

Introdução ao Linux Aula 4

Abordagem prática para novos usuários

Fernando Borges Whitaker

Comando ps

- ▶ Saída contém PID TTY TIME COMMAND
- ▶ ps -e (Lista todos os processos)
- ▶ ps -ef (Lista todos os processos, com mais detalhes)
- ▶ ps -fu “alguem” (Lista todos os processos de alguém)

- ▶ PID: process id
- ▶ TTY: terminal device
- ▶ TIME: tempo de cpu usado até agora

Comando Para Matar

- ▶ Comando kill
- ▶ kill -SIGKILL pid (mata processo)
- ▶ kill -SIGSTOP pid (para processo)
- ▶ kill -SIGCONT pid (continua processo)
- ▶ kill -SIGSEGV pid (faz o processo morrer como se tivesse tido um erro de segmentação)
- ▶ ps -fu fulano | awk '{print \$2}' | grep -v PID
- ▶ kill -9 `ps -fu fulano | awk '{print \$2}' | grep -v PID`
- ▶ Comando **top** (digitar k para matar)
 - ▶ M para ordenar pelo uso de memória em ordem decrescente
 - ▶ u usuario

Comandos diversos

- file - detecta tipo de arquivo
- Gerenciamento
 - free - memória ram
 - df - disco
 - passwd - trocar senha

Execução em paralelo

- Rodando processos em background
 - `firefox &`
 - `gedit &`
- Rodando processos em background mesmo sem sessão
 - `nohup programa > saida.txt 2> saída.err &`

Execução de Programas

- Colocar um processo interativo em background
 - comando Ctrl+Z → bg <enter>
 - **jobs** lista programas em background
 - **fg** voltar para execução foreground

Exercícios 1

- ▶ Use o comando `ps` para saber
 - ▶ Que processos o usuário `curso` está rodando
 - ▶ Que processos todos os usuários estão rodando
 - ▶ Abra o Firefox e ache o seu PID
 - ▶ Mate o Firefox com o comando `kill`
- ▶ Abra o `gedit` via linha de comando e o coloque em background com `Control+Z bg <enter>`
 - ▶ Traga o `gedit` de novo pra foreground com `fg`
- ▶ Use o `top` para determinar que processos estão consumindo mais recursos
- ▶ Use o `man top` para entender tudo que é listado
- ▶ Mate algum processo não essencial com `top` e tecla “k”

Exercícios 2

Crie um script chamado `forever.sh` usando o vim: `vim forever.sh`
Digite o conteúdo abaixo e salve.

```
#!/bin/bash
echo "Starting the Infinite Process"
while [ 1 ]
do
    sleep 60
    echo "Inside the loop that will go forever"
done
```

Mude as permissões do arquivo para conceder permissão de EXECUÇÃO para `forever.sh`.

Rode o script com `./forever.sh` Coloque o script em background

Tente achá-lo com `ps` e mate-o com `kill`

Rode-o novamente com `nohup` e um `&` (e comercial) no final da linha. Feche o terminal e abra novamente.

O processo `forever.sh` continua rodando?

Sumário de comandos



Gerenciamento de diretórios

- ❑ ls - lista
- ❑ cd - troca diretório atual
- ❑ pwd - exibe diretório atual
- ❑ mkdir - cria diretório
- ❑ rmdir - remove diretório vazio
- ❑ rm - apaga arquivos e diretórios

Manipulação de arquivos

- ❑ cat - concatena arquivo(s) e imprime (tac - inverso)
- ❑ rm - apaga arquivos e diretórios
- ❑ touch - altera data de modificação, cria arquivo
- ❑ cp - copia arquivo
- ❑ mv - move arquivo, renomear
- ❑ head - exibir inicio de arquivo

Manipulação de arquivos

- tail - exibir final de arquivo
- **grep** - busca em arquivo
- more and less
- sort - ordenar
- **wc** - contar letras, palavras e linhas
- cut - exibir partes
- **diff** - diferença entre arquivos

Comandos diversos

- **date** - data do sistema
- **find** - procura arquivos/ diretórios
- **time** - tempo de execução
- **which** - localizador de comandos
- **hostname** - retorna o nome da máquina

Filtrando linhas de um arquivo com comando grep

- ▶ `grep "teste" arquivo` (todas as linhas que contêm a palavra teste)
- ▶ `grep "^Teste" arquivo` (todas as linhas que começam com a palavra Teste)
- ▶ `grep "teste$" arquivo` (todas as linhas que terminam com a palavra Teste)
- ▶ `grep "^[A-Z]" arquivo` (todas as linhas que começam com letras maiúsculas)
- ▶ `grep "^[A-Z]" arquivo` (todas as linhas que começam com letras maiúsculas)
- ▶ `grep "[aeiou]" arquivo` (todas as linhas que têm alguma vogal minúscula)
- ▶ `grep -i "[aeiou]" arquivo` (todas as linhas que têm alguma vogal)
- ▶ `grep -c "teste" arquivo` (quantas linhas contêm a palavra teste)

Exercícios 3

- ▶ Qual o nome da sua máquina?
- ▶ Qual a data e hora do sistema?
- ▶ Em que caminho está o comando cp?
- ▶ Tente adaptar os 10 exemplos de usos do comando find para localizar arquivos no seu home (<https://e-tinet.com/linux/comando-find-linux/>)
- ▶ Vá pra /home/curso e tente localizar todos os arquivos com extensão .txt em todos os subdiretórios usando o comando find (consulte manual)
- ▶ Tente usar o comando find com a opção exec para apagar os arquivos com extensão .so abaixo de /home/curso/fulano (você os criou na aula 2)
- ▶ Baseie-se na dica em: <https://www.vivaolinux.com.br/dica/Utilizando-find-+-exec-e-a-gilizando-tarefas>

Exercícios 4

- ▶ Quantas palavras, caracteres e linhas tem o arquivo nomes.txt?
- ▶ Use o grep para pegar apenas os nomes que começam com A
 - ▶ Dica: padrão “^A” letra A no início de uma linha
 - ▶ padrão “A” letra A em qualquer posição da linha
- ▶ Use o grep para pegar apenas os nomes que NÃO começam com A. Coloque o resultado em nomes_ao_a.txt
 - ▶ Dica: opção -v (inverte o sentido do grep)
- ▶ Use o comando diff para comparar nomes.txt com nomes_ao_a.txt
- ▶ O arquivo pesquisa.txt contém resultados brutos dos votos de uma eleição com vários candidatos. Copie-o para sua área. Use uma combinação de sort, uniq e wc para dizer quantos candidatos diferentes foram votados.
- ▶ Elabore uma linha de comando para imprimir na tela o nome de cada candidato e seu número de votos em ordem decrescente de número de votos