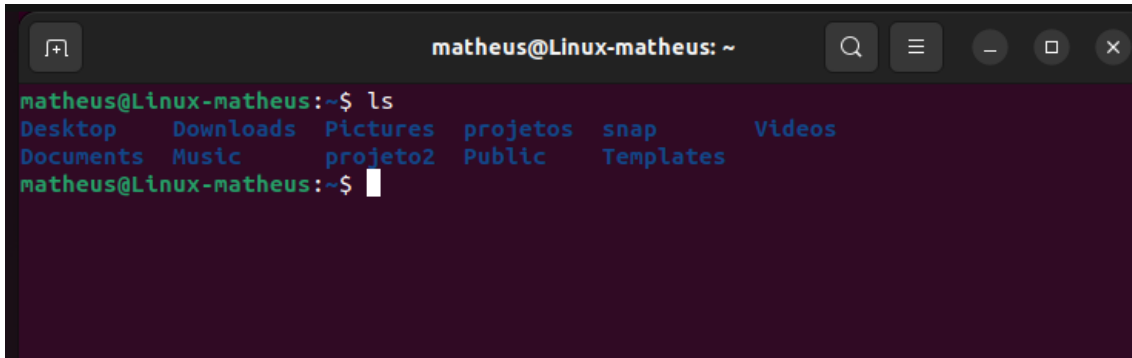


## QUESTÃO 02

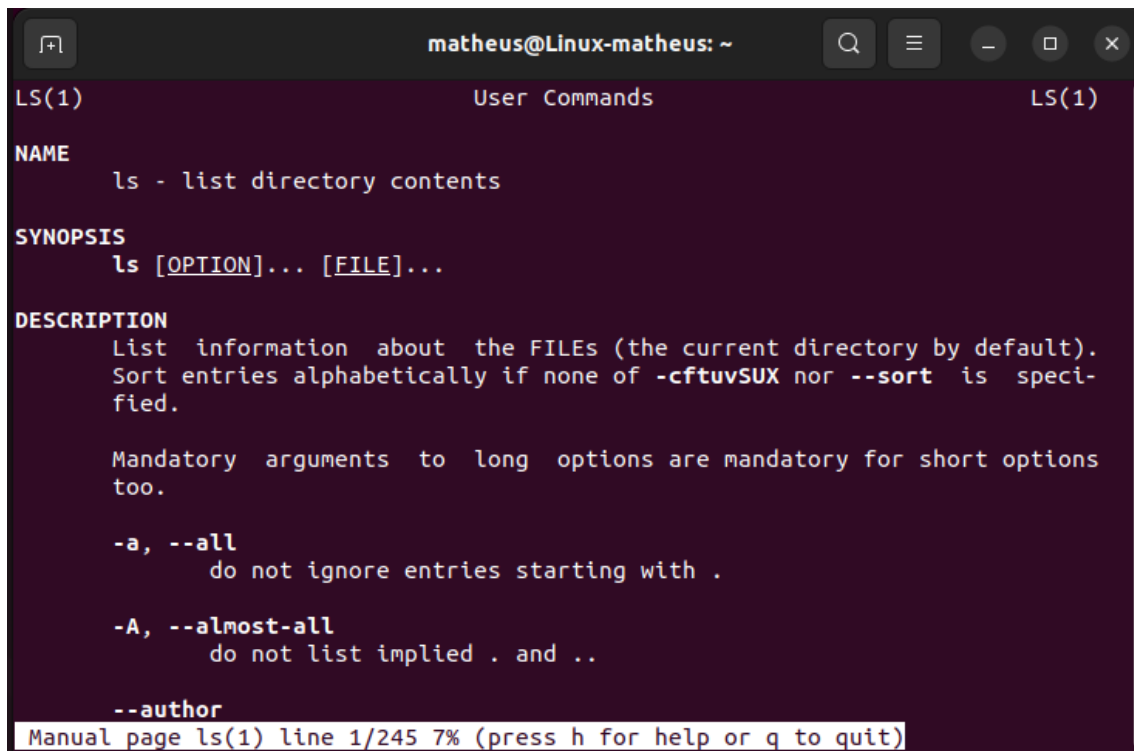
### LS

A terminal window titled 'matheus@Linux-matheus: ~' with search, menu, and window control icons. The command 'ls' has been executed, displaying a list of directories in the home folder: Desktop, Downloads, Pictures, projetos, snap, Videos, Documents, Music, projeto2, Public, and Templates. The prompt 'matheus@Linux-matheus:~\$' is visible at the bottom.

```
matheus@Linux-matheus:~$ ls
Desktop  Downloads  Pictures  projetos  snap      Videos
Documents Music      projeto2  Public    Templates
matheus@Linux-matheus:~$
```

O comando ls serve para listar os arquivos e/ou pastas que se encontram dentro de diretórios, por exemplo na imagem acima o comando ls mostra as pastas que se encontram dentro da pasta home indicada pela ~.

### MAN

A terminal window titled 'matheus@Linux-matheus: ~' showing the manual page for the 'ls' command. The page is titled 'LS(1) User Commands LS(1)'. It includes sections for NAME, SYNOPSIS, and DESCRIPTION. The NAME section states 'ls - list directory contents'. The SYNOPSIS section shows 'ls [OPTION]... [FILE]...'. The DESCRIPTION section explains that 'ls' lists information about files in the current directory by default, sorts entries alphabetically, and lists mandatory arguments for long options. It also shows options like -a, --all, -A, --almost-all, and --author. The bottom of the screen shows 'Manual page ls(1) line 1/245 7% (press h for help or q to quit)'.

```
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

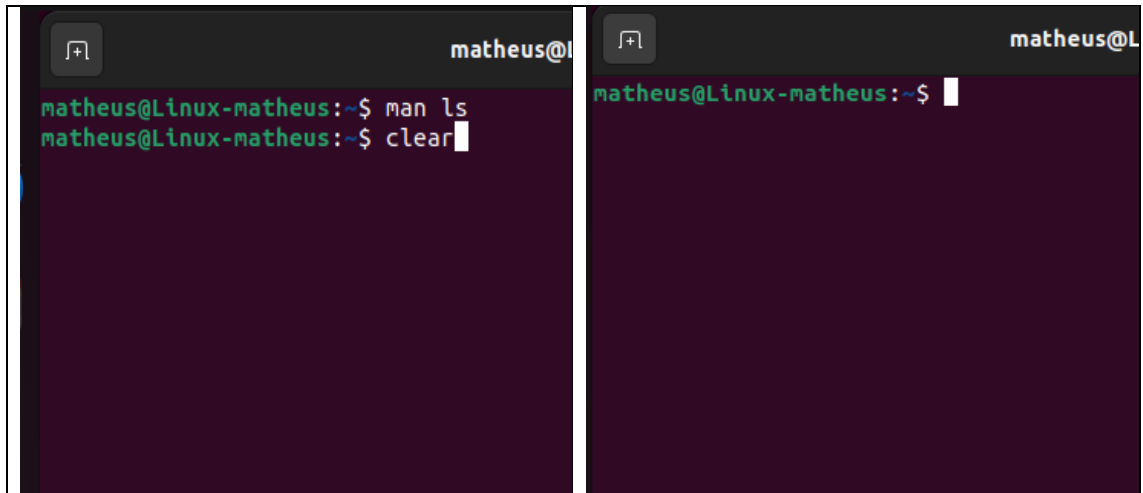
    -A, --almost-all
        do not list implied . and ..

    --author

Manual page ls(1) line 1/245 7% (press h for help or q to quit)
```

O comando man necessita ser acompanhado de outro comando, ele serve para mostrar um manual de explicação de outro comando, a imagem acima representa o manual do comando ls.

## CLEAR

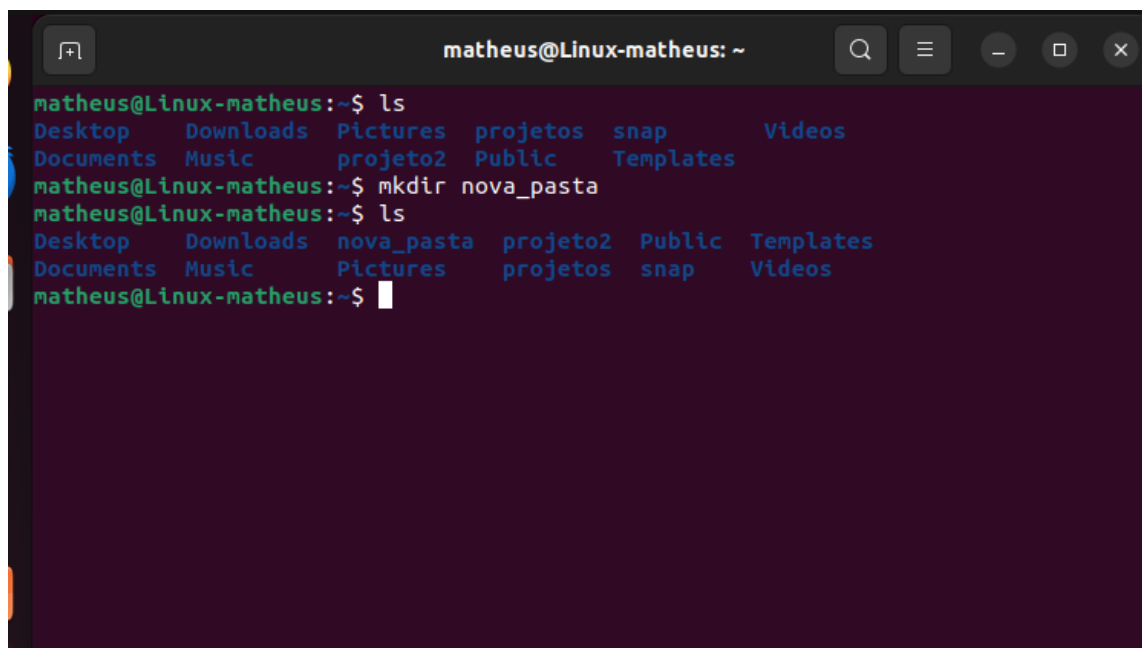


```
matheus@Linux-matheus:~$ man ls
matheus@Linux-matheus:~$ clear

matheus@Linux-matheus:~$
```

O comando clear é um dos mais simples sua função é limpar a tela, ou seja, voltar a visualização ao que era quando abriu o terminal.

## mkdir



```
matheus@Linux-matheus:~$ ls
Desktop  Downloads  Pictures  projetos  snap      Videos
Documents Music      projeto2  Public    Templates

matheus@Linux-matheus:~$ mkdir nova_pasta
matheus@Linux-matheus:~$ ls
Desktop  Downloads  nova_pasta  projeto2  Public  Templates
Documents Music      Pictures    projetos  snap    Videos

matheus@Linux-matheus:~$
```

A função mkdir é relativamente simples, sua função é criar pastas. Olhe no exemplo acima a criação de nova\_pasta.

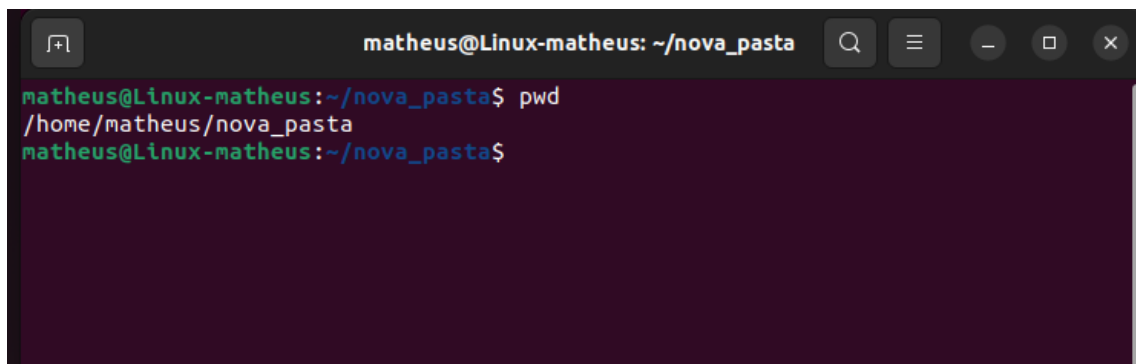
## Cd



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~$ ls
Desktop  Downloads  nova_pasta  projeto2  Public  Templates
Documents Music      Pictures    projetos  snap    Videos
matheus@Linux-matheus:~$ cd nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus@Linux-matheus:~/nova_pasta$
```

O comando `cd`, serve especificamente para entrar em uma pasta e acessar as informações dela, note que o terminal agora mostra informações da nova pasta e não mais da home.

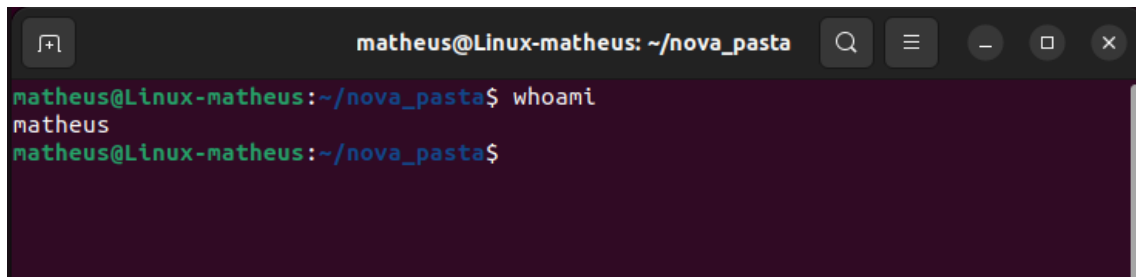
## Pwd



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ pwd
/home/matheus/nova_pasta
matheus@Linux-matheus:~/nova_pasta$
```

É importante muitas vezes saber o caminho de trabalho, ou seja, onde posso encontrar a pasta em que estou trabalhando, destacando o `~` e para isso é usado comando `pwd`.

## Whoami



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ whoami
matheus
matheus@Linux-matheus:~/nova_pasta$
```

Em empresas principalmente é comum haver mais do que um usuário, e para saber qual usuário você está trabalhando é possível usar o comando `whoami`.

## Redirecionamento

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ whoami
matheus
matheus@Linux-matheus:~/nova_pasta$ whoami >> matheus.txt
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  teste01
matheus@Linux-matheus:~/nova_pasta$
```

O redirecionamento é um recurso que me permite passar informações do terminal para um arquivo, nesse caso foi possível passar as informações de whoami para um txt.

## Nome composto

```
matheus@Linux-matheus: ~/nova_pasta/pasta composta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt
matheus@Linux-matheus:~/nova_pasta$ mkdir "pasta composta"
matheus@Linux-matheus:~/nova_pasta$ cd "pasta composta"
matheus@Linux-matheus:~/nova_pasta/pasta composta$
```

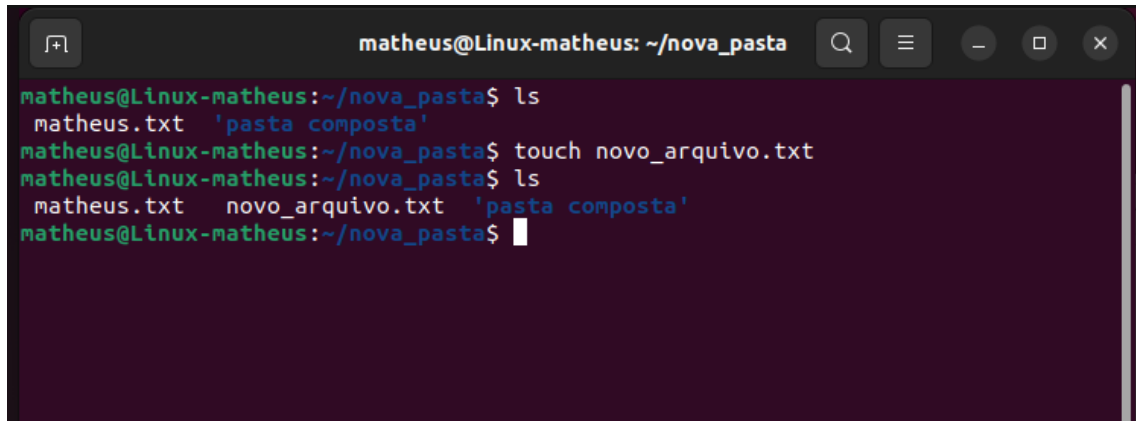
Para criar e/ou acessar pastas compostas o mais comum e indicado é utilizar aspas, porém é possível utilizar \ e espaço para indicar espaçamento.

## Exemplo

```
matheus@Linux-matheus: ~
matheus@Linux-matheus:~$ ls
'Area de Trabalho'  Downloads  Pictures  Public  Videos
Desktop             Music      projeto2  snap
Documents           nova_pasta projetos  Templates
matheus@Linux-matheus:~$ cd Area\ de\ Trabalho/
```

Exemplo do uso de \ e espaço.

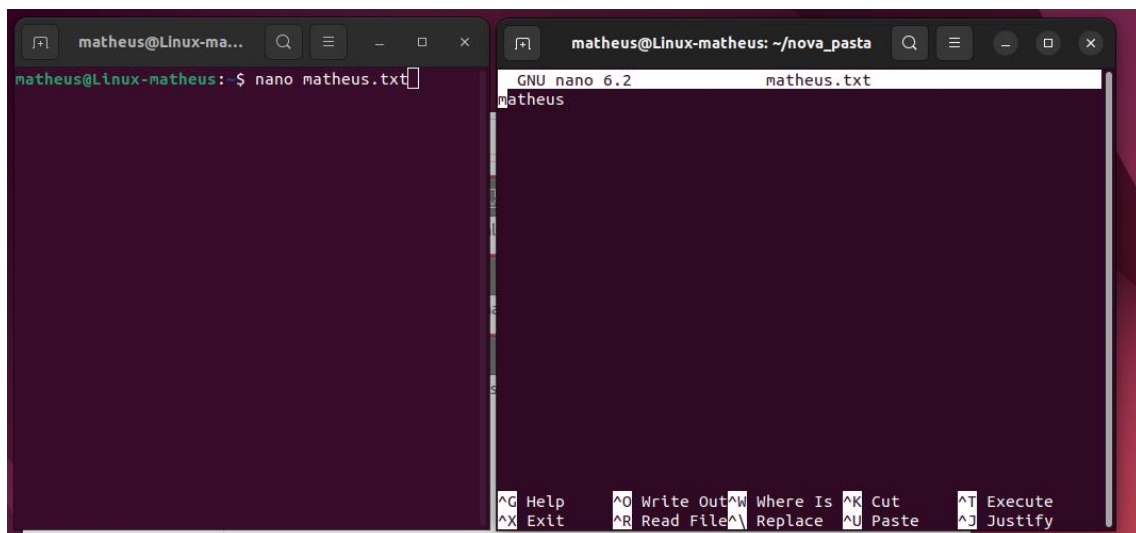
## Touch



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  'pasta composta'
matheus@Linux-matheus:~/nova_pasta$ touch novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  novo_arquivo.txt  'pasta composta'
matheus@Linux-matheus:~/nova_pasta$
```

Esse comando serve para criar arquivos sem a necessidade de se utilizar >>. E esse arquivo é iniciado vazio.

## Nano

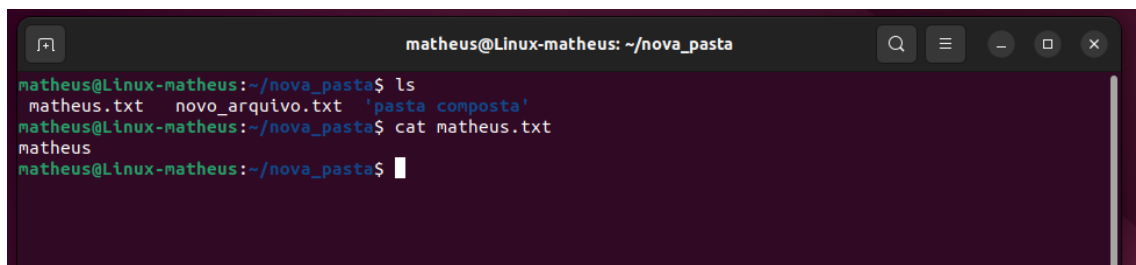


```
matheus@Linux-matheus:~$ nano matheus.txt
GNU nano 6.2 matheus.txt
matheus

^G Help      ^O Write Out ^W Where Is   ^K Cut       ^I Execute
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify
```

Além de criar arquivos é importante poder editá-los e para isso o Linux possui integrado o editor nano que permiti essa edição.

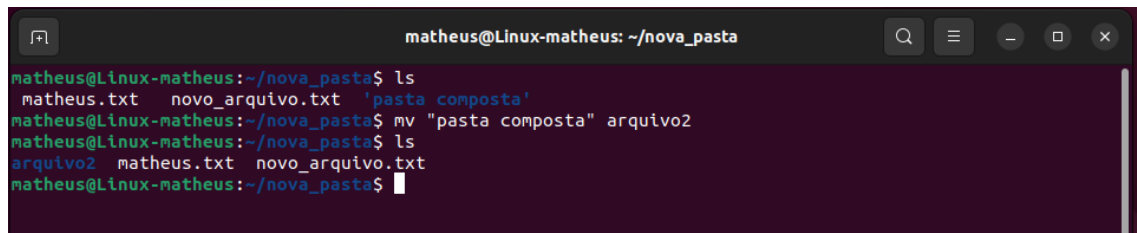
## Cat



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  novo_arquivo.txt  'pasta composta'
matheus@Linux-matheus:~/nova_pasta$ cat matheus.txt
matheus
matheus@Linux-matheus:~/nova_pasta$
```

Com o Linux podemos acessar o conteúdo de um arquivo sem precisar usar um editor. O comando `cat` mostra o conteúdo de arquivos pequenos.

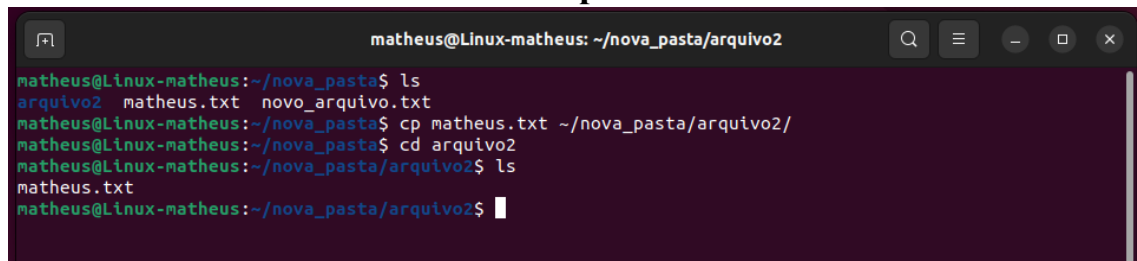
## MV



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  novo_arquivo.txt  'pasta composta'
matheus@Linux-matheus:~/nova_pasta$ mv "pasta composta" arquivo2
matheus@Linux-matheus:~/nova_pasta$ ls
arquivo2  matheus.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$
```

O comando `mv` é utilizado para renomear arquivos e pastas.

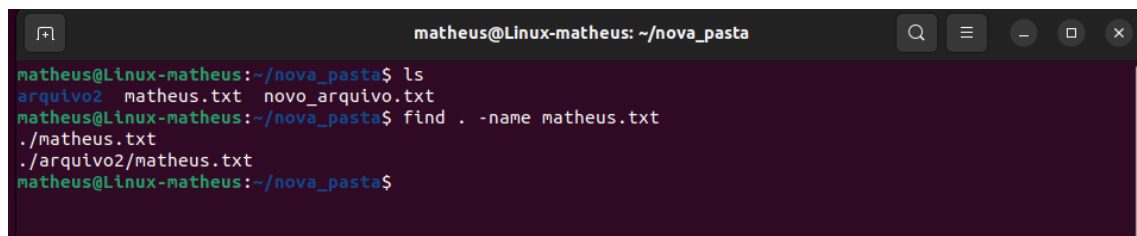
## Cp



```
matheus@Linux-matheus: ~/nova_pasta/arquivo2
matheus@Linux-matheus:~/nova_pasta$ ls
arquivo2  matheus.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ cp matheus.txt ~/nova_pasta/arquivo2/
matheus@Linux-matheus:~/nova_pasta$ cd arquivo2
matheus@Linux-matheus:~/nova_pasta/arquivo2$ ls
matheus.txt
matheus@Linux-matheus:~/nova_pasta/arquivo2$
```

Esse comando serve para copiar arquivos de uma pasta para outra.

## Find



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
arquivo2  matheus.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ find . -name matheus.txt
./matheus.txt
./arquivo2/matheus.txt
matheus@Linux-matheus:~/nova_pasta$
```

O comando `find` serve para procurar arquivos dentro de diretórios

## Head

```
matheus@Linux-matheus: /var/log$ head alternatives.log
update-alternatives 2023-08-07 22:52:51: run with --install /usr/bin/awk awk /usr/bin/mawk 5 --slave /usr/share/man/man1/awk.1.gz awk.1.gz /usr/share/man/man1/mawk.1.gz --slave /usr/bin/nawk nawk /usr/bin/mawk --slave /usr/share/man/man1/nawk.1.gz nawk.1.gz /usr/share/man/man1/mawk.1.gz
update-alternatives 2023-08-07 22:52:51: link group awk updated to point to /usr/bin/mawk
update-alternatives 2023-08-07 22:52:54: run with --install /usr/bin/which which /usr/bin/which.debianutils 0 --slave /usr/share/man/man1/which.1.gz which.1.gz /usr/share/man/man1/which.debianutils.1.gz --slave /usr/share/man/de/man1/which.1.gz which.de1.gz /usr/share/man/de/man1/which.debianutils.1.gz --slave /usr/share/man/es/man1/which.1.gz which.es1.gz /usr/share/man/es/man1/which.debianutils.1.gz --slave /usr/share/man/fr/man1/which.1.gz which.fr1.gz /usr/share/man/fr/man1/which.debianutils.1.gz --slave /usr/share/man/it/man1/which.1.gz which.it1.gz /usr/share/man/it/man1/which.debianutils.1.gz --slave /usr/share/man/ja/man1/which.1.gz which.ja1.gz /usr/share/man/ja/man1/which.debianutils.1.gz --slave /usr/share/man/pl/man1/which.1.gz which.pl1.gz /usr/share/man/pl/man1/which.debianutils.1.gz --slave /usr/share/man/sl/man1/which.1.gz which.sl1.gz /usr/share/man/sl/man1/which.debianutils.1.gz
update-alternatives 2023-08-07 22:52:54: link group which updated to point to /usr/bin/which.debianutils
update-alternatives 2023-08-07 22:52:54: run with --install /usr/bin/awk awk /usr/bin/mawk 5 --slave /usr/share/man/man1/awk.1.gz awk.1.gz /usr/share/man/man1/mawk.1.gz --slave /usr/bin/nawk nawk /usr/bin/mawk --slave /usr/share/man/man1/nawk.1.gz nawk.1.gz /usr/share/man/man1/mawk.1.gz
update-alternatives 2023-08-07 22:52:55: run with --install /usr/share/man/man7/builtins.7.gz builtins.7.gz /usr/share/man/man7/bash-builtins.7.gz 10
update-alternatives 2023-08-07 22:52:55: link group builtins.7.gz updated to point to /usr/share/man/man7/bash-builtins.7.gz
update-alternatives 2023-08-07 22:52:55: run with --install /usr/sbin/rmt rmt /usr/sbin/rmt-tar 50 --slave /usr/share/man/man8/rmt.8.gz rmt.8.gz /usr/share/man/man8/rmt-tar.8.gz
update-alternatives 2023-08-07 22:52:55: link group rmt updated to point to /usr/sbin/rmt-tar
update-alternatives 2023-08-07 22:52:55: run with --install /usr/bin/pager pager /bin/more 50 --slave /usr/share/man/man1/pager.1.gz pager.1.gz /usr/share/man/man1/more.1.gz
matheus@Linux-matheus: /var/log$
```

Em arquivos muito grandes o comando cat não é indicado para isso é necessário o comando head que irá imprimir o topo do arquivo, o cabeçario por assim dizer

## Tail

```
matheus@Linux-matheus: /var/log$ tail alternatives.log
update-alternatives 2024-01-27 14:05:37: link group fakeroot updated to point to /usr/bin/fakeroot-sysv
update-alternatives 2024-01-27 14:05:37: run with --install /usr/bin/fakeroot fakeroot /usr/bin/fakeroot -tcp 30 --slave /usr/share/man/man1/fakeroot.1.gz fakeroot.1.gz /usr/share/man/man1/fakeroot-tcp.1.gz --slave /usr/share/man/man1/faked.1.gz faked.1.gz /usr/share/man/man1/faked-tcp.1.gz --slave /usr/share/man/es/man1/fakeroot.1.gz fakeroot.es.1.gz /usr/share/man/es/man1/fakeroot-tcp.1.gz --slave /usr/share/man/es/man1/faked.1.gz faked.es.1.gz /usr/share/man/es/man1/faked-tcp.1.gz --slave /usr/share/man/fr/man1/fakeroot.1.gz fakeroot.fr.1.gz /usr/share/man/fr/man1/fakeroot-tcp.1.gz --slave /usr/share/man/fr/man1/faked.1.gz faked.fr.1.gz /usr/share/man/fr/man1/faked-tcp.1.gz --slave /usr/share/man/sv/man1/fakeroot.1.gz fakeroot.sv.1.gz /usr/share/man/sv/man1/fakeroot-tcp.1.gz --slave /usr/share/man/sv/man1/faked.1.gz faked.sv.1.gz /usr/share/man/sv/man1/faked-tcp.1.gz
update-alternatives 2024-01-27 14:05:37: run with --quiet --install /usr/bin/cc cc /usr/bin/gcc 20 --slave /usr/share/man/man1/cc.1.gz cc.1.gz /usr/share/man/man1/gcc.1.gz
update-alternatives 2024-01-27 14:05:37: link group cc updated to point to /usr/bin/gcc
update-alternatives 2024-01-27 14:05:37: run with --quiet --install /usr/bin/c89 c89 /usr/bin/c89-gcc 20 --slave /usr/share/man/man1/c89.1.gz c89.1.gz /usr/share/man/man1/c89-gcc.1.gz
update-alternatives 2024-01-27 14:05:37: link group c89 updated to point to /usr/bin/c89-gcc
update-alternatives 2024-01-27 14:05:37: run with --quiet --install /usr/bin/c99 c99 /usr/bin/c99-gcc 20 --slave /usr/share/man/man1/c99.1.gz c99.1.gz /usr/share/man/man1/c99-gcc.1.gz
update-alternatives 2024-01-27 14:05:37: link group c99 updated to point to /usr/bin/c99-gcc
update-alternatives 2024-01-27 14:05:37: run with --install /usr/bin/c++ c++ /usr/bin/g++ 20 --slave /usr/share/man/man1/c++.1.gz c++.1.gz /usr/share/man/man1/g++.1.gz
update-alternatives 2024-01-27 14:05:37: link group c++ updated to point to /usr/bin/g++
matheus@Linux-matheus: /var/log$
```

E enquanto o head mostra o começo do arquivo o comando tail mostra o final do arquivo

## Less

```
matheus@Linux-matheus: /var/log
update-alternatives 2023-08-07 22:52:51: run with --install /usr/bin/awk awk /usr/bin/mawk 5 --slave /usr/share/man/man1/awk.1.gz awk.1.gz /usr/share/man/man1/mawk.1.gz --slave /usr/bin/nawk nawk /usr/bin/mawk --slave /usr/share/man/man1/nawk.1.gz nawk.1.gz /usr/share/man/man1/mawk.1.gz
update-alternatives 2023-08-07 22:52:51: link group awk updated to point to /usr/bin/mawk
update-alternatives 2023-08-07 22:52:54: run with --install /usr/bin/which which /usr/bin/which.debianutils 0 --slave /usr/share/man/man1/which.1.gz which.1.gz /usr/share/man/man1/which.debianutils.1.gz --slave /usr/share/man/de/man1/which.1.gz which.de1.gz /usr/share/man/de/man1/which.debianutils.1.gz --slave /usr/share/man/es/man1/which.1.gz which.es1.gz /usr/share/man/es/man1/which.debianutils.1.gz --slave /usr/share/man/fr/man1/which.1.gz which.fr1.gz /usr/share/man/fr/man1/which.debianutils.1.gz --slave /usr/share/man/it/man1/which.1.gz which.it1.gz /usr/share/man/it/man1/which.debianutils.1.gz --slave /usr/share/man/ja/man1/which.1.gz which.ja1.gz /usr/share/man/ja/man1/which.debianutils.1.gz --slave /usr/share/man/pl/man1/which.1.gz which.pl1.gz /usr/share/man/pl/man1/which.debianutils.1.gz --slave /usr/share/man/sl/man1/which.1.gz which.sl1.gz /usr/share/man/sl/man1/which.debianutils.1.gz
update-alternatives 2023-08-07 22:52:54: link group which updated to point to /usr/bin/which.debianutils
update-alternatives 2023-08-07 22:52:54: run with --install /usr/bin/awk awk /usr/bin/mawk 5 --slave /usr/share/man/man1/awk.1.gz awk.1.gz /usr/share/man/man1/mawk.1.gz --slave /usr/bin/nawk nawk /usr/bin/mawk --slave /usr/share/man/man1/nawk.1.gz nawk.1.gz /usr/share/man/man1/mawk.1.gz
update-alternatives 2023-08-07 22:52:55: run with --install /usr/share/man/man7/builtins.7.gz builtins.7.gz /usr/share/man/man7/bash-builtins.7.gz 10
update-alternatives 2023-08-07 22:52:55: link group builtins.7.gz updated to point to /usr/share/man/man7/bash-builtins.7.gz
update-alternatives 2023-08-07 22:52:55: run with --install /usr/sbin/rmt rmt /usr/sbin/rmt-tar 50 --slave /usr/share/man/man8/rmt.8.gz rmt.8.gz /usr/share/man/man8/rmt-tar.8.gz
update-alternatives 2023-08-07 22:52:55: link group rmt updated to point to /usr/sbin/rmt-tar
update-alternatives 2023-08-07 22:52:55: run with --install /usr/bin/pager pager /bin/more 50 --slave /usr/share/man/man1/pager.1.gz pager.1.gz /usr/share/man/man1/more.1.gz
update-alternatives 2023-08-07 22:52:55: link group pager updated to point to /bin/more
update-alternatives 2023-08-07 22:53:04: run with --quiet --install /usr/bin/pager pager /usr/bin/less 77 --slave /usr/share/man/man1/pager.1.gz pager.1.gz /usr/share/man/man1/less.1.gz
update-alternatives 2023-08-07 22:53:04: link group pager updated to point to /usr/bin/less
update-alternatives 2023-08-07 22:53:06: run with --install /etc/newt/palette newt-palette /etc/newt/palette.ubuntu 50
alternatives.log
```

Caso seja necessário ler todo o conteúdo de um arquivo grande, é indicado o comando less, que irá ler o arquivo por partes e não de uma vez como o cat.

## Rm

```
matheus@Linux-matheus: ~/nova_pasta/arquivo2
matheus@Linux-matheus:~/nova_pasta/arquivo2$ ls
matheus.txt
matheus@Linux-matheus:~/nova_pasta/arquivo2$ rm matheus.txt
matheus@Linux-matheus:~/nova_pasta/arquivo2$ ls
matheus@Linux-matheus:~/nova_pasta/arquivo2$
```

O comando rm irá remover um arquivo, porém ele não exclui pastas

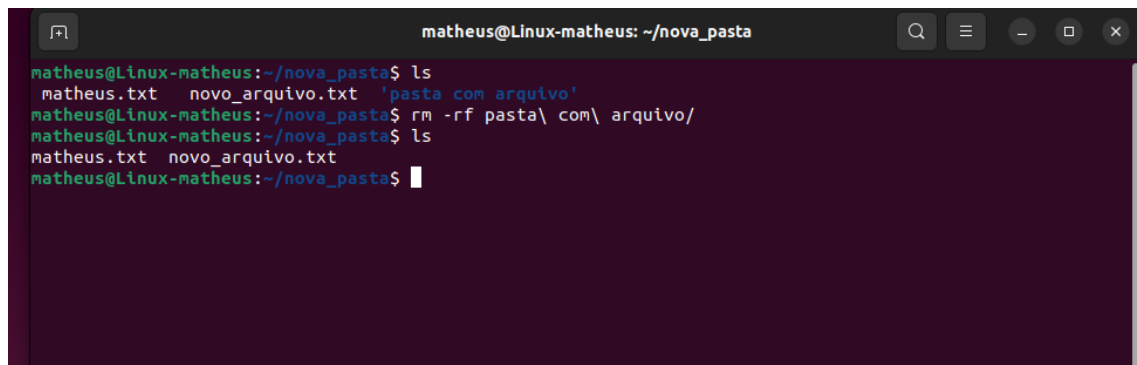
## Rmdir

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
arquivo2 matheus.txt novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ rmdir arquivo2
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$
```



Para excluir pasta é necessário utilizar o comando rmdir, porém somente se a pasta estiver vazia

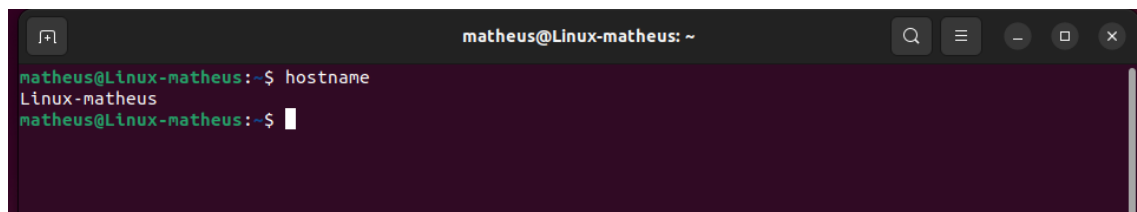
## Rm -rf



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  novo_arquivo.txt  'pasta com arquivo'
matheus@Linux-matheus:~/nova_pasta$ rm -rf pasta\ com\ arquivo/
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$
```

Para apagar pastas com arquivos é preciso usar um parâmetro do comando rm que é o -rf que irá forçar a exclusão da pasta, porém muito cuidado com esse comando.

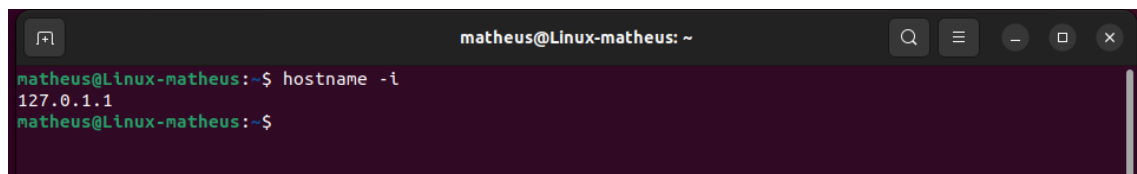
## Host name



```
matheus@Linux-matheus: ~
matheus@Linux-matheus:~$ hostname
Linux-matheus
matheus@Linux-matheus:~$
```

Esse comando serve para você conseguir saber o nome da máquina que está trabalhando

## Hostame -i



```
matheus@Linux-matheus: ~
matheus@Linux-matheus:~$ hostname -i
127.0.1.1
matheus@Linux-matheus:~$
```

Esse comando irá mostrar o ip da sua maquina

## Ip a

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ hostname -i  
127.0.1.1  
matheus@Linux-matheus:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:5e:23:30 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 81272sec preferred_lft 81272sec  
    inet6 fe80::ae65:1bab:f598:9b9/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
matheus@Linux-matheus:~$
```

Nesse caso, o comando ip a irá informar, mais detalhadamente da conexão ip do seu computador

## Grep

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ ip a | grep inet  
inet 127.0.0.1/8 scope host lo  
inet6 ::1/128 scope host  
inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3  
inet6 fe80::ae65:1bab:f598:9b9/64 scope link noprefixroute  
matheus@Linux-matheus:~$
```

O comando grep serve para fazer uma filtragem dos dados, pois muitas vezes não é necessária toda a informação.

## Ping

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ ping google.com  
PING google.com (142.250.218.174) 56(84) bytes of data:  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=1 ttl=116 time=23.4 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=2 ttl=116 time=16.3 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=3 ttl=116 time=15.9 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=4 ttl=116 time=15.4 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=5 ttl=116 time=17.3 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=6 ttl=116 time=16.1 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=7 ttl=116 time=18.7 ms  
64 bytes from gru06s62-in-f14.1e100.net (142.250.218.174): icmp_seq=8 ttl=116 time=18.5 ms  
^C  
--- google.com ping statistics ---  
8 packets transmitted, 8 received, 0% packet loss, time 7553ms  
rtt min/avg/max/mdev = 15.381/17.702/23.435/2.438 ms  
matheus@Linux-matheus:~$
```

Mesmo que tenha um ip, não é garantia que a maquina esteja conectada a internet, para isso é utilizado o comando ping que irá acessar um site e informar a velocidade de resposta.

## Free -h / free -m

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ free -h  
              total        used        free      shared  buff/cache   available  
Mem:          1,9Gi       743Mi       122Mi        39Mi        1,1Gi        1,0Gi  
Swap:         2,6Gi         0,0Ki       2,6Gi  
matheus@Linux-matheus:~$ free -m  
              total        used        free      shared  buff/cache   available  
Mem:          1958         743         122         39       1093       1030  
Swap:         2679          0       2679  
matheus@Linux-matheus:~$
```

Esses dois comandos servem, para mostrar o uso que seu computador faz do hardware, a diferença deles é que o -h imprime valores mais fáceis de entender e o -m irá imprimir em megabytes.

## Top

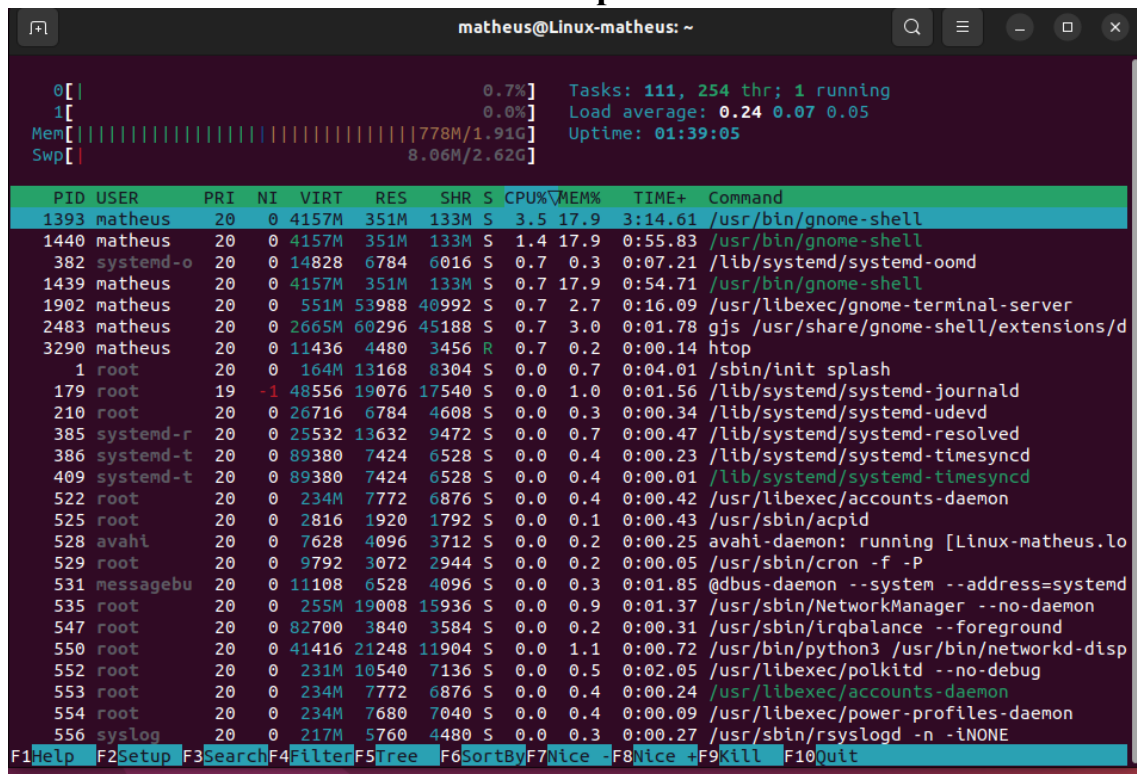
```
matheus@Linux-matheus: ~  
top - 14:41:04 up 1:36, 1 user, load average: 0,00, 0,01, 0,03  
Tasks: 179 total, 2 running, 176 sleeping, 1 stopped, 0 zombie  
%Cpu(s): 0,5 us, 0,8 sy, 0,0 ni, 98,5 id, 0,0 wa, 0,0 hi, 0,2 si, 0,0 st  
MiB Mem : 1959,0 total, 122,6 free, 742,9 used, 1093,5 buff/cache  
MiB Swap: 2680,0 total, 2680,0 free, 0,0 used. 1030,5 avail Mem  


| PID  | USER     | PR  | NI  | VIRT    | RES    | SHR    | S | %CPU | %MEM | TIME+   | COMMAND                     |
|------|----------|-----|-----|---------|--------|--------|---|------|------|---------|-----------------------------|
| 1393 | matheus  | 20  | 0   | 4253704 | 359108 | 136568 | S | 3,0  | 17,9 | 3:08.00 | gnome-shell                 |
| 58   | root     | 20  | 0   | 0       | 0      | 0      | I | 0,3  | 0,0  | 0:05.74 | kworker/u4:4-events_unbound |
| 382  | systemd+ | 20  | 0   | 14828   | 6784   | 6016   | S | 0,3  | 0,3  | 0:07.06 | systemd-oomd                |
| 1588 | matheus  | 20  | 0   | 312240  | 8320   | 7424   | S | 0,3  | 0,4  | 0:00.23 | gsd-housekeepin             |
| 2483 | matheus  | 20  | 0   | 2729580 | 60296  | 45188  | S | 0,3  | 3,0  | 0:01.73 | gjs                         |
| 3107 | root     | 20  | 0   | 0       | 0      | 0      | R | 0,3  | 0,0  | 0:00.29 | kworker/1:2-events          |
| 3113 | matheus  | 20  | 0   | 13416   | 4096   | 3456   | R | 0,3  | 0,2  | 0:00.05 | top                         |
| 1    | root     | 20  | 0   | 167992  | 13168  | 8304   | S | 0,0  | 0,7  | 0:04.01 | systemd                     |
| 2    | root     | 20  | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.01 | kthreadd                    |
| 3    | root     | 0   | -20 | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | rcu_gp                      |
| 4    | root     | 0   | -20 | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | rcu_par_gp                  |
| 5    | root     | 0   | -20 | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | slub_flushwq                |
| 6    | root     | 0   | -20 | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | netns                       |
| 8    | root     | 0   | -20 | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | kworker/0:0H-events_highpri |
| 11   | root     | 0   | -20 | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | mm_percpu_wq                |
| 12   | root     | 20  | 0   | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | rcu_tasks_kthread           |
| 13   | root     | 20  | 0   | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | rcu_tasks_rude_kthread      |
| 14   | root     | 20  | 0   | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:00.00 | rcu_tasks_trace_kthread     |
| 15   | root     | 20  | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.69 | ksoftirqd/0                 |
| 16   | root     | 20  | 0   | 0       | 0      | 0      | I | 0,0  | 0,0  | 0:03.55 | rcu_preempt                 |
| 17   | root     | rt  | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.05 | migration/0                 |
| 18   | root     | -51 | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.00 | idle_inject/0               |
| 19   | root     | 20  | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.00 | cpuhp/0                     |
| 20   | root     | 20  | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.00 | cpuhp/1                     |
| 21   | root     | -51 | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.00 | idle_inject/1               |
| 22   | root     | rt  | 0   | 0       | 0      | 0      | S | 0,0  | 0,0  | 0:00.43 | migration/1                 |


```

O comando top, traz bem mais informações do sistema que o comando free, porém com uma interface visual difícil de entender.

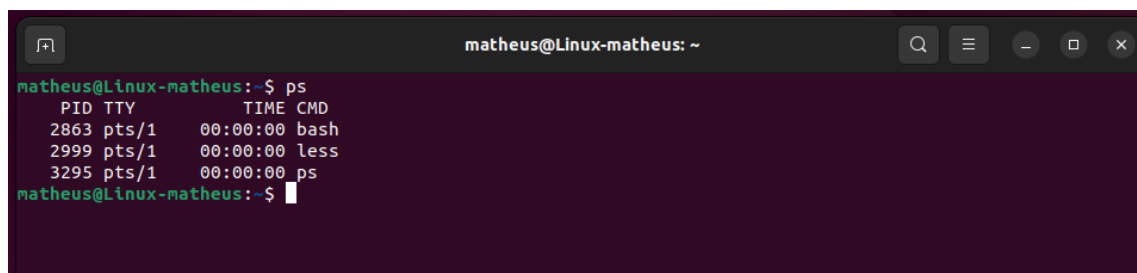
## Htop



```
matheus@Linux-matheus: ~  
0[| 0.7%] Tasks: 111, 254 thr: 1 running  
1[| 0.0%] Load average: 0.24 0.07 0.05  
Mem[| 778M/1.91G] Uptime: 01:39:05  
Swp[| 8.06M/2.62G]  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
1393 matheus 20 0 4157M 351M 133M S 3.5 17.9 3:14.61 /usr/bin/gnome-shell  
1440 matheus 20 0 4157M 351M 133M S 1.4 17.9 0:55.83 /usr/bin/gnome-shell  
382 systemd-o 20 0 14828 6784 6016 S 0.7 0.3 0:07.21 /lib/systemd/systemd-oomd  
1439 matheus 20 0 4157M 351M 133M S 0.7 17.9 0:54.71 /usr/bin/gnome-shell  
1902 matheus 20 0 551M 53988 40992 S 0.7 2.7 0:16.09 /usr/libexec/gnome-terminal-server  
2483 matheus 20 0 2665M 60296 45188 S 0.7 3.0 0:01.78 gjs /usr/share/gnome-shell/extensions/d  
3290 matheus 20 0 11436 4480 3456 R 0.7 0.2 0:00.14 htop  
1 root 20 0 164M 13168 8304 S 0.0 0.7 0:04.01 /sbin/init splash  
179 root 19 -1 48556 19076 17540 S 0.0 1.0 0:01.56 /lib/systemd/systemd-journald  
210 root 20 0 26716 6784 4608 S 0.0 0.3 0:00.34 /lib/systemd/systemd-udev  
385 systemd-r 20 0 25532 13632 9472 S 0.0 0.7 0:00.47 /lib/systemd/systemd-resolved  
386 systemd-t 20 0 89380 7424 6528 S 0.0 0.4 0:00.23 /lib/systemd/systemd-timesyncd  
409 systemd-t 20 0 89380 7424 6528 S 0.0 0.4 0:00.01 /lib/systemd/systemd-timesyncd  
522 root 20 0 234M 7772 6876 S 0.0 0.4 0:00.42 /usr/libexec/accounts-daemon  
525 root 20 0 2816 1920 1792 S 0.0 0.1 0:00.43 /usr/sbin/acpid  
528 avahi 20 0 7628 4096 3712 S 0.0 0.2 0:00.25 avahi-daemon: running [Linux-matheus.lo  
529 root 20 0 9792 3072 2944 S 0.0 0.2 0:00.05 /usr/sbin/cron -f -P  
531 messagebu 20 0 11108 6528 4096 S 0.0 0.3 0:01.85 @dbus-daemon --system --address=systemd  
535 root 20 0 255M 19008 15936 S 0.0 0.9 0:01.37 /usr/sbin/NetworkManager --no-daemon  
547 root 20 0 82700 3840 3584 S 0.0 0.2 0:00.31 /usr/sbin/irqbalance --foreground  
550 root 20 0 41416 21248 11904 S 0.0 1.1 0:00.72 /usr/bin/python3 /usr/bin/networkd-disp  
552 root 20 0 231M 10540 7136 S 0.0 0.5 0:02.05 /usr/libexec/polkitd --no-debug  
553 root 20 0 234M 7772 6876 S 0.0 0.4 0:00.24 /usr/libexec/accounts-daemon  
554 root 20 0 234M 7680 7040 S 0.0 0.4 0:00.09 /usr/libexec/power-profiles-daemon  
556 syslog 20 0 217M 5760 4480 S 0.0 0.3 0:00.27 /usr/sbin/rsyslogd -n -iNONE  
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

Para melhorar a visualização do comando top surgiu o comando htop, que mostra de maneira mais visual os dados do sistema.

## Ps



```
matheus@Linux-matheus: ~  
matheus@Linux-matheus: $ ps  
PID TTY TIME CMD  
2863 pts/1 00:00:00 bash  
2999 pts/1 00:00:00 less  
3295 pts/1 00:00:00 ps  
matheus@Linux-matheus:~$
```

O comando ps indica quais processos estão em execução nessa unidade.

## Ps aux

```
matheus@Linux-matheus: ~  
matheus 1642 0.0 0.4 386356 8704 ? Ssl 13:06 0:00 /usr/libexec/gsd-smartcard  
matheus 1644 0.0 0.4 319660 9984 ? Ssl 13:06 0:00 /usr/libexec/gsd-sound  
matheus 1655 0.0 0.3 237628 7680 ? SL 13:06 0:00 /usr/libexec/ibus-dconf  
matheus 1656 0.0 1.0 414656 21764 ? Ssl 13:06 0:00 /usr/libexec/gsd-wacom  
matheus 1659 0.0 1.4 347592 28424 ? SL 13:06 0:02 /usr/libexec/ibus-extension-gtk3  
matheus 1667 0.0 0.3 237584 7808 ? SL 13:06 0:00 /usr/libexec/ibus-portal  
matheus 1681 0.0 0.3 232268 7424 ? SL 13:06 0:00 /usr/libexec/gsd-disk-utility-notify  
matheus 1689 0.0 3.3 815204 66684 ? SL 13:06 0:00 /usr/libexec/evolution-data-server/ev  
matheus 1747 0.0 0.7 342680 15360 ? SL 13:06 0:00 /usr/libexec/gsd-printer  
matheus 1757 0.0 0.3 163804 7808 ? SL 13:06 0:01 /usr/libexec/ibus-engine-simple  
matheus 1778 0.0 1.2 344224 24956 ? Ssl 13:06 0:00 /usr/libexec/xdg-desktop-portal-gtk  
matheus 1810 0.0 1.4 2599824 28300 ? SL 13:06 0:00 /usr/bin/gjs /usr/share/gnome-shell/o  
matheus 1864 0.0 0.3 163372 6912 ? Ssl 13:07 0:00 /usr/libexec/gvfsd-metadata  
matheus 1902 0.2 2.6 564568 53988 ? Ssl 13:07 0:16 /usr/libexec/gnome-terminal-server  
matheus 1949 0.0 1.3 345516 27232 ? SL 13:07 0:00 update-notifier  
root 2468 0.0 0.0 0 0 ? I 13:15 0:00 [kworker/0:0-events]  
matheus 2483 0.0 3.0 2729580 60296 ? SL 13:15 0:01 gjs /usr/share/gnome-shell/extensions  
root 2680 0.0 0.0 0 0 ? I 13:40 0:03 [kworker/u4:2-events_unbound]  
matheus 2758 0.2 3.8 1064724 76608 ? SL 13:45 0:09 /usr/bin/nautilus --gapplication-serv  
root 2795 0.0 0.0 0 0 ? I 13:46 0:00 [kworker/1:0-mm_percpu_wq]  
matheus 2796 0.0 0.4 315272 9088 ? SL 13:46 0:00 /usr/libexec/gvfsd-recent --spawn  
matheus 2809 0.0 0.4 389096 9344 ? SL 13:46 0:00 /usr/libexec/gvfsd-network --spawn  
matheus 2824 0.0 0.4 317148 9472 ? SL 13:46 0:00 /usr/libexec/gvfsd-dnssd --spawn  
matheus 2863 0.0 0.2 11736 5632 pts/1 Ss 13:49 0:00 bash  
root 2946 0.0 0.0 0 0 ? I 14:08 0:00 [kworker/0:1-events]  
matheus 2999 0.0 0.1 9080 2816 pts/1 T 14:15 0:00 less alternatives.log  
root 3077 0.0 0.0 0 0 ? I 14:27 0:00 [kworker/u4:0-events_unbound]  
root 3107 0.0 0.0 0 0 ? I 14:35 0:00 [kworker/1:2-events]  
root 3130 0.0 0.0 0 0 ? I 14:42 0:00 [kworker/0:2-events]  
root 3289 0.0 0.0 0 0 ? I 14:43 0:00 [kworker/u4:1-events_power_efficient]  
root 3296 0.0 0.0 0 0 ? I 14:45 0:00 [kworker/1:1-events]  
matheus 3298 0.0 0.1 12996 3584 pts/1 R+ 14:46 0:00 ps aux  
matheus@Linux-matheus: $
```

Já o comando `ps aux` irá demonstrar todos os processos que estão em execução.

## kill

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ pgrep gnome-terminal  
3327  
matheus@Linux-matheus:~$ kill 3327
```

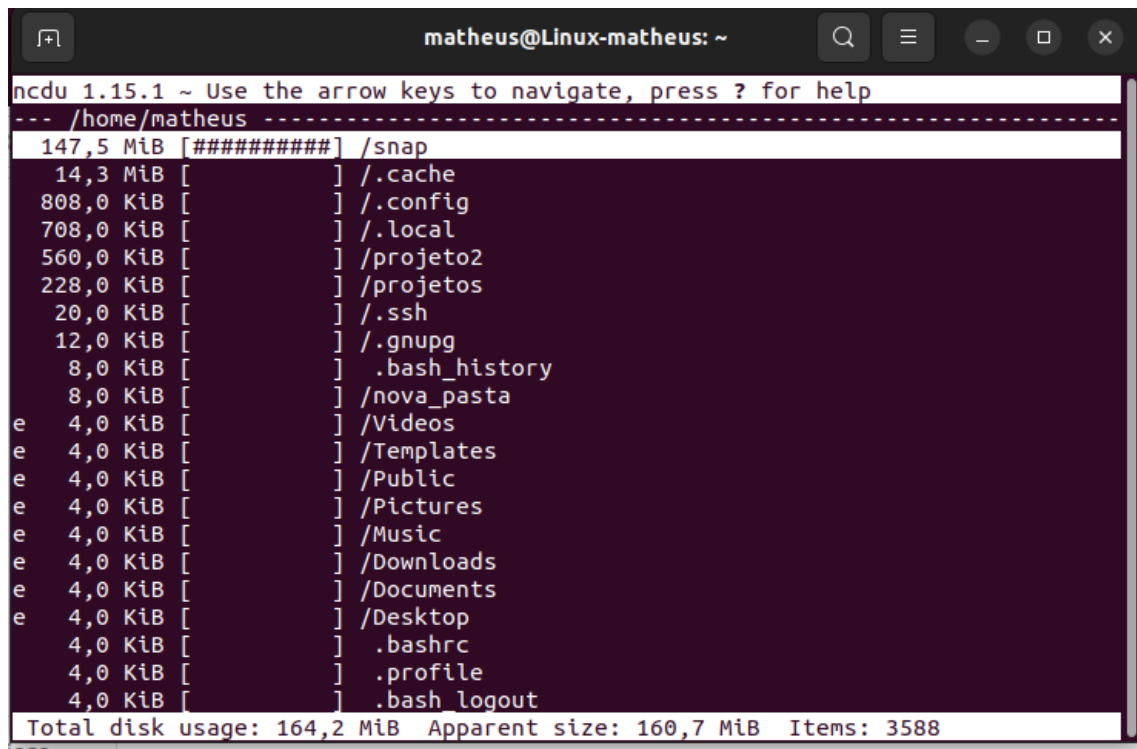
O comando `kill` irá gerenciar os processos em execução e irá fecha-lo por exemplo o comando acima irá fechar o terminal.

## Df-h

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
tmpfs            196M  1,5M  195M   1% /run  
/dev/sda3        24G   14G   8,9G  61% /  
tmpfs            980M    0  980M   0% /dev/shm  
tmpfs            5,0M   4,0K   5,0M   1% /run/lock  
/dev/sda2        512M   6,1M  506M   2% /boot/efi  
tmpfs            196M  100K   196M   1% /run/user/1000  
matheus@Linux-matheus:~$
```

Esse comando irá indicar o uso da memória quanto espaço possui e quanto está sendo utilizado

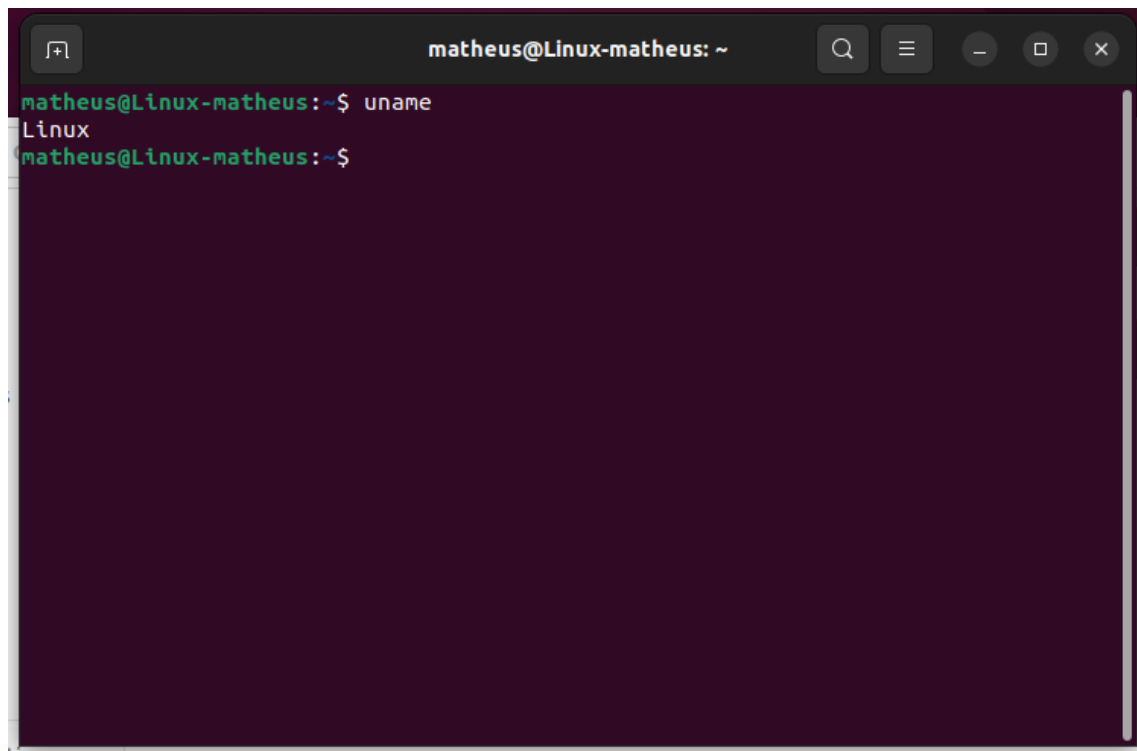
## Ncd



```
matheus@Linux-matheus: ~  
ncdu 1.15.1 ~ Use the arrow keys to navigate, press ? for help  
--- /home/matheus -----  
147,5 MiB [#####] /snap  
14,3 MiB [ ] /.cache  
808,0 KiB [ ] /.config  
708,0 KiB [ ] /.local  
560,0 KiB [ ] /projeto2  
228,0 KiB [ ] /projetos  
20,0 KiB [ ] /.ssh  
12,0 KiB [ ] /.gnupg  
8,0 KiB [ ] .bash_history  
8,0 KiB [ ] /nova_pasta  
4,0 KiB [ ] /Videos  
4,0 KiB [ ] /Templates  
4,0 KiB [ ] /Public  
4,0 KiB [ ] /Pictures  
4,0 KiB [ ] /Music  
4,0 KiB [ ] /Downloads  
4,0 KiB [ ] /Documents  
4,0 KiB [ ] /Desktop  
4,0 KiB [ ] .bashrc  
4,0 KiB [ ] .profile  
4,0 KiB [ ] .bash_logout  
Total disk usage: 164,2 MiB Apparent size: 160,7 MiB Items: 3588
```

Esse comando irá demonstrar o uso da memória também porém agora de forma mais visual.

## Uname



```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ uname  
Linux  
matheus@Linux-matheus:~$
```

Esse comando é usado para saber informações do kernel do sistema operacional

## Iscpu

```
matheus@Linux-matheus: ~  
L1d:          96 KiB (2 instances)  
L1i:          64 KiB (2 instances)  
L2:           2,5 MiB (2 instances)  
L3:           16 MiB (2 instances)  
NUMA:  
  NUMA node(s):      1  
  NUMA node0 CPU(s): 0,1  
Vulnerabilities:  
  Gather data sampling: Not affected  
  Itlb multihit:      Not affected  
  L1tf:               Not affected  
  Mds:                Not affected  
  Meltdown:           Not affected  
  Mmio stale data:    Not affected  
  Retbleed:           Mitigation; Enhanced IBRS  
  Spec rstack overflow: Not affected  
  Spec store bypass:  Vulnerable  
  Spectre v1:         Mitigation; usercopy/swapgs barriers and __user pointer sanitization  
  Spectre v2:         Mitigation; Enhanced / Automatic IBRS, RSB filling, PBR SB-eIBRS SW sequence  
  Srbds:              Not affected  
  Tsx async abort:    Not affected  
matheus@Linux-matheus:~$
```

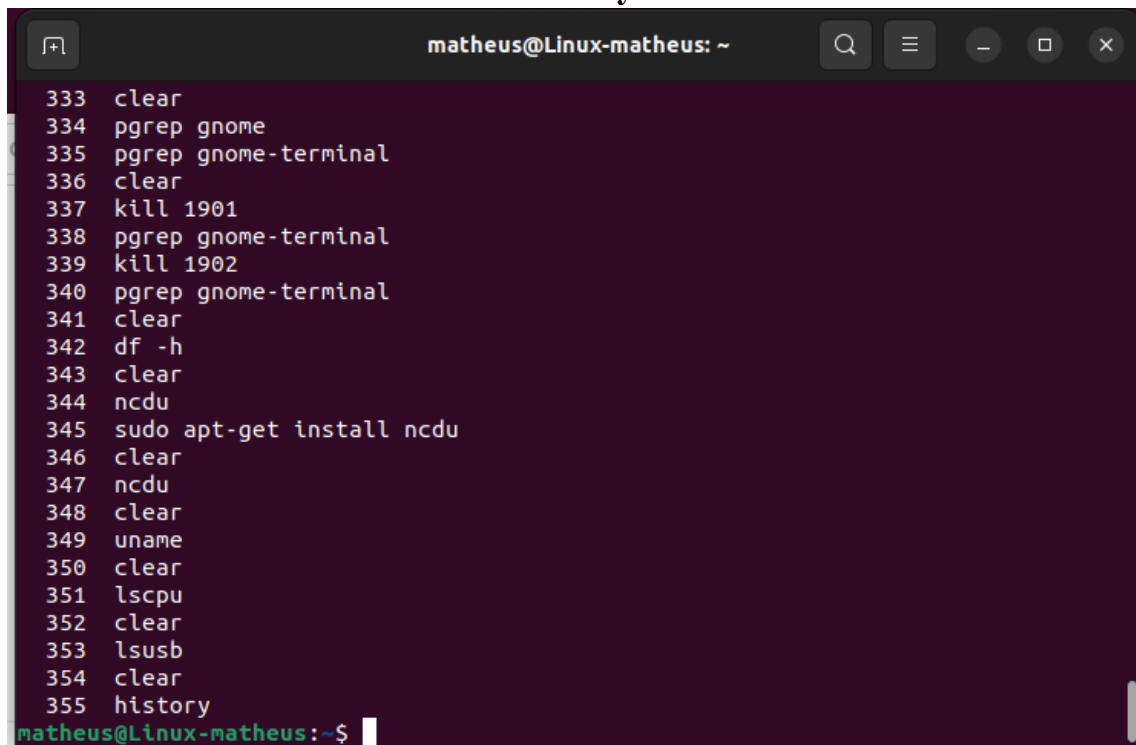
Esse comando mostra informações a respeito do processador

## Isusb

```
matheus@Linux-matheus: ~  
matheus@Linux-matheus:~$ lsusb  
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet  
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub  
matheus@Linux-matheus:~$
```

Esse comando demonstra informações das conexões usb

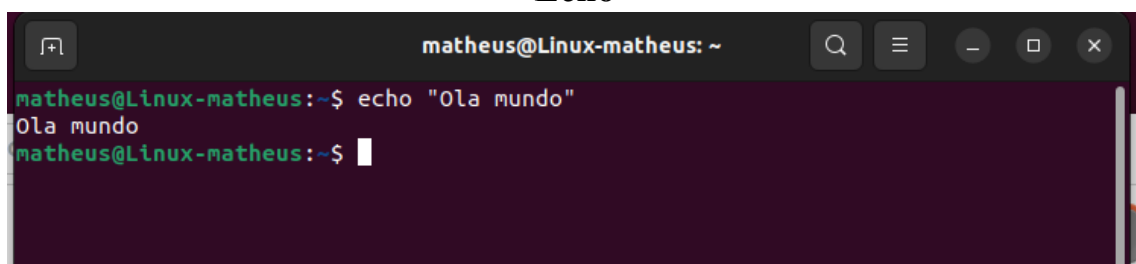
## History

A terminal window titled 'matheus@Linux-matheus: ~' with a search icon, a menu icon, and window control buttons. The terminal displays a list of 13 commands with their corresponding line numbers from 333 to 355. The commands are: clear, pgrep gnome, pgrep gnome-terminal, clear, kill 1901, pgrep gnome-terminal, kill 1902, pgrep gnome-terminal, clear, df -h, clear, ncdu, sudo apt-get install ncdu, clear, ncdu, clear, uname, clear, lscpu, clear, lsusb, clear, and history. The prompt 'matheus@Linux-matheus:~\$' is visible at the bottom.

```
333 clear
334 pgrep gnome
335 pgrep gnome-terminal
336 clear
337 kill 1901
338 pgrep gnome-terminal
339 kill 1902
340 pgrep gnome-terminal
341 clear
342 df -h
343 clear
344 ncdu
345 sudo apt-get install ncdu
346 clear
347 ncdu
348 clear
349 uname
350 clear
351 lscpu
352 clear
353 lsusb
354 clear
355 history
matheus@Linux-matheus:~$
```

History, é o comando que guarda todos os comandos utilizados e demonstrar toda sua atividade no terminal.

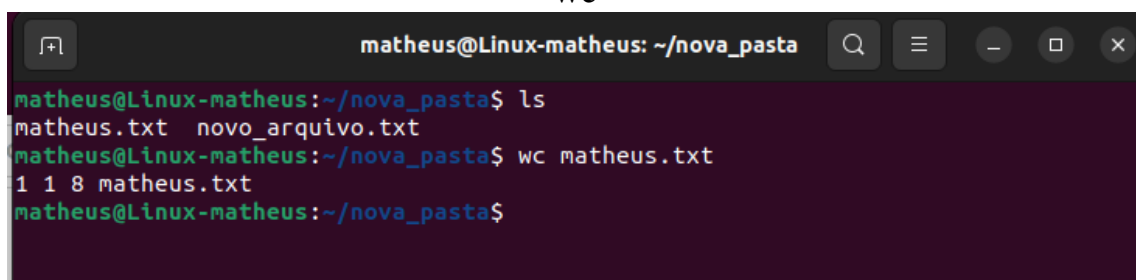
## Echo

A terminal window titled 'matheus@Linux-matheus: ~' with a search icon, a menu icon, and window control buttons. The terminal shows the command 'echo "Ola mundo"' being entered and executed, resulting in the output 'Ola mundo'. The prompt 'matheus@Linux-matheus:~\$' is visible at the bottom.

```
matheus@Linux-matheus:~$ echo "Ola mundo"
Ola mundo
matheus@Linux-matheus:~$
```

O comando echo no Linux é um comando interno que permite aos usuários exibir linhas de texto ou cadeias de caracteres que são passadas como argumentos.

## wc

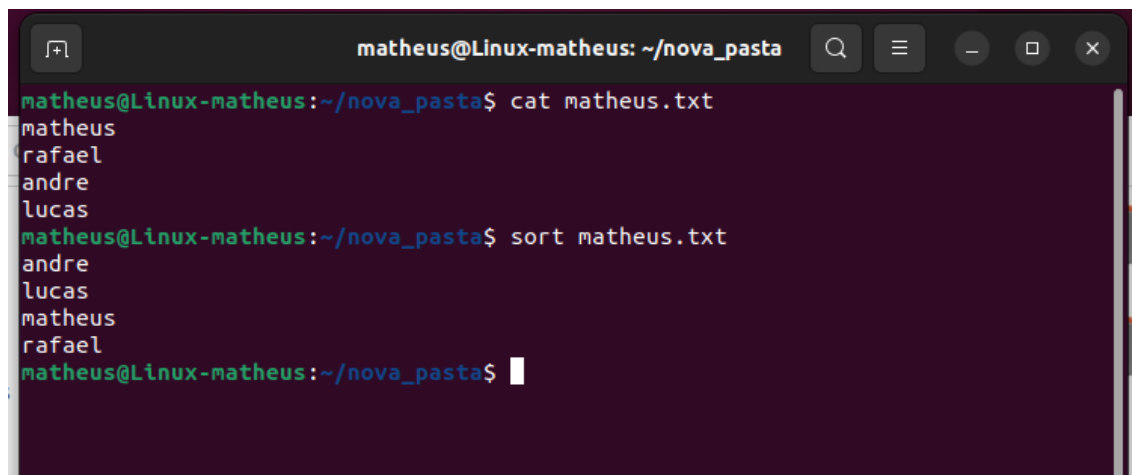
A terminal window titled 'matheus@Linux-matheus: ~/nova\_pasta' with a search icon, a menu icon, and window control buttons. The terminal shows the user navigating to the directory '~/nova\_pasta' and listing files with 'ls', showing 'matheus.txt' and 'novo\_arquivo.txt'. Then, the user runs 'wc matheus.txt', which outputs '1 1 8 matheus.txt'. The prompt 'matheus@Linux-matheus:~/nova\_pasta\$' is visible at the bottom.

```
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ wc matheus.txt
1 1 8 matheus.txt
matheus@Linux-matheus:~/nova_pasta$
```



wc significa contagem de palavras. Como o nome indica, é usado principalmente para fins de contagem. Seus dados apresentam número de linhas, contagem de palavras, contagem de bytes e caracteres.

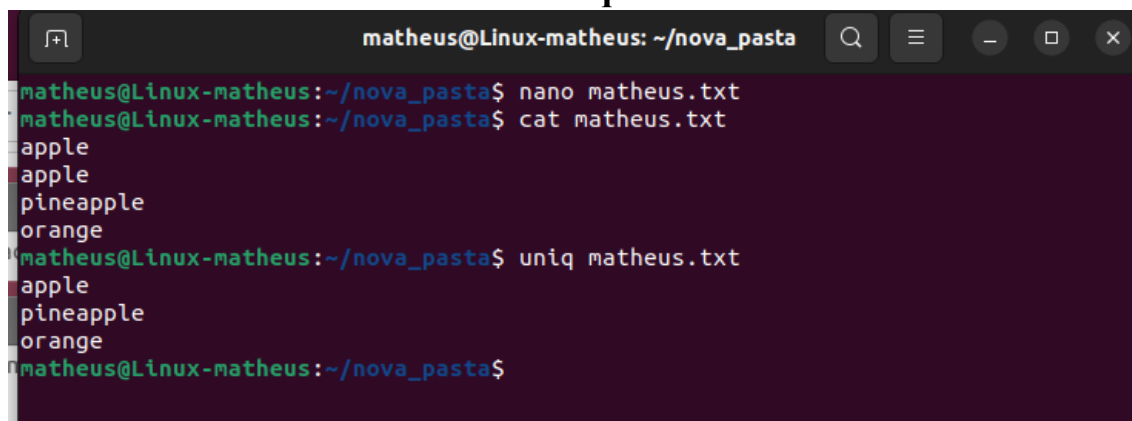
## Sort

A terminal window titled 'matheus@Linux-matheus: ~/nova\_pasta' showing the execution of the 'sort' command. The user first runs 'cat matheus.txt' which displays the names 'matheus', 'rafael', 'andre', and 'lucas'. Then, the user runs 'sort matheus.txt', which displays the names in alphabetical order: 'andre', 'lucas', 'matheus', and 'rafael'.

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ cat matheus.txt
matheus
rafael
andre
lucas
matheus@Linux-matheus:~/nova_pasta$ sort matheus.txt
andre
lucas
matheus
rafael
matheus@Linux-matheus:~/nova_pasta$
```

O comando Sort no Linux é usado para classificar o conteúdo dos arquivos de texto.

## uniq

A terminal window titled 'matheus@Linux-matheus: ~/nova\_pasta' showing the use of the 'uniq' command. The user creates a file 'matheus.txt' using 'nano matheus.txt' and then runs 'cat matheus.txt' which shows 'apple', 'apple', 'pineapple', and 'orange'. Finally, the user runs 'uniq matheus.txt', which displays only the unique lines: 'apple', 'pineapple', and 'orange'.

```
matheus@Linux-matheus:~/nova_pasta$ nano matheus.txt
matheus@Linux-matheus:~/nova_pasta$ cat matheus.txt
apple
apple
pineapple
orange
matheus@Linux-matheus:~/nova_pasta$ uniq matheus.txt
apple
pineapple
orange
matheus@Linux-matheus:~/nova_pasta$
```

O comando uniq no Linux remove as linhas duplicadas de um arquivo ordenado. Por isso ele é muito usado em conjunto com o comando sort.

## Diff

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ diff matheus.txt nomes.txt
1,4c1,5
< apple
< apple
< pineapple
< orange
---
> Jupiter
> Saturno
> Urano
> Mercurio
> Terra
matheus@Linux-matheus:~/nova_pasta$
```

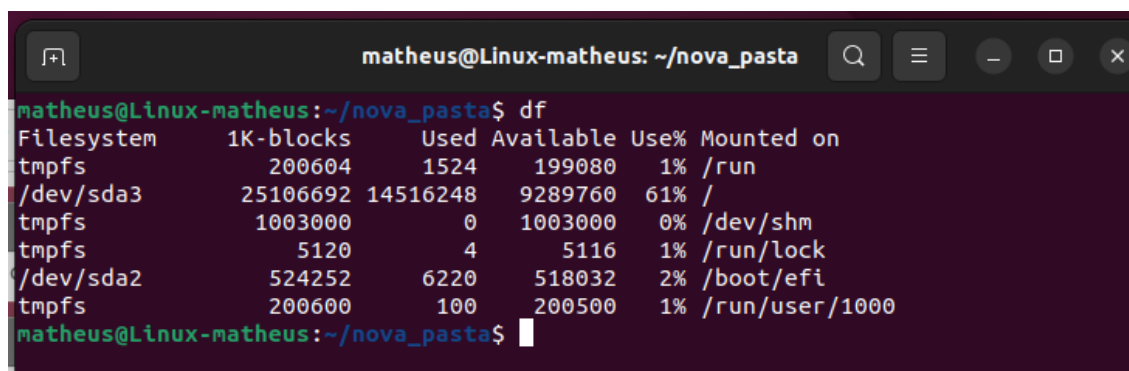
O comando diff no Linux serve para mostrar a diferença entre o conteúdo de um arquivo e outro. Muito útil por exemplo para comparar arquivos de configuração.

## Du

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ du -h /home
4,0K    /home/matheus/Pictures
4,0K    /home/matheus/Templates
12K     /home/matheus/.local/share/keyrings
4,0K    /home/matheus/.local/share/applications
4,0K    /home/matheus/.local/share/icc
24K     /home/matheus/.local/share/Trash/info
4,0K    /home/matheus/.local/share/Trash/files/Semana01
4,0K    /home/matheus/.local/share/Trash/files/teste01
4,0K    /home/matheus/.local/share/Trash/files/Area de Trabalho
4,0K    /home/matheus/.local/share/Trash/files/Semana03
4,0K    /home/matheus/.local/share/Trash/files/Semana02
24K     /home/matheus/.local/share/Trash/files
52K     /home/matheus/.local/share/Trash
4,0K    /home/matheus/.local/share/nautilus/scripts
356K    /home/matheus/.local/share/nautilus/tags
364K    /home/matheus/.local/share/nautilus
76K     /home/matheus/.local/share/gvfs-metadata
8,0K    /home/matheus/.local/share/gnome-shell
4,0K    /home/matheus/.local/share/evolution/tasks/trash
8,0K    /home/matheus/.local/share/evolution/tasks/system
16K     /home/matheus/.local/share/evolution/tasks
4,0K    /home/matheus/.local/share/evolution/mail/trash
```

O comando du no Linux (disk usage) vai fornecer uma lista detalhada sobre a utilização do disco.

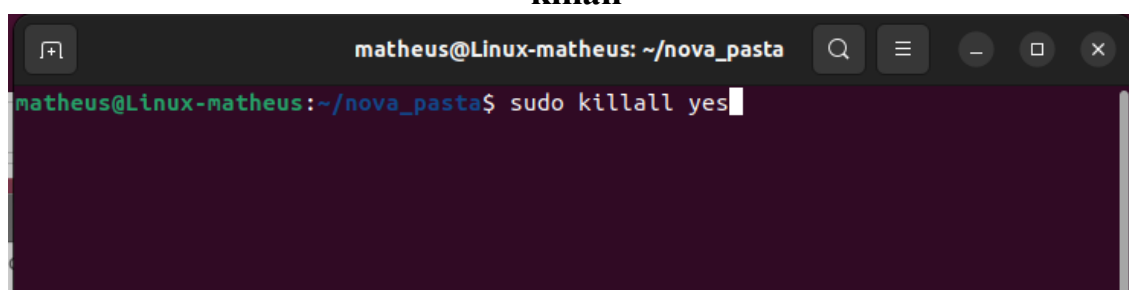
## Df



```
matheus@Linux-matheus: ~/nova_pasta$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs            200604      1524    199080    1% /run
/dev/sda3       25106692 14516248    9289760   61% /
tmpfs           1003000        0    1003000    0% /dev/shm
tmpfs            5120         4      5116    1% /run/lock
/dev/sda2       524252     6220    518032    2% /boot/efi
tmpfs           200600      100    200500    1% /run/user/1000
matheus@Linux-matheus:~/nova_pasta$
```

O comando df no Linux (disk free) mostra a capacidade utilizada de um sistema de arquivos em termos de espaço e inodes.

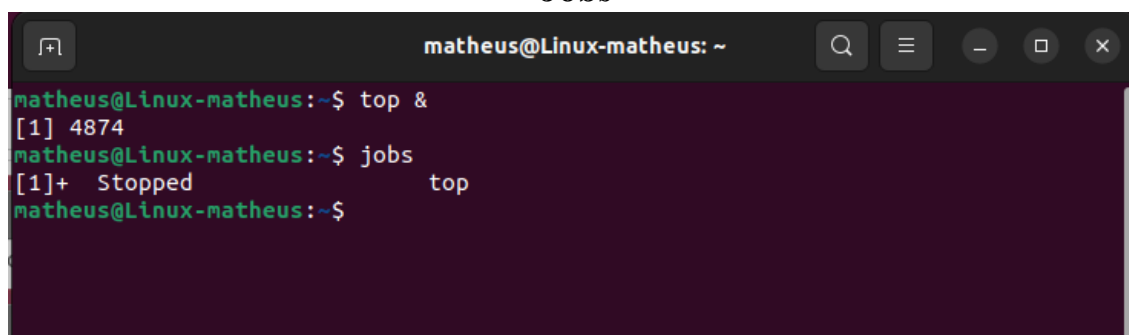
## killall



```
matheus@Linux-matheus: ~/nova_pasta$ sudo killall yes
```

A diferença de kill e killall é que o "kill" termina os ciclos de processo dependentes do número de ID do processo, já os killall terminam os ciclos de execução dependentes de seus nomes e atributos diferentes.

## Jobs



```
matheus@Linux-matheus:~$ top &
[1] 4874
matheus@Linux-matheus:~$ jobs
[1]+  Stopped                  top
matheus@Linux-matheus:~$
```

O Comando jobs no Linux lista os processos que estão em execução em segundo plano. Se um número da tarefa é fornecido o comando retornará as informações pertinentes somente à tarefa em questão.

## Bg

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ bg top
[1]+ top &          (wd: ~)

top: failed tty set: Interrupted system call
matheus@Linux-matheus:~/nova_pasta$ ^C
[1]+  Exit 1          top (wd: ~)
(wd now: ~/nova_pasta)
matheus@Linux-matheus:~/nova_pasta$
```

O comando bg no Linux coloca em segundo plano um processo em execução que foi “congelado” pelo sinal SIGTSTP.

## Fg

```
matheus@Linux-matheus: ~
matheus@Linux-matheus:~$ jobs
[1]+  Stopped          top
matheus@Linux-matheus:~$ fg jobs
bash: fg: jobs: no such job
matheus@Linux-matheus:~$ jobs
[1]+  Stopped          top
matheus@Linux-matheus:~$ fg 1
top

top: failed tty set: Interrupted system call
matheus@Linux-matheus:~$ jobs
matheus@Linux-matheus:~$
```

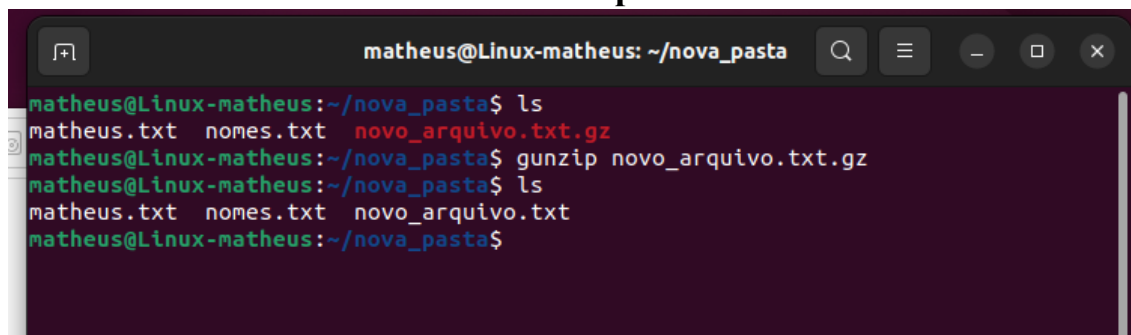
O Comando fg no Linux faz exatamente o oposto do comando bg, colocando a tarefa ou processo em segundo plano em primeiro plano e ligada a um terminal.

## Gzip

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ gzip novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt  novo_arquivo.txt.gz
matheus@Linux-matheus:~/nova_pasta$
```

O comando gzip compacta arquivos. Cada arquivo único é compactado em um único arquivo. O arquivo compactado consiste em um cabeçalho zip GNU e dados deflacionados.

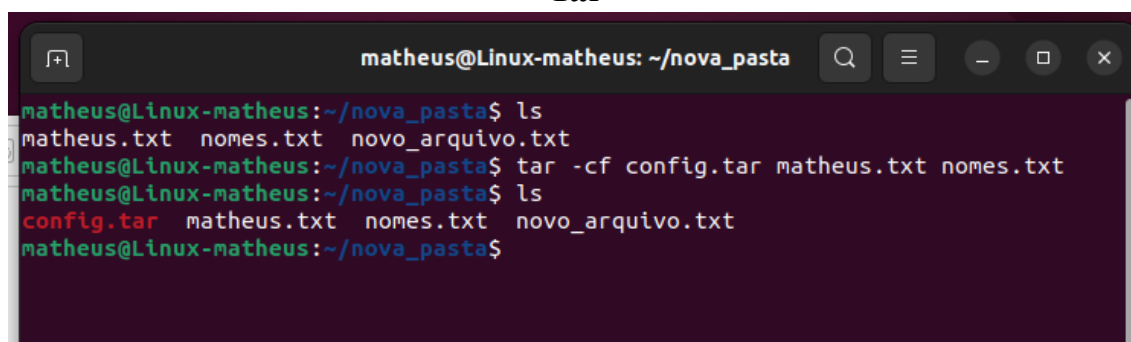
## Gunzip



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt  novo_arquivo.txt.gz
matheus@Linux-matheus:~/nova_pasta$ gunzip novo_arquivo.txt.gz
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$
```

gunzip comando é usado para descompactar ou expandir um arquivo ou uma lista de arquivos no Linux. Ele aceita todos os arquivos com extensão como .gz, .Z, \_Z, -gz, -Z, .Z, .taz ou .tgz.

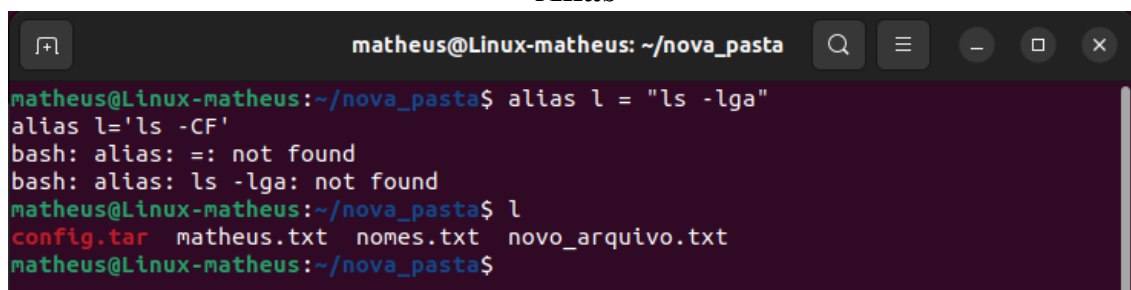
## Tar



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$ tar -cf config.tar matheus.txt nomes.txt
matheus@Linux-matheus:~/nova_pasta$ ls
config.tar  matheus.txt  nomes.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$
```

O comando tar no Linux serve para criar arquivamentos de arquivos no Linux. Estes arquivamentos vêm do tempo que se usava fitas de backup para criar cópias de segurança de dados.

## Alias



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ alias l = "ls -lga"
alias l='ls -CF'
bash: alias: ==: not found
bash: alias: ls -lga: not found
matheus@Linux-matheus:~/nova_pasta$ l
config.tar  matheus.txt  nomes.txt  novo_arquivo.txt
matheus@Linux-matheus:~/nova_pasta$
```

Existe a possibilidade de criarmos apelidos ou atalhos no Linux para os comandos. Ele pode ser utilizado para simplificar comandos com muitas opções. O comando alias cria estes atalhos.

## Xargs

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
matheus.txt  nomes.txt
matheus@Linux-matheus:~/nova_pasta$ cat nomes.txt | xargs mkdir
matheus@Linux-matheus:~/nova_pasta$ ls
Jupiter      matheus.txt  nomes.txt    Saturno
matheus@Linux-matheus:~/nova_pasta$
```

O comando xargs no Linux executa o comando ou programa e passa como argumento para esse comando o que foi recebido como entrada padrão.

## Ln

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ ls
Jupiter      matheus.txt  nomes.txt    Saturno
matheus@Linux-matheus:~/nova_pasta$ ln -s /Saturno/informacoes planeta1
matheus@Linux-matheus:~/nova_pasta$ ls
Jupiter      matheus.txt  nomes.txt    planeta1      Saturno
matheus@Linux-matheus:~/nova_pasta$ ls -lga
total 24
drwxrwxr-x  4 matheus 4096 jan 28 16:24 .
drwxr-x--- 19 matheus 4096 jan 28 14:17 ..
drwxrwxr-x  2 matheus 4096 jan 28 16:19 Jupiter
-rw-rw-r--  1 matheus  29 jan 28 15:24 matheus.txt
-rw-rw-r--  1 matheus  16 jan 28 16:19 nomes.txt
lrwxrwxrwx  1 matheus  20 jan 28 16:24 planeta1 -> /Saturno/informacoes
drwxrwxr-x  3 matheus 4096 jan 28 16:23 Saturno
matheus@Linux-matheus:~/nova_pasta$
```

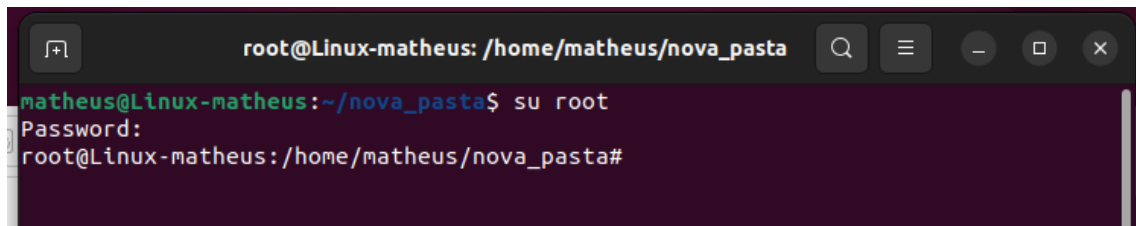
O comando ln no Linux pode ser usado para criar links simbólicos ou físicos.

## Who

```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ who
matheus  tty2          2024-01-28 13:06 (tty2)
matheus@Linux-matheus:~/nova_pasta$
```

O comando who imprime informações sobre todos os usuários que estão conectados no momento.

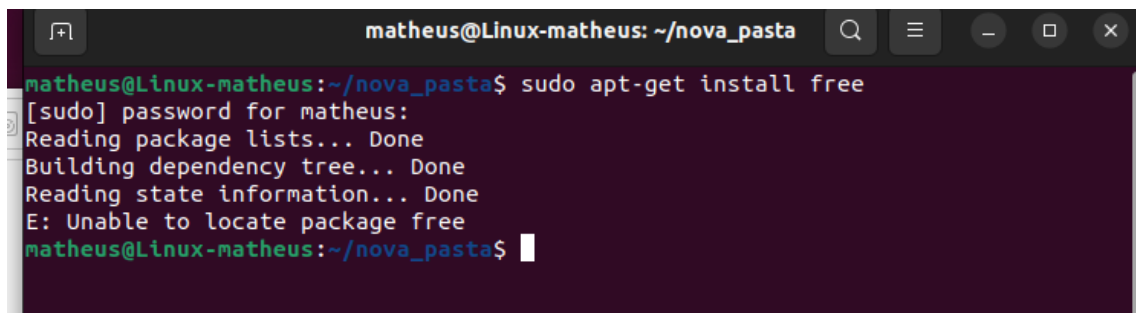
## Su



```
root@Linux-matheus: /home/matheus/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ su root
Password:
root@Linux-matheus: /home/matheus/nova_pasta#
```

O comando su, permite que o usuário atual atue como outro usuário durante a sessão de login atual .

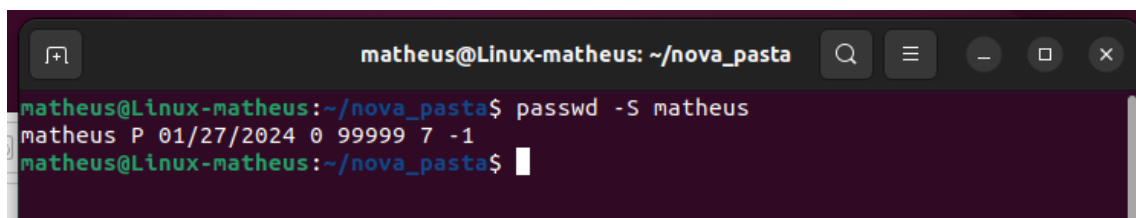
## Sudo



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ sudo apt-get install free
[sudo] password for matheus:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package free
matheus@Linux-matheus:~/nova_pasta$
```

O sudo permite que um usuário permitido execute um comando como outro usuário, de acordo com as especificações no arquivo / etc / sudoers.

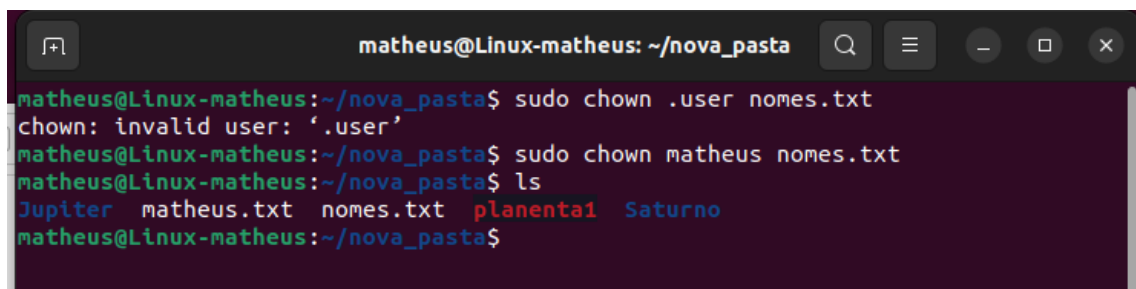
## Passwd



```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ passwd -S matheus
matheus P 01/27/2024 0 99999 7 -1
matheus@Linux-matheus:~/nova_pasta$
```

O comando passwd no Linux entra com a senha para uma determinada conta de usuário. Se a conta de usuário não for fornecida como parâmetro o passwd muda a senha da conta utilizada no momento.

## Chown



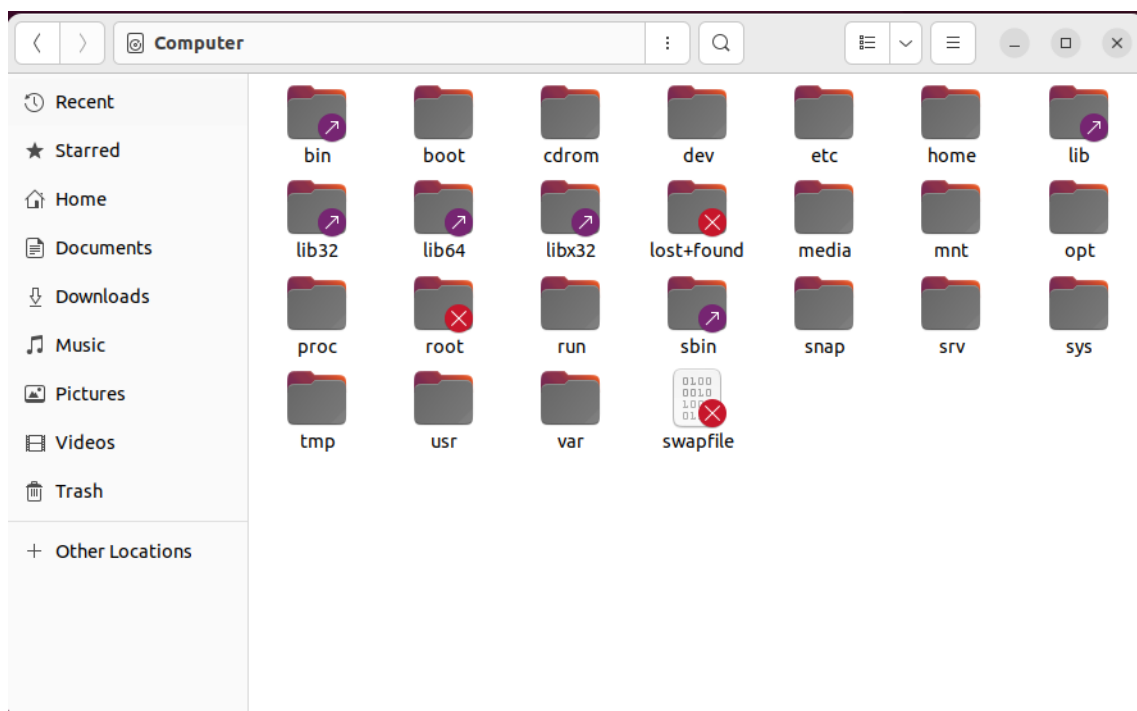
```
matheus@Linux-matheus: ~/nova_pasta
matheus@Linux-matheus:~/nova_pasta$ sudo chown .user nomes.txt
chown: invalid user: '.user'
matheus@Linux-matheus:~/nova_pasta$ sudo chown matheus nomes.txt
matheus@Linux-matheus:~/nova_pasta$ ls
Jupyter matheus.txt nomes.txt planenta1 Saturno
matheus@Linux-matheus:~/nova_pasta$
```

O comando chown no Linux altera o dono do arquivo e pode também alterar o grupo a que este arquivo pertence.

### Questão 3

Para entender o sistema de diretórios do Linux, é importante ressaltar que o Linux organiza seus diretórios em forma de árvore, basicamente ele funciona com a estrutura FHS é um acrônimo para File Sytem Hierarchy, que é um padrão que todas as distribuições Linux devem seguir para organizar os seus diretórios. Quando o Linux estava começando, viu-se diferentes formas para estruturar as pastas dentre as distribuições, o que causava muita confusão quando um usuário mudava de uma para a outra. Então, foi definido este padrão ao qual as distribuições seguem restritamente. O primeiro diretório, conhecido por barra, ou Root File System, é a raiz do nosso sistema. Isso significa que todos os outros diretórios estarão dentro dele. Antes de partir para explicação dos diretórios é importante destacar a simbologia do Linux. O símbolo da seta representa atalhos, ou seja, ao entrar neles você está na verdade entrando em outro diretório, enquanto o símbolo de x demonstra arquivos que só podem ser acessados pelo administrador do sistema, ou seja, o usuário root.

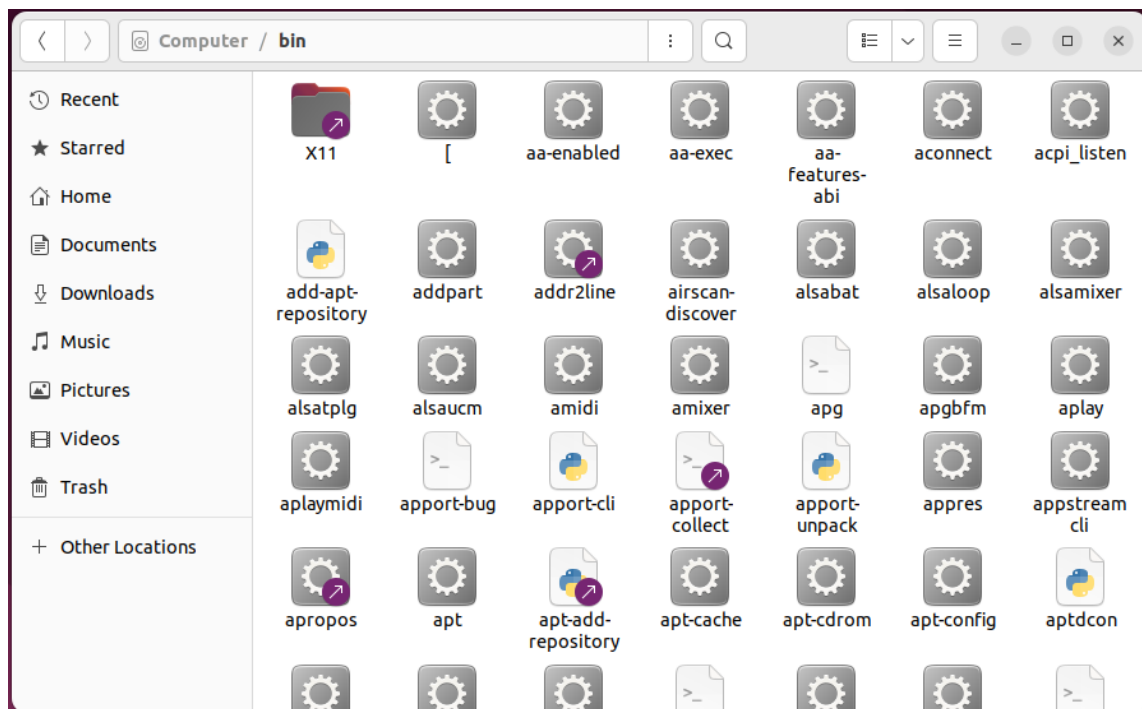
Diretório raiz



É importante ressaltar os principais diretórios também, o primeiro e um dos mais importantes é o diretório /bin que contém os binários executáveis essenciais para o funcionamento do sistema, que são necessários para o usuário em modo único e para a inicialização do sistema. Exemplos de comandos encontrados neste diretório incluem: ls, cp, mv, rm, cat, etc.

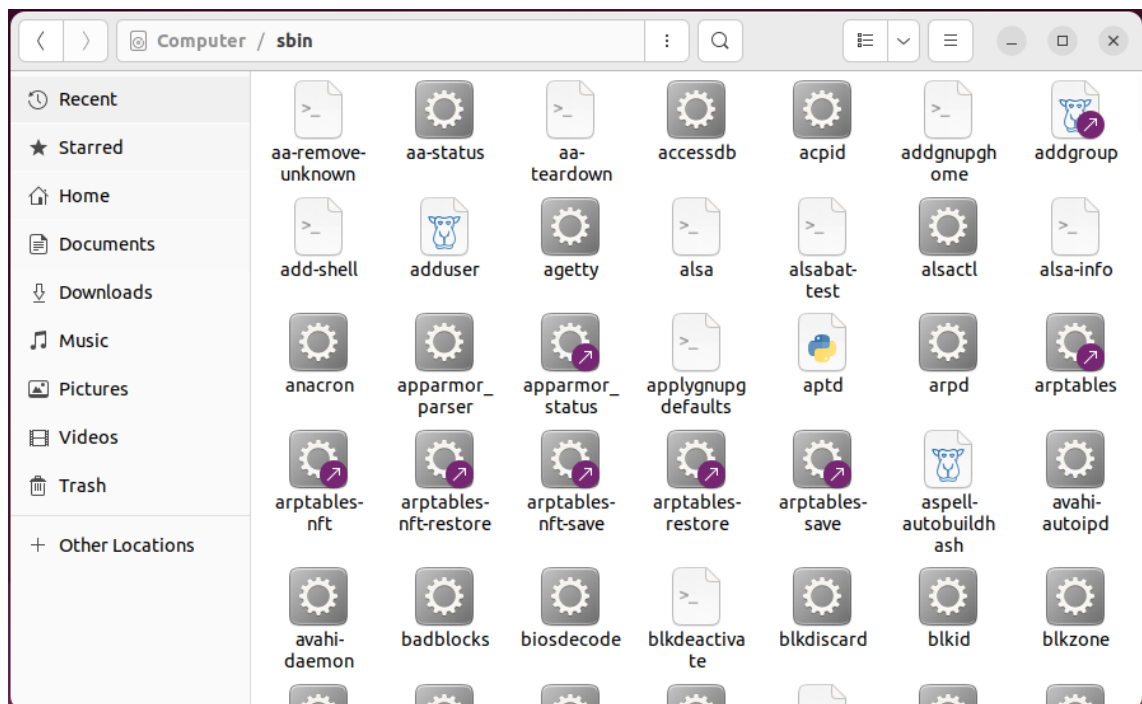


## Diretório /bin



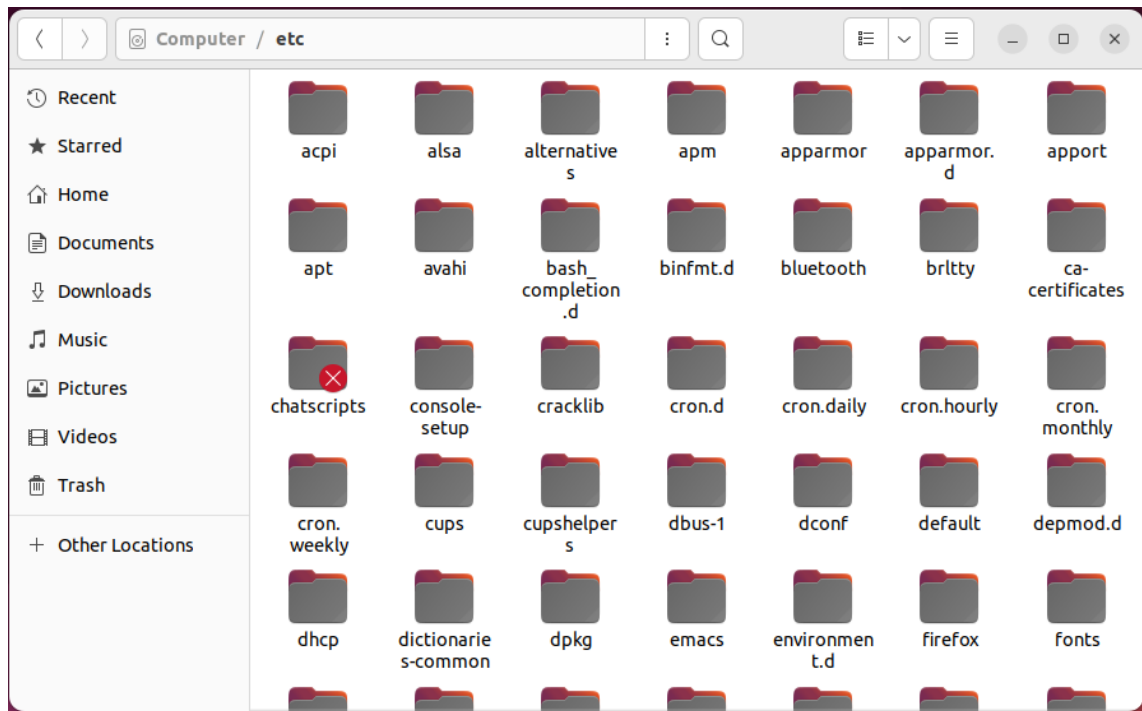
Um diretório bem semelhante ao /bin é o /sbin que também contém binários executáveis. No entanto, esses são usados principalmente para tarefas de administração do sistema, como montagem de volumes, inicialização do sistema e outras tarefas relacionadas ao superusuário.

## Diretório /sbin



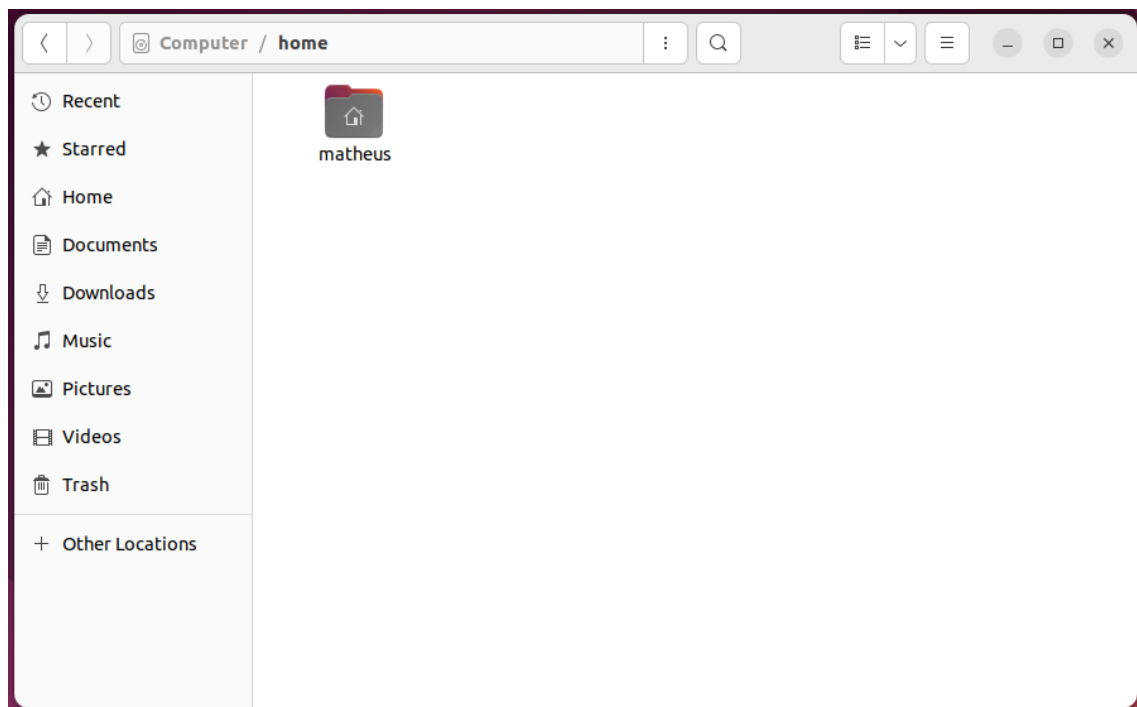
Um diretório que possui muita discussão em sua nomenclatura é o /etc, porém apesar do nome este diretório é dedicado a arquivos de configuração de sistema e scripts de inicialização do sistema. Todos os dados neste diretório são somente leitura para os usuários, mas o superusuário tem permissões de gravação.

### Diretório /etc



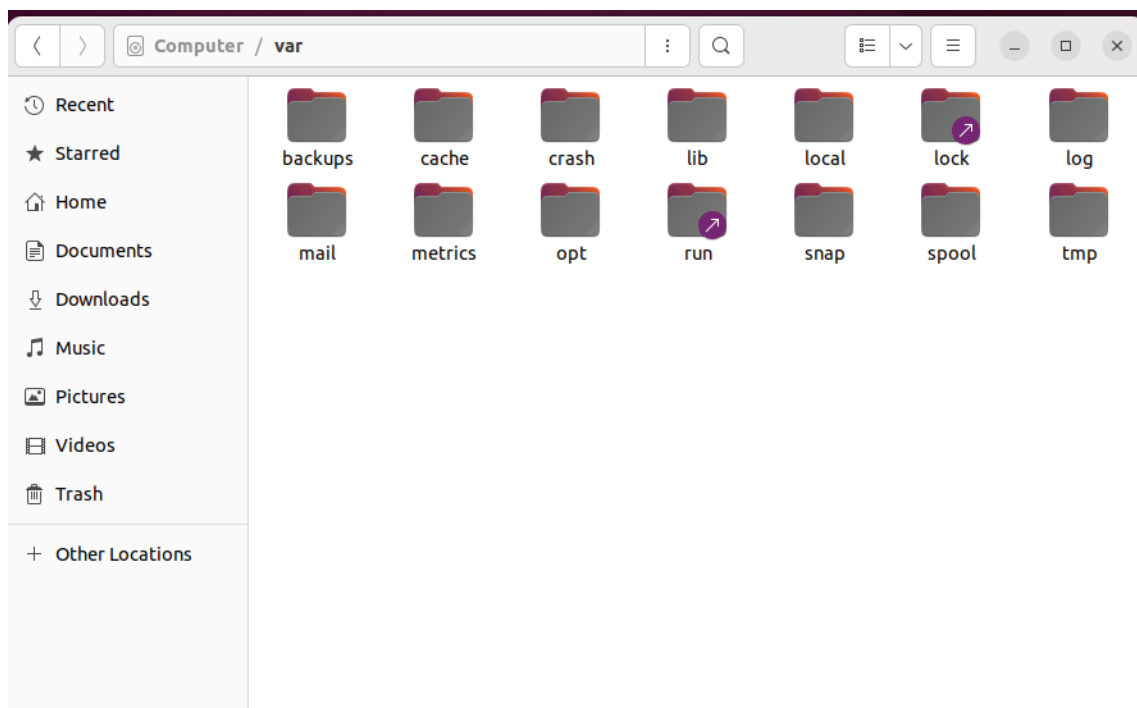
O diretório, mais comum para os usuários Linux é o home, ele pode ser comparado a área de trabalho, pois esse é o diretório onde os usuários têm suas pastas pessoais. Cada usuário do sistema tem uma pasta aqui, que é o diretório padrão ao iniciar a sessão. Aqui, os usuários podem armazenar arquivos pessoais, baixar arquivos, etc.

## Diretório /home



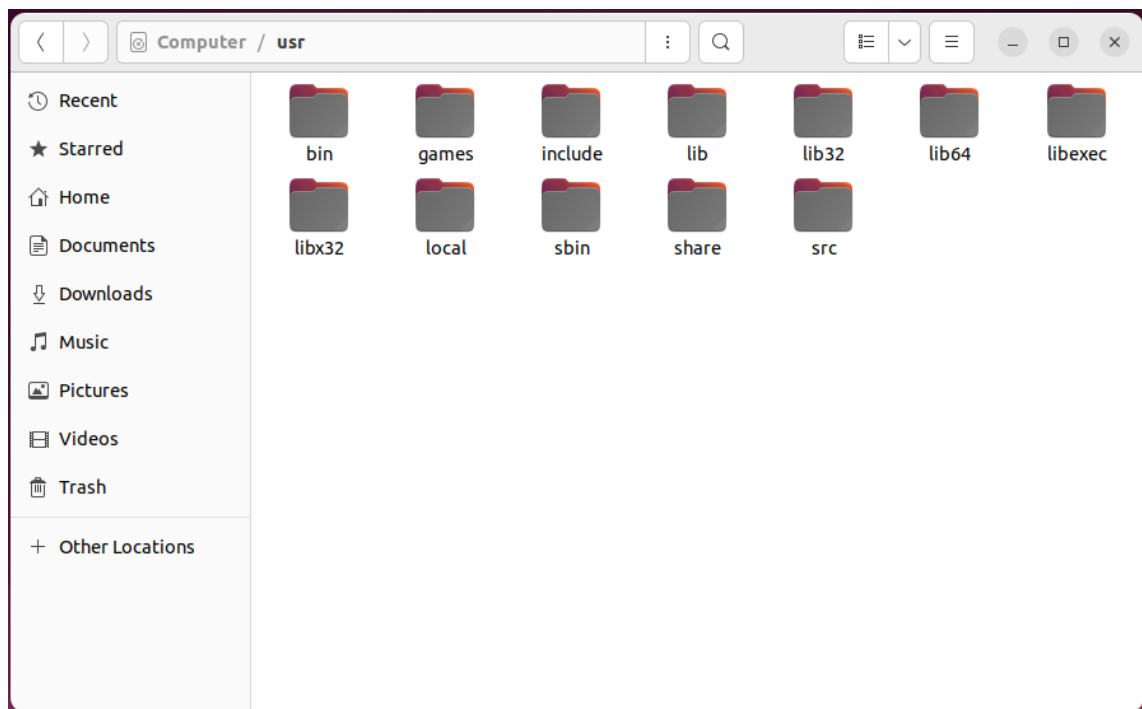
O próximo diretório importante é o /var, este diretório contém arquivos cujo conteúdo deve ser preservado entre reinicializações do sistema. Inclui spool para tarefas de email e impressão, arquivos de log do sistema e arquivos temporários.

## Diretório /var



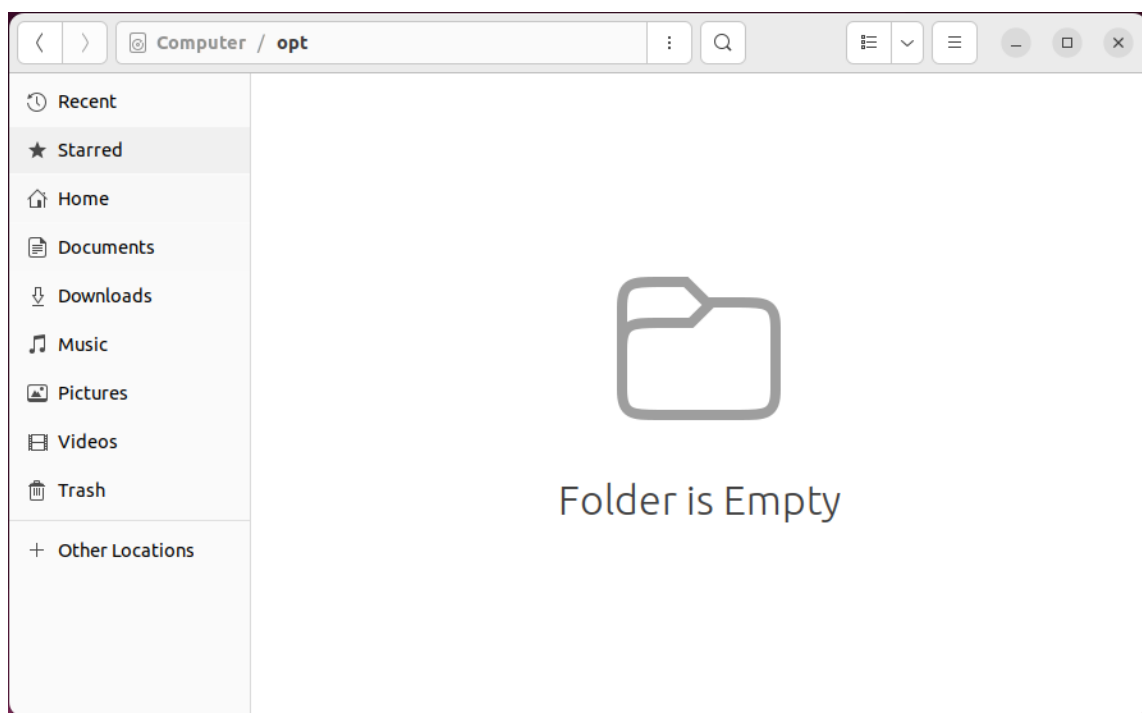
Continuando a lista temos o /usr que é um diretório, onde são armazenados os arquivos compartilhados pelos usuários, incluindo a biblioteca de sistema, a documentação do sistema e a maioria dos binários.

## Diretório /usr



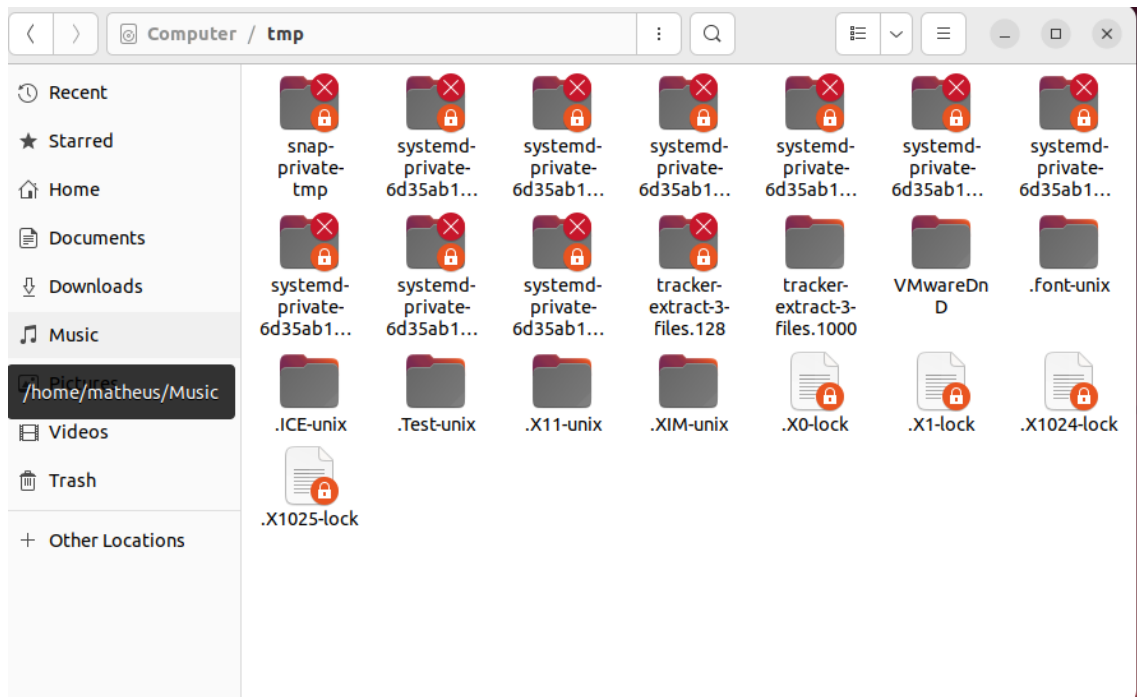
Por conseguinte, temos o diretório /opt que é reservado para a instalação de software e pacotes adicionais que não fazem parte da instalação padrão do Linux. Ele permite aos desenvolvedores de software colocar seus aplicativos em um lugar específico, sem interferir com o resto do sistema, para usuários normais ele costuma ser vazio.

## Diretório /opt



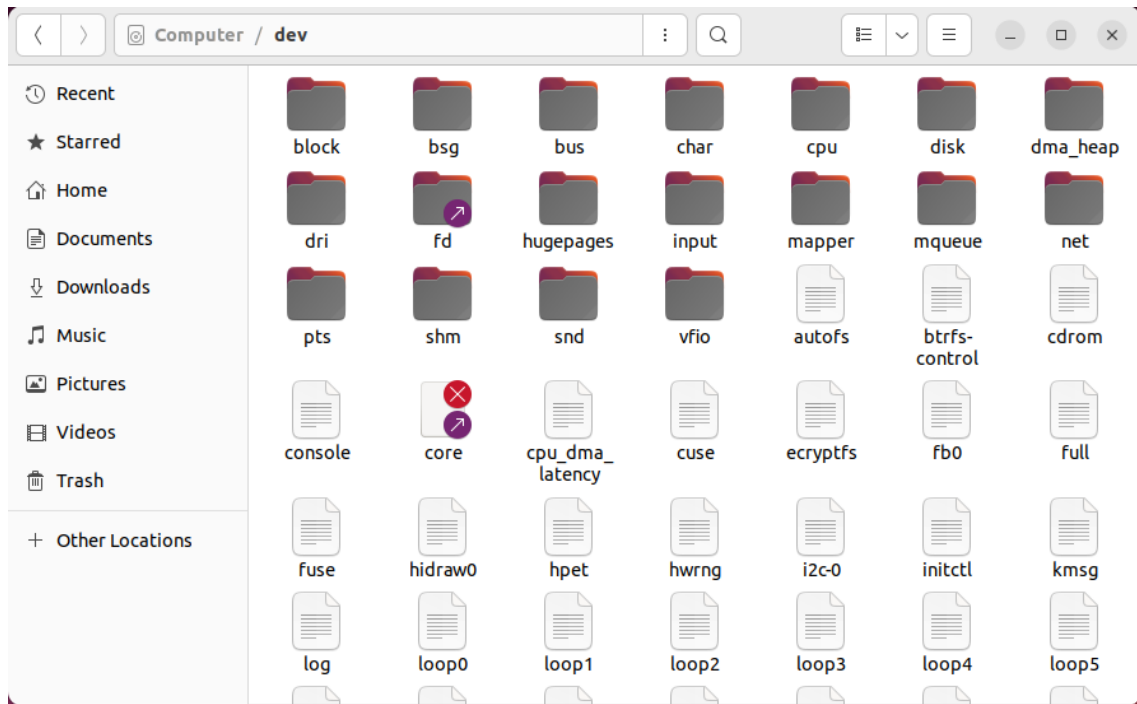
Este próximo diretório é mais comum para usuários Windows, porque o window possui um diretório com nome e funcionalidade parecidas que é o /tmp. Este é um diretório temporário, utilizado por aplicativos e serviços para armazenar arquivos temporários. Geralmente, esses arquivos são apagados quando o sistema é reiniciado.

### Diretório /tmp



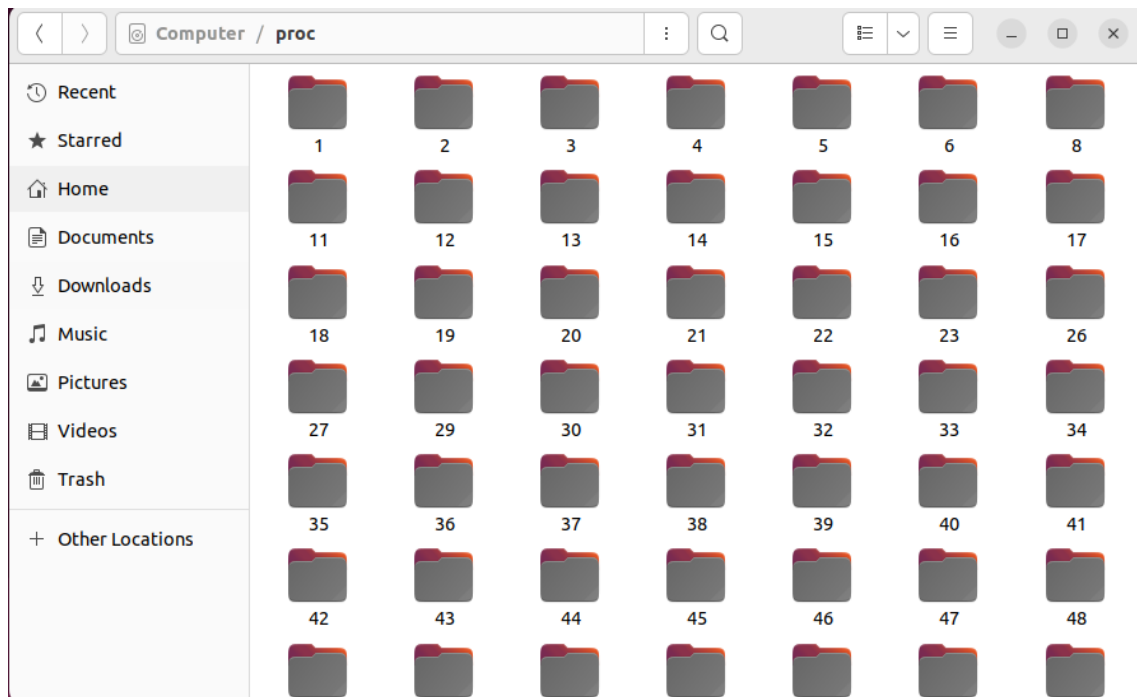
O próximo é o diretório /dev. Ele contém arquivos de dispositivos. No Linux, cada dispositivo de hardware é representado como um arquivo. Por exemplo, /dev/sda é o primeiro dispositivo de disco rígido do sistema.

## Diretório /dev



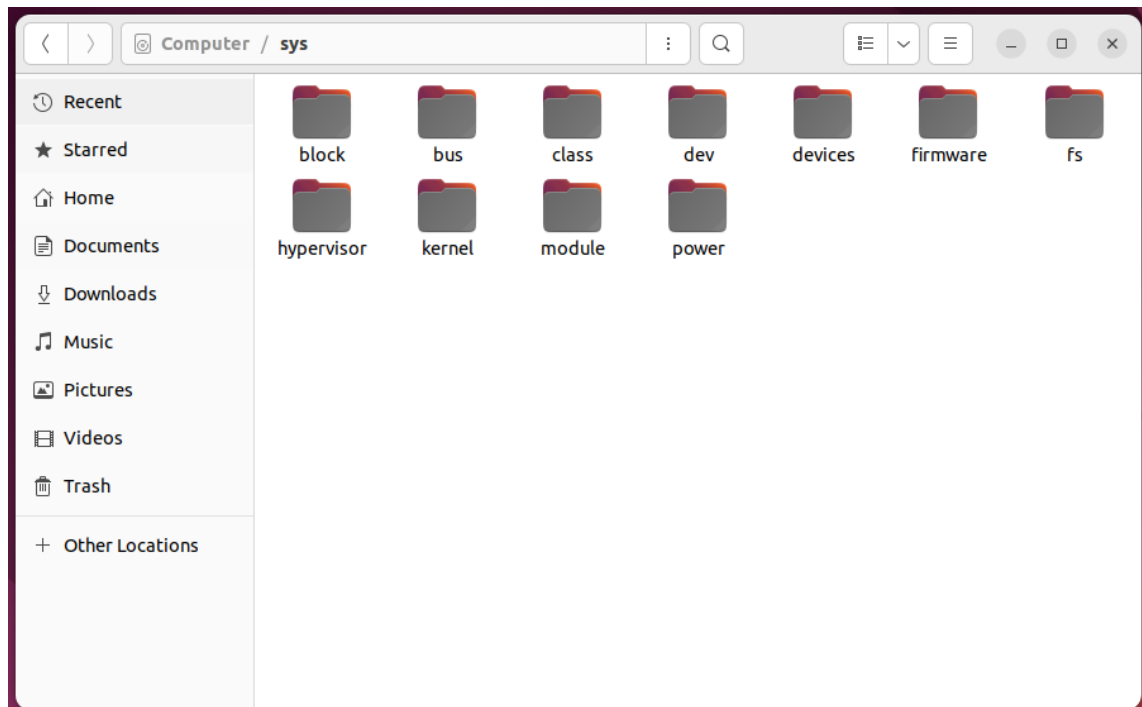
Este próximo diretório é um Este é um diretório especial que não existe fisicamente no disco, mas é criado pelo sistema. Ele contém informações sobre o estado do sistema, incluindo os processos atualmente em execução, utilização da memória, informações do hardware, etc. Ele é indicado por /proc.

## Diretório /proc



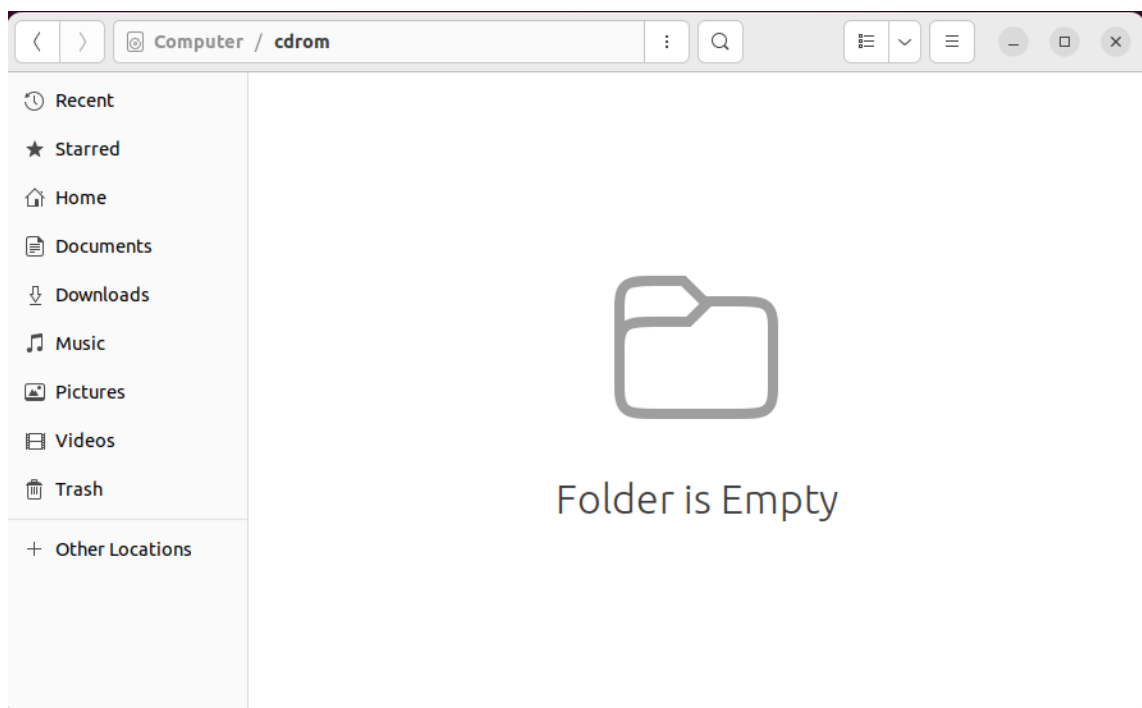
E semelhante ao Semelhante ao diretório /proc, o /sys também é um sistema de arquivos virtual criado pelo kernel. Ele contém informações sobre dispositivos, drivers e alguns aspectos do núcleo do sistema.

## Diretório /sys



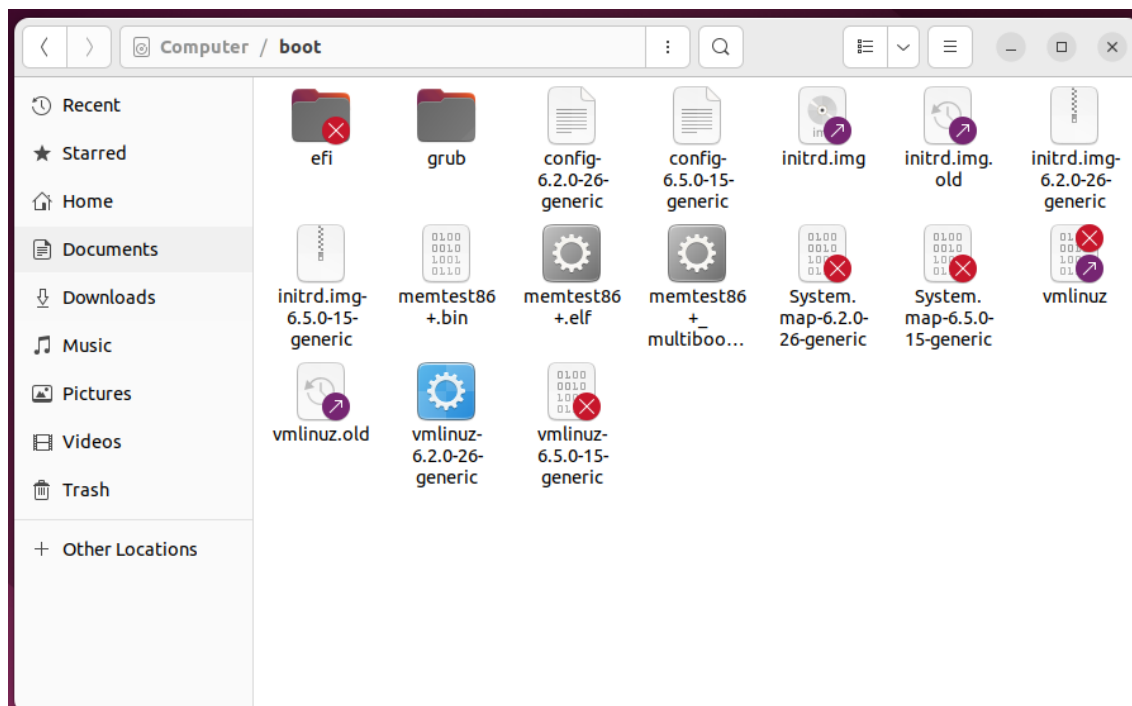
Um diretório antigo que pode existir em alguns dispositivos é o `cdrom`, ele é responsável por armazenar arquivos de discos, atualmente ele é normalmente vazio.

## Diretório /cdrom



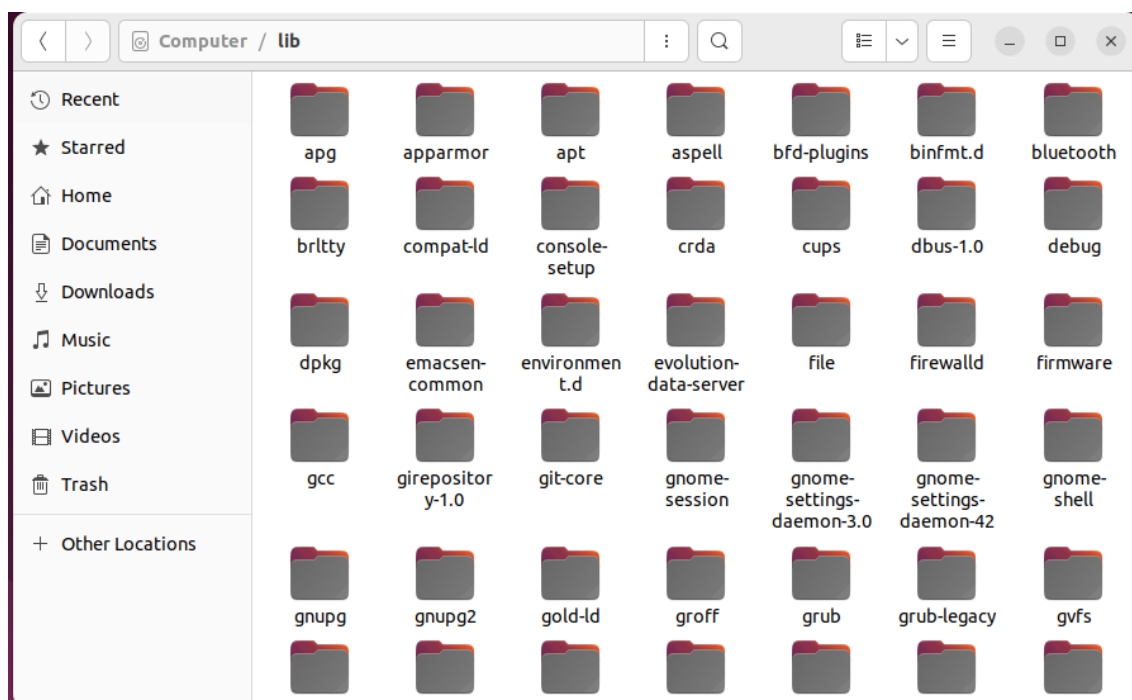
O próximo diretório também é um dos mais importantes, que é o `/boot`, como indicado pelo nome nele contém os arquivos necessários para o computador realizar a inicialização.

## Diretório /boot



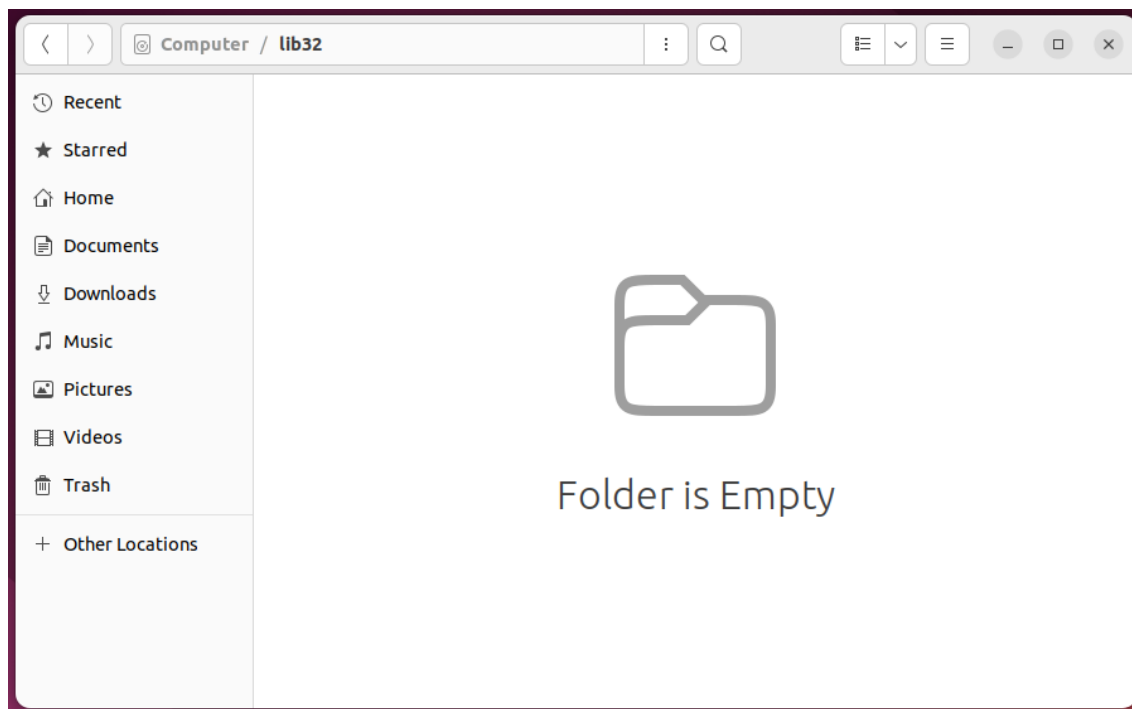
Os diretórios /lib, /lib32, /lib64 e /libx32, possuem praticamente a mesma função, eles são basicamente pastas que contêm bibliotecas de softwares e arquivos para sistemas operacionais. /lib são geralmente a multi arquitetura vão parar, enquanto as bibliotecas de 32 e 64 bits já indicam que são bibliotecas de quantidade de bits de 32 e 64 e pôr fim a libx32 é raramente usada, mas indica um software específico que pode usar essa pasta, porém o comum é ela estar vazia.

## Diretório /lib

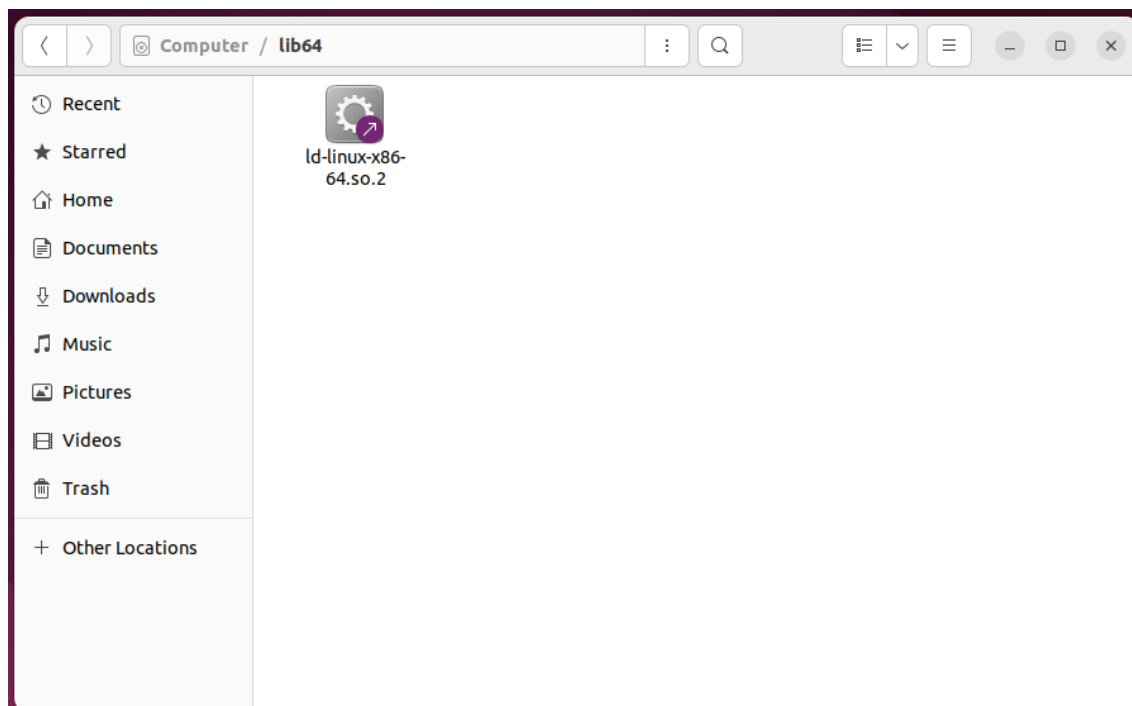




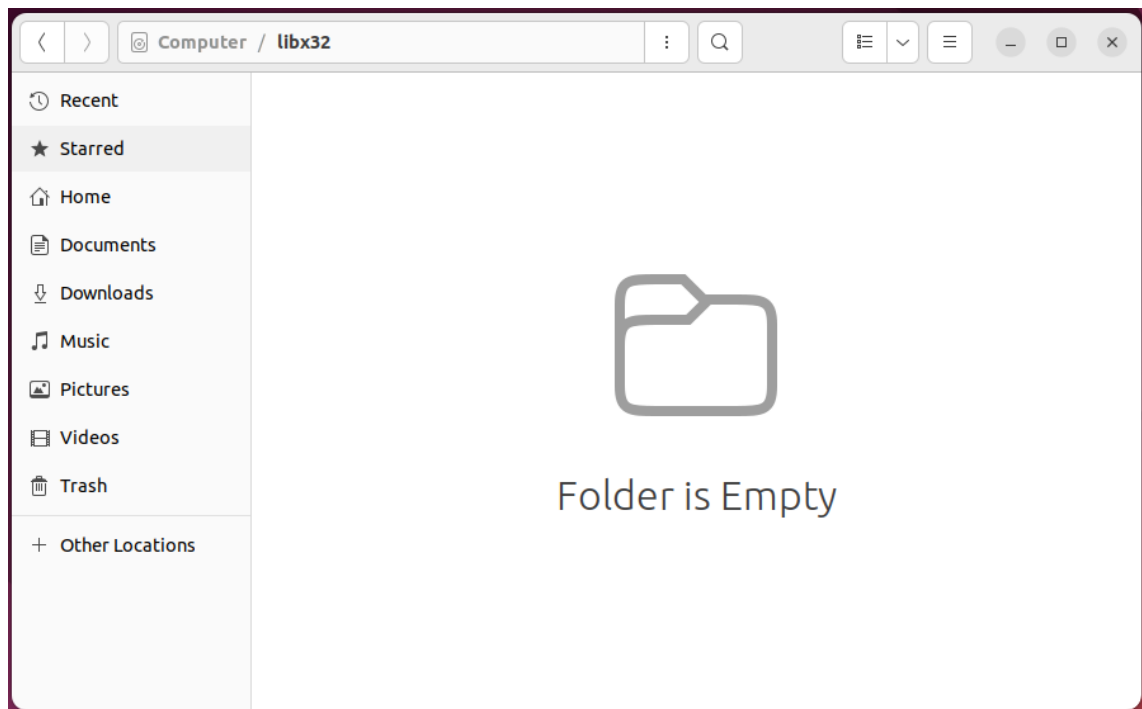
## Diretório /lib32



## Diretório /lib64

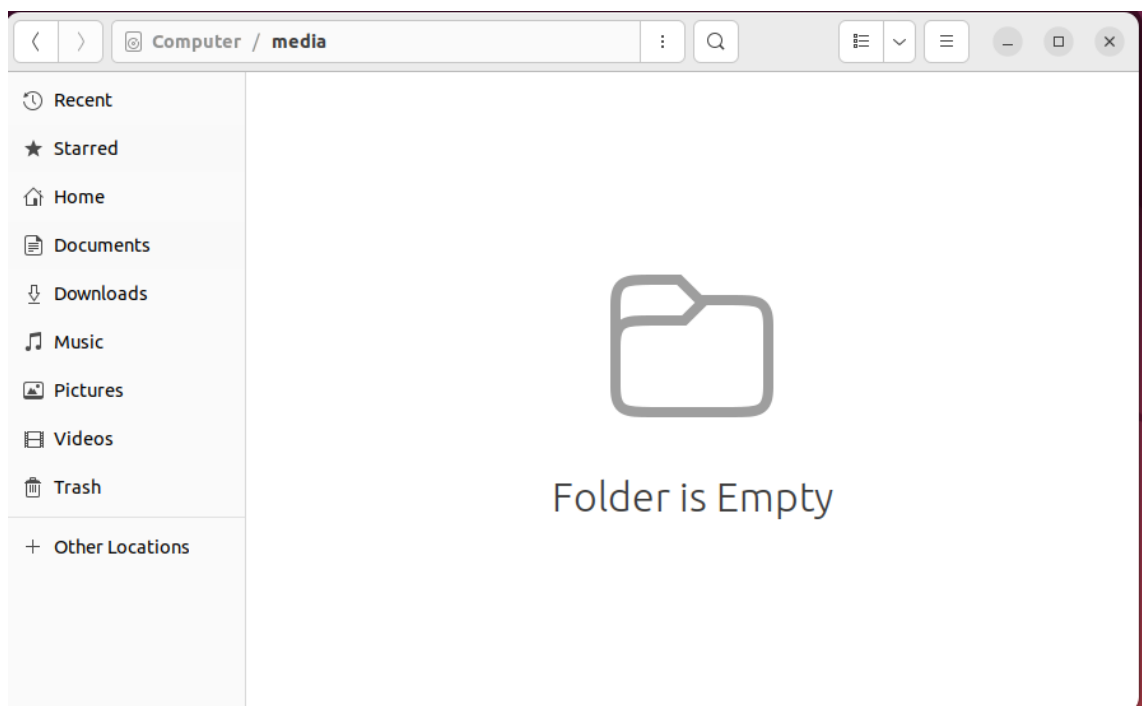


## Diretório /libx32



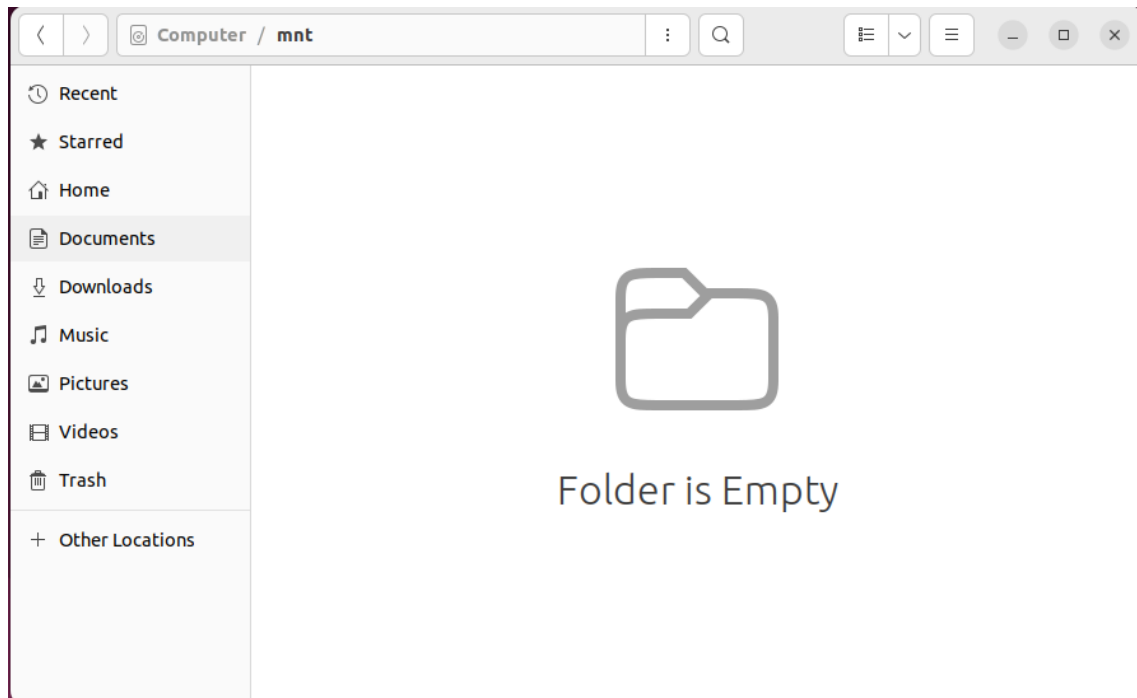
O /media é o diretório onde serão montadas automaticamente as unidades removíveis do sistema como por exemplo o pendrive, hd externo entre outros.

#### Diretório /media



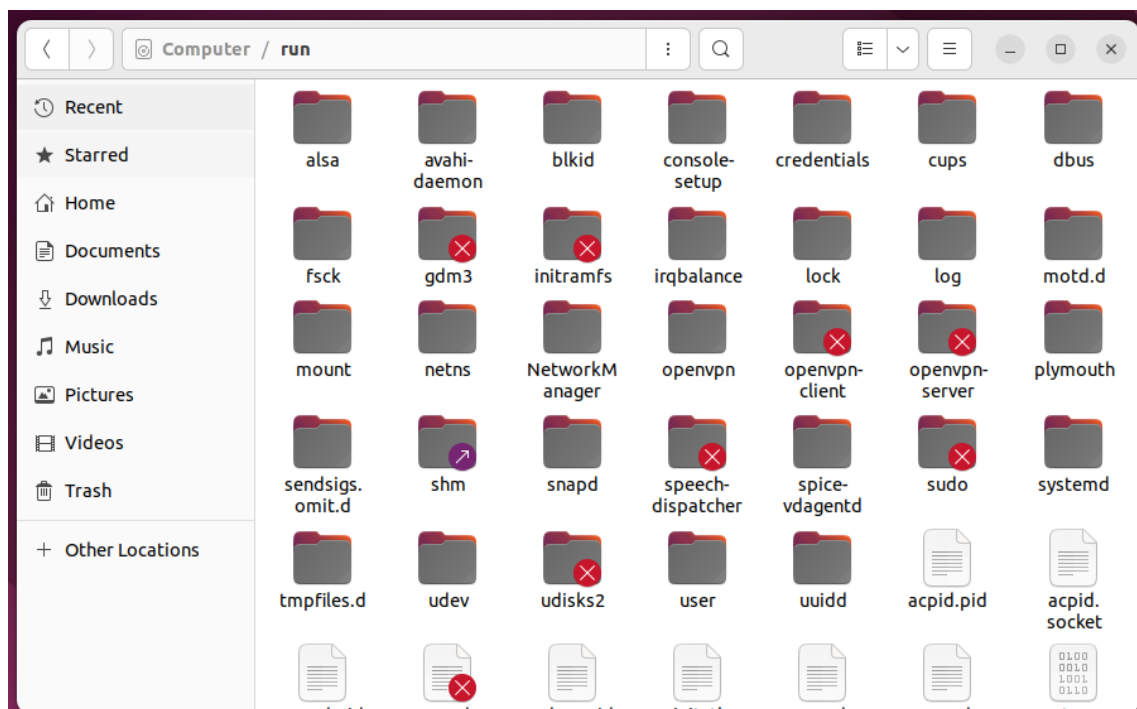
O /mnt faz basicamente a mesma função que o \media, a única diferença é que nesse caso o /mnt contem a montagem manual e não mais a automática feita pelo sistema.

## Diretório /mnt



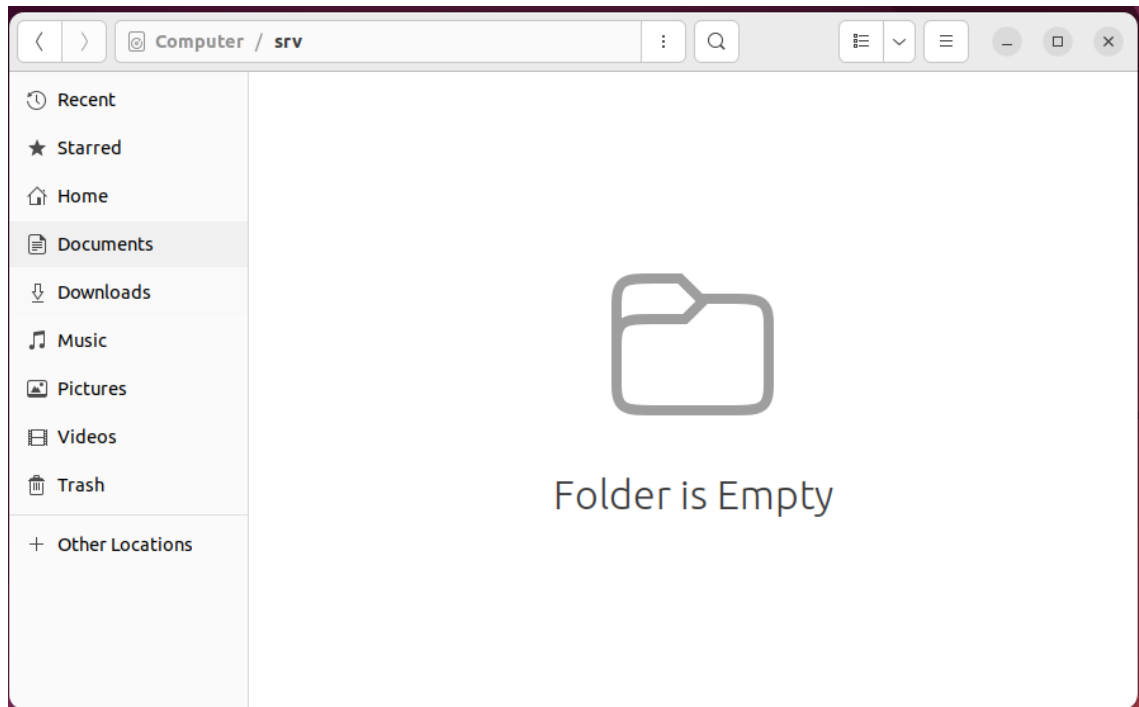
O próximo diretório é o /run. Este diretório é onde ficam arquivos utilizados por processos do sistema em runtime, principalmente os necessários durante o boot. Esses processos poderiam ser usados no /tmp? Até poderiam, mas o tmpwatch remove tudo que fica parado durante um tempo e causaria problema nos serviços em background, por isso foi criado este diretório.

## Diretório /run



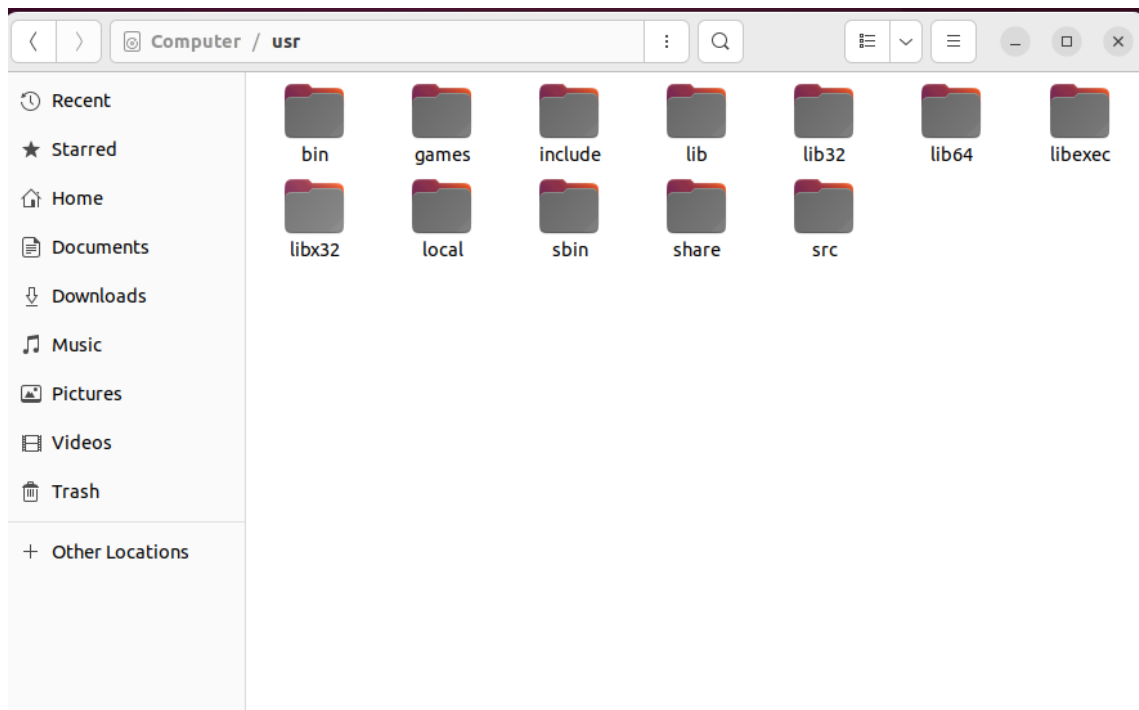
O próximo diretório é relativamente simples /srv é o diretório utilizado por serviços como FTP.

#### Diretório /srv



O próximo diretório é o /usr que é onde ficam os arquivos de programas específicos para usuários. Se você der uma olhada dentro deste diretório, verá praticamente um outro / dentro dele, isto é, vários diretórios repetidos. Isto porque alguns programas são distribuídos dentro do \_/usr.

#### Diretório /usr



Por ultimo temo o /root onde é a mesma coisa que o /home, porém, é somente para o super-usuário (root).

Diretório /root

