

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA
Campus Salvador

Usando o terminal -- Linux --

Flávia Maristela (flavia@flaviamaristela.com)



Os comandos que já vimos

- man
- info
- cal
- date
- uname
- clear
- nano
- pico
- clear
- exit
- pwd
- ls
- cd
- locate
- mkdir
- rmdir
- cp
- mv
- rm
- cat
- tree
- file
- reboot
- shutdown
- more
- less
- find
- df
- id
- who
- whoami
- touch
- head
- tail
- diff
- sort
- uniq
- grep
- paste
- du
- time
- uptime
- dmesg

O que veremos hoje

- Expressão Regular
- Curingas
- Outros comandos
 - Gerência de Usuários
 - Gerência de Grupos
 - Permissões
- Gerência de Processos
 - Estados
 - Transições

Expressão Regular

- Forma geral de listar uma cadeia de caracteres sem listar todos os elementos do conjunto
- Operações:
 - Alternância: |
 - Agrupamento: ()
 - Quantificação: ?, +, *
 - Ex.: flavia, flavio → flavi(a|o)

Expressão Regular

■ Quantificação

- ?: 0 ou 1 ocorrência do elemento precedente
 - cl?aro → claro, caro
- +: 1 ou mais ocorrência do elemento precedente
 - ab+a → aba, abba, abbba, ...
- *: 0 ou mais ocorrência do elemento precedente
 - ab*a → aa, aba, abba, abbba, ...

Curingas

- Especificam um ou mais arquivos ou diretórios do sistema de uma só vez.
- *: faz referência a um nome completo/restante de um arquivo/diretório.
- ?: faz referência a uma **letra** naquela posição.
- [padrão]: faz referência a uma faixa de caracteres de um arquivo/diretório.
 - [a-z][0-9]: caracteres de a à z seguido de um caracter de 0 à 9.
 - [a,z][1,0]: caracteres a e z seguido de um caracter 1 ou 0 naquela posição.
 - [a-z,1,0]: intervalo de caracteres de a à z ou 1 ou 0 naquela posição.
 - ^: identifica qualquer caracter exceto o da expressão.
 - Exemplo: [^tes] faz referência a qualquer caracter exceto t, e e s.
- {padrões} : expande e gera strings para pesquisa de padrões de um arquivo/diretório.
 - X{ab,01}: referencia a seqüencia Xab ou X01
 - X{a-z,10}: referencia a seqüencia de caracteres Xa-z e X10.

Outros comandos

- **mount**: monta um sistema de arquivos numa partição
 - Para que? O linux não acessa os dados de uma partição diretamente
 - Sintaxe: **mount -t** [fs] [particao] [ponto de montagem]
 - Exemplo: `mount /dev/sda1 /mnt/sda1`
- **umount**: desmonta um sistema de arquivos de uma partição
 - Sintaxe: **umount** [ponto de montagem]
 - Exemplo: `umount /mnt/sda1`
- **tar**: compacta/descompacta arquivos
 - Sintaxe: `tar [opcoes] [pacote de saida]`
 - `tar -c` : compactar
 - `tar -x`: descompactar

Atualização de pacotes do Debian

- `apt-get update`
- `apt-get install <pacote>`
- `apt-get remove <pacote>`
- `apt-get upgrade`
- `apt-get dist-upgrade`

Links no Linux

- *Links* são referências para arquivos ou diretórios em outra localização;
- Podem ser “simbólicos” ou “hard links”
 - Simbólicos:
 - funcionam como atalhos simples
 - Se o arquivo for movido, o link deixa de funcionar
 - Exemplo: `ln -s arquivo1.txt flavia.txt`
 - Hard Links:
 - Estão fortemente ligados aos arquivos
 - Se o arquivo mudar de localização, o link continua funcionando
 - Exemplo: `ln /home/aluno/arquivo1.txt arquivo2.txt`

Criação de usuários

- `adduser`: comando para a criação de usuários
 - Sintaxe: `adduser [usuario]`
- `userdel`: comando para exclusão de usuários
 - Sintaxe: `userdel [usuario]`
- `passwd`: comando para alterar a senha de um usuário
 - Sintaxe: `passwd [usuario]`

Criação de Grupos

- `addgroup`: comando para a criação de novos grupos de usuários
 - Sintaxe: `addgroup [group]`
- `newgrp`: altera a identificação de grupo do usuário.
 - Sintaxe: `newgrp [grupo]`
- `groupdel`: exclui um grupo
 - Sintaxe: `groupdel [grupo]`
- `users`: apresenta todos os usuários logados no sistema
 - Sintaxe: `users`
- `groups`: apresenta todos os grupos de usuário do sistema
 - Sintaxe: `groups`

Gerenciando arquivos

- Atributo dos arquivos que são gerenciados pelo Linux
 - Nome
 - Localização (no disco)
 - Tamanho (em bytes)
 - Ligações (nomes pelos quais os arquivos são conhecidos)
 - propriedade (usuário que é dono do arquivo)
 - Grupo (grupo de usuários que acessa o arquivo)
 - Tipo (tipo do arquivo)
 - Criação (data e hora da criação do arquivo)
 - Modificação (data e hora da modificação do arquivo)
 - Acesso (permissões de acesso ao arquivo)

Permissões de arquivo

- Informações sobre os arquivos → ls -l

—

	<u>permissões</u>	<u>nº links</u>	<u>Dono arquivo</u>	<u>grupo</u>	<u>tamanho bytes</u>	<u>data</u>	<u>arquivo / pasta</u>
	d rwx r-x r-x	2	aluno	aluno	4096	Mai 19 11:57	Desktop
	<u> </u> <u> </u> <u> </u>						
	dono grupo outros						
	└─┬─┘						
	Tipo de arquivo						

— Tipo do arquivo:

- - arquivo
- d: diretório
- l: link simbólico
- s, b, p, c: arquivos especiais

— Dono (owner) – proprietário do arquivo

— Grupo (group) – grupo ao qual o dono pertence

— Outros (others): usuários que não fazem parte do grupo

Permissões de arquivo

- **chown**: para mudar o dono de um arquivo
 - Sintaxe: **chown** novo dono [: novo grupo] arquivo
- **chgrp**: para mudar o dono do grupo
 - Sintaxe: **chgrp** novo grupo arquivo
- **chmod**: para mudar as permissões dos arquivos
 - Sintaxe: **chmod** ugo + rwx arquivo
 - Exemplo **chmod 764 arquivo1.txt**
 - 7: 111 → r w x
 - 6: 110 → r w -
 - 4: 100 → r - -

Outros comandos

- **umask**: (user mask) define as permissões iniciais de um arquivo

- Sintaxe: **umask** [permissao]

```
-----  
|          | ARQUIVO | DIRETÓRIO |  
| UMASK |-----|          |  
|          | Binário | Texto    | |
|---|---|---|---|
| 0      | r-x   | rw-   | rwx   |  
| 1      | r--   | rw-   | rw-   |  
| 2      | r-x   | r--   | r-x   |  
| 3      | r--   | r--   | r--   |  
| 4      | --x   | -w-   | -wx   |  
| 5      | ---   | -w-   | -w-   |  
| 6      | --x   | ---   | --x   |  
| 7      | ---   | ---   | ---   |  
-----
```

- Exemplo: **umask** 102 touch teste.txt

Outros comandos

- | (pipe): faz com que a saída de um comando passe a ser entrada de outro comando
 - Ex: `cat /etc/passwd | uniq`
- grep: identifica padrões num arquivo
 - Ex: `grep flavia /home;arquivo1.txt`
- cut: separa partes de um arquivo de acordo com algum separador
 - Ex: `cut -d “,” -f 1 /home/aluno/arquivo1.txt`

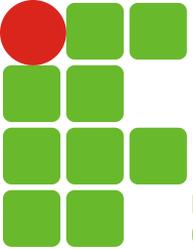
-
- **tee**: envia o resultado de um programa para a tela e para um arquivo ao mesmo tempo.
 - Sintaxe: comando | tee [arquivo]
 - Exemplo: ls -la | tee arquivo.txt
 - **wc**: conta o número de palavras, bytes e linhas em um arquivo ou entrada padrão.
 - Sintaxe: **wc** [opções] [arquivo]

Outros comandos

- **finger**: mostra detalhes sobre os usuários de um sistema.
 - Algumas versões do finger possuem bugs
 - Sintaxe: **finger** [opcao] [usuario]

Os comandos que já vimos

- man
- info
- cal
- date
- uname
- clear
- nano
- pico
- clear
- exit
- pwd
- ls
- cd
- locate
- mkdir
- rmdir
- cp
- mv
- rm
- cat
- tree
- file
- reboot
- shutdown
- more
- less
- find
- df
- id
- who
- whoami
- touch
- head
- tail
- diff
- sort
- uniq
- grep
- paste
- du
- time
- uptime
- dmesg
- mount
- umount
- tar
- apt-get
- ln
- adduser
- userdel
- passwd
- addgroup
- newgrp
- groupdel
- users
- groups
- chown
- chmod
- chgrp
- cut
- umask
- tee
- wc
- finger



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA
Campus Salvador

Gerência de Processos -- Linux --

Flávia Maristela (flavia@flaviamaristela.com)

Reapitulando...

- Processos representam programas em execução;
- Processos são organizados hierarquicamente;
- Processos podem criar ou chamar outros processos

Windows vs. Linux

- No Windows:
 - Processos realizam chamadas de sistema via *dll*
- No Linux:
 - Processos independentes estão preparados para executar ao mesmo tempo
 - Porque isto não acontece no Windows?

Atributos de um processo

- Identificador
- Proprietário
 - Identificado a partir de um uid e de um gid
- Estado
- Prioridade

Atributos de um processo

- Identificador (pid)
 - Numero inteiro que identifica unicamente um processo
 - Gerenciado pelo *kernel*
 - Processos instanciados possuem *ppid* (identificador do processo pai)

Atributos de um processo

- Proprietário (uid)
 - Numero inteiro que identifica o dono do processo
 - Como usuário sempre pertence a um grupo, cada processo também tem um **gid** (identificador do grupo)

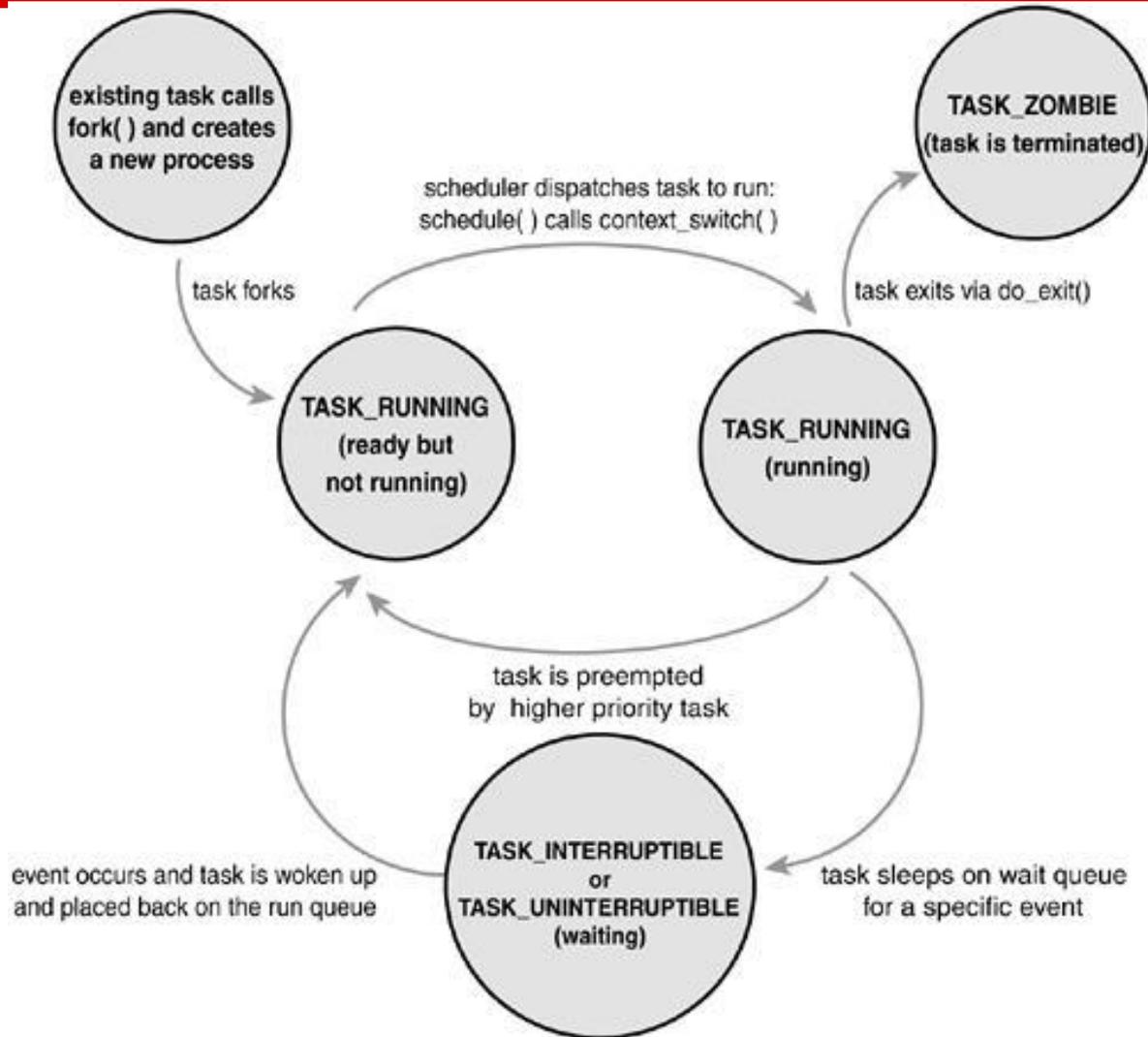
Falando nisso...

- O que é o usuário *root*?
 - Usuário que tem acesso irrestrito ao sistema
 - Desnecessário dizer que é um usuário extremamente perigoso
 - Por padrão, o Linux associa arquivos de configuração apenas para o usuário “*root*”
 - É possível para um usuário qualquer se tornar “*root*”?
 - Comando: **su** (substitute user)
 - Sintaxe: **su** [usuario]

Acessando o sistema como root

- Comando: `sudo`
 - Converte o usuário corrente em usuário root
 - Determina comandos específicos que podem ser feitos com o usuário
 - Arquivo: *sudoers*

Estados do Linux



Atributos do processo

(-- Estado --)

- Criação de processo:
 - fork(): criação de um processo filho
 - exec(): substitui a área de código do processo filho
- Em execução:
 - **TASK_RUNNING**: executando ou esperando para ser executado
 - Não existem dois estados: “pronto” e “em execução”
 - Lista única com apontador para processo em execução

Atributos do Processo

- **TASK_INITERUPTIBLE**: bloqueado, aguardando que determinada condição seja satisfeita
 - Final da operação de I/O
 - Liberação do recurso de sincronização
 - Liberação de recurso bloqueado por interrupção de software emitida por outro processo
- **TASK_UNINTERRUPTIBLE**: bloqueado, porém neste caso aguardando a condição crítica associada a hardware

Atributos do Processo

- **TASK_STOPPED**: processo fica parado por conta de uma interrupção de software, emitida por outro processo
 - Interrupções de software (signal)
 - STOP
 - CONT
 - TERM
 - KILL
- **TASK_ZOMBIE**: estado que o processo filho assume enquanto o processo pai identifica que ele finalizou.

Visualizando processos

- Finalização
 - `exit()`
 - `kill()`

- Como visualizar processos?
 - Comandos:
 - `ps` (estático)
 - `top` (dinâmico)

Comandos para gerência de processos

- **ps**: lista os processos em execução
- **top**: lista os processos em execução
- **pstree**: mostra processos relacionados em formato de árvore.
 - *Sintaxe: pstree -opção PID*
- **kill**: finaliza um processo indicado como parâmetro
 - *Sintaxe: kill [pid]*

Comandos para gerência de processos

- **bg**: coloca um processo que está em foreground em background
 - Sintaxe: `bg [pid]`
- **fg**: coloca um processo que está em background em foreground
 - Sintaxe: `fg [pid]`
- **jobs**: lista os jobs que estão executando
 - Sintaxe: `jobs [pid]`

-
- No Linux os processos são divididos em três grandes grupos
 - Interativos: tempo de resposta médio
 - Batch: vazão
 - Tempo Real: cumprimento de prazos
 - Processos podem ainda ser divididos em I/O-Bound e CPU-Bound

-
- Não identifica processos interativos e batch;
 - Distingue apenas processos de tempo real
 - Privilegia processos I/O-Bound para oferecer melhor tempo de resposta para as aplicações interativas
 - Escalonador é time-sharing
 - Quantum
 - Preempção

-
- O escalonador do Linux possui um algoritmo que divide o tempo de processamento em épocas
 - Prioridade dinâmica: ajuste
 - Prioridade Estática: processos de tempo real
 - Prioridade Dinâmica: processos batch e interativos
 - Como é calculada a prioridade?
 - Quantidade de tempo restante no quantum

Gerência de Memória

- Projetos de sistemas operacionais focam no uso de maior quantidade de memória física
- Linux não explora segmentação:
 - Gerência de paginação é mais simples
 - Nem sempre o hardware do processador dá suporte a segmentação
 - É possível converter segmentação em paginação:
 - Todo endereço virtual é mapeado num único segmento