# Administração de Sistemas Operacionais Unix (2)

Centro de Informática de Ribeirão Preto - CIRP

MSc. Eng. Ali Faiez Taha

# Sumário

# Capítulo 1

# Network File System - NFS

## 1.1 Introdução

O NFS, Sistema de Arquivos em Rede, permite que sistemas de arquivos sejam compartilhados pela rede. É transparente ao usuário.

Foi lançado pela Sun em 1985. Originalmente, foi implementado como um Sistema de Arquivos para máquinas sem disco.

Todos os UNIX oferecem uma versão de NFS.

O NFS roda sobre o protocolo RPC da Sun (Remote Procedure Call), que define uma maneira independente de sistema para que processos se comuniquem em uma rede. Os protocolos UDP e TCP podem ser usados no NFS.

Originalmente o NFS utilizava o UDP.

O Sistema de Arquivos em Rede tem três importantes características:

* Possibilita o compartilhamento de arquivos sobre uma rede local.

* Funciona muito bem.

* Possibilita diversos problemas de segurança que são bem conhecidos por intrusos, e podem ser explorados na obtenção de acesso (leitura, gravação e remoção) de todos os arquivos de um sistema.

## 1.2 Recursos necessários:

- Programa **portmap** (ou **rpc.portmap**): mapeador de portas **DARPA** para números de programas **RPC (Remote Procedure Call)**

- Daemons **nfsd** e **mountd**

Arquivo de configuração do **nfsd** : **/etc/exports**

Inicialização dos serviços : **/etc/init.d/**

**Pacotes necessários :**

**nfs-server, nfs-utils, nfs-common, portmap, nmap, rpcinfo**

Basicamente o que você precisa é de editar o arquivo **/etc/exports** e colocar as seguintes linhas:

**/home/ ip_da_maquina_remota(rw,no_root_squash)**

O que temos aqui é o seguinte:

* **/home/** - Este é o diretório que será exportado.

* **ip_da_maquina_remota** - Este é o ip da máquina que terá acesso ao dire tório exportado. Pode-se usar o coringa "*" para liberar para qualquer máquina.

* (rw,no_root_squash) - São as opções que usadas aqui, onde **rw** dá permissão de leitura e gravação para os IPs especificados e "**no_root_squash**" dá permissão de acesso ao **root** remoto também.

Pode-se escolher outras opções, tais como **somente leitura (ro).**

Há ainda formas melhores de incluir diversas máquinas no arquivo **exports**. Pode-se, por exemplo, usar grupos de rede caso se esteja utilizando **NIS**, ou subredes **IP** como máquinas que têm permissão para montar algo. Porém é necessário considerar que é possível obter acesso ao servidor de forma não autorizada caso se utilize autorizações tão genéricas.

Os **Daemons mountd** (**rpc.mountd**) e **nfsd** (**rpc.n fsd**) vão ler o arquivo **/etc/exports**

**Inicializar os serviços, no servidor :**

**/etc/init.d/nfs-server start**
**/etc/init.d/portmap start**
**/etc/init.d/mountd start**

## 1.3 Alternativa para controle do servidor NFS :

Caso se edite o **/etc/exports** deve-se estar seguro de que os programas **nfsd** e **mountd** fiquem cientes destas alterações. A forma tradicional é através da execução do comando **exportfs**.

Muitas distribuições Linux não possuem o programa **exportfs**. Caso este seja o seu caso, instale o seguinte programa na máquina local:

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo re-exportando sistemas de arquivos
```

O programa acima deve ser salvo, como por exemplo como **/usr/sbin/exportfs**, e deve ser executado o comando **chmod a+rx exportfs**. Agora, toda vez que uma alteração for efetuada, deve-se executar o comando **exportfs** a seguir, com privilégios de superusuário.

Agora deve-se checar se **mountd** e **nfsd** estão sendo adequadamente executados.

Inicialmente deve-se executar o comando **rpcinfo -p** . Ele deverá apresentar uma saída similar a:

```
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100005 1 udp 745 mountd
100005 1 tcp 747 mountd
100003 2 udp 2049 nfs
100003 2 tcp 2049 nfs
```

Caso se obtenha uma mensagem similar a **rpcinfo: não foi possível contactar o po rtmapper: RPC**: **Erro no sistema remoto** ou algo similar, possivelmente o **portmapper** não esteja sendo executado.

Caso se obtenha uma mensagem similar a **Nenhum programa remoto registrado,** então, ou o **portmapper** não deseja falar com a máquina local ou existe existe algum erro.

Pode-se finalizar o **nfsd, o mountd e o portmapper** e tentar reiniciá-los nesta ordem novamente.

Para tanto, use os comandos:

**/etc/init.d/nfsd stop**
**/etc/init.d/mountd stop**
**/etc/init.d/portmapper stop**

Para o Linux Debian os arquivos são :
**/etc/init.d/nfs-common, /etc/init.d/nfs-user.server**
**/etc/init.d/portmap e /etc/init.d/mountnfs.sh**

Após verificar os serviços disponíveis segundo o **portmapper**, pode-se fazer uma checagem através do comando **ps**.

O **portmapper** continuará a reportar um serviço, mesmo após o programa responsável ter sido finalizado com erro, por exemplo.

Então um comando **ps** poderá ser a maneira mais simples de descobrir que programas estão efetivamente sendo executados.

Evidentemente, será necessário modificar os arquivos **rc, ou /etc/rcN.d,** do sistema para inicializar o **mountd** e o **nfsd**, assim como o **portmapper**, quando o sistema operacional for carregado.

É muito provável que estes programas já existam na máquina local e que se deva somente descomentar as seções adequadas ou ativá-los nos níveis de execução corretos.

Páginas de manual online que já devem ter sido visitadas até agora: **portmap, mo untd, nfsd, e exports.**

## 1.4 Configuração do Cliente NFS:

Inicialmente é necessário ter um **KERNEL** com o suporte a sistemas de arquivo **NFS** compilado ou como um módulo. Isso deve ser configurado antes da compilação do **KERNEL**. No Debian linux, verifique com o comando **modconf.**

Pode-se, agora, na linha de comandos do superusuário, informar o comando de montagem apropriado e o sistema de arquivos estará disponível. Continuando com o exemplo anterior, deseja-se montar **/mnt/servidor/local** a partir de **servidor**.

Isso deve ser feito através do seguinte comando:

**mount -t nfs -o rsize=1024,wsize=1024 servidor:/mnt/servidor/local /mnt**

O sistema de arquivos está agora disponível sob **/mnt** e pode-se acessá-lo atravé s do comando **cd**, assim como verificar o seu conteúdo através do comando **ls,** e observar os arquivo individualmente.

Pode-se perceber que ele não é tão rápido quando um sistema local, mas muito mais amigável que o uso do **ftp**.

Se, ao invés de montar um sistema de arquivos, o comando **mount** apresente uma mensagem de erro como:

**mount:servidor:/mnt/servidor/local falhou,** razão fornecida pelo servidor:
**Permissão negada**, então o arquivo **/etc/exports** contém algum erro.

Caso ele informe **mount clntudp_create: RPC: Programa não registrado** isso significa que os programas **nfsd** ou **mountd** não estão sendo executados no servidor.

Para desmontar o sistema de arquivos basta usar o comando:

**umount /mnt**

Para que um sistema de arquivos **nfs** seja montado na inicialização do sistema operacional, deve-se editar o arquivo **/etc/fstab** da forma usual. No caso de nosso exemplo, deve-se adicionar a seguinte linha:

# dispositivo pto.montagem tipo_sist_arqs opções dump ordem verif. ...
**servidor:/mnt/servidor/local /mnt nfs rsize=1024,wsize=1024 0 0 ...**

## 1.5 Opções de Montagem

Há algumas opções que devem ser consideradas. Eles definem a forma como o cliente **NFS** lida com uma queda do servidor ou da rede. Um dos aspectos mais interessantes sobre **NFS** é que ele lida com estas situações com elegância, desde que o cliente esteja corretamente configurado. Há dois tipos distintos de parâmetros de tratamento de falhas:

**soft**

O cliente **NFS** reporta um erro ao processar o acesso a um arquivo localizado em um sistema de arquivos montado via **NFS**.

Alguns programas trabalham bem com esse erro reportado, outros não. **Esta opção não é recomendada**.

**hard**

O programa que acessa um arquivo em um sistema de arquivos montado via NFS irá travar sempre que o servidor não responder. O processo não pode serinterrompido ou finalizado.

Quando o servidor **NFS** estiver novamente ativo, o programa irá continuar a partir do ponto onde parou. Isso é provavelmente o que se deseja. Recomenda-se o uso do parâmetro **hard,intr** em todos os sistemas de arquivos montados via **NFS**.

A partir do exemplo anterior, esta seria a entrada no arquivo **/etc/fstab:**

# dispositivo pto.montagem tipo_sist_arqs opções dump ordem verif. ...
**servidor:/mn/servidor/local /mnt nfs rsize=1024,wsize=1024,hard,intr 0 0 ...**

## 1.6 Otimizando o NFS

Normalmente, caso as opções **rsize** e **wsize** sejam especificadas, o **NFS** irá ler e gravar blocos de 4096 e 8172 bytes.

Algumas combinações de **kernel do Linux** e placas de rede não podem lidar com blocos grandes e não podem ser otimizados.

Então vamos tentar descobrir como encontrar os parâmetros **rsize** e **wsize** que funcionem da maneira mais otimizada possível.

É possível testar a velocidade das opções com um simples comando.

Dado o comando **mount** conforme descrito acima, logo temos acesso de gravação ao disco, podendo executar um teste de performance de gravação seqüencial:

**time dd if=/dev/zero of=/mnt/testfile bs=16k count=4096**

Este comando criará um arquivo de 64 Mb de bytes zerados (que deve ser grande o suficiente para que o **cache** não altere significativamente a performance. Pode ser usado um arquivo maior caso o sistema local tenha muita memória).

Neste caso, o importante é medir o tempo de **relógio** e o tempo efetivamente ga sto na conexão. Após, pode-se testar a performance da leitura ao se ler o arquivo de volta:

**time dd if=/mnt/testfile of=/dev/null bs=16k**

Isso pode ser feito algumas vezes. Após, deve-se executar o comando **mount** e **umount** novamente com tamanhos maiores em **rsize** e **wsize**. Eles devem ser provavelmente múltiplos de 1024, e não maior que 16384 visto que este é o tamanho máximo do NFS versão 2.

Exatamente após a montagem de um tamanho maior, acesse o sistema de arquivos montado através do comando **cd** e explore-o através do comando **ls**, para estar seguro que ele está funcionando perfeitamente.

Caso os parâmetros **rsize/wsize** sejam muito grandes, os sintomas não são muito óbvios. Um típico sintoma é uma lista incompleta dos arquivos produzida pelo comando **ls** e nenhuma mensagem de erro. Ou ao se ler um arquivo ele falha misteriosamente, sem mensagens de erro. Após definir que os parâmetros **rsize/wsize** funcionam perfeitamente deve-se executar os testes de performance.

**SunOS e Solaris** tem a reputação de funcionar muito melhor com blocos de 4096 bytes.

**Kernels** mais recentes do Linux (**desde o 1.3**) executam a leitura antecipada para rsizes maiores ou iguais ao tamanho de página da máquina. Em máquinas **Intel** o tamanho de página é de 4.096 bytes.

A leitura adiantada aumenta significativamente a performance de leitura do NFS. Ou seja, sempre que possível deve-se usar o rsize de 4.096 bytes em máquinas **Intel**.

Lembre-se de editar o arquivo **/etc/fstab** com os valores de **rsize/wsize** encontrados.

Uma sugestão para incrementar a performance de gravação do **NFS** é desabilitar o sincronismo de gravação do servidor.

A especificação **NFS** indica que a gravação **NFS** solicitada não pode ser considera da finalizadas antes dos dados serem gravados em um meio não volátil (normalmente o disco rígido). Isso restringe a performance de gravação de alguma forma, enquanto que gravaçõe s assíncronas irão aumentar a velocidade do NFS.

O servidor **Linux nfsd** nunca faz gravações síncronas uma vez que a própria implementação do sistema de arquivos não o faz, mas em servidores em sistemas operacionais diferentes isso pode aumentar a per formance através do seguinte parâmetro no arquivo exports:

**/dir -async,access=linuxbox**
ou algo similar.

Verifique a página de manual online da máquina em questão. Cabe salientar que esta opção aumenta o risco de perda de dados no caso de algum problema ocorrer antes da efetiva gravação dos dados.

Exercícios:

1 - Comparar os recursos disponibilizados por clientes FTP e NFS.

2 - Preparar um Cliente Linux para montar, via NFS, dois compartilhamentos distintos, de dois servidores NFS. O compartilhamento deve ser montado no **boot** de cada cliente.

3 - Compare os tempos de escrita e de leitura de arquivos usando:

a) Disk dump (**comando dd**) com **if=/dev/zero e of=/mnt/arqteste**

b) Disk dump com **if=/mnt/arqteste e of=/dev/null**

c) Leitura de um arquivo NFS. Gravação no cliente.

d) Gravação de um arquivo no servidor, via NFS.

4 - Para quê servem as opções **rsize e wsize ?** Como devem ser utilizadas ?

5 - Configure um servidor de modo a permitir o controle de compartilhamento NFS com:

a) **TCP WRAPPER**, com os arquivos **hosts.allow e hosts.deny**

b) Leitura e escrita em compartilhamento NFS

c) Compartilhamento de Subdiretórios

d) Controle a nível de usuários

6 - Quais são os usuários que usam o compartilhamento NFS. Explique os perigos que podem acontecer.

7 - Para quê servem as opções **hard,intr** no NFS ?

8 - Quais as diferenças entre NFS Síncrono e NFS Assíncrono ?

9 - Elabore um servidor dedicado de arquivos NFS.

10 - Como medir a performance de um servidor NFS ?

Referências bibliográficas:

http://www.linorg.cirp.usp.br/Guias.Conectiva/Guias_V.9.0/servidor/intranet.html#NFS

# Capítulo 2

# Servidor SAMBA

## 2.1 Introdução

Samba é um pacote de Software Livre, que permite que computadores Windows, ligados em rede, utilizem o Linux, ou Unix, como servidor de arquivos e impressão, como se eles estivessem acessando um servidor Windows (NT, 2000, XP, Server 2003).

Numa rede, é necessário compartilhar dados. O Samba permite que clientes Windows compartilhem arquivos através do **NetBios**.

Com o servidor Samba, é possível compartilhar diretórios, impressoras, acessar arquivos na rede, exatamente como em redes Microsoft.

Neste caso, o servidor pode ser um Unix rodando uma aplicação específica.

Características:

* Compatível com estações Windows (praticamente todas as versões) e servidores.

NT 4.0/2000/XP/Server 2003.

Entre servidores e estações Unix/Linux, a compatibilidade é ainda maior;

* Totalmente configurável, com a grande vantagem de centralizar esta configuração em um único arquivo: o **smb.conf**.

* Também é possível configurar o Samba remotamente, através de acesso seguro, com Browsers, e receber por **e-mail**, informações do estado do servidor ( para isso é necessário usar um **script** específico que busca informações nos arquivos de **log** e cria um arquivo que pode ser enviado via **e-mail**);

Site do Samba **(http://www.samba.org)**

## 2.2 Como instalar o SAMBA

Para instalar o Samba é necessário antes saber se o pacote obtido está no formato **DEB, RPM ou .tar (.tgz ou .tar.gz)**.

Se sua distribuição for Debian, utilize: **apt-get install samba smbclient smbfs** .

Se o pacote for um RPM, usado pela Red Hat, execute o comando: **rpm -ivh samba-3.0.x.rpm**, substituindo o **x** pelo número relativo a versão do pacote ou o nome correto do pacote, dependendo da sua distribuição.

O arquivo de configuração do SAMBA será o **smb.conf,** geralmente locallizado no diretório **/etc/.**

Este arquivo contém praticamente tudo o que é necessário para se fazer o compartilhamento de Diretórios e Impressoras, autenticação e controle de usuários, etc.

## 2.3 Exemplos de configuração

Exemplos de configuração do arquivo **smb.conf**:

```
# Samba config file created using SWAT
    # from localhost (127.0.0.1)
    # Date: 2004/06/20 14:46:29
    # Global parameters
    [global]
    workgroup = ADMUNIX
    netbios name = ADMUNIX
    server string = Servidor de Impressao
    interfaces = 143.107.200.X
    security = SHARE
    smb passwd file = /etc/samba/smbpasswd
    password level = 34
    log level = 1
    log file = /var/log/samba/log.%m
    max log size = 100
    logon script = logon.bat
    domain logons = Yes
    os level = 64
    preferred master = Yes
    domain master = Yes
    printing = cups
    read only = No
    hosts allow = 143.107.200.68 143.107.200.53

    [netlogon] comment = Netlogon service em ADMUNIX
    path = %H
    read only = Yes

    [lj6l] comment = Impressora LaserJet 6L
    path = /var/spool/samba
    create mask = 0700
    guest ok = Yes
    printable = Yes
    print command = /usr/bin/lpr
    lpq command = /usr/bin/lpq
    lprm command = /usr/bin/lprm
    printer name = lj6l
    oplocks = No
    share modes = No

    [homes] comment = "Diretorio Home de : %U"
    path = /home/%U
    create mask = 0644
    directory mask = 0700
    guest ok = Yes
```

## 2.4 Ferramentas para configuração

Como mostrado acima, o arquivo **smb.conf** deve ser editado e o **SAMBA** deverá ser reinicializado a cada modificação.

É comum encontrar ferramentas WEB para fazer as configurações do SAMBA.

O SWAT (Samba Web Administration Tool) e o WEBMIN são dois exemplos.

Para instalar o **swat**, no Debian, simplesmente use o comando:

**$> apt-get install swat**

O pacote será instalado e a porta TCP utilizada será a 901. Para acessar a ferramenta, o Browser deve apontar para :

**http://localhost:901**

No arquivo **/etc/inetd.conf** tem uma linha mostrando este serviço. Lembre-se de que para reiniciar o **swat**, o **superdaemon inetd** deve ser reinicializado.

Isso pode ser feito com o comando : **killall -HUP inetd.**

A linha a ser adicionada no **/etc/inetd.conf** é a seguinte:

**swat stream tcp nowait.400 root /usr/local/samba/bin/swat swat**

Pode-se colocar o **swat** para funcionar subordinaod ao **XINETD**, que substitui o **INETD.**

Para tanto, basta criar um arquivo **swat** no diretório **/etc/xinetd.d**, da seguinte forma:

```
service swat
{
port = 901
socket_type= stream
wait= no
only_from = localhost
user =root
server= /usr/local/samba/bin/swat
log_on_failure = USERID
disable = no
}
```

Acrescentar no arquivo **/etc/services** a seguinte linha:

**swat 901/tcp**

Para a utilização com o WEBMIN, basta instalar o WEBMIN e procurar pelo item **Samba Windows file sharing**, que pertence ao tópico **servers**.

## 2.5 SAMBA com Impressoras

Uma utilização comum, já citada, é o compartilhamento de Impressoras com o Samba e a administração dos recursos, controle de quotas de impressão, disponibilidade, etc.

A seguir é mostrado uma alternativa ao **smb.conf** que ilustra outra maneira de se usar os comandos para impressão.

O arquivo que vai gerenciar a impressão não é o padrão, **/usr/bin/lpr, lpq, lpc, etc.**

```
#======================= Global Settings =====================
[global]
message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &
passwd program = /usr/bin/passwd %u
dns proxy = no encrypt
passwords = yes socket
```

```
    options = TCP_NODELAY
    invalid users = root
    max log size = 1000
    preferred master = no
    hosts allow = 192.168.0.
    localhost obey pam restrictions = yes
    passwd chat = *Enter\snew\sUNIX\spassword:* %n\n *Retype\snew\sUNIX\spassword:*
%n\n
    workgroup = servidor
    server string = %h server (Samba %v) ;removed-by-linpopup-install
    message command = cat %s|smbclient -U "%f" -M %t
    syslog = 0
    netbios name = servidor
    log file = /var/log/samba/log.%m
    os level = 20
    socket address = 192.168.0.1

    [homes]
    browseable = no
    comment = Home Directories
    writeable = yes
    create mask = 0700
    directory mask = 0700

    [printers]
    path = /tmp
    browseable = yes
    writeable = yes create
    mode = 0700
    comment = All Printers
    printable = yes
    use client driver = yes
    print command = /etc/samba/imprime '%T' '%U' '%J' '%c' '%z' '%p' '%s'

    [geral]
    path = /home/geral
    writeable = yes
    create mode = 777
    directory mode = 777
    force group = root
    valid users = fulano,ciclano
    force user = root
```

A linha :

```
    print command = /etc/samba/imprime '%T' '%U' '%J' '%c' '%z' '%p' '%s'
```

faz o desvio de solicitações de impressão para o arquivo **/etc/samba/imprime**, que vai gerenciar a impressão.
O arquivo **/etc/samba/imprime** está assim:

```
#!/bin/sh
    # Script chamado via SAMBA apos o trabalho ser submetido para impressao
    # Parametros recebidos (linha de comando)
    # $1 - Data e hora: Enviados como um so parametro (%T)
```

```
# $2 - Login do usuario que submeteu o trabalho (%U)
# $3 - Nome do job informado pelo cliente Windows (%J)
# $4 - Numero de paginas informado pelo Windows (%c)
# $5 - Tamanho do arquivo informado pelo Windows (%z)
# $6 - Nome da impressora (%p)
# $7 - Nome do arquivo de spool (%s)
# Arquivo de log
# O diretorio do arquivo de log deve ter acesso 777
ARQLOG="/var/log/printing.log"
# Verifica e/ou cria o arquivo de log
if [ ! -f ${ARQLOG} ];
then touch ${ARQLOG}
chmod 777 ${ARQLOG}
echo "# Arquivo de log de trabalhos enviados via SAMBA" >> ${ARQLOG}
echo "# Criado em: $1" >> ${ARQLOG}
echo "# data hora login_usuario impressora nome_do_job numero_paginas tamanho_documento"
>> ${ARQLOG}
fi
# Obtem o tamanho do arquivo
# tamanho=`ls -al /tmp/$7|awk '{print $5}'`
# copias=`tail -c 126 /tmp/smbprn.000010.yVq96P|head -c 7`
# Faz o log do trabalho de impressao
echo "$1 '$2' '$6' '$3' $4 $5" >> ${ARQLOG}
# Cria instrucao SQL para insercao na tabela
echo "insert into trabalhos ( data, hora, usuario, impressora, descricao, paginas,
tamanho) values ( curdate(), curtime(), '$2', '$6', '$3', $4 , $5 )" > /tmp/imprime.sql
/usr/local/mysql/bin/mysql -uroot -psenha impressao < /tmp/imprime.sql
rm /tmp/imprime.sql
# Submete o trabalho ao sistema de impressao Linux
/usr/bin/lpr -P $6 $7
# Remove o arquivo de spool
rm $7
# FIM
```

O arquivo **extrato.sql** está assim:

```
select UPPER(impressora) as IMPRESSORA, " - ", UPPER(usuario) as USUARIO,
" - ", sum(tamanho) as TOTAL_USO from trabalhos where (month(data) = month(curdate()))
group by impressora, usuario order by impressora, TOTAL_USO desc; #select UP-
PER(impressora) as IMPRESSORA, sum(tamanho) as TOTAL_GERAL from tra-
balhos where (month(data) = month(curdate())) group by impressora;
```

O arquivo **imprime** envia para o banco de dados **MySQL** os dados referentes a : (data, hora, usuário, impressora, descrição, páginas, tamanho) de cada impressão. O arquivo **extrato.sql** faz a contabilidade, mostrando o número de impressões de cada usuário cadastrado no **MySQL**.

## 2.6 Referências bibliográficas

Muitos artigos descrevem como o SAMBA pode ser usado numa rede baseada em plataforma Windows, como clientes de domínios Windows, controladores de domínio, Primary Domain Controller (PDC), compartilhamento de simples impressoras, diretórios, etc.

**Os artigos podem ser encontrados nos seguintes locais :**

http://www.linuxrapido.org/modules.php?name=Sections&op=viewarticle&artid=83
http://www.rnp.br/newsgen/0211/linux_samba_windows.html
http://www.neurix.com.br/testes/servidor_de_arquivos.html
http://infsr.unijui.tche.br/~heini/freebsd/samba.html
http://www.linuxrapido.org/modules.php?name=Sections&op=listarticles&secid=3

**Tutorial sobre configuração do SAMBA:**

http://www.linorg.cirp.usp.br/Guias.Conectiva/Guias_V.9.0/servidor/intranet.html#IMPLEMENTA-SAMBA

**Samba com LDAP:**

http://www.unav.es/cti/ldap-smb/smb-ldap-3-howto.html
http://samba.idealx.org/samba-ldap-howto.pdf
http://www.solis.coop.br/modules/ldap/files/files/sambaldap.pdf
http://opensourceschools.org/article.php?story=2003060302590131

# Capítulo 3

# Serviços Telnet e SSH

## 3.1 Introdução

Um dos serviços de conexão remota com os servidores é o **telnet**. Bem antigo e bastante usado, já teve sua fase de glória e ajudou muitos usuários a se conectar em servidores locais e remotos. Os mais remotos servidores ficavam tão próximos graças ao **telnet**.

Usado para autenticar usuários, as senhas trafegavam pela rede de maneira totalmente insegura. Sem nenhum esquema criptográfico.

O **telnet** substitui o **rlogin (remote login)** e o **rcp (remote copy).**

Atualmente está em desuso e aconselha-se fortemente que não se use o **telnet** em conexões remotas ou locais.

O seu substituto é o **SSH**, que será descrito mais adiante neste capítulo.

## 3.2 Características do telnet :

- Conexão rápida (não utiliza transmissão de dados criptografada), recomendado para ambientes seguros.
- Possui uma versão com suporte a criptografia via **ssl**.
- Possui controle de acesso **tcpd** (usando **/etc/hosts.allow e /etc/hosts.deny**).
- A maioria dos Sistemas Operacionais trazem este utilitário por padrão como sistema de acesso remoto a máquinas UNIX.
- Suporte a terminais ANSI (cores e códigos de escape especiais para o console) e uma grande variedade de outros terminais.

* Apesar de poder trabalhar com **SSL (Socket Secure Layer)**, o uso do **telnet** ainda é desaconselhado.

Para o Debian Linux, os seguintes pacotes e utilitários **telnet** estão disponíveis:

**Pacotes:**

* **telnet** - Cliente telnet com suporte a autenticação.

* **telnetd** - Servidor telnet com suporte a autenticação.

* **telnet-ssl** - Cliente telnet com suporte a autenticação e ssl. Também suporta conexão a servidores telnet padrão quando o servidor não suporta **ssl**. Por padrão é tentada a conexão usando **ssl**, se esta falhar será assumida a transmissão em texto plano.

* **telnetd-ssl** - Servidor telnet com suporte a autenticação e ssl. Também suporta conexão de clientes telnet padrão (sem suporte a ssl).

**Utilitários:**

* **in.telnetd** - Servidor **telnet**

* **telnet** - Cliente **telnet** padrão (quando o pacote **telnet-ssl** está instalado, é simplesmente um link para **telnet-ssl**).

* **telnet-ssl** - Cliente telnet com suporte a **ssl**.

Os pacotes com o **-ssl** no final possuem suporte a criptografia **ssl**. Por padrão a porta usada para executar o serviço **telnet** é a 23 (ou outro número de porta definido no **/etc/services**).

A instalação do servidor **telnet** é feita via **inetd** (no arquivo **/etc/inetd.conf**), ou **xinetd,** e o controle de acesso ao serviço é feito através dos arquivos **/etc/hosts.allow** e **/etc/hosts.deny.**

## 3.3 Exemplo de utilização:

Os arquivos de configuração do **TCP WRAPPER** (**hosts.allow e hosts.deny**) permitem o controle de uso do serviço **telnet**.

# **/etc/hosts.allow**
# # Permite que qualquer um envie e-mails
in.smtpd: ALL
# Permitir telnet e ftp somente para hosts locais e meudominio.com.br
**in.telnetd,** in.ftpd: LOCAL, meudominio.com.br
# Permitir finger para qualquer um mas manter um registro de quem é
in.fingerd: ALL: (finger @%h | mail -s "finger from %h" root)

Recomenda-se a utilização do **telnet** apenas em redes seguras, visto que as senhas são transmitidas em texto puro, sem criptografia, tornando fácil um **sniffer** capturar as senhas e usá-las para futuras conexões **indesejadas**.

**telnet [endereço] [porta]** para realizar conexões com uma máquina rodando o servidor **telnet.**

Adicionalmente as seguintes opções podem ser usadas:

* **-l [usuario]** - Envia o nome de usuário ao computador remoto. Muito útil com o **telnet-ssl**.

* **-E** - Desativa o caracter de escape

* **-a** - Tenta fazer o **login** automático usando o nome de usuário local.

Se o login falhar, será solicitado o nome de usuário. Esta opção é usada por padrão com o cliente **telnet-ssl**.

* **-r** - Emula o comportamento do programa **rlogin**.

* Conecta-se ao servidor **telnet** rodando na porta 23 de sua própria máquina **telnet localhost**

* Conecta-se ao servidor **telnet** 200.200.200.200 operando na porta 53454 usando o nome de usuário **zemane** :

**telnet -l zemane 200.200.200.200 53454**

## 3.4 Serviço SSH

Escrito por Tatu Ylonen, é um substituto seguro para **rlogin, rcp e telnet**. Utiliza autenticação criptográfica para confirmar a identidade do usuário e faz a criptografia de toda a comunicação entre os **dois hosts**.

Sua forma e estrutura é feita a partir de um projeto de código-fonte aberto livremente distribuído (SSH) para um produto comercial (SSH2).

O SSH2 está disponível para **download** para fins não comerciais.

O grupo **OpenBSD** trabalhou o SSH e desenvolveu o **OpenSSH**. É o produto mais recomendado para substituir o **telnet, rlogin e rcp**.

Os principais componentes do SSH são :

1. **daemon** de servidor : **sshd**

2. Comandos em nível de usuário: **ssh** para **logins** remotos e **scp** para copiar arquivos, e **sftp** para transferências de arquivos.

Outros componentes do SSH são um comando **ssh-keygen** que gera pares de chave pública e um par de utilitários que ajuda a suportar X Windows seguro.

O **sshd** pode autenticar **logins** de várias maneiras diferentes.

**Método A:** Se o nome do **host** remoto em que o usuário está efetuando o **login** estiver listado em **~/.rhosts, ~/.shosts, /etc/hosts.equiv ou /etc/shosts.equiv,** o usuário é conectado automaticamente sem uma verificação de senha. Este esquema é um reflexo do antigo **daemon rlogin**.

**Método B:** O **sshd** pode utilizar criptografia de chave pública para verificar a identidade do **host** remoto. Para isso acontecer, a chave pública do **host** remoto (gerada na instalação) deve estar listada no arquivo **/etc/ssh_known_hosts** do **host** local ou no arquivo **~/.ssh/known_hosts** do usuário. Se o **host** remoto puder provar que conhece a chave privada correspondente (normalmente armazenada em **/etc/ssh_host_key,** um arquivo não legível para todas as pessoas), então o usuário é conectado sem que se peça a senha. O **método B** é mais restrito que o **A**, mas ele ainda não é o completamente seguro o suficiente. Se a segurança do **hsot** de origem estiver comprometida, o **site** local também estará.

**Método C:** o **sshd** pode utilizar criptografia de chave pública para estabelecer a identidade do usuário. No **login**, o usuário deve ter acesso a uma cópia de seu arquivo de chave privada e fornecer uma senha para decriptá-la. Esse método é mais seguro, mas é um tanto trabalhoso para configurar. Também significa que você nao pode efetuar **login** ao viajar, a menos que traga uma cópia de seu arquivo de chave privada com você (talvez no seu **laptop**).

**Método D**: por fim, o **sshd** pode simplesmente permitir que o usuário digite sua senha normal de **login**. Isso faz o **ssh** comportar-se de modo muito parecido com o **telnet**, exceto pelo fato de que tanto a senha como a sessão são criptografadas. As desvantagens principais desse método são que as senhas de **login** de sistema são relativamente fracas (frequentemente limitadas a oito caracteres significativos) e que há ferramentas prontas para utilizar (como o **crack**) projetadas para quebrá-las. Mas esse método é provavelmente a melhor escolha para uso normal.

## 3.5 Configurações de SSH

A política de autenticação é configurada no arquivo **/etc/ssh/sshd_config.** O arquivo apresenta muitos detalhes de configuração e uma maneira bem **didática** de configurá-lo é usando a ferramenta **WEBMIN**. As opções relevantes para a autenticação são:

| Opção | Método | Valor Default | Significado quando ativada |
|---|---|---|---|
| **RhostsAuthentication** | A | não | Permite login via **~/.shosts, /etc/shosts.equiv**, etc |
| **RhostsRSAAuthentication** | B | sim | Permite **~/.shosts** e outros, mas também requer a chave de **host** |
| **IgnoreRhosts** | A,B | não | Ignora os arquivos **~/.rhosts** e **/etc/hosts.equiv** [a] |
| **IgnoreRootRhosts** | A,B | não[b] | Impede autenticação de **rhosts/shosts** para **root** |
| **RSAAuthentication** | C | sim | Permite autenticação de chave pública criptografada por usuário |
| **PasswordAuthentication** | D | sim | Permite utilização de senha normal de **login** |

a - Mas continua a honrar os arquivos **~/.shosts e shosts.equiv**

b - Padrões para o valor **IgnoreRhosts**

Uma boa sugestão, que ativa os métodos C e D mas não os métodos A ou B, é como se segue:
**RhostsAuthentication no**
**RhostsRSAAuthentication no**
**RSAAuthentication yes**
**PasswordAuthentication yes**

## 3.6 Utilização do SSH

* Muitos detalhes de utilização e configuração, geração de chaves, diferenças entre as versões do SSH podem ser vistas no seguinte URL:

**http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Avancado/index.html/ch-s-ssh.html**

O método mais simples de se utilizar o **ssh** é fazer uma conexão ao servidor:

**$> ssh usuario@servidor.com.br**

A primeira questão que aparece é a seguinte:

**The authenticity of host 'servidor.com.br (143.107.X.Y)' can't be established.**
**DSA key fingerprint is f7:ce:4b:a3:05:38:97:a1:7c:07:e8:01:cc:b1:1d:18.**
**Are you sure you want to continue connecting (yes/no)?**

Isso significa que a conexão vai usar a chave DSA gerada (**DSA key fingerprint**) indicada na segunda linha.

Responda **yes** para continuar a conexão, e, depois, entre com a senha.

É recomendável que se consulte os manuais do comando **ssh** para entender melhor como os parâmetros podem ser utilizados.

**Exemplos:**

**$> ssh -V —> mostra a versão do ssh**

**$> sshd -V —> mostra a versão do daemon ssh, ou OpenSSH**

**$> ssh –help —> um pequeno help de utilização do ssh**

**$> ssh -l usuario servidor.com.br**

**$> ssh -X -l usuario servidor.com.br**

**permite que programas executados em ambiente XWindows sejam mostrados no host local.**

**$> sftp usuario@servidor.com.br —> conecta no servidor.com.br para transferências via ftp.**

**$> scp -C /pub/teste/script.sh usuario@servidor.com.br:~/**

**copia o arquivo script.sh para a área do usuário no servidor.com.br**

**$> scp -C usuario@servidor.com.br:~/script.sh /pub/teste**

**operação inversa ao anterior**

**Exercícios:**

1 - Utilizando o **tcpdump**, identifique os **logins e senhas** que trafegam pela rede. Faça os testes usando o protocolos:

**telnet e telnet-ssl**

2 - Configure o servidor **ssh** para :

a) Não permitir **login** de **root.**

b) Disponibilizar o serviço apenas na interface de rede local **eth0, port 22**

c) Não permitir consulta aos arquivos **~/.shosts e ~/.rhosts.**

d) Bloquear alguns usuários e grupos de usuários

e) Permitir apenas o protocolo versão 2

f) Permitir redirecionamento de conexões XWindows

g) Limitar tempos de conexão

3 - Instale um cliente **SSH**, num computador com Windows, e utilize os recursos de conexão **ssh e sftp.**

Exemplos de clientes SSH: PUTTY, SSH Secure Shell Client

Exemplos de clientes Telnet: QVTNET, PUTTY

4 - Utilize os comandos **sftp** para transferir arquivos e **scp** para copiar arquivos a partir do servidor.

5 - Instale os seguintes serviços no Debian Linux :

- **rsh-server** - rsh servers

- **krb5-rsh-server** - Secure replacements for rshd and rlogind using MIT Kerberos

Utilize os programas **rsh, rlogin e rexec** e faça uma comparação com os serviços **telnet** e **ssh.**

6 - Com o **ssh** gere um par de chaves pública/privada, com RSA ou DSA, altere seu tamanho e faça os testes de funcionalidade.

Mude as configurações dos clientes SSH para Windows de forma a trabalhar com as chaves novas.

7 - Instale o pacote **vim** e utilize o **vim** com o argumento **-x**, crie um arquivo texto e use o recurso de criptografar o arquivo.

8 - Mostre as diferenças entre um cliente **ftp** e uma conexão com **sftp**.

9 - Utilizando o TCP WRAPPER, faça um esquema para controlar os endereços IP autorizados a fazer conexões **ssh, telnet, telnet-ssl, rsh, rlogin e rexec.** Juntamente com as configurações do **daemon sshd**, restrinja ao máximos as conexões **ssh** permitidas.

10 - Monitore os arquivos de **log** e identifique as conexões da questão anterior.

# Capítulo 4

# Correio Eletrônico - Postfix

## 4.1 Introdução

O correio eletrônico é um dos serviços mais utilizados na Internet, e cada vez mais pessoas e empresas utilizam-no para trocar informações de maneira rápida e eficiente.

Neste capítulo será visto como funciona e como implementar um serviço de correio eletrônico. Será visto também como implementar um **webmail** (um serviço que permite que os usuários acessem as suas mensagens através de um navegador Internet), como criar filtros para barrar mensagens não solicitadas.

## 4.2 Funcionamento do Correio Eletrônico

Antes de implementar um serviço de correio eletrônico é importante que o administrador entenda como funciona a troca de mensagens, seja na Internet, seja em uma rede local. Para uma simples troca de mensagens entre dois usuários, pode ser necessária a utilização de vários protocolos e de várias aplicações.

Um usuário que queira enviar uma mensagem para outro utilizará um aplicativo cliente de e-mail, também conhecido como **MUA**, ou **Agente de Mensagens do Usuário**. Ao terminar de redigir a sua mensagem o MUA enviará a mensagem a um **MTA (Agente Transportador de Mensagens)** que se encarregará então de entregar a mensagem ao **MTA** do destinatário, caso ele se encontre em outra máquina ou simplesmente colocar a mensagem na caixa postal do destinatário, caso ele se encontre no mesmo servidor.

A transferência da mensagem entre o **MUA** e o **MTA** se efetua utilizando-se um protocolo chamado **SMTP,** ou **Protocolo Simples de Transferência de Mensagens**. O protocolo **SMTP** será utilizado também entre o **MTA** do remetente e o **MTA** do destinatário.

O servidor de **e-mail** do destinatário, ao receber uma mensagem para um dos seus usuários, simplesmente a coloca na caixa postal deste usuário.

Se o usuário possui uma conta **shell** neste servidor ele poderá ler os seus **e-mails** direto no servidor, caso contrário o usuário deverá transferir suas mensagens para sua máquina a fim de lê-las com o seu cliente de **e-mail**. A transferência de mensagens recebidas entre o servidor e o cliente de **e-mail** requer a utilização de outros programas e protocolos. Usualmente é utilizado para este fim o protocolo **POP**, Protocolo de "Agência" de Correio, que recebe este nome por agir como uma agência de correios mesmo, que guarda as mensagens dos usuários em caixas postais e aguarda que estes venham buscar suas mensagens. Outro protocolo que pode ser utilizado para este mesmo fim é o **IMAP**, **Protocolo para Acesso de Mensagens via Internet**, que implementa além das funcionalidades fornecidas pelo **POP** muitos outros recursos. Os protocolos **POP** e **IMAP** são protocolos para recebimentos de mensagens, ao contrário do protocolo **SMTP** que serve para enviar mensagens, logo, possuem funcionalidades diferenciadas, como, por exemplo, autenticação do usuário.

Para a utilização dos protocolos **POP** e **IMAP** é necessário a instalação do servidor apropriado, que vai ser o responsável por atender as solicitações do cliente de **e-mail** por novas mensagens. O recebimento de mensagens pelo cliente se dá através da solicitação do **MUA** do usuário ao seu servidor de **e-mail**, que após a autenticação do usuário vai informar se existem mensagens em sua caixa postal e quantas são. A seguir o **MUA** solicita a transferência das mensagens para a máquina local, finalizando assim o processo de troca de mensagens entre dois usuários.

## 4.3 Postfix

O **Postfix** é o **MTA** padrão de muitas distribuições Linux.

O **Postfix** vem se consolidando como uma alternativa ao **Sendmail** (www.sendmail.org) em razão de suas características, como maior robustez, melhor desempenho e maior facilidade na manutenção e configuração. Além do mais o **Postfix** é capaz de emular várias funções do **Sendmail**, evitando assim modificações nas aplicações que utilizam o **Sendmail**.

Outra característica importante do **Postfix** é a sua construção modular, facilitando a manutenção do código e permitindo a implementação de novas funcionalidades mais facilmente.

Para uma implementação bem-sucedida do Postfix é necessário:

*uma interface de rede instalada e configurada;

*um servidor **DNS** instalado e configurado.

A instalação no Debian Linux é bastante simples, basta usar o seguinte comando:

**$> apt-get install postfix**
**$> apt-get install mailx —> quem realmente envia os E-Mails**

**Opcionais :**
**$> apt-get install procmail —> Filtro de E-Mail**
**$> apt-get install squirrelmail —> WEBMAIL**
**$> apt-get install php4 (ou php5) —> Linguagem PHP**

**Outros WEBMAILs :**
**Openwebmail - http://www.openwebmail.org**
**IMP Horde - http://www.horde.org/imp**

## 4.4 Configuração do Postfix

Os arquivos de configuração do **Postfix** estão localizados no diretório **/etc/postfix.**

Os principais arquivos são : **main.cf e master.cf**

As configurações devem ser modificadas de acordo com os seguintes itens: **hostname, dominio, Servidor DNS, arquivos de controle.**

Há muitos itens para serem configurados, tais como:

**Informação de caminho local**

Área de armazenamento temporária (**spool**), e o caminho para os binários dos programas e serviços que compõem o Postfix.

**Proprietário da fila e processo**

O **Postfix** roda como um usuário comum do sistema e com permissões de usuários comuns, tornando-o assim mais seguro. São configurados o usuário **dono** dos arquivos de armazenamento temporário de mensagens e as permissões do programa.

**Nome da domínio e da máquina**

Nome da máquina (**nome + domínio**) e o **domínio**. Normalmente esses valores são obtidos automaticamente, no entanto pode ser necessário configurá-los manualmente, principalmente se a máquina em questão possui um nome e um apelido. Este nome e o domínio serão utilizados como padrão ao se enviar mensagens a partir dessa máquina.

**Enviando mail**

O nome padrão a ser utilizado ao serem enviadas mensagens a partir dessa máquina, ou seja, especifica a máquina ou domínio de onde **e-mails** postados localmente parecerão ter vindo. A configuração padrão é utilizar o nome da máquina, e assim, um usuário, ao enviar uma mensagem, seria identificado como **usuario@maquina**, por exemplo. É possível utilizar apenas o domínio (**usuario@dominio**), ou ainda, um outro nome qualquer. Note que isso só afeta o nome padrão adotado por programas clientes de **e-mail** desta máquina, e que os usuários podem configurar os seus programas de e-mail para enviar mensagens utilizando outro nome de domínio.

**Recebendo mail**

Regras básicas para recebimento de mensagens. Configurar os endereços de interface de rede pelos quais o sistema receberá mensagens. Por padrão o Postfix aceita mensagens por todos os endereços. No campo Destino (mydestination) insira os endereços ou informe um arquivo contendo estes endereços para os quais essa máquina é o destino final. Por exemplo, se o seu domínio é **minhaorganizacao** e esta é a máquina encarregada de receber as mensagens deste domínio, insira "minhaorganizacao" neste campo, mas atenção, não insira aqui os **domínios virtuais** hospedados nessa máquina pois essa configuração é feita em outro local. A configuração inadequada deste campo causará a mensagem de erro **"Mail for xx.xx.xx loops back to myself"** (**mensagem para xx.xx.xx retorna para mim**), pois o **Postfix** não está configurado para ser o destino final desse endereço.

**Rejeitando usuários locais desconhecidos**

O administrador poderá indicar um arquivo que contém os usuários que pertencem ao domínio e que podem receber mensagens. Não é preciso preencher nada nessa opção para uma configuração básica.

**Controle de depuração**

Configurações relativas às mensagens de registro utilizadas pelo **Postfix** para a resolução de problemas. Para uma configuração simples utilize os valores padrões.

**Utilitários de linha de comando:**

**postfix** inicia e pára o sistema de correio

**postalias** faz o comando **newaliases** funcionar

**postcat** lista o conteúdo de arquivos na fila

**postconf** exibe e edita o arquivo de configuração de correio, **main.cf**

**postdrop** adiciona mensagens à fila de maildrop

**postkick,postlock,postlog** fornece bloquio e registro em **log** para **scripts** de **shell**

**postmap** cria tabela de base de dados, como o comando **makemap** do Unix

**postsuper** gerencia as filas (executa na inicialização)

Mais características, funcionalidades e demais detalhes de configuração do Postfix podem ser obtidas nos seguintes URLs:

**http://www.postfix.org**
**http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch11s02.html**
**www.linorg.cirp.usp.br/Guias.Conectiva/Guias_V.9.0/servidor/correioeletronico.html#POSTFIX**

## 4.5 Testando a Configuração

Para testar a configuração é necessário certificar-se de que as mensagens estão chegando corretamente ao seu servidor e que as mensagens com destino fora do seu servidor estão sendo enviadas corretamente.

O administrador poderá iniciar os seus testes, experimentando enviar uma mensagem a partir do próprio servidor para um outro usuário também localizado no servidor (verifique se o pacote **mailx** está instalado em seu sistema).

Isso pode ser feito através de um terminal, digitando-se na linha de comando:

# **echo teste | mail usuario@minhaorganizacao**

**LEMBRETE: Após fazer as alterações nos arquivos de configuração, o servidor deve ser reinicializado. Use os comandos a seguir:**

**$> postfix stop**
**$> postfix start ou**
**$> postfix reload**

**Verificação das configurações :**
**$> postconf -n**

## 4.6 Exemplo de configuração

O arquivo abaixo (**main.cf**) traz um exemplo de configuração de um servidor Postfix instalado num Debian Linux.

A versão do Postfix é **1.1.11-0.**
**Utilize sempre a versão mais nova. Obtenha-a a partir do site (www.postfix.org)**

*** Arquivo /etc/postfix/main.cf:**

```
# See /usr/share/postfix/main.cf.dist for a commented, fuller
    # version of this file.
    # Do not change these directory settings - they are critical to Postfix operation.
    # Diretórios e arquivos componentes do Postfix
    command_directory = /usr/sbin
    daemon_directory = /usr/lib/postfix
    program_directory = /usr/lib/postfix
    smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
    setgid_group = postdrop
    biff = no
    masquerade_domains = $mydomain
    # appending .domain is the MUA's job.
    append_dot_mydomain = no
    # Hostname e domainname de seu servidor
    myhostname = servidor.unidade.usp.br
    mydomain = servidor.unidade.usp.br
    # Lista de e-mails e servidores SUSPEITOS
    alias_maps = hash:/etc/aliases
    alias_database = hash:/etc/aliases
    myorigin = $mydomain
    mydestination = servidor.unidade.usp.br, localhost.servidor.unidade.usp.br, localhost
    # Quem faz RELAY
    relayhost =
    mynetworks = 143.107.X.0/24, 143.107.Y.0/24, 127.0.0.0/8
    mailbox_command = procmail -a "$EXTENSION"
    #Tamanho das mensagens
    message_size_limit = 10240000
    #Tamanho do mailbox
    mailbox_size_limit = 20480000
    recipient_delimiter = +
    #Diretório de fila de e-mails
    mail_spool_directory = /var/spool/mail
    # Quem envia os E-Mails
    mailbox_command = /usr/bin/procmail
    # JUNK MAIL CONTROLS
    # # The controls listed here are only a very small subset. See the file
```

```
# sample-smtpd.cf for an elaborate list of anti-UCE controls.
# The disable_vrfy_command parameter allows you to disable the SMTP
# VRFY command. This stops some techniques used by spammers to harvest
# email addresses.
# disable_vrfy_command = yes
# The smtpd_helo_required parameter optionally turns on the requirement
# that SMTP clients must introduce themselves at the beginning of an
# SMTP session. smtpd_helo_required = yes
# The strict_rfc821_envelopes configuration parameter controls whether
# the Postfix SMTP server requires that MAIL FROM and RCPT TO addresses
# are specified within <>, and that MAIL FROM and RCPT TO addresses
# do not contain RFC822-style comments or phrases. It's great to
# stop SPAM mailers. But it also trips up broken peecee clients.
# # By default, Postfix SMTPD allows RFC822 syntax in MAIL FROM and RCPT TO.
# strict_rfc821_envelopes = yes
# Aqui, a checagem eh feita atraves do HELO, remetente e destinatario
# Bloqueia falsos usuários, falsos remetentes, etc.
smtpd_client_restrictions =
smtpd_helo_restrictions =
smtpd_sender_restrictions =
smtpd_recipient_restrictions =
permit_mynetworks,
 reject_non_fqdn_sender,
 reject_non_fqdn_recipient,
 reject_unknown_sender_domain,
 reject_unknown_recipient_domain,
 reject_unauth_destination,
 reject_unauth_pipelining,
 reject_unknown_client,
 reject_invalid_hostname,
 reject_non_fqdn_hostname,
 permit
# The header_checks parameter restricts what may appear in message
# headers. This requires that POSIX or PCRE regular expression support
# is built-in. Specify "/^header-name: stuff you do not want/ REJECT"
# in the pattern file. Patterns are case-insensitive by default. Note:
# specify only patterns ending in REJECT (reject entire message) or
# IGNORE (silently discard this header). Patterns ending in OK are
# mostly a waste of cycles.
#
# Examina os cabeçalhos e faz filtragem
#header_checks = regexp:/etc/postfix/rejected.he
#header_checks = pcre:/etc/postfix/filename
#
# Aqui, a diretiva body_checks checa por anexos indesejaveis
body_checks = regexp:/etc/postfix/rejected.bo
# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
# You MUST specify $myhostname at the start of the text. That is an
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version) (Mandrake Linux)
smtpd_banner = $myhostname NO UCE ESMTP
default_process_limit = 150
qmgr_message_recipient_limit = 40000
```

Alterações nos arquivos **/etc/mail/aliases, dentre outros,** devem ser atualizadas com o comando **postmap.**

Examine os comandos que pertencem ao pacote **Postfix** e veja como eles devem ser usados.

Comandos para controle de filas, de atualização, verificação de **logs**, etc.

Uma cláusula importante que pode ser adicionada ao arquivo **main.cf:**

**notify_classes**

Essa cláusula determina quais problemas são levados ao conhecimento do **postmaster**. Muitas classes podem resultar em tantas mensagens que ele pode não estar preparado. O padrão é:

**notify_classes = resource, software**

o que limita os erros a problemas da máquina **host** e a problemas de **software** do **Postfix**.

A tabela a seguir mostra os valores que podem ser incluidos em **notify_classes:**

| Classe | Problemas com o correio enviado |
|---|---|
| **bounce** | Mensagens que não podem ser entregues |
| **2bounce** | Rebatimentos duplos (quando a mensagem de rebatimento também rebate) |
| **delay** | Mensagens atrasadas na fila (somente cabeçalho) |
| **policy** | Rejeições ao **SPAM** (inclui transcrição de SMTP) |
| **protocol** | Erros de protocolo (inclui transcrição de SMTP) |
| **resource** | Problemas de recurso (falhas na gravação das filas, sistemas de arquivo cheio, etc) |
| **software** | Erros internos "isso não pode ocorrer" do Postfix |

A configuração do arquivo **master.cf** é mais complexa. Neste arquivo podem ser colocados itens para que o Postfix trabalhe com Softwares Antivírus, tais como : **Uvscan (da McAfee), Clamav, Dr. Web, Trendmicro, Sophos, etc, e com sistemas intermediários como o Amavis.**

Exemplos de configuração podem ser obtidos nos seguintes URLs:

http://www.securityanalyze.com/gsoares/clamav_postfix.html

http://www.rhbr.com.br/modules.php?name=News&file=article&sid=108

http://www.underlinux.com.br/artigo311.html

**Use o pacote WEBMIN para fazer as configurações do Postfix e veja as facilidades/dificuldades que podem ser encontradas.**

### 4.6.1 Exemplo do master.cf

O exemplo a seguir mostra parcialmente o arquivo preparado para o **Antivírus CLAMAV:**

```
# DO NOT SHARE THE POSTFIX QUEUE BETWEEN MULTIPLE POSTFIX INSTANCES.
# # ==========================================================
# service type private unpriv chroot wakeup maxproc command + args
# (yes) (yes) (yes) (never) (100)
# ==========================================================
smtp inet n - n - - smtpd
-o content_filter=clamav:clamav clamav unix - n n - - pipe
    flags=Rq user=clamav argv=/usr/lib/postfix/clamav-filter.sh -f ${sender} - ${recipient}
#submission inet n - n - - smtpd
# -o smtpd_etrn_restrictions=reject
#628 inet n - n - - qmqpd ....
....
....
```

## 4.7 Controle de SPAM

O **Postfix** utiliza expressões regulares, tabelas de base de dados e as listas negras do projeto **MAPS** para filtrar **SPAM**. A tabela a seguir mostra algumas das variáveis do **Postfix** relacionado ao **SPAM.**

Se uma mensagem corresponder a uma pesquisa em uma tabela e o valor de tabela for **REJECT**, a mensagem será rejeitada com uma mensagem de erro adequada. Para os **hackers** do

**Perl**, eis um exemplo de expressão regular utilizada em filtros de **SPAM** de um **site**:
**/^friend@.$/ 550 Stick this in your pipe $0**
Se realmente tiver um usuário chamado **"friend"** em seu domínio, você poderia excluir esse usuário da mensagem amigável de erro com:
**/^friend@(?!meusite.com).*$/ 550 Stick this in your pipe $0**

| Variável | Significado |
|---|---|
| **header_checks** | Filtros em cabeçalho |
| **smtpd_client_restrictions** | Filtros em conexões de cliente, listas negras, etc |
| **smtpd_sender_restrictions** | Filtros nos endereços do remetente |
| **smtpd_recipient_restrictions** | Filtros nos endereços do destinatário |
| **smtpd_helo_required** | Exige SMTP HELO com nome de **host** identificado |
| **smtpd_helo_restrictions** | Exige pesquisa inversa de DNS |
| **smtpd_etrn_restrictions** | Lista **hosts** com permissão para solicitar execuções de fila |

Para utilizar as **listas negras** do projeto **MAPS**, adicione o seguinte ao seu arquivo **main.cf**:
**maps_rbl_domains =opm.blitzed.org,**
**list.dsbl.org,**
**blackholes.easynet.nl,**
**zombie.dnsbl.sborbs.net,**
**bl.spamcop.net,**
**sbl.spamhaus.org**
**rbl.maps.vix.com**
**dul.maps.vix.com**
**relays.mail-abuse.org**
**smtpd_client_restrictions = reject_maps_rbl**

**Atenção para as modificações futuras:**
**warning: support for restriction "reject_maps_rbl" will be removed from Postfix; use "reject_rbl_client domain-name" instead**

**Consulte estas listas antes para ver os e-mails, domínios, endereços IPs, etc, cadastrados. Há listas que rejeitam todos os e-mails originados do Brasil.**

## 4.8 Postfix com Antivírus

Nos dias de hoje é comum receber uma infinidade de **e-mails** com vírus. Os servidores de **E-Mails** tem que entregar as mensagens.

Além de sua função básica, entregar mensagens, agora ele tem que tratar os **e-mails**, verificar o que é SPAM, o que é vírus e aplicar os filtros necessários. Clientes de Webmail, tais como o **Openwebmail** já estão com filtros prontos para utilizar.

Muitos tutoriais ensinam como instalar o **postfix + amavis + clamav, postfix + clamav, etc.**

O **AMAVIS** é uma interface que atua entre o servidor de **E-Mails** e o **Antivírus.** O site é **www.amavis.org**

O **CLAMAV** é um sistema de Antivírus para Software Livre (**www.clamav.net**)

O **UVSCAN** é um sistema de Antivírus da McAfee (**www.mcafee.com**)

O **SOPHOS** é um sistema de Antivírus e Anti-SPAM da Sophos (**www.sophos.com**)

Para instalar esse conjunto todo, deve-se observar os Softwares exigidos, suas versões, e obedecer todos os itens de configuração e detalhes envolvidos.

* **Alguns sites com excelentes tutorias:**
http://www.superphp.com.br/tutoriais/index.php?id=57
http://www.securityanalyze.com/gsoares/clamav_postfix.html

http://www.underlinux.com.br/modules.php?name=Sections&op=printpage&artid=311
http://www.nerdgroup.org/cooltrick/uvscan_postfix.txt
http://www.linuxit.com.br/article60.html

## 4.9 Clientes POP e IMAP

Todo servidor de **E-Mails** deve disponibilizar serviços para que os clientes possam usar seus **E-Mails**.

Os protocolos **POP3 e IMAP** são responsáveis pelo transporte de mensagens recebidas do servidor de **e-mail** para o cliente de **e-mail** do usuário. O protocolo **POP3** é mais antigo e mais simples, mas é mais popular e praticamente todos os programas clientes de **e-mail** o suportam. O protocolo **IMAP** é mais novo e possui mais funções que o **POP3**, no entanto nem todos os programas clientes de **e-mail** o suportam.

O suporte a estes protocolos não é feito pelo **Postfix**, mas sim por outros servidores.

### 4.9.1 Instalação

Para implementar um servidor **POP/IMAP** é necessário somente que um servidor de **e-mail** esteja instalado e configurado.

Será necessário também que o serviço **inetd** (ou **xinetd**) esteja ativo.

Instale os pacotes **uw-imapd e qpopper:**

**$> apt-get install uw-imapd qpopper**

Podem ser instalados outros pacotes de **POP e IMAP. Os mais comuns são:**

**POP:**

**$> apt-cache seach pop**

ipopd - POP2 and POP3 servers from UW
ipopd-ssl - POP2 and POP3 servers from UW
**qpopper - Enhanced Post Office Protocol server (POP3).**
qpopper-drac - Qpopper with DRAC Support
popa3d - A tiny POP3 daemon, designed with security as the primary goal
cyrus-pop3d - CMU Cyrus mail system (POP3 support)
courier-pop - POP3 daemon with PAM and Maildir support
courier-pop-ssl - POP3 daemon with SSL, PAM and Maildir support

**IMAP:**

**$> apt-cache search imapd**

cyrus-common - CMU Cyrus mail system (common files)
cyrus-imapd - CMU Cyrus mail system (IMAP support)
teapop - Powerful and flexible RFC-compliant POP3 server
teapop-mysql - Powerful and flexible RFC-compliant POP3 server
teapop-pgsql - Powerful and flexible RFC-compliant POP3 server
**uw-imapd - remote mail folder access server**
uw-imapd-ssl - remote mail folder access server

Para configurar uma conta **POP** não é necessária mais nenhuma configuração: ela usa as contas de usuários do sistema, ou as contas criadas pelo administrador **IMAP**.

## 4.10 Testando a Configuração

Para testar a configuração do **IMAP**, será necessário configurar um cliente de **e-mail** localizado em alguma máquina de sua rede para buscar suas mensagens no servidor. Mande algumas mensagens para um usuário e depois tente recuperar as mensagens deste usuário através do cliente

de **e-mail.** Caso não seja possível recuperar essas mensagens verifique os arquivos de registro **/var/log/messages** e **/var/log/maillog** em busca de mensagens de erro.

O teste de funcionamento da configuração do **POP** pode ser feito da mesma forma: configure um cliente de **e-mail** para que ele busque as mensagens de uma conta **POP**; envie uma mensagem para esta conta e depois tente recuperá-la. Caso não funcione, verifique se a senha foi digitada corretamente e se a configuração do cliente de **e-mail** está correta.

Outro teste da configuração do **POP** também é simples, e pode ser feito no próprio servidor:

**$> telnet localhost 110**

**Como o IMAP trabalha com a porta 143, faça o seguinte:**

**$> telnet localhost 143**

Se não obter as mensagens:

**Trying 127.0.0.1...**
**Connected to localhost.**
**Escape character is '^]'.**

Os serviços não estão funcionando corretamente e as configurações devem ser verificadas.

Consulte o arquivo /etc/services para ver em quais portas estes serviços devem trabalhar:

**$> grep imap /etc/services**
**$> grep pop /etc/services**

Use o programa **nmap** também para fazer os testes :

**$> nmap localhost**

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
(The 1538 ports scanned but not shown below are in state: closed)
Port State Service
9/tcp open discard
13/tcp open daytime
21/tcp open ftp
22/tcp open ssh
25/tcp open smtp
37/tcp open time
53/tcp open domain
80/tcp open http
**110/tcp open pop-3**
111/tcp open sunrpc
113/tcp open auth
**143/tcp open imap2**

Pode-se usar também os comandos **netstat** ou **lsof** para verificar as portas **TCP/UDP** que estão sendo usadas no momento, da seguinte maneira:

**$> lsof -n -i -P**
**$> netstat -natp**

## 4.11 Configuração do Webmail

A instalação do **Webmail** é bastante simples. O **Squirrelmail** já faz parte dos pacotes das distribuições Debian. Pode-se instalar outro **Webmail**, tal como o **Openwebmail** ou o **IMP Horde**.

O **Squirrelmail** é baseado em linguagem **PHP**. Instale os dois pacotes da seguinte forma:

**$> apt-get install php4 squirrelmail**

Configure o **Squirremail** da seguinte maneira:

**$> /etc/squirrelmail.conf.pl**

Estabeleça os nomes do **Domínio**, do **SMTP**, do **IMAP** e suas respectivas portas de conexão.

Como descrito anteriormente, foi instalado o **IMAP uw-imapd.**

Configure a linguagem padrão (**pt_BR**), as figuras que devem aparecer na tela inicial (logotipo), nome do servidor, etc.

Personalize as configurações e, consulte sempre o site do **Squirrelmail** para conhecer outros recursos que podem ser implementados, tais como troca de senhas, domínios virtuais, utilização com Banco de dados, etc.

Instale o servidor **WEB Apache:**
**$> apt-get install apache**

Se preferir o **Openwebmail**, que depende do **Perl** e não está nos pacotes distribuídos pelo Debian Linux, versão **stable**, será necessário obter os fontes a partir do site (**http://www.openwebmail.org**). Este pacote está disponível para a distribuição **testing** do Debian Linux:
**http://packages.debian.org/testing/web/openwebmail**
* Depois de instalado o **Squirrelmail**, usando um **Browser** acesse o seguinte URL:
**http://servidor.com.br/squirrelmail**
* Se necessário, confira as configurações de seu servidor **Apache**. Verifique se os módulos necessários para o **PHP4** funcionar estão presentes no arquivo **httpd.conf**.

**$> grep php /etc/apache/conf/httpd.conf**

A resposta deve ser linhas com algo do tipo:

**LoadModule php4_module /usr/lib/apache/1.3/libphp4.so**
**DirectoryIndex login.php index.html index.php index.htm index.shtml index.cgi**
**#AddType application/x-httpd-php3 .php3**
**#AddType application/x-httpd-php3-source .phps**
**#AddType application/x-httpd-php .php**
**#AddType application/x-httpd-php-source .phps**
* Se não houver estas linhas, elas terão que ser adicionadas no **httpd.conf** e o servidor **Apache** deve ser reinicalizado.
* Observe os **PATHS** dos arquivos que tem que ser carregados por **LoadModule.**

**Faca um teste para ver se o PHP4 está funcionando. Crie um script em PHP (teste.php) com o seguinte conteúdo:**
**<?**
**phpinfo();**
**?>**

Salve-o no **/var/www/**
Aponte o **Browser** para o **script** da seguinte maneira:
**http://servidor.com.br/teste.php**
O resultado deve ser uma página bem extensa mostrando todas as configurações do **PHP** instalado.

Caso seja mostrado **apenas o conteúdo** do **script teste.php**, ou se for pedido para especificar o lugar onde salvar o arquivo, verifique as configurações do **PHP4** e do servidor **Apache**. Alguma configuração no **Apache** nao está correta.

A cada modificacao nas configuracoes do servidor **Apache,** lembre-se de reinicia'-lo com o comando:
**$> /etc/init.d/apache restart**

*** Caso não consiga ter resposta do servidor verifique se o Postfix, inetd (ou xinetd), popd, imapd e Apache estão ativos.**

Lembre-se de habilitar os serviços **popd e imapd** no **TCP WRAPPER**, nos arquivos **/etc/hosts.allow e /etc/hosts.deny.**

Veja os detalhes de como isso deve ser feito, verifique as sintaxes necessárias, etc.

Para acompanhar a evolução das mensagens de **log**, verifique constantemente os arquivos de **log**, mais especificamente os arquivos **/var/log/auth.log, /var/log/messages, /var/log/syslog, /var/log/apache/access.log e error.log, /var/log/mail/.**

Use os comandos :

**$> tail -f /var/log/mail/mail.log**
**$> tail -f /var/log/mail/mail.err**
**$> tail -f /var/log/mail/mail.info**
**$> tail -f /var/log/mail/mail.warn**

Faça o mesmo para os outros arquvos de **log.** Veja os **logs** do servidor **Apache** e consulte as mensagens de erro. Elas servem para lhe orientar melhor na busca de problemas de configuração.

# Capítulo 5

# Sistemas de Backup

## 5.1 Introdução

O Administrador Unix frequentemente necessita de restaurar arquivos perdidos, ou até mesmo o sistema de arquivamento inteiro. As razões são várias. Por exemplo, uma queda de energia elétrica pode, em algumas situações, danificar um sistema de arquivamento. Outra situação comum é a remoção, por engano, de arquivos pelos usuários. Para que esta tarefa seja executada a contento, é necessário que se tenha cópias de segurança (**backups**) atualizadas.

A seguir serão detalhados alguns comandos Unix para a criação de **backups**, cópia de arquivos, verificação de integridade.

## 5.2 Política de backup

Consiste na definição de uma série de parâmetros, tais como a frequência em que o **backup** deve ser executado, quais sistemas de arquivamento devem ser salvos, em qual meio físico, tempo de disponibilidade, etc. Para que o Administrador Unix defina uma política adequada à sua instalação, deve-se levar em consideração muitos fatores. Exemplo:

- Particionamento do disco: se está dividido em várias partições ou possui apenas uma partição (**root**);
- A frequência de alteração dos arquivos: se todos os sistemas de arquivamento são alterados com a mesma frequência ou não;
- A importância dos dados contidos nos sistemas de arquivos: se todos os sistemas de arquivos possuem a mesma importância ou não;
- O tempo gasto para criar um **backup**: a geração de um **backup** pode demorar muito tempo;
- O meio físico a ser utilizado.

## 5.3 Tipos de Backup

**Backup** total é uma forma de **backup** em que todos os arquivos são salvos, independente de terem ou não sido modificados desde o último **backup** efetuado.

**Backup** incremental é uma política de **backup** em que somente arquivos modificados em relação ao **backup** anterior são salvos.

Um **backup** de nível 0 (total) salva todo o sistema de arquivamento.

Um **backup** de um determinado nível só salva os arquivos modificados desde o instante do último **backup** de nível menor.

### 5.3.1 Comando ufsdump

Copia sistemas de arquivamento inteiros ou arquivos e diretórios selecionados (seletivo).

Pode ser total ou incremental.

Um **backup** seletivo será sempre de nível 0 (total)

No Debian Linux não existe este pacote. O **ufsdump** é muito usado em **Sistema Operacional Solaris, e outros Unixes comerciais.**

Antes de executar o comando **ufsdump,** deve-se observar o seguinte:

- Verifique a atividade do sistema, pois a melhor hora é quando a atividade estive baixa;
- Entre em modo de manutenção **(S, ou Single User)**. Se não for possível entrar em nível S, no mínimo desmonte o sistema de arquivos a ser salvo, já que não deve ser utilizado durante o **backup**;
- Lembre-se que, para ser feito **backup** com verificação, o sistema de arquivamento deve estar desmontado. O sistema de arquivos **raiz ( / )** e o **/usr** nunca podem ser desmontados, pois o sistema precisa deles para funcionar. Portanto, para se fazer um **backup** destes sistemas de arquivamento com verificação, deve ser dada carga de **CD-ROM**.
- Notifique os usuários
- É uma boa prática verificar o sistema de arquivos, via comando **fsck**, antes de efetuar um **backup**.

### 5.3.2 Comando ufsrestore

Para se restaurar arquivos ou subdiretórios é preciso que se tenha os volumes que contêm a cópia mais recente dos arquivos. Pode-se começãr pelo volume de **backup** mais recente, independente do nível. Se o arquivo não for encontrado, tenta-se o anterior, e assim por diante. Pode-se também, antes de tentar restaurar, listar os volumes para localizar a cópia desejada do arquivo.

O comando **ufsrestore** extrai sistemas de arquivos completos, ou arquivos/diretórios selecionados de um **backup** criado pelo **ufsdump**.

Os arquivos são, usualmente, restaurados para o diretório **/var/tmp** ou **/tmp**, de modo a evitar escrita acidental em arquivos já existentes.

## 5.4 Comando tar

O comando **tar** salva arquivos para algum meio de armazenamento, tipicamente uma fita magnética, podendo ser o disco flexível ou mesmo um arquivo regular.

É o meio mais fácil de se salvar ou copiar arquivos regulares.

Os arquivos são salvos seguindo um formato particular do comando **tar**, não sendo entendido por outros comandos.

Os arquivos operados pelo **tar** podem ser regulares ou diretórios não-vazios. Diretórios vazios e arquivos especiais não são operados pelo comando **tar**.

O comando **tar** pode restaurar arquivos específicos ou diretórios, recursivamente.

Permite o uso de nomes de arquivos relativos ou absolutos. Ao se usar um nome absluto, o arquivo, ao ser restaurado, é colocado exatamente no lugar original. Se o nome for relativo, o arquivo pode ser restaurado para qualquer posição no sistema de arquivamento. Recomenda-se sempre usar nomes relativos.

### 5.4.1 Sitaxe do comando tar

$> tar opções [ argumentos ] arquivos ...

**c** Cria um meio de salvamento, contendo os arquivos referenciados por arquivos. Diretórios são copiados recursivamente. Arquivos especiais e diretórios vazios são ignorados;

**x** Percorre o meio de salvamento, restaurando (**extract**) os arquivos referenciados por arquivos. As datas, dono, grupo e modo do arquivo são também restaurados, se possível. Diretórios são restaurados recursivamente. Se não forem fornecidos nomes, todos os arquivos contidos no meio de salvamento serão restaurados;

**t** Percorre o meio de salvamento, listando todas as ocorrências dos arquivos referenciados por arquivos. Se não forem fornecidos nomes, todos os arquivos contidos no meio de salvamento serão listados;

**f** Utiliza argumento a seguir como dispositivo de saída. Se o argumento for "**-**", o comando lê da entrada padrão e escreve na saída padrão;

**v** Exibe os nomes dos arquivos quando da restauração. É o modo **verbose**.

**p** Restaura os nomes com as permissões que estão gravadas no arquivo de salvamento, ignorando a máscara de criação de arquivos;

**b** fator de blocagem, em blocos de 512 bytes;

**M** cria/lista/extrai arquivos com multi-volumes;

**d** acha diferenças entre arquivo tar e o sistema de arquivos;

**z** comprime usando **gzip**;

**Z** comprime usando **compress**;

**j** comprime usando **bzip2**

A ordem dos argumentos associados às opções deve ser a mesma ordem das opções. Exemplo:
O comando abaixo vai colocar no **disquete (/dev/fd0)** os arquivos com extensão **.conf** do diretório **/etc/ .**

**$> tar -cvf /dev/fd0 /etc/*.conf**

### 5.4.2 Exemplos de utilização:

- Gerar arquivo **tar** a partir do diretório (artigos) em um **disquete (/dev/fd0)**, compactando:

**$> tar cvzf /dev/fd0 artigos**

- Gerar cópia do diretório **/home/aluno** no diretório **/tmp**, sem compactação:

**$> tar cvf /tmp/aluno.tar /home/tar**

- Gera cópia do diretório **/home/aluno** no diretório **/tmp,** com compactação e preserva os atributos dos arquivos, datas de criação, permissões, etc. :

**$> tar zcvpf /tmp/aluno.tar.gz /home/tar**

- Mostrar o que está em **aluno.tar :**

**$> tar tvf aluno.tar**

- Extrair o que foi gravado em **aluno.tar :**

**$> tar xvf aluno.tar**

- Para copiar uma árvore de diretório para outro diretório. Copiar **/dir2/olddir** para **/dir1/newdir**:
- **$> mkdir -p /dir1/newdir**
  **$> cd /dir2/olddir**
  **$> tar cfv - . | (cd /dir1/newdir ; tar xvf -)**
- O comando **tar cvMf /dev/fd0 <lista de arquivos>**, cria um pacote com todos os arquivos selecionados (na **<lista de arquivos>**) e grava-os em vários disquetes, solicitando ao usuário a troca de disquete quando cada um vai ficando cheio.
- Para recuperar os arquivos gravados, podemos executar **tar xvMf /dev/fd0**, e para listar o conteúdo desse conjunto de disquetes, **tar tvMf /dev/fd0.**

**Veja os manuais do comando tar para mais opções.**

## 5.5 Comando cpio

O comando **cpio** tem funcionalidade semelhante ao comando **tar**, mas apresenta algumas vantagens:

- Permite o uso de metacaracteres para identificar os arquivos a serem operados;
- Pode copiar subárvores ou sistemas de arquivamento completos, mantendo-os exatamente iguais, pois opera arquivos especiais e diretórios vazios.
- Permite multivolume

A desvantagem em relação ao **tar** é a sua sintaxe complicada.

Como no **tar,** é possível o uso de nomes de arquivos relativos ou absolutos. Ao se usar um nome absoluto, o arquivo, ao ser restaurado, é colocado exatamente no lugar original. Se o nome for relativo, o arquivo pode ser restaurado para qualquer posição no sistema de arquivamento. Recomenda-se sempre usar nomes relativos.

O comando:

- **$> find *diretoriodeorigem* -depth -print | cpio -pdm diretoriodedestino**

faz uma cópia da árvore dos diretórios **diretoriodeorigem** para **diretoriodedestino**. A maioria das versões do **cpio** não permite vários volumes de fita. Algumas versões de **cpio** não lidam com **pipes** de maneira elegante e apenas o superusuário pode copiar arquivos especiais. Ao utilizar o **cpio**, leia suas páginas **man** cuidadosamente; as opções variam significativamente entre sistemas.

## 5.6 Comando dd

Usado para cópia e conversão de arquivos. A menos que seja instruído a fazer algum tipo de conversão, **dd** simplesmente copia seu arquivo de entrada para seu arquivo de saída. Se um usuário lhe trouxer uma fita gravada em algum sistema que não o Unix, **dd** pode ser a única maneira de se ler essa fita.

Uma utilização histórica de **dd** era criar uma cópia de sistema de arquivos inteiro. Entretanto, atualmente a melhor opção é fazer **newfs** para o sistema de arquivos de destino e, então executar o **dump** canalizado (**piped**) para **restore. dd** pode, às vezes, danificar as informações de particionamento se for utilizado incorretamente. Ele apenas pode copiar sistemas de arquivos entre partições que tenham exatamente o mesmo tamanho.

**dd** também pode ser utilizado para fazer uma cópia de uma fita magnética. Com duas unidades de fita, digamos, **/dev/rmt8 e /dev/rmt9**, você utilizaria o comando:

- **$> dd if=/dev/rmt8 of=/dev/rmt9 cbs=16b**

Com uma unidade (**/dev/rmt8**), a sequência de comandos será:

**dd if=/dev/rmt8 of=arquivo cbs=16b**
/*troca de fitas */
**dd if=arquivo of=/dev/rmt8 cbs=16b**
**rm arquivo**

Naturalmente, se tiver apenas uma unidade de fita, deve-se ter espaço suficiente em disco para armazenar uma imagem da fita inteira.

Outra utilização histórica do **dd** era fazer uma conversão entre várias versões de fita QIC que diferiam apenas pela sua ordem de **bytes**. Por exemplo, para ler em uma máquina da Sun uma fita **tar** gravada em uma máquina SGI. O comando seria:

**$> dd if=/dev/rst8 conv=swab | tar xf -**

## 5.7 Comando volcopy

**volcopy** faz uma cópia exata de um sistema de arquivos em outro dispositivo, alterando o tamanho do blocl conforme apropriado. Está disponível em sistemas Solaris, HP-UX e Linux. Pode-se utilizar o **volcopy** para fazer um **backup** de um sistema de arquivos em um disco removível ou fazer uma cópia completa de um sistema de arquivos em fita.

## 5.8 Comando mt

Este comando controla a unidade de fita e permite ao usuário posicionar a fita, retorná-la ou desativá-la.

**A** sua sintaxe é:
**mt [-f tapename ] comando ... [ contador ]**

**Onde:**

**-f** Indica a unidade de fita a ser acessada.

**contador** Por **default,** o comando **mt** executa a operação desejada uma vez. Operações múltiplas podem ser executadas especificando-se **contador.**

**comandos:**

**fsf** # Avançar a fita com o número de espaços especificados;

**bsf** # Retornar a fita com o número de espaços especificados;

**rew** Ou **rewind** , rebobina a fita até o início;

**offl** Ou **offline**, coloca a fita em **off-line;**

**status** Informações sobre o **status** da fita;

## 5.9 Comandos dump e restore

O comando **dump** oferece mais recursos para a realização de **backups**:

- **Backup** pode ser distribuído em múltiplas fitas
- Arquivos de qualquer tipo podem ser copiados e restaurados
- Permissões, dono e data podem ser restauradas
- **Backups** incrementais podem ser executados

O arquivo **/etc/dumpdates** contém informações acerca dos **dumps** realizados

**Limitações:**

- O **dump** deve ser feito separadamente em cada partição
- Aceita apenas sistemas de arquivo locais, mas pode usar unidade de fita remota
- No linux somente para sistemas **ext2**

**Opções :**

**0-9:** nível do **backup**

**u:** atualiza o **/etc/dumpdates** após o **backup**

**s:** tamanho da fita em **pés** (**feet**)

**d:** densidade da fita em **bpi**

**f:** nome do **device** onde deve ser feito o **backup**

**Exemplos (BSD):**

Backup de nível 3 com opções de fita **default** do sistema de arquivo **/home/users:**
**$> dump 3u /home/users**

Backup de nível 2, fita com 2300 ps e 6250 bpi, device = /dev/rmt1, do /home/data:
**$> dump 2usfd 2300 /dev/rmt1 6250 /home/data**

### 5.9.1 Restaurando Arquivos (restore)

Opções do **restore** (**rrestore - remoto**):

**r:** restaura um sistema de arquivos completamente

**x:** extrai os arquivos especificados

**f:** nome do device

**s:** determina qual fita do backup deve ser utilizada

**i:** modo interativo

**Exemplo 1:**

**$> cd /tmp**
**$> restore -x -f /dev/rmt1 home/adriana/a.out**
**$> ls /tmp/home/adriana**
**a.out**
**$>ls /home/users/adriana**
**mail teste/**
**$> cp /tmp/home/adriana/a.out /home/users/adriana**
**$> chown adriana /home/users/adriana/a.out**
**$> chgrp staff /home/users/adriana/a.out**

Seu arquivo **a.out** foi recuperado como pedido e foi colocado no seu diretório de trabalho

**Exemplo 2:**
**$> cd /tmp**
**$> restore -if /dev/rmt1**
**restore > ls home/ usr/ var/**
**restore> cd home**
**restore> ls adriana/ bia/ janaina/**
**restore> cd adriana**
**restore> ls**
**a.out c/ mail teste/**
**restore> add a.out**
**restore> extract**

You have not read any volumes yet.
Unless you know which volume your files are on you should
start with the last volume and work towards the first.

**Specify next volume#: 1**
**set owner /mode for .? [yn] n**
**restore> quit**

### 5.9.2 Devices para Fitas

O nome do device de acesso à fita varia bastante entre as diferentes plataformas

**Tipos de devices :**
normal
no rewind
densidade baixa, mdia, alta e ultra

Exemplos de Nomes de Devices
**AIX**
**/dev/rmt0: rebobinar no final**
**/dev/rmt0.1: não rebobina**
**Solaris**
**/dev/rmt/0: normal**
**/dev/rmt/0n: sem rebobinar**
**FreeBSD**
**/dev/rst0: normal**
**/dev/nrst0: sem rebobinar**

### 5.9.3 Vários Backups em uma Mesma Fita

É possível colocar mais de um **backup** na mesma fita através do comando **mt**.
O comando abaixo avança para o terceiro **backup** da fita:
# **mt -f /dev/nrst0 fsf 2**
# **mt rewind**
# **mt fsf 2**
# **dump ...**
# **mt bsf 1**
# **restore ...**

## 5.10 Ferramentas para backup

Há muitos aplicativos para sistemas de **Backup.** Para Software Livre, podem ser citados:

- backup2l (http://backup2l.sourceforge.net)
- Amanda (http://www.amanda.org)
- bacula (http://www.bacula.org)

- O site abaixo mostra uma completa lista para Linux:

http://www.linux.org/apps/all/Administration/Backup.html

- Softwares comerciais também podem ser encontrados:
- ADSM/TSM - Tivoli Storage Manager (TSM) - IBM
- VERITAS (www.veritas.com)
- Legato ( www.legato.com)

## 5.11 Scripts para Backup

A seguir são mostrados dois exemplos de **scripts** para se fazer **backup.**
O primeiro faz **backup** em fitas DAT e o Sistema Operacional é FreeBSD 5.1
O segundo faz **backup** em disco e o Sistema Operacional é Linux

```
# Autor:
#Fabiano Caixeta Duarte
#Seção Técnica de Informática
#FEA-RP/USP
#
#!/usr/bin/bash
########
## Definição das constantes
########
LOGGER="/usr/bin/logger -t backup"
PG_DUMP_FILE=/tmp/dump.psql
TARGET="/etc /home /usr/local /var/log /var/mail /var/www /var/exceeded $PG_DUMP_FILE"
CUR_DATE="date +%d/%m/%Y[%H:%M]"
########
## Loga o início do backup
########
$LOGGER Iniciando processo de backup
########
## Testa se há fita no drive
########
mt fsf 1
if [ `echo $?` -eq 1 ]; then
$LOGGER Backup cancelado! Não há fita no drive
echo "Falta fita no drive." | mail -s "[ATENÇÃO] Backup não efetuado!" sti
exit
fi
mt bsf 1
########
## Dump de todas as bases + estrutura do banco
########
$LOGGER Gerando dump do pgsql
/usr/local/bin/pg_dumpall -c -U pgsql > $PG_DUMP_FILE
########
## Gera backup em fita
########
$LOGGER Gerando backup em fita
tar cpf /dev/sa0 –exclude /home/ftp –totals $TARGET 2> /tmp/tar.totals
########
## Fim do backup
########
#Rebobina e ejeta a fita
mt rewind
mt offline
#Registro no log
$LOGGER Backup concluído - Total: `grep Total /tmp/tar.totals | cut -d: -f2`
#Remove o dump do PostgreSQL
rm $PG_DUMP_FILE
#Remove arquivo de totais do tar
rm /tmp/tar.totals
```

```
#Autor : Thiago M. Zerbinato
#thiagozerbinato@yahoo.com.br
#Oracle Certified Professional 8i
#GNU/Linux User #286429 / Debian User #534
#http://www.thiagomz.hpg.com.br
# Cria backup Incrementais Diariamente e
# Full uma vez por semana dos Emails dos usuarios.
#
#!/bin/bash
#
# Data Criacao - 30/05/2004
# Ultima alteracao - 08/06/2004
dados="/home /var/mail /etc /root"
list="/tmp/backlist_$$.txt"
data_ini=`date +%d-%m-%Y_%H.%M`
data_log=`date +%d-%m-%Y`
backup_title="backup-ruby-$data_ini"
file_log="/tmp/$backup_title.log"
tipo_bkp="/tmp/tipo_bkp"
file_del="/tmp/file_del"
message="/tmp/message"
#
set $(date)
#
if test "$1" = "Sun" ; then
# Backup Full Domingo
echo "Backup Full do Servidor Ruby" > $file_log
echo "Backup Full do Servidor Ruby" > $tipo_bkp
echo " " >> $file_log
echo "Inicio as $data_ini" >> $file_log
echo " " >> $file_log
tar cfz "/mnt/backup/bkp_ruby/backup_full_$data_log.tgz" $dados >> $file_log
echo "Os Diretorio Incluidos sao :\n $dados ">> $file_log
elif test "$1" = "Wed" ; then
# Backup Full Quarta
echo "Backup Full do Servidor Ruby" > $file_log
echo "Backup Full do Servidor Ruby" > $tipo_bkp
echo " " >> $file_log
echo "Inicio as $data_ini" >> $file_log
echo " " >> $file_log
tar cfz "/mnt/backup/bkp_ruby/backup_full_$data_log.tgz" $dados >> $file_log
echo "Os Diretorio Incluidos sao :\n $dados ">> $file_log
elif test "$1" = "Fri" ; then
# Backup Full Sexta
echo "Backup Full do Servidor Ruby" > $file_log
echo "Backup Full do Servidor Ruby" > $tipo_bkp
echo " " >> $file_log
echo "Inicio as $data_ini" >> $file_log
echo " " >> $file_log
tar cfz "/mnt/backup/bkp_ruby/backup_full_$data_log.tgz" $dados >> $file_log
echo "Os Diretorio Incluidos sao :\n $dados ">> $file_log
else
# Backup Diario Incremental:
#
```

```
echo "Backup Incremental do Servidor Ruby" > $file_log
echo "Backup Incremental do Servidor Ruby" > $tipo_bkp
echo "Inicio as $data_ini" >> $file_log
echo " " >> $file_log
find $dados -depth -type f \( -ctime -1 -o -mtime -1 \) -print > $list
tar cfzT "/mnt/backup/bkp_ruby/backup_incremental_$data_log.tgz" "$list" >> $file_log
fi
# Fim do Backup
data_fim=`date +%d-%m-%Y_%H.%M`
#Lista o backup
cat "$list" >> $file_log
echo " ">> $file_log
echo "e terminou as $data_fim" >> $file_log
# Removendo arquivos de backup com mais de 7 dias
find /mnt/backup/bkp_ruby/ -ctime +7 > $file_del
find /mnt/backup/bkp_ruby/ -ctime +7 -exec rm \{} \;
# Envia email para o Administrador
echo "From: bkp-ruby@ruby" >$message
echo "To: root@logisticaeprocessos.com.br">>$message
echo "Subject: [backup] $backup_title">>$message
echo " ">>$message
echo `cat $tipo_bkp` >> $message
echo "———————— " >> $message
echo "Segue em anexo o log do backup" >>/tmp/message
echo " " >>$message
echo "Arquivo de Backup ==> $backup_title" >>$message
echo " " >>$message
echo "Iniciado as $data_ini" >>$message
echo " " >>$message
echo "Terminou as $data_fim" >>$message
echo " " >>$message
echo "Arquivos de Backup Removidos com + de 7 dias " >>$message
echo `cat $file_del` >> $message
echo " " >>$message
echo "Espaço usado no disco de backup - Total de 80Gb" >>$message
echo `du -sh /mnt/backup/` >>$message
echo " " >>$message
zip /tmp/mail_list.zip $file_log
uuencode /tmp/mail_list.zip mail_list.zip >>$message
exec cat $message<<EOF| /usr/lib/sendmail -i -t -B8BITMIME
EOF
sleep 15
# Removendo os Arquivos Temporarios
rm -f "$file_del"
rm -f "$file_log"
rm -f "$tipo_bkp"
rm -f /tmp/mail_list.zip
rm -f $message
rm -f "$list"
```

**Referências bibliográficas**


- IBM RedBooks :

http://www.redbooks.ibm.com/abstracts/sg246228.html

- Dica para usar fitas DAT no Linux:

http://www.ent.com.br/index.asp?cod=12

- Fazendo Cópias de Segurança e restaurando dados:

http://br.tldp.org/projetos/howto/arquivos/html/ftape/ftape.pt_BR-622.html

- Linux Administratin / Backup:

http://www.linux.org/apps/all/Administration/Backup.html