

# Comandos para manipulação de arquivos e textos e de redirecionamento

Nano ⇒ Editor de textos do Ubuntu.

No menu do Nano, toda vez que você vir um ^ significa Ctrl.

Se você digita: "nano teste.txt", você abre o arquivo txt no Nano.

## Comandos no Nano:

Ctrl + G → Mostra o arquivo de ajuda do Nano

Ctrl + X → Sair

Ctrl + O → Gravar (Salvar) o arquivo

Ctrl + \ → Substituir (Você digita o texto a ser substituído e depois o texto substituto.)

Ctrl + U → Colar o texto

Ctrl + K → Recortar o texto

Ctrl + J → Justificar

Ctrl + T → Verifica a Ortografia

ALT + A → Vai marcando o texto (Coloca na frente do texto e vai baixando com a seta para baixo)

ALT + U → Desfaz

ALT + E → Refaz

Alt + 6 → Copiar o texto

Você não precisa abrir o Nano para apenas visualizar o conteúdo de um arquivo. Você pode fazer isso através do comando cat.

## Comandos para manipulação de arquivos:

cat → Visualizar o conteúdo de um arquivo.

Ex.: cat teste.txt

tac → Visualizar o conteúdo do arquivo mas com as linhas invertidas.

Ex.: tac teste.txt

head → Mostra as 10 primeiras linhas de um arquivo.

Ex.: head teste.txt

Você pode salvar essa visualização em um outro arquivo.

Ex.: head teste.txt > novoarquivo.txt

tail → Mostra as 10 últimas linhas de um arquivo.

Ex.: tail teste.txt

Você pode salvar essa visualização em um outro arquivo.

Ex.: tail teste.txt > novoarquivo.txt

file → Ele retorna o tipo do arquivo.

Ex.: file teste.txt

Se você colocar file e o nome de um diretório, ele retorna também que aquilo é um diretório.

## **Redirecionamento:**

Comando de redirecionamento (>)

Se você quiser criar um arquivo com a saída de um comando você também consegue:

Ex.: cal > calendario-mes.txt

(Cria o arquivo e adiciona o calendário desse mês a esse arquivo.)

E se você quiser adicionar mais informações, você também consegue.

Ex.: date >> calendario-mes.txt

(Altera o arquivo adicionando a data.)

No caso acima você tem que colocar os dois >> ou então, colocando só um >, ele substitui todo o conteúdo do arquivo pelo novo conteúdo.

Ex.: cal 2021 > calendario-mes.txt

| → (pipe) Outro comando de redirecionamento para quando desejamos utilizar dois comandos juntos.

Ex.: tail teste.txt | grep Maria

tail → Como visto anteriormente, mostra as 10 últimas linhas do arquivo.

grep → Faz uma busca por algum texto dentro desse arquivo.

Quando a gente executar o comando acima, o grep vai marcar de vermelho todas as vezes que ele achar a palavra Maria dentro das 10 últimas linhas do arquivo teste.txt

Outro exemplo usando o grep:

```
grep Anna lista_nomes.txt
```

(Procura o nome Anna no arquivo)

Outro comando muito utilizado com o | (pipe) é o more.

more → É um comando de paginação de textos.

Para facilitar a leitura de um arquivo grande você pode utilizar o pipe more ou o pipe less.

```
Ex.: cat teste.txt | more
```

Nesse caso, será mostrado o começo do arquivo e, ao final, vai ter - -More- - e a medida que você vai pressionando Enter ele vai mostrando mais. Ou seja, o arquivo está paginado.

less → Ele também faz a paginação, mas no less não vai aparecer o - -More- -, o que aparece são : e a medida que você vai pressionando Enter ele também vai mostrando mais do arquivo.

```
Ex.: cat teste.txt | less
```

Operadores de redirecionamento (& e &&)

```
Ex.: cat teste.txt & cat teste2.txt
```

Quando você usa o &, a saída ele vai separar por linha de terminal, ou seja, na primeira linha de terminal a primeira saída do primeiro comando e na segunda linha de terminal a segunda saída, do segundo comando.

Quando você apertar Enter, ele finaliza a saída.

Já quando você usa o &&, ele executa os dois em consecutivo. Ou seja, vai ter uma saída só em uma linha de terminal.

```
Ex.: cat teste.txt && cat teste2.txt
```

Outro exemplo do && é quando você quer criar uma pasta e, juntamente, já quer acessar ela no terminal. Assim, você faz a utilização de dois comandos ao mesmo tempo.

```
Ex.: mkdir NomeDaPasta && cd nomeDaPasta
```