

Linux para o LPI

Linux e LPI

www.lpi.org

João Bosco Teixeira Junior

boscojr@gmail.com

Linux – Conceitos Básicos

- Histórico
 - Antes
 - Unix
 - Projeto GNU
 - 1991 – Kernel
 - 1992 – 1a Distribuição
 - 1994 – RPM e Red Hat
- Distribuições
- Licenciamento
 - Proprietário
 - GPL (Restritiva)
 - BSD (Permissiva)

Linux – Conceitos Básicos

- Sistema Operacional
 - Monolítico
 - Modular (Drivers de dispositivos)
 - Multi-Tarefa
 - Multi-Usuário
 - Multi-Processado
- Diversas Arquiteturas de Hardware
- Aplicativos
 - A maioria GPL

Linux – Conceitos Básicos

- Login
 - Usuário
 - Senha
- Interpretador de Comandos (shell)
 - Bash
 - Ambiente
 - Logout
 - Shutdown
- Usuário
 - Administrador: root (uid=0)
 - Ordinário

Linux – Conceitos Básicos

- MS/Windows
 - Case Insensitive
 - \
 - Unidade de Disco
 - Arquivos != diretórios
 - extensões
- GNU/Linux
 - Case sensitive
 - /
 - Montagem
 - Tudo é arquivo
 - Não existe extensão

O Processo de Login

Terminais virtuais
<Alt> + <Fx>
Onde 1 < X < 5

Prompt
De
comando

```
Debian GNU/Linux 5.0 debian-LPI
debian-LPI login: aluno
password:
```

Usuário que
logou @ Nome do host

```
aluno@debian-LPI:~$
```

```
aluno@debian-LPI:~$ pwd
/home/aluno
aluno@debian-LPI:~$ cd ..
aluno@debian-LPI:~$ touch teste
aluno@debian-LPI:~$ cd
aluno@debian-LPI:~$ pwd
/home/aluno
```

comando

Execução de Comandos

- Os comandos do Linux são executados no prompt do interpretador de comandos;
- Em geral os comandos não interativos, não oferecem resposta, eles simplesmente terminam, ou seja, se você executar um comando (não interativo) e ele não te der resposta na tela é por que deu tudo certo (na maioria dos casos);
- Uma linha comando tem três partes:
 - O comando em si
 - Os modificadores (opções)
 - Parâmetros
- Dos itens acima o único obrigatório para todos os comandos é o comando em si. Ex.:

```
$ comando opções parâmetros
```

Prática

Façam!

Esse \$ é o prompt
Por favor não digite

```
$ ls
```

Lista os arquivos de uma pasta, por omissão a pasta atual.
Se a pasta tiver vazia ele mostrará o somente prompt.

```
$ ls -a
```

A opção **-a** modifica o comportamento padrão do comando **ls**, neste caso o comando passa a mostrar também os arquivos ocultos.

```
$ ls -a /etc
```

Agora foi informado uma pasta para a listagem.
Ou seja **/etc/** foi passado como parâmetro.

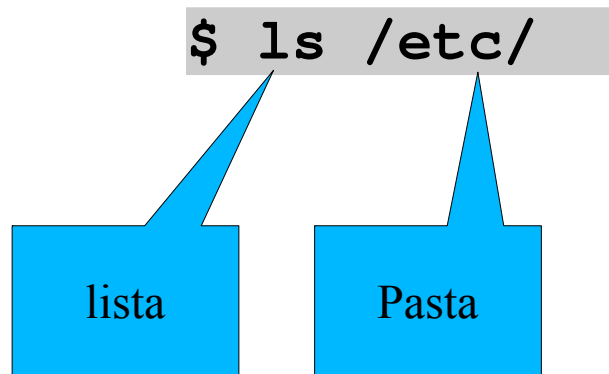
Dicas úteis:

Shift-PgUp permite ver as “telas” anteriores
Seta para cima e para baixo passa os comandos
A tecla tab completa comandos e nomes de arquivos

Trabalhando com arquivos

Listando pastas

- O comando para listar arquivos ou pastas é o **ls**.
- Sua sintaxe é da forma:
 - `ls [opções] Arquivo`
- **Ex:**



Cópia de Arquivos

- O comando para cópia de arquivos ou pastas é o **cp**.
- Sua sintaxe é da forma:
 - cp [opções] Arquivo Arquivo
 - cp [opções] Arquivo Diretório
 - cp -r Diretório Diretório
- Ex:

```
$ cp /etc/passwd .
```

Copia

Arquivo

P/

Diretório atual

FAÇA!
Como você verifica
Se o arquivo
Foi copiado?

Movendo arquivos

- O comando para mover arquivos ou pastas é o **mv**.
- Sua sintaxe é da forma:
 - mv [opções] Arquivo Arquivo
 - mv [opções] Arquivo Diretório
 - mv Diretório Diretório
- Ex:

```
$ mv passwd usuarios
```

Move
(renomeia
Nesta caso)

Arquivo

P/

Outro Arquivo

FAÇA!
Como você verifica
Se o arquivo
Foi renomeado?

Exibindo o conteúdo de um arquivo texto

- O comando para exibir o conteúdo de um arquivo é o `cat` (ele serve para muito mais!).
- Sua sintaxe é da forma:
 - `cat [opções] Arquivo`
- Ex:

```
$ cat usuarios
```

exibe

Arquivo

FAÇA!

Removendo um arquivo

- O comando para remover um arquivo é o **rm**.
- Sua sintaxe é da forma:
 - `rm [opções] Arquivo`
- Ex:

```
$ rm usuarios
```

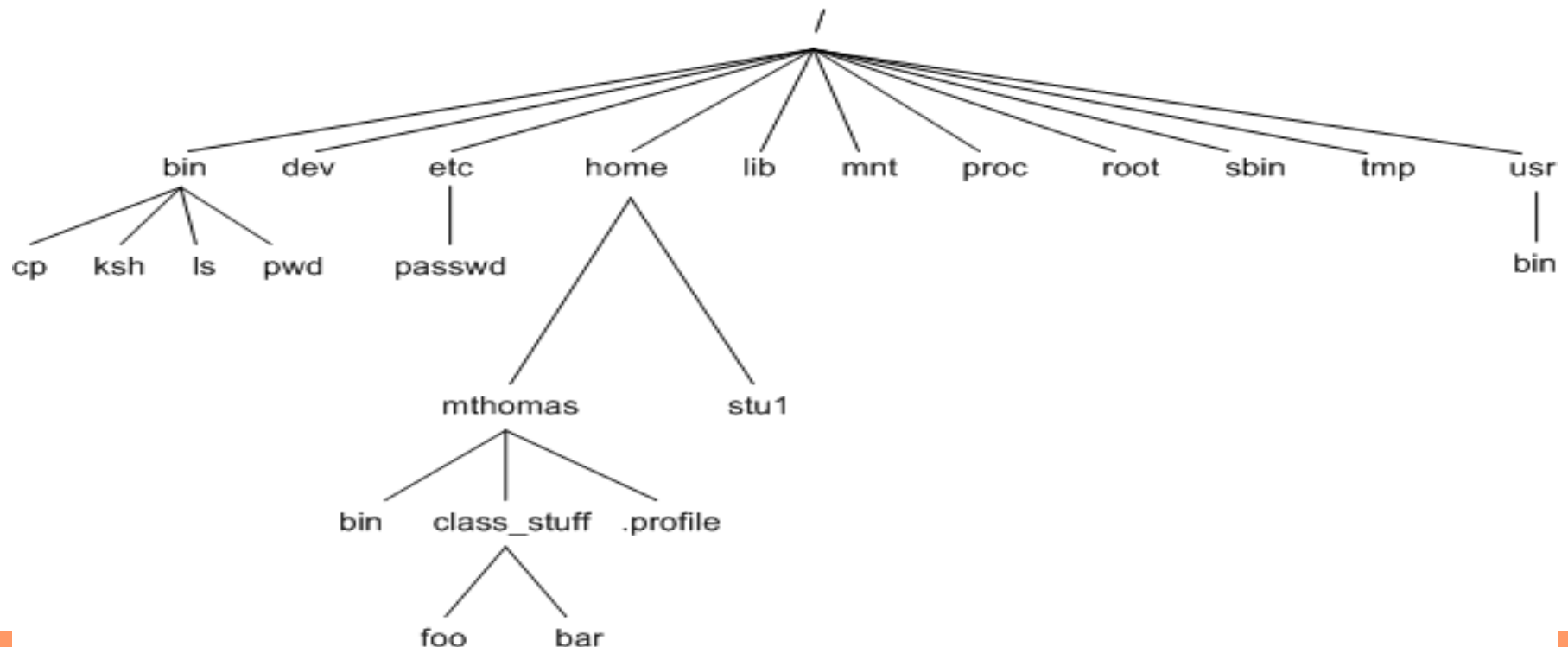
remove

Arquivo

FAÇA!

Diretórios

- No Linux não existe o conceito de unidade de disco (C:, D: , etc..) todos os arquivos disponíveis no sistema devem ser acessados a partir do diretório raiz /. Ex.



Como se achar na hierarquia?

- Com o comando PWD (Path of Work Directory)

```
$ pwd  
/home/aluno
```



Pasta
atual

Como se deslocar na hierarquia?

- Com o comando CD (change directory)

```
$ pwd
/home/aluno
$ cd /tmp
$ pwd
/tmp
$ cd -
$ pwd
/home/aluno
```

The diagram illustrates the process of navigating through a directory hierarchy using the 'cd' command. It shows a terminal session where the user starts in the directory /home/aluno, moves to /tmp, and then returns to /home/aluno. Three callout boxes provide explanations for each step: 'Pasta atual' points to the initial directory, 'Foi para a Pasta /tmp' points to the directory after the first 'cd' command, and 'Voltou para A pasta anterior' points to the directory after the second 'cd' command.

Pasta atual

Foi para a Pasta /tmp

Voltou para A pasta anterior

Criação de pastas

- O comando usado para criar pastas é o **mkdir**.
 - A sua sintaxe é:
 - `mkdir [opções] Pasta`
 - Exemplo:

```
$ mkdir teste
$ cd teste
$ pwd
$ /home/aluno/teste
$ cd -
$ pwd
/home/aluno
```

Home Directory (Diretório pessoal)

- Nos sistemas unix de maneira geral o caracter ~ é usado para representar o home directory.
- Ou seja:
- Deslocar-se de um lugar para a **pasta pessoal**
 - # cd ~
- Copiar um conteúdo para a **pasta pessoal**
 - # cp /etc/hosts ~
- Listar o conteúdo da **pasta pessoal**
 - # ls ~

O recurso de Autocompletar

<TAB>

- A tecla <tab> é sua amiga e pode ser usada por padrão para completar comandos e arquivos.
- Ex: digamos que você queira criar um arquivo na pasta /etc/network/run.
- O comando seria
- # touch /etc/network/run/teste.txt
 - 32 toques
- # tou<tab> /e<tab>/netw<tab>/r<tab>/teste.txt
 - 26 toques

Obtendo ajuda

- Internet:
 - Howtos: Tutoriais
 - Listas de Discursão
 - Blogs (brasil: vivaolinux)
- No sistema
 - Help on console: Opção --help dos comandos
 - Man pages: Man comando

103 – Comandos básicos

103.1 – Comandos cobrados no LPI

- Skills
 - Usar comandos do shell na linha de comandos para resolver tarefas básicas
 - Usar e modificar o ambiente shell, incluir, referenciar e exportar variáveis.
 - Usar e editar o histórico de comandos
 - Invocar comandos dentro e fora do caminho definido.
- Comandos:
 - Bash, **echo** , **export**, **pwd**, env, set , unset, **man**, **uname**, **history**

103 – Comandos básicos

103.1 – Linha de Comando

- **man:** Exibe a página de manual de um comando
 - Sintaxe: `man [pagina][comando]`
- **apropos:** Busca uma expressão nas páginas **man**.
 - Sintaxe: `apropos expressão`

103 – Comandos básicos

103.1 – Linha de Comando

- **uname:** exibe informações do sistema
 - Sintaxe: `uname [-opcoes]`
 - Opções:
 - -a todas as informações (all)
 - -v versão do kernel
 - -n nome do host (nodename)
 - Informações disponíveis:
 - versão do kernel, release do kernel, cpu, Arquitetura
- **history:** Lista dos comandos digitados
 - Sintaxe: `history [-opcoes]`
 - Opções:
 - -c : Limpa o histórico.

103 – Comandos básicos

103.1 – Linha de Comando

- **history:** Lista dos comandos digitados
 - Sintaxe: `history [-opcoes]`
 - Opções:
 - `-c` : Limpa o histórico.
 - Arquivo
 - `~/.bash_history`

103 – Comandos básicos

103.1 – Linha de Comando

- **pwd:** exibe o diretório de trabalho atual
 - Sintaxe: `pwd`
 - Comando embutido
- **export:** exporta variáveis de ambiente
 - Sintaxe: `export VARIÁVEL`
 - Comando Embutido
 - Uso de variáveis:
 - Configuração do ambiente, ex: `PATH`, `TMOUT`, `PS1`
 - Informativas: `SHELL`, `HOME`, `PWD`
 - Uso do `$` somente para leitura (exibição).

103 – Comandos básicos

103.1 – Linha de Comando

- **echo:** exibe uma linha de texto.
 - Sintaxe: `echo [opções] texto`
 - Principais opções
 - `-n` : Não imprime a quebra de linha no final do texto
 - `-e` : Interpreta as sequências de escape
 - Exemplo de uso: `echo -n "isso é um texto"`
 - Sequencias de escape
 - `\a` - Alerta, `\xNNN` caracter em hexadecimal
 - Maiores detalhes na sobre sequencias na man page.
 - Uso das Aspas
 - Simples: Imprime o texto que sem interpretar caracteres
 - Duplas: Interpreta o texto antes de imprimir

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- Skills
 - Copiar, mover e remover arquivos e pastas individuais;
 - Copiar, move e remover arquivos e pastas recursivamente;
 - Copiar e Remover múltiplos arquivos e pastas recursivamente;
 - Usar caracteres coringas para filtrar arquivos e pastas;
 - Usar o comando file para listar e agir em arquivos com base no tipo, tamanho ou tempo;
 - Uso do cpio and dd.
- Comandos:
 - **cp, find, mkdir, mv, ls, rm, rmdir, touch, tar, cpio, dd, file, gzip, gunzip, bzip2, xz, file globbing.**

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **mkdir:** criação de diretórios
 - Sintaxe: `mkdir [opções] diretorios`
 - Principais opções
 - `-p --parent`: Cria o diretório pai se não existir
 - `-m --mode=MOD0`: Ajusta as permissões do diretório criado
 - Exemplo de uso: `mkdir -p /tmp/dir/subdir`
- **rmdir:** remoção de diretórios
 - Sintaxe: `rmdir [opções] diretorios`
 - Principal opção:
 - `-p --parent`: Remove o diretório pai se existir
 - Obs: Só remove diretórios vazios

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **cd**: navegação na árvore de diretórios
 - Comando embutido do interpretador
 - Sintaxe: `cd diretorio`
 - Alguns exemplos de uso
 - `cd ~` : muda para a pasta HOME do usuário
 - `cd` : muda para a pasta HOME do usuário (também)
 - `cd -` : volta para a pasta onde estava anteriormente
 - Deslocamento
 - Absoluto (passando todo o caminho)
 - Relativo (usando-se os diretórios especiais `'.'` e `'..'`)

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **cp: cópia de arquivos**
 - Sintaxe: `cp [opções] Origem Destino`
 - Alguns exemplos de uso
 - `cp arquivo diretório`
 - `cp -r diretório diretório`
 - `cp arquivo arquivo`
 - Principais opções
 - `-r, -R --recursive`: cópia diretório e seu conteúdo
 - `-p --preserve=[mode, owner, timestamp]`: preserva atributos
 - `-v --verbose`: mostra o que está sendo feito
 - `-i --interactive`: Pergunta na hora de sobrescrever

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **mv:** move arquivos e diretórios
 - Sintaxe: `mv [opções] Origem Destino`
 - Alguns exemplos de uso
 - `mv arquivo diretório`
 - `mv diretório diretório`
 - `mv arquivo arquivo`
- **rm:** apaga arquivos de diretórios
 - Sintaxe: `rm [opções] diretorios`
 - Principais opções:
 - `-r, -R --recursive` : remove o diretório e o seu conteúdo
 - `-f` : nunca pergunta antes de remover
 - `-i` : pergunta antes de cada remoção

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **ls:** lista arquivos e diretórios
 - Sintaxe: `ls [opções] parametros`
 - Parâmetros pode ser uma lista contendo arquivos e dirs
 - Exemplo de uso: `ls -l /tmp /home`
 - Principais opções:
 - `-l` : Lista longa (vários atributos)
 - `-a --all`: lista também os arquivos iniciados por `'.'`
 - `-A --almost-all`: O mesmo que `-a` exclui o `'.'` e o `'..'`
 - `-d --directory`: Não entra na pasta
 - `-h --human-readable`: Tamanhos em K,M,G

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **touch**: altera a data de modificação e acesso de um arquivo ou diretório
 - Sintaxe: `touch [opções] parametros`
 - Parâmetros pode ser uma lista contendo arquivos e dirs
 - Exemplo de uso: `touch /tmp/arq`
 - Principais opções:
 - `-c` : Não cria arquivo

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **find:** busca no arquivos de diretórios no sistema
 - Sintaxe: `find local [opções] expressao`
 - Local é onde a busca deve ser feita
 - Expressão é o padrão de que deve ser procurado
 - Opções podem ser testes e/ou ações
 - Exemplos de uso:
 - `find /etc/ -name 'h*'`
 - `find /etc/ -type d -name 'b*'`
 - `find -amin +1 -name '*.c' -exec rm -rf {} \;`

- **Atividade:**

- Encontre todos os arquivos do sistema com permissões 400

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **dd:** Faz um dump de um arquivo
 - Sintaxe: `dd [opções]`
 - Algumas Opções
 - `if`: arquivo de entrada
 - `of`: arquivo de saída
 - `bs`: tamanho do bloco para leitura ou escrita (Por vez)
 - `count`: quantidade de vezes que os blocos serão lidos ou escritos
 - Exemplos de uso:
 - `dd if=/dev/sda of=/dev/sdb`
 - `dd if=/dev/urandom of=teste.img bs=1024 count=4`
- **Atividade:**
 - Crie um arquivo de 1M com zeros (`if=/dev/zero`)

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **file:** determina o tipo do arquivo
 - Sintaxe: `file [opções] arquivo`
 - Algumas Opções
 - `-f --file-from FILE`: Obtém a Lista de um arquivo
 - `-z --uncompress`: Testa dentro de arquivos comprimidos
 - `-p --preserve-date`: Preserva o tempo de acesso
 - Magic Numbers
 - Faz parte do formato
 - Determina o tipo de arquivo (assinatura)

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **gzip: GNU zip (Compactador)**
 - Sintaxe: `gzip [opções] arquivo`
 - Algumas Opções
 - `-c --stdout` : Não altera o arquivo, joga na saída padrão
 - `-d --decompress` : Descomprime o arquivo compactado
 - `-S --suffix=SUF` : Sufixo dos arquivos gerados
 - `-1` : compacta mais rápido
 - `-9` : compacta melhor
 - Exemplo de Uso:
 - `gzip arquivo.txt`
- **gunzip: `gzip -d`**

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **bzip2: GNU zip (Compactador)**
 - Sintaxe: `bzip2 [opções] arquivo`
 - Algumas Opções
 - `-c --stdout` : Não altera o arquivo, joga na saída padrão
 - `-d --decompress` : Descomprime o arquivo compactado
 - `-s --small` : Usa menos memória
 - Exemplo de Uso:
 - `bzip2 arquivo.txt`
- **bunzip2: `bzip2 -d`**

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **tar: Tape Archiver (“Aglutinador”)**

- Sintaxe: `tar [opções]`

- Algumas Opções

- `-c --compress` : aglutina os arquivos

- `-x --extract`: Extrai os arquivos

- `-z --gzip` : Usa o gzip para comprimir ou descomprimir

- `-j --bzip2` : Usa o bzip2 para comprimir ou descomprimir

- `-f --file`: Gera em arquivo

- `-v --verbose`: Exibe os arquivos que estão sendo aglutinados

- `-t --list` : Exibe a lista de arquivos

- Exemplo de uso:

- `tar -czvf arquivo.tar.gz /etc`

- `tar -xzvf arquivo.tar.gz`

103 – Comandos básicos

103.3 – Comando para gerência de arquivos

- **cpio**: Archiver (“Aglutinador”)
 - Sintaxe: `cpio [opções]`
 - Algumas Opções
 - `-o --create` : Obtém a Lista de um arquivo
 - `-i --extract`: Testa dentro de arquivos comprimidos
 - `-t --list`: Exibe os arquivos que estão sendo aglutinados
 - Exemplo de uso
 - `find /etc/ | cpio -o > etc.cpio`
 - `cpio -t < etc.cpio`
 - `cpio -i < etc.cpio`

103 – Comandos básicos

- 103.4 – Usar streams, pipe e redirecionamento
 - Redirecionar entrada e saída padrão e saída de erro
 - Enviar a saída de um comando para entrada de outro via PIPE
 - Usar a saída de um comando como argumento para outro comando
 - Enviar a saída padrão e de erro para um arquivo
- Comandos:
 - tee, xargs

103 – Comandos básicos

- 103.5 – Manipulação de Processos
- Skills
 - Executar jobs em foreground and background
 - Permitir que um processo execute após o logout
 - Monitorar processos ativos
 - Selecionar e ordenar processos para exibição
 - Enviar sinais a processos
- Comandos
 - **&, bg, fg, jobs, kill, nohup, ps, top, free, uptime, pgrep, pkill, killall, screen**
 -

103 – Comandos básicos

- 103.6 – Prioridade de Processos
- Skills
 - Conhecer a prioridade padrão do sistema
 - Rodar um programa com uma prioridade diferente da padrão
 - Mudar a prioridade de um processo em execução
- Comandos
 - Nice, **ps**, renice, **top**

103 – Comandos básicos

- 103.4/5/6 – Processos
 - Programas em execução
 - Atributos Principais
 - UID, GID, PID, PPID
 - Estados dos Processos
 - Pronto ou executando (R), dormindo (S), parado (T) e zumbi(Z)
 - Proc file system (/proc)
 - Comunicação entre Processos
 - Composição de comandos, PIPE e Redirecionamento
 - Código de Retorno \$?
 - Entrada(0) e Saída(1) Padrão e Saída(2) de Erro

103 – Comandos básicos

- 103.4/5/6 – Processos (Cont)
 - Criação e Interrupção de processos
 - Foreground e Background
 - &, bg, fg, nohup
 - Interrupção de Processos (ctrl-z)
 - Prioridade
 - Nice, Renice
- Comandos
 - jobs, kill, killall, ps, top, free, uptime, pgrep, fuser, pstree, tee, xargs, nice, renice

103 – Comandos básicos

103.4/5/6 – Processos

- **jobs:** mostra o status dos jobs que foram iniciados na sessão (shell) atual
 - Sintaxe: `jobs [opções]`
- **kill:** Mata(termina) processos pelo PID
 - Sintaxe: `kill -SINAL [opcoes]`
`lista_de_PIDs`
 - `-l` : Obtém a Lista de um arquivo
 - Sinais
 - `-l SIGHUP` : Reinicia
 - `-9 SIGKILL`: Termina imediatamente
 - `-15 SIGTERM`: Permissão para terminar
 - Exemplo de uso
 - `kill -l`
 - `kill -9 $$`

103 – Comandos básicos

103.4/5/6 – Processos

- **killall:** O mesmo que kill, mas usa o nome invés do PID
 - Sintaxe: `killall -SINAL [opções] processo`
 - Exemplo de uso:
 - `killall -9 bash`
- **free:** Exibe informações sobre o uso de memória
 - Sintaxe: `free [opcoes]`
 - `-k, m, g` : exibe em kilobytes, megabytes, gigabytes
 - `-s time` : atualiza a informação em TIME segundos
 - Exemplo de uso
 - `free -m -s 1`
 - Questão: Qual o sinal default para o comando kill e killall?

103 – Comandos básicos

103.4/5/6 – Processos

- **uptime:** Informações sobre o sistema (carga, tempo no ar, quantidade de usuários que usaram o sistema)
 - Sintaxe: `uptime`
 - O que é a carga?
- **ps tree:** Exibe a árvore de processos
 - Sintaxe: `ps tree [opcoes]`
 - `-a` : exibe os argumentos de linha de comando
 - `-p` : mostra o pid dos processos
 - Exemplo de uso:
 - `ps tree -a`

103 – Comandos básicos

103.4/5/6 – Processos

- **ps**: Lista os processo dos sistema
 - Sintaxe: `ps [opcoes]`
 - Algumas opções:
 - `-e` : exhibe todos os processo
 - `-x` : exhibe informações de ambiente
 - `-o`: seleciona os campos
 - `-f`: lista cheia
 - Exemplo de uso:
 - `ps -ef`
 - `Ps -eo pid,cmd,stat`

103 – Comandos básicos

103.4/5/6 – Processos

- **top:** Lista os processo dos sistema em tempo real
 - Sintaxe: `top [opcoes]`
 - Algumas opções:
 - `-u USER` : mostra só os processos de USER
 - `-p PID` : exhibe só o processo com pid igual a PID
 - `-d TIME`: faz atualizações a cada TIME segundos
 - `-n`: Número de interações
 - Exemplo de uso:
 - `top -d 1`
 - `top -d 1 -u root -n 10`
 - Questão: Por quanto tempo o comando anterior vai executar?

103 – Comandos básicos

103.4/5/6 – Processos

- **pgrep:** Lista os processo dos sistema de acordo com um padrão
 - Sintaxe: `pgrep [opcoes] padrao`
 - Algumas opções:
 - `-u UID` : mostra só os processos do usuário com uid igual a UID
 - `-d DELIM` : separa a listagem com o delimitador igual a DELIM
 - Exemplo de uso:
 - `pgrep -u 0`
 - `pgrep -u 0 -d ', ' bash`

103 – Comandos básicos

103.4/5/6 – Processos

- **fuser:** trata processos usando arquivos
 - Sintaxe: `fuser [opcoes] ponto_de_montagem`
 - Algumas opções:
 - `-a --append` : Adiciona o conteúdo ao arquivo
 - Exemplo de uso:
 - `fuser -k /home`

103 – Comandos básicos

103.4/5/6 – Processos

- **tee:** Ler da entrada padrão e escreve na saída padrão e no arquivo.
 - Sintaxe: `tee [opcoes] arquivo`
 - Algumas opções:
 - `-a --append` : Adiciona conteúdo ao arquivo informado
 - Exemplo de uso:
 - `ls | tee listagem.txt`

103 – Comandos básicos

103.4/5/6 – Processos

- **xargs:** Usa o texto recebido na entrada padrão e usa como argumento do comando informado.
 - Sintaxe: `xargs [opcoes] comando`
 - Algumas opções:
 - `-a arquivo` : Usa o texto do arquivo ao invés da entrada padrão.
 - Exemplo de uso:
 - `ls | xargs rm -rf`

103 – Comandos básicos

103.2 – Processamento de texto com filtros

- Skills
 - Enviar arquivos de textos e streams de texto através dos utilitários de filtragem para modificar a saída gerada usando comandos disponíveis no pacote GNU/textutils.
- Comandos
 - **cat, cut, expand, fmt, head, join, less, nl, od, paste, pr, sed, sort, split, tail, tr, unexpand, uniq, wc.**

103 – Comandos básicos

103.2 – Processamento de texto e filtros

- Manipulação de texto
- Criar Pipelines complexos (|)
 - Contar Palavras e Linhas
 - Converter Caracteres
- Fluxos de Entrada e Saída
 - Stdin (0) – Entrada Padrão
 - Stdout (1) – Saída Padrão
 - Stderr (2) – Saída de Erro

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **cat** : concatena arquivos e lista na saída padrão
 - Abreviação para concatenate
 - Sintaxe: `cat [opções] parametros`
 - Parâmetros pode ser uma lista contendo arquivos
 - Exemplo de uso: `cat arq1.txt arq2.txt`
 - Concatena `arq1.txt` e `arq2.txt` e exibe na saída padrão
 - Outro Exemplo: `cat > arq3.txt`
 - Tudo que for digitado na entrada padrão será escrito no arquivo até que se digite `ctrl-d`.
 - Principais opções:
 - `-n --number` : número as linhas na saída
 - `-E --show-ends` : mostra \$ no final de cada linha
 - `-s --squeeze-blank`: não exibe linhas em branco
 - `-A --show-all` : Exibe caracteres não “imprimíveis”

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **od** (octal dump): Faz o dump de um arquivo (octal, hexa, asc, etc.)
 - Sintaxe: `od [opções] arquivo`
 - Outro uso comum: `od -x -N 512 arquivo`
 - Exibe os primeiros 512 bytes (-N 512) em hexa (-x)
 - Principais opções:
 - `-x` : imprime a saída em formato hexadecimal
 - `-N NUM`: imprime os NUM primeiros bytes do arquivo
 - **Maiores Detalhes**
 - `od --help`
 - `man od`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **split** : Divide um arquivo em várias partes
 - Sintaxe: `split [opções] arquivo PREFIX`
 - Outro uso comum: `split -b 8k arquivo.txt`
`arq`
 - Vai criar arquivos com 8k (-b 8k) cujo os nomes terão arq como prefixo
 - Principais opções:
 - `-b --size=NUM` : divide arquivo em arquivos de NUM bytes
 - `-l --lines=NUM`: divide arquivo em arquivos de NUM linhas
 - **Maiores Detalhes**
 - `split --help`
 - `man split`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- Atividade

- Faça um backup do seu diretório /etc/ usando tar
- Faça um split do arquivo gerado de forma que cada parte gerada possua 100k e cada arquivo deverá ter no nome o prefixo etc.
 - Quantos arquivos foram gerados?
- E Agora? Como fazer para restaurar o arquivo gerado?

Para Dividir:

```
# tar cvf etc.tar /etc/  
# split -b 600k etc.tar etc
```

Para Juntar Novamente:

```
# cat etcaa etcab etcac > etc2.tar
```

Para Testar:

```
# tar xvf etc2.tgz
```

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **wc**: conta palavras, linhas, caracteres de um arquivo texto
 - Sintaxe: `wc [opções] arquivo`
 - Exemplo de uso: `dmesg | wc -l`
 - Exibe a quantas linhas (-l) existem no log do kernel (dmesg)
 - Principais opções:
 - `-l --lines` : exibe o número de linhas do arquivo
 - `-w --words` : exibe o número de palavras
 - Maiores Detalhes:
 - `wc --help`
 - `man wc`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **head:** imprime as primeiras linha de um arquivo
 - Sintaxe: `head [opções] arquivo`
 - Exemplo de uso: `dmesg | head -3`
 - Exibe as três primeiras (-3) linhas do log do kernel (dmesg)
 - Principais opções:
 - `-n --lines N`: exibe as primeiras N linhas do arquivo
 - Maiores Detalhes
 - `head --help`
 - `man head`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **tail:** Imprime as últimas linhas de um arquivo
 - Sintaxe: `tail [opções] arquivo`
 - Exemplo de uso: `dmesg | tail -3`
 - Exibe as três últimas (-3) linhas do log do kernel (dmesg)
 - Principais opções:
 - `-n --lines N`: exibe as últimas N linhas do arquivo
 - `-f --follow`: acompanha o crescimento do arquivo
 - Maiores Detalhes
 - `tail --help`
 - `man tail`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- ATIVIDADE (juntando as peças)

- Baixe o arquivo de LOGS:

```
# wget http://www.boscojr.com/secure.gz
```

- LOGS: Registros de Atividades de Software ou Sistema

- Informações no Arquivo

- Data e Horário (Dia/Mes/Ano Hora:Minuto:Segundo)
- Hostname: Nome da Máquina
- Sistema [pid] : Software que gerou o log;
 - PID: Número do Processo no Sistema
- Registro: Informação que foi registrada
-

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- ATIVIDADE (juntando as peças)

- Baixe o arquivo:

```
#wget http://www.boscojr.com/secure.gz
```

- Descopacte-o

- Trata-se de um arquivo de log de acessos, bem sucedidos ou não, a um determinado servidor

- Questões

- Como faço para saber quando se iniciam os logs?
- E Quando terminam?
- E como faço para descobrir a ocorrência que está registrada na linha 399 e 400?
- Quantas linhas tem o arquivo?

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **tr:** traduz ou deleta caracteres
 - Sintaxe: `tr [opções] SET SET2 < arquivo`
 - Exemplo de uso: `tr ' ' '\t' < /etc/hosts`
 - Troca espaço por tabulações no arquivo `/etc/hosts`
 - Principais opções:
 - `-d --delete` : deleta o caractere especificado
 - Maiores Detalhes
 - `tr --help`
 - `man tr`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **expand**: converte tabulações para espaço
 - Sintaxe: `expand [opções] arquivo`
 - Exemplo de uso: `expand -t 1 /etc/hosts`
 - Principais opções:
 - `-t NUM` : substitui por NUM espaços em branco
 - Maiores Detalhes
 - `expand --help`
 - `man expand`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **unexpand:** converte espaço em branco para tabulações
 - OBS: tem que ter pelo menos 2 espaços em branco
 - Sintaxe: `unexpand [opções] arquivo`
 - Exemplo de uso: `unexpand -t 3 /etc/hosts`
 - Substitui no arquivo `/etc/hosts` em todos os lugares onde houver uma sequência de três (-t 3) ou mais caracteres branco substitui por tabulação.
 - Principais opções:
 - `-t NUM` : substitui a partir de NUM tabulações
 - Maiores Detalhes
 - `expand --help`
 - `man expand`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **sort:** ordena as linhas de arquivo texto
 - Sintaxe: `sort [opções] arquivo`
 - Outro uso comum: `comando | sort`
 - Principais opções:
 - `-b --ignore-leading-blanks` : ignora espaço branco
 - `-f --ignore-case`: campos que serão exibidos
 - `-r --reverse` : ordena na ordem inversa
 - `-o --output=ARQ` : escreve a saída em ARQ
 - Maiores Detalhes
 - `sort --help`
 - `man sort`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **uniq**: reporta ou omite linhas repetidas
 - Sintaxe: `uniq [opções] arquivo`
 - Outro uso comum: `comando | uniq`
 - Principais opções:
 - `-c --count` : informa a quantidade de ocorrências
 - `-i --ignore-case`: campos que serão exibidos
 - `-d --repeated` : exhibe só o que tiver repetição
 - `-u --unique` : exhibe só que tiver uma única ocorrência
 - Maiores Detalhes
 - `uniq --help`
 - `man uniq`
 - OBS: O `uniq` só compara linhas adjacentes

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros

- **cut** : imprime partes selecionadas de um arquivo
 - Sintaxe: `cut [opções] parametros`
 - Parâmetros é o nome de um arquivo
 - Exemplo de uso: `cut -f 1 arq1.txt`
 - Principais opções:
 - `-d --delimiter=DELIM` : ajusta o separador de campos
 - `-f --fields=LIST` : Seleciona somente os campos da lista
 - `-s --only-delimited`: somente imprime linhas que tenham o delimitador

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros

- **join:** Concatena registro de dois arquivos baseado em algum campo específico
- O Campo deve estar ordenado nos dois arquivos
 - Sintaxe: `join [opções] arq1 arq2`
 - Principais opções:
 - `-t DELIM`: usa DELIM como delimitador de entrada e de saída
 - `-j NUM`: Baseia-se no campo do arquivo NUM (1 ou 2)
 - **Maiores Detalhes**
 - `join --help`
 - `man join`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros

- **paste:** Cola linhas de arquivos diferentes lado a lado
 - Sintaxe: `paste [opções] arq1 arq2`
 - Outro uso comum: `paste -d ', ' arq1 arq2`
 - Principais opções:
 - `-d DELIM`: usa DELIM como delimitador ao invés de TAB
 - **Maiores Detalhes**
 - `paste --help`
 - `man paste`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros

- Outros Comandos
- **pr**: formata arquivo texto para impressão
 - Cabeçalho, data, número da página, etc.
 - Sintaxe: `pr [opções] arquivo`
 - Exemplo de uso: `pr /etc/hosts`
- **nl**: numera as linhas de um arquivo texto
 - Sintaxe: `pr [opções] arquivo`
 - Exemplo de uso: `nl /etc/hosts`
- **fmt**: formata o conteúdo do arquivo texto de modo a caber nas margens
 - Sintaxe: `fmt [opções] arquivo`
 - Exemplo de uso: `fmt -w 20 /etc/hosts`

103 – Comandos básicos

103.2 – Processamento de Texto e Filtros - Comandos

- **sed** (Stream EDitor): edição não interativa de arquivos
 - Com o sed é possível:
 - Remover linhas de Arquivo
 - Alterar palavras dentro do texto
 - Inserir palavras
 - O resultado da alteração é enviado para a saída padrão
 - Sintaxe: `sed [opções] comando arquivo`
 - Exemplo de uso: `sed 's/a/A/g' /etc/hosts`
 - Troca todas as ocorrências do caractere 'a' pelo caractere 'A'
 - Outro exemplo de uso: `sed -i '2,3d' /etc/hosts`
 - Deleta as linhas 2 e 3 do arquivo direto no /etc/hosts
 - Principais opções:
 - `-i` : altera direto no arquivo
 - `-f FILE` : sed lerá os comandos de file

103 – Comandos básicos

103.7 – Busca em Arquivos textos com RegEx

- Skills
 - Criar expressões regulares simples contendo diversos elementos.
 - Usar ferramentas que suportem expressões regulares para realizar busca de arquivos no disco e conteúdos em arquivos.
- Comandos
 - **grep, egrep, fgrep, sed**

103 – Comandos básicos

103.7 – Busca em Arquivos textos com RegEx

- RegEX = Expressões Regulares
 - Método formal para especificar um padrão de texto
 - Não tem relação com os coringas usados para listar arquivos
 - Termos
 - ER, RE, regexp, (regex)*
 - Metacaracteres
 - Caracteres que ditam o funcionamento das expressões regulares
 - Caracteres literais
 - Caracteres sem efeito para a expressão regular
 - Padrão (pattern)
 - String que será usada na busca do texto
 - Casamento (Matching)
 - Quando alguma parte do texto combina com o padrão especificado

103 – Comandos básicos

103.7 – Busca em Arquivos textos com RegEx

- Tipos de Metacaracteres
 - Representante
 - Tem função de representar um ou mais caracteres
 - Ponto (.) - Casa com qualquer coisa
 - Lista [] - Casa somente se o caractere estiver na lista
 - Classes de caractere [:upper:], [:lower:], [:digit:], etc...
 - Lista Negada [^] - Casa somente se o caractere não estiver na lista
 - Quantificador (Depois do caractere)
 - Informa o número de repetições permitidas
 - Opcional ? - Zero ou uma vez
 - Asterisco (*) - Zero ou mais vezes
 - Mais (+) - Pelo menos uma vez
 - Âncora:
 - Circunflexo (^) - Casa com o começo da linha
 - Cifrão (\$) - Casa com o final da linha
 - Borda (\b) – Casa com o começo ou final de uma palavra

103 – Comandos básicos

103.7 – Buscar texto usando expressões regulares

- Comandos
 - **grep** – conjunto restrito de expressões regulares
 - **egrep** – conjunto estendido (completo)
 - **fgrep** – Não usa expressões regulares
- No fim todos são grep
 - `egrep = grep -E`
 - `fgrep = grep -F`

103 – Comandos básicos

Atividade (SUPER PIPE)

- Log de tentativas de acesso a um determinado host
- Gerar uma lista dos endereços IPS que estão tentando acessar o servidor.
 - Peçaço do Log:

```
<data> sshd[6382]: pam_unix(sshd:auth): check pass; user unknown
```

```
<data> sshd[6382]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=rhost=200.215.223.43
```

```
<data> sshd[6382]: pam_succeed_if(sshd:auth): error retrieving information about user franci
```

```
<data> sshd[6382]: Failed password for invalid user franci from 200.215.223.43 port 51344 ssh2
```

- Baixe o arquivo:

```
#wget http://www.boscojr.com/secure.gz
```

- Descompacte o arquivo

- A linha em negrito deve ser filtrada no arquivo

- use o comando **grep** com a string “Failed password”

```
#grep -i "failed password" [arquivo-descompactado]
```


103 – Comandos básicos

- 103.8 – vi
- Skills
 - Navegar em um documento usando vi;
 - Usar os modos básicos do vi;
 - Inserir, editar, deletar, copiar e encontrar texto.
- Comandos do vi
 - Vi, /, ? , h,j,k,l, i, o, a, c, d, p, y, dd, yy, ZZ, :w!, :q! , :e!
 -
 -
 -
 -

103 – Comandos básicos

- 103.8 – vi
 - Editor de Textos disponível em todos os unix
 - Poderoso mas nada intuitivo.
 - Comandos Basicos
 - Inserção (i,O,o)
 - Substituição (R,r)
 - Movimentação: [jkl], w, b, :nro, GG, shift G
 - Copiar ou Recortar: [N]yy ou [N]dd
 - Salvar e Sair: ZZ, :w, :q, :wq, :x
 - Prompt de Comandos
 - ! / ? :
 - Substituição
 - 1,\$s/dope/hope/g

FIM
103