

DNS

DOMAIN NAME SYSTEM

Como ser um administrador DNS em pouco tempo.

Nicolai Langfeldt janl@math.uio.no

Tradução: Ivan Luis Seibel <mailto:seibel@infsr.unijui.tche.br>

Índice Geral:

1. Introdução.
2. Servidor de nomes de ``um cache''.
 - 2.1. /var/named/root.cache
 - 2.2. /etc/nsswitch.conf
 - 2.3. /etc/host.conf
 - 2.4. Inicialização de named
3. Um domínio Simples.
 - 3.1. Primeiro algo de teoria.
 - 3.2. Nosso próprio domínio.
 - 3.3. Relacionamentos.
4. Um exemplo de domínio real
 - 4.1. /etc/named.boot (o /var/named/named.boot)
 - 4.2. /var/named/root.cache
 - 4.3. /var/named/zone/127.0.0
 - 4.4. /var/named/zone/land-5.com
 - 4.5. /var/named/zone/206.6.177
5. Manutenção
6. Configuração de Conexões Automáticas via telefone
7. Perguntas Frequentes (FAQ)
 - 7.1. Como uso DNS desde dentro de um firewall?
 - 7.2. Como faço que DNS rote mostre as direções possíveis para um serviço, por exemplo para `www.sempre.ocupado` para obter balanço de carga ou similar?
 - 7.3. Quero configurar DNS em uma intranet (fechada). O que faço?
 - 7.4. Meu sistema não tem o programa `ndc` . O que faço?
 - 7.5. Como configuro um servidor de nomes secundário?
 - 7.6. Quero que `bind` se execute quando me desconecto da rede.
 - 7.7. Onde armazena seu cache não servidor de nomes? Há alguma forma de controlar o tamanho do cache?
 - 7.8. Salva `named` não cache nas reinicializações? Posso guardá-lo?
8. Como fazer-se um grande administrador DNS.

1. Introdução.

O que é isto e o que não é.

Para os que começam (como vc), DNS é o Domain Name System (sistema de Nomes de Domínio), as regras de nomenclatura das máquinas e o software que mapeia os nomes a números IP. Este documento trata de como definir tais conversões usando um sistema Linux. Uma conversão é simplesmente uma associação entre duas coisas, neste caso um nome de máquina, como ftp.linux.org e o número IP da máquina, 199.249.150.4.

O DNS é, para os não iniciantes, uma das áreas mas opacas da administração de uma rede. Este howto tratará de esclarecer algumas coisas.

Este documento descreve como configurar um servidor de nomes DNS mples. Começaremos com um servidor caching only server (-- Servidor que se limita a guardar em uma cache o IPs dos nomes de máquina mais solicitados, obtendo-as de servidores externos.--) , e continuaremos com a configuração de um servidor DNS primário para um domínio.

Antes de começar, deve configurar seu sistema convenientemente, de forma que possa fazer telnet como fazia na sua máquina, efetuando satisfatoriamente toda classe de conexões de rede, especialmente telnet 127.0.0.1 entrando em sua própria máquina. Também necessita que os arquivos /etc/host.conf (ou /etc/nsswitch.conf), /etc/resolv.conf e /etc/hosts sejam corretos como ponto de partida, já que não explicarei suas funções aqui. Se não tem nenhuma destas configurações e não funciona em rede, o NET-2 HOWTO explica como fazê-lo. Leia-o.

Se está usando SLIP ou PPP necessitará que funcionem corretamente. Leia o PPP-HOWTO se não estiver assim.

Quando digo ``sua máquina'' quero dizer a máquina na qual estamos tentando configurar DNS.

Supondo que não está atrás de qualquer classe de firewalls que bloqueie petições de nomes. Se necessita uma configuração especial.

O serviço de nomes não Unix é levado a cabo por um programa, chamado named.

Este forma parte do pacote bind, que é coordenado por Paul Vixie para The Internet Software Consortium. named está incluído na maioria das distribuições de Linux e geralmente se instala como /usr/sbin/named. Se tem o arquivo named provavelmente poderá usá-lo; se não o tem, pode obter o binário num ftp de Linux, ou conseguir os últimos e mais volumosos fontes não ftp://ftp.vix.com/pub/bind.

DNS é uma base de dados cujo ambiente é a Rede. Tenha cuidado com o que põe nela. Se pôr incongruências, os demais obterão incongruências dela. Mantenha seu DNS limpo e consistente e conseguirá um bom serviço dela. Aprenda a usá-la, administrá-la, depurá-la e será outro bom administrador, salvando a rede de cair sobre suas malhas sobrecarregada por falta de manutenção.

Neste documento exponho de forma clara várias coisas que não são completamente verdade (são ao menos meias verdades). Todo isto faço com minha Simplicidade. Todas funcionarão (provavelmente ;-) se acreditar não que digo.

Aviso:

Faça uma cópia de segurança de todos os arquivos que lhe indico e se depois não funcionar, poderá voltar ao início.

2. Servidor de nomes de ``um cache''.

Um primeiro ataque à configuração DNS, muito útil para os usuários de conexões telefônicas.

Um servidor de nomes de ``um cache'' (caching only nameserver) obterá a resposta às solicitações de nome provenientes de sua rede perguntando a servidores externos, gravando a resposta para a próxima vez que a necessite.

O primeiro que necessitar este arquivo chamado /etc/named.boot. Este arquivo é lido quando se inicia o named. Por hora conterá Simplesmente:

```
; Arquivo boot de servidor de nomes de um cache:
;
directory /var/named
;
; tipo          domínio          arquivo ou maquina atual
cache           .                root.cache
primary         0.0.127.in-addr.arpa  pz/127.0.0
```

MUITO IMPORTANTE:

Em algumas versões deste documento, não conteúdo dos arquivos que aqui aparecem há um par de espaços ou tabulações antes do primeiro caracter em branco. Se supõe que estes caracteres NÃO estão não arquivo. Apague qualquer espaço inicial dos arquivos e use o deste HOWTO.

A linha directory indica ao named onde buscar os arquivos. Todos os arquivos indicados a continuação serão relativos a este diretório. /var/named é o diretório correto de acordo com o LFS, Linux File system Standard. Assim, pz é um diretório baixo /var/named, isto é, /var/named/pz.

2.1. /var/named/root.cache

Vamos descrever o arquivo chamado /var/named/root.cache nomeado não arquivo boot.named.

/var/named/root.cache deveria conter isto:

```
.      518400  NS      D.ROOT-SERVERS.NET.
.      518400  NS      E.ROOT-SERVERS.NET.
.      518400  NS      I.ROOT-SERVERS.NET.
.      518400  NS      F.ROOT-SERVERS.NET.
.      518400  NS      G.ROOT-SERVERS.NET.
.      518400  NS      A.ROOT-SERVERS.NET.
.      518400  NS      H.ROOT-SERVERS.NET.
.      518400  NS      B.ROOT-SERVERS.NET.
.      518400  NS      C.ROOT-SERVERS.NET.
;
D.ROOT-SERVERS.NET.  3600000 A      128.8.10.90
E.ROOT-SERVERS.NET.  3600000 A      192.203.230.10
I.ROOT-SERVERS.NET.  3600000 A      192.36.148.17
F.ROOT-SERVERS.NET.  3600000 A      192.5.5.241
```

G.ROOT-SERVERS.NET.	3600000	A	192.112.36.4
A.ROOT-SERVERS.NET.	3600000	A	198.41.0.4
H.ROOT-SERVERS.NET.	3600000	A	128.63.2.53
B.ROOT-SERVERS.NET.	3600000	A	128.9.0.107
C.ROOT-SERVERS.NET.	3600000	A	192.33.4.12

Recorde o que disse sobre os espaços iniciais!

Este arquivo descreve os servidores de nomes raiz não mundo. Este arquivo mudará com o passar do tempo e tem que ser mantido e atualizado com uma certa regularidade. Veja a seção de manutenção para saber como mantê-lo atualizado. Este arquivo está descrito na página man de named.

A seguinte linha de named.boot é a linha primary. Explicarei seu uso num capítulo posterior: Por hora, veja um arquivo chamado 127.0.0 não subdiretório pz:

@	IN	SOA	ns.linux.bogus. hostmaster.linux.bogus. (
			1 ; Número de Série
			28800 ; Taxa de Atualização
			7200 ; Taxa de Retentativa
			604800 ; Vencimento para secundário
			86400) ; Validação para Clientes
		NS	ns.linux.bogus.
1		PTR	localhost.

A continuação necessita o arquivo /etc/resolv.conf, que será algo similar a este:

```
search subdomínio.seu-domínio.edu su-domínio.edu
nameserver 127.0.0.1
```

A linha `search' especifica em que domínios se buscaria para qualquer nome de máquina ao qual queira conectar. A linha `nameserver' especifica a direção de seu servidor de nomes, neste caso sua própria máquina, já que é aqui que o named estará sendo executado. Se quiser uma lista de vários servidores ponha uma linha nameserver para cada um. (Nota: named nunca lê este arquivo).

Para ilustrar o que faz este arquivo:

para fulanão, fulanão.subdomínio.seu-domínio.edu se colocará primeiro, a continuação fulanão.seu-domínio.edu, e finalmente fulanão. Se um cliente tenta buscar sunSete.unc.edu, sunsite.unc.edu.subdomínio.seu-domínio.edu se põem primeiro, depois sunSete.unc.edu.seu-domínio.edu, e finalmente sunSete.unc.edu. Pode ser que não queira pôr demasiados domínios na linha search, perde-se tempo efetuando as buscas.

O exemplo supõe que pertence ao domínio subdomínio.seu-domínio.edu, sua máquina provavelmente se chame sua-maquina.subdomínio.seu-domínio.edu. A linha search não deveria conter seu TLD (Top Level Domain o Domínio de Nível Superior, `edu' neste caso). Se necessita conectar freqüentemente com máquinas de outro domínio, pode anexar esse domínio a linha search como segue:

```
search subdomínio.seu-domínio.edu seu-domínio.edu outro-domínio.com
```

e assim sucessivamente. Obviamente necessita pôr um domínio real em seu lugar. Por favor, observe a falta de pontos não final destes nomes de domínio.

O seguinte, dependendo de sua versão da biblioteca libc pode necessitar modificar /etc/nsswitch.conf ou /etc/host.conf. Se já tiver nsswitch.conf corrigiremos este, em outro caso mudaremos host.conf.

2.2. /etc/nsswitch.conf

Se trata de um extenso arquivo onde se especifica de onde obter as diferentes classes de tipos de dados, e de qual arquivo ou base de dados. Geralmente contém comentários úteis ao começo, que por certo deveria considerar ler agora. Depois busque a linha que começa por `hosts:`, deve-se ler:

```
hosts:      files dns
```

Se não há uma linha que comece por `'hosts:'` ponha-a. Isso indica que os programas devem olhar primeiro não arquivo `/etc/hosts`, e depois comprovar DNS de acordo com `resolv.conf`.

2.3. `/etc/host.conf`

Provavelmente contém varias linhas, uma delas deveria começar com `order` e teria que parecer-se ao Seguinte:

```
order hosts,bind
```

Se não há uma linha `order` tem que incluí-la. Isto lhe indica às rotinas de resolução de nomes que busquem primeiro em `/etc/hosts`, e pergunte logo ao servidor de nomes (que digo em `resolv.conf` que está em `127.0.0.1`). Estes dois últimos arquivos estão documentados na página de manual `resolv(8)` (digitando `man 8 resolv`) na maioria das distribuições Linux. Esta página do manual é de leitura obrigatória, e todos, especialmente os administradores DNS, deveriam lê-la. Faça-o agora, se disser a si mesmo `'o farei mais tarde'` então nunca o fará.

2.4. Inicialização de `named`

Depois de tudo isso, já é hora de iniciar `named`. Se está utilizando uma conexão telefônica, conecte-se primeiro. Tecle `ndc start` e pressione `return`, sem opções. Se tiver problemas tente `/usr/sbin/ndc start` no seu lugar. Agora já pode comprovar sua configuração. Se ver no arquivo de mensagens de `syslog` (geralmente chamado `/var/adm/messages`, ou no diretório `/var/log`) mensagens de inicio do `named`, (digite `tail -f /var/adm/messages`) deveria ver algo como isto:

```
Jun 30 21:50:55 roke named[2258]: starting.  named 4.9.4-REL Sun Jun 30
21:29:03 MET DST 1996
janl@roke.slip.ifi.uio.no:/var/tmp/bind/named
Jun 30 21:50:55 roke named[2258]: cache zone "" loaded (serial 0)
Jun 30 21:50:55 roke named[2258]: primary zone "0.0.127.in-addr.arpa"
loaded (serial 1)
```

Se houver qualquer mensagem de erro deverá haver algum erro. `named` determinará o arquivo que ocasiona o erro (de `named.boot` ou `root.cache`). Apague o `named` e volte a editar o arquivo.

Agora é o momento de iniciar `nslookup` para examinar seu trabalho:

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

>

Se é isso o que aparece então está funcionando. Assim espero. Em qualquer outro caso, volte atrás e edite-o todo. Cada vez que muda o arquivo `named.boot` tem que reinicializar o `named` usando o comando `ndc restart`.

Agora pode introduzir uma consulta. Tente buscar alguma máquina longe à sua.

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

Name:    pat.uio.no
Address: 129.240.2.50
```

nslookup agora solicita a named que busque a máquina pat.uio.no. Contatará com alguma das máquinas servidoras de nomes nomeadas no arquivo root.cache, e perguntará ali. Pode tardar um pouco antes de conseguir o resultado já que busca todos os domínios indicados em /etc/resolv.conf.

Se tentar de novo obterá isto:

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

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

Note a linha ``Non-authoritative answer:'' : lhe dedicaremos um pouco de tempo. Isto significa que named não sai da rede para perguntar desta vez, em seu lugar procura em seu cache e o encontra ali. Mas a informação da cache pode não estar atualizada. Então informamos deste fato (de modo um tanto eufemístico) com Non-authoritative answer:. Quando nslookup mostrou isto a segunda vez que pergunta por uma máquina, é um sinal seguro de que o named armazena a informação na cache e que está funcionando. Agora pode sair de nslookup usando o comando exit.

Se é um usuário de conexões telefônicas, (ppp, slip) por favor leia a seção sobre conexões telefônicas, há algumas advertências.

Agora já sabe como configurar um servidor de nomes de ``um cache''. Tome uma cerveja, um copo de leite ou qualquer outra coisa que prefira para comemorar.

3. Um domínio Simples.

Como configurar seu próprio domínio.

3.1. Mas primeiro algo de teoria.

Antes de começar realmente com esta seção, vou dar um pouco de teoria sobre como funciona DNS. E é bom ler porque será melhor para você.

O DNS é um sistema hierárquico. A raiz se escreve como `.' e se denomina `root'. Debaixo há certo número de Domínios de Nível Superior (Top Level Domains, TLDs), os mais conhecidos são ORG, COM, EDU e NET, mas há muitos mais.

Quando se busca uma máquina, a pergunta procede recursivamente na hierarquia começando desde cima. Se quiser localizar a direção de prep.ai.mit.edu, seu servidor de nomes deve encontrar primeiro um servidor de nomes que sirva a edu. Perguntar ao servidor no arquivo root.cache), e o servidor . proporcionará uma lista de servidores edu:

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

Começa perguntando a um servidor raiz.

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

Põe o tipo de petição (Query) a NS (Name Server records).

```
> set q=ns
```

Pergunta por edu.

```
> edu.
```

O ponto (".") final aqui é significativo, indica ao servidor que lhe pedimos um edu que está justo debaixo de ".", e isto reduz a busca um pouco.

```
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
```

isto nos diz que *.root-servers.net serve a edu., e assim podemos seguir perguntando a C. Agora queremos saber quem serve o seguinte nível o nome de domínio: mit.edu.

```
> mit.edu.
Server:  c.root-servers.net
Address:  192.33.4.12
```

```
Non-authoritative answer:
mit.edu nameserver = STRAWB.mit.edu
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu
```

```
Authoritative answers can be found from:
STRAWB.mit.edu  internet address = 18.71.0.151
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
```

steawb, w20ns e bitsy servem a mit, seleciona um e pergunta por ai.mit.edu:

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


Os nomes de máquina não são sensíveis a maiúsculas/minúsculas, mas como eu uso o mouse para copiar e colar, obtenho uma cópia tal e qual aparece nas linhas.

```
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 = WHEATIES.AI.MIT.EDU
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
```

Authoritative answers can be found from:

```
AI.MIT.EDU      nameserver = WHEATIES.AI.MIT.EDU
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
WHEATIES.AI.MIT.EDU internet address = 128.52.32.13
WHEATIES.AI.MIT.EDU internet address = 128.52.35.13
ALPHA-BITS.AI.MIT.EDU internet address = 128.52.32.5
ALPHA-BITS.AI.MIT.EDU internet address = 128.52.37.5
GRAPE-NUTS.AI.MIT.EDU internet address = 128.52.32.4
GRAPE-NUTS.AI.MIT.EDU internet address = 128.52.36.4
TRIX.AI.MIT.EDU internet address = 128.52.32.6
TRIX.AI.MIT.EDU internet address = 128.52.38.6
MUESLI.AI.MIT.EDU internet address = 128.52.32.7
MUESLI.AI.MIT.EDU internet address = 128.52.39.7
```

Então wheaties.ai.mit.edu é um servidor de nomes para ai.mit.edu:

```
> server WHEATIES.AI.MIT.EDU.
Default Server: WHEATIES.AI.MIT.EDU
Addresses: 128.52.32.13, 128.52.35.13
```

Agora mude o tipo de solicitação; encontrado o servidor de nomes pergunte tudo o que queremos saber sobre prep.ai.mit.edu.

```
> set q=any
> prep.ai.mit.edu.
Server: WHEATIES.AI.MIT.EDU
Addresses: 128.52.32.13, 128.52.35.13
```

```
prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
    inet address = 18.159.0.42, protocol = tcp
    #21 #23 #25 #79
prep.ai.mit.edu preference = 1, mail exchanger = life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = wheaties.ai.mit.edu
ai.mit.edu      nameserver = grape-nuts.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 = life.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
life.ai.mit.edu internet address = 128.52.32.80
```

alpha-bits.ai.mit.edu	internet address = 128.52.32.5
wheaties.ai.mit.edu	internet address = 128.52.35.13
wheaties.ai.mit.edu	internet address = 128.52.32.13
grape-nuts.ai.mit.edu	internet address = 128.52.36.4
grape-nuts.ai.mit.edu	internet address = 128.52.32.4
mini-wheats.ai.mit.edu	internet address = 128.52.32.11
mini-wheats.ai.mit.edu	internet address = 128.52.54.11
mintaka.lcs.mit.edu	internet address = 18.26.0.36

De esta forma começando em . irá encontrar os sucessivos servidores de nomes para o seguinte nível no nome de domínio. Se foi usado seu próprio servidor DNS em lugar de usar todos esses outros servidores, seu named, desde logo, terá armazenado no cache toda a informação que tiver encontrado na busca, e em consequência não terá que perguntar de novo durante um tempo.

Se fala muito menos sobre ele, mas um domínio importante é in-addr.arpa. Também está anexado como os domínios nome da máquina quando conhecemos sua direção IP. Uma coisa importante aqui é observar que as direções IP estão escritas em ordem inverso no domínio in-addr.arpa. Se tiver a direção da máquina 192.128.52.43, named procede como para o exemplo de prep.ai.mit.edu: Busca os servidores arpa.. Busca os servidores in-addr.arpa., os servidores 192.in-addr.arpa., os servidores 128.192.in-addr.arpa. , e os servidores 52.128.192.in-addr.arpa. e finalmente, os registros necessários para 43.52.128.192.in-addr.arpa. Inteligente? (Diga `sim'). A inversão de números pode ser confusa nos 2 primeiros anos.

Foi contada uma mentira. DNS não funciona como te digo de forma literal. Mas é bastante parecido.

3.2. Nosso próprio Domínio

Agora vamos definir nosso próprio domínio. Vamos criar o domínio linux.bogus e definir máquinas nele. Uso um nome de domínio totalmente falso para estar seguro de que não mexemos com nada de fora.

Já estamos começando esta parte com a seguinte linha em named.boot:

```
primary          0.0.127.in-addr.arpa          pz/127.0.0
```

Por favor observe a ausência de `.' ao final dos nomes de domínio em este arquivo. A primeira linha nomeia o arquivo pz/127.0.0 como definição de 0.0.127.in-addr.arpa. Já temos configurado este arquivo, nele poderemos ler:

@	IN	SOA	ns.linux.bogus. hostmaster.linux.bogus. (
			1 ; Numero de Serie
			28800 ; Taxa de Atualização
			7200 ; Taxa de Retentativa
			604800 ; Vencimento para secundário
			86400) ; Tempo de Validade para Clientes
		NS	ns.linux.bogus.
1		PTR	localhost.

Por favor observe os `.' ao final dos nomes de domínio completo em contraste com o arquivo named.boot anterior. Algumas pessoas gostam de iniciar cada arquivo com uma diretiva \$ORIGIN, mas isto é supérfluo. A origem (lugar da hierarquia DNS de onde pertence) de um arquivo de zona se especifica na coluna domínio do arquivo named.boot; neste caso é 0.0.127.in-addr.arpa.

Este ``arquivo de zona'' contém três registros de recursos (RRs): Um RR SOA, um RR NS e um RR PTR. SOA é uma abreviatura de Start Of

Authority. A '@' é uma notação especial que simboliza a origem, e como a coluna domínio para este arquivo indica 0.0.127.in-addr.arpa. A primeira linha realmente significa:

```
0.0.127.IN-ADDR.ARPA. IN      SOA ...
```

NS é o RR Name Server (Servidor de Nomes), e indica a DNS que máquina é o servidor de nomes do domínio. e finalmente o registro PTR ter valor 1 (igual a 1.0.0.127.IN-ADDR.ARPA, isto é, 127.0.0.1) que é o localhost de named.

O registro SOA é o preâmbulo de todos os arquivos de zona e deve haver um exatamente em cada arquivo de zona, como primeiro registro de todos. O registro SOA descreve zona, de onde provém (uma máquina chamada linux.bogus), quem é o responsável de seu conteúdo (hostmaster@linux.bogus), que versão do arquivo de zona é (Número de Serie, 1), e outras coisas que têm que ver com o cache e os servidores secundários DNS. Para o resto dos campos (Taxa de Atualização, Taxa de Retentativa, Vencimento para secundário e Tempo de Validade para Clientes) use os valores que aparecem aqui para maior segurança.

O registro NS nos indica quem efetua o serviço DNS para 0.0.127.in-addr.arpa, que é ns.linux.bogus. O registro PTR nos diz que 1.0.0.127.in-addr.arpa (aka 127.0.0.1) é conhecido como localhost.

Agora reiniciamos named (o comando é `ndc restart`) e usamos `nslookup` para examinar o que temos:

```
$ 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
```

assim obtemos localhost de 127.0.0.1, bem. Agora para nossa tarefa principal, o domínio linux.bogus, insere uma nova linha, primary, em named.boot:

```
primary                linux.bogus                pz/linux.bogus
```

Observe que continua a ausência de "." final no nome de domínio do arquivo named.boot.

No arquivo de zona de linux.bogus poremos alguns dados totalmente falsos (-- N do T bogus em inglês significa precisamente falso.--) :

```

;
; Arquivo de zona para linux.bogus
;
; Mínimo indispensável para ter funcionando um domínio
;
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                                199511301      ; Numero de série, fecha de hoy + n.
de série de hoy
                                28800          ; Taxa de Atualização, em segundos
                                7200           ; Taxa de Retentativa, em segundos
                                3600000        ; Vencimento para secundário, em
segundos
```

```

segundos                                86400 )           ; Tempo de Validade para Clientes, em
NS      ns.linux.bogus.
NS      ns.friend.bogus.
MX      10 mail.linux.bogus      ; Intercambiador de Correio
Primário
MX      20 mail.friend.bogus. ; Intercambiador de Correio
Secundário

localhost      A      127.0.0.1
ns              A      127.0.0.2
mail           A      127.0.0.4

```

Devemos observar duas coisas sobre os registros SOA. `ns.linux.bogus` deve ser uma máquina atual com um registro A. Não é legal ter um registro CNAME para a máquina mencionada no registro SOA. Seu nome não necessita ser `ns`, poderia ser qualquer nome válido de máquina. A continuação, em `hostmaster.linux.bogus` deverá aparecer algo como `hostmaster@linux.bogus`; isto seria um alias de email, ou uma conta de correio, de onde a(s) pessoa(s) que realizam a manutenção de DNS deveriam ler com frequência o correio. Qualquer email respectivo do domínio será mandado à direção aqui indicada. O nome não tem por que ser `hostmaster`, pode ser qualquer direção email válido, mas a direção email `hostmaster` funcionará bem.

Há um novo tipo de RR em este arquivo, o MX, o Mail eXchanger. Este indica o sistema de correio de onde mandar o correio dirigido a `alguem@linux.bogus`, podendo ser também `mail.linux.bogus` ou `mail.friend.bogus`. O número que precede a cada nome de máquina é a prioridade do RR MX. O RR com o número mais baixo (10) é aquele ao que o correio será enviado primeiro. Se este falha, pode ser mandado a outro com um número mais alto, que será gestor secundário de correio, como `mail.friend.bogus` que tem uma prioridade 20 aqui.

Reinicie `named` executando `ndc restart`. Examine os resultados com `nslookup`:

```

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

linux.bogus
    origin = linux.bogus
    mail addr = hostmaster.linux.bogus
    serial = 199511301
    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      nameserver = ns.friend.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
linux.bogus      nameserver = ns.friend.bogus
ns.linux.bogus   internet address = 127.0.0.2
mail.linux.bogus      internet address = 127.0.0.4

```

Com um exame cuidadoso poderá descobrir um erro. A linha

```

linux.bogus      preference = 10, mail exchanger =
mail.linux.bogus.linux.bogus

```

está equivocada. Deveria ser

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

Cometi o erro de forma deliberada para que aprenda com ele :-) Olhando no arquivo de zona podemos ver que a linha

```
@          MX      10 mail.linux.bogus      ; Intercambiador de
Correio Primário
```

não tem ponto. Ou tem demasiados linux.bogus. Se um nome de máquina não termina em ponto em um arquivo de zona, o origem é ignorado no seu final. Assim,

```
@          MX      10 mail.linux.bogus.      ; Intercambiador de
Correio Primário
```

ou

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

serão corretos. Eu prefiro a última forma, pois se escreve menos. Em um arquivo de zona o domínio deveria ser escrito e terminado com um ponto, ou não deve ser incluído, em cujo caso se referirá ao origem por defeito. Devo dizer que no arquivo named.boot não deveria haver pontos depois dos nomes de domínio. Não tem idéia de quantas vezes um '.' por estar ou por não estar no seu lugar falhar toda uma configuração e confundir horrorosamente a gente...

Uma vez feita esta pontualização, temos aqui o novo arquivo de zona,

com algo de informação extra também:

```
;
; Arquivo de zona para linux.bogus
;
; mínimo indispensável para fazer funcionar um domínio
;
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
de série de hoy      199511301      ; Numero de Serie, fecha de hoy + n.

      28800      ; Taxa de Atualização, em segundos
      7200      ; Taxa de Retentativa, em segundos
      604800     ; Vencimento para secundário, em segundos
      86400 )    ; Validade para Clientes, em segundos

      NS      ns      ; Direção de Internet do servidor de nomes
      NS      ns.friend.bogus.
      MX      10 mail      ; Intercambiador de Correio Primário
      MX      20 mail.friend.bogus. ; Intercambiador de Correio

Secundário

localhost      A      127.0.0.1
ns              A      127.0.0.2
mail           A      127.0.0.4
;
; Extras
;
@              TXT      "Linux.Bogus, your DNS consultants"

ns             MX      10 mail
              MX      20 mail.friend.bogus.
              HINFO     "Pentium" "Linux 1.2"
              TXT       "RMS"
```

```

richard      CNAME    ns
www          CNAME    ns

donald       A         127.0.0.3
             MX        10 mail
             MX        20 mail.friend.bogus.
             HINFO     "i486"  "Linux 1.2"
             TXT       "DEK"

mail         MX        10 mail
             MX        20 mail.friend.bogus.
             HINFO     "386sx"  "Linux 1.0.9"

ftp          A         127.0.0.5
             MX        10 mail
             MX        20 mail.friend.bogus.
             HINFO     "P6"    "Linux 1.3.59"

```

Pode que se queira ignorar os três primeiros registros tipo A (localhost, ns e mail) junto com os outros registros de seu mesmo tipo (donald, mail, e ftp), em vez de colocá-los separados ao princípio como aqui.

Há vários registros novos aqui: HINFO (Host INfOrmation), tem duas partes. A primeira parte é o hardware ou CPU da máquina, e a segunda parte corresponde ao software ou Sistema Operacional da mesma. ns tem uma CPU Pentium com Linux 1.2. O registro TXT é um texto em formato livre que pode usar para qualquer coisa que lhe interesse. CNAME (Canonical NAME) é uma forma de dar a cada máquina vários nomes. Por tanto richard e www são alias para ns. É importante observar que os registros A, MX, CNAME e SOA nunca devem fazer referência ao registro CNAME, só podem referir-se a registros A.

```

fulanito CNAME    richard          ; ;;NÃO!!!

```

Sendo correto ter

```

fulanito CNAME    ns              ; ;;SE!!!

```

Também é importante observar que CNAME não é um nome de máquina válido para direções de correio: webmaster@www.linux.bogus é uma direção email inválida dada na configuração anterior. Encontrará poucos administradores de correio por aí afora que recomendem esta regra. A forma de evitar isto é usar um registro A em seu lugar:

```

www      A         127.0.0.2

```

Paul Vixie, o principal gurú de named recomenda não usar CNAME. Por tanto considere ou não utilize-o seriamente.

Carregue a nova base de dados executando ndc reload, isto provoca a leitura de seus arquivos de novo.

```

$ nslookup
Default Server:  localhost
Address:  127.0.0.1

> ls -d linux.bogus

```

isto faria que todos os registros fossem listados.

```

[localhost]
linux.bogus.      SOA    ns.linux.bogus hostmaster.linux.bogus.
(199511301 28800 7200 604800 86400)
linux.bogus.      NS     ns.linux.bogus

```

```

linux.bogus.      NS      ns.friend.bogus
linux.bogus.      MX      10      mail.linux.bogus
linux.bogus.      MX      20      mail.friend.bogus
linux.bogus.      TXT      "Linux.Bogus, your DNS consultants"
localhost         A       127.0.0.1
mail              A       127.0.0.4
mail              MX      10      mail.linux.bogus
mail              MX      20      mail.friend.bogus
mail              HINFO    386sx      Linux 1.0.9
donald            A       127.0.0.3
donald            MX      10      mail.linux.bogus
donald            MX      20      mail.friend.bogus
donald            HINFO    i486      Linux 1.2
donald            TXT      "DEK"
www               CNAME    ns.linux.bogus
richard           CNAME    ns.linux.bogus
ftp               A       127.0.0.5
ftp               MX      10      mail.linux.bogus
ftp               MX      20      mail.friend.bogus
ftp               HINFO    P6        Linux 1.3.59
ns                A       127.0.0.2
ns                MX      10      mail.linux.bogus
ns                MX      20      mail.friend.bogus
ns                HINFO    Pentium    Linux 1.2
ns                TXT      "RMS"
linux.bogus.      SOA      ns.linux.bogus hostmaster.linux.bogus. (199511301
28800 7200 604800 86400)

```

isto está bem. Comprovemos o que diz para www só:

```

> set q=any
> www.linux.bogus.
Server: localhost
Address: 127.0.0.1

```

```
www.linux.bogus canonical name = ns.linux.bogus
```

www.linux.bogus é ns.linux.bogus

```

linux.bogus      nameserver = ns.linux.bogus
linux.bogus      nameserver = ns.friend.bogus
ns.linux.bogus   internet address = 127.0.0.2

```

e ns.linux.bogus tem a direção 127.0.0.2. Parece correto também.

3.3. Relaxemos

Desde logo, este domínio é falso, e como tal são todas suas direções. Para um exemplo real de domínio veja a seguinte seção.

4. Um exemplo de domínio real

De onde descreveremos alguns arquivos de zona reais.

Os usuários têm sugerido que se incluía um exemplo real de domínio que esteja em funcionamento como explicação das diferenças entre um domínio em funcionamento e o exemplo falso que não era de todo muito claro.

Uma coisa sobre este exemplo: NÃO o use em seu servidor de nomes!. Use-o só como leitura de referência. Se quiser experimentar, faça-o com o exemplo falso. Eu uso este exemplo com permissão de David Bullock e LAND-5. Estos arquivos eram os usados em 24 de Septiembre de 1996, e poderiam diferir dos que encontre se

perguntar agora ao servidor de nomes LAND-5. Também tenha em mente eliminar os espaços iniciais ;-).

4.1. /etc/named.boot (ou /var/named/named.boot)

Aqui encontramos as linhas primary para as zonas que necessitamos: a rede 127.0.0.0 e também a sub-rede 206.6.177 de LAND-5. Uma linha primary para a zona de redirecionamento (forward) land-5.com de land-5. Observe também que em lugar de situar os arquivos em um diretório chamado pz, como faço neste HOWTO, ele os situa em um diretório chamado zone.

```
; Arquivo de inicialização para o servidor de nomes LAND-5
;
directory /var/named
;
; tipo          domínio          arquivo ou máquina origem
cache          .                  root.cache
primary        0.0.127.in-addr.arpa  zone/127.0.0
primary        177.6.206.in-addr.arpa zone/206.6.177
primary        land-5.com           zone/land-5.com
```

4.2. /var/named/root.cache

Tenha em mente que este arquivo varia com muita frequência, e que o listado aqui é velho. Melhor utilizar o criado agora.

```
; <<>> DiG 2.1 <<>>
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr rd ra; Ques: 1, Ans: 9, Auth: 0, Addit: 9
;; QUESTIONS:
;;      ., type = NS, class = IN

;; ANSWERS:
.      518357 NS      H.ROOT-SERVERS.NET.
.      518357 NS      B.ROOT-SERVERS.NET.
.      518357 NS      C.ROOT-SERVERS.NET.
.      518357 NS      D.ROOT-SERVERS.NET.
.      518357 NS      E.ROOT-SERVERS.NET.
.      518357 NS      I.ROOT-SERVERS.NET.
.      518357 NS      F.ROOT-SERVERS.NET.
.      518357 NS      G.ROOT-SERVERS.NET.
.      518357 NS      A.ROOT-SERVERS.NET.

;; ADDITIONAL RECORDS:
H.ROOT-SERVERS.NET.      165593 A      128.63.2.53
B.ROOT-SERVERS.NET.      165593 A      128.9.0.107
C.ROOT-SERVERS.NET.      222766 A      192.33.4.12
D.ROOT-SERVERS.NET.      165593 A      128.8.10.90
E.ROOT-SERVERS.NET.      165593 A      192.203.230.10
I.ROOT-SERVERS.NET.      165593 A      192.36.148.17
F.ROOT-SERVERS.NET.      299616 A      192.5.5.241
G.ROOT-SERVERS.NET.      165593 A      192.112.36.4
A.ROOT-SERVERS.NET.      165593 A      198.41.0.4

;; Total query time: 250 msec
;; FROM: land-5 to SERVER: default -- 127.0.0.1
;; WHEN: Fri Sep 20 10:11:22 1996
;; MSG SEZE sent: 17 rcvd: 312
```

4.3. /var/named/zone/127.0.0

O básico, ou registro obrigatório SOA, é o registro que mapea

127.0.0.1 a localhost. Se requerem ambos. Não deveria haver mais nenhum neste arquivo. Provavelmente nunca se necessitará atualizá-lo, salvo se mudar seu servidor de nomes ou a direção dele

hostmaster.

```

@           IN      SOA      land-5.com. root.land-5.com. (
                                199609203   ; Numero de Serie
                                28800       ; Taxa de Atualização
                                7200        ; Taxa de Retentativa
                                604800      ; Vencimento para secundário
                                86400)      ; Validade para clientes
                                NS          land-5.com.

1           PTR      localhost.

```

4.4. /var/named/zone/land-5.com

Aqui vemos o registro SOA e os registros NS necessários. Podemos observar que dispõe de um servidor de nomes secundário ns2.pSe.net. isto é como deve ser, tenha sempre um servidor secundário de segurança. Também podemos ver que tem uma máquina principal chamada land-5 que se encarga de todos os diferentes serviços, e que foi feita usando CNAME (uma alternativa ao uso dos registros A).

Como se pode ver no registro SOA, o origem do arquivo de zona é land-5.com, a pessoa de contato é root@land-5.com. hostmaster é outro uso freqüente para a pessoa de contato. O número de série o formato habitual yyyymmdd com o número de série de hoy añadido; nesta é provavelmente a sexta versão do arquivo de zona do de 20 de Setembro de 1996. Recorde que o número de série deve incrementar-se vagarosamente, aqui há só um dígito para as séries de hoje, assim que depois de 9 edições terá que esperar até a manhã, antes de poder editar o arquivo de novo. Considere o uso de dois dígitos.

```

@           IN      SOA      land-5.com. root.land-5.com. (
série de hoy                                199609206   ; Numero de Serie, fecha de hoy + numero de
                                           10800         ; Taxa de Atualização, em segundos
                                           7200          ; Taxa de Retentativa, em segundos
                                           10800        ; Vencimento para secundário, em segundos
                                           86400 )       ; Validade para Clientes, em segundos
                                           NS          land-5.com.
                                           NS          ns2.pSe.net.
                                           MX          10 land-5.com. ; Intercambiador Primário de Correio

localhost   A        127.0.0.1

router      A        206.6.177.1

land-5.com. A        206.6.177.2
ns          CNAME    land-5.com.
ftp         CNAME    land-5.com.
www         CNAME    land-5.com.
mail        CNAME    land-5.com.
news        CNAME    land-5.com.

funn        A        206.6.177.3
ilusions    CNAME    funn.land-5.com.
@           TXT      "LAND-5 Corporation"

;
;   Estações de Trabalho
;

```

```

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
; {Muitas definições repetitivas borradas}
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

```

Outra coisa a ter em conta é que as estações de trabalho não têm nomes próprios, senão um prefixo seguido pelas duas últimas porções dos números IP. Usar tal convenção pode simplificar a manutenção significativamente, mas pode se tornar um pouco impessoal.

4.5. /var/named/zone/206.6.177

Comentarei este arquivo depois.

```

@              IN      SOA      land-5.com. root.land-5.com. (
                                199609206 ; Numero de Serie
                                28800      ; Taxa de Atualização
                                7200       ; Taxa de Retentativa
                                604800     ; Vencimento para secundário
                                86400)     ; Validade para Clientes
                                NS        land-5.com.
                                NS        ns2.pSe.net.
;
;      Servidores
;
1      PTR      router.land-5.com.
2      PTR      land-5.com.
3      PTR      funn.land-5.com.
;
;      Estações de Trabalho
;
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.
; {Eliminadas muitas definições repetitivas}
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.

```

A zona de resolução inversa é a parte da configuração que parece criar mais dores de cabeça. Se usa para encontrar o nome da

máquina a partir de sua direção IP. Exemplo: suponha que está em um servidor irc e aceita conexões de clientes irc. O servidor irc é norueguês e só quer aceitar conexões de clientes da Noruega e outros países escandinavos. Quando se produz uma conexão de um cliente, a biblioteca C é capaz de indicar o número IP da máquina conectada porque o número IP do cliente está contido em todos os pacotes que se passam através da rede. Agora pode chamar a uma função chamada `gethostbyaddr` que busca o nome da máquina dada sua direção IP.

`gethostbyaddr` interrogará a um servidor DNS o qual efetuará uma busca DNS para a máquina. Supondo que a conexão cliente venha de `ws_177200.land-5.com`, a direção IP que a biblioteca C proporção ao servidor irc será `206.6.177.200`. Para encontrar o nome da máquina necessitamos encontrar `200.177.6.206.in-addr.arpa`. O servidor DNS primeiro encontra os servidores `arpa.`, depois os servidores `in-addr.arpa.`, a continuação segue por `206`, `6` e ao final busca o servidor para a zona `177.6.206.in-addr.arpa` em `land-5`. Aqui obterá finalmente que para `200.177.6.206.in-addr.arpa` teremos um registro `'PTR ws_177200.land-5.com'`, que significa que o nome que vai com `206.6.177.200` é `ws_177200.land-5.com`. Com a explicação de como buscar `prep.ai.mit.edu`, isto é ligeiramente fictício.

Voltando ao exemplo do servidor irc. O servidor irc só aceita conexões dos países escandinavos, `osea`, `*.no`, `*.se`, e `*.dk`; o nome `ws_177200.land-5.com` claramente não se ajusta a nenhum deles, e o servidor negará a conexão. Se não tivesse havido resolução inversa de `206.2.177.200` mediante a zona `in-addr.arpa` o servidor estaria sendo incapaz de encontrar o nome e haveria tido que comparar `206.2.177.200` com `*.no`, `*.se` e `*.dk`, é dizer, cifras com nomes, nenhuma das quais concordaria.

Algumas pessoas lhe dirão que a resolução inversa só é importante para os servidores, o que não tem importância. Não é assim; muitos servidores de `ftp`, `news`, `irc` e incluso alguns servidores `http` (`WWW`) NÃO aceitarão conexões de máquinas das quais não são capazes de resolver o nome. Por tanto o mapeamento inverso de máquinas é de feito obrigatório.

5. Manutenção

Mantendo em funcionamento.

Há uma tarefa de manutenção que tem que se realizar com `named`, depois de pô-lo em funcionamento. Esta tarefa é manter o arquivo `root.cache` atualizado. A forma mais fácil é usar `dig`, primeiro execute `dig` sen argumentos, conseguirá `root.cache` de acordo com seu próprio servidor. Então pergunte a algum dos servidores raiz listados com

```
dig @rootserver
```

Poderá observar que a saída se parece muito com o arquivo `root.cache` exceto por um par de números extras. Esses números não ocasionam problemas. Guarde-o em um arquivo

```
dig @rootserver . ns > root.cache.new
```

e substitua o antigo `root.cache` com ele.

Lembre-se de reiniciar o `named` depois de substituir o arquivo `cache`.

Ao Longyear me enviou este script que pode ser executado automaticamente para atualizar `root.cache`, instale uma entrada no `crontab` para executá-lo uma vez ao mês. O script supõe

que trabalha com correio e que o alias de mail hostmaster está definido. Deve editá-lo para ajustá-lo a sua configuração.

```
#!/bin/sh
#
# Atualização do cache do servidor de nomes uma vez ao mês.
# isto é executado automaticamente mediante uma entrada de cron
#
(
  echo "To: hostmaster <hostmaster>"
  echo "From: system <root>"
  echo "Subject: Atualização automática do arquivo named.boot"
  echo

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

  dig @rs.internic.net . ns >root.cache.new

  echo "O arquivo named.boot está sendo atualizado para conter a
  seguinte informação:"
  echo
  cat root.cache.new

  chown root.root root.cache.new
  chmod 444 root.cache.new
  rm -f root.cache.old
  mv root.cache root.cache.old
  mv root.cache.new root.cache
  ndc restart
  echo
  echo "O servidor de nomes está sendo reinicializado a fim de assegurar que a
  atualização é completa."
  echo "O anterior arquivo root.cache foi renomeado para
  /var/named/root.cache.old."
  ) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Alguns de vocês pode ter observado que o arquivo root.cache está também disponível mediante ftp em Internic. Por favor NÃO utilize ftp para atualizar root.cache, o método anterior é muito mais amistoso com a rede.

6. Configuração de Conexões Automáticas via telefone.

Esta seção explica como se dispõe as coisas para automatizá-lo todo. Meu método pode não se adaptar completamente ao seu, mas pode obter idéias de algumas das coisas que faço. Também, uso ppp para marcar, ainda que muita gente usa slip o cslip e por tanto quase toda sua configuração pode ser distinta da minha. Mas o programa de slip dip deveria poder fazer muitas das coisas que eu faço.

Normalmente, quando não estou conectado à rede tenho um arquivo resolv.conf que simplesmente contém a linha

```
domain uio.no
```

Isso me assegura que não tenho que esperar a que a biblioteca de resolução de nomes do sistema tente conectar com um servidor de nomes que não pode ajudar-me. Mas quando me conecto quero iniciar meu named e ter um resolv.conf parecido aos descritos anteriormente. Consigo isto tendo dois arquivos resolv.conf chamados resolv.conf.local e resolv.conf.connected. O último se parece ao resolv.conf descrito anteriormente neste documento.

Para conectar-me automaticamente na rede executo um script chamado ppp-on:

```
#!/bin/sh
echo chamando...
pppd
```

pppd tem um arquivo chamado options que indica as características da conexão. Uma vez que minha conexão ppp está ativa pppd chama a um script chamado ip-up (este está descrito na página pppd (8) de man). Eis aqui uma parte do script:

```
#!/bin/sh
interface="$1"
device="$2"
speed="$3"
myip="$4"
upip="$5"
...
cp -v /etc/resolv.conf.connected /etc/resolv.conf
...
/usr/sbin/named
```

é dizer, inicializo o named desde aqui. Quando se corta a conexão ppp, pppd executa um script chamado ip-down:

```
#!/bin/sh
cp /etc/resolv.conf.local /etc/resolv.conf
read namedpid < /var/run/named.pid
kill $namedpid
```

Assim configuramos as coisas de uma forma quando estamos conectados e as desconfiguramos quando nos desconectamos.

Alguns programas, irc e talk me vem à mente, fazem algumas suposições, e para que em irc o comportamento das capacidades dcc, e talk funcionem bem tem que modificar seu arquivo hosts. Eu tenho inserido em meu script ip-up o seguinte:

```
cp /etc/hosts.ppp /etc/hosts
echo $myip      roke >>/etc/hosts
```

hosts.ppp simplesmente contém

```
127.0.0.1      localhost
```

e echo insere a direção IP que é necessária para meu nome de host (roke). Você deverá usar em seu lugar o nome de sua máquina. Este nome se pode saber com o comando hostname.

Provavelmente não seja inteligente executar named quando não está conectado à rede, isto é porque named tentará enviar solicitações à rede e isso consome tempo, e você terá que esperar este tempo cada vez que algum programa tente resolver um nome. Se está usando conexões telefônicas deveria iniciar named quando se conecta e matá-lo quando se desconecta.

Algumas pessoas gostam de usar a diretiva forwarders para conexões de baixa velocidade. Se seu provedor de Internet tem servidores DNS em 1.2.3.4 e 1.2.3.5 pode inserir a linha

```
forwarders 1.2.3.4 1.2.3.5
```

no arquivo named.boot. Deixe também vazio o arquivo root.cache. isto diminuirá o tráfego IP que origina em sua máquina. isto é especialmente importante pois se paga por cada byte que circula pelo cabo.

7. Perguntas de Uso Freqüente (FAQ)

Nesta seção incluo algumas das perguntas mais freqüentes realizadas relativas a DNS e este HOWTO. E as repostas :-) Por favor, leia esta seção antes de enviar-me correio eletrônico.

7.1. Como uso DNS desde dentro de um firewall?

Um quantas pistas: `forwarders', `slave', e dar uma olhada na literatura que há ao final deste HOWTO.

7.2. Como faço que DNS rote entre as direções disponíveis para um serviço, por exemplo para `www.sempre.ocupado` para obter balanço de carga ou similar?

Faça vários registros A para `www.sempre.ocupado` e use bind 4.9.3 ou posterior. bind fará uma rotação tipo round-robin das repostas. isto não funcionará com versões anteriores de bind.

7.3. Quero configurar DNS em uma intranet (fechada) que faço?

Elimine o arquivo de cache e faça os arquivos de zona. Isso também significa que nunca terá que atualizar o arquivo de cache.

7.4. Meu sistema não tem o programa `ndc`. Que faço?

O bind instalado em seu sistema é velho e de alguma forma obsoleto. Se a segurança é importante para você: atualize bind imediatamente. Em lugar de executar `ndc start` execute `named`; `ndc reload` será `named.reload` e `ndc restart` será `named.restart`. Esses programas provavelmente estarão em `/usr/sbin`.

7.5. Como configuro um servidor de nomes secundário?

Se o servidor primário tem a direção `127.0.0.1`, ponha a seguinte linha no arquivo `named.boot` de seu secundário:

```
secondary      linux.bogus          127.0.0.1      sz/linux.bogus
```

7.6. Quero que bind se execute quando me desconecto da rede.

Recebi este mail de Ian Clark, `ic@deakin.edu.au` onde explica a forma de fazer isto:

```
`Executo named na máquina que tem masquerading aqui. Tenho dois
arquivos root.cache, um chamado root.cache.real que contém o
servidor de nomes raiz real e o outro chamado root.cache.falso que
contém...
```

```
    ; root.cache.falso
    ; este arquivo não contém informação
```

Quando deixo de estar conectado copio o arquivo `root.cache.falso` em `root.cache` e reinicio `named`.

Quando me conecto copio `root.cache.real` em `root.cache` e reinicio `named`.

isto se faz desde `ip-down` & `ip-up` respectivamente.

A primeira vez que efetuo uma consulta off line sobre um nome de

domínio do qual named não tem detalhes, este põe uma entrada como esta em messages...

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

com a qual posso conviver sem problemas.

isto certamente parece funcionar-me. Posso usar o servidor de nomes para máquinas locais quando não estou conectado sem o retardo com nomes de domínio externos, e quando estou conectado, funciona de forma normal com domínios externos.'

7.7. Há alguma forma de controlar o tamanho do cache? Onde armazena seu cache o servidor de nomes?

O cache se armazena em memória completamente. Não se grava em disco em nenhum momento. Cada vez que mata o named se perde o cache. O cache não é controlável de nenhuma forma, named o manipula de acordo com umas regras simples. Não pode controlar nem o cache nem seu tamanho de nenhuma forma nem por nenhum motivo. Se quer, pode mudar isto trocando os fontes de named, o que não é recomendável.

7.8. Salvar named ou cache entre reinícios? Posso guardá-lo?

Não, named não salva o cache quando morre. Isto significa que o cache se deve reconstruir de novo cada vez que se mate e reinicie o named. Não há forma de fazer com que named salve o cache em um arquivo. Se quiser, pode mudar isto alterando os fontes de named, o que não é recomendável.

8. Como ser um grande administrador DNS.

Documentação e ferramentas.

Existe Documentação Real. Em linha e impressa. Se reque a leitura desta documentação para seguir os passos de pequeno a grande administrador DNS. Em formato impresso o livro standard é DNS and BIND de C. Liu e P. Albitz de O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. Leia-o, é excelente. Há também uma seção sobre DNS em TCP/IP Network Administration, de Craig Hunt de O'Reilly..., ISBN 0-937175-82-X. Outros livros necessários para um Bom Administrador é DNS (ou bom para qualquer coisa da matéria) é Zen and the Art of Motorcycle Maintenance de Robert M. PriSeg :-). Disponível com ISBN 0688052304 e outros.

Pode encontrar material em linha em <http://www.dns.net/dnsrd/>, <http://www.vix.com/isc/bind/>; uma FAQ, um manual de referencia (BOG; Bind Operations Guide) assim como papers e definição de protocolos e diversos retoques o hacks de DNS (estes são a maioria, se não todas as referencias mencionadas acima, estão também contidas na distribuição de bind). Não li a maioria, mas tampouco sou um grande administrador DNS. Arnt Gulbrandsen, por outra parte leu o BOG e está extasiado com ele :-). O grupo de notícias comp.protocols.tcp-ip.domains é sobre DNS.

Nota: A tradução deste documento foi feita sem objetivos financeiros nem morais. O motivo da transcrição do howto foi amenizar a dificuldade que os novos usuários têm enfrentado com o material em inglês. Qualquer sugestão, crítica ou correção da tradução feita por Ivan Luis Seibel, favor enviar para seibel@infshr.unijui.tche.br. Obrigado.