



UTFPR – Universidade Tecnológica Federal do Paraná – *Campus Campo Mourão*
DACOM – Departamento de Computação

COCIC – Coordenação de Ciência da Computação

Curso: Ciência da Computação

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

SISTEMAS OPERACIONAIS

Laboratório 01

INSTALAÇÃO E COMPILAÇÃO

DO KERNEL LINUX

Reginaldo Gregório de Souza Neto 2252813

Marcos Bezner Rampaso 2149435

Carolina Yumi Fujii 2335468

Universidade Tecnológica Federal do Paraná – UTFPR

Bacharelado em Ciência da Computação – BCC

Campo Mourão - Agosto / 2022

INSTALAÇÃO E COMPILAÇÃO DO KERNEL LINUX

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.

Resumo

O sistema operacional de um computador é uma camada de software que faz a interface entre o hardware, o usuário e os programas utilitários. Neste relatório iremos abordar a instalação e configuração do sistema operacional Debian (Linux), que é um Sistema Operacional de código aberto onde todos podem contribuir para a melhoria do mesmo. Além de compilar o Kernel (núcleo) Linux em sua versão mais recente estável.

Palavras-chave: Linux. Kernel. Sistema Operacional. Instalação. Configuração.

1 Introdução: Configurações de Hardware e testes iniciais.

O hardware utilizado para a execução desta atividade se trata de um computador desktop com:

Processador: AMD Ryzen 3 3200G com Radeon Vega Graphics 3.60 GHz

RAM instalada: 8,00 GB (utilizável: 5,95 GB)

Tipo de sistema: Sistema operacional Windows 10 de 64 bits, processador baseado em x64.

A máquina virtual utilizada foi a Oracle Content Management VirtualBox, e a distribuição executada foi a Debian na versão 11.4.0, com 2GB de memória RAM e 26.8GB de espaço de armazenamento interno dinamicamente alocado. Assim que foi inicializado o VirtualBox apresentou um problema de inicialização da máquina virtual, retratado na figura a seguir:

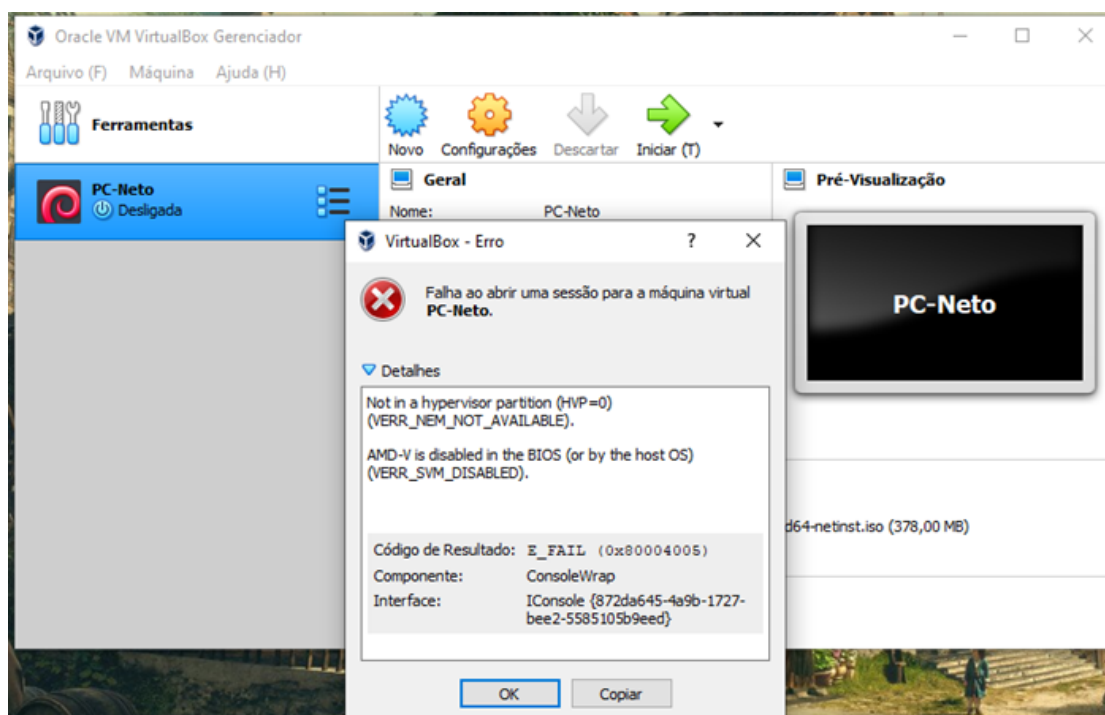


Figura 1 – Erro de sessão com a máquina virtual.

Para solucionar esse problema foi preciso ativar dois recursos do Windows para possibilitar a inicialização da VM. “Plataforma de Máquina Virtual” e “Plataforma de Hipervisor do Windows”.

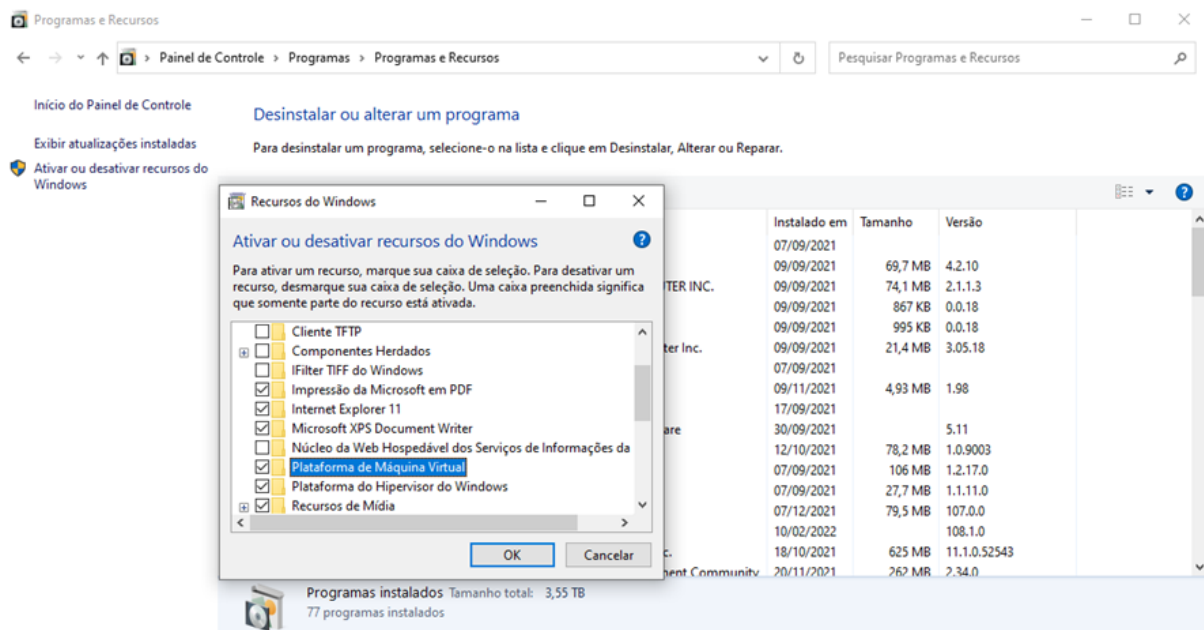


Figura 2 – Ativação de recursos do Windows.

Reiniciando a máquina, a BIOS teve de ser acessada para ativar a opção de máquina virtual no processador “.



Figura 3 – Ativando a máquina virtual através da BIOS.

Após a instalação do VirtualBox, configuramos o ambiente para o Debian (Versão 11 de 64 bits), e adicionamos a imagem .ISO. Desse modo, foi possível

iniciar a configuração de instalação do Sistema Operacional escolhido (debian) de modo “não-gráfico”, ou seja, através de janelas de texto.

Os passos foram respectivamente: Após seguir as instruções de criação de usuário, chegamos a parte de particionamento de disco, nesta aba, selecionamos o modo manual de particionamento (figura 4).

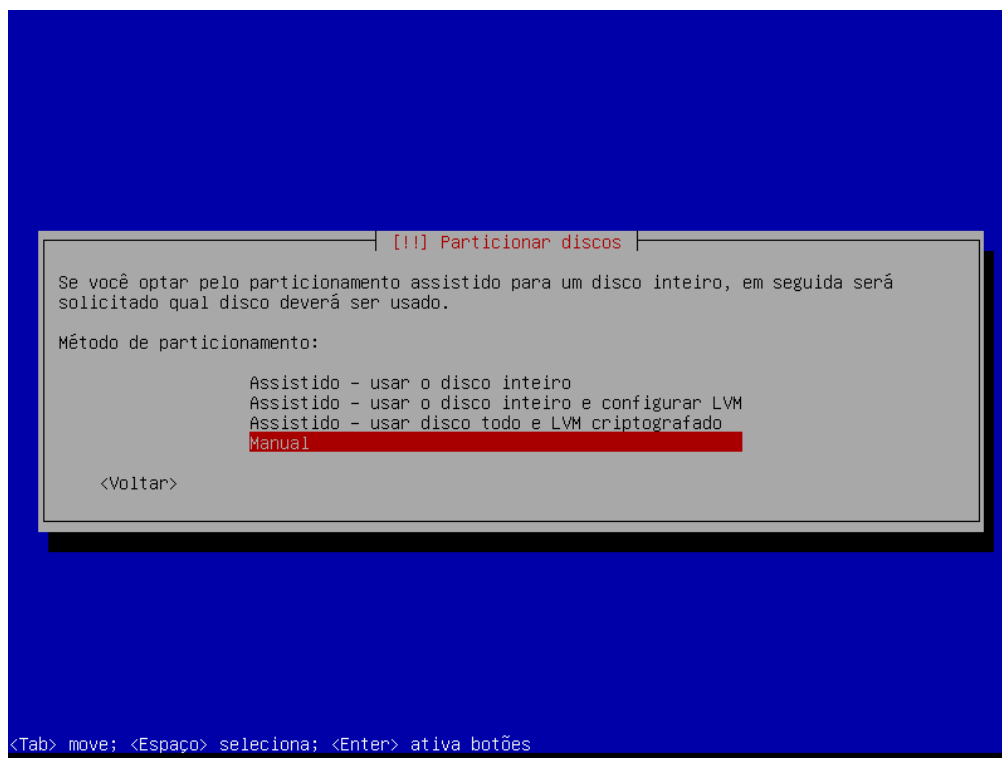


Figura 4 - Escolha do método de particionamento.

Após a seleção manual, as partições foram selecionadas com base em valores que executariam o Debian sem problemas de falta de espaço. Tais seleções nas partições foram (imagem 6):

- Unidade primária de armazenamento, contendo /root ou / (999.3MB);
- Unidade lógica de armazenamento, contendo o /home (2GB);
- Unidade lógica de armazenamento, contendo o /usr (16.0GB);
- Unidade lógica de armazenamento, contendo o swap (999.3MB);
- Unidade lógica de armazenamento, contendo o /var (2.0GB);
- Unidade primária ou lógica de armazenamento, contendo espaço livre (4.8GB)

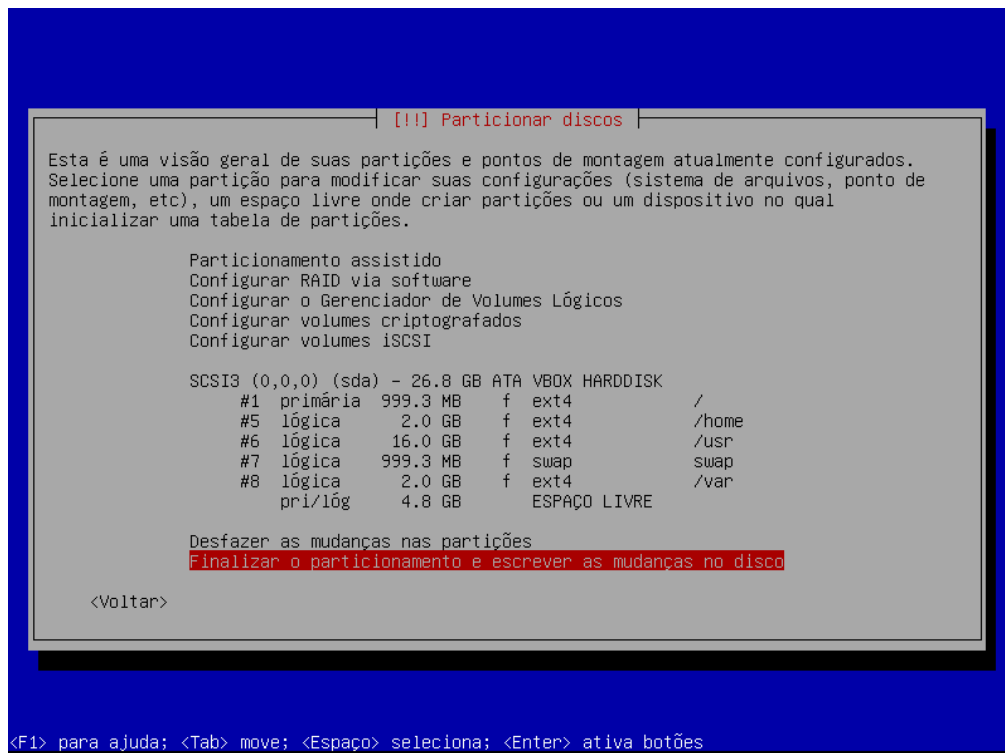


Figura 5 - Escolhas das partições.

Em seguida, foi solicitado o país de instalação para que fosse possível realizar, caso houvesse, uma atualização de arquivos que seriam buscados na nuvem. Deste modo, o país escolhido foi “Brasil” e o servidor: *ftp.br.debian.org/debian*.

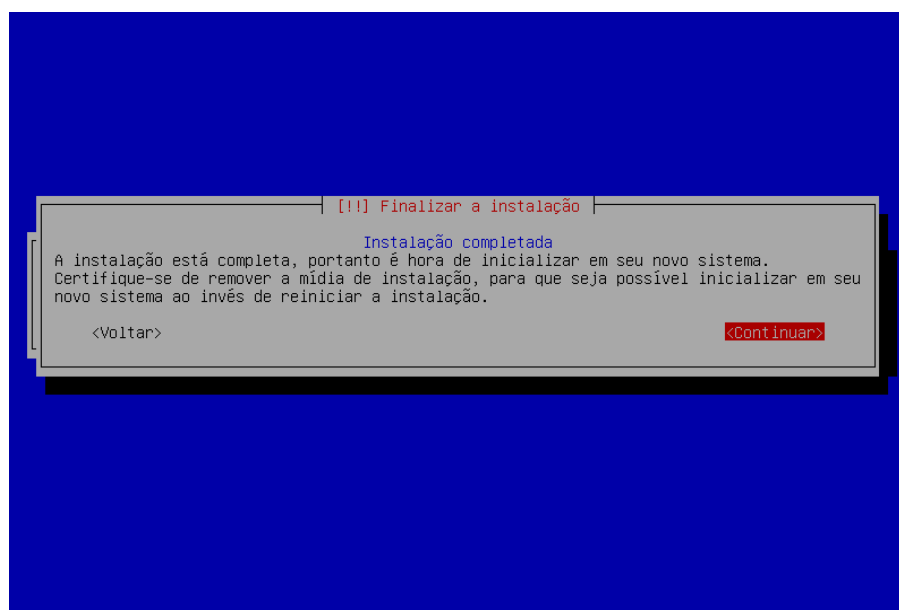


Figura 6 - Mensagem de "Instalação completa".

Como dito anteriormente, a criação do usuário padrão e do superusuário foi realizada diretamente no processo de instalação. Portanto, é possível observar que o acesso ao SO é feito através do usuário “neto”. E na figura 12 pode-se analisar a necessidade de realizar o acesso ao superusuário.

Sendo feita a instalação e inicialização do sistema, abrimos o terminal e executamos alguns comandos. O primeiro a ser executado foi o comando: `ps aux` (figura 7). Através da listagem do “`ps aux`”, é possível notar que cada processo possui:

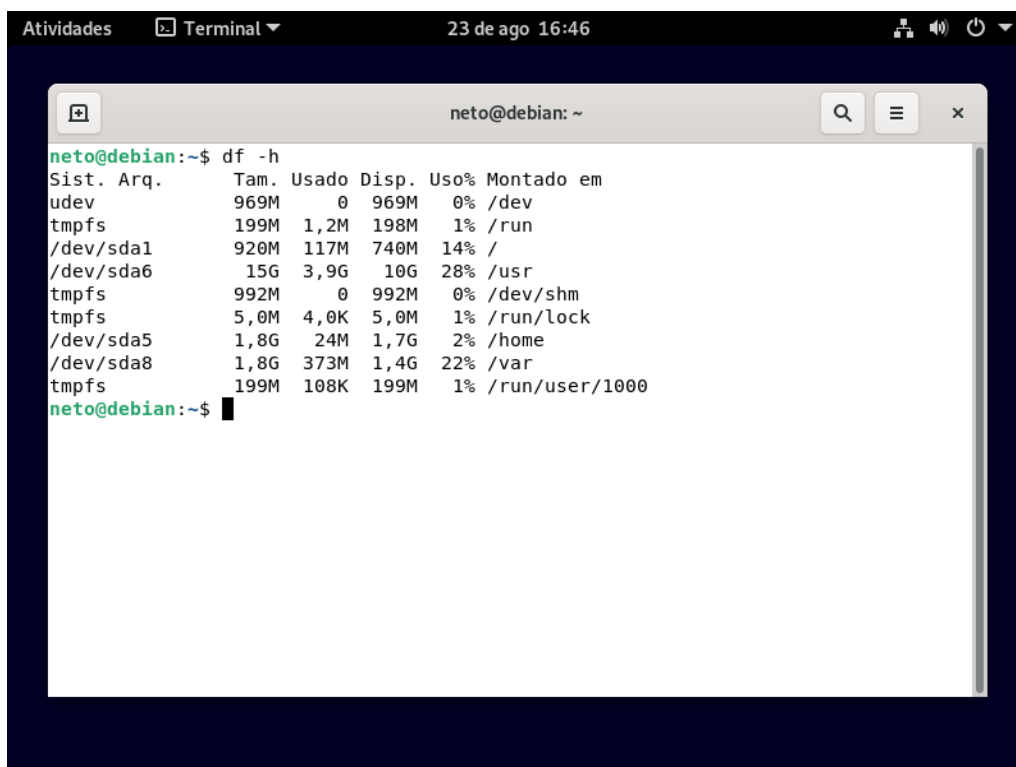
- inicializador (root ou USER);
- PID (Identificação do processo);
- A porcentagem de uso da CPU e da memória RAM (%CPU, %RAM);
- VSZ (Tamanho da memória virtual);
- RSS (porção da memória que está ocupada por um processo, mantido na memória RAM);
- STAT (estados de processo);
- O horário de início e o tempo de uso da CPU (Start, Time);
- Command (o comando utilizado para executar o processo).

```

neto@debian:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root             1  0.0  0.5 163996 10452 ?        Ss   15:49   0:03 /sbin/init
root             2  0.0  0.0      0     0 ?        S    15:49   0:00 [kthreadd]
root             3  0.0  0.0      0     0 ?        I<   15:49   0:00 [rcu_gp]
root             4  0.0  0.0      0     0 ?        I<   15:49   0:00 [rcu_par_gp]
root             6  0.0  0.0      0     0 ?        I<   15:49   0:00 [kworker/0:0H]
root             8  0.0  0.0      0     0 ?        I<   15:49   0:02 [kworker/0:1H]
root             9  0.0  0.0      0     0 ?        I<   15:49   0:00 [mm_percpu_wq]
root            10  0.0  0.0      0     0 ?        S    15:49   0:00 [rcu_tasks_ru]
root            11  0.0  0.0      0     0 ?        S    15:49   0:00 [rcu_tasks_tr]
root            12  0.0  0.0      0     0 ?        S    15:49   0:00 [ksoftirqd/0]
root            13  0.0  0.0      0     0 ?        I    15:49   0:00 [rcu_sched]
root            14  0.0  0.0      0     0 ?        S    15:49   0:00 [migration/0]
root            16  0.0  0.0      0     0 ?        S    15:49   0:00 [cpuphp/0]
root            18  0.0  0.0      0     0 ?        S    15:49   0:00 [kdevtmpfs]
root            19  0.0  0.0      0     0 ?        I<   15:49   0:00 [netns]
root            20  0.0  0.0      0     0 ?        S    15:49   0:00 [kauditd]
root            21  0.0  0.0      0     0 ?        S    15:49   0:00 [khungtaskd]
root            22  0.0  0.0      0     0 ?        S    15:49   0:00 [oom_reaper]
root            23  0.0  0.0      0     0 ?        I<   15:49   0:00 [writeback]
root            24  0.0  0.0      0     0 ?        S    15:49   0:00 [kcompactd0]
root            25  0.0  0.0      0     0 ?        SN   15:49   0:00 [ksmd]
root            26  0.0  0.0      0     0 ?        SN   15:49   0:02 [khugepaged]
  
```

Figura 7 - Verificação da listagem de processos em execução “`ps aux`”.

Em seguida, o comando “`df -h`” foi executado, tornando possível a visualização das pastas que foram criadas no particionamento do disco, juntamente com algumas pastas novas (figura 8). Com seus respectivos espaços de armazenamento e porcentagem de uso.



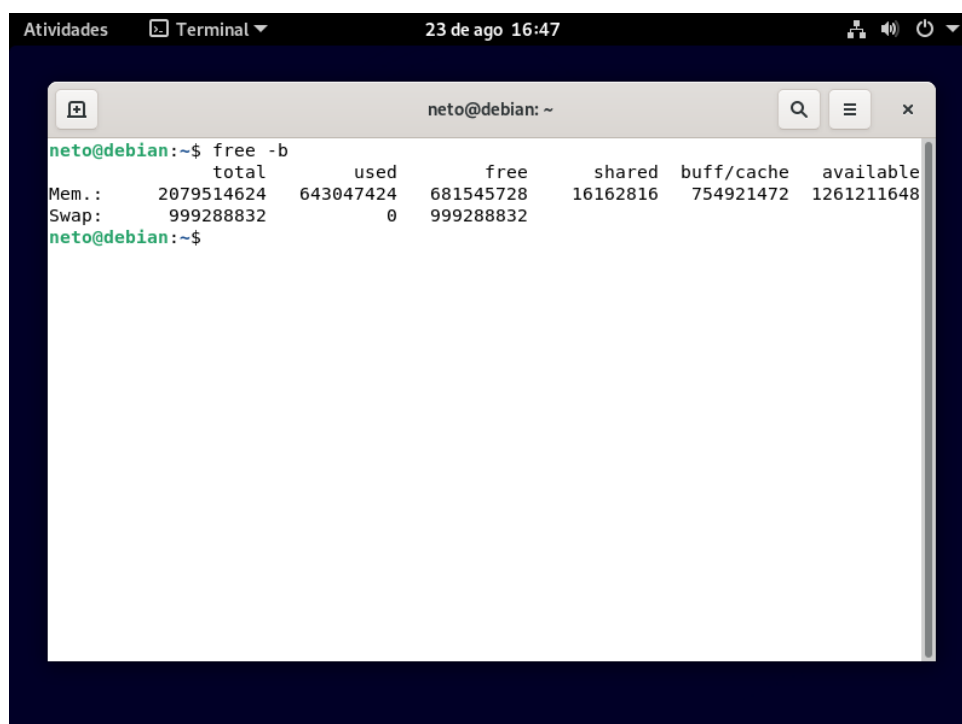
```

neto@debian:~$ df -h
Sist. Arq.      Tam. Usado Disp.  Uso% Montado em
udev           969M    0  969M   0% /dev
tmpfs          199M  1,2M  198M   1% /run
/dev/sda1      920M  117M  740M  14% /
/dev/sda6      15G   3,9G   10G  28% /usr
tmpfs          992M    0  992M   0% /dev/shm
tmpfs          5,0M   4,0K   5,0M   1% /run/lock
/dev/sda5      1,8G   24M   1,7G   2% /home
/dev/sda8      1,8G  373M   1,4G  22% /var
tmpfs          199M  108K   199M   1% /run/user/1000
neto@debian:~$

```

Figura 8 - Espaço da "df -h"

O comando `free -b` evidencia o quanto de memória o computador possui livre e a memória utilizada, também nos mostra a swap, a memória compartilhada e os buffers utilizados no kernel do sistema (figura 9).



```

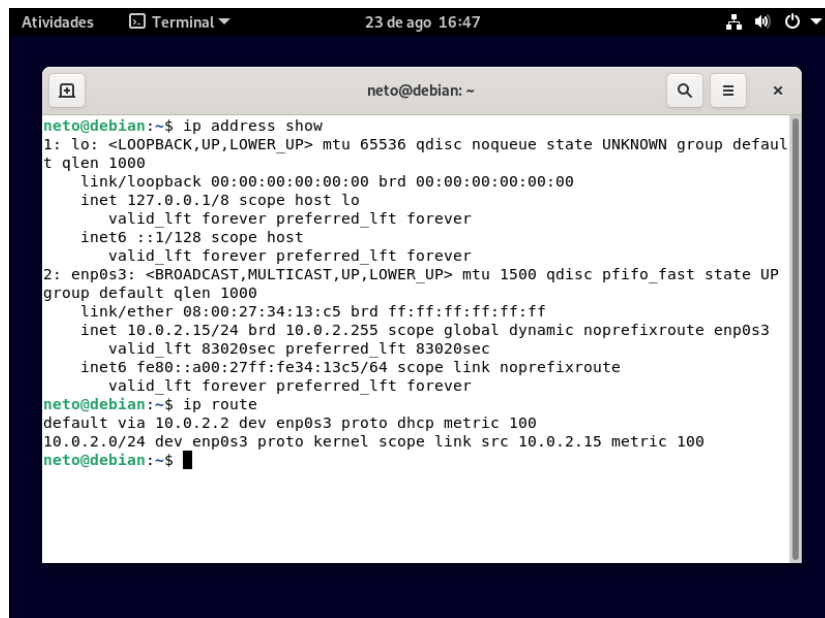
neto@debian:~$ free -b
              total        used         free       shared    buff/cache   available
Mem.:    2079514624    643047424    681545728    16162816    754921472    1261211648
Swap:      999288832           0    999288832
neto@debian:~$

```

Figura 9 - Memória disponível no "free -b".

Após o comando `free -b`, o grupo executou uma série de comandos relacionados à rede, sendo eles respectivamente:

- `ip address show`, o comando mostra o endereço de seu IP na rede (figura 10);
- `ip route`, retorna a rota de ip da máquina (figura 10);
- `cat /etc/resolv.conf`.

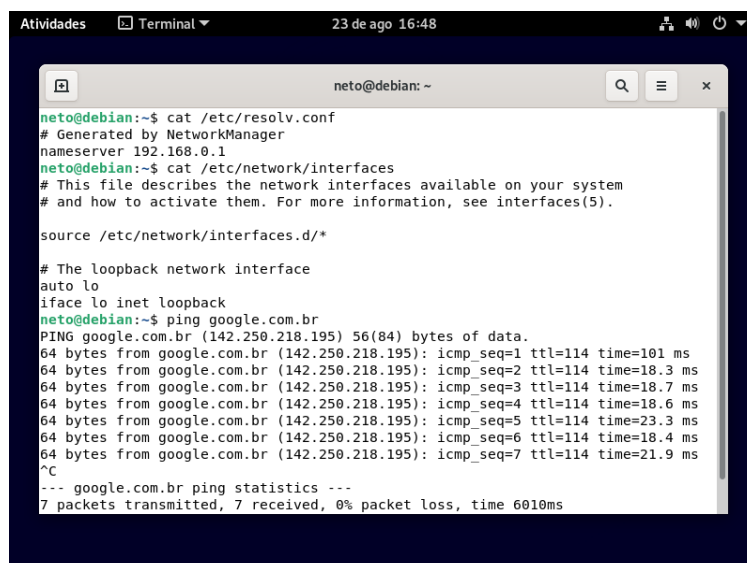


```

neto@debian:~$ 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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:34:13:c5 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 83020sec preferred_lft 83020sec
    inet6 fe80::a00:27ff:fe34:13c5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
neto@debian:~$ 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
neto@debian:~$
  
```

Figura 10 - Verificação da rede "`ip address show`", e "`ip route`".

Podemos analisar que mesmo se tratando de uma máquina virtual ela possui as duas interfaces de rede que são comumente encontradas em máquinas reais, ambas com endereços MAC e IP.



```

neto@debian:~$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.0.1
neto@debian:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

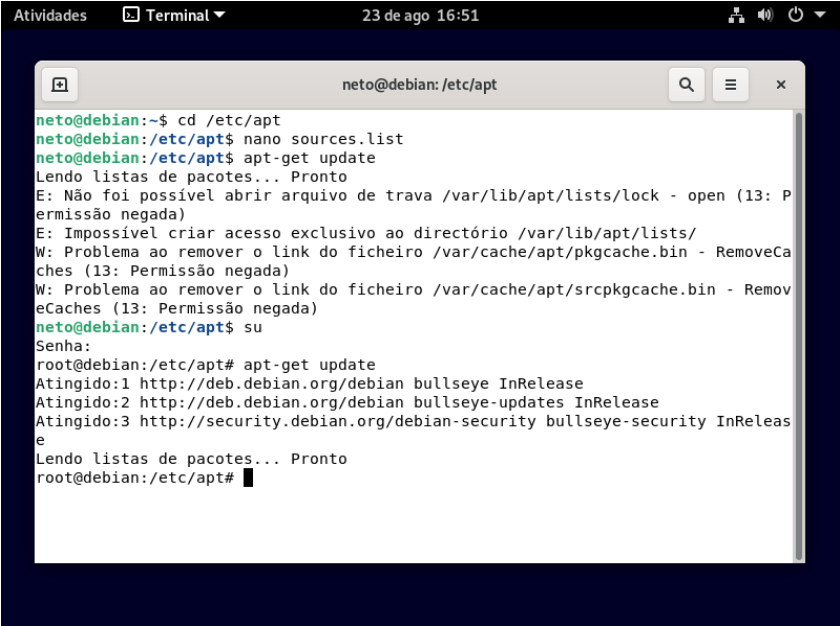
# The loopback network interface
auto lo
iface lo inet loopback
neto@debian:~$ ping google.com.br
PING google.com.br (142.250.218.195) 56(84) bytes of data:
64 bytes from google.com.br (142.250.218.195): icmp_seq=1 ttl=114 time=101 ms
64 bytes from google.com.br (142.250.218.195): icmp_seq=2 ttl=114 time=18.3 ms
64 bytes from google.com.br (142.250.218.195): icmp_seq=3 ttl=114 time=18.7 ms
64 bytes from google.com.br (142.250.218.195): icmp_seq=4 ttl=114 time=18.6 ms
64 bytes from google.com.br (142.250.218.195): icmp_seq=5 ttl=114 time=23.3 ms
64 bytes from google.com.br (142.250.218.195): icmp_seq=6 ttl=114 time=18.4 ms
64 bytes from google.com.br (142.250.218.195): icmp_seq=7 ttl=114 time=21.9 ms
^C
--- google.com.br ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
  
```

Figura 11 - Verificação da rede "`cat /etc/resolv.conf`" e "`cat /etc/network/interfaces`".

Os comandos `cat` realizados acima, exibem o servidor de DNS ("nameserver") e as interfaces de redes presentes no dispositivo virtual

respectivamente. Deste modo, constatando que está tudo em ordem, podemos realizar o comando de ping diretamente no endereço “google.com.br”.

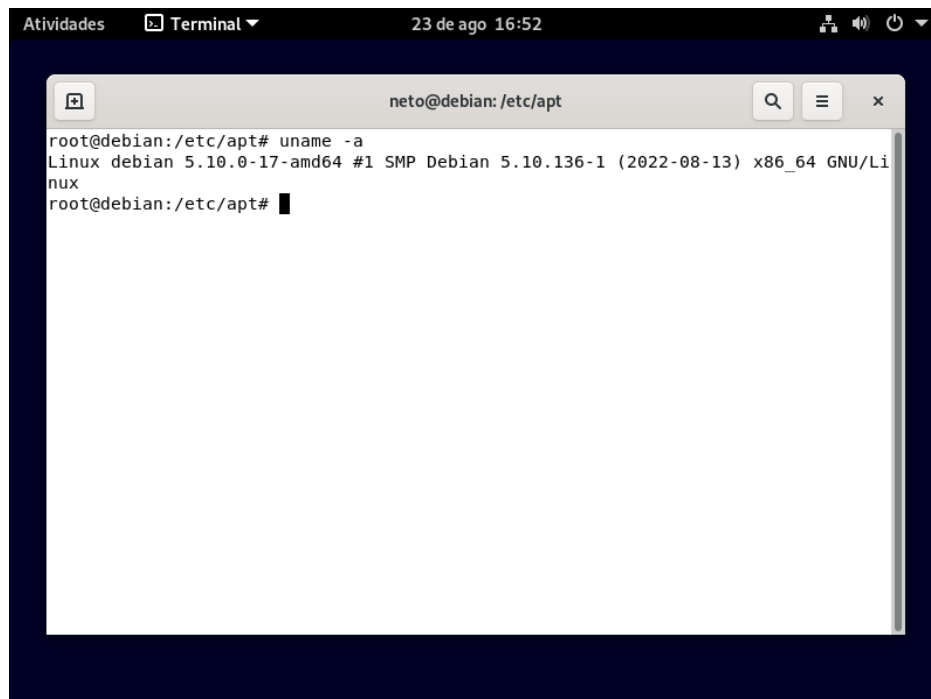
Após executados os comandos de rede, foram testados os comandos que adicionaram o repositório de atualização de segurança (cd /etc/apt, nano sources.list e apt-get update).



```
neto@debian:~$ cd /etc/apt
neto@debian:/etc/apt$ nano sources.list
neto@debian:/etc/apt$ apt-get update
Lendo listas de pacotes... Pronto
E: Não foi possível abrir arquivo de trava /var/lib/apt/lists/lock - open (13: Permissão negada)
E: Impossível criar acesso exclusivo ao directório /var/lib/apt/lists/
W: Problema ao remover o link do ficheiro /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permissão negada)
W: Problema ao remover o link do ficheiro /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permissão negada)
neto@debian:/etc/apt$ su
Senha:
root@debian:/etc/apt# apt-get update
Atingido:1 http://deb.debian.org/debian bullseye InRelease
Atingido:2 http://deb.debian.org/debian bullseye-updates InRelease
Atingido:3 http://security.debian.org/debian-security bullseye-security InRelease
Lendo listas de pacotes... Pronto
root@debian:/etc/apt#
```

Figura 12 - Adicionado os repositórios de atualização de segurança "cd /etc/apt", "nano sources.list" e "apt-get update".

A próxima etapa do experimento foi manusear mais intimamente os arquivos do sistema operacional através do terminal, começamos com o simples comando “uname -a”, que nos retorna o nome e a versão do Kernel atual (figura 13). O segundo comando (ls - l) lista os conteúdos do directório que está sendo acessado (figura 14).

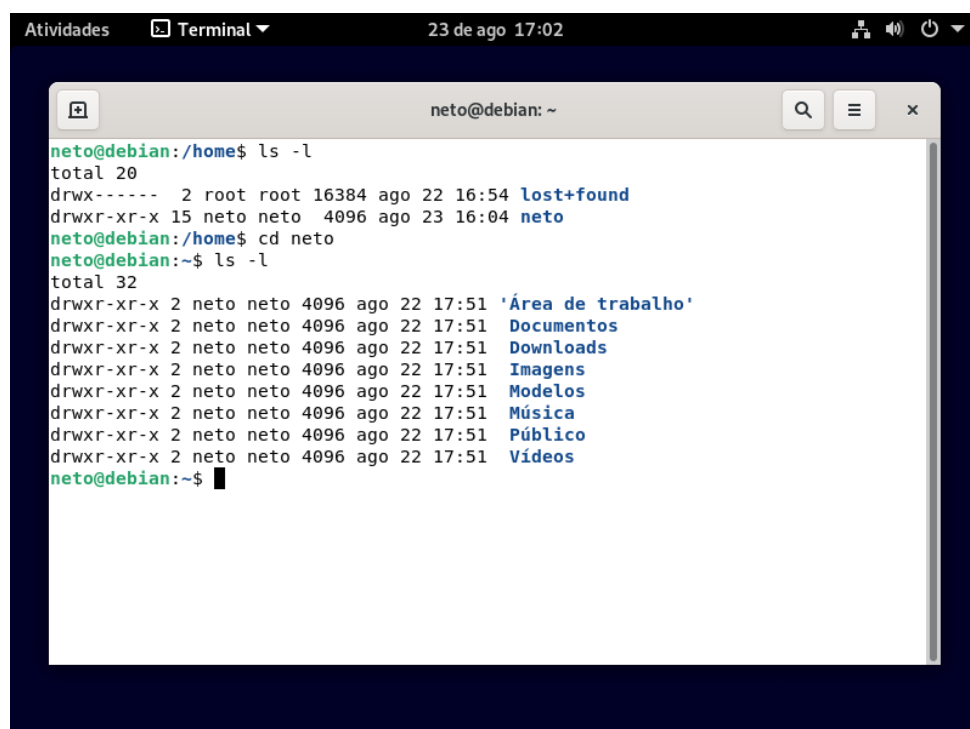


A terminal window titled 'neto@debian: /etc/apt' is shown. The user is root. The command 'uname -a' has been executed, and the output is displayed. The output shows the system is a Linux Debian 5.10.0-17-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64 GNU/Linux.

```
root@debian:/etc/apt# uname -a
Linux debian 5.10.0-17-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64 GNU/Linux
root@debian:/etc/apt#
```

Figura 13 - Verificação do kernel atual: *uname -a*

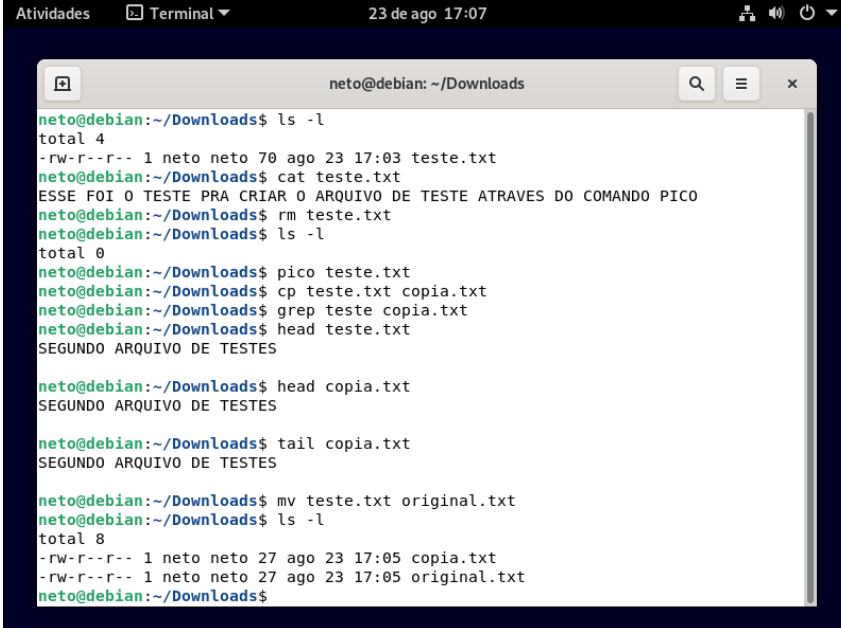
Em seguida foram realizados alguns comandos básicos de navegação em diretórios e listagem de arquivos.



A terminal window titled 'neto@debian: ~' is shown. The user is neto. The command 'ls -l' is executed in the home directory, showing the output. Then, the user changes to the 'neto' directory and runs 'ls -l' again, showing the output of the files in that directory.

```
neto@debian:/home$ ls -l
total 20
drwx----- 2 root root 16384 ago 22 16:54 lost+found
drwxr-xr-x 15 neto neto 4096 ago 23 16:04 neto
neto@debian:/home$ cd neto
neto@debian:~$ ls -l
total 32
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 'Área de trabalho'
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Documentos
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Downloads
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Imagens
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Modelos
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Música
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Público
drwxr-xr-x 2 neto neto 4096 ago 22 17:51 Vídeos
neto@debian:~$
```

Figura 14 - Teste do comando "ls -l".




```

neto@debian: ~/Downloads
neto@debian:~/Downloads$ ls -l
total 4
-rw-r--r-- 1 neto neto 70 ago 23 17:03 teste.txt
neto@debian:~/Downloads$ cat teste.txt
ESSE FOI O TESTE PRA CRIAR O ARQUIVO DE TESTE ATRAVES DO COMANDO PICO
neto@debian:~/Downloads$ rm teste.txt
neto@debian:~/Downloads$ ls -l
total 0
neto@debian:~/Downloads$ pico teste.txt
neto@debian:~/Downloads$ cp teste.txt copia.txt
neto@debian:~/Downloads$ grep teste copia.txt
neto@debian:~/Downloads$ head teste.txt
SEGUNDO ARQUIVO DE TESTES
neto@debian:~/Downloads$ head copia.txt
SEGUNDO ARQUIVO DE TESTES
neto@debian:~/Downloads$ tail copia.txt
SEGUNDO ARQUIVO DE TESTES
neto@debian:~/Downloads$ mv teste.txt original.txt
neto@debian:~/Downloads$ ls -l
total 8
-rw-r--r-- 1 neto neto 27 ago 23 17:05 copia.txt
-rw-r--r-- 1 neto neto 27 ago 23 17:05 original.txt
neto@debian:~/Downloads$

```

Figura 15 - Teste dos comandos de manipulação de arquivos.

Próximo ao fim de nosso experimento, é chegada a hora de compilar o kernel linux em nossa VM. Para tal, acessamos o terminal com o comando “sudo” a fim de tornar o usuário um super usuário, para com isso poder acessar a root. Também foi feita a instalação do kernel em sua versão mais recente estável. Em seguida foi feita a descompactação dos arquivos e por fim a configuração e compilação do kernel em si.



```

neto@debian: /
Configurando g++-10 (10.2.1-6) ...
Configurando gcc (4:10.2.1-1) ...
Configurando libelf-dev:amd64 (0.183-1) ...
Configurando g++ (4:10.2.1-1) ...
update-alternatives: a usar /usr/bin/g++ para disponibilizar /usr/bin/c++ (c++)
em modo auto
Configurando build-essential (12.9) ...
A processar 'triggers' para man-db (2.9.4-2) ...
A processar 'triggers' para libc-bin (2.31-13+deb11u3) ...
root@debian:~# wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.3.ta
r.xz
--2022-08-23 17:12:50-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19
.3.tar.xz
Resolvendo cdn.kernel.org (cdn.kernel.org)... 199.232.113.176, 2a04:4e42:5c::432
Conectando-se a cdn.kernel.org (cdn.kernel.org)[199.232.113.176]:443... conectad
o.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 131620076 (126M) [application/x-xz]
Salvando em: "linux-5.19.3.tar.xz"

linux-5.19.3.tar.xz 100%[=====] 125,52M 10,7MB/s em 14s
2022-08-23 17:13:04 (9,27 MB/s) - "linux-5.19.3.tar.xz" salvo [131620076/1316200
76]
root@debian:~#

```

Figura 16 - Configuração e compilação da última versão estável do kernel, utilizando o comando wget.

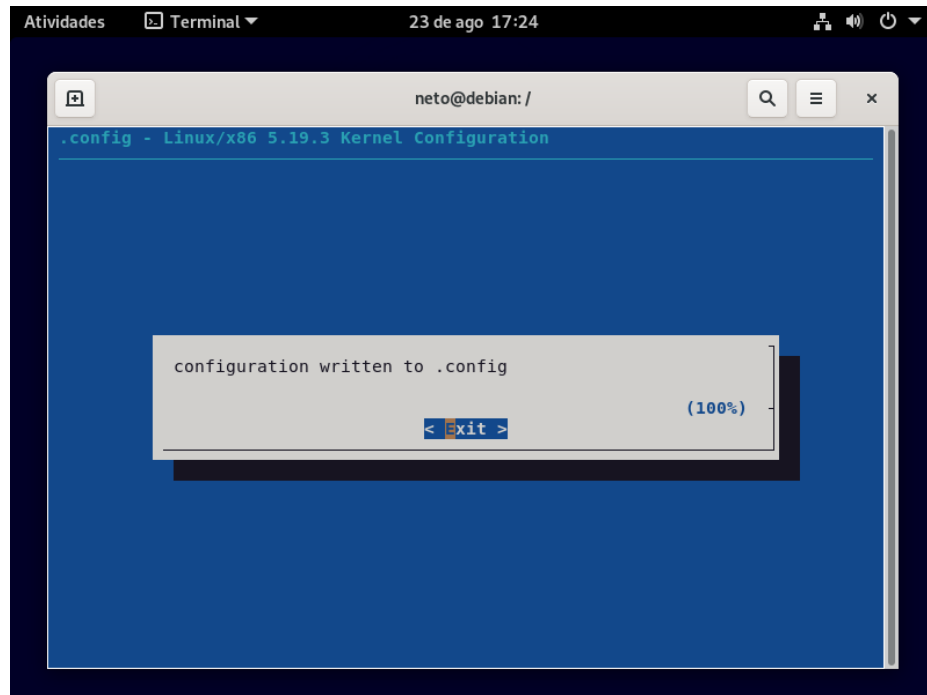


Figura 17 - Tela de conclusão da configuração do kernel.

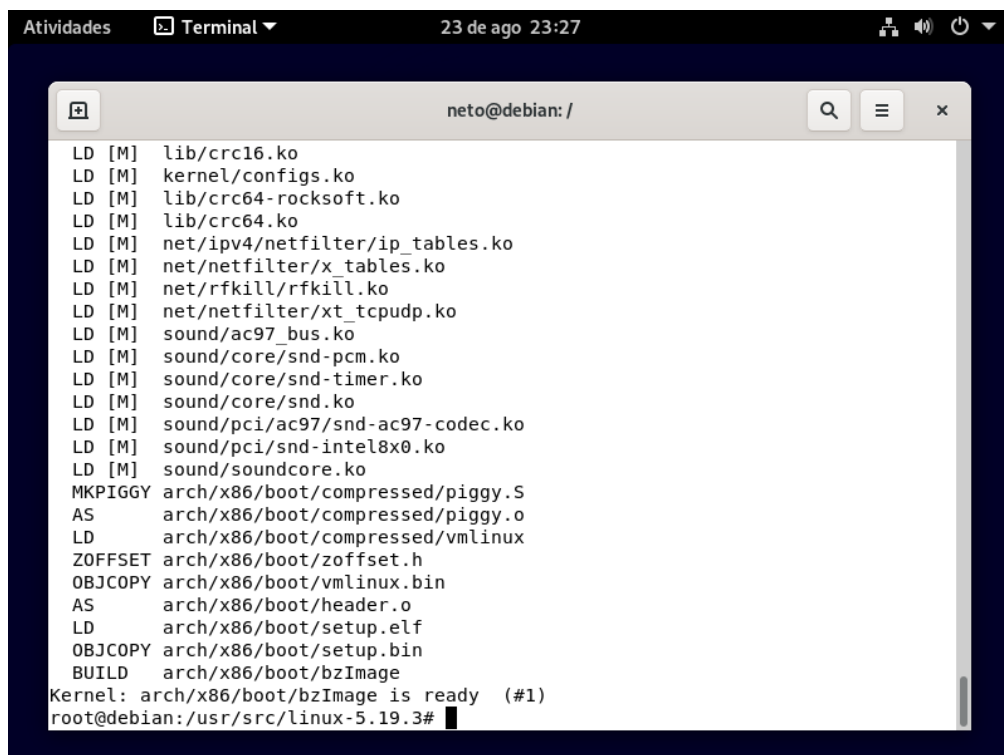
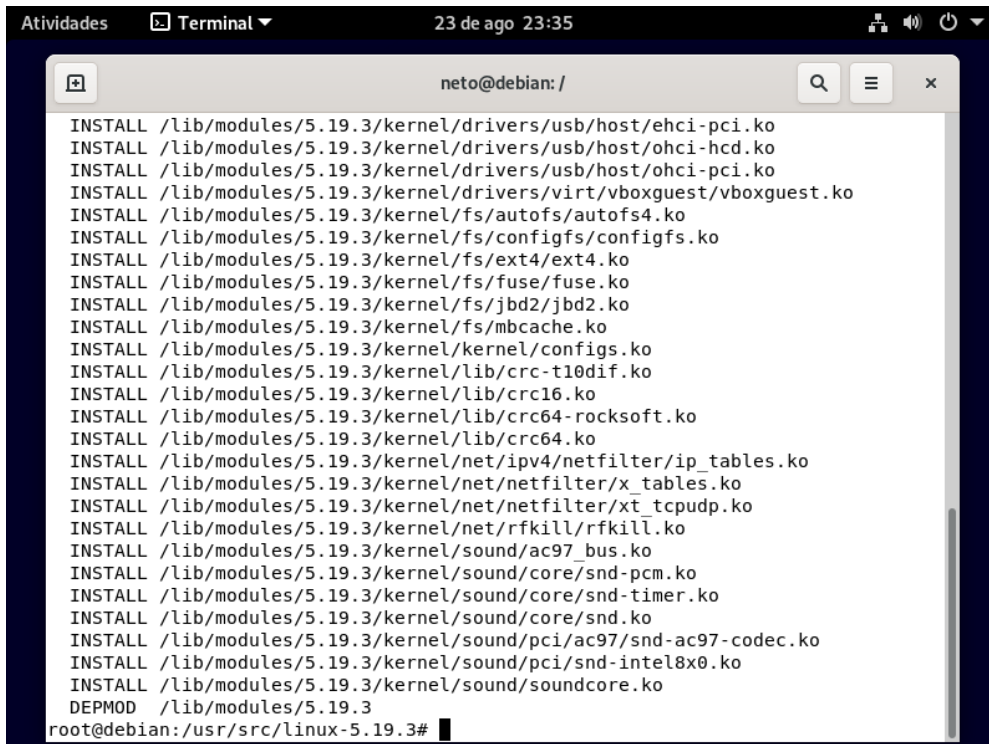


Figura 18 - Configuração e compilação da última versão estável do kernel.

Após configuradas as preferências, os arquivos foram descompactados com o comando: `tar xvf linux-5.19.3 .tar.xz -C /usr/src`. Feita a descompactação, o diretório linux da pasta src teve de ser acessado com o comando: `cd /usr/src/linux-5.19.3`. Ao final, o comando `make -j4` foi executado para compilar o kernel, seguidos pelos comandos: `make modules_install` e `make install`.

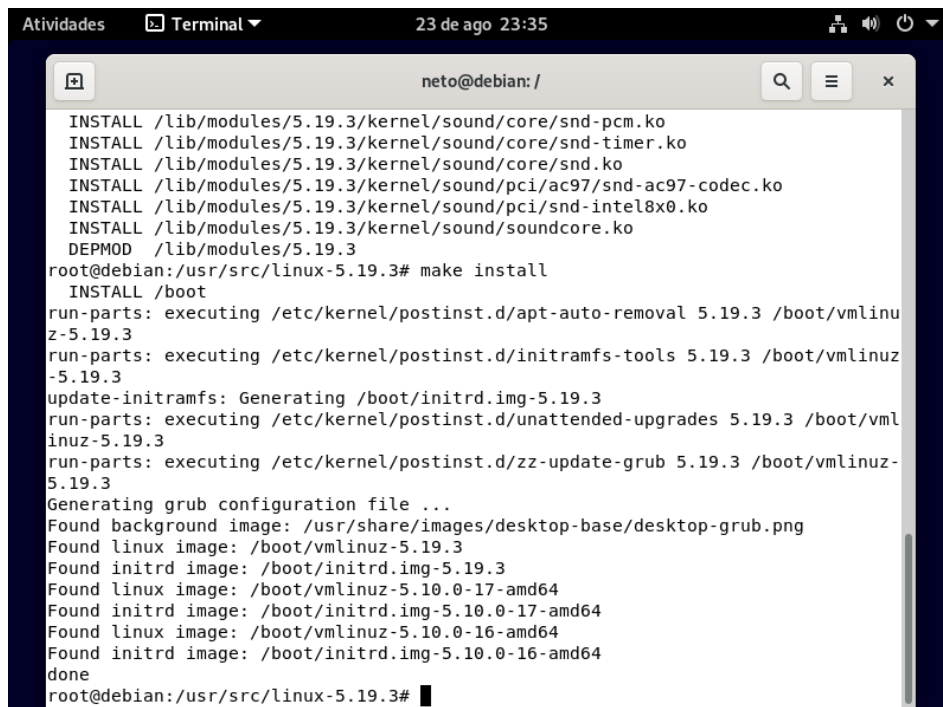


```

Atividades Terminal 23 de ago 23:35
neto@debian: /
INSTALL /lib/modules/5.19.3/kernel/drivers/usb/host/ehci-pci.ko
INSTALL /lib/modules/5.19.3/kernel/drivers/usb/host/ohci-hcd.ko
INSTALL /lib/modules/5.19.3/kernel/drivers/usb/host/ohci-pci.ko
INSTALL /lib/modules/5.19.3/kernel/drivers/virt/vboxguest/vboxguest.ko
INSTALL /lib/modules/5.19.3/kernel/fs/autofs/autofs4.ko
INSTALL /lib/modules/5.19.3/kernel/fs/configfs/configfs.ko
INSTALL /lib/modules/5.19.3/kernel/fs/ext4/ext4.ko
INSTALL /lib/modules/5.19.3/kernel/fs/fuse/fuse.ko
INSTALL /lib/modules/5.19.3/kernel/fs/jbd2/jbd2.ko
INSTALL /lib/modules/5.19.3/kernel/fs/mbcache.ko
INSTALL /lib/modules/5.19.3/kernel/kernel/configs.ko
INSTALL /lib/modules/5.19.3/kernel/lib/crc-t10dif.ko
INSTALL /lib/modules/5.19.3/kernel/lib/crc16.ko
INSTALL /lib/modules/5.19.3/kernel/lib/crc64-rocksoft.ko
INSTALL /lib/modules/5.19.3/kernel/lib/crc64.ko
INSTALL /lib/modules/5.19.3/kernel/net/ipv4/netfilter/ip_tables.ko
INSTALL /lib/modules/5.19.3/kernel/net/netfilter/x_tables.ko
INSTALL /lib/modules/5.19.3/kernel/net/netfilter/xt_tcpudp.ko
INSTALL /lib/modules/5.19.3/kernel/net/rfkill/rfkill.ko
INSTALL /lib/modules/5.19.3/kernel/sound/ac97_bus.ko
INSTALL /lib/modules/5.19.3/kernel/sound/core/snd-pcm.ko
INSTALL /lib/modules/5.19.3/kernel/sound/core/snd-timer.ko
INSTALL /lib/modules/5.19.3/kernel/sound/core/snd.ko
INSTALL /lib/modules/5.19.3/kernel/sound/pci/ac97/snd-ac97-codec.ko
INSTALL /lib/modules/5.19.3/kernel/sound/pci/snd-intel8x0.ko
INSTALL /lib/modules/5.19.3/kernel/sound/soundcore.ko
DEPMOD /lib/modules/5.19.3
root@debian:/usr/src/linux-5.19.3#

```

Figura 19 - Instalação da última versão estável do kernel do Linux a partir do código-fonte.



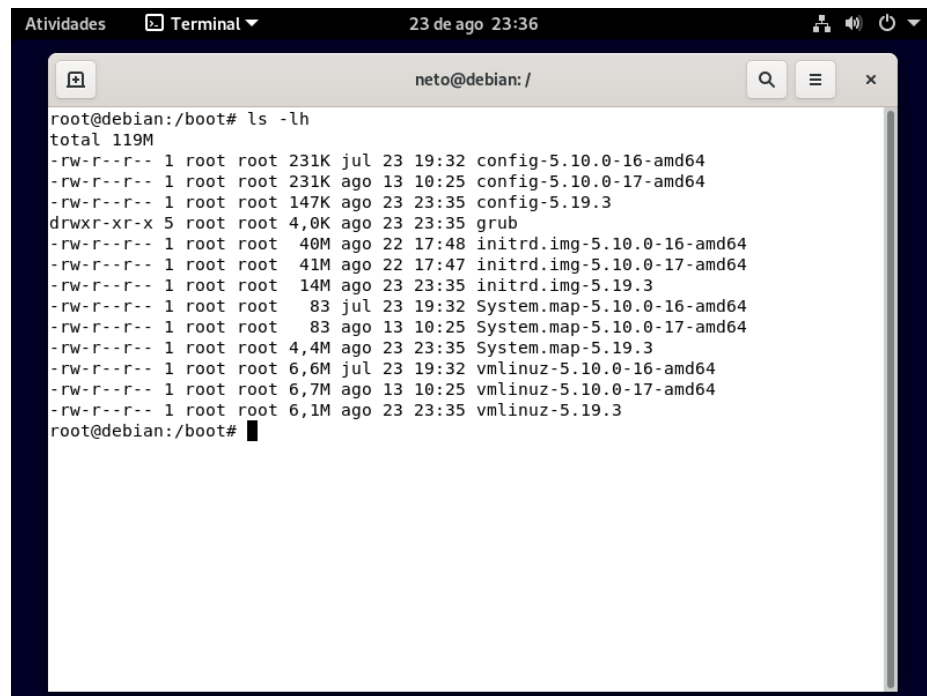
```

Atividades Terminal 23 de ago 23:35
neto@debian: /
INSTALL /lib/modules/5.19.3/kernel/sound/core/snd-pcm.ko
INSTALL /lib/modules/5.19.3/kernel/sound/core/snd-timer.ko
INSTALL /lib/modules/5.19.3/kernel/sound/core/snd.ko
INSTALL /lib/modules/5.19.3/kernel/sound/pci/ac97/snd-ac97-codec.ko
INSTALL /lib/modules/5.19.3/kernel/sound/pci/snd-intel8x0.ko
INSTALL /lib/modules/5.19.3/kernel/sound/soundcore.ko
DEPMOD /lib/modules/5.19.3
root@debian:/usr/src/linux-5.19.3# make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.19.3 /boot/vmlinuz-5.19.3
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.19.3 /boot/vmlinuz-5.19.3
update-initramfs: Generating /boot/initrd.img-5.19.3
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.19.3 /boot/vmlinuz-5.19.3
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.19.3 /boot/vmlinuz-5.19.3
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-5.19.3
Found initrd image: /boot/initrd.img-5.19.3
Found linux image: /boot/vmlinuz-5.10.0-17-amd64
Found initrd image: /boot/initrd.img-5.10.0-17-amd64
Found linux image: /boot/vmlinuz-5.10.0-16-amd64
Found initrd image: /boot/initrd.img-5.10.0-16-amd64
done
root@debian:/usr/src/linux-5.19.3#

```

Figura 20 - Instalação da última versão estável do kernel do Linux.

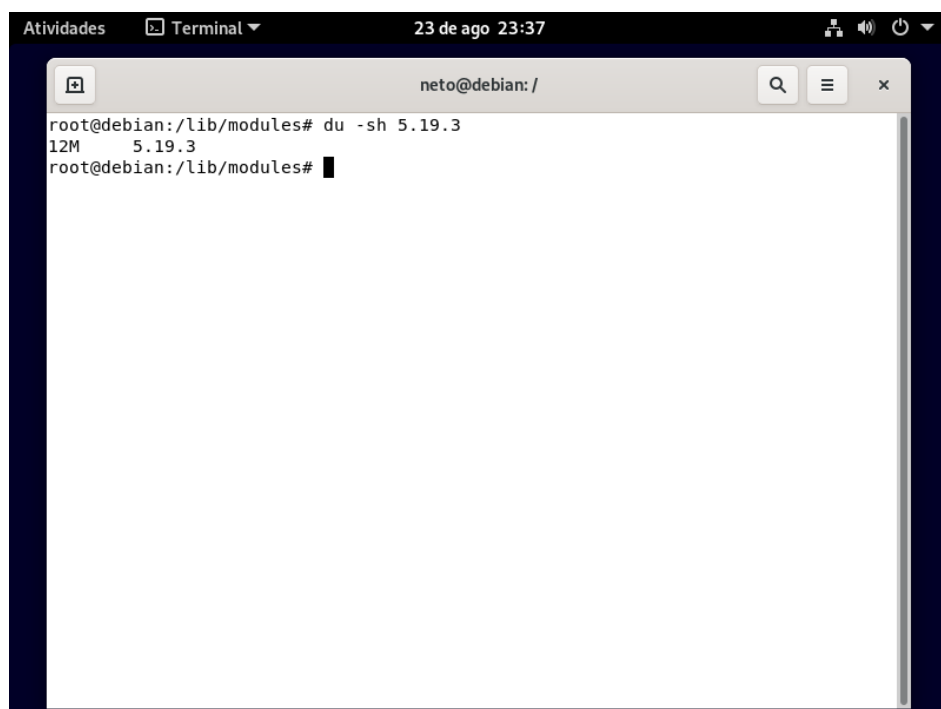
Finalizada a instalação do Kernel através do código-fonte, foram verificados os tamanhos dos arquivos (figura 22) e também o tamanho da pasta com os módulos compilados (figura 23). Conclui-se que o kernel é relativamente pequeno, se comparado a alguns programas atuais, ou até mesmo filmes via streaming, que chegam facilmente a marca dos Gigabytes, em contrapartida, o Kernel não passou dos 13MB.



```
Atividades Terminal 23 de ago 23:36
neto@debian: /

root@debian:/boot# ls -lh
total 119M
-rw-r--r-- 1 root root 231K jul 23 19:32 config-5.10.0-16-amd64
-rw-r--r-- 1 root root 231K ago 13 10:25 config-5.10.0-17-amd64
-rw-r--r-- 1 root root 147K ago 23 23:35 config-5.19.3
drwxr-xr-x 5 root root 4,0K ago 23 23:35 grub
-rw-r--r-- 1 root root 40M ago 22 17:48 initrd.img-5.10.0-16-amd64
-rw-r--r-- 1 root root 41M ago 22 17:47 initrd.img-5.10.0-17-amd64
-rw-r--r-- 1 root root 14M ago 23 23:35 initrd.img-5.19.3
-rw-r--r-- 1 root root 83 jul 23 19:32 System.map-5.10.0-16-amd64
-rw-r--r-- 1 root root 83 ago 13 10:25 System.map-5.10.0-17-amd64
-rw-r--r-- 1 root root 4,4M ago 23 23:35 System.map-5.19.3
-rw-r--r-- 1 root root 6,6M jul 23 19:32 vmlinuz-5.10.0-16-amd64
-rw-r--r-- 1 root root 6,7M ago 13 10:25 vmlinuz-5.10.0-17-amd64
-rw-r--r-- 1 root root 6,1M ago 23 23:35 vmlinuz-5.19.3
root@debian:/boot#
```

Figura 21 - Execução do comando ls -lh



```
Atividades Terminal 23 de ago 23:37
neto@debian: /

root@debian:/lib/modules# du -sh 5.19.3
12M    5.19.3
root@debian:/lib/modules#
```

Figura 22 - comando du -sh (versão do Kernel)