



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Engenharia Mecânica
Curso de Graduação em Engenharia Mecatrônica

Sistemas Digitais para Mecatrônica – Profº Eder Alves de Moura

Semana01b – Linux Instalação e Uso

Marco Tulio Vilela Fonseca.....12021EMT016

SUMÁRIO

1. INTRODUÇÃO	3
2. COMANDOS BÁSICOS DO TERMINAL LINUX.....	4
3. COMANDOS DO TERMINAL LINUX MAIS POPULARES.....	23
4. SISTEMAS DE ARQUIVOS NO LINUX.....	38

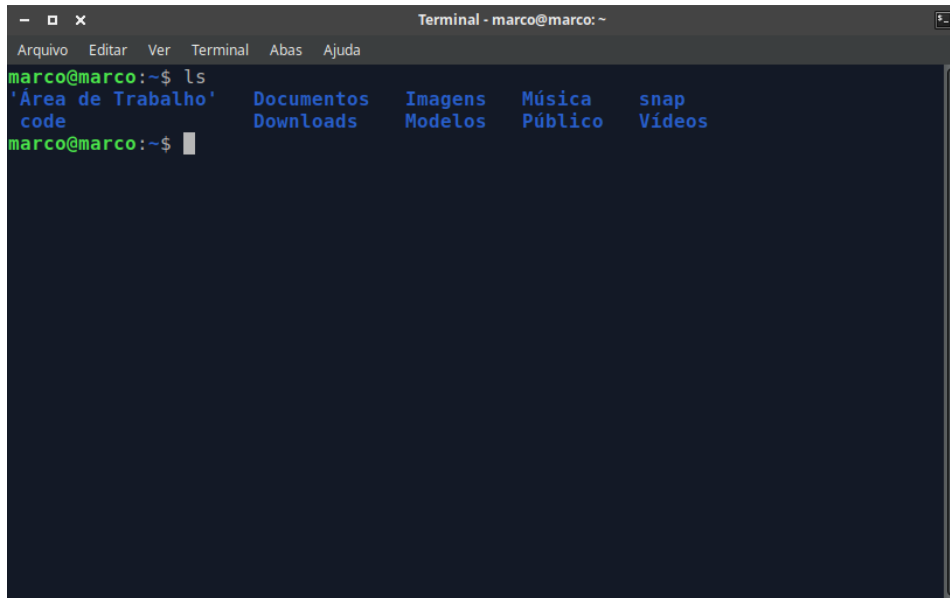
1. INTRODUÇÃO

Após a instalação do Xubuntu 22.04 via dual boot, o próximo desafio foi aprender e compreender o sistema operacional. Assim, este trabalho mostra os comandos básicos e populares do terminal além de uma breve explicação sobre o sistema de arquivos do Linux.

2. COMANDOS BÁSICOS DO TERMINAL LINUX

❖ *ls*

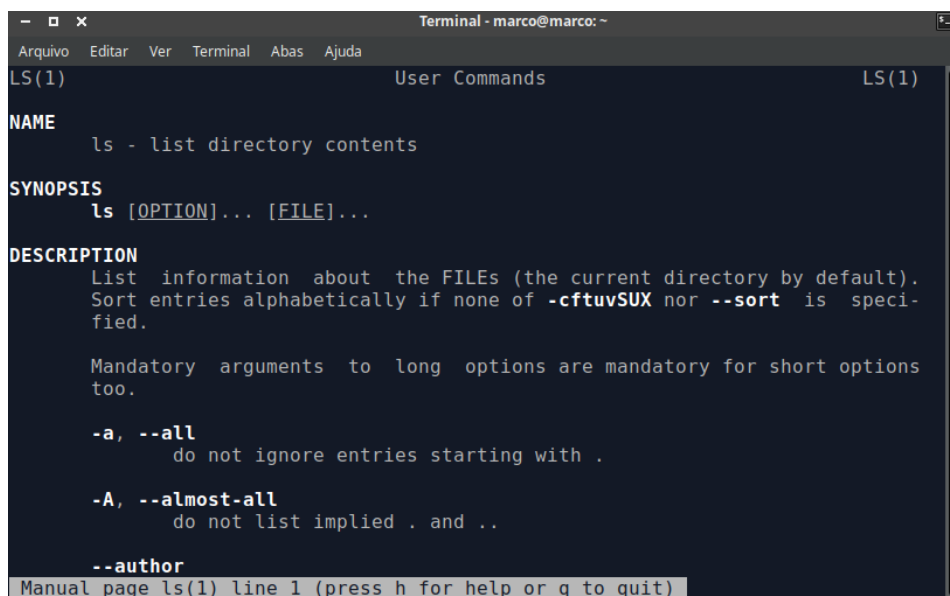
O comando *ls* no terminal é utilizado para listar os arquivos e diretórios presentes no diretório atual.

A screenshot of a Linux terminal window titled "Terminal - marco@marco: ~". The window has a menu bar with "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The terminal shows the command "marco@marco:~\$ ls" and its output: "'Área de Trabalho' Documentos Imagens Música snap", "code Downloads Modelos Público Vídeos". The prompt "marco@marco:~\$" is followed by a cursor.

❖ *man*

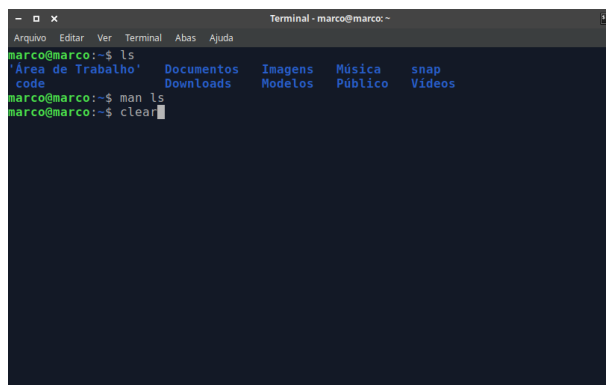
O comando *man* exibe o manual de qualquer comando no terminal Linux, oferecendo detalhes sobre suas opções e uso. Basta digitar *man* seguido do comando desejado.

Comando utilizado para exemplo: *man ls*

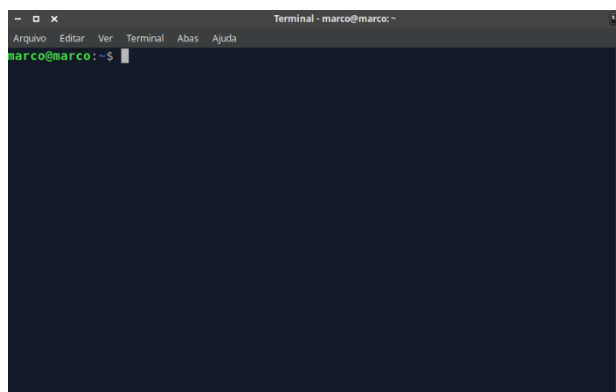
A screenshot of a Linux terminal window titled "Terminal - marco@marco: ~". The window shows the manual page for the 'ls' command. The header "LS(1) User Commands LS(1)" is at the top. The content includes sections for "NAME", "SYNOPSIS", and "DESCRIPTION". The "NAME" section says "ls - list directory contents". The "SYNOPSIS" section shows "ls [OPTION]... [FILE]...". The "DESCRIPTION" section explains that 'ls' lists information about files and sorts them alphabetically by default. It also lists several options: "-a, --all" (do not ignore entries starting with .), "-A, --almost-all" (do not list implied . and ..), and "--author". At the bottom, it says "Manual page ls(1) line 1 (press h for help or q to quit)".

❖ *clear*

O comando *clear* limpa a tela do terminal, removendo todo o conteúdo exibido anteriormente, deixando-o vazio para novas entradas.



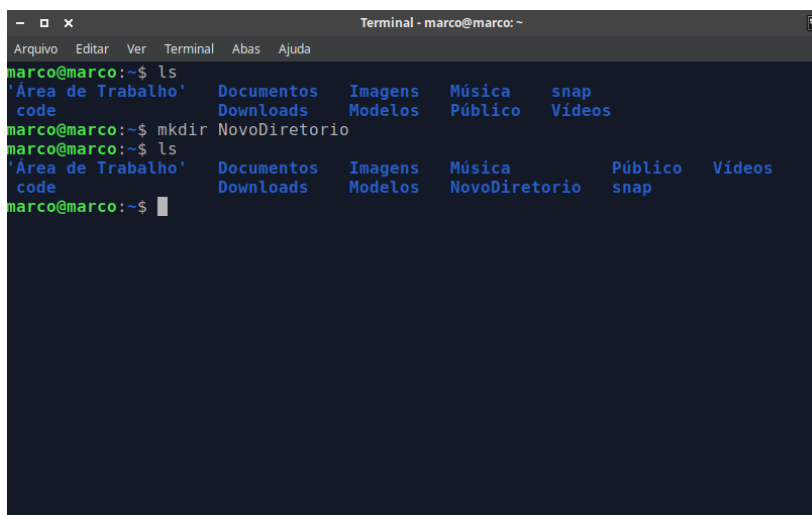
```
Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$ ls
'Área de Trabalho' Documentos Imagens Música snap
code Downloads Modelos Público Vídeos
marco@marco:~$ man ls
marco@marco:~$ clear
```



```
Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$
```

❖ *mkdir*

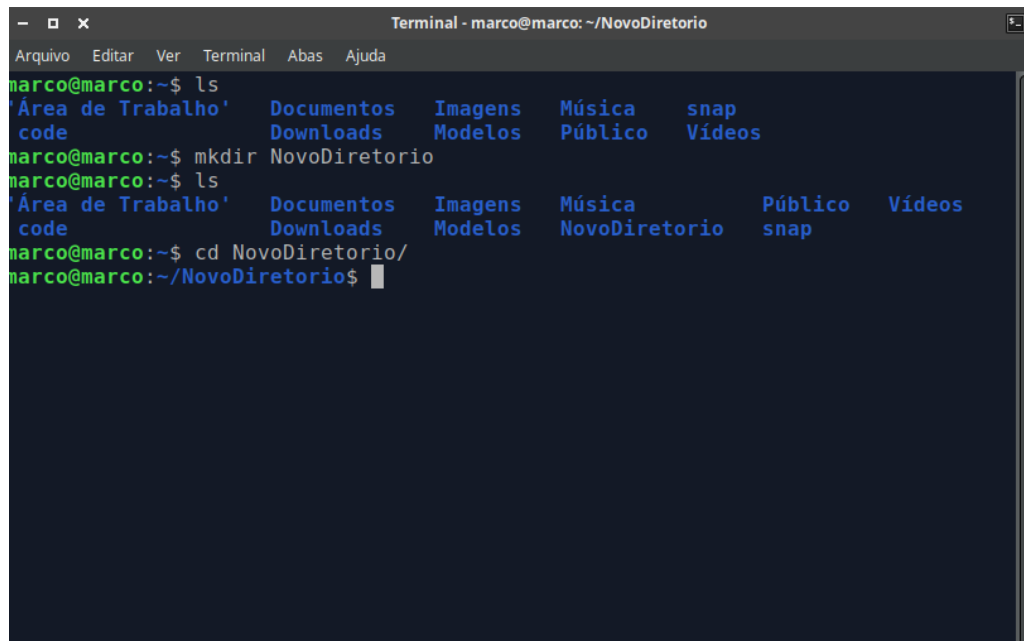
O comando *mkdir* cria um novo diretório no sistema de arquivos, especificando o nome e a localização desejada.



```
Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$ ls
'Área de Trabalho' Documentos Imagens Música snap
code Downloads Modelos Público Vídeos
marco@marco:~$ mkdir NovoDiretorio
marco@marco:~$ ls
'Área de Trabalho' Documentos Imagens Música Público Vídeos
code Downloads Modelos NovoDiretorio snap
marco@marco:~$
```

❖ *cd*

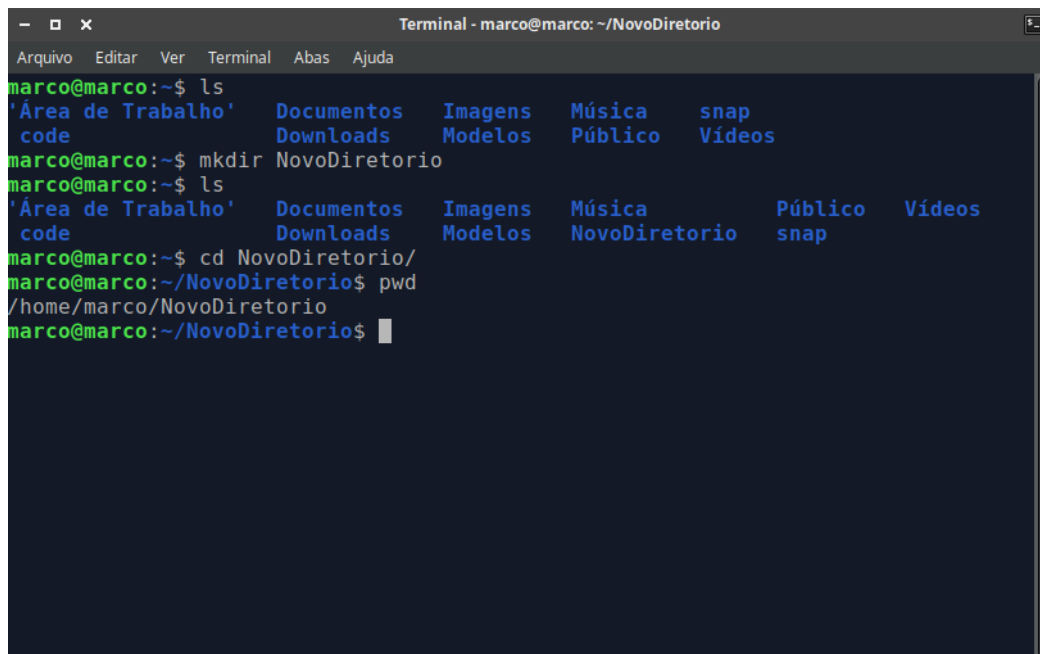
O comando *cd* muda o diretório atual no terminal, permitindo navegar entre diferentes pastas do sistema de arquivos.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The user runs 'ls' showing a directory listing. Then runs 'mkdir NovoDiretorio'. Then runs 'ls' again, showing the new directory. Finally, runs 'cd NovoDiretorio/' and the prompt changes to ~/NovoDiretorio\$.

```
marco@marco:~$ ls
'Área de Trabalho'  Documentos  Imagens    Música      snap
code                Downloads  Modelos    Público     Vídeos
marco@marco:~$ mkdir NovoDiretorio
marco@marco:~$ ls
'Área de Trabalho'  Documentos  Imagens    Música      NovoDiretorio  Público  Vídeos
code                Downloads  Modelos    NovoDiretorio  snap
marco@marco:~$ cd NovoDiretorio/
marco@marco:~/NovoDiretorio$
```

❖ *pwd*

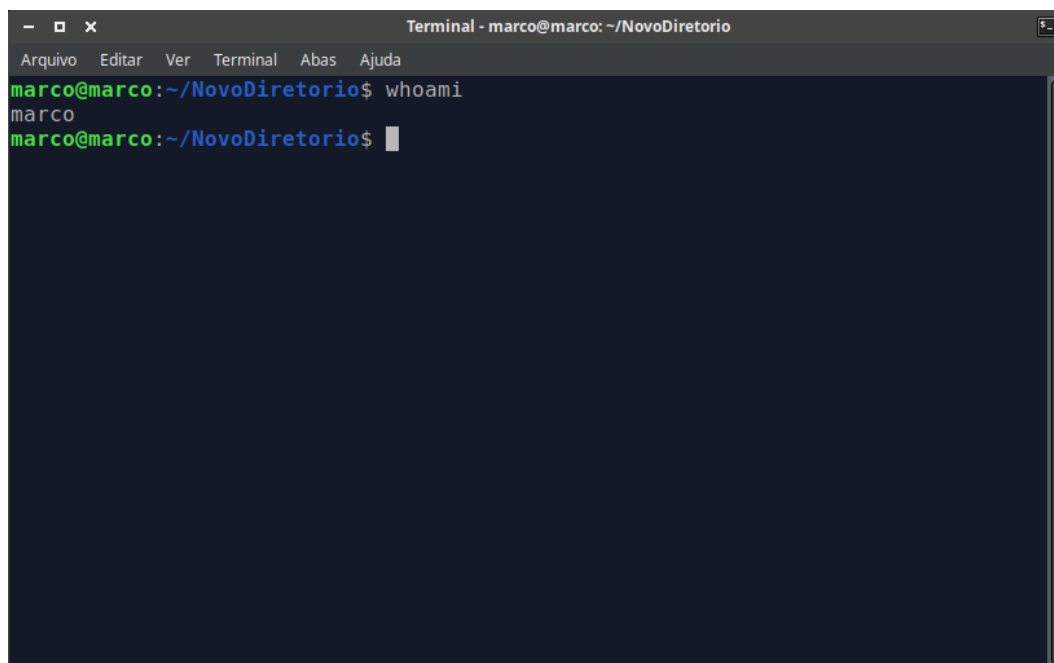
O comando *pwd* exibe o caminho completo do diretório atual no terminal, mostrando onde você está no sistema de arquivos.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The user runs 'ls', 'mkdir NovoDiretorio', and 'cd NovoDiretorio/'. Then runs 'pwd', which outputs the full path: /home/marco/NovoDiretorio.

```
marco@marco:~$ ls
'Área de Trabalho'  Documentos  Imagens    Música      snap
code                Downloads  Modelos    Público     Vídeos
marco@marco:~$ mkdir NovoDiretorio
marco@marco:~$ ls
'Área de Trabalho'  Documentos  Imagens    Música      NovoDiretorio  Público  Vídeos
code                Downloads  Modelos    NovoDiretorio  snap
marco@marco:~$ cd NovoDiretorio/
marco@marco:~/NovoDiretorio$ pwd
/home/marco/NovoDiretorio
marco@marco:~/NovoDiretorio$
```

❖ *whoami*

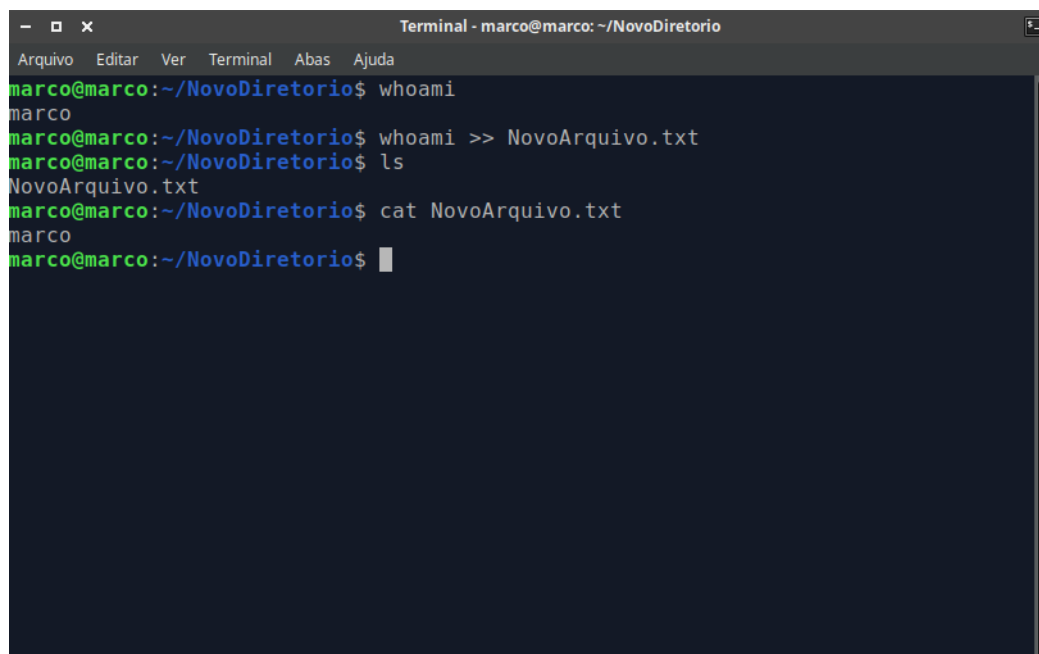
O comando *whoami* exibe o nome do usuário atualmente logado no sistema.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar containing "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The prompt is "marco@marco:~/NovoDiretorio\$". The command "whoami" has been entered, and the output "marco" is displayed on the next line. The prompt is now "marco@marco:~/NovoDiretorio\$" with a cursor.

```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ whoami
marco
marco@marco:~/NovoDiretorio$
```

❖ *Redirecionadores (>>)*

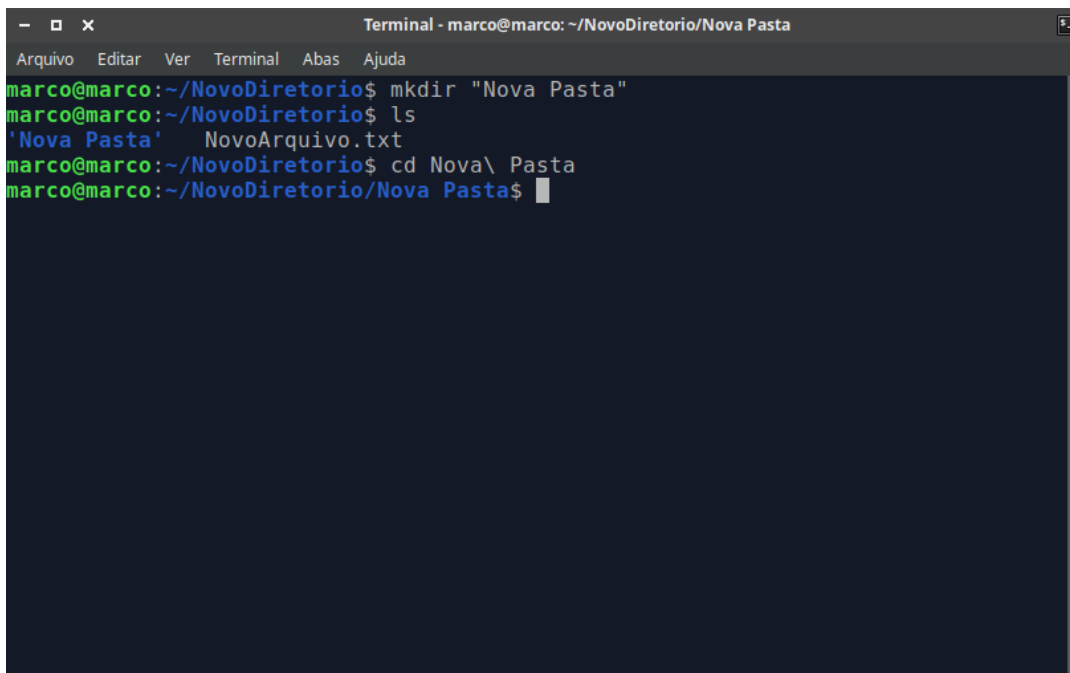
O operador *redirecionador* `>>` no terminal Linux é usado para redirecionar a saída de um comando para um arquivo, anexando o conteúdo ao final do arquivo sem sobrescrevê-lo. Se o arquivo não existir, ele será criado.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar containing "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The prompt is "marco@marco:~/NovoDiretorio\$". The command "whoami" is entered, outputting "marco". Then, "whoami >> NovoArquivo.txt" is entered. Next, "ls" is entered, showing "NovoArquivo.txt" in the directory listing. Finally, "cat NovoArquivo.txt" is entered, outputting "marco". The prompt is now "marco@marco:~/NovoDiretorio\$" with a cursor.

```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ whoami
marco
marco@marco:~/NovoDiretorio$ whoami >> NovoArquivo.txt
marco@marco:~/NovoDiretorio$ ls
NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cat NovoArquivo.txt
marco
marco@marco:~/NovoDiretorio$
```

❖ *pasta com nome composto*

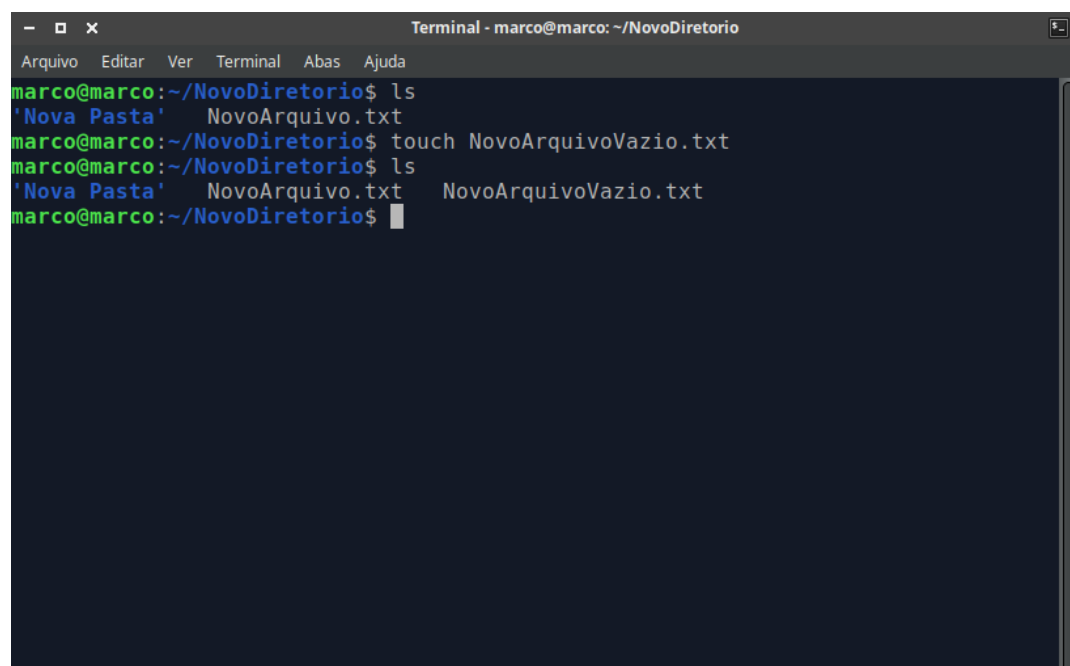
Para criar e acessar pastas com nomes compostos no terminal Linux, use aspas ("*Nova Pasta*") ou barras invertidas (*Nova\Pasta*). Isso garante que o sistema reconheça os espaços no nome.



```
Terminal - marco@marco: ~/NovoDiretorio/Nova Pasta
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ mkdir "Nova Pasta"
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cd Nova\Pasta
marco@marco:~/NovoDiretorio/Nova Pasta$
```

❖ *touch*

O comando *touch* cria um novo arquivo vazio.

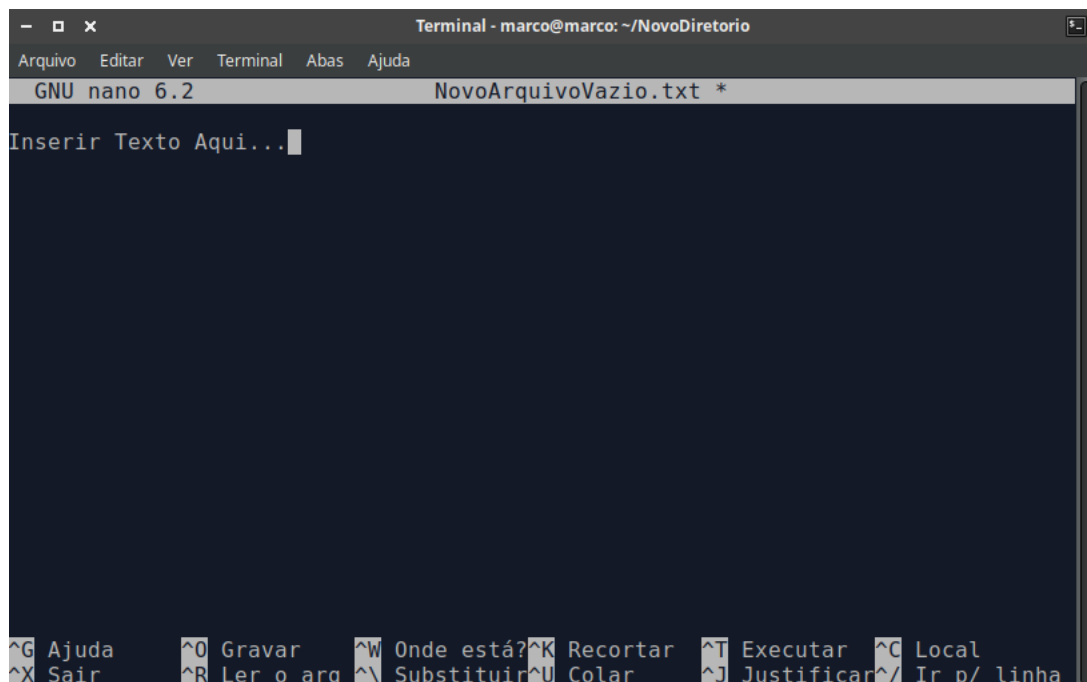


```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ touch NovoArquivoVazio.txt
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivo.txt  NovoArquivoVazio.txt
marco@marco:~/NovoDiretorio$
```

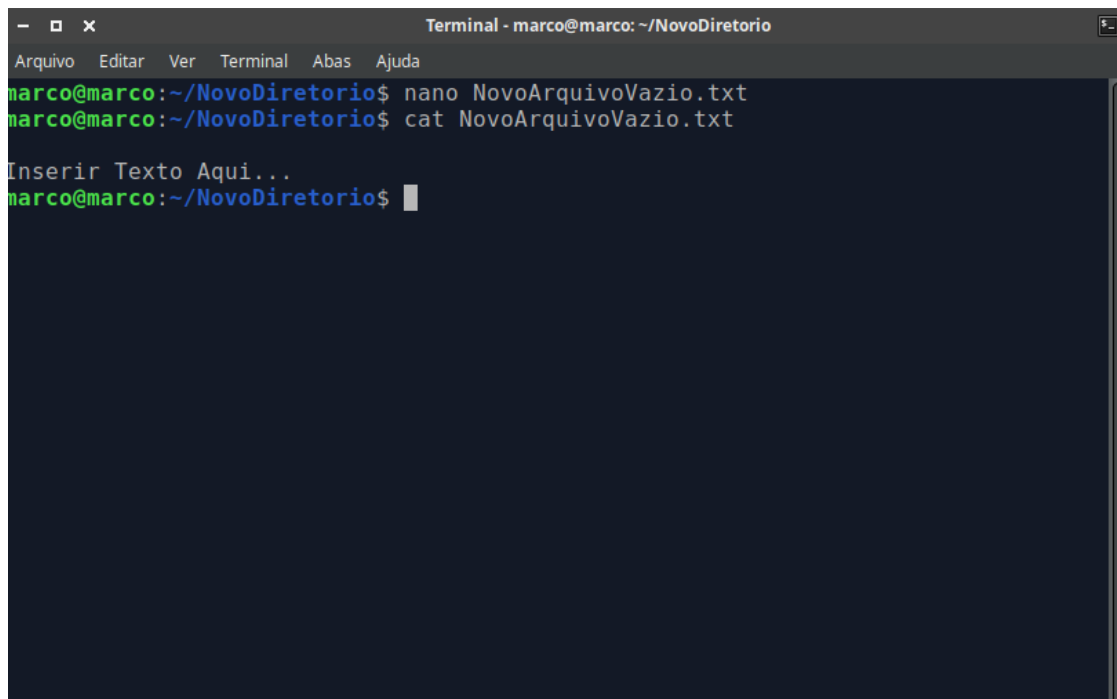

❖ *nano*

O comando *nano* abre um editor de texto simples no terminal Linux, permitindo criar e editar arquivos diretamente na linha de comando.

Comando utilizado para exemplo: *nano NovoArquivoVazio.txt*



The screenshot shows the nano text editor running in a terminal window. The title bar reads "Terminal - marco@marco: ~/NovoDiretorio". The menu bar includes "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The status bar at the top indicates "GNU nano 6.2" and "NovoArquivoVazio.txt *". The main editing area is dark blue with the text "Inserir Texto Aqui..." and a cursor. The bottom status bar displays various keyboard shortcuts: ^G Ajuda, ^O Gravar, ^W Onde está?, ^K Recortar, ^T Executar, ^C Local, ^X Sair, ^R Ler o arquivo, ^\ Substituir, ^U Colar, ^J Justificar, and ^_ Ir p/ linha.



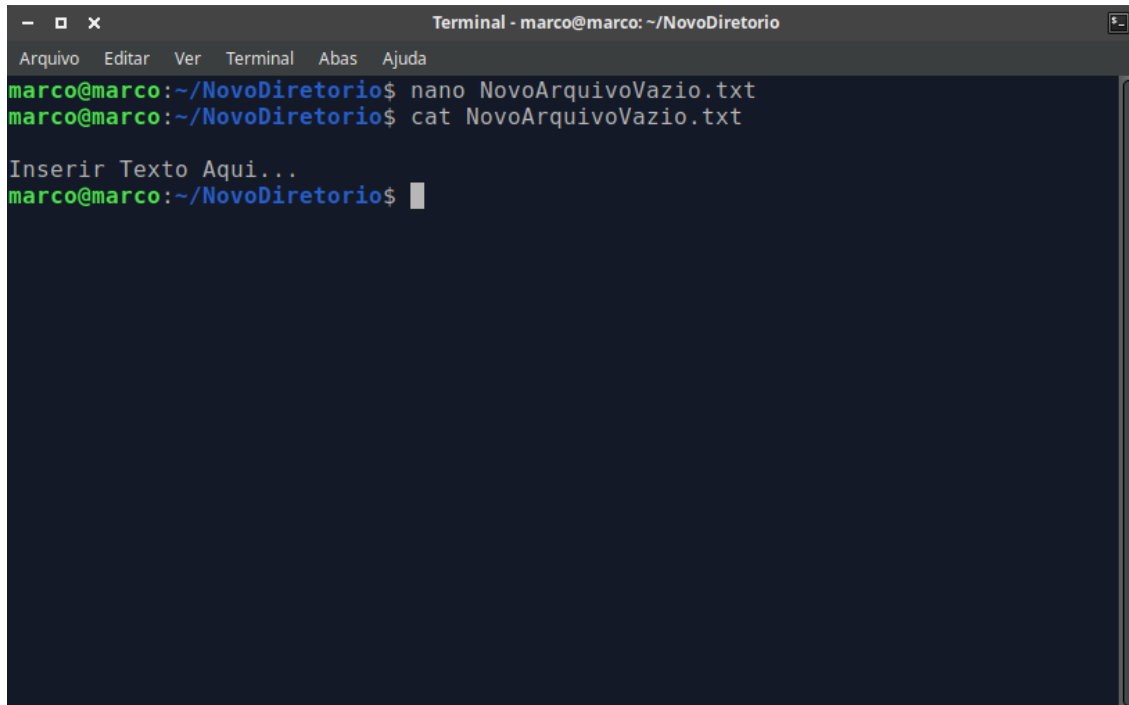
The screenshot shows a terminal window with the following commands and output:

```
marco@marco:~/NovoDiretorio$ nano NovoArquivoVazio.txt
marco@marco:~/NovoDiretorio$ cat NovoArquivoVazio.txt
```

The terminal output shows the file was created and is currently empty, displaying "Inserir Texto Aqui..." with a cursor.

❖ *cat*

O comando *cat* exibe o conteúdo de arquivos diretamente no terminal.

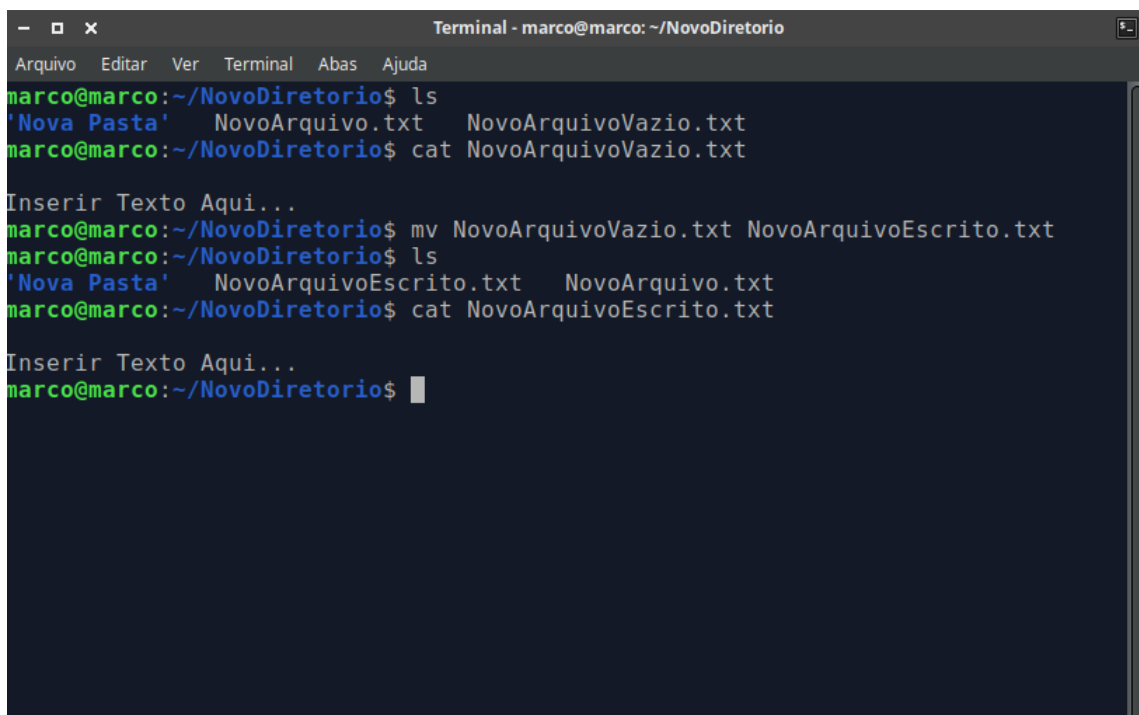
A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar containing "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The terminal shows the following commands and output:

```
marco@marco:~/NovoDiretorio$ nano NovoArquivoVazio.txt
marco@marco:~/NovoDiretorio$ cat NovoArquivoVazio.txt

Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$
```

❖ *mv*

O comando *mv* move ou renomeia arquivos e diretórios.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar containing "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The terminal shows the following commands and output:

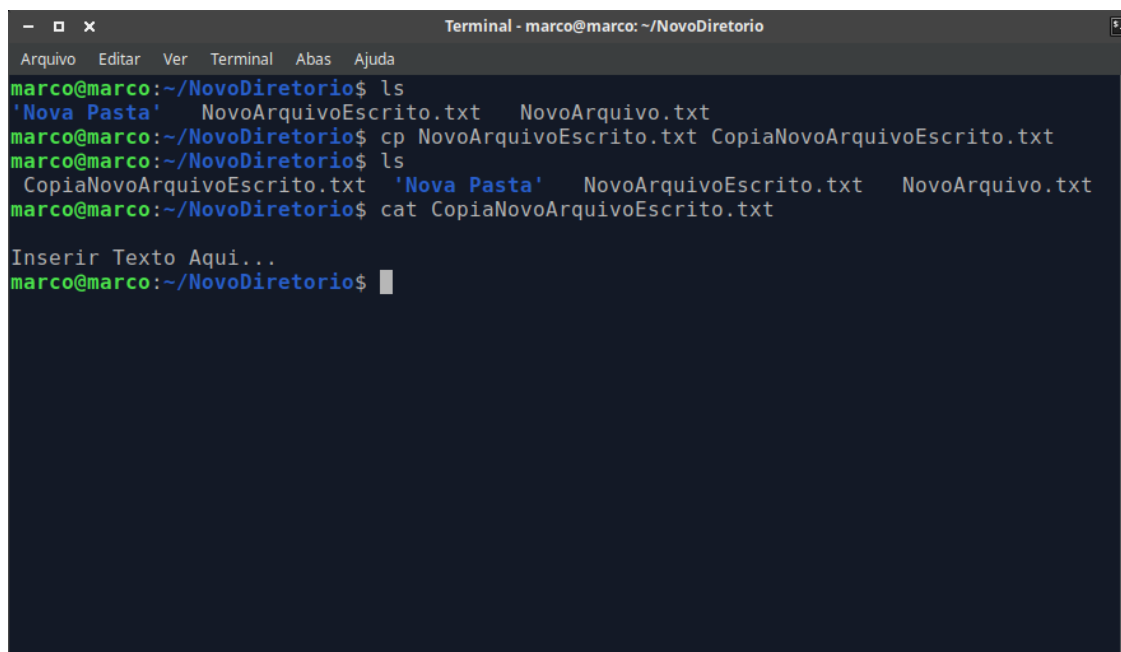
```
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivo.txt  NovoArquivoVazio.txt
marco@marco:~/NovoDiretorio$ cat NovoArquivoVazio.txt

Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$ mv NovoArquivoVazio.txt NovoArquivoEscrito.txt
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cat NovoArquivoEscrito.txt

Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$
```

❖ *cp*

O comando *cp* copia arquivos ou diretórios.



```

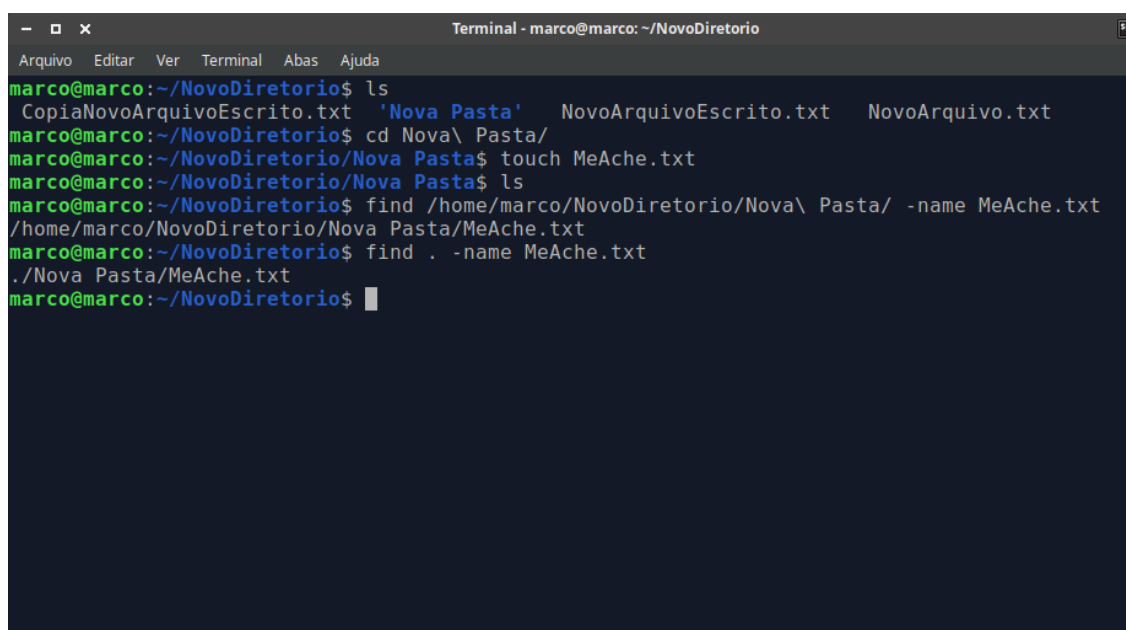
Terminal - marco@marco: ~/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cp NovoArquivoEscrito.txt CopiaNovoArquivoEscrito.txt
marco@marco:~/NovoDiretorio$ ls
CopiaNovoArquivoEscrito.txt  'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cat CopiaNovoArquivoEscrito.txt

Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$

```

❖ *find*

O comando *find* busca arquivos e diretórios em um caminho específico baseado em critérios como nome, tipo ou data de modificação.



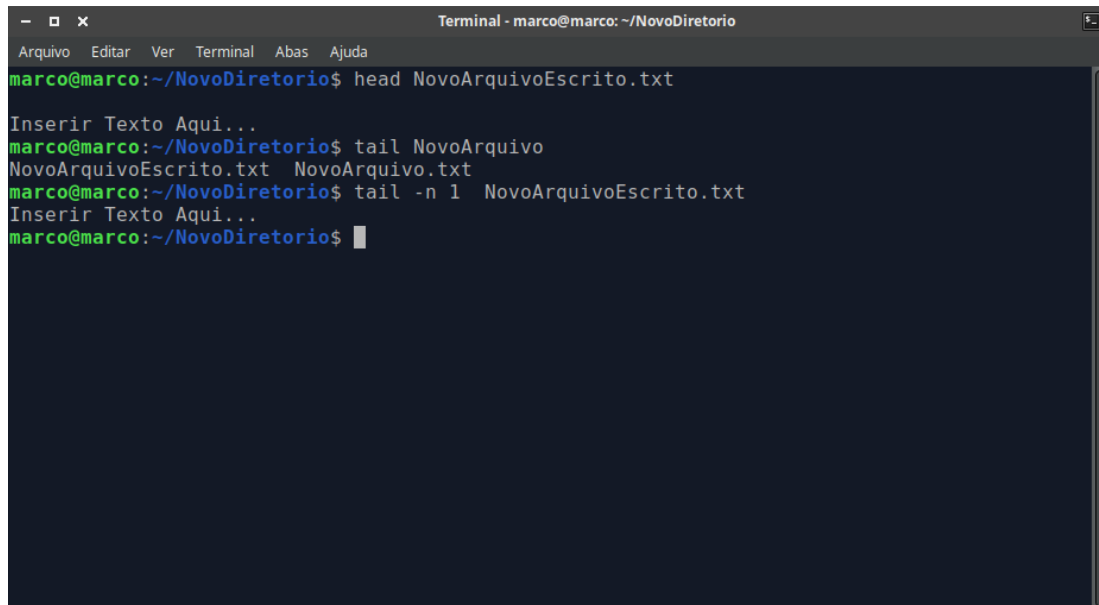
```

Terminal - marco@marco: ~/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ ls
CopiaNovoArquivoEscrito.txt  'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cd Nova\ Pasta/
marco@marco:~/NovoDiretorio/Nova Pasta$ touch MeAche.txt
marco@marco:~/NovoDiretorio/Nova Pasta$ ls
marco@marco:~/NovoDiretorio$ find /home/marco/NovoDiretorio/Nova\ Pasta/ -name MeAche.txt
/home/marco/NovoDiretorio/Nova Pasta/MeAche.txt
marco@marco:~/NovoDiretorio$ find . -name MeAche.txt
./Nova Pasta/MeAche.txt
marco@marco:~/NovoDiretorio$

```

❖ *head & tail*

Os comandos *head* e *tail* exibem, respectivamente, as primeiras e as últimas linhas de um arquivo. Por padrão, ambos mostram 10 linhas, mas isso pode ser ajustado com opções como *-n*.

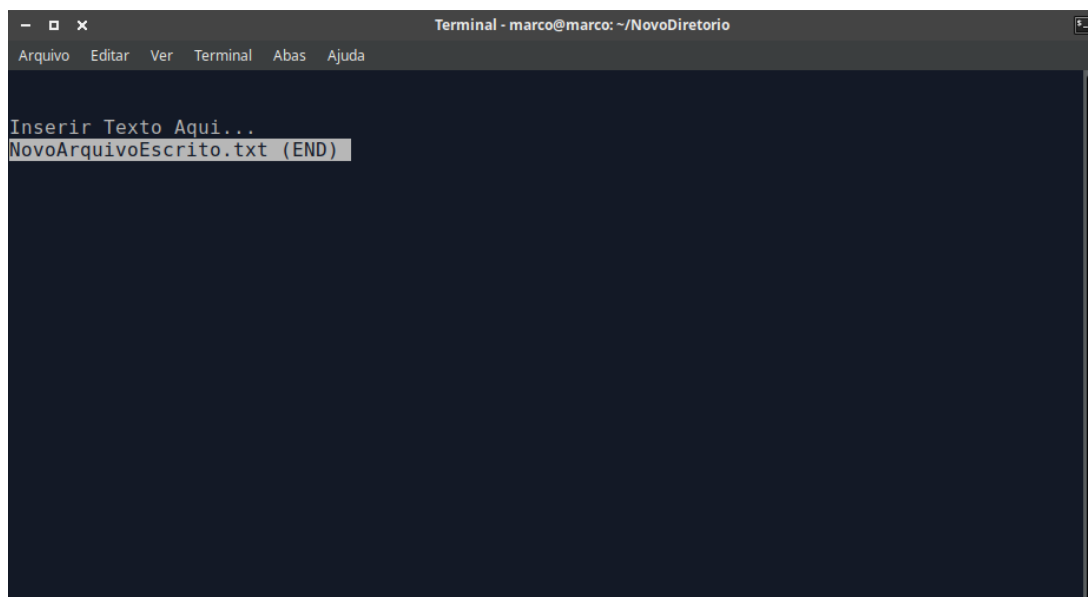
A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The prompt is "marco@marco:~/NovoDiretorio\$". The user enters "head NovoArquivoEscrito.txt", followed by "Inserir Texto Aqui...". Then the user enters "tail NovoArquivo", followed by "NovoArquivoEscrito.txt NovoArquivo.txt". Then the user enters "tail -n 1 NovoArquivoEscrito.txt", followed by "Inserir Texto Aqui...". Finally, the user enters "marco@marco:~/NovoDiretorio\$" with a cursor.

```
marco@marco:~/NovoDiretorio$ head NovoArquivoEscrito.txt
Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$ tail NovoArquivo
NovoArquivoEscrito.txt NovoArquivo.txt
marco@marco:~/NovoDiretorio$ tail -n 1 NovoArquivoEscrito.txt
Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$
```

❖ *less*

O comando *less* permite visualizar o conteúdo de um arquivo de forma paginada, facilitando a navegação por arquivos longos.

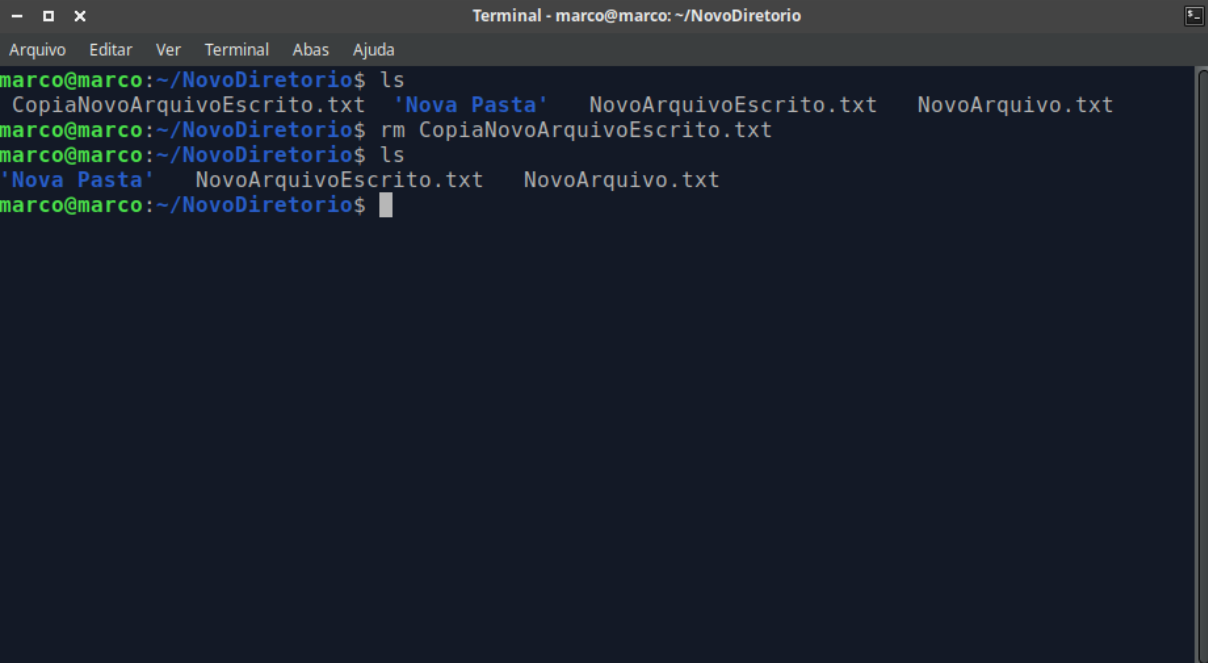
Comando utilizado para exemplo: *less NovoArquivoEscrito.txt*

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The prompt is "marco@marco:~/NovoDiretorio\$". The user enters "less NovoArquivoEscrito.txt", and the output "Inserir Texto Aqui..." is displayed. The prompt "marco@marco:~/NovoDiretorio\$" is visible below the output.

```
marco@marco:~/NovoDiretorio$ less NovoArquivoEscrito.txt
Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$
```

❖ *rm*

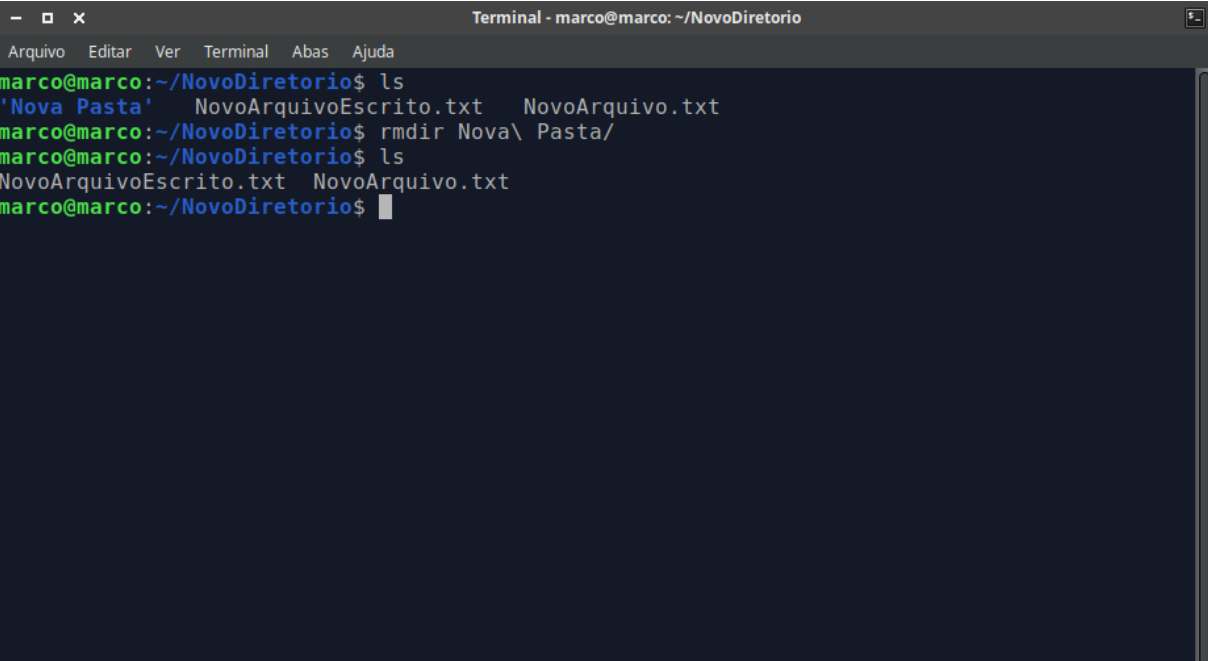
O comando *rm* remove arquivos.



```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ ls
CopiaNovoArquivoEscrito.txt  'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ rm CopiaNovoArquivoEscrito.txt
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$
```

❖ *rmdir*

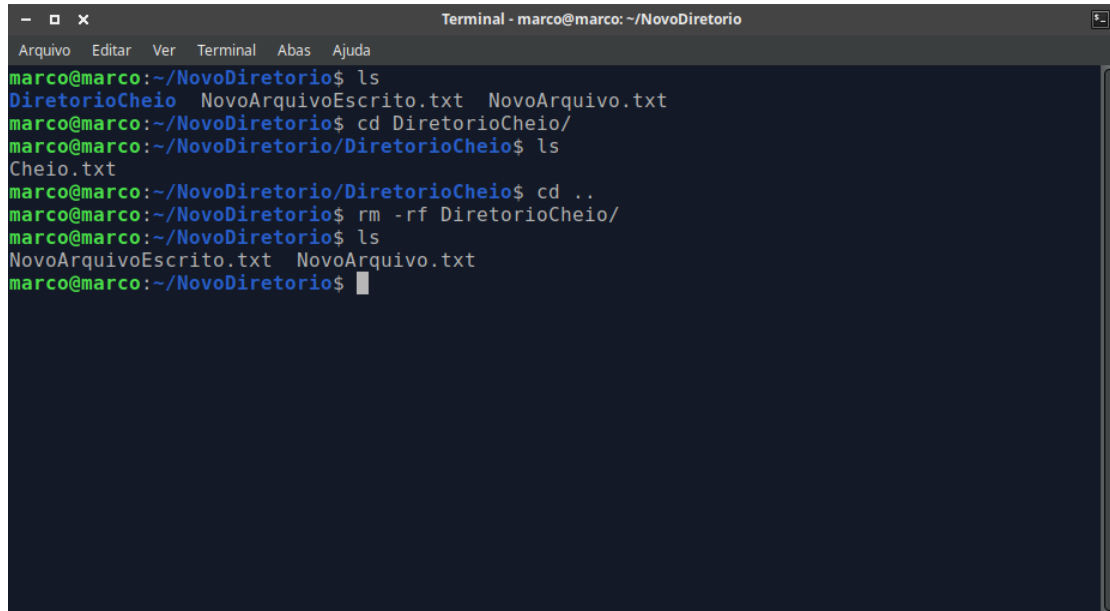
O comando *rmdir* remove diretórios vazios.



```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ ls
'Nova Pasta'  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ rmdir Nova\ Pasta/
marco@marco:~/NovoDiretorio$ ls
NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$
```

❖ *rm -rf*

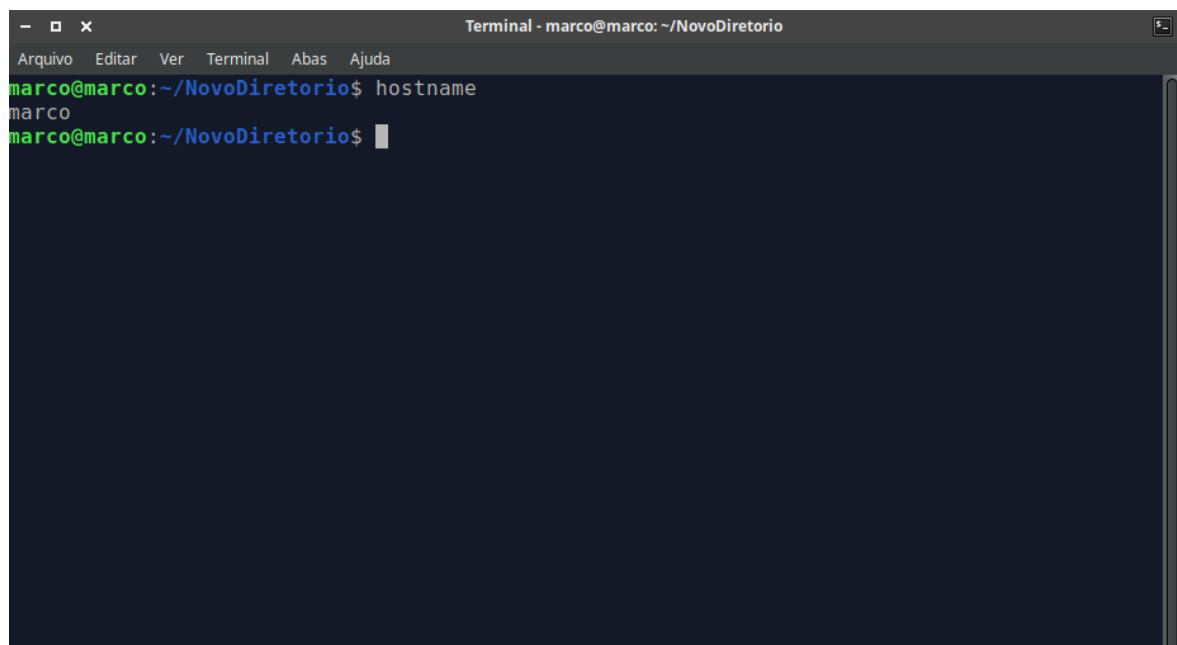
O comando *rm -rf* remove de forma recursiva (*-r*) e forçada (*-f*) arquivos e diretórios, incluindo seu conteúdo, sem pedir confirmação.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The terminal shows the following commands and output:

```
marco@marco:~/NovoDiretorio$ ls
DiretorioCheio  NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ cd DiretorioCheio/
marco@marco:~/NovoDiretorio/DiretorioCheio$ ls
Cheio.txt
marco@marco:~/NovoDiretorio/DiretorioCheio$ cd ..
marco@marco:~/NovoDiretorio$ rm -rf DiretorioCheio/
marco@marco:~/NovoDiretorio$ ls
NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$
```

❖ *hostname*

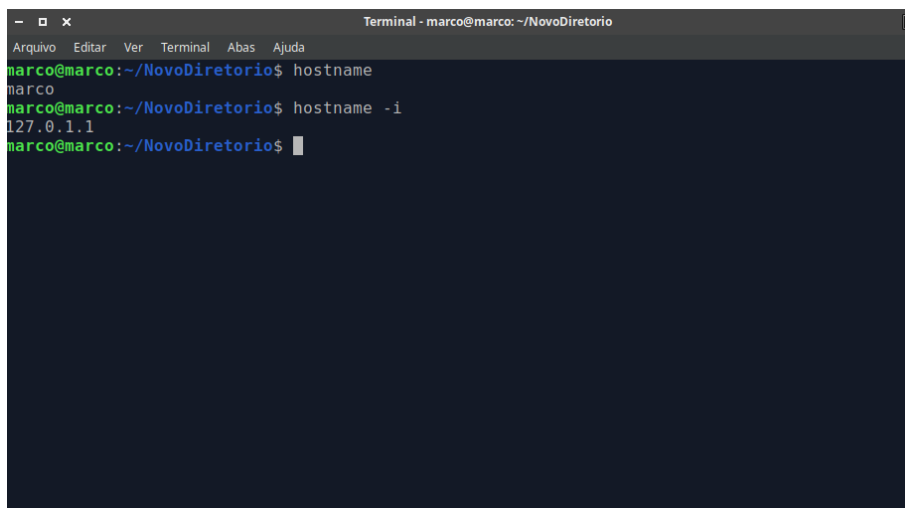
O comando *hostname* exibe o nome do host (nome da máquina) no qual o sistema está sendo executado.

A terminal window titled "Terminal - marco@marco: ~/NovoDiretorio" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The terminal shows the following commands and output:

```
marco@marco:~/NovoDiretorio$ hostname
marco
marco@marco:~/NovoDiretorio$
```

❖ *hostname -i*

O comando *hostname -i* exibe o endereço IP associado ao nome do host atual. Ele mostra o IP local da máquina no sistema de rede.



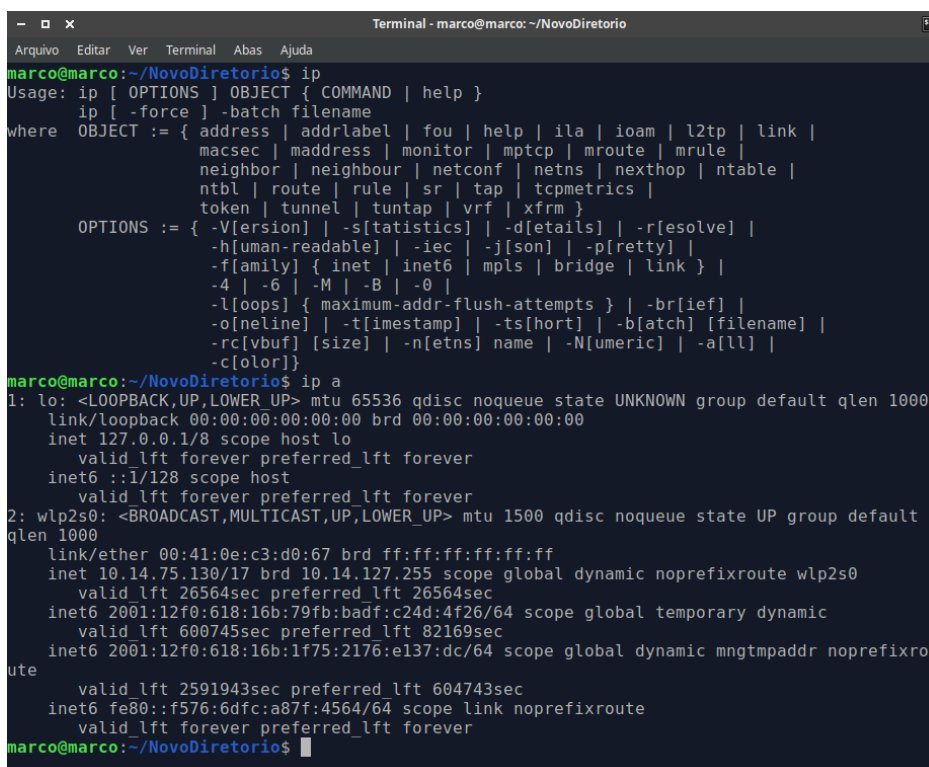
```

Terminal - marco@marco: ~/NovoDiretorio
marco@marco:~/NovoDiretorio$ hostname
marco
marco@marco:~/NovoDiretorio$ hostname -i
127.0.1.1
marco@marco:~/NovoDiretorio$

```

❖ *ip a*

O comando *ip a* exibe informações detalhadas sobre todas as interfaces de rede disponíveis, incluindo endereços IP, estados das interfaces, e outras configurações de rede. É um comando fundamental para visualizar a configuração de rede no Linux.



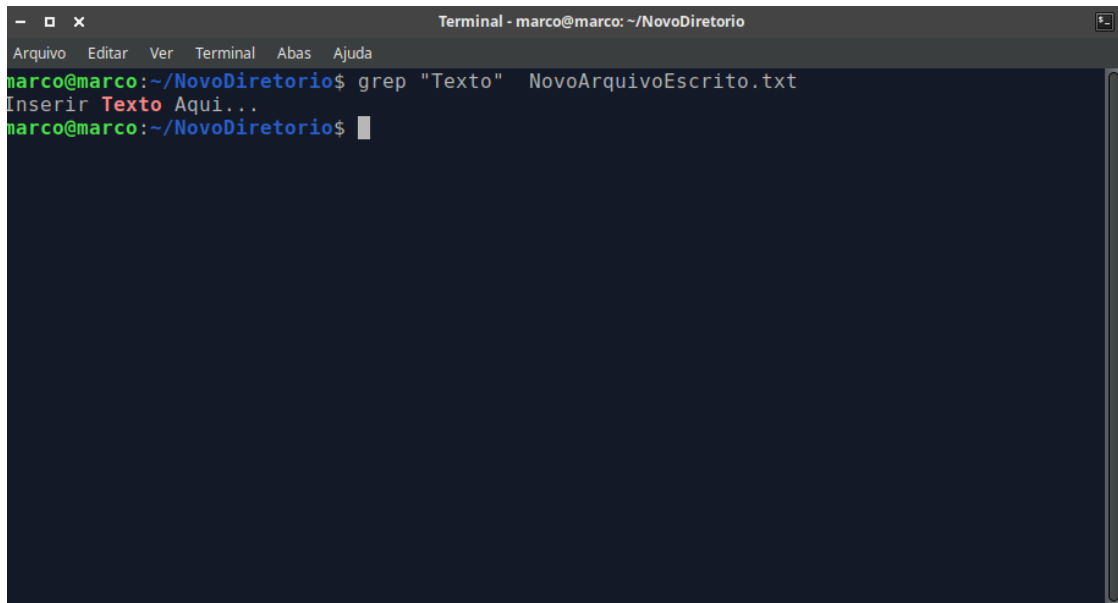
```

Terminal - marco@marco: ~/NovoDiretorio
marco@marco:~/NovoDiretorio$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { address | addrlabel | fou | help | ila | ioam | l2tp | link |
                 macsec | maddress | monitor | mptcp | mroute | mrule |
                 neighbor | neighbour | netconf | netns | nexthop | ntable |
                 ntbl | route | rule | sr | tap | tcpmetrics |
                 token | tunnel | tuntap | vrf | xfrm }
       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                  -h[uman-readable] | -iec | -j[son] | -p[retty] |
                  -f[amily] { inet | inet6 | mpls | bridge | link } |
                  -4 | -6 | -M | -B | -0 |
                  -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                  -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                  -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                  -c[olor]}
marco@marco:~/NovoDiretorio$ 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: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
   qlen 1000
   link/ether 00:41:0e:c3:d0:67 brd ff:ff:ff:ff:ff:ff
   inet 10.14.75.130/17 brd 10.14.127.255 scope global dynamic noprefixroute wlp2s0
       valid_lft 26564sec preferred_lft 26564sec
   inet6 2001:12f0:618:16b:79fb:badf:c24d:4f26/64 scope global temporary dynamic
       valid_lft 600745sec preferred_lft 82169sec
   inet6 2001:12f0:618:16b:1f75:2176:e137:dc/64 scope global dynamic mngtmpaddr noprefixro
       ute
       valid_lft 2591943sec preferred_lft 604743sec
   inet6 fe80::f576:6dfc:a87f:4564/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
marco@marco:~/NovoDiretorio$

```

❖ *grep*

O comando *grep* busca por padrões específicos de texto dentro de arquivos ou na saída de outros comandos. Ele é usado para filtrar e exibir apenas as linhas que correspondem ao padrão fornecido.



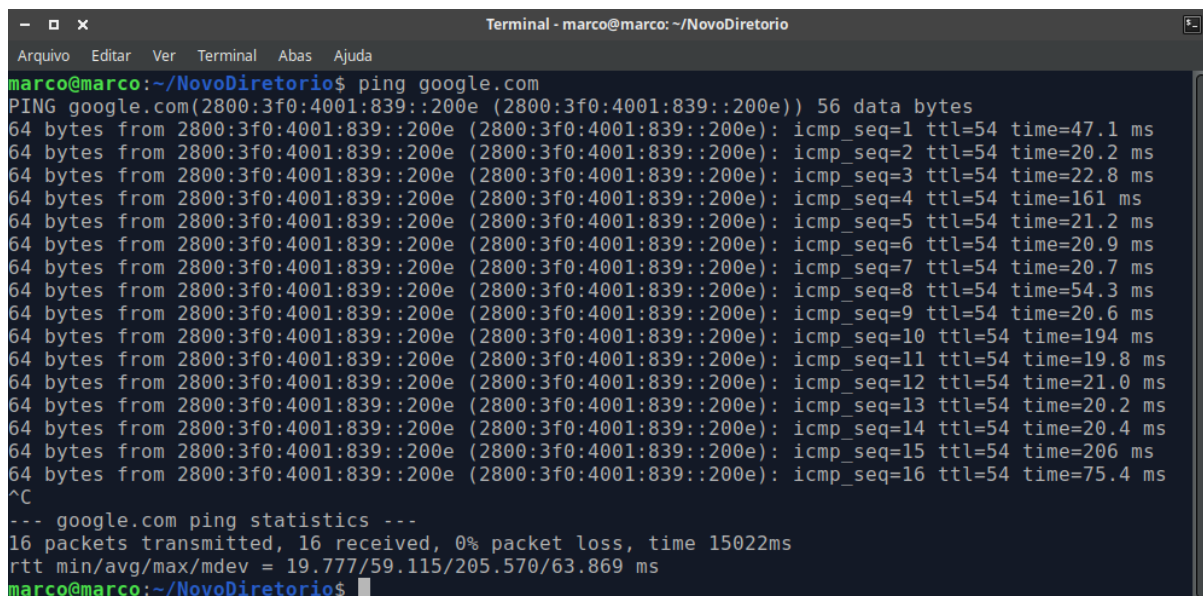
```

Terminal - marco@marco: ~/NovoDiretorio
marco@marco:~/NovoDiretorio$ grep "Texto" NovoArquivoEscrito.txt
Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$

```

❖ *ping*

O comando *ping* envia pacotes ICMP para verificar a conectividade entre o seu computador e outro dispositivo na rede, medindo o tempo de resposta. Ele é comumente usado para testar a disponibilidade de um host.



```

Terminal - marco@marco: ~/NovoDiretorio
marco@marco:~/NovoDiretorio$ ping google.com
PING google.com(2800:3f0:4001:839::200e (2800:3f0:4001:839::200e)) 56 data bytes
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=1 ttl=54 time=47.1 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=2 ttl=54 time=20.2 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=3 ttl=54 time=22.8 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=4 ttl=54 time=161 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=5 ttl=54 time=21.2 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=6 ttl=54 time=20.9 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=7 ttl=54 time=20.7 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=8 ttl=54 time=54.3 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=9 ttl=54 time=20.6 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=10 ttl=54 time=194 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=11 ttl=54 time=19.8 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=12 ttl=54 time=21.0 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=13 ttl=54 time=20.2 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=14 ttl=54 time=20.4 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=15 ttl=54 time=206 ms
64 bytes from 2800:3f0:4001:839::200e (2800:3f0:4001:839::200e): icmp_seq=16 ttl=54 time=75.4 ms
^C
--- google.com ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15022ms
rtt min/avg/max/mdev = 19.777/59.115/205.570/63.869 ms
marco@marco:~/NovoDiretorio$

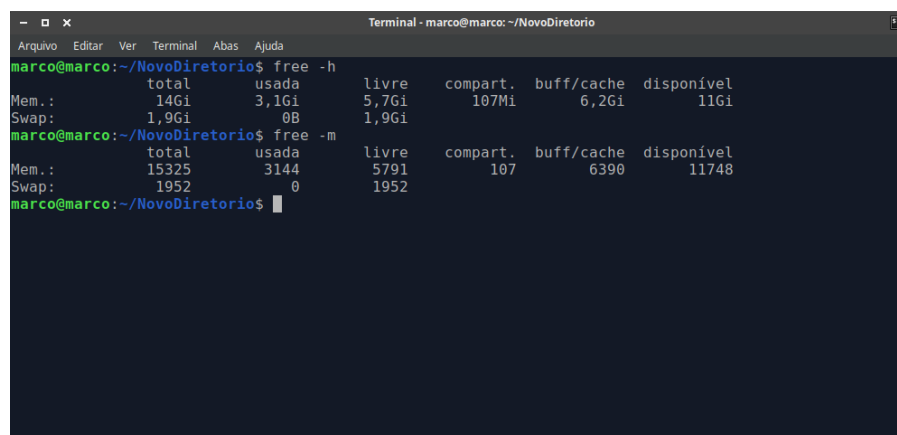
```


❖ *free -h & free -m*

Os comandos *free -h* & *free -m* exibem informações sobre o uso de memória no sistema, mas com diferentes unidades:

- *free -h*: Mostra o uso de memória em um formato legível, como KB, MB ou GB, dependendo do tamanho.
- *free -m*: Exibe o uso de memória especificamente em megabytes (MB).

Ambos mostram informações sobre memória total, usada, livre e swap.



```

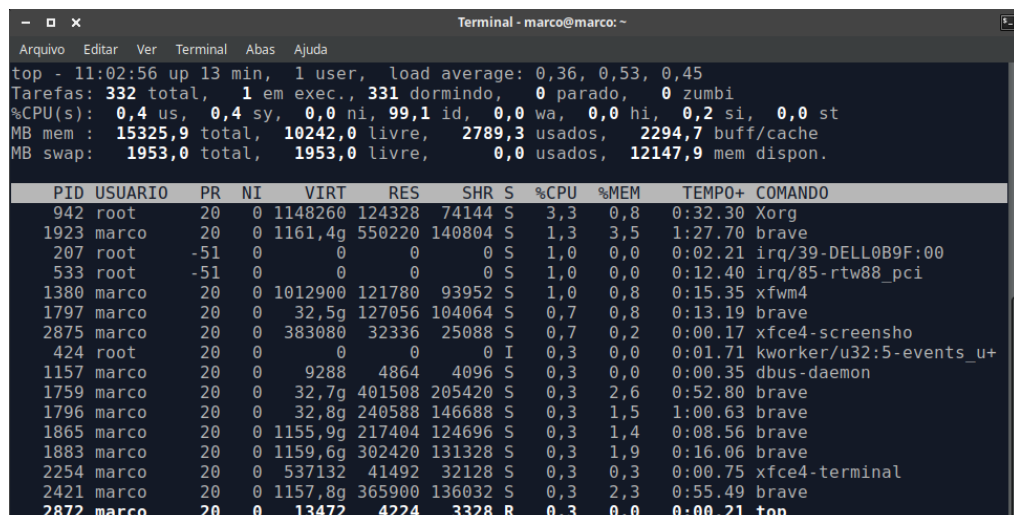
Terminal - marco@marco: ~/NovoDiretorio
marco@marco:~/NovoDiretorio$ free -h
               total        usada        livre      compart.  buff/cache    disponível
Mem.:          14Gi        3,1Gi        5,7Gi        107Mi        6,2Gi        11Gi
Swap:           1,9Gi           0B        1,9Gi
marco@marco:~/NovoDiretorio$ free -m
               total        usada        livre      compart.  buff/cache    disponível
Mem.:         15325        3144        5791         107        6390        11748
Swap:           1952           0        1952
marco@marco:~/NovoDiretorio$

```

❖ *top*

O comando *top* exibe uma visão em tempo real dos processos em execução no sistema, mostrando o uso de CPU, memória, e outras métricas essenciais. Ele permite monitorar e gerenciar o desempenho do sistema diretamente pelo terminal.

Comando utilizado para exemplo: *top*



```

Terminal - marco@marco: ~
top - 11:02:56 up 13 min, 1 user, load average: 0,36, 0,53, 0,45
Tarefas: 332 total, 1 em exec., 331 dormindo, 0 parado, 0 zumbi
%CPU(s): 0,4 us, 0,4 sy, 0,0 ni, 99,1 id, 0,0 wa, 0,0 hi, 0,2 si, 0,0 st
MB mem: 15325,9 total, 10242,0 livre, 2789,3 usados, 2294,7 buff/cache
MB swap: 1953,0 total, 1953,0 livre, 0,0 usados, 12147,9 mem dispon.

  PID USUARIO   PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TEMPO+  COMANDO
  942 root       20   0 1148260 124328 74144 S   3,3   0,8   0:32.30 Xorg
 1923 marco     20   0 1161,4g 550220 140804 S   1,3   3,5   1:27.70 brave
 207 root      -51   0      0      0      0 S   1,0   0,0   0:02.21 irq/39-DELL0B9F:00
 533 root      -51   0      0      0      0 S   1,0   0,0   0:12.40 irq/85-rtw88_pci
1380 marco     20   0 1012900 121780 93952 S   1,0   0,8   0:15.35 xfwm4
1797 marco     20   0 32,5g 127056 104064 S   0,7   0,8   0:13.19 brave
2875 marco     20   0 383080 32336 25088 S   0,7   0,2   0:00.17 xfce4-screensho
 424 root       20   0      0      0      0 I   0,3   0,0   0:01.71 kworker/u32:5-events_u+
1157 marco     20   0 9288 4864 4096 S   0,3   0,0   0:00.35 dbus-daemon
1759 marco     20   0 32,7g 401508 205420 S   0,3   2,6   0:52.80 brave
1796 marco     20   0 32,8g 240588 146688 S   0,3   1,5   1:00.63 brave
1865 marco     20   0 1155,9g 217404 124696 S   0,3   1,4   0:08.56 brave
1883 marco     20   0 1159,6g 302420 131328 S   0,3   1,9   0:16.06 brave
2254 marco     20   0 537132 41492 32128 S   0,3   0,3   0:00.75 xfce4-terminal
2421 marco     20   0 1157,8g 365900 136032 S   0,3   2,3   0:55.49 brave
2872 marco     20   0 13472 4224 3328 R   0,3   0,0   0:00.21 top

```

❖ *htop*

O comando *htop* é uma versão aprimorada do *top*, oferecendo uma interface visual mais amigável e colorida para monitorar processos e recursos do sistema em tempo real.

```

Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda

0[ ] 4.4% 3[ ] 0.0% 6[ ] 2.0% 9[ ] 0.0%
1[ ] 3.3% 4[ ] 0.7% 7[ ] 1.3% 10[ ] 2.0%
2[ ] 1.4% 5[ ] 0.0% 8[ ] 2.0% 11[ ] 0.0%
Mem[ ] 2.91G/15.0G Tasks: 134, 616 thr; 1 running
Swp[ ] 0K/1.91G Load average: 0.21 0.52 0.52
Uptime: 00:25:01

  PID USER   PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
  ---
  942 root    20    0 1121M  121M  74180 S   4.6  0.8  0:57.37 /usr/lib/xorg/Xorg -core :0 -s
 3189 marco   20    0 12048  5632  3584 R   2.0  0.0  0:00.23 htop
 2026 marco   20    0 1157G  254M  123M S   1.3  1.7  0:19.47 /opt/brave.com/brave/brave --t
 3190 marco   20    0 446M  32656  25216 S   1.3  0.2  0:00.13 xfce4-screenshooter -r
  965 root    20    0 1121M  121M  74180 S   0.7  0.8  0:07.36 /usr/lib/xorg/Xorg -core :0 -s
 1380 marco   20    0 989M  118M  93952 S   0.7  0.8  0:26.45 xfwm4 --replace
 1433 marco   20    0 641M  43588  35072 S   0.7  0.3  0:01.30 /usr/lib/x86_64-linux-gnu/xfce
 1759 marco   20    0 33.5G  387M  204M S   0.7  2.5  1:14.18 /opt/brave.com/brave/brave
 1792 marco   20    0 33.5G  387M  204M S   0.7  2.5  0:00.35 /opt/brave.com/brave/brave
 1797 marco   20    0 33.2G  124M  101M S   0.7  0.8  0:18.42 /opt/brave.com/brave/brave --t
 1804 marco   20    0 33.2G  124M  101M S   0.7  0.8  0:16.26 /opt/brave.com/brave/brave --t
 1865 marco   20    0 1155G  212M  122M S   0.7  1.4  0:14.86 /opt/brave.com/brave/brave --t
 1869 marco   20    0 1155G  212M  122M S   0.7  1.4  0:00.61 /opt/brave.com/brave/brave --t
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit

```

❖ *ps*

O comando *ps* exibe uma lista dos processos em execução no sistema, fornecendo informações como ID do processo (PID), tempo, usuário.

```

Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda

marco@marco:~$ ps
  PID TTY          TIME CMD
  2279 pts/0        00:00:00 bash
  3261 pts/0        00:00:00 ps
marco@marco:~$

```

❖ *ps aux*

O comando *ps aux* exibe uma lista detalhada de todos os processos em execução no sistema, independentemente de qual usuário os iniciou, mostrando informações como PID, usuário, uso de CPU e memória, e o comando que iniciou cada processo.

```

Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

marco@marco:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 168056 12756 ?        Ss   10:49   0:01 /sbin/init splash
root         2  0.0  0.0      0     0 ?        S    10:49   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   10:49   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   10:49   0:00 [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<   10:49   0:00 [slub_flushwq]
root         6  0.0  0.0      0     0 ?        I<   10:49   0:00 [netns]
root         9  0.0  0.0      0     0 ?        I    10:49   0:00 [kworker/0:1-events]
root        11  0.0  0.0      0     0 ?        I<   10:49   0:00 [mm_percpu_wq]
root        12  0.0  0.0      0     0 ?        I    10:49   0:00 [rcu_tasks_kthread]
root        13  0.0  0.0      0     0 ?        I    10:49   0:00 [rcu_tasks_rude_kthread]
root        14  0.0  0.0      0     0 ?        I    10:49   0:00 [rcu_tasks_trace_kthread]
root        15  0.0  0.0      0     0 ?        S    10:49   0:00 [ksoftirqd/0]
root        16  0.1  0.0      0     0 ?        I    10:49   0:02 [rcu_preempt]
root        17  0.0  0.0      0     0 ?        S    10:49   0:00 [migration/0]
root        18  0.0  0.0      0     0 ?        S    10:49   0:00 [idle_inject/0]
root        19  0.0  0.0      0     0 ?        S    10:49   0:00 [cpuhp/0]
root        20  0.0  0.0      0     0 ?        S    10:49   0:00 [cpuhp/2]
root        21  0.0  0.0      0     0 ?        S    10:49   0:00 [idle_inject/2]
root        22  0.0  0.0      0     0 ?        S    10:49   0:00 [migration/2]
root        23  0.0  0.0      0     0 ?        S    10:49   0:00 [ksoftirqd/2]
root        25  0.0  0.0      0     0 ?        I<   10:49   0:00 [kworker/2:0H-events_highpri]

```

❖ *kill*

O comando *kill* encerra um processo específico usando seu ID (PID).

```

Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

marco@marco:~$ ps aux | grep gnome
marco      1153  0.0  0.0 241272  7324 ?        Sl   10:49   0:00 /usr/bin/gnome-keyring-daemon --d
aemonize --login
marco      1443  0.0  0.1 187916 17152 ?        Sl   10:49   0:00 /usr/lib/policykit-1-gnome/polkit
-gnome-authentication-agent-1
marco      3468  0.0  0.0   9240   2432 pts/0    S+   11:26   0:00 grep --color=auto gnome
marco@marco:~$ kill 1153
marco@marco:~$ ps aux | grep gnome
marco      1443  0.0  0.1 187916 17152 ?        Sl   10:49   0:00 /usr/lib/policykit-1-gnome/polkit
-gnome-authentication-agent-1
marco      3473  0.0  0.0   9240   2432 pts/0    S+   11:26   0:00 grep --color=auto gnome
marco@marco:~$

```

❖ *df -h*

O comando *df -h* exibe o uso de espaço em disco, mostrando o espaço total, usado e disponível em cada sistema de arquivos.

```

Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$ df -h
Sist. Arq.      Tam. Usado Disp.  Uso% Montado em
tmpfs          1,5G  1,9M  1,5G   1% /run
/dev/nvme0n1p11 19G   15G  3,3G  82% /
tmpfs          7,5G  102M  7,4G   2% /dev/shm
tmpfs          5,0M   4,0K  5,0M   1% /run/lock
efivarfs       148K   89K   55K  62% /sys/firmware/efi/efivars
/dev/nvme0n1p8 431M  188M  210M  48% /boot
/dev/nvme0n1p10 48G   2,4G   43G   6% /home
/dev/nvme0n1p1 196M  100M   97M  51% /boot/efi
tmpfs          1,5G   96K  1,5G   1% /run/user/1000
marco@marco:~$

```

❖ *ncdu*

O comando *ncdu* é uma ferramenta interativa que exibe o uso de disco de forma detalhada e navegável. Ele analisa e mostra o tamanho de diretórios e arquivos, permitindo ao usuário explorar e identificar quais itens estão ocupando mais espaço no sistema.

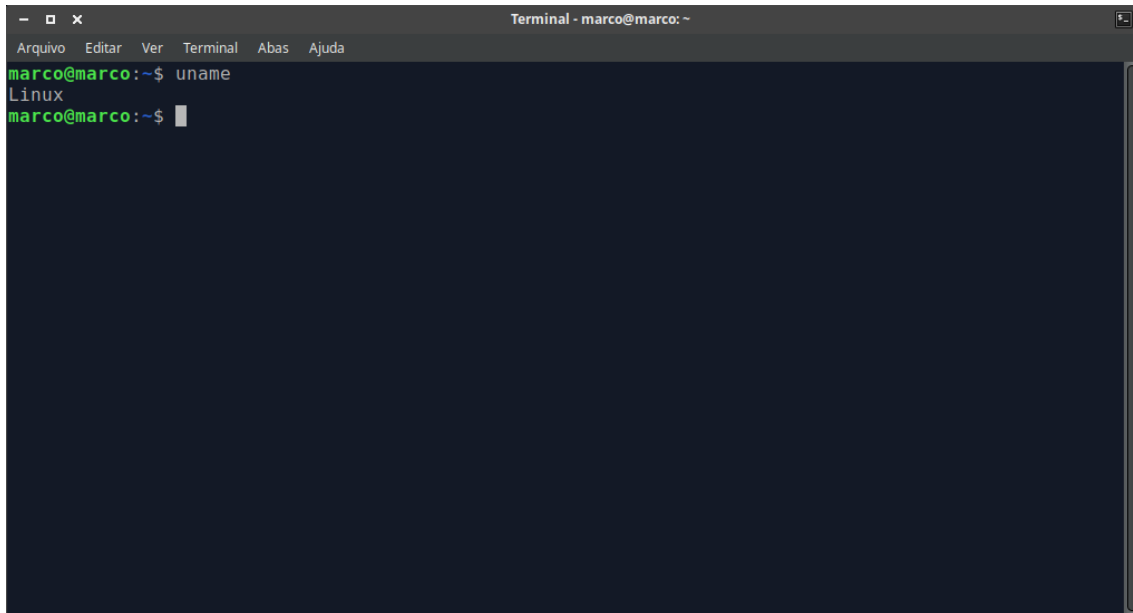
```

Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
ncdu 1.15.1 ~ Use the arrow keys to navigate, press ? for help
--- /home/marco -----
1,0 GiB [#####] /wine
793,1 MiB [#####] /config
573,2 MiB [#####] /cache
84,7 MiB [#####] /snap
1,7 MiB [#####] /vscode
768,0 KiB [#####] /local
456,0 KiB [#####] /Downloads
340,0 KiB [#####] /Área de Trabalho
304,0 KiB [#####] /code
148,0 KiB [#####] /dotnet
116,0 KiB [#####] /git
76,0 KiB [#####] /pki
44,0 KiB [#####] /xsession-errors.old
28,0 KiB [#####] /ssh
20,0 KiB [#####] /Modelos
16,0 KiB [#####] /xsession-errors
12,0 KiB [#####] /bash_history
12,0 KiB [#####] /NovoDiretorio
e 4,0 KiB [#####] /Vídeos
e 4,0 KiB [#####] /Público
e 4,0 KiB [#####] /Música
e 4,0 KiB [#####] /Imagens
e 4,0 KiB [#####] /Documentos
Total disk usage: 2.4 GiB Apparent size: 2.3 GiB Items: 25428

```

❖ *uname*

O comando *uname* exibe informações básicas sobre o sistema, como o nome do kernel.



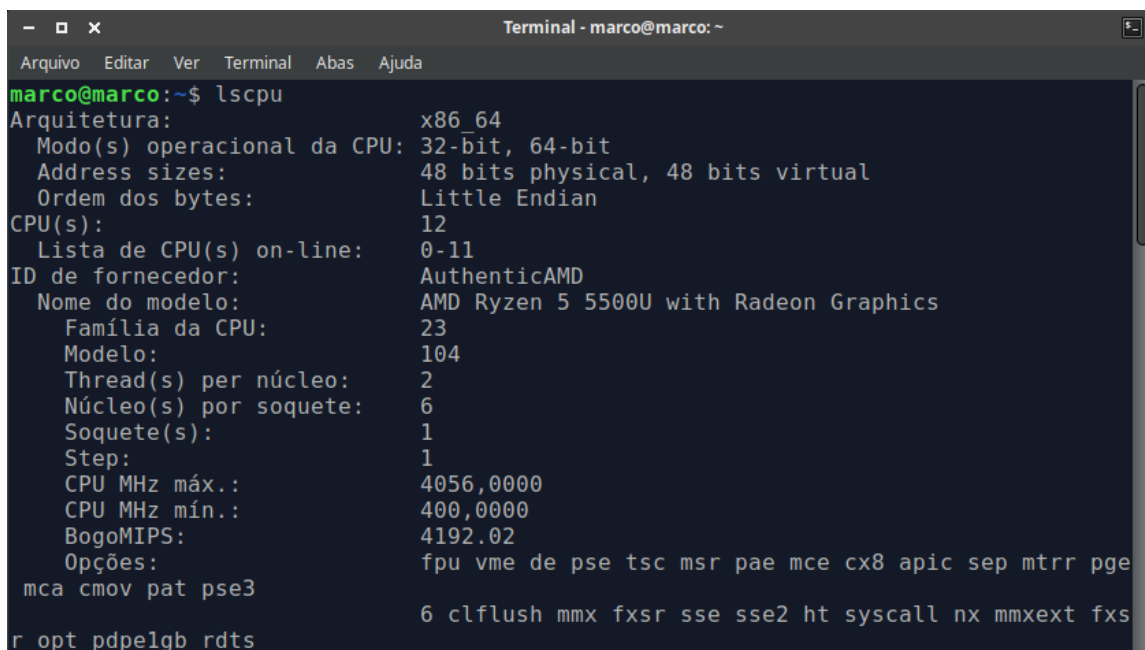
```

Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ uname
Linux
marco@marco:~$

```

❖ *lscpu* & *lsusb*

Os comandos *lscpu* & *lsusb* fornecem informações detalhadas sobre o hardware do sistema. O *lscpu* exibe detalhes sobre a CPU, como arquitetura e número de núcleos, enquanto o *lsusb* lista todos os dispositivos USB conectados, incluindo IDs e descrições.



```

Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ lscpu
Arquitetura:                x86_64
Modo(s) operacional da CPU: 32-bit, 64-bit
Address sizes:              48 bits physical, 48 bits virtual
Ordem dos bytes:            Little Endian
CPU(s):                     12
Lista de CPU(s) on-line:    0-11
ID de fornecedor:           AuthenticAMD
Nome do modelo:             AMD Ryzen 5 5500U with Radeon Graphics
Família da CPU:             23
Modelo:                     104
Thread(s) per núcleo:       2
Núcleo(s) por soquete:      6
Soquete(s):                 1
Step:                       1
CPU MHz máx.:               4056,0000
CPU MHz mín.:               400,0000
BogoMIPS:                   4192.02
Opções:                     fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse3          6 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr
r opt pdpe1gb rdtsc

```

```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 003: ID 0c45:6d1a Microdia Integrated Webcam HD
Bus 003 Device 002: ID 0bda:c829 Realtek Semiconductor Corp. Bluetooth Radio
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 27c6:639c Shenzhen Goodix Technology Co.,Ltd. Goodix USB2.0
MISC
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
marco@marco:~$
```

❖ *history*

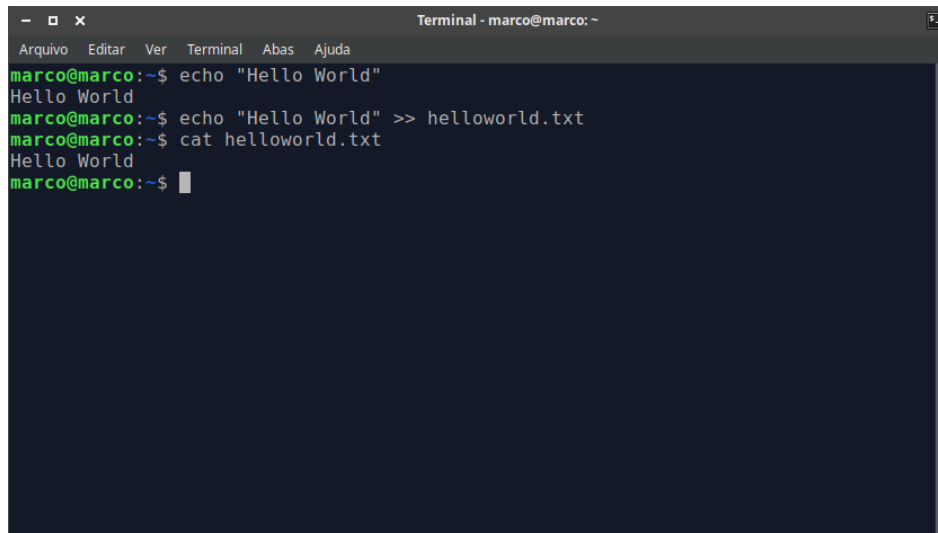
O comando *history* exibe a lista de comandos previamente executados no terminal, permitindo que você visualize e reutilize comandos passados.

```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ history
 1 sudo update-grub
 2 cd
 3 cd ls
 4 ls
 5 cd 'Área de Trabalho'
 6 cd
 7 /
 8 exit
 9 ls
10 dir
11 ls
12 cd
13 ls
14 dir
15 sudo app-ge
16 sudo apt-get install chrome
17 sudo apt-get install brave-browser
18 sudo apt install brave-browser
19 sudo apt install curl
20 sudo apt update
21 sudo apt install brave-browser
```

3. COMANDOS DO TERMINAL LINUX MAIS POPULARES

❖ *echo*

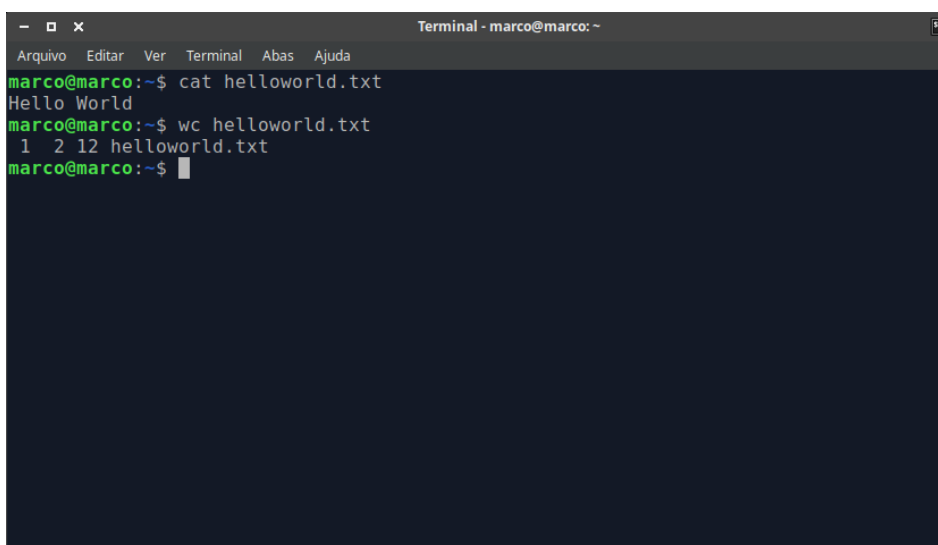
O comando *echo* exibe uma linha de texto ou variável no terminal. Ele é frequentemente usado para mostrar mensagens, valores de variáveis, ou redirecionar texto para arquivos.

A terminal window titled "Terminal - marco@marco: ~" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The terminal shows the following commands and output:

```
marco@marco:~$ echo "Hello World"
Hello World
marco@marco:~$ echo "Hello World" >> helloworld.txt
marco@marco:~$ cat helloworld.txt
Hello World
marco@marco:~$
```

❖ *wc*

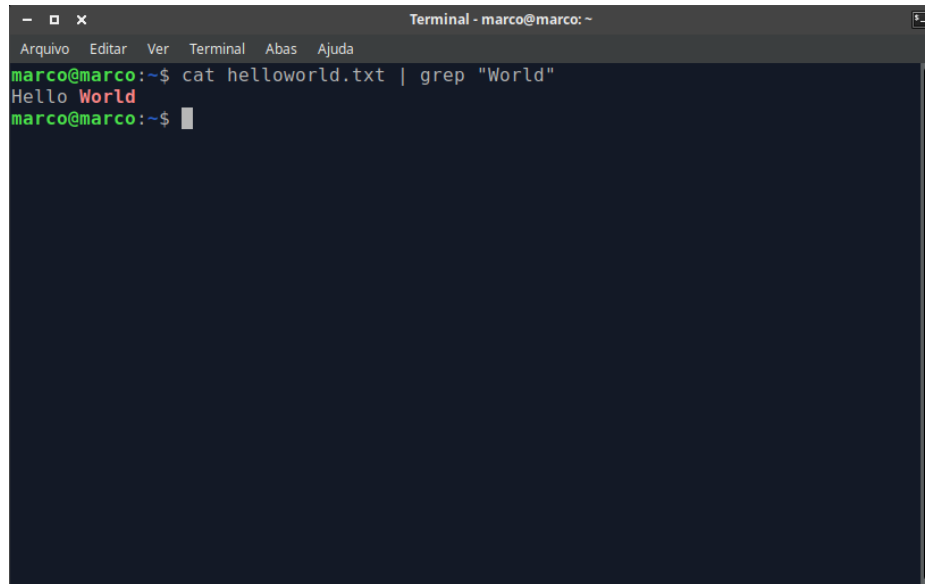
O comando *wc* conta o número de linhas, palavras e caracteres em um arquivo ou entrada de texto no terminal.

A terminal window titled "Terminal - marco@marco: ~" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The terminal shows the following commands and output:

```
marco@marco:~$ cat helloworld.txt
Hello World
marco@marco:~$ wc helloworld.txt
 1  2 12 helloworld.txt
marco@marco:~$
```

❖ *piping* |

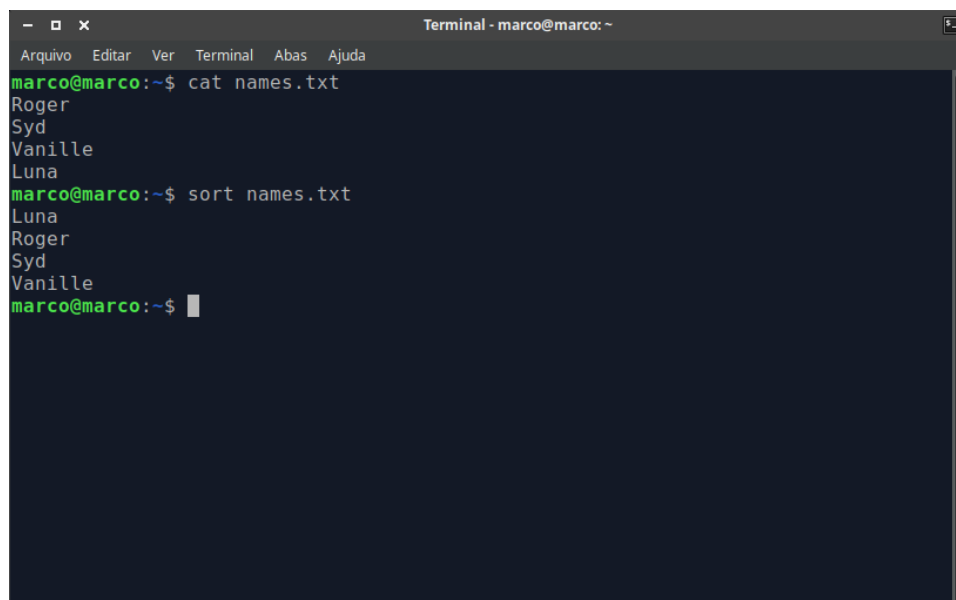
O *piping*, representado pelo operador |, conecta a saída de um comando à entrada de outro, permitindo encadear múltiplos comandos em uma única linha.



```
Terminal - marco@marco: ~  
Arquivo Editar Ver Terminal Abas Ajuda  
marco@marco:~$ cat helloworld.txt | grep "World"  
Hello World  
marco@marco:~$
```

❖ *sort*

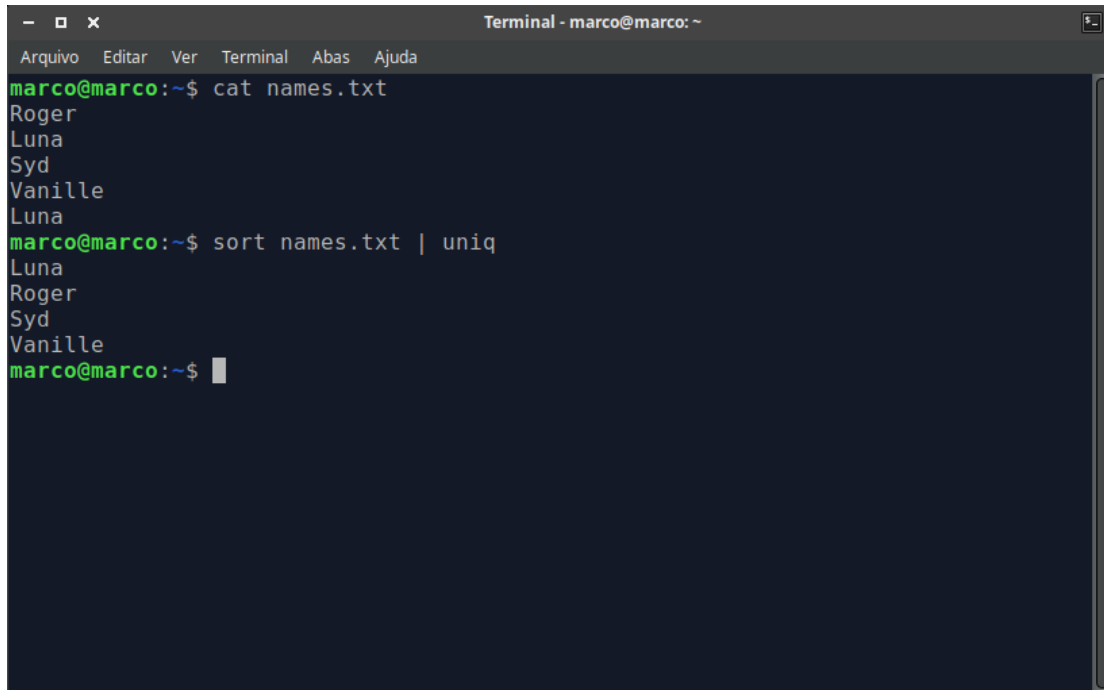
O comando *sort* organiza linhas de texto ou dados em um arquivo ou entrada em ordem alfabética ou numérica.



```
Terminal - marco@marco: ~  
Arquivo Editar Ver Terminal Abas Ajuda  
marco@marco:~$ cat names.txt  
Roger  
Syd  
Vanille  
Luna  
marco@marco:~$ sort names.txt  
Luna  
Roger  
Syd  
Vanille  
marco@marco:~$
```


❖ *uniq*

O comando *uniq* filtra linhas duplicadas em um arquivo ou entrada de texto, mostrando apenas linhas únicas ou destacando repetições.

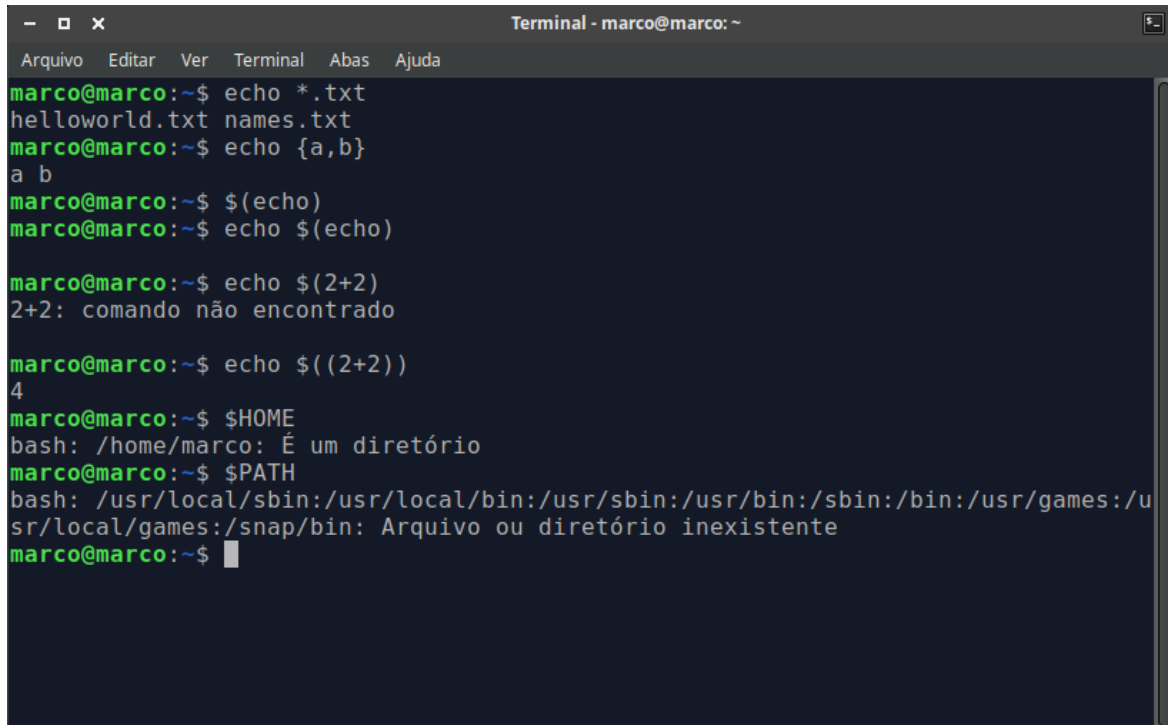
A terminal window titled "Terminal - marco@marco: ~" with a menu bar containing "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The prompt is "marco@marco:~\$". The first command is "cat names.txt", which outputs: "Roger", "Luna", "Syd", "Vanille", "Luna". The second command is "sort names.txt | uniq", which outputs: "Luna", "Roger", "Syd", "Vanille". The prompt returns to "marco@marco:~\$".

```
marco@marco:~$ cat names.txt
Roger
Luna
Syd
Vanille
Luna
marco@marco:~$ sort names.txt | uniq
Luna
Roger
Syd
Vanille
marco@marco:~$
```

❖ *expansions*

Em Linux, *expansions* refere-se a uma série de mecanismos que o shell usa para transformar ou expandir comandos antes de executá-los. Isso inclui:

- i. Pathname Expansion: Expande padrões de arquivo, como **.txt*, para corresponder a todos os arquivos que correspondem ao padrão.
- ii. Brace Expansion: Gera strings com padrões, como *{a,b}* que expande para *a* e *b*.
- iii. Command Substitution: Substitui a saída de um comando, como *\$(command)* ou *`command`*.
- iv. Arithmetic Expansion: Executa cálculos aritméticos, como *echo \$((2 + 2))*.
- v. Variable Expansion: Expande variáveis, como *\$HOME* para mostrar o valor armazenado na variável *HOME*.



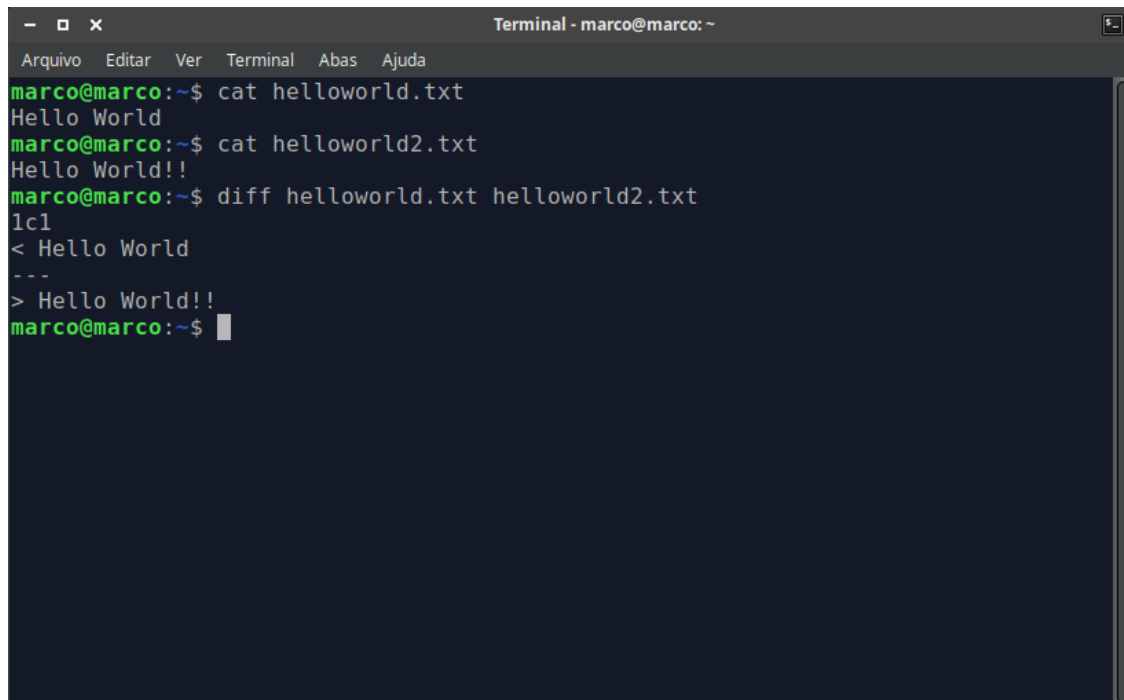
```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ echo *.txt
helloworld.txt names.txt
marco@marco:~$ echo {a,b}
a b
marco@marco:~$ $(echo)
marco@marco:~$ echo $(echo)

marco@marco:~$ echo $(2+2)
2+2: comando não encontrado

marco@marco:~$ echo $((2+2))
4
marco@marco:~$ $HOME
bash: /home/marco: É um diretório
marco@marco:~$ $PATH
bash: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin: Arquivo ou diretório inexistente
marco@marco:~$
```

❖ *diff*

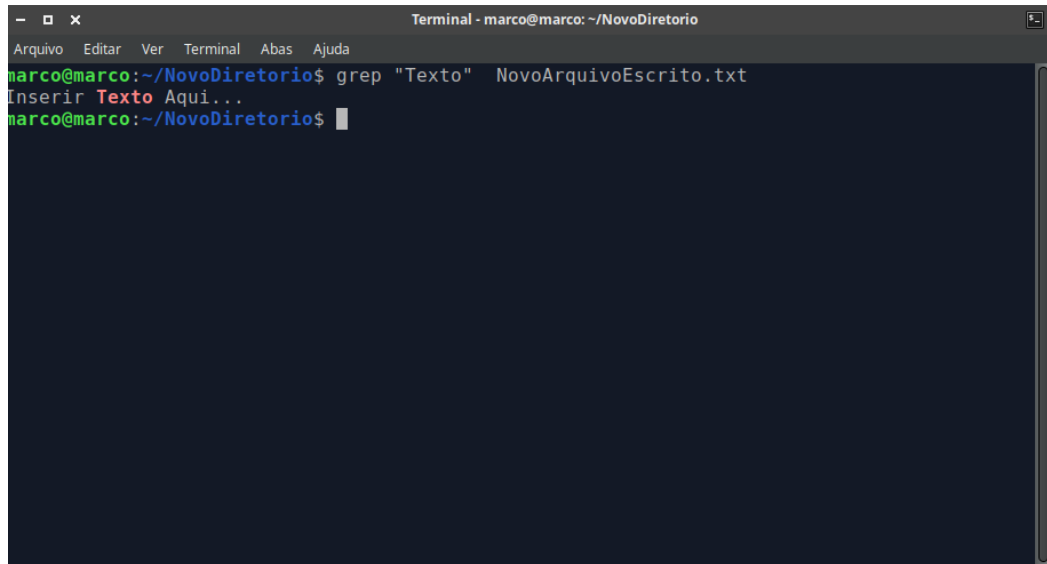
O comando *diff* compara o conteúdo de dois arquivos linha por linha, exibindo as diferenças entre eles.



```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ cat helloworld.txt
Hello World
marco@marco:~$ cat helloworld2.txt
Hello World!!
marco@marco:~$ diff helloworld.txt helloworld2.txt
1c1
< Hello World
---
> Hello World!!
marco@marco:~$
```

❖ *grep*

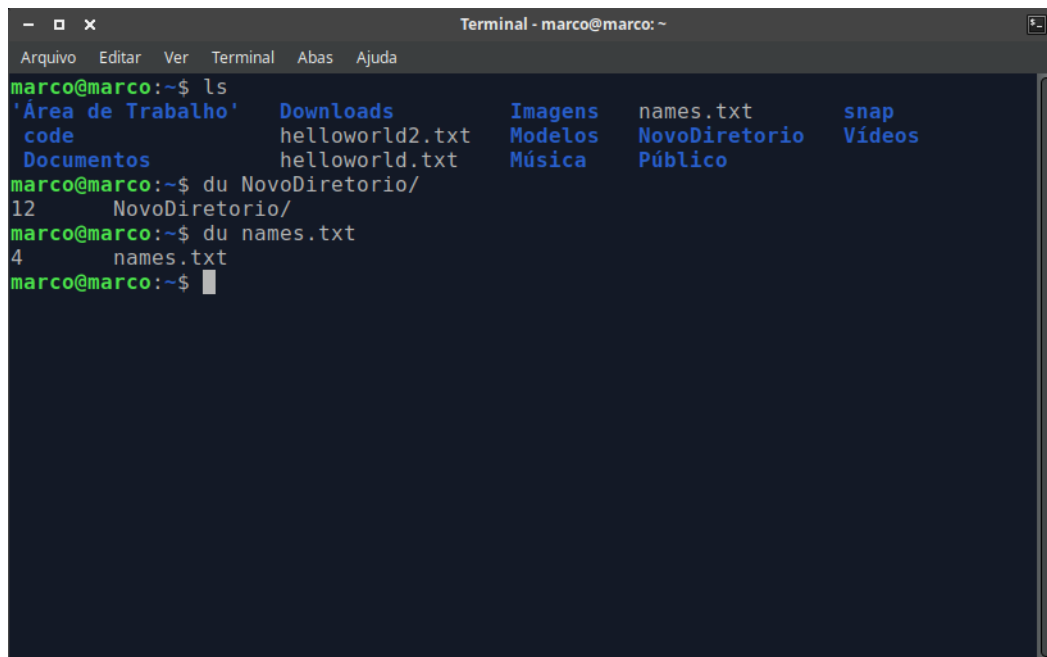
O comando *grep* busca por padrões específicos de texto dentro de arquivos ou na saída de outros comandos. Ele é usado para filtrar e exibir apenas as linhas que correspondem ao padrão fornecido.



```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ grep "Texto" NovoArquivoEscrito.txt
Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$
```

❖ *du*

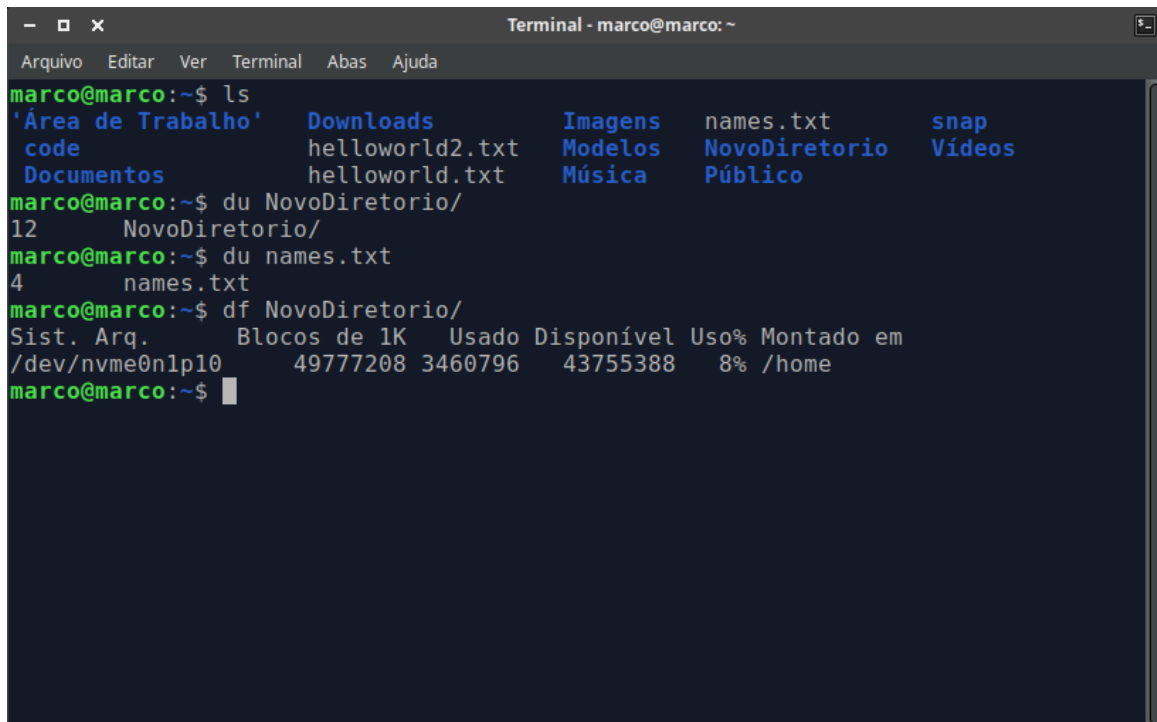
O comando *du* calcula e exibe o uso de espaço em disco de diretórios e arquivos, mostrando o tamanho de cada um.



```
Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$ ls
'Área de Trabalho' Downloads Imagens names.txt snap
code helloworld2.txt Modelos NovoDiretorio Vídeos
Documentos helloworld.txt Música Público
marco@marco:~$ du NovoDiretorio/
12 NovoDiretorio/
marco@marco:~$ du names.txt
4 names.txt
marco@marco:~$
```

❖ *df*

O comando *df* exibe o uso do espaço em disco para todos os sistemas de arquivos montados, mostrando informações como espaço total, usado e disponível.

A terminal window titled "Terminal - marco@marco: ~" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The user runs 'ls' showing a directory listing with files like 'Área de Trabalho', 'Downloads', 'Imagens', 'names.txt', 'snap', 'code', 'helloworld2.txt', 'Modelos', 'NovoDiretorio', 'Vídeos', 'Documentos', 'helloworld.txt', and 'Música'. Then, the user runs 'du NovoDiretorio/' showing 12 blocks for the directory and 'du names.txt' showing 4 blocks for the file. Finally, the user runs 'df NovoDiretorio/' displaying disk usage statistics for the /dev/nvme0n1p10 partition.

```
marco@marco:~$ ls
'Área de Trabalho'  Downloads  Imagens  names.txt  snap
code                helloworld2.txt  Modelos  NovoDiretorio  Vídeos
Documentos          helloworld.txt  Música   Público

marco@marco:~$ du NovoDiretorio/
12      NovoDiretorio/

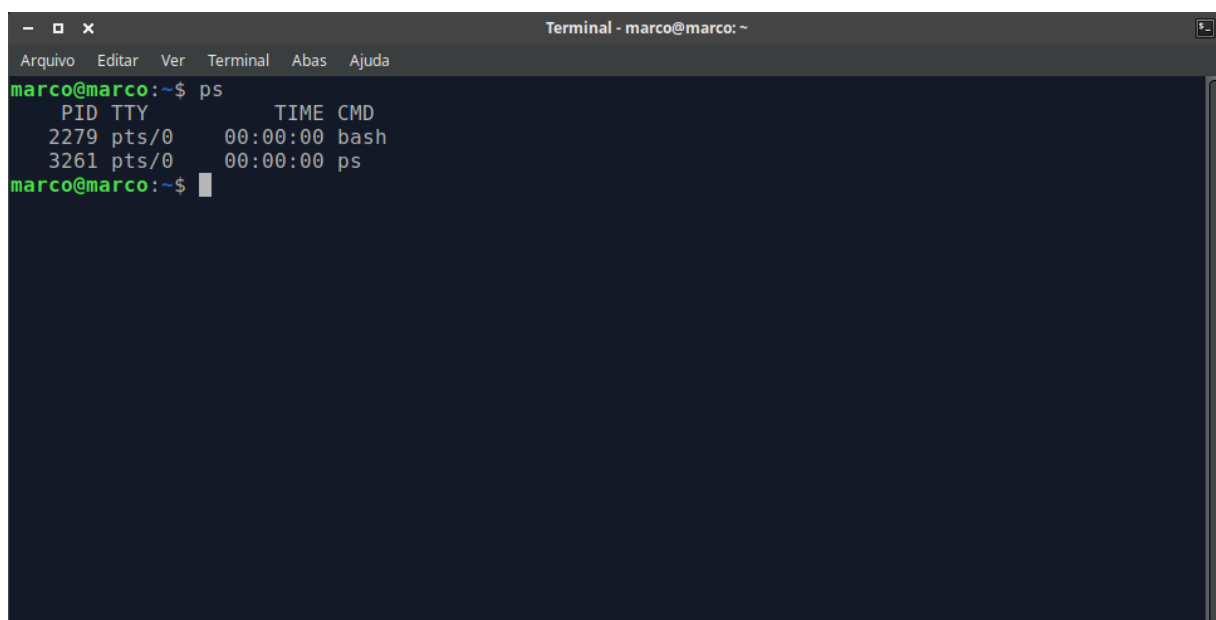
marco@marco:~$ du names.txt
4       names.txt

marco@marco:~$ df NovoDiretorio/
Sist. Arq.      Blocos de 1K  Usado  Disponível  Uso%  Montado em
/dev/nvme0n1p10  49777208  3460796  43755388    8%  /home

marco@marco:~$
```

❖ *ps*

O comando *ps* exibe uma lista dos processos em execução no sistema, fornecendo informações como ID do processo (PID), tempo, usuário.

A terminal window titled "Terminal - marco@marco: ~" with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda). The user runs 'ps', which displays a table of running processes with columns for PID, TTY, TIME, and CMD. Two processes are listed: 'bash' with PID 2279 and 'ps' with PID 3261.

```
marco@marco:~$ ps
  PID TTY          TIME CMD
 2279 pts/0    00:00:00 bash
 3261 pts/0    00:00:00 ps

marco@marco:~$
```

❖ *top*

O comando *top* exibe uma visão em tempo real dos processos em execução no sistema, mostrando o uso de CPU, memória, e outras métricas essenciais. Ele permite monitorar e gerenciar o desempenho do sistema diretamente pelo terminal.

Comando utilizado para exemplo: *top*

```

-  x  Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
top - 11:02:56 up 13 min, 1 user, load average: 0,36, 0,53, 0,45
Tarefas: 332 total, 1 em exec., 331 dormindo, 0 parado, 0 zumbi
%CPU(s): 0,4 us, 0,4 sy, 0,0 ni, 99,1 id, 0,0 wa, 0,0 hi, 0,2 si, 0,0 st
MB mem : 15325,9 total, 10242,0 livre, 2789,3 usados, 2294,7 buff/cache
MB swap: 1953,0 total, 1953,0 livre, 0,0 usados, 12147,9 mem dispon.

  PID  USUARIO  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TEMPO+  COMANDO
  942   root    20   0 1148260 124328 74144 S   3,3   0,8   0:32.30 Xorg
 1923  marco    20   0 1161,4g 550220 140804 S   1,3   3,5   1:27.70 brave
   207   root   -51   0      0      0      0 S   1,0   0,0   0:02.21 irq/39-DELL0B9F:00
   533   root   -51   0      0      0      0 S   1,0   0,0   0:12.40 irq/85-rtw88_pci
 1380  marco    20   0 1012900 121780 93952 S   1,0   0,8   0:15.35 xfwm4
 1797  marco    20   0   32,5g 127056 104064 S   0,7   0,8   0:13.19 brave
 2875  marco    20   0 383080 32336 25088 S   0,7   0,2   0:00.17 xfce4-screensho
   424   root    20   0      0      0      0 I   0,3   0,0   0:01.71 kworker/u32:5-events_u+
 1157  marco    20   0   9288   4864   4096 S   0,3   0,0   0:00.35 dbus-daemon
 1759  marco    20   0   32,7g 401508 205420 S   0,3   2,6   0:52.80 brave
 1796  marco    20   0   32,8g 240588 146688 S   0,3   1,5   1:00.63 brave
 1865  marco    20   0 1155,9g 217404 124696 S   0,3   1,4   0:08.56 brave
 1883  marco    20   0 1159,6g 302420 131328 S   0,3   1,9   0:16.06 brave
 2254  marco    20   0 537132 41492 32128 S   0,3   0,3   0:00.75 xfce4-terminal
 2421  marco    20   0 1157,8g 365900 136032 S   0,3   2,3   0:55.49 brave
 2872  marco    20   0  13472   4224   3328 R   0,3   0,0   0:00.21 top

```

❖ *kill*

O comando *kill* encerra um processo específico usando seu ID (PID).

```

-  x  Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$ ps aux | grep gnome
marco    1153   0.0  0.0 241272  7324 ?        Sl    10:49   0:00 /usr/bin/gnome-keyring-daemon --d
aemonize --login
marco    1443   0.0  0.1 187916 17152 ?        Sl    10:49   0:00 /usr/lib/policykit-1-gnome/polkit
-gnome-authentication-agent-1
marco    3468   0.0  0.0   9240  2432 pts/0    S+   11:26   0:00 grep --color=auto gnome
marco@marco:~$ kill 1153
marco@marco:~$ ps aux | grep gnome
marco    1443   0.0  0.1 187916 17152 ?        Sl    10:49   0:00 /usr/lib/policykit-1-gnome/polkit
-gnome-authentication-agent-1
marco    3473   0.0  0.0   9240  2432 pts/0    S+   11:26   0:00 grep --color=auto gnome
marco@marco:~$

```

❖ *killall*

O comando *killall* encerra todos os processos que correspondem ao nome especificado.

The image shows three terminal windows. The first window shows the output of the `ps a` command, listing running processes. The second window shows the output of the `killall ping` command, which terminates all ping processes. The third window shows the output of the `ping google.com` command, which sends ICMP echo requests to google.com.

```
marco@marco:~$ ps a
PID TTY STAT TIME COMMAND
801 tty1 Ss+ 0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
942 tty7 Ss+ 3:05 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/ro
7886 pts/0 Ss 0:00 bash
8489 pts/1 Ss 0:00 bash
8505 pts/2 Ss 0:00 bash
8807 pts/1 S+ 0:00 ping google.com
8808 pts/2 S+ 0:00 ping google.com
8810 pts/0 R+ 0:00 ps a
marco@marco:~$ killall ping
marco@marco:~$
```

```
marco@marco:~$ ping google.com
PING google.com (142.250.218.206) 56(84) bytes of data:
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=1 ttl=117 time=10.7 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=2 ttl=117 time=10.5 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=3 ttl=117 time=15.8 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=4 ttl=117 time=11.4 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=5 ttl=117 time=15.6 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=6 ttl=117 time=10.8 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=7 ttl=117 time=15.9 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=8 ttl=117 time=15.2 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=9 ttl=117 time=15.3 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=10 ttl=117 time=15.3 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=11 ttl=117 time=15.5 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=12 ttl=117 time=15.0 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=13 ttl=117 time=15.3 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=14 ttl=117 time=17.5 ms
64 bytes from gru6s28-in-f14.1e100.net (172.217.29.238): icmp_seq=15 ttl=117 time=15.0 ms
Terminado
marco@marco:~$
```

```
marco@marco:~$ curl -s -o /dev/null -w '%{url}\n' http://www.google.com
http://www.google.com
marco@marco:~$
```

❖ *Jobs, bg & fg*

Os comandos *jobs*, *bg* & *fg* gerenciam processos em segundo plano e em primeiro plano no terminal:

- jobs*: Lista os processos em segundo plano e os que estão suspensos, exibindo seus números de trabalho e status.
- bg*: Retoma um processo suspenso e o coloca em segundo plano, permitindo que continue sua execução. Por exemplo, `bg %1` retoma o trabalho número 1.
- fg*: Traz um processo em segundo plano para o primeiro plano, permitindo interagir com ele diretamente no terminal. Por exemplo, `fg %1` traz o trabalho número 1 para o primeiro plano.

The image shows a terminal window with the output of the `jobs` command, which lists the background processes.

```
marco@marco:~$ jobs
marco@marco:~$
```

❖ *gzip*

O comando *gzip* compacta arquivos usando o formato de compressão *gzip*. Ele reduz o tamanho dos arquivos, substituindo o arquivo original por uma versão compactada com a extensão *.gz*.

```

marco@marco:~$ ls
'Área de Trabalho'  Downloads  Modelos  NovoDiretorio  Vídeos
code                helloworld.txt  Música  Público
Documentos          Imagens       names.txt  snap
marco@marco:~$ gzip -c helloworld.txt > helloworld.gz
marco@marco:~$ ls
'Área de Trabalho'  Downloads  Imagens  names.txt  snap
code                helloworld.gz  Modelos  NovoDiretorio  Vídeos
Documentos          helloworld.txt  Música  Público
marco@marco:~$ ls -l
total 56
drwxr-xr-x 3 marco marco 4096 ago 13 15:39 'Área de Trabalho'
drwxrwxr-x 3 marco marco 4096 jun 11 18:46 code
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Documentos
drwxr-xr-x 2 marco marco 4096 ago 14 14:47 Downloads
-rw-rw-r-- 1 marco marco 47 ago 14 15:51 helloworld.gz
-rw-rw-r-- 1 marco marco 12 ago 14 12:05 helloworld.txt
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Imagens
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Modelos
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Música
-rw-rw-r-- 1 marco marco 30 ago 14 14:57 names.txt
drwxrwxr-x 2 marco marco 4096 ago 13 21:48 NovoDiretorio
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Público
drwx----- 4 marco marco 4096 jun 11 15:23 snap
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Vídeos
marco@marco:~$

```

❖ *gunzip*

O comando *gunzip* descompacta arquivos que foram compactados com *gzip*, removendo a extensão *.gz* e restaurando o arquivo original.

```

marco@marco:~$ ls
'Área de Trabalho'  Downloads  Imagens  names.txt  snap
code                helloworld.gz  Modelos  NovoDiretorio  Vídeos
Documentos          helloworld.txt  Música  Público
marco@marco:~$ ls -l
total 56
drwxr-xr-x 3 marco marco 4096 ago 13 15:39 'Área de Trabalho'
drwxrwxr-x 3 marco marco 4096 jun 11 18:46 code
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Documentos
drwxr-xr-x 2 marco marco 4096 ago 14 14:47 Downloads
-rw-rw-r-- 1 marco marco 47 ago 14 15:51 helloworld.gz
-rw-rw-r-- 1 marco marco 12 ago 14 12:05 helloworld.txt
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Imagens
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Modelos
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Música
-rw-rw-r-- 1 marco marco 30 ago 14 14:57 names.txt
drwxrwxr-x 2 marco marco 4096 ago 13 21:48 NovoDiretorio
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Público
drwx----- 4 marco marco 4096 jun 11 15:23 snap
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Vídeos
marco@marco:~$ gunzip helloworld.gz
marco@marco:~$ ls
'Área de Trabalho'  Downloads  Imagens  names.txt  snap
code                helloworld  Modelos  NovoDiretorio  Vídeos
Documentos          helloworld.txt  Música  Público
marco@marco:~$

```

❖ *tar*

O comando *tar* é usado para empacotar e comprimir arquivos e diretórios. Ele pode criar arquivos de arquivamento (*.tar*) e também compactar ou descompactar esses arquivos com opções como *-z* para *gzip* e *-j* para *bzip2*.

```

Terminal - marco@marco: ~
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~$ tar -cf helloworld.tar helloworld helloworld.txt
marco@marco:~$ ls
'Área de Trabalho' Downloads helloworld.txt Música snap
code helloworld Imagens NovoDiretorio Vídeos
Documentos helloworld.tar Modelos Público
marco@marco:~$ ls -l
ls: não foi possível acessar '-': Arquivo ou diretório inexistente
ls: não foi possível acessar 'l': Arquivo ou diretório inexistente
marco@marco:~$ ls -lh
total 64K
drwxr-xr-x 3 marco marco 4,0K ago 13 15:39 'Área de Trabalho'
drwxrwxr-x 3 marco marco 4,0K jun 11 18:46 code
drwxr-xr-x 2 marco marco 4,0K jun 11 10:56 Documentos
drwxr-xr-x 2 marco marco 4,0K ago 14 14:47 Downloads
-rw-rw-r-- 1 marco marco 12 ago 14 15:51 helloworld
-rw-rw-r-- 1 marco marco 10K ago 14 16:06 helloworld.tar
-rw-rw-r-- 1 marco marco 12 ago 14 12:05 helloworld.txt
drwxr-xr-x 2 marco marco 4,0K jun 11 10:56 Imagens
drwxr-xr-x 2 marco marco 4,0K jun 11 10:56 Modelos
drwxr-xr-x 2 marco marco 4,0K jun 11 10:56 Música
drwxrwxr-x 2 marco marco 4,0K ago 13 21:48 NovoDiretorio
drwxr-xr-x 2 marco marco 4,0K jun 11 10:56 Público
drwx----- 4 marco marco 4,0K jun 11 15:23 snap
drwxr-xr-x 2 marco marco 4,0K jun 11 10:56 Vídeos
marco@marco:~$

```

❖ *nano*

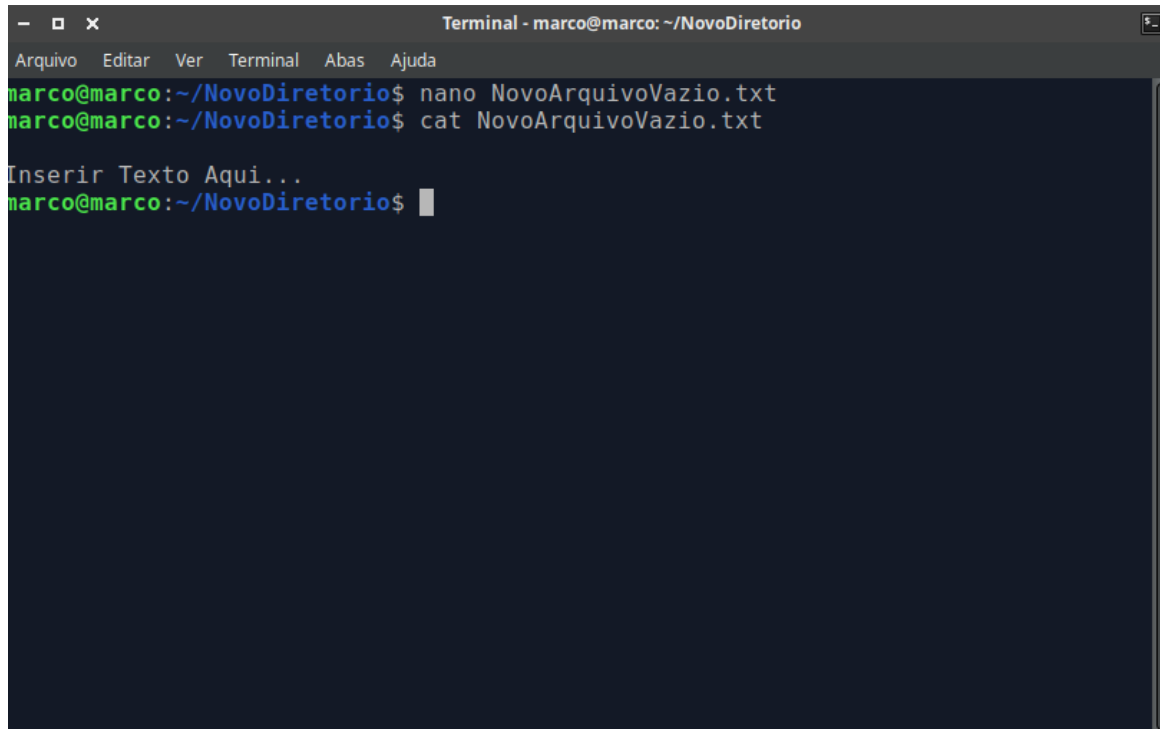
O comando *nano* abre um editor de texto simples no terminal Linux, permitindo criar e editar arquivos diretamente na linha de comando.

Comando utilizado para exemplo: *nano NovoArquivoVazio.txt*

```

Terminal - marco@marco: ~/NovoDiretorio
GNU nano 6.2 NovoArquivoVazio.txt *
Inserir Texto Aqui...
^G Ajuda ^O Gravar ^W Onde está? ^K Recortar ^T Executar ^C Local
^X Sair ^R Ler o arg ^\ Substituir ^U Colar ^J Justificar ^_ Ir p/ linha

```

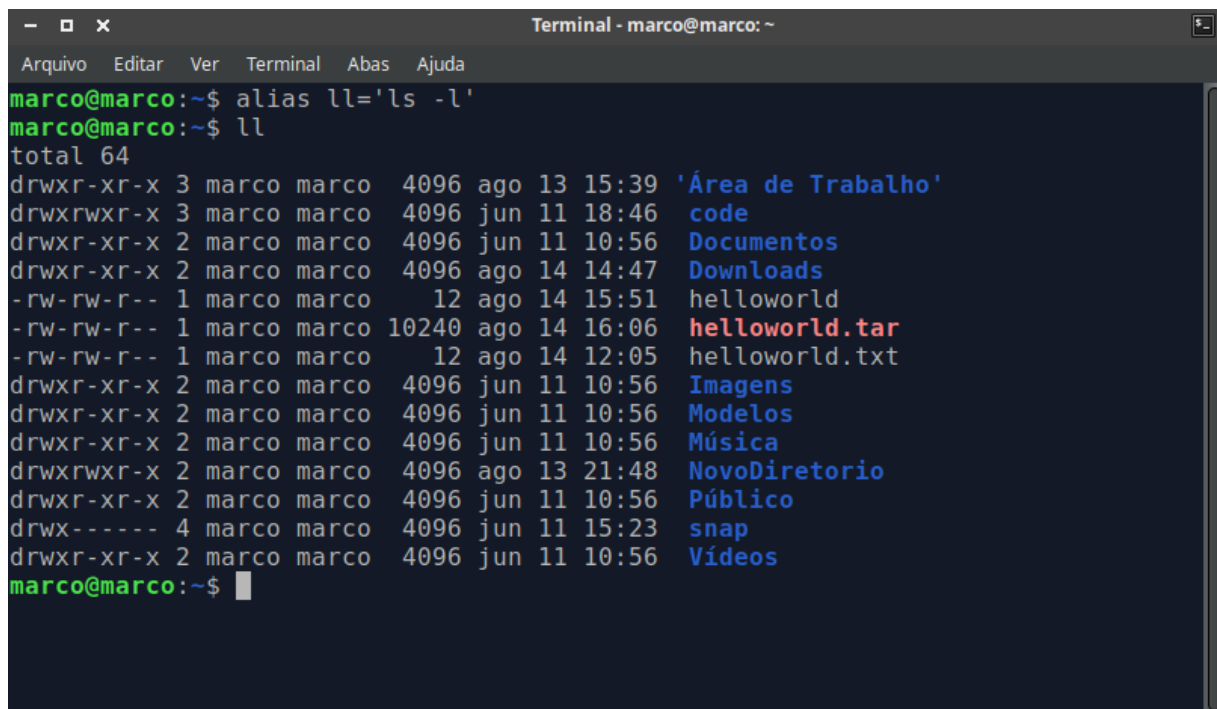



```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ nano NovoArquivoVazio.txt
marco@marco:~/NovoDiretorio$ cat NovoArquivoVazio.txt

Inserir Texto Aqui...
marco@marco:~/NovoDiretorio$
```

❖ *alias*

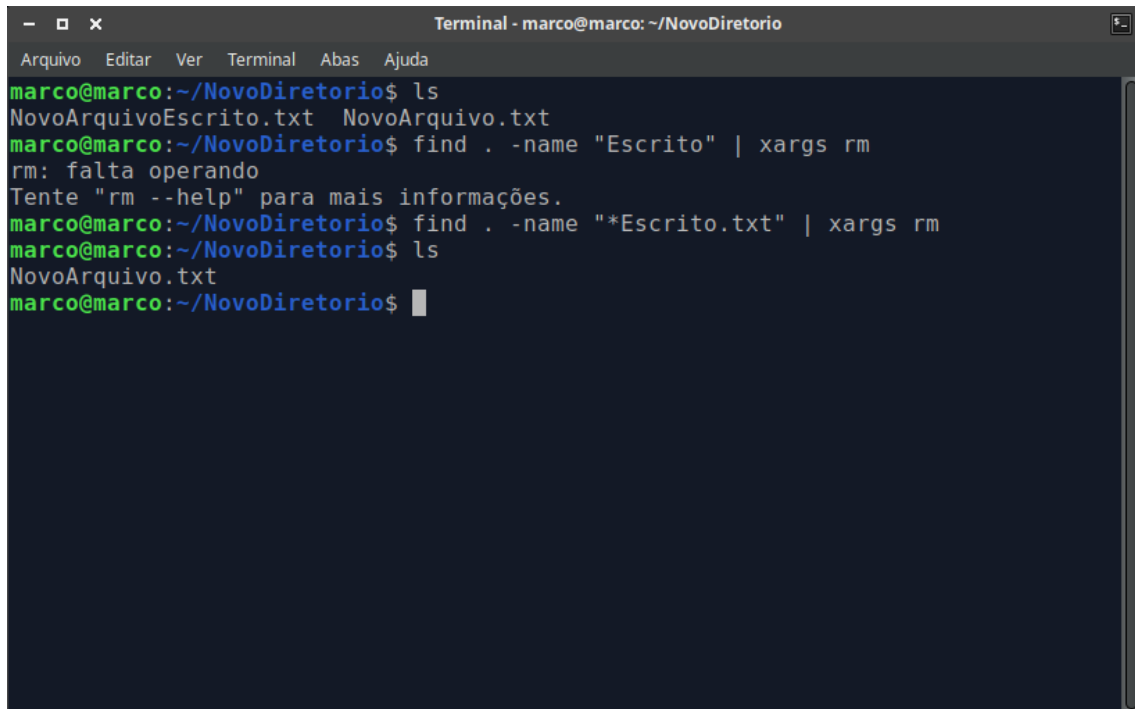
O comando *alias* cria atalhos personalizados para comandos no terminal, permitindo substituir comandos longos ou complexos por abreviações mais simples.



```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ alias ll='ls -l'
marco@marco:~$ ll
total 64
drwxr-xr-x 3 marco marco 4096 ago 13 15:39 'Área de Trabalho'
drwxrwxr-x 3 marco marco 4096 jun 11 18:46 code
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Documentos
drwxr-xr-x 2 marco marco 4096 ago 14 14:47 Downloads
-rw-rw-r-- 1 marco marco 12 ago 14 15:51 helloworld
-rw-rw-r-- 1 marco marco 10240 ago 14 16:06 helloworld.tar
-rw-rw-r-- 1 marco marco 12 ago 14 12:05 helloworld.txt
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Imagens
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Modelos
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Música
drwxrwxr-x 2 marco marco 4096 ago 13 21:48 NovoDiretorio
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Público
drwx----- 4 marco marco 4096 jun 11 15:23 snap
drwxr-xr-x 2 marco marco 4096 jun 11 10:56 Vídeos
marco@marco:~$
```

❖ *xargs*

O comando *xargs* constrói e executa comandos a partir da entrada padrão, como o resultado de outro comando. Ele é frequentemente usado para passar uma lista de argumentos a outro comando que não os aceita diretamente.



```
Terminal - marco@marco: ~/NovoDiretorio
marco@marco:~/NovoDiretorio$ ls
NovoArquivoEscrito.txt  NovoArquivo.txt
marco@marco:~/NovoDiretorio$ find . -name "Escrito" | xargs rm
rm: falta operando
Tente "rm --help" para mais informações.
marco@marco:~/NovoDiretorio$ find . -name "*Escrito.txt" | xargs rm
marco@marco:~/NovoDiretorio$ ls
NovoArquivo.txt
marco@marco:~/NovoDiretorio$
```

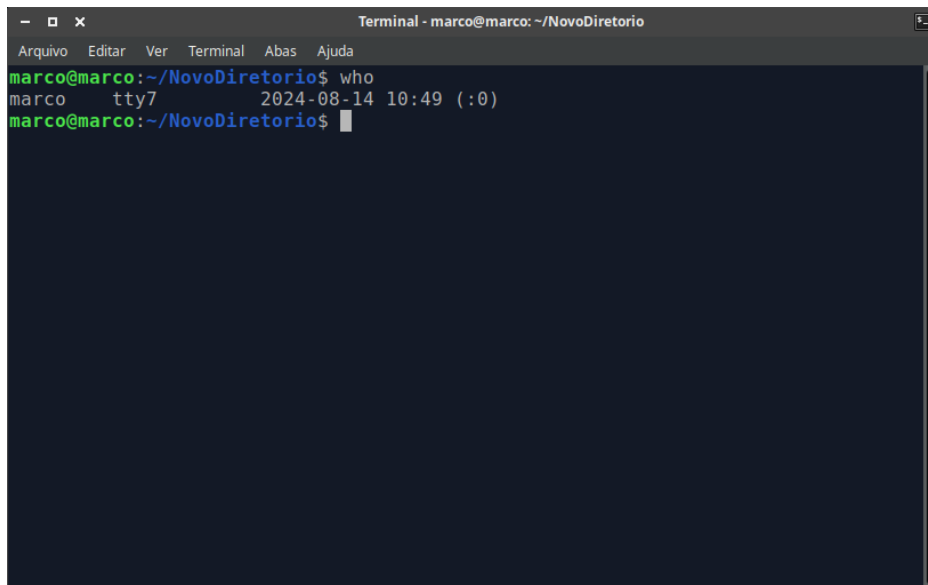
❖ *ln*

O comando *ln* cria links entre arquivos no sistema de arquivos. Existem dois tipos principais de links que podem ser criados:

- i. Link simbólico (soft link): Criado com *ln -s*, este é um tipo de ponteiro que aponta para o arquivo original. Se o arquivo original for removido, o link simbólico ficará quebrado.
- ii. Link físico (hard link): Criado sem a opção *-s*, este tipo de link é uma cópia do inode do arquivo original, apontando para o mesmo conteúdo. Mesmo que o arquivo original seja deletado, o conteúdo ainda pode ser acessado através do hard link.

❖ *who*

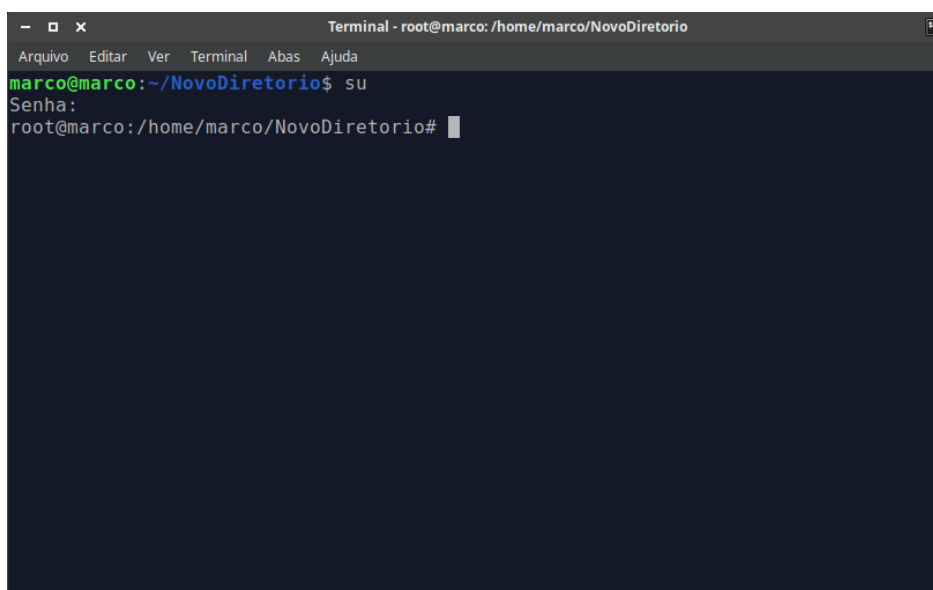
O comando *who* exibe informações sobre os usuários atualmente logados no sistema, incluindo seus nomes de usuário, terminais, horários de login e, às vezes, o IP ou hostname de onde estão conectados.



```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ who
marco    tty7        2024-08-14 10:49 (:0)
marco@marco:~/NovoDiretorio$
```

❖ *su*

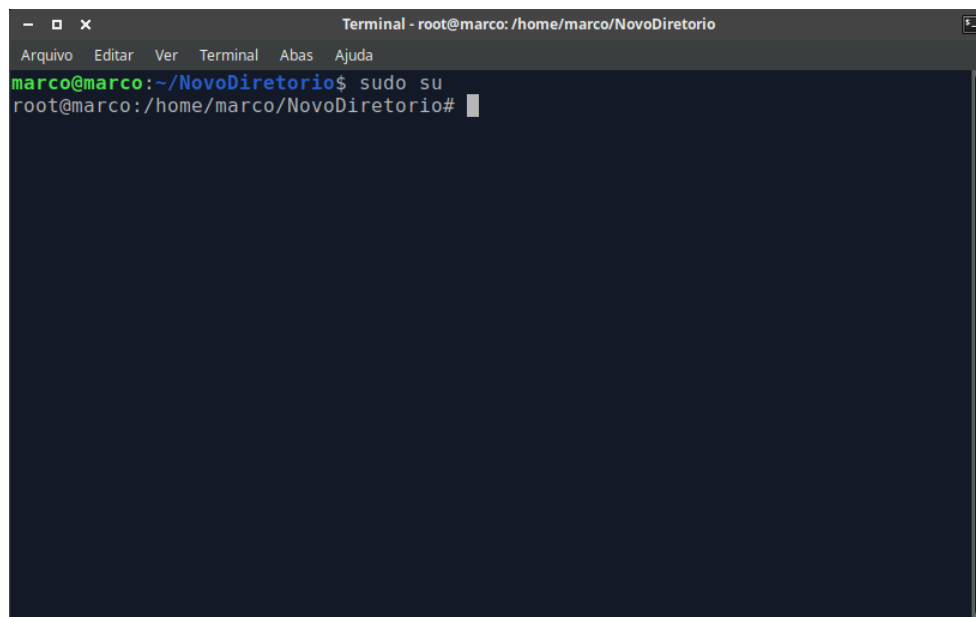
O comando *su* permite mudar para outro usuário no sistema, geralmente para obter privilégios administrativos. Quando usado como *su* sozinho, ele troca para o usuário root.



```
Terminal - root@marco: /home/marco/NovoDiretorio
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
marco@marco:~/NovoDiretorio$ su
Senha:
root@marco:/home/marco/NovoDiretorio#
```

❖ *sudo*

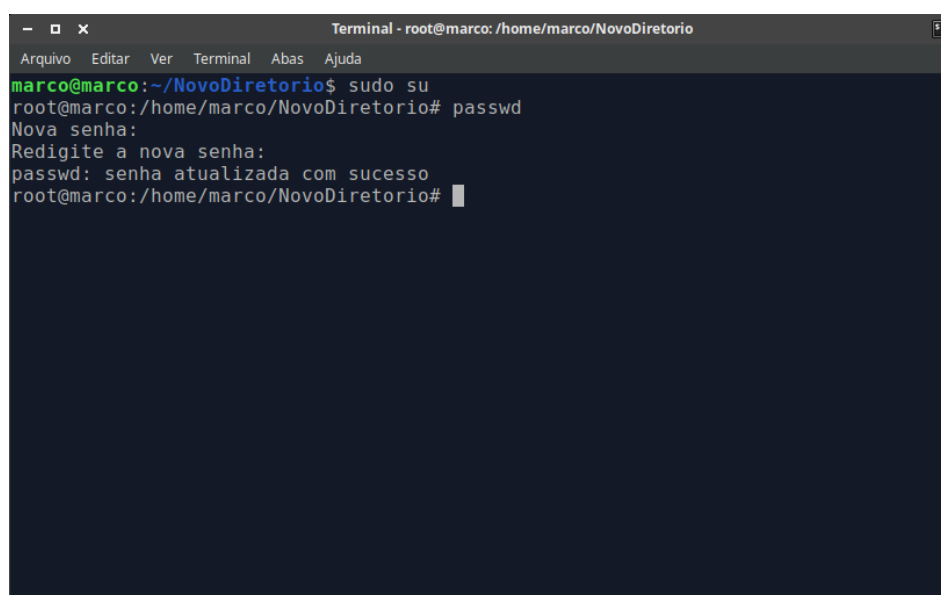
O comando *sudo* permite que um usuário execute comandos com privilégios de superusuário (root) temporariamente. Ele é usado para tarefas administrativas que exigem permissões elevadas, como instalar software ou modificar arquivos de sistema.



```
Terminal - root@marco: /home/marco/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ sudo su
root@marco:/home/marco/NovoDiretorio#
```

❖ *passwd*

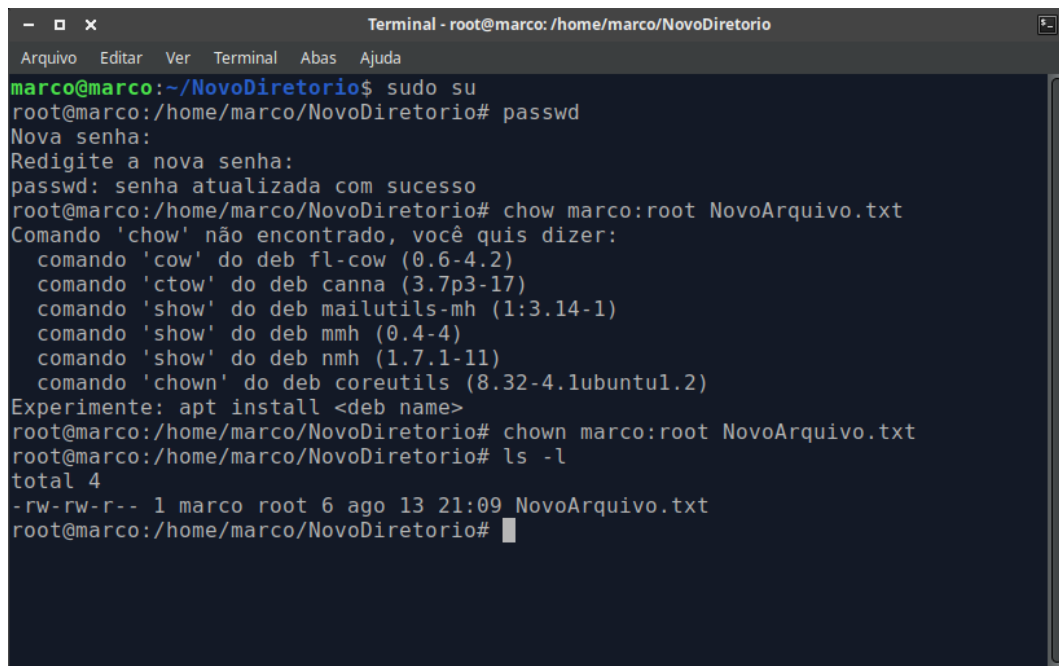
O comando *passwd* altera a senha de um usuário no sistema.



```
Terminal - root@marco: /home/marco/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ sudo su
root@marco:/home/marco/NovoDiretorio# passwd
Nova senha:
Redigite a nova senha:
passwd: senha atualizada com sucesso
root@marco:/home/marco/NovoDiretorio#
```

❖ *chown*

O comando *chown* altera o proprietário e/ou o grupo de um arquivo ou diretório no sistema.



```

Terminal - root@marco: /home/marco/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ sudo su
root@marco:/home/marco/NovoDiretorio# passwd
Nova senha:
Redigite a nova senha:
passwd: senha atualizada com sucesso
root@marco:/home/marco/NovoDiretorio# chow marco:root NovoArquivo.txt
Comando 'chow' não encontrado, você quis dizer:
  comando 'cow' do deb fl-cow (0.6-4.2)
  comando 'ctow' do deb canna (3.7p3-17)
  comando 'show' do deb mailutils-mh (1:3.14-1)
  comando 'show' do deb mmh (0.4-4)
  comando 'show' do deb nmh (1.7.1-11)
  comando 'chown' do deb coreutils (8.32-4.1ubuntu1.2)
Experimente: apt install <deb name>
root@marco:/home/marco/NovoDiretorio# chown marco:root NovoArquivo.txt
root@marco:/home/marco/NovoDiretorio# ls -l
total 4
-rw-rw-r-- 1 marco root 6 ago 13 21:09 NovoArquivo.txt
root@marco:/home/marco/NovoDiretorio#

```

❖ *Understanding permissions*

As permissões no Linux controlam quem pode ler, escrever ou executar arquivos e diretórios, e são divididas em três categorias: dono, grupo e outros. Cada categoria possui três tipos de permissões: leitura (r), escrita (w) e execução (x). Essas permissões são representadas em uma string de 10 caracteres, como -rwxr-xr--, onde os primeiros três caracteres após o tipo do arquivo representam as permissões do dono, os próximos três as do grupo, e os últimos três as dos outros usuários. Para alterar essas permissões, usa-se o comando *chmod*, e para modificar o dono ou grupo de um arquivo, utiliza-se *chown*.

❖ *chmod*

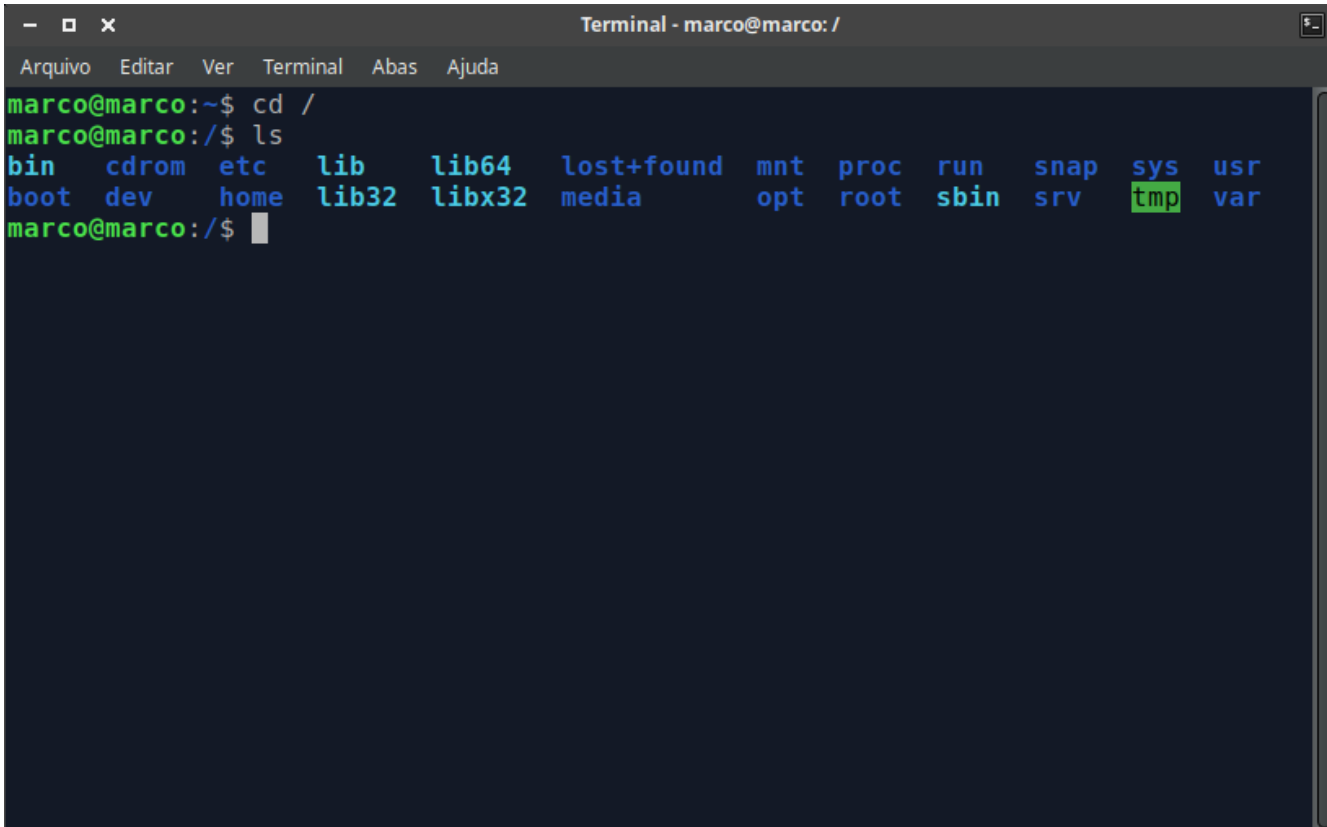
O comando *chmod* altera as permissões de arquivos e diretórios no Linux, usando modos simbólico ou numérico. No modo simbólico, você ajusta permissões com letras (u, g, o, a) e símbolos (+, -, =). No modo numérico, usa-se uma combinação de valores (4 para leitura, 2 para escrita, 1 para execução) para definir permissões para dono, grupo e outros.

4. SISTEMAS DE ARQUIVOS NO LINUX

O sistema de arquivos no Linux organiza e gerencia a forma como os dados são armazenados e acessados no disco rígido. Ele fornece a estrutura necessária para armazenar e recuperar arquivos e diretórios, garantindo que os dados sejam armazenados de maneira eficiente e segura.

No Windows, o disco C: é o principal local onde o sistema operacional e os aplicativos são instalados, e a estrutura é menos visível para o usuário, que interage principalmente por meio de interfaces gráficas. Em contraste, no Linux, o sistema de arquivos é organizado hierarquicamente com diretórios específicos para funções distintas, e a gestão é mais acessível através do terminal. Enquanto o Windows utiliza letras de unidade (C:, D:), o Linux utiliza uma estrutura única de diretórios a partir do diretório raiz (/), que proporciona uma visão mais coesa e integrada dos sistemas de arquivos.

Como dito anteriormente o Linux utiliza uma estrutura hierárquica de diretórios para organizar arquivos. O diretório raiz (/) é o ponto de partida, com diversos subdiretórios padrão, cada um com um propósito específico:

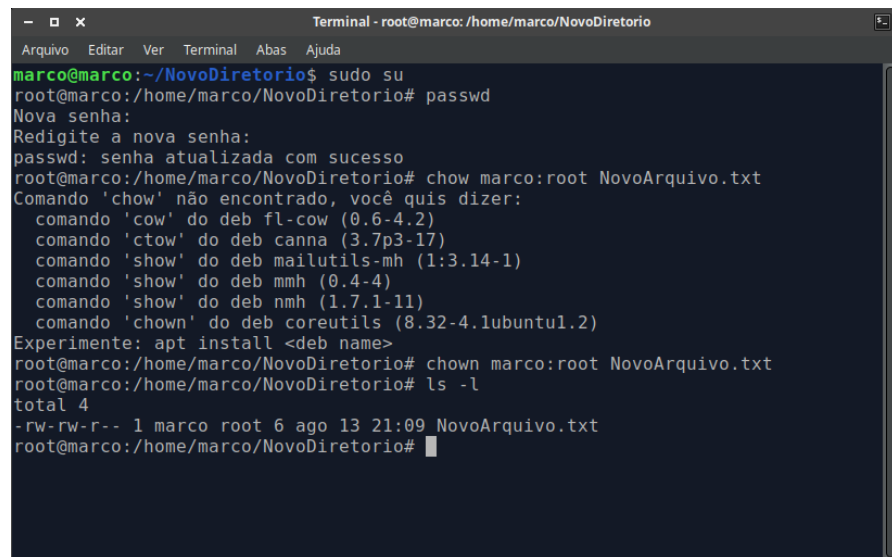


```
Terminal - marco@marco: /  
Arquivo Editar Ver Terminal Abas Ajuda  
marco@marco:~$ cd /  
marco@marco:/$ ls  
bin    cdrom  etc    lib    lib64  lost+found  mnt  proc  run  snap  sys  usr  
boot  dev    home  lib32  libx32  media      opt  root  sbin  srv   tmp  var  
marco@marco:/$
```

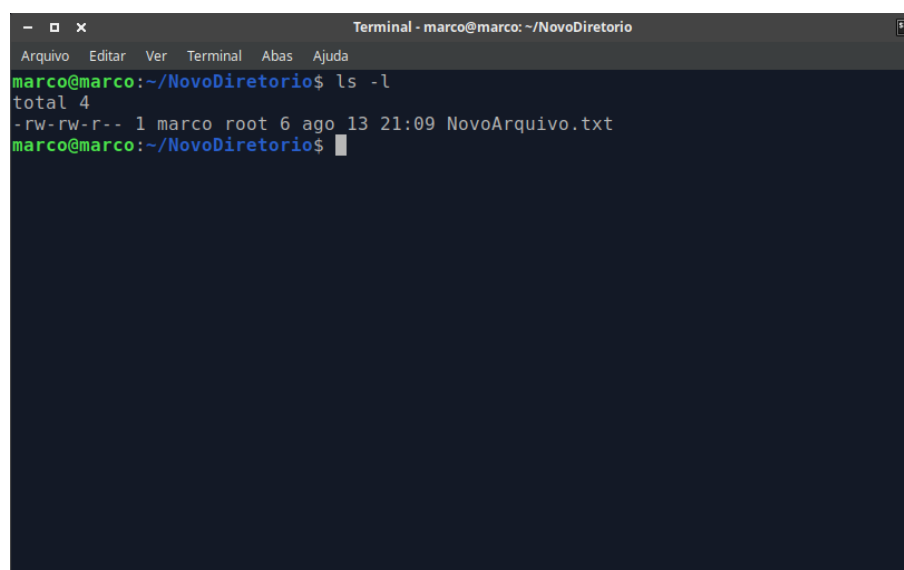
- **/:** Diretório raiz, ponto de partida da hierarquia de diretórios.
- **/bin:** Contém comandos binários essenciais para todos os usuários, como ls, cp e mv.
- **/boot:** Armazena arquivos necessários para o processo de inicialização do sistema, incluindo o kernel.
- **/dev:** Contém arquivos de dispositivos que representam hardware e dispositivos de sistema.
- **/etc:** Armazena arquivos de configuração do sistema e dos aplicativos.
- **/home:** Contém diretórios pessoais dos usuários, onde ficam seus arquivos e configurações.
- **/lib:** Contém bibliotecas essenciais para os binários localizados em /bin e /sbin.
- **/media:** Ponto de montagem para mídias removíveis, como CDs e pendrives.
- **/mnt:** Ponto de montagem para sistemas de arquivos temporários, geralmente usado para montagens manuais.
- **/opt:** Contém aplicativos adicionais e pacotes de software de terceiros.
- **/proc:** Sistema de arquivos virtual que fornece informações sobre o estado do kernel e processos em execução.
- **/root:** Diretório home do usuário root (administrador do sistema).
- **/run:** Contém dados temporários e de execução, como arquivos de PID e informações de sessão.
- **/sbin:** Contém comandos binários essenciais para a administração do sistema, como fsck e reboot.
- **/srv:** Contém dados para serviços fornecidos pelo sistema, como servidores web ou FTP.
- **/sys:** Sistema de arquivos virtual que expõe informações sobre o sistema e dispositivos.
- **/tmp:** Diretório para arquivos temporários que podem ser limpos após reinicializações.
- **/usr:** Contém dados de aplicativos e programas, incluindo bibliotecas, arquivos de documentação e binários não essenciais para o sistema.
- **/var:** Utilizado para armazenar dados variáveis, como logs, cache e arquivos temporários de aplicativos.

O Linux usa um sistema de permissões para controlar o acesso a arquivos e diretórios. As permissões podem ser de leitura (r), escrita (w) e execução (x). Comandos como `ls` exibem permissões, enquanto `chmod` e `chown` são usados para alterar permissões e propriedades dos arquivos.

Por exemplo sendo executado, temos o comando `chown` que altera o proprietário e/ou o grupo de um arquivo ou diretório no sistema. Observa-se também ao executar o comando `ls -l` as permissões que são representadas em uma string de 10 caracteres, como `-rwxr-xr--`, onde os primeiros três caracteres após o tipo do arquivo representam as permissões do dono, os próximos três as do grupo, e os últimos três as dos outros usuários.



```
Terminal - root@marco: /home/marco/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ sudo su
root@marco:/home/marco/NovoDiretorio# passwd
Nova senha:
Redigite a nova senha:
passwd: senha atualizada com sucesso
root@marco:/home/marco/NovoDiretorio# chow marco:root NovoArquivo.txt
Comando 'chow' não encontrado, você quis dizer:
  comando 'cow' do deb fl-cow (0.6-4.2)
  comando 'ctow' do deb canna (3.7p3-17)
  comando 'show' do deb mailutils-mh (1:3.14-1)
  comando 'show' do deb mmh (0.4-4)
  comando 'show' do deb nmh (1.7.1-11)
  comando 'chown' do deb coreutils (8.32-4.1ubuntu1.2)
Experimente: apt install <deb name>
root@marco:/home/marco/NovoDiretorio# chown marco:root NovoArquivo.txt
root@marco:/home/marco/NovoDiretorio# ls -l
total 4
-rw-rw-r-- 1 marco root 6 ago 13 21:09 NovoArquivo.txt
root@marco:/home/marco/NovoDiretorio#
```



```
Terminal - marco@marco: ~/NovoDiretorio
Arquivo Editar Ver Terminal Abas Ajuda
marco@marco:~/NovoDiretorio$ ls -l
total 4
-rw-rw-r-- 1 marco root 6 ago 13 21:09 NovoArquivo.txt
marco@marco:~/NovoDiretorio$
```


Para monitorar e gerenciar o espaço em disco, o Linux oferece comandos como `df` e `du`. O comando `df` exibe o espaço total, usado e disponível em sistemas de arquivos, enquanto `du` mostra o uso de espaço de diretórios e arquivos específicos, ajudando a identificar quais itens estão ocupando mais espaço.

```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ ls
'Área de Trabalho'  Downloads  Imagens  names.txt  snap
code                helloworld2.txt  Modelos  NovoDiretorio  Vídeos
Documentos          helloworld.txt  Música  Público
marco@marco:~$ du NovoDiretorio/
12      NovoDiretorio/
marco@marco:~$ du names.txt
4       names.txt
marco@marco:~$ df NovoDiretorio/
Sist. Arq.      Blocos de 1K  Usado  Disponível  Uso%  Montado em
/dev/nvme0n1p10  49777208  3460796  43755388   8%  /home
marco@marco:~$
```

```
Terminal - marco@marco: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
marco@marco:~$ ls
'Área de Trabalho'  Downloads  Imagens  names.txt  snap
code                helloworld2.txt  Modelos  NovoDiretorio  Vídeos
Documentos          helloworld.txt  Música  Público
marco@marco:~$ du NovoDiretorio/
12      NovoDiretorio/
marco@marco:~$ du names.txt
4       names.txt
marco@marco:~$
```