



Curso Linux



ESTRUTURA DE DIRETÓRIOS DO LINUX



Debian



- A estrutura de diretórios do Linux é baseada na organização lógica e padronizada do Unix.
- O diretório raiz é a origem de tudo sendo representado por uma barra(/).
- Cada diretório da árvore básica tem uma função específica.
- É regulada por uma norma chamada **Filesystem Hierarchy Standard(FHS)**.
- Comando **tree**



Debian



/bin contém os comandos básicos acessados por qualquer usuário.

/boot kernel e arquivos de inicialização

/cdrom link para /media/cdrom

/dev contém os arquivos especiais p/ dispositivos

/etc contém arquivos de administração (configuração)

/home arquivos dos usuários do sistema

/initrd Fornece uma estrutura de RAM disk.

/lib contém bibliotecas de programas e módulos do kernel.



Debian



/lost+found arquivos que perderam seu vínculo após recuperação do filesystem.

/media ponto de montagem de mídias removíveis.

/mnt ponto de montagem de mídias não removíveis.

/proc diretório com informações sobre processos, memória, CPU e estado do sistema

/tmp contém arquivos temporários

/usr sub-árvore relativa aos programas dos usuários (pacotes da distribuição).

/var arquivos dos programas de tamanho variável (spool, log, temporários, etc)



Debian



/usr/bin
programas

arquivos binários dos principais

/usr/sbin

arquivos executáveis de administração

/usr/include

arquivos de cabeçalho para compilação

/usr/lib

contém bibliotecas dos programas

/usr/share
programas

outros arquivos usados pelos

/usr/src

contém o fonte do kernel

/usr/local
(não nativos)

nova sub-árvore para programas locais



Curso Linux



Login no Sistema



Login no sistema

O Ambiente Shell



- O login no sistema pode ser no Modo texto (shell) ou Modo Gráfico (desktop)
- O shell é a famosa tela preta.
- O shell é um interpretador de comandos.
- Existem vários ambientes shell no Linux(Arquivo /etc/shells). Cada um com sua peculiaridade.
- O shell padrão na maioria das distribuições é o bash(Bourne Again Shell).



Login no sistema

O Ambiente Shell



- A tela apresentada logo após a inicialização do sistema é:

```
Debian GNU/Linux 4.0 micro0 tty1
```

```
micro0 login:
```

- O primeiro login pode ser realizado como **root** e a senha que foi cadastrada na instalação do sistema.



Login no sistema O Ambiente Shell



- Após se logar é apresentada a tela:

Last login: Tue May 3 15:04:15 2005

micro0:~#

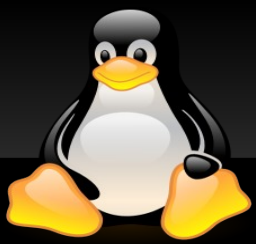
- **micro0**: É o nome da máquina que foi escolhido na instalação.
- ~: Indica o diretório home do usuário.
- #: indica que o usuário é root.
- \$: indica que o usuário é comum.



O Ambiente Shell



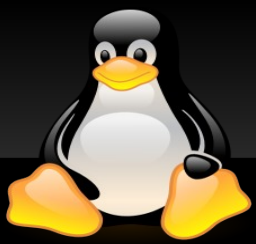
- O shell é uma interface com o sistema operacional
- Apresenta uma mensagem de solicitação (prompt)
- Possui:
 - Um conjunto de programas executáveis ou **comandos**
 - Uma linguagem que permite escrever aplicações(scripts) e interagir com outros programas



O Ambiente Shell



- É um programa como os outros, um arquivo executável
- Originalmente /bin/sh, conhecido como Bourne Shell
- Outros shells existiram em seguida:
C-shell (csh), tcsh, bash (bourne again shell)
- É especificado para cada usuário na criação da conta
- O usuário pode mudar de shell à qualquer momento

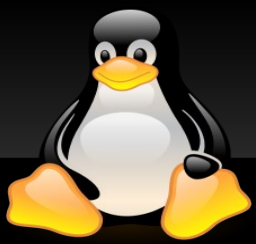


O Ambiente Shell



- Um comando é nada mais que um arquivo executável
- O espaço é separador entre comandos e argumentos
- Pode-se digitar um comando em avanço, sem esperar o prompt
- Uma lista ordenada de diretórios indica a maneira de efetuar a busca pelo comando (variável de ambiente PATH)

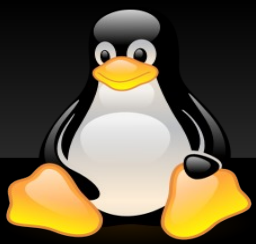
```
PATH=HOME/bin:/usr/ucb:/bin:/usr/bin:/usr/local/bin:
```



O Ambiente Shell



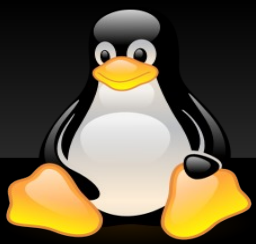
- Tudo que estiver depois do nome do comando é considerado argumento.
- Normalmente os comandos dividem os argumentos em duas categorias:
 - opções, normalmente começam com um ou dois hífen
 - nomes de arquivos, diretórios, etc para se operar



O Ambiente Shell



- & permite lançar um programa em background
 - mozilla &
 - 4866 indicação do número do processo
- " " permite delimitar uma cadeia de caracteres
 - rm "arquivo com espaço"
- \ permite suprimir a interpretação de um caractere
 - rm arquivo\ com\ espaço
- ! permite executar um comando dado recentemente apenas com os primeiros caracteres



O Ambiente Shell



` ` permite obter o resultado da execução de um comando

```
# a = `ls /home`
```

* substitui qualquer conjunto de caracteres em um nome

```
# ls *.c
```

```
# ls a*b*
```

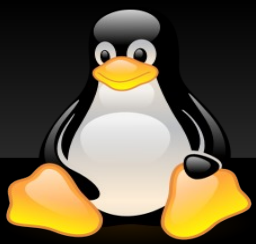
```
# rm *.o
```

? substitui um único caractere em um nome

```
# ls a?
```

[] permite definir uma série de caracteres opcionais

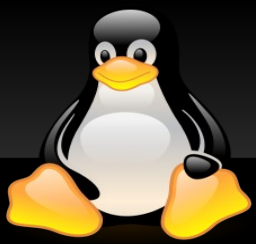
```
# ls [abcABC]*
```



O Ambiente Shell



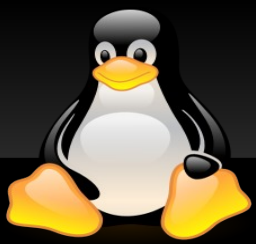
- Funções especiais de teclas no shell:
 - **Home:** Vai para o início da linha
 - **End:** Vai para o fim da linha
 - **Seta para cima/baixo:** Mostra os últimos comandos emitidos.
 - **Shift+Page Up:** Mostra telas visualizadas anteriormente(até 9 telas).
 - **Shift+Page Down:** É o inverso do anterior.
 - **TAB:** completa nome de comandos, arquivos e diretórios.



O Ambiente Shell



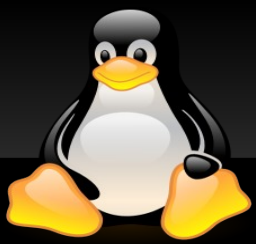
- Normalmente, comandos executados com êxito não exibem mensagens
- Mensagens são mostradas em **caso de erro**



O Ambiente Shell



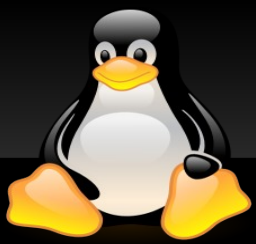
- Durante a inicialização do sistema, o Linux já carrega 6 ambientes shell(ctrl+alt+Fx).
- O prompt normalmente termina com # (para o usuário root) ou \$(para todos os outros usuários)
- Um comando é escrito no prompt do shell e executado após um **enter**.
- Exemplos:
 - **date**
 - **uname**
 - **echo**
 - **exit**
 - **whoami, hostname, uname**



O Ambiente Shell



- Para cancelar a execução de um comando:
 - **CTRL+C**
- Para passar um comando para background
 - **CTRL+Z**



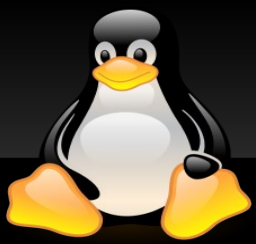
Logout do Sistema



- **Para sair do sistema (logout) utilize um dos comandos abaixo:**

```
# logout
```

```
# exit
```



Desligamento do Sistema



- Para desligar o sistema utilize um dos comandos:
shutdown -h now
shutdown -h 17:00
shutdown -h +30
halt
init 0
- Para reiniciar o sistema utilize um dos comandos:
shutdown -r now
init 6



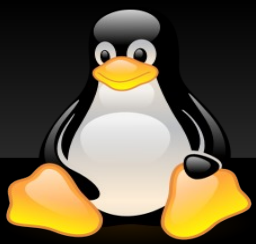
- **Sistema de Arquivos do Linux**



Arquivos e diretórios



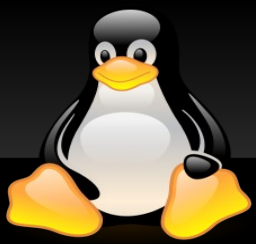
- Os diferentes tipos de arquivos:
 - Diretórios ou pastas
 - Arquivos comuns
 - Arquivos especiais
- Um diretório é um conjunto de arquivos e outros diretórios
- O nome do diretório raiz (root directory) é /
- **Diretório home** é o diretório do usuário. Cada usuário (inclusive o root) possui seu próprio diretório.



Arquivos e diretórios



- Um nome de arquivo é composto de até 1024 caracteres
- O Linux **DIFERENCIA** letras maiúsculas e minúsculas para nome de arquivos e diretórios.
- Cada arquivo possui um nome único representado por um caminho absoluto a partir da raiz:
 - /home/mesquita/teste.c
 - /home/mesquita
 - /home
- Arquivos que começam com . (ponto) são considerados 'ocultos'.



Arquivos e diretórios



- Se o nome do arquivo não começar pelo caractere /, fica a cargo do sistema construir o caminho absoluto
- O usuário indica apenas um caminho relativo
 - teste.c
 - freire/teste.c
- O caractere . (**ponto**) corresponde ao diretório corrente
- Os caracteres .. indicam o diretório pai



O sistema de arquivos

Barra comum e barra invertida



- No Linux pouco se utiliza a barra invertida(\).
- Para se designar diretórios deve-se usar **sempre** a barra comum(/)
- A barra invertida tem duas funcionalidades no Linux:
 - Indicar que um comando irá continua na próxima linha.
 - Proteger uma caractere especial(\$).



O sistema de arquivos

Extensões de arquivos



- É comum se colocar uma extensão, no fim do nome de um arquivo
- A extensão vai indicar o tipo do arquivo
 - .txt .jpg .mpeg .mp3 .tar .tar.gz .tgz
- O Kernel trata extensões como uma parte do nome
- Alguns programas usam as extensões para determinar o tipo do arquivo



O sistema de arquivos

Diretório HOME



- Cada usuário do sistema normalmente tem um diretório HOME, armazenando:
 - arquivos de configuração
 - arquivos pessoais
- O símbolo ~ é uma abreviação do diretório home
 - echo ~



Comandos de Ajuda



O Ambiente Shell

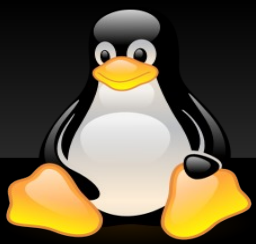
Comandos de Ajuda



- **man:**
 - É o comando mais utilizado para ajuda no Linux
 - Exemplos: `man ls`; `man mkdir`
- **info:** Similar ao `man`
- **whatis:** Mostra resumidamente para que serve um comando.
- **apropos:**
 - Busca por comandos a partir de uma expressão
 - Exemplo: `apropos reiser`



Comandos para manipulação de arquivos e diretórios



O sistema de arquivos

Comandos



- **pwd** mostra o diretório atual(print working directory)
- **cd** muda o diretório atual (change directory)

```
$ cd ..
```
- **touch:** Atualiza a data/hora do arquivo. Se o arquivo não existir, é criado um arquivo vazio.
- **which:** Mostra o caminho completo de um comando.



O sistema de arquivos

Comandos



- **ls:**
 - Mostra os arquivos e diretórios existentes no disco.
 - Opções:
 - **-a**: mostra arquivos ocultos
 - **-l**: mostra detalhes
 - **-h**: mostra o tamanho em notações humanas
 - **-S**: mostra em ordem de tamanho
 - Exemplos:
 - ls
 - ls -al /etc/init.d

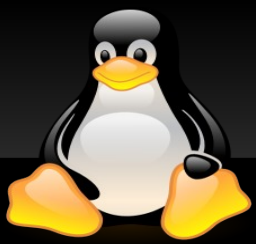


O sistema de arquivos

Comandos



- **mkdir:**
 - Cria diretórios.
 - Exemplos:
 - mkdir teste
 - mkdir /etc/teste1
- **rmdir:** Remove diretórios vazios.

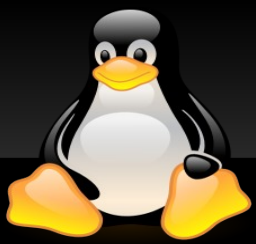


O sistema de arquivos

Comandos



- **cp:**
 - Copia arquivos e diretórios.
 - Opções:
 - -p: preserva permissões, data, hora
 - -r ou -R: implementa recursividade
 - -v: mostra o que está acontecendo.
 - Exemplos:
 - `cp /home/teste/texto .`
 - `cp /home/teste/texto /etc/texto2`



O sistema de arquivos

Comandos



- **mv:**
 - Move ou renomeia arquivos e diretórios.
 - Opções:
 - -f: não pergunta antes de sobrescrever um arquivo
 - -v: mostra o que está acontecendo.
 - Exemplos:
 - `mv /usr/teste .`
 - `mv /etc/teste1 /etc/teste2`



O sistema de arquivos

Comandos



- **rm:**
 - Remove diretórios(vazios ou não) e arquivos.
 - Opções:
 - -i: pede confirmação
 - -r: remove diretórios e arquivos recursivamente
 - -f: remove sem pedir confirmação
 - Exemplos:
 - rm teste
 - rm -rf mala
 - rm -rf / : **NUNCA EXECUTE ESSE COMANDO!!**



O sistema de arquivos

Comandos – Mais Exemplos



- **cp** **copia um ou mais arquivos**

```
$ cp CV.pdf old_CV.pdf  
$ cp CV.pdf /tmp  
$ cp *.gif *.jpg imagens/
```

- **rm** **remove um ou mais arquivos**

```
$ rm lista_compras  
$ rm lista compras
```

- **mv** **move ou muda o nome de arquivo**

```
$ mv CV.pdf old_CV.pdf  
$ mv CV.pdf /tmp  
$ mv dir1 dir2
```



O sistema de arquivos

Comandos

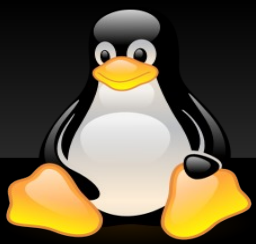


- **cat:**

- Concatena arquivos. Se usado apenas com um arquivo mostra seu conteúdo.
- Opção: -n – numera as linhas

- **head:**

- Mostra as primeiras linhas de um arquivo.
- Opção
 - -n x: Mostra as n primeiras linhas do arquivo x.



O sistema de arquivos

Comandos



- **tail:**
 - Mostra as últimas linhas de um arquivo.
 - Opções
 - -n x:Mostra as n últimas linhas do arquivo x.
 - -f: Lê o arquivo ininterruptamente.
- **md5sum:**
 - Calcula o hash md5 de arquivos.
 - Importante para verificar integridade de arquivos.
 - Exemplos:
 - md5sum /etc/fstab
 - md5sum /etc/*



O sistema de arquivos

Comandos



- **diff:**

- Mostra a diferença entre dois arquivos textos
- Exmplo: `diff texto1 texto2`.

- **du:**

- Mostra o espaço ocupado em disco por um diretório, recursivamente.
- Opções:
 - -s: Apresenta apenas um sumário
 - -h: Utiliza notações humanas.

- **patch:**

- Aplica correções entre dois arquivos textos
- Exmplo: `patch -p0 texto1 texto2`.



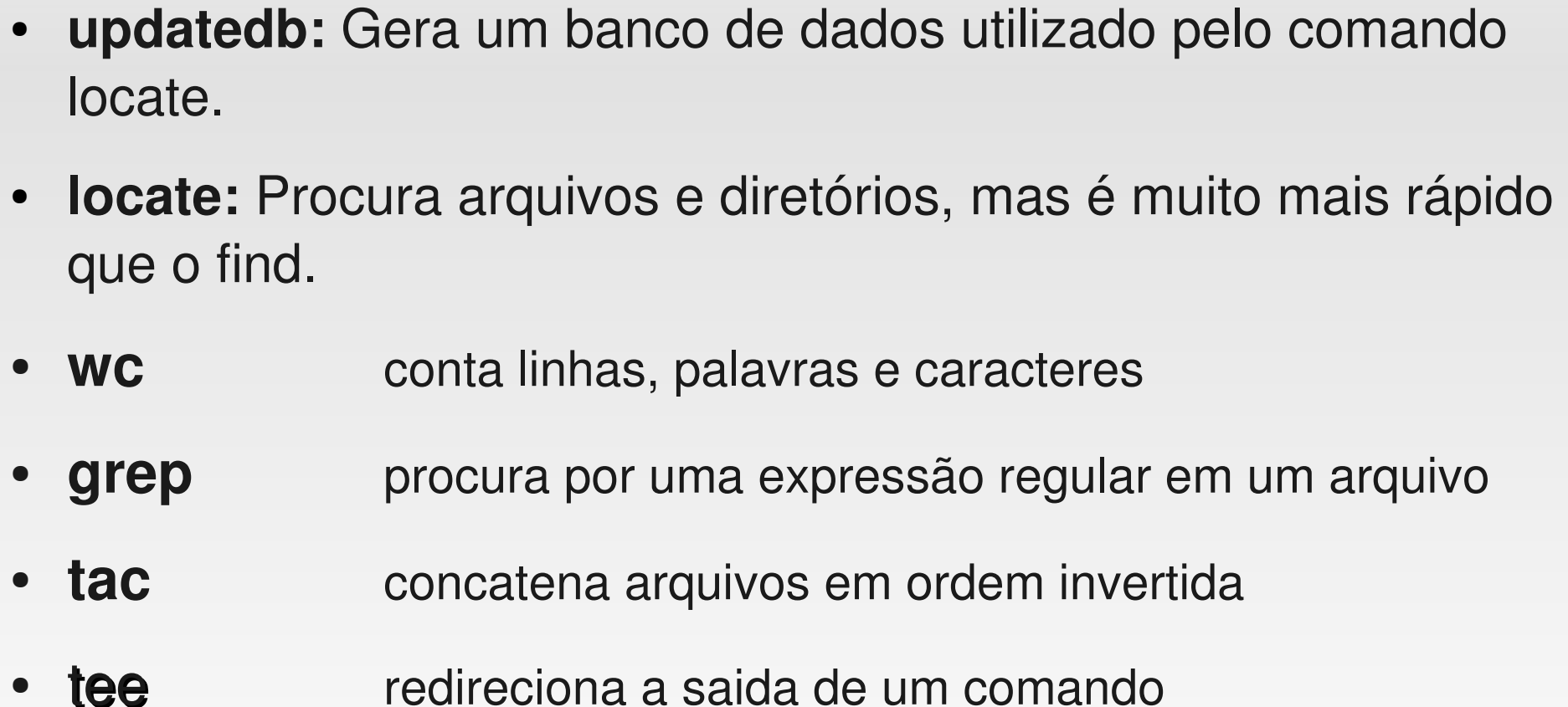
O sistema de arquivos

Comandos



- **find:**

- Procura por arquivos e diretórios no disco.
- Opção: -name: Procura pelo nome
- Exemplos:
 - `find / -name 'win*'`
 - `find /etc -iname '*loc*'`
 - `find / -lname apache`
 - `find / -perm 4000 -print`
 - `find . -iname 'teste*' -exec mv {} ../busca \;`





O sistema de arquivos

Outros comandos para arquivos



- **xargs** recebe argumentos de outros comandos e executa comandos nesses argumentos
 - `echo 1 2 3 4 5 6 | xargs -n 2`
 - `find . -iname 'teste*' -print0 | xargs -0 -I file /bin/mv file file.bak`
 - `find / -iname '*.gif' -print0 | xargs -0 -I {} mv {} ../busca`
 - `find ../busca -iname 'ylwm*.gif' -print0 | xargs -0 -I file /usr/bin/gimp file`
 - `find / -iname '*.avi' -print0 | xargs -0 -I arquivo /usr/bin/totem arquivo`



Editores de texto para shell



Editor “vi”



- Modo de comando
 - i** modo inserção
 - x** apaga um caractere
 - yy** copia a linha corrente
 - dd** apaga a linha corrente
 - /** busca do cursor para a frente
 - n** próxima ocorrência
- Modo de edição
 - ESC** volta ao modo de comando



Editor “vi”



- Modo ex:
 - :x** sai e salva o arquivo
 - :wq** sai e salva o arquivo
 - :w** salva
 - :q** sai
 - :q!** sai sem salvar
 - :1** vai para a primeira linha
 - :\$** vai para última linha



LINUX





Sistemas de Arquivo

mount e umount



- Sub-árvores podem ser *ligadas* à árvore principal
 - O ponto de ligação é um **diretório existente**, vazio ou não
- A sub-árvore pode ser *montada* e *desmontada*
- A localização física dos arquivos pode ser desconhecida pelo usuário
 - Um diretório montado pode estar em um outro computador da rede
- **O que montar ?**



Sistemas de Arquivo

mount e umount



- Os comandos ***mount*** e ***umount*** montam/desmontam uma partição na árvore do sistema
 - ***mount -t vfat /dev/hda1 /mnt/windows***
- Normalmente só o **root** tem permissão para montagem
- Nunca retire um dispositivo antes de desmontá-lo.
- O dispositivo não irá desmontar se estivermos dentro do diretório em que foi montado.



Sistemas de Arquivo

mount e umount



- Na inicialização do sistema, montagem padrão definida em /etc/fstab
- Pode ser especificado apenas o nome do diretório:
 - ***mount /cdrom***
- Montando uma partição FAT32:
 - ***mount -t vfat /dev/hda1 /mnt/windows***
- Montando uma partição NTFS(read-only):
 - ***mount -t ntfs /dev/hda1 /mnt/windows***
- Montando uma pendrive:
 - ***mount /dev/sda1 /mnt/pendrive***



Sistemas de Arquivo

mount e umount



- Arquivo ***/etc/fstab*** (Tabela de file system)
 - ✓ Dispositivo: nome da partição
 - ✓ Ponto de montagem: nome do diretório
 - ✓ Tipo
 - ext3, swap, vfat, iso9660, auto, etc
 - ✓ Opções
 - ro – read only,
 - rw – monta a partição como read-write
 - noauto - não será montado automaticamente durante o boot
 - nouser – usuário comum não pode montar o file system



Sistemas de Arquivo

mount e umount



- Arquivo /etc/fstab(Tabela de file system)
 - ✓ Opções
 - user - usuário comum pode montar o file system
 - sw – indica que o filesystem é de swap
 - exec – permite a execução de binários
 - auto – permite montagem em série com o comando mount -a
 - - Dump: Essa coluna indica se o comando dump irá fazer um backup do file system. O valor 0 indica que não haverá dump e 1 haverá
 - - Pass-no: Ordem de checagem de integridade. Valores: 0, 1 ou 2



Sistemas de Arquivo

mount e umount

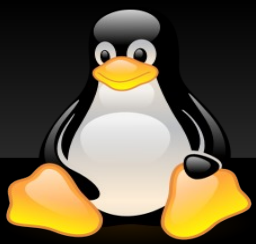


- Para visualizar as partições montadas:
 - ➔ ***mount***
 - ➔ Arquivo /etc/mtab
- Unidades de CD-ROM e disquete, podem estar configuradas com montagem automática ou manual
- Imagens de CD-ROM também podem ser montadas
 - ➔ ***mount -t iso9660 -o loop imagem.iso /mnt/disk***



LINUX





Sistemas de Arquivo

Os *links* simbólicos



- Atalhos para outros arquivos ou diretórios
- É feita uma substituição : um nome é substituído pelo outro
- Os *links* simbólicos podem ser feitos sobre diretórios ou arquivos pertencentes à diferentes partições
- Existe a criação de um arquivo especial tendo como dado o nome efetivo do arquivo (ou outro link simbólico)
 - ***ln -s /var/spool/mail /var/mail***



Sistemas de Arquivo

Os *hard links*



- Os *hard links* não se relacionam com o nome do arquivo, mas com sua identificação no sistema operacional (*inode*)
 - Um *inode* é uma estrutura que descreve um arquivo no disco (tipo, tamanho, permissões, etc)
 - Uma nova entrada pode ser criada em um diretório apontando para um *inode* existente : o arquivo terá então 2 links



Sistemas de Arquivo

Os *hard links*

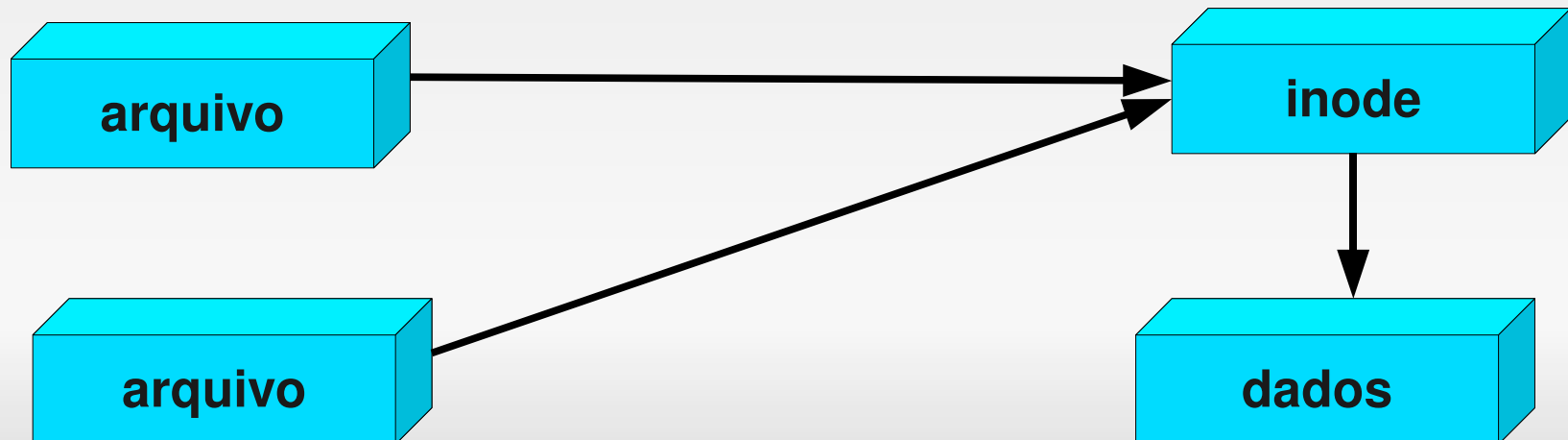


- A criação é feita com o comando **ln** :
→ ***ln /bin/ls /bin/dir***
- Os *hard links* não são permitidos em diretórios ou sobre arquivos pertencentes à diferentes partições
- A supressão é feita pelo comando **rm** (*unlink*)
- O arquivo é apagado, se ele não tiver pelo menos um *link*



Sistemas de Arquivo

links simbólicos e hard links





Sistemas de Arquivo

links simbólicos e hard links



- Exemplo
 - Criar um arquivo texto
 - Criar um *hard link* e um *link* simbólico para o arquivo
 - Verificar com **ls** os *links* criados
 - Apagar o arquivo original



LINUX





Sistemas de Arquivo

Problemas que podem ocorrer



- Falta de espaço em disco
 - *df -h*
 - *du -hs **
- Sistema de arquivos corrompido
 - *fsck*
- Falta de espaço para inodes
 - *df -i*



Sistemas de Arquivo

Proteção de arquivos



- Para o sistema, cada usuário é caracterizado por:
 - seu nome (uid)
 - um grupo de usuários ao qual pertence (gid)
- O super usuário (root) tem todos os direitos
- Para um arquivo, existem 3 diferentes classes de usuários:
 - O proprietário (criador)
 - Usuários pertencentes ao mesmo grupo
 - Todos os outros usuários



Sistemas de Arquivo

Proteção de arquivos



- Para o sistema, cada usuário é caracterizado por:
 - seu nome (uid)
 - um grupo de usuários ao qual pertence (gid)
- O super usuário (root) tem todos os direitos
- Para um arquivo, existem 3 diferentes classes de usuários:
 - O proprietário (criador)
 - Usuários pertencentes ao mesmo grupo
 - Todos os outros usuários



Sistemas de Arquivo

Proteção de arquivos



- Existem 3 permissões de acesso para um arquivo:
 - r = leitura, w = escrita, x = execução
- Para diretórios:
 - r listar os arquivos, sem acesso
 - w criar novos ou remover arquivos existentes
 - x acessar o diretório, ou usá-lo no caminho
 - t usada por diretórios comuns (/tmp), somente a remoção de arquivos que pertencem ao usuário é permitida



Padrões de arquivos: exemplos

```
ls -l exemplo
```

```
-rwxr-x--x 1 daniel staff 872 Oct 24 19:55  
exemplo
```

- Tipo do arquivo (d para diretório, b ou c para arquivo especial, l para link simbólico, p para pipe, s para socket)

rwX permissões para o proprietário: leitura, escrita, execução

r-X permissões para o grupo: leitura, execução

--X permissões para outros: execução

1 número de *hard links*

daniel staff usuário e grupo dono do arquivo



Padrões de arquivos: comandos

- ***chmod*** mudar a proteção atual
 - # *chmod a+x exemplo*
 - # *chmod ugo-w exemplo*
 - # *chmod 640 exemplo*
- ***chown*** mudar o proprietário (grupo) do arquivo
- ***chgrp*** mudar o grupo do arquivo
- ***cp -p*** copiar um arquivo com as permissões



Sistemas de Arquivo

Mudança de identidade



- Na execução de um programa, é possível ter os mesmos direitos que o proprietário do programa ou os usuários do mesmo grupo
- Existe um bit suplementar chamado bit **s**

```
chmod +s exemplo
```

```
ls -l exemplo
```

```
rwsr-x--x exemplo
```

```
chmod g+s teste
```

```
ls -l teste
```

```
rwxr-s--x teste
```



Sistemas de Arquivo

Proteção de arquivos



- Exemplo
 - Verifique as permissões do seu diretório **HOME**
 - Mude as permissões de forma a permitir/negar que outros usuários vejam arquivos do seu **HOME**
 - Verificar as permissões de **/bin** e **/bin/l**
 - Verificar as permissões de **/etc/passwd** e **/etc/shadow**



LINUX





Interpretador de comandos

Cadastramento de Usuários



- Apenas executado pelo root
- Comando ***adduser***
 - Pede a senha e confirmação da senha;
 - Cria o diretório ***home*** do usuário.
- Comando ***passwd***
 - Alteração da senha do usuário;
 - O root pode alterar a senha de qualquer usuário. Um usuário muda apenas sua própria senha.



Interpretador de comandos

Gerência de Cadastro



- ***useradd***: Adiciona um usuário ao sistema, mas não adiciona senha, nem cria o diretório ***home***.
- ***userdel***: Exclui usuários do sistema.
- ***passwd***
 - Cadastra ou altera a senha de um usuário. Também pode ser usado para bloquear um usuário.
 - Opções
 - **-l** bloqueia um usuário
 - **-u** desbloqueia um usuário



Interpretador de comandos

Gerência de Usuário



- ***su***
 - Troca o usuário corrente;
 - Com a opção “-” carrega o ambiente do usuário.
- ***who***: Mostra os usuários que estão conectados no momento.
- ***w***: Semelhante a ***who***, com mais detalhes
- ***whoami***: Exibe o nome do usuário que está conectado.
- ***id***: Exibe o uid, gid e grupos do usuário

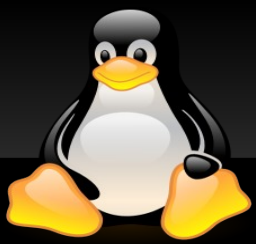


Interpretador de comandos

Gerência de Grupos e Usuários



- ***addgroup***: Adiciona um grupo ao sistema.
- ***groupdel***: Remove um grupo do sistema.
- ***lastlog***: Mostra o último login dos usuários.
- ***last***: Mostra a entrada e saída de usuário no sistema.



Interpretador de comandos

Aliases



- ***alias***

- Gera uma atalho para um comando;
- Se digitado sozinho, mostra os aliases já definidos;
- Exemplo: ***alias teste='ls -al'***.

- ***unalias***

- Desfaz um alias
- Exemplo: ***unalias teste***

- O alias criado funciona apenas na sessão atual do usuário. Para um alias ficar definitivo, basta adicioná-lo no arquivo ***.bashrc***



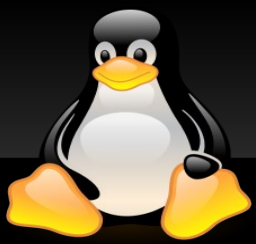
Interpretador de comandos

Execução em segundo plano



- Executar um programa em segundo plano é fazer com que ele rode sem ocupar o ambiente shell.
- Pode ser feito colocando **&** no final do comando.
- Exemplo:

→ ***# updatedb &***



Interpretador de comandos

Gerência de memória



- ***free***

- Mostra os espaços livre e ocupados da memória principal e do swap;

- Opções:

- ***-m*** resultados em MB
 - ***-k*** resultados em KB

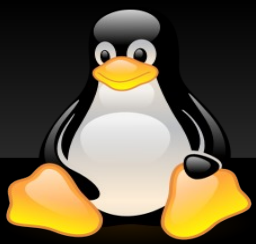


Interpretador de comandos

Processos



- Cada programa rodando no sistema é um processo;
- O kernel identifica cada processo por um número chamado ***process id (pid)***;
- O kernel mantém o controle das propriedades de cada processo
 - ***uid , gid, ppid, etc***
 - ***# ps -elf***
 - ***# top***
 - ***# kill -9 pid***



Interpretador de comandos

Gerência de Processos

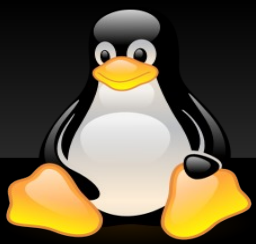


- ***ps:***

- Mostra os processos que estão sendo em executados;
- Opções:
 - ***-a:*** mostra os processos que rodam em todos os terminais
 - ***-x:*** mostra os processos que rodam independentes de terminais
 - ***-u:*** mostra outros dados, inclusive os usuários donos de processos

- Exemplo:

- ***ps -aux***



Interpretador de comandos

Gerência de Processos



- **kill:**
 - Encerra um processo que está sendo executado;
 - Opção: -9: mata o processo, não permitindo bloqueios por partes do sistema;
 - Exemplo:
 - **kill 3125**
 - **kill -9 3125**
- **killall:** Similar ao kill, porém usa-se o nome ao invés do PID.



Interpretador de comandos

Gerência de Processos



- ***time:***

- Mede o tempo de execução de um programa;

- Exemplo:

- ***time updatedb***

- ***nohup:***

- Faz com que um comando continue a ser executado, mesmo que o usuário faça logout;

- Exemplo:

- ***nohup updatedb &***



Interpretador de comandos

Gerência de memória e de processos



- ***top:***
 - Mostra interativamente, a utilização de CPU e memória pelos processos;
 - Opções:
 - ***shift p***: ordena por uso de processador
 - ***shift m***: ordena por uso de memória
 - ***k***: solicita um PID para fazer kill
 - ***q***: sai do ambiente top



Interpretador de comandos

Gerência do Sistema



- ***clear***: Limpa a tela.
- ***set***: Mostra as variáveis de ambiente.
- ***df***: Mostra os espaços livres e ocupados em disco.
- ***history***: Mostra os últimos comandos emitidos.



Interpretador de comandos

Gerência do Sistema



- ***date:***
 - Mostra ou altera a data e a hora do sistema;
 - Opção: **-s** : altera a hora;
 - Exemplo:
 - ***Date mmdaaa***
 - ***Date -s hh:mm:ss***
- ***cal:*** calendário
- ***logout ou exit:*** sai do sistema



Interpretador de comandos

Gerência do Sistema



- ***reset***: Restabelece o terminal, útil quando o mesmo ficar desconfigurado.
- ***export***: Utilizado para alterar ou criar variáveis de ambiente.
- ***watch***: Mantém um determinado comando em constante execução. Opção: ***--interval=x***
- ***sleep***: Põe a máquina em espera por um tempo **x**.



Interpretador de comandos

Gerência do Sistema



- ***lspci***: Mostra dispositivos pci.
- ***lsusb***: Mostra dispositivos usb.
- ***cat /proc/cpuinfo***: Contém dados sobre o processador.
- ***cat /proc/meminfo***: Contém dados sobre a memória.
- ***lshw***: Faz um levantamento completo do hardware.
- ***hwinfo***: Faz um levantamento completo do hardware.



Interpretador de comandos Pipes



- Quantos usuários estão conectados ?
 - **# who > users**
 - **# wc -l users**
- Utiliza-se o caractere “ | ”(barra vertical) para o pipe
 - **# who | wc -l**
- Supressão dos arquivos intermediários
- Todos os comandos envolvidos ficam em execução
 - **# tail -100 /var/log/messages | grep error**



Interpretador de comandos

Filtros Básicos



- ***more:*** Usado para exibir informações em várias páginas.
- ***less:*** Funciona como o more, mas permite uso de setas.
- ***grep:***
 - Busca linhas de texto com base em expressões
 - Opções:
 - ♦ **-i:** ignora a caixa
 - ♦ **-r:** opera recursivamente
- **>:** Redireciona uma saída para um arquivo.



Interpretador de comandos

Redirecionamento



- O redirecionamento da entrada padrão (stdin) e das saídas (stdout e stderr) pode ser feito:
 - *# prog < arquivo_entrada*
 - *# prog > arquivo_saida*
 - *# prog 2> saida_de_erros*
- Se ao invés de > for usado >>, os dados são adicionados à saída.



Interpretador de comandos

Variáveis



- Variáveis do shell (bash)
 - ***\$ arquivos = "notas.txt relat.txt"***
 - ***\$ echo \$arquivos***
 - ***\$ set***
- Variáveis de ambiente (bash)
 - ***\$ export arquivos = "notas.txt relat.txt"***
 - ***\$ echo \$arquivos***
 - ***\$ env***
 - ***\$ echo \$PATH***



Interpretador de comandos

Scripts do Shell



- É um arquivo texto
 - Normalmente com especificação do shell na primeira linha (`#!/bin/sh`).
- Permite concentrar a execução de vários comandos:

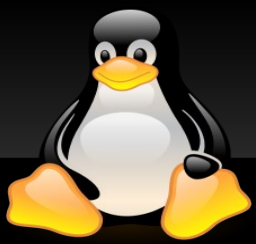
```
for file in $var;  
do  
mv -v $file $file.old;  
done
```

- Sua execução pode ser feita pela chamada direta ao arquivo, se ele possuir o bit **x**



LINUX





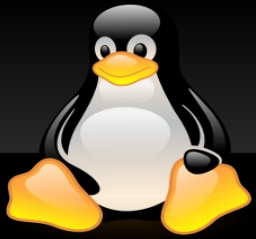
Iniciando e Desligando

- Runlevel: 0 a 6
 - *(0=halt, 1=single user, 6=reboot)*
- Arquivo /etc/inittab
 - runlevel padrão
 - ♦ ***id:3:initdefault:***
 - trap ctrl+alt_del
- Arquivo /etc/fstab
 - Pontos de montagem padrão



Iniciando e Desligando

- Diretório /etc/init.d
 - Contém scripts de inicialização para cada serviço que pode ser iniciado/parado
- Diretório /etc/rcN.d
 - links para os scripts que devem iniciar/parar no runlevel N
- init N
 - Muda para o runlevel N



Linux daemons



- Devem estar habilitados apenas os serviços usados
- Exemplo
 - ***crond***
 - ***httpd***
 - ***iptables***
 - ***network***
 - ***smb***
 - ***sshd***
 - ***xfs***



Gerenciador de Inicialização

- Passa parâmetros para o kernel
- Permite escolher entre vários sistemas operacionais
- Gerenciadores mais populares: lilo e grub
- LILO
 - Arquivo `/etc/lilo.conf`
 - É necessário executar o comando ***lilo*** para carregar as mudanças
- GRUB
 - Arquivo `/boot/grub/menu.lst`



Gerenciador de Inicialização

- Exemplo:
 - Remova o gerenciador de inicialização, reinicie a máquina;
 - Instale de novo o gerenciador de inicialização;
 - GRUB
 - ***grub-install /dev/hda***
 - LILO
 - ***lilo***



LINUX





Linux

Pacotes DEB



- Instalar
 - ***dpkg -i arquivo.deb***
- Remover
 - ***dpkg -r <nome_do_pacote>***
 - ***dpkg -r <nome_do_pacote>***
- Listar todos instalados
 - ***dpkg -l***
- Mostra a descrição de um pacote
 - ***dpkg -l <nome_do_pacote>***



Linux

Pacotes RPM



- Instalar
 - ***rpm -ivh arquivo.rpm***
- Remover
 - ***rpm -e <nome_do_pacote>***
- Listar todos instalados
 - ***rpm -qav***
- Mostra a descrição de um pacote
 - ***rpm -qav <nome_do_pacote>***



LINUX





- O administrador deve se prevenir contra
 - Falha de discos
 - Erros de configuração
 - Apagamento acidental
 - Vírus ou ataque remoto
 - Roubo, incêndio, inundação, etc



Backup e Recuperação

- Mídias de backup
 - Unidades de fita (melhor opção)
 - Outros discos rígidos
 - CD gravados
- O Backup deve ser regular e bem conhecido pelos administradores



Linux

Usando o TAR



- Originalmente desenvolvido para trabalhar com unidades de fita
 - ***tar -cvf /dev/st0 /home***
- Junta vários arquivos em um único
 - ***tar -cvf arquivo.tar ****
- Outros comandos para backup:
 - ***cpio, dump, restore***



Linux

Usando o TAR



- Principais ações:
 - **c** create
 - **t** list
 - **x** extract
 - **f** especifica o nome de um arquivo ou dispositivo
 - **z** realiza compactação gzip
 - **p** conserva as permissões dos arquivos extraídos



Linux

Usando o TAR



- Escolhendo os diretórios:
 - */tmp, /dev, /proc, não são necessários*
 - */bin, /usr, /lib, podem ser recuperados com uma nova instalação*
 - */home, /etc, /var, normalmente são importantes*
- É comum fazer scripts de backup e agendar sua execução para a noite.



Agendando Tarefas

- Para executar tarefas regularmente: ***cron***
- Usando o ***cron***
 - Daemon ***crond*** em execução
 - /bin, /usr, /lib, podem ser recuperados com uma nova instalação
 - Diretórios /etc/cron.hourly , /etc/cron.daily , /etc/cron.weekly, /etc/cron.monthly
- Arquivo ***/etc/crontab***
 - Linhas no formato:



Linux

Agendando Tarefas

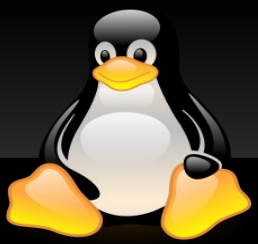


- Exemplo
 - Verifique se o **crond** está em execução;
 - Crie um script para copiar um arquivo qualquer para um diretório de backup;
 - Agende no **cron** a execução do script para alguns minutos no futuro;
 - Veja se o script foi executado com sucesso.



LINUX





Comandos úteis para rede

- **ifconfig**
- **route**
- **ping**
- **nmap**
- **dig**
- **telnet/ftp**
- **ssh/scp**
- **netstat**



Comandos úteis para rede

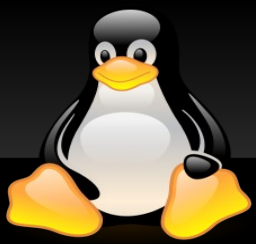
- **ifconfig**

- Serve para visualizar e editar as configurações de rede;
- Se for executado sem argumentos mostras as interfaces de rede;
- Para configurar a interface eth0, basta dar o seguinte comando:
 - ♦ ***ifconfig eth0 10.45.2.10 netmask 255.255.255.0 up***
- O IP configurado pelo ifconfig só vale temporariamente, ou seja, quando a máquina reiniciar, volta para as configurações originais.



Comandos úteis para rede

- As configurações das placas de redes ficam nos arquivos:
 - No Mandriva/Conectiva/Fedora:
 - ♦ ***/etc/sysconfig/network-scripts/ifcfg-ethX, onde X pode ser 0, 1, 2, etc.***
 - No Debian/Kurumin:
 - ♦ ***/etc/network/interfaces***
- O script de inicialização de rede é:
 - No Mandriva/Conectiva/Fedora: ***/etc/init.d/network***
 - No Debian/Kurumin: ***/etc/init.d/networking***



Comandos úteis para rede

- O arquivo ***/etc/sysconfig/network-scripts/ifcfg-eth0***
 - ***DEVICE eth0***
 - ***ONBOOT = yes,no***
 - ***BOOTPROTO= static, dhcp ou none***
 - ***IPADDR = 10.45.2.50***
 - ***NETMASK = 255.255.255.0***
 - ***GATEWAY = 10.45.2.1***
- Se BOOTPROTO for dhcp, não precisa escrever os outros parâmetros, já que os mesmos serão com figurados pelo servidor DHCP.



Comandos úteis para rede

- ***route***
 - Serve para visualizar e editar a tabela de rotas da máquinas, que é a tabela usada para a máquina saber para onde enviar um pacote TCP/IP;
 - Para visualizar a tabela de rotas basta dar o comando sem parâmetros;
 - Se o comando demorar a fornecer a resposta, tente usar **route -n**, esse parâmetro indica que o comando não deve tentar resolver(transformar) o endereço da rede em um nome.



Comandos úteis para rede

- ***route***

- ➔ As duas situações mais comuns de alterações na tabela de rotas são adicionar uma rota para uma rede específica e adicionar uma rota padrão.
- ➔ Para adicionar um rota para uma rede específica, basta dar o comando da seguinte forma:
 - ♦ ***route add -net 10.45.20.0 netmask 255.255.255.0 gw 10.45.2.1 dev eth0***
- ➔ **net** indica a rede, **netmask** a máscara de subrede, **gw** o gateway para aquela rede, e **dev** a interface de rede por onde o pacote deve ser enviado.



Comandos úteis para rede

- ***route***

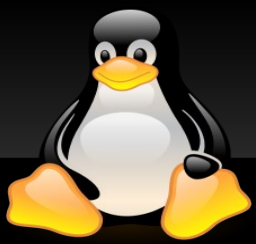
- Para adicionar uma rota padrão, basta dar o comando da seguinte forma:

- ***route add default gw 10.45.2.1 dev eth0***

- Vale lembrar que a rota padrão é aquela usada, se o pacote não pertencer a nenhuma das redes existentes na tabela de rotas;

- As rotas adicionadas com o comando ***route*** não valerão após reiniciar o serviço de rede;

- Para tornar as rotas sempre válidas, pode-se colocar o comando no arquivo ***/etc/rc.local***.



Comandos úteis para rede

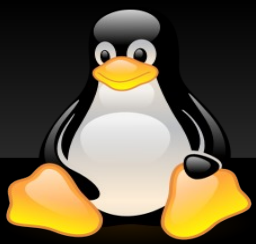
- *ping*

- Serve para verificar se uma máquina está ativa;
- Em caso de não haver resposta, a máquina pode não estar ativa(situação mais comum), ou pode estar protegida por firewall;
- Principais parâmetros:
 - ♦ **-b**: envia um ping de broadcast para um endereço de rede.
 - ♦ **-c 200**: envia 200 pacotes ping
 - ♦ **-q**: envia os pacotes mas não fica exibindo as respostas na tela.



Comandos úteis para rede

- ***nmap***
 - Serve para verificar as portas que esperam conexão em uma máquina, com isso é possível saber alguns dos serviços que rodam numa máquina;
 - Por padrão ele não é instalado na maioria das distribuições;
 - Exemplo:
 - ♦ ***nmap 10.45.4.25***



Comandos úteis para rede

- ***dig***

- Serve para fazer uma consulta a um servidor de DNS;
- Exemplo:
 - ♦ `dig @10.45.6.4 www.exercito.gov.br` - Faz uma consulta ao DNS 10.45.6.4 pedindo o endereço IP do site `www.exercito.gov.br`
 - ♦ `dig @10.45.6.4 NS 5cta.eb.mil.br` Faz uma consulta ao DNS 10.45.6.4 pedindo o endereço IP do servidor de nomes do domínio `5cta.eb.mil.br`
- Se não for especificado o IP do servidor DNS, a consulta é feita ao DNS primário de máquina.



Comandos úteis para rede

- ***dig***
- ***telnet / ftp***
- ***ssh / scp***
- ***netstat***



O que fazer se a máquina não acessa a internet

- Verifique as configurações de rede com o comando ***ifconfig***. Se necessário, altere-as.
- Verifique com o comando ***route*** se há uma rota padrão. Se não houver, adicione-a.
- Verifique se a máquina pinga para as outras máquinas da rede.
- Verifique se a máquina pinga para o default gateway.
- Verifique se o servidor DNS está correto no arquivo ***/etc/resolv.conf***. Se necessário acrescente o servidor DNS, de preferência na primeira linha do arquivo.
- Com o comando ***dig*** verifique se a máquina consegue resolver os endereços de internet.



LINUX

