

# Introdução ao Linux e comandos de E/S

Prof. MSc. Rodrigo Malara  
rmalara.uniara@gmail.com

# Sobre o treinamento

- Tutorial de enfoque prático
  - Uso de ferramentas básicas
  - Configurações básicas para administração do sistema
  - Instalação de software
  - Focado na distribuição Fedora Linux

# O que é o Linux?

- Sistema Operacional padrão UNIX
  - Multiusuário
  - Multitarefa
- É software livre
- Estritamente falando: "Linux é somente o kernel"
- Distribuições
  - Kernel
  - Utilitários
  - Ferramentas
  - Instalador e configurador
  - Ex: Red Hat, Conectiva, Suse, Mandrake, Debian,...

# Instalação (prática em lab.)

- Bootar Live CD
- Usar “Install to Hard Drive” (System Tools)
- Fazer o particionamento
- Instalar e reiniciar o Sistema
- Aceitar licença, criar usuário, etc
- Configurar resolução de tela
- Configurar layout de teclado e mouse

# Informações Iniciais

- É necessário autenticar-se para acessar uma máquina Linux.
- Os comandos são invocados por um shell
- O uso de shells é intenso para administrar o sistema.
- Ex:  

```
# date  
Sat Sep 13 17:27:58 BRT 2017  
#
```
- Maiúsculas e minúsculas são diferenciadas pelo S.O.

# Informações Iniciais (2)

- A maioria das instalações inclui o modo gráfico (X-Windows).
- Existem vários Gerenciadores do X-Windows
  - GNOME (padrão Fedora, UBUNTU)
  - KDE
  - BlackBox
- O acesso ao shell não é direto. Ele deve ser invocado

# Informações Iniciais (3)

## Estrutura de diretórios da partição de dados

/	Partição padrão
/etc	Configurações
/tmp	Partição de dados temporários
/var	Spool de impressão, mails, logs
/home	Contas dos usuários cadastrados
/boot	Imagem do sistema operacional
/opt	Aplicativos
/proc	Informações sobre o SO (execução)
/dev	Ponto de acesso aos dispositivos (hardware)
/usr	Aplicativos e documentações
/mnt	Diretório onde são “montados” outros drives
/media	Diretório onde aparece pen-drives, CDs, etc

# Shell

- Quase tudo no Linux é feito a partir de um shell
- Shell = Casca → Interface com o sistema operacional (kernel)
- Pode-se abrir vários shells na mesma janela (tabs)
- Existem vários interpretadores
  - bash (padrão)
  - sh
  - ksh e outros
- O shell preferido do usuário é definido na criação da sua conta na máquina



# Interagindo com o Linux

- Pode-se utilizar a tecla TAB para se completar comandos e nomes de arquivos ou diretórios.
- A princípio, nenhum comando é embutido no shell.
- Comandos executados no shell são programas binários.
- Programas binários **não** possuem extensão (.exe)
- Para se executar um arquivo binário:
  - ./meubinario (executa binário no diretório atual)
  - meubinario (o binário deve estar num diretório que esteja listado na variável de ambiente PATH)
- Variáveis de ambiente ??? O que é isso?

# Comandos Básicos

- Manipulação do sistema de arquivos
- Edição de arquivos texto
- Acessar manuais/info sobre comandos
- Manipulação de arquivos texto
- Compactadores de arquivos e diretórios
- Localizando arquivos armazenados
- Comandos relacionados ao acesso à rede

# Comandos: Sistema de Arquivos

Barra de separação de diretórios: / (divisão)

Dica: Não use espaços, símbolos ou letras acentuadas em nomes de arquivos

cd	muda de diretório
ls	lista arquivos do diretório atual
cp	faz cópias de arquivos e diretórios
rm	apaga arquivos e diretórios
mv	renomeia/move arquivos e diretórios
mkdir	cria diretórios

# Comandos: Sistema de Arquivos (2)

- Criar diretório temporário no seu “HOME”  
# cd  
# mkdir tmp
- Copiar todos os arquivos do diretório /etc para esse novo diretório:  
# cd tmp  
# cp -R /etc/\* .

# Sistema de Arquivos

- O sistema de arquivos no UNIX é totalmente unificado.
- Os drives estão dispostos em diretórios.  
Ex:
  - Disquete 3 ½"                      /mnt/floppy
  - CD-ROM                                /mnt/cdrom
- Os drives precisam ser montados para serem usados

... montados ???

# Sistema de Arquivos (2)

- Montando drives: comando mount

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

- isso é complicado ...

- Solução: Arquivo fstab no diretório /etc/

- Configurações dos *filesystems* mais montados
- Faça um backup do fstab (fstab.orig)
- Edite o arquivo e adicione a seguinte linha:

```
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
```

- Agora, montar drives ficou mais fácil:

```
# mount /mnt/cdrom
```

- Para mais informações sobre o arquivo fstab: `man fstab`

# Sistema de Arquivos (3)

- É necessário se “desmontar” drives montados  
# umount /mnt/cdrom
- Senão o CD não ejeta do drive
- Outros comandos:  
du : utilização de espaço em disco  
df : utilização de espaço nos *filesystems*

# Sistema de Arquivos

- Verifique quanto espaço o diretório `/etc` ocupa  
# `du /etc`
- Verifique espaço disponível no *filesystem* onde o diretório `/etc` se encontra  
# `df /etc`
- Verifique espaço disponível em todos os *filesystem*  
# `df`



# Manipulação de Arquivos Texto

- O UNIX é poderoso no uso de arquivos texto
  - cat      mostra conteúdo do arquivo todo
  - more     mostra conteúdo de arquivo por página
  - diff     exibe diferenças entre dois arquivos texto
  - grep     procura por string em arquivos
  - tail     exibe últimas linhas de um arquivo texto
  - awk      processa arquivo texto através de programa em linguagem própria
  - sed      processa arquivo texto baseado em padrões de modificação

# Manipulação de Arquivos Texto (2)

- Exemplos:

```
cat /etc/fstab
```

```
more /etc/fstab
```

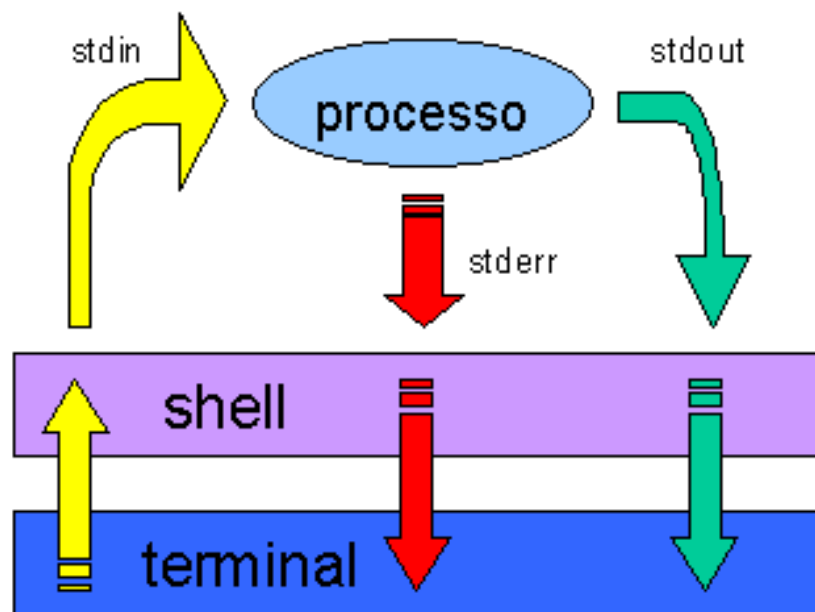
```
diff /etc/fstab /etc/fstab.orig
```

```
grep LANG */*/*
```

```
tail /etc/fstab
```

```
tail -f /etc/httpd/logs/access.log
```

# E/S e Redirecionamentos



# E/S e Redirecionamentos (2)

- *Entrada padrão* (stdin - *standard input*): onde o comando vai ler seus dados de entrada.
- *Saída padrão* (stdout - *standard output*): onde o comando vai escrever seus dados de saída.
- *Saída de erro* (stderr - *standard error*): onde o comando vai enviar mensagens de erro.

# E/S e Redirecionamentos (3)

- **Saída em arquivo:** a saída padrão (*stdout*) do comando é desviada para um arquivo usando o operador ">". Vejamos um exemplo:
  - `ls > listagem.txt`
- **Entrada de arquivo:** a entrada padrão (*stdin*) pode ser obtida a partir de um arquivo usando o operador "<". Um exemplo:
  - `sort < listagem.txt`
- **Uso combinado:** os dois operadores podem ser usados simultaneamente.
  - `sort < listagem.txt > listagem_ordenada.txt`

# E/S e Redirecionamentos (4)

- **Concatenação:** a saída padrão pode ser concatenada a um arquivo existente usando-se o operador ">>", como mostra o exemplo:
  - `ls /etc >> listagem.txt`
- **Saída de erros:** a saída de erros (*stderr*) pode ser redirecionada juntamente com a saída standard. Para isso basta usar o modificador "&1" em conjunção com "2>" ou "2>>". Vejamos um exemplo:
  - 
  - `ls /xpto > teste.txt`  
`ls: /xpto: No such file or directory`
  - `ll /xpto 2> erro.txt`
  - `cat error.txt`  
`ls: /xpto: No such file or directory`

# E/S e Redirecionamentos (5)

## Conduto:

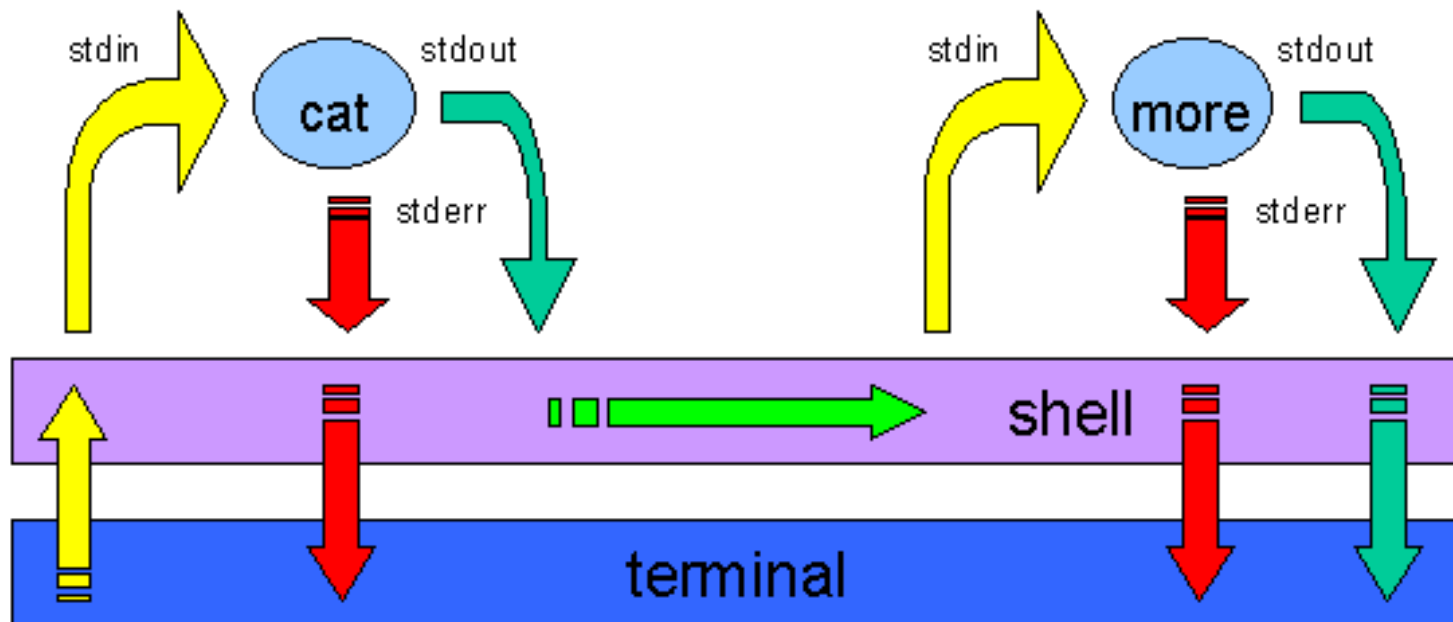
- Conecta *stdout* de um comando à *stdin* de outro
- O resultado da execução de um comando pode ser utilizado como parâmetro de entrada na execução de outro comando
- Meio de condução para transportar dados de um comando para o outro

- Exemplos:

```
cat meuarq | grep palavra | wc -l
```

```
ls -la | grep -i "*.txt"
```

# E/S e Redirecionamentos (6)





# Compactação: arquivos e diretórios

- Compactadores:
  - tar
    - Concatena sem compactar (\*.tar)
  - gzip e gunzip
    - Padrão Unix (\*.gz)
    - Trabalha em conjunto com o tar
  - zip e unzip
    - Compatível com Winzip (\*.zip)
  - bzip2 e bunzip2
    - Maior poder de compactação (\*.bz2)

# Compactação: arquivos e diretórios (2)

## Exemplos:

Os comandos abaixo fazem o backup do diretório /etc :

`tar cf etcback.tar /etc` Gera arquivo etcback.tar (não compactado)

`gzip etcback.tar` Gera arquivo etcback.tar.gz (compactado)

## Possuem o mesmo efeito que:

`tar zcvf etcback.tgz /etc` Gera arquivo etcback.tgz  
(compactado)

## Para descompactar:

`gunzip etcback.tar.gz`

`tar xvf etcback.tar`

## Com o mesmo efeito de:

`tar zxvf etcback.tgz`

# Compactação: arquivos e diretórios (3)

## Exemplos:

Backup do diretório /etc :

```
zip -r etcbac.zip /etc
```

Para descompactar

```
unzip etcbac.zip
```

Compactar o arquivo etcbac.tar

```
bzip2 etcbac.tar          Gera arquivo  
etcbac.tar.bz2
```

Descompactar arquivo etcbac.tar.bz2

```
bunzip2 etcbac.tar.bz2
```

# Localizando arquivos

- Ferramentas

`find` procura por arquivos

`which` procura por executáveis nos diretórios que estão na variável `PATH`

`locate` Procura por arquivo. Mantém base de dados que deve ser atualizada sobre os arquivos em disco.

`locate -u` Atualiza base de dados de arquivos

# Localizando arquivos

## Exemplos

- Procura pelo arquivo fstab  
`find / -name fstab`
- Procura por arquivos sem dono  
`find / -nouser`
- Procura pelo programa ls no PATH  
`which ls`
- Procura por todos os arquivos que contenham a palavra ls no seu nome  
`locate ls`

# Variáveis de Ambiente

- São variáveis globais que controlam o shell.
- Existem várias:
  - HOME → diretório padrão do usuário atual
  - PATH → diretórios onde aplicativos executáveis se encontram
  - MANPATH → diretórios onde estão os manuais dos aplicativos executáveis
- Para se visualizar todas as variáveis de ambiente:  
`# export <ENTER>`
- Para se criar ou trocar o valor de uma variável:  
`# export HOME=/home/jose <ENTER>`

# Variáveis de Ambiente (2)

- Modificar a variável de ambiente HISTSIZE
- HISTSIZE define a quantidade comandos a serem “lembrados” usando setas para cima e para baixo.

```
# export HISTSIZE=3
```

- Execute vários comandos e verifique que somente os últimos 3 são “lembrados” pelo shell.

# Acesso à rede

- Ferramentas que ajudam na verificação da conexão de rede:

`ping` Verifica se há conexão com a máquina sendo “pingada”

`ifconfig` Mostra todas as informações sobre as placas de rede do sistema

`traceroute` Mostra todos os hosts por onde a conexão passa para chegar no destino



# Acesso à rede (2)

```
# /sbin/ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:4D:BB:60
          inet addr:200.210.49.145  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:54186 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21503 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:16666741 (15.8 Mb)  TX bytes:2093701 (1.9 Mb)
          Interrupt:11 Base address:0xcc00
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1125 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1125 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:83498 (81.5 Kb)  TX bytes:83498 (81.5 Kb)
```

# Acesso à rede (3)

## Exemplos

- `ping localhost`
- `ping www.w3s.com.br`
- `ping 200.206.132.150`
  
- `tracert www.w3s.com.br`
- `tracert 200.206.132.150`

# Segurança Interna

- A segurança interna é baseada em usuários e grupos.
- O cadastro pode ser feito através de interface gráfica, ou através de comandos executados no próprio shell.

## Segurança Interna (2)

- Existem grupos cadastrados
  - Usuários são associados a um ou mais grupos
- Para tudo é necessário acesso ao sistema de arquivos.
- Existem 3 possíveis operações:
  - Leitura                      r
  - Escrita                        w
  - Execução                     x

# Segurança Interna (3)

- As operações são restringidas ou liberadas para:
  - O usuário que criou o arquivo
  - Os usuários do grupo a que ele pertence
  - Todos os outros usuários da máquina

# Segurança Interna (4)

- Para verificar quais os direitos em relação à cada arquivo e cada diretório da máquina:

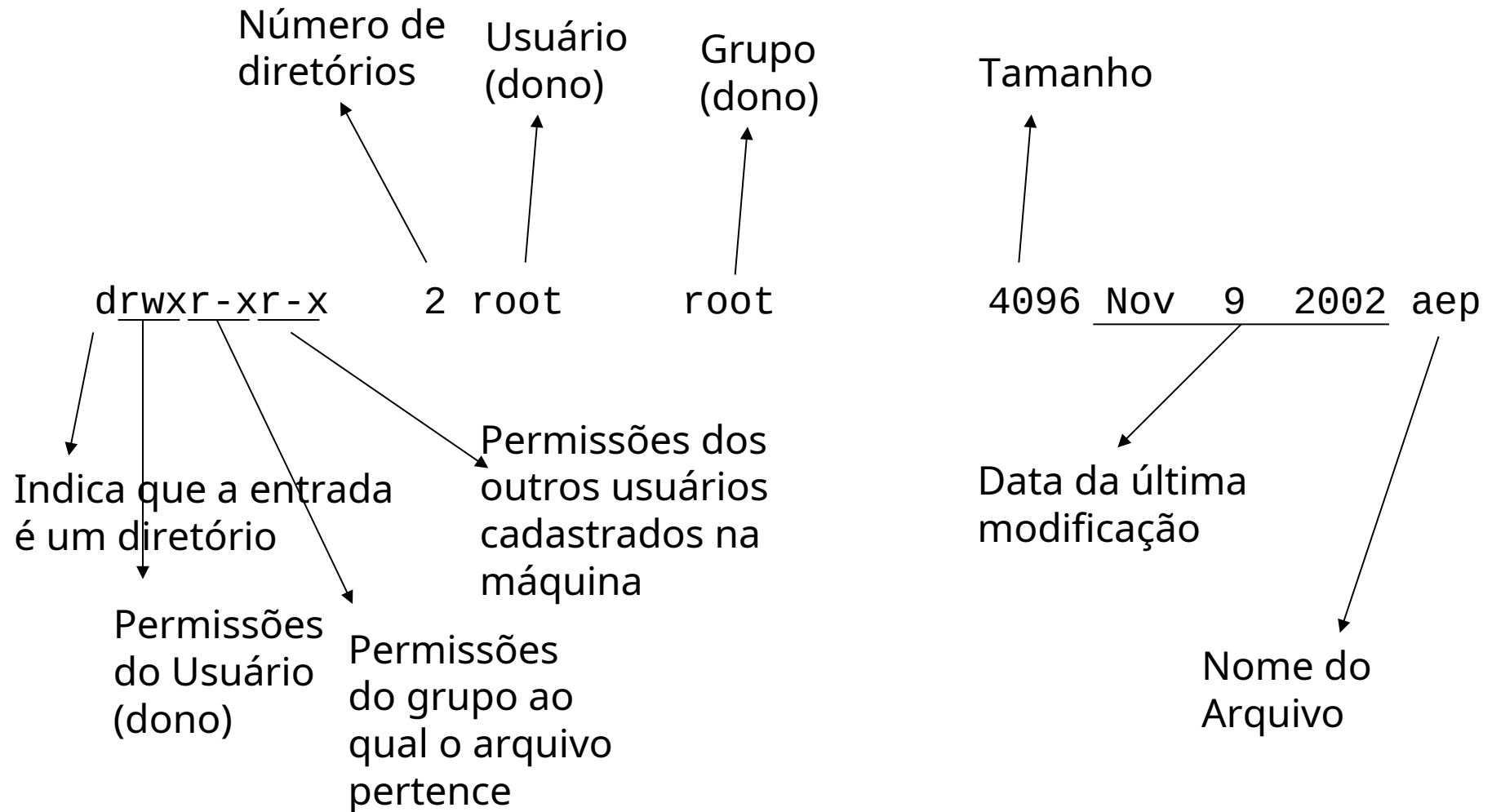
```
ls -l
```

- Ex:

```
# ls -l /etc
```

drwxr-xr-x	2	root	root	4096	Nov	9	2002	aep
-rw-r--r--	1	root	root	688	Aug	23	2002	fstab

# Segurança Interna (5)



# Segurança Interna (6)

- Mudar direitos:  
chmod
- Mudar o usuário proprietário do arquivo  
chown
- Mudar o grupo proprietário do arquivo  
chgrp



# Segurança Interna (7)

- Exemplos

```
chmod o-w *           // Outros usuários não escrevem
chmod g+x *           // Todos do grupo executam
chmod a+r *txt         // Todos os usuários podem ler
chmod o-rwx *txt      // Outros usuários: acesso pleno
chmod u+rwx *txt      // Dono: acesso pleno
chmod u+rwx, a-rwx *   // Não reage como o esperado
chmod a-rwx, u+rwx *   // Reage como o esperado
```

```
chown jose *          // Muda o dono de todos os arquivos do
diretório
```

```
chgrp alunos *rpm     // Troca grupo dos arquivos rpm do
diretório
```

```
chown jose:alunos *log // Troca dono e grupo dos arquivos
                        // *log do diretório atual
```

# Edição de Arquivos

- Edição de arquivos texto é comum no Linux
- Programas disponíveis:
  - vi  
`http://www.eng.hawaii.edu/Tutor/vi.html`
  - emacs  
`http://www.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html`
  - gedit

# Manuais e Infos

- Dois comandos principais para obter ajuda:  
man [<seção>] <comando>  
info <comando>
- São complementares (não duplicados)
- O man possui várias seções:
  - 1 comandos de usuário
  - 1M comandos de administração
  - 2 chamadas do sistema
  - 3 funções em bibliotecas
  - 5 Miscelânea
  - 7 Devices
  - 9 Glossário / Termos usados no Unix

# Manuais e Infos

- Ex:
  - Manual do comando passwd  
`man passwd`
  - Manual do arquivo de configuração passwd  
`man 5 passwd`
  - Manual do comando grub  
`man grub`
  - Info sobre o comando grub (mais completo)  
`info grub`

# Shell Scripts

- Shell scripts seriam os correspondentes no Linux aos arquivos com extensão .bat do MS-DOS
- É possível se escrever shell scripts numa variedade de linguagens, cada uma com sua sintaxe e potencialidades. Ex:
  - Interpretadores nativos: bash, sh, ksh, ...
  - Perl
  - PHP
- Os interpretadores de shell acima suportam uma grande variedade de comandos e podem contar com o poder das ferramentas do UNIX vistas anteriormente

# Shell Scripts

## Criando scripts:

- Os scripts podem ser criados com qualquer editor de textos (gedit, vi, emacs...)
- Após ser criado pela primeira vez, é necessário que o script se torne executável:

```
# chmod a+x <nome do script>
```

- Para executar use o interpretador:

```
# <interpretador> <nome do script>
```

```
# sh <nome do script>
```

Obs: Neste caso, o script deve ter sido escrito para ser executado pelo interpretador sh

# Shell Scripts

Exemplo:

Salve o seguinte conteúdo no arquivo `ola.sh`

```
# Estas sao 3 linhas de comentarios
# Meu primeiro script
# Exibe Ola seguido do nome do usuário atual
clear
echo "Ola $USER"
```

Execute o script acima:

```
# sh ola.sh
```

# Shell Scripts

- Estes scripts podem ser altamente complexos
- As estruturas de controle disponíveis dependem do interpretador
- Um dos interpretadores mais utilizados é o sh (existe em todos os UNIX)
- As particularidades de cada interpretador impõem dificuldades.
- Para se chamar o script diretamente, deve-se colocar o interpretador na primeira linha



# Shell Scripts

Exemplo:

Salve o seguinte conteúdo no arquivo 2

```
#!/bin/sh  
# Exibe Ola seguido do nome do  
  usuário atual  
clear  
echo "Ola $USER"
```

Execute o script acima:

```
# ./ola2.sh
```

# Startup e Shutdown

- Processo de inicialização do Linux:
  - O boot loader invoca o Linux
  - O kernel é descompactado
  - O kernel monta e verifica o filesystem / (root)
  - O kernel invoca o processo `init` que inicia todos os serviços do sistema de acordo com o runlevel setado no arquivo `/etc/inittab`
  - O sistema inicializa os serviços associados ao runlevel inicializado

Runlevel????

# Runlevels

- O *runlevel* determina o tipo de inicialização do sistema. Existem 6 runlevels:
  - 0: halt → Runlevel que permite o sistema ser desligado
  - 1: Single user → Sistema funciona em modo monousuário. Usado para manutenções.
  - 2: Multiuser → Aceita vários usuários locais, mas sem acesso a rede
  - 3: Full Multiuser → Vários usuários com acesso a rede
  - 4: não utilizado
  - 5: X Windows → Multi usuário, com rede e ambiente gráfico
  - 6: reboot → Runlevel que permite a reinicialização da máquina

## Runlevels (2)

- Dependendo do runlevel, diferentes serviços são inicializados ou desativados.
  - exemplos de serviços: apache, mysql, postgres, ftp, ...
- Geralmente servidores operam no runlevel 3 e estações de trabalho no 5.
- Para mudar o runlevel, use `telinit`  
`# /sbin/telinit <runlevel>`
- Para verificar o runlevel atual:  
`# /sbin/runlevel`

# Runlevels (3)

- Configuração manual para algumas distribuições:
    - Diretório dos runlevels: `/etc/rc.d`
    - Os scripts dos serviços ficam em `/etc/rc.d/init.d`
    - Cada runlevel possui um diretório.
      - Ex: `/etc/rc.d/rc3.d/`
    - Os scripts dos serviços são relacionados com os runlevels através de links simbólicos
- ```
# cd /etc/rc.d/rc3.d
# ln -s ../init.d/httpd ./S08httpd
```

# Gerenciamento de Processos

- O Linux é um sistema operacional multiusuário e multitarefas.
- Não são raras as situações onde é necessária a intervenção do usuário na execução de um processo.
- Cada usuário tem o direito de modificar o estado de seus processos apenas (menos o root)

# Gerenciamento de Processos (2)

- Para visualizar o estado atual da máquina, quanto a processos, memória, uso da CPU:

```
# top
```

- Para visualizar todos os processos em execução:

```
# ps aux
```

Cada processo possui um número de identificação

- Para visualizar somente os processos do usuário root

```
# ps aux | grep root
```

# Gerenciamento de Processos (3)

- Para se cancelar a execução de um processo:  
# kill -9 <nro do processo> ou  
# kill KILL <nome do processo>
- Para mudar a prioridade de um processo:  
# renice <nova prioridade> <nro do processo>  
# snice <nova prioridade> <nome do processo>



# Segurança Externa

- É uma das maiores preocupações de qualquer administrador de sistemas.
- No Linux, existem duas ferramentas suportadas diretamente pelo kernel:
  - iptables (kernel versão 2.4 e superior)
  - ipchains (kernel versão 2.2)
  - ipfwadm (kernel versão 2.0)
- A configuração manual é complicada
- Felizmente existe uma ferramenta gráfica que atende às necessidades em 90% dos casos.
  - Application -> Other -> Firewall

# Segurança Externa

Você pode ativar e desativar o Firewall

Também pode abrir e fechar portas de serviços confiáveis ou demais portas

É comum permitir acesso via SSH, para conexão remota segura com o sistema.

Esta ferramenta é um *front-end* para o uso do iptables

# Segurança Externa

Existirão situações onde é necessário o uso do iptables diretamente.

Antes de usá-lo, é recomendado que se leia alguns documentos:

<http://www.netfilter.org/documentation/>

Basicamente o iptables manipula os pacotes IP que fluem pelo sistema. Isto é feito através da manipulação de duas tabelas:

- `filter` → Filtragem de pacotes
  - É isso o que um firewall faz
- `nat` → Redirecionamento de pacotes
  - NAT: Network Address Translation
    - Usado quando se possui poucos endereços IPs disponíveis.
    - O sistema serve de *gateway* para outras máquinas em uma rede interna, que são invisíveis a partir da rede externa. Esta técnica é também chamada *masquerading*.

# Segurança Externa (6)

- O iptables pode ser configurado em qualquer momento pelo usuário root utilizando o programa iptables no *shell*.
- A configuração do iptables é armazenada no arquivo `/etc/sysconfig/iptables`.
  - É aconselhável que se mantenham cópias (backups) deste arquivo.
- A atualização deste arquivo é feita através do comando `iptables-save` que exibe o estado das duas tabelas (`filter` e `nat`) da memória.  

```
# iptables-save > /etc/sysconfig/iptables
```
- Pode-se restaurar uma configuração do iptables usando o `iptables-restore` a partir de um arquivo de configuração:  

```
# iptables-restore < /etc/sysconfig/iptables
```

# Segurança Externa (7)

- Exemplo NAT, sistema com duas placas de rede:
  - eth0: conexão com a internet: IP fixo → 200.206.132.150
  - eth1: conexão com rede interna (192.168.0.\*)
- Comando para se permitir a conexão da rede interna com a internet:  

```
# iptables -t nat -A POSTROUTING -s 192.168.0.0/255.255.255.0 -  
j SNAT --to-source 200.206.132.150
```
- Listando as regras de NAT:  

```
# iptables -t nat -L
```
- Listando as regras de firewall:  

```
# iptables -t filter -L
```

# FIM

Outros temas importantes para administradores de sistemas UNIX:

- Servidores de endereços IP (DHCP)
- NIS (Network Information Services)
- Instalação e configuração do PostgreSQL
- Instalação de servidor FTP e SFTP