



# LINUX

SANTO OLIANI JUNIOR ●

# UBUNTU

**Server** – Linhas de comando, mais usado para criar um servidor em uma corporação

**Desktop** - Interface Gráfica e Aplicativos pré-instalados

# SISTEMAS DE ARQUIVOS SUPORTADOS

**Ext4** - é o sistema de arquivos padrão usado na maioria das distribuições Linux modernas. Ele suporta arquivos grandes e partições grandes, é eficiente em termos de armazenamento e possui recursos de segurança aprimorados, como suporte a atributos estendidos e controle de acesso.



# SISTEMAS DE ARQUIVOS SUPORTADOS

**Btrfs** - é um sistema de arquivos avançado que oferece suporte a recursos avançados, como instantâneos, clonagem de subvolume, compressão, balanceamento de carga e gerenciamento de quotas. É um sistema de arquivos de próxima geração que é frequentemente usado em aplicativos de armazenamento em nuvem e em servidores de banco de dados.

# SISTEMAS DE ARQUIVOS SUPORTADOS

**XFS** - é um sistema de arquivos de alto desempenho que é capaz de lidar com grandes volumes de dados e grandes arquivos. Ele suporta partições grandes e possui recursos avançados de gerenciamento de armazenamento, como a capacidade de expandir o tamanho da partição em tempo real.

# SISTEMAS DE ARQUIVOS SUPORTADOS

**NTFS** - é um sistema de arquivos comum em sistemas operacionais Windows e pode ser usado no Linux para compartilhar arquivos com sistemas Windows. Ele suporta arquivos grandes, partições grandes e suporte a recursos de segurança, como criptografia e controle de acesso.

**FAT32** - é um sistema de arquivos antigo que foi projetado para sistemas operacionais Windows e é usado em dispositivos de armazenamento portáteis, como unidades flash USB. Ele tem limitações em termos de tamanho máximo de arquivo e tamanho máximo de partição, mas é amplamente suportado em diferentes sistemas operacionais.



# DIRETÓRIOS IMPORTANTES

**/** - A raiz do sistema de arquivos do Linux.

**/bin** - Contém arquivos executáveis essenciais necessários para o sistema operacional funcionar corretamente.

**/boot** - Contém arquivos necessários para inicializar o sistema operacional, como arquivos de configuração do bootloader e imagens do kernel.

**/dev** - Contém arquivos de dispositivo que representam dispositivos de hardware conectados ao sistema.

# DIRETÓRIOS IMPORTANTES

**/etc** - Contém arquivos de configuração do sistema, como arquivos de configuração do sistema de login, do firewall, do servidor de rede, entre outros.

**/home** - Contém as pastas de usuário, onde cada usuário tem uma pasta com seu nome de usuário para armazenar seus arquivos pessoais.

**/lib** - Contém bibliotecas compartilhadas necessárias para executar os programas no sistema.

**/media** - Contém pontos de montagem para dispositivos de armazenamento removíveis, como unidades USB ou CD/DVD.



# DIRETÓRIOS IMPORTANTES

**/mnt** - Contém pontos de montagem para sistemas de arquivos temporários ou permanentes, como unidades de rede ou discos rígidos externos.

**/opt** - Contém software de terceiros instalado no sistema.

**/proc** - Contém informações do sistema, como informações de memória, informações de hardware e informações do processo.

**/root** - A pasta home do usuário root.

# DIRETÓRIOS IMPORTANTES

**/run** - Contém arquivos temporários e informações do sistema usadas durante a inicialização do sistema.

**/sbin** - Contém arquivos executáveis do sistema usados por administradores para gerenciar o sistema.

**/tmp** - Contém arquivos temporários usados por aplicativos em execução no sistema.

**/usr** - Contém programas e arquivos de dados usados por aplicativos do sistema, como bibliotecas e arquivos de cabeçalho.

**/var** - Contém arquivos de dados variáveis que mudam durante a operação do sistema, como arquivos de log e spools de impressora.

# EXECUTÁVEIS LINUX

Os executáveis no Linux ficam armazenados em várias pastas do sistema de arquivos, dependendo do tipo de arquivo e sua finalidade.



# EXECUTÁVEIS LINUX

**/bin** - Contém executáveis essenciais para o sistema operacional funcionar corretamente.

**/sbin** - Contém executáveis do sistema usados por administradores para gerenciar o sistema.

**/usr/bin** - Contém executáveis de programas instalados pelos usuários.

**/usr/sbin** - Contém executáveis de programas de sistema instalados pelos usuários.

**/usr/local/bin** - Contém executáveis de programas instalados manualmente pelo usuário, geralmente fora do gerenciador de pacotes padrão do sistema operacional.

**/opt** - Contém software de terceiros instalado no sistema.

# COMANDOS BÁSICOS DO LINUX

**ls** - Mostra o conteúdo do diretório atual.

**cd** - Altera o diretório atual.

**mkdir** - Cria um novo diretório.

**touch** - Cria um novo arquivo vazio.

**cat** - Mostra o conteúdo de um arquivo.

**cp** - Copia um arquivo de um local para outro.

# COMANDOS BÁSICOS DO LINUX

**mv** - Move ou renomeia um arquivo ou diretório.

**rm** - Remove um arquivo.

**rmdir** - Remove um diretório vazio.

**sudo** - Permite executar um comando com privilégios de administrador.

**nano** - editor de texto simples



# OPÇÕES COMANDOS

`rm -R` (recursive) apaga todo conteúdo de um diretório

`sudo rm -R nome-do-diretório` (com privilégios de administrador)

`<comando> --help` – Lista sintaxe e opções do comando

# Caminho relativo e caminho absoluto

~\$: <caminho\_relativo>

~\$: /<caminho\_absoluto>

/home/user#: Conectado como root

# Pipes e filtros |

Os comandos de filtragem ajudam a classificar, exibir e selecionar partes da saída de um comando.

Filtros tomam a entrada padrão Linux e realizam uma operação sobre ela, e enviam o resultado para uma saída padrão Linux.



# Pipes e filtros |

**wc** – o wc conta caracteres, linhas e palavras.

**sort** – Classifica entrada padrão, então envia o resultado classificado para a saída padrão.

**tee** – O comando tee envia o resultado do programa ao mesmo tempo para a saída padrão (tela) e para um arquivo. Este comando deve ser usado com o pipe “|”.

**find** – Find retorna arquivos com nomes de arquivo que correspondem ao argumento passado para find.

# Pipes e filtros |

**grep** – Grep retorna o texto que corresponde ao padrão de string passado para grep.

**tr** – localiza e substitui uma string por outra. O tr pode ser usado para executar tarefas como conversões de MAIÚSCULAS/minúsculas

**head** – Envia o resultado das primeiras poucas linhas a partir da entrada. Útil para obter o cabeçalho de um arquivo.

**tail** – Envia o resultado das últimas linhas a partir de sua entrada. Útil para obter resultados como por exemplo as entradas mais recentes de um arquivo de log.

# USUÁRIOS

- Criar um usuário:

```
sudo adduser [nome_do_usuario]
```



# USUÁRIOS

- Excluir um usuário:

```
sudo deluser [nome_do_usuario] <-r>
```

-r indica a remoção do diretório do usuário no /home

# USUÁRIOS

- Alterar nome do usuário no Linux

```
usermod -l {login-novo} {login-atual}
```

# USUÁRIOS

- Alterar a senha do usuário:

`sudo passwd [nome_do_usuario]`: permite alterar a senha do usuário.

- Listar usuários no Linux:

`sudo cat /etc/passwd`



# GRUPOS

- Adicionar um usuário ao grupo:  
`sudo groupadd nome_do_grupo`
- Deletar um grupo  
`sudo groupdel nome_do_grupo`
- Listar Grupos  
`sudo groupdel nome_do_grupo`

# GRUPOS

- Alterar grupo principal de um usuário:

```
sudo usermod -g nome_do_grupo  
nome_do_usuario
```

- Adicionar um usuário a vários grupos:

```
sudo usermod -aG <grupo1>,<grupo2>  
<nomeUsuario>
```

O parâmetro -aG significa que você está adicionando o usuário a um ou mais grupos e não substituindo os grupos existentes. Se você omitir o -a, o comando substituirá os grupos existentes pelo(s) grupo(s) especificado(s).

# GRUPOS

Após executar o comando, você pode verificar se o usuário foi adicionado aos grupos usando o comando `id` seguido do nome de usuário. Por exemplo:

```
id <usuario>
```



# USUÁRIOS E GRUPOS

Alterando o dono / grupo de um arquivo e ou diretório:

`chown [opções] [novo_proprietário]`

`nomes_arquivos`

`chgrp [opções] [novo_grupo] arquivo`

# Permissões

Adicionar permissões a um arquivo ou diretório

```
sudo chmod [permissões] [nome_do_arquivo_ou_diretório]
```

1 = execução

2 – Escrita

4 – Leitura

Leitura + Execução =  $1+4 = 5$

Escrita + leitura =  $2+4 = 6$

Permissão	Binário	Decimal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

-----	000
r-----	400
r--r--r--	444
rw-----	600
rw-r--r--	644
rw-rw-rw-	666
rwX-----	700
rwXr-x---	750
rwXr-xr-x	755
rwXrwXrwX	777



Símbolo

**u**        =>        **usuário**  
**g**        =>        **grupo**  
**O** (letra 'o' maiúscula) =>  
          **outro**  
**a** => **todos**

Símbolo

**r** => **leitura**  
**w** => **gravação**  
**x** => **execução**

**+ (sinal de adição) => adicionar permissão**  
**- (sinal de subtração) => remover permissão**  
**= (sinal de igualdade) => definir permissão**

**Exemplo: chmod u+w teste.old**

# Arquivos em lote

No Linux não é a extensão do arquivo e sim a permissão como visto no slide anterior que o transforma em “executável”

# Processos no Linux

`ps` – lista os processos

`ps -ef` – lista todos os processos

`Kill -9 pid` (obtido através do `os`)

`ps | grep <palavra>` - filtra todos os processos com aquela palavra



# OBRIGADO



santo.oliani@fatec.sp.gov.br



santo.oliani@etec.sp.gov.br



santo@oliani jr.com.br