



# ***Administração de Redes em Software Livre***

***Marcos Monteiro***

***<http://www.marcosmonteiro.com.br>  
[contato@marcosmonteiro.com.br](mailto:contato@marcosmonteiro.com.br)***

Marcos Monteiro





# Software Livre

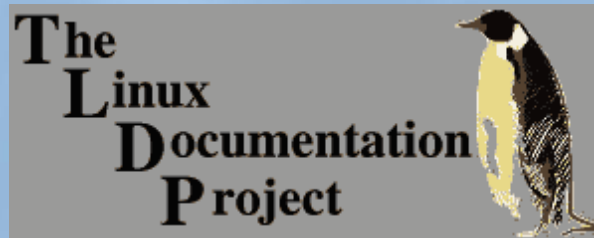
## Definição e História





# Documentação Vasta

- Comando **man**, **help** e **info**;



- <http://tldp.org/>

- <http://www.guiafoca.org/>





# Tipos de Arquivos

- **Arquivo regular:**

- tipo comum que contém dados somente. Os arquivos regulares podem ser dos mais variados tipos, guardando os mais diferentes tipos de informações. Existem arquivos de áudio, vídeo, imagem, texto, enfim. Os arquivos se dividem em 2 (duas) categorias principais:

- Binários:

- Arquivos binários são compostos por bits 1 e 0 e só podem ser interpretados pelo sistema operacional, sendo incompreensíveis por nós humanos. Se um arquivo binário for aberto por um editor de textos, serão exibidos vários símbolos estranhos. Programas e bibliotecas são exemplos de arquivos binários;

- Texto:

- Arquivos do tipo texto são compostos por informações em forma de texto, que podem ser entendidas pelo usuário comum. Arquivos desse tipo não necessariamente contém texto propriamente dito. Quando se diz que um arquivo é do tipo texto estamos querendo dizer que, se o abrirmos num editor de texto, serão exibidos informações legíveis (ainda que possam não fazer muito sentido).





# Tipos de Arquivos

- **Diretórios:**

- Diretórios são utilizados para separar um grupo de arquivos de outros. Um diretório pode conter arquivos e outros diretórios, que serão chamados subdiretórios;

- **Dispositivos:**

- todo componente de hardware instalável é chamado dispositivo. Placas de vídeo, som, rede, drives de CD-ROM, tudo o que se liga na interface USB, memória RAM, são dispositivos. Os dispositivos podem ser:

- De bloco:

- Dispositivos de bloco utilizam buffer para leitura/gravação. Geralmente são unidades de disco, como HD's, CD's, etc;

- De caracter:

- esses dispositivos não utilizam buffer para leitura/gravação. A maioria dos dispositivos PCI e outros dispositivos como impressoras, mouse, etc. são do tipo caracter;

- Fifo:

- trata-se de um canal de comunicação, através do qual pode-se ver os dados que estão trafegando por um dispositivo;





# Tipos de Arquivos

- **Links:**

- Os links são arquivos utilizados para fazer referência a um outro arquivo localizado em outro local. Em outras palavras, são atalhos. Os links podem ser de 2 (dois) tipos:

- Simbólicos:

- fazem uma referência ao arquivo através de seu endereço lógico no disco ou memória. São os links mais comuns;

- Absolutos:

- Fazem referência ao arquivo através do seu endereço físico no disco rígido ou memória.

- **FIFO:**

- canal de comunicação, utilizado para direcionar os dados produzidos por um processos para um outro processo.





# Tipos de Diretórios

- Os sistemas GNU/Linux possuem um padrão rígido e específico a respeito da organização hierárquica dos diretórios, definido pela Filesystem Hierarchy Standard, ou simplesmente FHS, constante da LSB.
- `/bin/`
  - aqui encontram-se os programas que são usados freqüentemente pelos usuários;
- `/boot/`
  - arquivos estáticos utilizados durante a inicialização do sistema;
- `/dev/`
  - encontramos aqui os dispositivos de hardware. Existem vários arquivos, um para cada dispositivo;
- `/etc/`
  - nesse diretório ficam os arquivos de configuração do sistema e dos programas instalados;
- `/home/`
  - os usuários cadastrados no sistema possuem um diretório com seu nome dentro de `/home`. Em geral é a única área acessível aos usuários para gravarem seus arquivos;





# Tipos de Diretórios

- `/lib/`
  - as bibliotecas essenciais e os módulos do kernel Linux ficam aqui. Bibliotecas são conjuntos de funções e recursos utilizados por programas;
- `/media`
  - aqui devem ser inseridos os pontos de montagem para as mídias removíveis, como CD's e disquetes;
- `/mnt/`
  - diretório utilizado para conexão com volumes presentes em outros computadores da rede ou para acessar dispositivos removíveis. Em outras palavras, aqui ficam os pontos de montagem temporários;
- `/opt/`
  - softwares adicionais instalados de maneira não padrão devem ficar aqui;
- `/root`
  - diretório pessoal do usuário root, o administrador do sistema;
- `/sbin/`
  - local dos programas essenciais para o funcionamento e manutenção do sistema. Somente o administrador (root) tem permissão de executar esses programas;







# Tipos de Diretórios

- `/srv`
  - dados dos serviços fornecidos pelo sistema;
- `/tmp`
  - aqui se encontram os arquivos temporários gerados pelo sistema;
- `/usr/`
  - os arquivos acessados pelos usuários se encontram contidos neste diretório, principalmente programas e os arquivos utilizados por esses programas;
- `/var/`
  - aqui se encontram informações variáveis do sistema, como spool de impressora, caixas postais, logs do sistema, cache de programas, etc.





# principais subdiretórios

- `/etc/X11/`
  - arquivos de configuração do servidor X;
- `/etc/skel/:`
  - arquivos gravados no diretório pessoal do novo usuário;
- `/etc/network/:`
  - arquivos de configuração da rede;
- `/lib/modules/:`
  - diretório dos módulos do kernel Linux. Dentro do diretório modules existirá um subdiretório para cada versão de kernel disponível no sistema. O nome desse subdiretório será a versão do kernel correspondente;
- `/media/floppy/:`
  - ponto de montagem da unidade de disquete principal;
- `/media/cdrom/:`
  - ponto de montagem da unidade de CD-ROM principal;
- `/usr/bin/:`
  - programas utilizados pelos usuários;
- `/usr/lib/:`
  - bibliotecas utilizadas pelos programas em `/usr/bin`;
- `/var/spool/:` diretório de spool dos processos.





# principais subdiretórios

- `/usr/sbin/`:
  - programas utilizados pelo administrador do sistema;
- `/usr/src/`:
  - códigos-fonte de programas devem ficar aqui, incluindo do kernel Linux;
- `/usr/X11R6/`:
  - programas, bibliotecas e demais arquivos do servidor X;
- `/var/cache/`:
  - aqui ficam armazenados os arquivos de cache utilizados por diversos programa. Dentro do diretório `cache/` são criados diretórios com o nome dos processos criadores;
- `/var/lock/`:
  - usado para bloquear a execução de programas. Se dentro desse diretório houver um arquivo com o nome de um determinado processo, ele não poderá ser executado;
- `/var/log/`:
  - arquivos de log gerados pelos sistema e outros programas;
- `/var/mail/`:
  - mensagens enviadas ao usuário root e emails ficam armazenados aqui (requer configuração);
- `/var/run/`:
  - dados dos programas em execução;





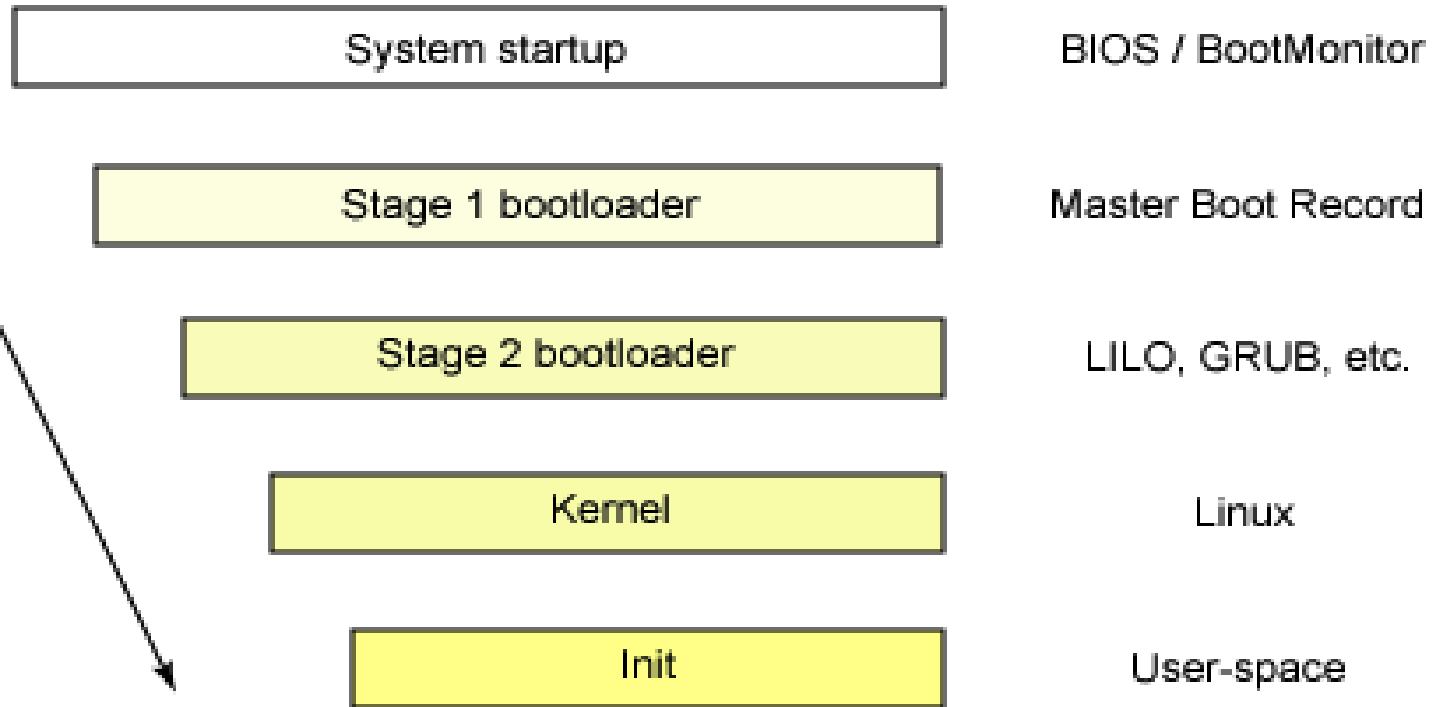
# Instalando o linux





# Boot

Power-up / Reset



Operation





# Modos de Inicialização

Inicialização	Scripts	Função
Nível 0	/etc/rc.d/rc0.d	desligamento da máquina
Nível 1	/etc/rc.d/rc1.d	ativa o modo de manutenção ou monousuário
Nível 2	/etc/rc.d/rc2.d	ativa o modo multiusuário modo texto
Nível 3	/etc/rc.d/rc3.d	entra em modo multiusuário modo texto com suporte a NFS
Nível 4	/etc/rc.d/rc4.d	não é usado nas distribuições Debian e Red Hat
Nível 5	/etc/rc.d/rc5.d	responsável pelo carregamento do ambiente gráfico
Nível 6	/etc/rc.d/rc6.d	reinicialização da máquina



Normalmente, os computadores pessoais utilizam o Runlevel nível 5, já os servidores que não precisam de interface gráfica, utilizam o Runlevel nível 3.

Os níveis 0 e 6, não podem ser usados na configuração de inicialização, caso contrário, o sistema nunca será carregado.

Os níveis de execução dos Runlevel podem ser mudados com o computador em funcionamento, basta executar o comando:

```
# init x
```

Onde 'x' varia de 0 à 6.

O arquivo de configuração do Runlevel é o `"/etc/inittab"`. Para mudar o nível que o sistema irá executar, basta modificar a linha:

```
id:5:initdefault:
```

No exemplo, está inicializando com o Runlevel nível 5, que carrega a interface gráfica. Poderíamos colocar para inicializar no Runlevel nível 3, basta mudar o número, deixando da seguinte maneira:

```
id:3:initdefault:
```

Comando `runlevel` mostra qual nível está iniciado.





# Comandos do sistema de arquivos

`cd`

Muda o diretório de trabalho (diretório no qual o script irá atuar durante sua execução).

`pwd`

Retorna o diretório de trabalho corrente.







# Comandos do sistema de arquivos

`cd`

Muda o diretório de trabalho (diretório no qual o script irá atuar durante sua execução).

`pwd`

Retorna o diretório de trabalho corrente.





# Comandos básicos - ls

Comando básico para listar o conteúdo de um diretório. Alguns parâmetros:

-R

lista os subdiretórios recursivamente

-S

ordena pelo tamanho do arquivo

-t

ordena pela hora de modificação

-r

reverte a ordenação

-a

mostra arquivos ocultos (iniciados por .)

-h

mostra em formato mais apropriado para leitura humana

-l

utiliza formato longo para mostrar atributos de arquivos





cat

Envia o conteúdo do arquivo para a saída padrão (por padrão o monitor). O parâmetro `-n` faz com que as linhas do arquivo sejam numeradas.

tac

Similar ao `cat`, porém as linhas do arquivo são mostradas da última para a primeira.

rev

Envia o conteúdo do arquivo para a saída padrão (por padrão o monitor), mas mostra as linhas de trás para frente. A ordem das linhas são mantidas, porém são mostrados do último ao primeiro caractere da linha.





## User Mask – umask e chmod

```
508762 -rw-r--r-- 1 marcos marcos 220 Jul 13 17:41 .bash_logout
508767 -rw-r--r-- 1 marcos marcos 3184 Jul 13 17:41 .bashrc
637140 drwxr-xr-x 5 marcos marcos 4096 Jul 13 17:55 .config
637143 drwx----- 3 marcos marcos 4096 Jul 13 17:42 .dbus
508777 -rw-r--r-- 1 marcos marcos 41 Jul 13 21:52 .dmrc
637136 drwxr-xr-x 2 marcos marcos 4096 Jul 13 17:42 Documentos
637133 drwxr-xr-x 2 marcos marcos 4096 Jul 13 17:42 Downloads
637148 drwx----- 4 marcos marcos 4096 Jul 13 21:52 gconf
```

umask

Para alterar o valor de umask, basta incluir o novo valor após o comando umask. Por exemplo, o comando

```
umask 002
```

define que, por padrão, novos arquivos terão permissão 664, enquanto novos diretórios terão permissão 775.

Suponha que o usuário aluno tenha definido o umask como 002. Portanto, quando este usuário cria o arquivo teste, temos as seguintes permissões

```
-rw-rw-r-- 1 aluno aluno 0 May 28 17:20 teste
```

que corresponde ao valor 664 (666 - 002).





cp

Cópia de arquivos. Alguns parâmetros:

- f força a cópia
- i interativo (pergunta antes de sobrescrever)
- R, -r copia diretórios recursivamente
- v mostra os arquivos que estão sendo copiados

mv

Move arquivos. Também utilizado para renomear. Alguns parâmetros:

- f força a movimentação
- i interativo (pergunta antes de sobrescrever)
- v mostra os arquivos que estão sendo movidos





rm

Remove (deleta) arquivos. Alguns parâmetros:

- f força a remoção
- i interativo (pergunta antes de remover)
- R, -r remove diretórios recursivamente
- v mostra os arquivos que estão sendo removidos

rmdir

Remove diretórios vazios.





# Montagem mount

`mount -t [tipo] [caminho da partição] [ponto de montagem]`

`mount -t vfat /dev/hda1 /mnt/win`

`umount`

`/etc/fstab`

`fdisk -l` e `df -h`





`mkdir`

Cria diretório.

`chmod`

Muda os atributos de arquivos e diretórios.

`ln`

Cria links para arquivos e diretórios.

`man, info`

Fornece informações sobre o funcionamento dos comandos externos.







## Comandos para manipulação de data e hora

date

Mostra a data e a hora do sistema. Também é utilizados para acertar a data/hora.

time

Mostra estatísticas de utilização de tempo do processo executado como parâmetro.

cal

Mostra um calendário.

hwclock, clock

Consulta ou acerta a hora no relógio do hardware





# Comandos para processamento de texto

sort

Mostra um arquivo em ordem alfabética.

uniq

Remove linhas duplicadas de um arquivo ordenado.

head

Envia as primeiras linhas de um arquivo para a saída padrão.

tail

Envia as últimas linhas de um arquivo para a saída padrão. Quando utilizado com o parâmetro -f o comando continua mostrando novas linhas que são adicionadas ao arquivo (útil para monitorar arquivos de log).

wc

Conta a quantidade de linhas, palavras e caracteres em um arquivo.

nl

Numera as linhas de um arquivo.





# Comandos para arquivamento

tar

Junta vários arquivos em um único arquivo ou dispositivo.

gzip

Faz a compressão de um arquivo. Normalmente utilizado em conjunto com o tar quando é necessário fazer a compressão de mais de um arquivo.

bzip2

Outro programa para compressão de arquivo. Normalmente mais eficiente que o gzip.

zip, unzip

Programas para manipulação de arquivo comprimido no formato zip.

arj, unarj

Programas para manipulação de arquivo comprimido no formato arj.

rar, unrar

Programas para manipulação de arquivo comprimido no formato rar.





# Comandos que atuam sobre variáveis

unset

Remove o valor de variáveis internas do shell. Na prática as coloca com valor null.

export

Torna a variável passada como parâmetro disponível para processos filho.

readonly

Torna uma variável disponível somente para leitura (a torna uma constante). Tentativas de alterar seu valor irão provocar uma mensagem de erro.





# Comandos para controle de jobs

jobs

Lista os processos executando em segundo plano no shell, fornecendo o número do job.

fg

Trás para o primeiro plano um job que está executando em segundo plano.

bg

Coloca em execução em segundo plano um job cuja execução tenha sido suspensa.

kill

Força o encerramento de um processo enviando ao processo um código de encerramento.

killall

Força o encerramento de um processo passado a ele o nome do processo.





# Arquivos

file

Identifica o tipo de um arquivo.

which

Mostra em que diretório se encontra determinado arquivo.

diff

Compara 2 arquivos e mostra as diferenças entre eles.

sum, cksum, md5sum, sha1sum, sha256sum

Comandos para gerar hash de arquivos a fim de verificar sua integridade.

more, less

Pagina arquivos e os envia para a saída padrão. O comando less permite um maior controle sobre a paginação.





# Comandos do sistema

`su` e `sudo` e `sudoers`

Executa um programa como um usuário substituto ou inicia um shell como outro usuário.

`uname`

Exibe informações sobre o sistema.

`free`

Mostra o total de memória RAM e swap utilizada pelo sistema.

`du`

Mostra o total de espaço em disco utilizado (recursivamente) por arquivos e/ou diretórios.

`df`

Mostra a utilização dos discos/partições.

`dmesg`

Mostra todas as mensagens emitidas durante o processo de boot do sistema.

`uptime`

Mostra por quanto tempo o sistema está em execução.





# Redirecionamento

Para todo processo existem sempre 3 arquivos abertos por padrão:

stdin ? entrada padrão (teclado)

stdout ? saída padrão (monitor)

stderr ? saída de erros (monitor)

Estes e quaisquer outros arquivos abertos podem ser redirecionados. Redirecionamento significa capturar a saída de um arquivo, comando, programa, script ou até mesmo um bloco de código em um script e o enviar como entrada para outro arquivo, comando, programa ou script.







Cada arquivo aberto recebe um descritor de arquivo. Os descritores de arquivo para stdin, stdout e stderr são 0, 1 e 2, respectivamente.

O “>” redireciona a saída padrão (stdout) para um arquivo. Se o arquivo não existir ele será criado, caso contrário será sobrescrito e seu conteúdo anterior será perdido.

O “>>” também redireciona a saída padrão (stdout) para um arquivo. Se o arquivo não existir ele será criado. Porém, caso o arquivo exista, a saída do comando irá ser acrescentada ao final do arquivo.

O “2>” redireciona a saída de erros (stderr) para um arquivo. Se o arquivo não existir ele será criado, caso contrário será sobrescrito e seu conteúdo anterior será perdido.

O “2>>” também redireciona a saída padrão (stdout) para um arquivo. Se o arquivo não existir ele será criado. Porém, caso o arquivo exista, a saída do comando irá ser acrescentada ao final do arquivo.





# Configurando Rede





# Configurando Interface

- Interface
  - Ifconfig eth0 IP/mask
  - Ifconfig eth0 <up> ou <down>
- DNS
  - /etc/resolv.conf
    - nameserver IP
- Gateway
  - route add default gw IP
- Proxy
  - export https\_proxy=http://username:password@proxy:porta
  - export http\_proxy=http://username:password@proxy:porta
  - export ftp\_proxy=http://username:password@proxy:porta





# Configurando Interface

- Como Serviço no Debian
  - /etc/network/interfaces

```
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet static # use dhcp ao invés de static, se tiver um server dhcp
address x.x.x.x # troque x pelo seu ip
netmask x.x.x.x # troque x pela sua mascara
gateway x.x.x.x
dns-nameservers x.x.x.x
```





# Configurando Interface

- Como Serviço no Debian
  - /etc/network/interfaces

```
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet static # use dhcp ao invés de static, se tiver um server dhcp
address x.x.x.x # troque x pelo seu ip
netmask x.x.x.x # troque x pela sua mascara
gateway x.x.x.x
dns-nameservers x.x.x.x
```





# Outros comandos para rede

- dhclient
- ping
- traceroute
- netstat
- arp





▪

# Instalando programas





# A partir de um código fonte

- \$ ./configure
- # make
- # make install

e é só :D

obs: \$ é seu usuário e # é o root







# Pacotes .rpm (Red Hat)

- instalar:
  - **rpm -i pacote.rpm**
- atualizar:
  - **rpm -Uvh pacote.rpm**
- desinstalar:
  - **rpm -e pacote.rpm**
- para forçar o pacote:
  - **rpm -i pacote.rpm --force**
- instalar s/ dependências:
  - **rpm -i pacote.rpm --nodeps**
- verificar pacote instalado:
  - **rpm -q pacote.rpm**





# Pacotes .deb (Debian)

- `dpkg -i <pacote>`
  - Instala um pacote.
- `dpkg -r <pacote>`
  - Desinstala um pacote.
- `dpkg -P <pacote>`
  - Desinstala um pacote e os seus arquivos de configuração.
- `dpkg -l <padrão>`
  - Lista pacotes que combinam com <padrão>.
- `dpkg -L <pacote>`
  - Lista arquivos instalados no sistema a partir de <pacote>
- `dpkg -I <pacote>`
  - Mostra informações sobre o pacote.
- `dpkg -c <pacote>`
  - Mostra o conteúdo do pacote.





# Pacotes .deb (Debian)

- `dpkg -s <pacote>`
  - Mostra o estado do pacote.
- `dpkg -S <padrão>`
  - Procura por um pacote a partir dos pacotes instalados.
- `dpkg -C`
  - Procura pacotes que foram parcialmente instalados. O `dpkg` sugerirá o que fazer para que funcionem.
- `dpkg --get-selections <padrão>`
  - Obtém uma lista dos pacotes instalados que contenham `<padrão>` no nome. Caracteres especiais como `*` e `?` podem ser utilizados.
- `dpkg-reconfigure <pacote>`
  - Reconfigura um pacote instalado.
- `/etc/dpkg/dpkg.cfg` Arquivo de configuração do `dpkg`.
- `/var/lib/dpkg/available` Lista de pacotes disponíveis.
- `/var/lib/dpkg/status` Contém o estado dos pacotes disponíveis.





# APT (Advanced Package Tool)

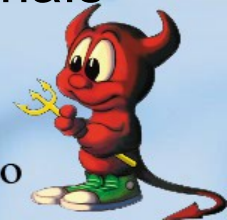
- apt-get
  - apt-get update
    - Actualiza a lista de pacotes disponíveis nos servidores. Deve ser executado antes de começar a instalar ou actualizar pacotes, e sempre que o arquivo sources.list for modificado.
  - apt-get install
    - Instala ou actualiza um pacote e todas as suas dependências.
  - apt-get remove [–purge]
    - Remove um pacote e todos os outros pacotes que dependem dele. A opção –purge faz com que os arquivos de configuração do pacote sejam removidos também.





# APT (Advanced Package Tool)

- apt-get
  - apt-get upgrade
    - Actualiza todos os pacotes do sistema, excepto aqueles que possuam mudanças nas suas dependências. Dependências novas não serão instaladas e o pacote não será actualizado.
  - apt-get dist-upgrade
    - Semelhante ao apt-get upgrade, porém instala e remove pacotes para satisfazer as novas dependências.
  - apt-get clean
    - Limpa a cache de pacotes do sistema (/var/cache/apt/archives).
  - apt-get autoclean
    - Funciona semelhante ao apt-get clean. A diferença é que apaga da cache somente os pacotes que não podem mais ser baixados (versões antigas de pacotes).





# APT (Advanced Package Tool)

- apt-cache
  - apt-cache search
    - Procura pacotes e descrições por <padrão>.
  - apt-cache show
    - Retorna a descrição completa de <pacote>.
  - apt-cache showpkg
    - Retorna mais detalhes sobre <pacote> e seu relacionamento com outros pacotes.
  - apt-cache depends
    - Retorna os pacotes que são dependência de <pacote>





# APT (Advanced Package Tool)

- apt-cdrom

- Usado para adicionar um novo CD-ROM na lista de servidores de pacotes (/etc/apt/sources.list). Dessa forma, o comando apt-get pode instalar pacotes contidos no CD-ROM.
- apt-cdrom add
  - Adiciona o CD-ROM à sources.list. Isto desmontará o dispositivo de CD-ROM, pedirá para inserir o CD de pacotes, e então irá procurar no CD arquivos de índice.





# APT (Advanced Package Tool)

- apt-file
  - Ferramenta para busca de pacotes através de arquivos contidos no pacote. É semelhante ao uso de "dpkg - S", porém devolve pacotes não instalados também. Para isso possui uma base de dados com informações sobre cada pacote de servidores contidos no sources.list. A atualização da base de dados é feita em conjunto com o pacote "curl", que precisa de ser instalado também, mesmo não sendo uma dependência de apt-file.
- apt-file update
  - Sincroniza a base de dados do apt-file.
- apt-file search
  - Procura por um pacote através de um arquivo integrante.
- apt-file list
  - Lista todos os arquivos contido em um pacote.







# APT (Advanced Package Tool)

- Consertando dependências

Problemas de conflitos entre dependências e instalações interrompidas podem ocorrer algumas vezes, impossibilitando a instalação de novos pacotes. Quando ocorre algum erro desse tipo, o sistema geralmente imprime uma mensagem no ecrã, aconselhando a execução dos seguintes comandos capazes de resolver esses conflitos:

- `apt-get install -f`
- `apt-get upgrade -f`
- `apt-get dist-upgrade -f`
- `dpkg --configure -a`





# Nossa source list

- Em **`/etc/apt/sources.list`**

```
deb http://192.168.10.1:3142/ftp.br.debian.org/debian/ squeeze main
deb http://192.168.10.1:3142/security.debian.org/ squeeze/updates main
```

Depois não esqueça do **`apt-get update`**





# E no RedHat?

No mundo Red Hat, há o comando **yum**,  
muito parecido com o apt





▪

# Gerenciando usuários e grupos





# Gerenciando usuários e grupos

- **useradd:**
  - Comando utilizado para criação de um usuário.
- **userdel:**
  - Comando utilizado para remoção de um usuário.
- **usermod:**
  - Comando usado para modificar os dados de um usuário.
- **passwd:**
  - Comando usado para definir e ou modificar a senha de um usuário.





# Gerenciando usuários e grupos

- **groupadd:**
  - Comando usado para criar um grupo.
- **groupdel:**
  - Comando usado para remover um grupo.
- **groupmod:**
  - Comando usado para modificar os dados de um grupo.





# useradd

- Sintaxe: `useradd [opções] <username>`
  - -d - Caminho do diretório home do usuário.
  - -g - Especifica o grupo do usuário.
  - -c - Inclui um comentário referente ao usuário, tais como nome, setor, etc
  - -s - Especifica o shell de comando que o usuário irá utilizar.
  - -m - Cria o diretório home do usuário e copia os arquivos de `/etc/skel/` para o home criado.
  - -p - Essa opção serve para especificarmos uma senha já criptografada para o usuário.

```
# useradd -g admin -s /bin/bash -d /home/sup1 -c "Usuário Suporte 1" -m sup1
```

No exemplo acima criamos o usuário `sup1`, que tem como grupo `admin`, usando o shell `/bin/bash`, o home criado foi o `/home/sup1` e tem o comentário "Usuário Administrativo de Suporte 1".





# passwd

- Sintaxe: `passwd [opções] <username>`
  - -d - Permite o usuário acessar (logar) o sistema sem senha.
  - -l - Bloqueia/trava a conta do usuário. O usuário não consegue logar.
  - -u - Desbloqueia/destrava a contado usuário (bloqueado pela opção "-l").
  - -S - Mostra o status da senha do usuário.
- `# passwd sup1`
- `# passwd -l sup1`
- `# passwd -u sup1`







# usermod

## usermod [opções] <username>

- -d - Modifica o caminho do diretório home do usuário.
- -g - Modifica o grupo do usuário.
- -c - Modifica o comentário referente ao usuário.
- -s - Modifica o Shell de comando que o usuário irá utilizar.
- -p - Substitui a senha já criptografada do usuário.

– Exemplo 1: Nesse exemplo estamos modificando o grupo e o comentário do usuário sup1 ao mesmo tempo.

```
# usermod -g <novoGrupo> -c "<novoComentario>" sup1
```





# userdel

## userdel [opções] <username>

- -r - Ao usarmos essa opção o diretório HOME e Mailbox do usuário será removido. É importante ter certeza ao fazer isso, pois muitas vezes é melhor remover apenas o usuário ou até mesmo suspendê-lo mantendo seus arquivos para auditoria.
- Exemplo 1: Remover o usuário sem excluir seus arquivos.
  - # userdel sup1
- Exemplo 2: Remover o usuário e seus arquivos
  - # userdel -r sup1





# groupadd

- `groupadd [opções] <groupname>`
  - `-g` - Ao usarmos esta opção, podemos especificar o GID do grupo que estamos criando.
- Exemplo 1: Criando um grupo chamado "administracao".  
`# groupadd administracao`
- Exemplo 2: Criando um grupo chamado oragroup e especificando o GID 1521.  
`# groupadd -g 1521 oragroup`





# groupmod

- `groupmod [opções] <groupname>`
  - `-g` - Ao usarmos esta opção, podemos modificar o GID do grupo.
  - `-n` - Para trocarmos o nome do grupo.
- Exemplo 1: Modificando o GID do grupo "administracao".  

```
# groupmod -g 666 administracao
```
- Exemplo 2: Modificando o nome do grupo oragroup.  

```
# groupadd -n oracle oragroup
```





# Diretório e arquivos

- Diretório:
  - /etc/skel
    - Neste diretório são armazenados arquivos, por padrão ocultos (arquivos que iniciam com um "."), que são copiados para o diretório HOME do usuário no momento da criação do usuário. Se precisarmos incluir alguma configuração padrão, podemos usar esse diretório para incluir ou até mesmo editar os arquivos existentes e conseqüentemente fazer o ajuste a suas necessidades.





# Diretório e arquivos

- Arquivos:
  - /etc/passwd
    - Arquivo que contém várias informações sobre o usuário. Ele é utilizado por vários comandos de sistema e aplicações. Antigamente até mesmo as senhas eram armazenadas nele, porém a algum tempo as senhas estão sendo armazenadas em /etc/shadow, arquivo que falaremos a seguir. Só o administrador do sistema consegue modificar esse arquivo.
  - /etc/shadow
    - Onde estão armazenadas as senhas criptografadas dos usuários, além de outras informações como expiração da senha etc.





# Diretório e arquivos

## – /etc/gshadow

- Tem a mesma finalidade do /etc/shadow, só que para grupos e não usuários.

## – /etc/group

- É onde se encontram os grupos existentes no sistema. Cada grupo pode estar associado a vários usuários, este arquivo também é responsável por esta associação.

## – /etc/motd

- Esse arquivo contém as informações que serão exibidas após o login do usuário.

## – /etc/default/useradd e /etc/login.defs

- Arquivos onde se encontram as configurações default de criação de usuários. As configurações podem ser diferentes dependendo de cada "distro", umas por exemplo, não há a necessidade da opção "-m" para a criação do diretório HOME do usuário.





# Gerenciando grupos e senhas

- `gpasswd`
  - Com esse comando podemos criar senhas para administradores de grupo que terão poder de incluir e remover, além de modificar a senha de um grupo.
- `grpconv`
  - Através desse comando é criado o arquivo `gshadow` utilizando o arquivo `group` para isso.
- `grpunconv`
  - Esse comando recria um arquivo `group` utilizando os arquivos `group` e `gshadow` e após isso ele remove os dois arquivos.







# Gerenciando grupos e senhas

- pwconv
  - Utilizando o arquivo passwd ele cria o arquivo shadow, hoje em dia esse comando é pouco utilizado, pois os novos sistemas já trabalham diretamente com senhas criptografadas já localizadas no shadow.
- pwunconv
  - Recria o arquivo passwd utilizando os arquivos passwd e shadow e depois os removendo.





# Gerenciando grupos e senhas

- chage
  - Usado para modificar parâmetros de expiração de senha dos usuários. Pode ser executado pelo superusuário seguido do nome do usuário a que iremos modificar os parâmetros. Nesse caso só teremos que ir respondendo o que nos é perguntado. Outra coisa que temos que ter em mente é que esse comando funciona apenas se estivermos usando o esquema de senhas criptografadas (/etc/shadow).
- Ver artigo <http://www.vivaolinux.com.br/artigo/Como-gerenciar-usuarios-e-grupos/?pagina=3>





# chown

- O comando `chown` executado pelo root permite alterar o proprietário ou grupo do arquivo ou diretório, alterando o dono do arquivo ou grupo.
  - Abaixo alguns exemplos:
    - # chown usuário arquivo**
    - # chown usuário diretório**
  - Outro exemplo é alterar o grupo que pertence o diretório. Por exemplo, o diretório `/mp3` pertence a `daia` e o grupo é `root`. Quero que pertença ao grupo `inf`, fica assim:

```
# chown daia:inf /mp3
```





# chgrp

- comando chgrp altera o grupo dos arquivos e diretórios.
  - As opções mais usadas são:
    - v : Reporta o grupo dos arquivos, inclusive as mudanças.
    - c : Reporta somente a mudança de grupo.
    - R : Muda o grupo de todos os arquivos e diretório recursivamente dentro da mesma hierarquia.
  - Exemplo:

```
# chgrp -R users /home
```

Neste exemplo estou alterando o grupo de todos os arquivos e diretórios dentro de /home para users.





# Agendar Tarefas

- crontab
  - crontab -e
    - Edita o crontab ou cria um caso ainda não exista;
  - crontab -l
    - Mostra o conteúdo do crontab;
  - crontab -r
    - Remove o crontab;
  - crontab -v
    - Mostra quando foi a última vez que o crontab foi editado (disponível em apenas alguns sistemas).

O crontab controla as tarefas agendadas por cada usuário. Existe um arquivo para o sistema (/etc/crontab) e um para cada usuário, localizado em /var/spool/cron/<usuário>.





# \$ crontab -e

O crontab respeita esta sintaxe.

-----

|||||

|||+----- dia da semana (0 - 6) (domingo=0)

||+----- mês (1 - 12)

||+----- dia do mês (1 - 31)

|+----- hora (0 - 23)

+----- minuto (0 - 59)





# crontab

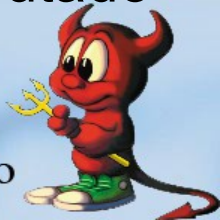
Obs 1: Em dia\_da\_Semana, 0 refere-se a domingo; e 6, ao sábado. No caso de dia da semana funciona também as três primeiras letras (em inglês) do dia da semana (SUN,MON,TUE,WED,THU,FRI,SAT)

Obs 2: Em qualquer posição pode-se usar o \* (asterisco) quando não se importar com o campo em questão.

Obs 3: Pode-se utilizar intervalos nesses campos. O caracter para intervalo é o - (hifen).

Obs 4: Pode-se utilizar lista de valores nesses campos. O caracter para a lista é a , (vírgula).

Obs 5: Qualquer texto colocado após o programa que será executado será considerado comentário e não será interpretado pelo cron





# Agendando tarefas

```
00 3,5,7 * * * root /usr/local/sbin/backup.sh
```

Diariamente, às 03:00h, 05:00h e 07:00h será executado, pelo usuário root, o script backup.sh;

```
40 16 1 6-8 * /home/oracle/sqldump.sh
```

Às 16:40 dos dias 01 de junho, julho e agosto, será executado o script sqldump.sh.

A especificação do usuário para executar o script é opcional.

```
*/5 * * * * php /var/www/cacti/poller.php > /dev/null 2>&1
```

Executa, a cada 5 minutos, o arquivo poller.php com usuário php.







# at

O comando "at" pode agendar tarefas de forma semelhante ao cron, e é integrado à interface de linha de comando do Linux. É muito eficiente se aplicado no agendamento de tarefas que sejam disparadas somente uma vez.

- `echo ls /var/log | at 10am today`
  - Executa às 10 da manhã de hoje;
- `echo updatedb | at 10:00pm today`
  - Executa às 10 da noite de hoje;
- `echo netstat | at 14:30 tomorrow`
  - Executa o comando netstat às 14:30 da tarde;
- `at -f commands.txt teatime`
  - Executa os comandos especificados no arquivos "commands.txt" no horário do chá da tarde (às 16:00 horas);
- `at -f commands.txt +3 minutes`
  - Executa comandos especificados no arquivo "commands.txt" daqui a 3 minutos.

Todas as tarefas agendadas são armazenadas em arquivos dentro do diretório `/var/spool/cron/atjobs`. Para ver as tarefas, digite: `# atq`

Para remover uma tarefa, use comando "atrm" seguido do número da tarefa obtida pelo atq.

Marcos Monteiro





▪

# Entrega da Primeira Atividade Estruturada





# Quota de disco

## # apt-get install quota

Vamos habilitar quota de disco para a partição de disco que queremos utilizar quotas. Para isso é necessário editar o arquivo de configuração das tabelas de sistemas de arquivos, o `/etc/fstab`, e incluir as opções

- "usrquota"
  - (para quota por usuários), e/ou
- "grpquota"
  - (para quota por grupos).

<code>/dev/hda2</code>	<code>/</code>	<code>reiserfs</code>	<code>notail</code>	<code>0 1</code>
<code>/dev/hda4</code>	<code>/share</code>	<code>ext3</code>	<code>defaults,usrquota,grpquota</code>	<code>0 2</code>
<code>/dev/hda3</code>	<code>/mnt/win98</code>	<code>vfat</code>	<code>defaults,utf8,umask=007,gid=46</code>	<code>0 1</code>
<code>/dev/hda1</code>	<code>none</code>	<code>swap</code>	<code>sw</code>	<code>0 0</code>

OBS: Somente sistemas de arquivos ext2, ext3 e xfs possuem suporte a quota de disco no Linux.





# Quota de disco

- # mount -o remount,rw /share

Execute o comando "mount" para verificar se as opções de quotas estão habilitadas.

– # mount | grep share

» /dev/hda4 on /share type ext3 (rw,usrquota,grpquota)

- quotachek

– É utilizado para checar quotas de disco em todos os sistemas de arquivos em /etc/fstab. E também é utilizado para criar os arquivos com informações sobre as quotas: aquota.user e aquota.group.

- quotaon

– É utilizado para habilitar a checagem de quota de disco.

- quotaoff

– Desabilita quotas de disco no sistema

Ambos os comandos quotacheck e quotaon são normalmente incluídos nos scripts de inicialização do sistema.

Marcos Monteiro





# Quota de disco

- `quotacheck` e `quotaon`:
  - a - Checa todos os sistemas de arquivos em `/etc/fstab` que estão habilitados como "automount";
  - u - Checa quotas de usuários (opção padrão);
  - g - Checa quotas de grupos;
  - v - Mostra mais detalhes na saída do comando (modo verbose).

## # `quotacheck -augv`

```
quotacheck: Quota for users is enabled on mountpoint /share so quotacheck might damage the file.
```

```
Please turn quotas off or use -f to force checking.
```

## # `quotaon -augv`

```
quotaon: using /share/aquota.group on /dev/hda4 [/share]: Device or resource busy
```

```
quotaon: using /share/aquota.user on /dev/hda4 [/share]: Device or resource busy
```





# Quota de disco

Agora vamos verificar se os arquivos `aquota.user` e `aquota.group` foram criados em `/share`:

OBS: Perceba que os arquivos `aquota.user` e `aquota.group` tem permissão de leitura e escrita somente para o usuário `root`, já que somente o usuário `root` pode gerenciar quotas no sistema.

Para editar o limite de quotas de disco, utilizamos o comando `edquota` (de EDitar QUOTA).

```
edquota [ -u | -g ] [ -F format-name ] [ -f filesystem ] -t
```

```
# edquota roberto
```

Disk quotas for user roberto (uid 1000):

```
Filesystem blocks soft hard inodes soft hard
```

```
/dev/hda4 10020 12000 13000 3 5 6
```





Disk quotas for user roberto (uid 1000):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda4	10020	12000	13000	3	5	6

- Filesystem
  - Partição que terá a quota do usuário/grupo editada. No exemplo /dev/hda4.
- Blocks
  - Número máximo de blocos (especificado em Kbytes) que o usuário possui atualmente. No exemplo 10020 Kbytes.
- soft
  - Restrição mínima de espaço em disco usado. No exemplo 12000 Kbytes.
- hard
  - Limite máximo aceitável de uso em disco para o usuário/grupo sendo editado. O sistema de quotas nunca deixará este limite ser ultrapassado. No exemplo 13000 Kbytes.
- inodes
  - Número máximo de arquivos (inodes) que o usuário possui atualmente na partição especificada. No exemplo 3 inodes.
- soft
  - Restrição mínima de número de arquivos que o usuário/grupo possui no disco. No exemplo 5 inodes.
- hard
  - Restrição máxima de número de arquivos que o usuário/grupo possui no disco. No exemplo 6 inodes.





# Quota de disco

Quando o limite soft é atingido, o usuário é alertado por ter ultrapassado sua quota com a mensagem "hda4: warning, user file quota exceeded." (quota do usuário excedida).

Quando o limite hard é atingido, o usuário é alertado com a mensagem "hda4: write failed, user file limit reached." e posteriormente com a mensagem "mkdir: não é possível criar o diretório `novo\_dir': Disk quota exceeded". E o usuário será bloqueado para criar novos arquivos ou diretórios. O limite hard nunca será ultrapassado.

Para desativar as restrições de quota coloque "0" no campo soft ou hard.

O limite soft é válido somente por um determinado tempo, após expirar este tempo, o valor do limite soft é utilizado como limite hard. Lembrando que o hard nunca poderá ser ultrapassado.

Outras opções leia o artigo <http://www.vivaolinux.com.br/artigo/Linux-Quota-de-disco/?pagina=2>

