Centro Federal de Educação Tecnológica
Unidade São José
Área de Telecomunicações
Odilson Tadeu Valle

Material Didático para o Curso de Gerência de Redes Aplicado à Intelbrás

Centro Federal de Educação Tecnológica
Unidade São José
Área de Telecomunicações
Odilson Tadeu Valle

Sumário

1 Introdução 5
1 LVM - Logical Volume Manager 6
1.1 Introdução 6
1.2 Implantando LVM 8
1.3 Aumentando o tamanho de partições LVM 9
2 Gerência de usuários e grupos 10
2.1 Introdução 10
2.2 Criação de conta 10
2.3 Parâmetros das Contas 11
2.4 Alterando parâmetros das contas 12
3 Cotas em disco para usuários e grupos 13
3.1 Introdução 13
3.2 Implementação 13
3.3 Estabelecendo cotas para vários usuários e/ou grupos 15
3.4 Verificando cotas de usuários 15
4 Agendamento de tarefas com Crontab 16
4.1 Introdução 16
4.2 Uso do Crontab 16
5 Servidor de backups Amanda 18
5.1 Introdução 18
5.2 Configuração do servidor Amanda 19
5.2.1 amanda.conf 19
5.2.2 disklist 20
5.3 Configurando o cliente 21
5.4 Backups com o Amanda 22
5.5 Restaurando os backups com Amanda 22
5.6 Comandos Extras do Amanda 23
6 Configuração da interface de rede 23
6.1 Introdução 23
6.2 Configuração 23
6.2.1 Configuração do ifcfg-ethN 24
6.2.2 Configuração do network 24
6.2.3 Configuração do resolv.conf 24
6.3 Apelidos de ip 25
7 Roteadores e sub-redes 25
7.1 Introdução 25
7.2 Configurando o roteador 26
7.3 Configurando sub-redes 27
7.4 Caso de estudo 28

7.4.1 Roteadores 28
7.4.2 Configuração do Cliente 28
7.4.3 Testes 29
8 NAT - Network Address Translator 29
9 Servidor DNS - Domain Name System com Bind 30
9.1 Introdução 30
9.2 Configuração de um servidor DNS 31
9.2.1 Caso de estudo 31
9.2.2 Testes 33
10 Servidor de páginas Apache 33
10.1 Introdução 33
10.2 Instalação e configuração 34
10.3 Domínios virtuais 34
10.4 Páginas de Usuários 35
11 Servidor de correio eletrônico Postfix 36
11.1 Introdução 36
11.2 Instalação e configuração 37
11.3 Testes 37
12 Servidor cache/proxy Squid 38
12.1 Introdução 38
12.2 Instalação e configuração 39
12.3 Testes 39
13 Servidor SMB, Server Message Block, Samba 41
13.1 Introdução 41
13.2 Instalação e configuração 42
13.3 Testes 43
14 Servidor NFS Network File System 44
14.1 Introdução 44
14.2 Instalação e configuração 44
14.3 Testes 45
15 Servidor DHCP Dynamic Host Configuration Protocol 46
15.1 Introdução 46
15.2 O protocolo DHCP 46
15.3 Instalação e configuração 50
15.4 Testes 51
16 Servidor FTP File Transfer Protocol 51
16.1 Introdução 51
16.2 Instalação e configuração 52
16.3 Testes 52
17 Servidor SSH Secure Shell com OpenSSH 53
17.1 Introdução 53
17.2 Instalação e configuração 53
17.3 Testes 54
18 Webmin 54

18.1 Introdução.....54
18.2 Instalação e configuração.....55
19 Referências bibliográficas.....57

Centro Federal de Educação Tecnológica
Unidade São José
Área de Telecomunicações
Odilson Tadeu Valle

1 Introdução

Como princípio o gerenciamento de redes consiste em prover serviços de rede aos usuários da maneira transparentee fácil aos mesmos.

Normalmente o usuário procura, além dos sistemas próprios da empresa, os seguintes serviços: correio eletrônico, navegação na internet, servidor para hospedar as páginas de seus projetos e um lugar seguro para guardar seus dados e documentos.

Por outro lado, do ponto de vista do administrador, para prover estes serviços de maneira segura são necessário uma série de outros serviços que, a princípio, não interessa ao usuário e em muitos casos "atrapalha" o mesmo.

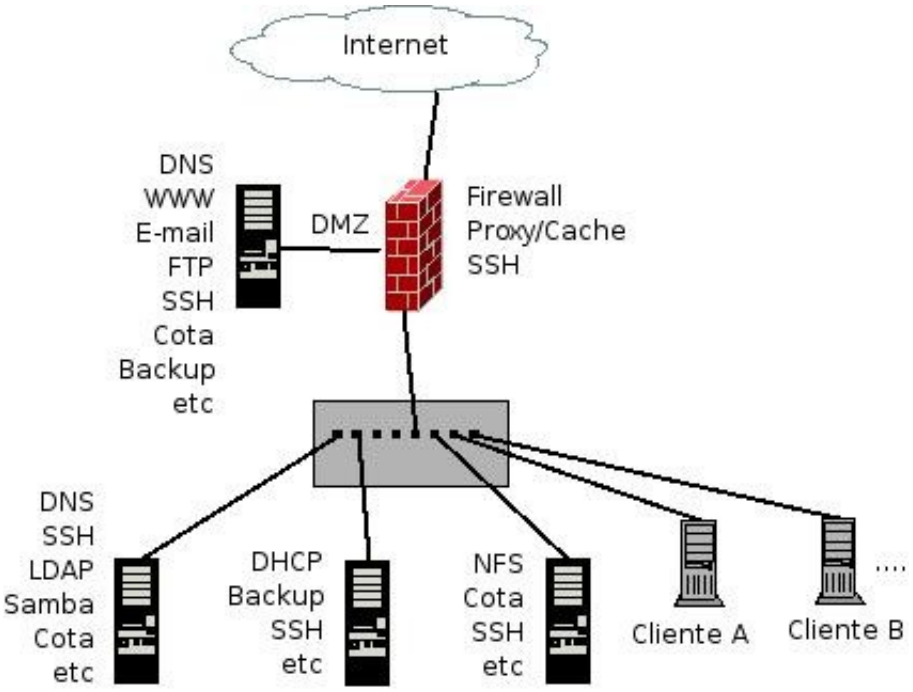


Ilustração 1: Serviços de rede

Na Ilustração 1 vemos um diagrama com os serviços básicos de rede que um administrador de sistemas deve ofertar. Devemos observar que os serviços podem ou não estar agrupados na mesma máquina.

Nos capítulos abaixo vamos detalhar uma série de serviços que podemos fornecer aos usuários ou que são necessários à boa administração do sistema.

Nosso objetivo é prover uma instalação básica de todos os serviços para o conhecimento básico teórico/prático dos mesmos. Em caso de necessidades de configurações "avançadas" nos serviços, devemos consultar as referências

bibliográficas.

1 *LVM - Logical Volume Manager*

1.1 Introdução

O LVM tem por objetivo facilitar a administração do espaço em disco e facilitar a ampliação de partições com a inserção de novos discos nos servidores. Como principais características podemos citar:

- O LVM permite o aumento ou diminuição do tamanho de partições sem a reformatação e a reinicialização da máquina.
- Em partições XFS e ReiserFS não é necessário sequer o desmonte da partição para alterar o tamanho da mesma. Ou seja, pode-se alterar o tamanho de uma partição com o sistema *on-line*.
- O melhor momento de configuração do LVM é na instalação do servidor. Deve-se reservar uma área mínima para o / (raiz), já que somente o / e o /boot não podem ser montados em partições LVM, e deixar todo o espaço restante do disco como LVM, Ilustração 2.

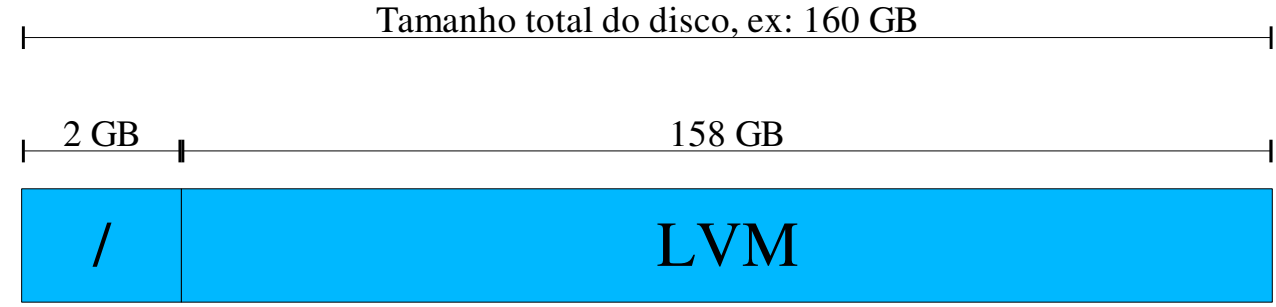


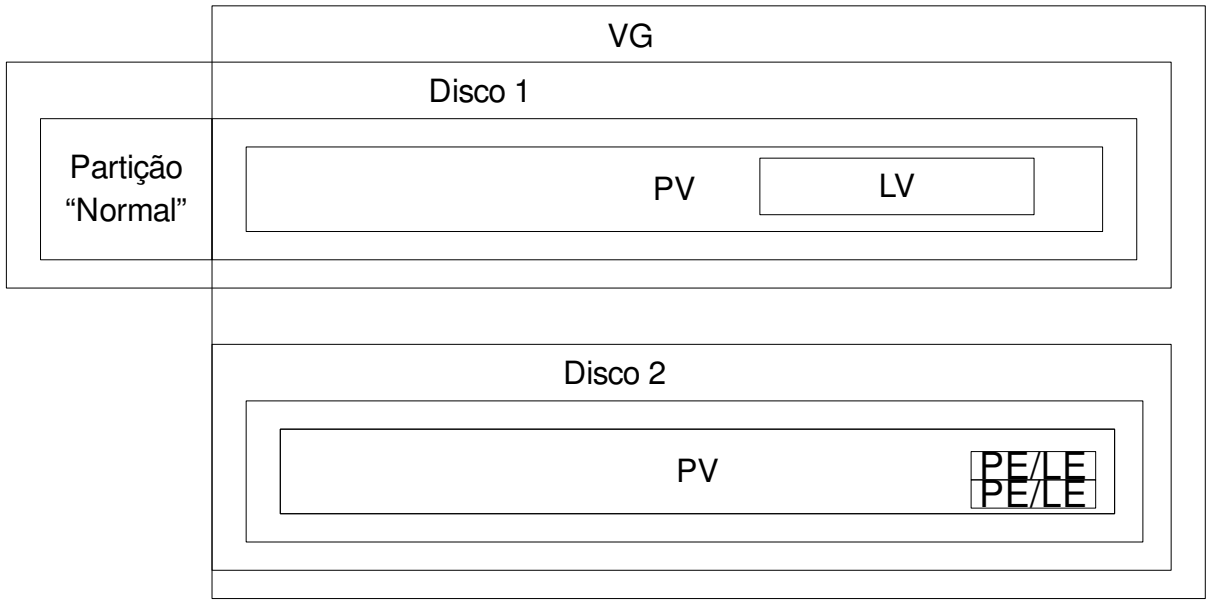
Ilustração 2: Particionamento com LVM

O Gerenciador de Volumes Lógicos consiste em uma camada adicional entre os dispositivos físicos e a interface de E/S no kernel para fornecer uma visão lógica no armazenamento.

Ao contrário do método tradicional de particionamento, a implementação **LVM** cria um grande disco virtual, que pode inclusive ter mais de um dispositivo de armazenamento, e *divide* em *partições virtuais*. A grande vantagem é permitir o redimensionamento das áreas de modo dinâmico, ou seja, com o sistema operacional sendo utilizado.

A grande desvantagem é que por ser um único disco virtual, a recuperação de dados em uma eventual pane no sistema de armazenamento é bastante prejudicada.

Tecnicamente o LVM é montado/composto conforme descrito abaixo, Ilustração 3.



PV Physical Volume — VG Volume Group
PE/LE Physical/Logical extent — LV Logical Volume

Ilustração 3: Como é fisicamente o LVM

PV (Physical Volume) - Os volumes físicos são as partições de discos alocadas para o LVM. No Linux é necessário criar a partição e alterar o tipo para "Linux LVM", tipo 8e do fdisk, para que ela possa ser utilizada no LVM.

VG (Volume Group) - Um conjunto de PV podem ser necessários para criar filesystems maiores que a limitação física de um disco rígido. Esses PV são agrupados em um VG.

PE (Physical Extent) - Quando um PV é inserido em um VG o LVM o divide em várias partes de igual tamanho e essas partes são associadas a uma LE (Logical Extent), o menor valor de alocação dentro de um VG (do ponto de vista do LVM). No AIX são conhecidos como PP (Physical Partition) e LP (Logical Partition), respectivamente.

LV (Logical Volume) - Esse elemento é uma área de alocação das LE, na qual criamos o filesystem. Ao criarmos um volume lógico, recebemos um device para referenciarmos, ao criar ou manipular, o sistema de arquivos. O nome do device é /dev/NOME_DO_VG/NOME_DO_LV.

VGDA (Volume Group Descriptor Area) - Numa analogia mais grosseira, essa área é uma tabela de alocação do VG. Nela há todos os dados do VG. É dividida em quatro partes básicas: descritor de PV, descritor de VG, descritor de LV e vários descritores de PE e LE. Os backup automáticos da VGDA são guardados em /etc/lvm-conf/.

1.2 Implantando LVM

Se o LVM não foi instalado juntamente com o sistema podemos instalar/configurar o mesmo posteriormente. Como primeira ação devemos instalar os pacotes necessários para gerenciamento e criação de partições LVM. Para isto usamos o comando:

```
urpmi lvm
```

No exemplo abaixo vamos criar um grupo de volumes de nome vg, numa partição já existente na máquina. Primeiro "inicialize" o LVM com o comando:

```
vgsacn
```

Crie o volume físico com o comando:

```
pvcreate /dev/hdaX (X = número da partição)
```

Inclua o volume físico no volume lógico com o comando:

```
vgcreate vg /dev/hdaX
```

Atualize o LVM com o comando:

```
vgscan
```

Ative o volume lógico com o comando

```
vgchange -a y
```

Verifique a criação com o comando

```
vgdisplay -v vg
```

Dentro do grupo de volumes crie, por exemplo, dois volumes lógicos lv1 de 300 MB e lv2 de 500 MB com os comandos:

```
lvcreate -L 300M -n lv1 vg

lvcreate -L 500M -n lv2 vg
```

Ative os volumes lógicos com os comandos:

```
lvchange -a y /dev/vg/lv1

lvchange -a y /dev/vg/lv2
```

Formate as "partições" lógicas criadas com os comandos abaixo, ou de acordo com o sistema de arquivos usado.

```
mkfs.xfs /dev/vg/lv1
```

```
mkfs.xfs /dev/vg/lv2
```

Crie dois diretórios, onde serão montadas as partições, com o comando:

```
mkdir /dados /backup
```

Monte as partições com os comandos:

```
mount /dev/vg/lv1 /dados
```

```
mount /dev/vg/lv2 /backup/
```

Verifique as partições montadas com o comando

```
df
```

As partições já estão disponíveis para uso. Podem ser copiados arquivos e diretórios. Se for necessário pode-se aumentar o tamanho da partições.

1.3 Aumentando o tamanho de partições LVM

Para aumentar o tamanho de uma partição é preciso primeiro ver seu tamanho e a disponibilidade de espaço no volume group.

Para verificar quais são os volumes lógicos utiliza-se o comando:

```
lvdisplay
```

O espaço disponível pode ser visto com o comando:

```
vgdisplay
```

No exemplo abaixo aumenta-se o volume lógico lv em 2 GB:

```
lvextend -L +2G /dev/vg/lv
```

Pronto a partição já foi aumentada. O único problema é que o sistema de arquivos ainda não sabe disso. Execute o próximo passo de acordo com o seu sistema de arquivos:

- Sistemas de arquivos XFS (pode ser executado com o sistema *on-line*, ou seja, com a partição montada):

```
xfs_growfs /home
```

- Sistemas de arquivos ReiserFS (pode ser executado com o sistema on-line, ou seja, com a partição montada):

```
resize_reiserfs -f /dev/rootvg/home
```

- Sistemas de arquivos EXT3:

```
umount /home

resize2fs /dev/rootvg/home

mount /dev/rootvg/home /home
```

2 Gerência de usuários e grupos

2.1 Introdução

Um usuário Linux é uma entidade que possui uma identificação no sistema onde os principais parâmetros são: *login*, senha, e número de identificação. Estas informações permitem ao Linux controlar como o acesso é garantido aos usuários e o que eles podem fazer depois de obter a permissão de acesso.

Um grupo é um conjunto de usuários. Cada grupo também possui identificação única no sistema, um nome e um número. O administradores de sistemas normalmente fazem controle de acesso por meio dos grupos.

2.2 Criação de conta

Para criação de uma conta de usuário usa-se o comando:

```
adduser login
```

Podemos ainda “sofisticar” a criação de contas com algumas *flags*, as principais são:

- -d caminho – Podemos informar qual será o diretório home do usuário.
- -g grupo – Podemos definir o grupo primário a que o usuário pertencerá. Se este não for informado o linux cria um grupo com o mesmo nome de usuário.
- -G grupo1,grupo2 – Define o(s) grupo(s) suplementar(s) ao(s) qual(is) o usuário pertencerá.
- -c comentário – Normalmente o nome completo do usuário.
- -s shell – Esta opção é interessante se desejamos, por exemplo, que um usuário não acesse a máquina com sua conta. Para isto informamos /bin/false.

Em seguida **definimos a senha** com o comando:

```
passwd login
```

2.3 Parâmetros das Contas

As contas de usuários ficam armazenadas nos seguintes arquivos passwd, group e shadow.

O arquivo **/etc/passwd** define todos os usuários cadastrados no sistema, segundo o molde:

login:x:503:500:comentário:/home/login:/bin/bash

A descrição dos campos são:

o login do usuário.

senha, mais comumente encontrada em shadow.

UID, *User Identification*, número que identifica o usuário.

GID, *Group Identification*, número que identifica o grupo primário.

comentários, pode conter nome de usuário, endereço, etc entre aspas simples e campos separados por vírgulas.

define o diretório home do usuário.

o shell do usuário.

O arquivo **/etc/group** tem uma relação dos grupos do sistema, segundo molde:

nome_do_grupo:senha:GID:lista_de_usuários

As descrições dos campos são:

o nome do grupo

a senha (criptografada) do grupo. Se este campo estiver vazio, nenhuma senha é necessária. (gpasswd)

o identificador numérico do grupo.

nomes de usuário de todos os membros suplementares do grupo, separados por vírgulas.

O arquivo **/etc/shadow** contém as senhas dos usuários, segundo os moldes:

login:1/clCUNfk$9ULCTE27T94Po9qRp5oJi0:13581:0:99999:7:::

As descrições dos campos são:

login

senha criptografada

dias, desde 01/01/1970, que a senha sofreu a última alteração

dentro de quantos dias a senha não pode ser alterada.

dentro de quantos dias a senha deverá ser alterada.

quantos dias antes da expiração da senha o usuário receberá aviso.

quantos dias após a expiração da senha a conta será desabilitada.

dias, desde 01/01/1970, que a conta está desabilitada.

campo reservado.

O arquivo **/etc/login.defs** contém uma série de diretivas e padrões que serão utilizados na criação das próximas contas de usuários. Seu principal conteúdo é:

```
MAIL_DIR dir # Diretório de e-mail
PASS_MAX_DAYS 99999 #Número de dias até que a senha expire
PASS_MIN_DAYS  0 #Número mínimo de dias entre duas trocas senha
PASS_MIN_LEN 5  #Número mínimo de caracteres para composição da senha
PASS_WARN_AGE 7 #Número de dias para notificação da expiração da senha
UID_MIN 500 #Número mínimo para UID
UID_MAX 60000 #Número máximo para UID
GID_MIN 500 #Número mínimo para GID
GID_MAX 60000 #Número máximo para GID
CREATE_HOME yes #Criar ou não o diretório home
```

Como o login.defs o arquivo **/etc/default/useradd** contém padrões para criação de contas. Seu principal conteúdo é:

```
GROUP=100 #GID do primeiro usuário criado
HOME=/home #Diretório a partir do qual serão criados os “homes”
INACTIVE=-1 #Quantos dias após a expiração da senha a conta é desativada
EXPIRE=AAAA/MM/DD #Dia da expiração da conta
SHEL=/bin/bash #Shell atribuído ao usuário.
SKEL=/etc/skel #Arquivos e diretórios padrão para os novos usuários.
```

2.4 Alterando parâmetros das contas

Para modificarmos uma conta já existente podemos usar o comando

```
usermod opçoes login
```

onde as principais opções são:

-c comentário

-d diretório_home

-e data_de_expiração data, na forma YYYY-MM-DD, que a conta será desativada

-g grupo grupo primário

-G grupo grupo(s) suplementar(es)

-l login novo *login*

-L trava a senha

-s shell

3 Cotas em disco para usuários e grupos

3.1 Introdução

O sistema de cotas em disco é muito importante pois permite um controle efetivo do espaço em disco a ser utilizado por usuários e grupos. Permite que o administrador controle o sistema de tal modo a não ocorrer travamentos por partições lotadas.

A implementação do sistema de cotas no Linux se dá por partições, ou seja, em todas as partições onde desejamos um efetivo controle devemos configurar o sistema de cotas. As cotas são determinadas por usuário e/ou grupo, sendo que podemos impor limites por espaço ocupado e/ou por número de arquivos e diretórios criados. Por exemplo, se um determinado usuário recebe uma cota de 100 MB, ele poderá ocupar no máximo 100 MB de espaço na partição, seja qual for o diretório da partição. Ao mesmo tempo não terá contabilizado em sua cota algum arquivo ou diretório salvo em outra partição.

3.2 Implementação

Como primeira etapa devemos instalar os pacotes que permitirão o uso do sistema de cotas, com o comando:

```
urpmi quota
```

Em seguida devemos informar ao sistema de arquivos em qual(is) partição(ões) pretendemos implantar o sistema de cotas. Para isto devemos editar o arquivo /etc/fstab e inserir ao final da quarta coluna, separado por vírgula, a diretiva usrquota e/ou grpquota, dependendo se desejamos quotas para usuários, grupos ou ambas. No exemplo abaixo habilitamos cotas para usuários no /home e para usuários e grupos no /dados.

- Arquivo original

  /dev/hda5 / ext3 defaults 1 1

/dev/hda7 /home ext3 defaults 1 2

/dev/hda9 /dados ext3 defaults 1 2

/dev/hda /mnt/cdrom auto umask=0,users,iocharset=utf8,noauto,ro,exec0 0

/dev/hda1 /mnt/win_c ntfs umask=0,nls=utf8,ro 0 0

/dev/hda8 /mnt/win_d vfat umask=0,iocharset=utf8 0 0

none /proc proc defaults 0 0

/dev/hda6 swap swap defaults 0 0

- Arquivo modificado

  /dev/hda5 / ext3 defaults 1 1

  /dev/hda7 /home ext3 defaults,usrquota 1 2

  /dev/hda9 /dados ext3 defaults,usrquota,grpquota 1 2

  /dev/hda /mnt/cdrom auto umask=0,users,iocharset=utf8,noauto,ro,exec0 0

  /dev/hda1 /mnt/win_c ntfs umask=0,nls=utf8,ro 0 0

  /dev/hda8 /mnt/win_d vfat umask=0,iocharset=utf8 0 0

  none /proc proc defaults 0 0

  /dev/hda6 swap swap defaults 0 0

Após isto devemos desmontar e montar a(s) partição(ões) modificadas para que o sistema de arquivos do kernel releia o arquivo e monte segundo os novos parâmetros. Para isto devemos usar a seguinte seqüência de comandos:

```
<Ctrl>+<Alt>+<F1>       # Chaveamos para o modo texto puro.

Logar como root

init 3                  # Matamos o ambiente gráfico

umount -a               # Desmontamos todas as partições. Exceção ao /

mount -a                # Montamos todas as partições do fstab

quotacheck -augv        # Vai inicializar o sistema de cotas
```

Após isto serão criados os arquivos /home/aquota.user, /home/aquota.group, /dados/aquota.user e /dados/aquota.group. Estes arquivos são uma espécie de banco de dados que contém uma relação entre usuários/grupos e o espaço_em_disco/arquivos_e_diretórios usados pelos mesmos.

Agora podemos estabelecer as cotas por usuários ou grupos com o comando:

```
edquota login
```

que abrirá um editor com algo parecido com:

Disk quotas for user login (uid 534):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda7	43460	102400	112640	482	0	0
/dev/hda9	57360	62400	72640	432	0	0

Na primeira linha tem a informação do usuário. As demais linhas são divididas em colunas com os significados:

- Filesystem informa qual(is) partição(ões) tem o sistema de cotas habilitada(s).
- blocks informa o espaço em disco já em uso pelo referido usuário.
- soft a cota em disco
- hard o limite máximo a ser atingido pelo usuário. O limite acima de *soft* e até *hard* poderá ser usado por até uma semana, após a qual o usuário não conseguirá salvar mais nenhum arquivo, só conseguirá apagar até que baixe do valor de *soft*.
- inodes informa o número de arquivos e diretórios em nome do usuário.
- soft e hard informa as cotas para número de arquivos e diretórios. Zero significa que não há limites.

Para edição da cota de grupo o processo é exatamente o mesmo sendo que deve-se adicionar a *flag* -g no comando, por exemplo:

```
edquota -g grupo
```

3.3 Estabelecendo cotas para vários usuários e/ou grupos

No caso que tenhamos que implantar cotas para vários usuários e/ou grupos não é viável ficarmos editando as cotas individualmente. Neste caso recomenda-se editar a cota para um determinado usuário padrão e replicar a mesma para os demais com o comando:

```
edquota -p padrão login
```

3.4 Verificando cotas de usuários

Para tal usa-se o comando:

```
repquota -a
```

4 Agendamento de tarefas com Crontab

4.1 Introdução

O "*cron*" é um programa de agendamento de tarefas. Com ele pode-se fazer a programação para execução de qualquer programa numa certa periodicidade ou até mesmo em um exato dia, numa exata hora. Um uso bem comum do cron é o agendamento de tarefas administrativas de manutenção do seu sistema, como por exemplo, análise de segurança do sistema, backup, entre outros. Estas tarefas são programadas para, todo dia, toda semana ou todo mês, serem automaticamente executadas através da crontab e um *script shell* comum. A configuração do cron geralmente é chamada de crontab.

Os sistemas Linux possuem o cron na instalação padrão. A configuração tem duas partes: uma global, e uma por usuário. Na global, que é o root quem controla, o crontab pode ser configurado para executar qualquer tarefa de qualquer lugar, como qualquer usuário. Já na parte por usuário, cada usuário tem seu próprio crontab, sendo restringido apenas ao que o usuário pode fazer (e não tudo, como é o caso do root).

4.2 Uso do Crontab

Para configurar um crontab por usuário, utiliza-se o comando "*crontab*", junto com um parâmetro, dependendo do que se deseja fazer. Abaixo uma relação:

Comando	**Função**
crontab -e	Edita a crontab atual do usuário logado
crontab -l	Exibe o atual conteúdo da crontab do usuário
crontab -r	Remove a crontab do usuário

Se você quiser verificar os arquivos crontab dos usuários, você precisará ser root. O comando crontab coloca os arquivos dos usuários no diretório:

```
/var/spool/cron/usuario
```

Onde "*usuario*" corresponde ao usuário dono do arquivo crontab.

Agora se deseja-se editar o crontab global, este fica no arquivo "*/etc/crontab*", e só pode ser manipulado pelo root.

Vamos estudar o formato da linha do crontab, que é quem vai dizer o que executar e quando. Vamos ver um exemplo:

```
0       4       *       *       *       who
```

A linha é dividida em 6 campos separados por tabs ou espaço:

Campo	Função
1o.	Minuto
2o.	Hora
3o.	Dia do mês
4o.	Mês
5o.	Dia da semana
6o.	Programa para execução

Todos estes campos, sem contar com o 6o., são especificados por números. Veja a tabela abaixo para os valores destes campos:

Campo	Função
Minuto	0-59
Hora	0-23
Dia do mês	1-31
Mês	1-12
Dia da semana	0-6 (o "0" é domingo, "1" segunda, etc)

Então o que nosso primeiro exemplo estava dizendo? A linha está dizendo: "*Execute o comando 'who' todo dia de todo mês sendo o dia qualquer dia da semana, às 4 horas e 0 minutos.*".

Vamos analisar mais alguns exemplos:

```
1,21,41 *       *       *       *       echo "Meu crontab rodou mesmo!"
```

Aqui está dizendo: "*Executar o comando do sexto campo toda hora, todo dia, nos minutos 1, 21 e 41*".

```
30      4       *       *       1       rm -rf /tmp/*
```

Aqui está dizendo: "*Apagar todo conteúdo do diretório /tmp toda segunda-feira, as 4:30 da manhã.*"

```
45      19      1,15    *       *       /usr/local/bin/backup
```

Aqui está dizendo: "*Executar o comando 'backup' todo dia 1 e 15 às 19:45.*".

E assim pode-se ir montando inúmeros jeitos de agendamento possível. No arquivo do crontab global, o sexto campo pode ser substituído pelo nome do usuário, e um sétimo campo adicionado com o programa para a execução, como mostrado no

exemplo a seguir:

```
*/5 * * * * root /usr/bin/mrtg /etc/mrtg/mrtg.cfg
```

Aqui está dizendo: "*Executar o mrtg como usuário root, de 5 em 5 minutos sempre.*"

```
0      19-23/2      *    *       *       /root/script
```

Aqui está dizendo: “Executar o 'script' entre as 19 e 23 de 2 em duas horas.”

5 Servidor de b*ackups* Amanda

5.1 Introdução

Um dos principais quesitos de segurança de redes é a integridade física dos dados e informações armazenadas.

O *backup* é uma ou várias cópias de segurança dos dados, para a recuperação dos dados em caso de acidentes. Objetiva assegurar a integridade contra possíveis quedas do sistema ou problemas com o disco principal. Assegurar a recuperação de arquivos de usuários apagados/corrompidos acidentalmente.

O **Amanda** (*Advanced Maryland Network Disk Archiver*) é um sistema de *backup* cliente/servidor, melhor que um único programa. Um servidor **Amanda** irá realizar numa única controladora de fita o *backup* de qualquer número de computadores que tenham o cliente do **Amanda** e uma conexão de com o servidor **Amanda**. Um problema comum em locais com um grande número de discos é que a quantidade de tempo requerida para o *backup* dos dados diretamente na fita excede a quantidade de tempo para a tarefa. O **Amanda** resolve este problema utilizando um disco auxiliar para realizar o *backup* de diversos sistemas de arquivos ao mesmo tempo. O **Amanda** cria ``conjuntos de arquivos '': um grupo de fitas utilizadas sobre o tempo para criar os *backups* completos de todos os sistemas de arquivos listados no arquivo de configuração do **Amanda**. O ``conjunto de arquivos'' também pode conter *backups* incrementais (ou diferenciais) noturnos de todos os sistemas de arquivos. Para restaurar um sistema de arquivos é necessário o *backup* completo mais recente e os incrementais, este controle é feito pelo próprio **Amanda**.

O arquivo de configurações provê um controle total da realização dos *backups* e do tráfico de rede que o **Amanda** gera. O **Amanda** utilizará qualquer programa de *backup* para gravar os dados nas fitas, por exemplo tar. O **Amanda** está disponível como pacote, porém ele não é instalado por padrão.

5.2 Configuração do servidor Amanda

http://wiki.zmanda.com/index.php/Main_Page

http://www.amanda.org/docs/

Para instalar o Amanda devemos primeiramente copiar os pacotes necessários da máquina do professor (por questões de velocidade os mesmo já foram baixados de http://rpm.pbone.net/) com o comando:

```
scp -r aluno@192.168.2.1:amanda ./
```

Instalar alguns pacotes para satisfazer as dependências com os comandos:

```
urpmi xinetd

urpmi gnuplot

urpmi mt-st
```

Mude para o diretórios recém criado amanda com o comando:

```
cd amanda
```

Instale os pacotes do servidor e cliente Amanda com o comando:

```
rpm -ivh *
```

Após a instalação o arquivo padrão /etc/amanda/DailySet1/amanda.conf, estará criado. Como exemplo configuraremos o sistema para um backup fictício, já que não dispomos de unidades de fita para um verdadeiro backup.

Para "*Backup* em HD com Amanda" pode-se ver um tutorial em http://www.cybershark.net/tutoriais/amanda_hd. Muito útil também o tutorial http://www.amanda.org/docs/howto-filedriver.html

5.2.1 amanda.conf

O Amanda é bastante flexível. Poderíamos ter uma série de conjuntos independentes de backup no mesmo servidor. Para isto bastaria criar tantos diretórios de configuração quanto tivéssemos necessidade, a exemplo do cefetsc. Cada um tendo características independentes de periodicidade e dados a serem armazenados.

Em nosso exemplo vamos criar um único conjunto de *backups*. Para isso faça uma cópia do exemplo /etc/amanda/DailySet1 para cefetsc com o comando:

```
cp -rf /etc/amanda/DailySet1 /etc/amanda/cefetsc
```

Edite o amanda.conf do cefetsc e modifique somente as linhas abaixo, as demais

definições são as padrão:

```
org "cefetsc"
mailto "suporte@sj.cefetsc.edu.brodilson@sj.cefetsc.edu.br"
tapecycle 21 tapes      #Número de fitas disponíveis.
tapedev "/dev/nst0"     #Dispositivo da fita.
tapetype DDS4           #Tipo de fita
labelstr "^cefetsc[0-9][0-9]*$"        #Rótulos para fitas = cefetsc01 à 99.
infofile "/var/lib/amanda/cefetsc/curinfo"   #database filename
logdir   "/var/lib/amanda/cefetsc"         #log directory
indexdir "/var/lib/amanda/cefetsc/index"   #index directory
```

5.2.2 disklist

Edite o /etc/amanda/DailySet1/disklist, vá ao final do arquivo e acrescente todos os diretórios, de todas as máquinas, que deseja backup, veja o exemplo:

```
dk.uned.sj /scripts comp-root-tar
dk.uned.sj /etc comp-root-tar
[....]
hendrix.sj.cefetsc.edu.br /etc comp-root-tar
hendrix.sj.cefetsc.edu.br /root comp-root-tar
[....]
titas.uned.sj   /var/www comp-root-tar
titas.uned.sj   /home comp-root-tar
[....]
```

Neste caso estaríamos configurando o backup do diretório /scripts e /etc da máquina dk.uned.sj o /etc e /root da hendrix.sj.cefetsc.edu.br e /var/www e /home da titas.uned.sj. É evidente que os clientes deverão responder por estes nomes (DNS) e ter o cliente **Amanda** configurado, conforme procedimento abaixo.

5.3 Configurando o cliente

Normalmente desejamos que o servidor **Amanda** também seja cliente. Em nosso caso já instalamos os pacotes necessários no item “Configuração do Servidor Amanda”.

Após isto devemos editar o arquivo /etc/xinetd.d/amanda e modificarmos uma única diretiva como abaixo:

```
disable = no
```

Reiniciamos o serviço com o comando:

```
service xinetd restart
```

Agora Vamos testar as configurações, no servidor, com o comando:

```
/usr/sbin/amcheck cefetsc
```

Se tudo estiver correto obteremos uma mensagem do tipo:

```
Amanda Tape Server Host Check

-----------------------------

Holding disk /amanda: 22985136 kB disk space available, that's plenty

NOTE: skipping tape-writable test

Tape cefetsc06 label ok

WARNING: tapecycle (21) <= runspercycle (21).

Server check took 0.023 seconds


Amanda Backup Client Hosts Check

--------------------------------

Client check: 3 hosts checked in 0.090 seconds, 0 problems found


(brought to you by Amanda 2.4.5)
```

5.4 Backups com o Amanda

Para fazermos backup dos diretórios usamos o **amdump** com a sintaxe:

```
amdump cefetsc
```

Este também é o comando que deve ser programado na crontab para os *backups* periódicos.

5.5 Restaurando os *backups* com Amanda

Uma forma de restaurar os *backups* é utilizando o '**amrecover**'. Ela é a opção mais poderosa, por isso deve ser dado preferência ao uso da mesma.

Na máquina servidora, como root, crie um diretório de restore:

```
mkdir restore

cd restore
```

Agora, chame o programa de recuperação:

```
amrecover cefetsc
```

Dê o comando que determina a data que você quer restaurar, no formato AAAA-MM-DD, onde AAAA é o ano com quatro dígitos, MM é o mês com dois dígitos e DD é o dia com dois dígitos:

```
setdate 2006-09-05
```

O próximo passo é determinar de qual cliente se quer restaurar o *backup*:

```
sethost localhost
```

A seguir determina-se de qual "disco" se restaurará o *backup*. Abaixo, um exemplo, de como extrair somente um arquivo do diretório /etc:

```
setdisk /etc
```

Agora pode-se navegar pelos diretórios:

```
cd rc.d
```

O próximo passo é adicionar o arquivo a ser restaurado. Lembre-se que você pode usar coringas, como o * para adicionar todos os arquivos, ou pode adicionar um diretório:

```
add rc.local
```

Continue adicionando diretórios e arquivos, conforme o necessário. Depois disso

resta extrair os arquivos:

```
extract
```

5.6 Comandos Extras do Amanda

Para descarregar o disco de suporte, temporário, devemos fazer uso do comando **amflush**.

Para limpeza geral no serviço Amanda execute o **amcleanup**.

Para gravar rótulos em fitas use o **amlabel**. Esta operação é muito importante para não haver sobreposição de dados pelo sistema amanda. As fitas, tendo rótulo, serão manipuladas devidamente pelo Amanda.

Para tarefas administrativas **amadmin**.

Amcheck verifica de você está utilizando a fita correta (esperada), se há espaço livre suficiente no disco de suporte e se as máquinas clientes estão configuradas adequadamente.

Para gerenciamento de empilhadores e trocadores de fita, se houver, use o **amtape**.

Para esboçar gráfico da atividade do Amanda em cada *dump* que for executado use o **amplot**.

6 Configuração da interface de rede

6.1 Introdução

Até o momento trabalhamos com o sistema "desconectado". A partir de agora trabalharemos com serviços em rede e a primeira tarefa é justamente configurar a(s) interface(s).

A configuração da(s) interface(s) de rede é um processo bastante simples, desde que se tenha um conhecimento prévio de protocolos TCP/IP e classes de redes.

6.2 Configuração

Para adicionarmos uma máquina à rede é obrigatória a configuração de no mínimo os seguintes parâmetros: endereço ip, máscara de rede. Com estes dois parâmetros a máquina já se comunica com outras máquinas da rede local.

Para uma configuração completa é necessário configurarmos ainda o nome de máquina, o servidor de nomes (DNS) e o roteador padrão (*default gateway*).

Todos estes parâmetros podem ser configurados estaticamente ou por meio de um

servidor DHCP.

No caso de servidores de rede é praticamente obrigatório, para alguns serviços é obrigatório, que a configuração seja estática. Para a maioria dos clientes a configuração clássica é como cliente DHCP.

6.2.1 Configuração do ifcfg-ethN

O ifcfg-ethN é o arquivo de configuração de cada uma das interfaces de rede existentes na máquina. Edita-se o arquivo /etc/sysconfig/network-scripts/ifcfg-ethN, onde N é o número da interface. No caso de uma única interface este número é 0 (zero). Este arquivo tem os seguintes parâmetros mínimos:

```
DEVICE=eth0        # Nome do dispositivo
BOOTPROTO=static   # Configuração estática. Dinâmico seria dhcp
IPADDR=192.168.2.X # Endereço ip
NETMASK=255.255.255.0   # Máscara de rede
BROADCAST=192.168.2.255 # Endereço de broadcast – para todas as máquinas.
GATEWAY=192.168.2.1# Roteador padrão
ONBOOT=yes         # Interface inicializa no boot da máquina
MS_DNS1=192.168.1.1 # Endereço do servidor DNS primário
MS_DNS2=200.135.233.1   #  Endereço do servidor DNS secundário
```

6.2.2 Configuração do network

Este arquivo define o nome da máquina e quem é o roteador padrão da mesma. Edita-se o arquivo /etc/sysconfig/network, que tem por exemplo os seguintes parâmetros:

```
NETWORKING=yes     # Trabalha ou não em rede
GATEWAY=192.168.2.1# Roteador padrão
GATEWAYDEV=eth0    # Interface de acesso ao roteador padrão
HOSTNAME=mX.redesX.edu.br     # Nome da máquina
```

6.2.3 Configuração do resolv.conf

Este arquivo define qual(is) é(são) o(s) servidore(s) DNS da máquina. Edita-se o arquivo /etc/resolv.conf, que tem por exemplo os seguintes parâmetros:

```
nameserver 192.168.1.1
```

nameserver 200.135.233.1

Uma vez ajustados estes parâmetros basta reiniciar a interface de rede com o comando:

```
service network restart
```

Os parâmetros ajustados acima serão assumidos como o padrão da máquina e, sempre que a máquina ou interfaces de rede forem reiniciadas, os valores assumidos serão dados por estes arquivos.

O comando ifconfig, sem parâmetros, serve para mostrar a atual configuração da(s) interface(s) de rede. Se desejarmos, para um teste por exemplo, podemos mudar os parâmetros da interface de rede sem modificar estes arquivos. Para isto usamos o comando ifconfig com a seguinte sintaxe:

```
ifconfig interface ip/mascara up/down

ifconfig eth0 192.168.2.X/24 up
```

No comando acima estamos mudando a configuração da interface de rede eth0 para assumir o endereço ip 192.168.2.**X** com máscara de rede 255.255.255.0. Isto passará a valer imediatamente mas ao reiniciarmos a interface ou máquina os arquivos acima serão lidos e "configurarão" a interface.

6.3 Apelidos de ip

No Linux, a mesma interface de rede pode responder por mais de um endereço ip. Isto pode ser útil em algumas configurações especiais de rede, por exemplo, duas sub-redes no mesmo domínio de colisão ou um servidor Apache atendendo a domínios virtuais. Para fazermos isto basta criarmos interfaces virtuais com nomes ethN:0, ethN:1 etc. Por exemplo:

```
ifconfig eth0:0 192.168.1.65/24 up
```

Ou criando um arquivo /etc/sysconfig/network-scripts/ifcfg-eth0:0 com somente o conteúdo abaixo e reiniciarmos a rede.

IPADDR=192.168.2.1**X**

NETMASK=255.255.255.0

7 Roteadores e sub-redes

7.1 Introdução

Um roteador, por definição, é um equipamento com no mínimo duas interfaces de

rede que encaminha os pacotes oriundos de uma das interfaces à outra, de acordo com regras pré-definidas. No mercado existem roteadores com uma interface ethernet e uma, duas ou três interfaces WAN (*Wide Area Network*), normalmente utilizados para conexão da rede local com a internet. Existem também ou chamados modem/router que além de roteadores são modens, comumente usados para conexão ADSL.

O roteamento é sem dúvida um dos principais serviços (protocolos) da rede TCP/IP, já que é por meio dele que é possível um pacote originado na Brasil chegar rapidamente ao Japão, por exemplo.

Uma máquina Linux, com duas ou mais interfaces de rede, também funcionar como um roteador. Esta pode ser uma opção interessante se desejarmos criar sub-redes na instituição e obrigatória na implementação de um *firewall* transparente.

O roteamento estático trabalha com uma tabela que é lida linha-a-linha de tal modo que quando for encontrada uma regra que atenda a "demanda" o sistema pára imediatamente. Analisemos uma tabela de roteamento de uma estação qualquer:

```
Destino         Roteador        MáscaraGen.     Opções Métrica Ref   Uso Iface

192.168.1.0     0.0.0.0         255.255.255.0   U     10     0        0 eth0

127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo

0.0.0.0         192.168.1.254   0.0.0.0         UG    10     0        0 eth0
```

A primeira linha informa que, para pacotes destinados à rede 192.168.1.0/24, basta "jogar" os mesmos pela interface eth0, já que o roteador (0.0.0.0) não é definido.

A segunda linha trata da rota para a interface de *loopback*.

A terceira linha trata do roteador padrão. Para destino qualquer (0.0.0.0) deve-se encaminhar os pacotes para o endereço 192.168.1.254.

De um outro modo, esta tabela nos diz o seguinte: se um pacote for destinado à 192.168.1.0/24 o "roteamento" analisará somente a primeira linha e simplesmente "jogará" o pacote na interface eth0, se o pacote for destinado a qualquer endereço iniciado com 127 o pacote será "jogado" pela interface virtual lo e por último para qualquer outro destino (0.0.0.0) o pacote será encaminhado para 192.168.1.254 e este "que se vire". Observe que o endereço 192.168.1.254 é um endereço "atingível" pela interface eth0.

7.2 Configurando o roteador

Para transformarmos nossa máquina, de uma estação com duas interfaces de rede,

em um roteador basta setarmos o bit ip_forward para 1. Isto pode ser feito com o comando:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

E imediatamente nossa máquina passará a rotear pacotes de uma interface à outra. Este roteamento ocorrerá somente se os pacotes tiverem um destino explícito à outra interface, caso contrário os pacotes não serão roteados, ou seja, um roteador segmenta a rede, e seu tráfego, criando sub-redes distintas.

7.3 Configurando sub-redes

Como caso de estudos vamos montar a estrutura de sub-redes mostrada no diagrama esquemático abaixo, Ilustração 4.

Neste diagrama percebemos que, após as configurações necessárias, teremos 6 sub-redes compostas de 6 roteadores e 6 clientes para testes. A máquina "Professor", que também é um roteador, interligará estas sub-redes à rede da Unidade São José do CEFETSC. Esta máquina será o roteador padrão de cada um dos 6 roteadores.

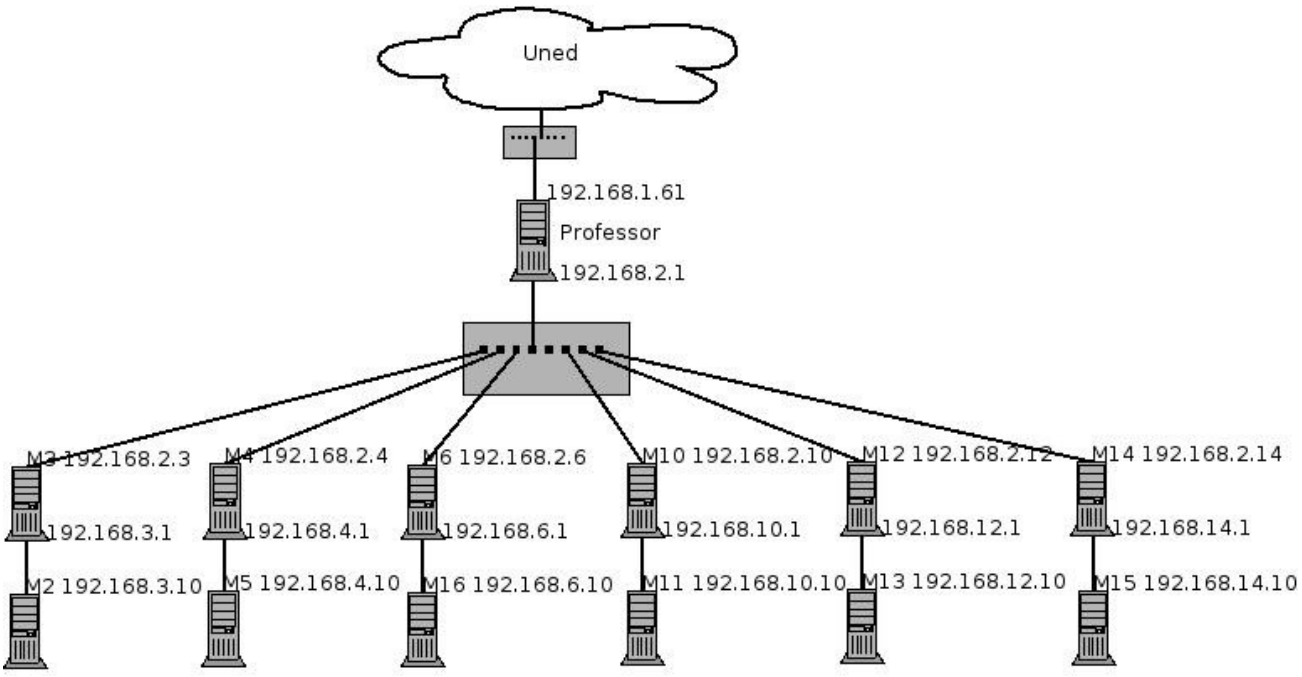


Ilustração 4: Diagrama de sub-redes

7.4 Caso de estudo

7.4.1 Roteadores

No roteador devemos, em primeiro lugar, definir os parâmetros da segunda interface de rede. Por questões de facilidade, vamos usar *ip aliases,* ou seja, uma única interface de rede do roteador responderá pelos dois ip's, mesmo sendo de classes diferentes. Isto permitirá que não seja necessário reestruturar o cabeamento.

Devemos adotar a numeração do diagrama esquemático e seguir o modelo abaixo:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0:1
```

IPADDR=192.168.**X**.**X**

NETMASK=255.255.255.0

Reiniciamos o serviço de rede com o comando:

```
service network restart
```

Adicionamos as rotas para as 5 demais sub-redes com 5 comandos baseados no modelo abaixo:

```
route add -net 192.168.X.0/24 gw 192.168.2.X
```

Onde o parâmetro net identificará a rede a ser atingida e o gw identificará qual a interface "conhecida" (endereço) do roteador da respectiva rede.

Configuramos o roteamento do roteador com o comando:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

7.4.2 Configuração do Cliente

No cliente devemos redefinir os parâmetros de rede, conforme modelo abaixo:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

DEVICE=eth0

BOOTPROTO=static

IPADDR=192.168.**X**.10

NETMASK=255.255.255.0

BROADCAST=192.168.**X**.255

ONBOOT=yes

```
vi /etc/sysconfig/network
```

GATEWAY=192.168**X**.1

Reiniciamos o serviço de rede com o comando:

```
service network restart
```

7.4.3 Testes

1. A partir do cliente "pingar" a interface mais próxima do roteador. Se este ping não funcionar devemos revisar a configuração física e lógica entre este e o roteador.
2. A partir do cliente "pingar" a interface externa do roteador. Se não pingar será por que o roteador não está roteando.
3. A partir do roteador "pingar" para 192.168.2.1. Se não pingar é por que tem algum erro de configuração física ou lógica na interface externa do roteador.
4. A partir do roteador "pingar" para a interface interna de um outro roteador, por exemplo 192.168.4.1. Se houver problemas os motivos podem ser dois: não foi escrita uma rota adequada para tal rede, verificamos com o comando route -n, e/ou porque o roteador "pingado" está mal configurado. Lembre-se que os pacotes devem rota para ida e volta.
5. A partir do cliente pingar para outro cliente. Se houver problemas pode ser por má configuração do roteador "local" ou do roteador da rede "pingada".

Obs.: desfaça somente as tabelas de roteamento para podermos implementar NAT.

8 NAT - *Network Address Translator*

A tradução de endereço de rede é um procedimento que objetiva criar sub-redes e também a segurança das mesmas.

Podemos implementar um NAT de diversos modos mas a mais recomendada por facilidades e segurança é o mascaramento da seguinte forma:

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j MASQUERADE
```

Esta regra diz o seguinte: todos os pacotes que passarem (POSTROUTING) por esta máquina com origem de 192.168.1.0/24 e sairem pela interface eth0 serão mascarados, ou seja sairão desta máquina com o endereço de origem como sendo

da eth0.

Com estas configurações o cliente acessa qualquer site na internet mas não pode ser acessado. A partir do cliente faça testes “pingando” para sites externos, roteadores vizinhos e tente pingar nos clientes vizinhos.

Obs.: desfaça todas as alterações, NAT, roteador etc para poder prosseguir.

9 Servidor DNS - *Domain Name System com Bind*

9.1 Introdução

Se o roteamento mantém a "conexão" entre as milhares de máquinas ligadas a internet o DNS faz o papel de dar nome às mesmas já que, para nós seres humanos, é difícil guardar números mas fácil gravar nomes. Já para as máquinas vale o oposto, entendem bem números mas nem tanto nomes. A interface entre estes dois mundos é feita pelo DNS.

O DNS é portanto uma tabela relacionando nomes e números ip. O DNS direto relaciona o número ip de uma máquina e seu nome. O DNS reverso relaciona o nome e seu ip.

O processo de resolução de nomes segue o esquema da Ilustração 5.

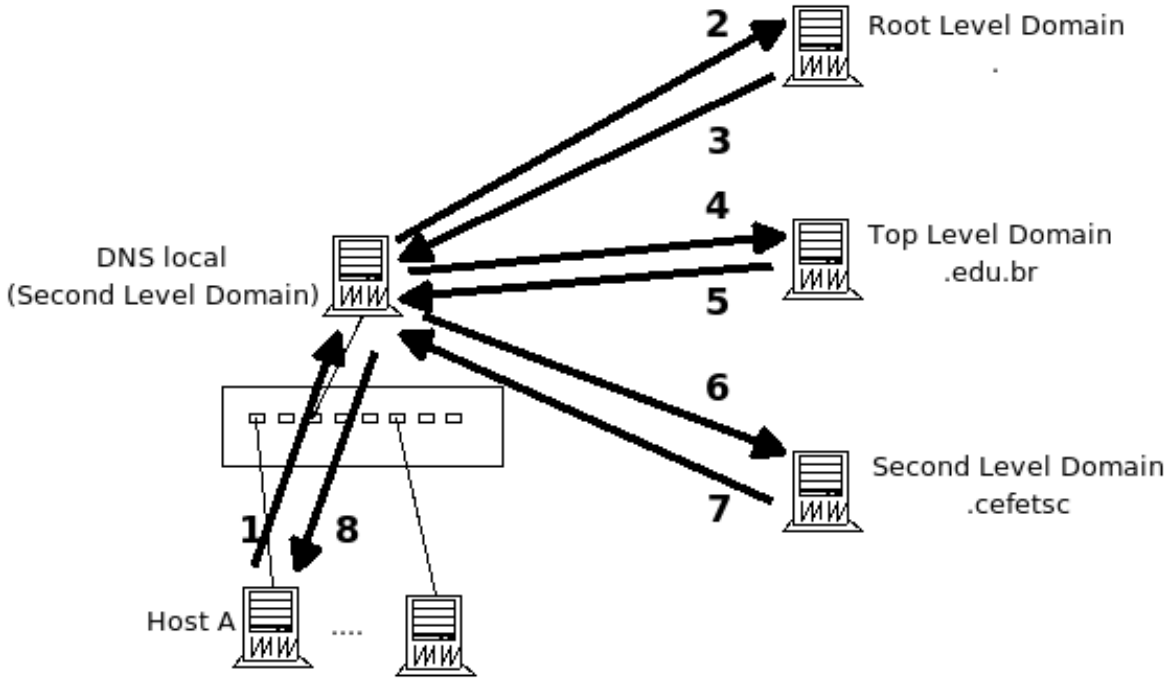


Ilustração 5: Processo de resolução de nomes

Vamos detalhar as setas em destaque:

1. O cliente, host A, deseja acessar algum serviço da máquina www.cefetsc.edu.br. Este "olha" em sua tabela de *cache* DNS e verifica que não tem o ip desta máquina então solicita esta informação ao DNS local, seu "nameserver".
2. O DNS local busca em sua *cache* o ip de tal máquina, caso não encontre pede diretamente ao *Root Level Domain*.
3. Este responde que não conhece explicitamente o endereço da máquina mas sabe quem é o *Top Level Domain* responsável por aquele endereço.
4. O DNS local pede então ao *Top Level Domain* qual o ip da máquina.
5. Este responde que não conhece explicitamente o endereço da máquina mas sabe quem é o *Second Level Domain* responsável por aquele endereço.
6. O DNS local pede então ao *Second Level Domain* qual o ip da máquina.
7. O *Second Level Domain*, responsável pelo domínio "cefetsc.edu.br." informa então o ip da máquina "www.cefetsc.edu.br." ao DNS Local.
8. O DNS local armazena na tabela DNS *cahe* a correspondência entre ip e nome e entrega a informação ao cliente – Host A – que também guarda em seu cache.

9.2 Configuração de um servidor DNS

Como primeiro passo devemos instalar o pacote bind com o comando:

```
urpmi bind
```

Como primeiro passo devemos instalar o pacote bind com o comando:

```
urpmi bind
```

9.2.1 Caso de estudo

Para podermos verificar o funcionamento do DNS vamos montar a estrutura lógica mostrada na Ilustração 6. **A máquina "professor" terá uma cópia de todos os arquivos de todos os domínios diretos e será a única a ter domínio reverso**. Ela será o servidor DNS de todos.

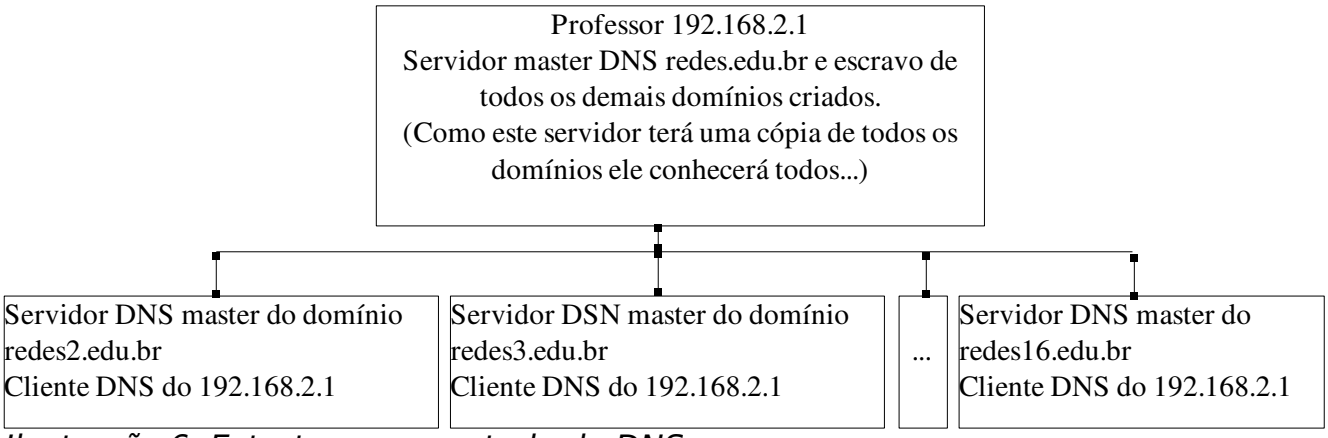


Ilustração 6: Estrutura para estudo do DNS

Para isto edite seu arquivo /etc/named.conf e crie o domínio redes**X**.edu.br incluindo no final do arquivo as seguintes linhas:

```
zone "redesX.edu.br" IN {                   # nome do domínio
        type master;                        # servidor master (slave etc)
        file "master/redesX.zone";          # arquivo de definição do domínio
        allow-update { none; };             # sem atualizações dinâmicas
};
```

Agora edite o arquivo /var/lib/named/var/named/master/redes**X**.zone de acordo com o modelo abaixo. Obs.: copie o arquivo localdomain.zone para referência inicial.

```
$TTL    86400
@             IN SOA  mX.redesX.edu.br root (
                                2007032000          ; serial
                                3H          ; refresh
                                15M          ; retry
                                1W          ; expiry
                                1D )        ; minimum
              IN NS         mX.redesX.edu.br.
              IN MX   0     mX.redesX.edu.br.
localhost     IN A          127.0.0.1
$ORIGIN redesX.edu.br.
mX            A     192.168.2.X
www           A     192.168.2.X
```

Inicie o serviço de DNS com o comando:

```
service named start
```

Como primeiro teste configure a sua máquina para ser sua própria cliente editando o /etc/resolv.conf e adicionando a diretiva “nameserver 192.168.2.**X**” no início do

arquivo. Em seguida dê um ping para m**X**.redes**X**.edu.br. Se "pingar" é sinal de que o próprio servidor está funcionando.

Para podermos "enxergar" as demais máquinas devemos configurar a nossa máquina para ser cliente DNS da máquina "professor", editando o arquivo /etc/resolv.conf e deixando-o somente com o conteúdo:

```
nameserver 192.168.2.1
```

9.2.2 Testes

Após as devidas configurações poderemos "pingar" para nomes de máquinas, por exemplo:

```
ping www.redes3.edu.br

nslookup m5.redes5.edu.br
```

Neste caso, antes de efetivamente "pingar" a máquina, o sistema converte o nome para número e mostra na tela, em seguida inicia-se o processo de ping. Se não funcionar a fonte do problema pode ser duas, ou o cliente mau configurado ou o servidor.

Para testar a resolução reversa de nomes usamos o seguinte comando, por exemplo:

```
host 192.168.2.5
```

Este comando deverá retornar o nome da máquina em questão.

10 Servidor de páginas Apache

10.1 Introdução[1]

O servidor Apache (*Apache server*) é o mais bem sucedido servidor web livre. Foi criado em 1995 por Rob McCool,então funcionário do NCSA (National Center for Supercomputing Applications), Universidade de Illinois. Numa pesquisa realizada em dezembro de 2005, foi constatado que a utilização do Apache supera 60% nos servidores ativos no mundo.

O servidor é compatível com o protocolo HTTP versão 1.1. Suas funcionalidades são mantidas através de uma estrutura de módulos, podendo inclusive o usuário escrever seus próprios módulos — utilizando a API do software.

É disponibilizado em versões para os sistemas Windows, Novell Netware, OS/2 e diversos outros do padrão POSIX (Unix, GNU/Linux, FreeBSD, etc).

1 Texto obtido a partir de http://pt.wikipedia.org/wiki/Servidor_Apache

10.2 Instalação e configuração

Como primeiro passo devemos instalar o pacote com o comando:

```
urpmi apache

urpmi apache-doc
```

Obs.: se for o caso escolha a opção 1, **estável**.

Em seguida inicia-se o servidor com o comando:

```
service httpd start
```

O servidor Apache já estrá rodando e pode ser testado usando um navegador com o endereço http://localhost/.

Acesse também o endereço http://localhost/manual/, que contém o manual (apache-doc) do servidor com uma série textos e *links* importantes.

Os principais arquivos de configuração do Apache são:

- /etc/httpd/conf/httpd.conf que está dividido em três seções: global, opções do servidor e máquinas virtuais. Esta última na verdade remete ao arquivo comentado no próximo item.
- /etc/httpd/conf/vhosts.d/vhosts.conf. Este arquivo define os domínios virtuais.

Como primeira configuração vamos mudar a página apresentada por nosso servidor. Para isto criamos um arquivo index.html com o conteúdo:

> <html><body><h1>Esta é minha página de testes. Servidor 192.168.2.**X**</h1></body></html>

E copiamos este arquivo para /var/www/html/, que é o diretório padrão de hospedagem de páginas, com o comando:

```
cp -f index.html /var/www/html/
```

Acessamos novamente a página http://localhost/ e observamos o resultado. Obs.: pode ser necessário fazer uma atualização da página, no navegador, para que o mesmo releia o arquivo.

10.3 Domínios virtuais

O recurso de domínios virtuais é muito interessante pois permite que um servidor Apache responda pelas páginas de vários domínios de maneira independente. A princípio o usuário que acessará estes domínios não saberá que se trata do mesmo servidor. Os domínios podem ser tanto por nomes, desde que se tenha o registro formal do domínio, como por ips.

Para não ficarmos só na configuração básica vamos criar um domínio virtual baseado em ip. Como primeiro passo devemos conferir se nosso apelido de ip está ativo, com o comando:

```
ifconfig
```

Onde deve aparacer a configuração da interface eth0 e eth0:0.

Agora criamos o arquivo /etc/httpd/conf/vhosts.d/vhosts.conf com o seguinte conteúdo:

<VirtualHost 192.168.2.1**X**>

DocumentRoot /var/www/html/virtual

</VirtualHost>

Criamos o diretório virtual com o comando:

```
mkdir /var/www/html/virtual
```

Criamos mais um arquivo index.html dentro deste diretório com o conteúdo:

<html><body><h1>Esta é minha página virtual. Servidor 192.168.2.1**X**</h1></body></html>

Reiniciamos o servidor para que ele releia as configurações de domínios virtuais com o comando:

```
service httpd restart
```

Agora podemos testar e observar que possuímos duas páginas independentes:

http://192.168.2.**X**/

http://192.168.2.1**X**/

10.4 Páginas de Usuários

Para permitir que os usuários tenham sua página pessoal, seja em formato html ou simplesmente como repositório de arquivos (Indexes) procedemos do seguinte modo. Primeiramente instalamos o módulo userdir com o comando:

```
urpmi apache-mod_userdir<Directory /home/*/*/public_html>
```

Em seguida editamos o arquivo /etc/httpd/conf/httpd.conf e acrescentamos ao final do mesmo o seguinte conteúdo (contêiner):

```
<Directory /home/*/public_html>     #Contêiner diretório home dos usuários

    AllowOverride All   #Aceita todo tipo de diretivas de autenticação
```

```
    Allow from all      #Permite acesso a todos

    Options Indexes FollowSymLinks MultiViews   #Indexes, se não houver o arquivo
                                                index.html mostra em formato de
                                                diretório. FollowSymLinks, pemite
                                                seguir os links da  página.
                                                MultiViews, tenta servir a página na
                                                língua do usuário.

    <IfModule mod_access.c>   #Se o módulo de controle de acesso, access, existir...

      Order allow,deny  #Ordem de avaliação das diretivas para permitir ou negar
                        acesso ao recurso.

      Allow from all    #Permite para todos

    </IfModule>         #Fim do if

</Directory>            #Fim do contêiner
```

Reiniciamos o apache com o comando:

```
service httpd restart
```

Assim qualquer usuário, que tiver um diretório public_html dentro de seu diretório de entrada, terá uma página no ar. Dentro do public_html pode ser colocado um arquivo index.html, em linguagem html, ou simplesmente arquivos para *download* externo.

11 Servidor de correio eletrônicoPostfix

11.1 Introdução[2]

Um servidor de correio eletrônico gerencia os e-mails que são enviados e recebidos. Os servidores de e-mail podem ser servidores Internet, onde e-mails enviados e recebidos podem ser transitados para qualquer lugar do mundo, ou servidores de correio de intranet onde as mensagens trafegam apenas dentro da empresa. Através do correio eletrônico podem ser criados grupos de discussão sobre quaisquer assuntos. Estes grupos são chamados de listas ou refletores. Um refletor é uma caixa postal eletrônica falsa. Todas as mensagens enviadas para esta caixa postal, são transmitidas para as pessoas cadastradas na lista deste refletor. Desta forma cada membro do grupo passa a dispor das mensagens enviadas para o refletor em sua caixa postal ou mailbox. Cada membro, pode ler as mensagens e dar a sua opinião sobre elas enviando uma nova mensagem para o

2 Texto obtido em http://pt.wikipedia.org/wiki/Sistema_de_correio_eletr%C3%B4nico

refletor.

Como exemplo de sistemas de correio eletrônico livres podemos citar o Postfix, que é um dos candidatos a substituir o SendMail. O Postfix é hoje uma das melhores alternativas para todas as empresas que desejam utilizar um servidor de email sem ter grandes gastos, ele foi escrito de forma direta e clara e visa facilitar e ajudar o Administrador Linux já que esse software é muito fácil de utilizar, além de ser um agente de transporte de email muitas vezes chamado simplesmente de servidor de email. Além de apresentar grande facilidade para sua configuração ele é um servidor de email robusto e com vários recursos.

11.2 Instalação e configuração

Devemos instalar os pacotes com o comando:

```
urpmi postfix
```

Configuramos o "segundo bloco" do arquivo /etc/postfix/main.cf, acrescentando/mudando somente os seguintes parâmetros:

```
# User configurable parameters
myhostname = mX.redesX.edu.br          # Nome da máquina
mydomain = redesX.edu.br               # Nome do domínio
myorigin = $mydomain                   # Especifica o domínio que aparece
                                       quando se envia um email
inet_interfaces = all                  # Interfaces que o servidor usa
mynetworks_style = subnet              # A rede do servidor
```

Obs.: Outra diretiva interessante a se acrescentar em um caso real é a seguinte: **home_mailbox = Maildir/**. Esta diretiva salvará os emails destinados a determinado usuário em seu diretório home, dentro de uma pasta chamada Maildir. Contabilizando os arquivos em sua cota e deixando os mesmos fisicamente separados dos demais usuários.

11.3 Testes

Para procedermos alguns testes em nosso servidor devemos ter uma ferramente cliente de email. Para isto vamos usar uma ferramente a nível de linha de comando, mail (se não existir deve-se instalar o pacote mailx).

Para enviar uma mensagem proceda do seguinte modo:

- mail usuario@redesX.edu.br <Enter>,
- inserir o subjet <Enter>,

- inserir a mensagem,
- <Ctrl>+<d>.

Como primeiro teste podemos enviar uma mensagem para um usuário da própria máquina e monitorar com o comando:

```
tail -f  /var/log/mail/info
```

Se aparecer algo do tipo:

> Mar 23 09:56:29 professor postfix/pickup[15108]: 81A7C2C44C80: uid=1572 from=<odilson>
>
> Mar 23 09:56:29 professor postfix/cleanup[15113]: 81A7C2C44C80: message-id=<20070323125629.81A7C2C44C80@professor.redes.edu.br>
>
> Mar 23 09:56:29 professor postfix/qmgr[15109]: 81A7C2C44C80: from=<odilson@redes.edu.br>, size=454, nrcpt=1 (queue active)
>
> Mar 23 09:56:29 professor postfix/local[15115]: 81A7C2C44C80: to=<root@redes.edu.br>, orig_to=<root>, relay=local, delay=0.29, delays=0.23/0.01/0/0.04, dsn=2.0.0, **status=sent** (delivered to mailbox)
>
> Mar 23 09:56:29 professor postfix/qmgr[15109]: 81A7C2C44C80: removed

é porque está tudo certo. O principal aviso é o status=sent (em negrito).

Uma vez que esteja funcionando localmente, pode-se enviar email para os colegas e, inclusive, emails externos. Lembrando que os emails externos não chegarão/retornarão ao nosso servidor, pois não termos um domínio válido.

Para ler mensagens basta digitar mail (logado na conta “certa”), aparecerá uma listagem de emails, e em seguida o número da mensagem.

12 Servidor *cache/proxy* Squid

12.1 Introdução[3]

Podemos resumir o significado de servidor proxy como uma espécie de "cache comunitário", onde toda página que um usuário visualizar ficará armazenada e quando outro (ou o mesmo) usuário requisitar a mesma página, ela não será trazida da Internet novamente, simplesmente será lida do disco e entregue, economizando tráfego de rede (isso se a página não tiver sido modificada na origem, caso isto tenha acontecido, ela será trazida da Internet novamente). Um proxy pode, além disso, fazer o controle de conteúdo, barrando o acesso a certos sites, por exemplo.

3 http://www.fundao.wiki.br/articles.asp?cod=199

12.2 Instalação e configuração

Para instalar o squid basta digitarmos o comando:

```
urpmi squid
```

Em seguida iniciamos o serviço com o comando:

```
service squid start
```

Para funcionar como cache, armazenamento centralizado de páginas, basta isto. Já a função de filtro (*firewall*) será estudada no próximo módulo, Segurança e Monitoramento de Redes.

O principal arquivo de configuração é o /etc/squid/squid.conf. As principais diretivas deste arquivo são:

http_port 3128	# Porta à qual o squid atenderá
cache_dir ufs /var/spool/squid 100 16 256	# Diretório de cache do tipo ufs, com armazenamento em /var/spool/squid, tamanho total de 100 MB, com 16 subdiretórios e cada um deles com 256 subdiretórios. Obs.: recomenda-se aumentar o tamanho total e mais nenhum outro parâmetro.
cache_mem 8 MB	# Tamanho da cache em RAM. Dependendo do uso da máquina, ocupe metade da RAM total.
maximum_object_size 4096 KB	# Tamanho máximo de um único objeto. Recomenda-se 16384 (16 MB). Isto é interessante quando faz-se *downloads* de arquivos.
visible_hostname m**X**.redes**X**.edu.br	# Nome real do servidor.

Se mudar algum parâmetro lembre-se de reiniciar o serviço (service squid restart).

12.3 Testes

Para testar devemos configurar nosso navegador para usar o nosso próprio proxy. Para isto abra o Firefox e clique em Editar, Preferências, Avançado, Aba Rede, Configurações, ajuste para Configuração manual de proxy e acrescente em HTTP: localhost, Porta: 3128 e clique em “Usar este proxy para todos os protocolos”. Feche o navegador para que a configuração se torne válida, veja Ilustração 7.

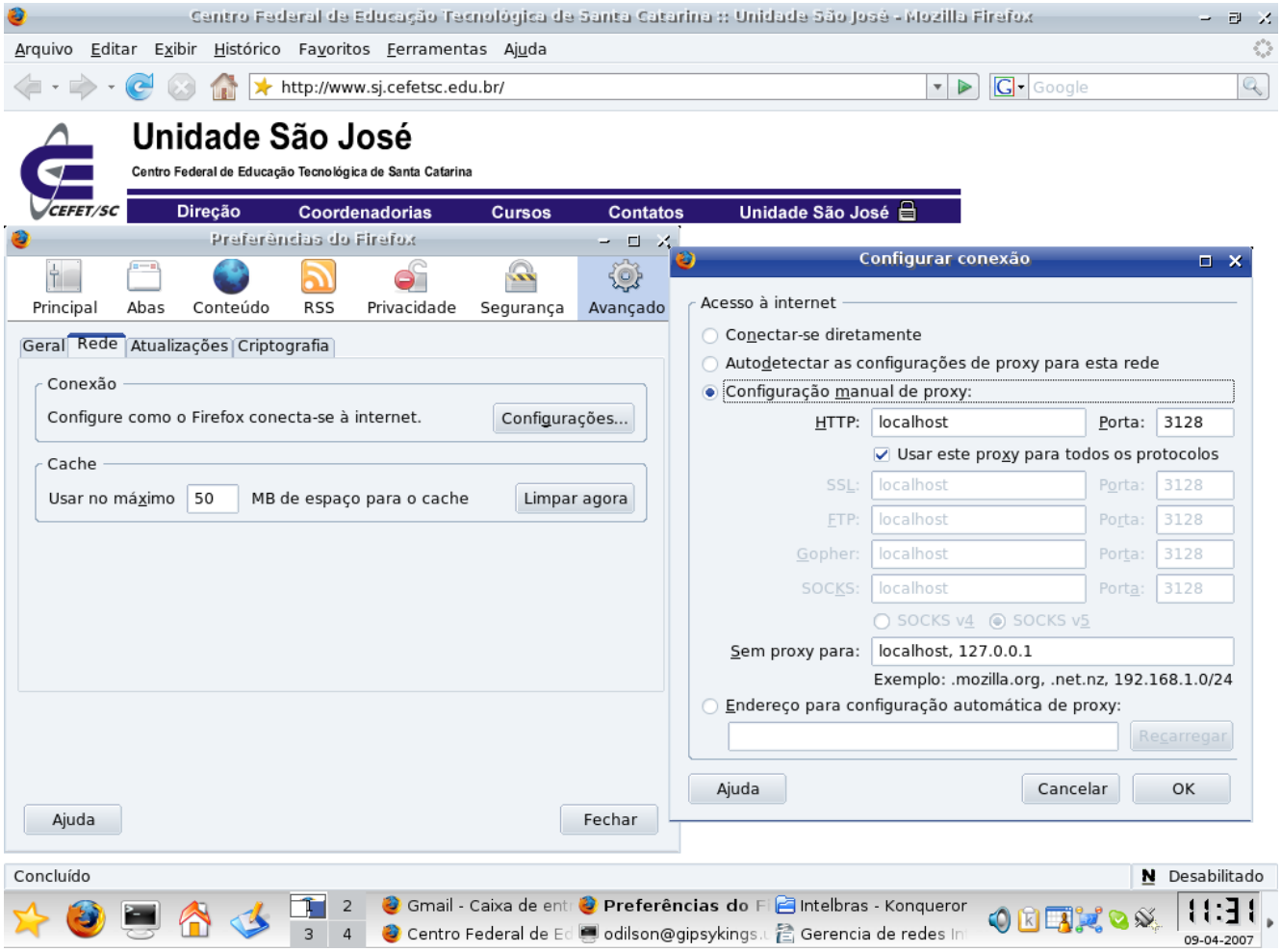


Ilustração 7: Configuração manual do proxy no Firefox

Em seguida acesse o site http://rpm.pbone.net/ e baixe algum arquivo, por exemplo amsn. Meça o tempo.

Vasculhe o diretório /var/spool/squid, por exemplo com du -s /var/spool/squid/*, e procure por um diretório/arquivo de mais ou menos o tamanho que você baixou.

Baixe novamente o mesmo arquivo e do mesmo lugar. Meça o tempo e compare com o anterior. Se tudo correu bem a velocidade agora foi muito maior, já que o navegador buscou o arquivo no próprio disco.

13 Servidor SMB, *Server Message Block*, Samba

13.1 Introdução

O SAMBA é um software criado por Andrew Tridgell, que veio para facilitar a

integração do mundo UNIX e o mundo Windows, integrando-os por meio do protocolo SMB (*Service Message Blocks*). Tem como função principal o compartilhamento de arquivos e impressoras com a família Windows©.

Um domínio Windows é um conjunto de computadores que residem na mesma sub-rede e pertencem ao mesmo grupo de trabalho, e um deles atua como controlador de domínio.

O PDC, *Primary Domain Controller*, é o controlador de domínio primário, onde está contido o banco de dados SAM, *Security Account Manager*, que é o banco de dados dos usuários do domínio Windows. O SAM é usado para validar os usuários no domínio. As mudanças, que por ventura ocorrerem, são propagadas para o BDC.

O BDC, *Backup Domain Controller*, é o reserva do controlador de domínio. Pode haver nenhum, um ou mais de um BDC num domínio mas um único PDC.

Pelo ambiente de rede Windows podemos navegar pelos computadores que estão disponíveis pela sub-rede e acessar seus recursos compartilhados.

O servidor WINS, *Windows Internet Name Server*, é uma implementação do servidor de nomes NetBIOS, *Network Basic Input/Output System.* O WINS é dinâmico: quando um cliente é iniciado são requeridos seu nome, endereço e grupo de trabalho. Este servidor manterá estas informações para futuras consultas e atualizações.

O Samba tem condições de exercer todos os papéis de uma rede Windows, com exceção do BDC. Mais especificamente o Samba pode ser: servidor de arquivo, servidor de impressão, PDC e servidor WINS, entre outros.

13.2 Instalação e configuração

Para instalar o samba basta executarmos o comando:

```
urpmi samba
```

O principal arquivo de configuração do samba é o /etc/samba/smb.conf. Este arquivo é dividido em duas secções, global e compartilhamentos. Devemos editá-lo e modificarmos/acrescentarmos as diretivas segundo o modelo abaixo:

```
[global]
    workgroup = redesX          # nome do grupo de trabalho ou do domínio
```

Para um caso real devemos nos preocupar ainda com mais mais algumas diretivas, as mais importantes são:

```
[global]
    security = user             # Modo de segurança: user, share
    encrypt passwords = yes     # Criptografia de senha: sim, não
```

```
        smb passwd file = /etc/samba/smbpasswd   # arquivo com senhas SMB
        load printers = yes              # Compartilhamento de impressora: sim, não
        printcap name = cups             # Servidor de impressão, cups ou lprng
    [homes]
        .....                            # mantenha o original
```

Para criarmos outros compartilhamentos devemos criar "contêiners", abaixo do [homes], com a seguinte sintaxe:

```
    [software]                           # Nome do compartilhamento
        comment = Softwares              # Comentário
        path = /dados/software           # O diretório a ser compartilhado
        guest ok = no                    # Convidado: sim, não
        public = no                      # Público: sim, não
        writable = yes                   # Permissão de escrita: sim, não
        browseable = yes                 # Navegação: sim, não
        create mode = 0555               # Tipo de permissão dos arquivos criados
        directory mode = 0555            # Tipo de permissão dos diretórios criados
        veto files = /*.mp3/             # Arquivos que não poderão ser salvos
        valid users = @admin             # Grupo (de usuários) que tem acesso
        force group = admin              # Grupo atribuído ao criar arquivos/diretórios
```

Como o padrão de senhas do SMB é diferente do Linux devemos criar as senhas "SMB" para os usuários de nossa máquina. Isto é feito com o comando:

```
smbpasswd -a usuario
```

Após a configuração devemos iniciar ou reiniciar o serviço com o comando:

```
service smb restart
```

13.3 Testes

Para fazermos os testes devemos ir à máquina Windows, se houver livre, se não na máquina do professor e fazer um mapeamento de rede apontando para //redes**X**/user. Fazer testes criando/copiando/removendo arquivos e diretórios.

14 Servidor NFS *Network File System*

14.1 Introdução

NFS é o sistema nativo Linux (Unix) para compartilhamento de arquivos em rede local. Permite que os diretórios remotos (servidor) sejam montados localmente (cliente) passando a impressão ao usuário de que o sistema de arquivos é local.

A segurança aos arquivos e diretórios é dada pelo permissionamento de arquivos e diretório padrão do Linux, sobreposto à uma "máscara" configurada no servidor de arquivos. Deve-se tomar o cuidado para garantir que todos os usuários tenham a mesma identificação (UID e GID) no servidor e cliente para não gerar "furos" na segurança de arquivos.

14.2 Instalação e configuração

Para instalar o servidor NFS usamos o comando

```
urpmi nfs-server
```

A configuração dos diretórios a serem exportados é feita por meio do arquivo /etc/exportfs, que por padrão não existe e deverá ser criado com o primeiro compartilhamento. O formato deste arquivo é bastante simples. Nele devem ser informados todos os diretórios, um por linha, a serem exportados seguindo o formato:

diretório [cliente(s) opções]

Onde diretório é o próprio diretório a ser exportado/compartilhado. Se informarmos simplesmente o diretório, todos as máquinas terão permissão de escrita e leitura no dito diretório.

Em cliente deve pode ser informado:

- o nome da máquina cliente ou ip
- curingas de domínio como *uned.sj, todas as máquinas na uned.sj
- pares de endereço ip/máscara, 192.168.2.0/24.
- *, qualquer máquina

Opções pode ser ro (*read only*) rw (*read and write*) e *no_root_squash*. *Nesta última* o root do cliente passa a ter permissões de root no servidor. Não recomenda-se o uso desta opção a não ser entre servidores e com muito cuidado. Devemos observar que se compartilharmos um diretório com opção rw e as permissões do diretório em si são somente leitura o usuário terá permissão de somente leitura. A regra geral é que o que vale é a permissão mais restritiva.

Exemplos reais:

/home 192.168.2.0/24 (rw) # a rede 192.168.2.0 terá acesso para escrita e leitura ao diretório /home.

/usr 192.168.2.7 (ro) 192.168.2.1 (rw,no_root_squash) # a máquina 192.168.2.7 poderá ler e a máquina 192.168.1.1 ler e escrever e o usuário root do cliente será "replicado", tendo permissões de root no servidor.

/var/www/html www.sj.cefetsc.edu.br (rw) # a máquina www.sj.cefetsc.edu.br terá acesso de leitura e escrita ao dito diretório.

Após definirmos o que queremos compartilhar devemos informar ao sistema as nossas atualizações com os comandos:

```
service nfs start

exportfs -a
```

14.3 Testes

Para verificar os compartilhamentos atuais em um determinado servidor usamos o comando:

```
showmount --exports servidor
```

Podemos fazer isto com nossa própria máquina trocando servidor pelo nosso ip. O comando retornará a listagem de todos os diretórios compartilhados.

Agora podemos montar o diretório compartilhado no cliente. Para fins de testes o cliente pode ser nossa própria máquina. Sendo assim criamos um diretório com o comando:

```
mkdir nfs
```

e montamos o compartilhamento neste diretório com o comando:

```
mount 192.168.2.X:/usr nfs
```

Podemos conferir listando o conteúdo do diretório nfs e/ou com o comando df.

15 Servidor DHCP *Dynamic Host Configuration Protocol*

15.1 Introdução[4]

Toda máquina que for participar de uma rede, deve primeiro, ter um endereço IP. Em uma rede pequena (até 20 máquinas), a tarefa de configurar IPs é relativamente simples. Mas em uma rede grande com centenas de máquinas, esta tarefa de endereçamento torna-se trabalhosa.

Para facilitar as coisas, foi criado um mecanismo de endereçamento automático de IP para máquinas em uma rede TCP/IP: o DHCP (*Dynamic Host Configuration Protocol* – Protocolo de configuração de máquinas dinâmico).

Um servidor DHCP pode facilitar muito a vida do administrador da rede. Dentre as configurações de serviços que podem serpassadas ao host cliente por dhcp são:

- Endereçamento IP, máscara de subrede, Gateway, Servidor(es) DNS, nome de host e/ou de domínio;
- Servidores e domínio NIS (autenticação);
- Servidores WINS (para redes Microsoft®);
- Servidores NTP (Hora);
- Imagens de boot para Terminais burros;

Como podemos observar, tudo o que é necessário para que uma máquina esteja em condições de ingressar em uma rede e usufruir de tudo o que ela possa oferecer, o DHCP se faz útil para sua configuração automática.

15.2 O protocolo DHCP

Entenda, com a explicação a seguir, como funciona o protocolo DHCP.
a) DHCP Discover – Quando uma máquina é ligada, ela tem um serviço (daemon) cliente do DHCP configurado para localizar o servidor neste momento. Este cliente DHCP envia um pacote UDP com destino à porta 67 do servidor chamado "***DHCP Discover***". Este pacote *broadcast* tem o endereço IP de destino 255.255.255.255 e mac address de destino ff:ff:ff:ff:ff:ff – Ilustração 8.

4 http://marcio.katan.googlepages.com/dhcp-mandriva

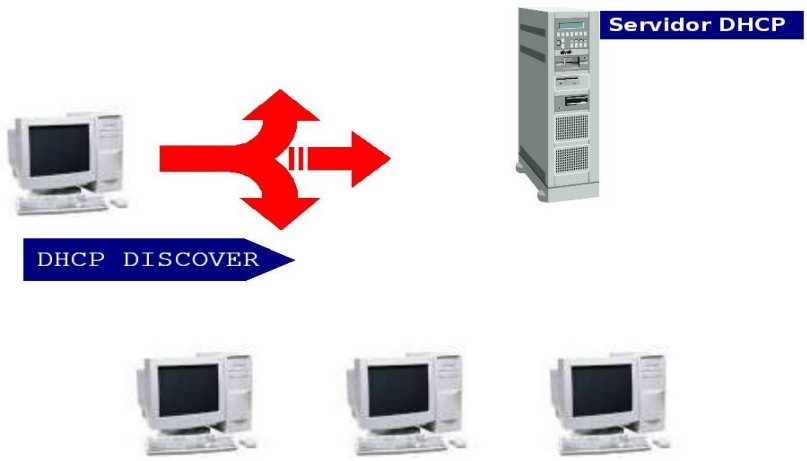


Ilustração 8: DHCP Discover. Pacote enviado pela estação

b) DHCP Offer – O servidor ao receber o referido pacote em sua porta ethernet, irá analisá-lo e, em sua tabela de IPs, reservar um endereço e preparar um pacote de resposta ao cliente solicitante. Este pacote de resposta chama-se DHCP Offer – Ilustração 9.

Ilustração 9: DHCP Offer. Resposta do Servidor DHCP à estação

O único meio de a estação cliente saber que o pacote DHCP Offer se destina à ela, é através do mac address.

c) DHCP Request – O cliente ao receber o pacote do servidor, decide se aceita a configuração oferecida pois pode receber mais de uma oferta. Em caso positivo, retorna um novo pacote ao servidor, comunicando o aceitamento da oferta. Este pacote chama-se DHCP Request - Ilustração 10.

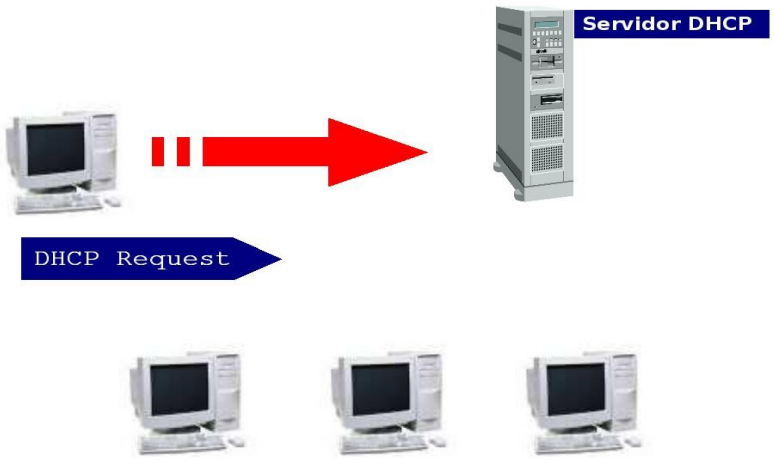


Ilustração 10: DHCP request. Confirmação da estação

d) DHCP Ack – Para finalizar a "conversação" entre cliente e servidor DHCP, este finaliza (efetiva) o aluguel (lease) do endereço ao cliente em sua tabela de IPs, e envia àquele, um pacote DHCP Ack para que ele ajuste suas configurações – Ilustração 11.

Ilustração 11: DHCP Ack. Confirmação do lease do endereço pelo Servidor

15.3 Instalação e configuração

Para instalarmos o servidor DHCP devemos executar o seguinte comando:

```
urpmi dhcp-server
```

Como primeiro passo da configuração devemos copiar o arquivo modelo de configuração com o comando:

```
cp /etc/dhcpd.conf.sample /etc/dhcpd.conf
```

Agora editamos o arquivo /etc/dhcpd.conf de acordo com o modelo:

```
ddns-update-style none;         # Esta opção serve para indicar se o servidor DNS será atualizado quando um aluguel de ip for solicitado.

subnet 192.168.0.0 netmask 255.255.255.0 {    # Todas as diretivas entre as chaves ({}) serão explicitamente aplicadas aos clientas da rede (subnet) declarada.

    option routers 192.168.2.1;      # Roteador padrão

    option subnet-mask 255.255.255.0; # Máscar de rede

    option domain-name "redesX.edu.br";       # Nome do domínio
```

```
        option domain-name-servers mX.redesX.edu.br; # Servidor DNS
        range dynamic-bootp 192.168.0.128 192.168.0.254;    # A faixa de
endereços que serão alugados
        default-lease-time 21600;       # Tempo padrão de aluguel. Após este
tempo o cliente tenta alugar novamente o mesmo ou um novo endereço.
        max-lease-time 43200;           # Tempo máximo de aluguel. Se este
tempo for excedido o cliente sairá de rede.
        host novo {  # Podemos fornecer um ip fixo a um determinado cliente
que receberá o nome "novo".
                hardware ethernet 12:34:56:78:AB:CD;       # O endereço mac do
cliente
                fixed-address 192.168.2.100;    # O ip que será fixado a ele.
        }
}
```

No arquivo de configuração poderíamos ter a declaração de mais de uma sub-rede (subnet), sendo que poderíamos ter parâmetros globais, declarados acima da sub-redes, e parâmetros específicos, dentro das sub-redes.

Após a configuração do arquivo (re)iniciamos o servidor com o comando:

```
service dhcpd restart
```

15.4 Testes

Para fazermos um teste podemos usar o comando dhclient eth? na máquina sobressalente, se houver, se não na máquina do professor. Se o servidor estiver corretamente configurado este alugará algum ip e demais características para esta máquina.

16 Servidor FTP *File Transfer Protocol*

16.1 Introdução

FTP significa Protocolo de Transferência de Arquivos, e é uma forma bastante rápida e versátil de transferir arquivos e diretórios, sendo uma das mais usadas na internet.

Pode referir-se tanto ao protocolo quanto ao programa que implementa este protocolo (neste caso, tradicionalmente aparece em letras minúsculas, por influência do programa de transferência de arquivos do Unix).

A transferência de dados em rede de computadores envolve normalmente transferência de arquivos e acesso a sistemas de arquivos remotos (com a mesma interface usada nos arquivos locais). É o padrão da pilha TCP/IP para transferir arquivos, é um protocolo genérico independente de hardware e do sistema operacional e transfere arquivos por livre arbítrio, tendo em conta restrições de acesso e propriedades dos arquivos.

16.2 Instalação e configuração

Para instalar o servidor FTP devemos primeiramente escolher um dos servidores disponíveis na distribuição dentre proftpd, pure-ftpd, vsftpd e wu-ftpd. Um dos mais usados na distribuição Mandriva é o proftpd. Cabe salientar que todos são similares na funcionalidade e configuração. Para instalá-lo basta executarmos o comando:

```
urpmi proftpd
```

O Proftpd vem absolutamente pronto para uso, não sendo necessária nenhuma configuração preliminar. Mas se desejarmos alguma configuração especial devemos editar o arquivo /etc/proftpd.conf.

Agora devemos (re)iniciar o serviço com o comando:

```
service proftpd restart
```

16.3 Testes

Para testar podemos usar nossa própria máquina como cliente. Para isto basta executarmos o comando:

*ftp 192.168.2.**X***

E surgirá um prompt com algo parecido com:

> Connected to 192.168.2.1.
>
> 220 (vsFTPd 2.0.2)
>
> 530 Please login with USER and PASS.
>
> 530 Please login with USER and PASS.
>
> KERBEROS_V4 rejected as an authentication type
>
> Name (192.168.2.1:odilson):

Então devemos informar o usuário ou <Enter> para o padrão, nesta caso odilson. Em seguida será requisitada a senha e após esta estaremos conectado no servidor remoto e poderemos usar praticamente todos os comandos normais do shell para

visualizar, criar, modificar ou apagar arquivos e diretórios. Lembramos que todos os comandos podem ser executados no servidor remoto ou na máquina local, neste caso iniciando o comando com "!". Para transferências de arquivos usa-se o comando "put arquivo", para enviar da máquina local ao servidor, e "get arquivo", no caso contrário. Para sairmos da aplicação digitamos o comando "bye".

17 Servidor SSH *Secure Shell* com OpenSSH

17.1 Introdução[5]

Em informática, o *Secure Shell* ou SSH é, simultaneamente, um programa de computador e um protocolo de rede que permite a conexão com outro computador na rede, de forma a executar comandos de uma máquina remota. Possui as mesmas funcionalidades do TELNET, com a vantagem da conexão entre o cliente e o servidor ser criptografada.

O SSH faz parte da suíte de protocolos TCP/IP que torna segura a administração remota de um servidor Linux/Unix.

O scp (*Secure Copy*) é uma maneira segura de fazer cópias de arquivos e diretórios usando o protocolo SSH.

17.2 Instalação e configuração

Para instalarmos o servidor SSH devemos executar o seguinte comando:

```
urpmi openssh-server ou urpmi ssh
```

Por padrão o servidor OpenSSH já vem completamente configurado, não sendo necessário nenhuma ajuste de configuração para as operações padrão. Mas, se quisermos fazer alguns ajustes devemos editar o arquivo /etc/ssh/sshd_config. A recomendação para este arquivo é descomentar somente o que pretende-se mudar do padrão. Se descomentarmos uma linha e deixarmos o valor padrão podem ocorrem instabilidades no serviço. Os principais parâmetros que podem ser modificados são:

> PermitRootLogin no # yes ou no. O usuário root poderá abrir um conexão ssh diretamente? Por questões de segurança recomenda-se deixar no, logar como usuário normal e em seguida dar o comando su.
>
> X11Forwarding yes # yes ou no. Se no servidor e cliente existirem as bibliotecas gráficas ativas o usuário poderá executar um programa em modo gráfico remotamente, sendo que o processo estará rodando no servidor e a janela será exibida no terminal do cliente.

5 http://pt.wikipedia.org/wiki/SSH

Além dos parâmetros “normais” podemos acrescentar alguns como por exemplo:

AllowUsers root fulano beltrano # Se esta diretiva existir somente os usuários listados poderão abrir conexão ssh.

DenyUsers root fulano beltrano # Se esta diretiva existir os usuários listados NÃO poderão fazer conexão ssh.

AllowGroups grupo # Idem AlloUsers

DenyGroups # Idem DenyUsers

Feitas as configurações podemos (re)iniciar os serviço com o comando:

```
service sshd restart
```

17.3 Testes

Para testar podemos usar nossa própria máquina como cliente. Para isto basta executarmos o comando:

*ssh usuario@192.168.2.**X***

Com isto abriremos uma conexão com o servidor, podendo executar todos os comandos, como se estivéssemos logados localmente.

Podemos também fazer cópias de arquivos com o scp. Por exemplo:

```
scp -r usuario@192.168.2.X:diretorio ./
```

Com este comando copiamos recursivamente o diretório “diretorio” do servidor 192.168.2.X para o diretório corrente local.

18 Webmin

18.1 Introdução[6]

O Webmin é um gerenciador de sistema baseado numa interface web. Com este utilitário você pode administrar sua(s) máquina(s) pela rede através de um navegador comum. Ele é bem completo e tem módulos para configuração de várias e várias coisas. É uma mão e tanta para os administradores de sistema.

Algumas das tarefas que você pode fazer com o Webmin atualmente:

- Mudar senhas, configurar o crontab, configurar scripts de inicialização, backup, configuração do pam, quotas, gerência de processos, pacotes, usuários e grupos.

6 http://www.devin.com.br/eitch/webmin/

- Configura e administrar servidores majordomo, cvs, sendmail, qmail, postfix, fetchmail, jabber, samba, postgresql, proftpd, ssh, squid, wu-ftpd, apache, dhcp, dns bind, MySQL.
- Configura rede, exportações NFS, NIS, PPP, túneis SSL.
- Administração de impressoras, gerenciadores de boot, cd-roms, raid, partições, lvm, clustering.
- Além de outras coisas como shell via web, gerenciador de arquivos, módulos perl, etc.

Então dá pra ver que o sistema é bem completo né? E ele é também amplamente usado. Vamos através deste tutorial saber como instalar e configurar de um modo bem prático e direto.

18.2 Instalação e configuração

Para instalar e rodar o Webmin devemos executar os comandos:

```
urpmi webmin

service webmin start
```

Uma vez instalado ele já estará absolutamente pronto para o uso, para isto basta acessar com um navegador qualquer o endereço https://192.168.2.**X**:10000/. Ou seja uma conexão segura, https, na porta 10000 de seu servidor. Agora devemos informar o usuário e senha, que são as mesmas cadastradas em nossa máquina. Se desejamos fazer manutenção nos serviços o ideal é usar o próprio root.

No primeiro acesso teremos uma janela do tipo mostrado na Ilustração 12:

Ilustração 12: Primeira janela do Webmin

Como primeira configuração devemos alterara linguagem de apresentação clicando no ícone "*Change Language and Theme*". Escolhemos "*Portuguese (Brasilian) (PT_BR)*" setamos a opção "*Personal Choice*"e clicamos em "*Make Change*". Agora teremos a interface em português, Ilustração 13.

Agora podemos dar uma navegada nas diversas janelas, principalmente em "Servidores", onde termos acesso a todas as configurações dos servidores que já instalamos configuramos. Ou seja, podemos fazer a manutenção/configuração do nosso servidor remotamente, através de um navegador qualquer e de maneira bastante amigável.

Ilustração 13: Webmin em português

19 Referências bibliográficas

Tibet, Chuck V. Linux: Administração e Suporte. Novatec Editora. ISBN: 85-85184-95-7. 2001.

Ferreira, Rubens E. Linux: Guia do Administrador do Sistema. Novatec Editora. ISBN: 85-7522-038-1. 2003.

Hunt, Craig. Linux: Servidores de rede. Editora Ciência Moderna. ISBN: 85-7393-321-6. 2004.

Stanger, James; Lane, Patrick T.; Danielyan, Edgar. Rede Segura Linux. Editora Alta Books. ISBN: 85-88745-10-0. 2002.

Nemeth, Evi; Snyder, Garth; Seeebas, Scott; Hein, Trnt T. Manual de Administração do Sistema Unix. Editora Boojman. ISBN: 85-7307-979-7. 2002.

Página de Gerência de Redes do CEFETSC - Unidade São José. http://www.sj.cefetsc.edu.br/wiki/index.php/Ger%C3%AAncia_de_Redes_%28p%C3%A1gina%29

Sites de dicas Linux:

Multiterminais/Multiterminal com Ruby – Wikibooks - http://pt.wikibooks.org/wiki/Multiterminais/Multiterminal_com_Ruby

Index of /MPlayer/releases/codecs - http://www4.mplayerhq.hu/MPlayer/releases/codecs/
Roger Lovato Â» Webcam DSB-C110 no Mandriva Linux - http://www.roger.lovato.com.br/artigos/11
Piter Punk's HomePage - dicas - http://piterpunk.info02.com.br/dicas.html
AURELIO.NET - http://aurelio.net/
iMasters - Por uma Internet mais criativa e dinÃ¢mica - http://www.imasters.com.br/
Comunidade de profissionais, estudantes e mestres em tecnlogias e ferramentas voltadas para o desenvolvimento web.
Guia Foca GNU/Linux - http://focalinux.cipsga.org.br/guia/avancado/index.htm#contents
Linux Operating System and Linux Distributions - http://linux.about.com/About_Focus_on_Linux.htm
This site is your Internet destination for information and resources for the Linux operating system and various Linux distributions.
Linux-Tip.net - Remote access Mandriva 2007 Free using FreeNX - http://www.linux-tip.net/cms/content/view/254/26/
Linux-Tip.net, Linux-Tip.eu, Linux-Tip.com Linux Tips and Tricks, workshops, news and articles, Linux Security, NoMachine NX is a Terminal Server and remote access solution based on a comprising set of enterprise class open source technologies. NX makes it possible to run any graphical application on any operating system across any network connection at incredible speed.
Linux in Brazil: Novo artigo: AutenticaÃ§Ã£o com o PAM-LDAP - http://br-linux.org/news2/006653.html
PrÃticas de SeguranÃ§a para Administradores de Redes Internet - http://www.cert.br/docs/seg-adm-redes/
PrÃticas de SeguranÃ§a para Administradores de Redes Internet
PÃgina principal - Nagios-BR - http://nagios-br.sourceforge.net/wiki/index.php/PÃgina_principal
RelatÃ³rios em Java - http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/documentviews/relatorios.htm
Easy Urpmi - http://easyurpmi.zarb.org/
Jarbas Teixeira - Usando o URPMI - Parte 01 - http://www.imasters.com.br/artigo/5127/linux/usando_o_urpmi_-_parte_01/
Um pacote trÃs tambÃ©m informaÃ§Ãµes sobre prÃ©-requisitos para ser instalado ou as chamadas dependÃªncias.
Jarbas Teixeira - URPMI - Parte 02 http://www.imasters.com.br/artigo/5127/linux/usando_o_urpmi_-_parte_02/
Vamos estudar mais detalhes de como usar o URPMI e configurar mÃdias via http.
Comunidade LinuxBSD - Montando partiÃ§Ãµes no fstab - http://www.linuxbsd.com.br/phpLinuxBSD/modules/artigos_tecnicos/fstab.htm

Aldeia NumaBoa - BIND (DNS) na jaula - http://www.numaboa.com/content/view/424/167/1/0/
Aldeia NumaBoa - um portal diferente em PortuguÃªs do Brasil

Sites com dicas do Amanda

ConsideraÃ§Ãµes bÃ¡sicas sobre backup - http://www.openit.com.br/freebsd-hb/backup-basics.html
Backup em HD com Amanda â€" Projeto CyberShark.net - http://www.cybershark.net/tutoriais/amanda_hd
AMANDA, The Advanced Maryland Automatic Network Disk Archiver - http://www.amanda.org/
[Dicas-L] Amanda: Backup DistribuÃdo - http://www.dicas-l.com.br/dicas-l/20000711.php
Chapter 13. How to use the Amanda file-driver - http://www.amanda.org/docs/howto-filedriver.html
Amanda Faq-O-Matic: How to configure for tapeless operation? - http://amanda.sourceforge.net/fom-serve/cache/191.html

Sites com dicas do Postfix e listas de discussão

Postfix : INTEGRANDO O POSTFIX COM O CLAMAV - http://www.unitednerds.org/thefallen/docs/index.php?area=Postfix&tuto=Clamav-gsoares
Filtros para o Postfix | LinuxMan - http://www.linuxman.pro.br/node/4
The Postfix Home Page - http://www.postfix.org/
Integrating amavisd-new Into Postfix For Spam- And Virus-Scanning | HowtoForge - Linux Howtos and Tutorials - http://www.howtoforge.com/amavisd_postfix_debian_ubuntu
Linux: GNU Mailman email list installation and configuration - http://www.yolinux.com/TUTORIALS/LinuxTutorialMailman.html
Mailman Tutorial. The YoLinux portal covers topics from desktop to servers and from developers to users
GNU/Mailman Administrator's Manual - http://radonio.iq.usp.br/outros/mailman/manual_mailman.html
Tutoriais/FreeBSD/Mailman - UnderLinux Wiki - http://under-linux.org/wiki/index.php/Tutoriais/FreeBSD/Mailman
Sistema de correio eletrÃ´nico - WikipÃ©dia - http://pt.wikipedia.org/wiki/Sistema_de_correio_eletrÃ´nico
Linux: Como configurar o servidor de correio eletrÃ´nico Postfix [Artigo] - http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=189
Viva o Linux - Porque nÃ³s amamos a liberdade! Seja livre, use Linux. A maior e melhor comunidade para se aprender Linux do Brasil, artigos, dicas, forum e muito mais.
BOM - Correio EletrÃ´nico - http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch11.html