# DNS HOWTO

Nicolai Langfeldt janl@math.uio.no v2.2, 11 February 1999

Kako za čisto kratek čas postati DNS administrator. V slovenščino prevedel Andrej Lajovic andrej.lajovic@guest.arnes.si, 7. avgusta 2000

## Kazalo

# 1 Predgovor

Ključne besede: DNS, bind, bind-4, bind-8, named, povezava na klic, ppp, slip, isdn, Internet, domena, ime, poizvedovanje, predpomnenje.

Ta dokument je del projekta Linux Documentation Project.

## 1.1 Pravne zadeve



V slovenščino prevedel Andrej Lajovic `andrej.lajovic@guest.arnes.si`, 7. avgusta 2000. Za razširjanje slovenskega prevoda veljajo enaki pogoji, kot za original.

## 1.2 Zahvala in prošnja za pomoč.

Rad bi se zahvalil Arntu Gulbrandsenu, ki se je pregrizel skozi vse delovne različice tega dokumenta in ki mi je dal mnogo uporabnih predlogov. Rad bi se zahvalil tudi mnogim ljudem, ki so mi poslali predloge in opozorila po elektronski pošti.

Tole ne bo nikoli zaključen dokument, zato mi sporočite o vaših težavah in uspehih - tako lahko naredimo ta HOWTO boljši. Komentarje in/ali vprašanja ter denar pošljite na janl@math.uio.no. Preden mi pošljete sporočilo, na katerega želite tudi odgovor, prosim preverite, da je vaš naslov, naveden v pismu *zagotovo* pravilen in deluje. Preden me kontaktirate, preberite tudi poglavje 8 (Vprašanja in odgovori). Še nekaj: razumem le norveški in angleški jezik.

Če ste prevedli ta HOWTO, vas prosim, da me na to opozorite, da bom lahko vzdrževal seznam jezikov, v katerih je bil objavljen in da vas bom ob izidu nove različice lahko o tem obvestil.

## 1.3 Posvetilo

Ta HOWTO je posvečen Anne Line Norheim Langfeldt, čeprav ga verjetno ne bo nikoli prebrala - pač ni takšne vrste dekle.

# 2 Uvod.

**Kaj to je in kaj ni.**

DNS (Domain Name System) pretvarja imena računalnikov v IP številke, ki jih imajo vsi računalniki v internetu. Preslikava imena v naslove, naslove v imena in še nekaj drugih stvari. Ta HOWTO prikazuje, kako definirati takšne preslikave na Linux sistemu. Preslikava je nekakšna povezava med imenom računalnika, na primer `ftp.linux.org` ter njegovo IP številko (ali naslovom) `199.249.150.4`.

DNS je za nepoučene ljudi (vas ;-) eno od bolj nejasnih področij omrežne administracije. Ta HOWTO bo poskusil narediti nekaj stvari jasnejših. Opisuje, kako postaviti *preprost* DNS imenski strežnik. Začeli bomo z imenskim strežnikom s predpomnilnikom in prešli na postavitev primarnega DNS strežnika za domeno. Za zapletenejše nastavitve lahko pogledate v poglavje 8 (Vprašanja in odgovori). Če ni opisana tam, boste morali *prebrati* nekaj Prave Dokumentacije. Kaj sestavlja Pravo Dokumentacijo, bom povedal v 9 (zadnjem poglavju).

Preden začnete, morate svoj računalnik nastaviti tako, da se lahko telnetate vanj in iz njega ter uspešno delate različne povezave v omrežje. Še posebej pomembno je, da lahko storite `telnet 127.0.0.1` in dobite svoj lasten računalnik (preizkusite takoj!). Za začetek potrebujete tudi dobre datoteke `/etc/nsswitch.conf` (ali `/etc/host.conf`), `/etc/resolv.conf` in `/etc/hosts`, ker njihove funkcije tu ne bom razlagal. Če vam vse to še ne deluje, poglejte v NET-3-HOWTO in/ali PPP-HOWTO, kjer so navodila za nastavitev.

Ko rečem 'vaš računalnik', mislim s tem računalnik, na katerem postavljate DNS in ne kakšen drug računalnik, ki ga posedujete.

Predpostavljam, da niste za kakršnim koli požarnim zidom, ki bi preprečeval imenske poizvedbe. Če ste, boste potrebovali posebno konfiguracijo, zato poglejte v poglavje 8 (Vprašanja in odgovori).

Strežnik za imenske poizvedbe je na Unix-u program `named`. Je del paketa "bind", katerega koordinator za Konzorcij internetnega programja je Paul Vixie. `Named` je vključen v večino distribucij Linuxa in je navadno nameščen kot `/usr/sbin/named`. Vkolikor named imate, ga verjetno lahko uporabite, sicer pa lahko preveden program dobite z Linuxove ftp strani ali pa si pretočite zadnjo različico vsemogočne izvorne kode z *ftp.isc.org:/isc/bind/src/cur/bind-8/*. Ta HOWTO govori o bind različici 8. Starejša verzija tega HOWTO-ja, ki velja za bind 4, je še vedno dosegljiva na *http://www.math.uio.no/˜janl/DNS/* v primeru, da še vedno uporabljate bind 4. Če vaš named priročnik (man) govori (na koncu, v razdelku FILES) o `named.conf`, imate bind 8, če govori o `named.boot`, pa bind 4. Če imate različico 4 in vam je varnost pomembna, bi morali vsekakor nadgraditi na različico 8.

DNS je čez cel internet segajoča baza. Pazite, kaj vpišete vanjo. Če boste vanjo vpisali neumnosti, boste vi in drugi ljudje neumnosti iz nje tudi dobili. Vzdržujte svoj DNS organiziran in pregleden, da vam bo dobro služil. Naučite se ga uporabljati, administrirati, odpravljati napake in bodite še eden dober administrator, ki skrbi, da internet ne pade na kolena zaradi nepravilne uporabe.

V tem dokumentu zaradi poenostavljenosti navajam nekaj dejstev, ki niso popolnoma resnična (čeprav so vsaj napol resnična). Stvari bodo (verjetno ;-) delovale, če verjamete, kar vam bom povedal.

**Nasvet:** Naredite si rezervne kopije vseh datotek, ki jih boste spreminjali - v primeru, da kaj ne bo delovalo, lahko sistem še vedno vrnete v staro, delujoče stanje.

# 3 Imenski strežnik s predpomnilnikom.

### Prvi korak v konfiguraciji DNS-a, zelo uporaben za uporabnike klicnega dostopa

Imenski strežnik s predpomnilnikom bo našel odgovore na imenske poizvedbe in si jih zapomnil za naslednjič, ko jih boste potrebovali. To bo precej skrajšalo čakalni čas, še posebej, če imate počasno povezavo.

Za začetek potrebujete datoteko `/etc/named.conf`, ki jo named prebere, ko se zažene. Za zdaj naj vsebuje le:

```
// Konfiguracijska datoteka za imenski strežnik s predpomnilnikom

options {
        directory "/var/named";

        // Odkomentiranje tega lahko pomaga, če ste za požarnim zidom in
        // stvari ne delujejo, kot bi morale:

        // query-source port 53;
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};
```

Vrstica '`directory`' pove named, kje naj pogleda za datoteke. Vse datoteke, navedene zatem, se nahajajo relativno na to mesto - mapa `pz` se nahaja v `/var/named` in je torej v resnici `/var/named/pz`. `/var/named` je prava mapa glede na *Standard Linux-ovega datotečnega sistema*.

V `named.conf` je omenjena tudi datoteka `/var/named/root.hints`. Ta naj vsebuje:

```
;
; Če to datoteko že imate, se lahko tu nahajajo komentarji.
; Če se ne, ne skrbite.
;
.                       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
```

Datoteka opisuje korenske imenske strežnike po svetu. Ti podatki se sčasoma spreminjajo, zato jih morate ohranjati točne. Preberite si poglavje 6 (Vzdrževanje) za navodila, kako to storiti.

Naslednji razdelek v `named.conf` je zadnji vnos `zone`. Njegov namen in uporabo bom razložil v naslednjem poglavju, za zdaj samo ustvarite datoteko `127.0.0` v podmapi `pz`:

```
@               IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                                1       ; Serijska številka
                                8H      ; Osveževanje
                                2H      ; Ponoven poskus
                                1W      ; Potek
                                1D)     ; Minimum TTL
                        NS      ns.linux.izmislek.
1                       PTR     localhost.
```

Vaš `/etc/resolv.conf` mora izgledati približno takole:

```
search poddomena.vasa-domena.edu vasa-domena.edu
nameserver 127.0.0.1
```

Vrstica ‘`search`’ pove, katere domene je potrebno preiskati za vsako ime računalnika, na katerega se želite priključiti. ‘`nameserver`’ vsebuje naslov vašega imenskega strežnika, v tem primeru vašega lastnega računalnika, na katerem bo named tekel (127.0.0.1 je v redu, tudi če ima vaš računalnik še kakšen naslov). Če želite navesti več imenskih strežnikov, vpišite za vsakega po eno ‘`nameserver`’ vrstico. (Zapomnite si: named te datoteke ne bo nikoli prebral, potreboval jo bo le del sistema, ki bo uporabljal named.)

Za ilustracijo, kaj ta datoteka počne: če poskuša program poizvedeti za `foo`, bo najprej poskusil `foo.poddomena.vasa-domena.edu`, nato `foo.vasa-domena.edu` in šele na koncu `foo`. Če program poskuša poizvedeti za `sunsite.unc.edu`, bo najprej poskusil `sunsite.unc.edu.poddomena.vasa-domena.edu` (priznam, da se zdi butasto, ampak tako stvar pač deluje), nato `sunsite.unc.edu.vasa-domena.edu` in končno `sunsite.unc.edu`. Pametno je, da v to vrstico ne vnesete preveč domen, ker iskanje po vseh zahteva kar nekaj časa.

Primer predpostavlja, da spadate v domeno `poddomena.vasa-domena.edu` in je vaš računalnik potemtakem `vas-racunalnik.poddomena.vasa-domena.edu`. Vrstica naj ne vsebuje vaše vrhnje domene (ang.: TLD, Top Level Domain), v tem primeru ‘`edu`’. Če se pogosto priključujete na računalnike v drugi domeni, lahko dodate tudi to:

```
search subdomain.your-domain.edu your-domain.edu druga-domena.com
```

in tako naprej. Vsekakor morate tukaj navedene primere zamenjati s pravimi domenami. Prosim, zapomnite si, da na koncu domen ni pik. To je zelo pomembno, prosim, zapomnite si, da na koncu domen ni pik.

Sedaj je treba glede na različico vaše C knjižnice (libc) popraviti `/etc/nsswitch.conf` ali `/etc/host.conf`. Če že imate `/etc/nsswitch.conf`, bomo popravili to, sicer pa `/etc/host.conf`.

**/etc/nsswitch.conf**

To je dolga datoteka, v kateri je navedeno, kje dobiti različne tipe podatkov - iz datoteke ali baze podatkov. Na začetku je navadno nekaj komentarjev, ki vam bodo pomagali pri razumevanju. Nato poiščite vrstico, ki se začne z '`hosts:`' - popraviti jo morate v:

```
hosts:      files dns
```

Če te vrstice še nimate, jo dodajte. Ta vrstica pove programom, naj najprej pogledajo v datoteko `/etc/hosts` in šele nato preverijo DNS glede na `/etc/resolv.conf`.

**/etc/host.conf**

Verjetno že vsebuje nekaj vrstic, med katerimi se ena začne z `order` in mora izgledati takole:

```
order hosts,bind
```

Če takšne vrstice še ni, jo dodajte. Ta vrstica pove podprogramom, ki imajo opraviti z imensko poizvedbo, naj najprej pogledajo v datoteko `/etc/host.conf`, nato pa povprašajo imenski strežnik (za katerega v `/etc/resolv.conf` piše, da je na 127.0.0.1).

## 3.1 Zagon named

Po vsem tem je čas, da zaženemo named. Če uporabljate povezavo na klic, jo vzpostavite. Napišite '`ndc start`' in pritisnite enter. Če to ne deluje, poskusite '`/usr/sbin/ndc start`'. Če vam tudi tu spodleti, si poglejte poglavje 8 (Vprašanja in odgovori). V datoteki, kamor vaš syslog piše sporočila (navadno `/var/adm/messages`, lahko pa tudi `/var/log/messages`), se mora med zagonom named (naredite `tail -f /var/log/messages`) izpisati nekaj takšnega:

(vrstice, ki se končajo z \ se nadaljujejo v naslednji vrstici)

```
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 (ippp0)
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.
```

Če vidite kakršna koli sporočila o napakah, ste se verjetno nekje zmotili. Named vam bo povedal, kje je napaka (upam, da v eni izmed datotek named.conf ali root.hints :-) Ubijte named in še enkrat preglejte datoteko.

Sedaj lahko preverimo vaše nastavitve. Poženite naslookup in si oglejte svoje delo.

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

>
```

Če dobite to, vse deluje, kot je treba. Upajmo. Če ne, potem pojdite nazaj in še enkrat preglejte vse. Vsakič, ko spremenite datoteko `named.conf`, morate ponovno zagnati named, kar lahko storite z ukazom `ndc restart`.

Sedaj lahko vnesete poizvedbo. Poskusite poizvedeti o kakšnem računalniku, ki je blizu vas. Blizu mene je na primer `pat.uio.no` - univerza v Oslu.

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Name:    pat.uio.no
Address:  129.240.130.16
```

nslookup je sedaj povprašal named za poizvedbo o `pat.uio.no`. Ta je nato kontaktiral enega od imenskih strežnikov, navedenih v datoteki `root.hints` in iskal od tam naprej. Poizvedba utegne trajati nekaj časa, ker je potrebno preiskati vse domene v `/etc/resolv.conf`.

Če še enkrat vprašate za isto poizvedbo, dobite tole:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    pat.uio.no
Address:  129.240.2.50
```

Bodite pozorni na vrstico "`Non-authoritative answer:`", ki smo jo dobili tokrat. Pomeni, da named ni šel še enkrat iskati iste poizvedbe po internetu, temveč jo ima sedaj v predpomnilniku. Lahko pa se zgodi, da je informacija v tem času postala napačna. Named vas na to (izredno majhno) možnost opozori tako, da reče "`Non-authoritative answer:`". Ko `nslookup` pove to drugič, je to zanesljivo znamenje, da named shranjuje poizvedbe v predpomnilnik. Iz programa `nslookup` pridete tako, da napišete '`exit`'.

## 3.2 Naredimo nekaj še boljšega

V velikih, dobro organiziranih akademskih omrežjih ali omrežjih ponudnikov interneta boste včasih ugotovili, da ima omrežje nastavljeno posredovalno hierarhijo DNS strežnikov, kar pomaga razbremeniti tako samo omrežje, kot tudi zunanje strežnike. Pogosto je težko ugotoviti, ali ste v takšnem omrežju ali ne. To pravzaprav sploh ni pomembno, če si nastavite DNS strežnik vašega ponudnika interneta kot "posredovalnik" in tako naredite poizvedbe hitrejše ter zmanjšate obremenitev vašega omrežja. Če uporabljate dostop preko modema, je to lahko kar velik dosežek. Za namene tega primera bomo predpostavili, da ima vaš ponudnik interneta dva imenska strežnika, ki jih želite uporabljati, njuni IP številki pa sta `10.0.0.1` in `10.1.0.1`. V vaši datoteki `named.conf` torej vstavite v razdelek "options" sledeče vrstice:

```
          forward first;
          forwarders {
               10.0.0.1;
               10.1.0.1;
           };
```

Ponovno zaženite vaš imenski strežnik in ga preverite z nslookup. Stvar mora delovati.

### 3.3 Čestitam

Sedaj znate postaviti named s predpomnilnikom. Vzemite si pivo, mleko ali kar pač imate najraje in proslavite to.

## 4 *Preprosta* domena.

**Kako postaviti svojo lastno domeno.**

### 4.1 Najprej nekaj teorije

Preden *resnično* začnemo to poglavje, vam bom serviral nekaj teorije in primerov o tem, kako DNS deluje. In vi boste to prebrali, ker je dobro za vas. Če nočete prebrati, bi bilo dobro, da vsaj na hitro preletite. Leteti prenehajte, ko pridete do tistega, kar mora iti v vašo datoteko `named.conf`.

DNS je hierarhičen, drevesno strukturiran sistem. Vrh se imenuje '`.`' in izgovarja 'koren' (ang.: root). Pod `.` je veliko število vrhnjih domen, od katerih so najbolj poznane `ORG`, `COM`, `EDU` in `NET`, seveda pa jih je še mnogo več. Struktura ja prav takšna kot drevo - ima korenine in iz njih poganja veje. Če imate vsaj nekaj računalniškega znanja, boste v DNS prepoznali iskalno drevo, lahko boste našli razvejanja, liste in robove.

Ko iščemo določen računalnik, gre poizvedba rekurzivno po hierarhiji, začenši pri vrhu. Če hočete izvedeti naslov `prep.ai.mit.edu`, mora vaš imenski strežnik najprej najti strežnik, ki je odgovoren za `edu`, zato vpraša `.` strežnik (`.` strežnike že pozna, ker so zapisani v `root.hints`), ta pa mu posreduje seznam strežnikov za `edu`:

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

Začnemo s spraševanjem korenskega strežnika:

```
> server c.root-servers.net.
Default Server:  c.root-servers.net
Address:  192.33.4.12
```

Postavimo tip poizvedbe na NS (zapisi imenskih strežnikov, ang.: name server records)

```
> set q=ns
```

Vprašajmo za `edu`:

```
> edu.
```

Končna pika (.) je tu zelo pomembna, ker pove `nslookup`, da je `edu`, o katerem poizvedujemo takoj pod `.` (in ne pod eno od naših `search` domen - to pospeši iskanje).

```
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
```

To nam pove, da vsi strežniki `ROOT-SERVERS.NET` služijo tudi za `EDU.`, zato lahko sprašujemo katerega koli od njih. Mi bomo nadaljevali s spraševanjem `C`. Sedaj hočemo vedeti, kdo je strežnik za naslednji nivo domene: `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
```

`strawb`, `w20ns` in `bitsy` vsi strežejo `mit.edu`, zato izberemo enega od njih in poizvedujemo še en nivo navzgor: `ai.mit.edu`:

```
> server W20NS.mit.edu.
```

Vseeno je, ali so imena računalnikov zapisana z velikimi ali malimi črkami, vendar uporabljam miško za kopiranje in so zato vsa imena takšna, kot se izpišejo na zaslon.

```
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
```

`museli.ai.mit.edu` je torej imenski strežnik za `ai.mit.edu`:

```
> server MUESLI.AI.MIT.EDU
Default Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7
```

Spremenimo tip poizvedbe - našli smo imenski strežnik in sedaj sprašujemo za vse podatke 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
```

Tako smo s pričetkom na `.` našli imenske strežnike za vsak nivo domene. Če bi namesto ročnega iskanja vseh teh uporabili svoj DNS strežnik, bi si named seveda zapomnil vse informacije, ki bi jih našel pri iskanju in mu nekaj časa ne bi bilo treba ponovno poizvedovati.

V drevesu vsaka "`.`" v imenu pomeni razvejanje in vsak del med pikami je ime določene veje v drevesu.

Po drevesu se vzpenjamo tako, da vzamemo ime, o katerem poizvedujemo (`prep.ai.mit.edu`), najprej najdemo korenine (`.`) in nato poiščemo naslednjo vejo za vzpenjanje, v tem primeru `edu`. Ko jo najdemo, se vzpnemo po njej tako, da se priključimo na strežnik, ki je odgovoren za ta del imena. Zatem poiščemo vejo `mit`, ki raste iz veje `edu` (kombinirano ime je `mit.edu`) in se priključimo na strežnik, ki streže `mit.edu`. Spet poiščemo naslednjo vejo, tokrat `ai.mit.edu` in se priključimo na strežnik, odgovoren zanjo. Sedaj smo prispeli na končni strežnik, ki ga nato povprašamo za `prep.ai.mit.edu`. V računalniški terminologiji `prep` navadno imenujemo *list* na drevesu.

Domena, o kateri se manj govori, pa je vseeno zelo pomembna je `in-addr.arpa`. Prav tako, kot 'navadne' domene je vgnezdena, vendar nam omogoča ravno nasprotno - da izvemo imena računalnikov iz njihovih naslovov. Pomembna stvar, ki si jo morate zapomniti je, da so IP naslovi v domeni `in-addr.arpa` zapisani v obratnem vrstnem redu. Če je naslov računalnika `192.128.52.43`, je postopek, ki ga ubere named prav takšen, kot za `prep.ai.mit.edu`: najdi strežnik za `arpa.`, najdi strežnik `in-addr.arpa.`, najdi strežnik za `192.in-addr.arpa.`, najdi strežnik za `128.192.in-addr.arpa.`, najdi strežnik za `52.128.192.in-addr.arpa.` in končno najdi zapise za `43.52.128.192.in-addr.arpa.` Pametno, kajne? (Recite 'da'.) Prav dejstvo, da so številke obrnjene, lahko povzroči veliko zmešnjavo.

Pravkar sem se zlagal. DNS ne deluje natanko tako, kot sem vam povedal. Vsekakor pa deluje skoraj tako.

## 4.2 Naša lastna domena

Sedaj bomo definirali našo lastno domeno - `linux.izmislek` in v njej definirali računalnike. Uporabil sem popolnoma izmišljeno domeno; tako smo lahko gotovi, da ne bomo motili koga Tam Zunaj.

Še ena stvar, preden začnemo: imena računalnikov ne smejo vsebovati določenih znakov - omejeni smo na črke angleške abecede: a-z, številke: 0-9 in znak '-' (pomišljaj). Držite se jih. Velikosti črk DNS ne razlikuje, zato je `pat.uio.no` isto kot `Pat.UiO.No`.

Delo smo začeli že s tem razdelkom v `named.conf`:

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};
```

Prosim, zapomnite si, da na koncu domen v tej datoteki ni pike '`.`'. Ta razdelek pove, da bomo sedaj definirali `0.0.127.in-addr.arpa`, da smo za to domeno glavni strežnik in da so podatki zanjo shranjeni v datoteki `pz/127.0.0`. Tudi slednjo smo že uredili, glasi pa se:

```
@               IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                                1       ; Serijska številka
                                8H      ; Osveževanje
                                2H      ; Ponoven poskus
                                1W      ; Potek
                                1D)     ; Minimum TTL
                        NS      ns.linux.izmislek.
1                       PTR     localhost.
```

Prosim, zapomnite si pike '`.`' na koncu vseh polnih domen v tej datoteki, kot nasprotje datoteki `named.conf`. Nekateri ljudje začnejo vsako 'zone' datoteko z ukazom `$ORIGIN`, vendar je to popolnoma odveč. Izvor (kam v DNS hierarhiji spada) datoteke 'zone' je namreč naveden v `named.conf`, v tem primeru je to `0.0.127.in-addr.arpa`.

Ta datoteka 'zone' vsebuje tri zapise virov (ang.: RRs, resource records): SOA, NS in PTR. SOA pomeni začetek pristojnosti (ang.: Start Of Authority). '@' je oznaka za izvor in ker je v stolpcu 'domena' za to datoteko navedeno 0.0.127.in-addr.arpa, prva vrstica v resnici pomeni

```
0.0.127.in-addr.arpa.   IN      SOA ...
```

NS je zapis vira imenskega strežnika (ang.: Name Server RR). Na začetku te vrstice ni oznake '@', ker se to ohranja še iz prejšnje vrstice - takšen način prihrani veliko tipkanja. Zapis vira NS bi torej lahko zapisali tudi kot

```
0.0.127.in-addr.arpa.   IN      NS      ns.linux.izmislek.
```

DNS-u pove, kateri računalnik je strežnik za domeno `0.0.127.in-addr.arpa` in to je `ns.linux.izmislek`. 'ns' je standardno ime za imenske strežnike, kot je `www.`*nekaj* standardno ime za spletne strežnike, vendar to ni pravilo - ime je lahko karkoli.

Na koncu sledi še zapis vira PTR, ki pravi, da je računalniku na naslovu 1 v podmreži `0.0.127.in-addr.arpa`, torej 127.0.0.1, ime `localhost`.

*Vse* datoteke ‘zone’ morajo vsebovati natanko en zapis vira SOA. Ta zapis vsebuje podatke o domeni, od kod prihaja (računalnik z imenom `ns.linux.izmislek`), kdo je odgovoren za njeno vsebino (`hostmaster@linux.izmislek` - tukaj vstavite svoj e-mail naslov), katera različica datoteke ‘zone’ je to (serijska številka: 1) in druge podatke, ki imajo opraviti z DNS strežniki s predpomnilnikom ter sekundarnimi DNS strežniki. Za vsa ostala polja (osveževanje, ponoven poskus, potek in minimum) se zanašajte na številke, ki so navedene v tem HOWTO-ju in vse bi moralo delovati kot je treba.

Sedaj ponovno zaženite named (ukaz je `ndc restart`) in uporabite nslookup za ogled vašega dela:

```
$ 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
```

Aha, torej uspe preslikati 127.0.0.1 v `localhost`. Odlično. Sedaj se lahko spravimo k našemu glavnemu opravilu, domeni `linux.izmislek`. Najprej vstavimo nov razdelek ’zone’ v `named.conf`:

```
zone "linux.izmislek" {
        notify no;
        type master;
        file "pz/linux.izmislek";
};
```

Spet si zapomnite, da na koncu domen v `named.conf` ni pik ‘`.`’.

V datoteko ‘zone’ `linux.izmislek` bomo napisali nekaj popolnoma izmišljenih podatkov:

```
;
; Datoteka zone za linux.izmislek
;
; Popolna datoteka zone
;
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151       ; serijska številka = datum + serijska #
                        8H              ; osveževanje, sekunde
                        2H              ; ponoven poskus, sekunde
                        1W              ; potek, sekunde
                        1D )            ; minimum, sekunde
;
                NS      ns              ; Internetni naslov imenskega strežnika
                MX      10 mail.linux.izmislek     ; Primarni poštni strežnik
                MX      20 mail.prijatelj.izmislek.   ; Sekundarni poštni strežnik
```

```
;
localhost       A       127.0.0.1
ns              A       192.168.196.2
mail            A       192.168.196.4
```

O zapisu SOA si morate zapomniti dve stvari. `ns.linux.izmislek` *mora* biti resničen računalnik z zapisom A. Prepovedano je imeti zapis CNAME za računalnik, naveden v zapisu SOA. Vsekakor ni pomembno, da je njegovo ime 'ns', lahko je kakršno koli veljavno ime. Druga stvar; hostmaster.linux.izmislek se bere kot hostmaster@linux.izmislek in mora biti obstoječ naslov ali preusmeritev naslova, na katerega bo oseba, ki vzdržuje DNS prejemala pošto (ta oseba naj pošto tudi redno pregleduje). Ni nujno, da je to ravno 'hostmaster' - lahko je vaš navaden e-mail naslov, vendar se pogosto pričakuje, da obstaja tudi naslov 'hostmaster'.

V tej datoteki je nov zapis vira, in sicer MX, kar pomeni poštni strežnik (ang.: Mail eXchanger). Ta zapis pove sistemom, kam naj pošljejo pošto, ki je naslovljena na `nekdo@linux.izmislek`, v tem primeru na `mail.linux.izmislek` ali `mail.prijatelj.izmislek`. Številka pred imeni računalnikov pomeni prednost. Če je le mogoče, se pošta pošlje strežniku z najmanjšo številko (10), sicer pa naslednjemu z najmanjšo številko, v tem primeru `mail.prijatelj.izmislek`, ki ima prednost 20.

Ponovno zaženite named z ukazom `ndc restart` in opazujte rezultate z nslookup:

```
$ nslookup
> set q=any
> linux.izmislek
Server:  localhost
Address:  127.0.0.1

linux.izmislek
        origin = ns.linux.izmislek
        mail addr = hostmaster.linux.izmislek
        serial = 199802151
        refresh = 28800 (8 hours)
        retry   = 7200 (2 hours)
        expire  = 604800 (7 days)
        minimum ttl = 86400 (1 day)
linux.izmislek     nameserver = ns.linux.izmislek
linux.izmislek     preference = 10, mail exchanger = mail.linux.izmislek.linux.izmislek
linux.izmislek     preference = 20, mail exchanger = mail.prijatelj.izmislek
linux.izmislek     nameserver = ns.linux.izmislek
ns.linux.izmislek  internet address = 192.168.196.2
mail.linux.izmislek        internet address = 192.168.196.4
```

Po pazljivem opazovanju boste odkrili napako. Vrstica

```
linux.izmislek     preference = 10, mail exchanger = mail.linux.izmislek.linux.izmislek
```

je popolnoma napačna. Morala bi biti

```
linux.izmislek     preference = 10, mail exchanger = mail.linux.izmislek
```

Namenoma sem naredil napako, da se boste lahko iz nje naučili :-) Če pogledate v datoteko 'zone', boste ugotovili, da v vrstici

```
MX      10 mail.linux.izmislek     ; Primary Mail Exchanger
```

manjka pika. Ali drugače povedano, ima 'linux.izmislek' preveč. Če se v datoteki 'zone' ime računalnika ne konča s piko, se mu doda izvor in v tem primeru povzroči dvojen `linux.izmislek.linux.izmislek`. Torej je

```
                MX      10 mail.linux.izmislek.    ; Primary Mail Exchanger
```

ali

```
                MX      10 mail                 ; Primary Mail Exchanger
```

pravilno. Priporočam uporabo druge možnosti, ker je potrebnega manj tipkanja. Nekateri bind eksperti se s tem ne bodo strinjali, nekateri se bodo. V datoteki 'zone' je lahko domena napisana in se konča s '`.`', ali pa ni napisana in se namesto nje privzame izvorna domena.

Poudariti moram, da v named.conf *ne sme* biti pike na koncu domen. Še sanja se vam ne, kolikokrat je '`.`' preveč ali premalo zamešala stvari in popolnoma zmedla ljudi.

Zdaj, ko sem vam vse to povedal, je tu nova, popravljena datoteka 'zone' s še nekaj dodatnimi informacijami:

```
;
; Datoteka zone za linux.izmislek
;
; Popolna datoteka zone
;
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151       ; serijska številka = datum + serijska #
                        8H              ; osveževanje, sekunde
                        2H              ; ponoven poskus, sekunde
                        1W              ; potek, sekunde
                        1D )            ; minimum, sekunde
;
                TXT     "Linux.Izmislek, vaši DNS konzultanti"
                NS      ns              ; Internetni naslov imenskega strežnika
                NS      ns.prijatelj.izmislek.
                MX      10 mail         ; Primarni poštni strežnik
                MX      20 mail.prijatelj.izmislek. ; Sekundarni poštni strežnik

localhost       A       127.0.0.1

gw              A       192.168.196.1
                HINFO   "Cisco" "IOS"
                TXT     "Usmerjevalnik"

ns              A       192.168.196.2
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "Pentium" "Linux 2.0"
```

```
www             CNAME   ns


donald          A       192.168.196.3
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "i486"  "Linux 2.0"
                TXT     "DEK"

mail            A       192.168.196.4
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "386sx" "Linux 1.2"

ftp             A       192.168.196.5
                MX      10 mail
                MX      20 mail.prijatelj.izmislek.
                HINFO   "P6" "Linux 2.1.86"
```

Tukaj smo spoznali še cel kup novih zapisov: HINFO, informacija o računalniku (ang.: Host INFOrmation), ima dva dela in dobra navada je, da ju izpolnite. Prvi del je strojna oprema ali procesor računalnika, drugi pa operacijski sistem. Računalnik z imenom 'ns' ima procesor Pentium in poganja Linux 2.0. CNAME, kanonično ime (ang.: Canonical NAME) je način za poimenovanje istega računalnika z več različnimi imeni. V našem primeru je www alternativno ime za ns.

Uporaba zapisa CNAME je nekoliko sporna, vendar se je popolnoma varno ravnati po pravilu, da se zapisi MX, CNAME in SOA ne smejo *nikoli* nanašati na zapis CNAME, temveč samo na nekaj z zapisom A. Na primer, ni priporočljivo, da imate

```
foobar          CNAME   www                     ; NE!
```

medtem, ko je takole pravilno

```
foobar          CNAME   ns                      ; Da!
```

Prav tako se je varno zanašati na to, da CNAME ni dovoljeno ime računalnika za e-mail naslov: `webmaster@www.linux.izmislek` na primer ni dovoljen naslov glede na zgoraj navedene nastavitve. Pričakujete lahko, da bo kar nekaj administratorjev Tam Zunaj hotelo, da to pravilo upoštevate, čeprav za vas morda deluje. Način, da preprečite težave je uporaba zapisov A (in še nekaterih drugih, na primer MX):

```
www             A       192.168.196.2
```

Nekaj bind strokovnjakov celo priporoča, naj se zapisi CNAME *sploh* ne uporabljajo, vendar razprava 'zakaj' in 'zakaj ne' ne spada v ta HOWTO.

Kot ste lahko že opazili, se ta HOWTO in mnogo internetnih strani ne drži tega pravila.

Naložite novo bazo podatkov z ukazom `ndc reload`, ki pove named, naj še enkrat prebere svoje datoteke.

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

> ls -d linux.izmislek
```

Ta ukaz prelista vse zapise. Rezultat je sledeč:

```
[localhost]
$ORIGIN linux.izmislek.
@                       1D IN SOA       ns hostmaster (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                        1D IN NS        ns
                        1D IN NS        ns.prijatelj.izmislek.
                        1D IN TXT       "Linux.Izmislek, vaši DNS konzultanti"
                        1D IN MX        10 mail
                        1D IN MX        20 mail.prijatelj.izmislek.
gw                      1D IN A         192.168.196.1
                        1D IN HINFO     "Cisco" "IOS"
                        1D IN TXT       "Usmerjevalnik"
mail                    1D IN A         192.168.196.4
                        1D IN MX        10 mail
                        1D IN MX        20 mail.prijatelj.izmislek.
                        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.prijatelj.izmislek.
                        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.prijatelj.izmislek.
                        1D IN HINFO     "P6" "Linux 1.3.59"
ns                      1D IN A         192.168.196.2
                        1D IN MX        10 mail
                        1D IN MX        20 mail.prijatelj.izmislek.
                        1D IN HINFO     "Pentium" "Linux 1.2"
```

Tole je v redu. Kot lahko vidite, je izpis zelo podoben sami datoteki 'zone'. Poglejmo, kaj pravi samo za `www`:

```
> set q=any
> www.linux.izmislek.
Server:  localhost
```

```
Address:  127.0.0.1

www.linux.izmislek canonical name = ns.linux.izmislek
linux.izmislek     nameserver = ns.linux.izmislek
linux.izmislek     nameserver = ns.prijatelj.izmislek
ns.linux.izmislek  internet address = 192.168.196.2
```

Z drugimi besedami, pravo ime `www.linux.izmislek` je `ns.linux.izmislek`, poleg tega pa vam poda še nekaj drugih informacij o ns, dovolj, da bi se lahko priključili nanj, če bi bili program.

Sedaj smo na polovici poti.

## 4.3 Obratni vnosi

Trenutno lahko programi pretvorijo imena v linux.izmislek v naslove, na katere se lahko priključijo, potrebujemo pa še obratne vnose, ki bodo omogočili DNS-u, da pretvarja naslove v imena. Imena so pomembna informacija za mnoge strežnike (FTP, IRC, WWW in druge), ki na podlagi tega odločajo, ali se bodo sploh pogovarjali z vami ali ne ter kakšno prednost vam bodo dali v primeru, da se bodo. Za popoln dostop do uslug na internetu potrebujete obratne vnose.

Dodajte tole v `named.conf`:

```
zone "196.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "pz/192.168.196";
};
```

Stvar je natanko takšna, kot pri `0.0.127.in-addr.arpa`, pa tudi vsebina je podobna:

```
@       IN      SOA     ns.linux.izmislek. hostmaster.linux.izmislek. (
                        199802151 ; Serijska številka = datum + serijska #
                        8H      ; Osveževanje
                        2H      ; Ponoven poskus
                        1W      ; Potek
                        1D)     ; Minimum TTL
                NS      ns.linux.izmislek.

1               PTR     gw.linux.izmislek.
2               PTR     ns.linux.izmislek.
3               PTR     donald.linux.izmislek.
4               PTR     mail.linux.izmislek.
5               PTR     ftp.linux.izmislek.
```

Ponovno zaženite named (`ndc restart`) in preglejte vaše delo z nslookup:

```
> 192.168.196.4
Server:  localhost
Address:  127.0.0.1

Name:    mail.linux.izmislek
Address:  192.168.196.4
```

Zaenkrat izgleda v redu, preglejmo še ostale dele:

```
> ls -d 196.168.192.in-addr.arpa
[localhost]
$ORIGIN 196.168.192.in-addr.arpa.
@                       1D IN SOA       ns.linux.izmislek. hostmaster.linux.izmislek. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                        1D IN NS        ns.linux.izmislek.
1                       1D IN PTR       gw.linux.izmislek.
2                       1D IN PTR       ns.linux.izmislek.
3                       1D IN PTR       donald.linux.izmislek.
4                       1D IN PTR       mail.linux.izmislek.
5                       1D IN PTR       ftp.linux.izmislek.
@                       1D IN SOA       ns.linux.izmislek. hostmaster.linux.izmislek. (
                                        199802151       ; serial
                                        8H              ; refresh
                                        2H              ; retry
                                        1W              ; expiry
                                        1D )            ; minimum
```

Odlično! Če vaš izpis ne izgleda takole, poglejte v syslog za sporočila o napakah, kot sem razložil na začetku tega poglavja.

## 4.4 Opozorila

k0Rad bi dodal še nekaj stvari. IP številke, ki sem jih uporabil v primerih, navedenih tukaj, so vzete iz blokov 'privatnih mrež', kar pomeni, da se jih v internetu ne sme uporabljati za prave naslove. So pa zato varne za uporabo v primerih. Druga stvar, na katero vas moram opozoriti je vrstica `notify no;`, ki pove named, naj svojih sekundarnih strežnikov ne opozarja na spremembe datotek 'zone'. V bind-8 lahko named opozori druge strežnike, navedene v zapisih NS, kadar se kakšna od teh spremeni. To je sicer pripravno pri resnični uporabi, za privatne poskuse pa mora biti izključeno - saj vendar nočemo, da bi naši poskusi onesnažili internet, kajne?

In seveda, ta domena je popolnoma izmišljena, kot so izmišljeni tudi naslovi v njej. Za primer prave domene glejte naslednje poglavje.

### 4.5 Zakaj obratne poizvedbe ne delujejo.

Obstaja nekaj zank pri postavljanju obratnih vnosov, ki jih navadno rešimo z imenskimi poizvedbami. Preden greste naprej, potrebujete delujoče obratne vnose - če vam ne delujejo, pojdite nazaj in jih popravite.

Razložil bom dve možnosti, zaradi katerih obratni vnosi ne delujejo, kot so videti iz izven vašega omrežja.

#### 4.5.1 Obratni vnosi niso pristojni.

Ko dobite od ponudnika svoj del omrežja in domeno, je domena navadno pristojna za ta del omrežja. Pristojnost je zapis NS, lepilo, ki vam omogoča priti od enega imenskega strežnika do drugega, kot sem pojasnil v poglavju o teoriji. Če vaši obratni vnosi ne delujejo, pojdite nazaj in ga preberite. Zdaj.

Tudi obratni vnosi morajo biti pristojni za določen del omrežja. Če ste dobili omrežje `192.168.192` z domeno `linux.izmislek`, mora ponudnik v svoje datoteke 'zone' zapisati tako vnos NS za imenske poizvedbe, kot tudi vnos NS za obratne poizvedbe. Če sledite verigi od `in-addr.arpa.` do vas, boste verjetno našli prekinitev - najverjetneje pri vašem ponudniku. Kontaktirajte ga in ga prosite za odpravo napake.

#### 4.5.2 Imate brezrazredno podmrežo

To je dokaj zapletena tema, vendar so brezrazredne podmreže dandanes zelo pogoste in če niste vsaj srednje veliko podjetje, ste v njej najverjetneje tudi vi.

Brezrazredne podmreže danes ohranjajo internet pri življenju. Nekaj let nazaj je bilo veliko razprav o pomanjkanju IP številk. Pametni ljudje v IETF (Internet Engineering Task Force, organizacija, ki skrbi za internet) so staknili glave in iznašli rešitev. Za določeno ceno. Ta cena je, da dobite manj kot "C" podmrežo, poleg tega pa se lahko zgodi, da nekaj stvari ne bo delalo. *Vprašajte Gospoda DNS* na *http://www.acmebw.com/askmrdns/00007.htm* za dobro razlago, kako se spopasti s tem problemom.

Ste prebrali? Tega ne bom razlagal, zato si, prosim, preberite.

Prvi del problema je, da mora vaš ponudnik interneta razumeti tehniko, ki jo opisuje Gospod DNS. Nekateri majhni ponudniki je ne razumejo popolnoma, zato jim jo boste morda morali razložiti. Prej se prepričajte, da jo sami obvladate ;-) Ponudnik bo nato postavil obratne vnose na njihovem strežniku, kar lahko preverite z nslookup.

Drugi in zadnji del problema pa je, da morate tudi vi razumeti tehniko. Če o tem niste prepričani, pojdite nazaj in preberite še enkrat, nato pa lahko postavite vaše lastne brezrazredne obratne vnose, kot vam to opisuje Gospod DNS.

Obstaja pa še ena težava. Stari programi *ne bodo* sposobni slediti triku `CNAME` v verigi poizvedovanja in zato ne bodo sposobni naresti obratnih poizvedb za vaš računalnik. To lahko pomeni, da vas bo strežnik razvrstil v napačen razred, vam prepovedal dostop oziroma storil kaj podobnega. Če njegove usluge resnično potrebujete, je edina možnost (ki jo jaz poznam), da poprosite vašega ponudnika, naj vstavi vaš PTR zapis v njihovo brezrazredno datoteko 'zone', namesto v trik s CNAME vnosom.

Nekateri ponudniki interneta vam bodo ponudili tudi drugačne rešitve, kot so spletni forumi za obratne vnose in podobni avtomagični sistemi.

## 5 Primer prave domene

**Tukaj je nekaj *pravih* datotek 'zone'**

Uporabniki so predlagali, da poleg učnih primerov vključim tudi datoteke prave in delujoče domene.

Te primere uporabljam z dovoljenjem Davida Bullocka z LAND-5. Datoteke so bile posnete 24. septembra 1996 in sem jih nato nekoliko popravil, da delujejo z bind 8, zato se to, kar vidite tu, nekoliko razlikuje od rezultatov, ki jih dobite iz poizvedovanja pri imenskih strežnikih LAND-5.

## 5.1 /etc/named.conf (ali /var/named/named.conf)

Tukaj lahko najdemo glavne razdelke 'zone' za dvoje obratne vnose: omrežje 127.0.0 in omrežje LAND-5 `206.6.177`. Tu je tudi primarni razdelek za `land-5.com`. Pozorni bodite na to, da datoteke niso shranjene v mapi `pz`, kot v tem HOWTO-ju, temveč v mapi `zone`.

```
// Boot file for LAND-5 name 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";
};
```

Če boste to datoteko skopirali v vaš named.conf, da bi se igrali, **PROSIM**, vpišite še vrstico "`notify no;`" v razdelke 'zone' za obe `land-5`, da preprečite nesreče.

## 5.2 /var/named/root.hints

Zapomnite si, da je ta datoteka dinamična in zato tale ni več veljavna. Svojo lahko naredite z dig-om, kot sem razložil.

```
; <<>> 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

Samo osnove; nujen zapis SOA in zapis, ki preslika 127.0.0.1 v `localhost`. Ni potrebno, da je v tej datoteki še kaj. Verjetno ne bo nikoli potrebno, da jo popravljate, razen, če se spremeni imenski strežnika ali e-mail administratorja DNS.

```
@               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

Tukaj lahko vidimo zapis SOA in potreben zapis NS. Kot lahko opazite, ima LAND-5 tudi sekundarni imenski strežnik na `ns2.psi.net`. Tako tudi mora biti: *vedno* mora obstajati izven določene strani še sekundarni imenski strežnik, za vsak slučaj. Vidimo lahko, da obstaja glaven računalnik, `land-5`, ki skrbi za veliko različnih internetnih uslug in ima tudi nekaj alternativnih imen narejenih s CNAME (druga možnost je uporaba zapisov A).

Iz zapisa SOA je razvidno, da domena izvira iz `land-5.com`, kontaktna oseba je `root@land-5.com`. Tudi `hostmaster` je pogosto uporabljen naslov za kontaktno osebo. Serijska številka je v formatu yyyymmdd (leto, mesec, dan) s pripeto dnevno serijsko številko; to je verjetno šesta verzija te datoteke na dan 20. septembra 1996. Zapomnite si, da se serijska številka *mora* povečevati monotono in ker je današnja serijska številka omejena na *eno* cifro, mora po 9 popravkih počakati do naslednjega dne, da lahko ponovno popravlja datoteko. Izognite se uporabi dveh cifer.

```
@       IN      SOA     land-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
                TXT     "LAND-5 Corporation"

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

;
;       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
```

Če v resnici opazujete imenske strežnike land-5, boste ugotovili, da so imena računalnikov v obliki `ws_`*številka*. Po bind različici 4 je named namreč začel zahtevati, da se točno držite znakov, ki se smejo uporabiti v imenih. Bind-8 s tem sploh ne bi deloval več, zato sem zamenjal '_' s pomišljajem (-).

Še ena zanimiva stvar je dejstvo, da delovne postaje nimajo individualnih imen, temveč so le-ta sestavljena iz predpone in zadnjih delov IP številk. Uporaba takšnega sistema lahko močno olajša vzdrževanje, vendar je precej neosebna in je lahko povod za jezo vaših strank.

Vidimo lahko tudi, da je `funn.land-5.com` alternativno ime za `land-5.com`, vendar z uporabo zapisa A in ne CNAME. To je dobra politika, kot sem povedal že prej.

## 5.5 /var/named/zone/206.6.177

Komentarji na to datoteko sledijo za njo.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609206       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; 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.
```

Obratni vnosi so del nastavitev, ki povzroča največ težav. Uporabljajo se za iskanje imena računalnika, če imate njegovo IP številko. Primer: ste IRC strežnik in sprejemate povezave IRC klientov. Toda - ste norveški IRC strežnik in želite sprejemati le povezave klientov na norveškem in v drugih skandinavskih država. Ko se na vas priključi klient, vam lahko vaša C knjižnica pove IP številko računalnika, ki se je priključil, ker je le-ta napisana v vsakem paketu podatkov, ki pride po internetu. Sedaj lahko pokličete funkcijo gethostbyaddr, ki preslika IP številko v ime računalnika. Gethostbyaddr bo vprašal DNS strežnik in ta bo prevzel iskanje imena. Predpostavimo, da se je poskusil priključiti klient z ws-177200.land-5.com. IP številka, ki vam jo pove C knjižnica, je 206.6.177.200. Da ugotovimo ime računalnika moramo najti `200.177.6.206.in-addr.arpa`. DNS strežnik bo najprej našel strežnike za `arpa.`, nato strežnike za `in-addr.arpa.` in sledil po številkah 206 in 6 ter končno našel strežnik za `177.6.206.in-addr.arpa.` na LAND-5. Od njega bo dobil odgovor, da ima za `200.177.6.206.in-addr.arpa` zapis “`PTR ws-177200.land-5.com`”, kar pomeni, da k naslovu `206.6.177.200` spada ime `ws-177200.land-5.com`.

Pojdimo nazaj k primeru IRC strežnika. Strežnik sprejema samo povezave iz skandinavskih dežel, to je `*.no`, `*.se` in `*.dk`. Ime `ws-177200.land-5.com` vsekakor ne ustreza nobeni od teh možnosti, zato bo strežnik preprečil povezavo. Če *ne bi* bilo obratnih preslikav za `206.6.177.200` čez `in-addr.arpa`, strežnik sploh ne bi mogel najti imena računalnika in vse, kar bi imel za primerjati z `*.no`, `*.se` in `*.dk` bi bilo `206.2.177.200` - kar se seveda ne bi ujemalo.

Nekateri ljudje vam bodo govorili, da obratne poizvedbe sploh niso pomembne za strežnike, oziroma, da sploh niso pomembne. Ni res: mnogi ftp, novičarski, IRC in celo nekateri http (WWW) strežniki vam *ne bodo* dovolili, da se priključite z računalnika, katerega imena ne morejo izvedeti. Zato so obratne poizvedbe v resnici izredno pomembne.

# 6 Vzdrževanje

**Skrbite, da bo vse delovalo.**

Poleg tega, da skrbite, da bo vse delovalo, imate še eno nalogo - vzdržujte datoteko `root.hints` točno. To je najenostavneje z uporabo programa dig. Najprej ga zaženite brez argumentov in dobili boste `root.hints` glede na vaš strežnik, nato pa zaženite dig še z enim od korenskih strežnikov: `dig @korenski-strežnik`. Izhod, ki ga dobite, je nova datoteka root.hints. Shranite ga v datoteko (`dig @e.root-servers.net . ns >root.hints.nov`) in zamenjajte staro datoteko `root.hints` z njo.

Ne pozabite ponovno zagnati named potem, ko ste zamenjali datoteko.

Al Longyear mi je poslal skript za vzdrževanje `root.hints`, ki ga lahko poganjate avtomatično - v crontab vpišite vnos, ki ga bo pognal vsak mesec, nato pa lahko nanj pozabite. Skript predpostavlja, da imate delujočo elektronsko pošto in da na vašem računalniku obstaja naslov 'hostmaster'. Da bo ustrezala vašim nastavitvam, jo morate pohekati sami.

---

```
#!/bin/sh
#
# Posodobimo predpomnilniško datoteko imenskega strežnika enkrat na mesec.
# Ta skript poganja avtomatično vnos v crontabu.
#
# Original je napisal Al Longyear
# Za bind 8 priredil Nicolai Langfeldt
# David A. Ranch je poročal o različnih možnih napakah
# Test s pingom je predlagal Martin Foster
#
(
 echo "To: hostmaster <hostmaster>"
 echo "From: system <root>"
 echo "Subject: Avtomatična posodobitev datoteke root.hints"
 echo

 PATH=/sbin:/usr/sbin:/bin:/usr/bin:
 export PATH
 cd /var/named

 # Smo povezani v internet? Pingajmo strežnik vašega ponudnika
 case `ping -qnc ponudnikov.streznik.net` in
   *'100% packet loss'*)
        echo "Ni povezave z internetom. Datoteka root.hints ni bila posodobljena"
        echo
        exit 0
        ;;
```

```
 esac

 dig @rs.internic.net . ns >root.hints.nov 2>&1

 case `cat root.hints.nov` in
   *NOERROR*)
        # Delovalo je
        :;;
   *)
        echo "Posodobitev datoteke root.hints NI USPELA."
        echo "Dig je sporočil tole:"
        echo
        cat root.hints.nov
        exit 0
        ;;
 esac

 echo "Datoteka root.hints sedaj vsebuje naslednje informacije:"
 echo
 cat root.hints.nov

 chown root.root root.hints.nov
 chmod 444 root.hints.nov
 rm -f root.hints.star
 mv root.hints root.hints.star
 mv root.hints.nov root.hints
 ndc restart
 echo
 echo "Imenski strežnik je bil ponovno zagnan in sedaj ve za morebitne spremembe."
 echo "Prejšnja datoteka root.hints se zdaj imenuje /var/named/root.hints.star."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Morda ste izvedeli, da je datoteka `root.hints` na voljo tudi preko ftp na Internic. Prosim vas, da za posodobitev `root.hints` uporabljajte raje dig, ker je ta možnost bolj prijazna tako do interneta kot do Internica.

# 7 Pretvarjanje iz različice 4 v različico 8

To poglavje, ki ga je napisal David E. Smith (dave@bureau42.ml.org) je bilo v originalu poglavje o uporabi bind 8. Da se sklada z novim naslovom, sem ga nekoliko popravil.

Na to temo ni potrebno veliko besed. Razen tega, da se uporablja named.conf namesto named.boot, je vse popolnoma enako. Bind8 ima tudi skript v perlu, ki vam pretvori stare datoteke v nove. Primer named.boot (starejša različica) za imenski strežnik s predpomnilnikom:

```
directory /var/named
```

```
cache   .                                         root.hints
primary 0.0.127.IN-ADDR.ARPA                      127.0.0.zone
primary localhost                                 localhost.zone
```

V mapi bind8/src/bin/named (*tole predvideva, da imate distribucijo izvorne kode. Če imate paket s prevedenim programom, je skript verjetno prav tako nekje inštaliran, vendar ne vem točno, kje. -ed*) v ukazno vrstico napišite:

```
./named-bootconf.pl < named.boot > named.conf
```

To naredi 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";
};
```

Deluje za vse, kar je lahko v `named.boot`, čeprav ne doda novih pridobitev in možnosti, ki jih omogoča bin8. Tule je bolj popoln `named.conf`, ki deluje enako, vendar nekoliko bolj učinkovito.

```
// To je nastavitvena datoteka za named (BIND 8.1 ali novejši).
// Navadno bo inštalirana v /etc/named.conf
// Edina razlika med to in 'že pripravljeno' datoteko (poleg tega
// komentarja :) je to, da sem odkomentiral vrstico 'directory', ker
// sem datoteke 'zone' že imel v /var/named.

options {
        directory "/var/named";
        datasize 20M;
};
```

```
zone "localhost" IN {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "127.0.0.zone";
};

zone "." IN {
        type hint;
        file "root.hints";
};
```

V distribuciji bind 8 lahko v mapi bind8/src/bin/named/test najdete tole in nekaj izvodov datotek ‘zone’, ki jih večina ljudi lahko le skopira in jih uporabi takoj.

Formati datotek ‘zone’ in `root.hints` so enaki, tako kot tudi ukazi za njihovo posodobljanje.

# 8 Vprašanja in odgovori

Prosim, preberite to poglavje preden me kontaktirate.

1. Moj named zahteva datoteko named.boot

   Berete napačen HOWTO. Poglejte si starejšo različico tega HOWTO-ja, ki pokriva bind 4, na *http://www.math.uio.no/~janl/DNS/*

2. Kako uporabljam DNS izza požarnega zidu?

   Namig: `forward only;`, Verjetno boste potrebovali tudi

   ```
   query-source port 53;
   ```

   v razdelku “options” v datoteki `named.conf`, kot je napisano v primeru 3 (Imenski strežnik s predpomnilnikom).

3. Kako naj prepričam DNS, da periodično obrača razpoložljive naslove za določeno uslugo, na primer za www.zelo-zaseden.com, da se obremenitev porazdeli na več računalnikov?

   Naredite več **A** zapisov za www.zelo-zaseden.com in uporabite bind 4.9.3 ali novejšega. Potem bo bind avtomatsko spreminjal svoje odgovore. S prejšnjimi različicami binda to *ne bo* delovalo.

4. Rad bi postavil DNS na (zaprtem) intranetu. Kaj moram storiti?

   Izpustite datoteko root.hints in napišite le datoteke ‘zone’. To tudi pomeni, da vam ne bo treba posodabljati datoteke root.hints.

5. Kako postavim sekundarni (pomožni) imenski strežnik?

   Če ima primarni/glavni strežnik naslov 127.0.0.1, dodajte tale razdelek v named.conf sekundarnega strežnika:

```
zone "linux.bogus" {
      type slave;
      file "sz/linux.bogus";
      masters { 127.0.0.1; };
};
```

---

V polje `masters` lahko vpišete tudi več glavnih strežnikov, ločenih s ';' (podpičji).

6. Rad bi poganjal bind tudi, ko nisem priključen na internet.

   Na to temo so napisane tri rešitve:

   - To pismo sem prejel od Iana Clarka, <ic@deakin.edu.au>. V njem opisuje, kako to stori.

     ```
     Named tukaj poganjam na računalniku za 'Maškerado'. Imam dve datoteki
     root.hints, ena se imenuje root.hints.prava in vsebuje resnične podatke,
     druga pa je root.hints.neprava in vsebuje...

     ----
     ; root.hints.neprava
     ; ta datoteka ne vsebuje nobenih podatkov
     ----

     Ko se izključim iz interneta, skopiram datoteko root.hints.neprava v
     root.hints in ponovno zaženem named.

     Ko pa se spet priključim v internet, skopiram root.hints.prava v root.hints
     in ponovno zaženem named.

     Oboje naredita skripta ip-down & ip-up.

     Prvič, ko naredim poizvedbo o domeni, named nima podatkov o njej in v
     'messages' napiše...

     Jan 28 20:10:11 hazchem named[10147]: No root nameserver for class IN

     s čimer lahko živim.

     Zame vsekakor odlično deluje. Imenski strežnik lahko uporabljam za lokalne
     računalnike, medtem, ko niso priključeni v internet, in sicer brez poteka
     časa za zunanje domene, medtem, ko pa sem priključen v internet, delujejo
     poizvedbe normalno.
     ```

   - Od Karl-Maxa Wagnerja sem prejel tudi informacijo, kako bind deluje vzajemno z NFS in portmapper-jem na računalniku, ki večinoma ni priključen v internet:

     ```
     Navadno poganjam lasten named na vseh računalnikih, ki so le občasno
     priključeni v internet preko modema. Imenski strežnik se obnaša le kot
     predpomnilnik, nima področja pristojnosti in za vse poizvedbe sprašuje
     strežnike iz datoteke root.hints. Kot je v navadi za Slackware, se tudi
     zažene pred nfsd in mountd.
     ```

```
Z enim od mojih računalnikov (Libretto 30 notebook) sem imel težavo - včasih
sem ga lahko dosegel z drugega sistema, priključenega v moj LAN, večino časa
pa to ni delovalo. Ista težava se je pojavila ne glede nato, ali sem
uporabljalj PLIP, PCMCIA mrežno kartico ali PPP po serijski povezavi.

Po nekaj ugibanja in poskušanja sem ugotovil, da je named nekako pokvaril
postopek registracije, ki ga nfsd in mountd izvedeta s portmapper-jem pri
zagonu (navadno takrat, ko zaženem računalnik). Zagon named po nfsd in
mountd je to težavo popolnoma odpravil.

Ker tako spremenjena zagonska procedura nima stranskih učinkov, vam
priporočam, da jo popravite in tako preprečite morebitne težave.
```

- Zadnja možnost so HOWTO informacije *Vprašajte Gospoda DNS* na *http://www.acmebw.com/askmrdns/#linux-ns*, ki pa govorijo o bind 4 in jih morate zato prilagoditi za bind 8.

7. Kje ima named svoj predpomnilnik? Lahko kako vplivam na njegovo velikost?

   Predpomnilnik je v celoti shranjen v spominu in ni *nikoli* zapisan na disk, zato vsakič, ko ubijete named, izgubite njegovo vsebino. Predpomnilnika *ni mogoče* upravljati. Named ga uredi po nekih preprostih pravilih in to je to. Njegove vsebine in velikosti ne morete na noben način spremeniti ali omejiti. Če želite, lahko to "popravite" s hekanjem named, vendar vam tega ne priporočam.

8. Ali named shranjuje predpomnilnik med ponovnimi zagoni? Ga lahko pripravim do tega, da ga shrani?

   Named svojega predpomnilnika *ne* shranjuje, ko umre. To pomeni, da mora predpomnilnik zgraditi na novo vsakič, ko ga ubijete in ponovno zaženete. *Ni načina*, da bi named shranil svoj predpomnilnik v datoteko. Če želite, lahko to "popravite" s hekanjem named, vendar vam tega ne priporočam.

9. Kako lahko dobim domeno? Želel bi svojo lastno domeno (na primer) `linux-rules.net`. Kako bi lahko to domeno pripisal nase?

   Kontaktirajte svojega ponudnika interneta, on vam bo zagotovo lahko pomagal. Vendar si zapomnite, da je v večini delov sveta potrebno za domeno plačati nekaj denarja.

# 9 Kako postati DNS administrator za dalj časa.

### Dokumentacija in orodja.

Prava Dokumentacija obstaja, na internetu in v tiskani obliki. Potrebno je, da preberete nekaj te dokumentacije in tako naredite korak od majhnega do velikega administratorja. V tiskani obliki je standardna knjiga *DNS in BIND* avtorja sta C. Liu in P. Albitz iz O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. Sam sem jo prebral in je odlična, čeprav se opira na bind 4, kar ni velik problem. Poglavje o DNS-u je tudi v *Administracija omrežja TCP*IP/, čigar avtor je Craig Hunt iz O'Reilly..., ISBN 0-937175-82-X. Še eno nujno branje za Dobro administracijo DNS-a (oziroma dobro karkoli) je *Zen in umetnost vzdrževanja motornih koles*, avtor Robert M. Pirsig :-) Dosegljivo je kot ISBN 0688052304 in druge.

Na internetu boste našli dobre stvari na `<http://www.dns.net/dnsrd/>` (Mapa z zapisi o DNS), `<http://www.isc.org/bind.html>`; FAQ, in referenčni priročnik (BOG; Bind Operations Guide), pa tudi članki in definicije protokola in DNS trikov (skoraj, če ne popolnoma vse RFC-je navedene spodaj najdete tudi v distribuciji bind-a). Večino teh sploh nisem prebral in zato tudi nisem velik DNS administrator. Arnt Gulbrandsen pa je prebral BOG in je popolnoma zanesen :-). Obstaja novičarska skupina o DNS, *comp.protocols.tcp-ip.domains*, poleg tega pa je na internetu tudi veliko število RFC-jev o DNS-u. Najpomembnejši so verjetno tile:

**RFC 2052**

A. Gulbrandsen, P. Vixie, *DNS RR za določevanje lokacij servisov (DNS SRV)*, Oktober 1996

**RFC 1918**

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Dodelitev naslovov za privatna omrežja*, 02/29/1996.

**RFC 1912**

D. Barr, *Pogoste napake pri obratovanju in konfiguraciji DNS-a*, 02/28/1996.

**RFC 1912 Errors**

B. Barr *Napake v RFC-ju 1912*, dosegljiv je na *http://www.cis.ohio-state.edu/˜barr/rfc1912-errors.html*

**RFC 1713**

A. Romao, *Orodja za razhroščevanje DNS-a*, 11/03/1994.

**RFC 1712**

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS kodiranje geografskega položaja*, 11/01/1994.

**RFC 1183**

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *Nove definicije DNS RR*, 10/08/1990.

**RFC 1035**

P. Mockapetris, *Domene - izvedba in specifikacije*, 11/01/1987.

**RFC 1034**

P. Mockapetris, *Domene - koncepti in možnosti*, 11/01/1987.

**RFC 1033**

M. Lottor, *Vodič za delovanje administratorjev domen*, 11/01/1987.

**RFC 1032**

M. Stahl, *Vodič za administratorje domen*, 11/01/1987.

**RFC 974**

C. Partridge, *Usmerjanje pošte in sistem domen*, 01/01/1986.