



# LINUX

AULA 2



# MANIPULANDO ARQUIVOS

# AUTOCOMPLETAR

- O bash oferece a funcionalidade de sugerir os nomes dos arquivos ou diretórios sem que tenhamos que digitá-los.
- Basta digitar o prefixo e pressionar a tecla TAB.
- Se mais de um arquivo possuir o mesmo prefixo, o shell exibe a lista de todos eles.

## Exemplo

```
[gauss@physic.edu gauss]$ user[TAB]
useradd      userdel      usermod      users
[gauss@physic.edu gauss]$ user
```

# SÍMBOLOS ESPECIAIS

- O bash oferece alguns símbolos especiais para facilitar a manipulação de arquivos ou diretórios:

? : Coringa que substitui 1 caractere

\* : Coringa que substitui 1 ou mais caracteres.

[abcd] : Define um grupo de caracteres.

[a-z] : Define um intervalo de caracteres.

! : Negação

## Exemplo 1

```
cp pub/* backup/
```

Copia todos os arquivos do diretório pub para o diretório backup

## Exemplo 2

```
ls [a-f]*.txt
```

Lista todos os arquivos que começam com letras de “a” a “f” com a extensão “txt”

# NOMES DE ARQUIVOS

- O sistema Linux é sensível a maiúsculas e minúsculas;
- Podemos, mas não devemos usar símbolos especiais como espaço, ~, \*, &, ^, \$ ou ?.
- Para manipularmos símbolos especiais em arquivos no shell utilize:
  - 'abc def.txt'
- Os arquivos não precisam ter extensão.



# EXIBINDO MENSAGENS (ECHO)

- Exibe mensagens na tela.

## Sintaxe

```
echo <mensagem>
```

## Exemplo

```
[gauss@physic.edu pub]$ echo "Fatec"
```

```
Fatec
```

```
[gauss@physic.edu pub]$
```

# COPIANDO ARQUIVOS (CP)

- O comando cp (copy) copia arquivos ou diretórios.

## Sintaxe

```
cp [opcoes] <origem> <destino>  
cp [opcoes] <origem...> <diretório destino>
```

## Exemplo

```
[gauss@physic.edu pub]$ cp net/wi-fi.pdf docs/  
[gauss@physic.edu pub]$ cp firewire.txt ieee/1394.txt  
[gauss@physic.edu pub]$ cp globus.pdf secure.pdf papers
```

# MOVENDO E RENOMEANDO ARQUIVOS (MV)

- Para mover ou renomear arquivos ou diretórios utilize o comando mv (move).

## Sintaxe

```
mv [opções] <origem> <destino>  
mv [opções] <origem...> <diretório destino>
```

## Exemplo

```
[gauss@physic.edu pub]$ mv docs/wi-fi.pdf net/  
[gauss@physic.edu pub]$ mv *.tmp lixo/  
[gauss@physic.edu pub]$ mv ieee/ docs/
```



# EXCLUINDO ARQUIVOS (RM)

- Para excluirmos arquivos utilizamos o comando rm (remove).

## Sintaxe

```
rm [opções] <arquivo...>
```

## Exemplo

```
[gauss@physic.edu pub]$ rm globus.pdf
```

```
[gauss@physic.edu pub]$ rm *.txt
```

```
[gauss@physic.edu pub]$ rm -r beta
```

# CRIANDO ARQUIVOS (TOUCH)

- O comando touch altera a data de último acesso e de última modificação de um arquivo.
- Se o arquivo não existe, o comando irá criá-lo.

## Sintaxe

```
touch [opções] <arquivo...>
```

## Exemplo

```
[newton@physic.edu pub]$ touch arquivo1 arquivo2
```

```
[newton@physic.edu pub]$ touch -t 0503011200 relatorio
```

# LISTANDO O CONTEÚDO (CAT)

- Para exibir o conteúdo de um arquivo texto utilize o comando `cat`.

## Sintaxe

```
cat [opcoes] [arquivo...]
```

## Exemplo

```
[newton@physic.edu pub]$ cat docs/genios.txt  
Newton, Issac 1642-1727 Inglatterra  
Einstein, Albert 1879-1955 Alemanha  
Gauss, Friedrich 1777-1855 Alemanha  
[newton@physic.edu pub]$
```

# LISTANDO O CONTEÚDO (MORE)

- Ao contrário do cat, o comando more permite pagnar a saída, exibindo o conteúdo conforme é solicitado.

## Sintaxe

```
more [opcoes] [arquivo...]
```

## Exemplo

```
[newton@physic.edu pub]$ more docs/genios.txt  
Newton, Issac 1642-1727 Inglatterra  
Einstein, Albert 1879-1955 Alemanha  
Gauss, Friedrich 1777-1855 Alemanha  
--More--(78%)
```

# LISTANDO O CONTEÚDO (LESS)

- O comando `less` faz a paginação, permitindo controlarmos a exibição do arquivo para cima ou para baixo.

## Sintaxe

```
less [opcoes] [arquivo...]
```

## Exemplo

```
[newton@physic.edu pub]$ less docs/genios.txt  
Newton, Issac 1642-1727 Inglatterra  
Einstein, Albert 1879-1955 Alemanha  
Gauss, Friedrich 1777-1855 Alemanha  
:
```

# LISTANDO LINHAS INICIAIS (HEAD)

- Utilize o comando head para exibir apenas as linhas iniciais de um arquivo.

## Sintaxe

```
head [opcoes] [arquivo...]
```

## Exemplo

```
[newton@physic.edu pub]$ head -n 2 docs/guitarristas.txt  
Vai, Steve      1879-1955  for_the_love_of_god.tab  
Satriani, Joe   1777-1855  flying_in_a_blue_dream.tab  
[newton@physic.edu pub]$
```



# LISTANDO LINHAS FINAIS (TAIL)

- Utilize o comando tail para exibir apenas as linhas finais de um arquivo.

## Sintaxe

```
tail [opcoes] [arquivo...]
```

## Exemplo

```
[newton@physic.edu pub]$ tail -n 2 docs/genios.txt  
Einstein, Albert 1879-1955 Alemanha  
Gauss, Friedrich 1777-1855 Alemanha  
[newton@physic.edu pub]$
```

# CONTANDO LINHAS (WC)

- Utilize o comando `wc` para contar as linhas, palavras e caracteres de um arquivo

## Sintaxe

```
wc [opcoes] [arquivo...]
```

## Exemplo

```
[newton@physic.edu pub]$ wc /var/log/messages  
1576 19486 149604 /var/log/messages  
[newton@physic.edu pub]$
```

# NUMERANDO LINHAS (NL)

- O comando nl (number lines) numera linhas de arquivos.

## Sintaxe

```
nl [opcoes] <arquivo>
```

## Exemplo

```
[gauss@physic.edu pub]$ nl /etc/passwd  
    1  Felipe Martins  
    2  Rogério Mello  
[gauss@physic.edu pub]$
```

# LAB

- Faça os exercícios práticos abaixo:

1. Crie um arquivo chamado `janeiro.txt` e `fevereiro.txt` no diretório `~/arquivos/docs`

2. Mova o diretório `docs` para dentro do diretório `relatorios`. A estrutura ficará:

`~/arquivos/relatorios/docs`

3. Crie um arquivo chamado **`cria_usuario.sh`** no diretório

`~/arquivos/scripts` (observe o espaço no nome do arquivo)

4. Copie o arquivo `/etc/services` para o diretório `~/arquivos/scripts`

5. Liste o conteúdo do arquivo `~/arquivos/scripts/services` com o comando `cat` e em seguida com o comando `less`.



# COMANDOS DE FILTRO

# FILTRANDO COLUNAS (CUT)

- O comando **cut** filtra colunas de um determinado arquivo texto ou saída de comando.

## Sintaxe

```
cut [opcoes] <arquivo>
```

## Exemplo

```
[gauss@physic.edu pub]$ cut -d ":" -f 1,2 /etc/passwd  
felipe:x  
olympio:x  
[gauss@physic.edu pub]$
```



# LINHAS REPETIDAS (UNIQ)

- O comando `uniq` verifica linhas repetidas em arquivos.

## Sintaxe

```
uniq [opções] <arquivo>
```

## Exemplo

```
[gauss@physic.edu pub]$ uniq -d /var/log/secure  
asdf  
[gauss@physic.edu pub]$
```

# ORDENANDO ARQUIVOS (SORT)

- O comando sort ordena linhas de arquivos alfabeticamente ou baseados em algum padrão.

## Sintaxe

```
sort [opções] <arquivo>
```

## Exemplo

```
[gauss@physic.edu pub]$ sort /etc/passwd  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
apache:x:48:48:Apache:/var/www:/sbin/nologin  
...  
[gauss@physic.edu pub]$
```

# TRADUZINDO CARACTERES (TR)

- O comando `tr` traduz qualquer caractere em um caractere especificado.

## Sintaxe

```
<entrada> | tr <origem> <destino>
```

## Exemplo

```
[gauss@physic.edu pub]$ cat /etc/passwd | tr 'a' 'b'
```

## Exemplo

```
[gauss@physic.edu pub]$ tr 'a-z' 'A-Z' < origem > destino
```

# OUTROS FORMATOS (OD)

- O comando `od` visualiza arquivos textos em outros formatos.

## Sintaxe

```
od [opções] <arquivo>
```

## Exemplo

```
[gauss@physic.edu pub]$ od -x arq1
```

# LAB

- Faça os exercícios práticos abaixo:
  1. Traduza todas as letras minúsculas do arquivo `/etc/passwd` em maiúsculas e escreva estas alterações no arquivo `/root/passwd_filtrado`
  2. Exiba apenas as colunas respectivas ao `home_dir`, `username` e `shell` do arquivo `/etc/passwd`.
  3. Ordene por caracteres alfanuméricos e ignorando case-sensitive o conteúdo do arquivo `/etc/passwd`.
  4. Visualize de forma hexadecimal o conteúdo do arquivo `/etc/shadow`.



# REDIRECIONANDO ENTRADA E SAÍDA



# REDIRECIONANDO ENTRADA E SAÍDA

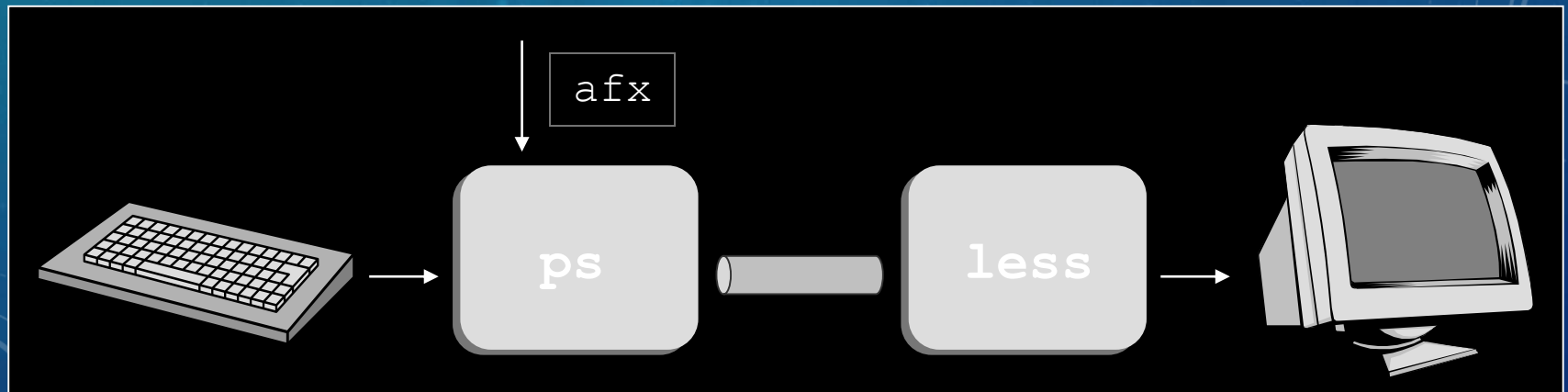
- O sistema é capaz de controlar a entrada e a saída de processos.
- Quando um processo é criado ele é associado a 3 descritores de arquivo:
  - Entrada padrão (stdin)
  - Saída padrão (stdout)
  - Saída de erro padrão (stderr)

# PIPES

- Através dos pipes conseguimos redirecionar a saída de um processo como argumento de entrada para outro processo.

## Exemplo

```
[gauss@physic.edu /]$ ps afx | less
```



# REDIRECIONANDO SAÍDA

- A saída de um processo também pode ser redirecionada para um arquivo.
- Para criar um arquivo com a informação que seria exibida na tela utilize o operador “>;

## Exemplo

```
[gauss@physic.edu /]$ ps afx > processos.txt
```

- Para preservar a informação armazenada no arquivo, utilize o operador “>>”

## Exemplo

```
[gauss@physic.edu /]$ w >> usuarios_logados
```

# REDIRECIONANDO ENTRADA

- No shell é possível alterar a entrada de um processo do teclado para um arquivo.
- Utilize o operador “<” e o nome do arquivo.

## Exemplo

```
[gauss@physic.edu /]$ gunzip < listagem.gz
```

O comando `gunzip` quando lê um dado da entrada padrão, o descompacta e retorna o resultado na saída padrão. Se o arquivo fosse passado como parâmetro, o comando iria descompactá-lo e não exibiria nada na tela.

# LABS

- Faça os exercícios práticos abaixo:
  1. Armazene as 20 últimas linhas do arquivo `/etc/services` no arquivo `~/arquivos/services_final.txt`
  2. Armazene em um arquivo chamado `/root/arquivos/nologin.txt` todos os usuários cujo shell é `/sbin/nologin`.





AJUDA ONLINE



# HELP

- A maioria dos comandos do Linux possuem uma ajuda rápida.
- A ajuda pode ser acessada através do parâmetro “--help”.

## Exemplo

```
[newton@physic.edu pub]$ cp --help
Usage: cp [OPTION]... SOURCE DEST
      or:  cp [OPTION]... SOURCE... DIRECTORY
      or:  cp [OPTION]... --target-directory=DIRECTORY
SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
...
```

# MANUAIS

- Os manuais compõem uma das fontes mais completas de informação sobre o sistema.
- Através deles obtemos informações sobre comandos, arquivos de configuração, chamadas ao sistema etc.

## Sintaxe

```
man [seção] assunto
```

# COMANDO INFO

- O comando info representa o sistema de documentação do projeto GNU.
- Organiza os documentos em formato hipertexto.

## Sintaxe

```
info comando
```

# APROPOS E WHATIS

- Para facilitar as buscas por alguma documentação, utilize o comando `whatis` ou `apropos`.
- Os comandos fazem busca em uma base de dados pela palavra procurada.

## Sintaxe

```
whatis palavra_chave
```

# DOCUMENTOS HOWTO

- Os documentos howto são documentos objetivos que visam elucidar a execução de uma determinada tarefa
- Muitos deles fazem parte do projeto The Linux Document Project (tldp)
- São encontrados no endereço:  
<http://www.tldp.org>