



Sistemas embarcados II

Trabalho 1

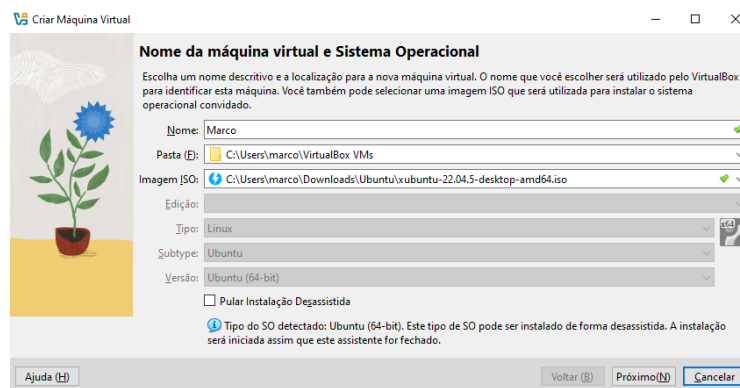
Professor: Eder Alves de Moura

Alunos: Marco Túlio França

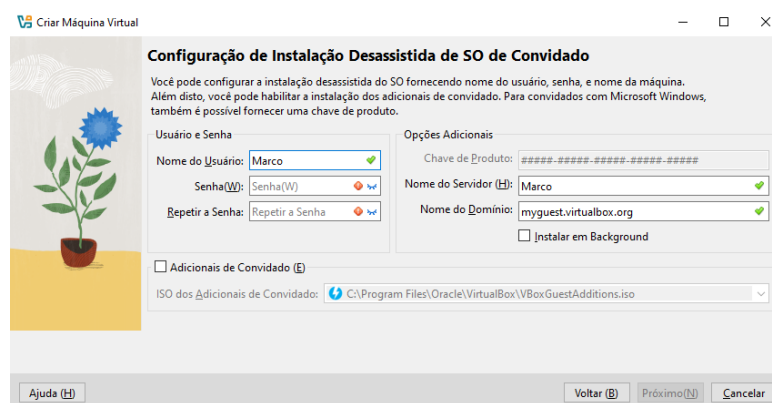
Matrícula: 11911ETE002

07/02/2025


Primeiramente foi iniciado a criação de uma maquina virtual a partir do Ubuntu 22.04



foi iniciado a configuração utilizando nome do usuário “marco”



Os dados da maquina virtual foram colocados de acordo com o que foi solicitado no enunciado.



Hardware

Você pode alterar o hardware da máquina virtual definindo a quantidade de RAM e CPUs virtuais. Também é possível habilitar o EFI.

Memória Base: 4 MB 8192 MB

Processadores: 1 CPU 4 CPUs

☐ Habilitar EFI (SOs especiais apenas)

Disco Rígido Virtual

Você pode acrescentar um disco rígido virtual à nova máquina virtual. Você pode criar um novo arquivo de disco rígido virtual ou selecionar um existente. Você também pode criar uma máquina virtual sem um disco rígido.

☒ Criar um novo disco rígido virtual agora

Tamanho do Disco: 4,00 MB 2,00 TB

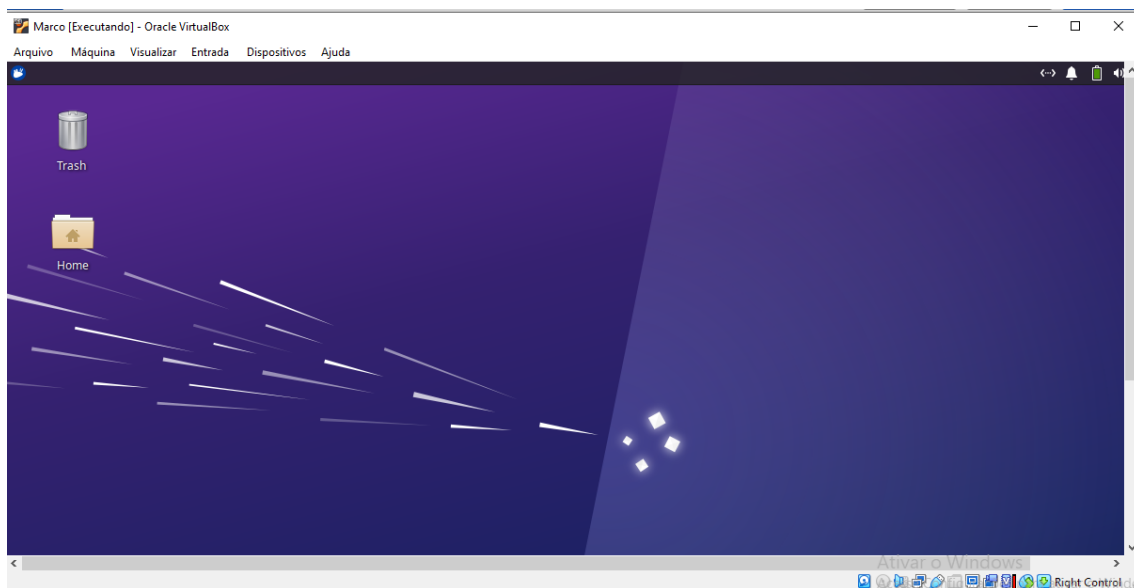
☐ Pré-alocar Tamanho Total (F)

☐ Utilizar um disco rígido virtual existente

Vazio

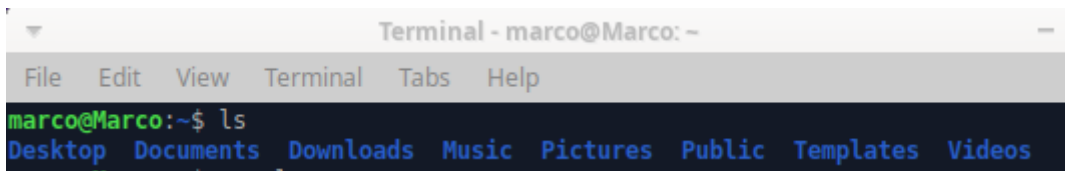
☐ Não Acrescentar um Disco Rígido Virtual

Após esse processo, foi confirmada a instalação e já se tinha o ambiente de trabalho disponível, como é visto abaixo:



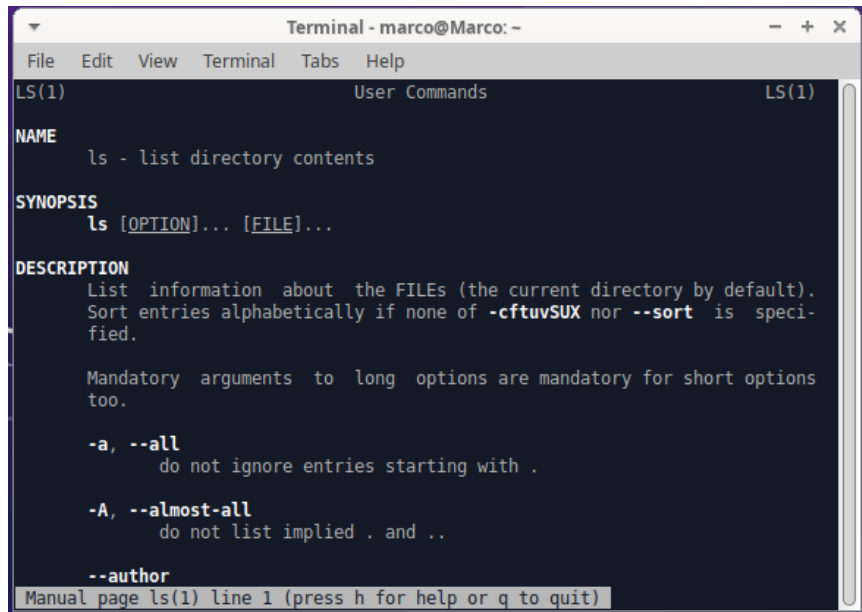
Nesta parte agora iremos abordar os comandos listados do primeiro vídeo:

ls: Lista o conteúdo do diretório atual. Exemplo: ls exibe os arquivos e pastas do diretório em uso.



```
Terminal - marco@Marco: ~
File Edit View Terminal Tabs Help
marco@Marco:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

man: Mostra o manual de uso de comandos do sistema. Exemplo: `man ls` exibe informações detalhadas sobre o comando `ls`.



```
Terminal - marco@Marco: ~
File Edit View Terminal Tabs Help
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.

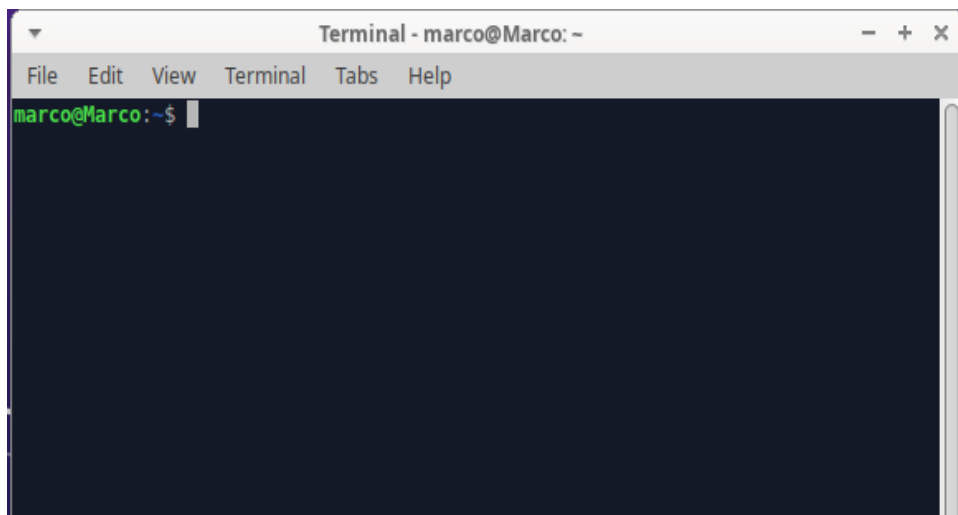
Mandatory arguments to long options are mandatory for short options
too.

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

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

--author
Manual page ls(1) line 1 (press h for help or q to quit)
```

clear: Limpa a tela do terminal para facilitar a visualização.



```
Terminal - marco@Marco: ~
File Edit View Terminal Tabs Help
marco@Marco:~$
```

mkdir: Cria um novo diretório. Exemplo: `mkdir exemplo` cria uma pasta chamada "exemplo".

cd: Navega entre diretórios. Exemplo: `cd exemplo` entra na pasta "exemplo".

pwd: Exibe o caminho completo do diretório atual.

whoami: Mostra o nome do usuário atualmente logado.

Redirecionadores: Usados para redirecionar a entrada/saída. Exemplo: `whoami >> exemplo.txt` salva o arquivo "exemplo.txt".

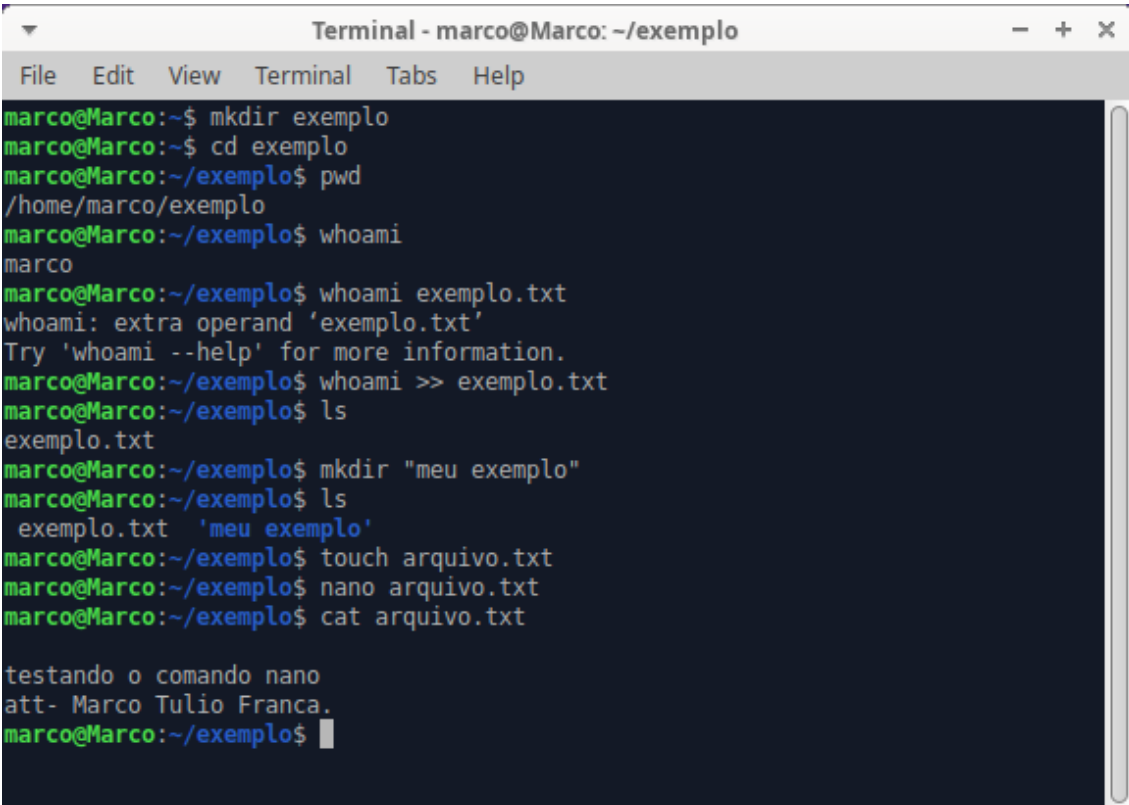
Criar e acessar pasta com nome composto: Utilize aspas. Exemplo: `mkdir "meu exemplo"` e `cd "meu exemplo"`.

touch: Cria arquivos vazios. Exemplo: `touch arquivo.txt` cria um arquivo chamado "arquivo.txt".

nano: Editor de texto no terminal. Exemplo: `nano arquivo.txt` permite editar o arquivo "arquivo.txt".

cat: Exibe o conteúdo de arquivos. Exemplo: `cat arquivo.txt` mostra o conteúdo de "arquivo.txt".

A lista de comandos anteriores está exemplificada no print abaixo:



```
Terminal - marco@Marco: ~/exemplo
File Edit View Terminal Tabs Help
marco@Marco:~$ mkdir exemplo
marco@Marco:~$ cd exemplo
marco@Marco:~/exemplo$ pwd
/home/marco/exemplo
marco@Marco:~/exemplo$ whoami
marco
marco@Marco:~/exemplo$ whoami exemplo.txt
whoami: extra operand 'exemplo.txt'
Try 'whoami --help' for more information.
marco@Marco:~/exemplo$ whoami >> exemplo.txt
marco@Marco:~/exemplo$ ls
exemplo.txt
marco@Marco:~/exemplo$ mkdir "meu exemplo"
marco@Marco:~/exemplo$ ls
exemplo.txt 'meu exemplo'
marco@Marco:~/exemplo$ touch arquivo.txt
marco@Marco:~/exemplo$ nano arquivo.txt
marco@Marco:~/exemplo$ cat arquivo.txt

testando o comando nano
att- Marco Tulio Franca.
marco@Marco:~/exemplo$
```

mv: Move ou renomeia arquivos e pastas. Exemplo: `mv arquivo.txt arquivo2.txt` renomeia o arquivo para "arquivo2.txt".

cp: Copia arquivos e pastas. Exemplo: `cp arquivo.txt copia.txt` cria uma cópia chamada "copia.txt".

find: Busca arquivos ou pastas. Exemplo: `find . -name "arquivo2.txt"` busca o arquivo "arquivo2.txt".

head & tail: Mostra o início (head) ou fim (tail) de um arquivo. Exemplo: `head arquivo.txt` exibe as primeiras linhas de "arquivo.txt", e `tail arquivo.txt` exibe as últimas.

```
marco@Marco:~/exemplo$ mv arquivo.txt arquivo2.txt
marco@Marco:~/exemplo$ cp arquivo2.txt copia.txt
marco@Marco:~/exemplo$ ls
arquivo2.txt  copia.txt  exemplo.txt  'meu exemplo'
marco@Marco:~/exemplo$ find. -name "arquivo2.txt"
Command 'find.' not found, did you mean:
  command 'findg' from deb ncl-ncarg (6.6.2-10build2)
  command 'find' from deb findutils (4.8.0-1ubuntu3)
  command 'findv' from deb polylib-utils (5.22.5-4+dfsg)
Try: apt install <deb name>
marco@Marco:~/exemplo$ ^C
marco@Marco:~/exemplo$ find . -name arquivo2.txt
./arquivo2.txt
marco@Marco:~/exemplo$ head arquivo2.txt

testando o comando nano
att- Marco Tulio Franca.
marco@Marco:~/exemplo$ tail arquivo2.txt

testando o comando nano
att- Marco Tulio Franca.
```

less: Exibe o conteúdo de arquivos com navegação por página. No exemplo foi usando less arquivo2.txt.

```
testando o comando nano
att- Marco Tulio Franca.
(END)
```

rm: Remove arquivos. Exemplo: rm arquivo.txt exclui "arquivo.txt".

rmdir: Remove diretórios vazios.

rm -rf: Remove diretórios com conteúdo, de forma forçada.

hostname: Exibe o nome do host do sistema.

hostname -i: Mostra o endereço IP associado ao hostname.

ip a: Exibe informações de interfaces de rede.

```
marco@Marco:~/exemplo$ less arquivo2.txt
marco@Marco:~/exemplo$ ls
arquivo2.txt  copia.txt  exemplo.txt  'meu exemplo'
marco@Marco:~/exemplo$ rm copia.txt
marco@Marco:~/exemplo$ rmdir meu\ exemplo/
marco@Marco:~/exemplo$ hostname
Marco
marco@Marco:~/exemplo$ hostname i
hostname: you must be root to change the host name
marco@Marco:~/exemplo$ hostname -i
127.0.1.1
marco@Marco:~/exemplo$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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
    link/ether 08:00:27:f3:c1:da 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 76580sec preferred_lft 76580sec
```

grep: Filtra conteúdo de textos. Exemplo: grep "palavra" arquivo.txt busca "palavra" no arquivo.

ping: Testa a conectividade com outro dispositivo.

```
marco@Marco:~/exemplo$ 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 fd00::c4df:49f0:b10e:6efb/64 scope global temporary dynamic
inet6 fd00::d8c9:bf7c:3d8e:90d3/64 scope global dynamic mngtmpaddr noprefixroute
inet6 fe80::ed95:9b5a:4b56:223b/64 scope link noprefixroute
marco@Marco:~/exemplo$ ping google.com.br
PING google.com.br (142.251.132.227) 56(84) bytes of data.
64 bytes from gru14s46-in-f3.1e100.net (142.251.132.227): icmp_seq=1 ttl=255 time=15.1 ms
64 bytes from gru14s46-in-f3.1e100.net (142.251.132.227): icmp_seq=2 ttl=255 time=16.1 ms
64 bytes from gru14s46-in-f3.1e100.net (142.251.132.227): icmp_seq=3 ttl=255 time=14.9 ms
64 bytes from gru14s46-in-f3.1e100.net (142.251.132.227): icmp_seq=4 ttl=255 time=15.5 ms
```

free -h & free -m: Mostra o uso de memória em formatos legíveis.

```
marco@Marco:~/exemplo$ free -h
              total        used        free      shared  buff/cache   available
Mem:           2,5Gi         489Mi         695Mi          1,0Mi          1,4Gi          1,9Gi
Swap:          5,2Gi           0,0Ki          5,2Gi
marco@Marco:~/exemplo$ free -m
              total        used        free      shared  buff/cache   available
Mem:           2595         489         695           1         1409         1930
Swap:          5371           0         5371
```

top: Monitora os processos do sistema em tempo real.

```
top - 09:33:37 up 3:34, 1 user, load average: 0,01, 0,04, 0,05
Tasks: 160 total, 1 running, 159 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,7 us, 1,6 sy, 0,0 ni, 91,7 id, 2,5 wa, 0,0 hi, 1,6 si, 0,0 st
MiB Mem : 2595,3 total, 694,9 free, 490,6 used, 1409,8 buff/cache
MiB Swap: 5372,0 total, 5371,7 free, 0,3 used. 1929,9 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
  661 root        20   0 331680 96992 52232 S   4,9   3,6   0:35.00 Xorg
 45709 marco      20   0 493356 48484 37888 S   3,0   1,8   0:13.21 xfce4-terminal
  1103 marco      20   0 471844 46072 29236 S   1,0   1,7   0:06.97 xfdesktop
 45878 marco      20   0 23804   4096  3456 R   1,0   0,2   0:00.25 top
  1034 marco      20   0 926612 99868 77552 S   0,7   3,8   0:12.69 xfwm4
   781 marco      20   0  9024   5120  4096 S   0,3   0,2   0:12.33 dbus-daemon
  1065 marco      20   0 666380 43448 34252 S   0,3   1,6   0:38.73 panel-9-pulseau
```

htop: Similar ao top, mas com interface visual melhorada.(Não foi possível exemplificar)

```
marco@Marco:~/exemplo$ sudo apt install htop
[sudo] password for marco:
marco is not in the sudoers file. This incident will be reported.
```

ps: Lista os processos em execução.

ps aux: Mostra detalhes de todos os processos.

```
marco@Marco:~/exemplo$ ps
  PID TTY          TIME CMD
  45728 pts/0    00:00:00 bash
  45892 pts/0    00:00:00 ps
marco@Marco:~/exemplo$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.2   0.4 167616 13020 ?        Ss   05:58   0:36 /sbin/init splash
root         2   0.0   0.0      0     0 ?        S    05:58   0:00 [kthreadd]
root         3   0.0   0.0      0     0 ?        S    05:58   0:00 [pool_workqueue_release]
root         4   0.0   0.0      0     0 ?        I<   05:58   0:00 [kworker/R-rcu_g]
root         5   0.0   0.0      0     0 ?        I<   05:58   0:00 [kworker/R-rcu_pl]
root         6   0.0   0.0      0     0 ?        I<   05:58   0:00 [kworker/R-slub_]
root         7   0.0   0.0      0     0 ?        I<   05:58   0:00 [kworker/R-netns]
```

kill: Encerra processos pelo ID. Exemplo: kill 1234 encerra o processo com ID 1234.

```
root      45808  0.0  0.0      0     0 ?        I   09:12   0:00 [kworker/u5:3-events_power_effici
root      45824  0.0  0.0      0     0 ?        I   09:19   0:00 [kworker/u5:1-events_unbound]
root      45841  0.0  0.0      0     0 ?        I   09:24   0:00 [kworker/u6:2-events_unbound]
root      45868  0.0  0.0      0     0 ?        I   09:32   0:00 [kworker/u6:0-events_power_effici
root      45891  0.0  0.0      0     0 ?        I   09:37   0:00 [kworker/u6:3-flush-8:0]
marco     45912  0.0  0.2 173632 6656 ?        Ssl  09:40   0:00 /usr/libexec/gvfsd-metadata
marco     45924  1.2  0.9 419672 26480 ?        SNl  09:40   0:00 /usr/lib/x86_64-linux-gnu/tumbler
root      45937  0.0  0.0      0     0 ?        I   09:40   0:00 [kworker/u5:0-events_power_effici
marco     45940 14.6  1.7 552724 45700 ?        Sl   09:41   0:02 gnome-sudoku
marco     45958  0.0  0.1 23396  3584 pts/0    R+   09:41   0:00 ps aux
marco@Marco:~/exemplo$ ^C
marco@Marco:~/exemplo$ kill 45940
```

df -h: Exibe o uso de disco de forma legível.

ncdu: Analisa e exibe o uso de espaço em disco.

uname: Exibe informações do sistema.

lscpu & lsusb: Exibe detalhes sobre a CPU (lscpu) e dispositivos USB (lsusb).

```
marco@Marco:~/exemplo$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            260M  1,3M  259M   1% /run
/dev/sda3        98G   15G   78G  17% /
tmpfs            1,3G     0  1,3G   0% /dev/shm
tmpfs            5,0M  4,0K  5,0M   1% /run/lock
/dev/sda2        512M   6,1M  506M   2% /boot/efi
tmpfs            260M  100K  260M   1% /run/user/1000
marco@Marco:~/exemplo$ ncdu
Command 'ncdu' not found, but can be installed with:
apt install ncdu
Please ask your administrator.
marco@Marco:~/exemplo$ apt install ncdu
E: Could not open lock file /var/lib/dpkg/lock-frontent - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), are you root?
marco@Marco:~/exemplo$ uname
Linux
marco@Marco:~/exemplo$ lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Address sizes:        36 bits physical, 48 bits virtual
Byte Order:           Little Endian
CPU(s):               2
On-line CPU(s) list: 0,1
```


history: Mostra o histórico de comandos executados.

```
marco@Marco:~/exemplo$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 003: ID 80ee:0021 VirtualBox USB Tablet
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
marco@Marco:~/exemplo$ history
 1  ls
 2  man ls
 3  clear
 4  mkdir exemplo
 5  cd exemplo
 6  pwd
 7  whoami
 8  whoami exemplo.txt
 9  whoami >> exemplo.txt
10  ls
11  mkdir "meu exemplo"
12  ls
13  touch arquivo.txt
14  nano arquivo.txt
15  cat arquivo.txt
16  mv arquivo.txt arquivo2.txt
17  cp arquivo2.txt copia.txt
18  ls
```

Agora iremos abordar os comandos listados no segundo vídeo:

echo: Exibe uma mensagem ou valor no terminal. Exemplo: `echo "Olá, mundo!"` imprime "Olá, mundo!".

wc: Conta linhas, palavras e caracteres em arquivos. Exemplo: `wc arquivo.txt` exibe essas contagens para "arquivo.txt".

piping: Conecta a saída de um comando à entrada de outro. Exemplo: `ls | wc -l` conta o número de arquivos listados.

sort: Ordena linhas de texto em arquivos. Exemplo: `sort arquivo.txt` ordena as linhas de "arquivo.txt".

uniq: Filtra linhas duplicadas. Exemplo: `uniq arquivo.txt` remove duplicatas consecutivas em "arquivo.txt".

expansions: Expande variáveis, comandos ou caracteres especiais. Exemplo: `echo $(date)` exibe a data atual.

```
marco@Marco:~/exemplo$ echo "hello world"
hello world
marco@Marco:~/exemplo$ wc arquivo2.txt
 3  8 50 arquivo2.txt
marco@Marco:~/exemplo$ ls | wc -l
wc: invalid option -- 'l'
Try 'wc --help' for more information.
marco@Marco:~/exemplo$ ls -l | wc
   3   20  115
marco@Marco:~/exemplo$ sort arquivo2.txt

att- Marco Tulio Franca.
testando o comando nano
marco@Marco:~/exemplo$ uniq arquivo2.txt

testando o comando nano
att- Marco Tulio Franca.
marco@Marco:~/exemplo$ echo $(date)
bash: syntax error near unexpected token `('
marco@Marco:~/exemplo$ echo $(date)
sáb 25 jan 2025 09:59:56 -03
```

diff: Compara diferenças entre arquivos. Exemplo: `diff arquivo2.txt`

`exemplo.txt` mostra as diferenças entre os dois arquivos.

du: Mostra o uso de disco por diretórios e arquivos. Exemplo: `du -h` exibe o uso de espaço de forma legível.

df: Exibe o uso do sistema de arquivos. Exemplo: `df -h` mostra o espaço disponível e usado.

killall: Encerra todos os processos com o mesmo nome. Exemplo: `killall firefox` encerra todos os processos "firefox".

```
marco@Marco:~/exemplo$ ls
arquivo2.txt  exemplo.txt
marco@Marco:~/exemplo$ diff arquivo2.txt exemplo.txt
1,3c1
<
< testando o comando nano
< att- Marco Tulio Franca.
---
> marco
marco@Marco:~/exemplo$ du -h
12K  .
marco@Marco:~/exemplo$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           260M  1,3M  259M   1% /run
/dev/sda3       98G   15G   78G  17% /
tmpfs           1,3G    0  1,3G   0% /dev/shm
tmpfs           5,0M   4,0K  5,0M   1% /run/lock
/dev/sda2       512M   6,1M  506M   2% /boot/efi
tmpfs           260M  100K  260M   1% /run/user/1000
marco@Marco:~/exemplo$ killall sudoku
sudoku: no process found
marco@Marco:~/exemplo$ killall firefox
```

jobs, bg, and fg: Gerencia processos em segundo plano. Exemplo: `jobs` lista processos ativos, `bg` coloca um processo em segundo plano, e `fg` o traz para o primeiro plano.

```
[1]+ Stopped top
marco@Marco:~/exemplo$ jobs
[1]+ Stopped top
marco@Marco:~/exemplo$ bg 1
[1]+ top &
```

E após executar o comando `fg 1`:

```
top - 10:16:28 up 4:17, 1 user, load average: 0,22, 0,26, 0,26
Tasks: 171 total, 1 running, 170 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4,6 us, 1,1 sy, 0,0 ni, 93,7 id, 0,0 wa, 0,0 hi, 0,7 si, 0,0 st
MiB Mem : 2595,3 total, 462,2 free, 514,9 used, 1618,2 buff/cache
MiB Swap: 5372,0 total, 5371,7 free, 0,3 used, 1904,1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
  661 root        20   0   330440   95676   52628 S   7,3   3,6   1:25.61 Xorg
45709 marco      20   0   493356   48612   37888 S   4,3   1,8   0:34.57 xfce4-terminal
47191 marco      20   0   23804    4224    3456 R   1,0   0,2   0:00.22 top
 1034 marco      20   0   926612   99868   77552 S   0,7   3,8   0:19.91 xfwm4
 1039 marco      20   0   236752   24036   20068 S   0,3   0,9   0:02.71 xfsettingsd
 1103 marco      20   0   472116   46328   29236 S   0,3   1,7   0:10.10 xfdesktop
    1 root        20   0   167616   13020    8284 S   0,0   0,5   68:24.88 systemd
    2 root        20   0      0         0         0 S   0,0   0,0   0:00.04 kthreadd
```

gzip: Comprime arquivos. Exemplo: `gzip arquivo2.txt` cria "arquivo2.txt.gz".

gunzip: Descomprime arquivos compactados com `gzip`. Exemplo: `gunzip arquivo2.txt.gz` descomprime "arquivo2.txt.gz".

```
marco@Marco:~/exemplo$ gzip arquivo2.txt
marco@Marco:~/exemplo$ ls
arquivo2.txt.gz  exemplo.txt
marco@Marco:~/exemplo$ gunzip arquivo2.txt.gz
marco@Marco:~/exemplo$ ls
arquivo2.txt  exemplo.txt
```

tar: Arquia e comprime arquivos. Exemplo: `tar -cvf arquivo2.tar pasta` cria um arquivo "arquivo2.tar" contendo "pasta".

alias: Cria atalhos para comandos. Exemplo: `alias l='ls -la'` cria um atalho para listar arquivos detalhadamente.

xargs: Passa argumentos para comandos. Exemplo: `echo "arquivo1 arquivo2" | xargs rm` remove os arquivos especificados.

ln: Cria links simbólicos ou físicos. Exemplo: `ln -s exemplo.txt link.txt` cria um link simbólico "link.txt".

who: Exibe informações sobre usuários conectados.

```
marco@Marco:~/exemplo$ alias l='ls -la'
marco@Marco:~/exemplo$ l
total 40
drwxrwxr-x  2 marco marco  4096 jan 25 10:25 .
drwxr-x--- 15 marco marco  4096 jan 25 10:02 ..
-rw-rw-r--  1 marco marco 10240 jan 25 10:21 arquivo2.tar
-rw-rw-r--  1 marco marco   50 jan 25 09:12 arquivo2.txt
-rw-rw-r--  1 marco marco 10240 jan 25 10:25 arquivo.tar
-rw-rw-r--  1 marco marco    6 jan 25 09:06 exemplo.txt
marco@Marco:~/exemplo$ echo "arquivo2.txt" | xargs rm
marco@Marco:~/exemplo$ ls
arquivo2.tar  arquivo.tar  exemplo.txt
marco@Marco:~/exemplo$ ln -s exemplo.txt link.txt
marco@Marco:~/exemplo$ ls
arquivo2.tar  arquivo.tar  exemplo.txt  link.txt
marco@Marco:~/exemplo$ who
marco      tty7          2025-01-20 22:18 (:0)
```

su: Troca para outro usuário. Exemplo: `su username` troca para o usuário "username".

sudo: Executa comandos com privilégios administrativos. Exemplo: `sudo apt update` atualiza pacotes com privilégios de administrador.

passwd: Altera a senha do usuário atual ou de outro, se permitido.

```
marco@Marco:~/exemplo$ ln -s exemplo.txt link.txt
marco@Marco:~/exemplo$ ls
arquivo2.tar  arquivo.tar  exemplo.txt  link.txt
marco@Marco:~/exemplo$ who
marco      tty7          2025-01-20 22:18 (:0)
marco@Marco:~/exemplo$ su username
su: user username does not exist or the user entry does not contain all the required fields
marco@Marco:~/exemplo$ sudo apt update
[sudo] password for marco:
marco is not in the sudoers file. This incident will be reported.
marco@Marco:~/exemplo$ passwd
Changing password for marco.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```