpd # uucp démon
remotefs     556/tcp rfs_server rfs # Brunhoff remote filesystem
klogin       543/tcp # Kerberized 'rlogin' (v5)
kshell       544/tcp krcmd # Kerberized 'rsh' (v5)
kerberos-adm 749/tcp # Kerberos 'kadmin' (v5)
#
webster      765/tcp # Network dictionary
webster      765/udp
#
# From ''Assigned Numbers'':
#
#> The Registered Ports are not controlled by the IANA and on most
#> systems can be used by ordinary user processes or programs
#> executed by
#> ordinary users.
#
#> Ports are used in the TCP [45,106] to name the ends of logical
#> connections which carry long term conversations. For the
```

```
#> purpose of
#> providing services to unknown callers, a service contact port is
#> defined. This list specifies the port used by the server process
#> as its contact port. While the IANA can not control uses of these
#> ports
#> it does register or list uses of these ports as a convenience
#> to the community.
#
ingreslock 1524/tcp
ingreslock 1524/udp
prospero-np 1525/tcp # Prospero non-privileged
prospero-np 1525/udp
rfe        5002/tcp # Radio Free Ethernet
rfe        5002/udp # Actually uses UDP only
bbs        7000/tcp # BBS service
#
#
# Kerberos (Project Athena/MIT) services
# Note that these are for Kerberos v4 and are unofficial. Sites
# running v4 should uncomment these and comment out the v5 entries
# above.
#
kerberos4 750/udp kdc # Kerberos (server) udp
kerberos4 750/tcp kdc # Kerberos (server) tcp
kerberos_master 751/udp # Kerberos authentication
kerberos_master 751/tcp # Kerberos authentication
passwd_server 52/udp # Kerberos passwd server
krb_prop      754/tcp # Kerberos slave propagation
krbupdate 7 60/tcp kreg # Kerberos registration
kpasswd      761/tcp kpwd # Kerberos "passwd"
kpop         1109/tcp # Pop with Kerberos
knetd       2053/tcp # Kerberos de-multiplexor
zephyr-srv  2102/udp # Zephyr server
zephyr-clt  2103/udp # Zephyr serv-hm connection
zephyr-hm   2104/udp # Zephyr hostmanager
eklogin     2105/tcp # Kerberos encrypted rlogin
#
# Unofficial but necessary (for NetBSD) services
```

```

#
supfilesrv      871/tcp # SUP server
supfiledbg      1127/tcp # SUP debugging
#
# Datagram Delivery Protocol services
#
rtmp             1/ddp # Routing Table Maintenance Protocol
nbp             2/ddp # Name Binding Protocol
echo            4/ddp # AppleTalk Echo Protocol
zip             6/ddp # Zone Information Protocol
#
# Debian GNU/Linux services
rmtcfg          1236/tcp # Gracilis Packeten remote config server
xtel            1313/tcp # french minitel
cfinger         2003/tcp # GNU Finger
postgres        4321/tcp # POSTGRES
mandelspawn     9359/udp mandelbrot # network mandelbrot

# Local services

```

5.8.2 /etc/inetd.conf

Jedná se o konfigurační soubor pro serverového démona `inetd`. Jeho funkcí je sdělit `inetd` co dělat, když obdrží žádost o spojení na určitou službu. U každé služby, pro kterou chcete přijmout spojení, musíte `inetd` sdělit, jak a který démon síťového serveru spustit.

Jeho formát je také velice jednoduchý. Jedná se o textový soubor s každým řádkem popisujícím službu, kterou chcete poskytnout. Jakýkoliv text, který se na řádku objeví za znakem „#“, je ignorován a posuzován jako komentář. Každý řádek obsahuje sedm polí, oddělených libovolným množstvím prázdného místa (tabulátory nebo mezery). Obecný formát vypadá takto:

```

služba  typ_soketu  protokol  volby  uživatel  cesta_k_serveru\
        argumenty_serveru

```

služba Je služba, příslušná této konfiguraci, převzatá ze souboru `/etc/services`.

typ_soketu Toto pole popisuje typ socketu. Možné hodnoty jsou: **stream**, **dgram**, **raw**, **rdm** nebo **seqpacket**, což je sice trochu více technické, ale platí zde pravidlo, že skoro všechny služby **tcp** používají **stream** a skoro všechny služby **udp** používají **dgram**. Pouze velmi speciální typy serverových démonů využívají některé ze zbývajících hodnot.

protokol	Protokol, platný pro tento údaj. Měl by odpovídat příslušnému údaji v souboru <code>/etc/services</code> a bývá tcp nebo udp . Servery Sun RPC budou využívat rpc/tcp nebo rpc/udp .
volby	Toto pole má jen dvě možná nastavení. Zde si <code>inetd</code> zjistí, jestli program síťového serveru po spuštění uvolní socket a jestli má tedy <code>inetd</code> při následující žádosti o spojení spustit další, nebo jestli má <code>inetd</code> čekat a předpokládat, že nějaký již běžící serverový démon se o žádost o nové připojení postará. Opět je zde chápání trochu zjednodušené, ale pravidlem je, že všechny servery tcp by měly mít nastaveno nowait a většina serverů udp by měla mít wait . Jsou zde však časté výjimky, takže pokud si nejste jisti, raději se řiďte podle příkladu.
uživatel	Toto pole popisuje, který uživatelský účet z <code>/etc/passwd</code> bude pro spuštěného síťového démona nastaven jako vlastník. Užitečné zejména z hlediska bezpečnosti. Uživatele můžete nastavit jako nobody , takže pokud se naruší bezpečnost serveru, možné ztráty budou minimální. Většinou je zde ale nastaven root , protože mnohé servery ke své správné funkci vyžadují práva superuživatele.
cesta_k_serveru	Cesta k aktuálnímu serverovému programu, spouštěnému pro tento údaj.
argumenty_serveru	Tohle pole tvoří zbytek řádku a je volitelné. Zde se umísťují všechny parametry příkazového řádku, které chcete předat serverovému démonu při jeho spuštění.

Ukázkový soubor `/etc/inetd.conf`

Stejně jako u `/etc/services` všechny současné distribuce Linuxu nabízí správný soubor `/etc/inetd.conf`. Pro doplnění sem umísťuji kopii souboru `/etc/inetd.conf`, který je v distribuci *Debian* <<http://www.debian.org/>>.

```
# /etc/inetd.conf: see inetd(8) for further informations.
#
# Internet server configuration database
#
# Modified for Debian by Peter Tobias <tobias@et-inf.fho-emden.de>
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path>
#   <args>
```

```
#
# Internal services
#
#echo          stream    tcp      nowait   root    internal
#echo          dgram    udp      wait     root    internal
discard       stream    tcp      nowait   root    internal
discard       dgram    udp      wait     root    internal
daytime       stream    tcp      nowait   root    internal
daytime       dgram    udp      wait     root    internal
#chargen      stream    tcp      nowait   root    internal
#chargen      dgram    udp      wait     root    internal
time          stream    tcp      nowait   root    internal
time          dgram    udp      wait     root    internal
#
# These are standard services.
#
telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd
ftp    stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
#fsp   dgram  udp  wait   root  /usr/sbin/tcpd /usr/sbin/in.fspd
#
# Shell, login, exec and talk are BSD protocols.
#
shell  stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rshd
login  stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rlogind
#exec  stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rexecd
talk   dgram  udp  wait   root  /usr/sbin/tcpd /usr/sbin/in.talkd
ntalk  dgram  udp  wait   root  /usr/sbin/tcpd /usr/sbin/in.ntalkd
#
# Mail, news and uucp services.
#
smtp    stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.smtpd
#nntp   stream tcp nowait news /usr/sbin/tcpd /usr/sbin/in.nntpd
#uucp   stream tcp nowait uucp  /usr/sbin/tcpd /usr/lib/uucp/uucico
#comsat dgram  udp  wait   root  /usr/sbin/tcpd /usr/sbin/in.comsat
#
#      Pop      et      al
#
#pop-2  stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.pop2d
```



```
#pop-3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.pop3d
#
# 'cfinger' is for the GNU finger server available for Debian.
# (NOTE: The current implementation of the 'finger' daemon allows it
# to berun as 'root'.)
#
#cfinger stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.cfingerd
#finger stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.fingerd
#netstat stream tcp nowait nobody /usr/sbin/tcpd /bin/netstat
#sysstat stream tcp nowait nobody /usr/sbin/tcpd /bin/ps -auwx
#
# Tftp service is provided primarily for booting. Most sites
# run this only on machines acting as "boot servers."
#
#tftp dgram udp wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd
#tftp dgram udp wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd/boot
#bootps dgram udp wait root /usr/sbin/bootpd bootpd -i -t 120
#
# Kerberos authenticated services (these probably need to be
# corrected)
#
#klogin stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rlogind -k
#eklogin stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rlogind
#-k -x
#kshell stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rshd -k
#
# Services run ONLY on the Kerberos server (these probably need to be
# corrected)
#
#krbupdate stream tcp nowait root /usr/sbin/tcpd /usr/sbin/registerd
#kpasswd stream tcp nowait root /usr/sbin/tcpd /usr/sbin/kpasswdd
#
# RPC based services
#
#mountd/1 dgram rpc/udp wait root /usr/sbin/tcpd /usr/sbin/rpc.mountd
#rstatd/1-3 dgram rpc/udp wait root /usr/sbin/tcpd /usr/sbin/rpc.rstatd
#rusersd/2-3 dgram rpc/udp wait root /usr/sbin/tcpd /usr/sbin/rpc.rusersd
#walld/1 dgram rpc/udp wait root /usr/sbin/tcpd /usr/sbin/rpc.rwalld
```

```
#
# End of inetd.conf.
ident stream tcp nowait nobody /usr/sbin/identd identd -i
```

5.9 Další síťové konfigurační soubory

V Linuxu existuje množství různých síťových konfiguračních souborů, které by vás mohly zajímat. Nebudete je muset nikdy upravovat, ale stojí zato si je popsat, abyste věděli, co obsahují a k čemu slouží.

5.9.1 /etc/protocols

Tento soubor je databází, která mapuje identifikační čísla protokolů na názvy protokolů. Využívají jej programátoři, aby mohli ve svých programech specifikovat protokoly prostřednictvím názvů, a také některé programy (například `tcpdump`), aby na výstupu zobrazily názvy místo čísel. Obecná syntaxe je:

```
jméno číslo přezdívky
```

Soubor `/etc/protocols` je v distribuci *Debian* <<http://www.debian.org/>> v následující podobě:

```
# /etc/protocols:
# $Id: protocols,v 1.1 1995/02/24 01:09:41 imurdock Exp $
#
# Internet (IP) protocols
#
# from: @(#)protocols 5.1 (Berkeley) 4/17/89
#
# Updated for NetBSD based on RFC 1340, Assigned Numbers (July
1992).

ip          0   IP          # internet protocol, pseudo protocol number
icmp       1   ICMP         # internet control message protocol
igmp       2   IGMP         # Internet Group Management
ggp        3   GGP          # gateway-gateway protocol
ipencap    4   IP-ENCAP    # IP encapsulated in IP (officially ''IP'')
st         5   ST           # ST datagram mode
tcp        6   TCP          # transmission control protocol
egp        8   EGP          # exterior gateway protocol
```

```

pup      12 PUP      # PARC universal packet protocol
udp      17 UDP      # user datagram protocol
hmp      20 HMP      # host monitoring protocol
xns-idp  22 XNS-IDP   # Xerox NS IDP
rdp      27 RDP      # "reliable datagram" protocol
iso-tp4  29 ISO-TP4 # ISO Transport Protocol class 4
xtp      36 XTP      # Xpress Transfer Protocol
ddp      37 DDP      # Datagram Delivery Protocol
idpr-cmtp 39 IDPR-CMTP # IDPR Control Message Transport
rspf     73 RSPF     # Radio Shortest Path First.
vmtp     81 VMTP     # Versatile Message Transport
ospf     89 OSPFIGP  # Open Shortest Path First IGP
ipip     94 IPIP     # Yet Another IP encapsulation
encap    98 ENCAP   # Yet Another IP encapsulation

```

5.9.2 /etc/networks

Tento soubor má podobnou funkci, jakou má /etc/hosts. Poskytuje jednoduchou databázi síťových názvů a síťových adres. Jeho formát se liší v tom, že na řádku jsou jen dvě pole a ty vypadají následovně:

```
jméno_sítě   adresa_sítě
```

Příklad může vypadat takto:

```

loopnet 127.0.0.0
localnet 192.168.0.0
amprnet 44.0.0.0

```

Když používáte příkazy, jako je `route`, s místem určení nějaké sítě, o které existuje v /etc/networks záznam, směrovací příkaz zobrazí u sítě místo adresy název.

5.10 Zabezpečení sítě a kontrola přístupu

Dovolte mi začít tuto část upozorněním, že zabezpečení sítě proti nežádoucím útokům je složitě umění. Na tohle téma se necítím příliš povoláním odborníkem. I když je mechanismus, který popíši, jistě přínosem, jestli chcete zabezpečení brát vážně, doporučuji vám používat vlastní postupy, které vám nejlépe vyhovují. Na dané téma je možné z Internetu získat mnoho dobrých odkazů.

Důležité pravidlo zní: „**Nespouštějte servery, které nehodláte používat.**“ Mnoho distribucí má nakonfigurovány různé služby, které se automaticky spustí. Alespoň minimální úroveň zabezpečení zajistíte, když si projdete soubor `/etc/inetd.conf` a vložíte do komentáře (umístěním znaku „#“ na začátek řádku) údaje ke službám, které nehodláte používat. Dobrymi kandidáty jsou služby, jako **shell**, **login**, **exec**, **uucp**, **ftp** a informační služby, jako **finger**, **netstat** a **systat**.

Existují různé druhy zabezpečovacích a přístupových mechanismů, já zde ale popíši jenom ty nejzákladnější.

5.10.1 /etc/ftpusers

Tento soubor je jednoduchým mechanismem, který vám umožňuje pro některé uživatele zakázat přístup na váš stroj pomocí FTP. Soubor je čten ftp-démonem (`ftpd`) při obdržení přicházejícího ftp-spojení. Seznam je jednoduchým seznamem těch uživatelů, kterým není povolen přístup. Může vypadat asi takto:

```
# /etc/ftpusers - uživatelé, kterým není povoleno ftp-spojení
root
uucp
bin
mail
```

5.10.2 /etc/securetty

Soubor, umožňující nastavení zařízení **tty**, na která se může přihlásit **root**. Tento soubor je čten programem `login` (obvykle `/bin/login`). Jeho formátem je seznam názvů zařízení **tty**, na která se **root** může přihlásit (na zbývající nemůže):

```
# /etc/securetty - zařízení tty, na která se může přihlásit root
tty1
tty2
tty3
tty4
```

5.10.3 Mechanismus kontroly přístupu `tcpd`

Program `tcpd`, který jste viděli v `/etc/inetd.conf`, poskytuje přihlašovací a přístupové kontrolní mechanismy službám, ke kterým je nakonfigurován, aby je chránil.

Po vyvolání programem `inetd` čte dva soubory, obsahující přístupová pravidla, a umožňuje nebo zakazuje přístup na server, který chrání.

Prohledává soubory s pravidly, dokud nenalezne první shodu. Jestliže není nalezena žádná shoda, usoudí, že je možné povolit přístup každému. Soubory, které postupně prochází, jsou `/etc/hosts.allow`, `/etc/hosts.deny`. Oba dva ještě popíši. Kompletní popis naleznete na příslušných manuálových stránkách (dobrým začátkem je **host_access(5)**).

/etc/hosts.allow

Konfigurační soubor programu `/usr/sbin/tcpd`. Soubor `hosts.allow` obsahuje pravidla, popisující, kteří hostitelé *mohou* přistoupit ke službám na vašem stroji.

Formát souboru je jednoduchý:

```
# /etc/hosts.allow
#
# <seznam služeb>: <seznam hostitelů> [: příkaz]
```

seznam služeb Čárkami oddělený seznam názvů serverů, na které se tohle pravidlo vztahuje. Příklady názvů serverů jsou: `ftpd`, `telnetd` a `fingerd`.

seznam hostitelů Čárkami oddělený seznam názvů hostitelů. Můžete zde také použít IP-adresy. Názvy hostitelů nebo adresy můžete dodatečně určit pomocí zástupných znaků, aby odpovídaly celým skupinám. Příklady jsou: **gw.vk2ktj.ampr.org**, odpovídající jednomu hostiteli; **uts.edu.au**, odpovídající jakémukoliv názvu, končícímu tímto řetězcem; **44.**, odpovídající jakékoliv IP-adrese, začínající na toto číslo. Existuje pár speciálních slov, která zjednodušují konfiguraci - některé z nich jsou: **ALL** (odpovídá všemu), **LOCAL** (odpovídá všemu, co neobsahuje tečku, což znamená, že je ve stejné doméně jako váš stroj) a **PARANOID** (odpovídá všem hostitelům, jejichž název neodpovídá jejich adrese). Ještě je užitečné **EXCEPT**, umožňující nastavení seznamu s výjimkami. Později uvedeme příklad.

příkaz Operační parametr. Tento parametr určuje plnou cestu k příkazu, spouštěnému vždy, když je pravidlo splněno. Může například spouštět příkaz, který se pokusí identifikovat, kdo je přihlášen na připojícím se hostiteli, nebo generovat zprávu nebo jiné upozornění systémovému správci, že se někdo snaží připojit. Je možné využít množství rozšíření, nejběžnějšími příklady jsou: **%h** se expanduje na název připojovaného hostitele nebo adresu, jestliže nemá název, **%d** na název démona, který je volán.

Příklad:

```
# /etc/hosts.allow
#
# Poštu povolíme všem.
in.smtpd: ALL
# Telnet a ftp povolíme pouze počítačům z naší domény nebo počítači
doma.
telnetd, ftpd: LOCAL, myhost.athome.org.au
# Službu finger povolíme všem, ale poznamenáme si je.
fingerd: ALL: (finger @%h | mail -s "finger from %h" root)
```

/etc/hosts.deny

Konfigurační soubor programu `/usr/sbin/tcpd`. Soubor `hosts.deny` obsahuje pravidla, popisující, kteří hostitelé *nesmí* přistoupit ke službám na vašem stroji.

Jednoduchá ukázka vypadá následovně:

```
# /etc/hosts.deny
#
# Zakážeme všechny služby těm, kteří nemají v pořádku jméno hostitele
ALL: PARANOID
#
# Zakážeme vše
ALL: ALL
```

Údaj **PARANOID** je zde nadbytečný, protože ostatní údaje stejně všechno odchytí. V závislosti na svých požadavcích si některé z těchto údajů můžete nastavit jako implicitní.

Nejbezpečnější konfigurací je v `/etc/hosts.deny` implicitní hodnota **ALL: ALL** a následné umožnění služeb některým hostitelům v `/etc/hosts.allow`.

5.10.4 */etc/hosts.equiv*

Soubor se využívá k zajištění přístupu na váš stroj bez hesla pro některé hostitele nebo uživatele. To je užitečné v bezpečném prostředí, kde ovládáte všechny stroje, jinak je to velmi riskantní. Váš stroj je zabezpečen tak, jako nejméně zabezpečený hostitel, kterému důvěřujete. Aby se zabezpečení zvýšilo, nepoužívejte tento mechanismus a zařídte, aby vaši uživatelé nevyužívali soubor `.rhosts`.

5.10.5 Řádně konfigurujte vašeho ftp-démona

Množství hostitelů využívá anonymní server, umožňující vkládání a stahování souborů bez specifické uživatelské identifikace. Jestliže se rozhodnete tuto možnost nastavit, musíte pro anonymní přístup řádně nakonfigurovat ftp-démona. Většina manuálových stránek pro `ftpd(8)` v určité délce popisuje, jak se s tím vypořádat. Vždy se řiďte doporučenými instrukcemi. Důležitým tipem je nepoužívat kopii vašeho souboru `/etc/passwd` v adresáři `/etc` anonymního účtu. Odstiňte všechny podrobnosti o účtech, kromě těch nejnnutnějších, jinak budete zranitelní metodami rozkódování hesla hrubou silou.

5.10.6 Firewally

Neumožnit datagramům dosáhnout na váš stroj nebo server je skvělý příklad zabezpečení. Viz dokument `<Firewall-HOWTO.html>`.

5.10.7 Další návrhy

Zde je pár dalších, potenciálně zbožných návrhů, které můžete vzít v úvahu.

sendmail

Navzdory popularitě se démon `sendmail` objevuje se železnou pravidelností v hlášeních o nedostatcích v zabezpečení. Je to na vás, ale já bych jej nepoužíval.

NFS a další služby Sun RPC

Pozor na ně. Tyto služby umožňují všechny druhy možných průniků. Ke službám, jako je NFS, se obtížně hledá náhrada, ale pokud je použijete, ujistěte se, komu umožňujete právo připojení.

6 Informace, specifické k síťovým technologiím

Následující části jsou specifické pro určité síťové technologie. Informace zde obsažené se nemusí nutně vztahovat na jiné síťové technologie.

6.1 ARCNet

Názvy zařízení ARCNet jsou „arc0e“, „arc1e“, „arc2e“ atd. nebo „arc0s“, „arc1s“, „arc2s“ atd. První detekované kartě je přiřazen název „arc0e“ nebo „arc0s“ a zbylé názvy jsou přiřazovány postupně v pořadí, v jakém jsou detekovány. Písmeno na konci názvu označuje, zda jste vybrali paket formátu ethernetové zapouzdření nebo paket formátu RFC1051.

Volby překladačného jádra:

```
Network device support  --->
  [*] Network device support
  <*> ARCnet support
  [ ]   Enable arc0e (ARCnet "Ether-Encap" packet format)
  [ ]   Enable arc0s (ARCnet RFC1051 packet format)
```

Jakmile máte sestaveno jádro s podporou ethernetové karty, je již vlastní konfigurace karty jednoduchá:

Zpravidla byste měli použít zápis podobný tomu následujícímu:

```
# ifconfig arc0e 192.168.0.1 netmask 255.255.255.0 up
# route add -net 192.168.0.0 netmask 255.255.255.0 arc0e
```

Podrobnější informace najdete v souborech `/usr/src/linux/Documentation/networking/arcnet.txt` a `/usr/src/linux/Documentation/networking/arcnet-hardware.txt`.

Autorem podpory ARCNet je Avery Pennarun, apenwarr@foxnet.net.

6.2 Appletalk (AF_APPLETALK)

Podpora protokolu Appletalk nepoužívá žádné speciální názvy zařízení, protože využívá síťová zařízení.

Volby překladačného jádra :

```
Networking options  --->
  <*> Appletalk DDP
```

Podpora Appletalk umožňuje vašemu počítači spolupracovat se sítěmi Appletalk. Má důležité využití při sdílení zdrojů jako jsou tiskárny a disky mezi linuxovými počítači a počítači Apple. Potřebujete k tomu dodatečný software, který se nazývá `netatalk`. Wesley Craig netatalk@umich.edu reprezentuje tým nazvaný „Research Systems Unix Group“ na University of Michigan, který tento balík vytvořil. Poskytuje software, který implementuje protokol Appletalk a některé užitečné utility. Balík `netatalk` bude buď součástí vaší distribuce Linuxu nebo si ho musíte stáhnout pomocí služby z University of Michigan <ftp://terminator.rs.itd.umich.edu/unix/netatalk/>.

Balík sestavíte a nainstalujete pomocí následujících příkazů:


```
# cd /usr/src
# tar xvfz ../netatalk-1.4b2.tar.Z
- V tuto chvíli možná bude nutné upravit soubor ,Makefile', přesněji
  změnit proměnnou DESTDIR, která definuje místo pozdější instalaci
  souborů. Implicitní adresář je /usr/local/atalk je ovšem poměrně
  bezpečný.
# make
- jako root:
# make install
```

6.2.1 Konfigurace softwaru Appletalk

Nejprve je třeba zkontrolovat, zda jsou v souboru `/etc/services` příslušné záznamy. Jsou to tyto záznamy:

```
rtmp      1/ddp    # Routing Table Maintenance Protocol
nbp       2/ddp    # Name Binding Protocol
echo      4/ddp    # AppleTalk Echo Protocol
zip       6/ddp    # Zone Information Protocol
```

Dalším krokem je vytvoření konfiguračních souborů Appletalk v adresáři `/usr/local/atalk/etc` (nebo v adresáři, do kterého jste nainstalovali balík Appletalk).

První soubor má název `/usr/local/atalk/etc/atalkd.conf`. Zpočátku stačí, když bude tento soubor obsahovat pouze jediný řádek, na kterém bude uveden název síťového zařízení, které podporuje síť, k níž jsou připojeny vaše počítače Apple:

```
eth0
```

Démon Appletalk do něj po svém spuštění přidá další podrobnosti.

6.2.2 Export souborových systémů prostřednictvím protokolu Appletalk

Souborové systémy z vašeho linuxového počítače lze exportovat do sítě, aby je mohl sdílet počítač Apple připojený k téže síti.

Aby to bylo možné, je třeba nejdříve upravit soubor `/usr/local/atalk/etc/AppleVolumes.system`. Existuje ještě jeden konfigurační soubor jménem `/usr/local/atalk/etc/AppleVolumes`, který má úplně stejný formát a popisuje souborové systémy, které budou mít k dispozici uživatelé přihlašující se na účet `guest`.

Podrobnosti týkající se způsobu konfigurace těchto souborů a popis různých voleb najdete v manuálových stránkách programu `afpd`.

Toto je jednoduchý příklad:

```
/tmp Scratch
/home/ftp/pub "Public Area"
```

Tyto příkazy způsobí export vašeho souborového systému `/tmp` jako svazek AppleShare „Scratch“ a váš veřejný ftp-adresář jako svazek AppleShare „Public Area“. Názvy svazků nejsou povinné, démon si nějaké zvolí.

6.2.3 Sdílení linuxové tiskárny prostřednictvím protokolu Appletalk

Sdílení linuxové tiskárny na počítači Apple je poměrně jednoduché. Potřebujete spustit program `papd`, což je Appletalk Printer Access Protocol Daemon. Tento program bude po spuštění přijímat požadavky počítačů Apple a předávat tiskové úlohy vašemu démonu tiskárny k vytištění.

Konfigurace démona se provádí v souboru `/usr/local/atalk/etc/papd.conf`. Syntaxe tohoto souboru je stejná jako souboru `/etc/printcap`. Název, který přidělíte definici, je registrován pomocí názvového protokolu Appletalk, NBP.

Vzorový konfigurační soubor by mohl například obsahovat následující záznam:

```
TricWriter:\
:pr=lp:op=cg:
```

Ten by zpřístupnil tiskárnu jménem „TricWriter“ vaší síti Appletalk a všechny přijaté úlohy by byly vytištěny na linuxové tiskárně „lp“ (dle definice v souboru `/etc/printcap`) pomocí `lpd`. Zápis „op=cg“ říká, že operátorem tiskárny je uživatel „cg“.

6.2.4 Spuštění softwaru appletalk

Nyní bychom mohli otestovat základní konfiguraci. S balíkem `netatalk` je dodáván soubor `rc.atalk`, který by nám měl vyhovovat, takže stačí přidat jeho cestu do některého startovacího souboru

```
# /usr/local/atalk/etc/rc.atalk
```

a vše by mělo fungovat tak, jak má. Neměly by se objevit žádné chybové zprávy a software by měl posílat zprávy o svém stavu na konzolu.

6.2.5 Otestování softwaru appletalk

Chcete-li otestovat správnou funkci softwaru, vyvolejte na některém z počítačů menu Apple, zvolte položku Chooser, klepněte na AppleShare a mělo by se objevit okno Linux.

6.2.6 Výhrady proti softwaru appletalk

- Možná bude nutné spustit podporu Appletalk před konfigurací sítě IP. Máte-li problémy se spuštěním programů Appletalk nebo po jejich spuštění nefunguje IP-sít, zkuste spustit software Appletalk ještě před spuštěním souboru `/etc/rc.d/rc.inet1`.
- Démon `afpd` (Apple Filing Protocol Daemon) vážně zaneřádí váš pevný disk. Pod přípojnými body vytvoří dvojici adresářů nazvaných `.AppleDesktop` a `Network Trash Folder`. Potom pro každý adresář, do kterého vstoupíte vytvoří adresář `.AppleDouble`, aby do něj mohl uložit resource forks atd. Takže dříve než exportujete adresář `/`, si to pořádně rozmyslete, jinak budete mít spoustu práce s čištěním disku.
- Program `afpd` očekává od počítačů Apple hesla v čistě textové podobě. To může být vážný bezpečnostní problém, takže pokud budete spouštět tohoto démona na počítači připojenému k Internetu, můžete za případný průnik do systému vinit jen sami sebe.
- Existující diagnostické nástroje, jako je `netstat` a `ifconfig`, nepodporují Appletalk. Hrubé informace najdete v adresáři `/proc/net/`.

6.2.7 Více informací

Podrobnější popis způsobu konfigurace softwaru Appletalk pro Linux najdete na stránce *Linux Netatalk-HOWTO* Andrease Brownwothe na adrese <http://thehamptons.com/anders/netatalk/>.

6.3 ATM

Werner Almesberger <werner.almesberger@lrc.di.epfl.ch> vede projekt, který by umožnil podporu režimu ATM (Asynchronous Transfer Mode) v Linuxu. Aktuální informace o stavu tohoto projektu získáte na adrese <http://lrcwww.epfl.ch/linux-atm/>.

6.4 AX25 (AF_AX25)

Názvy zařízení protokolu AX.25 jsou ve verzi 2.0 „sl0“, „sl1“ atd. nebo „ax0“, „ax1“ atd. ve verzi jádra 2.1.*.

Volby kompilace jádra:

```
Networking options --->
[*] Amateur Radio AX.25 Level 2
```

Protokol AX25, Netrom a Rose jsou popisovány v dokumentu *AX25-HOWTO* <AX25-HOWTO.html>. Tyto protokoly používají radioamatéři po celém světě k experimentům s paketovým rádiem.

Velký díl práce při implementaci těchto protokolů má na svědomí Jonathon Naylor, jsn@cs.nott.ac.uk.

6.5 DECNet

Na podpoře protokolu DECNet se v současné době pracuje. Předpokládá se, že se objeví ve verzi jádra č. 2.1.*.

6.6 EQL – multiple line traffic equaliser

Název zařízení protokolu EQL je „eql“. Ve standardním zdrojovém kódu jádra můžete mít pouze jediné zařízení EQL na počítač. EQL poskytuje způsob využití více linek point to point, např. PPP, SLIP nebo PLIP, jako jediného logického spojení pro přenos protokolu TCP/IP. Vždy je levnější použít větší množství pomalejších linek, než mít nainstalovanou jedinou vysokorychlostní linku.

Volby kompilace jádra

```
Network device support --->
[*] Network device support
<*> EQL (serial line load balancing) support
```

Aby mohl být tento mechanismus podporován, musí počítač na druhém konci spojení také podporovat EQL. Linux, Livingstone Portmasters a novější přístupové servery podporují kompatibilní prostředky.

Ke konfiguraci EQL budeme potřebovat nástroje `eql`, které lze získat na adrese <ftp://sunsite.unc.edu/pub/linux/system/Serial/eql-1.2.tar.gz>.

Konfigurace je poměrně jednoduchá. Začnete konfigurací rozhraní `eql`. Je to rozhraní jako každé jiné síťové rozhraní. Pomocí utility `ifconfig` nastavíte IP-adresu a `mtu`, například takto:

```
ifconfig eql 192.168.10.1 mtu 1006
```

Potom musíte ručně navázat všechna spojení, která budete používat. Může se jednat o kombinaci síťových služeb point to point. Způsob navázání spojení bude záviset na jejich typu. Podrobnosti najdete v příslušných statích tohoto dokumentu.

nakonec je potřeba sdružit sériové spojení se zařízením EQL, kterému se říká „enslaving“ a provádí se příkazem `eql_enslave` níže uvedeným způsobem:

```
eql_enslave eql sl0 28800
eql_enslave eql ppp0 14400
```

Parametr „*estimated speed*“, který předáte programu `eql_enslave`, sám o sobě nic neprovádí. Ovladač EQL podle něj určí, jaký typ sdílení datagramů má zařízení obdržet, takže pomocí této hodnoty můžete doladit zatížení linek.

Ke zrušení přiřazení mezi linkou a zařízením EQL slouží příkaz `eql_emancipate`, který se použije následujícím způsobem:

```
eql_emancipate eql sl0
```

Směrování přidáte stejně jako byste to provedli pro kterékoliv jiné spojení point to point, jen vaše trasy by spíše než na skutečná sériová zařízení měly odkazovat na zařízení eql. Typický zápis bude vypadat takto:

```
route add default eql
```

Ovladač EQL napsal Simon Janes, simon@ncm.com.

6.7 Ethernet

Názvy ethernetových zařízení jsou „eth0“, „eth1“, „eth2“ atd. První detekované kartě je přiřazen název „eth0“ a delším pak názvy v tom pořadí, v jakém jsou detekovány.

Jak rozchodit vaši ethernetovou kartu pod Linuxem se dozvíte v dokumentu *Ethernet-HOWTO*.

Jakmile máte sestavené jádro s podporou vaší ethernetové karty, je již vlastní konfigurace karty jednoduchá.

Typicky použijete následující zápis:

```
# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
# route add -net 192.168.0.0 netmask 255.255.255.0 eth0
```

Většinu ethernetových ovladačů vytvořil Donald Becker, becker@CESDIS.gsfc.nasa.gov.

6.8 FDDI

Názvy zařízení FDDI jsou „fddi0“, „fddi1“, „fddi2“ atd. První detekované kartě je přiřazen název „fddi0“ a delším pak postupně názvy v tom pořadí, v jakém jsou detekovány.

Larry Stefani, lstefani@ultranet.com, napsal ovladač pro karty Digital Equipment Corporation FDDI EISA a PCI.

Volby kompilace jádra

```
Network device support --->
[*] FDDI driver support
[*] Digital DEFEA and DEFPA adapter support
```

Jakmile máte sestavené jádro s podporou ovladače FDDI, je vlastní konfigurace rozhraní FDDI takřka identická s konfigurací ethernetového rozhraní. Stačí zadat příslušný název rozhraní FDDI příkazům `ifconfig` a `route`.

6.9 Frame Relay

Názvy zařízení Frame Relay jsou „dlci00“, „dlci01“ atd pro DLCI-zařízení zapouzdření a „sd-la0“, „sdla1“ atd. pro FRAD.

Frame Relay je nová síťová technologie, která je navržena pro datové komunikace, které mají impulzivní nebo střídavou povahu. K síti Frame Relay se připojíte pomocí zařízení zvaného Frame Relay Access Device (FRAD). Frame Relay pro Linux podporuje IP přes Frame Relay, dle popisu v dokumentu RFC-1490.

Volby kompilace jádra

```
Network device support --->
<*> Frame relay DLCI support (EXPERIMENTAL)
(24) Max open DLCI
(8) Max DLCI per device
<*> SDLA (Sangoma S502/S508) support
```

Autorem podpory Frame Relay a konfiguračních nástrojů je Mike McLagan, mike.mclagan@linux.org.

V současné době je FRAD podporován pouze firmou Sangoma Technologies
<<http://www.sangoma.com/>> S502A, S502E a S508.

Ke konfiguraci zařízení FRAD a DLCI po sestavení jádra budete potřebovat konfigurační nástroje Frame Relay. Ty lze získat na adrese <ftp://ftp.invlogic.com/pub/linux/fr/frad-0.15.tgz>. Kompilace a instalace těchto nástrojů je poměrně jednoduchá, ale díky absenci souboru Makefile je to potřeba provést manuálně:

```
# cd /usr/src
# tar xvfz ../frad-0.15.tgz
# cd frad-0.15
# for i in common dlci frad; do make -C $i clean; make -C $i; done
# mkdir /etc/frad
# install -m 644 -o root -g root bin/*.sfm /etc/frad
# install -m 700 -o root -g root frad/fradcfg /sbin
# install -m 700 -o root -g root dlci/dlcicfg /sbin
```

Po nainstalování těchto nástrojů je třeba vytvořit soubor `/etc/frad/router.conf`. Můžete k tomu využít tuto šablonu, která je upravenou verzí jednoho z příkladů:

```
# /etc/frad/router.conf
# This is a template configuration for frame relay.
# All tags are included. The default values are based on the code
# supplied with the DOS drivers for the Sangoma S502A card.
#
# A '#' anywhere in a line constitutes a comment
# Blanks are ignored (you can indent with tabs too)
# Unknown [] entries and unknown keys are ignored
#

[Devices]
Count=1                # number of devices to configure
Dev_1=sdla0            # the name of a device
#Dev_2=sdla1           # the name of a device

# Specified here, these are applied to all devices and can be
# overridden for
# each individual board.
#
```

```
Access=CPE
Clock=Internal
KBaud=64
Flags=TX
#
# MTU=1500          # Maximum transmit IFrame length, default is # #
#                  # 4096
# T391=10          # T391 value      5 - 30, default is 10
# T392=15          # T392 value      5 - 30, default is 15
# N391=6           # N391 value      1 - 255, default is 6
# N392=3           # N392 value      1 - 10, default is 3
# N393=4           # N393 value      1 - 10, default is 4

# Specified here, these set the defaults for all boards
# CIRfwd=16        # CIR forward     1 - 64
# Bc_fwd=16        # Bc forward      1 - 512
# Be_fwd=0         # Be forward      0 - 511
# CIRbak=16        # CIR backward    1 - 64
# Bc_bak=16        # Bc backward     1 - 512
# Be_bak=0         # Be backward     0 - 511

# Device specific configuration
#

# The first device is a Sangoma S502E
#
[sdla0]
Type=Sangoma      # Type of the device to configure, currently
                  # only SANGOMA is recognised

#
# These keys are specific to the 'Sangoma' type
#
# The type of Sangoma board - S502A, S502E, S508
Board=S502E
#
# The name of the test firmware for the Sangoma board
# Testware=/usr/src/frad-0.10/bin/sdla_tst.502
#
```



```
# The name of the FR firmware
# Firmware=/usr/src/frad-0.10/bin/frm_rel.502
#
Port=360          # Port for this particular card
Mem=C8           # Address of memory window, A0-EE, depending
                 # on card
IRQ=5            # IRQ number, do not supply for S502A
DLCIs=1          # Number of DLCI's attached to this device
DLCI_1=16        # DLCI #1's number, 16 - 991
# DLCI_2=17
# DLCI_3=18
# DLCI_4=19
# DLCI_5=20
#
# Specified here, these apply to this device only,
# and override defaults from above
#
# Access=CPE      # CPE or NODE, default is CPE
# Flags=TXIgnore,RXIgnore,BufferFrames,DropAborted,Stats,MCI,AutoDLCI
# Clock=Internal  # External or Internal, default is Internal
# Baud=128        # Specified baud rate of attached CSU/DSU
# MTU=2048        # Maximum transmit IFrame length, default is # #
                 # 4096
# T391=10         # T391 value      5 - 30, default is 10
# T392=15         # T392 value      5 - 30, default is 15
# N391=6          # N391 value      1 - 255, default is 6
# N392=3         # N392 value      1 - 10, default is 3
# N393=4         # N393 value      1 - 10, default is 4
#
# The second device is some other card
#
# [sdla1]
# Type=FancyCard  # Type of the device to configure.
# Board=          # Type of Sangoma board
# Key=Value       # values specific to this type of device
#
```

```
# DLCI Default configuration parameters
# These may be overridden in the DLCI specific configurations
#
CIRfwd=64          # CIR forward    1 - 64
# Bc_fwd=16        # Bc forward    1 - 512
# Be_fwd=0         # Be forward    0 - 511
# CIRbak=16        # CIR backward  1 - 64
# Bc_bak=16        # Bc backward   1 - 512
# Be_bak=0         # Be backward   0 - 511

#
# DLCI Configuration
# These are all optional. The naming convention is
# [DLCI_D<devicenum>_<DLCI_Num>]
#

[DLCI_D1_16]
# IP=
# Net=
# Mask=
# Flags defined by Sangoma: TXIgnore,RXIgnore,BufferFrames
# DLCIFlags=TXIgnore,RXIgnore,BufferFrames
# CIRfwd=64
# Bc_fwd=512
# Be_fwd=0
# CIRbak=64
# Bc_bak=512
# Be_bak=0

[DLCI_D2_16]
# IP=
# Net=
# Mask=
# Flags defined by Sangoma: TXIgnore,RXIgnore,BufferFrames
# DLCIFlags=TXIgnore,RXIgnore,BufferFrames
# CIRfwd=16
# Bc_fwd=16
# Be_fwd=0
```

```
# CIRbak=16
# Bc_bak=16
# Be_bak=0
```

Po sestavení souboru `/etc/frad/router.conf` zbývá nakonfigurovat vlastní zařízení. Pouze tato část je o něco obtížnější, než konfigurace normálních síťových zařízení. Nesmíte zapomenout předložit zařízení FRAD ještě před zapouzdřenými zařízeními DLCI.

```
# Configure the frad hardware and the DLCI parameters
/sbin/fradcfg /etc/frad/router.conf || exit 1
/sbin/dlcicfg file /etc/frad/router.conf
#
# Bring up the FRAD device
ifconfig sdla0 up
#
# Configure the DLCI encapsulation interfaces and routing
ifconfig dlci00 192.168.10.1 pointopoint 192.168.10.2 up
route add -net 192.168.10.0 netmask 255.255.255.0 dlci00
#
ifconfig dlci01 192.168.11.1 pointopoint 192.168.11.2 up
route add -net 192.168.11.0 netmask 255.255.255.0 dlci00
#
route add default dev dlci00
#
```

6.10 IP-účetnictví (accounting)

IP-účetnictví jádra Linuxu vám umožňuje shromažďovat a analyzovat některá data týkající se využití sítě. Mezi sesbíraná data patří počet paketů a počet bajtů nahromaděných od posledního vynulování součtů. Ke kategorizaci součtů lze zadat množství nejrůznějších pravidel tak, aby výsledek vyhovoval vašim potřebám.

Volby kompilace jádra

```
Networking options --->
[*] IP: accounting
```

Po kompilaci a instalaci jádra je třeba pomocí příkazu `ipfwadm` nastavit IP-účetnictví. Existuje mnoho různých způsobů specifikací účetnictví, z nichž můžete vybírat. Vybral jsem jednoduchý příklad zahrnující to podstatné. Další informace najdete v manuálových stránkách příkazu `ipfwadm`.

Scénář: Máte ethernetovou síť, která je připojena k Internetu prostřednictvím spojení PPP. Na Ethernetu máte počítač, který nabízí několik služeb, a zajímá vás, jaký provoz způsobuje telnet, rlogin, FTP a WWW.

Můžete použít následující sadu příkazů:

```
#
# Vyprázdnit účetní pravidla
ipfwadm -A -f
#
# Pravidla pro lokální ethernetový segment
ipfwadm -A in -a -P tcp -D 44.136.8.96/29 20
ipfwadm -A out -a -P tcp -S 44.136.8.96/29 20
ipfwadm -A in -a -P tcp -D 44.136.8.96/29 23
ipfwadm -A out -a -P tcp -S 44.136.8.96/29 23
ipfwadm -A in -a -P tcp -D 44.136.8.96/29 80
ipfwadm -A out -a -P tcp -S 44.136.8.96/29 80
ipfwadm -A in -a -P tcp -D 44.136.8.96/29 513
ipfwadm -A out -a -P tcp -S 44.136.8.96/29 513
ipfwadm -A in -a -P tcp -D 44.136.8.96/29
ipfwadm -A out -a -P tcp -D 44.136.8.96/29
ipfwadm -A in -a -P udp -D 44.136.8.96/29
ipfwadm -A out -a -P udp -D 44.136.8.96/29
ipfwadm -A in -a -P icmp -D 44.136.8.96/29
ipfwadm -A out -a -P icmp -D 44.136.8.96/29
#
# Pravidla pro ostatní
ipfwadm -A in -a -P tcp -D 0/0 20
ipfwadm -A out -a -P tcp -S 0/0 20
ipfwadm -A in -a -P tcp -D 0/0 23
ipfwadm -A out -a -P tcp -S 0/0 23
ipfwadm -A in -a -P tcp -D 0/0 80
ipfwadm -A out -a -P tcp -S 0/0 80
ipfwadm -A in -a -P tcp -D 0/0 513
```

```

ipfwadm -A out -a -P tcp -S 0/0 513
ipfwadm -A in -a -P tcp -D 0/0
ipfwadm -A out -a -P tcp -D 0/0
ipfwadm -A in -a -P udp -D 0/0
ipfwadm -A out -a -P udp -D 0/0
ipfwadm -A in -a -P icmp -D 0/0
ipfwadm -A out -a -P icmp -D 0/0
#
# Výpis pravidel
ipfwadm -A -l -n
#

```

Poslední příkaz vypíše všechna pravidla účetnictví a zobrazí shromážděné souhrny.

Je důležité vědět, že při analýze IP-účetnictví se budou **celkové výsledky všech vyhovujících pravidel zvyšovat**, takže k získání různých součtů je třeba příslušná matematika. Pokud bych chtěl například vědět, jakou část tvořila data, která nepatří ke službě FTP, telnet, rlogin nebo WWW, musel bych odečíst jednotlivé součty od pravidla, které vyhovuje všem portům.

```

# ipfwadm -A -l -n
  IP accounting rules
pkts  bytes  dir   prot  source          destination      ports
  0      0   in    tcp   0.0.0.0/0       44.136.8.96/29  * -> 20
  0      0   out   tcp   44.136.8.96/29  0.0.0.0/0       20 -> *
  0      0   in    tcp   0.0.0.0/0       44.136.8.96/29  * -> 23
  0      0   out   tcp   44.136.8.96/29  0.0.0.0/0       23 -> *
  10     1166  in    tcp   0.0.0.0/0       44.136.8.96/29  * -> 80
  10     572  out   tcp   44.136.8.96/29  0.0.0.0/0       80 -> *
  242    9777  in    tcp   0.0.0.0/0       44.136.8.96/29  * -> 513
  220   18198  out   tcp   44.136.8.96/29  0.0.0.0/0       513 -> *
  252   10943  in    tcp   0.0.0.0/0       44.136.8.96/29  * -> *
  231   18831  out   tcp   0.0.0.0/0       44.136.8.96/29  * -> *
  0      0   in    udp   0.0.0.0/0       44.136.8.96/29  * -> *
  0      0   out   udp   0.0.0.0/0       44.136.8.96/29  * -> *
  0      0   in    icmp  0.0.0.0/0       44.136.8.96/29  *
  0      0   out   icmp  0.0.0.0/0       44.136.8.96/29  *
  0      0   in    tcp   0.0.0.0/0       0.0.0.0/0       * -> 20
  0      0   ou    tcp   0.0.0.0/0       0.0.0.0/0       20 -> *
  0      0   in    tcp   0.0.0.0/0       0.0.0.0/0       * -> 23

```

```

0      0  out  tcp  0.0.0.0/0      0.0.0.0/0      23 -> *
10     1166 in  tcp  0.0.0.0/0      0.0.0.0/0      * -> 80
10     572  out  tcp  0.0.0.0/0      0.0.0.0/0      80 -> *
243    9817 in  tcp  0.0.0.0/0      0.0.0.0/0      * -> 513
221    18259 out  tcp  0.0.0.0/0      0.0.0.0/0      513 -> *
253    10983 in  tcp  0.0.0.0/0      0.0.0.0/0      * -> *
231    18831 out  tcp  0.0.0.0/0      0.0.0.0/0      * -> *
0      0  in   udp  0.0.0.0/0      0.0.0.0/0      * -> *
0      0  out  udp  0.0.0.0/0      0.0.0.0/0      * -> *
0      0  in   icmp 0.0.0.0/0      0.0.0.0/0      *
0      0  out  icmp 0.0.0.0/0      0.0.0.0/0      *
#
```

6.11 Přidělování IP-přezdívek

U některých aplikací je výhodné přidělit jednomu síťovému zařízení více IP-adres. Poskyvatelé internetových služeb využívají tuto vlastnost k poskytování zákaznických nabídek služeb World Wide Web a FTP.

Volby kompilace jádra

```

Networking options  --->
    ....
    [*] Network aliasing
    ....
    <*> IP: aliasing support
```

Po kompilaci a instalaci jádra s podporou IP_Alias je vlastní konfigurace velice jednoduchá. Přezdívky (aliasy) jsou přidány k virtuálním síťovým zařízením sdruženým se skutečným síťovým zařízením. Jednoduchá názvová konvence pak vypadá třeba takto: <název zařízení>:<číslo virtuálního zařízení>, tj. eth0:0, ppp0:10 atd.

Předpokládejme například, že máte ethernetovou síť, která současně podporuje dvě různé podsítě IP, a vy chcete, aby měl váš počítač přímý přístup k oběma. V takovém případě můžete použít tento zápis:

```

#
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
```

```
# ifconfig eth0:0 192.168.10.1 netmask 255.255.255.0 up
# route add -net 192.168.10.0 netmask 255.255.255.0 eth0:0
#
```

Budete-li chtít odstranit nějakou přezdívku, stačí přidat na konec jejího názvu pomlčku, jako v následujícím výpisu:

```
# ifconfig eth0:0- 0
```

Všechny trasy sdružené s tímto aliasem budou také automaticky odstraněny.

6.12 IP-firewall

IP-firewall a problémy kolem firewallů jsou podrobněji probírány v dokumentu *Firewall-HOWTO*. IP-firewall chrání váš počítač před neautorizovaným síťovým přístupem, což provádějí filtrováním nebo propouštěním datagramů z a na vámi nominované IP-adresy. Existují tři různé třídy pravidel: příchozí filtrování, odchozí filtrování a postupující filtrování. Příchozí pravidla jsou aplikována na datagramy, které obdrží síťové zařízení. Odchozí pravidla jsou aplikována na datagramy, které mají být síťovým zařízením přenesena. Postupující pravidla jsou aplikována na datagramy, které zařízení obdrží, ale nejsou určeny pro daný počítač, tj. datagramy, které je třeba nasměrovat.

Volby kompilace jádra

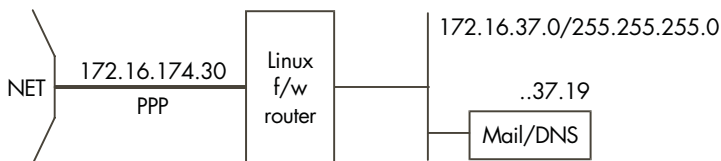
```
Networking options  --->
    [*] Network firewalls
    ....
    [*] IP: forwarding/gatewaying
    ....
    [*] IP: firewalling
    [ ] IP: firewall packet logging
```

Konfigurace pravidel IP-firewallu se provádí pomocí příkazu `ipfwadm`. Jak jsem již uvedl, nepatřím v otázkách bezpečnosti mezi experty, takže ačkoli vám zde představím jednoduchý příklad, který můžete použít, je-li pro vás bezpečnost prvořadá, měli byste zkoumat sami na vlastní pěst a vytvořit si svá vlastní pravidla.

Pravděpodobně nejběžnějším využitím IP-firewallu je případ, kdy používáte váš linuxový počítač jako router a firewallovou bránu k ochraně vaší lokální počítačové sítě před neautorizovaným přístupem z vnějšku.

Následující konfigurace vychází z příspěvku, jehož autorem je Arnt Gulbrandsen, <agulbra@troll.no>.

Příklad popisuje konfiguraci pravidel firwallu na počítači-firwallu/bráně ilustrovaném na tomto diagramu:



Následující příkazy by byly normálně umístěny v souboru `rc`, aby došlo k jejich automatickému spuštění při startu systému. Pro zajištění maximální bezpečnosti by měly být tyto příkazy provedeny až po konfiguraci síťových rozhraní, ale před vlastním spuštěním těchto zařízení, aby nemohl nikdo získat přístup v době, kdy počítač startuje.

```

#!/bin/sh

# Flush the 'Forwarding' rules table
# Change the default policy to 'accept'
#
/sbin/ipfwadm -F -f
/sbin/ipfwadm -F -p accept
#
# .. and for 'Incoming'
#
/sbin/ipfwadm -I -f
/sbin/ipfwadm -I -p accept

# First off, seal off the PPP interface
# I'd love to use '-a deny' instead of '-a reject -y' but then it
# would be impossible to originate connections on that interface
# too.
# The -o causes all rejected datagrams to be logged. This trades
# disk space against knowledge of an attack of configuration error.
#
/sbin/ipfwadm -I -a reject -y -o -P tcp -S 0/0 -D 172.16.174.30

# Throw away certain kinds of obviously forged packets right away:

```



```
# Nothing should come from multicast/anycast/broadcast addresses
#
/sbin/ipfwadm -F -a deny -o -S 224.0/3 -D 172.16.37.0/24
#
# and nothing coming from the loopback network should ever be
# seen on a wire
#
/sbin/ipfwadm -F -a deny -o -S 127.0/8 -D 172.16.37.0/24

# accept incoming SMTP and DNS connections, but only
# to the Mail/Name Server
#
/sbin/ipfwadm -F -a accept -P tcp -S 0/0 -D 172.16.37.19 25 53
#
# DNS uses UDP as well as TCP, so allow that too
# for questions to our name server
#
/sbin/ipfwadm -F -a accept -P udp -S 0/0 -D 172.16.37.19 53
#
# but not "answers" coming to dangerous ports like NFS and
# Larry McVoy's NFS extension.  If you run squid, add its port here.
#
/sbin/ipfwadm -F -a deny -o -P udp -S 0/0 53 \
-D 172.16.37.0/24 2049 2050

# answers to other user ports are okay
#
/sbin/ipfwadm -F -a accept -P udp -S 0/0 53 \
-D 172.16.37.0/24 53 1024:65535

# Reject incoming connections to identd
# We use 'reject' here so that the connecting host is told
# straight away not to bother continuing, otherwise we'd experience
# delays while ident timed out.
#
/sbin/ipfwadm -F -a reject -o -P tcp -S 0/0 -D 172.16.37.0/24 113

# Accept some common service connections from the 192.168.64 and
```

```
# 192.168.65 networks, they are friends that we trust.
#
/sbin/ipfwadm -F -a accept -P tcp -S 192.168.64.0/23 \
              -D 172.16.37.0/24 20:23

# accept and pass through anything originating inside
#
/sbin/ipfwadm -F -a accept -P tcp -S 172.16.37.0/24 -D 0/0

# deny most other incoming TCP connections and log them
# (append 1:1023 if you have problems with ftp not working)
#
/sbin/ipfwadm -F -a deny -o -y -P tcp -S 0/0 -D 172.16.37.0/24

# ... for UDP too
#
/sbin/ipfwadm -F -a deny -o -P udp -S 0/0 -D 172.16.37.0/24
```

Správná konfigurace firewallu je poměrně obtížná. Tento příklad by vám měl posloužit jako odrazový můstek. Manuálové stránky příkazu `ipfwadm` nabízí jistou pomoc při používání tohoto nástroje. Budete-li konfigurovat firewall, poptejte se kolem a snažte se získat co nejvíce rad ze zdrojů, které považujete za spolehlivé, a požádejte někoho, aby rozumně otestoval vaši konfiguraci z vnějšku.

6.13 IPIP-zapouzdření

Proč byste chtěli zapouzdřovat IP-datagramy do IP-datagramů? Pokud jste o tom zatím neslyšeli, bude vám to asi připadat zvláštní. Zde je tedy dvojice využití: Mobile-IP a IP-Multicast. Pravděpodobně nejrozšířenější využití má tato technika v nejméně známé oblasti, kterou je amatérské rádio.

Volby kompilace jádra

```
Networking options  --->
  [*] TCP/IP networking
  [*] IP: forwarding/gatewaying
  ....
  <*> IP: tunneling
```

Zařízení provádějící IP-tunelování se nazývají „tunl0“, „tunl1“ atd.

„Ale proč?“ No protože konvenční IP-směrování nařizuje, že síťová IP-adresa se skládá ze síťové adresy a síťové masky. Takto vzniká množství po sobě jdoucích adres, které mohou být všechny směrovány přes jeden směrovací záznam. To je poměrně výhodné, ale znamená to, že všechny konkrétní IP-adresy můžete používat jen v případě, kdy jste připojeni ke konkrétní části sítě, které tyto adresy náleží. Ve většině případů je to v pořádku, ale patříte-li k mobilním uživatelům sítě, pak nemůžete zůstat po celou dobu připojen k jednomu místu. IPIP zapouzdření (IP-tunelování) dovoluje toto omezení překonat tím, že dovolí IP-datagramy určené pro vaši IP-adresu zabalit a přeměřovat na jinou IP-adresu. Víte-li dopředu, že budete nějakou dobu pracovat v jiné IP-síti, můžete nastavit počítač ve vaší domovské síti tak, aby přijímal datagramy určené pro vaši IP-adresu a přeměřoval je na adresu, kterou budete ve skutečnosti dočasně používat.

6.13.1 Konfigurace tunelované sítě

Věřím, že diagram mi jako vždy ušetří spoustu matoucího textu, takže tady je:

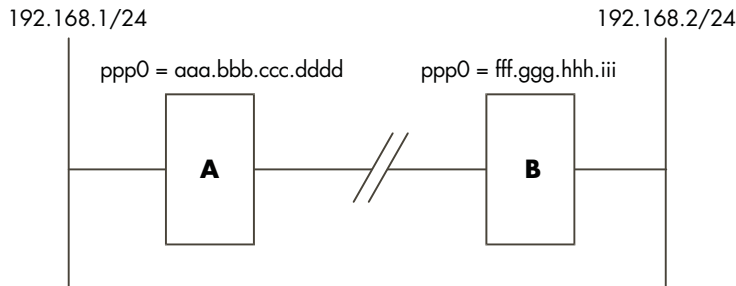


Diagram ilustruje další možný důvod použití IPIP zapouzdření, kterým je virtuální soukromá síť (VPN). Tento příklad předpokládá, že máte dva počítače, každý s jedním vytáčeným internetovým připojením. Každému hostiteli je přidělena právě jedna IP-adresa. Za těmito počítači je určitá soukromá lokální počítačová síť s rezervovanými síťovými IP-adresami. Dejme tomu, že chcete libovolnému uživateli v síti A povolit připojení k libovolnému hostiteli v síti B, stejně jako by byli řádně připojeni k Internetu prostřednictvím síťového routeru. Právě tento stav nám pomůže dosáhnout IPIP zapouzdření. Všimněte si, že zapouzdření neřeší problém ohledně způsobu, jakým hostitelé v sítích A a B komunikují s kterýmkoliv jiným hostitelem v Internetu. Stále k tomu budete potřebovat triky typu IP-maškaráda (Masquerade). Zapouzdřování normálně provádí počítač fungující jako router.

Linuxový router „A“ by byl nakonfigurován takto:

```
#
PATH=/sbin:/usr/sbin
```

```
#
# Ethernet configuration
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.1 up
route add -net 192.168.2.0 netmask 255.255.255.0 gw fff.ggg.hhh.iii tunl0
```

Linuxový router „B“ by byl nakonfigurován takto:

```
#
PATH=/sbin:/usr/sbin
#
# Ethernet configuration
ifconfig eth0 192.168.2.1 netmask 255.255.255.0 up
route add -net 192.168.2.0 netmask 255.255.255.0 eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.2.1 up
route add -net 192.168.1.0 netmask 255.255.255.0 gw aaa.bbb.ccc.ddd tunl0
```

Příkaz

```
route add -net 192.168.1.0 netmask 255.255.255.0 gw aaa.bbb.ccc.ddd tunl0
```

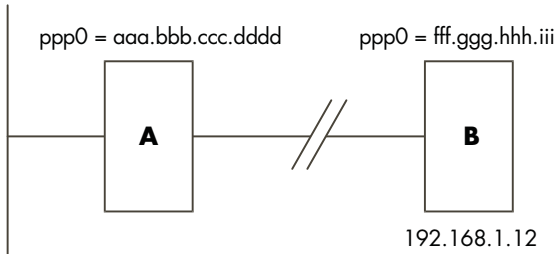
říká: „Všechny datagramy určené pro adresu 192.168.1.0/24 pošli do IPIP zapouzdřeného datagramu s cílovou adresou aaa.bbb.ccc.ddd.“

Všimněte si, že na sebe obě konfigurace působí. Tunelovací zařízení používá „gw“ v trase jako cíl IP-datagramu, kam pošle datagram, který obdrželo pro směrování. Tento počítač musí umět zapouzdřovat IPIP datagramy, tj. musí být také nakonfigurován s tunelovacím zařízením.

6.13.2 Konfigurace tunelovacího hostitele

Směrování nemusí pokrývat celou síť. Můžete třeba směřovat pouze jedinou IP-adresu. V takovém případě stačí nastavit zařízení tunl na „vzdáleném“ počítači s jeho IP-adresou a na konci A použít hostitelský router (a Proxy Arp) a nikoli síťový router přes tunelovací zařízení. Pojďme tedy naši konfiguraci příslušným způsobem upravit. Nyní máme pouze hostitele „B“, u kterého chceme, aby byl připojen k Internetu a zároveň byl součástí vzdálené sítě podporované hostitelem „A“:

192.168.1/24



Linuxový router „A“ by byl nakonfigurován takto:

```
#
PATH=/sbin:/usr/sbin
#
# Ethernet configuration
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.1 up
route add -host 192.168.1.12 gw fff.ggg.hhh.iii tunl0
#
# Proxy ARP for the remote host
arp -s 192.168.1.12 xx:xx:xx:xx:xx:xx pub
```

Linuxový router „B“ by byl nakonfigurován takto:

```
#
PATH=/sbin:/usr/sbin
#
# ppp0 configuration (start ppp link, set default route)
pppd
route add default ppp0
#
# Tunnel device configuration
ifconfig tunl0 192.168.1.12 up
route add -net 192.168.1.0 netmask 255.255.255.0 \
    gw aaa.bbb.ccc.ddd tunl0
```

Tento druh konfigurace je typičtější pro aplikace Mobile-IP, kdy se chce jeden hostitel potulovat po Internetu a přitom si po celou dobu zachovat jedinou použitelnou IP-adresu. Zajímá-li vás praktická stránka tohoto tématu, pak si přečtěte stať Mobile-IP, kde najdete více informací.

6.14 IPX (AF_IPX)

Protokol IPX je častěji využíván v lokálních počítačových sítích Novell NetWare(tm). Linux obsahuje podporu tohoto protokolu a lze ho nakonfigurovat tak, aby se choval jako síťový koncový bod nebo router IPX.

Volby kompilace jádra

```
Networking options  --->
    [*] The IPX protocol
    [ ] Full internal IPX network
```

Protokol IPX a NCPFS jsou podrobněji rozebírány v dokumentu *IPX-HOWTO*.

6.15 IPv6

Když už máte pocit, že začínáte rozumět protokolu IP, změní se síťová pravidla. IPv6 je zkratka 6. verze internetového protokolu (IP). Tento protokol byl vyvinut zejména k vyřešení nedostatku IP-adres v Internetu. Adresy protokolu IPv6 jsou dlouhé 16 bajtů (128 bitů). Protokol IPv6 obsahuje také několik dalších změn, většinou se jedná o zjednodušení, což přispěje k lepší ovladatelnosti sítí IPv6 ve srovnání se sítěmi IPv4.

V jádru Linuxu 2.1.* již je zabudována fungující, nikoli však úplná, implementace protokolu IPv6.

Pokud si chcete zaexperimentovat s internetovou technologií příští generace nebo ji už potřebujete, měli byste si přečíst dokument IPv6-FAQ, který najdete na adrese <http://www.terra.net/ipv6/>.

6.16 ISDN

ISDN (Intergrated Services Digital Network) je sada standardů, které specifikují přepínané digitální datové sítě pro obecné použití. „Volání“ ISDN vytvoří synchronní datovou službu point to point směrem k cíli. ISDN je obecně posílán po vysokorychlostních linkách, které jsou rozděleny na několik oddělených kanálů. Rozeznáváme dva různé typy kanálů, tzv. „B kanály“ slouží k vlastnímu přenosu dat a jediný „D kanál“, který slouží k posílání řídicích informací službě ISDN o navázání spojení a dalších funkcí. V Austrálii například může být ISDN doručován po lince o rychlosti 2 Mbps, která je rozdělena na 30 samostatných B kanálů o rychlosti 64 kbps a jeden D kanál o rychlosti 64 kbps. Současně může být využíván libovolný počet a kombinace kanálů. Je možné například navázat 30 samostatných hovorů o rychlosti 64 kbps s 30 různými místy nebo 15 hovorů s 15 různými místy o rychlosti 128 kbps (každý hovor bude využívat dva kanály) případně jen malý počet hovorů a zbytek kanálů nechat zahálet. Kanál může být využíván buď pro příchozí, nebo odchozí hovor. Původním záměrem ISDN bylo umožnit telekomunikačním společnostem poskytovat jedinou datovou službu, která by umožnila doručování buď telefonních (prostřednictvím digitalizace hlasu), nebo datových služeb do domácností nebo firem bez nutnosti provádění zvláštních změn v konfiguraci.

Existují dva různé způsoby, jakými se může váš počítač připojit ke službě ISDN. První způsob spočívá ve využití zařízení zvaného „Terminal Adaptor“ (terminálový adaptér), který se zapojí do jednoho ze sériových rozhraní Network Terminating Unit, kterou vám nainstalovali pracovníci telekomunikací při zavádění služby ISDN. Jedno z těchto rozhraní slouží k zadávání příkazů pro navázání hovorů a konfiguraci a ostatní jsou skutečně připojeny k síťovým zařízením, které budou využívat datové obvody po jejich vytvoření. Linux bude v této konfiguraci fungovat bez jakýchkoliv úprav. S portem na terminálovém adaptéru se zachází stejným způsobem, jako s kterýmkoliv jiným sériovým zařízením. Jiný způsob, pro který je navržena podpora jádra ISDN, spočívá v nainstalování karty ISDN do linuxového počítače a v nechání softwaru obsluhovat protokoly a provádět vlastní hovory.

Volby kompilace jádra

```
ISDN subsystem --->
<      *> ISDN support
      [ ] Support synchronous PPP
```

```
[ ] Support audio via ISDN
< > ICN 2B and 4B support
< > PCBIT-D support
< > Teles/NICCY1016PC/Creatix support
```

Linuxová implementace ISDN podporuje několik různých typů interních karet ISDN.

- ICN 2B a 4B
- Octal PCBIT-D
- Karty Teles ISDN a kompatibilní

Některé z těchto karet vyžadují pro správnou funkci natažení softwaru. K tomu slouží samostatná utilita.

Všechny podrobnosti o konfiguraci linuxové podpory ISDN najdete v adresáři `/usr/src/linux/Documentation/isdn/` a v dokumentu FAQ věnovanému programu *isdn4linux*, který je dostupný na adrese <http://www.lrz-muenchen.de/~ui161ab/www/isdn/>. (Anglickou verzi tohoto dokumentu získáte klepnutím na anglickou vlajku.)

Poznámka o PPP: Sada protokolů PPP bude fungovat po asynchronních i synchronních sériových linkách. Běžně dodávaný démon protokolu PPP pro Linux s názvem „pppd“ podporuje pouze asynchronní režim. Pokud hodláte provozovat protokoly PPP přes službu ISDN, potřebujete speciálně upravenou verzi. Podrobnosti ohledně jejího získání najdete ve výše uvedené dokumentaci.

6.17 IP-Masquerade (Maškaráda)

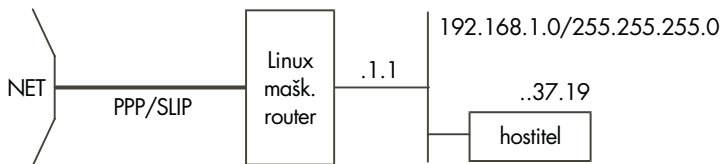
Spousta lidí vlastní jednoduchý vytáčený účet pro připojení k Internetu. Téměř každému, kdo používá tuto konfiguraci, přidělil poskytovatel internetových služeb (Internet Service Provider) jednu IP-adresu. Za normálních okolností to stačí ke zprostředkování plného přístupu k síti pouze jednomu hostiteli. IP-Masquerade je trik, díky němuž může jedinou IP-adresu využívat více počítačů, protože vypadají jako jiní hostitelé, proto se používá termín maškaráda. Nevýhodou je, že tato maškaráda funguje skoro vždy jen jedním směrem, to znamená, že maškarádovaní hostitelé mohou volat ven, ale nemohou přijímat síťová spojení od vzdálených hostitelů. To znamená, že zde nefungují některé síťové služby, například *talk* a další, třeba *ftp*, je třeba nastavit na pasivní režim (PASV). Naštěstí nejpoužívanější síťové služby, jako je *telnet*, World Wide Web a *irc* fungují dobře.

Volby kompilace jádra

```
Code maturity level options  --->
    [*] Prompt for development and/or incomplete code/drivers
Networking options  --->
    [*] Network firewalls
    ....
    [*] TCP/IP networking
    [*] IP: forwarding/gatewaying
    ....
    [*] IP: masquerading (EXPERIMENTAL)
```

Normálně byste svůj linuxový počítač podporující protokol SLIP nebo PPP nechali vytáčet stejně, jako by se jednalo o samostatný počítač. Kromě toho by měl nastaveno další síťové zařízení, možná Ethernet, s jednou z rezervovaných síťových adres. Hostitelé, kteří mají být maskováni, by byli na této druhé síti. Každý z těchto hostitelů by měl nastavenou IP-adresu ethernetového portu linuxového počítače jako implicitní bránu nebo router.

Typická konfigurace by mohla vypadat asi takto:



Nejdůležitější příkazy této konfigurace jsou:

```
# Network route for ethernet
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
#
# Default route to the rest of the internet.
route add default ppp0
#
# Cause all hosts on the 192.168.1/24 network to be masqueraded.
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

Více informací o IP-maškarádě pod Linuxem získáte na stránce *IP-Masquerade Resource Page* <http://www.hw401.com/achau/ipmasq/>.

6.18 IP-transparentní proxy

IP-transparentní proxy je rys, který umožňuje přeměrovat servery nebo služby určené pro jiný počítač na tyto služby na daném počítači. Typicky se používá v případě linuxového počítače pracujícího jako router a zároveň i proxy server. Všechna spojení určená pro tuto službu byste měli vzdáleně přeměrovat na lokální proxy server.

Volby kompilace jádra

```
Code maturity level options  --->
    [*] Prompt for development and/or incomplete code/drivers
Networking options  --->
    [*] Network firewalls
    ....
    [*] TCP/IP networking
    ....
    [*] IP: firewalling
    ....
    [*] IP: transparent proxy support (EXPERIMENTAL)
```

Konfigurace transparentního proxy se provádí pomocí příkazy `ipfwadm`.

Toto je jeden z užitečných příkladů:

```
ipfwadm -I -a accept -D 0/0 telnet -r 2323
```

Tento příklad způsobí přeměrování všech pokusů o spojení libovolného hostitele s portem 23 (telnet) na port 2323 na tomto hostiteli. Takto je možné zajistit automatické přeměrování veškerého telnetového provozu z vaší sítě na lokální port.

6.19 Mobilní IP

Termín „IP-mobilita“ popisuje schopnost hostitele, který umí přesunout své síťové spojení z jednoho místa na Internetu do jiného bez změny své IP-adresy nebo ztráty konektivity. Když IP-hostitel změní místo svého připojení, musí zpravidla změnit také svoji IP-adresu. Mobilita tento problém překonává alokováním pevné IP-adresy mobilnímu hostiteli a používáním IP-zapouzdření (tunelování) s automatickým směrováním, čímž se zajistí, že datagramy určené pro tohoto hostitele jsou směrovány na skutečnou IP-adresu, kterou právě používá.

V současné době se pracuje na projektu, jehož cílem je kompletní sada IP-mobilních nástrojů pro Linux. Stav tohoto projektu a nástrojů zjistíte na stránce *Linux Mobile IP Home Page* <http://anchor.cs.binghamton.edu/~mobileip/>.

6.20 Multicast

IP-Multicast umožňuje simultánní směrování IP-datagramů na libovolný počet IP-hostitelů ve vzájemně neslučitelných IP-sítích. Tento mechanismus je využíván k šíření materiálů „broadcast“ po Internetu, například audio a video přenosů a dalších nových aplikací.

Volby kompilace jádra

```
Networking options  --->
    [*] TCP/IP networking
    ....
    [*] IP: multicasting
```

Je nutná sada nástrojů a některé malé úpravy sítě. Zdroj informací týkajících se způsobu instalace a konfigurace pro Linux najdete na adrese www.teksouth.com <<http://www.teksouth.com/linux/multicast/>>.

6.21 NAT – Network Address Translation

IP Network Address Translation (převod síťových adres) je více než standardizovaným bratrem linuxové IP-maškarády. Podrobně je tato vlastnost popisována v dokumentu RFC-1631. NAT disponuje vlastnostmi, které IP-maškarádě chybí, takže je neobyčejně vhodná pro použití ve firewallových routerech a velkých instalacích.

Jádro alfa-verze NAT pro Linux 2.0.29 vytvořil Michael Hasenstein, **Michael.Hasenstein@informatik.tu-chemnitz.de**. Jeho dokumentaci a implementaci lze získat na stránce:

Linux IP Network Address Web Page <http://www.csn.tu-chemnitz.de/HyperNews/get/linux-ip-nat.html>.

Novější jádra Linuxu 2.1.* obsahují také některé funkce NAT ve směrovacím algoritmu.

6.22 NetRom (AF_NETROM)

Názvy zařízení NetRom jsou „nr0“, „nr1“ atd.

Volby kompilace jádra

```
Networking options  --->
  [*] Amateur Radio AX.25 Level 2
  [*] Amateur Radio NET/ROM
```

Protokoly AX25, NetRom a Rose jsou popsány v dokumentu *AX25-HOWTO*. Tyto protokoly používají amatérští rádiooperátoři při experimentech s paketovým rádiem.

Většinu práce na implementaci těchto protokolů má na svědomí Jonathon Naylor, jsn@cs.nott.ac.uk.

6.23 PLIP

Názvy zařízení PLIP jsou „plip0“, „plip1“ atd.

Volby kompilace jádra

```
Networking options  --->
  <*> PLIP (parallel port) support
```

Protokol *PLIP* (Paralel Line IP) je podobný protokolu *SLIP* v tom, že slouží k poskytování síťových spojení typu *point to point* mezi dvěma počítači. ale je navržen pro paralelní tiskový port místo portu sériového (schéma zapojení najdete v příslušné stati na konci dokumentu). Protože pomocí paralelního portu je možné přenášet více než jeden bit současně, dosahuje se u rozhraní *PLIP* vyšších rychlostí než u standardního sériového zařízení. Kromě toho i ten nejjednodušší paralelní port lze použít místo poměrně drahého rozhraní 16550AFN UART pro sériové porty. Protokol *PLIP* využívá v porovnání se spojením přes sériový port velkou část CPU a pokud můžete získat nějaké levné ethernetové karty, není dobrou volbou. Pokud však nemáte k dispozici nic jiného, můžete se na něj spolehnout. Při dobrém spojení můžete očekávat přenosové rychlosti kolem 20 kilobajtů za sekundu.

Ovladače zařízení protokolu *PLIP* soutěží s ovladačem paralelních zařízení o paralelní port. Pokud chcete používat oba tyto ovladače, pak byste je měli přeložit jako moduly, abyste mohli volit, který port chcete pro *PLIP* a které porty pro ovladač tiskárny. Podrobnější informace najdete v dokumentu *Modules-HOWTO* <Modules-HOWTO.html>.

Některé přenosné počítače používají chipsety, které nebudou s protokolem *PLIP* fungovat, protože neumožňují některé kombinace signálů, na které tento protokol spoléhá a tiskárny je nepoužívají.

Linuxové rozhraní *PLIP* je kompatibilní s *Crynwyr Packet Driver PLIP*, což znamená, že prostřednictvím *PLIP* můžete spojit váš linuxový počítač s dosovým počítačem, na kterém běží kterýkoliv jiný druh softwaru TCP/IP.

V jádru verzí 2.0.* jsou zařízení PLIP přiděleny následující V/V porty a IRQ:

Zařízení	V/V	IRQ
plip0	0x3bc	5
plip1	0x378	7
plip2	0x278	2

Pokud vašemu paralelnímu portu žádná z těchto kombinací nevyhovuje, můžete změnit IRQ portu pomocí příkazu `ifconfig`, jemuž předáte parametr „irq“. Pokud to váš BIOS umožňuje, nezapomeňte na tiskových portech povolit příslušná IRQ.

V pozdějších verzích jádra 2.1.* s podporou Plug'n'Play jsou zařízení PLIP alokována postupně v tom pořadí, v jakém jsou detekována, podobně jako ethernetová zařízení. První alokované zařízení je označeno `plip0`.

Při kompilaci jádra je třeba upravit pouze jediný soubor. Jmenuje se `/usr/src/linux/driver/net/CONFIG` a obsahuje časovače *PLIP* v milisekundách. Implicitní hodnoty jsou ve většině případů vyhovující. Zvýšit je bude nutné jen pokud máte zvláště pomalý počítač, a zvyšovat budete časovače na **druhém** počítači. Program nazvaný `plipconfig` umožňuje změnit tato nastavení bez nutnosti rekompilace jádra. Je dodáván jako součást většiny distribucí Linuxu.

Konfigurace rozhraní *PLIP* znamená **doplnění** následujících řádků do sítového souboru `rc`:

```
#
# Attach a PLIP interface
#
# configure first parallel port as a plip device
/sbin/ifconfig plip0 IPA.IPA.IPA.IPA pointopoint IPR.IPR.IPR.IPR up
#
# End plip
```

Kde:

IPA.IPA.IPA.IPA zastupuje vaši IP-adresu

IPR.IPR.IPR.IPR zastupuje IP-adresu vzdáleného počítače

Parametr *pointpoint* má stejný význam jako u protokolu SLIP v tom, že specifikuje adresu počítače na druhém konci spojení.

Téměř ve všech ohledech můžete zacházet s rozhraním *PLIP* jako by se jednalo o rozhraní *SLIP*, jen s tou výjimkou, že není zapotřebí ani možné používat příkazy *dip* a *slattach*.

Další informace najdete v dokumentu *PLIP-mini-HOWTO*.

6.24 PPP

Názvy zařízení PPP jsou „ppp0“, „ppp1“ atd. Zařízení jsou číslována sekvenčně, přičemž první nalezené zařízení bude mít označení „0“.

Volby kompilace jádra

```
Networking options  --->
    <*> PPP (point-to-point) support
```

Konfigurace protokolu PPP je podrobněji probírána v dokumentu *PPP-HOWTO*.

6.24.1 Zajištění permanentního připojení k síti pomocí *pppd*

Máte-li to štěstí, že vlastníte „polopermanentní“ připojení k síti a byli byste rádi, kdyby váš počítač v případě ztráty spojení PPP toto automaticky znovu navázal, mám pro vás jednoduché řešení.

Pomocí následujícího příkazu nastavte PPP tak, jako by ho spustil uživatel *root*.

```
# pppd
```

Nezapomeňte v souboru */etc/ppp/options* uvést volbu „-detach“. Potom vložte do sekce *getty* souboru */etc/inittab* následující řádek:

```
pd:23:respawn:/usr/sbin/pppd
```

Takto bude program *init* monitorovat běh programu *pppd* a pokud skončí, automaticky ho znovu spustí.

6.25 Protokol Rose (AF_ROSE)

Názvy zařízení Rose jsou ve verzi jádra 2.1.* „rs0“, „rs1“ atd. Protokol Rose je dostupný od verzí 2.1.*.

Volby kompilace jádra

```
Networking options  --->
  [*] Amateur Radio AX.25 Level 2
  <*> Amateur Radio X.25 PLP (Rose)
```

Protokoly AX25, NetRom a Rose jsou popsány v dokumentu *AX25-HOWTO*. Tyto protokoly používají amatérští radiooperátoři při experimentech s paketovým rádiem.

Většinu práce na implementaci těchto protokolů má na svědomí Jonathon Naylor, jsn@cs.nott.ac.uk.

6.26 SAMBA – podpora 'NetBEUI', 'NetBios'

SAMBA je implementací protokolu Session Management Block. Umožňuje systémům firmy Microsoft a jiným připojovat a používat disky a tiskárny.

Protokol SAMBA a jeho konfigurace jsou rozebírány v dokumentu *SMB-HOWTO*.

6.27 SLIP-klient

Názvy zařízení SLIP jsou „sl0“, „sl“ atd. První nalezené zařízení bude mít označení „0“ a zbytek pak rostoucí čísla v pořadí, v jakém byly konfigurovány.

Volby kompilace jádra

```
Network device support  --->
  [*] Network device support
  <*> SLIP (serial line) support
  [ ] CSLIP compressed headers
  [ ] Keepalive and linefill
  [ ] Six bit SLIP encapsulation
```

Protokol SLIP (Serial Line Internet Protocol) dovoluje provozovat protokol TCP/IP po sériové lince, ať už se jedná o telefonní linku s vytáčeným modemem nebo nějaký druh pronajaté linky. Samozřejmě, že abyste mohli používat protokol SLIP, potřebujete mít ve vaší oblasti přístup k *SLIP-serveru*. Spousta univerzit a společností po celém světě takovýto přístup poskytuje.

Zařízení SLIP používá sériové porty vašeho počítače k přenosu IP-datagramů. K tomu potřebuje převzít kontrolu nad sériovým zařízením. Zařízení SLIP jsou pojmenována „sl0“,

„s/l“ atd. Jak to koresponduje s vašimi sériovými zařízeními? Síťový kód pomocí tzv. volání `ioctl` (řízení V/V operací) změní sériová zařízení na SLIP zařízení. Tuto proceduru zvládají dva programy, nazývají se `dip` a `slattach`.

6.27.1 *dip*

`dip` (Dialup IP) je chytrý program, který umí nastavit rychlost sériového zařízení, přikázat vašemu modemu, aby vytočil vzdálený konec linky, automaticky vás přihlásí ke vzdálenému serveru, hledá zprávy, které vám server poslal, a vytáhne z nich informace, jako je IP-adresa, a spustí volání `ioctl`, které je nutné pro přepnutí sériového portu do režimu SLIP. Program `dip` má mocnou podporu skriptů, což můžete využít k automatizaci přihlašovací procedury.

Tento program najdete na adrese <ftp://sunsite.unc.edu/pub/Linux/system/Network/serial/dip/dip3370-uri.tgz>.

Při instalaci proveďte následující posloupnost příkazů:

```
#
# cd /usr/src
# gzip -dc dip3370-uri.tgz | tar xvf -
# cd dip-3.3.70

<editace souboru Makefile>

# make install
#
```

Soubor `makefile` předpokládá existenci skupiny jménem `uucp`, ale tento název můžete v závislosti na vaší konfiguraci změnit buď na `dip` nebo `SLIP`.

6.27.2 *slattach*

Program `slattach` je ve srovnání s programem `dip` velice jednoduchý, snadno se používá, ale není tak důmyslný, jako `dip`. Nedisponuje podporou skriptů a veškerá jeho činnost spočívá v konfiguraci vašeho sériového zařízení jako zařízení SLIP. Předpokládá, že máte všechny potřebné informace a před jeho spuštěním je navázáno spojení po sériové lince. Program `slattach` je ideální tam, kde máte trvalé spojení se serverem, například fyzickým kabelem nebo vyhrazenou linkou.

6.27.3 Kdy který program použít?

Program `dip` byste měli použít, pokud se ke SLIP-serveru připojujete prostřednictvím modemu nebo nějakého jiného dočasného spojení. Program `slattach` byste měli použít, je-li váš počítač spojen se serverem vyhrazenou linkou, třeba kabelem, a pro správnou funkci tohoto spojení není třeba provádět nějaké speciální činnosti. Více informací získáte ve stati „Trvalé spojení SLIP“.

Konfigurace protokolu SLIP je podobná konfiguraci ethernetového rozhraní (nahlédněte do stati „Konfigurace ethernetového rozhraní výše“). Jsou zde však některé klíčové rozdíly.

Předně spojení SLIP se liší od ethernetových sítí tím, že zde jsou v síti vždy jen dva hostitelé, na každém konci jeden. Na rozdíl od Ethernetu, který lze používat ihned po připojení kabelů, u protokolu SLIP je třeba, v závislosti na typu spojení, určitým speciálním způsobem síťové spojení inicializovat.

Pokud používáte program `dip`, nedejde k tomu normálně při startu systému, ale o něco později, až budete připraveni navázat spojení. Tuto proceduru je možné automatizovat. Používáte-li program `slattach`, bude nutné přidat do souboru `rc.inet1` novou sekci, viz dále.

Existují dva základní typy SLIP-serverů: Servery s dynamickou IP-adresou a servery se statickou IP-adresou. Skoro každý SLIP-server vás po vytočení vyzve k přihlášení za pomoci vašeho uživatelského jména a hesla. Program `dip` vás může přihlásit automaticky.

6.27.4 Statický SLIP-server s vytáčenou linkou a programem DIP

Statickému SLIP-serveru poskytnete IP-adresu, která je výlučně vaše. Pokaždé, když se připojíte k tomuto serveru, nastavíte SLIP-port na tuto adresu. Statický SLIP-server odpoví vašemu modemu, pravděpodobně vás vyzve k zadání vašeho uživatelského jména a hesla a potom vám přes toto spojení pošle veškeré datagramy určené pro vaši adresu. Máte-li server se statickou IP-adresou, můžete název vašeho hostitele a IP-adresu (pokud je dopředu znáte) umístit do souboru `/etc/hosts`. Také bude potřeba překonfigurovat některé další soubory: `rc.inet2`, `host.conf`, `resolv.conf`, `/etc/HOSTNAME` a `rc.local`. Nezapomeňte, že při úpravě souboru `rc.inet1` do něj nemusíte přidávat žádné speciální příkazy týkající se SLIP-spojení, protože program `dip` provede veškerou práci související s konfigurací rozhraní za vás. Stačí, když programu `dip` poskytnete příslušné informace, a on nejprve přikáže modemu, aby navázal spojení a přihlásil se k vašemu SLIP-serveru a potom sám rozhraní nakonfiguruje.

Pokud váš SLIP-server funguje tímto způsobem, pak přeskočte na další stať – „Používání programu DIP“, kde se dozvíte informace o jeho konfiguraci.

6.27.5 Dynamický SLIP-server s vytáčenou linkou a programem DIP

Dynamický SLIP-server vám při každém přihlášení přidělí IP-adresu náhodně ze skupiny adres. To znamená, že nelze zaručit, že budete mít pokaždé jednu konkrétní adresu a že tuto adresu může po vašem odhlášení využívat někdo jiný. Správce sítě, který prováděl konfiguraci SLIP-serveru, mu přidělil skupinu adres, které může používat. Když server obdrží nový hovor, vyhledá první nepoužívanou adresu, provede volajícího přihlašovací procesem a potom zobrazí uvítací zprávu, která obsahuje přidělenou IP-adresu a tuto adresu pak bude používat po celou dobu trvání hovoru.

Konfigurace tohoto typu serveru je podobná konfiguraci statického serveru. Pouze obsahuje jeden krok navíc, ve kterém je třeba získat IP-adresu, kterou vám server vyhradil, a předat ji zařízení SLIP.

I v tomto případě za vás provede veškerou práci program `dip`, jehož novější verze vás nejenom přihlásí, ale umí také automaticky vytáhnout a uložit z uvítací zprávy IP-adresu, kterou pak můžete použít při konfiguraci zařízení SLIP.

6.27.6 Používání programu DIP

Již jsme si řekli, že program `dip` je mocný nástroj, který vám může za pomoci příkazů `ifconfig` a `route` ulehčit a zcela automatizovat proces vytáčení SLIP-serveru, přihlášení, navázání spojení a konfigurace SLIP-zařízení.

V podstatě napíšete skript tvořený příkazy, kterým program `dip` rozumí a na jejichž základě provede příslušné akce. Představu o fungování programu `dip` si můžete utvořit podle následujícího vzorového skriptu `sample.dip`, který je dodáván společně s tímto programem. Program `dip` je poměrně výkonný a obsahuje množství voleb.

Nebudeme se jimi zde zabývat, ale v případě zájmu vás odkazují na manuálové stránky, soubor `README` a vzorové soubory, které získáte současně s vaší verzí programu `dip`.

Možná jste si všimli, že skript `sample.dip` předpokládá, že používáte statický SLIP-server, takže budete dopředu vědět svou IP-adresu. Pro dynamické SLIP-servery obsahují novější verze programu `dip` příkaz, který slouží k automatickému přečtení IP-adresy, kterou vám dynamický server přidělil, a následnému nakonfigurování zařízení SLIP. Následující ukázka je upravenou verzí souboru `sample.dip`, který je součástí balíku `dip337j-uri.tgz` a má vám posloužit jako jakýsi odrazový můstek. Soubor uložte pod názvem `/etc/dipscript` a upravte podle vaší konfigurace.

```
#
# sample.dip      Dialup IP connection support program.
#
#               This file (should show) shows how to use the DIP
#               This file should work for Annex type dynamic servers, if you
#               use a static address server then use the sample.dip file that
#               comes as part of the dip337-uri.tgz package.
#
#
# Version:       @(#)sample.dip  1.40      07/20/93
#
# Author:       Fred N. van Kempen, <waltje@uWalt.NL.Mugnet.ORG>
#
main:
# Next, set up the other side's name and address.
# My dialin machine is called 'xs4all.hacktic.nl' (== 193.78.33.42)
get $remote xs4all.hacktic.nl
# Set netmask on sl0 to 255.255.255.0
netmask 255.255.255.0
# Set the desired serial port and speed.
port cua02
speed 38400

# Reset the modem and terminal line.
# This seems to cause trouble for some people!
reset

# Note! "Standard" pre-defined "errlevel" values:
# 0 - OK
# 1 - CONNECT
# 2 - ERROR
#
# You can change those grep'ping for "addchat()" in *.c...

# Prepare for dialing.
```

```
send ATQ0V1E1X4\r
wait OK 2
if $errlvl != 0 goto modem_trouble
dial 555-1234567
if $errlvl != 1 goto modem_trouble

# We are connected.  Login to the system.
login:
sleep 2
wait ogin: 20
if $errlvl != 0 goto login_trouble
send MYLOGIN\n
wait ord: 20
if $errlvl != 0 goto password_error
send MYPASSWD\n
loggedin:
# We are now logged in.
wait SOMEPROMPT 30
if $errlvl != 0 goto prompt_error

# Command the server into SLIP mode
send SLIP\n
wait SLIP 30
if $errlvl != 0 goto prompt_error

# Get and Set your IP address from the server.
#   Here we assume that after commanding the SLIP server into SLIP
#   mode that it prints your IP address
get $locip remote 30
if $errlvl != 0 goto prompt_error

# Set up the SLIP operating parameters.
get $mtu 296
# Ensure "route add -net default xs4all.hacktic.nl" will be done
default

# Say hello and fire up!
done:
```

```

print CONNECTED $locip ---> $rmtip
mode CSLIP
goto exit

prompt_error:
print TIME-OUT waiting for sliplogin to fire up...
goto error

login_trouble:
print Trouble waiting for the Login: prompt...
goto error

password:error:
print Trouble waiting for the Password: prompt...
goto error

modem_trouble:
print Trouble occurred with the modem...
error:
print CONNECT FAILED to $remote
quit

exit:
exit

```

Výše uvedený příklad předpokládá volání *dynamického* SLIP-serveru. Voláte-li *statický* SLIP-server, použijte soubor `sample.dip`, který je součástí balíku `dip337j-uri.tgz`.

Když předáte programu `dip` příkaz `get $local`, prohledá příchozí text, zda neobsahuje řetězec, který vypadá jako IP-adresa, to znamená řetězec čísel oddělených tečkami. Tato úprava byla do programu přidána kvůli *dynamickým* SLIP-serverům, aby mohl být proces čtení IP-adresy poskytnuté serverem automatizován.

Výše uvedený příklad automaticky vytvoří implicitní trasu přes vaše SLIP-spojení. Pokud si to nepřejete, může ponechat ethernetové spojení jako implicitní trasu, potom odstranit příkaz `default` ze skriptu. Po skončení tohoto skriptu provedete příkaz `ifconfig`, zjistíte, že máte nové zařízení jménem `sl0`. To je vaše zařízení SLIP. Dle potřeby můžete manuálně upravit jeho konfiguraci po skončení příkazu `dip` pomocí příkazů `ifconfig` a `route`.

Všimněte si, že příkaz `mode` programu `dip` umožňuje výběr několika různých protokolů, nejpopulárnějším příkladem je CSLIP, což znamená SLIP s kompresí. Všimněte si také, že musí souhlasit oba konce spojení, takže cokoliv zvolíte by mělo souhlasit s nastavením vašeho serveru.

Výše uvedený příklad je poměrně robustní a měl by odolat většině chyb. Podrobnější informace najdete v manuálových stránkách programu `dip`. Přirozeně byste mohli třeba vytvořit skript, který by prováděl opakované vytáčení serveru, pokud by nebylo navázáno spojení po uplynutí stanovené doby nebo vyzkoušet skupinu serverů, pokud přistupujete k více než jednomu serveru.

6.27.7 Trvalé spojení SLIP s vyhrazenou linkou a programem `slattach`

Máte-li dva počítače propojeny kabelem, případně vyhrazenou linkou nebo nějakým jiným typem trvalého sériového spojení, nepotřebujete se zatěžovat všemi problémy kolem používání programu `dip` na sériové linky. Při konfiguraci spojení úplně vystačíte s jednoduchou utilitou jménem `slattach`.

Protože bude vaše spojení trvalé, bude nutné přidat určité příkazy do souboru `rc.inet1`. V podstatě je potřeba nastavit správnou rychlost sériového zařízení a přepnout ho do režimu SLIP. Díky programu `slattach` vám k tomu stačí jeden příkaz. Následující řádky **přidejte** do souboru `rc.inet1`.

```
#
# Attach a leased line static SLIP connection
#
# configure /dev/cua0 for 19.2kbps and cslip
/sbin/slattach -p cslip -s 19200 /dev/cua0 &
/sbin/ifconfig sl0 IPA.IPA.IPA.IPA pointopoint IPR.IPR.IPR.IPR up
#
# End static SLIP.
```

Kde:

IPA.IPA.IPA.IPA zastupuje vaši IP-adresu

IPR.IPR.IPR.IPR zastupuje IP-adresu vzdáleného počítače

Program `slattach` přidělí první volné zařízení SLIP specifikovanému sériovému zařízení. Začne označením `sl0`. Takto první příkaz `slattach` přiřadí specifikovanému zařízení SLIP zařízení `sl0`, další pak `sl1` atd.

Program `slattach` umožňuje pomocí argumentu `-p` konfigurovat různé protokoly. Ve vašem případě použijete buď *SLIP*, nebo *CSLIP*, podle toho, zda chcete nebo nechcete používat kompresi.

6.28 SLIP-server

Pokud máte počítač, který je připojen k síti, a chcete, aby se k němu mohli připojovat i jiní uživatelé, budete muset tento počítač nakonfigurovat jako server. Chcete-li jako protokol sériové linky používat SLIP, máte, co se týče způsobu konfigurace vašeho linuxového počítače jako SLIP-server, tři možnosti. Já osobně bych dal přednost první z nich, která využívá program `sliplogin` a připadá mi nejjednodušší co se konfigurace a pochopení týče. Nicméně představím vám všechny tři, abyste si mohli udělat vlastní úsudek.

6.28.1 SLIP-server používající program `sliplogin`

Program `sliplogin` lze použít místo klasického přihlašovacího programu pro uživatele, kteří převedou terminálovou linku na SLIP-linku. Umožňuje nastavit váš linuxový počítač jako *statický server* (uživatelé získají při každém zavolání vždy stejnou adresu) nebo jako *dynamický server* (uživatelům je přidělena adresa, která nutně nemusí být stejná, jako při předchozím spojení).

Volající se přihlásí jako při normálním přihlašování, zadá své uživatelské jméno a heslo. Ovšem místo příkazového interpretu se po přihlášení spustí program `sliplogin`, který vyhledá v konfiguračním souboru (`/etc/slip.hosts`) záznam s uživatelským jménem, které odpovídá jménu volajícího. Pokud takový záznam najde, nastaví linku jako 8bitovou a pomocí udání `ioctl` ji převede na SLIP-linku. Po skončení tohoto procesu začne poslední fáze konfigurace, kdy program `sliplogin` spustí skript příkazového interpretu, který přidělí rozhraní SLIP IP-adresu, síťovou masku a nastaví příslušné směrování. Tento skript se většinou nazývá `/etc/slip.login`, ale vyžadují-li někteří volající speciální inicializaci, můžete vytvořit konfigurační skripty nazvané `/etc/slip.login.přihlašovacíjméno`, které budou spouštěny místo tohoto implicitního.

Pro správnou funkci programu `sliplogin` potřebujete tři nebo čtyři soubory. Podrobně vám popíši, jak a kde získáte příslušný software a jak ho nakonfigurujete. Jedná se o tyto soubory:

- `/etc/passwd` pro vytáčené uživatelské účty
- `/etc/slip.hosts` pro ukládání informací, které jsou pro každého uživatele jedinečné
- `/etc/slip.login` řídí konfiguraci směrování, kterou je třeba provádět pro každého uživatele
- `/etc/slip.tty` je nutný pouze v případě, že má server nastavenou *dynamickou alokaci adres* a obsahuje tabulku přidělovaných adres

- `/etc/slip.logout` obsahuje příkazy, které se provedou po zavěšení nebo odhlášení uživatele

Kde získáte program `sliplogin`? Je možné, že jste balík `sliplogin` získali jako součást vaší distribuce Linuxu. Pokud nikoliv, získáte program `sliplogin` na adrese `ftp://sunsite.unc.edu/pub/linux/system/Network/serial/sliplogin-2.1.1.tar.gz`. Archiv ve formátu `tar` obsahuje zdrojový kód, předkompilované binární soubory a manuálovou stránku.

Aby mohli program `sliplogin` používat pouze autorizovaní uživatelé, měli byste do souboru `/etc/group` přidat záznam podobný tomu následujícímu:

```
..
slip::13:radio,fred
..
```

Po nainstalování balíku `sliplogin` změní soubor `Makefile` vlastnictví skupiny programu `sliplogin` na `slip`, což znamená, že ho mohou spouštět pouze uživatelé patřící do této skupiny. Výše uvedený zápis povolí spouštění tohoto programu pouze uživatelům `radio` a `fred`.

Následující posloupnost příkazů nainstaluje binární soubory do adresáře `/sbin` a manuálové stránky do sekce 8.

```
# cd /usr/src
# gzip -dc ../sliplogin-2.1.1.tar.gz | tar xvf -
# cd sliplogin-2.1.1
# <..editujte Makefile pokud nepoužíváte stínová hesla..>
# make install
```

Pokud budete chtít binární soubory ještě před instalací překompilovat, spusťte před příkazem `make install` ještě příkaz `make clean`. Budete-li chtít nainstalovat binární soubory někam jinam, musíte upravit pravidlo `install` v souboru `Makefile`.

Další informace získáte v souboru `README`, který je součástí balíku.

Konfigurace souboru `/etc/passwd` pro SLIP hostitele. Normálně byste pro tyto uživatele vytvořili v souboru `/etc/passwd` speciální účty. Běžně se ovšem před jméno volajícího hostitele přidává písmeno „S“. Takže když se například volající hostitel jmenuje `radio`, vytvoříte v souboru `/etc/passwd` následující záznam:

```
Sradio:FvKurok73:1427:1:radio SLIP login:/tmp:/sbin/sliplogin
```

Pokud je pro vás účet smysluplný, nezáleží na tom, jak se nazývá.

Poznámka: Volající nepotřebuje mít žádný speciální adresář, protože nebude mít přístup k příkazovému interpretu, takže postačí adresář `/tmp`. Nezapomeňte také, že místo normálního přihlašovacího příkazového interpretu se používá program `sliplogin`.

Konfigurace souboru `/etc/slip.hosts`. V souboru `/etc/slip.hosts` hledá program `sliplogin` záznamy vyhovující uživatelskému jménu, aby získal konfigurační informace pro příslušného volajícího. V tomto souboru specifikujete IP-adresu a síťovou masku, která bude přidělena volajícímu. Následují vzorové záznamy pro dva hostitele, první je statická konfigurace pro hostitele `radio` a druhá dynamická pro uživatele `albert`:

```
#
Sradio  44.136.8.99    44.136.8.100  255.255.255.0  normal      -1
Salbert 44.136.8.99    DYNAMIC      255.255.255.0  compressed  60
#
```

Záznamy v souboru `/etc/slip.hosts` jsou složeny z následujících položek:

1. Přihlašovací jméno volajícího uživatele.
2. IP-adresa serveru, to znamená tohoto počítače.
3. IP-adresa, která bude přidělena volajícímu. Je-li v tomto poli uveden řetězec „DYNAMIC“, bude IP-adresa přidělena v závislosti na informaci obsažené v souboru `/etc/slip.tty`, který jsme probírali před chvílí.
Poznámka: Aby vše fungovalo tak, jak má, musíte používat program `sliplogin` alespoň verze 1.3.
4. Síťová maska přidělená volajícímu počítači v tečkové notaci, tj. 255.255.255.0 pro síťovou masku třídy C.
5. Nastavení režimu SLIP, kterým můžete povolit nebo zakázat kompresi. Povolené jsou hodnoty „normal“ nebo „compressed“.
6. Časový údaj, který specifikuje dobu, po kterou může linka zůstat v nečinnosti (nepřijímá žádné datagramy), než dojde k automatickému zrušení spojení. Záporná hodnota tuto vlastnost ruší.
7. Volitelné argumenty.

Poznámka: Pole 2 a 3 mohou obsahovat buď názvy hostitelů, nebo IP-adresy. Použijete-li názvy hostitelů, musí být váš počítač schopen zjistit příslušnou IP-adresu, jinak se skript po spuštění zhroutí. Můžete si to ověřit příkazem `telnet`, kterému předáte jako parametr název hostitele. Pokud se objeví zpráva „Trying nnn.nnn.nnn...“, podařilo se vašemu počítači nalézt IP-

adresu odpovídající tomuto názvu hostitele. Objeví-li se zpráva „*Unknown host*“, potom se to nepodařilo. V takovém případě buď použijte IP-adresu převedenou na tečkovou desítkovou notaci, nebo upravte konfiguraci resolveru (viz. stať Konfigurace resolveru).

Nejpoužívanější režimy SLIP jsou:

normal - zapíná normální nekomprimovaný SLIP

compressed – zapíná van Jacobsonovu kompresi hlaviček (CSLIP)

Samozřejmě, že se tyto režimy navzájem vylučují, takže musíte použít buď jeden, nebo druhý. Další informace o ostatních volbách najdete v manuálových stránkách.

Konfigurace souboru `/etc/slip.login`. Jakmile program `sliplogin` nalezl vyhovující záznam v souboru `/etc/slip.hosts`, pokusí se zpracovat soubor `/etc/slip.login` a dojde na vlastní konfiguraci rozhraní SLIP, kterému je předána IP-adresa a síťová maska.

Vzorový soubor `/etc/slip.login`, který je dodáván s balíkem `sliplogin`, vypadá takto:

```
#!/bin/sh -
#
#      @(#)slip.login  5.1  (Berkeley)  7/1/90
#
# generic login file for a SLIP line.  sliplogin invokes this with
# the parameters:
#      $1      $2      $3      $4, $5, $6 ...
#      SLIPunit ttyspeed  pid  the arguments from the slip.host entry
#
/sbin/ifconfig $1 $5 pointopoint $6 mtu 1500 -trailers up
/sbin/route add $6
arp -s $6 <hw_addr> pub
exit 0
#
```

Jistě si všimnete, že tento skript používá ke konfiguraci SLIP-zařízení příkazy `ifconfig` a `route`, kterým předá IP-adresu, vzdálenou IP-adresu a síťovou masku, a vytvoří směrování pro vzdálenou adresu prostřednictvím rozhraní SLIP. Stejného výsledku bychom dosáhli i pomocí programu `slattach`.

Nezapomeňte také použít *Proxy ARP*, aby ostatní hostitelé, kteří jsou připojeni ke stejnému Ethernetu jako server, věděli, jak se dostanou k volajícímu hostiteli. Pole `<hw_addr>` by mělo obsahovat hardwarovou adresu ethernetové karty v tomto počítači. Pokud váš server není součástí ethernetové sítě, můžete tento řádek vynechat.

Konfigurace souboru `/etc/slip.logout`. Jakmile volající zavěsí, budete chtít obnovit normální stav sériového zařízení, aby se mohli dovolat další volající. Tuto funkci plní soubor `/etc/slip.logout`. Jeho formát je poměrně jednoduchý a je spouštěn se stejným argumentem, jako soubor `/etc/slip.login`.

```
#!/bin/sh -
#
#           slip.logout
#
/sbin/ifconfig $1 down
arp -d $6
exit 0
#
```

Jeho cílem je „shodit“ rozhraní, které odstraní dříve vytvořené manuální směrování. Pomocí příkazu `arp` také smaže všechny záznamy v tabulce ARP. I zde platí, že pokud server neobsahuje ethernetový port, nemusí být ve skriptu příkaz `arp`.

Konfigurace souboru `/etc/slip.tty`. Používáte-li dynamické přidělování adres (máte u některých hostitelů nastaveno v souboru `/etc/slip.hosts` klíčové slovo `DYNAMIC`) musíte v souboru `/etc/slip.tty` uvést seznam adres přidělených jednotlivým portům. K dynamickému přidělování adres potřebujete pouze tento soubor.

Soubor `/etc/slip.tty` je vlastně tabulka obsahující zařízení `tty`, která budou podporovat vytáčené SLIP-spojení a IP-adresy, které budou přiděleny uživatelům, kteří na příslušný port zavolají.

Formát souboru vypadá takto:

```
# slip.tty      tty -> IP address mappings for dynamic SLIP
# format: /dev/tty?? xxx.xxx.xxx.xxx
#
/dev/ttyS0      192.168.0.100
/dev/ttyS1      192.168.0.101
#
```

Tato tabulka říká, že volajícimu, který vytočí port `/dev/ttyS0` a má v souboru `/etc/slip.hosts` nastavené pole pro vzdálenou adresu na `DYNAMIC`, přiřadí adresu `192.168.0.100`.

Takto vám pro všechny uživatele, kteří nevyžadují vyhrazené adresy, stačí alokovat jednu adresu pro každý port. Vyhnete se tak plýtvání adresovým prostorem.

6.28.2 SLIP-server používající program `dip8`

Dovolte mi začít konstatováním, že některé níže uvedené informace pocházejí z manuálových stránek programu `dip`, kde je stručně popisována konfigurace SLIP-serveru pod Linuxem. Předesílám též, že následující fakta vycházejí z balíku `dip3370-uri.tgz` a pravděpodobně se nebudou úplně shodovat s jinými verzemi tohoto programu.

Program `dip` může pracovat v tzv. vstupním režimu, kdy automaticky vyhledá záznam uživatele, který ho spustil, a nakonfiguruje sériovou linku jako SLIP linku podle informací uložených v souboru `/etc/diphosts`. Tento vstupní režim aktivujete, když program `dip` spustíte jako `diplogin`. Tímto způsobem používáte program `dip` jako SLIP-server, vytvoříte speciální účty, kde program `diplogin` funguje jako přihlašovací příkazový interpret.

Nejprve je třeba vytvořit následující symbolický odkaz:

```
# ln -sf /usr/sbin/dip /usr/sbin/diplogin
```

Potom doplníte určité záznamy do souborů `/etc/passwd` a `etc/diphosts`. Tyto záznamy mají níže popsany formát:

Při konfiguraci Linuxu coby SLIP-serveru potřebujete vytvořit pro uživatele určité speciální SLIP-účty, přičemž program `dip` (ve vstupním režimu) slouží jako přihlašovací příkazový interpret. Doporučuje se označovat všechny účty SLIP s počátečním písmenem „S“, např. „Sfredm“.

Takto vypadá vzorový záznam SLIP-uživatele v souboru `/etc/passwd`:

```
Sfredm:ij/SMxiTlGVC0:1004:10:Fred:/tmp:/usr/sbin/diplogin
```

↑ ↑ ↑ ↑ ↑ ↑

Přihlašovací jméno Heslo UID GID Domovský adresář Plné jméno uživatele diplogin jako přihlašovací příkazový interpret

Jakmile se uživatel přihlásí, program `login`, pokud nalezne a ověří uživatele, spustí příkaz `diplogin`. Program `dip`, je-li spuštěn jako `diplogin`, ví, že má automaticky předpokládat, že je používán jako přihlašovací příkazový interpret. Pokud je spuštěn jako `diplogin`, zjistí nejprve pomocí funkce `getuid()` id uživatele, který ho spustil. Potom vyhledá v souboru `/etc/diphosts` první záznam, který odpovídá buď id uživatele, nebo názvu zařízení `tty`, ze kterého hovor pochází a příslušným způsobem se nakonfiguruje. Moudrým rozhodnutím, zda vyhradit uživateli záznam v souboru `diphosts` nebo mu přidělit implicitní konfiguraci, získáte směs uživatelů s dynamicky a staticky přidělovanými adresami.

Program `dip`, je-li spuštěn ve vstupním režimu, automaticky přidá záznam „Proxy-ARP“, takže se nemusíte zabývat manuálním přidáváním těchto záznamů.

Konfigurace souboru `/etc/diphosts`.

Soubor `/etc/diphosts` slouží programu `dip` k vyhledávání přednastavených konfigurací vzdálených hostitelů. Tito vzdálení hostitelé mohou být uživatelé, kteří se přihlašují k vašemu linuxovému počítači, nebo to mohou být stroje, ke kterým se přihlašujete z vašeho počítače.

Následuje obecný formát souboru `/etc/diphosts`:

```
..
Suwalt::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:CSLIP,1006
ttyS1::145.71.34.3:145.71.34.2:255.255.255.0:Dynamic ttyS1:CSLIP,296
..
```

Toto jsou jednotlivá pole každého záznamu:

1. přihlašovací jméno: vrácené funkcí `getpwuid(getuid())` nebo název `tty`
2. nevyužito: slučitelné s heslem
3. Vzdálená adresa: IP-adresa volajícího hostitele, buď číselná, nebo názvová
4. Lokální adresa: IP-adresa tohoto počítače, opět buď číselná, nebo názvová
5. Síťová maska: v tečkové notaci
6. Pole komentáře: sem můžete napsat cokoliv
7. Protokol: SLIP, CSLPI atd.
8. MTU: desítkové číslo

Příklad záznamu v souboru `/etc/diphosts` pro vzdáleného SLIP-uživatele může vypadat třeba takto:

```
Sfredm::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:SLIP,296
```

což udává SLIP-spojení se vzdálenou adresou 145.71.34.1 a MTU 296 nebo jiný příklad:

```
Sfredm::145.71.34.1:145.71.34.2:255.255.255.0:SLIP uwalt:CSLIP,1006
```

kteřý specifikuje spojení CSLIP se vzdálenou adresou 145.71.34.1 a MTU 1006.

Proto všichni uživatelé, kterým chcete povolit staticky přidělovaný vytáčený IP-přístup, musí mít záznam v souboru `/etc/diphosts`. Pro uživatele, kteří vytáčejí konkrétní port, s nímž si dynamicky domlouvají detaily spojení, potřebujete mít záznam pro zařízení `tty`. Nezapomeňte nakonfigurovat alespoň jeden záznam pro každé zařízení `tty`, které používají tito uživatelé, aby pro ně byla k dispozici vhodná konfigurace bez ohledu na modem, se kterým se spojí.

Když se uživatel přihlásí, odpoví na normální přihlašovací výzvu svým přihlašovacím SLIP id a heslem. Po ověření neuvidí uživatel žádné zprávy a zařízení se přepne do režimu SLIP. Uživatel by se potom měl bez potíží připojit a příslušnému zařízení budou nastaveny parametry ze souboru `diphosts`.

6.28.3 SLIP-server používající balík dSLIP

Matt Dillon <dillon@apollo.west.oic.com> napsal balík, který se umí nejenom přihlásit, ale také odhlásit. Jde o kombinaci malých programů a skriptů, které se starají o vaše spojení. Potřebujete mít nainstalovaný program `tcsh`, protože ho ke své činnosti vyžaduje minimálně jeden skript. Matt Dillon dodává také utilitu `expect` (v binárním tvaru), kterou také vyžaduje jeden ze skriptů. Ke zprovoznění tohoto balíku budete potřebovat určitou zkušenost s programem `expect`, ale tím se nenechte odradit.

Matt Dillon popsal instalaci v souboru `README`, takže se jí zde nebudu zabývat.

Balík `dSLIP` získáte na jeho domovské adrese:

apollo.west.oic.com /pub/linux/dillon_src/dSLIP203.tgz

nebo na adrese:

sunsite.unc.edu /pub/Linux/system/Network/serial/dSLIP203.tgz

Před spuštěním příkazu `make install` si přečtěte soubor `README` a vytvořte příslušné záznamy v souborech `/etc/passwd` a `/etc/group`.

6.29 Podpora protokolu STRIP (Standard Radio IP)

Názvy zařízení STRIP jsou „st0“, „st1“ atd.

Volby kompilace jádra

```
Network device support  --->
    [*] Network device support
    ....
    [*] Radio network interfaces
    < > STRIP (Metricom starmode radio IP)
```

Protokol STRIP byl navržen speciálně pro rádiové modemy Metricom v rámci výzkumného projektu vedeného na Stanford University a pojmenovaného MosquitoNet Project <http://mosquitonet.Stan-ford.EDU/mosquitonet.html>. Je to velice zajímavé čtení, a to i pro ty, kdo se o tento projekt přímo nezajímají.

Rádiové modemy Metricom se připojují k sériovému portu, využívají široké spektrum technologií a zvládají přenosové rychlosti 100 kbps. Informace o rádiových modemech Metricom získáte na adrese <http://www.metricom.com/>.

V současné době nepodporují standardní síťové nástroje ovladač protokolu STRIP, takže si budete muset ze serveru MosquitoNet stáhnout nějaké upravené nástroje. Podrobnosti ohledně softwaru, který potřebujete, najdete na stránce <http://mosquitonet.Stanford.EDU/strip.html>.

Co se konfigurace týče, pomocí upraveného programu `slattach` nastavíte chování linky sériového zařízení `tty` na STRIP a potom nakonfigurujete výsledné zařízení „st[0-9]“ jako pro Ethernet, jen s jednou výjimkou. Z technických důvodů nepodporuje protokol STRIP protokol ARP, takže je třeba ručně upravit záznamy ARP pro každého hostitele ve vaší podsíti. To by nemělo být obtížné.

6.30 Token Ring

Názvy zařízení token ring jsou „tr0“, „tr1“ atd. Token Ring je standardní LAN protokol vytvořený firmou IBM, který zabraňuje kolizím pomocí mechanismu, dávajícího v určitou dobu pouze jedné stanici v síti LAN právo vysílat. „Token“ vlastní vždy pouze jedna stanice a pouze ona může přenášet data. Po skončení přenosu předá token další stanici. Token putuje dokola po všech aktivních stanicích, odtud tedy název „Token Ring“.

Volby kompilace jádra

```
Network device support  --->
  [*] Network device support
  ....
  [*] Token Ring driver support
  < > IBM Tropic chipset based adaptor support
```

Konfigurace protokolu Token Ring je kromě názvu síťového zařízení identická s konfigurací Ethernetu.

6.31 X.25

X.25 je kruhový protokol založený na přepínání paketů definovaný organizací C.C.I.T.T. (Sdružení, které navrhuje standardy, jež se používají telekomunikačními společnostmi téměř na celém světě). Na implementaci protokolů AX.25 a LAPB se pracuje a poslední jádra verzí 2.1.* je již obsahují.

Vývoj vede Jonathon Naylor jsn@cs.nott.ac.uk a za účelem řešení problémů kolem protokolu AX.25 byla zřízena diskusní skupina. Chcete-li se do ní přihlásit, pošlete zprávu na adresu majordomo@vger.rutgers.edu a do jejího těla napište text „subscribe linux-x25“.

Dřívější verze konfiguračních nástrojů získáte na Jonathonově anonymním FTP na adrese <ftp://ftp.cs.nott.ac.uk/jsn/>.

6.32 Karta WaveLan

Názvy zařízení WaveLan jsou „eth0“, „eth1“ atd.

Volby kompilace jádra

```
Network device support  --->
  [*] Network device support
  ....
  [*] Radio network interfaces
  ....
  <*> WaveLAN support
```

Karta WaveLan je bezdrátová síťová karta. Při používání je velmi podobná ethernetové kartě a stejným způsobem se i konfiguruje.

Informace o kartě WaveLan získáte na adrese <http://www.wavelan.com/>.

7 Kabely a kabeláž

Ti z vás, kteří mají zkušenosti s pájením, si možná budou chtít vyrobit vlastní kabely, kterými by propojili své linuxové počítače. Následující diagramy kabelů by vám měly pomoci.

7.1 Sériový kabel null modem

Ne všechny kabely null modem jsou stejné. Většina těchto kabelů umí jen simulovat přítomnost všech příslušných signálů a přehodit vstupní a výstupní data. To je sice pěkné, ale znamená to, že musíte používat softwarovou kontrolu toku dat (XON/XOFF), která je méně efektivní než hardwarová. Následující kabel poskytuje nejlepší možnou signalizaci mezi počítači a umožňuje používat hardwarovou (RTS/CTS) kontrolu toku dat.

Název PINu	PIN	PIN
Tx Data	2	3
Rx Data	3	2
RTS	4	5
CTS	5	4
Uzemění	7	7
DTR	20	8
DSR	6	20
RLSD/DCD	8	6

7.2 Kabel pro paralelní port (kabel PLIP)

Pokud hodláte používat protokol PLIP ke komunikaci dvou počítačů, můžete použít tento kabel bez ohledu na typ instalovaného paralelního portu.

Poznámky:

- Nespojujte piny označené hvězdičkou („*“).
- Zvlášť uzemněny jsou piny 18,19,20,21,22,23 a 24.
- Má-li vámi použitý kabel kovové stínění, měl by být **na jednom konci** opatřen kovovou zástrčkou DB-25.

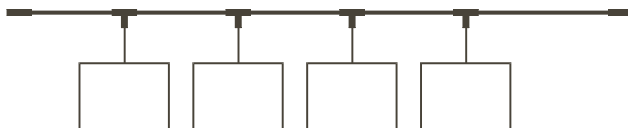
Upozornění: Špatně zapojený kabel PLIP může zničit kartu řadiče. Při zapojování buďte proto velice opatrní a dvakrát zkontrolujte každé spojení, abyste se vyhnuli problémům.

Název PINu	PIN	PIN
STROBE	1 *	
D0 - ERROR	2	15
D1 - BUSY	3	13
D2 - PAPOUT	4	12
D3 - ACK	5	10
D4 - BUSY	6	11
D5	7 *	
D6	8 *	
D7	9 *	
ACK - D3	10	5
BUSY - D4	11	6
PAPOUT - D2	12	4
SLCT - D1	13	3
FEED	14 *	
ERROR - D0	15	2
INIT	16 *	
SLCTIN	17 *	
Uzemnění	25	25

I když můžete kabel PLIP používat na velké vzdálenosti, měli byste se tomu, je-li to možné, vyhnout. Specifikace tohoto kabelu umožňuje délku kolem 1 metru. Při používání dlouhých kabelů tohoto typu dávejte proto pozor na zdroje elektromagnetických polí, jako je blesk, vodiče elektrického proudu a rádiové vysílače, které je mohou rušit a někdy též zničit váš řadič. Pokud opravdu potřebujete spojit dva počítače na velkou vzdálenost, měli byste si opatřit dvojici ethernetových karet a použít raději koaxiální kabel.

7.3 Ethernetový kabel 10base2 (tenký koaxiál)

10base2 je standard ethernetového kabelu, který specifikuje použití 52ohmového koaxiálního kabelu o průměru kolem 5 milimetrů. Při spojování počítačů kabelem 10base2 je třeba dodržet dvě důležitá pravidla. Zaprvé musíte použít terminátory **na obou koncích** kabelu. Terminátor je rezistor 52 ohmů, který zajišťuje, aby byl signál po dosažení konce absorbován a nikoliv odražen. Bez terminátoru na každém konci kabelu může být Ethernet nespolehlivý nebo nemusí vůbec fungovat. Normálně se používají k propojení počítačů konektory ve tvaru T, takže získáte následující zapojení:



- na každém konci kabelu reprezentuje terminátor
- reprezentuje délku koaxiálního kabelu s BNC-konektory na obou koncích
- T** reprezentuje konektor ve tvaru T. Délka kabelu mezi T-konektorem a vlastní ethernetovou kartou by měl být co nejmenší, v ideálním případě je T-konektor zapojen přímo do ethernetové karty.

7.4 Ethernetový kabel Twisted pair

Pokud máte pouze dvě ethernetové karty twisted pair a chcete je propojit, nepotřebujete rozbočovač. Tyto karty můžete propojit přímo. Schéma zapojení najdete v dokumentu *Ethernet-HOWTO*.

8 Některé termíny používané v tomto dokumentu

Následuje seznam některých důležitých termínů používaných v tomto dokumentu.

ARP	Zkratka názvu protokolu Address Resolution Protocol a jedná se o způsob, jakým síťový počítač sdružuje IP-adresy se síťovými adresami.
ATM	Zkratka režimu Asynchronous Transfer Mode. Síť ATM balí data do bloků standardních velikostí, které může efektivně dopravovat z jednoho místa na druhé. ATM je síťová technologie založená na kruhovém přepínání paketů.
klient	Většinou se jedná o software na straně uživatele. Existují výjimky z tohoto tvrzení, například v okenním systému X11 běží na straně uživatele server a na vzdáleném počítači klient. Klient je program, který přijímá nějakou službu poskytovanou serverem. V případě systému <i>peer to peer</i> , jako je <i>SLIP</i> a <i>PPP</i> , navazuje klient spojení a vzdálený konec, ten, který je volán, se nazývá server.
datagram	Datagram je vyhrazený balík dat a hlaviček, který obsahuje adresy. Jedná se o základní přenosovou jednotku v síti IP. Někdy se také datagramům říká pakety

DLCI	DLCI znamená Data Link Connection Identifier a slouží k identifikaci jedinečného virtuálního spojení typu point to point v síti Frame Relay. Identifikátor DLCI normálně přiděluje poskytovatel sítě Frame Relay.
Frame Relay	Frame Relay je síťová technologie, která je ideální pro impulsní nebo ojedinelý přenos. Sdílejí-li stejnou kapacitu sítě mnoho zákazníků, kteří používají síť v jinou dobu, mohou se značně zredukovat síťové náklady.
Hardwarová adresa	Jedná se o číslo, které slouží k jedinečné identifikaci hostitele v rámci fyzické sítě na úrovni přístupu k médiu. Příkladem jsou <i>ethernetové adresy</i> a <i>adresy AX.25</i> .
ISDN	Zkratka Integrated Services Digital Network. ISDN je standard, s jehož pomocí mohou telekomunikační společnosti doručovat hlasové nebo datové informace do domovů svých zákazníků. Z technického hlediska je ISDN kruhově přepínaná datová síť.
ISP	Zkratka Internet Service Provider (Poskytovatelé internetových služeb). Jsou to organizace nebo společnosti, které lidem poskytují připojení k Internetu.
IP-adresa	Jedná se o číslo, které slouží k jedinečné identifikaci TCP/IP-hostitele v síti. Adresa je 4 bajty dlouhá a zpravidla se zapisuje v tzv. tečkové notaci, kde je každý bajt uveden jako desítkové číslo a ta jsou vzájemně oddělená tečkami.
MSS	Maximum Segment Size (maximální velikost segmentu) je největší objem dat, který lze najednou přenést. Chcete-li zabránit lokální fragmentaci, měla by být MSS rovna velikosti MTU IP-hlavičky.
MTU	Maximum Transmission Unit (maximální přenosová jednotka) je parametr, který určuje největší datagram, který lze přenést přes IP-rozhraní, aniž by ho bylo nutné rozdělit na menší jednotky. Hodnota MTU by měla být větší než největší datagram, který chcete přenášet nefragmentován. Uvědomte si, že tak zabráníte pouze místní fragmentaci. Některá jiná linka totiž může mít nastavenou nižší hodnotu MTU a k fragmentaci datagramu pak dojde zde. Typické hodnoty jsou pro Ethernet 1500 bajtů a 576 bajtů pro rozhraní SLIP.
trasa (route)	Trasa je cesta, po které putují datagramy sítě ke svému cíli.

- server** Zpravidla se jedná o software na vzdáleném konci spojení. Server poskytuje určité služby jednomu nebo více klientům. Mezi servery patří *FTP*, *NFS* nebo *DNS*. V případě systémů *peer to peer*, jakým je například *SLIP* nebo *PPP*, je server na straně, která přijímá spojení.
- okno** Okno je největší množství dat, které je přijímací strana schopna v určitém okamžiku přijmout.

9 Linux pro ISP?

Pokud vás zajímá využití Linuxu pro účely ISP, doporučuji vám navštívit *Linux ISP homepage* <http://www.anime.net/linuxisp/>, kde najdete velký seznam odkazů na informace, které se vám mohou hodit.

10 Poděkování

Rád bych poděkoval následujícím lidem za jejich příspěvky do tohoto dokumentu (bez ohledu na pořadí): Terry Dawson, Axel Boldt, Arnt Gulbrandsen, Gary Allpike, Cees de Groot, Alan Cox, Jonathon Naylor, Claes Ensson, Ron Nessim, John Minack, Jean-Pierre Cocatrix, Erez Strauss.

Zvláštní díky patří Alessandru Rubinimu za jeho připomínky a korektury.

11 Autorská práva

Dokument NET-3-HOWTO, informace o instalaci a konfiguraci síťové podpory v systému Linux. Copyright (c) 1995 Terry Dawson.

Tento program je volně šířitelný. Můžete ho šířit a/nebo upravovat v souladu s 2. nebo pozdější verzí licence GNU General Public License publikovanou nadací Free Software Foundation.

Tento program je šířen s nadějí, že bude užitečný, ale BEZ JAKÝCHKOLIV ZÁRUK. Dále bez záruk OBCHODOVATELNOSTI nebo ZPŮSOBILOSTI PRO KONKRÉTNÍ ÚČEL. Podrobnosti najdete v General Public License.

Společně s tímto programem byste měli obdržet kopii licence GNU General Public License. Pokud nikoli, pak si o ni napište na adresu:

Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Linux Intranet Server

Pramod Karnad, karnad@indiamail.com v2.11, 7. srpna 1997

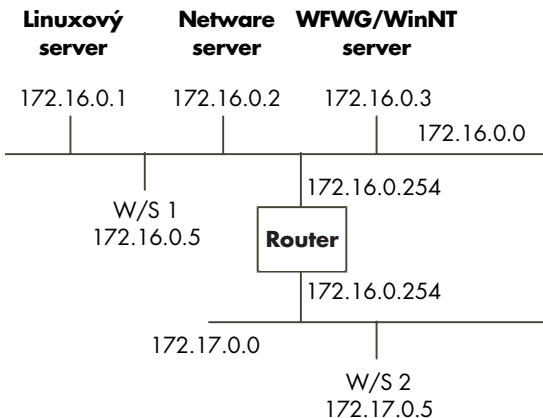
Tento dokument popisuje, jak nastavit Intranet s linuxovým serverem, který propojuje Unix, NetWare, NT a Windows. Tak po připojení na Linux získáte přístup ke všem různým platformám. Je zde podrobný popis k nastavení HTTP pomocí NCSA-serveru a k připojení k němu pomocí TCP/IP-klientů na Novellu; Microsoft pod Windows3.1, WFWG, Win95 a WinNT; a MacTCP na Apple PowerMac.

1 Úvod

Intranet je zjednodušeně řečeno implementací technologií Internetu v rámci nějaké uzavřené organizace bez nutnosti připojení ke globálnímu Internetu. Tato implementace se provádí takovým způsobem, aby bylo možné při minimálních nákladech, času a námaze poskytnout veškeré informační zdroje každému počítači. Tento dokument se pokouší osvětlit, jak vytvořit Intranet pomocí nástrojů, které jsou k dispozici, ale které jsou přesto velmi levné nebo jsou zcela zdarma.

Dokument předpokládá, že již víte, jak na vašem linuxovém serveru instalovat TCP/IP a jak jej fyzicky připojit pomocí ethernetové karty k vaší LAN. Předpokládá se zde také základní znalost systémů NetWare, WinNT a Mac. Konfigurace NetWare-serveru je zde ukázána na bázi verze 3.1x. Stejného výsledku dosáhnete za pomoci INETCFG. Na straně klienta se dokument zabývá systémy Windows 3.1x, Windows for Workgroups a Win95, WinNT a Apple PowerMac.

Soukromé síťové adresy (RFC-1918) 172.16.0.0 a 172.17.0.0 používám pouze jako příklady. V závislosti na vaší konfiguraci si můžete vybrat odpovídající adresy.



1.1 Co je nutné

Před instalací budete potřebovat následující software:

- Software HTTP-serveru, který je možné získat z One Step NCSA HTTPd Downloader na stránce <http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html>.*
- Novell NetWare Client, který je k dispozici na <http://support.novell.com/> (s klientem se dodávají i TCP/IP-soubory).
- Microsoft TCP/IP client, který je k dispozici na <http://www.microsoft.com/>.
- Apple MacTCP client, který je k dispozici na <http://www.apple.com/>.
- Prohlížeče WWW, jako je Netscape na <http://home.netscape.com/> nebo MS Internet Explorer na <http://www.microsoft.com/> nebo NCSA Mosaic na adrese <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html>.

1.2 Nové verze tohoto dokumentu

Nové verze tohoto dokumentu budou pravidelně posílány do comp.os.linux.announce a comp.os.linux.help. Budou také nahrávány na různé ftp-servery s Linuxem, včetně sunsite.unc.edu. Poslední verze tohoto dokumentu je k dispozici ve formátu HTML na adrese <http://www.inet.co.th/cyberclub/karnadp/http.html>.

* Poznámka korektora: Vhodnějším HTTP-serverem je server Apache, který můžete získat na adrese <http://www.apache.org>.

1.3 Ohlasy

Jestliže máte k dokumentu dotazy nebo komentáře, nebojte se je odeslat Pramodu Karnadovi - karnad@indiamail.com. Návrhy, kritika a pošta jsou vždy vítány. Jestliže v dokumentu naleznete chyby, dejte mi prosím vědět, abych je mohl v příští verzi opravit. Díky.

2 Instalace HTTP-serveru

Když chcete získat server, máte dvě možnosti: získat zdroj a sami si jej zkompileovat nebo získat již zkompileované binární soubory. Binární soubory pro verzi Linux (ELF) jsou k dispozici v NCSA, ale ne starší verze.

2.1 Přípravy před stažením

Server NCSA vás provede kroky s konfiguračními volbami a připraví pro vás různé soubory. Před stažením HTTPd si ale připravte odpovědi na následující dotazy.

2.1.1 Operační systém

Nejprve musíte vybrat, jestli se má stahovat zdroj nebo předkompilovaná verze softwaru. Jestliže v menu není váš systém, musíte vzít zavděk implicitním zdrojem a sami jej zkompileovat.

Zjištění verze vašeho Linuxu provedete v jeho příkazovém řádku pomocí

```
linux:~$ uname -a
```

a získáte tak zhruba takovouto odpověď

```
linux:~$ uname -a
Linux 2.0.29 #4 Tue Sep 13 04:05:51 CDT 1994 i586
linux:~$
```

Zbývající parametry mohou být určeny před stažením nebo nakonfigurovány později úpravou souboru `srm.conf` v adresáři `/usr/local/etc/httpd/conf`. Názvy direktiv, které se právě ukazují v souboru `httpd.conf`, jsou v závorkách. Jedinou výjimkou je zde `DocumentRoot`, který se objevuje v souboru `srm.conf`.

2.1.2 Typ procesu (ServerType)

Zde se určí, jak na vašem stroji poběží HTTPd-server. Základní metodou je „standalone“ (samostatně). Takto běží HTTP-démon nepřetržitě. Jestliže nahrajete HTTPd pod „inetd“, bude binární soubor serveru nahráván do paměti při každé žádosti, což by mohlo server zpomalit.

2.1.3 Navázání portu (Port)

Určuje, na který port vašeho stroje se HTTPd-démon naváže a bude naslouchat HTTP-žádostem. Jestliže se můžete přihlásit jako „root“, použijte implicitní nastavení 80. Jinak použijte hodnotu v rozsahu 1 025 až 65 536.

2.1.4 Identifikace uživatele na serveru (User)

Jedná se o id uživatele, na které se server změní při reakcích na žádosti a při práci se soubory. Tento dotaz zodpovídáte pouze pokud server používáte jako „standalone“. Jestliže nemáte rootovská práva, použijte vlastní uživatelské jméno. Jestliže jste systémovým správcem, můžete vytvořit speciálního uživatele, aby bylo možné kontrolovat přístupová práva k souborům.

2.1.5 Identifikace skupiny na serveru (Group)

Jedná se o id skupiny, na které se server změní při reakcích na žádosti a při práci se soubory. Je to podobné jako u Server User a používá se pouze pokud server používáte jako „standalone“.

Jestliže nemáte rootovská práva, použijte název vaší primární skupiny. Vaše skupiny zjistíte v Linuxu v příkazovém řádku pomocí příkazu **groups**.

2.1.6 E-mailová adresa administrátora serveru (ServerAdmin)

Toto je e-mailová adresa, na kterou by měl uživatel odesílat poštu s hlášením problémů na serveru. Sem můžete nastavit svoji osobní adresu.

2.1.7 Umístění adresáře serveru (ServerRoot)

Místo na vašem systému, na kterém sídlí server. Jestliže máte rootovská práva, ponechejte jej v doporučené podobě v adresáři `/usr/local/etc/httpd`. Jestliže se jako root přihlásit nemůžete, zvolte podadresář ve vašem domovském adresáři. Cestu k vašemu domovskému adresáři zjistíte příkazem `pwd`.

2.1.8 Umístění HTML-souborů (DocumentRoot)

Zde jsou uloženy HTML-soubory. Implicitní umístění je `/usr/local/etc/httpd/htdocs`. Můžete je nastavit do domovského adresáře speciálního uživatele, kterého vyberete ze Server User identifiy, nebo do podadresáře svého domovského adresáře, pokud se nemůžete přihlásit jako root.

Když nevíte, použijte implicitní nastavení. Nyní, když máte odpovědi na předchozí otázky, můžete si stáhnout NCSA HTTPd z <http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html>. Před instalací si na <http://hoohoo.ncsa.uiuc.edu/docs/> pročtete dokumentaci k HTTPd. Jestliže se chystáte kompilovat kód, musíte upravit Makefile v každém z adresářů `support`, `src` a `cgi-src`. Jestliže je vaše verze Linuxu podporována, stačí vám v hlavním adresáři (např. `/usr/local/etc/httpd`) zadat `make linux`.

2.2 Kompilace HTTPd

Kompilace je snadná, zadejte v kořenovém adresáři serveru do příkazového řádku `make linux`.

Poznámka: Uživatelé předELFového Linuxu musí před kompilací HTTPd v souboru `portability.h` zrušit komentář u řádku `#define NO_PASS` a v souboru `Makefile` nastavit `DBM_LIBS= -ldb`.

3 Testování HTTPd

Po instalaci HTTPd se přihlašte jako root a spustěte jej zadáním `httpd &` (předpokládáme instalaci jako standalone). Nyní by měl být vidět v seznamu příkazu `ps`. Nejjednodušším způsobem testování HTTPd je `telnet`. V příkazovém řádku Linuxu zadejte:

```
linux:~$ telnet 172.16.0.1 80
```

kde 80 je implicitní port pro HTTP. Jestliže jste „Port“ nastavili na jinou hodnotu, napište ji sem místo 80. Odpověď bude vypadat asi takto:

```
Trying 172.16.0.1...
Connected to linux.mydomain
Escape character is '^]'
```

Nyní, když zadáte jakýkoliv znak a stisknete **Enter**, objeví se odpověď, podobná té následující.

```
HTTP/1.0 400 Bad Request
```

Date: Wed, 10 Jan 1996 10:24:37 GMT

Server: NCSA/1.5

Content-type: text/html

```
<HEAD><TITLE>400 Bad Request < /TITLE> < /HEAD>
<BODY><H1>400 Bad Request < /H1>
Your client sent a query that this server could
not understand.<P>
Reason: Invalid or unsupported method.<P>
</BODY>
```

Nyní jsme připraveni se na tento server připojit pomocí jiného PC a prohlížeče WWW.

4 Připojování na linuxový server

Použité adresovací schéma je vidět z diagramu v první části. Pracovní stanice 1 (W/S1) je na síti 172.16.0.0 a může se připojit na linuxový server přímo, zatímco pracovní stanice 2 (W/S2) je na síti 172.17.0.0 a musí pro připojení na Linux využít bránu (router) 172.17.0.254. Tato informace o bráně musí být použita pouze při konfiguraci klientů na W/S2. NetWare se na bránu odkazuje jako na „ip_router“ (ip-router).

Já pro ilustraci nastavení klienta použiji W/S2. Při nastavení W/S1 stačí zaměnit adresy 172.17.0.5 na 172.16.0.5 a ignorovat všechny odkazy na bránu/router.

Jestliže nemáte router, můžete tuto část přeskočit a postoupit až k

- 4.2, jestliže používáte NetWare server
- 4.4, jestliže používáte Microsoft Client

4.1 Nastavení linuxového serveru

Jestliže nemáte router, můžete tuto část vynechat.

Musíte nakonfigurovat linuxový server, aby rozpoznal router a umožnil W/S2-připojení na webový server. Aby bylo možné nastavit linuxový server, musíte se přihlásit jako root. Na příkazovém řádku serveru zadejte

```
route add gw default 172.16.0.254
```

Aby bylo možné tuto bránu využívat při každém zavedení systému linuxového serveru, můžete editovat soubor `/etc/rc.d/rc.inet1*` a změnit řádek, obsahující definici brány na `GATEWAY = „172.16.0.254“`. Tento řádek nesmí být v komentáři.

Nebo můžete směrování přidat do sítí na druhé straně routeru. To se provádí pomocí

```
route add -net 172.17.0.0 gw 172.16.0.254
```

Směrování pak přidáním příkazu do souboru `/etc/rc.d/rc.local` nastavíte pro každé zavedení systému.

4.2 Nastavení NetWare-serveru

Aby bylo možné nastavit NetWare-server, musíte mít přístupová práva Supervisor nebo alespoň Console operator. Jestliže je nemáte, musíte požádat vašeho síťového správce (Network Administrator), aby vám s nastavením pomohl. Serveru nastavte typ rámce LAN na Ethernet_II. K tomu použijte tyto příkazy (můžete je také vložit do souboru `AUTOEXEC.ncf`).

```
load NE2000 frame=Ethernet_II name=IPNET
load TCPIP
bind IP to IPNET addr=172.16.0.2 mask=FF.FF.FF.0
```

Při nahrávání ovladače NE2000 můžete v závislosti na konfiguraci vašeho stroje určit slot nebo číslo desky (například: `load NE2000 slot=3 frame=.....`).

4.3 Nastavení NetWare-klienta

Na PC máte možnosti Win3.1, WFWG nebo Win95. Instalační procedura se mezi Win95 a staršími verzemi liší podle toho, jestli používáte 32bitového klienta od Microsoftu nebo Novellu. Jestliže použijete 16bitového klienta, procedura bude stejná a vy se můžete řídit instalačními instrukcemi pro Windows 3.x. Při instalaci 32bitového klienta pro Win95 přejděte k části 4.3.2.

4.3.1 Windows 3.x

Jestliže používáte Win3.1 nebo WFWG, můžete nainstalovat NetWare Client (VLMs) a některé další soubory, které jsou na TCP/IP-disketě, jmenovitě

```
TCPIP.exe, VTCPIP.386, WINSOCK.dll a WLIBSOCK.dll
```

* Poznámka překladatele: V konkrétní distribuci může být umístění tohoto souboru jiné, případně mohou být soubory zcela jinak uspořádány.

Povšimněte si, že soubor se liší od stejného souboru z Win95 a Trumpet. Nainstalujte NetWare Client s podporou pro Windows. Okopírujte VTCPIP.386, WINSOCK.dll a WLIB-SOCK.dll do adresáře SYSTEM a TCPIP.exe do adresáře NWCLIENT. Nyní upravte STARTNET.bat v adresáři NWCLIENT na

```
lsl
ne2000          ---> ovladač vaší síťové karty
c:\windows\odihlp.exe  ----> pokud používáte WFWG
ipxodi
tcpip          ---> doplňte tento řádek
nwip          ---> pokud používáte NetWare/IP
vlm
```

Vytvořte podadresář (například) \NET\TCP a okopírujte soubory HOSTS, NETWORKS, PROTOCOLS a SERVICES z /etc vašeho linuxového serveru nebo adresáře SYS:ETC vašeho NetWare-serveru. Editujte okopírovaný soubor HOSTS, kam přidáte řádek pro váš nový linuxový server. Díky tomu se budete moci ve vašem prohlížeči WWW na linuxový server odkazovat jako na <http://linux.mydomain/> místo <http://172.16.0.1/>

```
127.0.0.1      localhost
172.16.0.1    linux.mydomain
```

Editujte v adresáři NWCLIENT soubor NET.cfg

```
Link Driver NE2000
port 300
int 3
MEM D0000
FRAME Ethernet_802.2

; ---- přidejte tyto řádky ----

FRAME Ethernet_II

Protocol TCPIP
PATH TCP_CFG C:\NET\TCP
ip_address 172.17.0.5
ip_netmask 255.255.255.0
ip_router 172.17.0.254  ---> přidejte adresu vaší brány pouze
```

```

---> pokud musíte tuto bránu použít
---> k dosažení vašeho HTTP-serveru

```

Link Support

MemPool 6192 ---> minimum je 1024. Zkuste jiné hodnoty.

Buffers 10 1580 ---> Tato hodnota může být opět vyladěna.

```

;-----
; Pokud používáte NetWare/IP, budete muset přidat následující řádky
;
NWIP
    NWIP_DOMAIN_NAME mydomain
    NSQ_BROADCAST ON
    NWIP1_1 COMPATIBILITY OFF
    AUTORETRIES 1
    AUTORETRY SECS 10

```

Edituje v adresáři WINDOWS soubor SYSTEM.ini a pro VTCP/IP.386 přidejte tento údaj

```

[386Enh]
.....
network=*vnetbios, vipx.386, vnetware.386, VTCP/IP.386
.....

```

Provedte restart vašeho PC, spusíte STARTNET.bat a nyní můžete k přístupu na vaše webové stránky používat svůj oblíbený prohlížeč WWW. Nepotřebujete se přihlašovat na NetWare a nemusíte spouštět TCPMAN (pokud používáte Trumpet Winsock).

4.3.2 Windows 95

Tato část vysvětluje, jak ve Win95 instalovat 32bitového klienta. Nejprve musíte nainstalovat následující:

```

Client for NetWare Networks (od Microsoftu nebo Novellu)
Microsoft TCP/IP Protocol
Network Adapter

```

Tyto položky instalujete po klepnutí na My Computer, Control Panel, Networks. Klepněte na Add. Nyní se ocitnete v okně, které zobrazuje Client, Adapter, Protocol a Service. Při instalaci Client for NetWare Networks:

1. Klepněte dvakrát na Client
2. Klepněte na Microsoft nebo Novell
3. Klepněte dvakrát na Client for NetWare Networks

Při instalaci TCP/IP Protocol:

1. Klepněte dvakrát na Protocol
2. Klepněte na Microsoft
3. Klepněte dvakrát na TCP/IP

Windows 95 implicitně automaticky instaluje několik dalších protokolů. Odstraníte je tak, že na ně klepnete a pak klepnete na tlačítko Remove. Win95 většinou instaluje protokol Microsoft NetBeui a kompatibilní protokol IPX/SPX. Protokol NetBEUI můžete smazat, ale protokol IPX/SPX budete potřebovat, pokud se budete chtít přihlásit na NetWare Server.

Nastavení TCP/IP provedete klepnutím na TCP/IP, klepnutím na Properties, klepnutím na tabulku IP-adress

V boxu Specify an IP address zadejte vaši IP-adresu 172.17.0.5

V boxu Subnet Mask zadejte 255.255.255.0

vyberte tabulku Gateway

V boxu New gateway zadejte adresu vaší brány 172.17.0.254

Klepněte na tlačítko Add

Adresa brány by se nyní měla objevit v boxu s instalovanými branami. Nyní klepněte na OK.

Obrázíte zprávu o restartu systému. Proveďte restart. Nyní budete moci k připojení na váš HTTP-server využívat prohlížeč.

4.4 Nastavení Microsoft Client

Jestliže pro přístup na síť používáte Microsoft Client, tato část popisuje, jak instalovat TCP/IP pro:

- 4.4.1 Windows for Workgroups
- 4.4.2 Windows 95
- 4.4.3 Windows NT

Poznámka: Abyste se ve svém prohlížeči WWW a všech internetových příkazech mohli na linuxový server odkazovat jako na `http://linux.mydomain/` místo `http://172.16.0.1/`, musíte editovat soubor `hosts`. Pro každý server (NetWare, Unix, WinNT) můžete přidat více údajů. Windows obsahují soubor `HOSTS` v adresáři `\WINDOWS` nebo `\WINDOWS\SYSTEM`, v závislosti na verzi. Editujte tento soubor a pro váš linuxový server přidejte řádek:

```
127.0.0.1 localhost
        172.16.0.1   linux.mydomain
        172.16.0.2   netware.mydomain
        172.16.0.3   winNT.mydomain
        172.16.0.5   ws_1
```

4.4.1 Windows for Workgroups

Tato část popisuje, jak instalovat 32bitového klienta na WFWG. Nejprve si musíte od Microsoftu obstarat ovladače TCP/IP pro Windows. Aktuální verzi je 3.11b a k dispozici je na adrese `ftp://ftp.microsoft.com` i na jiných místech, jako `tcp32b.exe`. Před nahráním 32bitového ovladače TCP/IP musíte mít ale nahrán Win32s.

Po vložení souborů TCP/IP do dočasného adresáře (například `C:\TEMP`) zkontrolujte adresář `\WINDOWS\SYSTEM`, zda neobsahuje kopie `OEMSETUP.INF`. Jestliže zde nějaké jsou, přejmenujte je. Nyní okopírujte soubor `OEMSETUP.INF` z adresáře `TEMP` do adresáře `\WINDOWS\SYSTEM`. Jestliže máte v systému jiné TCP/IP-ovladače, nejprve je odstraňte.

Spusťte nastavení Network Setup nebo Windows Setup/Change Network

Klepněte na tlačítko Networks
 Klepněte na Install Microsoft Windows Network
 Zvolte podporu pro další síť (je-li to nutné)
 Klepněte na OK

Měli byste být vyzváni k výběru vašeho síťového adaptéru - vyberte jej. Jestliže k výzvě nedojde, potom

Klepněte na tlačítko Adapter
 vyberte adaptér (řekněme NE2000)
 Klepněte na OK
 Klepněte na tlačítko Protocol
 vyberte protokol MS TCP/IP-32
 klepněte na OK

Nyní budete vyzváni ke konfiguraci sady TCP/IP-protokolů. Vždy je můžete překonfigurovat pomocí nastavení TCP/IP-protokolu v boxu Adapters a klepnutí na tlačítko Setup.

V boxu IP address zadejte 172.17.0.5

V boxu Subnet Mask zadejte 255.255.255.0

V boxu Default gateway zadejte adresu vaší brány 172.17.0.254

Klepněte na OK. Obdržíte zprávu o restartu systému. Provedte restart. Nyní budete moci k připojení na váš HTTP server využívat prohlížeč.

4.4.2 Windows 95

Tato část popisuje, jak instalovat 32bitového klienta pro Microsoft na Win95. Nejprve musíte nainstalovat následující

Client for Microsoft Networks

Microsoft TCP/IP Protocol

Network Adapter

Tyto položky instalujete po klepnutí na My Computer, Control Panel, Networks. Klepněte na Add. Nyní se ocitnete v okně, které zobrazuje Client, Adapter, Protocol a Service. Při instalaci Client for Microsoft Networks:

1. Klepněte dvakrát na Client
2. Klepněte na Microsoft
3. Klepněte dvakrát na Client for Microsoft Networks

Při instalaci TCP/IP Protocol:

1. Klepněte dvakrát na Protocol
2. Klepněte na Microsoft
3. Klepněte dvakrát na TCP/IP

Windows 95 implicitně automaticky instaluje několik dalších protokolů. Odstraní je tak, že na ně klepnete a pak klepnete na tlačítko Remove. Win95 většinou instaluje protokol Microsoft NetBeui.

Nastavení TCP/IP provedete klepnutím na TCP/IP, klepnutím na Properties, klepnutím na tabulku IP-adress

V boxu Specify an IP address zadejte vaši IP-adresu 172.17.0.5

V boxu Subnet Mask zadejte 255.255.255.0

vyberte tabulku Gateway

V boxu New gateway zadejte adresu vaší brány 172.17.0.254
Klepněte na tlačítko Add

Adresa brány by se nyní měla objevit v boxu s instalovanými branami. Nyní klepněte na OK.

Obdržíte zprávu o restartu systému. Proveďte restart. Nyní budete moci k připojení na váš HTTP-server využívat prohlížeč.

4.4.3 Windows NT

Tato část popisuje instalaci TCP/IP-klienta pro WinNT 4.0. Spusťte Control Panel/Network

Vyberte tabulku Adapter.

Klepněte na Add (jestliže ještě nemáte, tak přidáte nový adaptér)

Měli byste být vyzváni k výběru vašeho síťového adaptéru - vyberte jej. Pro přidání protokolů:

Vyberte tabulku protocols

Klepněte na Add

Vyberte protokol TCP/IP

Klepněte na OK

Nyní budete vyzváni ke konfiguraci sady TCP/IP-protokolů. Vždy je můžete překonfigurovat pomocí nastavení TCP/IP-protokolu a klepnutí na tlačítko Properties.

Vyberte tabulku IP Address

Označte box 'Specify an IP address'

V boxu IP address zadejte 172.17.0.5

V boxu Subnet Mask zadejte 255.255.255.0

V boxu Default gateway zadejte adresu vaší brány 172.17.0.254

Klepněte na OK. Obdržíte zprávu o restartu systému. Proveďte restart. Nyní budete moci k připojení na váš HTTP-server využívat prohlížeč.

4.5 Nastavení TCP/IP na počítači Macintosh

Jestliže pro přístup na síť používáte Macintosh, tato část popisuje, jak instalovat MacTCP na PowerMac.

Poznámka: Abyste se ve svém prohlížeči WWW a všech internetových příkazech mohli na linuxový server odkazovat jako na `http://linux.mydomain/` místo `http://172.16.0.1/`, musíte editovat soubor `hosts`. Formát souboru `hosts` se liší od formátu, který používá Unix. Soubor `hosts` je zde vytvořen na základě RFC-1035. Pro každý server (NetWare, Unix, WinNT) můžete přidat více údajů. MacOS ukládá soubor `HOSTS` do **Preferences folder** pod **System folder**. Editujte tento soubor a pro váš linuxový server přidejte řádek:

```
linux.mydomain      A  172.16.0.1
netware.mydomain    A  172.16.0.2
winNT.mydomain      A  172.16.0.3
ws_1                 A  172.16.0.5
```

4.5.1 MacTCP

Tato část popisuje instalaci MacTCP. Nejprve musíte od Applu získat soubory MacTCP nebo je nainstalovat z Internet Connection CD. Při konfiguraci MacTCP klepněte na Apple Menu/Control Panels/ TCP/IP. Na obrazovce změňte nastavení pro „Connect via:“ na „Ethernet“.

Změňte nastavení „Configure“ na „Manually“.

V boxu IP address zadejte 172.17.0.5

V boxu Subnet Mask zadejte 255.255.255.0

V boxu Router address zadejte adresu vaší brány 172.17.0.254

Klepněte na OK. Nyní budete moci k připojení na váš HTTP-sserver využívat prohlížeč.

5 Nastavení Intranetu

Intranet by nebyl Intranetem bez sdílení zdrojů na různých platformách. Budete potřebovat podporu pro další systémy souborů, aby bylo možné využívat data, která jsou na nich k dispozici. Tento dokument nabízí instrukce pro připojení k těmto populárním systémům souborů.

- 5.1 NCPFS pro NetWare
- 5.2 SMBFS pro Windows
- 5.3 NFS pro Unix

Tyto systémy souborů mohou být kompilovány do jádra Linuxu nebo přidány jako moduly, v závislosti na verzi Linuxu. Jestliže nejste v kompilaci jádra příliš sběhlí, můžete se více do-

zvědět z dokumentu o jádru a o modulech. Tyto moduly jsou k dispozici na adrese <http://sunsite.unc.edu/mdw/HOWTO/Kernel-HOWTO.html> a <http://sunsite.unc.edu/mdw/HOWTO/Module-HOWTO.html>.

5.1 Ncpfs

Při sdílení souborů na NetWare-serveru budete potřebovat podporu pro NCP (ncpfs). Ncpfs funguje s jádrem verze 1.2.x a od 1.3.71 výše. Nefunguje se starými jádry 1.3.x. Nevyužívá NDS-databázi NetWare 4.0, ale dokáže využít bindery. Jestliže používáte NetWare 4.x, můžete na konzole na určitých místech pomocí příkazu `Set Bindery Context` umožnit podporu bindery:

```
set Bindery Context = CORP.MYDOM;WEBUSER.MYDOM
```

V předchozím případě byla podpora bindery umožněna na dvou místech.

Z URL <ftp://sunsite.unc.edu/pub/Linux/system/fileystems/ncpfs/ncpfs.tgz> (aktuálně `ncpfs-2.0.10`) ze Sunsite musíte získat utility pro ncpfs.

5.1.1 Instalace

Utility `ncpfs` nainstalujete zadáním

```
zcat ncpfs.tgz | tar xvf -
```

soubory se tak vloží do vlastního adresáře. V tomto případě budete mít adresář `ncpfs-2.0.10`. Před započítím instalace sem změňte aktuální adresář. Pročtete si `README` a je-li to nutné, editujte `Makefile`.

Instalace `ncpfs` závisí na verzi jádra, kterou používáte. Pro jádro 1.2 stačí zadat „make“. Následné „make install“ nainstaluje spustitelné soubory a manuálové stránky.

Jestliže používáte Kernel 1.3.71 nebo pozdější, možná budete muset svoje jádro překompilovat. U těchto jader je část jádra pro `ncpfs` již začleněna v hlavním zdrojovém stromu. Jestli jádro musíte překompilovat, zjistíte zadáním

```
cat /proc/fileystems
```

Měl by se objevit řádek, sdělující, že jádro zná `ncpfs`.

Jestliže zde `ncpfs` není, můžete buď překompilovat jádro, nebo přidat `ncpfs` jako modul. Při kompilaci jádra byste měli zadat „make config“ a jakmile se vás dotáže na

```
The IPX protocol (CONFIG_IPX) [N/y/?]
```

odpovězte jednoduše „y“. Plnou vnitřní síť, která bude v dotazech následovat, pravděpodobně potřebovat nebudete. Jakmile je jádro úspěšně nainstalováno, proveďte restart systému, zkontrolujte `/proc/filesystems`, a jestliže je vše v pořádku, pokračujte v instalaci utilit `ncpfs`. Změňte aktuální adresář na adresář s přetaženými soubory `ncpfs` a zadejte „make“. Po dokončení kompilace zadejte „make install“, čímž se nainstalují různé utility a manuálové stránky.

5.1.2 Připojení Ncpfs

Kontrolu instalace proveďte pomocí

```
ipx_configure --auto_interface=on --auto_primary=on
```

....počkejte 10 sekund a napište

```
slist
```

Měli byste být schopni vidět seznam vašich NetWare-serverů. Nyní jsme připraveni ke sdílení souborů z NetWare-serveru.

Předpokládejme, že potřebujeme využít HTML-soubory z adresáře `\home\htmldocs` z VOL1: na serveru MYDOM_NW. Doporučuji na tomto serveru vytvořit nového uživatele (například „EXPORT“ s heslem „EXP123“, kterému pomocí `SYSCON` nebo `NWADMIN` dáte příslušná přístupová práva do tohoto adresáře.

Na linuxovém stroji vytvořte nový adresář `/mnt/MYDOM_NW`. Nyní zadejte příkaz

```
ncpmount -S MYDOM_NW -U EXPORT -P EXP123 /mnt/MYDOM_NW
```

kterým se připojí systém souborů Netware. Zadáním příkazu

```
ls /mnt/MYDOM_NW/vol1/home/htmldocs
```

zobrazíte seznam všech souborů v `MYDOM_NW/VOL1:\HOME\HTMLDOCS` (v systému pojmenování souborů podle NetWare). Jestliže máte nějaké problémy, přečtěte si dokument o IPX, který naleznete na <http://sunsite.unc.edu/mdw/HOWTO/IPX-HOWTO.html>.

5.2 Smbfs

Aby bylo možné sdílet soubory na serveru Windows, budete potřebovat podporu pro SMB (smbfs).

Z <ftp://sunsite.unc.edu/pub/Linux/system/filesystems/smbfs/smbfs.tgz> (aktuálně smbfs-2.0.1) ze Sunsite musíte získat utility pro smbfs.

5.2.1 Instalace

Utility nainstalujete zadáním

```
zcat smbfs.tgz | tar xvf -
```

soubory se tak vloží do vlastního adresáře. V tomto případě budete mít adresář `smbfs-2.0.10`. Před započítím instalace sem změňte aktuální adresář. Pročtete si `README` a je-li to nutné, editujte `Makefile`.

Instalace `smbfs` závisí na verzi jádra, kterou používáte. Pro jádro 1.2 stačí zadat „`make`“. Následné „`make install`“ nainstaluje spustitelné soubory a manuálové stránky.

Jestliže používáte Kernel 2.0 nebo pozdější, možná budete muset svoje jádro překompilovat. U těchto jader je jejich část pro smbfs již začleněna v hlavním zdrojovém stromu. Zda jádro musíte překompilovat, zjistíte zadáním

```
cat /proc/filesystems
```

Měl by se objevit řádek, sdělující, že jádro zná smbfs.

Jestliže zde smbfs není, můžete buď překompilovat jádro, nebo přidat smbfs jako modul. Při kompilaci jádra byste měli zadat „`make config`“ a jakmile se vás dotáže na podporu SMBFS, odpovězte jednoduše „`y`“. Jakmile je jádro úspěšně nainstalováno, proveďte restart systému, zkontrolujte `/proc/filesystems`, a jestliže je vše v pořádku, pokračujte v instalaci utilit smbfs. Změňte aktuální adresář na adresář s přetaženými soubory smbfs a zadejte „`make`“. Po dokončení kompilace zadejte „`make install`“, čímž se nainstalují různé utility a manuálové stránky.

5.2.2 Připojení smbfs

V našem příkladu předpokládejme, že WinNT server se nazývá „`MYDOM_NT`“ a sdílí svůj adresář `C:\PUB\HTMLDOCS` se sdíleným názvem „`HTMLDOCS`“ a bez hesla. Na linuxovém stroji vytvořte nový adresář `/mnt/MYDOM_NT`. Nyní zadejte příkaz

```
smbmount //MYDOM_NT/HTMLDOCS /mnt/MYDOM_NT -n
```

kterým se připojí smbfs (sdílení ve Windows). Jestliže příkaz nefunguje, vyzkoušejte

```
smbmount //MYDOM_NT/COMMON /mnt/MYDOM_NT -n -I 172.16.0.3
```

Zadáním příkazu

```
ls /mnt/MYDOM_NT
```

zobrazíte seznam všech souborů v `\\MYDOM_NT\PUB\HTMLDOCS` (v systému pojmenování souborů podle Windows).

5.3 NFS

Nejprve budete potřebovat jádro s NFSFS buď zakompilovaným, nebo k dispozici jako modul.

Předpokládejme, že máte server s Unixem, na kterém běží NFS s názvem MYDOM_UNIX a IP-adresou 172.16.0.4. Kontrolu zde exportovaných (sdílených) adresářů provedete zadáním příkazu

```
showmount -e 172.16.0.4
```

Jakmile známe exportované adresáře, můžeme je připojit zadáním příslušného příkazu `mount`. Doporučuji pod „/mnt“ vytvořit podadresář (řekněme) „MYDOM_UNIX“ a využít jej jako vaše připojovací centrum.

```
mount -o rsize=1024,wsiz=1024 172.16.0.4:/pub/htmldocs\  
/mnt/MYDOM_UNIX
```

`Rsize` a `wsiz` mohou být v závislosti na vašem prostředí změněny.

Jestliže máte nějaké problémy, přečtěte si dokument k NFS, který je k dispozici na <http://sunsite.unc.edu/mdw/HOWTO/NFS-HOWTO.html>.

6 Přístup na Web

Nyní, když máme nastavený HTTP-server, klienty a propojili jsme linuxový server s ostatními servery, musíme na linuxovém serveru provést některé menší úpravy, aby bylo možné přistupovat k těmto systémům souborů i z WWW-prohlížeče.

6.1 Přístup k připojeným systémům souborů

Při přístupu k připojeným adresářům na vaší stránce HTML máte dvě možnosti:

Vytvořit odkaz (link) na DocumentRoot (/usr/local/etc/httpd/htdocs), aby se odkazoval na připojený adresář jako

```
ln -s /mnt/MYDOM_NW/vol1/home/htmldocs netware
```

nebo

```
ln -s /mnt/MYDOM_NT winNT
```

nebo

```
ln -s /mnt/MYDOM_UNIX unix
```

Editovat ve vašem adresáři /usr/local/etc/httpd/conf soubor srm.conf a přidat nový alias.

```
# Přezdívka pro skutečné jméno
Alias /icons/ /usr/local/etc/httpd/icons/

# alias pro Netware server
Alias /netware/ /mnt/MYDOM_NW/vol1/home/htmldocs/
Alias /winNT/ /mnt/MYDOM_NT/
Alias /unix/ /mnt/MYDOM_UNIX
```

A znovu spustit váš server HTTPd. K dokumentům na serveru NetWare přistoupíte pomocí <http://linux.mydomain.netware/index.htm> podobně u ostatních.

6.2 Připojení k Internetu

Svůj Intranet můžete nakonec připojit k Internetu, čímž využijete e-mail a všechny ty báječné informace, co tam jsou. Stručnou poznámku k této problematice pravděpodobně přepíšete v některých dalších verzích dokumentu. Podrobný popis naleznete v dokumentech ISP Hookup na adrese <http://sunsite.unc.edu/mdw/HOWTO/ISP-Hookup-HOWTO.html> a Diald na adrese <http://sunsite.unc.edu/mdw/HOWTO/mini/Diald>.

6.3 Další využití

Server HTTP je možné využít na úřadu, aby poskytl přímý přístup k informacím na různých serverech, místech a adresářích. Data mohou být jednoduché dokumenty Wordu, tabulky Lotusu nebo složité databáze.

Aplikace této technologie jsou většinou následující:

- Vydávání dokumentů v organizaci. Tyto dokumenty mohou obsahovat vývěsky, roční zprávy, mapy, budovy v organizaci, ceníky, literaturu s informacemi o výrobcích a jakékoliv dokumenty, které mohou mít v rámci organizace nějakou hodnotu.
- Přístup k adresářům, ve kterých je možné vyhledávat. Rychlý přístup k telefonním seznamům v organizaci nebo podobným věcem. Tato data je možné zobrazit na serveru WWW nebo může server WWW (pomocí skriptu CGI) sloužit jako brána k připravovaným nebo novým aplikacím. To znamená, že využitím stejného standardního přístupového mechanismu může být informace lépe snadněji k dispozici. To znamená, že je možné pro generování informací v reálném čase vytvořit rozhraní s RDBMS, jako ORACLE, a SYBASE. Zde je seznam odkazů na takové servery WWW.
- Web Access - http://cscsun1.larc.nasa.gov/~beowulf/db/web_access.html/ - CGI gateways
- <http://www.w3.org/hypertext/WWW/RDBGate/Overview.html/>
- Stránka organizace/oddělení/osobní. Jak se mění v organizaci přístup směrem k větší samostatnosti oddělení, technologie Intranetu nabízí ideální médium pro rozšíření aktuálních informací oddělením i jednotlivcům. Mocné vyhledávací procedury nabízí služby lidem, kteří hledají skupinu nebo jednotlivce s odpovědí na běžné každodenní otázky, vznikající při chodu organizace.
- Jednoduché aplikace Groupware. S podporou HTML mohou servery poskytovat podpísané archy, průzkumy a jednoduché plánování.
- Distribuce softwaru. Administrátoři mohou Intranet využít k předávání software a aktualizací uživatelům v rámci organizace. K tomu slouží „Java“, umožňující vytvoření a přímé dodání požadovaných objektů, ne tedy jen dat a aplikací. To je více podporováno v nových verzích Linuxu se zabudovanou podporou Javy.
- Pošta. S přesunem k využívání intranetových poštovních produktů se standardními a jednoduchými metodami přiřazení dokumentu, zvuku, videa a dalších multimédií mezi jednotlivci, se pošta stává jednoduchou komunikační metodou. Pošta je brána zejména jako komunikace mezi jednotlivci nebo jednotlivcem a malou skupinou. K nastavení systému elektronické pošty je pro Linux k dispozici několik systémů, jako jsou `sendmail`, `pop3d`, `imapd`.
- Uživatelské rozhraní. Technologie Intranetu se vyvíjí tak rychle, že použité nástroje (zde HTML) je možné využít k podstatné změně naší komunikace se systémy. Pomocí HTML můžete vytvořit rozhraní, ovlivněné pouze představivostí tvůrce. Na technologiích Intranetu je pěkné to, že se snadno používají. Klepnutím na hyperlink se můžete v HTML přesunout na další stránku, spustit poplach, spustit několikaměsíční proceduru nebo cokoliv jiného, co zvládají počítačové programy.

7 Další možnosti

Následuje seznam dalších zajímavých věcí, které je možné s vaším linuxovým intranetovým serverem provádět. Veškerý níže zmíněný software spadá do kategorie free software nebo sharewaru.

- Prohlížení linuxového serveru pomocí Network Neighbourhood ve Win95/NT; Nastavte server NBT, podobný WINS. Projděte si stránku WWW SAMBA na <http://lake.canberra.edu.au/pub/samba/samba.html>.
- Implementace vyhledávacích procedur na vašem Intranetu. Připojte se k ht://Dig na <http://htdig.sdsu.edu/>.
- Využití CUSeeMe nastavením lokálního reflektoru viz domovské stránky v Cornellu na <http://cu-seeme.cornell.edu/>.
- Založení WWW-konferencí. Použijte COW z <http://thecity.sfsu.edu/COW/>.
- Založení databáze SQL. Viz domovská stránka mSQL na <http://Hughes.com.au/>.
- Nastavení FTP, Gopher, Finger, Bootp-serverů na serveru NetWare. Získáte je na <http://mft.ucs.ed.ac.uk/>.
- Emulace serveru NetWare. Vyzkoušejte NCP Utilities na internetové adrese <ftp://sunsite.unc.edu/pub/Linux/system/fileystems/ncpfs/>.

Jestliže naleznete další možnosti využití pro linuxový intranetový server, nebojte se mi svoje návrhy zaslat.

8 Autorské a právní záležitosti

8.1 Poděkování

Děkuji lidem z NCSA za to, že mi poskytli perfektní dokumentaci. Davidu Andersonovi a všem ostatním za to, že dokument přečetli a poslali své komentáře. Podrobnosti o NetWare/IP pochází od Romela Florese (rom@mnl.sequel.net).

8.2 Informace o autorských právech

Tento dokument má copyright (c) 1996, 1997 Pramod Karnad a rozšiřuje se podle následujících pravidel:

- Dokumenty projektu Linux HOWTO (jak na to) mohou být kopírovány a šířeny celé nebo i částečně, na jakémkoliv fyzickém i elektronickém médiu, ale tato poznámka o autorských právech zde musí vždy zůstat zachována. Komerční šíření je povoleno a vítáno; autor by ale na takové šíření byl rád upozorněn.
- Všechny překlady, odvozené práce nebo výtahy dokumentů Linux HOWTO musí být prováděny s dodržением této poznámky o autorských právech. To znamená, že nemůžete vytvářet od těchto dokumentů odvozenou práci a uvalit na ni dodatečná omezení šíření. Výjimky z tohoto pravidla mohou být poskytnuty za určitých podmínek; zde prosím kontaktujte koordinátora celého projektu na níže uvedené adrese.
- Jestliže máte dotazy, kontaktujte prosím na adrese **gregh@sunsite.unc.edu** Grega Hankinse, koordinátora pro Linux HOWTO. Telefonní číslo a adresy získáte pomocí `finger`.

Elektronická pošta a Linux

Guyllhem Aznar <guyllhem@danmark.linux.eu.org> verze 2.0, leden 1998

Tento dokument popisuje nastavení, údržbu a provoz elektronické pošty pod operačním systémem Linux. Tento dokument byste si měli pročíst v případě, že budete lokálně nebo přes síť posílat elektronickou poštu. Pokud elektronickou poštu nepoužíváte (ať už mezi uživateli jednoho systému nebo mezi uživateli různých systémů), dokument si číst **nemusíte**.

1 Úvod, autorská práva a zodpovědnost za správnost

1.1 E-mail

Jestliže se kdekoliv v elektronické adrese vyskytne řetězec „at“, nahraďte jej znakem „@“.

Pro lidi je to jednoduché, ale ne pro roboty prohledávající WWW; proto je to dostatečná ochrana před nevyžádanou poštou pro přispěvatele tohoto dokumentu.

1.2 Zaměření

Účelem tohoto dokumentu je nalézt odpovědi na některé často se vyskytující otázky a komentáře (FAQ) obecně k aplikacím elektronické pošty pro Linux a specificky k verzím pro distribuce Linux Debian a RedHat.

1.3 Nové verze

Nové verze tohoto dokumentu budou pravidelně posílány do diskusních skupin **comp.os.linux.announce**, **comp.answers** a **mail.answers**. Objeví se také na různých anonymních ftp-serverech, které archivují takový typ informací.

Kromě toho by mělo být vždy možné nalézt tento dokument na domovské stránce WWW pro Linux, která je na <http://sunsite.unc.edu/mdw/linux.html>.

1.4. Ohlasy

Zajímají mě jakékoliv vaše ohlasy (elektronickou poštou), týkající se tohoto dokumentu, kladné i záporné. Určitě mi dejte vědět, pokud objevíte chyby nebo zjevné překlepy.

Všechny dopisy obdržené elektronickou poštou, si přečtu, ale ne na všechny odpovídám. Žádosti budou zvažovány a vyřizovány v závislosti na mém volném čase, na stupni žádosti a na mém krevním tlaku :-).

Co se mi nebude líbit, to skončí v `/dev/null`, takže můžete být klidní.

Ohlasy na formát dokumentu by měly být odesílány koordinátorovi projektu HOWTO: Gregu Hankinsovi (**gregh at sunsite.unc.edu**).

1.5 Autorská práva

Vlastníkem autorských práv k dokumentu Mail-HOWTO je (c) 1998 Gylhem Aznar. Distribuce se provádí v licenci LDP. Všechny dotazy prosím směřujte na koordinátora Linux HOWTO - Grega Hankinse (**gregh at sunsite.unc.edu**).

1.6. Omezení záruky

Samozřejmě se distancuji od jakékoliv zodpovědnosti za obsah tohoto dokumentu. Použití návodů, příkladů a jiného obsahu dokumentu provádíte zcela na vlastní riziko.

2 Další zdroje informací

2.1 USENET

Na konfiguraci a provozu aplikací elektronické pošty pro Linux není nic „specifického“ (jednou provždy). Vzhledem k tomu **NEPOSÍLEJTE** dotazy, týkající se využití elektronické pošty, do skupin **comp.os.linux**.*

Do hierarchie **comp.os.linux** posílejte pouze dotazy, specifické pro Linux, jako jsou: „Se kterými volbami byl kompilován Debian 1.2 sendmail?“ nebo „RedHat 5.0 smail se po spuštění zhroutí.“

Dovolte mi to ještě zopakovat.

Pro posílání čehokoliv, co se týká pošty, do hierarchie **comp.os.linux** již není prakticky žádný důvod. Na **VŠECHNY** vaše otázky by měly stačit skupiny v hierarchii comp.mail.*

*JESTLIŽE POSÍLÁTE NA COMP.OS.LINUX. * OTÁZKY, KTERÉ SE NETÝKAJÍ KONKRÉTNĚ LINUXU, HLEDÁTE POMOC NA ŠPATNÉM MÍSTĚ. ODBORNÍCI NA ELEKTRONICKOU POŠTU VYŘIZUJÍ DOTAZY NA MÍSTECH, ZMÍNĚNÝCH VÝŠE, PŘIČEMŽ NEMUSÍ NUTNĚ POUŽÍVAT LINUX.*

POSÍLÁNÍM OTÁZEK, KTERÉ SE NETÝKAJÍ KONKRÉTNĚ LINUXU, DO LINUXOVÉ HIERARCHIE JEDINĚ MARNÍTE SVŮJ A CIZÍ ČAS, NAVÍC SE TÍM PRODLOUŽÍ VAŠE ČEKÁNÍ NA ODPOVĚĎ.

PATŘIČNÁ MÍSTA jsou:

comp.mail.elm	poštovní systém ELM
comp.mail.mh	system Rand Message Handling
comp.mail.mime	Multipurpose Internet Mail Extensions
comp.mail.misc	všeobecná diskuze o počítačové poště
comp.mail.multi-	media Multimedia Mail
comp.mail.mush	Mail User's Shell (MUSH)
comp.mail.sendmail	BSD sendmail
comp.mail.smail	smail
comp.mail.uucp	pošta v prostředí uucp

2.2 Emailové konference

Pro sendmail, smail a qmail existuje velké množství e-mailové konference.

Adresy je možné nalézt v `/usr/doc/tenkterýjsteszvolil`.

2.3. Další dokumenty z LDP

V jiných dokumentech Linux HOWTO a v projektu Linux DOC je k dispozici množství zajímavého materiálu.

Konkrétně se můžete podívat na následující:

- na vašem vlastním počítači je adresář `/usr/doc/sgml` nebo `/usr/doc/sendmail` :-)
- Příručka správce sítě
- Serial Communications HOWTO (komunikace po sériové lince)
- Ethernet HOWTO
- UUCP HOWTO, jestliže používáte UUCP

2.4 Knihy

Následuje sada knih, které by vám mohly být užitečné:

- „**Managing UUCP and USENET**“ od O'Reilly and Associates je podle mého názoru nejlepší knihou o programech a protokolech, používaných při provozu serveru pro USENET.
- „**Unix Communications**“ od The Waite Group obsahuje perfektní popis všech součástí a způsobu jejich spojení.
- „**Sendmail**“ od O'Reilly and Associates je nejlepší příručkou pro sendmail-v8 a sendmail+IDA. Nutná koupě pro každého, kdo chce využívat sendmail a nehodlá se učit až z vlastních chyb při jeho provozu.
- „**The Internet Complete Reference**“ od Osborne je dobrou příručkou, vysvětlující různé služby, dostupné na Internetu, a současně je skvělým zdrojem informací o různých zdrojích Internetu, jako jsou news, a elektronická pošta.
- „**Příručka správce sítě**“ od Olafa Kircha z Linux Documentation Project je k dispozici na síti a určitě ji vydává O'Reilly a SSC.

Je dobré si zjistit vše, co vlastně o síťovém používání systému Unix potřebujete vědět.

3 Požadavky

3.1 Hardware

Pro používání pošty v Linuxu neexistují žádné specifické požadavky.

Budete potřebovat nějaký druh „transportních“ programů, používaných pro připojení ke vzdáleným systémům, což zde znamená buď TCP/IP, nebo UUCP.

To znamená, že v závislosti na nastavení budete potřebovat modem nebo ethernetovou kartu.

Ve většině případů je vhodné mít nejrychlejší dostupný modem, dnes se jedná o modem s přenosovou rychlostí 57 600 bps. Obecně je vhodné mít na sériovém portu nebo přímo v modemu 16550 UART, aby bylo možné zvládnout rychlosti nad 9 600 baudů.

Jestliže nechápete smysl poslední věty, informace si zjistíte ve skupině **comp.dcom.modems** nebo využijte USENET a jeho FAQ a periodicky zasílané informace, týkající se komunikace pomocí modemu a sériové komunikace.

3.2 Software

Problém zní: K čemu bude váš poštovní software sloužit?

1. Počet hostitelů

Využíváte více než 100 hostitelů se složitými volbami pro názvy domén? *Zvolte sendmail!*

Využíváte méně než 100 hostitelů, přičemž s názvy domén se moc nezatěžujete? *Zvolte smail!*

2. Zabezpečení

Využíváte více nebo méně než 100 hostitelů, ale s vysokým zabezpečením? *Zvolte qmail!*

Využíváte méně než 100 hostitelů, ale se standardní úrovní zabezpečení? *Zvolte smail!*

3. Různé způsoby získávání pošty

Poštu získáváte a odesíláte přes UUCP nebo FIDO (pomocí ifmail)? Poštu získáváte a odesíláte pomocí POP a internetového SMTP? *Zvolte smail!*

Samozřejmě, že ve volbě poštovního software záleží rozhodnutí na vás. Předchozí informace vám mohou toto rozhodnutí usnadnit. *Sendmail* je vhodný pro větší množství hostitelů se složitými volbami, *qmail* zajišťuje vysoké zabezpečení; mezi *sendmailem* a *qmailem* leží kompromis, *smail*. Jestliže víte co děláte, zvolte *sendmail* (pak asi nebudete číst tento dokument); obecně však doporučuji *smail*.

4 Smail verze 3.1

Smail 3.1 je pro hostitele UUCP a pro některé hostitele SMTP vlastně téměř standardním přenosovým agentem. Snadno se konfiguruje, kompiluje se bez připojení zdrojů a jeho zabezpečení je dostatečné.

4.1 Konfigurování smailu

Nainstalujte si binární soubor pro `smail` z vaší distribuce Linuxu (doporučuji) nebo si získajte zdroje pro `smail` a vytvořte jej. Jestliže `smail` vytváříte ze zdrojových textů, musíte mít ve vašem souboru `os/linux` následující řádek (aby „`sed`“ dával scripty příkazového interpretu, které správně fungují).

```
CASE_NO_NEWLINES=true
```

Po nainstalování se do `/etc/smail` zapíše konfigurační soubory (pokud používáte starší verze zdroje Linuxu, situace se může lišit). Můžeme začít s jejich editací!

4.2 Soubor config

```
# Odkud
smart_path=polux
smart_transport=uux
# Na
hostname=danmark
domains=linux.eu.org
visible_name=danmark.linux.eu.org
uucp_name=danmark.linux.eu.org
# max_message_size=512k
# auth_domains=foo.bar
# more_hostnames=barberouge:barberouge.polux.freenix.fr
```

Takže zaprvé, kdo vám poštu doručuje? U mě je to „`polux`“ přes UUCP (transport pomocí `uux`); tento soubor si přirozeně změňte podle vlastní situace. Poštu vám může doručovat například „`bargw.bar.foobar.com`“ přes „`smtp`“, přičemž zde nepotřebujete transportní soubor, což definujete pomocí „`-transport_file`“.

Můžete také použít „`postmaster_address = vaše_jméno`“, pomocí „`visible_name`“ skryt síťovou topologii v adresách pro odesílání (jestliže jste brána) a pomocí „`more_hostnames`“ nastavit, které přezdívky mají být využity i pro doručenu poštu.

Více podrobností naleznete v dokumentaci pro `smail` nebo si prohlédněte příklady z `/usr/doc/smail/examples`, jestli vám náhodou některý nevyhovuje.

4.3 Soubor directors

```
# aliasinclude - expandovat ":include:filename" adresy, vytvořené
# aliasovými soubory. Tento a následující údaj je v podstatě
# dost běžný.
# K provedení větších úprav je jen málo důvodů, snad jen úprava
# adres ve formě:
#     :include:pathname
# které se mohou objevit v aliasových souborech nebo poštovních
# seznamech/předávacích souborech.
aliasinclude:
    driver = aliasinclude, # použijte tento ovladač speciálních znaků
    nobody;                # přiřazení uživatele nobody (nikdo)
                           # v případě narušení přístupových práv
    copysecure,           # získání práv z alias directoru
    copyowners,           # získání vlastníků z alias directoru
# forwardinclude - expandovat ":include:filename" adresy, vytvořené
# předávacími soubory
forwardinclude:
    driver = forwardinclude, # použijte tento ovladač speciálních znaků
    nobody;
    copysecure,            # získání práv z předávacího directoru
    copyowners,           # získání vlastníků z předávacího
                           # directoru

# aliases - vyhledání aliasových expanzí, uložených v databázi.
# Jedná se o standardní soubor aliases. Používá se pro běžné věci,
# jako je mapování roota, postmastera, MAILER-DAEMONa a uucp
# při administraci sítě, vytváření některých aliasových expanzí
# menších systémů a podobně. V konfiguraci této lokace je soubor
# aliases využit zejména pro aliasové a předávací informace,
# specifické pro jednotlivé stroje. Celkové předávací informace
# se vkládají do databáze "forward".
```

aliases:

```
driver=aliasfile,      # víceúčelový aliasový director
-nobody,               # všechny adresy jsou s nobody spojeny
                      # implicitně, takže toto nastavení není
                      # užitečné

sender_okay,          # neodstraňujte sender (odesilatele)
                      # z výrazů

owner=owner-$user;    # problémy se směřují na vlastníkovu adresu
file=/etc/aliases,
modemask=002,         # nemělo by to být globálně zapisovatelné
optional,             # ignoruj, pokud soubor neexistuje
proto=lsearch,        # neseříděný ASCII-soubor
```

```
# forward - vyhledání expanzí, uložených v předávací databázi.
# Jedná se o uživatelskou subdoménovou předávací databázi. Údaje jsou
# udržovány pro aktuální nebo minulé uživatele, kterým se pošta
# předává na preferované místo určení. Předávací databáze se
# po provedení změn dodává po síti TCP/IP, aby byla síť jednotná.
```

```
# forward:
```

```
# driver = aliasfile,      # víceúčelový aliasový director
# -nobody,                 # všechny adresy jsou s nobody spojeny
                          # implicitně, takže toto nastavení není
                          # užitečné

# owner = real-$user;     # problémy se směřují na vlastníkovu adresu
# file = /etc/forward,
# modemask = 002,
# proto = dbm,             # k přístupu využijte knihovnu dbm(3X)
```

```
# dotforward - expanduje soubory .forward v domovských adresářích
# uživatelů
# Pro uživatele, kteří mají údaj v databázi "forward", se soubor
# ".forward" využije pouze tehdy, pokud je na "domovském" stroji,
# definovaném v předávací databázi. Je-li použit, bere se spíše jako
# seznam adres, na které má být pošta doručena, než jako (nebo navíc
# jako) uživatel, určený v lokální adrese.
```

```
dotforward:
```

```
driver = forwardfile,    # víceúčelový předávací director
owner = postmaster, nobody, sender_okay;
```

```

file = ~/.forward,      # soubor .forward v domovských adresářích
checkowner,            # uživatel může tento soubor vlastnit
owners = root,         # nebo root může tento soubor vlastnit
modemask = 002,        # nemělo by to být globálně zapisovatelné
caution = daemon:root, # nespouštět nic jako root nebo daemon
# věnovat zvláštní pozornost na vzdáleně přístupné domovské
# adresáře
unsecure = "~uucp:/tmp:/usr/tmp:/var/tmp"

# forwardto - expanduje a "Forward to " (předat na) z uživatelských
# souborů v mailboxu
# Emuluje předávací mechanismus V6/V7/System-V, který využívá řádek
# předávacích adres, uložený na začátku uživatelských souborů
# mailboxu, s předponou "Forward to".
forwardto:
    driver = forwardfile,
    owner = postmaster, nobody, sender_okay;
    file = /var/spool/mail/${lc:user},      # ukazuje na uživatelské
                                           # soubory mailboxu

    forwardto,          # umožňuje funkci "Forward to "
    checkowner,         # uživatel může tento soubor vlastnit
    owners = root,      # nebo root může tento soubor vlastnit
    modemask = 0002,    # pod System V může zapisovat group mail
    caution = daemon:root, # nespouštět nic jako root nebo daemon
# user - odevzdá uživatelům na jejich lokálních strojích do mailboxu
user:    driver = user;    # ovladač pro zjištění uživatelského jména
        transport = local    # lokální transport směřuje do mailboxů

# real_user - nalezne uživatelská jména s předponou "real-"
# Toto je užitečné při umožnění adres, které explicitně dodávají
# do uživatelova souboru mailboxu. Sem mohou být dodány například
# chyby při expanzi souboru .forward nebo tak mohou být rozřešeny
# předávací smyčky mezi více stroji. Uživatelé mohou poštou také
# na svůj "nedomovský" stroj předávat data, přičemž využijí adresu
# ve tvaru real-login@vzdálený.hostitel.
real_user:
    driver = user;
    transport = local,

```

```
prefix = "real-"          # odpovídá například real-root

# lists - expanduje poštovní seznamy, uložené v adresáři list
# poštovní seznamy mohou být vytvořeny jednoduchým vytvořením
# souboru v adresáři /etc/maillists/.
lists: driver = forwardfile,
  caution,                # všechny adresy označit upozorněním
  nobody,                 # a potom přiřadit uživatele nobody
  owner = owner- $\$$ user;    # lokace se system V mohou využít o- $\$$ user
                          # - owner- $\$$ user by bylo moc dlouhé
  file = lists/ $\{\$ lc:user $\}$  # lists je pod  $\$$ maillib_dir

# owners - expanduje poštovní seznamy, uložené v adresáři list owner
# seznamy vlastníků mohou být vytvořeny vytvořením souboru
# v adresáři/etc/maillists/owner. Vlastníkům seznamů jsou posílány
# lokálně vzniklé chyby při práci se seznamy stejného názvu. Seznam
# vlastníků pro poštovní seznam vytvoříte souborem s názvem seznamu
# v /etc/maillists/owner. Tak se vytvoří seznam adres vlastníků-
# názvů seznamů, jak jej výše používá director "lists".
owners: driver = forwardfile,
  caution,                # všechny adresy označit upozorněním
  nobody,                 # a potom přiřadit uživatele nobody
  owner = postmaster;     # lokace se system V mohou využít o- $\$$ user
                          # - owner- $\$$ user by bylo moc dlouhé

  prefix = "owner-",
  file = lists/owner/ $\{\$ lc:user $\}$  # lists je pod  $\$$ maillib_dir

# request - expanduje poštovní seznamy, uložené v adresáři list
# request seznamy požadavků mohou být vytvořeny vznikem souboru
# v adresáři /etc/maillists/request. Zde vypsané adresy se
# využívají jako standardní adresy pro dotazy k poštovním seznamům.
# Například žádosti o přidání nebo smazání ze seznamu budou odesílány
# na "list-request", což je nutné nastavit na předání k příslušným
# osobám.
request: driver = forwardfile,
  caution,                # všechny adresy označit upozorněním
  nobody,                 # a potom přiřadit uživatele nobody
  owner = postmaster;     # lokace se system V mohou využít o- $\$$ user
```

```

# - owner-$user by bylo moc dlouhé
suffix = "-request",
file = lists/request/${lc:user} # lists je pod $smail_lib_dir

```

Tady nemusíte nic měnit, snad jen volbu pro poštovní konference (mailing list), jestliže ji v smailu budete využívat, nebo volbu forwards, jestliže budete chtít například znemožnit přeposílání pošty.

4.4 Soubor fidopaths

```

.f105.n324.z2.fidonet.org f105.n324.z2.fidonet.org!%s
.n324.z2.fidonet.org f105.n324.z2.fidonet.org!%s
.z2.fidonet.org f105.n324.z2.fidonet.org!%s
.fidonet.org f105.n324.z2.fidonet.org!%s

```

Takový soubor vytvořte pouze pokud používáte ifmail a FIDO.

4.5 Soubor routers

```

# forces - nastavit určité cesty
# Tato databáze existuje z důvodu nastavení cest na různé stroje
# nebo domény. Využívá se při vytváření dočasných převodů k dalším
# databázím cesty. Změny zde provedete editací souboru
# maps/force.path a zadáním make v adresáři maps/.
forces:
    driver = pathalias,                # směrovač, prohledávající
                                       # soubor paths
    method = /etc/smail/maps/table; # přenosy jsou v tomto souboru
    file = forcepaths,                # soubor, obsahující force path
                                       # info
    proto = lsearch,                  # použití setříděného souboru
    optional,
    reopen                             # zavřít, když se nepoužívá

uucp_neighbors:
    driver=uuname,                    # použít program, který vrací
                                       # sousedy
    transport=uux;

```

```

    cmd="/usr/bin/uuname -a",      # použít program uuname
# domain=uucp                    # oddělit koncovku .uucp

# smart_host - částečně určený director smarthost
# Jestliže je atribut smart_path v souboru config definován jako
# cesta z lokálního stroje na vzdálený, budou všechny stroje, které
# neodpovídají ničemu jinému, odeslány na vyznačený vzdálený stroj.
# Atribut smart_transport je možné použít k určení jiného
# transportu. Jestliže není určen atribut smart_path, tento router
# je ignorován.
smart_host:
    driver = smarthost,          # ovladač speciálních znaků
    transport = uux             # implicitně dodávat přes UUCP
# path=phreak

# ifmail - posílání pošty na fidonet a obráceně
ifmail:
    driver=pathalias,
    transport=ifmail;
    file=fidopaths,
    proto=lsearch

```

Pokud pro poštu z FIDO používáte ifmail, měli byste sem zahrnout část ifmail. Pověsimněte si, že je možné změnit transportní mód z „uux“ (přes UUCP) například na „SMTP“ nebo je dokonce možné vypsát cesty k různým strojům nebo doménám podle „/etc/maill/maill/maps/table“.

4.6 Soubor transports

```

# local - dodání pošty lokálním uživatelům
# Smail se má připojit přímo na soubory uživatelských mailboxů
# v /var/spool/mail
#local: driver = appendfile,    # připojit zprávu do souboru
# -return_path,                # začlenit pole Return-Path:
# local,                        # při předání využít lokální formy
# from,                          # dodat řádek From_
# unix_from_hack;              # v těle vložit > před From
#
# file = /var/spool/mail/${lc:user}, # použít tuto lokaci pro Linux

```



```
#                               # Všimněte si, že mail spool musí být 1777
# file = ~/mailfile,           # toto umístění zajišťuje lepší zabezpečení
# group = mail,                 # skupina, vlastnící soubor pro System V

# mode = 0660,                 # pod System V má group mail přístup
# suffix = "\n",               # připojení nového řádku navíc
# append_as_user,

# Takto může mít každý uživatel soubor ~/.procmailrc, kterým
# kontroluje filtrování pošty a její ukládání z poštovních seznamů
# v oddělených mailboxech (podle potřeby).
local: +inet,
    -uucp,
    driver = pipe,              # připojit zprávu do souboru
    return_path,               # začlenit pole Return-Path:
    local,                     # při předání využít lokální formy
    from,                      # dodat řádek From_
    unix_from_hack;           # v těle vložit > před From

    cmd = "/usr/bin/procmail", # použít procmail k lokálnímu
                                # předání
    parent_env,                # info o prostředí z nadřazeného
                                # adresáře
    pipe_as_user,              # použít user-id, asociovaný
                                # s adresou
    umask = 0022,              # umask pro dceřiný proces
# -ignore_status,             # status exit by měl být akceptován
# -ignore_write_errors,       # navazovat na broken pipes

# pipe - dodání pošty příkazům shellu
# To se využívá implicitně, když smail zjistí adresy, začínající
# svíslou čarou, jako "|/usr/lib/news/recnews talk.bizarre".
# Svíslá čára se z adresy odstraňuje před předáním k transportu.
# pipe: driver = pipe,         # předání zprávy jinému programu
# return_path, local, from, unix_from_hack;
#
# cmd = "/bin/sh -c $user",    # odeslání adresy do Bourne Shell
```

```
# parent_env,                # info o prostředí z nadřazeného
                             # adresáře
# pipe_as_user,              # použít user-id, asociovaný s adresou
# umask = 0022,              # umask pro dceřiný proces
# -log_output,               # nelogovat stdout/stderr
# ignore_status,             # na status exit se nehledí
# ignore_write_errors,       # ignorování broken pipes

# file - dodání pošty souborům
# To se využívá implicitně, když smail zjistí adresy, začínající
# šikmou čarou nebo tildou, jako "/usr/info/list_messages" nebo
# "~/Mail/inbox".
# file: driver = appendfile,
# return_path, local, from, unix_from_hack;
#
# file = $user,               # soubor se vezme z adresy
# append_as_user,            # použít user-id, asociovaný s adresou
# expand_user,                # expandovat ~ a $ v adrese
# check_path,
# suffix = "\n",
# mode = 0644

# uux - předání programu rmail na vzdáleném hostiteli UUCP
#
# Při jednom UUCP-převodu bude na vzdálenou lokaci předáno právě
# 5 adres.

uux:    driver = pipe,
        -uucp,
        inet,
# uucp,                # použít adresové formuláře ve stylu UUCP
        from,          # dodat řádek From_
        max_addrs = 5, # maximálně 5 adres na jedno vyvolání
        max_chars = 200; # maximálně 200 znaků adres

# přepínač -r zabraňuje okamžitému dodání, závorky kolem proměnné
# $user zabraňují speciální interpretaci v uux.
cmd = "/usr/bin/uux - -r -g$grade $host!rmail ${({strip:user}}$)",
```

```

# cmd="/usr/bin/uux - $host!rmail $($user)$",
  ignore_write_errors,      # ignorování broken pipes
  umask = 0022,
# pipe_as_sender,

# uux_one_addr - dodání pošty přes UUCP na vzdálenou lokaci, která
# zvládá v jednom okamžiku jednu adresu
# Toto je často nutné při dodávání na lokaci s neupravenou verzí
# 4.1BSD.

uux_one_addr:
  driver = pipe,
  uucp,                      # použít adresové formuláře ve stylu UUCP
  from;                      # dodat řádek From_
# přepínač -r zabraňuje okamžitému dodání
  cmd = "/usr/bin/uux - -r -g$grade $host!rmail (${strip:user})",
  umask = 0022,
  pipe_as_sender

queueonly:
  driver = pipe;            # odeslat zprávu na rouru
  cmd = "/usr/lib/sendmail -Q -f $sender -bm $user",
                                # použít getmail pro lokální předání
  user=root,                # spustit getmail jako "root"
  group=mail,               # spustit getmail jako "mail"
  parent_env,               # info o prostředí z nadřazeného adresáře
  -pipe_as_user,           # použít user-id, asociovaný s adresou
  umask = 0007,            # umask pro dceřiný proces

# dodání zprávy. smtp transport se začlení, pouze pokud existuje síť
# BSD. Pro přenosy v zóně UUCP je možné určit atribut uucp. Při
# přenosech na Internetu je nutné nastavit atribut inet.
# POZNÁMKA: Toto je optimální, ke zvládnutí více zpráv v jednom
# připojení by měla existovat nastavba.
# TAKÉ: Možná bude nutné omezit max_addrs na 100, protože toto je
# počet, pro který SMTP požaduje implementaci k obsluze jedné zprávy.
smtp:  driver=tcpsmtp,
      inet,                  # jestliže UUCP_ZONE není definováno

```

```
# uucp,                                # jestliže UUCP_ZONE je definováno
  -max_addrs, -max_chars; # žádné omezení v počtu adres

short_timeout=5m,                       # timeout pro krátké operace
long_timeout=2h,                         # timeout pro delší operace smtp
service=smtp,                            # připojení na tento servisní port
# Při použití na Internetu: zrušte komentář u následujících 4 řádků
use_bind,                                 # rozlišení MX a multiple A záznamů
defnames,                                # použití standardního vyhledávání domén
defer_no_connect,                       # nový pokus, je-li nameserver vypnutý
local_mx_okay,                          # selhat MX na lokálního hostitele
```

ifmail:

```
from,received,max_addrs=5,max_chars=200,
driver=pipe;
pipe_as_sender,
cmd="/usr/local/bin/ifmail -x9 -r$host ${strip:user})$)"
```

Pokud pro poštu z FIDO používáte ifmail, měli byste sem zahrnout část ifmail. Kromě toho byste neměli v tomto souboru editovat nic, co definuje agenty pro transport (uux, smtp...). Tyto parametry určíte v jiných konfiguračních souborech.

Povšimněte si, že některé části (jako „pipe“ nebo „file“) jsem dal do komentáře, aby se rozšířilo zabezpečení.

4.7 Adresář maps/

Obsahuje soubory map a table:

Nejprve soubor map

```
#N      foo.bar foo2.bar2
#S      AT 486/RedHat Linux 1.2.13
#O      organizace
#C      kontakt
#E      administrace (email)
#T      telefon
#P      adresa
#R
```

```

#U          hostitelé, připojení přes uucp
#W          vytvořeno/editováno kým
#
hname polux

hname linux.eu.org

hname = polux
hname = polux.linux.eu.org

```

Tento soubor si opět upravte podle vlastní situace (já poštu dostávám z polux.linux.eu.org). Nyní soubor `table`

```
* uux
```

Zde můžete určit různý způsob transportu do různých cílů. Například pro lokální síť určíte „smtp“ a pro zbytek světa „uux“ (přes UUCP) nebo obráceně (já využívám UUCP pro veškerou odesílanou poštu, proto jsem použil „*“).

4.8 Další dobré příklady

Předchozí soubory skutečně využívám, takže by neměly nastat problémy s jejich využitím jako základ pro vlastní soubory.

Následující soubory nabízím pouze jako příklady konfigurace `smailu` jiným způsobem.

```

#ident "@(#) transports,v 1.2 1990/10/24 05:20:46 tron Exp"

# Viz smail(5) s kompletním popisem obsahu tohoto souboru.

# local - předání pošty lokálním uživatelům
#
# Smail se má připojit přímo na soubory uživatelských mailboxů
# v /usr/mail
local: driver = appendfile,      # připojit zprávu do souboru
    return_path,                # začlenit pole Return-Path:
    local,                       # při předání využít lokální formy
    from,                         # dodat řádek From_
    unix_from_hack;              # v těle vložit > před From

```

```
file = /usr/mail/${lc:user},    # použít tuto lokaci pro System V
group = mail,                  # skupina, vlastníci soubor pro System V
mode = 0660,                  # pod System V má group mail přístup
suffix = "\n",                # připojení nového řádku navíc
append_as_user,

# pipe - dodání pošty příkazům shellu
#
# To se využívá implicitně, když smail zjistí adresy, začínající
# svislou čarou, jako "|/usr/lib/news/recnews talk.bizarre".
# Svislá čára se z adresy odstraňuje před předáním k transportu.
pipe: driver = pipe,          # předání zprávy jinému programu
      return_path, local, from, unix_from_hack;

cmd = "/bin/sh -c $user",     # odeslání adresy do Bourne Shell
parent_env,                   # info o prostředí z nadřazeného adresáře
pipe_as_user,                 # použít user-id, asociovaný s adresou
umask = 0022,                 # umask pro dceřiný proces
-log_output,                  # nelogovat stdout/stderr
ignore_status,                # na status exit se nehledí
ignore_write_errors,          # ignorování broken pipes

# file - dodání pošty souborům
#
# To se využívá implicitně, když smail zjistí adresy, začínající
# šikmou čarou nebo tildou, jako "/usr/info/list_messages" nebo
# "~/Mail/inbox".

file: driver = appendfile,
      return_path, local, from, unix_from_hack;

file = $user,                 # soubor se vezme z adresy
append_as_user,               # použít user-id, asociovaný s adresou
expand_user,                  # expandovat ~ a $ v adrese
check_path,
suffix = "\n",
mode = 0644
```

```
# uux - předání programu rmail na vzdálené UUCP lokaci
#
# Při jednom UUCP převodu bude na vzdálenou lokaci předáno právě
# 5 adres.
uux:    driver = pipe,
        uucp,                # použít adresové formuláře ve stylu UUCP
        from,                # dodat řádek From_
        max_addrs = 5,      # maximálně 5 adres na jedno vyvolání
        max_chars = 200;    # maximálně 200 znaků adres

# přepínač -r zabraňuje okamžitému dodání, závorky kolem proměnné
# $user zabraňují speciální interpretaci v uux.
cmd = "/usr/bin/uux - -r -g$grade $host!rmail ${({strip:user})$}",
umask = 0022,
pipe_as_sender,

# uux_one_addr - dodání pošty přes UUCP na vzdálenou lokaci, která
# zvládá v jednom okamžiku jednu adresu
#
# Toto je často nutné při dodávání na lokaci s neupravenou verzí
# 4.1BSD.
uux_one_addr:
    driver = pipe,
    uucp,                # použít adresové formuláře ve stylu UUCP
    from;                # dodat řádek From_

# přepínač -r zabraňuje okamžitému dodání
cmd = "/usr/bin/uux - -r -g$grade $host!rmail ${({strip:user})$}",
umask = 0022, pipe_as_sender

# demand - dodat vzdálenému programu rmail, který podává žádost
demand: driver = pipe,
        uucp, from, max_addrs = 5, max_chars = 200;

# bez přepínače -r se bude vzdálená lokace kontaktovat okamžitě
cmd = "/usr/bin/uux - -g$grade $host!rmail ${({user})$}",
umask = 0022, pipe_as_sender
```

```
# uusmtp - dodat programu rsmtplib na vzdálené hostitele UUCP
#
# Dodat pomocí jednoduchého protokolu Batched SMTP na vzdálený stroj.
# Umožňuje využít více libovolných adres. Odstraňuje také omezení
# adresátů na jedno vyvolání uux.
uusmtp: driver = pipe,
    bsmtplib,          # odeslat příkazy batched SMTP
    -max_addrs,       # není zde žádné omezení počtu nebo
    -max_chars;       # celkové velikosti adres adresátů

# přepínač -r zamezuje okamžitému dodání, adresáti jsou uloženi
# v datech, odesílaných na standardní výstup rsmtplib.
cmd = "/usr/bin/uux - -r -g$grade $host!rsmtplib",
umask = 0022, pipe_as_sender

# demand_uusmtp - dodat vzdálenému programu rsmtplib, který podává
# žádost
demand_uusmtp:
    driver = pipe,
    bsmtplib, -max_addrs, -max_chars;

# bez přepínače -r se bude vzdálený hostitel kontaktovat okamžitě
cmd = "/usr/bin/uux - -g$grade $host!rsmtplib",
umask = 0022, pipe_as_sender

# smtp - dodání pomocí SMTP přes TCP/IP
#
# Připojení ke vzdálenému hostiteli pomocí TCP/IP a inicializace
# SMTP-konverzace o dodání zprávy. SMTP transport je začleněn pouze
# s existující sítí BSD.

# POZNÁMKA: Možná bude nutné omezit max_addrs na 100, protože toto
# je počet, pro který SMTP požaduje implementaci k obsluze jedné
# zprávy.
smtp: driver = smtp,
    -max_addrs,
    -max_chars
```



```
#ident "@(#) table,v 1.2 1990/10/24 05:20:31 tron Exp"

# Tento soubor vyjmenovává transporty, které se při dodání na
# specifické lokace z bargw využijí.

#lokace      transport
#-----
curdsgw      demand_uusmtp      # pomocí batched SMTP
oldbsd       uux_one_addr        # lokace s 4.1BSD nemohou vzít
                                     # více než jednu adresu
sun          demand            # když odesíláte poštu,
                                     # volejte sun
*            uux                # u všeho ostatního volte intervaly
```

4.9 Restart inetd

Aby se `smail` spustil jako SMTP démon, přidejte do `/etc/inetd.conf`:

```
smtp stream tcp nowait root /usr/bin/smtpd smtpd
```

nebo:

```
smtp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.smtpd
```

Odesílaná pošta se bude při použití `elmu` odesílat automaticky.

4.10 Smail s SMTP

ISP většinou používají SMTP, takže s posíláním pošty byste neměli mít žádné problémy. Jestliže je vaše připojení na Internet v době odeslání pošty nefunkční, pošta se usadí do „`/var/spool/mailed/input`“. Při obnovení spojení se spouští „`runq`“, který poštu odešle. S obdržetím pošty je ale v takovém případě problém, protože váš poskytovatel Internetu nemá na starosti jen vás, ale i mnoho jiných klientů!

Vaši poštu je obvykle možné zachránit pomocí protokolu POP, viz níže.

5 Sendmail+IDA

Pro velké servery stojí za zvážení výběr sendmailu, už pro snadnost jeho použití v daném případě. Ale musíte si vybrat mezi verzemi sendmailu+IDA a sendmailu 8.x:

- Jestliže používáte staré jádro (1.0): sendmailu+IDA
- Předchozí jádro (1.2): sendmailu+IDA a editace zdrojového kódu
- Současné jádro (2.0): sendmailu 8.x

Ale linuxoví začátečníci nebo lidé, zaměřeni na zabezpečení nebo snadnost konfigurace, by si měli raději vybrat smail nebo qmail.

5.1 Instalace zdrojového textu

- `cd / ; tar -zxvf sendmail5.67b+IDA1.5.tgz`
- `cd /usr/local/lib/mail/CF` a okopírujte `sample.m4` `local.m4` na „jménovašehohostitele.m4“.

Provedte editaci dodaných jmen hostitelů, přezdivek, serverů a upravte je podle vaší situace. Implicitní soubor je pouze pro čistě UUCP-sít, kde jsou hlavičky v doménovém tvaru a používá se chytřejší hostitel. Potom make `jménovašehohostitele.cf` a výsledný soubor přesuňte do `/etc/sendmail.cf`.

- Jestliže využíváte čistě UUCP, **NEPOTŘEBUJETE** vytvářet žádné z tabulek, které jsou zmíněny v souboru `README.linux`

Aby fungoval Makefile, stačí na soubory použít touch. Editujte soubor `.m4`, vytvořte `sendmail.cf` a začněte jej testovat.

- Jestliže využíváte čistě UUCP a využíváte „chytřejší hostitele“, musíte pro každou lokaci přidat údaje `uucpstable` (jinak by pošta pro ně také chodila přes chytřejší hostitele) a proti upraveným `uucpstable` musíte spustit dbm.
- Jestliže spouštíte původní binární distribuci Riche Brauna 5.67a a změníte soubor `.cf` pomocí „`/usr/lib/sendmail -bz`“, musíte přerušit konfiguraci, aby se změny projevíly.

Také byste měli přejít na verzi alespoň 5.67b, protože ve verzích 5.67a a starších je nepěkná díra v zabezpečení. Další zajímavou věcí je to, že když máte nastaven `mail.debug` a spustíte `syslogd`, vaše doručená a odeslaná pošta bude zaznamenávána. Podrobnosti viz soubor „`/etc/syslog.conf`“.

Zdroje pro sendmail+IDA naleznete na `vixen.cso.uiuc.edu`; pokud například používáte jádro 1.00, nevyžadují žádné úpravy pro běh v Linuxu.

Jestliže používáte jádro > 1.1.50, budete si muset pohrát s obrácením linuxových specifikací, které naleznete ve zdrojových textech (já jsem vám to říkal, že `sendmail` je pouze pro „stará“ jádra.

Jak se to provede je jistě jasné: stačí zadat „make“, počkat si, podívat se na příslušné řádky zdrojových textů a dát do komentáře kód, specifický pro Linux.

Jestliže chcete používat `sendmail+IDA`, silně doporučuji až verzi `sendmail5.67b+IDA1.5`, protože všechny nutné linuxové odlišnosti jsou nyní ve zdrojových textech a bylo odstraněno několik bezpečnostních děr, které BYLY (!) ve starších verzích, vytvořených přibližně před 1. prosincem 1993.

Nyní je poslední verze jádra 2.0.x. Místo `sendmail+IDA` byste měli používat `sendmail 8.x`.

5.2 Soubor `sendmail.m4`

`Sendmail+IDA` požaduje spíše vytvoření souboru `sendmail.m4` místo přímé editace souboru `sendmail.cf`. Na tom je pěkné, že lze jednoduše nastavit konfigurace v `smailu` nebo normálním `sendmailu` prakticky nenastavitelné (pro některé lidi).

Soubor `sendmail.m4`, který odpovídá předchozímu příkladu pro `smail`, vypadá následovně:

```
dnl #----- UKÁZKOVÝ SOUBOR SENDMAIL.M4 -----
dnl #
dnl # (řetězec 'dnl' je m4 ekvivalentem řádku v komentáři)
dnl #
dnl # asi nebudete chtít přepsat LIBDIR z kompilovaných cest
dnl #define(LIBDIR,/usr/local/lib/mail)dnl      # kam jdou všechny
dnl                                           # podpůrné soubory
define(Local_mailer_DEF, mailers.linux)dnl   # poštovní program pro
dnl                                           # lokální dodávku
define(POSTMASTERBOUNCE)dnl                # chyby na adresu
                                           # postmaster
define(PSEUDODOMAINS, BITNET UUCP)dnl      # zde nezkoušejte DNS
dnl #
dnl #-----
dnl #
dnl # jména, pod kterými nás znají
define(PSEUDONYMS, myhostname.subdomain.domain myhostname.UUCP)
dnl #
```

```
dnl # naše primární jméno
define(HOSTNAME, myhostname.subdomain.domain)
dnl #
dnl # naše UUCP-jméno
define(UUCPNAME, myhostname)dnl
dnl #
dnl #-----
dnl #
define(UUCPNODES, |uname|sort|uniq)dnl # naši UUCP-sousedé
define(BANGIMPLIESUUCP)dnl           # zajistěte tohle UUCP
define(BANGONLYUUCP)dnl             # pošta je brána korektně
define(RELAY_HOST, my_uucp_neighbor)dnl # náš chytrý relay host
define(RELAY_MAILER, UUCP-A)dnl     # do moria přes UUCP
dnl #
dnl #-----
dnl #
dnl # různé vyhledávací tabulky dbm
dnl #
define(ALIASES, LIBDIR/aliases)dnl   # systémové aliasy
define(DOMAINTABLE, LIBDIR/domaintable)dnl # doménové lokace
define(PATHTABLE, LIBDIR/pathtable)dnl # databáze cest
define(GENERICFROM, LIBDIR/generics)dnl # obecné adresy "z"
define(MAILERTABLE, LIBDIR/mailertable)dnl # poštovní programy na
dnl                                     # hostitele nebo doménu
define(UUCPXTABLE, LIBDIR/uucphtable)dnl # cesty na hostitele,
dnl                                     # které zásobujeme
define(UUCPRELAYS, LIBDIR/uucprelays)dnl # cesty krátkých okruhů
dnl #
dnl #-----
dnl #
dnl # začlenění 'skutečného' kódu, který to vše zprovozní
dnl # (se zdrojovým kódem)
dnl #
include(Sendmail.mc)dnl              # POŽADOVANÝ ÚDAJ !!!
dnl #
dnl #----- KONEC UKÁZKOVÉHO SOUBORU SENDMAIL.M4 -----
```

5.3 Určení lokálního doručovatele

Na rozdíl od většiny verzí Unixu se Linux implicitně nedodává s lokálním doručovatelem. Nyní se většinou instalují `deliver` nebo `procmail`, takže tohle velice složité nastavení se již dále nekomplikuje. Doporučuji tedy využít běžně dostupné programy `deliver` a `procmail`, které se v některých distribucích Linuxu mohou vyskytovat jako volitelné komponenty.

Musíte pak v souboru `sendmail.m4` definovat `LOCAL_MAILER_DEF`, který bude ukazovat na takovýto soubor:

```
# -- /usr/local/lib/mail/mailers.linux --
#      (lokální doručovatelé pro použití v Linuxu )
Mlocal, P=/usr/bin/deliver, F=SlsmFDMP, S=10, R=25/10, A=deliver $u
Mprog,  P=/bin/sh,          F=lsDFMeuP,  S=10, R=10, A=sh -c $u
```

V souboru `Sendmail.mc` je také zabudovaná implicitní hodnota pro `deliver`, která se dostane i do souboru `sendmail.cf`. K jejímu určení nepoužijete soubor `mailers.linux`, ale v souboru `sendmail.m4` místo toho nadefinujete:

```
dnl --- (v souboru sendmail.m4) ---
define(LOCAL_MAILER_DEF, DELIVER)dnl      # doručovatel pro lokální
dnl                                         # dodávku
```

Naneštěstí `Sendmail.mc` předpokládá instalaci `deliver` v `/bin`, což není správné v případě `Slackware1.1.1` (instaluje jej do `/usr/bin`). V takovém případě si musíte buď vypomoci ošálením pomocí odkazu, nebo podle zdrojových textů `deliver` přeinstalovat do `/bin`. Povšimněte si, že `procmail` je ve většině případů lepší než `deliver`, například při filtrování pošty.

5.4 Tabulky sendmail+IDA dbm

Nastavení speciálního chování hostitelů nebo domén se provádí přes množství volitelných tabulek `dbm` místo přímé editace souboru `sendmail.cf`.

Podrobnější informace naleznete v červencovém čísle časopisu **Linux Journal** z roku 1994 (jestli ale naleznete ten časopis, v dokumentacích ve zdrojích nebo v kapitole o `sendmailu` v nejnovější verzi **Příručky správce sítě** v Linux DOC Project, která je součástí této knihy.

- `mailertable` - určuje zvláštní chování vzdálených hostitelů nebo domén
- `uucphtable` - nastavuje doručování pošty pomocí UUCP pro adresy ve formátu DNS

- `path`table - definuje UUCP bang-path na vzdálené hostitele nebo domény
- `uucprelays` - zkracuje cestu `pathalias` na známé vzdálené hostitele
- `genericfrom` - převádí vnitřní adresy na obecné, viditelné zvenku
- `xaliases` - převádí na/z platných vnitřních adres
- `decnetxtable` - převádí RFC-822 adresy na adresy ve stylu DECnet

5.5 Takže které údaje jsou opravdu nutné?

Jestliže nejsou použity žádné volitelné tabulky `dbm`, `sendmail` poštu doručuje pomocí `RELAY_HOST` a `RELAY_MAILER`, v závislosti na souboru `sendmail.m4`, použitého pro vytvoření `sendmail.cf`. Takové chování je možné snadno upravit pomocí údajů v `domaintable` nebo `uucpxtable`.

Normální hostitel, který je na Internetu a slyší na systém DNS, nebo je čistě UUCP a předává veškerou poštu pomocí UUCP přes `RELAY_HOST`, nepotřebuje žádné zvláštní tabulkové údaje.

Prakticky všechny systémy by měly nastavovat makra `DEFAULT_HOST` a `PSEUDONYMS`, která určují kanonické jméno hostitele a přezdívky, pod kterými je známa.

Pravděpodobně také nastavení `RELAY_MAILER` a `RELAY_HOST`, umožňující automatické směřování pomocí předávání pošty na chytřejšího hostitele.

Použitý přenos pošty je definován v `RELAY_MAILER` a pro UUCP hostitele by měl být obvykle nastaven na UUCP-A. Jestliže je váš hostitel čistě SMTP a rozumí DNS, měli byste změnit `RELAY_MAILER`.

Jestliže je váš hostitel SLIP, můžete využít jednoduchý způsob směřování veškeré odesílané pošty k vašemu poskytovateli služeb, který si s ní už poradí. K tomu je nutné definovat `ISOLATED_DOMAINS` a `VALIDATION_DOMAINS` na vaši doménu, `RELAY_HOST` musí být váš poskytovatel služeb a `RELAY_MAILER` bude TCP. K takovému nastavení systému pro převádění byste také samozřejmě měli vyžádat povolení.

5.6 Sendmail 8.x

`Sendmail 8.7.x` byl poslední větší inovací od dob `sendmail5`. Měl nádhernou zabudovanou podporu pro kompilaci v Linuxu: „`make linux`“ a všechno je nastavené.

Nejlépe vám pravděpodobně poslouží nějaká binární podoba programu, kterou si přetáhnete z některých linuxových archivních serverů. Je to lepší než se potýkat například s Berkeley `dbm`.

Na sunsite.unc.edu/pub/Linux/system/Mail/delivery/sendmail-8.6.12-bin.tgz naleznete skvělou distribuci pro sendmail 8.6.12 od Jasona Haara - **j.haar at lazer-jem.demon.co.uk**, která má dokumentaci ke zdrojovým textům a pěkný rychlý popis pro běh sendmailu v8 při běžných konfiguracích.

5.7 Ukázkový soubor mc pro 8.7.x

Podobně jako sendmail+IDA využívá sendmail v8 m4 pro převod konfiguračního souboru config na plný sendmail.cf, využívaný sendmailem. Následuje můj aktuální mc soubor, využívaný na mém hostiteli (PPP na Internet pro odesílanou poštu, UUCP pro doručenou poštu).

```
dnl divert(-1)
#-----
#
# tohle je soubor .mc pro linuxového hostitele, nastavenou následovně:
#
#   - připojená na Internet pro výchozí poštu (je zde ppp)
#   - připojená přes UUCP pro příchozí poštu
#   - doménové hlavičky
#   - žádný lokální doručovatel (místo toho 'deliver')
#   - žádný běžící DNS, takže nic výchozího tudy neprojde
#   - veškerá nelokální výchozí pošta jde na RELAY_HOST přes SMTP
# (používáme ppp a necháváme našeho poskytovatele, aby se staral)
#
#                               vds 3/31/95
#-----

include('../m4/cf.m4')
VERSIONID('linux nodns relays to slip service provider
smarthost')dnl
Cwmyhostname.myprimary.domain myhostname.UUCP localhost
OSTYPE(linux)
FEATURE(nodns)dnl
FEATURE(always_add_domain)dnl
FEATURE(redirect)
FEATURE(nocanonify)
dnl MAILER(local)dnl
```

```
MAILER(smtp) dn1
MAILER(uucp) dn1
define('RELAY_HOST', smtp:my.relay.host.domain)
define('SMART_HOST', smtp:my.relay.host.domain)
define('UUCP_RELAY', smtp:my.relay.host.domain)
define('LOCAL_MAILER_PATH', '/bin/deliver')
define('LOCAL_MAILER_ARGS', 'deliver $u')
```

5.8 Lahůdky v sendmailu v8

Existuje několik rozdílů, které předpokládám u uživatelů IDA. Zatím jsem se setkal s tímto: místo „runq“ pro spuštění fronty zadáváte „sendmail -q“.

5.9 Agenti pro lokální doručování

Na rozdíl od většiny operačních systémů neměl Linux poštu „přímo v sobě“ (built-in): potřebovali jste program, který by ji lokálně doručoval (například „lmail“, „procmail“ a „deliver“).

Nyní již každá nová distribuce obsahuje lokálního doručovatele!

Dokumentaci k jejich využití ve vaší distribuci naleznete v binární podobě `sendmail5.67b+IDA1.5` (na [sunsite](#)).

6 POP mail

Na síti pracovních stanic byla pošta vždy problémová:

- Buď využijete „**uživatel@počítač.foo.com**“ s problémy, když je „počítač“ vypnutý, s identifikací vaší sítě pro zbytek světa, s různými adresami pro jednu osobu u různých počítačů,
- nebo využijete poštovní přípojku na „**poštovniserver.foo.com**“ s přepisovacími právy, takže každý uživatel má poštu z jedné adresy, i když sedí u různých počítačů.

V takovém případě ale nastává problém, jak bude uživatel poštu číst? Pomocí `rsh` a `elm`? :-)
) Tím by se přetížila poštovní přípojka! Řešením by mohlo být přeposílání, UUCP, SMTP..., ale to by bylo příliš složité.

Pak přišli POP/IMAP (oba zpočátku s problémy v zabezpečení v nových verzích jsou problémy překonány použitím `ssh`): poštovní program někdy musí být nastaven lokálně (jako `sendmail`, `smail`, `qmail` při použití například `elmu`, ale u mozilly se tomu vyhnete). Odesílání a příjem pošty je pak ale jednodušší.

6.2 Příjem pošty

Zde se dostáváme k hlavnímu neduhu protokolu POP: heslo se přes síť posílá jako čistý text a někteří příjemci POP neznají: musíte zvolit takový program, který POP umí (například `pine`, `emacs`, `netscape`, `mutt`...).

Problém s heslem se odstraní vytvořením šifrovaného „kanálu“ pro POP nebo použitím rozšíření APOP nebo RPOP. Problém s příjmem je možné vyřešit buď změnou programu pro čtení pošty (`mozilla`, `emacs`, `pine`,) nebo použitím lokálního poštovního programu s podporou protokolu POP.

`Gwpop` se hodí k vytvoření šifrovaného „kanálu“ a vložení pošty přímo do poštovní přihrádky. Ale závisí na `perlu`...

Mohu také doporučit `fetchmail`, který je aktivně podporován.

Jinak je možné použít jeden z mnoha POP klientů, dostupných pro Linux. Pokud je vaše uživatelské jméno `john` a heslo `PrisneTajne`, spustíte:

```
$ popclient -3 -v mail.acme.net -u john -p "PrisneTajne"
-k -o JOHN-INET-MAIL
```

(Přesný význam předchozích voleb naleznete na manuálových stránkách programu `popclient`.)

6.3 Odeslání pošty

Zde musíte použít software, který umí SMTP, jako je `qmail`, `sendmail` nebo `mozilla` (tenhle umí všechno: čte poštu, obdrží POP a odešle přes SMTP!).

Vraťte se k jedné z předchozích částí, podle které nainstalujete a nakonfigurujete ten program, který vám vyhovuje. Jakmile začnete testovat, zkuste odeslat nějakou poštu na lokální účet na poštovní přípojce.

6.4 Čtení pošty

Jestliže váš program nezvládá všechno sám, můžete si nainstalovat `elm`, `pgp`, `mush`, `pine`... k dispozici je mnoho dobrých programů pro Linux!

6.5 Testování

To, že váš poštovní server zvládá POP, vyzkoušíte takto:

```
$ telnet mailhost 110
```

Jestliže to funguje, objeví se něco jako: „OK Pop server (...) starting“. Zadejte „quit“. Instalaci šifrovaného „kanálu“ `ssh` proveďte po testování vašeho poštovního serveru:

```
$ ssh mailhost date
```

Jestliže obdržíte datum, je vše v pořádku. Povšimněte si, že `ssh` se neptá na heslo, proto musíte na poštovním serveru vytvořit soubor „`shost`“, obsahující jméno klienta. Testování `ssh` přesměrování portů (které využívá `gwpop`) provedete takto:

```
$ ssh -n -f -L 12314:localhost:110 mailhost sleep 30
```

potom

```
$ telnet localhost 12314
```

Pak můžete očekávat objevení uvítací zprávy vzdáleného POP-serveru. Jestliže nepoužíváte `ssh`, nezapomeňte ve skriptu pro `gwpop` dát do komentáře `$ssh`. Zkoušku `procmail` provedete zadáním „`procmail -v`“.

6.6 Používání

Nyní můžete editovat skript `perlu` pro `gwpop`, zkontrolovat, jestli je vše v pořádku, a spustit `gwpop`:

```
$ gwpop -v vašeuživatelskéjméno
POP password on mailhost: vašeheslo
```

Jestliže jsou „chybová hlášení“ `gwpopu` normální, pošta bude z poštovní přípojky přehrána na váš lokální systém, kamkoliv určíte (tohle si prosím vyzkoušejte).

Gwpop je možné použít také jako démon:

```
$ gwpop -d $HOME/tmp vašeuživatelskéjméno
```

Zprávy gwpopu jsou potom odeslány do souboru `syslog` a gwpop bude stále běžet; signál „HUP“ přinutí gwpop vyzvednout vaši poštu.

POP-software je možné získat například zde:

```
ftp://ftp.pasteur.fr/pub/Network/gwpop
ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail
http://www.cs.hut.fi/ssh/
```

7 Poštovní „uživatelské agenty“

Tato část obsahuje informace, vztahující se k „uživatelským agentům“, což znamená software, který uživatel vidí a používá. Tento software spoléhá na transportní software, zmiňovaný výše. Nyní je k dispozici množství dalších „uživatelských agentů“ (`pine`, `mush`...), ale k těm neexistují žádné informace, specifické pro Linux. Dejte mi prosím vědět, jestli jsem na něco nezapomněl!

7.1 Elm

Elm se pod Linuxem kompiluje, instaluje a spouští bez problémů. Více informací naleznete ve zdrojových textech k elmu a v instalačních instrukcích. Elm a `filter` musí mít práva 2755 (skupin mail), `/var/spool/mail/775` a skupin mail.

Jestliže používáte binární distribuci, budete muset vytvořit soubor `„/usr/local/lib/elm/elm.rc“`, čímž přepíšete zakompilovaný název serveru a informace o doméně:

- nahraďte „`subdomain.domain`“ názvem vaší domény
- nahraďte „`myhostname`“ vaším názvem serveru (bez názvu domény)

```
#----- /usr/local/lib/elm/elm.rc -----
#
# toto je nekvalifikovaný název hostitele
hostname = myhostname
#
# toto je lokální doména
hostdomain = subdomain.domain
```

```
#
# toto je plně kvalifikovaný název hostitele
hostfullname = myhostname.subdomain.domain
#
#-----
```

Jednu věc si ale uvědomte, pokud máte totiž Elm kompilován s nastaveným MIME, musíte mít nainstalován a v cestě `metamail`, jinak by Elm nedokázal číst obdrženou MIME-poštu. Metamail je k dispozici na **thumper.bellcore.com** a samozřejmě také přes „archie“.

Do kategorie „příliš dobré, než aby to byla pravda“ spadá distribuce Elm-2.4.24, která umí PGP. Naleznete ji na adrese `ftp://ftp.viewlogic.com/pub/elm-2.4pl24pgp3.tar.gz`, což je elm2.4.24 s přidanou podporou PGP. Konfiguruje se a instaluje jako normální Elm, což pravděpodobně znamená přidání výše zmíněných dodatků. Za zmínku stojí, že já tuto verzi používám a jsem s ní spokojen. Samozřejmě, že k dispozici je jistě i mnoho novějších verzí, včetně elm-ME+.

Následující věc sice není specifická pro Linux, ale často je považována (neprávem) za chybu v Elmu. Slyšeli jsme, že Elm někdy padá a hlásí, že nemůže alokovat v paměti nějaký vysoký počet bajtů. Náprava je v odstranění zpracovaných globálních poštovních přezdivek (`aliases.dir` a `aliases.pag`).

TOHLE ALE NENÍ CHYBA ELMU, je to chyba v konfiguraci Elmu, kterou prováděl ten, od koho máte binární distribuci.

Elm má rozšířený a nekompatibilní formát přezdivek; musíte zajistit, aby cesta, kterou Elm pro přezdívky využívá, byla odlišná od cesty, kterou využívá `sendmail/smmail`. Vzhledem k množství zpráv o tomhle problému je zřejmé, že alespoň jedna z větších distribucí „z ulice“ byla špatně nakonfigurována (**scot at catzen.gun.de (Scot W. Stevenson)**).

Aktuální balík `metamailu` vyžaduje pro některé skripty `csh`. Nemáte-li `csh` (nebo `tcsh`), objeví se zajímavé chyby...

7.2 Mailx

Ušetřete si námahu: sežeňte si ze Slackware verze 2.1.0 nebo pozdější `mailx` kit, který obsahuje pěknou implementaci `mailx5.5`. Jestliže chcete kompilovat ze zdrojů, `mailx v5.5` se v Linuxu kompiluje bez dodatků, pokud máte nainstalován „**pmake**“. Jestli je to ještě aktuální, doporučuji náhradu starého „`edmailu`“ ze SLS1.00 za `mailx`.

7.3 Další uživatelské agenty

Tyto by pod Linuxem také měly běžet. Podrobnosti k jejich nalezení viz [archie...](#)

- Pine - z Univ. of Washington
- Metamail - umožňuje podporu MIME
- mh - další způsob zpracování pošty
- deliver - shromažďuje a zpracovává poštu podle zadaných pravidel
- procmail - shromažďuje a zpracovává poštu podle zadaných pravidel
- Majordomo - spravuje poštovní konference
- Mserv - poskytuje soubory poštou

8 Poděkování

Následující lidé přispěli svými informacemi a zkušenostmi, čímž napomohli dokončení tohoto dokumentu:

Steve Robbins, Ian Kluft, Rich Braun, Ian Jackson, Syd Weinstein, Ralf Sauther, Martin White, Matt Welsh, Ralph Sims, Phil Hughes, Scot Stevenson, Neil Parker, Stephane Bortzmayer a zvláštní díky patří Vince Skahanovi za jeho jedinečnou spolupráci.

Jestliže jsem na někoho zapomněl, omlouvám se: stačí mi poslat email!

Nicolai Langfeldt janl@math.uio.no verze 2.0.3, 13. března 1998

Jak se stát úplně malým administrátorem systému DNS.

1 Předmluva

Klíčová slova: DNS, bind, bind-4, bind-8, named, dialup, ppp, slip, isdn, Internet, domain, name, hosts, resolving.

1.1 Autorská práva

(C)opyright 1997 Nicolai Langfeld. Neupravujte bez doplnění autorských práv, rozšiřujte bez omezení, ale ponechejte tento odstavec.

1.2 Podíly a žádosti o pomoc

Chtěl bych poděkovat Arntu Gulbrandsenovi, který mnohokrát četl návrhy této práce a navrhl mnohá užitečná vylepšení. Chtěl bych také poděkovat lidem, kteří mi poslali návrhy a poznámky.

Tento dokument nebude nikdy dokončen. Pošlete mi prosím zprávy o vašich problémech a úspěších, dokument tak mohu vylepšovat. Peníze, komentáře nebo otázky posílejte na janl@math.uio.no. Jestliže budete odesílat elektronickou zprávu, *zajistěte prosím*, aby byla návratová adresa funkční. Předtím, než mi napíšete, si také **prosím** přečtěte část 8.

Jestliže chcete tento dokument překládat, sdělte mi to, abych věděl, ve kterých jazycích jsem byl publikován, současně vás mohu upozornit na změny v dokumentu.

1.3 Věnování

Dokument bych rád věnoval Anne Line Norheim Langfeldtové. I když jej pravděpodobně nikdy nebude číst, protože není takovým typem dívky, která by jej četla.

2 Úvod

Co to je a co není

Pro začátečníky: DNS je Domain Name System (systém pojmenování domén). DNS převádí názvy strojů na IP-adresy, dále mapuje názvy na adresy a adresy na názvy. Tento dokument ukazuje, jak takové mapování provádět v Linuxu. Mapování je vlastně vztah mezi dvěma věcmi, v našem případě mezi názvem stroje (např. *ftp.linux.org*) a IP-adresou stroje (zde 199.249.150.4).

DNS je pro nezavěšené jednou z méně průhledných oblastí síťové správy. Tento dokument by měl alespoň něco osvětlit. Popisuje, jak nastavit *jednoduchý* DNS-server. Začneme přitom s DNS-serverem pouze s vyrovnávací pamětí a přejdeme až k nastavení primárního DNS-serveru pro doménu. Složitější nastavení naleznete v 8. části. Jestliže zde nenaleznete to, co potřebujete, musíte si pročíst opravdovou dokumentaci. K té se dostanu v 9. části.

Před dalším postupem je vhodné nakonfigurovat váš stroj, aby bylo možné se z něj a na něj přihlásit pomocí telnetu a provést všechny druhy připojení na síť. Zejména je nutné vyzkoušet telnet 127.0.0.1 a mít vlastní stroj (zkuste to ihned!). Jako počáteční bod potřebujete také prospěšné soubory `/etc/nsswitch.conf` (nebo `/etc/host.conf`), `/etc/resolv.conf` a `/etc/hosts`. Nebudu zde totiž vysvětlovat jejich funkci. Jestliže ještě nemáte všechno nastavené a funkční, pomohou vám dokumenty NET-3 HOWTO a PPP HOWTO. Pročtete si je.

Když říkám „vlastní stroj“, myslím stroj, na kterém má DNS být. Ne tedy jakýkoliv další váš stroj, který používáte ve vašem síťovém snažení.

Předpokládám, že se nenacházíte za žádným druhem firewallu, který by blokoval DNS-dotazy. Jestliže se ale za ním nacházíte, budete potřebovat speciální konfiguraci, viz část 8.

Převody názvů jsou v Unixu prováděny programem `named`. Ten je součástí balíku, který je koordinován Paulem Vixiem. `Named` je obsažen ve většině distribucí Linuxu a instaluje se jako `/usr/sbin/named`. Jestliže už `named` máte, pravděpodobně jej také můžete používat; jest-

liže jej nemáte, můžete si z linuxového FTP-serveru přetáhnout binární podobu, nebo si můžete z `ftp.isc.org:/isc/bind/src/cur/bind-8/` stáhnout poslední zdrojovou podobu. Tento dokument se zabývá bind verzí 8. Používáte-li bind verzi 4, stará verze tohoto dokumentu (vztahující se k bind verzi 4) je pro vás stále k dispozici na `http://www.math.uio.no/janl/DNS/`. Jestliže manuálová stránka k named hovoří o named.conf, máte bind 8, jestliže hovoří o named.boot, máte bind 4. Jestliže máte 4 a staráte se o zabezpečení, měli byste přejít na 8.

DNS je databází rozloženou na síti. Starejte se o to, co do ní vkládáte. Jestliže sem vložíte nesmysly, budou je vidět všichni. Svůj systém DNS udržujte konzistentní a jasný - získáte tak kvalitnější služby. Naučte se jej používat, spravovat, odladovat a bude z vás další „hodný“ síťový správce, který udržuje síť před přetížením ze špatné správy.

V tomto dokumentu nastíním několik věcí, které nejsou zcela pravdivé (jsou to spíše polo-pravdy). Vše v zájmu zjednodušení. Když mi budete věřit, všechno bude (pravděpodobně) fungovat.

Tip: U všech souborů, které vám navrhuji změnit, si vytvořte záložní kopie. Kdyby se stalo, že nebude nic fungovat, můžete se ještě vrátit ke staré konfiguraci.

3 DNS-server pouze s vyrovnávací pamětí

První zásah do konfigurace DNS, velmi užitečný pro uživatele modemů

DNS-server pouze s vyrovnávací pamětí nalezne odpověď na dotazy na názvy a bude si pro příště pamatovat odpověď. Tím se příště zkrátí doba čekání, zejména pokud se nacházíte na pomalých připojeních.

Nejprve potřebujete soubor, nazvaný `/etc/named.conf`. Ten je čten při spuštění named. Pro tentokrát by měl jednoduše obsahovat:

```
// Config file for caching only DNS server

options {
    directory "/var/named";
    // Zrušení následujícího komentáře vám může pomoci
    // pokud při průchodu firwallem narazíte na problémy
    // query-source address * port 53;
};
```

```
zone "." {
    type hint;
    file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "pz/127.0.0";
};
```

Řádek „directory“ sděluje named, kde se mají soubory hledat. Zde budou všechny soubory, které bude named používat. Proto je pz adresářem pod /var/named, tedy /var/named/pz. /var/named je adresář, který odpovídá *Linux File System Standard*.

Soubor /var/named/root.hints by měl obsahovat následující:

```
.      6D IN NS      G.ROOT-SERVERS.NET.
.      6D IN NS      J.ROOT-SERVERS.NET.
.      6D IN NS      K.ROOT-SERVERS.NET.
.      6D IN NS      L.ROOT-SERVERS.NET.
.      6D IN NS      M.ROOT-SERVERS.NET.
.      6D IN NS      A.ROOT-SERVERS.NET.
.      6D IN NS      H.ROOT-SERVERS.NET.
.      6D IN NS      B.ROOT-SERVERS.NET.
.      6D IN NS      C.ROOT-SERVERS.NET.
.      6D IN NS      D.ROOT-SERVERS.NET.
.      6D IN NS      E.ROOT-SERVERS.NET.
.      6D IN NS      I.ROOT-SERVERS.NET.
.      6D IN NS      F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET. 5w6d16h IN A      192.112.36.4
J.ROOT-SERVERS.NET. 5w6d16h IN A      198.41.0.10
K.ROOT-SERVERS.NET. 5w6d16h IN A      193.0.14.129
L.ROOT-SERVERS.NET. 5w6d16h IN A      198.32.64.12
M.ROOT-SERVERS.NET. 5w6d16h IN A      202.12.27.33
A.ROOT-SERVERS.NET. 5w6d16h IN A      198.41.0.4
H.ROOT-SERVERS.NET. 5w6d16h IN A      128.63.2.53
B.ROOT-SERVERS.NET. 5w6d16h IN A      128.9.0.107
C.ROOT-SERVERS.NET. 5w6d16h IN A      192.33.4.12
```

```
D.ROOT-SERVERS.NET. 5w6d16h IN A      128.8.10.90
E.ROOT-SERVERS.NET. 5w6d16h IN A      192.203.230.10
I.ROOT-SERVERS.NET. 5w6d16h IN A      192.36.148.17
F.ROOT-SERVERS.NET. 5w6d16h IN A      192.5.5.241
```

VELICE DŮLEŽITÉ: V některých verzích tohoto dokumentu jsou výpisy programů uvedeny s prázdnými znaky nebo tabulátory před prvním znakem na řádku. Tak by to ve skutečných souborech nemělo být. Jestliže tedy soubory využíváte z tohoto dokumentu, **smažte všechny mezery na začátcích řádků.**

Nezapomeňte na to, co jsem říkal o mezerách na začátcích řádků!

Soubor popisuje světové kořenové DNS-servery. Ty se během doby mění a *musí* být udržovány. Jejich aktualizaci probírám v 6. části.

V `named.conf` následuje řádek `primary`. Jeho použití vysvětlím později, prozatím stačí v adresáři `pz` vytvořit soubor `127.0.0:`

```
@           IN      SOA      nslinux.bogus. hostmaster.linux.bogus. (
                1          ; Serial
                8H        ; Refresh
                2H        ; Retry
                1W        ; Expire
                1D)       ; Minimum TTL
                NS        ns.linux.bogus.

1           PTR      localhost.
```

Pak potřebujete `/etc/resolv.conf`, který vypadá přibližně takto:

```
search subdomain.your-domain.edu your-domain.edu
nameserver 127.0.0.1
```

Řádek „`search`“ určuje, ve kterých doménách budou vyhledány názvy hostitelů, ke kterým se chcete připojit. Řádek „`nameserver`“ určuje adresu vašeho DNS-serveru - zde vašeho stroje, protože na něm spouštíte `named` (správně je `127.0.0.1`, nezávisle na případných dalších adresách vašeho stroje). Jestliže hodláte vypsat několik DNS-serverů, pro každý z nich vložte jeden řádek „`nameserver`“ (poznámka: `named` tento soubor nikdy nečte, ale resolver, který používá `named`, jej čte).

Aby bylo jasné, co tento soubor dělá: Jestliže se klient pokusí vyhledat `foo`, pak se nejprve vyzkouší `foo.poddoména.vaše-doména.edu`, potom `foo.vaše-doména.edu` a nakonec `foo`. Jestli-

že se klient pokouší vyhledat **sunsite.unc.edu**, vyzkouší se nejprve **sunsite.unc.edu.poddoména.vaše-doména.edu** (ano, je to hloupé, ale tak to funguje), potom **sunsite.unc.edu.vaše-doména.edu** a nakonec **sunsite.unc.edu**. Na vyhledávací řádek tedy raději nedávejte příliš mnoho domén, vyhledávání by mohlo trvat dlouho.

Příklad předpokládá, že patříte do domény **poddoména.vaše-doména.edu**, váš stroj se tedy nazývá **váš-stroj.poddoména.vaše-doména.edu**. Vyhledávací řádek by neměl obsahovat vaši TLD (doménu nejvyšší úrovně, zde „edu“). Jestliže se často připojujete na hostitele v jiné doméně, můžete tuto doménu přidat do vyhledávacího řádku:

```
search subdomain.your-domain.edu your-domain.edu other-domain.com
```

A tak dále. Přirozeně, že zde použijete funkční názvy domén. Povšimněte si, že za názvy domén chybí tečky.

Poté musíte upravit buď `/etc/nsswitch.conf`, nebo `/etc/host.conf` (v závislosti na verzi `libc`). Jestliže máte `nsswitch.conf`, budeme jej upravovat, jestliže jej nemáte, budeme upravovat `host.conf`.

`/etc/nsswitch.conf` Tohle je dlouhý soubor, který určuje, odkud se vezmou různé datové typy, ze kterého souboru nebo databáze. Obsahuje nahoře obvykle užitečné komentáře, které si raději hned přečtete. Poté naleznete řádek, začínající „`hosts:`“, měl by vypadat takto:

```
hosts: files dns
```

Jestliže zde není žádný řádek, který by začínal „`hosts:`“, pak si jej vezměte odtud a vložte jej tam. Sděluje, že program by se měl dívat nejprve do souboru `/etc/hosts`, potom podle `resolv.conf` kontrolovat DNS.

`/etc/host.conf` Obsahuje pravděpodobně několik řádků, jeden by měl začínat **order** a vypadat takto:

```
order hosts,bind
```

Jestliže zde není žádný řádek „`order`“, přidejte jej. Sděluje, že rozhodovací rutiny jména hostitele budou hledat nejprve v `/etc/hosts`, poté požádají DNS-server (v `resolv.conf` jste sdělili, že je to 127.0.0.1). Tyto dva poslední soubory jsou ve většině linuxových distribucích dokumentovány v manuálové stránce `resolv(8)` (provedte „`man 8 resolv`“). Podle mého názoru je tato manuálová stránka celkem čitelná a měl by si ji přečíst každý, zejména správci DNS. Přečtete si ji hned, když to budete odkládat, nikdy se k tomu nedostanete.

3.1 Spuštění named

Takže nastal čas spustit named. Jestliže používáte připojení přes modem, tak se nejprve připojte. Zadejte „`ndc start`“ a stiskněte **Enter** (bez voleb). Jestli to zlobí, zkuste místo toho „`/usr/sbin/ndc start`“. Pokud to pořád zlobí, přejděte k 8. části. Nyní můžete otestovat nastavení. Jestliže si při spuštění named (provedte `tail -f /var/log/messages`) zobrazíte váš soubor se zprávami (obvykle se nazývá `/var/adm/messages`, ale může být i v adresáři `/var/log` nebo v souboru `syslog`), měl by se objevit podobný výpis:

(některé řádky jsou rozloženy do dvou)

```
Feb 15 01:26:17 roke named[6091]: starting. named 8.1.1 Sat Feb 14 \
00:18:20 MET 1998 ^Ijanl@roke.uio.no:/var/tmp/bind-8.1.1/src/bin/named
Feb 15 01:26:17 roke named[6091]: cache zone ""\
(IN) loaded (serial 0)
Feb 15 01:26:17 roke named[6091]: master zone "0.0.127.in-addr.arpa" \
(IN) loaded (serial 1)
Feb 15 01:26:17 roke named[6091]: listening [127.0.0.1].53 (lo)
Feb 15 01:26:17 roke named[6091]:\
listening [129.240.230.92].53 (ipp0)
Feb 15 01:26:17 roke named[6091]:\
Forwarding source address is [0.0.0.0].1040
Feb 15 01:26:17 roke named[6092]: Ready to answer queries.
```

Jestliže jsou zde nějaké zprávy o chybách, pak ta chyba někde musí být. Named pojmenuje soubor, ve kterém ji naleznete (doufám, že jeden z `named.conf` a `root.hints`). Pomocí `kill` zrušte named, vraťte se zpět a soubor zkontrolujte.

Nyní nastal čas spustit `nslookup` a vyzkoušet výsledek vašeho snažení.

```
$ nslookup
Default Server: localhost
Address: 127.0.0.1
>
```

Jestli dostanete uvedený výsledek, tak to funguje. Doufejme. Pokud se objeví něco jiného, vraťte se a všechno zkontrolujte. Pokaždé, když změníte soubor `named.conf`, musíte pomocí příkazu `ndc restart named` znovu spustit.

Nyní můžete zadávat dotazy. Zkuste vyhledávat některé blízké stroje. Pro mě je to **pat.uio.no** (na univerzitě v Oslo):

```
> pat.uio.no
Server: localhost
Address: 127.0.0.1
```

```
Name: pat.uio.no
Address: 129.240.130.16
```

Nslookup nyní požádal váš `named` o nalezení stroje **pat.uio.no**. Poté kontaktoval jeden z DNS-serverů, vyjmenovaných ve vašem souboru `root.hints`, a požádal o odpověď odtud. Výsledek můžete dostat až za chvíli, protože se prohledávají všechny domény, vyjmenované v `/etc/resolv.conf`.

Jestliže se na to stejně dotážete ještě jednou, dostanete následující:

```
> pat.uio.no
Server: localhost
Address: 127.0.0.1

Non-authoritative answer:
Name: pat.uio.no
Address: 129.240.2.50
```

Povšimněte si řádku „`Non-authoritative answer:`“, který se nám nyní objevil. Znamená to, že `named` se pro odpověď neodebral na síť, ale jen do své vyrovnávací paměti, kde ji také našel. Ale tato informace *by mohla* být stará. Proto jste na tohle (velmi malé) nebezpečí upozornění sdělením „`Non-authoritative answer:`“ (neautorizovaná odpověď). Když vám to `nslookup` sdělí při druhém dotazu na stejnou lokaci, jedná se o znamení, že `named` funguje a ukládá si informace do vyrovnávací paměti. `Nslookup` ukončíte příkazem „`exit`“.

Nyní víte, jak nastavit `named` s vyrovnávací pamětí. Dejte si na oslavu jedno pivo, mléko nebo to, čemu dáváte přednost.

4 Jednoduchá doména

Jak nastavit vlastní doménu

4.1 Nejprve ale trochu suché teorie

Předtím, než *opravdu* začneme tuto část, vám předložím teorii o funkci DNS. A vy si ji přečte- te, protože je pro vás poučná. Jestli se vám „nechce“, tak ji alespoň v rychlosti projděte. Ale nezapomeňte se zastavit u obsahu souboru `named.conf`.

DNS je hierarchický systém. Jeho vrchol se píše „.“ a vyslovuje „root“. Pod „.“ je mnoho domén nejvyšší úrovně (TLD). Nejznámější jsou ORG, COM, EDU a NET, ale je zde mnoho dalších.

Při vyhledávání stroje se dotaz zpracovává do hierarchie rekurzivně od vrcholu. Jestliže hodlá- te nalézt adresu pro **prep.ai.mit.edu**, musí váš DNS-server nalézt DNS-server, který obsluhu- je doménu edu. Požádá .server (kořenový server je znám, od toho je zde soubor `root.hints`). Kořenový server poskytne seznam serverů edu:

```
$ nslookup
Default Server: localhost
Address: 127.0.0.1
```

Začátek dotazů na kořenový server:

```
> server c.root-servers.net.
Default Server: c.root-servers.net
Address: 192.33.4.12
```

Nastavení typu dotazu na NS (záznamy DNS serverů):

```
> set q=ns
```

Dotaz na edu:

```
> edu.
```

Tečka je zde důležitá - sděluje serveru, že se ptáme na edu přímo pod .(tím se vyhledávání zjednoduší).

```
edu      nameserver = A.ROOT-SERVERS.NET
edu      nameserver = H.ROOT-SERVERS.NET
```

```
edu      nameserver = B.ROOT-SERVERS.NET
edu      nameserver = C.ROOT-SERVERS.NET
edu      nameserver = D.ROOT-SERVERS.NET
edu      nameserver = E.ROOT-SERVERS.NET
edu      nameserver = I.ROOT-SERVERS.NET
edu      nameserver = F.ROOT-SERVERS.NET
edu      nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET  internet address = 198.41.0.4
H.ROOT-SERVERS.NET  internet address = 128.63.2.53
B.ROOT-SERVERS.NET  internet address = 128.9.0.107
C.ROOT-SERVERS.NET  internet address = 192.33.4.12
D.ROOT-SERVERS.NET  internet address = 128.8.10.90
E.ROOT-SERVERS.NET  internet address = 192.203.230.10
I.ROOT-SERVERS.NET  internet address = 192.36.148.17
F.ROOT-SERVERS.NET  internet address = 192.5.5.241
G.ROOT-SERVERS.NET  internet address = 192.112.36.4
```

Tohle nám říká, že ***.root-servers.net** obsluhuje **edu**, takže můžeme požádat **c**. Nyní chceme vědět, kdo obsluhuje následující úroveň názvu domény – **mit.edu**:

```
> mit.edu.
Server: c.root-servers.net
Address: 192.33.4.12
```

Non-authoritative answer:

```
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu
mit.edu nameserver = STRAWB.mit.edu
```

Authoritative answers can be found from:

```
W20NS.mit.edu internet address = 18.70.0.160
BITSY.mit.edu internet address = 18.72.0.3
STRAWB.mit.edu internet address = 18.71.0.151
```

Takže **mit** obsluhují **steawb**, **w20ns** a **bitsy**. Vyberte si jeden a pokračujte s **ai.mit.edu**:

```
> server W20NS.mit.edu.
```


U názvů hostitelů nezáleží na velkých a malých písmenech, zde jsou přímo okopírovány (pomocí paste) z obrazovky.

```
Server: W20NS.mit.edu  
Address: 18.70.0.160
```

```
> ai.mit.edu.  
Server: W20NS.mit.edu  
Address: 18.70.0.160
```

Non-authoritative answer:

```
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU  
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU  
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU  
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU  
ai.mit.edu      nameserver = LIFE.AI.MIT.EDU  
ai.mit.edu      nameserver = BEET-CHEX.AI.MIT.EDU  
ai.mit.edu      nameserver = MINI-WHEATS.AI.MIT.EDU  
ai.mit.edu      nameserver = COUNT-CHOCULA.AI.MIT.EDU  
ai.mit.edu      nameserver = MINTAKA.LCS.MIT.EDU
```

Authoritative answers can be found from:

```
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU  
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU  
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU  
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU  
AI.MIT.EDU      nameserver = LIFE.AI.MIT.EDU  
AI.MIT.EDU      nameserver = BEET-CHEX.AI.MIT.EDU  
AI.MIT.EDU      nameserver = MINI-WHEATS.AI.MIT.EDU  
AI.MIT.EDU      nameserver = COUNT-CHOCULA.AI.MIT.EDU  
AI.MIT.EDU      nameserver = MINTAKA.LCS.MIT.EDU  
ALPHA-BITS.AI.MIT.EDU      internet address = 128.52.32.5  
GRAPE-NUTS.AI.MIT.EDU      internet address = 128.52.36.4  
TRIX.AI.MIT.EDU      internet address = 128.52.37.6  
MUESLI.AI.MIT.EDU      internet address = 128.52.39.7  
LIFE.AI.MIT.EDU      internet address = 128.52.32.80  
BEET-CHEX.AI.MIT.EDU      internet address = 128.52.32.22
```

```
MINI-WHEATS.AI.MIT.EDU      internet address = 128.52.54.11
COUNT-CHOCULA.AI.MIT.EDU  internet address = 128.52.38.22
MINTAKA.LCS.MIT.EDU        internet address = 18.26.0.36
```

Takže DNS-server pro **ai.mit.edu** je **muesli.ai.mit.edu**:

```
> server MUESLI.AI.MIT.EDU
Default Server: MUESLI.AI.MIT.EDU
Address: 128.52.39.7
```

Nyní změním typ požadavku (query). Nalezli jsme DNS-server, takže se ho zeptáme na vše, co ví o **prep.ai.mit.edu**.

```
> set q=any
> prep.ai.mit.edu.
Server: MUESLI.AI.MIT.EDU
Address: 128.52.39.7
```

```
prep.ai.mit.edu CPU = dec/decstation-5000.25 OS = unix
prep.ai.mit.edu
  inet address = 18.159.0.42, protocol = tcp
  ftp telnet smtp finger
prep.ai.mit.edu preference = 1, mail exchanger = gnu-life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = beet-chex.ai.mit.edu
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
gnu-life.ai.mit.edu      internet address = 128.52.32.60
beet-chex.ai.mit.edu    internet address = 128.52.32.22
alpha-bits.ai.mit.edu  internet address = 128.52.32.5
mini-wheats.ai.mit.edu internet address = 128.52.54.11
trix.ai.mit.edu        internet address = 128.52.37.6
```

```

muesli.ai.mit.edu      internet address = 128.52.39.7
count-chocula.ai.mit.edu internet address = 128.52.38.22
mintaka.lcs.mit.edu   internet address = 18.26.0.36
life.ai.mit.edu        internet address = 128.52.32.80

```

Takže počínaje u „*„*“ jsme postupně našli všechny DNS-servery pro následující úroveň názvu domény. Pokud byste místo použití všech ostatních serverů použili vlastní DNS-server, váš `named` by si samozřejmě všechny získané informace ukládal do vyrovnávací paměti a případný další dotaz na to stejné by už zodpověděl sám.

Méně diskutovanou, ale stejně důležitou doménou je **in-addr.arpa**. Je to „stejná“ doména jako ostatní. Umožňuje získat názvy hostitelů podle adres. Zde je důležité si povšimnout, že IP-adresy jsou v doméně `in-addr.arpa` napsány obráceně. Jestliže máte adresu stroje 192.128.52.43, `named` pokračuje stejně jako u předchozího příkladu **prep.ai.mit.edu**. Nalezne servery **arpa.**, nalezne servery **in-addr.arpa.**, nalezne servery **192.in-addr.arpa.**, nalezne servery **128.192.in-addr.arpa.**, nalezne servery **52.128.192.in-addr.arpa.** Nalezne požadovaný záznam pro **43.52.128.192.in-addr.arpa**. Jasně? (odpovězte „Jasně!“) Tak dva roky se vám obrácené pořadí čísel může ještě zdát matoucí.

Teď jsem lhal. DNS sice nepracuje tak, jak jsem říkal, ale zato dost podobně.

4.2 Naše vlastní doména

Nyní k definici vaší vlastní domény. Chystáme se vytvořit doménu `linux.bogus` a definovat v ní stroje. Využívám zde fiktivní (bogus) název domény, aby bylo jasné, že tam nikoho nevyrušíme.

Ještě něco, než začneme: Ve jménech hostitelů nejsou povoleny všechny znaky. Jsme omezeni na znaky anglické abecedy (a-z), číslice (0-9) a pomlčky „-“. Držte se toho. Pro DNS se nerozlišují velká a malá písmenka. Takže **pat.uio.no** je to stejné, jako **Pat.UiO.No**.

Tuto část jsme již začali s tímto řádkem v `named.conf`:

```

zone "0.0.127.in-addr.arpa" {
    type master;
    file "pz/127.0.0";
};

```

Povšimněte si, že zde na koncích názvů domén není „.“. To znamená, že nyní budeme definovat zónu **0.0.127.in-addr.arpa**, pro kterou jsme hlavním serverem a která je uložena v souboru `pz/127.0.0`. Tento soubor už jsme nastavili, vypadá takto:

```

@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                        1          ; Serial
                        8H         ; Refresh
                        2H         ; Retry
                        1W         ; Expire
                        1D)        ; Minimum TTL
                        NS      ns.linux.bogus.
1      PTR     localhost.

```

Povšimněte si zde „@“ na konci všech plných názvů domén, což je v kontrastu se souborem `named.conf`. Někteří lidé rádi začínají každý zónový soubor direktivou `$ORIGIN`, ale je to zbytečné. Původ (kam náleží v hierarchii DNS) zónového souboru je určen na zónové řádce v souboru `named.conf`, v tomto případě to je **0.0.127.in-addr.arpa**.

Tento „zónový soubor“ obsahuje 3 „zdrojové záznamy“ (RR): SOA, NS a PTR. SOA je zkratkou pro začátek autority. „@“ je speciální označení, znamenající počátek, a protože „doménový“ sloupec pro tento soubor říká **0.0.127.in-addr.arpa**, první řádek ve skutečnosti znamená

```
0.0.127.in-addr.arpa. IN SOA ...
```

NS je zdrojový záznam pro DNS-server. Na začátku tohoto řádku není žádné „@“, je to *implicitní*, protože poslední řádek začínal na „@“. Tím se ušetří trocha psaní. Takže NS-řádek skutečně vypadá takto:

```
0.0.127.in-addr.arpa. IN NS ns.linux.bogus
```

Sděluje DNS, který stroj je DNS-serverem domény **0.0.127.in-addr.arpa**. Je to **ns.linux.bogus**. „ns“ je obecný název pro DNS-servery, ale název může být i jiný, stejně jako u webových serverů, které by měly být pojmenovány *www.něco*, ale často jsou pojmenovány jinak.

A konečně záznam PTR sděluje, že hostitel na adrese 1 v podsíti **0.0.127.in-addr.arpa** (127.0.0.1) je pojmenován **localhost**.

Záznam SOA je úvodem ke *všem* zónovým souborům a v každém zónovém souboru by měl být právě jeden - jako první záznam. Popisuje zónu, ze které pochází (stroj, pojmenovaný **ns.linux.bogus**), která je zodpovědná za jeho obsah (**hostmaster@linux.bogus**), jaké verze je tento zónový soubor (serial: 1) a další věci, které mají co do činění se sekundárními DNS-servery s vyrovnávací pamětí. Pro zbylá pole (refresh, retry, expire a minimum) použijte čísla z našeho dokumentu a měli byste být bez problémů.

Nyní znovu spusťte `named` (příkaz je `ndc restart`) a to co jste vytvořili otestujte pomocí `nslookup`:

```
$ nslookup
```

```
Default Server: localhost
Address: 127.0.0.1
```

```
> 127.0.0.1
Server: localhost
Address: 127.0.0.1
```

```
Name: localhost
Address: 127.0.0.1
```

Takže z 127.0.0.1 se získal **localhost**, to je dobré. Nyní pro náš hlavní úkol (doména **linux.bogus**) vložte do `named.conf` novou „zónovou“ část:

```
zone "linux.bogus" {
    notify no;
    type master;
    file "pz/linux.bogus";
};
```

Povšimněte si, že na konci názvu domény v souboru `named.conf` opět chybí „.“.

Do zónového souboru `linux.bogus` vložíme některá zcela fiktivní data:

```
;
; Zone file for linux.bogus
;
; The full zone file
;
@      IN      SOA    ns.linear.bogus.    hostmaster.linear.bogus. (
                                199802151    ; serial, todays date +
                                todays serial #
                                8H          ; refresh, seconds
                                2H          ; retry, seconds
                                1W          ; expire, seconds
                                1D )       ; minimum, seconds
;
```

```

NS      ns                ; Inet Address of DNS server
MX      10 mail.linux.bogus ; Primary Mail Exchanger
MX      20 mail.friend.bogus. ; Secondary Mail Exchanger
;
localhost A      127.0.0.1
ns        A      192.168.196.2
mail     A      192.168.196.4

```

K záznamu SOA musí být poznamenány ještě dvě věci. `ns.linux.bogus` musí být skutečným strojem s `A`-záznamem. Pro stroj, zmíněný v záznamu SOA, není povoleno mít záznam `CNAME`. Jeho jméno nemusí být „ns“ a může to být jakýkoliv povolený název hostitele. Dále `hostmaster.linux.bogus` by měl být čten jako **hostmaster@linux.bogus** a měl by to být poštovní alias nebo schránka, kde by měla osoba(y), spravující DNS, často číst poštu. Jakákoliv pošta, týkající se domény, je odesílána na zde vypsanou adresu. Název nemusí být „hostmaster“, ale nějaká jiná povolená e-mailová adresa. Ale i ten „hostmaster“ by *měl být* funkční.

V tomto souboru je jeden nový typ zdrojového záznamu, `MX` neboli Mail eXchanger (poštovní server). Sděluje poštovním systémům kam mají posílat poštu, určenou na **někdo@linux.bogus**, jmenovitě na **mail.linux.bogus** nebo **mail.friend.bogus**. Číslo před každým názvem stroje je priorita daného `MX`. Na `MX` s nejnižším číslem (10) by měla být pošta odesílána primárně. Jestliže to není možné, pošta je odeslána na `MX` s druhým nejvyšším číslem (sekundární), zde **mail.friend.bogus** s prioritou 20.

Spuštěním `ndc restart` znovu spusíte `named`. Výsledky zkontrolujte pomocí `nslookup`:

```

$ nslookup
> set q=any
> linux.bogus
Server: localhost
Address: 127.0.0.1

linux.bogus
  origin = ns.linux.bogus
  mail addr = hostmaster.linux.bogus
  serial = 199802151
  refresh = 28800 (8 hours)
  retry = 7200 (2 hours)
  expire = 604800 (7 days)
  minimum ttl = 86400 (1 day)

```

```
linux.bogus nameserver = ns.linux.bogus
linux.bogus preference = 10, mail exchanger = \
    mail.linux.bogus.linux.bogus
linux.bogus preference = 20, mail exchanger = mail.friend.bogus
linux.bogus nameserver = ns.linux.bogus
ns.linux.bogus internet address = 192.168.196.2
mail.linux.bogus internet address = 192.168.196.4
```

Po podrobném prozkoumání objevíte chybu. Řádek

```
linux.bogus preference = 10, mail exchanger = \
    mail.linux.bogus.linux.bogus
```

je celý špatně. Měl by vypadat

```
linux.bogus preference = 10, mail exchanger = mail.linux.bogus
```

Tu chybu jsem udělal schválně, abyste se mohli poučit. Když se podíváte do zónového souboru, zjistíte, že v řádku

```
MX 10 mail.linux.bogus ; Primary Mail Exchanger
```

chybí tečka. Jestliže název stroje nekončí v zónovém souboru tečkou, na jeho konec je přidán počátek, takže máme dvojité **linux.bogus.linux.bogus**. Proto správně je buď

```
MX 10 mail.linux.bogus. ; Primary Mail Exchanger
```

nebo

```
MX 10 mail ; Primary Mail Exchanger
```

Já dávám přednost druhé formě, je zde méně psaní. Existují úzkoprsí uživatelé, kteří teď nesouhlasí, ale také uživatelé, kteří souhlasí. V zónovém souboru by měla být doména buď vypsána a zakončena „.“, nebo by neměla být vůbec zmíněna a brala by se implicitně z počátku.

Musím zde znovu zdůraznit, že v `named.conf` by *neměly* být za názvy domén „.“. Nedokážete si představit, jak často tyto přebývající nebo chybějící tečky komplikují život.

Takže zde je nový zónový soubor s některými informacemi navíc:

```
;
; Zone file for linux.bogus
;
; The full zone file
;
@      IN      SOA      ns.linux.bogus.      hostmaster.linux.bogus. (
                                199802151      ; serial, todays date +
                                                todays serial #
                                8H              ; refresh, seconds
                                2H              ; retry, seconds
                                1W              ; expire, seconds
                                1D )           ; minimum, seconds
;

      TXT      "Linux.Bogus, your DNS consultants"
      NS       ns              ; Inet Address of DNS server
      NS       ns.friend.bogus.
      MX       10 mail         ; Primary Mail Exchanger
      MX       20 mail.friend.bogus. ; Secondary Mail Exchanger

localhost      A      127.0.0.1

gw              A 1      92.168.196.1
              HINFO    "Cisco" "IOS"
              TXT      "The router"

ns              A      192.168.196.2
              MX       10 mail
              MX       20 mail.friend.bogus.
              HINFO    "Pentium" "Linux 2.0"

www            CNAME    ns

donald         A      192.168.196.3
              MX       10 mail
              MX       20 mail.friend.bogus.
              HINFO    "i486" "Linux 2.0"
              TXT      "DEK"
```



```
mail          A      192.168.196.4
              MX    10 mail
              MX    20 mail.friend.bogus.
              HINFO "386sx" "Linux 1.2"

ftp           A      192.168.196.5
              MX    10 mail
              MX    20 mail.friend.bogus.
              HINFO "P6" "Linux 2.1.86"
```

Nalezneme zde množství nových zdrojových záznamů: HINFO (informace o hostiteli) má dvě části a je zvykem je oddělovat. První částí je hardware nebo procesor stroje a druhou částí je software nebo operační systém stroje. Stroj, nazvaný „ns“, má procesor (CPU) Pentium a používá Linux 2.0. CNAME (kanonické jméno) je způsob pojmenování stroje více názvy. Takže `www` je aliasem pro `ns`.

Používání záznamu CNAME je trochu kontroverzní. Bezpečné pravidlo ale říká, že záznamy MX, CNAME nebo SOA by se *neměly* odkazovat na záznam CNAME, měly by se odkazovat pouze na něco se záznamem A. Takže následující by bylo špatně

```
foobar       CNAME  www                ; NO!
```

ale toto by bylo správně

```
foobar       CNAME  ns                  ; Yes!
```

Pro adresy elektronické pošty je také bezpečné předpokládat, že CNAME není platný název hostitele:

webmaster@www.linux.bogus je podle předchozího nastavení neplatnou adresou elektronické pošty. Můžete předpokládat, že pouze málo poštovních správců „zvenku“ bude toto pravidlo akceptovat (i když vám funguje dobře). Způsob, jakým se situace vyřeší, je použití A-záznamů (a snad i dalších, jako jsou MX):

```
www          A      192.168.196.2
```

V určitých kruzích se použití CNAME *nedoporučuje*. Takže CNAME neberte *příliš* vážně. Ale tento dokument a množství hostitelů se tímto pravidlem neřídí.

Spuštěním `ndc reload` nahrajte novou databázi. Tím přinutíte `named`, aby svoje soubory znovu načel.

```
$ nslookup
Default Server: localhost
Address: 127.0.0.1
> ls -d linux.bogus
```

To znamená, že všechny záznamy by měly být vypsány. Výsledek je tento:

```
[localhost]
$ORIGIN linux.bogus.
@          1D IN SOA      ns hostmaster (
                199802151      ; serial
                8H           ; refresh
                2H           ; retry
                1W           ; expiry
                1D )         ; minimum

                1D IN NS      ns
                1D IN NS      ns.friend.bogus.
                1D IN TXT     "Linux.Bogus, your DNS consultants"
                1D IN MX      10 mail
                1D IN MX      20 mail.friend.bogus.
gw         1D IN A          192.168.196.1
                1D IN HINFO   "Cisco" "IOS"
                1D IN TXT     "The router"
mail      1D IN A          192.168.196.4
                1D IN MX      10 mail
                1D IN MX      20 mail.friend.bogus.
                1D IN HINFO   "386sx" "Linux 1.0.9"
localhost 1D IN A          127.0.0.1
www       1D IN CNAME     ns
donald    1D IN A          192.168.196.3
                1D IN MX      10 mail
                1D IN MX      20 mail.friend.bogus.
                1D IN HINFO   "i486" "Linux 1.2"
                1D IN TXT     "DEK"
ftp       1D IN A          192.168.196.5
                1D IN MX      10 mail
                1D IN MX      20 mail.friend.bogus.
```

```

ns          1D IN HINFO  "P6" "Linux 1.3.59"
           1D IN A      192.168.196.2
           1D IN MX     10 mail
           1D IN MX     20 mail.friend.bogus.
           1D IN HINFO  "Pentium" "Linux 1.2"
@          1D IN SOA   ns hostmaster (
                    199802151      ; serial
                    8H      ; refresh
                    2H      ; retry
                    1W      ; expiry
                    1D ) ; minimum

```

Toto je správné. Jak je vidět, vypadá to spíše jako samotný zónový soubor. Podívejme se, co se zde říká o samotném `www`:

```

> set q=any
> www.linux.bogus.
Server: localhost
Address: 127.0.0.1

```

```

www.linux.bogus canonical name = ns.linux.bogus
linux.bogus nameserver = ns.linux.bogus
linux.bogus nameserver = ns.friend.bogus
ns.linux.bogus internet address = 192.168.196.2

```

Jinými slovy **www.linux.bogus** má pravý název **ns.linux.bogus** a dává vám některé informace o ns, postačující k připojení.

Nyní máme polovinu za sebou.

4.3 Reverzní zóna

Nyní mohou programy převádět názvy z `linux.bogus` na adresy, ke kterým se mohou připojit. Nutná je ale také reverzní zóna, která DNS umožňuje převod adresy na název. Tento název je využíván mnoha servery různých druhů (FTP, IRC, WWW a další), aby se rozhodly, jestli s vámi budou mluvit nebo ne, a když se rozhodnou kladně, tak se ještě musí určit, jaká obdržíte práva. Pro plný přístup ke všem službám Internetu je nutná reverzní zóna.

Následující vložte do `named.conf`:

```
zone "196.168.192.in-addr.arpa" {
    notify no;
    type master;
    file "pz/192.168.196";
};
```

Je to stejná situace jako u **0.0.127.in-addr.arpa** a obsahy jsou podobné:

```
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                                199802151 ; Serial, todays date + todays serial
                                8H          ; Refresh
                                2H          ; Retry
                                1W          ; Expire
                                1D)        ; Minimum TTL
      NS      ns.linux.bogus.

1      PTR      gw.linux.bogus.
2      PTR      ns.linux.bogus.
3      PTR      donald.linux.bogus.
4      PTR      mail.linux.bogus.
5      PTR      donald.linux.bogus.
```

Nyní znovu spusťte `named` (`ndc restart`) a znovu otestujte výsledky vaší práce pomocí `nslookup`:

```
> 192.168.196.4
Server: localhost
Address: 127.0.0.1
```

```
Name: mail.linux.bogus
Address: 192.168.196.4
```

vypadá to v pořádku, takže vyzkoušejte úplně všechno:

```
> ls -d 196.168.192.in-addr.arpa
[localhost]
$ORIGIN 196.168.192.in-addr.arpa.
@          1D IN SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                                199802151 ; serial
                                8H          ; refresh
```

```

                                2H          ; retry
                                1W          ; expiry
                                1D )       ; minimum

1D IN NS      ns.linux.bogus.
1 1D IN PTR   gw.linux.bogus.
2 1D IN PTR   ns.linux.bogus.
3 1D IN PTR   donald.linux.bogus.
4 1D IN PTR   mail.linux.bogus.
5 1D IN PTR   donald.linux.bogus.
@ 1D IN SOA   ns.linux.bogus. hostmaster.linux.bogus. (
                                199802151    ; serial
                                8H          ; refresh
                                2H          ; retry
                                1W          ; expiry
                                1D )       ; minimum

```

Opět správně!

Tady bych měl doplnit ještě několik věcí. IP-čísla, která jsem v příkladu využil, jsou vyňata z bloku „soukromých sítí“. Tato čísla není povoleno na Internetu veřejně používat. Proto je jejich použití v našem příkladu bezpečné. Další záležitostí je řádek `notify no`; `Named` z něho zjistí, že při aktualizaci jednoho ze svých zónových souborů nemá upozorňovat svoje sekundární servery. V `bind-8` může `named` při aktualizaci zóny upozornit další servery, vypsané v záznamech NS. To je vhodné pro pravidelné využívání, ale pro soukromé experimenty se zónami by tato funkce měla být vypnuta. Přece nechceme, aby náš experiment zamožil Internet.

A samozřejmě musím dodat, že tato doména a její adresy jsou všechny smyšlené. Příklad skutečné domény bude v následující části.

5 Příklad skutečné domény

Kde nalezneme některé soubory *skutečných* domén

Uživatelé navrhli, abych vedle ukázkového příkladu použil příklad skutečné funkční domény.

Příklad jsem použil se souhlasem Davida Bullocka z LAND-5. Tyto soubory byly aktuální 24. 9. 1996 a potom byly mnou upraveny, aby odpovídaly omezením `bind-8` a využívaly rozšíření. Takže zde máte určitý rozdíl oproti současné situaci na DNS-serverech LAND-5.

5.1 /etc/named.conf (nebo /var/named/named.conf)

Zde nalezneme základní vazby pro dvě reverzní zóny, které jsou nutné: síť 127.0.0 a podsíť 206.6.177 v LAND-5. A také základní vazbu pro přesměrovací zónu `land-5.com`. Pověšim-něte si také, že místo hromadění souborů v adresáři `pz` (jak to dělám já v tomto dokumentu) jsou soubory v adresáři `zone`.

```
// Boot file for LAND-5 DNS server
```

```
options {
    directory "/var/named";
};

zone "." {
    type hint;
    file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "zone/127.0.0";
};

zone "land-5.com" {
    type master;
    file "zone/land-5.com";
};

zone "177.6.206.in-addr.arpa" {
    type master;
    file "zone/206.6.177";
};
```

Jestliže toto vložíte do vašeho souboru `named.conf`, abyste si s tím pohráli, *PROSÍM* použijte pro obě zóny `land-5` v zónových částech `notify no`; - vyhněte se tak nehodám.

5.2 /var/named/root.hints

Uvědomte si, že tento soubor je dynamický a ten, který jsem zde popsal, je již zastaralý. Pro vás bude lepší využít nějaký novější, vytvořený pomocí `dig`, jak je výše popsáno.

```
; <<>> DiG 8.1 <<>> @A.ROOT-SERVERS.NET. =
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;; ., type = NS, class = IN

;; ANSWER SECTION:
.           6D IN NS           G.ROOT-SERVERS.NET.
.           6D IN NS           J.ROOT-SERVERS.NET.
.           6D IN NS           K.ROOT-SERVERS.NET.
.           6D IN NS           L.ROOT-SERVERS.NET.
.           6D IN NS           M.ROOT-SERVERS.NET.
.           6D IN NS           A.ROOT-SERVERS.NET.
.           6D IN NS           H.ROOT-SERVERS.NET.
.           6D IN NS           B.ROOT-SERVERS.NET.
.           6D IN NS           C.ROOT-SERVERS.NET.
.           6D IN NS           D.ROOT-SERVERS.NET.
.           6D IN NS           E.ROOT-SERVERS.NET.
.           6D IN NS           I.ROOT-SERVERS.NET.
.           6D IN NS           F.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
G.ROOT-SERVERS.NET. 5w6d16h IN A           192.112.36.4
J.ROOT-SERVERS.NET. 5w6d16h IN A           198.41.0.10
K.ROOT-SERVERS.NET. 5w6d16h IN A           193.0.14.129
L.ROOT-SERVERS.NET. 5w6d16h IN A           198.32.64.12
M.ROOT-SERVERS.NET. 5w6d16h IN A           202.12.27.33
A.ROOT-SERVERS.NET. 5w6d16h IN A           198.41.0.4
H.ROOT-SERVERS.NET. 5w6d16h IN A           128.63.2.53
B.ROOT-SERVERS.NET. 5w6d16h IN A           128.9.0.107
C.ROOT-SERVERS.NET. 5w6d16h IN A           192.33.4.12
```

```

D.ROOT-SERVERS.NET. 5w6d16h IN A           128.8.10.90
E.ROOT-SERVERS.NET. 5w6d16h IN A           192.203.230.10
I.ROOT-SERVERS.NET. 5w6d16h IN A           192.36.148.17
F.ROOT-SERVERS.NET. 5w6d16h IN A           192.5.5.241

;; Total query time: 215 msec
;; FROM: roke.uio.no to SERVER: A.ROOT-SERVERS.NET. 198.41.0.4
;; WHEN: Sun Feb 15 01:22:51 1998
;; MSG SIZE sent: 17 rcvd: 436

```

5.3 /var/named/zone/127.0.0

Pouze základy – klasický záznam SOA a záznam, který mapuje 127.0.0 na **localhost**. Nutné jsou oba. V tomto souboru by nemělo být nic dalšího. Tyto soubory nemusí být nikdy aktualizovány, pokud se nezmění adresy vašeho DNS-serveru nebo hostmastera.

```

@           IN      SOA      land-5.com. root.land-5.com. (
                                199609203      ; Serial
                                28800          ; Refresh
                                7200          ; Retry
                                604800       ; Expire
                                86400)       ; Minimum TTL
                                NS      land-5.com.

1           PTR     localhost.

```

5.4 /var/named/zone/land-5.com

Zde vidíme povinný záznam SOA, požadovaný záznamy NS. Je vidět, že má sekundární DNS-server na `ns2.psi.net`. Tak to má být - jako zálohu *vždy* používat sekundární server na jiném hostiteli. Je zde také patrné, že jako hlavní hostitel je zde `land-5`, který se stará o mnoho různých internetových služeb. Využívá k tomu CNAME (alternativou je využití záznamů A).

Jak je vidět ze záznamu SOA, zónové soubory pochází z `land-5.com`, kontaktní osobou je **root@land-5.com**. **hostmaster** je další často používanou adresou kontaktní osoby. Sériové číslo je v upravitelném formátu `yyyymmdd` (rok, měsíc, den) s připojenými dnešními sériovými čísly; 20. 9. 1996 se jedná pravděpodobně o šestou verzi zónového souboru. Uvědomte si, že

sériové číslo se *musí* stále zvyšovat, zde máme pouze *jednu* číslici pro dnešní sériová čísla, takže po 9 editacích souboru se může v jeho editacích pokračovat až následující den. Zvažte využití dvou číslic.

```
@      IN      SOA 1 and-5.com. root.land-5.com. (
                199609206      ; serial, todays date
                + todays serial #
                8H              ; refresh, seconds
                2H              ; retry, seconds
                1W              ; expire, seconds
                1D )            ; minimum, seconds
      NS      land-5.com.
      NS      ns2.psi.net.
      MX 10   land-5.com. ; Primary Mail Exchanger

localhost      A      127.0.0.1

router         A      206.6.177.1

land-5.com.    A      206.6.177.2
ns             A      206.6.177.3
www           A      207.159.141.192

ftp           CNAME   land-5.com.
mail         CNAME   land-5.com.
news        CNAME   land-5.com.

funn         A      206.6.177.2
@           TXT     "LAND-5 Corporation"

;
;           Workstations
;
ws-177200    A      206.6.177.200
            MX      10 land-5.com. ; Primary Mail Host
ws-177201    A      206.6.177.201
            MX      10 land-5.com. ; Primary Mail Host
```

```

ws-177202      A      206.6.177.202
               MX      10 land-5.com.          ; Primary Mail Host
ws-177203      A      206.6.177.203
               MX      10 land-5.com.          ; Primary Mail Host
ws-177204      A      206.6.177.204
               MX      10 land-5.com.          ; Primary Mail Host
ws-177205      A      206.6.177.205
               MX      10 land-5.com.          ; Primary Mail Host
; {Many repetitive definitions deleted - SNIP}
ws-177250      A      206.6.177.250
               MX      10 land-5.com.          ; Primary Mail Host
ws-177251      A      206.6.177.251
               MX      10 land-5.com.          ; Primary Mail Host
ws-177252      A      206.6.177.252
               MX      10 land-5.com.          ; Primary Mail Host
ws-177253      A      206.6.177.253
               MX      10 land-5.com.          ; Primary Mail Host
ws-177254      A      206.6.177.254
               MX      10 land-5.com.          ; Primary Mail Host

```

Jestliže otestujete nameserver z land-5, zjistíte, že názvy hostitelů mají formát **ws_číslo**. Od pozdějších verzí bind 4 začal named omezovat znaky, které mohou být použity v názvech lokací. Takže v bind-8 by všechno nefungovalo a já jsem pomlčky „-“ nahradil podtržítky „_“.

Dále za zmínku stojí to, že pracovní stanice nemají jednotlivé názvy, ale spíše předpony, následované posledními dvěma částmi IP-adres. Použití takové konvence značně zjednoduší údržbu, ale je trochu neosobní a může být zdrojem odporu vašich uživatelů.

Zřejmě je také **funn.land-5.com** aliasem pro land-5.com, ale s použitím záznamu A, ne CNAME.

5.5 /var/named/zone/206.6.177

Tento soubor ještě později okomentuji.

```

@              IN      SOA      land-5.com.  root.land-5.com. (
               199609206      ; Serial
               28800          ; Refresh
               7200           ; Retry

```

```

6          04800          ; Expire
          86400)         ; Minimum TTL
          NS      land-5.com.
          NS      ns2.psi.net.

;
; Servers
;
1          PTR      router.land-5.com.
2          PTR      land-5.com.
2          PTR      funn.land-5.com.
;
; Workstations
;
200        PTR      ws-177200.land-5.com.
201        PTR      ws-177201.land-5.com.
202        PTR      ws-177202.land-5.com.
203        PTR      ws-177203.land-5.com.
204        PTR      ws-177204.land-5.com.
205        PTR      ws-177205.land-5.com.
; {Many repetitive definitions deleted - SNIP}
250        PTR      ws-177250.land-5.com.
251        PTR      ws-177251.land-5.com.
252        PTR      ws-177252.land-5.com.
253        PTR      ws-177253.land-5.com.
254        PTR      ws-177254.land-5.com.

```

Reverzní zóna je částí konfigurace, která způsobuje nejvíce neštěstí. Používá se k nalezení názvu hostitele, když máte IP-adresu stroje. Příklad: jste IRC-server a přijímáte připojení z IRC-klientů. Jste ale norský IRC-server a chcete přijímat připojení pouze z Norska a dalších skandinávských zemí. Když obdržíte od klienta připojení, knihovna C vám může sdělit IP-adresu připojovaného stroje, protože IP-adresa klienta je obsažena ve všech paketech, posílaných po síti. Nyní můžete volat funkci, nazvanou `gethostbyaddr`, která vyhledá název hostitele podle IP-adresy. `gethostbyaddr` požádá DNS-server, který stroj vyhledá. Předpokládejme, že klient se připojuje z **ws-177200.land-5.com**. IP-adresa, kterou knihovna C předá IRC-serveru, je **206.6.177.200**. Abychom našli název stroje, musíme nalézt **200.177.6.206.in-addr.arpa**. DNS server nejprve nalezne servery `arpa`, potom servery `in-addr.arpa`, pak bude pokračovat obrácenou cestou přes 206, poté 6, až nakonec nalezne server pro zónu `177.6.206.in-addr.arpa` na `land-5`. Odtud získá odpověď, že pro **200.177.6.206.in-addr.arpa** máme záznam „PTR ws-

177200.land-5.com“, což znamená, že název, který odpovídá 206.6.177.200, je **ws-177200.land-5.com**. Stejně jako u příkladu vyhledávání **prep.ai.mit.edu** je toto vysvětlení zjednodušené.

Vraťme se k příkladu IRC-serveru. IRC-server přijímá připojení pouze ze skandinávských zemí (*.no, *.se, *.dk), přičemž název ws-177200.land-5.com zjevně neodpovídá podmínce a server spojení nepovolí. Kdyby zde *nebylo* žádné zpětné mapování 206.6.177.200 přes zónu in-addr.arpa, server by vůbec nebyl schopen nalézt název. Musel by 206.6.177.200 srovnávat s *.no, *.se a *.dk, takže by nic neodpovídalo.

Někteří lidé vám řeknou, že zpětná vyhledávací mapování jsou důležitá pouze pro servery nebo nejsou důležitá vůbec. Ne tak zcela: Množství serverů FTP, news, IRC a dokonce HTTP (WWW) *nepřijme* připojení ze stroje, ke kterému nejsou schopny nalézt název. Takže zpětné mapování je u strojů v podstatě *povinné*.

6 Údržba

Udržujte v chodu

U `named` musíte kromě udržování chodu zajistit ještě jeden aspekt údržby. Jedná se o aktualizace souboru `root.hints`. Nejjednodušším způsobem je použití `dig`, nejprve jej spustíte bez argumentů, získáte `root.hints` podle vlastního serveru. Poté požádejte jeden z vypsaných kořenových serverů pomocí `dig @rootserver`. Zjistíte, že výstup vypadá jako soubor `root.hints`. Uložte jej do souboru (`dig @e.root-servers.net .ns >root.hints.new`) a nahraďte jím starý `root.hints`.

Po nahrazení souboru nezapomeňte znovu spustit `named`.

Al Longyear mi poslal tento skript, kterým se aktualizuje `root.hints`. Skript předpokládá, že máte funkční poštu a že je definován poštovní alias „`hostmaster`“. Aby skript odpovídal vašemu nastavení, musíte jej ještě upravit.

```
#!/bin/sh
#
# Update the nameserver cache information file once per month.
# This is run automatically by a cron entry.
#
(
echo "To: hostmaster <hostmaster>"
echo "From: system <root>"
```

```
echo "Subject: Automatic update of the named.conf file"
echo

export PATH=/sbin:/usr/sbin:/bin:/usr/bin:
cd /var/named

dig @rs.internic.net . ns >root.hints.new

echo "The named.conf file has been updated to contain the following
information:"
echo
cat root.hints.new

chown root.root root.hints.new
chmod 444 root.hints.new
rm -f root.hints.old
mv root.hints root.hints.old
mv root.hints.new root.hints
ndc restart
echo
echo "The nameserver has been restarted to ensure that the update is
complete."
echo "The previous root.hints file is now called

/var/named/root.hints.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Někteří z vás možná zjistili, že soubor `root.hints` je přes FTP k dispozici na Internicu. Prosím, *nepoužívejte* FTP k aktualizaci `root.hints`. Výše popsaná metoda je pro síť daleko přijatelnější.

7 Převod z verze 4 na verzi 8

Toto původně byla část, zabývající se použitím `bind` verze 8, kterou napsal David E. Smith (dave@bureau42.ml.org). Trochu jsem ji upravil, aby odpovídala svému novému názvu.

Není zde moc co dodat. Kromě použití `named.conf` místo `named.boot` je všechno stejné. A `bind8` se dodává s perlovým skriptem, který převádí soubory starých verzí na nové. Ukázkový `named.boot` (stará verze) pro DNS-servery pouze s vyrovnávací pamětí:

```
directory /var/named
cache . root.hints
primary 0.0.127.IN-ADDR.ARPA 127.0.0.zone
primary localhost localhost.zone
```

Na příkazovém řádku zadejte v adresáři `bind8/src/bin/named` (zde se předpokládá, že máte zdrojovou distribuci, jestliže máte binární - skript někde je, ale já nevím kde - ed.):

```
./named-bootconf.pl < named.boot > named.conf
```

Čímž se vytvoří `named.conf`:

```
// generated by named-bootconf.pl

options {
    directory "/var/named";
};

zone "." {
    type hint;
    file "root.hints";
};

zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "127.0.0.zone";
};

zone "localhost" {
    type master;
    file "localhost.zone";
};
```

Funguje to pro všechno, co může jít do souboru `named.conf`, ačkoliv se nepřidávají všechna nová rozšíření a konfigurační volby, které `bind8` umožňuje. Zde následuje úplnější `named.conf`, který provádí to stejné, ale trochu efektivněji.

```
// This is a configuration file for named (from BIND 8.1 or later).
// It would normally be installed as /etc/named.conf.
// The only change made from the 'stock' named.conf (aside from this
// comment :) is that the directory line was uncommented, since I
// already had the zone files in /var/named.

options {
    directory "/var/named";
    check-names master warn; /* default. */
    datasize 20M;
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
    check-names fail;
    allow-update { none; };
    allow-transfer { any; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "127.0.0.zone";
    check-names fail;
    allow-update { none; };
    allow-transfer { any; };
};

zone "." IN {
    type hint;
    file "root.hints";
};
```

Vše je obsaženo v `bind8/src/bin/named/test`, kde jsou i kopie zónových souborů, které můžete ihned využít.

Formáty zónových souborů a souborů `root.hints` jsou shodné, stejně jako příkazy pro jejich aktualizace.

8 FAQ

V této části je seznam některých často pokládaných otázek (FAQ), které se vztahují k DNS a tomuto dokumentu. A jsou zde také odpovědi. Předtím, než mi napíšete, si ještě přečtete tuto část.

1. Můj `named` požaduje soubor `named.boot`.

Čtete špatný dokument. Přečtěte si prosím starou verzi tohoto dokumentu, která pokrývá `bind 4`. Naleznete ji na <http://www.math.uio.no/janl/DNS/>

2. Jak mám DNS používat ze vnitřku firewallu?

Několik rad: „`forwarders`“, „`slave`“ a podívejte se na seznam literatury na konci tohoto dokumentu.

3. Jak DNS přinutit k cycklickému procházení adres, které jsou k dispozici pro nějakou službu (řekněme například `www.busy.site`), abychom získali efekt rovnoměrného nahrávání?

Vytvořte pro `www.busy.site` několik A-záznamů a používejte `bind 4.9.3` nebo pozdější. `Bind` cycklicky obslouží odpovědi. Tohle *nebude* fungovat ve starších verzích `bind`.

4. Chci DNS nastavit na (uzavřeném) intranetu. Co mám dělat?

Zbavte se souboru `root.hints` a vytvořte zónové soubory. Znamená to také, že nemůžete stále tento soubor aktualizovat soubory `hints`.

5. Jak mám nastavit sekundární (podřízený) DNS-server?

Jestliže má primární server adresu `127.0.0.1`, pak na svém sekundárním vložíte do souboru `named.conf` tento řádek:

```
zone "linux.bogus" {
  type slave;
  file "sz/linux.bogus";
  masters { 127.0.0.1; };
};
```

Na seznamu `masters` můžete vypsat několik alternativních hlavních serverů, ze kterých je možné kopírovat zónu. Odděluje je „;“ (středníkem).

6. Chci, aby `bind` běžel, i když jsem ze sítě odpojen.

Od Iana Clarka <ic@deakin.edu.au> jsem obdržel dopis, ve kterém popisuje způsob, který k tomu používá:

Named spouštím na svém „maškarádujícím“ stroji. Mám dva soubory `root.hints` – jeden se jmenuje `root.hints.real` (obsahuje právě názvy kořenových serverů) a druhý se jmenuje `root.hints.fake` a obsahuje...

```
-----
; root.hints.fake
; tento soubor neobsahuje žádné informace
-----
```

Když se odpojuji, kopíruji soubor `root.hints.fake` na `root.hints` a znovu spouštím `named`.

Když se připojuji, kopíruji soubor `root.hints.real` na `root.hints` a znovu spouštím `named`.

Tohle provádím přes `ip-down` (resp. `ip-up`).

Jakmile poprvé požaduji název domény, když nejsem připojen, `named` nemá k dispozici žádné podrobnosti, takže vypustí následující zprávu..

```
Jan 28 20:10:11 hazchem named[10147]: No root nameserver for
class IN
```

s čímž dokáži vyžít.

U mě vše spolehlivě funguje. DNS-server mohou pro lokální stroje využívat bez pauz při neúspěšném vyhledávání externích názvů domén (když nejsem připojen k Internetu). Když jsem připojen k Internetu, vyhledávání probíhá normálně.

7. Kam si DNS-server s vyrovnávací pamětí ukládá vyrovnávací paměť? Je zde nějaký způsob, jakým bych mohl kontrolovat její velikost?

Vyrovnávací paměť je celá uložena v paměti, nikdy se *neukládá* na disk. Kdykoliv ukončíte `named`, vyrovnávací paměť je ztracena. Vyrovnávací paměť *není* možné žádným způsobem kontrolovat. `Named` ji spravuje podle určitých jednoduchých pravidel a tak to má být. Vyrovnávací paměť nebo její velikost nemůžete z žádného důvodu ovládat. Jestli k tomu máte přesto chuť, musíte si programově upravit celý `named`. To samozřejmě příliš nedoporučujeme.

8. Ukládá `named` svoji vyrovnávací paměť mezi jednotlivými restarty? Mohu ho přinutit, aby ji uložil?

Ne, `named` při svém zániku vyrovnávací paměť *neukládá*. To znamená, že vyrovnávací paměť se musí při každém restartu znovu vytvořit. Neexistuje *žádný* způsob, jak `named` přinutit uložit svoji vyrovnávací paměť do souboru. Jestli k tomu máte přesto chuť, musíte si opět upravit celý `named`. To ale samozřejmě příliš nedoporučujeme.

9 Jak se stát lepším správcem DNS

Dokumentace a nástroje

Vhodná dokumentace existuje. Na síti i v tištěné podobě. K tomu, aby se z obyčejného správce DNS stal správce lepší, je nutné si část této dokumentace pročíst. Tištěná dokumentace, to je zejména kniha *DNS and BIND* od C. Liu a P. Albitze, vydaná v O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. Četl jsem ji, je skvělá. O DNS pojednává také část knihy *TCP/IP Network Administration* od Craiga Hunta, vydaná v O'Reilly..., ISBN 0-937175-82-X. Další nutnou koupí pro dobrou správu DNS (nebo spíše něčeho jiného) je *Zen and the Art of Motorcycle Maintenance* od Roberta M. Prisiga :-). K dispozici pod ISBN 0688052304 a další.

Na síti naleznete další zajímavosti na <http://www.dns.net/dnsrd/>, <http://www.isc.org/bind.html>; jsou to FAQ, referenční příručka (BOG) a definice protokolů spolu s úpravami DNS (většinu z toho, plus některá níže zmíněná RFC jsou obsažena v distribuci `bindu`). Většinu z toho jsem nečetl, ale já také nejsem žádný skvělý správce. Na druhou stranu Arnt Gulbrandsen četl BOG a líbilo se mu to :-). K DNS se vztahuje skupina `news – comp.protocols.tcp-ip.domains`. Navíc k DNS existuje množství RFC, z nichž nejdůležitější jsou pravděpodobně tato:

RFC 2052

A. Gulbrandsen, P. Vixie, *A DNS RR for specifying the location of services (DNS SRV)*, October 1996

RFC 1918

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear,

Address Allocation for Private Internets, 02/29/1996.

RFC 1912

D. Barr, *Common DNS Operational and Configuration Errors*, 02/28/1996.

RFC 1912 Errors

B. Barr, *Errors in RFC 1912*, k dispozici také na <http://www.cis.ohio-state.edu/~barr-rr/rfc1912-errors.html>

RFC 1713

A. Romao, *Tools for DNS debugging*, 11/03/1994.

RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 11/01/1994.

RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 10/08/1990.

RFC 1035

P. Mockapetris, *Domain names – implementation and specification*, 11/01/1987.

RFC 1034

P. Mockapetris, *Domain names – concepts and facilities*, 11/01/1987.

RFC 1033

M. Lottor, *Domain administrators operations guide*, 11/01/1987.

RFC 1032

M. Stahl, *Domain administrators guide*, 11/01/1987.

RFC 974

C. Partridge, *Mail routing and the domain system*, 01/01/1986.

Jádro Linuxu

Brian Ward, bri@blah.math.tu-graz.ac.at verze 0.80, 26. května 1997

Podrobný průvodce ke konfiguraci, kompilaci, aktualizaci a odstraňování problémů u jádra na systémech, založených na procesorech ix86.

1 Úvod

Měli byste vůbec číst tento dokument? Dobrá, tak si projděte následující příklady a zjistěte, jestli vám alespoň jeden neodpovídá:

- „Hanba! Tenhle wizzo-46.5.6 tvrdí, že potřebuje jádro verze 1.8.193 a já mám stále jen verzi 1.0.9!“
- V jednom z novějších jader je ovladač zařízení, které se chystáte připojit.
- Nevíte, jak se kompiluje jádro.
- „Je to, co se píše v README, *opravdu* všechno?“
- Zkusili jste to, ale nefunguje to.
- Musíte nějak uspokojit lidi, kteří po vás žádají instalaci jádra.

1.1 Nejprve si přečtete tohle! (myslím to vážně)

Některé z příkladů tohoto dokumentu předpokládají, že máte GNU `tar`, `find` a `xargs`. To je standard; zde by neměly nastat problémy. Předpokládám také, že na svém systému znáte strukturu systému souborů; pokud neznáte, měli byste si zapsat na papír výstup příkazu `mount` během normální funkce systému (nebo výpis `/etc/fstab`, jestliže jej můžete číst). Tato informace je důležitá a pokud nerozdělíte jinak disk, nepřidáte další, nepřeinstalujete systém nebo něco podobného, tak se nemění.

Poslední verzi jádra byla v době psaní dokumentu verze 2.0.30, což znamená, že odkazy a příklady se vztahují právě k ní. I když se snažím dokument pojmut nezávisle na verzích, jádro se stále vyvíjí, takže pokud budete mít nějaké novější, mohou se objevit odlišnosti. Větší problémy by se objevit neměly, ale k určitému zmatení by dojít mohlo.

Existují dvě verze zdrojových textů jádra Linuxu - „produkční“ a „vývojová“. Produkční verze začínají u 1.0.x a jsou číslovány sudě; 1.0.x je produkční, 1.2.x je produkční, stejně jako 2.0.x. Tato jádra jsou v době svého vydání považována za nejstabilnější a bez chyb. Vývojová jádra (1.1.x, 1.1.3 atd.) jsou považována za zkušební, určená lidem, kteří by je mohli testovat a odhalovat možné chyby. Takže varování jste již byli.

1.2 Poznámka ke stylu

Text, který vypadá takto, je buď něco, co se objeví na vaší obrazovce, název souboru nebo něco, co je možné přímo zadat (příkazy, volby). Jestliže si ale tento text prohlížíte v čistě textové podobě, rozdíl samozřejmě nepoznáte. Příkazy a další vstupy jsou často v uvozovkách („“), přičemž zde se objevuje klasický problém s tečkou. Když se položka v uvozovkách objeví na konci věty, v Americe se tečka píše ještě před poslední apostrof. Pokud zde tedy budu chtít naznačit, abyste napsali `„make config“`, napíší `„make config“`, ne `„make config.“`

2 Důležité otázky a odpovědi

2.1 Mimochodem, co vlastně jádro dělá?

Unixové jádro funguje jako prostředník mezi vašimi programy a hardwarem. Zejména provádí správu paměti všech běžících programů (procesů) a zaručuje, že všechny dostanou patřičný (nebo nepatřičný) podíl cyklu procesoru. Navíc poskytuje krásné, snadno propojitelné rozhraní programů pro komunikaci s hardwarem.

K operacím jádra náleží samozřejmě mnohem více, ale tyto základní funkce jsou nejdůležitější, které je potřeba znát.

2.2 Proč bych měl chtít aktualizovat jádro?

Novější jádra obecně nabízí schopnost komunikace s větším množstvím typů hardwaru (to znamená, že mají více ovladačů zařízení), mohou mít lepší správu procesů, mohou být rychlejší, stabilnější a upravují chyby starých verzí jader. Většinou lidí stačí jako důvod pro aktualizaci nové ovladače zařízení a nápravu chyb.

2.3 Jaký druh hardwaru novější jádra podporují?

Viz dokument o hardware. Kromě toho můžete nahlédnout do souboru „`config.in`“ ve zdrojovém textu Linuxu nebo si prostě vyzkoušíte „`make config`“. Tak se zobrazí veškerý hardware, podporovaný standardní distribucí jádra, ale ne všechny, který podporuje Linux; množství běžných ovladačů zařízení (například PCMCIA) je dodáváno odděleně ve formě modulů, určených k nahrání.

2.4 Jakou verzi gcc a libc potřebuji?

Verzi gcc doporučuje Linus v souboru `README`, dodávaném s linuxovým zdrojovým textem. Jestliže tuto verzi nemáte, dokumentace v doporučené verzi gcc by vám měla sdělit, jestli potřebujete aktualizovat libc. Postup není obtížný, ale je důležité přesně sledovat instrukce.

2.5 Co je to modul, určený k nahrání?

Existují části kódu jádra, které nejsou přilinkovány (obsaženy) přímo v jádru. Je možné je kompilovat odděleně a téměř kdykoliv je přidat nebo odstranit z běžícího jádra. Vzhledem k pružnosti se nyní jedná o upřednostňovaný způsob kódování určitých funkcí jádra. Mezi moduly, určenými k nahrání, nalezneme mnohé oblíbené ovladače zařízení, jako jsou ovladače PCMCIA a ovladače pásek QIC-80/40.

2.6 Kolik potřebuji místa na disku?

To závisí na vaší konkrétní systémové konfiguraci. Komprimovaný zdrojový text linuxu zabírá ve verzi 2.0.10 téměř 6 MB. Některé servery si jej uchovávají i v nezkomprimované podobě, která zabírá 24 MB. Ale to není vše – ke kompilaci potřebujete mnohem více. To závisí na tom, kolik toho nakonfigurujete do svého jádra. Já mám například na jednom stroji nakonfigurovanou podporu sítě, ovladač 3Com 3C509 a tři systémy souborů, což využívá 30 MB.

Na přidání komprimovaného linuxového zdrojového textu potřebujete při takové konfiguraci 36 MB. Na jiném systému bez podpory síťového adaptéru (ale s ponechanou podporou sítě) a navíc s podporou zvukové karty se požadované místo ještě zvětšuje. Nové jádro také vždy zabírá více místa než staré, takže pokud máte dostatečné možnosti v hardwaru, zajistěte si dostatek diskové kapacity (při dnešních cenách se vyplatí i mít i větší rezervu).

2.7 Jak dlouho to potrvá?

Pro většinu lidí je odpověď „dost dlouho“. Rychlost vašeho systému a množství paměti, které máte, rychlost přímo určí, ale ještě zde může mít menší vliv množství položek, konfigurovaných v jádru. Na 486DX4/100 s 16 MB RAM s jádrem v1.2 s pěti systémy souborů, podporou sítě a ovladači zvukových karet to zabere kolem 20 minut. Na 386DX/40 s 8 MB RAM a při podobné konfiguraci to bude téměř 1,5 hodiny. Zde se doporučuje trocha kávy, televize, plečení nebo jakákoliv jiná zábava, prováděná po dobu kompilace jádra. Jestliže máte opravdu pomalý stroj, můžete si jádro nechat kompilovat u někoho jiného (s rychlým strojem).

3 Jak skutečně nakonfigurovat jádro

3.1 Získání zdrojového textu

Zdrojový text je možné získat na anonymním ftp-serveru **ftp.funet.fi** v adresáři `/pub/Linux/PEOPLE/Linus`, na jeho mirrorech nebo jiných serverech. Označení je většinou `linux-x.y.z.tar.gz`, kde `x.y.z` je číslo verze. Novější (lepší?) verze a patche jsou v podadresářích, jako je „v1.1“ a „v1.2“. Nejvyšší číslo znamená poslední verzi a obvykle se jedná o testovací, takže pokud vás děsí alfa a beta verze, držte se raději těch základních.

Místo **ftp.funet.fi*** *silně* doporučuji mirrory. Zde je krátký seznam těch základních:

USA: `sunsite.unc.edu:/pub/Linux/kernel`

USA: `tsx-11.mit.edu:/pub/linux/sources/system`

Velká Británie: `sunsite.doc.ic.ac.uk:/pub/unix/Linux/sunsite.unc-mirror/kernel`

Rakousko: `ftp.univie.ac.at:/systems/linux/sunsite/kernel`

Německo: `ftp.Germany.EU.net:/pub/os/Linux/Local.EU.net/Kernel/Linus`

* Poznámka korektora: Primárním zdrojem pro zdrojové texty operačního systému Linux je nyní server `ftp.kernel.org`.

Německo: `sunsite.informatik.rwth-aachen.de:/pub/Linux/PEOPLE/Linus`
 Francie: `ftp.ibp.fr:/pub/linux/sources/system/patches`
 Austrálie: `sunsite.anu.edu.au:/pub/linux/kernel`

Obecně bývá nejlepším místem mirror **sunsite.unc.edu**. Soubor `/pub/Linux/MIRRORS` obsahuje seznam jeho známých mirrorů. Jestliže nemáte možnost používat ftp, na `comp.os.linux.announce` je pravidelně zasílán seznam systémů BBS, kde je možné Linux získat.

Jestliže hledáte obecné informace o Linuxu a jeho distribuci, je zde <http://www.linux.org>.

3.2 Rozbalení zdrojového textu

Přihlaste se jako „**root**“ (nebo použijte **su**) a proveďte `cd /usr/src`. Jestliže jste instalovali zdrojový text jádra již při instalaci Linuxu (což je běžné), měl by zde být adresář „`linux`“, který obsahuje celý starý zdrojový strom. Jestliže máte dostatek místa na disku a chcete postupovat bezpečně, tento adresář zachovejte. Dobrý nápad je zjistit, kterou verzi váš systém nyní používá, a podle toho adresář přejmenovat. Příkaz „`uname -r`“ vypíše verzi aktuálního jádra. Proto pokud „`uname -r`“ hlásí „**1.0.9**“, přejmenujete (pomocí „`mv`“) „`linux`“ na „`linux-1.0.9`“. Jestliže se nehodláte obtěžovat, prostě zrušte celý adresář. V každém případě nesmí být v `/usr/src` před rozbalením plného zdrojového kódu žádný adresář „`linux`“.

Nyní v `/usr/src` rozbalte zdrojový text pomocí „`tar xzpvf linux-x.y.z.tar.gz`“ (jestliže máte jen soubor `.tar` bez koncovky `.gz`, stačí „`tar xpvf linux.x.y.z.tar`“). Obsah archivního souboru se rozbalí. Poté se v `/usr/src` objeví nový adresář „`linux`“. Proveďte `cd linux` a projděte si soubor `README`. Měla by zde být část s nadpisem „`INSTALLING the kernel`“. Proveďte příslušné instrukce – vytvoření symbolických odkazů*, odstranění souborů `.o` atd.

3.3 Konfigurace jádra

Poznámka: Část je zde jen opakováním podobné části ze souboru „`README`“.

Příkaz „`make config`“, provedený v `/usr/src/linux` spustí skript, který vám položí mnohé otázky. Požaduje `bash`, takže ověřte, že `bash` je `/bin/bash`, `/bin/sh` nebo `$BASH`.

K „`make config`“ existují určité alternativy a vám se mohou jevit i snazší a pohodlnější. Pro ty, kteří používají X-Window System, je zde „`make xconfig`“ (máte-li instalováno Tk). „`make menuconfig`“ je pro ty, co nemají rádi ano/ne a raději používají textová menu. Tato rozhraní mají jednu výhodu – pokud se při volbě zmýlíte, není problém se vrátit a opravit.

* Poznámka korektora: Novější verze jádra Linuxu již tyto odkazy vytvářejí samy.

Připravte se na odpovědi na otázky, většinou „y“ (ano) nebo „n“ (ne). Ovladače zařízení mají většinou volbu „m“. Znamená to „modul“, což znamená kompilaci ne do jádra, ale jako modul, určený k nahrání. Trochu méně vážné vysvětlení pak zní jako „možná“. Některé jasnější a méně kritické volby se zde nepopisují; stručný popis několika dalších viz část „Další konfigurační volby“.

V 2.0.x a pozdějších je volba „?“ , která nabízí stručný popis konfiguračních parametrů. Tato informace je nejaktuálnější.

3.3.1 Matematická emulace v jádru

Jestliže nemáte matematický koprocesor (máte samotné 386 nebo 486SX), musíte zde odpovědět „y“. Pokud koprocesor máte a odpovíte „y“, příliš se nevzrušujte – koprocesor je stejně využíván a emulace ignorována. Jediný následek je zvětšení jádra (obsadí více RAM). Slyšel jsem, že emulace je pomalá; ačkoliv to by nám zde až tak vadit nemuselo, to se projeví až ve špatných výkonech X-Window System.

3.3.2 Podpora normálního (MFM/RLL) disku a IDE disku/cdrom

Pravděpodobně je potřeбуйте podporovat; to znamená, že jádro bude podporovat standardní PC pevné disky, které má většina lidí. Tento ovladač nezahrnuje ovladače SCSI; ty přijdou v konfiguraci později.

Poté budete dotázáni „old disk-only“ (pouze staré disky) a „new IDE drivers“ (nové IDE ovladače). Vyberete si jeden z nich; hlavní rozdíl je v tom, že starý ovladač podporuje pouze dva disky na jednom rozhraní, zatímco nový podporuje druhotné rozhraní a IDE/ATAPI cdrom mechaniky. Nový ovladač zabírá o 4 KB více než starý a je také trochu „vylepšen“ – to znamená, že kromě jiného množství chyb, které obsahuje, může zvýšit vaše výkony, obzvláště pokud vlastníte novější hardware (EIDE).

3.3.3 Podpora sítě

Jestliže je váš stroj na síti, jako je Internet, nebo chcete pro přístup přes modem používat SLIP, PPP, term..., tak v podstatě stačí odpovědět „y“. Ale množství balíků (například systém X-Window System) vyžaduje podporu sítě, i když na opravdové síti nejste, takže stejně musíte říci „y“. Později budete dotázáni, jestli chcete podporovat TCP/IP síť; opět odpovězte „y“, nejste-li si zcela jisti.

3.3.4 Omezení paměti pod 16 MB

Existují chybné ovladače 386 DMA, které mají problémy s adresací čehokoliv nad 16 MB RAM; jestliže jste tento (vzácný) případ, musíte odpovědět „y“.

3.3.5 System V IPC

Jednu z nejlepších definicí IPC (meziprocesorová komunikace) nalezneme v perlovém slovníku. Pak není překvapivé, že programátoři v Perlu mnoho balíčků (nevyjímaje například DOOM) takto nechávají procesy hovořit mezi sebou. Takže zde odpovídat „n“ není zrovna moudré, pokud ovšem přesně nevíte, co děláte.

3.3.6 Typ procesoru (386, 486, Pentium, PPro)

(ve starších jádrech: optimalizace pro 486 využijete s přepínačem -m486)

Většinou se zde kompilace prováděla s optimalizací pro určitý procesor; jádra běžela i na jiných procesorech, ale byla o něco větší. V novějších jádrech už to tak není, takže byste měli zadat procesor, pro který kompilaci provádíte. Jádro „386“ bude fungovat na všech strojích.

3.3.7 Podpora SCSI

Jestliže máte SCSI zařízení, odpovězte „y“. Budete požádáni o další informace, jako je podpora CD-ROM, disků a druh použitého SCSI adaptéru. Více podrobností viz dokument SCSI-HOWTO.

3.3.8 Podpora síťových zařízení

Jestliže máte síťovou kartu nebo byste rádi pro připojení k Internetu použili SLIP, PPP nebo adaptér na paralelní port, odpovězte „y“. Konfigurační skript se vás dotáže na druh karty a použitý protokol.

3.3.9 Systémy souborů

Konfigurační skript se vás pak zeptá, jestli si přejete podporovat následující systémy souborů:

Standard (minix) - novější distribuce systémy souborů minix nevytváří a množství lidí je nevyužívá, ale přesto se jejich nakonfigurování může hodit. Využívají je některé programy na „záchranu disku“ a dále množství disket, pro které je tento systém výhodnější.

Extended fs – tohle byla první verze rozšířeného systému souborů, který se ale už moc nepoužívá. Jestliže jej znáte, možná jej budete potřebovat, jestliže nevíte, tak ho ani nepotřebujete.

Second extended – V nových distribucích se používá masově. Jednu z nich pravděpodobně máte a musíte odpovědět „y“.

xiafs – svého času nebyl neznámý, ale dnes neznám nikoho, kdo by jej používal.

msdos – jestliže hodláte používat svoje dosové disky nebo diskety, odpovězte „y“.

umsdos – tento systém rozšiřuje systém souborů MS-DOS o obvyklé unixové věci, jako jsou dlouhé názvy souborů. Není nutný pro lidi (jako jsem já), kteří „nedělají v MS-DOSu“.

`/proc` – další z velkých věcí od dob sušeného mléka (myšlenka byla podle mého předpokladu hanebně ukradena z Bellových laboratoří). Nejedná se o systém souborů na disku; je to systém souborů jako rozhraní mezi jádrem a procesy. Využívají jej mnohé vypisovače procesů (jako je „ps“). Někdy si zkuste „`cat /proc/meminfo`“ nebo „`cat /proc/devices`“. Některé příkazové interprety (jmenovitě `rc`) používají pro I/O `/proc/self/fd` (na jiných systémech známý jako `/dev/fd`). Zde musíte skoro jistě odpovědět „y“; závisí na tom mnoho důležitých nástrojů Linuxu.

NFS – jestliže váš stroj žije na síti a vy chcete využívat systémy souborů z jiných systémů pod NFS, odpovězte „y“.

ISO9660 – je na většině CD-ROM. Jestliže máte mechaniku CD-ROM a chcete ji pod Linu-
xem používat, odpovězte „y“.

OS/2 HPFS – v době psaní tohoto dokumentu se jedná o systém souborů pro samotné čtení OS/2 HPFS.

System V and Coherent – System V a Coherent jsou další varianty na PC Unix.

Ale já nevím, které systémy souborů potřebuji! Dobrá, zadejte „`mount`“. Výstup by měl vypadat asi takto:

```
blah# mount
/dev/hda1 on / type ext2 (defaults)
/dev/hda3 on /usr type ext2 (defaults)
none on /proc type proc (defaults)
/dev/fd0 on /mnt type msdos (defaults)
```

Podívejte se na každý řádek; slovo vedle „**type**“ je typ systému souborů. V tomto příkladu jsou `múj` / `a /usr` systém souborů typu second extended, využívám `/proc` a je zde připojena disketová mechanika, využívající dosovský systém souborů.

Jestliže máte právě nastaveno `/proc`, můžete zkusit „`cat /proc/filesystems`“; vypíše se systémy souborů, aktuálně ve vašem jádru používané.

Nakonfigurováním zřídka kdy používaných, nekritických systémů souborů můžete jádro zahl-
tit; jak se tomu vyhnout zjistíte níže, v části o modulech, a později se také dozvíte, proč je za-
hlcené jádro nežádoucí.

3.3.10 Znaková zařízení

Zde se nastavují ovladače pro tiskárnu (paralelní), busmouse, myš PS/2 (množství notebooků používá tento protokol pro svoje trackbally), některé páskové ovladače a další podobná „znaková“ zařízení. Když je to nutné, odpovězte „y“.

Poznámka: Selection je program, který umožňuje využití myši mimo X-Window System k operacím cut&paste mezi virtuálními konzolami. Je sice pěkné, že máte sériovou myš, protože v X-Window System funguje dobře, ale ještě je nutné umožnit provádění určitých speciálních triků. Kdysi byla podpora selection volitelná, dnes je již standardní.

Poznámka 2: Selection se nyní považuje za zastaralý. Novější program má název „gpm“. Dokáže více věcí, jako je překlad protokolů myši, obsluhu více myší...

3.3.11 Zvuková karta

Jestli silně toužíte po zvuku, odpovězte „y“ a později se vás další konfigurační program zeptá na všechno o vaší zvukové kartě. Poznámka ke konfiguraci zvukové karty: když máte rozhodnout, jestli se má instalovat plná verze ovladače, odpovězte „n“ a ušetříte část paměti jádra tím, že vyberete jen to, co opravdu potřebujete. Jestliže máte zvukovou kartu, doporučuji vám k proctení dokument Sound-HOWTO.

3.3.12 Další konfigurační volby

Všechny konfigurační volby zde vypsány nejsou, protože se velmi často mění nebo mají význam zcela zřejmý již podle názvu (například 3Com 3C509 support je podpora ovladače dané ethernetové karty). Podrobný seznam všech voleb (a způsob jejich začlenění ve skriptu `Configure`) od Axela Boldta (axel@uni-paderborn.de) je k mání na adrese s následujícím URL:

http://math-www.uni-paderborn.de/~axel/config_help.html

nebo přes anonymní FTP na:

`ftp://sunsite.unc.edu/pub/Linux/kernel/config/krnl_cnfg_hlp.x.yz.tgz`

kde `x.yz` je číslo verze.

Pro pozdější jádra (od verze 2.0.x) je to začleněno do zdrojového stromu.

3.3.13 Úprava jádra

V README Linus píše:

Některé konfigurační volby (označené jako „kernel hacking“) mají obvykle za následek větší nebo pomalejší jádro (nebo obojí) a mohou zvýšit nestabilitu jádra – povolením některých rutin, určených k odstraňování problémových částí kódu jádra (kmalloc()). U „produkčního“ jádra je tedy lepší na tyto dotazy odpovědět „n“.

3.4 Co teď? (soubor Makefile)

Po provedení `make config` vám hlášení potvrdí ukončenou konfiguraci jádra, doporučí kontrolu nejvyššího Makefile s další konfigurací atd.

Takže se podívejte do **Makefile**. Nezměníte zde pravděpodobně nic, ale za podívání nic nedáte. Jakmile je nové jádro připraveno, můžete jeho volby změnit pomocí příkazu „`rdev`“.

4 Kompilace jádra

4.1 Čištění a závislosti

Jakmile se ukončí konfigurační skript, doporučí vám také „`make dep`“ a (možná) „`clean`“. Takže proveďte „`make dep`“. Tím zajistíte, že všechny závislosti, budou registrovány. Nemělo by to trvat dlouho, pokud nemáte opravdu pomalý počítač. U starších verzí jádra musíte po ukončení provést „`make clean`“. Tím se odstraní všechny objektové soubory a některé další věci, které stará verze po sobě zanechává. V každém případě na tento krok *nezapomeňte*, pokud se chystáte jádro překompilovat.

4.2 Doba kompilace

Po `dep` a `clean` můžete provést „`make zImage`“ nebo „`make zdisk`“ (tohle je ta část, která trvá tak dlouho). Jádro se kompiluje pomocí „`make zImage`“. V `arch/i386/boot` zůstane soubor „`zImage`“ (mimo jiné). Tohle je nové komprimované jádro. Stejnou věc provádí „`make zdisk`“, ale nový `zImage` ukládá na disketu, kterou musíte mít v mechanice „A:“. Volba „`zdisk`“ je vhodná pro testování nových jader; jestliže dojde k nějakému selhání, prostě vyjměte disketu z mechaniky a proveďte restart se starým jádrem. Jedná se také o šikovný způsob zavedení systému, pokud nechtěně smažete jádro (nebo něco stejně nepatřičného). Disketu můžete také využít při instalaci systému u někoho jiného – prostě mu předáte její obsah.

Všechna sudá pozdější jádra jsou komprimována (proto „z“ před jmény. Komprimované jádro se při spuštění automaticky dekomprimuje.

4.3 Další možnosti pro make

Volba „`make mrproper`“ provede mnohem rozsáhlejší „`clean`“. Někdy je nezbytná; můžete ji provést při každé úpravě. Tato volba také smaže váš konfigurační soubor, takže si jej (`.config`) můžete zálohovat, může-li být nějak užitečný.

Volba „`make oldconfig`“ se pokusí jádro konfigurovat podle starého konfiguračního souboru; který použije během procesu „`make config`“. Jestliže jste před tím ještě nikdy nekompilovali jádro nebo nemáte starý konfigurační soubor, nepoužívejte tuto volbu, protože byste se připravili o možnost úpravy implicitní konfigurace.

O „`make modules`“ pojednává část o modulech.

4.4 Instalace jádra

Jakmile máte nové jádro, které vypadá, že bude fungovat požadovaným způsobem, nastal čas jej nainstalovat. Většina lidí k tomu užívá LILO (Linux Loader). Volba „`make zlilo`“ nainstaluje jádro, spustí na něm LILO a připraví vás na zavedení systému, ALE POUZE v případě, že LILO je na vašem systému nakonfigurováno následujícím způsobem: jádro je `/vmlinuz`, LILO je v `sbin` a vaše lilo konfigurace LILO (`/etc/lilo.conf`) s tím souhlasí.

Jinak musíte LILO použít přímo. Tento balík se velice snadno instaluje a ovládá, ale často máte uživatele svým konfiguračním souborem. Podívejte se do něj (buď `/etc/lilo/config` u starších verzí nebo `/etc/lilo.conf` u novějších) a zjistěte aktuální nastavení. Konfigurační soubor vypadá asi takto

```
image = /vmlinuz
    label = Linux
    root = /dev/hda1
    ...
```

Řádek „`image =`“ je nastaven na aktuálně instalované jádro. Většina lidí používá `/vmlinuz`. Řádek „`label`“ se pro LILO využívá k určení, které jádro nebo operační systém se má zavádět, a „`root`“ je kořenovým adresářem příslušného operačního systému. Vytvořte záložní kopii vašeho starého jádra a okopírujte zImage, který jste právě vyrobili, na správné místo (jestliže používáte „`/vmlinuz`“, napíšete „`cp zImage /vmlinuz`“). Potom znovu spustíte LILO – na novějších systémech stačí spustit „`lilo`“, ale na starších budete možná muset provést `/etc/lilo/install` nebo dokonce `/etc/lilo/lilo -C /etc/lilo/config`.

Jestliže se chcete o konfigurování LILO dozvědět více nebo jestliže LILO nemáte, sežeňte si z vašeho oblíbeného ftp-serveru nejnovější verzi a řiďte se instrukcemi.

Při zavádění systému z jiného místa než obvykle (další způsob záchrany při pokažení nového jádra) okopírujte v konfiguračním souboru pro LILO řádky pod řádkem „**image = xxx**“ (a včetně) a změňte „**image = xxx**“ na „**image = yyy**“, kde „**yyy**“ je plná cesta k souboru, který jste uložili jako záložní jádro. Poté změňte „**label = zzz**“ na „**label = linux-backup**“ a znovu spusťte `lilo`. Do konfiguračního souboru možná ještě vložíte řádek „**delay=x**“, kde `x` je čas v setinách sekundy, který pro LILO určuje, jak dlouho má před zavedením systému čekat, abyste mohli zavedení ještě přerušit (například klávesou **Shift**) a zadat označení záložního obrazu jádra (kdyby například nebylo všechno v pořádku).

5 Úprava jádra

5.1 Aplikace záplat (patch)

Postupné aktualizace jádra se dodávají jako záplaty (patch). Jestliže máte například verzi 1.1.45 a existuje pro ni záplata „`patch46.gz`“, můžete aplikaci této záplaty přejít na verzi 1.1.46. Nejprve si můžete vytvořit záložní kopii zdrojového stromu („`make clean`“ a poté „`cd /usr/src; tar zcvf old-tree.tar.gz linux`“ vám vytvoří komprimovaný archiv `tar`).

Takže když budeme pokračovat v předchozím příkladu, předpokládejme, že „`patch46.gz`“ máte v `/usr/src`. Proveďte `cd /usr/src a „zcat patch46.gz | patch -p0“` (nebo „`patch -p0 < patch46`“, není-li záplata komprimována). Všechno vám pak na obrazovce proběhne (nebo projde, pokud máte pomalý systém) a sdělí, že se o něco úspěšně nebo neúspěšně pokouší. Většinou všechno proběhne příliš rychle, než aby se to dalo vnímat, takže použijte k `patch` přepínač `-s`, který zredukuje hlášení pouze na chybová (sice nebudete vnímat změny, prováděné vaším počítačem, ale možná je to tak lepší). Při hledání částí, které by nemusely proběhnout v pořádku, proveďte `cd /usr/src/linux` a zaměřte se na soubory s koncovkou `.rej`. Některé verze `patch` používají koncovku `#`. K vyhledání můžete využít příkaz „`find`“;

```
find . -name '*.rej' -print
```

Na standardní výstup se vytisknou všechny soubory s koncovkou `.rej`, které jsou v aktuálním adresáři nebo jeho podadresářích.

Jestliže vše proběhlo v pořádku, proveďte podle 3. a 4. části „`make clean`“, „`config`“ a „`dep`“.

Příkaz `patch` má několik voleb. Jak jsme již naznačili, `patch -s` odfiltruje všechna hlášení kromě chybových. Jestliže máte zdrojové texty jádra na jiném místě než je

`/usr/src/linux, patch -p1` (v požadovaném adresáři) se provede bez problémů. Další volby `patch` jsou přesně dokumentovány na manuálové stránce.

5.2 Jestliže je něco špatně

(Poznámka: Tato část se týká zejména starých jader.)

Nejčastějším problémem se objevil, když `patch` upravil soubor „`config.in`“, ale ne správně, protože jste si jej předtím upravili podle svého stroje. Dnes je již tento problém vyřešen, ale u starších verzí se s ním ještě můžete setkat. Nápravu provádíte po nahlédnutí do souboru `config.in.rej` a zjištění, co zbylo z původní podoby. Změny jsou většinou na začátku řádku označeny „`+`“ a „`-`“. Podívejte se na okolní řádky a vzpomeňte si, jestli byly nastaveny na „`y`“ nebo „`n`“. Nyní editujte `config.in` a na příslušných místech změňte „`y`“ na „`n`“ a obráceně. Proveďte

```
patch -p0 < config.in.rej
```

a jestliže se příkaz provede bez chyb, můžete v konfigurování a kompilaci pokračovat. Soubor `config.in.rej` vám zůstane, ale můžete ho smazat.

Jestli se objeví další problémy, může to být díky tomu, že máte špatnou verzi záplaty. Jestliže se objeví „`previously applied patch detected: Assume -R?`“, snažte se pravděpodobně provést starší úpravu na novější verzi, než právě máte; jestliže přesto odpovíte „`y`“, dojde k pokusu o degradaci vašeho zdrojového textu, která většinou neprojde. Nezbude vám než si opatřit novější zdrojový strom.

Vrácení záplaty provedete z původní úpravy pomocí „`patch -R`“.

Nejvhodnější postup při selhání záplat je začít úplně znovu s čistým zdrojovým stromem (například z jednoho ze souborů `linux-x.y.z.tar.gz`).

5.3 Odstranění souborů `.orig`

Po pár úpravách se začnou hromadit soubory `.orig`. Například strom 1.1.51 byl naposledy promazán v dobách 1.1.48. Odstraněním souborů `.orig` jsem získal více než 0,5 MB volného místa.

```
find . -name '*.orig' -exec rm -f {} ';'
```

by mělo potřebné promazání provést. Verze `patch`, které pro `.rej` používají koncovku `#`, používají místo `.orig` tildu (`~`). Existují ale i lepší způsoby smazání souborů `.orig`, které závisí na GNU `xargs`:

```
find . -name '*.orig' | xargs rm
```

nebo ještě „jistá, ale trochu delší“ metoda:

```
find . -name '*.orig' -print0 | xargs --null rm --
```

5.4 Další záplaty

Existují další záplaty (říkám jim „nestandardní“), odlišné od záplat, distribuovaných Linusem. Jestliže je využijete, nemusí fungovat správně. Potom je musíte vrátit, opravit zdrojový text nebo záplatu, instalovat nový zdrojový text nebo tohle všechno současně. Tohle vás může dost potrápit, takže pokud nechcete všechno opravovat sami, vraťte záplaty zpět nebo instalujte nový strom. Pak se teprve podívejte, jestli vám nestandardní záplaty fungují. Jestliže nefungují, máte možnost buď laborovat se starým jádrem, nebo si počkejte na novou verzi záplaty.

Jak běžné jsou záplaty nedodávané standardně? Pravděpodobně o nich uslyšíte. Jednu jsem používal pro svoje virtuální konzoly, protože nemám rád blikající kurzory (tato úprava byla zatím pro nové verze jádra vždy obnovována). Protože se ale nové ovladače zařízení vyvíjí jako moduly, snižuje se i rozsah použití „nestandardních“ záplat.

6 Dodatečné balíky

Vaše linuxové jádro má množství funkcí, které nejsou vysvětleny v samotném zdrojovém textu jádra; tyto funkce se většinou zajišťují pomocí externích balíčků. Nyní následuje výpis těch nejběžnějších.

6.1 kbd

Linuxová konzola má pravděpodobně více funkcí než kolik by potřebovala. Mimo jiné je zde možnost přepínat fonty, přemapovat vaši klávesnici, přepínat videomódy (v novějších jádrech) atd. Balík `kbd` obsahuje programy, které uživatelům tohle všechno zpřístupní, a navíc mnoho fontů a rozložení kláves pro téměř jakoukoliv klávesnici. Balík je k dispozici na stejných místech jako zdrojové texty jádra.

6.2 util-linux

Rik Faith (faith@cs.unc.edu) složil velkou sbírku linuxových utilit, která se nazývá `util-linux`. Nyní ji udržuje Nicolai Langfeldt (util-linux@math.uio.no). K dispozici je přes anonymní FTP ze sunsite.unc.edu v `/pub/Linux/system/misc`. Obsahuje programy, jako jsou `setterm`, `rdev` a `ctrlaltdel`, které souvisí s jádrem. Jak říká Rik, *neinstalujte bezmyšlenkovitě*. Nepotřebujete instalovat celý balík, mohli byste si způsobit vážné problémy.

6.3 hdparm

Stejně jako u mnoha jiných balíků se i zde původně jednalo o záplatu jádra a podpůrné programy. Záplaty se dostaly do oficiálního jádra a programy na optimalizaci a různé nastavování vašeho pevného disku jsou dodávány odděleně.

6.4 gpm

Gpm je zkratkou pro víceúčelovou myš (General Purpose Mouse). Tento program umožňuje přenos textu mezi virtuálními konzolami (pomocí cut&paste) a další věci, prováděné různými typy myší.

7 Některá úskalí

7.1 make clean

Jestliže vaše nové jádro provádí po aktualizaci opravdu podivné věci, je možné, že jste před kompilací nového jádra zapomněli na `make clean`. Příznaky mohou být různé – od pádu celého systému přes problémy se vstupními a výstupními zařízeními až po nevyrovnané výkony. Nezapomeňte také na `make dep`.

7.2 Velká nebo pomalá jádra

Jestliže vaše jádro spotřebovává příliš mnoho paměti, je velké nebo se kompiluje velmi dlouho i na vašem novém 786DX6/440, máte asi nakonfigurováno příliš mnoho nepotřebných věcí (ovladačů zařízení, systémů souborů atd.).

Kolik paměti jádro využívá zjistíte, když vezmete celkovou velikost paměti vašeho počítače a odečtete od ní množství „celkové paměti“ z `/proc/meminfo` nebo z výstupu příkazu `„free“`. K výsledku se dopravujete i pomocí `„dmesg“` (nebo nahlédnutím do log-souboru jádra, ať už je na systému kdekoliv). Bude zde řádek, který bude vypadat asi takto:

```
Memory: 15124k/16384k available (552k kernel code,
      384k reserved, 324k data)
```

Moje 386-ka (která má nakonfigurováno trochu méně zbytečností) říká:

```
Memory: 7000k/8192k available (496k kernel code,
      384k reserved, 312k data)
```

Když „prostě musíte“ mít velké jádro, ale systém vám to neumožní, můžete zkusit „make bzImage“. V takovém případě možná budete muset nainstalovat novou verzi LILO.

7.3 Jádro se nezkompiluje

Jestliže se nezkompiluje, bude to proto, že záplata se nepovedla nebo je jádro poškozené. Můžete mít také nesprávnou nebo poškozenou (například s chybou v souborech include) verzi gcc. Zkontrolujte, jestli jsou správně nastaveny symbolické odkazy, které Linus popisuje v README. Jestliže se standardní jádro nekompiluje, obecně je to proto, že je něco v systému opravdu špatně. Nutná bude nová instalace určitých nástrojů.

Nebo také kompilujete jádro 1.2.x kompilátorem ELF (gcc 2.6.3 a vyšší). Jestliže se vám během kompilace objeví spousta hlášení typu `to-a-to undefined` (není nadefinováno), je problém možná na vaší straně. Náprava je ve většině případů velice jednoduchá. Tyto řádky přidejte na začátek `arch/i386/Makefile`:

```
AS=/usr/i486-linuxaout/bin/as
LD=/usr/i486-linuxaout/bin/ld -m i386linux
CC=gcc -b i486-linuxaout -D__KERNEL__ -I$(TOPDIR)/include
```

Potom opět proveďte `make dep` a `zImage`.

Ve vzácných případech se může gcc zhroutit díky problémům hardwaru. Chybové hlášení bude pak vypadat asi takto: „xxx exited with signal 15“ (xxx ukončeno signálem 15). To většinou vypadá dost záhadně. Ani bych se o tom nezmínil, ale jednou se mi to stalo – měl jsem špatnou cache na základní desce a kompilátor by občas sahal do prázdna. Jestliže se objeví problémy, pokuste se nejprve o reinstalaci gcc. Obávat se můžete začít až když se vám jádro bude dobře kompilovat s vypnutou externí cache, se zmenšenou RAM atd.

Lidé se většinou při hlášení problémů hardwaru vylekají. Tyto problémy zde řešit nebudu, k tomu je zde FAQ – na adrese <http://www.bitwizard.nl/sig11/>.

7.4 Nová verze jádra se nespustí

Nespustili jste LILO nebo není správně nakonfigurováno. Jednou mě „dostala“ jedna věc – problém v konfiguračním souboru; bylo tam `„boot = /dev/hda1“` místo `„boot = /dev/hda“` (tohle může být zpočátku velmi mrzuté, ale jakmile vám konfigurační soubor funguje, nemusíte to měnit).

7.5 Zapomněli jste spustit LILO nebo se systém vůbec nezavede

Jéje! Tady je nejlepší zavést systém z diskety a připravit další zaváděcí disketu (například „make zdisk“). Musíte vědět, kde je váš kořenový (/) systém souborů a jakého je typu (second extended, minix atd.). V následujícím příkladu musíte také vědět, na jakém systému souborů je váš zdrojový strom `/usr/src/linux`, jakého je typu a kam se normálně připojuje (mount).

V našem příkladu je v oblasti `/dev/hda1` a systém souborů, na kterém je `/usr/src/linux`, je `/dev/hda3`, normálně připojovaný jako `/usr`. Oba dva systémy souborů jsou second extended. Funkční obraz jádra v `/usr/src/linux/arch/i386/boot` se nazývá `zImage`.

Vycházíme zde z toho, že když je `zImage` funkční, je možné jej použít pro další disketu. Další možnost, která může a nemusí být lepší (záleží na problému vašeho systému), bude následovat po příkladu.

Nejprve zavedte systém ze zaváděcí nebo záchranné diskety a připojte systém souborů, který obsahuje obraz funkčního jádra:

```
mkdir /mnt
mount -t ext2 /dev/hda3 /mnt
```

Jestliže vám `mkdir` sdělí, že adresář již existuje, ignorujte jej. Nyní proveďte `cd` na místo, kde byl obraz funkčního jádra. Povšimněte si, že

```
/mnt + /usr/src/linux/arch/i386/boot - /usr =
/mnt/src/linux/arch/i386/boot
```

Do disketové mechaniky (ne do zaváděcího nebo kořenového disku!) vložte naformátovanou disketu, okopírujte na ni obraz a nakonfigurujte jej pro váš kořenový systém souborů:

```
cd /mnt/src/linux/arch/i386/boot
dd if=zImage of=/dev/fd0
rdev /dev/fd0 /dev/hda1
```

Proveďte `cd /` a odpojte normální systém souborů `/usr`:

```
cd /
umount /mnt
```

Nyní byste měli být schopni z této diskety normálně zavést systém. Nezapomeňte po zavedení systému spustit `lilo` (nebo to, díky čemu jste se dostali do problémů)!

Jak už jsme naznačili, je zde ještě jedna alternativa. Jestliže máte funkční obraz jádra v / (například `/vmlinuz`), můžete jej využít pro zaváděcí disk. Když si vezmeme předchozí příklad a to, že můj obraz jádra je `/vmlinuz`, proveďte oproti příkladu tyto změny: změňte `/dev/hda3` na `/dev/hda1` (systém souborů `/`), `/mnt/src/linux` na `/mnt` a `if=zImage` na `if=vmlinuz`.

Poznámka: Vysvětlení jak odvodit `/mnt/src/linux` může být ignorováno.

Použití LILO na velkých mechanikách (discích s více než 1 024 cylindry) může způsobovat problémy. V takovém případě doporučujeme pročíst dokumentaci LILO.

7.6 Objeví se 'warning: bdflush not running'

Toto může být velký problém. Počínaje verzí jádra od 1.0 (kolem 20. dubna 1994) byl aktualizován/nahrazen program „update“, který pravidelně vyprazdňuje buffery systému souborů. Sežeňte si zdrojové texty k „bdflush“ (měli byste je nalézt tam, kde jste si opatřili zdrojový text jádra) a instalujte jej (přitom asi budete chtít systém spouštět pod starým jádrem). Sám se nainstaluje jako „update“ a po novém zavedení systému by si už nové jádro nemělo stěžovat.

7.7 Objeví se něco o nenadefinovaných symbolech a kompilace neproběhne

Pravděpodobně máte kompilátor ELF (`gcc` 2.6.3 a vyšší) a zdrojový text jádra 1.2.x (nebo starší). V takové situaci se většinou postupuje přidáním těchto tří řádků na začátek `arch/i386/Makefile`:

```
AS=/usr/i486-linuxaout/bin/as
LD=/usr/i486-linuxaout/bin/ld -m i386linux
CC=gcc -b i486-linuxaout -D__KERNEL__ -I$(TOPDIR)/include
```

Tak se bude jádro verze 1.2.x kompilovat s knihovnami `a.out`.

7.8 Nemohu zprovoznit svoji mechaniku IDE/ATAPI CD-ROM

Je zvláštní, že mnoha lidem nefungují jejich mechaniky ATAPI, což je pravděpodobně tím, že je zde množství faktorů, které to mohou zapříčinit.

Jestliže máte mechaniku CD-ROM jako jediné zařízení na některém rozhraní IDE, musíte ji nastavit jako „master“ nebo „single“. To bývá nejčastější chyba.

Firma Creative Labs někdy na své zvukové karty umísťuje rozhraní IDE. Tak vzniká zajímavý problém – někteří lidé mají jen jedno rozhraní, ale někteří mají na základní desce zabudována dvě rozhraní IDE (obvykle na IRQ 15), takže je nutné na zvukové kartě nastavit třetí port IDE (IRQ 11, jak mi bylo řečeno).

Tím vzniká problém u verze Linuxu 1.2.x, která nepodporuje třetí rozhraní IDE (podpora začíná až někdy od 1.3.x, ale takový už je vývoj, s tím se musíte smířit). Při řešení máte několik možností.

Jestliže máte druhý port IDE, je možné, že jej ještě nevyužíváte nebo na něm ještě nemáte dvě zařízení. Odpojte mechaniku ATAPI ze zvukové karty a dejte ji na druhé rozhraní. Pak můžete odstavit rozhraní ze zvukové karty, čímž mimochodem ušetříte jedno IRQ.

Jestliže nemáte druhé rozhraní, nastavte rozhraní zvukové karty (samozřejmě, že ne to zvukové) jako IRQ 15 – tedy druhé rozhraní. To by mělo fungovat.

Jestliže je připojení mechaniky na „třetím“ rozhraní prostě nutné nebo máte ještě jiné problémy, obstarejte si jádro novější a pročtěte si `drivers/block/README/ide`. Je zde mnohem více informací.

7.9 Objevují se podivná hlášení o zastaralých směrovacích požadavcích

Obstarejte si novou verzi programu `route` a dalších programů, které se používají při směrování. Změnil se `/usr/include/linux/route.h` (což je ve skutečnosti soubor v `/usr/src/linux`).

7.10 Ve verzi 1.2.0 nefunguje firewall

Přejděte na verzi alespoň 1.2.1.

7.11 „Not a compressed kernel image file“

Nepoužívejte jako obraz zaváděcího disku soubor `vmlinuz`, vytvořený v `/usr/src/linux`; tím správným je `[..]/arch/i386/boot/zImage`.

7.12 Problémy s konzolovým terminálem po přechodu na 1.3.x

V `/etc/termcap` v záznamu `console` změňte slovo `dumb` na `linux`. Můžete vytvořit i nový záznam v `terminfo`.

7.13 Po aktualizaci jádra se nic nekompiluje

Zdrojový text jádra linuxu obsahuje množství souborů `include` (končí na `.h`), které jsou standardně požadovány v `/usr/include`. Většinou se odkazují takto (`xyzyzy.h` je něco z `/usr/include/linux`):

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

Normálně je v `/usr/include` odkaz do adresáře `include/linux` vašeho zdrojového textu jádra (většinou `/usr/src/linux/include/linux`), nazývaný `linux`. Jestliže tento odkaz chybí nebo ukazuje na špatné místo, většina věcí se nebude kompilovat. Jestliže jste si řekli, že vám zdrojový text jádra zabíral na disku příliš mnoho místa a smazali jste jej, je to asi tím. Dalším problémem mohou být přístupová práva k souborům; jestliže má váš **root** `umask`, neumožňující ostatním uživatelům implicitně vidět jeho soubory a vy jste si vzali zdrojový text jádra bez volby **p** (zachovat mód souboru), tak ostatní uživatelé také nebudou schopni využívat C-kompilátor. I když je možné k nápravě použít příkaz `chmod`, jednodušší asi bude znovu získat soubory `include`. To můžete provést stejným způsobem, jako u celého zdrojového textu na začátku, pouze s přidaným argumentem:

```
blah# tar xzvpf linux.x.y.z.tar.gz linux/include
```

Poznámka: Jestliže zde není, „`make config`“ znovu odkaz `/usr/include/linux` vytvoří.

7.14 Zvýšení limitů

Následujících pár *vzorových* příkazů může pomoci těm, kteří chtějí zvýšit některé dolní limity, uvalené jádrem:

```
echo 4096 > /proc/sys/kernel/file-max
echo 12288 > /proc/sys/kernel/inode-max
echo 300 400 500 > /proc/sys/vm/freepages
```


8 Poznámka pro přechod k verzi 2.0.x

Jádro verze 2.0.x představilo pár změn ve své instalaci. Soubor `Documentation/Changes` ve zdrojovém stromu 2.0.x obsahuje informace, které byste měli znát při přechodu na verzi 2.0.x. Bude nutné aktualizovat několik základních balíčků, jako jsou `gcc`, `libc` a `SysVInit`, a snad také upravit některé systémové soubory, takže s tím počítejte a nezapomínejte.

9 Moduly

Moduly jádra, určené k nahrání, mohou ušetřit paměť a usnadnit konfiguraci. Moduly zahrnují již i systémy souborů, ovladače ethernetových karet, páskové ovladače, ovladače tiskáren a další.*

9.1 Instalace modulových utilit

Modulární utility jsou k dispozici ze stejného místa, jako jádro, a sice pod názvem `modules-x.y.z.tar.gz`; vyberte si nejvyšší číslo `x.y.z`, které je menší nebo rovno verzi vašeho aktuálního jádra. Rozbalte jej pomocí `„tar zxvf modules-x.y.z.tar.gz“`, proveďte `cd` do adresáře, který vytvoří (`modules-x.y.z`), nahlédněte do `README` a proveďte instalační instrukce zde popsané (samé jednoduché, jako je `make install`). Nyní byste měli mít v `/sbin` programy `insmod`, `rmmod`, `ksyms`, `lsmod`, `genksyms`, `modprobe` a `depmod`. Jestli chcete, otestujte utility pomocí příkladu ovladače „hw“ v `insmod`; podrobnosti ve stejném adresáři v souboru `INSTALL`.

Moduly vkládá do běžícího jádra `insmod`. Moduly mají obvykle koncovku `.o`; příklad ovladače (viz výše) se nazývá `drv_hello.o`, takže aby se použil, musí se napsat `„insmod drv_hello.o“`. Výpis modulů, právě používaných jádrem, získáte pomocí `lsmod`. Výstup vypadá takto:

```
blah# lsmod
Module: #pages: Used by:
drv_hello 1
```

„`drv_hello`“ je název modulu, využívá jednu stránku (4KB) paměti a v této chvíli na něm nezávisí žádné další moduly. Tento modul odstraní pomocí `„rmmod drv_hello“`. Po-

* Poznámka korektora: Nyní se tyto pomocné programy nazývají `modutils`.

všimněte si, že `rmmod` požaduje *module name* (název modulu), ne název souboru; získáte jej ze seznamu **lsmod**. Účely dalších modulových utilit jsou dokumentovány na jejich manuálových stránkách.

9.2 Moduly, dodávané s jádrem

Od verze 2.0.30 je skoro všechno k dispozici ve formě modulů. Při využívání modulů si nejprve uvědomte, jestli jste je nenakonfigurovali přímo do jádra; to znamená neříkat během „`make config`“ na patřičných místech „`y`“. Zkompilujte nové jádro a zaveďte s ním systém. Proveďte `cd /usr/src/linux a „make modules“`. Tím se kompilují všechny moduly, které jste neurčili v konfiguraci jádra, a v `/usr/src/linux/modules` se k nim přidají odkazy. Můžete je využít přímo v tomto adresáři nebo můžete spustit „`make modules_install`“, čímž se instalují do `/lib/modules/x.y.z`, kde `x.y.z` je verze jádra.

Toto je velice šikovné, obzvláště u systémů souborů. Systémy souborů minix nebo msdos třeba příliš často nepoužíváte. Když mám například dosovou disketu, použiji `insmod /usr/src/linux/modules/msdos.o` a po skončení `rmmod msdos`. Během normálního běhu jádra se tak ušetří kolem 50 KB RAM. Ještě malá poznámka, týkající se systému souborů minix: pro použití na „záchranných“ discích byste jej měli *vždy* konfigurovat přímo do jádra.

10 Další konfigurační volby

Tato část obsahuje popisy vybraných konfiguračních voleb (v `make config`), které nejsou vypsány v části o konfiguraci. Není zde vypsána většina ovladačů zařízení.

10.1 Celkové nastavení

Normal floppy disk support – podpora normálních disket. Možná je dobré si pročíst soubor `drivers/block/README.fd` – zejména pro uživatele IBM Thinkpad.

XT harddisk support – pokud byste chtěli používat ještě 8bitový XT řadič disků, který leží zaprášený někde v koutě.

PCI bios support – pokud máte PCI, můžete to zkusit; ale pozor, některé starší základní desky PCI mohou s touto volbou zlobit. Více informací o sběrnicích PCI pod Linuxem viz PCI-HOWTO.

Kernel support for ELF binaries – ELF je snahou o umožnění použití binárních souborů mezi architekturami a operačními systémy; Linux, jak se zdá, tento trend podporuje, takže volba je pro vás žádoucí.

Set version information on all symbols for modules – dříve byly moduly jádra překompilovány s každým novým jádrem. Jestliže odpovíte „y“, bude možné používat moduly, kompilované pod jinou verzí. Více podrobností viz `README.modules`.

10.2 Volby pro síť

Volby pro síť jsou popsány v příslušném dokumentu (`NET-3-HOWTO`).

11 Tipy a triky

11.1 Přesměrování výstupu příkazů `make` nebo `patch`

Jestliže potřebujete záznam toho, co provádí příkazy „`make`“ a „`patch`“, můžete výstup přesměrovat do souboru. Nejprve zjistěte, který příkazový interpret využíváte: „`grep root /etc/passwd`“ a hledejte něco jako „`/bin/csh`“.

Pokud používáte `sh` nebo `bash`, kopie výstupu „**příkazu**“ se umístí do „**výstupního souboru**“ takto:

```
(příkaz) 2>&1 | tee (výstupní soubor)
```

U `csh` nebo `tcsh` takto:

```
(příkaz) |& tee (výstupní soubor)
```

Pro `rc` (poznámka: `rc` pravděpodobně nepoužíváte) takto:

```
(příkaz) >[2=1] | tee (výstupní soubor)
```

11.2 Podmíněná instalace jádra

Kromě využití disket je zde ještě několik metod testování nového jádra bez ovlivnění starého. Na rozdíl od jiných unixových záležitostí je LILO schopné zavést systém z libovolného místa disku (jestliže máte velký disk – asi 500 MB a více – pročtěte si raději dokumentaci k LILO, protože zde mohou vyvstat problémy). Takže pokud na konec konfiguračního souboru LILO přidáte něco takového:

```
image = /usr/src/linux/arch/i386/boot/zImage
label = new_kernel
```

můžete si (samozřejmě po spuštění `lilo`) vybrat spuštění nově kompilovaného jádra bez narušení starého `/vmlinuz`. V **LIL**O je nejsnazší způsob zavedení nového jádra stisk klávesy **[Shift]** při zavádění systému (když je na obrazovce **LIL**O a nic jiného), získáte tak prompt a zadáte „`new_kernel`“.

Jestliže si chcete v systému ponechat několik různých zdrojových stromů jader (takhle obsadíte *hodně* místa na disku, pozor), obvykle se pojmenovávají `/usr/src/linux-x.y.z`, kde `x.y.z` je verze jádra. Zdrojový strom si pak můžete „vybrat“ pomocí symbolického odkazu; například „`ln -sf linux-1.2.1 /usr/src/linux`“ nastaví jako aktuální strom 1.2.2. Před vytvořením takového symbolického odkazu se ujistěte, že poslední argument u `ln` není skutečný adresář (staré symbolické odkazy jsou dobré); výsledek by nebyl takový, jaký byste očekávali.

11.3 Aktualizace jádra

Russel Nelson (nelson@crynwr.com) shrnul změny v nových verzích jádra. Jsou popsány stručně a před aktualizací do nich můžete nahlédnout. K dispozici jsou na anonymním FTP serveru na ftp.emlist.com v adresáři `pub/kchanges` nebo na URL

<http://www.crynwr.com/kchanges>*

12 Další dokumenty této série, které se mohou hodit

- Sound-HOWTO: zvukové karty a utility
- SCSI-HOWTO: vše o řadičích a ovladačích SCSI
- NET-3-HOWTO: síť
- PPP-HOWTO: PPP síť obecně
- PCMCIA-HOWTO: o ovladačích pro váš notebook
- ELF-HOWTO: co je ELF, převody..
- Hardware-HOWTO: přehled podporovaného hardwaru
- Module-HOWTO: něco k modulům jádra
- Kernel-d-HOWTO: o `kernel.d`
- BogoMips-HOWTO: když nevíte o co se jedná

* Poznámka korektora: Seznamy změn v novějších jádrech naleznete na adrese <http://www.linuxhq.com>.

13 Závěr

13.1 Autor

Autorem a správcem tohoto dokumentu je Brian Ward (**bri@blah.math.tu-graz.ac.at**). Posílejte mi prosím jakékoliv komentáře, dodatky a opravy (nejdůležitější jsou pro mě opravy).

Na jednom z těchto URL se můžete podívat na moji „domovskou stránku“:

<http://www.math.psu.edu/ward/>

<http://blah.math.tu-graz.ac.at/~bri/>

I když se snažím pozorně pročítat veškerou poštu, uvědomte si prosím, že jí každý den dostávám *opravdu* hodně, takže na vás může přijít řada až za dlouho. Tohle platí zejména v případě, kdy se na mne s něčím obracíte. Snažte se proto být struční a srozumitelní. Jestliže píšete o nefunkčním hardwaru (nebo něčem podobném), musím vědět, jaká je vaše hardwarová konfigurace. Když píšete o chybě, nepište jen: „Zkusil jsem to, ale objevila se chyba.“ Musím vědět, o jakou chybu se jedná. Měl bych také vědět, jakou verzi jádra, `gcc` a `libc` používáte. Když mi napíšete jen, že používáte tu a tu distribuci, moc mi to neřekne. Nevadí mi, že mi pokládáte jednoduché dotazy; uvědomte si, že když se nikdy nezeptáte, nikdo vám také neodpoví! Rád bych poděkoval všem za jejich ohlasy.

Jestliže jste mi napsali a odpověď v rozumné době (tři týdny a více) nepřišla, možná jsem omylem vaši zprávu smazal nebo něco podobného (znáte to). Pak zkuste dotaz poslat ještě jednou.

Dostávám spoustu dopisů, které se týkají problémů s hardwarem. V pořádku, ale uvědomte si, že neznám všechno hardware světa a nevím, jak moc vám mohu být nápomocen; já osobně používám stroje s disky IDE a SCSI, CD-ROM mechaniky SCSI, ethernetové karty 3Com a WD, sériové myši, základní desky PCI, řadiče NCR 810 SCSI, procesory AMD 386DX40 w/Cyrix, AMD 5x86, AMD 486DX4 a Intel 486DX4 (takže tohle všechno používám a znám já, ale neberte to jako doporučení pro vás). Pokud chcete, klidně se ptejte.)

Verze 0.1 byla napsána 3. října 1994. Tento dokument je k dispozici ve formátech SGML, PostScript, TeX, roff a v čistém textu.

13.2 Co ještě chybí

Část „Tipy a triky“ je příliš krátká. Chtěl bych ji rozšířit o další nápady dalších lidí.

To stejné platí o „dalších balících“.

Potřebuji také více informací o odladování a odstraňování havárií.

13.3 Příspěvky

Použil jsem malou část Linusova README (volby pro úpravy jádra). Díky!

uc@brian.lunetix.de (Ulrich Callmeier): `patch -s a xargs`.

quinlan@yggdrasil.com (Daniel Quinlan): úpravy a doplnění v mnoha částech.

nat@nataa.fr.eu.org (Nat Makarevitch): `mrproper`, `tar -p`, mnoho dalších věcí.

boldt@math.ucsb.edu (Axel Boldt): sestavil na síti popisy voleb konfigurace jádra a tento seznam mi poskytl.

lembark@wrkhors.psyber.com (Steve Lembark): návrh na využití více možností zavádění systému.

kbriggs@earwax.pd.uwa.edu.au (Keith Briggs): pár oprav a návrhů.

rmcguire@freenet.columbus.oh.us (Ryan McGuire): dodatky možností u `make`.

dumas@excalibur.ibp.fr (Eric Dumas): překlad do francouzštiny.

simazaki@ab11.yamanashi.ac.jp (Yasutada Shimazaki): překlad do japonštiny.

jjamor@lml.ls.fi.upm.es (Juan Jose Amor Iglesias): překlad do španělštiny.

mva@sbbs.se (Martin Wahlen): překlad do švédštiny.

jzp1218@stud.u-szeged.hu (Zoltan Vamosi): překlad do maďarštiny.

bart@mat.uni.torun.pl (Bartosz Maruszewski): překlad do polštiny.

donahue@tiber.nist.gov (Michael J Donahue): překlady.

rms@gnu.ai.mit.edu (Richard Stallman): poznámka k „volné“ šířitelnosti dokumentu.

dak@Pool.Informatik.RWTH-Aachen.DE (David Kastrup): NFS-záležitosti.

esr@snark.thyrsus.com (Eric Raymond): různé úpravy.

13.4 Autorská práva, licence a všechny tyhle záležitosti

Copyright (c) Brian Ward, 1994-1997.

Kopírování a rozšiřování tohoto manuálu je povoleno, přičemž však musí být zachována tato poznámka a uvedení autorských práv.

Kopírování a rozšiřování tohoto manuálu v upravené podobě je povoleno podle podmínek pro doslovné kopírování, přičemž odvozená práce musí být šířena se stejnou poznámkou, jako je tato. Překlady spadají do kategorie „upravených verzí“.

Záruky: žádné.

Doporučení: Komerční distribuce je povolena a vítána; doporučuji ale, aby distributor předtím kontaktoval autora a získal aktualizovanou verzi (můžete mi pak poslat kopii toho, co jste vydali, když už jsme u toho). Překladačům také doporučuji před započítím prací kontaktovat autora. Tištěná verze vypadá lépe. A podporujte recyklaci.

Linux IPX

Terry Dawson, terry@perf.no.itg.telstra.com.au v2.2, 29. března 1997

Cílem tohoto dokumentu je popsat způsob získání, instalace a konfigurace různých nástrojů, které jsou dostupné pro operační systém Linux a využívají podporu jádra Linuxu protokolu IPX.

1 Úvod

Toto je dokument Linux IPX-HOWTO. V kombinaci s tímto dokumentem byste si měli přečíst i dokument NET-3-HOWTO.

1.1 Změny oproti předchozím verzím

Dodatky:

Doplněny některé informace o typu rámce

- Opravy/aktualizace:
- Pro síťové IPX-adresy v souboru `/etc/ppp/options` je vyžadováno `0x`.
- Aktualizace verzí a umístění.
- Některé změny v abstraktním tisku a administrativních nástrojích.

1.2 Úvod

Jádro Linuxu má ve srovnání s jinými operačními systémy na bázi Unixu zcela novou implementaci síťové podpory. Schopnost nového přístupu k vývoji síťového softwaru jádra přispěla k tomu, že jádro Linuxu nabízí zabudovanou podporu i pro řadu jiných protokolů než TCP/IP. Protokol IPX je jedním z nich.

Jádro Linuxu podporuje pouze protokol IPX. Zatím neobsahuje podporu protokolů typu IPX/RIP, SAP nebo NCP. Tyto protokoly podporuje jiný software, o němž bude zmínka dále.

Podpora protokolu IPX je dílem Alana Coxe <alan@lxorguk.ukuu.org.uk> a značně ji vylepšil Greg Page <greg@caldera.com>.

2 Záruka

Nevím a ani nemohu vědět vše o síťovém softwaru operačního systému Linux. Proto vás předem upozorňuji, že tento dokument může obsahovat chyby. Přečtete si prosím soubory `README`, které jsou dodávány společně s programy popisovanými v tomto dokumentu, kde najdete podrobnější a přesnější informace. Budu se snažit, aby tento dokument obsahoval co nejméně chyb a byl co nejaktuálnější. Verze programů, o kterých bude řeč, byly aktuální v době psaní tohoto dokumentu.

Já ani autoři softwaru zmiňovaného v tomto dokumentu neposkytujeme žádnou ochranu proti vašim akcím. I v případě, že nakonfigurujete tento software podle popisu v tomto dokumentu a objeví se nějaké síťové problémy, nesete odpovědnost jenom vy. Toto varování zde uvádím proto, že návrh a konfigurace sítě IPX není vždy jednoduchá záležitost a špatný návrh nebo konfigurace vaší sítě mohou někdy vést k vzájemné interakci s ostatními routery a souborovými servery. Toto varování zde uvádím také proto, že mě jeden nešťastný člověk požádal, abych toto čtení uvedl trochu drsnějším způsobem.

3 Příbuzná dokumentace

V tomto dokumentu se předpokládá, že umíte sestavit jádro Linuxu s příslušně zvolenými síťovými volbami a že umíte používat základní síťové nástroje, jako jsou `ifconfig` a `route`. Pokud tomu tak není, měli byste si společně s tímto dokumentem přečíst i dokument `NET-3-HOWTO`, který se touto problematikou zabývá.

K dalším dokumentům `HOWTO`, které by pro vás mohly být užitečné, patří:

- Dokument *Ethernet-HOWTO*, který popisuje podrobnosti konfigurace ethernetového zařízení pro systém Linux.
- Dokument *PPP-HOWTO*, protože od verze 2.2.0d existuje pro implementaci protokolu PPP podpora IPX.

3.1 Nové verze tohoto dokumentu

Je-li vaše kopie tohoto dokumentu starší než dva měsíce, pak vám doporučuji stáhnout si novější verzi. Síťová podpora operačního systému Linux se velmi rychle mění, jsou přidávány nová vylepšení a funkce, takže se poměrně často mění i tento dokument. Poslední verzi tohoto dokumentu vždy získáte prostřednictvím anonymní služby FTP na adrese:

sunsite.unc.edu

v adresáři `/pub/Linux/docs/HOWTO/IPX-HOWTO`

nebo

`/pub/Linux/docs/HOWTO/other-formats/IPX-HOWTO{-html.tar,ps,dvi}.gz`

Prostřednictvím webu je tento dokument k dispozici na serveru `http://sunsite.unc.edu/LDP/linux.html`, na internetové stránce **IPX-HOWTO** `http://sunsite.unc.edu/LDP/HOWTO/IPX-HOWTO.html` nebo na mé adrese **<terry@perf.no.itg.telstra.com.au>**. Občas může být také zaslán do diskusních skupin *comp.os.linux.networking*, *comp.os.linux.answers* a *news.answers*.

3.2 Zpětná vazba

Jakékoliv komentáře, aktualizace nebo návrhy mi prosím pošlete na adresu **<terry@perf.no.itg.telstra.com.au>**. Čím dříve dostanu vaše připomínky, tím dříve budu moci aktualizovat a opravit tento dokument. Pokud v něm najdete nějaké chyby, pak mi prosím neprodleně napište, protože v současné době čtu diskusní skupiny jen zřídka.

3.3 Podpora konferencí

Diskusi o různých softwarových IPX-balících popisovaných v tomto dokumentu je také věnována jedna konference. Přihlásit se do ní můžete tak, že pošlete zprávu na adresu **listserv@sh.cvut.cz** a do těla zprávy vložíte řetězec „add linware“. Příspěvky do konference pak zasílejte na adresu **linware@sh.cvut.cz**.

Tato konference je archivována na WWW-stránce `http://www.kin.vslib.cz/hypermail/linware/`.

4 Některé termíny používané v tomto dokumentu

V tomto dokumentu se často setkáte s termíny **klient** a **server**. Normálně se jedná o poměrně specifické termíny, ale v tomto dokumentu jsem jejich definice trochu zevšeobecnil, takže zde mají následující význam:

- klient** Počítač nebo program, který zahajuje nějakou akci nebo spojení za účelem získání nějaké služby nebo dat.
- server** Počítač nebo program, který přijímá příchozí spojení od více počítačů a poskytuje jim služby nebo data.

Obě tyto definice sice nejsou příliš spolehlivé, ale jsou prostředkem, jak rozlišit dva konce systémů peer-to-peer, jako je *SLIP* nebo *PPP*, které ve skutečnosti nemají klienty ani servery.

Mezi další termíny, s nimiž se zde setkáte, patří:

- Bindery** *Bindery* je specializovaná databáze uchovávající síťové konfigurační informace na novellovském souborovém serveru. Klienti NetWare mohou této databázi posílat dotazy a získávat informace o dostupných službách, směrování a informace o uživateli.
- Typ rámce** (Frame Type) Je termín používaný k popisu skutečného protokolu použitého pro přenos datagramů IPX (a IP) přes ethernetové síťové segmenty. Existují čtyři běžné typy. Jsou to:

Ethernet_II Jedná se o vylepšenou verzi původního ethernetového standardu DIX. Novellu bylo přiděleno id původního protokolu a to znamená, že IPX a IP mohou v prostředí Ethernet_II poměrně dobře koexistovat. Tento typ rámce se v novellovských prostředích běžně používá a je dobrou volbou.

802.3 Jedná se o protokol podle standardu I.E.E.E, který definuje mechanismus CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Je založen na původním standardu DIX Ethernet, ale obsahuje důležité úpravy – typové pole (id protokolu) bylo převedeno na délkové pole. To proto, že IPX by zde neměl být použit. Standard IEEE 802.3 byl navržen **pouze** pro přenos rámců 802.3, existují však implementace, které ho využívají k přímému přenosu IPX-rámců. Pokud budete spolupracovat se sítí, která ho již používá, pak se tomuto standardu vyhněte.

802.2 Tento I.E.E.E protokol definuje sadu Logical Link Control procedur. Poskytuje zjednodušený způsob povolování soužití různých protokolů, ale v tomto ohledu je poměrně omezený. Firma Novell používá neoficiální Service Address Point (jako id protokolu), ale protože je používá skoro každý, nebyl to zatím velký problém.

SNAP SNAP znamená Sub Network Access Protocol. Tento protokol je navržen pro funkci nad protokoly 802.3 a 802.2. Rozšiřuje mezi-protokolovou kompatibilitu 802.2 a poskytuje určité měřítka kompatibility s existujícími typy rámců Ethernet a Ethernet_II.

IPX Internet Packet eXchange je protokol, s jehož pomocí poskytuje společnost Novell mezikomunikací podporu svého produktu NetWare(tm). Protokol IPX je funkcí podobný protokolu IP, který používají sítě TCP/IP.

Síťová IPX adresa

Toto číslo slouží k jedinečné identifikaci konkrétní IPX-sítě. Tato adresa je obvykle zapisována hexadecimálně. Příklad by vypadal asi takto: 0x23a91002.

Interní IPX síť

Jedná se o virtuální IPX-sít. Virtuální je proto, že nekoresponduje s fyzickou sítí. Slouží k jedinečné identifikaci a adresaci konkrétního IPX-hostitele. Tento způsob je obecně použitelný pouze pro IPX-hostitele, kteří figurují ve více fyzických IPX-sítích, kam patří například souborové servery. Adresa je zakódována stejným způsobem, jako pro fyzickou IPX-sít.

RIP Routing Information Protocol je protokol používaný k automatickému šíření síťových tras (route) v síti IPX. Jeho funkce je podobná protokolu RIP používanému v sítích TCP/IP.

NCP NetWare Core Protocol je protokol síťového souborového systému, který vyvinula společnost Novell Corporation pro svůj produkt NetWare (tm). Funkce protokolu NCP je podobná systému NFS používanému v sítích TCP/IP.

SAP Service Advertisement Protocol je protokol, který navrhla společnost Novell Corporation. Slouží k propagování síťových služeb v prostředí NetWare(tm).

Hardwarová adresa

Je číslo, které slouží k jedinečné identifikaci hostitele ve fyzické síti na úrovni přístupu k médium. Příkladem hardwarové adresy je *ethernetová adresa*. Ethernetová adresa je obecně tvořena šesti hexadecimálními hodnotami oddělenými znaky dvojtečka, tj: 00:60:8C:C3:3C:0F.

trasa (route) Trasa je cesta, po které vaše pakety putují sítí, než dorazí ke svému cíli.

5 Soubory IPX v souborovém systému /proc

Protokol IPX má v Linuxu několik podpůrných souborů, které jsou umístěny v adresáři /proc. Jsou to:

/proc/net/ipx_interface

Tento soubor obsahuje informace o rozhraních IPX konfigurovaných ve vašem počítači. Lze je konfigurovat buď manuálně pomocí nějakého příkazu, nebo je může systém detekovat a nakonfigurovat automaticky.

/proc/net/ipx_route

Tento soubor obsahuje seznam tras existujících ve směrovací tabulce protokolu IPX. Tyto trasy lze přidávat manuálně pomocí příkazu nebo automaticky pomocí směrovacího IPX démona.

/proc/net/ipx

Tento soubor obsahuje seznam IPX soketů, které jsou aktuálně otevřeny.

6 Nástroje IPX Grega Page

Greg Page <greg@caldera.com> ze společnosti Caldera Incorporated napsal sadu konfiguračních nástrojů protokolu IPX a zdokonalil podporu jádra Linuxu pro tento protokol.

Díky těmto vylepšením může být Linux nakonfigurován jako plnohodnotný IPX-most nebo router. Zdokonalená podpora protokolu IPX byla zabudována také do jádra většiny hlavních distribucí, takže ji zřejmě již máte.

Nástroje pro konfiguraci sítě umožňují nastavit síťová zařízení tak, aby podporovala protokol IPX a umožňovala konfigurovat IPX-směrování a další prostředky pod systémem Linux. Síťové nástroje podporující protokol IPX jsou dostupné na adrese <ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ipx.tgz>.

6.1 Nástroje IPX podrobněji

`ipx_interface` Tento příkaz slouží k manuálnímu přidávání, rušení a kontrole podpory protokolu IPX u existujících síťových zařízení. Normálně by bylo síťové zařízení ethernetovým zařízením, například `eth0`. Alespoň jedno rozhraní protokolu IPX musí být označeno jako *primární* a o to se postará pří-

volba `-p`. Chcete-li například ethernetovému zařízení `eth0` umožnit podporu IPX-rozhraní pomocí typu rámce IEEE 802.2 a síťové IPX-adresy `0x39ab0222`, pak použijete následující zápis:

```
# ipx_interface add -p eth0 802.2 0x39ab0222
```

Pokud se při spuštění tohoto programu objeví chyba a náhodou ještě nemáte nakonfigurován protokol TCP/IP, budete muset manuálně spustit rozhraní `eth0` pomocí příkazu:

```
# ifconfig eth0 up
```

`ipx_configure` Tento příkaz povoluje nebo zakazuje automatické nastavení konfigurace rozhraní a primárního rozhraní.

```
--auto_interface
```

umožňuje nastavit, zda mají být nová síťová zařízení automaticky nakonfigurována jako IPX-zařízení.

```
--auto_primary
```

umožňuje nastavit, zda má software protokolu IPX automaticky zvolit primární zařízení.

Typickým příkladem je povolení jak automatické konfigurace rozhraní, tak i automatického nastavení primárního rozhraní pomocí následujícího příkazu:

```
# ipx_configure --auto_interface=on --auto_primary=on
```

`ipx_internal_net`

Tento příkaz umožňuje nastavit nebo zrušit interní síťovou adresu. Interní síťová adresa je volitelná, ale je-li nastavena, bude vždy používána jako primární adresa. síťovou adresu IPX `ab000000` na IPX-uzlu 1 nastavíte následujícím způsobem:

```
# ipx_internal_net add 0xab000000 1
```

`ipx_route`

Tento příkaz umožňuje manuální úpravy ve směrovací tabulce protokolu IPX. Chcete-li například přidat trasu do IPX-sítě `0x39ab0222` prostřednictvím routeru s uzlem číslo `0x00608CC33C0F` v IPX-síti `0x39ab0108`, použijte následující příkaz:

```
# ipx_route add 0x39ab0222 0x39ab0108 0x00608CC33C0F
```

7 Konfigurace linuxového počítače jako IPX-routeru

Pokud máte hodně IPX-segmentů, které chcete vzájemně propojit do sítě, potřebujete služby routeru. V prostředí sítě Novell existují dvě informace, které je třeba předat. Jedná se o síťovou směrovací informaci šířenou prostřednictvím protokolu Novell RIP a informaci o oznamování služeb šířenou prostřednictvím protokolu Novell SAP. Každý router, aby byl použitelný ve většině situací, musí podporovat oba tyto protokoly.

Operační systém Linux má zabudovanou podporu pro oba tyto protokoly a poměrně snadno se může fungovat jako plnohodnotný novellovský router.

Podpora jádra Linuxu protokolu IPX ve skutečnosti řídí doručování IPX-paketů po rozhraních, dělá to však podle pravidel zakódovaných do směrovací tabulky protokolu IPX. Linux potřebuje k implementaci protokolů Novell RIP a SAP nějaký program, který zajistí správné vytvoření směrovací tabulky protokolu IPX a její pravidelnou aktualizaci, aby odražela stav sítě.

Volker Lendecke <lendecke@namu01.gwdg.de> vytvořil směrovacího démona, který to udělá za vás.

Najdete ho na adrese:

```
ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ipxripd-0.7.tgz
```

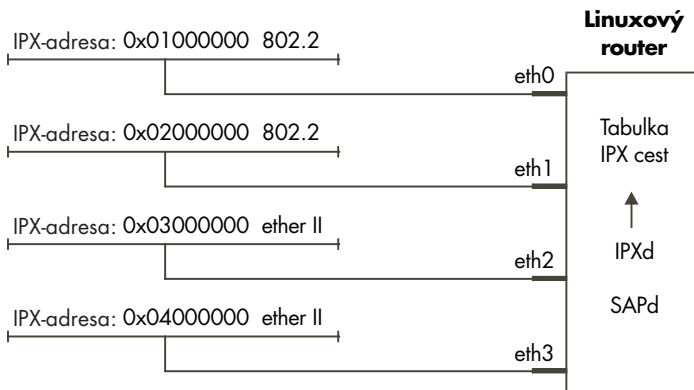
nebo na Volkerově domovském serveru:

```
ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/ipxripd-0.7.tgz
```

Konfigurace linuxového počítače jakožto routeru je poměrně přímočará. Řiďte se následujícími kroky:

1. Sestavte jádro s podporou protokolů IPX, Ethernet a souborového systému `/proc`.
2. Sežeňte si, zkompilujte a nainstalujte démona `ipxd`.
3. Nastartujte nové jádro a ujistěte se, že byly všechny ethernetové karty správně detekovány a nedochází k žádným hardwarovým konfliktům.
4. Na všech rozhraních povolte za pomoci výše popsaného příkazu `ipx_interface` protokol IPX.
5. Spusťte démona `ipxd`.

Uvažujte následující jednoduchou síť:



Konfigurace výše uvedené sítě by vypadala následovně:

```
# ipx_interface add eth0 802.2 0x0100000000
# ipx_interface add eth1 802.2 0x0200000000
# ipx_interface add eth2 etherii 0x0300000000
# ipx_interface add eth3 etherii 0x0400000000
# ipxd
```

Potom byste měli chvíli počkat a podívat se, zda byl soubor `/proc/net/ipx_route` rozšířen o IPX-routery důležité pro vaši konfiguraci a další informace získané od jiných routerů v síti.

7.1 Potřebuji konfigurovat interní síť?

V síti Novell se setkáte s pojmem interní síť, která zjednodušuje směrování v situacích, kdy je k hostiteli připojeno více než jedno síťové zařízení. To je užitečné v případě souborového serveru připojeného k více sítím, protože tak stačí oznámit jen jedinou trasu, pomocí které je možné serveru dosáhnout, bez ohledu na síť, z níž se o to pokoušíte.

V případě konfigurace, při které neprovozujete souborový server a váš počítač slouží pouze jako IPX-router, je odpověď na nastolenou otázku trochu složitější. Je známo, že konfigurace protokolu IPX/PPP funguje „lépe“, když současně zkonfigurujete i interní síť.

V každém případě je tato činnost jednoduchá, nicméně vyžaduje nové sestavení jádra. Když budete odpovídat na dotazy příkazu `make config`, musíte na otázku `Full internal IPX network` odpovědět `y`, jak je ukázáno v následujícím výpisu:

```
...  
...  
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] y  
...  
...
```

Ke konfiguraci interního síťového rozhraní použijte příkaz `ipx_internal_net`, který jsme si popsali dříve ve stati IPX-nástroje. Nejdůležitější je zajistit, aby síťová IPX-adresa, kterou mu přidělíte, byla ve vaší síti jedinečná a nepoužíval ji žádný jiný počítač nebo síť.

8 Konfigurace linuxového počítače jako NCP-klienta

Jste-li uživatelem sítě založené na různých technologiích, jež zahrnují protokol IP i IPX, je pravděpodobné, že někdy bude potřeba, aby váš linuxový počítač přistupoval k datům uloženým na novellovském souborovém serveru ve vaší síti. Firma Novell již dlouho nabízí pro své souborové servery balík NFS-serveru, který to umožňuje, ale jste-li jen malý systém nebo jen malý počet vašich uživatelů má o tuto funkci zájem, pak to nevyváží cenu tohoto komerčního balíku.

Volker Lendecke <lendecke@namu01.gwdg.de> napsal modul jádra linuxového souborového systému, který podporuje podmnožinu funkcí protokolu Novell NCP, jež umožňují připojení novellovských svazků k linuxovému souborovému systému bez dodatečných produktů pro souborový server. Volker Lendecke nazval tento balík `ncpfs` a nezbytné informace získal v knize „Netzwerkprogrammierung in C“, jejímiž autory jsou Manfred Hill a Ralf Zessin (další informace o této knize najdete v souboru `README`, který je součástí balíku `ncpfs`).

Tento software způsobí, že Linux emuluje pro souborové služby normální novellovskou pracovní stanicí. Obsahuje také malou tiskovou utilitu, která umožňuje tisknout do novellovské tiskové fronty (je zdokumentována ve stati „Tiskový klient“ dále). Balík `ncpfs` spolupracuje s novellovskými souborovými servery verze 3.x a vyšší. Se serverem Novell 2.x fungovat nebude. Klient `ncpfs` také spolupracuje s produkty, které jsou úzce kompatibilní s produkty firmy Novell, ale některé programy, které o sobě tvrdí, že jsou kompatibilní, nesplňují tuto vlastnost úplně. Aby bylo možné používat balík `ncpfs` se souborovými servery Novell 4.x, musí být souborový server nakonfigurován pro režim `bindery emulation`, protože `ncpfs` zatím nepodporuje NDS.

8.1 Získání balíku ncpfs

Poslední verze balíku ncpfs byla navržena pro verzi jádra 1.2.13 nebo verze vyšší než 1.3.71 (sem patří i verze 2.x.x). Pokud používáte jádro, které nespadá do žádné z těchto dvou kategorií, pak budete muset provést jeho upgrade. Podrobný popis najdete v dokumentu *Kernel-HOWTO*.

Balík ncpfs získáte prostřednictvím anonymní služby FTP z Volkerova domovského systému `ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/` nebo na adrese `ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs`, případně jeho zrcadlech. Aktuální verze měla v době psaní tohoto dokumentu označení `ncpfs-2.0.10.tgz`.

8.2 Sestavení ncpfs pro jádro verze 1.2.13

Sestavení jádra s podporou Ethernetu a protokolu IPX

Nejprve se musíte přesvědčit, že bylo vaše jádro sestaveno se zabudovanou podporou protokolu IPX. Ve verzi 1.2.13 je třeba jen kladně odpovědět na dotaz „The IPX protocol“, jak je ilustrováno v následujícím výpisu:

```
...
...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
...
...
```

Také je třeba zajistit zařazení příslušného ovladače pro vaši ethernetovou kartu. Pokud nevíte, jak na to, pak si přečtete dokument *Ethernet-HOWTO*.

Pak můžete přistoupit k sestavení jádra. Nezapomeňte po skončení spustit příkaz `lilo`, který nainstaluje bootovací manažer.

Rozbalení balíku ncpfs

```
# cd /usr/src
# tar xvfz ncpfs-2.0.10.tgz
# cd ncpfs
```

Kontrola souboru Makefile

Hodláte-li používat program `kernel.d` k automatickému zavedení modulu jádra `ncpfs`, pak musíte zrušit komentář na řádce v souboru `Makefile`, který odkazuje na `KERNELD`. Pokud si nejste jistí, co to znamená, měli byste si přečíst dokument *Kernel-HOWTO*, který vás seznámí s konfigurací modulů jádra.

Kompilace softwaru ncpfs

Celý balík by pak mělo být možné přeložit bez nutnosti další konfigurace:

```
# make
```

Zkopírování IPX-nástrojů na příslušné místo

Po skončení příkazu `make` by měly být všechny potřebné nástroje umístěny v adresáři `ncpfs/bin`. Pomocí příkazu

```
# make install
```

je nainstalujete do adresářů, které zvolil Volker Lendecke. Pokud používáte systém založený na ELF, pak bude nutné znovu spustit příkaz „`ldconfig -v`“, který zajistí dostupnost sdílené knihovny.

Zkopírování modulu ncpfs.o na příslušné místo

Po sestavení jádra verze 1.2.* najdete v adresáři `ncpfs/bin` soubor `ncpfs.o`. To je modul jádra `ncpfs`. Tento soubor byste měli zkopírovat na jiné užitečnější místo. V mé distribuci *Debian* jsem ho zkopíroval do adresáře `/lib/modules/1.2.13/fs` a do souboru `/etc/modules` jsem automaticky přidal záznam `ncpfs`, aby byl tento modul automaticky spouštěn při startu. Používáte-li nějakou jinou distribuci, pak si musíte zjistit, kde uchovává své moduly a modul `ncpfs.o` sem zkopírovat, případně ho zkopírovat do adresáře `/etc`. K manuálnímu natažení modulu použijte příkaz

```
# insmod ncpfs.o
```

8.3 Sestavení ncpfs pro jádra verzí 1.3.71++/2.0

Abyste mohli používat poslední verzi balíku `ncpfs`, musíte mít jádro verze 1.3.71 nebo novější. Sem patří i jádra 2.0.*.

Budete-li používat jádro verze 1.3.71 nebo novější, pak je balík `ncpfs` součástí standardní distribuce jádra. Stačí jen, když odpovíte `y` na následující dotaz:

```

Networking options  --->
    ...
    ...
    <*> The IPX protocol
    ...
Filesystems  --->
    ...
    ...
    <*> NCP filesystem support (to mount NetWare volumes)
    ...

```

Stále se ovšem budete muset řídit instrukcemi pro sestavení verzí jádra 1.2.*, takže budete moci zkompilovat nástroje, ale nevznikne tím žádný modul, který byste mohli nainstalovat.

8.4 Konfigurace a používání ncpfs

Konfigurace síťového IPX-softwaru

Existují dva způsoby konfigurace síťového IPX-softwaru. Můžete buď všechny síťové IPX-informace nastavit ručně, anebo můžete nechat volbu rozumných nastavení na softwaru. Druhého případu dosáhnete pomocí následujícího příkazu:

```
# ipx_configure --auto_interface=on --auto_primary=on
```

Toto nastavení by mělo vyhovovat ve většině případů. Pokud tomu tak není, pak si přečtěte výše uvedenou stať „IPX-nástroje“, kde se dozvíte, jak lze tento software nakonfigurovat ručně.

Otestování konfigurace

Po nakonfigurování IPX-sítě by mělo být možné pomocí příkazu `slist` získat seznam všech novellovských souborových serverů ve vaší síti:

```
# slist
```

Pokud příkaz `slist` vypíše zprávu typu: `ncp_connect: Invalid argument`, pak vaše jádro zřejmě nepodporuje protokol IPX. Zkontrolujte, zda jste skutečně spustili příslušné jádro. Při startu systému by se měly objevit zprávy o protokolu IPX a modulu `nepfs`. Pokud příkaz `slist` nevypíše všechny vaše souborové servery, pak bude zřejmě nutné přistoupit k ruční konfiguraci sítě.

Připojení novellovského svazku

Pracuje-li váš síťový IPX-*software* správně, měl by jít připojit svazek novellovského souborového serveru k vašemu linuxovému souborovému systému. K tomuto účelu slouží příkaz zajišťující odpojení, který vyžaduje zadání minimálně následujících parametrů:

1. Název souborového serveru.
2. Připojovací id souborového serveru. Je-li vyžadováno heslo, budete potřebovat také je.
3. Přihlašovací bod, tj. kde chcete, aby došlo k připojení. Měl by to být existující adresář na vašem počítači.

K odpojení připojeného souborového systému NCP slouží ekvivalentní příkaz `ncpmount`. Pokud normálně ukončíte práci s počítačem, dojde ke správnému odpojení souborových systémů NCP, takže pokud použijete příkazy `halt` nebo `shutdown`, nemusíte se o příkaz `ncpunmount` starat.

Jako příklad si ukážeme připojení souborového serveru `ACCT_FS01` s přihlašovacím id `guest` bez hesla k adresáři `/mnt/Accounts`:

```
# ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n
```

Všimněte si použití volby `-n`, která udává, že k přihlášení není potřeba heslo. Stejně přihlášení s heslem `secret` by vypadalo následovně:

```
# ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -P secret
```

Kontrola připojení

Pokud bylo připojení úspěšné, budou svazky přístupné příslušnému id pod adresářem, který jste uvedli jako bod připojení. Měli byste také být schopni procházet adresářovou strukturu a hledat příslušné soubory. Protože NCP neposkytuje uid ani gid vlastnictví souborů, budou mít všechny soubory nastavena stejná přístupová práva a vlastnictví jako adresář, který je přípojným bodem. Mějte to na paměti při sdílení připojení mezi linuxovými uživateli.

Nastavení automatického připojování

Potřebujete-li mít nějaký svazek NCP trvale připojen, budete muset výše uvedené příkazy zařadit do vašich souborů `rc`, aby byly prováděny automaticky při startu systému. Pokud již vaše distribuce Linuxu nabízí nějaký způsob konfigurace protokolu IPX, jako v případě distribuce Debian, pak vám doporučuji umístit příslušné příkazy do souboru `/etc/rc.local`. Můžete použít například následující syntaxi:

```
#
# Podpora pro souborový systém NCP
/sbin/insmod /lib/modules/1.2.13/fs/ncpfs.o

# konfigurace IPX-sítě
ipx_configure --auto_interface=on --auto_primary=on

# přihlášení na souborový server
ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n

#
```

Existuje ještě jeden způsob konfigurace připojení NCP, který spočívá ve vytvoření souboru `$HOME/.nwclient`. Tento soubor obsahuje informace o dočasných nebo uživatelských připojeních NCP, ke kterým pravidelně dochází. Můžete do něj umístit podrobnosti jednotlivých připojení, takže již příště nemusíte zadávat všechny detaily znovu.

Formát tohoto souboru je poměrně jednoduchý:

```
# První záznam je preferovaný server,
# který je použit když nspecifikujete server.
#
# Uživatel Terry se přihlašuje k serveru DOCS_FS01
# s heslem 'password'
DOCS_FS01/TERRY password
#
# Přihlášení k serveru ACCT_FS01 bez hesla.
ACCT_FS01/GUEST -
```

K aktivaci těchto připojení můžete použít třeba následující příkaz:

```
$ ncpmount /home/terry/docs
```

který připojí server `DOCS_FS`, k němuž se přihlásí jako uživatel `TERRY`, pod adresář `/home/terry/docs`. Tento záznam byl zvolen z toho důvodu, že nebyl zadán žádný souborový server. Pokud bychom použili následující příkaz:

```
$ ncpmount -S ACCT_FS01 /home/terry/docs
```

připojil by systém adresář pro uživatele GUEST na serveru ACCT_FS01.

Poznámka: Aby výše uvedený mechanismus fungoval, musí mít soubor `$HOME/nwclient` nastavena přístupová práva `0600`, čehož dosáhnete pomocí následujícího příkazu:

```
$ chmod 0600 $HOME/.nwclient
```

Chcete-li, aby mohli tento mechanismus využívat i jiní uživatelé než pouze uživatel root, musíte příkaz `ncpmount` pomocí následujícího příkazu nastavit `suid root`.

```
# chmod 4755 ncpmount
```

Vyzkoušení utility `nsend`

Součástí balíku je i utilita pro posílání zpráv novellovským uživatelům. Jmenuje se `nsend` a používá se následujícím způsobem:

```
# nsend rod hello there
```

Tento příkaz by poslal uživateli, který je přihlášen k vašemu preferovanému souborovému serveru (první server, který je uveden v souboru `.nwclient`) jako uživatel „rod“ zprávu „hello there“. Pomocí stejné syntaxe, jakou používá příkaz `ncpmount`, můžete zadat i jiný souborový server.

9 Konfigurace linuxového počítače jako NCP-serveru

K dispozici jsou dva balíky, které umožňují Linuxu simulovat funkce novellovského souborového serveru. Oba dva dovolují sdílet soubory uložené na vašem linuxovém počítači společně s uživateli, kteří používají klientský software Novell NetWare. Uživatelé si mohou připojovat a mapovat souborové systémy, které se jim na jejich počítačích budou jevit jako lokální disky, jako v případě skutečného novellovského souborového serveru. Můžete si vyzkoušet oba dva balíky a vybrat si ten, který bude lépe vyhovovat vašim účelům.

9.1 Balík `mars_nwe`

Martin Stover <mstover@freeway.de> vytvořil balík `mars_nwe`, díky němuž může Linux poskytovat klientům NetWare jak souborové, tak tiskové služby.

V případě, že vás zajímá původ názvu tohoto balíku, tak jde o zkratku z Martin Stovers NetWare Emulator.

9.1.1 Schopnosti balíku *mars_nwe*

Balík *mars_nwe* implementuje podmnožinu souborových služeb, disk based bindery a tiskových služeb Novell NCP. Pravděpodobně bude obsahovat chyby, ale v současné době ho používá velké množství lidí a počet chyb se dalšími verzemi neustále snižuje.

9.1.2 Získání balíku *mars_nwe*

Program *mars_nwe* získáte na adrese `ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/` nebo `ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/`.

Nejaktuálnější verze v době psaní této knihy byla `mars_nwe-0.98.pl8.tgz`.

9.1.3 Sestavení balíku *mars_nwe*

Sestavení jádra s podporou Ethernetu a protokolu IPX

Jste-li vlastníkem jádra verze 1.2.13, pak stačí, když při jeho kompilaci odpovíte kladně na otázku „The IPX protocol“ a záporně na otázku „Full internal IPX network’s“, jako v následujícím výpisu:

```
...
...
The IPX protocol (CONFIG_IPX) [n] y
...
...
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] n
...
...
```

Do novějších verzí jádra byl převzat podobný proces, i když skutečný text výzev se mohl mírně změnit.

Také je třeba zahrnout příslušný ovladač pro vaši ethernetovou kartu. Pokud nevíte, jak na to, pak byste si měli přečíst dokument *Ethernet-HOWTO*.

Pak můžete přistoupit ke kompilaci jádra. Nezapomeňte po skončení spustit příkaz `lilo`, který nainstaluje zaváděcí manažer.

Rozbalení balíku *mars_nwe*

```
# cd /usr/src
# tar xvfz mars_nwe-0.98.pl3.tgz
```

Kompilace balíku mars_nwe

Kompilace tohoto balíku je velice jednoduchá. Prvním krokem je spuštění programu `make`, čehož výsledkem bude soubor `config.h`. Potom byste měli tento soubor vyhledat a případně upravit. Můžete zde nastavit položky jako instalační adresáře, které se mají použít, a maximální počet relací a svazků, které bude příslušný server podporovat. Skutečně zajímavé položky, na které byste se měli zaměřit, jsou tyto:

<code>FILENAME_NW_INI</code>	umístění inicializačního souboru
<code>PATHNAME_PROGS</code>	kde se nacházejí podpůrné programy
<code>PATHNAME_BINDERY</code>	kde budou soubory bindery
<code>PATHNAME_PIDFILES</code>	adresář pro soubory s PID procesů
<code>MAX_CONNECTIONS</code>	maximální počet souběžných spojení
<code>MAX_NW_VOLS</code>	maximální počet svazků, které bude mars-nwe podporovat
<code>MAX_FILE_HANDLES_CONN</code>	maximální počet otevřených souborů na spojení
<code>WITH_NAME_SPACE_CALLS</code>	pokud chcete podporovat klienty ncpfs
<code>INTERNAL_RIP_SAP</code>	chcete, aby mars_nwe poskytoval RIP/SAP směrování?
<code>SHADOW_PWD</code>	používáte stínová hesla?

Implicitní hodnoty pravděpodobně budou v pořádku, ale i tak byste je měli zkontrolovat.

Potom pomocí následujících příkazů

```
# make
# make install
```

sestavíte servery a nainstalujete je do příslušných adresářů. Instalační skript také nainstaluje konfigurační soubor `/etc/nwserver.conf`.

Konfigurace serveru

Konfigurace je velice jednoduchá. Stačí upravit soubor `/etc/nwserver.conf`. Formát tohoto souboru vám možná na první pohled bude připadat tajemný, ale ve skutečnosti je dobře srozumitelný. Soubor obsahuje několik jednořádkových konfiguračních záznamů. Každý řádek je vymezen bílým místem a začíná číslem, které udává jeho obsah. Veškeré znaky následující za znakem `#` jsou považovány za komentář a jsou ignorovány. Martin Stover dodává jako součást balíku vzorový konfigurační soubor, ale já vám nyní jako alternativu představím zjednodušený příklad tohoto souboru.

```

# VOLUMES (max. 5)
# Only the SYS volume is compulsory. The directory containing the SYS
# volume must contain the directories: LOGIN, PUBLIC, SYSTEM, MAIL.
# The 'i' option ignores case.
# The 'k' option converts all filenames in NCP requests to lowercase.
# The 'm' option marks the volume as removable useful for cdroms etc.
# The 'r' option set the volume to read-only.
# The 'o' option indicates the volume is a single mounted filesystem.
# The 'P' option allows commands to be used as files.
# The 'O' option allows use of the OS/2 namespace
# The 'N' option allows use of the NFS namespace
# The default is upper case.
# Syntax:
# 1 <Volumename> <Volumepath> <Options>

1      SYS          /home/netware/SYS/          # SYS
1      DATA        /home/netware/DATA/        k          # DATA
1      CDROM        /cdrom                      kmr       # CDROM

# SERVER NAME
# If not set then the Linuxhostname will be converted to upper case
# and used. This is optional, the hostname will be used if this is
# not configured.
# Syntax:
# 2 <Servername>

2      LINUX_FS01

# INTERNAL NETWORK ADDRESS
# The Internal IPX Network Address is a feature that simplifies IPX
# routing for multihomed hosts (hosts that have ports on more than
# one IPX network).
# Syntax:
# 3 <Internal Network Address> [<Node Number>]
# or:
# 3 auto
#
# If you use 'auto' then your host IP address will be used. NOTE:

```

```
# this may be dangerous, please be sure you pick a number unique to
# your network.
# Addresses are 4byte hexadecimal (the leading 0x is required).

3 0x49a01010 1

# NETWORK DEVICE(S)
# This entry configures your IPX network. If you already have your
# IPX network configured then you do not need this. This is the same
# as using ipx_configure/ipx_interface before you start the server.
# Syntax:
# 4 <IPX Network Number> <device_name> <frametype> [<ticks>]
#           Frame types: ethernet_ii, 802.2, 802.3, SNAP

4 0x39a01010 eth0 802.3 1

# SAVE IPX ROUTES AFTER SERVER IS DOWNED
# Syntax:
# 5 <flag>
# 0 = don't save routes, 1 = do save routes

5 0

# NETWARE VERSION
# Syntax:
# 6 <version>
# 0 = 2.15, 1 = 3.11

6 1

# PASSWORD HANDLING
# Real Novell DOS clients support a feature which encrypts your
# password when changing it. You can select whether you want your
# mars server to support this feature or not.
# Syntax
# 7 <flag>
# <flag> is:
# 0 to force password encryption. (Clients can't change
```

```
# password)
# 1 force password encryption, allow unencrypted password
# change.
# 7 allow non-encrypted password but no empty passwords.
# 8 allow non-encrypted password including empty passwords.
# 9 completely unencrypted passwords (doesn't work with OS/2)

7 1

# MINIMAL GID UID rights
# permissions used for attachments with no login. These permissions
# will be used for the files in your primary server attachment.
# Syntax:
# 10 <gid>
# 11 <uid>
# <gid> <uid> are from /etc/passwd, /etc/groups

10 200
11 201

# SUPERVISOR password
# May be removed after the server is started once. The server will
# encrypt this information into the bindery file after it is run.
# You should avoid using the 'root' user and instead use another
# account to administer the mars fileserver.
#
# This entry is read and encrypted into the server bindery files, so
# it only needs to exist the first time you start the server to
# ensure that the password isn't stolen.
#
# Syntax:
# 12 <Supervisor-Login> <Unix username> [<password>]

12 SUPERVISOR terry secret

# USER ACCOUNTS
# This associates NetWare logins with unix accounts. Password are
# optional.
```

```
# Syntax:
#   13 <User Login> <Unix Username> [<password>]

13 MARTIN martin
13 TERRY terry

# LAZY SYSTEM ADMIN CONFIGURATION
# If you have a large numbers of users and could not be bothered
  using
# type 13 individual user mappings, you can automatically map
# mars_nwe
# logins to Linuxuser names. BUT, there is currently no means of
# making use of the Linuxlogin password so all users configured this
# way are will use the single password supplied here. My
# recommendation is not to do this unless security is absolutely no
# concern to you.
# Syntax:
#   15 <flag> <common-password>
#   <flag> is:      0 - don't automatically map users.
#                  1 - do automatically map users not configured
#                  above.
#                  99 - automatically map every user in this way.

15 0 duzzenmatta

# SANITY CHECKING
# mars_nwe will automatically ensure that certain directories exist
# if you set this flag.
# Syntax:
#   16 <flag>
#   <flag> is 0 for no, don't, or 1 for yes, do.

16 0

# PRINT QUEUES
# This associates NetWare printers with unix printers. The queue
# directories must be created manually before printing is attempted.
# The queue directories are NOT lpd queues.
# Syntax:
```

```
#    21 <queue_name> <queue_directory> <unix_print_cmd>

21  EPSON  SYS:/PRINT/EPSON lpr -h
21  LASER  SYS:/PRINT/LASER lpr -Plaser

# DEBUG FLAGS
# These are not normally needed, but may be useful if are you
# debugging a problem.
# Syntax:
#    <debug_item> <debug_flag>
#
#    100 = IPX KERNEL
#    101 = NWSERV
#    102 = NCPSEV
#    103 = NWCONN
#    104 = start NWCLIENT
#    105 = NWBIND
#    106 = NWROUTED
#
#    0 = disable debug, 1 = enable debug

100 0
101 0
102 0
103 0
104 0
105 0
106 0

# RUN NWSERV IN BACKGROUND AND USE LOGFILE
# Syntax:
#    200 <flag>
#
#    0 = run NWSERV in foreground and don't use logfile
#    1 = run NWSERV in background and use logfile

200 1

# LOGFILE NAME
# Syntax:
```

```
# 201 <logfile>

201 /tmp/nw.log

# APPEND LOG OR OVERWRITE
# Syntax:
# 202 <flag>
# 0 = append to existing logfile
# 1 = overwrite existing logfile

202 1

# SERVER DOWN TIME
# This item sets the time after a SERVER DOWN is issued that the
# server really goes down.
# Syntax:
# 210 <time>
# in seconds. (defaults 10)

210 10

# ROUTING BROADCAST INTERVAL
# The time is seconds between server broadcasts
# Syntax:
# 211 <time>
# in seconds. (defaults 60)

211 60

# ROUTING LOGGING INTERVAL
# Set how many broadcasts take place before logging of routing
# information occurs.
# Syntax:
# 300 <number>

300 5

# ROUTING LOGFILE
```



```
# Set the name of the routing logfile
# Syntax:
#   301 <filename>

301 /tmp/nw.routes

# ROUTING APPEND/OVERWRITE
# Set whether you want to append to an existing log file or
# overwrite it.
# Syntax:
#   302 <flag>
#       <flag> is 0 for append, 1 for create/overwrite

302 1

# WATCHDOG TIMING
# Set the timing for watchdog messages that ensure the network is
# still alive.
# Syntax:
#   310 <value>
#       <value> = 0 - always send watchdogs
#               < 0 - (-ve) for disable watchdogs
#               > 0 - send watchdogs when network traffic
#                   drops below 'n' ticks

310 7

# STATION FILE
# Set the filename for the stations file which determine which
# machines this fileserver will act as the primary fileserver for.
# The syntax of this file is described in the 'examples' directory
# of the source code.
# Syntax:
#   400 <filename>

400 /etc/nwserv.stations

# GET NEAREST FILESERVER HANDLING
```

```
# Set how SAP Get Nearest Fileserver Requests are handled.
# Syntax:
#     401 <flag>
#     <flag> is: 0 - disable 'Get Nearest Fileserver' requests.
#               1 - The 'stations' file lists stations to be
#                   excluded.
#               2 - The 'stations' file lists stations to be
#                   included.

401 2
```

Spuštění serveru

Pokud jste nastavili server tak, aby čekal na externí programy, které nastaví vaši síť a/nebo budou poskytovat směrovací funkce, pak byste měli tyto programy spustit ještě před spuštěním serveru. Za předpokladu, že jste nastavili server tak, aby sám nakonfiguroval rozhraní a poskytoval směrovací služby, stačí spustit pouze následující příkaz:

```
# nwserv
```

Otestování serveru

Abyste server otestovali, měli byste se nejprve zkusit připojit a přihlásit do vaší sítě z klienta NetWare. Pak nastavte `CAPTURE` z klienta a vyzkoušejte tisk. Budete-li v obou případech úspěšní, pak server funguje správně.

9.2 Balík `lward`

Aleš Dryák <A.Dryak@sh.cvut.cz> vytvořil balík `lward`, díky němuž může linuxový server fungovat jako souborový server využívající protokol NCP.

Aleš dal tomuto balíku název *lward*, protože je to zkratka *Lin Ware Daemon*.

9.2.1 Kompatibilita balíku `lward`

Server `lward` je schopen poskytovat část služeb protokolu Novell NCP. Umí posílat zprávy, ale neposkytuje žádné funkce týkající se tisku. V současné době příliš nefunguje s klienty Windows95 nebo Windows NT. Server `lward` spoléhá při vytvoření a aktualizaci IPX-směrování a tabulek SAP na externí programy. Nevychovaní klienti mohou způsobit jeho pád. Je také důležité upozornit, že součástí balíku nejsou utility pro převod názvů souborů.

Server funguje pod příkazovými interprety NETX a VLM NetWare.

9.2.2 Získání balíku lwarded

Balík `lwarded` lze přeložit pod jádrem verze 1.2.0, doporučuji však verzi 1.2.13, protože v tom případě nejsou nutné žádné záplaty. S příchodem verze 1.3.* došlo k určitým změnám protokolu IPX, takže pro správnou funkci tohoto balíku nyní potřebujete záplaty. Příslušné záplaty jsou součástí nových verzí jádra, takže i když musíte používat alfa verzi, bude vám `lwarded` fungovat správně.

Balík `lwarded` získáte pomocí anonymní služby FTP na adrese: `ftp://klokan.sh.cvut.cz/pub/linux/linware/`

nebo na

`ftp://sunsite.unc.edu/pub/Linux/system/network/demons` nebo na jejich zrcadlech. Aktuální verze v době psaní knihy nesla označení `lwarded-0.95.tar.gz`.

9.2.3 Sestavení balíku lwarded

Rozbalení balíku lwarded

Například:

```
# cd /usr/src
# tar xvpfz lwarded-0.95.tar.gz
```

Sestavení jádra s podporou Ethernetu a protokolu IPX

Pokud používáte alfa verzi 1.3.*, pak by to měla být verze 1.3.17 nebo novější, protože již obsahuje příslušné opravy chyb. Opravy verzí starší než 1.3.17 vyžadují manuální instalaci (*některé informace o tomto zákroku najdete v souboru `INSTALL`, který je součástí balíku `lwarded`*). Při instalaci oprav jádra verze 1.3.17 nebo vyšší zadejte následující příkaz:

```
# make patch
```

Po provedení nezbytných úprav je třeba se přesvědčit, že jste jádro přeložili s podporou protokolu IPX. Ve verzi jádra 1.2.13 je třeba pouze odpovědět kladně (Y) na dotaz „The IPX protocol“, jako v následujícím výpise:

```
...
...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
...
...
```

Do novějších verzích jádra byl převzat podobný proces, i když skutečný text výzev se mohl mírně změnit.

Také je třeba zahrnout příslušný ovladač pro vaši ethernetovou kartu. Pokud nevíte, jak na to, pak byste si měli přečíst dokument *Ethernet-HOWTO*.

Pak můžete přistoupit ke kompilaci jádra. Nezapomeňte po skončení spustit příkaz `lilo`, který nainstaluje bootovací manažer.

Kompilace a instalace balíku `lward`

Před kompilací balíku `lward` byste měli nejprve, je-li to nutné, upravit soubor `server/config.h`. Tento soubor obsahuje různá nastavení, která budou řídit způsob chování serveru při běhu. Implicitní hodnoty jsou nastaveny rozumně, i když bude možná dobré zkontrolovat, zda adresáře určené pro log soubory a konfigurační soubory vyhovují vašemu systému.

```
# make depend
# make
# make install
```

Zjistil jsem, že si příkaz „`make depend`“ stěžoval na to, že v mém systému nenašel soubor `float.h`, i přesto však fungoval. Také jsem při pokusu o kompilaci pomocí překladače `gcc 2.6.3` zjistil, že v souboru `lib/ipxkern.c` je třeba změnit řádek

```
#include <net/route.h>

na

#include <net/if_route.h>
```

protože tento soubor někdy změnil svůj název.

Příkaz „make install“ se pokusí nainstalovat server a směrovacího démona do adresáře `/usr/sbin`, program `lwpasswd` do adresáře `/usr/bin`, pomocné programy protokolu IPX budou nainstalovány do adresáře `/sbin` a v neposlední řadě manuálové stránky přijdou do adresáře `/usr/man`. Pokud vám některé z těchto umístění nevyhovuje, pak budete muset změnit cílové adresáře v souboru `Makefile`.

9.2.4 Konfigurace a používání balíku `lward`

Nyní trochu legrace.

Konfigurace sítě IPX

Nejprve je nutné nastavit ethernetová rozhraní tak, aby podporovala IPX-sítě, které bude podporovat váš server. K tomu budete potřebovat znát síťové IPX-adresy každého vašeho LAN-segmentu, které ethernetové zařízení (`eth0`, `eth1` atd.) je na kterém segmentu, jaký typ rámce (`802.3`, `EtherII` atd.) každý LAN-segment používá a které interní síťové adresy má váš server používat (to je nutné zejména v případě, že bude váš server obsluhovat více než jeden LAN-segment). Konfigurace serveru, který je na dvou různých segmentech se síťovými IPX-adresami `0x23a91300` a `0x23a91301` a interní síťovou adresou `0xbdefaced`, by mohla vypadat následovně:

```
# ipx_internal_net add BDEFACED 1
# ipx_interface add eth0 802.3 23a91300
# ipx_interface add eth1 etherii 23a91301
```

Spuštění směrovacích démonů

Vlastní software jádra ve skutečnosti provádí doručování IPX-paketů stejně jako v případě protokolu IP, ale jádro potřebuje ke správě aktualizací směrovací tabulky dodatečné programy. V případě protokolu IPX jsou potřeba dva démoni, z nichž oba jsou dodáváni s balíkem `lward`. Démon `ipxripd` spravuje IPX směrovací informace a démon `ipxsapd` řídí SAP-informace. Při spuštění těchto dvou démonů je pouze třeba zadat umístění jejich log-souborů:

```
# ipxripd /var/adm/ipxrip
# ipxsapd /var/adm/ipxsap
```

Konfigurace serveru `lward`

K tomu, aby se mohl uživatel přihlásit k serveru `lward`, je třeba ručně nastavit dva soubory. Jsou to:

```
/etc/lwpasswd
```

V tomto souboru uchovává server LinWare informace o uživatelských účtech. Program `lwpasswd` je udržuje aktuální. V nejjednodušší podobě vypadá soubor `/etc/lwpasswd` následovně:

```
ales:
terryd:
guest:
```

Jeho formát odpovídá seznamu přihlašovacích id následovaných znakem „:“ a zašifrovaným heslem. Platí zde tato dvě důležitá pravidla: Pokud zde není uvedeno žádné zašifrované heslo, je příslušný účet bez hesla. Uživatelé serveru LinWare musí mít linuxové účty, to znamená, že každý uživatel, který se objeví v souboru `/etc/lwpasswd`, musí mít svůj záznam také v souboru `/etc/passwd` a jedině uživatel `root` může změnit heslo jiného uživatele serveru LinWare. Pokud jste přihlášení jako `root`, můžete podle následujícího příkladu změnit heslo některého uživatele serveru LinWare:

```
# lwpasswd rodg
Changing password for RODG
Enter new password:
Re-type new password:
Password changed.
/etc/lwvtab
```

Tento soubor LinWare definuje tabulku svazků a uchovává informace o tom, které adresáře mají být přístupné uživatelům serveru LinWare (podobnou funkci má soubor `/etc/exports` v systému NFS). Následuje jednoduchý příklad formátu tohoto souboru:

```
SYS          /lwfs/sys
DATA         /lwfs/data

HOME        /home
```

Formát je jednoduchý: Název svazku následovaný bílým místem a exportovaným linuxovým adresářem. Soubor musí obsahovat **minimálně** záznam pro svazek `SYS`, aby se mohl server spustit. Chcete-li, aby i uživatelé DOSu mohli využívat server LinWare jako svůj primární server, musíte v adresáři, který exportujete jako svazek `SYS`, vytvořit standardní adresářovou strukturu svazku `SYS`. Protože jsou tyto soubory vlastnictvím firmy Novell, měli byste mít příslušnou licenci. Budou-li vaši uživatelé používat novellovský souborový server jako svůj primární server, pak tato licence není nutná.

Spuštění serveru lwared

```
# lwared
```

Je to takřka antiklimax, není-liž pravda? Dobře, takže jste dostali otázku, mám pravdu? Jaký je název oznamovaného serveru? Pokud jste server spustili výše uvedeným způsobem, bude inzerovaný název serveru LinWare záviset na tom, co vrátí linuxový program `hostname`. Pokud chcete používat jiný název, můžete ho předat serveru při jeho spuštění. Například:

```
# lwared -nlinux00
```

spustí server s názvem `linux00`.

Otestování serveru lwared

Ze všeho nejdříve vyzkoušejte, zda se váš server LinWare objeví v seznamu, který vrátí program `slist` spuštěný z dosového klienta ve vaší síti. Program `slist` je uložen na svazku `SYS` novellovského souborového serveru, takže ho musíte spustit z počítače, který je k němu již přihlášen. Pokud neuspějete, pak zkontrolujte, zda běží současně programy `ipxsapd` i `lwared`. Po úspěšném provedení příkazu `slist` se zkuste přihlásit k serveru a připojit nějaký svazek:

```
C:> attach linux00/ales
...
...
C:> map l:=linux00/data:
C:> l:
```

Potom by mělo být možné zacházet s novou mapou stejně jako s kteroukoliv jinou mapou. Přístupová práva k souborům, které budete mít, budou záviset na právech přidělených účtu `linux`, který se shoduje s vaším přihlášením k serveru LinWare.

10 Konfigurace linuxového počítače jako novellovského tiskového klienta

Balík `ncpfs` obsahuje dva malé programy, které umožňují posílat tisk z linuxového počítače na tiskárnu připojenou k novellovskému tiskovému serveru. Příkaz `nprint` umožňuje poslat soubor do tiskové fronty serveru NetWare. Příkaz `pqlist` vypíše seznam dostupných tiskových front na serveru NetWare.

Tyto programy získáte a nainstalujete podle výše uvedených instrukcí ohledně NCP-klienta.

Oba příkazy vyžadují zadání uživatelského jména a hesla, takže si možná budete chtít vytvořit skripty příkazového interpretu, které by vám celý proces tisku ulehčily:

Mohly by vypadat třeba takto:

```
# pqlist -S ACCT_FS01 -U guest -n
# nprint -S ACCT_FS01 -q LASER -U guest -n filename.txt
```

Přihlašovací syntaxe je stejná jako u příkazu `ncpmount`. Ve výše uvedeném příkladu předpokládáme, že na souborovém serveru `ACCT_FS01` existuje účet `quest` bez hesla, dále že existuje tisková fronta `LASER` a uživatel `quest` do ní může posílat tiskové úlohy.

11 Konfigurace linuxového počítače jako novellovského tiskového serveru

Součástí balíku `ncpfs` je program, který udělá z vašeho linuxového počítače tiskový server v síti NetWare. Instrukce, jak ho získat a přeložit, najdete výše ve stati „Klient NetWare“.

11.1 Nezbytné předpoklady

Konfigurace tohoto programu je poměrně jednoduchá, ale počítá, že máte dokončenou konfiguraci tiskárny pod Linuxem. Toto téma je hlouběji rozebíráno v dokumentu `Printing-HOWTO`.

11.2 Konfigurace

Jakmile máte fungující konfiguraci tiskárny a nainstalovanou utilitu `pserver`, zbývá přidat do `rc` souborů příkazy, které ji spustí.

Příkaz, který použijete, bude záviset na tom, jak chcete aby fungoval, ale v tomto jednoduchém příkladu postačí následující zápis:

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET
```


Tento příklad požádá utilitu `pserver`, aby se přihlásila k souborovému serveru `ACCT_01` pod uživatelským jménem `LASER` s heslem `secret` a převzala úlohy z tiskové fronty `LASERJET`. Po přijetí příchozí tiskové úlohy předá tuto tiskovou úlohu pomocí implicitního tiskového příkazu nebo programu `lpr` linuxovému tiskovému démonu.

Pokud byste chtěli, mohli byste k přijímání a tisku tiskových úloh používat jakýkoliv linuxový příkaz. Konkrétní tiskový příkaz umožňuje zadat argument `-c`. Například příkaz:

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET
-c "lpr -Plaserjet"
```

by provedl to samé, jako ten předchozí pouze s tím rozdílem, že by úlohu neposlal implicitnímu příkazu, ale na tiskárnu `laserjet`.

12 Přehled uživatelských a administrativních příkazů balíku `ncpfs`

Nové verze Volkerova balíku `ncpfs` obsahují paletu uživatelských a administrativních příkazů, které byste mohli využít. Tyto nástroje jsou překládány a instalovány v rámci instalace balíku `ncpfs`, takže pokud jste se neřídili instrukcemi uvedenými ve stati „Klient NetWare“ vedoucími k jejich instalaci, potom tak nyní učíte.

Podrobné informace najdete v dodávaných manuálových stránkách. Zde uvádíme pouze stručný popis jednotlivých příkazů.

12.1 Uživatelské příkazy

<code>ncopy</code>	Network Copy – umožňuje efektivní kopírování souborů pomocí funkce NetWare místo kopírování přes síť.
<code>nprint</code>	Network Print – umožňuje tisk souboru do tiskové fronty NetWare na serveru NetWare.
<code>nsend</code>	Network Send – umožňuje posílat zprávy jiným uživatelům na serveru NetWare.
<code>nwbols</code>	List Bindery Objects – umožňuje vypsát seznam obsahu bindery serveru NetWare.

<code>nwboprops</code>	List Properties of a Bindery Object – zpřístupní vlastnosti objektu NetWare bindery.
<code>nwbpset</code>	Set Bindery Property – slouží k nastavení vlastností objektu NetWare bindery.
<code>nwbpvalues</code>	Print NetWare Bindery Objects Property Contents – umožňuje vytisknout obsah vlastností objektu NetWare bindery.
<code>nwfsinfo</code>	Fileserver Information – vytiskne některé souhrnné informace o serveru NetWare.
<code>nwpasswd</code>	NetWare Password – umožňuje změnit hesla uživatelů serveru NetWare.
<code>nwrights</code>	NetWare Rights – zobrazí práva přidělená konkrétnímu souboru nebo adresáři.
<code>nwuserlist</code>	Userlist – vypíše seznam uživatelů aktuálně přihlášených k souborovému serveru NetWare.
<code>pqlist</code>	Print Queue List – zobrazí obsah tiskové fronty serveru NetWare.
<code>slist</code>	Server List – vypíše seznam známých serverů NetWare.

12.2 Administrativní nástroje

<code>nwbocreate</code>	Create a Bindery Object – umožňuje vytvořit objekt NetWare bindery.
<code>nwborm</code>	Remove Bindery Object – umožňuje smazat objekt NetWare bindery.
<code>nwbpadd</code>	Add Bindery Property – umožňuje nastavit hodnotu existující vlastnosti objektu NetWare bindery.
<code>nwbpcreate</code>	Create Bindery Property – umožňuje vytvořit novou vlastnost existujícího objektu NetWare bindery.
<code>nwbprm</code>	Remove Bindery Property – umožňuje odstranit vlastnost z objektu NetWare bindery.
<code>nwgrant</code>	Grant Trustee Rights – umožňuje přidělit správcovská (trustee) práva adresáři na souborovém serveru NetWare.
<code>nwrevoke</code>	Revoke Trustee Rights – umožňuje odstranit správcovská (trustee) práva adresáře na souborovém serveru NetWare.

13 Konfigurace protokolu PPP s podporou protokolu IPX

Nové verze démona `pppd` pro Linux umožňují přenášet IPX-pakety po sériové PPP-lince. Potřebujete k tomu minimálně verzi `ppp-2.2.0d`. Informace o tom, kde ji můžete získat, najdete v dokumentu *PPP-HOWTO*. Když budete tohoto démona překládat, musíte se ujistit, že soubor `/usr/src/linux/pppd-2.2.0f/pppd/Makefile.linux` obsahuje následující dva řádky, které zapínají podporu protokolu IPX.

```
IPX_CHANGE = 1
USE_MS_DNS = 1
```

Volba `IPX_CHANGE` začlení podporu IPX do protokolu PPP. Volba `USE_MS_DNS` umožňuje počítačům s Microsoft Windows95 provádět vyhledávání názvů (Name Lookups).

Pokud chcete tuto podporu rozchodit, musíte vědět, jak ji nakonfigurovat.

Je to možné udělat několika způsoby, ale zde si popíšeme dva, o kterých se mi podařily sehnat nějaké informace. Ani jeden z nich jsem nezkoušel, takže považujte tuto stať za experimentální a podaří-li se vám něco rozběhnout, pak mi dejte vědět.

13.1 Konfigurace serveru IPX/PPP

Nejprve je nutné nastavit váš linuxový počítač jako server IP/PPP. Nemějte strach, není to nic obtížného. Znovu stačí, když se budete řídit instrukcemi uvedenými v dokumentu *PPP-HOWTO*. Potom je ještě potřeba provést některé úpravy, aby při stejné konfiguraci fungovala i podpora protokolu IPX.

13.1.1 První kroky

Jedním z prvních kroků je konfigurace vašeho linuxového počítače jako IPX-routeru podle popisu, který uvádím dříve v tomto dokumentu. Nebudete muset používat příkaz `ipx_route`, protože konfiguraci rozhraní `ppp`, stejně jako IP, za vás udělá démon `pppd`. Pokud běží démon `ipxd`, automaticky rozpozná jakákoliv nová IPX-rozhraní a rozšíří pro ně trasy. Takto budou vaši vytáčení hostitelé po připojení automaticky viditelní pro ostatní počítače.

13.1.2 Návrh

Pokud fungujete jako server, budete zodpovědní za přiřazení síťových adres všem PPP-spojením po jejich navázání. Tento bod je velice důležitý, protože každé PPP-spojení bude IPX-sítí a bude mít jedinečnou síťovou IPX-adresu. To znamená, že se musíte rozhodnout, jak a jaké ad-

resy budete alokovat. Běžně se alokuje jedna síťová IPX-adresa pro každé sériové zařízení, které bude podporovat IPX/PPP. Mohli byste alokovat síťovou IPX-adresu v závislosti na přihlašovacím id uživatele, ale pro to neexistuje žádný konkrétní důvod.

Budu předpokládat, že jste provedli vše potřebné a že budeme používat dvě sériová zařízení (modemy). V tomto vymyšleném příkladu jsem jim přiřadil následující adresy:

zařízení	síťová adresa IPX
ttyS0	0xABCDEF00
ttyS1	0xABCDEF01

13.1.3 Konfigurace démona *pppd*

V souboru `/etc/ppp/options.ttyS0` uveďte následující záznamy:

```
ipx-network 0xABCDEF00
ipx-node 2:0
ipxcp-accept-remote
```

a v souboru `/etc/ppp/options.ttyS1` tyto:

```
ipx-network 0xABCDEF01
ipx-node 3:0
ipxcp-accept-remote
```

Tím požádáte démona `pppd`, aby po navázání spojení alokoval příslušnou síťovou IPX adresu, nastavil lokální číslo uzlu 2 nebo 3 a nechal vzdálený uzel přepsat uzlem, kterým myslí, že je. Všimněte si, že každá z těchto adres je hexadecimální číslo a že na začátku síťové adresy je vyžadován řetězec `0x`, nikoli však na začátku adresy uzlu.

Tuto informaci lze nastavit také na jiných místech. Pokud máte pouze jeden vytáčecí modem, pak by mohl být tento záznam součástí souboru `/etc/ppp/options`. Nebo lze tuto informaci předat démonu `pppd` na příkazové řádce,

13.1.4 Otestování konfigurace serveru

K otestování konfigurace serveru budete potřebovat funkční konfiguraci klienta. Když volající vytočí číslo, přihlásí se a spustí se démon `pppd`, přidělí serveru síťovou adresu, oznámí klientovi číslo uzlu serveru a sjedná číslo uzlu klienta. Když potom démon `ipxd` detekuje nové rozhraní, měl by být klient schopen navázat IPX-spojení se vzdálenými hostiteli.

13.2 Konfigurace klienta IPX/PPP

Při konfiguraci klienta závisí to, zda nakonfigurujete váš linuxový počítač jako IPX-router na tom, zda máte lokální počítačovou síť, pro kterou chcete, aby tento počítač fungoval jako IPX-router. Máte-li samostatný počítač, který se připojuje k vytáčenému IPX/PPP-serveru, pak nebudete potřebovat spouštět démona `ipxd`, ale máte-li lokální počítačovou síť a chcete, aby mohly IPX-router využívat všechny počítače v této síti, pak musíte nakonfigurovat a spustit démona `ipxd` tak, jak bylo popsáno. Tato konfigurace je mnohem jednodušší, protože nemusíte nastavovat více sériových zařízení.

13.2.1 Konfigurace démona `pppd`

Nejednodušší konfigurace je ta, která umožňuje serveru poskytovat veškeré informace o konfiguraci IPX-sítě. Tato konfigurace by byla kompatibilní s výše popsanou konfigurací serveru.

Znovu je třeba přidat nějaké záznamy do souboru `/etc/ppp/options`:

```
ipxcp-accept-network
ipxcp-accept-remote
ipxcp-accept-local
```

Tyto záznamy říkají démonu `pppd`, aby se choval zcela pasivně a přijal všechny detaily konfigurace od serveru. Pro servery, které neposkytují podrobné informace, zde můžete také uvést implicitní hodnoty pomocí záznamů `ipx-network` a `ipx-node`.

13.2.2 Testování klienta IPX/PPP

K otestování klienta budete potřebovat fungující server, se kterým byste se mohli spojit. Po vytvoření a spuštění démona `pppd` by měl program `ifconfig` vypsat podrobnosti konfigurace protokolu IPX na vašem zařízení `ppp0` a měl by fungovat příkaz `ncpmount`.

Nejsem si jistý, zda je pro spojení se vzdálenými souborovými servery nutné ručně přidat IPX-router. Zdá se to pravděpodobné. Budu však vděčen komukoliv, kdo s touto konfigurací pracuje a sdělí mi příslušné informace.

14 IPX-tunel přes protokol IP

Spousta z vás se dostane do situace, kdy máte dvě novellovské lokální počítačové sítě a mezi nimi pouze jediné IP-spojení. Jistě vás napadne, jak budete s tímto uspořádáním hrát deathmatch v hře DOOM pro DOS? Andreas Godzina [<ag@agsc.han.de>](mailto:ag@agsc.han.de) má pro vás odpověď v podobě nástroje zvaného `ipxtunnel`.

Program `ipxtunnel` poskytuje jakýsi most pro protokol IPX tím, že umožní zapouzdření IPX-paktů do IPXD/IP-datagramů, takže mohou být přenášeny po TCP/IP-spojení. Odposlouchává IPX-pakety a jakmile na nějaký narazí, zabalí ho do TCP/IP datagramu a přesměruje ho na vzdálenou IP-adresu, kterou zadáte. Aby to fungovalo, musí samozřejmě na počítači, na který směřujete zapouzdřený protokol IPX, také běžet stejná verze programu `ipxtunnel`, jakou používáte vy.

14.1 Získání programu ipxtunnel

Program `ipxtunnel` získáte na adrese

```
ftp://sunsite.unc.edu/pub/Linux/system/network/demons.
```

14.2 Přeložení programu ipxtunnel

Program `ipxtunnel` přeložíte pomocí následující sekvence příkazů:

```
# cd /usr/src
# tar xvfz ../ipxtunnel.tgz
# cd ipxtunnel
# make
```

14.3 Konfigurace programu ipxtunnel

Konfigurace programu `ipxtunnel` není vůbec složitá. Řekněme, že počítač vašich přátel se nazývá `gau.somewhere.com` a váš počítač `gim.sw.edu`. Program `ipxtunnel` používá konfigurační soubor jménem `/etc/ipxtunnel.conf`. Tento soubor umožňuje zadat implicitní UDP-port, který se má použít pro TCP/IP-spojení, kam se mají posílat zapouzdřená data, a určit které z vašich lokálních rozhraní má `ipxtunnel` odposlouchávat a doručovat na ně IPX-pakety.

Jednoduchý konfigurační soubor vypadá asi takto:

```
#
# Soubor /etc/ipxtunnel.conf pro gim.sw.edu
#
# Používaný UDP port:                               (implicitně 7666)
port 7777
#
# Vzdálený server, kam budete posílat pakety:
remote gau.somewhere.com
```

```
#
# Lokální rozhraní, kde čekáme na IPX pakety: (implicitně eth0)
interface eth0
interface eth1
```

Samozřejmě, že počítač na druhém konci by měl podobný konfigurační soubor, v němž by byl tento počítač uveden jako `remote` hostitel.

14.4 Testování a používání programu `ipxtunnel`

Program `ipxtunnel` funguje jako IPX-most, takže by IPX-sítě měly být na obou koncích spojeny stejně. Andreas Godzina nikdy netestoval svůj program v prostředí, které skutečně podporuje novellovské souborové servery, takže pokud se vám to podaří v reálném prostředí, dejte Andreasovi vědět, zda to funguje.

Běží-li program `ipxtunnel`, měla by jít spustit hra DOOM na obou koncích spojení pracujících v režimu IPX a tyto dva počítače by o sobě měly navzájem vědět.

Andreas Godzina používal tento kód pouze na vysokorychlostních linkách a netvrdí, že na pomalejších bude stejně rychlý. I v tomto případě mu napište, co vám fungovalo a co nikoliv.

15 Komerční podpora protokolu IPX pro Linux

15.1 Network Desktop společnosti Caldera

Společnost Caldera Inc. produkuje distribuci Linuxu, která obsahuje velké množství komerčně podporovaných vylepšení včetně podpory klienta Novell NetWare. Základem je respektovaná distribuce Red Hat a společnost Caldera k ní přidala svůj balík produktů nazvaný „Network Desktop“. Podpora NetWare obsahuje plnohodnotného klienta Novell NetWare založeného na technologii licencované společností Novell Corporation. Tento klient umožňuje úplný přístup k souborovým serverům Novell 3.x a 4.x a disponuje službami typu NDS (NetWare Directory Service) a šifrování RSA.

Mnohem více informací o tomto produktu a jeho získání najdete na WWW-serveru společnosti Caldera <http://www.caldera.com/>.

Pokud pracujete v prostředí NetWare 4.x a/nebo NDS, je pro vás Caldera NetWare Client jediným možným řešením.

Také pokud vlastníte obchodně důležitou aplikaci využívající podpory Novellu v Linuxu, měli byste o produktu společnosti Caldera popřemýšlet.

16 Některé často kladené dotazy

Kde najdu komerčně podporovaný IPX-softwar pro Linux?

Společnost Caldera Corporation nabízí plně licencovaného a plně podporovaného klienta NetWare 3.x a 4.x. Informace o něm získáte na adrese <http://www.caldera.com/>.

1. Funguje IPX-softwar se sítěmi ArcNet/Token Ring/atd?

Linuxový IPX-softwar funguje s rozhraními ArcNet a TokenRing. Neslyšel jsem o nikom, kdo by ho zkoušel s AX.25. Konfigurace je podobná konfiguraci Ethernetu, jen s tou výjimkou, že budete muset tam kde je to nutné nahradit příslušné názvy zařízení na místě „eth0“ a příslušné hardwarové adresy.

2. Jak nastavím víc než jedno IPX-rozhraní?

Pokud máte ve vašem počítači víc než jedno rozhraní, měli byste každé z nich nastavit pomocí příkazu `ipx_interface`. Neměli byste používat konfiguraci „plug and play“.

3. Jaké mám zvolit IPX-adresy?

Sítě IPX jsou podobné, ne však identické se sítěmi IP. Základní rozdíl spočívá ve způsobu, jakým používají adresy. Protokol IPX nepoužívá koncept podsítí, takže druh přiřazení mezi síťovými adresami a sítěmi je odlišný. Pravidla jsou poměrně jednoduchá:

- Každá síťová IPX-adresa musí být jedinečná v rámci WAN. Sem patří interní síťové adresy (Internal Network Addresses). Spousta organizací, které používají protokol IPX v dálkových sítích, bude mít nějaký druh adresovacího standardu, který musíte dodržovat.
- Každá adresa hostitele v samostatné síti musí být jedinečná. To znamená, že každý hostitel v každé IPX-síti musí mít přiřazenu jedinečnou adresu. V případě ethernetové sítě to není problém, protože každá karta má jedinečnou adresu. V případě sítí IPX/PPP je třeba přidělit jedinečné adresy všem hostitelům v síti, bez ohledu na to, ke kterému konci jsou připojeni. Hostitelské adresy nemusí být jedinečné v rámci dálkové sítě, protože k jedinečné identifikaci hostitele se používá síťová adresa v kombinaci s hostitelskou adresou.

4. Jaké typy rámců bych měl použít?

Existují rozmanité typy rámců, které lze použít pro protokol IPX. Nejběžnějším jsou popisovány ve stati „Některé termíny používané v tomto dokumentu“ (pod heslem „**Frame Type**“).

V případě, že instalujete váš počítač do existující sítě, pak musíte použít již zaběhnutý typ rámce, aby mohl váš počítač komunikovat s ostatními hostiteli v síti. Pokud ale zavádíte úplně novou síť, můžete k přenosu protokolu IPX použít kterýkoliv z možných protokolů. Já osobně vám pro přenos protokolu IPX i IP doporučuji typ rámce `Ethernet_II`.

5. Mým počítačům běžícím pod Windows95 se nepodařila autodetekce typu rámce.

Jak se zdá, někdy k tomu může dojít. Mohl bych vám dát spoustu rad, ale místo toho vám poradím použít manuální konfiguraci typu rámce. V každém případě je to zřejmě lepší metoda.

6. Proč se mi při konfiguraci protokolu IPX objevuje zpráva „invalid argument“ (neplatný argument)?

Pravděpodobně používáte jádro, které nepodporuje protokol IPX. Proto buď jádro znovu nastavte, nebo dvakrát zkontrolujte, že jste pomocí LILO skutečně nainstalovali a používáte nové jádro.

7. Proč se mi při konfiguraci protokolu IPX objevuje zpráva „package not installed“ (balík nebyl nainstalován)?

Pravděpodobně používáte jádro, které nepodporuje protokol IPX. Proto buď jádro znovu nastavte, nebo dvakrát zkontrolujte, že jste pomocí LILO skutečně nainstalovali a používáte nové jádro.

8. Proč dostávám od démona `pppd` zprávu „IPX support not in kernel“ (Jádro neobsahuje podporu protokolu IPX)?

Pravděpodobně jste přeložili podporu protokolu IPX jako modul a nezajistili jste jeho nahrání před démonem `pppd`.

9. Jak lze pomocí systému NFS exportovat připojený souborový systém NCP?

Abyste mohli exportovat souborový systém NCP pomocí NFS, musíte ho připojit příkazem `ncpmount` s volbou `-v`. Tato volba umožňuje připojit pouze jeden svazek souborového serveru, místo obvyklého připojení všech svazků. Když to uděláte, umožní vám démon NFS exportovat tento souborový systém obvyklým způsobem.

10. Proč v interní síti s balíkem `mars_nwe` nefunguje příkaz `slist`?

Musíte mít povolen požadavek „get nearest server“. To znamená, že záznam 401 v souboru `/etc/nwserver.conf` by měl být nulový, ledaže máte důvod neodpovídat na požadavky „get nearest server“. Pokud jen chcete, aby fungoval příkaz `slist` a neodpovídal na každý požadavek „get nearest server“, zařadte číslo interní sítě a uzlu do souboru `/etc/nwserver.stations` a nastavte záznam 401 v souboru `/etc/nwserver.conf` na hodnotu 2.

11. Funguje balík `ncpfs` s balíkem `mars_nwe`?

Martinův a Volkerův kód pomalu začínají konvergovat. Poslední verze balíku `mars_nwe` obsahuje volbu, která povoluje spolupráci s balíkem `ncpfs`. V souboru `config.h` balíku `mars_nwe` je třeba povolit volbu `WITH_NAME_SPACE_CALLS`.

12. Existuje nějaký volně šířitelný software pro DOS, který by fungoval s balíkem `mars_nwe`?

Vynalézavá otázka si žádá vynalézavou odpověď. Jsem rád, že jste ji položil, protože Martin Stover má balík, který distribuuje společně s balíkem `mars_nwe` a který nabízí volně šířitelnou dosovskou klientskou podporu pro server `mars_nwe`. Tento balík najdete na stejných místech jako server pod názvem `mars_dosutils-0.01.tgz`. Obsahuje zdrojový kód v jazyci C programů jako jsou `slist.exe`, `login.exe`, `map.exe` atd. Kód je přeložen pomocí překladače Borland(tm) C.

17 Poznámka o autorských právech

Dokument IPX-HOWTO, průvodce softwarovou podporou protokolu IPX pro Linux. Copyright (c) 1995 Terry Dawson.

Tento program je volně šířitelný. Můžete ho šířit a/nebo upravovat v souladu s 2. nebo pozdější verzí licence GNU General Public License publikovanou nadací Free Software Foundation.

Tento program je šířen s nadějí, že bude užitečný, ale BEZ JAKÝCHKOLI ZÁRUK. Dále bez záruk OBCHODOVATELNOSTI nebo ZPŮSOBILOSTI PRO KONKRÉTNÍ ÚČEL. Podrobnosti najdete v General Public License.

Společně s tímto programem byste měli obdržet kopii licence GNU General Public License. Pokud nikoli, pak si o ni napište na adresu:

Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

18 Různé a poděkování

David E. Storey <dave@tamos.gmu.edu> a Volker Lendecke <lendecke@namu01.gwdg.de> mi výrazně pomohli při získávání informací pro tento dokument. Gilbert Callaghan <gilbert@pokey.inviso.com>, David Higgins <dave@infra.com> a Chad Robinson <chadr@brtgate.brttech.com> přispěli informacemi o konfiguraci protokolu IPX/PPP. Bennie Venter <bjv@Gil-galad.paradigm-sa.com> přispěl některými užitečnými informacemi o typech rámců. Christopher Wall <vergil@idir.net> mi dal některé užitečné návrhy, jak vylepšit čitelnost a úpravu tohoto dokumentu. Axel Boldt <boldt@math.ucsb.edu> přispěl některými užitečnými nápady a jako zpětná vazba. Erik D. Olson <eriko@wrq.com> poskytl zpětnou vazbu a informace ohledně konfigurace protokolu PPP pro protokol IPX. Brian King <root@brian.library.dal.ca> přispěl dotazem do sekce FAQ (Často kladené dotazy).

„NetWare“ je registrovaná obchodní značka společnosti Novell Corporation <http://www.novell.com/>. „Caldera“ je registrovaná obchodní značka společnosti Caldera Corporation <http://www.caldera.com/>.

S pozdravem Terry Dawson, VK2KTJ <terry@perf.no.itg.telstra.com.au>.

Nicolai Langfeldt janl@math.uio.no verze 0.7, 3. listopadu 1997

Jak nastavit NFS klienty a servery.

1 Úvod

1.1 Autorská práva

(C)opyright 1997 Nicolai Langfeld. Nepravujte bez doplnění autorských práv, rozšiřujte bez omezení, ale ponechejte tento odstavec. Část FAQ je postavena na NFS FAQ, autor Alan Cox. Seznam problémů a řešení je odvozen podle seznamu, vytvořeného v IBM Corporation.

1.2 Další záležitosti

Tento dokument nebude nikdy dokončen. Pošlete mi prosím zprávy o vašich problémech a úspěších, dokument tak mohu vylepšovat. Peníze, komentáře nebo otázky pošlejte na janl@math.uio.no. Jestliže budete odesílat elektronickou zprávu, *zajistěte prosím*, aby byla návratová adresa funkční. Dostávám *spousty* dopisů a zjišťování vaší adresy může být dost pracné.

Jestliže chcete tento dokument překládat, dejte mi vědět, abych věděl, ve kterých jazycích jsem byl publikován).

Stížnosti a poděkování předejte Olafu Kirchovi, který mě k napsání dokumentu přivedl a dal mi pár dobrých návrhů :-).

Tento dokument pokrývá NFS ve verzích jádra 2.0. Ve verzích jádra 2.1 jsou značné rozdíly a změny.

1.3 Věnování

Dokument bych rád věnoval Anne Line Norheim Langfeldtové. I když jej pravděpodobně nikdy nebude číst, protože není takovým typem dívky, která by jej četla.

2 README.first

NFS (Network File System – síťový systém souborů) má tři důležité charakteristiky:

- Umožňuje na síti sdílení souborů.
- Většinou funguje dostatečně dobře.
- Otevírá rizika v zabezpečení, velmi dobře známá crackerům, kteří snadno získají přístup (čtení, zápis a smazání) ke všem vašim souborům.

Ke zmíněným záležitostem se ještě v dokumentu dostanu. Jestliže přečtete část dokumentu, věnující se zabezpečení, budete méně zranitelní vůči některým rizikům zabezpečení. Pasáže, věnující se zabezpečení jsou v některých místech dosti odborné a vyžadují určité znalosti o IP a použitých termínech. Jestliže se v termínech nevyznáte, můžete se vrátit k dokumentu o sítích nebo si sežeňte knihu o správě TCP/IP-sítě. S její pomocí se s TCP/IP lépe sžijete. Mimochodem to je dobrý nápad, když provádíte správu unixových/linuxových serverů. Konkrétně bych doporučil *TCP/IP Network Administration* od Craiga Hunta, vydávanou O'Reilly & Associates, Inc. Když knihu přečtete a pochopíte, budete mít vyšší hodnotu na trhu práce, takže určitě neztratíte ;-).

Zde naleznete dvě části, určené k odstraňování obtíží s NFS – *Seznam problémů s programem mount* a *FAQ*. Jestliže vám něco nefunguje podle předpokladů, řešení hledejte nejprve zde.

3 Nastavení NFS-serveru

3.1 Předpoklady

Před dalším čtením dokumentu si musíte být jisti, že je možné provozovat telnet mezi stroji, které budete využívat jako server a klient. Jestliže to nefunguje, musíte korektně nastavit síť (dokument networking/NET-2).

3.2 První krok

Předtím, než můžeme provést cokoli dalšího, musíme nastavit NFS server. Jestliže jste součástí nějaké úřední nebo univerzitní sítě, jistě zde již bude množství NFS serverů. Jestliže na ně máte přístup nebo jestli tento dokument čtete proto, abyste zjistili, jak přístup získat, můžete tuto část vynechat a přeskočit až na část 4.

Jestliže potřebujete server nastavit ne-linuxový, musíte si pročíst systémový manuál a zjistit, jak umožnit NFS-sluzby a export systémů souborů přes NFS. Jedna část tohoto dokumentu se této problematice věnuje pro mnoho různých systémů. Jakmile si vše urovnáte, můžete přikročit ke čtení další části. Nebo v této části ještě zůstaňte, protože některé věci, které zde odkryvám, jsou nezávislé na druhu stroje, použitého jako server.

Ti z vás, kteří tady čtou dál, budou muset nastavit množství programů.

3.3 Portmapper

Portmapperu se v Linuxu říká buď `portmap`, nebo `rpc.portmap`. Manuálová stránka na mém systému říká, že je to „DARPA port k mapovači čísel RPC-programů“. Je to první meze-
ra v zabezpečení, se kterou se setkáte při čtení tohoto dokumentu. Popis zaplnění jedné z mezer naleznete v části 6. Tu vám opět vřele doporučuji k prostudování.

Spusťte `portmapper`. Jmenuje se buď `portmap`, nebo `rpc.portmap` a měl by bydlit v adresáři `/usr/sbin` (na některých strojích se nazývá `rpcbind`). Teď ho můžete spustit ručně, ale bude potřeba, aby se spustil při každém natažení systému, takže bude nutné vytvořit/editovat rc skripty. Vaše rc skripty jsou blíže popsány v manuálové stránce pro `init`. Většinou se nachází v `/etc/rc.d`, `/etc/init.d` nebo `/etc/rc.d/init.d`. Jestliže zde bude nějaký skript, který se nazývá nejspíše `inet`, je to ten, který budete editovat. Ale co napsat nebo provést je už záležitost, která leží mimo rozsah tohoto dokumentu. Spusťte `portmap` a spuštěním `ps aux` zkontrolujte, jestli žije. Žije? V pořádku.

3.4 Mountd a nfsd

Následující programy, které musíme spustit, jsou `mountd` a `nfsd`. Nejprve ale budeme editovat jiný soubor – `/etc/exports`. Řekněme, že chci, aby byl systém souborů `/mn/eris/local` (je na stroji **eris**) k dispozici pro stroj **apollo**. Potom do `/etc/exports` na stroji **eris** vložím následující:

```
/mn/eris/local apollon(rw)
```

Předchozí řádek dává stroji `apollo` na `/mn/eris/local` práva pro čtení a zápis. Místo `rw` by zde mohlo být `ro`, což znamená pouze čtení (jestliže zde nepoužijete nic, je to i implicitní hodnota, která se automaticky použije). Můžete zde použít ještě další volby, přičemž zabezpečovacími se budu ještě zabývat. Všechny volby jsou popsány na manuálových stránkách pro `exports`. Alespoň jednou v životě si je přečtete. Existují i lepší způsoby, než je výpis všech serverů v souboru `exports`. Jestliže používáte NIS (nebo NYS, NIS byl znám jako YP), můžete využít síťové skupiny, určit zde domény pomocí zástupných znaků a IP-podsítě brát jako hostitele. Pak si ale uvědomte, kdo by tak mohl nechtěně získat přístup k serveru.

Poznámka: Tento soubor (`exports`) nemá stejnou syntaxi jako v ostatních Unixech. Souborům `exports` z ostatních Unixů zde v dokumentu věnujeme jednu samostatnou část.

Nyní můžeme spustit `mountd` (nebo se jedná o `rpc.mountd`) a `nfsd` (může být nazván `rpc.nfsd`). Oba budou číst soubor `exports`.

Jestliže budete editovat `/etc/exports`, musíte zajistit, aby `nfsd` a `mountd` věděly, že se soubory změnily. Většinou se to zajišťuje spuštěním `exportfs`. Mnoho distribucí Linuxu program `exportfs` neobsahuje. Jestliže je tomu tak i u vás, instalujte na vašem stroji následující skript:

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo re-exported file systems
```

Uložte jej například jako `/usr/sbin/exportfs` a nezapomeňte u něj provést `chmod a+rx`. Nyní po změně vašeho souboru `exports` jako root spustíte `exportfs`.

Nyní musíte zkontrolovat, jestli `mountd` a `nfs` skutečně fungují. Nejprve použijete `rpcinfo -p`. Mělo by se objevit zhruba následující:

```
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100005 1 udp 745 mountd
```



```
100005 1 tcp 747 mountd
100003 2 udp 2049 nfs
100003 2 tcp 2049 nfs
```

Jak je vidět, portmapper oznámil svoje služby, a totéž učinily i mountd a nfsd.

Jestliže obdržíte `rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused` nebo něco podobného, je jasné, že portmapper nefunguje. Napravte to. Jestliže dostanete `No remote programs registered`, pak buď s vámi portmapper nechce mluvit, nebo se něco pokazilo. Použijte na `nfsd`, `mountd` a `portmapper` příkaz `kill` a pokuste se o všechno znovu od začátku.

Po kontrole služeb portmapperu můžete také provést kontrolu pomocí `ps`. Portmapper bude hlásit svoje služby i v případě, že programy, které je využívaly, se zhroutí. Takže pokud se něco pokazí, kontrola pomocí `ps` bude na místě.

Samozřejmě potřebujete také upravit vaše systémové `rc` soubory, aby se `mountd` a `nfsd`, stejně jako `portmapper` spustily při startu systému. Většinou už se potřebné skripty na vašem stroji nachází, stačí v nich pouze zrušit komentář na patřičných místech nebo je aktivovat pro příslušné úrovně inicializace.

Nyní byste měli znát manuálové stránky pro: `portmap`, `mount`, `nfsd` a `exports`.

Jestliže jste všechno provedli, jak jsem navrhoval, můžete spustit NFS-klienta.

4 Nastavení NFS-klienta

Nejprve budete potřebovat jádro, kde bude NFS-systém zabudován nebo bude k dispozici jako modul. To se konfiguruje před kompilací jádra. Jestliže jste jádro ještě nekompilovali, přečtěte si dokument, týkající se jádra, a pokuste se o to. Jestliže používáte nějakou velmi dobrou distribuci (jako je Red Hat) a ještě jste se jádrem nebo moduly nezabývali (takže jste nic nezkažili :-), budete mít NFS k dispozici automaticky.

Nyní můžete jako `root` zadat příslušný příkaz `mount` a systém souborů se objeví. Když navážeme na příklad z předchozí části, budeme `mount` používat pro připojení `/mn/eris/local` ze stroje `eris`. To provedete následovně:

```
mount -o rsize=1024,wsize=1024 eris:/mn/eris/local /mnt
```

K volbám pro `rsize` a `wsize` se ještě dostaneme. Systém souborů je nyní k dispozici pod `/mnt`, takže tady můžete použít příkaz `cd` a `ls` a můžete se zde podívat na jednotlivé soubory

ry. Zjistíte, že je to pomalejší než lokální systém souborů, ale naopak zase pohodlnější než FTP. Jestliže se po použití `mount` místo připojení systému objeví chybové hlášení, například: „eris:/mn/eris/local failed, reason given by server“: **Permission denied**, pak je špatný soubor `exports` nebo jste po jeho editaci zapomněli spustit `exportfs`. Jestliže se objeví „mount clntudp_create: RPC: Program not registered“, znamená to, že na serveru neběží `nfsd` nebo `mountd`.

Systému souborů se zbavíte takto:

```
umount /mnt
```

Aby se systém souborů `nfs` připojil při startu systému, musíte editovat `/etc/fstab`. Pro náš příklad stačí připojit takovýto řádek:

```
# device mountpoint fs-type options  dump fsckorder
...
eris:/mn/eris/local /mnt nfs rsize=1024,wsiz=1024 0 0
...
```

K tomu vše, pokračujeme dál.

4.1 Volby pro mount

Ke zvážení zde máte několik možností. Jejich využitím je možné ovládat způsob, jakým se NFS-klient postaví k havárii serveru nebo poruše síť. Jednou z dobrých vlastností NFS je to, že se takové situace zvládají s přehledem. Pokud je klient nastaven správně. Existují zde dva různé módy poruch:

- `soft` NFS-klient nahlásí chybu procesu, který využívá soubor na přimontovaném systému souborů NFS. Některé programy se s takovým hlášením správně vyrovnají, některé ne. Takové nastavení nemohu doporučit.
- `hard` Program, který přistupuje k souboru na přimontovaném systému souborů NFS, se při havárii serveru zastaví. Proces není možné přerušit nebo zrušit, pokud neurčíte také `intr`. Když je NFS-server opět funkční, program bude pokračovat tam, kde přestal. To je pravděpodobně i pro vás to pravé. Na všech připojených NFS-systémech souborů doporučuji použít `hard`, `intr`.

V návaznosti na předchozí příklad vypadají údaje v `fstab` takto:

```
# device mountpoint fs-type options dump fsckorder
...
eris:/mn/eris/local /mnt nfs rsize=1024,wsizer=1024,hard,intr 0 0
...
```

4.2 Optimalizace NFS

Normálně (když nejsou určeny volby `rsize` a `wsizer`) bude NFS číst a zapisovat po 4 096 nebo 8 192 bajtech. Některé kombinace linuxových jader a síťových karet tak velké bloky nezvládají a navíc nemusí být optimální. Takže budeme muset experimentovat a nalézt nastavení `rsize` a `wsizer`, která jsou funkční a co nejrychlejší. Rychlost vašeho nastavení můžete otestovat několika jednoduchými příkazy. Po použití předchozího příkazu `mount` a za předpokladu, že máte práva pro zápis na disk, můžete takto otestovat výkon při postupném zápisu:

```
time dd if=/dev/zero of=/mnt/testfile bs=16k count=4096
```

Tím vytvoříte 64 MB soubor nulových bajtů (je natolik velký, že by zde na výkon neměla mít vliv vyrovnávací paměť, jestliže máte velkou paměť použijte větší soubor). Operaci proveďte vícekrát (5-10) a spočítejte průměrný čas. To je doba trvání zápisu, která nás zajímá. Pak můžete zpětným čtením souboru otestovat výkony při čtení:

```
time dd if=/mnt/testfile of=/dev/null bs=16k
```

Opět proveďte vícekrát a spočítejte průměr. Proveďte odpojení a připojení (`mount` a `umount`), ale s většími hodnotami `rsize` a `wsizer`. Mělo by se jednat o násobky 1 024 a neměly by být větší než 16 384 bajtů, což je maximální velikost u NFS verze 2. Ihned po připojení provádějte příkaz `cd` do připojeného systému a zde provádějte `ls` a trochu prozkoumejte `fs`, aby bylo zřejmé, že je vše v pořádku. Jestliže je nastavení `rsize/wsize` příliš vysoké, příznaky jsou *velice* podivné a někdy ne zcela zřejmé. Typický je nekompletní seznam souborů po zadání „`ls`“ a dále absence chybových hlášení. Nebo někdy bez chybových hlášení selže čtení souboru. Jakmile jste si jisti, že zadané hodnoty `rsize/wsize` jsou funkční, můžete opět přistoupit k testům rychlosti. Různé platformy serverů mají různé optimální hodnoty. SunOS a Solaris je například mnohem rychlejší při použití 4 096 bajtových bloků.

Nová jádra Linuxu (od 1.3) provádějí při nastavení `rsize` větším nebo rovném velikosti stránky procesoru čtení dopředu. Na procesorech Intel je velikost stránky 4 096 bajtů. Takové čtení zde *značně* zvýší čtecí výkony NFS. Takže na strojích Intel je dobré mít `rsize` alespoň 4 096.

Nezapomeňte nalezenou hodnotu `rsize/wsize` odrazit v `/etc/fstab`.

Výkony NFS při zápisu je možné zvýšit trikem – zrušením synchronních zápisů na serveru. Specifikace NFS určuje, že žádosti NFS o zápis by neměly být považovány za vyřízené, dokud jsou zapsaná data na stálém médiu (obvyčně disk). Tím se určitým způsobem sníží výkon při zápisu, asynchronní zápisy budou NFS-zápisy urychlovat. Linux nfsd nikdy neprovádí synchronní zápisy, protože implementace systému souborů v Linuxu se k tomu nepropůjčí, ale na ne-linuxových serverech můžete výkony zvýšit vložení následujícího řádku (nebo něčeho podobného) do vašeho souboru exports:

```
/dir -async,access=linuxbox
```

Podívejte se prosím u daného stroje na manuálovou stránku pro exports. Uvědomte si také, že se zvyšuje riziko ztráty dat.

5 NFS na pomalých linkách

Pomalé linky zahrnují modemy, ISDN a také další připojení na velké vzdálenosti.

Tato část je postavena na znalostech o použitých protokolech, ale zkušenosti z experimentování chybí. Můj domácí počítač byl 6 měsíců mimo provoz (nefunkční HDD), takže jsem nemohl testovat spojení přes modem. Jestli se o to pokusíte, dejte mi prosím vědět.

První věc, kterou musíme brát v úvahu, je to, že NFS je pomalý protokol. Má velké nároky. Použití NFS je skoro jako použití kermitu na přenos souborů. Je to *pomalé*. Skoro všechno je rychlejší než NFS. FTP je rychlejší. HTTP je rychlejší. Rcp je rychlejší. Ssh je rychlejší.

Pořád to chcete zkoušet? Dobrá.

Implicitní hodnoty NFS jsou určeny pro rychlé linky s malými prodlevami. Jestliže tyto implicitní hodnoty použijete na linkách s velkými prodlevami, NFS může hlásit chyby, ukončovat operace, předpokládat, že soubory jsou kratší než ve skutečnosti, nebo provádět jiné záhadné operace.

První věc, kterou musíte udělat, je *nepoužívat* pro příkaz mount volbu `soft`. Pak by prodlevy způsobily hlášení chyb aplikacím, které by většinou situaci správně nezvládly. Takto snadno vyvoláte různá záhadná selhání. Použijte tedy raději volbu `hard`. Když je aktivní **hard**, prodlevy místo ukončení způsobí další pokusy o pokračování, ať už se aplikace snaží o cokoliv. To je to, co potřebujete. Opravdu.

Nyní si musíte u příkazu mount pohrát s volbami `timeo` a `retrans`. Jsou popsány na manuálové stránce `nfs(5)`, ale tady vám nabízím její kopii:

`timeo=n` The value in tenths of a second before sending the first retransmission after an RPC timeout. The default value is 7 tenths of a second. After the first timeout, the timeout is doubled after each successive timeout until a maximum timeout of 60 seconds is reached or the enough retransmissions have occurred to cause a major timeout. Then, if the filesystem is hard mounted, each new timeout cascade restarts at twice the initial value of the previous cascade, again doubling at each retransmission. The maximum timeout is always 60 seconds. Better overall performance may be achieved by increasing the timeout when mounting on a busy network, to a slow server, or through several routers or gateways.

`retrans=n` The number of minor timeouts and retransmissions that must occur before a major timeout occurs. The default is 3 timeouts. When a major timeout occurs, the file operation is either aborted or a "server not responding" message is printed on the console.

Jinými slovy: Jestliže ve lhůtě 0,7 sekundy (700 ms) neobdržíte odpověď, NFS-klient bude žádost opakovat a lhůtu zdvojnásobí na 1,4 sekundy. Jestliže se odpověď opět neobjeví, žádost se opět opakuje a lhůta se opět zdvojnásobí (na 2,8 sekundy).

Rychlost linek je možné měřit pomocí `ping` se stejnou velikostí paketu, jakou mají nastaveny volby `rsize/wsize`.

```
$ ping -s 8192 lugulbanda
PING lugulbanda.uio.no (129.240.222.99): 8192 data bytes
8200 bytes from 129.240.222.99: icmp_seq=0 ttl=64 time=15.2 ms
8200 bytes from 129.240.222.99: icmp_seq=1 ttl=64 time=15.9 ms
8200 bytes from 129.240.222.99: icmp_seq=2 ttl=64 time=14.9 ms
```

```
8200 bytes from 129.240.222.99: icmp_seq=3 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=4 ttl=64 time=15.0 ms
```

```
--- lugulbanda.uio.no ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14.9/15.1/15.9 ms
```

Čas (time) zde znamená, jak dlouho se paket ping dostává do lugulbandy a zpět. 15 ms je dost rychlých. Přes linku s kapacitou 28 000 bps můžete očekávat tak 4 000 – 5 000 ms a jestli je linka přetížená, může se čas prodloužit, snadno i zdvojnásobit. Když je čas dost vysoký, říkáme, že je velká prodleva (high latency). Obecně se při zvětšování paketů a vyšším zatížení sítě zvyšuje i prodleva. Timeo zvýšte podle vaší linky a jejího zatížení. A protože se prodleva zvyšuje, když linku využíváte pro další věci: Když budete chtít použít FTP a NFS současně, měli byste se pokusit změřit časy pro ping při využití FTP pro přenos souborů.

6 Zabezpečení a NFS

Nejsem žádný odborník na zabezpečení počítačů. Ale pro otázky zabezpečení mám pár *malých* rad. Ale nezapomeňte, že tohle není kompletní seznam problémů kolem NFS a jestli si myslíte, že po přečtení a provedení tohoto všeho jste v bezpečí, určitě se mýlíte.

Tahle část se vás nemusí týkat – a sice v případě, že je vaše síť *uzavřená*, vy věříte všem uživatelům a nikdo jiný (komu nevěříte) se na síť nemůže dostat. Znamená to, že se na síť nikdo jiný nemůže přihlásit ani přes modem a ani přes jinou síť, ve které nevěříte všem. Znít to paradoxně? Ne tak zcela. Je to jen *základní* rada pro zabezpečení. A uvědomte si, že to, co tady říkám, je jen začátek všeho. *Bezpečná* síť vyžaduje pilného a dobrého správce, který ví, kde najít informace o aktuálních a možných problémech.

NFS má základní problém v tom, že klient (nebylo-li řečeno jinak) bude věřit NFS-serveru a obráceně. To může být špatné. Znamená to, že pokud je narušen rootovský účet na serveru, není už takový problém narušit i rootovské účty na klientech. A opačně. K tomu existuje množství kopírovacích strategií, ke kterým se brzy dostaneme.

Něco, co byste si měli přečíst, jsou vývěsky CERT na NFS, většina z níže uvedeného textu se zabývá záležitostmi, o kterých CERT píše ve vývěskách. Zde následují některé vývěsky, vztahující se k NFS:

```
CA-91:21.SunOS.NFS.Jumbo.and.fsirand 12/06/91
Vulnerabilities concerning Sun Microsystems, Inc. (Sun) Network
File System (NFS) and the fsirand program. These vulnerabilities
```

affect SunOS versions 4.1.1, 4.1, and 4.0.3 on all architectures. Patches are available for SunOS 4.1.1. An initial patch for SunOS 4.1 NFS is also available. Sun will be providing complete patches for SunOS 4.1 and SunOS 4.0.3 at a later date.

CA-94:15.NFS.Vulnerabilities 12/19/94

This advisory describes security measures to guard against several vulnerabilities in the Network File System (NFS). The advisory was prompted by an increase in root compromises by intruders using tools to exploit the vulnerabilities.

CA-96.08.pcnfsd 04/18/96

This advisory describes a vulnerability in the pcnfsd program (also known as rpc.pcnfsd). A patch is included.

6.1 Zabezpečení klienta

U klienta se můžeme rozhodnout, že serveru nebudeme tolik věřit, několika způsoby a možnostmi pro mount. Můžeme například pomocí volby `nosuid` zakázat práci programů `suid` mimo NFS-systém. Tohle je dobrý nápad a stojí za to zvážit, zda jej nevyužít na všech připojených (mount) discích. Znamená to, že `root` na serveru nemůže v systému souborů vytvořit rootovský `suid`-program, přihlásit se na klienta jako normální uživatel a využít rootovský `suid`-program a stát se tak `rootem` i na klientu. Pomocí volby `noexec` je také možné na připojeném systému souborů zakázat spouštění souborů. Tohle je ale oproti `nosuid` méně praktické, protože systém souborů může obsahovat *nějaké* skripty nebo programy, které musí být spouštěny. Tyto volby zadáte ve sloupci pro volby `s rsize` a `wsize`, oddělené čárkami.

6.2 Zabezpečení serveru: `nfsd`

Na serveru se můžeme rozhodnout, že nebudeme věřit rootovskému účtu klienta. Dosáhneme toho využitím volby `root_squash` v `exports`:

```
/mn/eris/local apollon(rw,root_squash)
```

Nyní pokud se uživatel s `UID 0` na klientovi pokusí přistoupit (číst, zapisovat, mazat) na systém souborů, server jeho `UID` nahradí `UID` účtu „`nobody`“ ze serveru. To znamená, že `root` z klienta nemůže pracovat se soubory, se kterými může pracovat jedině `root` na serveru. To je dobré, takže `root_squash` pravděpodobně použijete na všech exportovaných systémech souborů. Můžete ale namítnout, že `root` z klienta použije „`su`“, stane se tak jiným uživatelem

a stejně tyto uživatelské soubory použije! Zde je odpověď jasná: Ano, je to tak a v Unixu na NFS je to tak správně. Z toho plyne jedno ponaučení – všechny důležité soubory a binární programy by měl vlastnit **root**, ne tedy **bin** nebo jiný nerootovský účet, protože jediným účtem, na který se root z klienta nedostane, je rootovský účet na serveru. Na manuálových stránkách pro NFSd je vypsáno několik dalších voleb pro squash, takže se můžete rozhodnout, že někomu z klientů budete méně věřit. Pomocí squash je možné nastavit i libovolný rozsah UID a GID. Popis naleznete na manuálové stránce pro Linux NFSd.

V Linux NFSd je `root_squash` vlastně nastaveno implicitně. Pokud má mít root k systému souborů přístup, použijte

```
no_root_squash
```

Další důležitou věcí je zajistit, aby `nfsd` kontroloval, že všechny jeho žádosti přichází z privilegovaného portu. Jestliže přijme žádosti z jakéhokoliv starého portu na klientu, uživatel může bez zvláštních práv spustit program, který je možné snadno získat na Internetu. Hovoří protokolem nfs a určí, že uživatel je kdokoliv, kým chce být. No, trochu tady straším. Linux `nfsd` provádí kontrolu automaticky, v jiných operačních systémech ji musíte nastavit sami. Popis nastavení naleznete v manuálových stránkách pro `nfsd` v daném operačním systému.

Další věc. Nikdy neexportujte systém souborů na „localhost“ nebo na 127.0.0.1. Důvěřujte mi.

6.3 Zabezpečení serveru: portmapper

Základní `portmapper` má v kombinaci s `nfsd` problém, který umožňuje dostat se na soubory na NFS-serverech bez jakýchkoliv práv. Naštěstí je `portmapper`, využívaný v Linuxu, proti tomuto zneužití relativně bezpečný a může být ještě bezpečnější, pokud ve dvou souborech nakonfigurujete seznamy pro přístup.

Nejprve editujme `/etc/hosts.deny`. Měl by obsahovat řádek

```
portmap: ALL
```

Tento řádek zakáže přístup *všem*. Je to možná trochu drastické, takže přístup opět otevřeme editací `/etc/hosts.allow`. Nejprve si ale musíme rozmyslet, komu přístup otevřeme. Mělo by se jednat o všechny stroje, které by měly mít přístup k vašemu `portmapperu`. V běžícím linuxovém systému je jen málo strojů, které by z nějakého důvodu potřebovaly přístup. `Portmapper` se stará o `nfsd`, `mountd`, `ypbind/ypserv`, `pcnfsd` a „r“ služby, jako `ruptime` a `rusers`. Zde stojí za zvážení jedině `nfsd`, `mountd`, `ypbind/ypserv` a snad `pcnfsd`. Všechny stroje, které potřebují využívat služby na vašem stroji, by to měly mít také umožněno. Řekněme například, že adresa vašeho stroje je 129.240.223.254 a stroje, nalézající se v podsíti 129.240.223.0, by k němu měly mít přístup (použité termíny jsou vysvětleny v dokumentu k síťi). Potom v `hosts.allow` napíšete


```
portmap: 129.240.223.0/255.255.255.0
```

To je stejný případ jako síťová adresa, daná pro směrování, a podsíťová maska, použitá v `ifconfig`. Pro zařízení `eth0` by měl `ifconfig` na tomto počítači ukazovat

```
...
eth0 Link encap:10Mbps Ethernet HWaddr 00:60:8C:96:D5:56
inet addr:129.240.223.254 Bcast:129.240.223.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:360315 errors:0 dropped:0 overruns:0
TX packets:179274 errors:0 dropped:0 overruns:0
Interrupt:10 Base address:0x320
...
```

A `netstat -rn` by měl ukazovat

```
Kernel routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
...
129.240.223.0 0.0.0.0 255.255.255.0 U 0 0 174412 eth0
...
```

Síťová adresa je v prvním sloupci.

Soubory `hosts.deny` a `hosts.allow` jsou popsány na stejnojmenných manuálových stránkách.

DŮLEŽITÉ: V těchto souborech *nedávejte* na řádky `portmap` *nic jiného než IP-ČÍSLA*. Vyhledávání podle názvů lokací může nepřímo způsobit aktivitu `portmap`, která způsobí vyhledávání podle názvů lokací, které může nepřímo způsobit aktivitu `portmap`, která způsobí...

Předchozí věci posilují váš server. Jediným problémem, který zbývá (opravdu!), je něčí narušení roota (nebo natažení systému MS-DOS) na stroji, kterému věříme. Se získanými právy může dojít k odeslání žádostí ze zabezpečeného portu pod libovolným uživatelským jménem.

6.4 NFS a firewall

Dobrý nápad je použít na NFS firewall a portmapovat porty na vašem routeru a firewallu. `Nfsd` pracuje na portu 2049 s oběma protokoly `udp` i `tcp`. `Portmapper` je na portu 111 (`tcp` i `udp`) a `mountd` je na portu 745 a 747 (`tcp` a `udp`). Obvykle. Porty raději zkontrolujte příkazem `rpcinfo -p`.

Jestliže ale na druhou stranu chcete, aby NFS procházel přes firewall, existují volby pro novější NFSd a mountd, aby používaly určitý (nestandardní) port, který může být ve firewallu otevřen.

6.5 Shrnutí

Jestliže u aplikací `portmapper/nfs` používáte `hosts.allow/deny`, `root_squash`, `nosuid` a funkce privilegovaných portů, vyhnete se mnoha známým nedostatkům v NFS a můžete se cítit téměř bezpečně.

Ale kdyby přece jenom: Když má na vaši síť přístup nějaký vetřelec a jsou-li `/home` a `/var/spool/mail` připojeny přes NFS, může vložit do vašich souborů `.forward` nebo do mailboxu nějaké podivné příkazy. Z tohoto důvodu nikdy nepřistupujte k vašemu soukromému klíči PGP přes NFS. Nebo alespoň počítejte s rizikem. A nyní už něco znáte.

NFS a `portmapper` tvoří dohromady složitý podsystém, takže se někdy může stát, že jsou objeveny nové chyby – ať už v základním návrhu nebo v použité implementaci. Vyskytují se i známé chyby, které někdo zneužívá. Ale takový je život. Tyto záležitosti můžete sledovat alespoň v newsgroups `comp.os.linux.announce` a `comp.security.announce`.

7 Seznam problémů s programem mount

Tato část je odvozena podle seznamu, vytvořeného v IBM Corporation, který se zabývá problémy při připojení NFS. Zde jim musím poděkovat za zpřístupnění pro účely tohoto dokumentu. Jestliže se při připojení NFS setkáte s problémem, obraťte nejprve následující seznam. Teprve v případě neúspěchu svůj problém někam pošlete. Každá položka popisuje nějaký problém a nabízí jeho odstranění.

1. Systém souborů není exportován nebo není exportován danému klientovi.

Náprava: exportujte jej.

2. Rozlišení názvu neodpovídá exportnímu seznamu.

V exportním seznamu je například `export` na **johmad**, ale tento název je rozlišen jako **johmad.austin.ibm.com** a připojení není povoleno.

Náprava: exportujte na obě formy názvu.

Situace může nastat také v případě, že klient má 2 rozhraní s různými názvy pro oba adaptéry a v exportu je určen jen jeden.

Náprava: exportujte obě rozhraní.

Situace může nastat také v případě, že server nemůže na klientu provést `lookuphostbyname` nebo `lookuphostbyaddr` (jsou to funkce knihoven). Zajistěte, aby klient mohl provést `host <name>; host <ip_addr>;` a aby oba příkazy ukazovaly stejný stroj.

Náprava: srovnajte rozlišení názvu.

3. Systém souborů byl připojen po spuštění NFS (na tom serveru). V takovém případě server exportuje vyznačené místo připojení, ne připojený systém souborů.

Náprava: Ukončete `NFSd` a znovu jej spusťte.

Poznámka: Klienti, na kterých je připojeno vyznačené místo připojení, budou mít po restartu problémy s přístupem.

4. Datum je na jednom nebo obou strojích úplně mimo (mohou vzniknout zmatky).

Náprava: nastavte datum správně.

Autor projektu HOWTO doporučuje k synchronizaci hodin použití NTP. Protože na NTP existují v USA exportní omezení, pro Debian, RedHat nebo Slackware jej získáte na adrese `ftp://ftp.hacktic.nl/pub/replay/pub/linux` nebo na `mirrorech`.

5. Server nedokáže přijmout připojení od uživatele, který je členem více než 8 skupin.

Náprava: snižte počet skupin, ve kterých je uživatel členem, nebo proveďte připojení přes jiného uživatele.

8 FAQ

Tato část se věnuje FAQ, tedy často pokládaným otázkám. Většinu sepsal Alan Cox.

1. **Jestliže používám Linux jako NFS-server, často se mi objevují chyby „stale nfs handle“.**

Objevuje se jako chyba v některých starších verzích `nfsd`. Od verze `nfs-server2.2beta16` je chyba odstraněna.

2. **Když se pokusím připojit systém souborů, dostávám**

```
can't register with portmap: system error on send
```

Pravděpodobně používáte systém Caldera. Je zde chyba v `rc` skriptech. Nápravu si zjednejte od Caldery.

3. Proč není možné spustit soubor po jeho okopírování na NFS-server?

Důvod je v tom, že `nfsd` z důvodu zvýšení výkonů používá při práci se soubory vyrovnávací paměť (uvědomte si, že běží v uživatelském prostoru). `Nfsd` sice soubor otevře (jako v případě po zápisu do něj), ale jádro vám jej neumožní spustit. `Nfsd`, která jsou novější než z jara 95, otevírají soubory v několika sekundách, těm starším to může trvat i dny.

4. Všechny moje NFS-soubory jsou pouze pro čtení.

Linuxový NFS-server je implicitně pouze pro čtení. Více podrobností viz manuálové stránky pro „`exports`“ a `nfsd`. Budete muset upravit `/etc/exports`.

5. Připojuji se z linuxového NFS-serveru a `ls` mi funguje, ale číst a zapisovat soubory nemůžu.

Ve starších verzích Linuxu musíte NFS-server připojit s hodnotami `rsize=1024, wsize=1024`.

6. Připojuji se z linuxového NFS-serveru s velikostí bloku mezi 3 500 – 4 000 a pravidelně se zhroutí linuxový box.

Tak to nedělejte.

7. Dokáže Linux zvládnout NFS přes TCP?

Ne, zatím ne.

8. Když se pokouším připojit stroj z linuxového boxu, dostanu spoustu podivných chyb.

Uživatelé musí být v 8 a méně skupinách. Starší servery to vyžadují.

9. Když provedu restart svého stroje, někdy se při pokusu o odpojení NFS-serveru zasekne.

Při restartu **neprovádějte** odpojení NFS-serverů, ignorujte je, protože se stejně nic nestane. Příkaz pak zní `umount -avt nonfs`.

10. NFS-klienty Linuxu jsou při zápisu do systémů Sun a BSD velice pomalé.

NFS-zápisy jsou normálně synchronní (můžete to vypnout, pokud nechcete riskovat ztrátu dat). Horší je, že od BSD odvozená jádra zatím nebývají schopna pracovat v malých blocích. Proto když zapisujete 4 KB dat z linuxového boxu v 1 KB paketech, které používá, BSD provede tohle

čtení 4KB stránky

změna 1KB

zápis 4KB zpět na disk

čtení 4KB stránky

změna 1KB
zápis 4KB zpět na disk
atd..

9 Exportování systémů souborů

Způsob exportování systémů souborů pomocí NFS není pochopitelně u všech platforem konzistentní. Linux a Solaris 2 jsou v tomto případě rozdílné. Tato část vypisuje, jak se to dělá na většině systémů. Jestliže zde není váš systém obsažen, musíte si k němu projít manuálové stránky. Klíčová slova jsou `nfsd`, `system administration tool`, `rc scripts`, `boot scripts`, `boot sequence`, `/etc/exports`, `exportfs`. V této části budu průběžně využívat jeden příklad: jak exportovat `/mn/eris/local` na `apollo` `read/write`.

9.1 IRIX, HP-UX, Digital-UNIX, Ultrix, SunOS 4 (Solaris 1), AIX

Tyto operační systémy využívají tradiční Sunovský exportní formát. Do `/etc/exports` zapište:

```
/mn/eris/local -rw=apollo
```

Kompletní dokumentace je na manuálové stránce `exports`. Po editaci souboru spusťte `exports -av` a systém souborů bude exportován.

Syntaxe příkazu `exportfs` se systém od systému může lišit. Na některých operačních systémech bude předchozí zadaný řádek znít:

```
/mn/eris/local apollo
```

nebo dokonce ještě takto:

```
/mn/eris/local rw=apollo
```

Doporučuji formálnost. Riskujete, že příští verze `exportfs` bude v syntaxi přísnější a nic vám nepoběží.

9.2 Solaris 2

Sun při tvorbě systému Solaris 2 znovu vynalezl kolo. Takže tento systém se od ostatních značně liší. Budete provádět editaci souboru `/etc/dfs/dfstab`. Zde uložíte podle manuálové stránky `share(1M)` příkazy pro sdílení. Asi takto:

```
share -o rw=apollon -d "Eris Local" /mn/eris/local
```

Po editaci spusíte program `shareall`, který systémy souborů exportuje.

10 PC-NFS

PC-NFS nespouštějte. Pusťte si sambu.

Pardon: O PC-NFS nevím nic. Jestli se na to někdo cítí, ať to udělá a já jeho práci zařadím na toto místo.