



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Introdução a Sistemas Abertos

Comandos de Administração

A decorative graphic in the bottom right corner of the slide, consisting of a grid of light green squares. The grid is composed of 12 squares arranged in a 4x3 pattern, with the top-right square missing. The squares are semi-transparent and have rounded corners.

Objetivos

- Familiarização com os comandos do Linux
- Conhecimento e prática de comandos que informam ou manipulam informações do sistema
- Comandos de busca e filtros de arquivos
- Comandos diversos

Execução de comandos

- Para executar um comando, é necessário que se tenha permissões de execução
- Path é o caminho de procura dos arquivos/comandos executáveis
- O path é armazenado numa variável de ambiente \$PATH
 - Ex.: /usr/local/bin:/usr/bin:/bin:/usr/bin/X11
- O comando deve está no PATH do usuário para ser executado diretamente

Execução de comandos

- Para verificar o *path* atual, basta executar o comando:
 - `echo $PATH`
- Para executar um comando que não esteja no path, pode-se informar o caminho completo
 - `/usr/local/bin/Comando`

Comandos de gerência e monitoramento sistema

- Gerenciar sistemas requer constante monitoramento dos recursos e serviços
- O sistema operacional deve fornecer mecanismos que possibilitem o gerenciamento
- Constatada situação inadequada, o administrador deve intervir

Comandos de gerência e monitoramento sistema

■ df

- Mostra o espaço livre/ocupado de cada partição

```
sor@labsor:~$ df -h
Sist. Arq.          Tam    Usad  Disp  Uso%  Montado em
/dev/hda2          2,8G   693M   2,0G   27%   /
tmpfs              126M    0    126M    0%   /lib/init/rw
udev               10M    616K   9,4M    7%   /dev
tmpfs              126M    0    126M    0%   /dev/shm
sor@labsor:~$
```

Comandos de gerência e monitoramento sistema

■ du

- Mostra o espaço ocupado por arquivos e sub-diretórios no diretório especificado

```
sor@labsor:~$ du -h /home/sor/  
4,0K    /home/sor/dir_001  
4,0K    /home/sor/dir_002  
32K     /home/sor/  
sor@labsor:~$ _
```

Comandos de gerência e monitoramento sistema

■ free

- Mostra detalhes sobre a utilização da memória RAM do sistema.

```
lor@labsor:~$ free -m
              total          used          free          shared    buffers         cached
Mem:           250           29          221             0             2            15
-/+ buffers/cache:           11          239
Swap:           243             0          243
lor@labsor:~$ _
```

Comandos de gerência e monitoramento sistema

■ date

- Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário root para modificar a data e hora

```
root@labnor:~$ date  
Dom Abr 11 19:10:58 BRT 2010  
root@labnor:~$ _
```

Comandos de gerência e monitoramento sistema

- Alterando data do sistema com date:
 - Por padrão os parametros são:
MMDDhhmm

```
labsor:~# date
Ter Jul 20 12:13:08 BRT 2010
labsor:~# date 08231113
Seg Ago 23 11:13:00 BRT 2010
labsor:~# _
```



Execução de comandos

- Todos os programas quando são executados iniciam um processo. Todo processo tem um número de identificação (PID). É através do PID que o S.O. identifica os processos e pode manipulá-los



Execução de comandos

- Os comandos
 - ps
 - top

- Fornecem informações sobre os processos em execução, para que seja tomada alguma medida, como “matar” um processo.

Comandos de gerência e monitoramento sistema

■ ps

- Mostra os programas (processos) que estão sendo executados e informações de caminho e de quem executou
- ps aux (exemplo)

```
labsor:~# ps aux
USER          PID  %CPU  %MEM    VSZ   RSS TTY      STAT   START       TIME COMMAND
root           1   1.1   0.2   2100    692 ?        Ss     22:16       0:00 init [2]
root           2   0.0   0.0     0     0 ?        S<     22:16       0:00 [kthreadd]
root           3   0.0   0.0     0     0 ?        S<     22:16       0:00 [migration/0]
root           4   0.0   0.0     0     0 ?        S<     22:16       0:00 [ksoftirqd/0]
root           5   0.0   0.0     0     0 ?        S<     22:16       0:00 [watchdog/0]
root           6   0.1   0.0     0     0 ?        S<     22:16       0:00 [events/0]
root           7   0.0   0.0     0     0 ?        S<     22:16       0:00 [khelper]
```

Comandos de gerência e monitoramento sistema

■ top

- Mostra os programas em execução ativos, parados, tempo usado na CPU, detalhes sobre o uso da memória RAM, Swap, disponibilidade para execução de programas no sistema, etc
- Para sair do top pressione a tecla “q”.

```
top - 22:20:36 up 3 min, 1 user, load average: 0.03, 0.10, 0.05
Tasks: 39 total, 1 running, 38 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 1.0%sy, 0.0%ni, 99.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 256404k total, 29108k used, 227296k free, 2812k buffers
Swap: 248968k total, 0k used, 248968k free, 15948k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2080	root	20	0	2388	1088	884	R	1.3	0.4	0:00.20	top
1	root	20	0	2100	692	588	S	0.0	0.3	0:00.92	init
2	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0

Finalização de Processos

- Os processos que estão em execução podem ser finalizados com os comandos
 - `kill [opções] [sinal] [pid]`
 - `killall [opções] [sinal] [nome do processo]`
- Exemplo:
 - `Kill -9 532` (Mata o processo com PID=532)
 - `Killall -9 httpd` (Mata todos os processos com nome httpd)

Sinais do Sistema

Sinal	Valor	Ação	Comentário
HUP	1	A	Travamento detectado no terminal de controle ou finalização do processo controlado
INT	2	A	Interrupção através do teclado
QUIT	3	C	Sair através do teclado
ILL	4	C	Instrução Ilegal
ABRT	6	C	Sinal de abortar enviado pela função abort
FPE	8	C	Exceção de ponto Flutuante
KILL	9	AEF	Sinal de destruição do processo
SEGV	11	C	Referência Inválida de memória
PIPE	13	A	Pipe Quebrado: escreveu para o pipe sem leitores
ALRM	14	A	Sinal do Temporizador da chamada do sistema alarm
TERM	15	A	Sinal de Término

Execução de comandos

- Tipos de Execução de comandos
- Primeiro Plano (foreground)
 - É a execução normal, ou seja, um novo programa só poderá ser executado quando o anterior for concluído
- Segundo Plano (background)
 - Quando você não precisa esperar o término da execução de um programa para executar um novo comando



Execução de comandos

- Manipulando processos:
- Para parar um programa em execução em primeiro plano, basta um ctrl+z. O comando jobs informa os programas que estão em segundo plano, ou parados ou em execução.



Execução de comandos

- Os comandos servem para manipular processos
 - `bg` – Coloca processos em execução em segundo plano
 - `fg` – Coloca processos em primeiro plano
 - `Jobs` – Mostra a lista de processos em segundo plano



Execução de comandos

- Prática
- Colocar processos em segundo plano
 - Matar processos parado
 - Matar processos rodando em primeiro e em segundo plano.

Execução de comandos

- Prática:
 - Execute o comando `tail -f /var/log/messages`
 - Digite `Ctrl+Z` (Observe)
 - Digite `jobs`
 - Digite `bg 1` (Observe)
 - Digite `jobs`
 - Digite `fg 1`
- Coloque mais de um processo em segundo plano para fixar melhor

Comandos de filtragem de saída de arquivo

- grep

- Procura por um texto dentro de um arquivo ou no dispositivo de entrada padrão.

```
labsor:~# grep www /etc/group  
www-data:x:33:
```

Comandos de filtragem de saída de arquivo

- sort
 - Organiza (Ordena) as linhas de um arquivo texto ou da entrada padrão.

Ordem do arquivo

```
labsor:~# cat texto.txt
maria
pedro
joao
amelia
barbara
erika
labsor:~# _
```

Saída ordenada

```
labsor:~# sort texto.txt
amelia
barbara
erika
joao
maria
pedro
labsor:~# _
```

Comandos de filtragem de saída de arquivo

■ WC

- Conta o número de palavras, bytes e linhas em um arquivo ou entrada padrão. Se as opções forem omitidas, o wc mostra a quantidade de linhas, palavras, e bytes.

```
labsor:~# cat texto.txt
maria
pedro
joao antonio
amelia
barbara
erika
labsor:~# wc texto.txt
 6  7 46 texto.txt
labsor:~# _
```

- Exemplos de uso:

- wc -l, wc -w, wc -c, wc -m

Redirecionamento de Saída

>

Redireciona a saída de um comando para um arquivo especificado, criando caso não exista ou destruindo seu conteúdo anterior.

>>

Redireciona a saída de um comando para um arquivo especificado, anexando ao seu fim. Caso este arquivo não exista, será criado.

2>

Redireciona os erros gerados por um comando para o arquivo especificado. Mesmo que não ocorra erro na execução do comando, o arquivo será criado.

Redirecionamento Especial

|

O pipe serve para direcionar a saída de um comando para a entrada de outro. É muito útil!

ex:

```
$ ps -aux |grep nomeprocesso
```



- Modo de Inserção
- Modo Normal
- Comandos Básicos:
 - ESC - Ir para o modo normal
 - i, a ou o- Ir para o modo de inserção de texto
 - :q! - Sair sem salvar
 - :x - sair salvando
 - :w - gravar o arquivo
 - / <texto> - buscar um texto em um arquivo
 - G - ir para o final do arquivo
 - g - ir para o início do arquivo
 - dd - Apaga linha
 - yy - copia linha
 - p - cola linha
- Consultar mais comandos <http://aurelio.net/curso/material/vim-ref.html>

Comandos diversos

apropos

Para quando se deseja executar uma tarefa e não se sabe o nome do comando

whatis

Serve para mostrar uma breve descrição do comando.

whereis

Localiza o arquivo que contém uma página de manual. A pesquisa é feita usando-se os caminhos de páginas de manuais configuradas no sistema

Pratica Comandos diversos

- Descobrir como funcionam os comandos **seq**, **time** e **nl**

