

Felipe Archanjo da Cunha Mendes

Giorgio Artur Garcia Braz

Pamella Lissa Sato Tamura

Thaynara Ribeiro Falcão dos Santos

Laboratório 1: Instalar GNU/Linux e compilar o núcleo

Relatório técnico de atividade prática solicitado pelo professor Rodrigo Campiolo na disciplina de Sistemas Operacionais do Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Universidade Tecnológica Federal do Paraná – UTFPR

Departamento Acadêmico de Computação – DACOM

Bacharelado em Ciência da Computação – BCC

Campo Mourão

Março / 2022

Sumário

1	Introdução	3
2	Objetivos	3
3	Fundamentação	3
4	Materiais	3
4.1	Parte I	3
4.2	Parte II	3
5	Procedimentos e Resultados	4
5.1	Instalação do Ubuntu	4
5.1.1	Verificar a listagem de processos em execução	5
5.1.2	Verificar o espaço em disco em cada partição	5
5.1.3	Verificar a memória disponível	6
5.2	Verificar a rede	7
5.2.1	Configurar os repositórios do apt-get e adicionar os repositórios de atualização de segurança	10
5.2.2	Verificar o kernel atual	11
5.2.3	Testando comandos	11
6	Compilação do kernel Linux	16
7	Conclusões	18
8	Referências	18

1 Introdução

O relatório está dividido em duas partes, sendo a primeira reservada para descrever o processo de instalação e configuração do sistema operacional (Linux) em uma máquina virtual (VirtualBox¹), além dos comandos básicos presentes no Sistema Operacional. Já a segunda parte está relacionada à configuração, compilação e instalação do kernel do Linux.

2 Objetivos

O objetivo desse trabalho consiste em, primeiramente, instalar um sistema operacional (Linux). Além de compreender a estrutura básica do Linux e alguns de seus comandos. Por fim, configurar e compilar o Kernel Linux em sua última versão estável.

3 Fundamentação

A instalação e compilação do sistema operacional Ubuntu foram baseadas e fundamentadas por documentações apresentadas na referência, no final do documento, além do material fornecido pelo próprio Prof. Dr. Rodrigo Campiolo, assim como as definições de comandos e conceitos que foram obtidos por pesquisas sequenciais por cada membro da equipe (também referenciadas ao final do trabalho).

4 Materiais

4.1 Parte I

- Ubuntu 20.04 LTS (ISO)
- VirtualBox 6.1.32
- Windows 10 (64 bits)
- i7-i7-4500U
- 8GB de RAM

4.2 Parte II

- Ubuntu 20.04 LTS (ISO)

¹ <https://www.virtualbox.org>

- Parallels Desktop 17.1.1
- MacBook Air (mid. 2020)
- macOS Monterey (v12.2.1)
- Apple Silicon M1
- 8GB de RAM

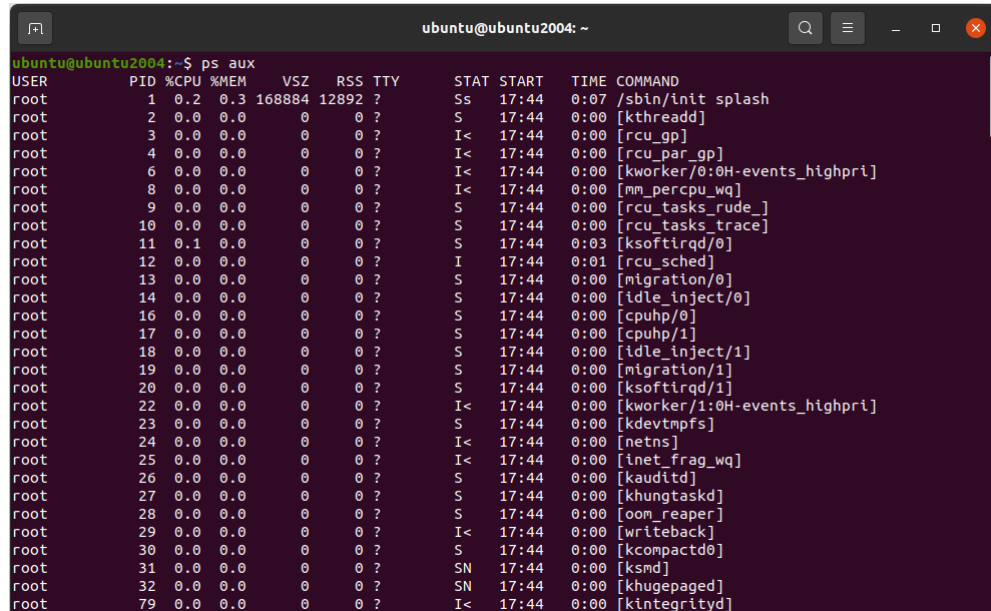
5 Procedimentos e Resultados

5.1 Instalação do Ubuntu

Para a instalação do Ubuntu na máquina virtual (VirtualBox) foi necessário obter a imagem do sistema operacional. Após a imagem formada e as configurações definidas, é possível verificar as seguintes ações:

5.1.1 Verificar a listagem de processos em execução

Foi utilizado o comando `ps aux` para a listagem de processos em execução. Segue imagem (Figura 1) da verificação do comando `ps aux`.



```

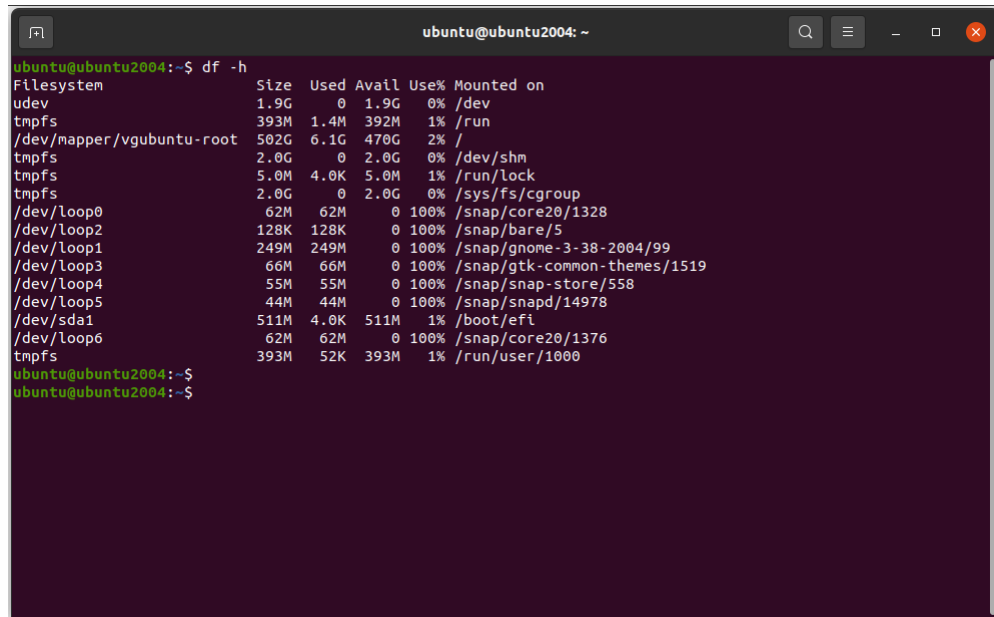
ubuntu@ubuntu2004: ~
ubuntu@ubuntu2004:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  0.3 168884 12892 ?        Ss   17:44   0:07 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    17:44   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   17:44   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   17:44   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   17:44   0:00 [kworker/0:0H-events_highpri]
root           8  0.0  0.0      0     0 ?        I<   17:44   0:00 [mm_percpu_wq]
root           9  0.0  0.0      0     0 ?        S    17:44   0:00 [rcu_tasks_rude_]
root          10  0.0  0.0      0     0 ?        S    17:44   0:00 [rcu_tasks_trace]
root          11  0.1  0.0      0     0 ?        S    17:44   0:03 [ksoftirqd/0]
root          12  0.0  0.0      0     0 ?        I    17:44   0:01 [rcu_sched]
root          13  0.0  0.0      0     0 ?        S    17:44   0:00 [migration/0]
root          14  0.0  0.0      0     0 ?        S    17:44   0:00 [idle_inject/0]
root          16  0.0  0.0      0     0 ?        S    17:44   0:00 [cpuhp/0]
root          17  0.0  0.0      0     0 ?        S    17:44   0:00 [cpuhp/1]
root          18  0.0  0.0      0     0 ?        S    17:44   0:00 [idle_inject/1]
root          19  0.0  0.0      0     0 ?        S    17:44   0:00 [migration/1]
root          20  0.0  0.0      0     0 ?        S    17:44   0:00 [ksoftirqd/1]
root          22  0.0  0.0      0     0 ?        I<   17:44   0:00 [kworker/1:0H-events_highpri]
root          23  0.0  0.0      0     0 ?        S    17:44   0:00 [kdevtmpfs]
root          24  0.0  0.0      0     0 ?        I<   17:44   0:00 [netns]
root          25  0.0  0.0      0     0 ?        I<   17:44   0:00 [inet_frag_wq]
root          26  0.0  0.0      0     0 ?        S    17:44   0:00 [kauditd]
root          27  0.0  0.0      0     0 ?        S    17:44   0:00 [khungtaskd]
root          28  0.0  0.0      0     0 ?        S    17:44   0:00 [oom_reaper]
root          29  0.0  0.0      0     0 ?        I<   17:44   0:00 [writeback]
root          30  0.0  0.0      0     0 ?        S    17:44   0:00 [kcompactd0]
root          31  0.0  0.0      0     0 ?        SN   17:44   0:00 [ksmd]
root          32  0.0  0.0      0     0 ?        SN   17:44   0:00 [khugepaged]
root          79  0.0  0.0      0     0 ?        I<   17:44   0:00 [kintegrityd]

```

Figura 1 – Demonstração do comando `ps aux`

5.1.2 Verificar o espaço em disco em cada partição

Foi utilizado o comando `df -h` para verificar o espaço em disco em cada partição. Ao observar a Figura 2, nota-se partições que foram já pré definidas a partir da imagem obtida do Ubuntu citada anteriormente (tópico 5.1).

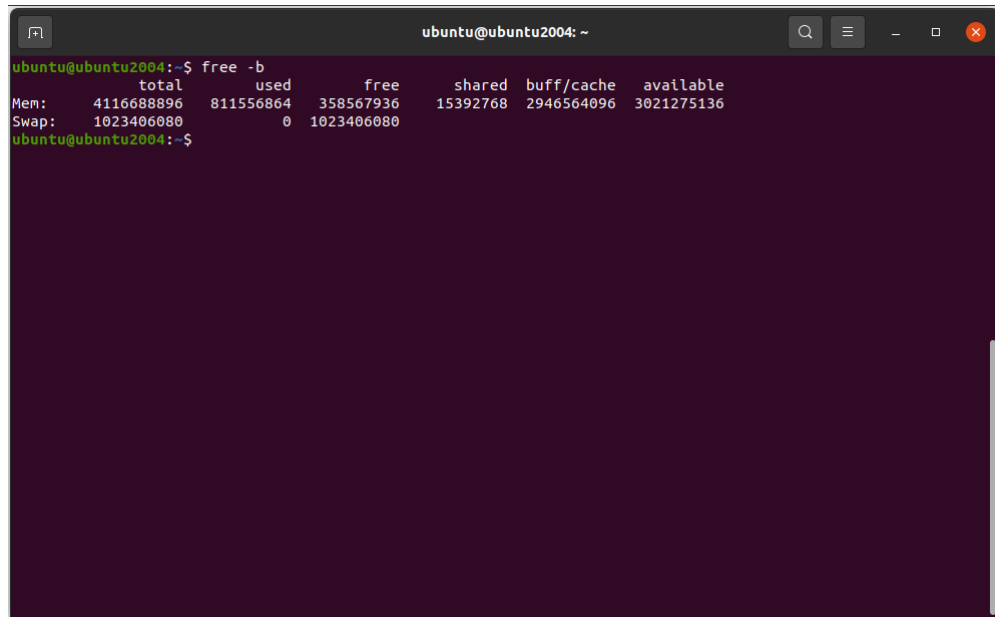


```
ubuntu@ubuntu2004: ~  
ubuntu@ubuntu2004:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
udev            1.9G   0    1.9G   0% /dev  
tmpfs           393M  1.4M  392M   1% /run  
/dev/mapper/vgubuntu-root 502G  6.1G  470G   2% /  
tmpfs           2.0G   0    2.0G   0% /dev/shm  
tmpfs           5.0M  4.0K  5.0M   1% /run/lock  
tmpfs           2.0G   0    2.0G   0% /sys/fs/cgroup  
/dev/loop0       62M   62M   0 100% /snap/core20/1328  
/dev/loop2      128K  128K   0 100% /snap/bare/5  
/dev/loop1      249M  249M   0 100% /snap/gnome-3-38-2004/99  
/dev/loop3       66M   66M   0 100% /snap/gtk-common-themes/1519  
/dev/loop4       55M   55M   0 100% /snap/snap-store/558  
/dev/loop5       44M   44M   0 100% /snap/snapd/14978  
/dev/sda1       511M  4.0K  511M   1% /boot/efi  
/dev/loop6       62M   62M   0 100% /snap/core20/1376  
tmpfs           393M  52K  393M   1% /run/user/1000  
ubuntu@ubuntu2004:~$  
ubuntu@ubuntu2004:~$
```

Figura 2 – Demonstração do comando `df -h`

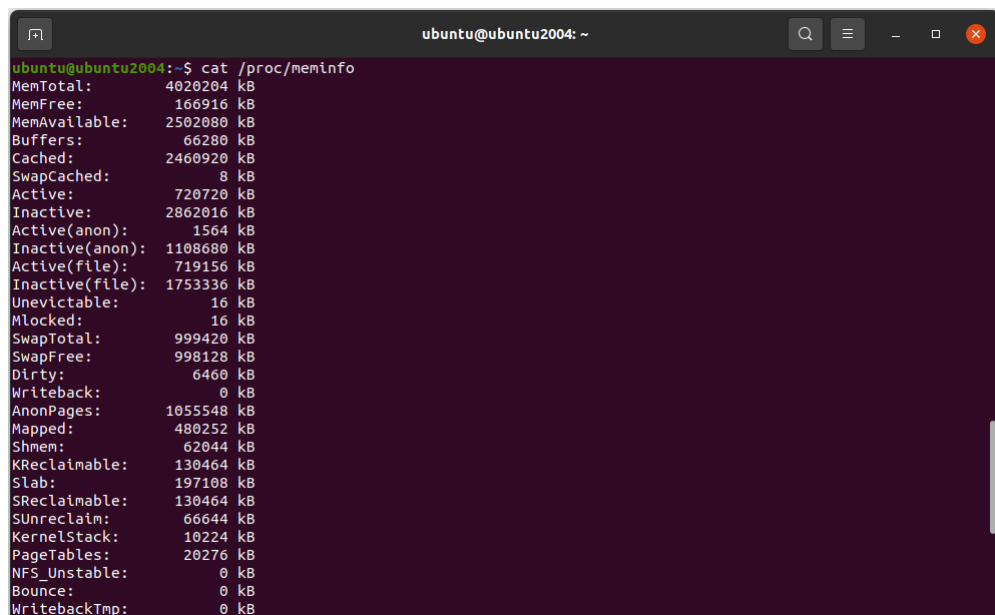
5.1.3 Verificar a memória disponível

Os comandos a seguir foram utilizados para verificar a memória disponível, sendo o comando `free -b` para verificar a memória total física e `cat /proc/meminfo` para verificar a memória livre. Pode-se observar a seguir os resultados respectivos desses comandos.



```
ubuntu@ubuntu2004: ~  
ubuntu@ubuntu2004:~$ free -b  
              total        used        free      shared  buff/cache   available  
Mem:      4116688896    811556864    358567936    15392768    2946564096    3021275136  
Swap:      1023406080           0    1023406080
```

Figura 3 – Demonstração do comando free -b



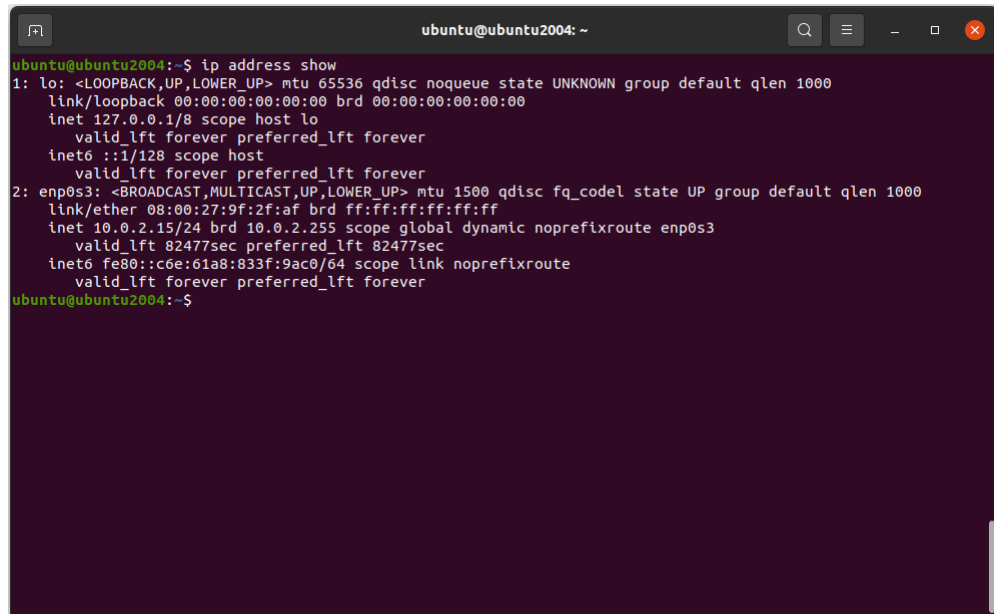
```
ubuntu@ubuntu2004:~$ cat /proc/meminfo  
MemTotal:        4020204 kB  
MemFree:         166916 kB  
MemAvailable:    2502080 kB  
Buffers:         66280 kB  
Cached:          2460920 kB  
SwapCached:       8 kB  
Active:          720720 kB  
Inactive:        2862016 kB  
Active(anon):     1564 kB  
Inactive(anon):  1108680 kB  
Active(file):     719156 kB  
Inactive(file):  1753336 kB  
Unevictable:      16 kB  
Mlocked:          16 kB  
SwapTotal:       999420 kB  
SwapFree:        998128 kB  
Dirty:           6460 kB  
Writeback:        0 kB  
AnonPages:       1055548 kB  
Mapped:          480252 kB  
Shmem:           62044 kB  
KReclaimable:    130464 kB  
Slab:            197108 kB  
SReclaimable:    130464 kB  
SUnreclaim:      66644 kB  
KernelStack:     10224 kB  
PageTables:       20276 kB  
NFS_Unstable:     0 kB  
Bounce:           0 kB  
WritebackTmp:     0 kB
```

Figura 4 – Demonstração do comando cat /proc/meminfo

5.2 Verificar a rede

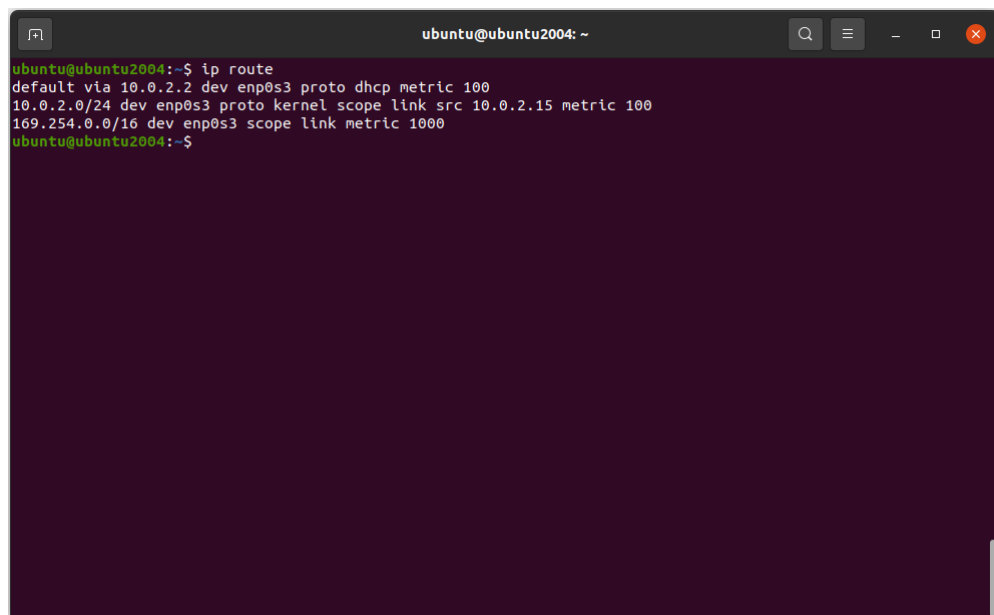
Foram utilizados cinco comandos para a verificação da rede. O primeiro, ip address show (Figura 5), para exibir as interfaces de rede. O segundo comando, ip route (Figura 6), com a função de exibir a tabela de roteamento. Em seguida foi utilizado cat /etc/resolv.conf (Figura 7) para a configuração do dns. A configuração das interfaces de rede foi definida pelo comando cat /etc/network/interfaces (Figura 8). E, por fim, o comando ping

(Figura 9) foi utilizado para verificar o tempo de transferência de um pacote de dados de um dispositivo para um servidor (taxa de latência).



```
ubuntu@ubuntu2004: ~  
ubuntu@ubuntu2004:~$ ip address show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
        inet 127.0.0.1/8 scope host lo  
            valid_lft forever preferred_lft forever  
        inet6 ::1/128 scope host  
            valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:9f:2f:af 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 82477sec preferred_lft 82477sec  
        inet6 fe80::c6e:61a8:833f:9ac0/64 scope link noprefixroute  
            valid_lft forever preferred_lft forever  
ubuntu@ubuntu2004:~$
```

Figura 5 – Demonstração do comando ip address show



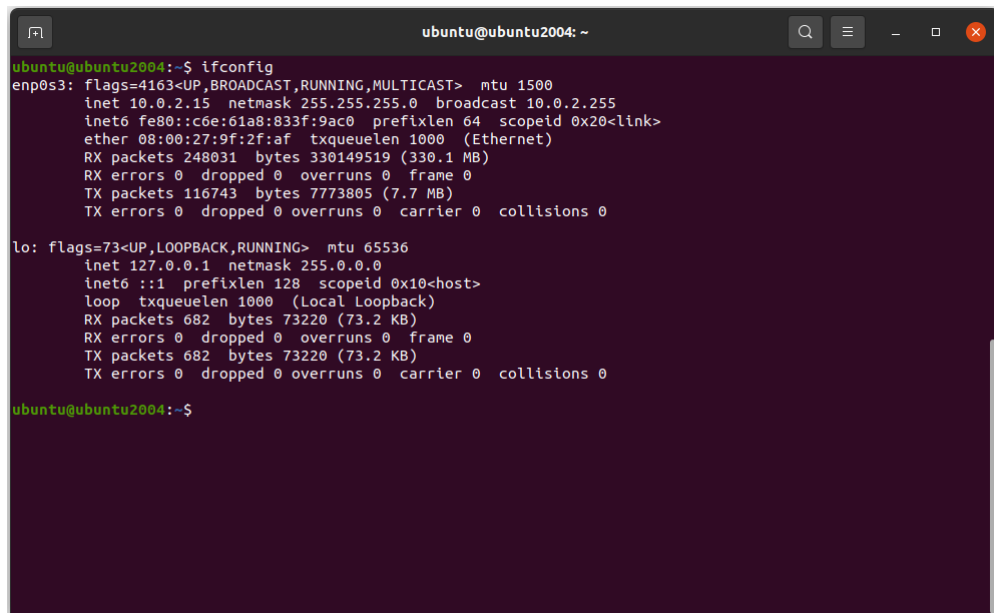
```
ubuntu@ubuntu2004: ~  
ubuntu@ubuntu2004:~$ ip route  
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100  
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100  
169.254.0.0/16 dev enp0s3 scope link metric 1000  
ubuntu@ubuntu2004:~$
```

Figura 6 – Demonstração do comando ip route



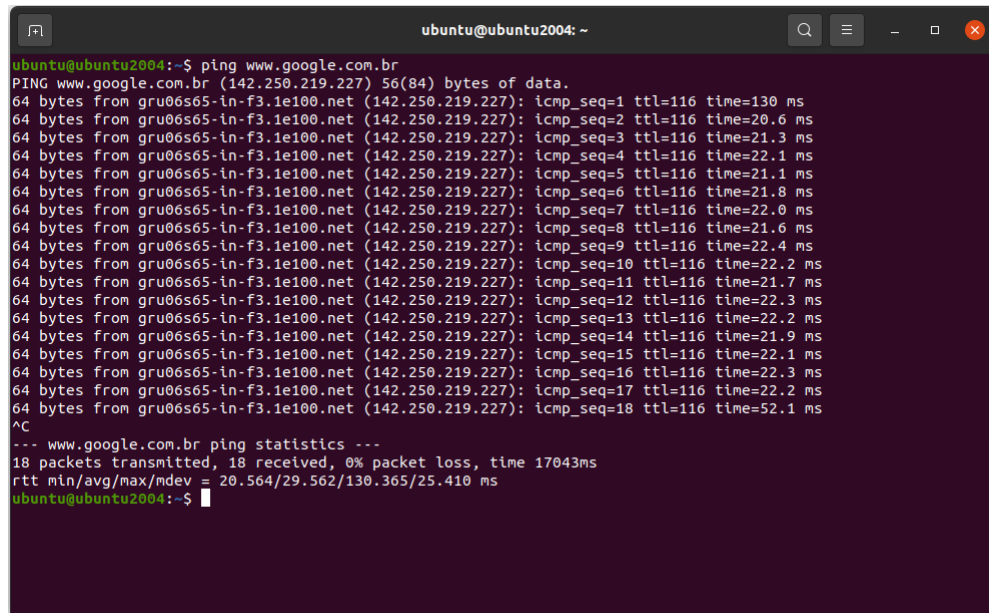
```
ubuntu@ubuntu2004: ~  
$ cat /etc/resolv.conf  
# This file is managed by man:systemd-resolved(8). Do not edit.  
#  
# This is a dynamic resolv.conf file for connecting local clients to the  
# internal DNS stub resolver of systemd-resolved. This file lists all  
# configured search domains.  
#  
# Run "resolvectl status" to see details about the uplink DNS servers  
# currently in use.  
#  
# Third party programs must not access this file directly, but only through the  
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,  
# replace this symlink by a static file or a different symlink.  
#  
# See man:systemd-resolved.service(8) for details about the supported modes of  
# operation for /etc/resolv.conf.  
  
nameserver 127.0.0.53  
options edns0 trust-ad  
ubuntu@ubuntu2004:~$
```

Figura 7 – Demonstração do comando `cat /etc/resolv.conf`



```
ubuntu@ubuntu2004: ~  
$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::c6e:61a8:833f:9ac0 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:9f:2f:af txqueuelen 1000 (Ethernet)  
    RX packets 248031 bytes 330149519 (330.1 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 116743 bytes 7773805 (7.7 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 682 bytes 73220 (73.2 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 682 bytes 73220 (73.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
ubuntu@ubuntu2004:~$
```

Figura 8 – Demonstração do comando `ifconfig`

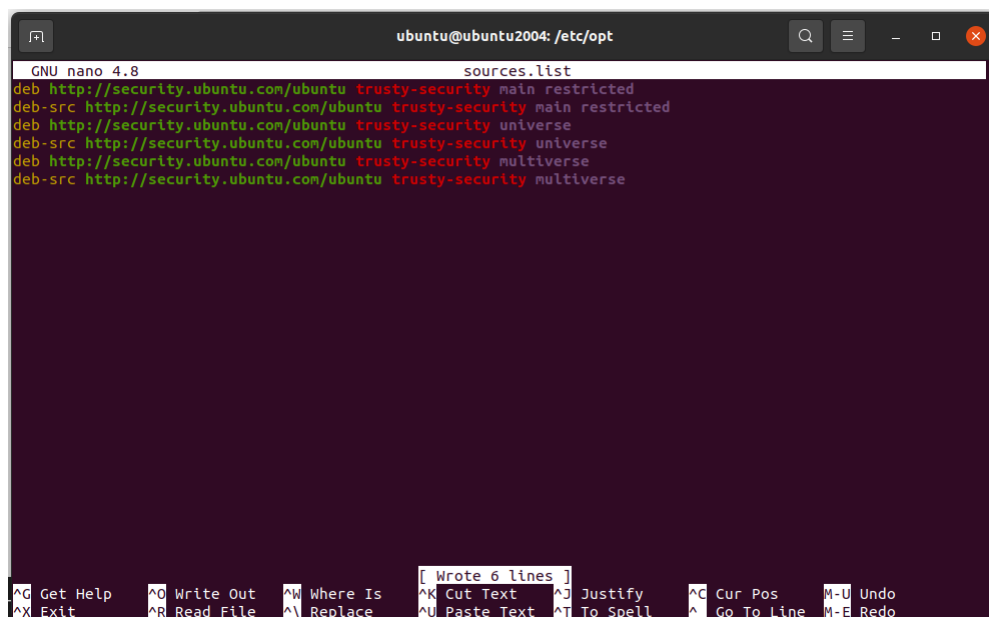


```
ubuntu@ubuntu2004: ~  
ubuntu@ubuntu2004:~$ ping www.google.com.br  
PING www.google.com.br (142.250.219.227) 56(84) bytes of data:  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=1 ttl=116 time=130 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=2 ttl=116 time=20.6 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=3 ttl=116 time=21.3 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=4 ttl=116 time=22.1 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=5 ttl=116 time=21.1 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=6 ttl=116 time=21.8 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=7 ttl=116 time=22.0 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=8 ttl=116 time=21.6 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=9 ttl=116 time=22.4 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=10 ttl=116 time=22.2 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=11 ttl=116 time=21.7 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=12 ttl=116 time=22.3 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=13 ttl=116 time=22.2 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=14 ttl=116 time=21.9 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=15 ttl=116 time=22.1 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=16 ttl=116 time=22.3 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=17 ttl=116 time=22.2 ms  
64 bytes from gru06s65-in-f3.1e100.net (142.250.219.227): icmp_seq=18 ttl=116 time=52.1 ms  
^C  
--- www.google.com.br ping statistics ---  
18 packets transmitted, 18 received, 0% packet loss, time 17043ms  
rtt min/avg/max/mdev = 20.564/29.562/130.365/25.410 ms  
ubuntu@ubuntu2004:~$
```

Figura 9 – Demonstração do comando ping www.google.com

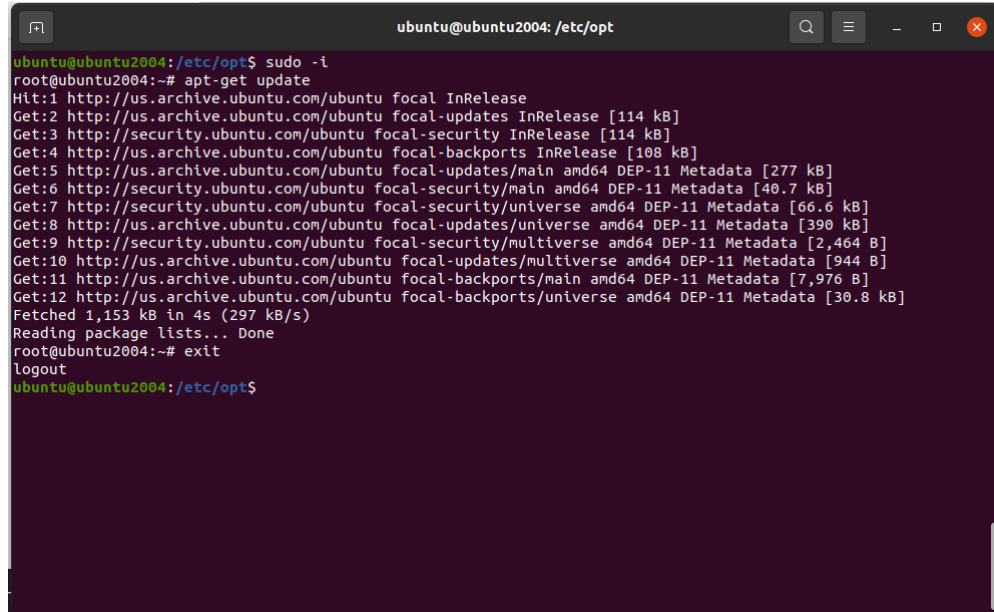
5.2.1 Configurar os repositórios do apt-get e adicionar os repositórios de atualização de segurança

Ao executar o comando `cd /etc/apt`, o usuário foi redirecionado para os diretórios respectivos. Em seguida, foi criado um arquivo `.list` para a adição do repositório de segurança com o comando `nano sources.list` (Figura 10). Assim, o `apt-get update` (Figura 11) foi executado para a atualização do arquivo.



```
ubuntu@ubuntu2004: /etc/apt  
GNU nano 4.8 sources.list  
deb http://security.ubuntu.com/ubuntu trusty-security main restricted  
deb-src http://security.ubuntu.com/ubuntu trusty-security main restricted  
deb http://security.ubuntu.com/ubuntu trusty-security universe  
deb-src http://security.ubuntu.com/ubuntu trusty-security universe  
deb http://security.ubuntu.com/ubuntu trusty-security multiverse  
deb-src http://security.ubuntu.com/ubuntu trusty-security multiverse  
[ Wrote 6 lines ]  
^C Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^M-U Undo  
^X Exit ^R Read File ^A Replace ^U Paste Text ^T To Spell ^G Go To Line ^M-E Redo
```

Figura 10 – Demonstração do comandos `cd /etc/apt` E `nano sources.list`

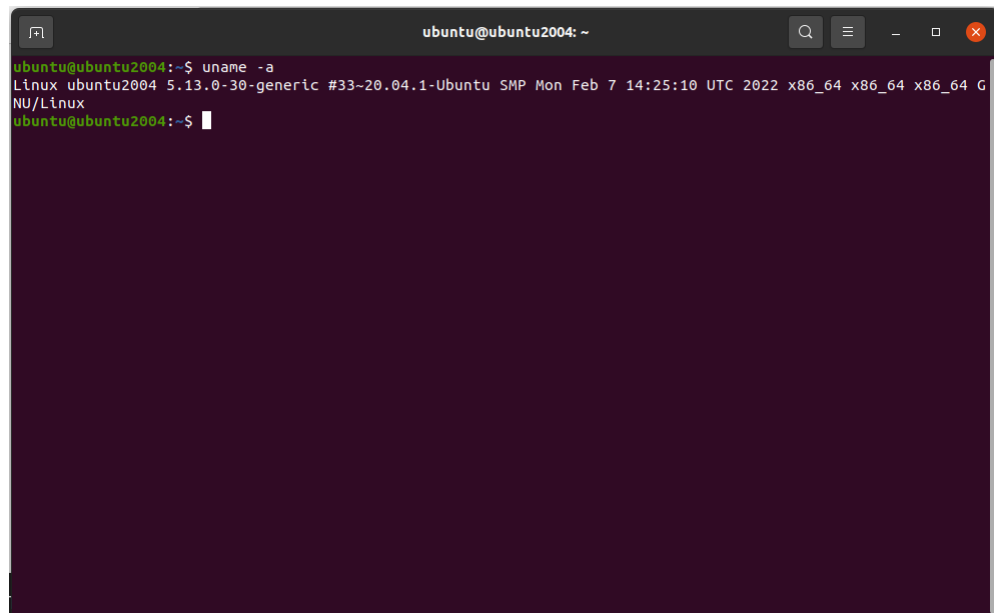
A terminal window titled 'ubuntu@ubuntu2004: /etc/opt' showing the execution of the 'apt-get update' command. The output lists various package repositories and their metadata sizes, such as 'focal InRelease', 'focal-updates InRelease', 'focal-security InRelease', and 'focal-backports InRelease'. It also shows the total size of the fetched data (1,153 kB) and the time taken (4s). The user then enters 'exit' and the prompt changes back to 'ubuntu@ubuntu2004: /etc/opt\$'.

```
ubuntu@ubuntu2004: /etc/opt$ sudo -i
root@ubuntu2004:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [277 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.7 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [66.6 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [390 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:11 http://us.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [7,976 B]
Get:12 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.8 kB]
Fetched 1,153 kB in 4s (297 kB/s)
Reading package lists... Done
root@ubuntu2004:~# exit
logout
ubuntu@ubuntu2004: /etc/opt$
```

Figura 11 – Demonstração do comando apt-get update

5.2.2 Verificar o kernel atual

Na Figura 12, podemos observar o resultado após a execução do comando `uname -a` para verificação do kernel atual.

A terminal window titled 'ubuntu@ubuntu2004: ~' showing the execution of the 'uname -a' command. The output displays system information including the kernel version (5.13.0-30-generic), architecture (x86_64), and the date and time of the execution (Mon Feb 7 14:25:10 UTC 2022).

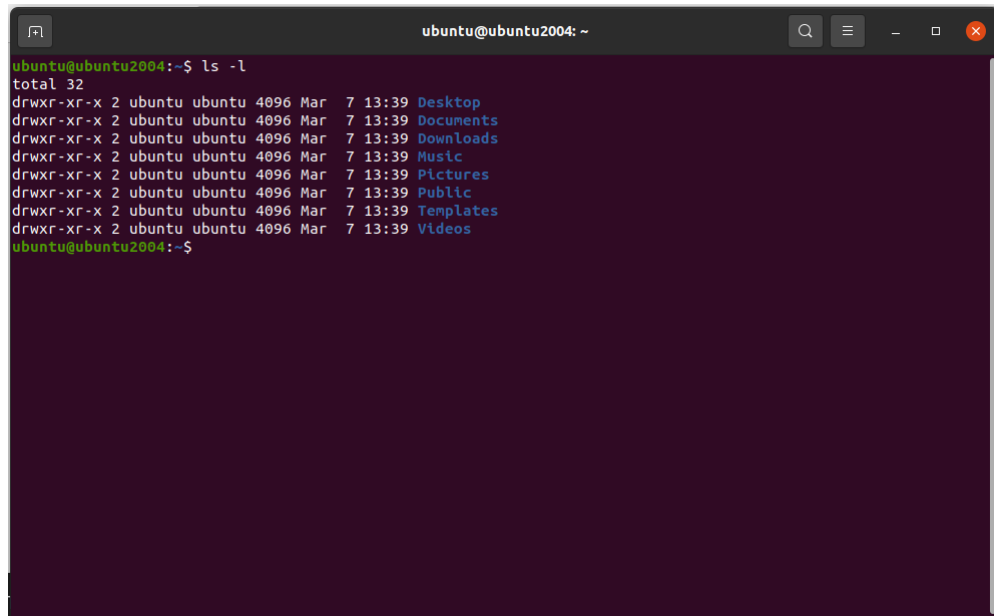
```
ubuntu@ubuntu2004:~$ uname -a
Linux ubuntu2004 5.13.0-30-generic #33~20.04.1-Ubuntu SMP Mon Feb 7 14:25:10 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
ubuntu@ubuntu2004:~$
```

Figura 12 – Demonstração do comando uname -a

5.2.3 Testando comandos

Clear: O comando `clear` tem a função de limpar o terminal. Normalmente, é executado após várias linhas de comando para apagar os históricos dos comandos anteriores.

ls -l: Esse comando é executado para obter a listagem do conteúdo do diretório atual. (Figura 13)

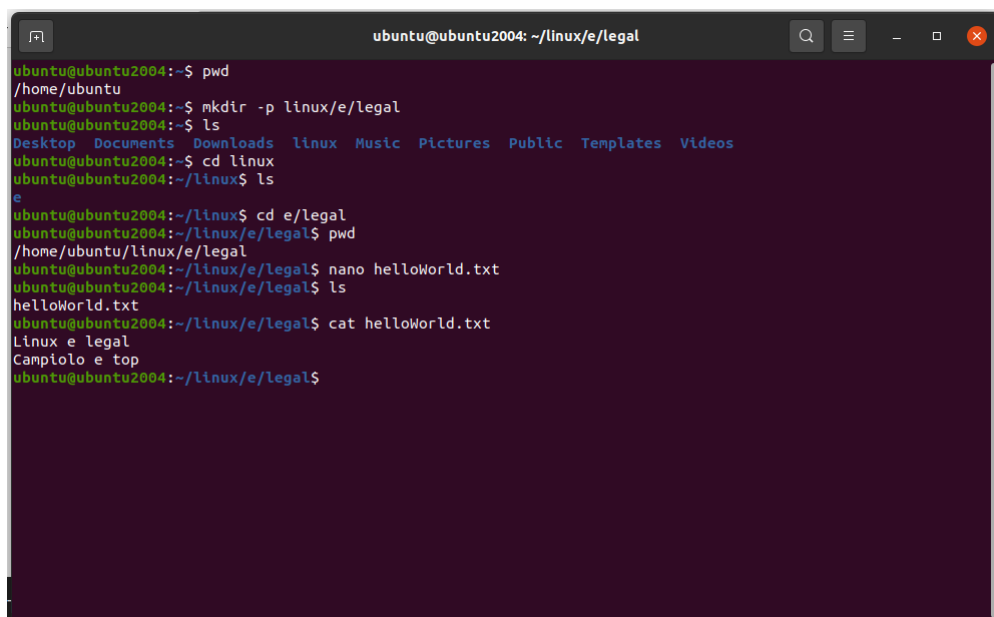
A terminal window titled 'ubuntu@ubuntu2004: ~' with a search icon, menu icon, and window controls. The command 'ls -l' has been executed, showing a detailed listing of the home directory. The output includes file permissions, owner, group, size, date, and filename for various directories like Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos.

```
ubuntu@ubuntu2004:~$ ls -l
total 32
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Desktop
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Documents
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Downloads
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Music
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Pictures
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Public
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Templates
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar  7 13:39 Videos
ubuntu@ubuntu2004:~$
```

Figura 13 – Demonstração do comando ls -l

cd: O comando acima é responsável por mudar a localização atual do usuário para outros diretórios. Onde é executado e sucedido por um determinado diretório. (Figura 14)

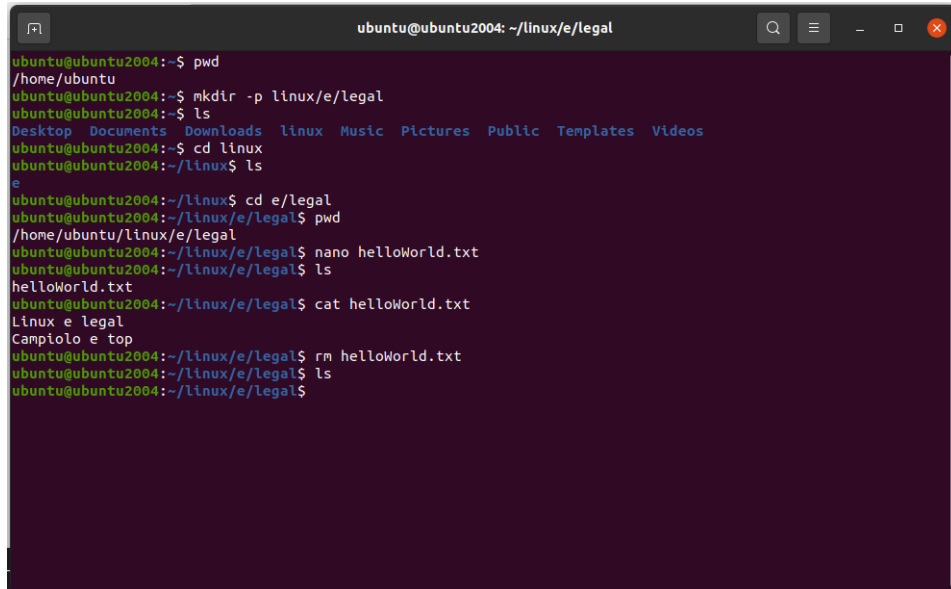
cat: O comando cat é utilizado para exibir o conteúdo do arquivo indicado por parâmetro pelo usuário. (Figura 14)

A terminal window titled 'ubuntu@ubuntu2004: ~/linux/e/legal' with a search icon, menu icon, and window controls. The user performs a series of commands: 'pwd' (shows /home/ubuntu), 'mkdir -p linux/e/legal' (creates the directory), 'ls' (shows the directory structure), 'cd linux' (changes to /home/ubuntu/linux), 'ls' (shows contents of /home/ubuntu/linux), 'cd e/legal' (changes to /home/ubuntu/linux/e/legal), 'pwd' (confirms the current directory), 'nano helloWorld.txt' (creates a file), 'ls' (confirms the file), and 'cat helloWorld.txt' (displays the file content: 'Linux e legal' and 'Campiolo e top').

```
ubuntu@ubuntu2004:~$ pwd
/home/ubuntu
ubuntu@ubuntu2004:~$ mkdir -p linux/e/legal
ubuntu@ubuntu2004:~$ ls
Desktop  Documents  Downloads  linux  Music  Pictures  Public  Templates  Videos
ubuntu@ubuntu2004:~$ cd linux
ubuntu@ubuntu2004:~/linux$ ls
e
ubuntu@ubuntu2004:~/linux$ cd e/legal
ubuntu@ubuntu2004:~/linux/e/legal$ pwd
/home/ubuntu/linux/e/legal
ubuntu@ubuntu2004:~/linux/e/legal$ nano helloWorld.txt
ubuntu@ubuntu2004:~/linux/e/legal$ ls
helloWorld.txt
ubuntu@ubuntu2004:~/linux/e/legal$ cat helloWorld.txt
Linux e legal
Campiolo e top
ubuntu@ubuntu2004:~/linux/e/legal$
```

Figura 14 – Demonstração dos comandos cat e cd

rm: Esse comando é executado para remover/deletar determinado arquivo indicado pelo usuário. É executado indicando o nome do arquivo posteriormente. (Figura 15)

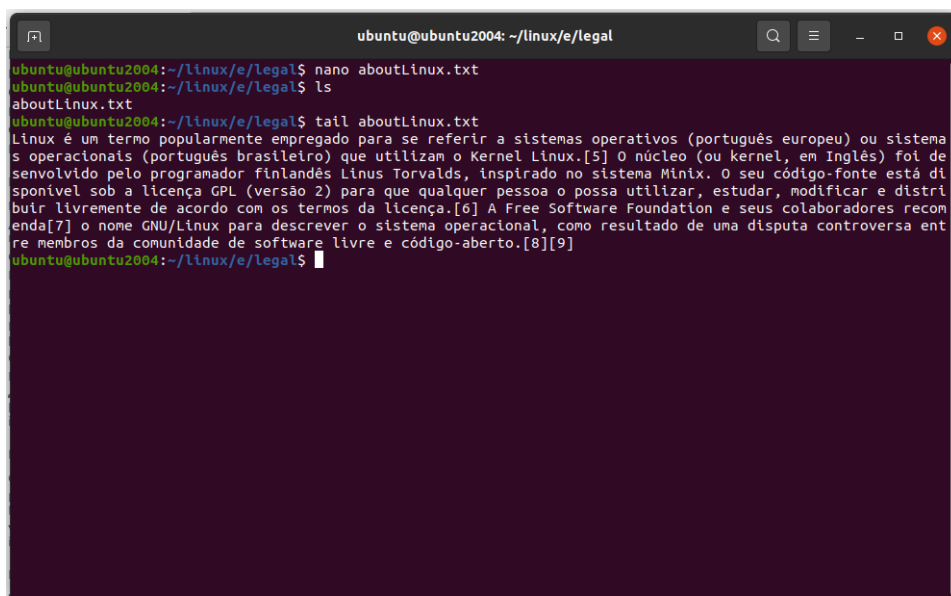


```
ubuntu@ubuntu2004: ~/linux/e/legal
ubuntu@ubuntu2004:~$ pwd
/home/ubuntu
ubuntu@ubuntu2004:~$ mkdir -p linux/e/legal
ubuntu@ubuntu2004:~$ ls
Desktop  Documents  Downloads  linux  Music  Pictures  Public  Templates  Videos
ubuntu@ubuntu2004:~$ cd linux
ubuntu@ubuntu2004:~/linux$ ls
e
ubuntu@ubuntu2004:~/linux$ cd e/legal
ubuntu@ubuntu2004:~/linux/e/legal$ pwd
/home/ubuntu/linux/e/legal
ubuntu@ubuntu2004:~/linux/e/legal$ nano helloWorld.txt
ubuntu@ubuntu2004:~/linux/e/legal$ ls
helloWorld.txt
ubuntu@ubuntu2004:~/linux/e/legal$ cat helloWorld.txt
Linux e legal
Campiolo e top
ubuntu@ubuntu2004:~/linux/e/legal$ rm helloWorld.txt
ubuntu@ubuntu2004:~/linux/e/legal$ ls
ubuntu@ubuntu2004:~/linux/e/legal$
```

Figura 15 – Demonstração do comando rm

nano (ou pico): O comando nano/pico é executado para abrir um editor de texto e, assim, permitir que o usuário crie um arquivo texto. (Figura 16)

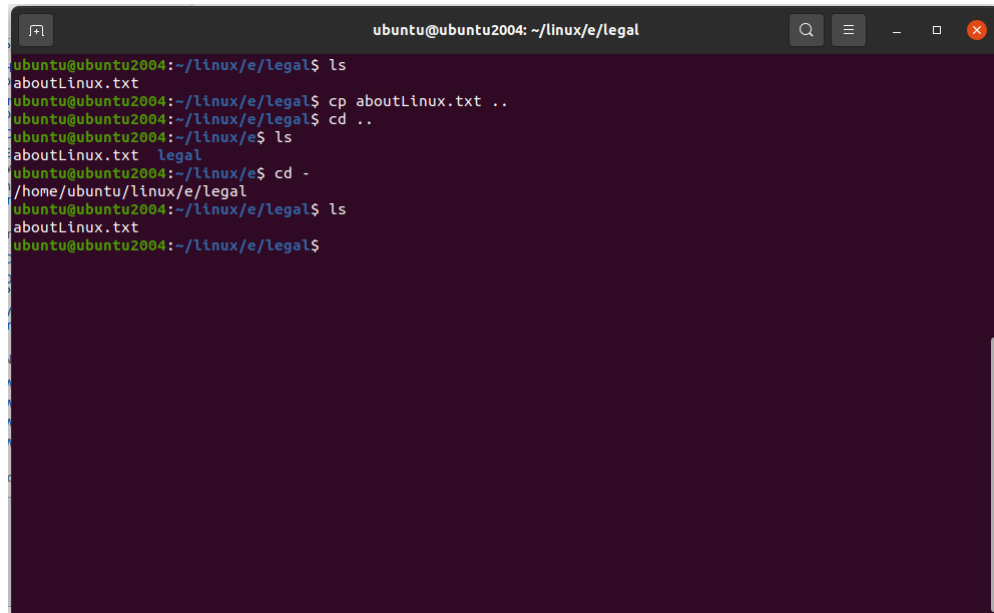
tail: O comando tail exhibe, normalmente, as 10 últimas linhas do arquivo texto. Que também pode ser alterada para outra quantidade definida pelo próprio usuário. É utilizado com a indicação tail seguido pelo nome do arquivo. (Figura 16)



```
ubuntu@ubuntu2004:~/linux/e/legal
ubuntu@ubuntu2004:~/linux/e/legal$ nano aboutLinux.txt
ubuntu@ubuntu2004:~/linux/e/legal$ ls
aboutLinux.txt
ubuntu@ubuntu2004:~/linux/e/legal$ tail aboutLinux.txt
Linux é um termo popularmente empregado para se referir a sistemas operativos (português europeu) ou sistemas operacionais (português brasileiro) que utilizam o Kernel Linux.[5] O núcleo (ou kernel, em inglês) foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema Minix. O seu código-fonte está disponível sob a licença GPL (versão 2) para que qualquer pessoa o possa utilizar, estudar, modificar e distribuir livremente de acordo com os termos da licença.[6] A Free Software Foundation e seus colaboradores recomendam[7] o nome GNU/Linux para descrever o sistema operacional, como resultado de uma disputa controversa entre membros da comunidade de software livre e código-aberto.[8][9]
ubuntu@ubuntu2004:~/linux/e/legal$
```

Figura 16 – Demonstração dos comandos nano e tail

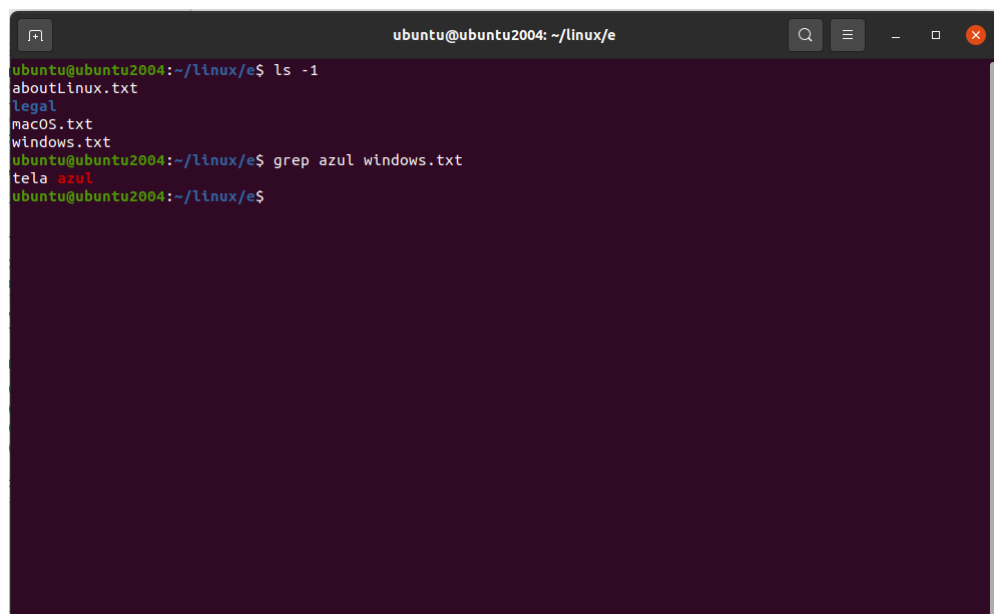
cp: O comando cp (copy) é utilizado para copiar arquivos e diretórios para outros locais. É executado pelo nome de arquivo de origem seguido pelo arquivo de destino (Figura 17)

A terminal window titled 'ubuntu@ubuntu2004: ~/linux/e/legal' with search, menu, and window control icons. The command history is as follows:

```
ubuntu@ubuntu2004:~/linux/e/legal$ ls
aboutLinux.txt
ubuntu@ubuntu2004:~/linux/e/legal$ cp aboutLinux.txt ..
ubuntu@ubuntu2004:~/linux/e/legal$ cd ..
ubuntu@ubuntu2004:~/linux/e$ ls
aboutLinux.txt  legal
ubuntu@ubuntu2004:~/linux/e$ cd -
/home/ubuntu/linux/e/legal
ubuntu@ubuntu2004:~/linux/e/legal$ ls
aboutLinux.txt
ubuntu@ubuntu2004:~/linux/e/legal$
```

Figura 17 – Demonstração do comando cp

grep: O comando grep é utilizado para exibir uma determinada string indicada pelo usuário seguido pelo nome do arquivo respectivo. (Figura 18)

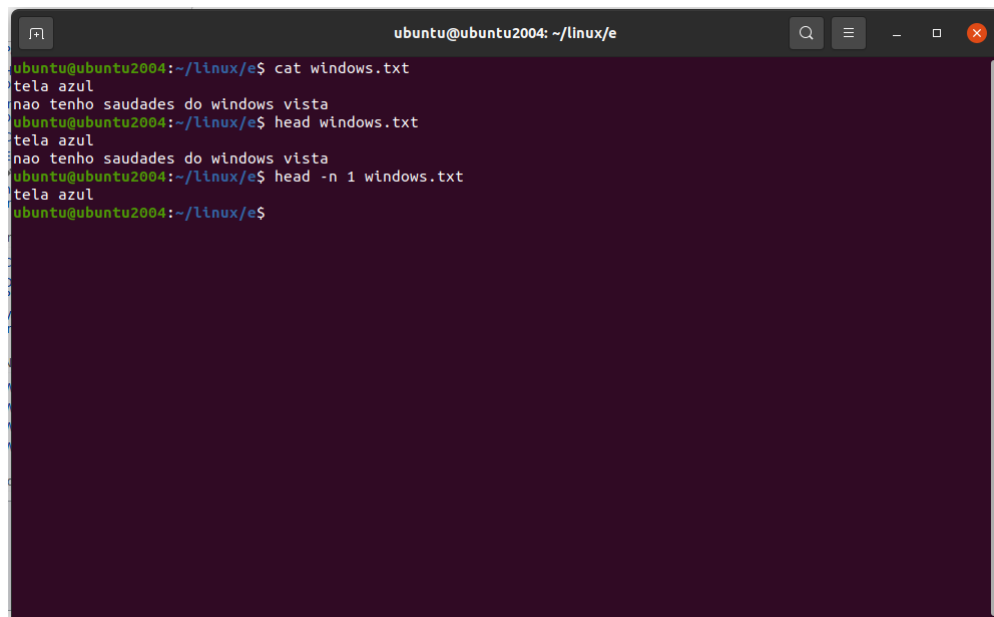
A terminal window titled 'ubuntu@ubuntu2004: ~/linux/e' with search, menu, and window control icons. The command history is as follows:

```
ubuntu@ubuntu2004:~/linux/e$ ls -l
aboutLinux.txt
legal
macOS.txt
windows.txt
ubuntu@ubuntu2004:~/linux/e$ grep azul windows.txt
tela azul
ubuntu@ubuntu2004:~/linux/e$
```

Figura 18 – Demonstração do comando grep

head: Esse comando é semelhante ao comando tail (tópico 5.1.7.7) porém ao invés de exibir as 10 últimas linhas ele mostra as primeiras 10 linhas do arquivo texto. Como

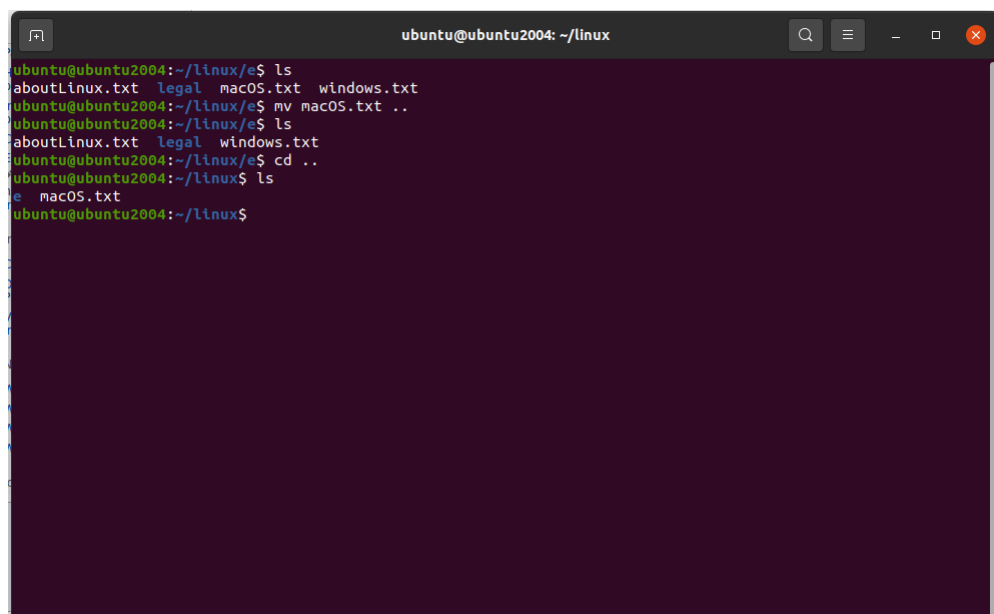
o .txt criado não possui dez linhas suficientes, é exibido as linhas totais desse arquivo. (Figura 19)



```
ubuntu@ubuntu2004: ~/linux/e
ubuntu@ubuntu2004:~/linux/e$ cat windows.txt
tela azul
nao tenho saudades do windows vista
ubuntu@ubuntu2004:~/linux/e$ head windows.txt
tela azul
nao tenho saudades do windows vista
ubuntu@ubuntu2004:~/linux/e$ head -n 1 windows.txt
tela azul
ubuntu@ubuntu2004:~/linux/e$
```

Figura 19 – Demonstração do comando head

mv: O comando mv é utilizado para mover arquivos de um diretório para outro. Ele é executado seguido pelo nome do arquivo de origem e pelo nome do destino. (Figura 20)



```
ubuntu@ubuntu2004: ~/linux
ubuntu@ubuntu2004:~/linux/e$ ls
aboutLinux.txt  legal  macOS.txt  windows.txt
ubuntu@ubuntu2004:~/linux/e$ mv macOS.txt ..
ubuntu@ubuntu2004:~/linux/e$ ls
aboutLinux.txt  legal  windows.txt
ubuntu@ubuntu2004:~/linux/e$ cd ..
ubuntu@ubuntu2004:~/linux$ ls
e  macOS.txt
ubuntu@ubuntu2004:~/linux$
```

Figura 20 – Demonstração do comando mv

6 Compilação do kernel Linux

A fim de iniciar o procedimento de compilação do kernel em si, preparou-se o ambiente utilizando um software de virtualização², como na primeira parte da atividade. Entretanto, utilizou-se um computador rodando macOS com um processador Apple Silicon (M1) de arquitetura ARM, por isso a opção por um software de virtualização diferente, já que o VirtualBox apenas é compatível com arquitetura x86.

O Parallels Desktop permite nativamente baixar imagens de distribuições Linux compatíveis com arquitetura ARM. Dentre as possíveis opções, decidiu-se por continuar trabalhando com o Ubuntu 20.04 LTS. É possível observar a configuração prévia da máquina virtual na Figura 21.

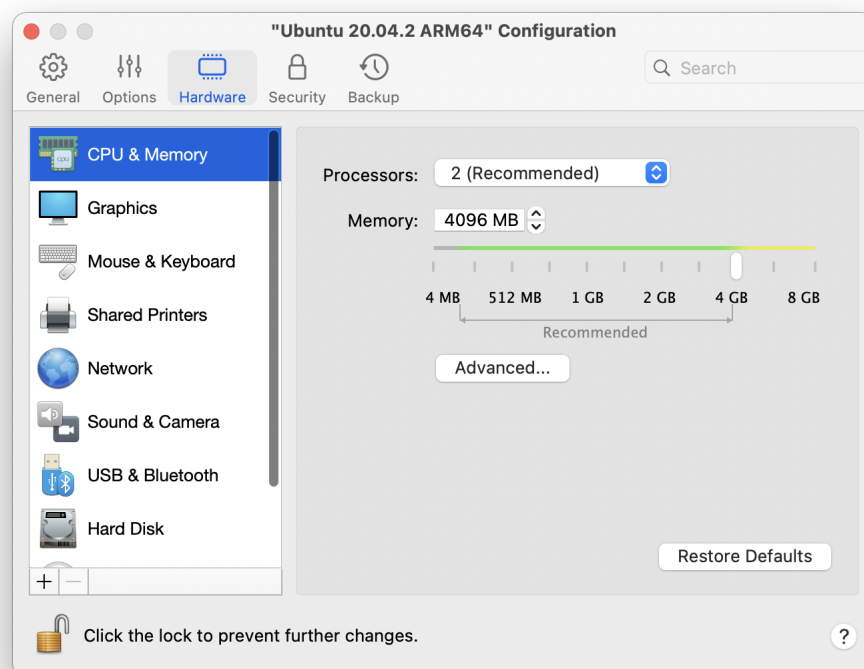


Figura 21 – Configuração da máquina virtual

Uma vez tendo a máquina virtual iniciada e usuário criado (com privilégio de execução em *root*) realizou-se as atualizações básicas necessárias além de um editor de texto de preferência.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt install vim
```

² <https://www.parallels.com>

Tais comandos foram sucedidos pela instalação de dependências e ferramentas necessárias para a compilação do kernel.

```
sudo su
apt-get install build-essential libncurses5-dev xz-utils
libssl-dev bc bison flex libelf-dev libncurses5-dev
```

O passo seguinte se deu por baixar e extrair o arquivo *tarball* com a versão mais recente do kernel Linux, 5.16.14. Para facilitar a configuração da compilação, fez-se uma cópia da configuração (atual) pré-compilação, com o comando demonstrado abaixo.

```
cp -v /boot/config-$(uname -r) .config
```

Para configurar o kernel com as opções específicas da respectiva arquitetura e módulos, optou-se pelo setup padrão oferecido via interface do sistema com o comando **menuconfig**. Os comandos seguintes têm a função de preparar e compilar os módulos necessários para, na sequência, compilar o kernel. São eles:

```
make -j2
make modules
make modules_install
make
```

O parâmetro `-j2` leva em conta o número de núcleos previamente configurados para a máquina virtual. Nesse caso foram dois núcleos. Ao executar o comando `make modules`, houve a necessidade de alterar uma configuração específica: `CONFIG_SYSTEM_TRUSTED_KEYS`. Anteriormente à execução dos próximos comandos, detectou-se a ausência de duas dependências. Utilizando o comando `apt-get install`, foram instalados os pacotes `flex` e `pahole`.

Para concluir o processo, utilizou-se o comando `TODO make install` para a compilação e o comando `sudo update-grub` seguido do comando `sudo reboot`. Os dois últimos executados fora do ambiente de root. Como resultado final da compilação, pode-se observar a Figura 22.

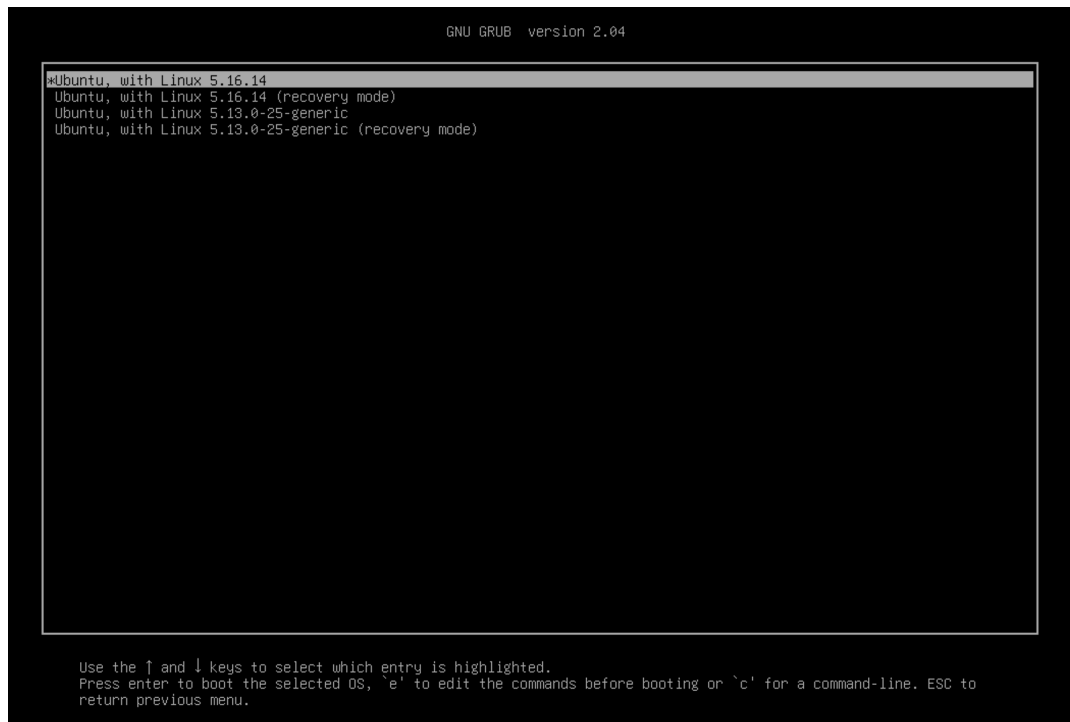


Figura 22 – Configuração da máquina virtual

7 Conclusões

Após realizados os procedimentos solicitados na descrição da atividade do laboratório 1, finalizou-se a compilação do kernel Linux em sua versão estável 5.16.14.

8 Referências

- [HOWTO: Compile Linux kernel \(on Ubuntu, applies to any distro\)](#)
- [Aprenda definitivamente como compilar o kernel do linux](#)
- [Fazer atualizações de segurança. para Debian e Ubuntu](#)
- [Network interface configuration for ifup and ifdown](#)