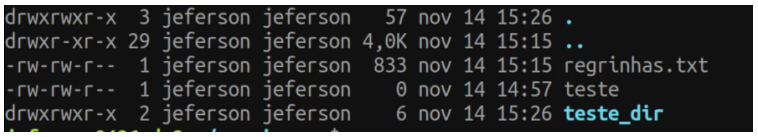
Capítulo 5: Permissões

☑ Reviewed ☐

▼ chmod

PERMISSÕES



1. **Primeiro caractere: Tipo de arquivos**
   - `` : Indica um arquivo regular.
   - `d` : Indica um diretório.
   - `l` : Indica um link simbólico.
   - `-` arquivo de texto
   - `c` Dispositivos de caractere (portas que transferem informações)
   - `s` socket
2. **Caracteres 2-4: Permissões do proprietário**
   - `r` : Permissão de leitura.
   - `w` : Permissão de escrita.
   - `x` : Permissão de execução.
   - `-` : Indica ausência de permissão.
3. **Caracteres 5-7: Permissões do grupo**
   - `r` : Permissão de leitura.
   - `w` : Permissão de escrita.

- `x`: Permissão de execução.
- `-`: Indica ausência de permissão.

4. **Caracteres 8-10: Permissões para outros usuários**
   - `r`: Permissão de leitura.
   - `w`: Permissão de escrita.
   - `x`: Permissão de execução.
   - `-`: Indica ausência de permissão.

```
-rw-rw-r--
```

OBSERÇÃO: Se você tiver subpastas e quer aplicar as permissões a todas elas, basta colocar como recursivo (chmod -R 511 teste_dir/)

COMANDOS

```
chmod - modifica as permissões
chown - modifica o dono/grupo do arquivo/diretório
chgrp - modifica o grupo do arquivo/diretório
umask - define a permissão padrão de arquivos e diretór
```

exemplos - Usando maneira 'Simbólica' - CHMOD

Permissões Básicas

- **r**: Leitura (read)
- **w**: Escrita (write)
- **x**: Execução (execute)

Categorias de Usuários

- **u**: Proprietário (user)

- **g**: Grupo (group)
- **o**: Outros (others)
- **a**: Todos (all)

Operadores

- **+**: Adiciona uma permissão
- **-**: Remove uma permissão
- **=**: Define permissões exatas

Exemplos de Uso do `chmod` com Forma Simbólica

1. **Adicionar Permissão de Execução para o Proprietário**

   ```
   chmod u+x arquivo.txt
   ```

   Adiciona a permissão de execução para o proprietário do arquivo `arquivo.txt`.

2. **Remover Permissão de Escrita para o Grupo**

   ```
   chmod g-w arquivo.txt
   ```

   Remove a permissão de escrita para o grupo em `arquivo.txt`.

3. **Adicionar Permissão de Leitura para Outros**

   ```
   chmod o+r arquivo.txt
   ```

   Adiciona a permissão de leitura para outros usuários em `arquivo.txt`.

4. **Definir Permissões Exatas para o Proprietário e Grupo**

```
chmod u=rwx,g=rx arquivo.txt
```

Define permissões de leitura, escrita e execução para o proprietário e permissões de leitura e execução para o grupo em `arquivo.txt`.

5. **Remover Todas as Permissões para Outros**

```
chmod o-rwx arquivo.txt
```

Remove todas as permissões para outros usuários em `arquivo.txt`.

6. **Adicionar Permissão de Leitura para o dono do grupo**

```
chmod u=r diretorio01_1/teste.txt
```

7. restaurar as permissões originais do arquivo `teste.txt` para o dono (usuário)

```
#~/diretorio01/diretorio01_1$ ls -l

#-rw-rw-r-- 1 ubuntu ubuntu 24 Sep 13 14:33 teste.txt


chmod u=rw,g=rw,o=r diretorio01/diretorio01_1/teste.txt
```

Exemplo com maneira 'Octal':

- Observação: NUNCA DEIXAR COMO '777', pois todos vão ter todas as permissões

Permissões Básicas e Seus Valores Octais

- **Leitura (r)**: 4
- **Escrita (w)**: 2
- **Execução (x)**: 1

As permissões são representadas por três dígitos octais:

1. **Primeiro Dígito**: Permissões para o Proprietário
2. **Segundo Dígito**: Permissões para o Grupo
3. **Terceiro Dígito**: Permissões para Outros

Cada dígito é a soma das permissões que você deseja conceder. Por exemplo:

- **7** = 4 (leitura) + 2 (escrita) + 1 (execução) = rwx
- **6** = 4 (leitura) + 2 (escrita) = rw-
- **5** = 4 (leitura) + 1 (execução) = r-x
- **4** = 4 (leitura) = r--

Exemplos de Uso do `chmod` com Notação Octal

1. **Definir Permissões Completas para o Proprietário, Leitura e Execução para o Grupo e Nenhuma Permissão para Outros**

   ```
   chmod 750 arquivo.txt
   ```

   - Proprietário: `rwx` (7)
   - Grupo: `r-x` (5)
   - Outros: `--` (0)

2. **Definir Permissões de Leitura e Escrita para o Proprietário e Grupo, e Nenhuma Permissão para Outros**

   ```
   chmod 664 arquivo.txt
   ```

   - Proprietário: `rw-` (6)
   - Grupo: `rw-` (6)

- Outros: `--` (4)

3. **Definir Permissões de Leitura e Execução para Todos**

```
chmod 555 arquivo.txt
```

   - Proprietário: `r-x` (5)
   - Grupo: `r-x` (5)
   - Outros: `r-x` (5)

4. **Definir Permissões Completas para o Proprietário e Somente Leitura para o Grupo e Outros**

```
chmod 744 arquivo.txt
```

   - Proprietário: `rwx` (7)
   - Grupo: `r--` (4)
   - Outros: `r--` (4)

5. **Remover Todas as Permissões para o Grupo e Outros**

```
chmod 700 arquivo.txt
```

   - Proprietário: `rwx` (7)
   - Grupo: `--` (0)
   - Outros: `--` (0)

Resumo dos Comandos `chmod` Octais

- **Definir Permissões:**

```
chmod [modo octal] arquivo
```

- **Exemplos de Modos Octais:**

- **777**: Permissões completas para todos (`rwxrwxrwx`)
- **755**: Permissões completas para o proprietário e leitura e execução para grupo e outros (`rwxr-xr-x`)
- **644**: Leitura e escrita para o proprietário, leitura para grupo e outros (`rw-r--r--`)

▼ CHOWN e CHGRP

- Serve para mudar o grupo/dono do arquivo/diretório

```
chown giropops teste
```

```
chown giropops:users teste ou chown giropops.users teste
```

```
chown :jeferson teste
```

CHGRP

```
chgrp jeferson teste_dir/
```

▼ STICK BIT, SUID, SGID e UMASK

STICK BIT

- representado pela letra t e na maneira octal pelo ‘1’
- apenas o proprietário do arquivo, o proprietário do diretório e o superusuário podem renomear ou excluir arquivos dentro desse diretório, mesmo que outros usuários tenham permissão de escrita no diretório.

```
chmod +t nome_do_diretorio
```

SUID

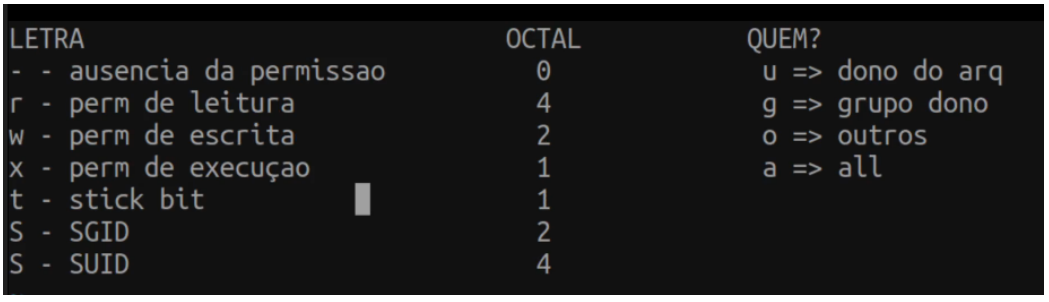
- Todo arquivo binário que tiver essa permissão habilitada, qualquer pessoa que executar esse executável vai ser como se fosse dono do executável
- Representado pela ''S'' e no Octal pelo ''4''

```
chmod 4755 nome_do_arquivo
```

GUID

- Todo arquivo binário que tiver essa permissão habilitada, o grupo que executar esse executável vai ser como se fosse dono do executável
- Representado pelo ''S'' e no Octal pelo ''2''

```
chmod 2755 nome_do_arquivo
```



UMASK

- A máscara de permissão (umask) determina as permissões padrão de um novo arquivo ou diretório criado por um usuário no sistema operacional Unix/Linux.

- Quando você fizer logout ele não vai ter mais o mesmo mask, então voce, se quiser colocar como padrão, precisa salvar o arquivo
- Colinha: Permissão de arquivos é sempre 666.Permissão de diretório é sempre 777.A permissão padrão de quando for criar um diretório ou arquivo é o valor (arquivo ou diretorio) - mask

```
umask valor
```

▼ *PROTEGENDO ATRIBUTOS*

`chattr` - alterar os atributos de um arquivo (imutável, escrita etc.)

`lsattr` - listar os atributos de um arquivo

`chattr +i nome_do_arquivo` - imutável (ex: chattr +i teste)

`chattr -i nome_do_arquivo` - tornar mutável novamente

`chattr +a nome_do_arquivo` - impede a sobrescrição de dados (echo "Linha 3" > teste 3) para testar

`chattr +c` - compactado

`chattr =ai *` - atribui os atributos a todos os arquivos

`chattr +D attr` - mesmo que alguém tenha permissões para escrever nesse diretório, não será possível excluir nenhum arquivo ou subdiretório dentro dele

`chattr +c` - sistema operacional tenta compactar automaticamente o arquivo quando está sendo gravado no disco

`chattr +s` - realiza atualizações síncronas no arquivo

`chattr +S` - grava rapidamente no disco.Diferente do 'sync', que demora um pouco mais.