Utilizar os operadores>(maior),>>(maior maior) e | (pipe) no terminal;

Maior > substituir

simplesmente regrava o arquivo de saída;
redireciona um resultado de uma saída para outra.

Ex : cat > nomedoarquivo.txt, e Enter. o sistema vai esperar valores que vc quer. Dá enter entre um e outro. E, ao finalizar, você dá um crtl C.

Maior duplo > >adicionar

adiciona a saída no final do arquivo, sem apagar o conteúdo.

Ex : cat >> nomedoarquivo.txt, enter, e coloca mais valores. Ctrl C para sair.

Pipe |

O Pipe é um comando que permite usar dois ou mais comandos, de forma que a saída de um comando sirva como entrada para o próximo.
Em resumo, a saída de cada processo diretamente como entrada para o próximo como um pipeline.
O símbolo '|' denota um cano. Pipes ajudam a mash-up de dois ou mais comandos ao mesmo tempo e executá-los consecutivamente.

Ex : crie um arquivo chamado top_skills.txt usando o skills2.txt, contendo as 3 primeiras skills em ordem alfabética =

```
cat skills2.txt | sort | head -n 3 >
top_skills.txt
```

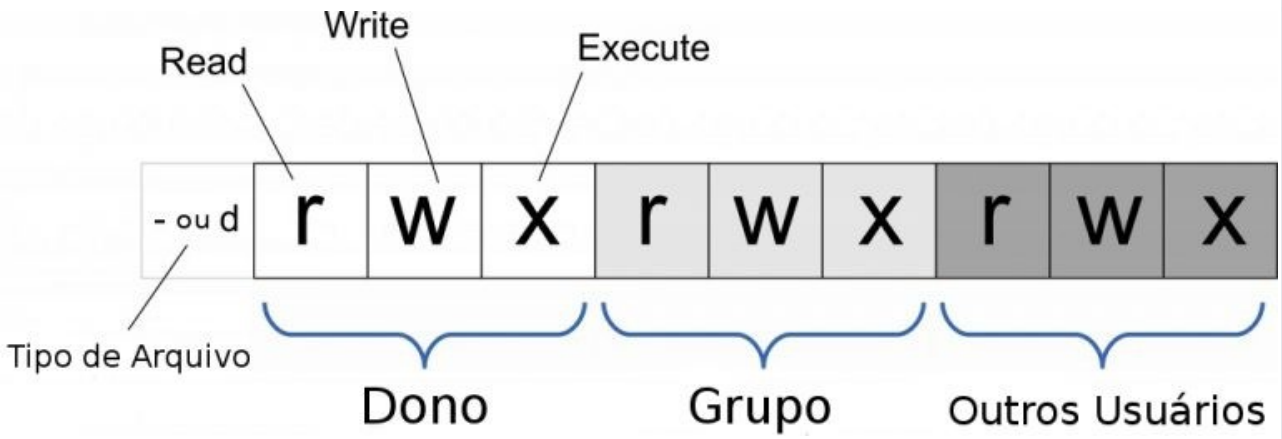
●Alterar as permissões de arquivos e diretórios;

Como verificar as permissões de um arquivo no Linux?

O comando chmod é usado para lidar com **permissões** de **arquivos** do sistema **Linux**. Em sistemas

como o **Linux**, cada **arquivo** possui um sistema de sinalizadores que indicam quais são as **permissões** que cada usuário tem para ler ou editar cada **arquivo**.

No Unix e sistemas baseados em Unix, como no Linux, há três modelos de controle de acesso básicos: Read, Write e Execution, ou seja, Leitura, Escrita e Execução.



O Tipo do Objeto(arquivo) significa que podem ser:

d => diretório;

- b => arquivo de bloco;
- c => arquivo especial de caractere;
- p => canal;
- s => socket;
- – => arquivo normal.

Já os outros caracteres significam:

•r => permissão de leitura (read);

•w => permissão de gravação (write);

•x => permissão de execução (execution);

•– => permissão desabilitada.

•Explicando um pouco mais:

Leitura (r)

Em arquivos, permite examinar o conteúdo do arquivo.
Em diretórios permite listar conteúdo do diretório.

Escrita (w):
Em arquivos, permite escrever, remover e alterar o arquivo.
Em diretórios, permite criar e remover arquivos dentro do diretório.
Execução (x):
Em arquivos, permite executar um arquivo como um programa.
Em diretório, permite ler e escrever em arquivos dentro do diretório

Ao listar um arquivo com o comando:

```
ls -l arquivo.extensão
```

Ele mostrará:

```
Permissões | Links  | Proprietário | Grupo | Tamanho | Data e Hora   | Nome
-----------|--------|--------------|-------|---------|---------------|---------
drwxr-xr-x | 2      | root         | root  | 4096    | Out 19 09:10  | composer/
```

Nas Permissões

r - Significa permissão de leitura (read);

w - Significa permissão de gravação (write);

x - Significa permissão de execução (execution);

- - Significa permissão desabilitada.

Links = Número de ligações que o item possui, no caso do diretório, número de subdiretórios que possui;

Proprietário =Quem é a pessoa dona, quem criou. É o diretório padrão da pessoa usuária, o home;

Grupo =Grupo ao qual pertence o item ou diretório. Utilizado para dar permissões à outras pessoas;

Tamanho = Em Bytes;

Data e Hora= Momento em que foi criado ou última modificação;

Nome= Nome do item ou diretório

Podemos também dar permissão através de letras:

U Usuário

G Grupo

O Outros

+ Adicionar permissão

- Remover permissão

= Igualdade

Exemplos:

```
chmod u+w arquivo.extensão
```

O "u" indica o usuário, o sinal de adição (+) indica que está sendo adicionada a permissão e "w" indica que a permissão que está sendo dada é de gravação.

```
chmod g+rw arquivo.extensão
```

Leitura e execução para o grupo.

```
chmod u+rwx arquivo.extensão
```

Aqui estamos dando permissão total para o dono do arquivo.

Utilizar o comando find

O find é um comando para pesquisar em diretórios por arquivos ou outras pastas, de acordo com os parâmetros passados a ele. Esses parâmetros podem ser name, date, size e type. Caso nenhum atributo seja passado, ele pesquisará tudo que estiver dentro do diretório atual.

Exemplo:

```
# Para listar todos os arquivos que terminam em .txt
find . -name "*.txt"
# Para localizar todos os diretórios
find . -type d
```

```
# Para localizar todos os arquivos
find . -type f
```

Utilizar o comando history

O history é um comando que mostra o histórico de comandos que você executou no terminal. A quantidade ou o tamanho desse *"histórico"* podem ser configurados para um número arbitrário de comandos ou para ver todo o histórico.

Exemplo:

```
# Mostra o histórico de comandos
history
# Pegar os últimos 10 comandos
history | tail
```

Utilizar o comando echo

O echoé um comando utilizado em scripts ou no terminal para exibir mensagens na tela ou em um arquivo.

Exemplo:

```
echo "Este é um teste"
# Resultado
Este é um teste
# Pode ser usado para colocar textos dentro de arquivos.
echo "Este é mais um teste" > teste.txt
cat teste.txt
Este é mais um teste
```