



LINUX

AULA 4



O EDITOR DE TEXTO VI

CONHECENDO O VI

- O vi é o editor de texto mais usado por por usuários Unix.
- O editor possui funcionalidades poderosas, porém usuários iniciantes costumam achá-lo complexo e complicado.
- O vi está presente em 100% das distribuições Linux.

EXECUTANDO O VI

- O vi deve ser aberto a partir do comando vi ou vim.
- Pode-se informar qual arquivo o vi deseja abrir.

Sintaxe

```
vi [opções] [arquivo...]
```

Exemplo

```
[kepler@physic.edu pub]$ vi  
[kepler@physic.edu pub]$ vi grid.txt
```

MODOS DO VI

- O vi possui 2 modos de funcionamento:
- Modo edição
 - O que o usuário digita se reflete em texto para ser incluído no arquivo.
- Modo comando
 - A maior parte das teclas são comandos.
 - Permite apagar, copiar, acessar funções etc.
 - Quando iniciado ele está neste modo.

ALTERNANDO ENTRE MODOS

- Tecla ESC
 - Modo edição > modo comando
- Tecla “i” (Insert)
 - Modo comando > modo edição
- Tecla “:”
 - Modo comando > modo linha

MOVIMENTANDO O CURSOR

- Para mover o cursor você pode usar as setas e as teclas pageUp, pageDown, home e end.
- **gg** - Vai para o início do arquivo
- **G** - Vai para o final do arquivo
- **:n** - Vai para a linha n.
- **w** - Vai para a próxima palavra
- **b** - Vai para a palavra anterior

MANIPULANDO ARQUIVOS

- **:w** - Grava o arquivo.
- **:q** - Sai do vi.
- **:q!** - Sai sem salvar
- **:w <nome>** - Grava com outro nome.
- **:wq** - Sai e salva.
- **:x** - O mesmo que **:wq**
- **:e <nome>** - Abre outro arquivo
- **:n** - Vai para o próximo arquivo na lista
- **:p** - Vai para o arquivo anterior na lista

COPIANDO E COLANDO

- **dd** - Recorta uma linha.
- **ndd** - Recorta n linhas. Mesmo que **dnd**
- **yy** - Copia uma linha.
- **nyy** - Copia n linhas.
- **p** - Cola na linha abaixo do cursor.
- **P** - Cola na linha acima ao cursor.

BUSCAS E SUBSTITUIÇÃO

- **/palavra** - Realiza a busca por “palavra”.
- **n** - Posiciona o cursor na próxima ocorrência.
- **N** - Posiciona o cursor na ocorrência anterior.

- Substituindo:

:%s/busca/texto/flags

flags:

- **g** - Substitui todas ocorrências em uma linha.
- **c** - Solicita confirmação.

FUNÇÕES DIVERSAS

- **u** : desfazer a última ação.
- **nu** : desfazer as n últimas ações.
- **U** : Desfaz todas as ações.
- **^r** : Refaz a última ação (Ctrl+r).
- **^g** : Exibe status da página (Ctrl+g).
- **J** : Une uma linha a outra.

LAB

- **1.** Crie um novo arquivo no vi com o nome de `~/arquivos/dados.txt`.
 - **2.** Digite em uma linha seu nome e em outra linha seu email.
 - **3.** Salve e saia do arquivo.
 - **4.** Faça uma cópia do arquivo `/etc/services` para o diretório `/tmp` e abra com o vi para editá-lo.
 - **5.** Insira a linha `"# arquivo de exemplo"` na última linha do arquivo e salve a alteração.
 - **6.** Copie das linhas 40 a 50 no início do arquivo.
 - **7.** Delete das linhas 100 a 120.
- Obs: As alterações no arquivo `services` têm como intuito apenas exercitar o uso do vi e não tem nenhum valor real para o sistema.

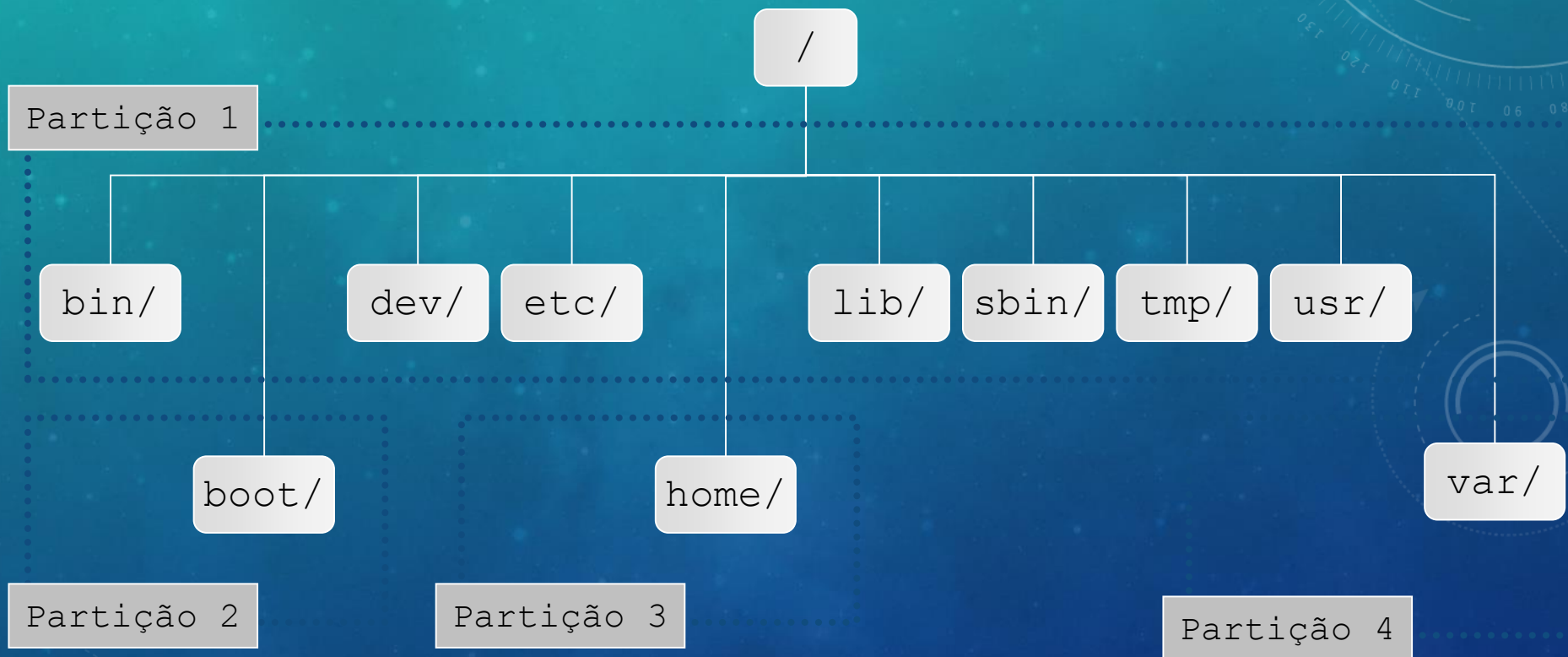


SISTEMA DE ARQUIVOS

O SISTEMA DE ARQUIVOS

- A organização dos dados de um sistema Unix é feita na forma de uma árvore;
- Associados a esta hierarquia, encontram-se:
 - Arquivos;
 - Diretórios;
 - Links;
 - Dispositivos; e outros elementos.
- O sistema de arquivos define o controle de acesso aos dados e programas.

ÁRVORE DO SISTEMA



ÁRVORE DO SISTEMA

- O sistema de arquivos é apresentado como uma única hierarquia que inicia no diretório “/”. Este diretório também é conhecido por raiz.
- A lista de diretórios que deve ser percorrida para localizar um arquivo em particular é chamada de “caminho” (*path*).
- Os diretórios seguem uma estrutura que, apesar de não ser 100% compatível com todos os Linux, têm objetivos bem definidos.

/bin : Comandos essenciais (cp, ls, rm etc).

/boot : Arquivos usados no carregamento do sistema (boot).

/dev : Arquivos especiais que são os caminhos para acessar algum dispositivos.

/etc : Arquivos de configuração. O sistema será configurado quase em sua totalidade através de arquivos localizados neste diretório.

/home : Diretório onde serão guardados os arquivos dos usuários.

ÁRVORE DO SISTEMA

- /lib** : Bibliotecas compartilhadas e módulos do kernel.
- /sbin** : Comandos essenciais do usuário administrador (fdisk, mount, ifconfig etc).
- /usr** : Uma segunda hierarquia. Reside a maioria dos programas gráficos, as bibliotecas, serviços de rede, bibliotecas e arquivos fonte.
- /tmp** : Arquivos temporários de todos os usuários.
- /var** : Diretório variável. Nele residem arquivos de log, spool de impressão e correio, dados dos banco de dados, páginas web (pode crescer muito).
- /mnt** : Diretório padrão para montagem de dispositivos de sistemas como Hard Disks, Pen Drives, entre outros.
- /opt** : Diretório para instalação opcional de programas não oficiais da distribuição utilizada.



CONTROLE DE ACESSO A ARQUIVOS E DIRETÓRIOS

SISTEMA DE PERMISSÕES

- O sistema de segurança do Linux é baseado na definição de propriedade e permissões de acesso aos arquivos.
- O sistema foi projetado para ser simples e eficiente.
- A conta root tem permissões para acessar qualquer arquivo no sistema, independente das permissões.

PROPRIETÁRIOS

- O controle de acesso aos arquivos é definido para 3 classes de usuários:
 - **Usuário dono:** Aquele que detém o arquivo. Em geral o usuário que o criou. Somente pode existir um usuário dono para o arquivo.
 - **Grupo dono:** Um grupo de usuários que pode controlar o arquivo. Em geral, é o grupo primário do dono do arquivo.
 - **Outros:** Se um usuário não é o dono, nem pertence ao grupo dono, então ele é classificado na classe outros.

PERMISSÕES DE ACESSO

- As permissões de acesso são controladas por três propriedades:
 - r** – Leitura (read);
 - w** – Escrita (write);
 - x** – Execução (execute)
- As permissões podem ser definidas para arquivos e diretórios;
- São aplicadas de forma independente ao usuário dono, grupo dono e outros.

PERMISSÕES DE ACESSO

- As permissões são definidas por bits presentes em cada arquivo:

Usuário dono			Grupo dono			Outros		
r	w	x	r	w	x	r	w	x

Leitura

Escrita

Execução

r	w	x
2^2	2^1	2^0
4	2	1

ALTERANDO AS PERMISSÕES

- Para alterar as permissões de um arquivo ou diretório utilize o comando **chmod**

Sintaxe

```
chmod [opções] modo arquivo...
```

Exemplo

```
[galileu@physic.edu pub]$ chmod 644 todo.txt  
[galileu@physic.edu pub]$ ls -l todo.txt  
-rw-r--r--  1 ricardo ricardo 0 Mar 15 15:51 todo.txt
```

Permissões 644

ALTERANDO O DONO/GRUPO

- Para alterar o usuário dono ou grupo de um arquivo utilize o comando `chown`.
- Usuários são capazes de alterar apenas o grupo de um arquivo, desde que também pertençam a ele.

Sintaxe

```
chown [opções] [dono][.[grupo]] arquivo...
```

Exemplo

```
[root@physic.edu pub]# chown galileu.fisicos estrela.doc  
[root@physic.edu pub]# chown newton. gravita.doc
```

ALTERANDO O GRUPO

- Para alterar apenas o grupo utilize o comando **chgrp**.

Sintaxe

```
chgrp [opções] grupo arquivo...
```

Exemplo

```
[galileu@physic.edu pub]$ chgrp fisicos *.txt
```

LAB

- **1.** Defina permissões `rw-r-----` para o arquivo `/tmp/services`
- **2.** Crie os arquivos `janeiro.txt` e `fevereiro.txt` dentro do diretório `~/arquivos/scripts/docs`
- **3.** Como root mude as permissões dos arquivos `janeiro.txt` e `fevereiro.txt` dentro do diretório para `rw-rw----`.
- **4.** Altere o grupo de todos os arquivos e diretórios do diretório `~/arquivos/` para `users`.
- **5.** Dê permissões de escrita para o grupo no diretório `~/arquivos`