

# Gerenciamento de Usuários

Conceitos Gerais

# 1. Introdução

- O gerenciamento de usuários no Linux é uma parte **fundamental da administração de sistemas**, porque está diretamente ligado à **segurança, organização e controle de acesso**.
- Em sistemas Linux (como o Debian ou Ubuntu), o sistema é **multiusuário por natureza**. Isso significa que várias pessoas (ou serviços) podem usar o sistema ao mesmo tempo. O gerenciamento de usuários permite:
  - **Segurança:** cada usuário tem permissões específicas
  - **Isolamento:** um usuário não interfere no outro
  - **Auditoria:** saber quem fez o quê no sistema
  - **Controle de recursos:** limitar acessos e ações

## 2. Gerenciamento de Usuários

- As informações dos usuários ficam armazenadas nos seguintes arquivos do sistema:

`/etc/passwd` → dados básicos dos usuários

`/etc/shadow` → senhas criptografadas

`/etc/group` → grupos

Por exemplo, o usuário "paulo" no `/etc/passwd`:

```
paulo:x:1001:1001:Paulo Soares:/home/paulo:/bin/bash
```

## 2. Gerenciamento de Usuários

- Explicando melhor cada parte das informações:

```
nome : senha : UID : GID : descrição : home : shell
```

1 – nome: É o **identificador textual do usuário**, usado para fazer login.

2 – UID: É o **ID numérico único do usuário no sistema**. O Linux usa o UID para controlar permissões, identificar dono de arquivos e gerenciar processos.

0 → root (superusuário)

1-999 → usuários do sistema (serviços)

1000+ → usuários comuns (humanos)

## 2. Gerenciamento de Usuários

3 – GID: É o **ID do grupo principal do usuário**. Todo usuário pertence a pelo menos um grupo. Esse grupo define permissões coletivas.

4 – Home: É a **pasta pessoal do usuário**.

5 - Shell padrão: É o **programa que será executado quando o usuário fizer login**. Normalmente é um interpretador de comandos (shell).

# 3. Criando Usuários

- Existem duas formas de criar usuários. O comando "adduser" cria o usuário, e automaticamente já permite definir a senha e o diretório home.
- Quando o usuário é criado com "adduser", os arquivos do /etc chamados passwd, shadow e group são atualizados com os dados do novo usuário.
- Antigamente a senha ficava salva no arquivo passwd (arquivo sem muito proteção). Atualmente, a senha é criptografada e fica no arquivo shadow (mais seguro).

```
#comando para criar usuário abel  
#uma serie de perguntas é feita para entrar com os dados do usuario  
adduser abel
```

# 3. Criando Usuários

- Existem duas formas de criar usuários. O comando "**adduser**" cria o usuário, e automaticamente já permite definir a senha e o diretório home.
- Quando o usuário é criado com "adduser", os arquivos do /etc chamados passwd, shadow e group são atualizados com os dados do novo usuário.
- Antigamente a senha ficava salva no arquivo passwd (arquivo sem muito proteção). Atualmente, a senha é criptografada e fica no arquivo shadow (mais seguro).

```
#comando para criar usuário abel  
#uma serie de perguntas é feita para entrar com os dados do usuario  
adduser abel
```

# 3. Criando Usuários

- O comando "adduser" faz algumas perguntas, como mostra a imagem abaixo:

```
paulo@notebook1:~$ su -  
Senha:  
root@notebook1:~# adduser abel  
Nova senha:  
Redigite a nova senha:  
passwd: senha atualizada com sucesso  
Modificando as informações de usuário para abel  
Informe o novo valor ou pressione ENTER para aceitar o padrão  
  Nome Completo []: Abel Ferreira  
  Número da Sala []: 10  
  Fone de Trabalho []: 44999577875  
  Fone Residencial []:  
  Outro []:  
Is the information correct? [Y/n] Y  
root@notebook1:~#
```

# 3. Criando Usuários

- O comando "**adduser**" também permite adicionar os parâmetros já no terminal:

```
sudo adduser --home /home/joao --shell /bin/bash --uid 1500 abel
```

# 3. Criando Usuários

- O comando "**useradd**" é similar ao anterior, no entanto ele apenas cria o usuário, não define senha e nem o diretório home por padrão.
- Geralmente ele é usado par criar usuários no meio de um código pertencente a um script. Para definir a senha, usa-se o comando "passwd".

```
sudo useradd abel  
passwd abel
```

## 4. Excluindo Usuários

- O comando "**userdel**" remove um usuário. Com os parâmetros "r" remove o home e com "f" força a remoção mesmo que o usuário esteja logado.

```
#deleta o usuário e seu diretório home  
sudo userdel -rf abel
```

## 5. Bloqueando Usuários

- O comando "**passwd**" além de cadastrar ou alterar a senha de usuário, também permite bloquear e desbloquear os usuários:

```
#bloqueia e desbloqueia  
sudo passwd -l abel  
sudo passwd -u abel
```

## 6. Alternando Usuários

- O "**su**" é um dos comandos mais importantes no Linux para **troca de usuário**, especialmente para virar root. Ele permite trocar de usuário no terminal e executar comandos como outro usuário.
- Quando é usado "su paulo", é trocado para usuário paulo, mas mantém o ambiente do usuário anterior. Quando usa "su -paulo", carrega variáveis de ambiente, PATH correto e diretório home do paulo.

```
#ganhar os poderes do usuario paulo
```

```
su -paulo
```

```
#se tornar root
```

```
su -
```

# 7. Consultando Usuários

- O comando "w" mostra os usuários logados naquele momento no Linux:

```
#verificando usuários logados  
w
```

```
root@notebook1:~# w  
20:44:36 up 1:45, 2 users, load average: 0,20, 0,47, 0,46  
USUARIO  TTY      DE              LOGIN@  IDLE   JCPU   PCPU   WHAT  
root     pts/1    -               20:17   0.00s  0.05s  0.03s  w  
paulo    tty2     -               19:00   1:45m  0.09s  0.08s  /usr/bin/startplasma-wayland
```

# 8. Gerencia de Permissões dos Usuários

- O Linux organiza os arquivos em 3 “níveis” de controle:

**1 - User (dono)** → quem criou ou é proprietário

**2 - Group (grupo)** → conjunto de usuários

**3 - Others (outros)** → todo o resto do sistema

- Quando o comando "ls -l" é dado em um diretório, são listadas as permissões

```
-rwxr-xr--
```

- O primeiro caractere define o tipo do arquivo, conforme a imagem:

- = arquivo comum

d = diretório

l = link



# 8. Gerencia de Permissões dos Usuários

- Para diretórios as permissões mudam um pouco:

1 – r : permite listar os arquivos do diretório

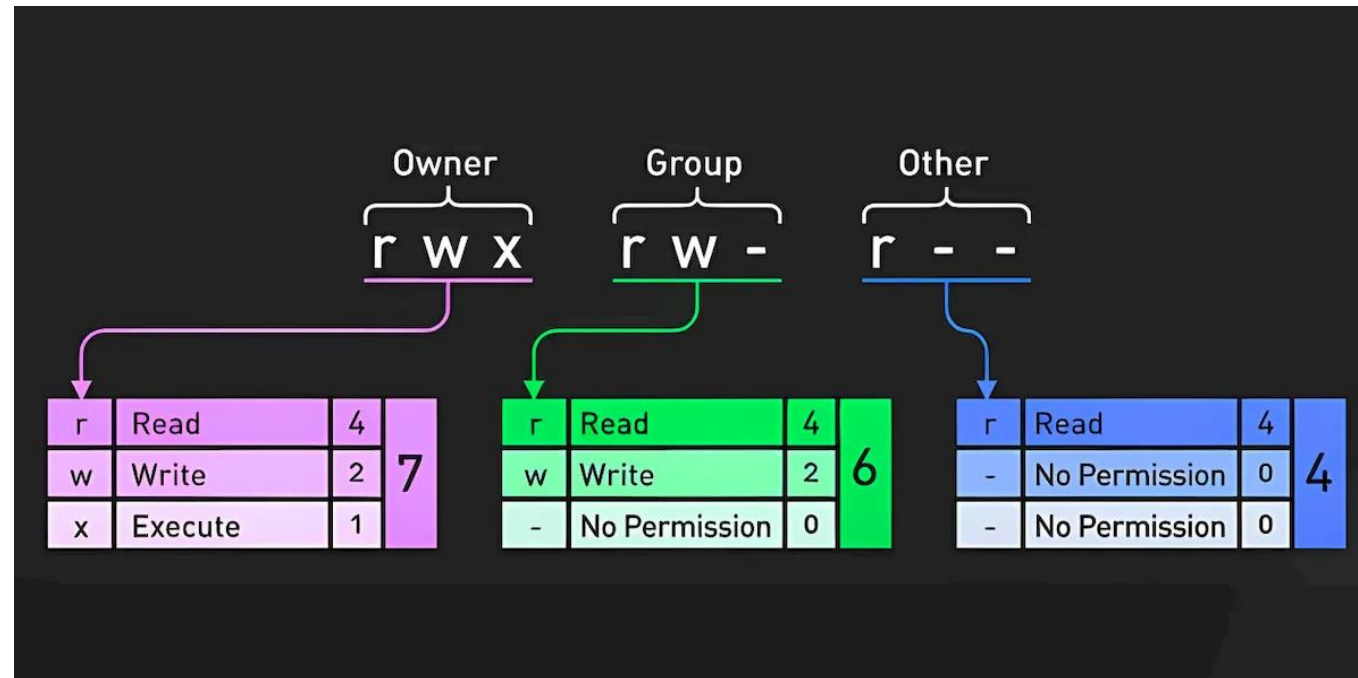
2 – w : permite adicionar e remover arquivos do diretório

3 – x : permite entrar no diretório



# 8. Gerencia de Permissões dos Usuários

- Para alterar as permissões de um arquivo, utiliza-se o comando "chmod", seguido do número que representa a permissão, como mostra o quadro abaixo:



## 8. Gerencia de Permissões dos Usuários

- Deve-se somar os números que representam cada uma das operações (1-x, 2-w, 4-r e 0-nada), para cada um dos grupos (dona, grupo e outros), como mostra o quadro abaixo:

```
#comando dá permissão total ao dono (1+2+4)  
#permissão de execução e leitura para o grupo  
#outros usuários não tem permissão para nada  
chmod 750 /etc/passwd
```

## 8. Gerencia de Permissões dos Usuários

- Geralmente, o dono do arquivo é o usuário que criou o mesmo. Entretanto, é possível alterar o dono e o grupo de qualquer arquivo ou diretório utilizando o comando "**chown**".

```
#alterando o dono do arquivo para usuário paulo  
#alterando o grupo do arquivo para grupo palmeiras  
chown paulo:palmeiras /home/logo.pdf
```

```
#altera apenas o dono para paulo  
chown paulo /home/logo.pdf
```

```
#altera apenas o grupo para palmeiras  
chown :palmeiras /home/logo.pdf
```

## 8. Gerencia de Permissões dos Usuários

- Os comandos "**chown**" e "**chmod**" aceitam recursividade com a chave "-R".

```
#todo o conteudo do diretório teste (arquivos e subdiretorios) terão permissão 764  
chmod -R 764 /teste
```

```
#altera recursivamente o dono do diretório para o usuario paulo  
chown -R paulo /home/teste
```

# 9. Criando Grupos

- O comando "**groupadd**" cria um grupo, para que usuários possam ser adicionados. O quadro abaixo mostra como usar:

```
#cria o grupo palmeirense  
sudo groupadd palmeirense
```

```
#adiciona o usuario paulo ao grupo  
#-a não remove dos outros grupos e -G grupo suplementar  
sudo usermod -aG palmeirense paulo
```

```
#remover o grupo  
sudo groupdel palmeirense
```

```
#visualiza grupos do usuário  
groups paulo
```