

Caio Rangel Ferreira Rodrigues

Guilherme Almeida Lopes

Ingrid Reupke Sbeguen Moran

Laboratório 1: Instalar GNU/Linux e compilar o núcleo

Relatório técnico de laboratório 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

Agosto / 2023

Resumo

Este documento descreve a instalação e configuração do Linux Debian, tais como seus comandos a serem executados e suas respectivas saídas. Além disso, apresenta, também, a configuração e compilação do núcleo do sistema operacional (kernel). Todos os comandos apresentam explicações detalhadas e imagens para exemplificação.

Palavras-chave: sistemas operacionais. linux. kernel. debian.

Sumário

1	Introdução	4
2	Objetivos	4
3	Materiais	4
4	Fundamentação	4
5	Procedimentos e Resultados	4
	5.1 Instalação e Configuração do Linux em uma Máquina Virtual	4
	5.2 Execução e Retorno dos Comandos	5
	5.2.1 Teste de Comandos	11
	5.3 Configuração do Kernel Linux	18
6	Conclusões	23
7	Referências	23

1 Introdução

Neste documento consta os objetivos do primeiro laboratório, os materiais utilizados para realizar os procedimentos, a fundamentação do conteúdo, os procedimentos efetuados e seus resultados e, por fim, a conclusão final.

2 Objetivos

O objetivo da realização do primeiro laboratório se constitui em compreender a estrutura básica do sistema operacional Linux e seus comandos. Para isso, foi necessário instalar o Linux e compilar e configurar seu kernel.

3 Materiais

Para realizar os procedimentos foi utilizado o software Virtual Box em computador de 4GB de RAM e 128GB de memória, além da última versão estável do Linux Debian e do Kernel.

4 Fundamentação

Por definição, um sistema operacional é uma camada de software que opera entre o hardware e os programas aplicativos voltados ao usuário final (MAZIERO, 2019). O Linux, sendo um exemplo de sistema operacional, o qual foi desenvolvido em 1991 por Linus Torvalds, pode ser definido como um sistema multiprogramado e multiusuário - além de ser tanto desktop como servidor - que possui um núcleo monolítico.

5 Procedimentos e Resultados

5.1 Instalação e Configuração do Linux em uma Máquina Virtual

O Virtual Box foi utilizado para fazer a instalação e configuração do Linux, usando a ISO do Debian 12. As configurações básicas, como idioma, fuso horário e periféricos foram preenchidas de acordo, tal como os usuários e senhas. Outras configurações como a RAM, o tamanho do disco e a quantidade de processadores também foram definidas previamente, sendo elas, respectivamente, 2048MB, 109GB e 1 processador.

Para o particionamento, foi necessário configurar manualmente as partições, seguindo os seguintes tamanhos de espaço reservados. Para a partição raiz, ou seja /, foi utilizado 4GB. Já para o /home e o /boot, foram utilizados, respectivamente, 2GB e 249.6MB. A área de troca, ou seja, o swap, recebeu 4GB enquanto o /usr recebeu 10GB,

devido a necessidade de compilar o núcleo. Por fim, foi utilizado 1.2GB para o /var. A imagem 2 mostra o particionamento descrito.

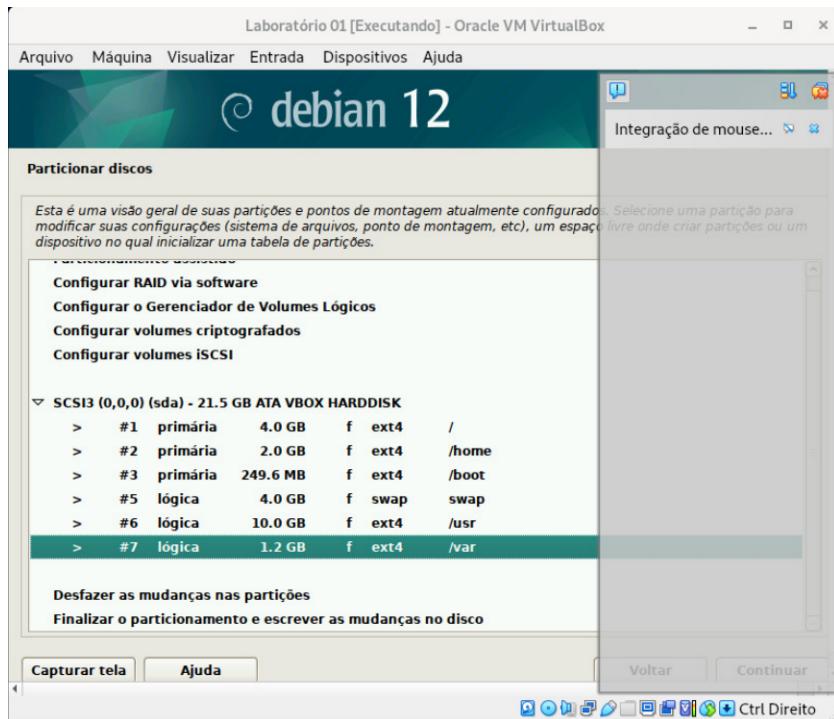


Figura 1 – Particionamento Manual

5.2 Execução e Retorno dos Comandos

Após a instalação do sistema, é possível executar comandos de verificação e concluir algumas configurações. Primeiramente, para analisar os processos em execução do sistema, basta executar o comando ps aux, que os lista.

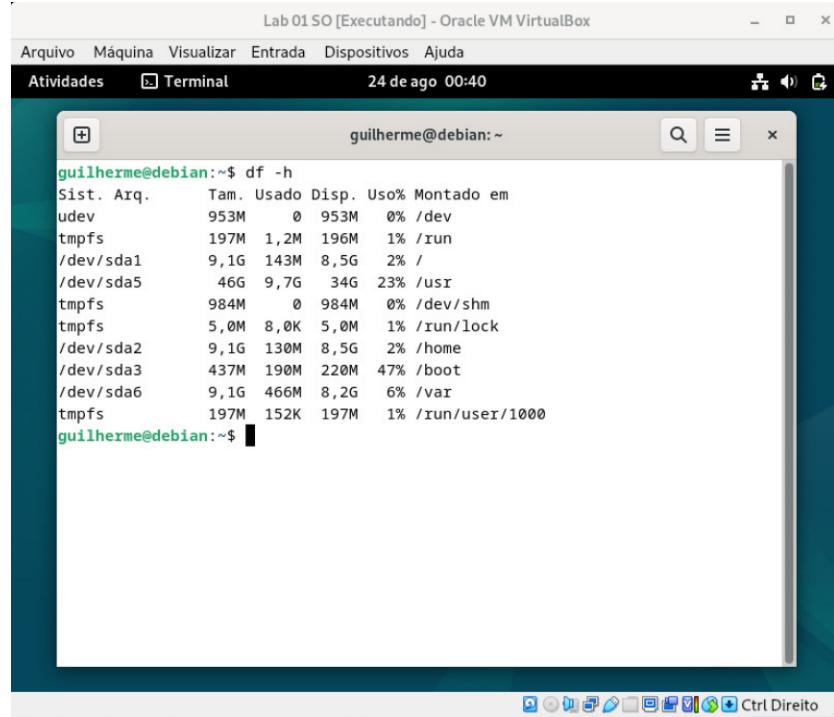
```

guilherme@debian:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root         1  0.0  0.2 168172  4468 ?        Ss  ago23  0:01 /sbin/init
root         2  0.0  0.0     0     0 ?        S  ago23  0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        I<  ago23  0:00 [rcu_gp]
root         4  0.0  0.0     0     0 ?        I<  ago23  0:00 [rcu_par_g
root         5  0.0  0.0     0     0 ?        I<  ago23  0:00 [slub_flus
root         6  0.0  0.0     0     0 ?        I<  ago23  0:00 [netns]
root        10  0.0  0.0     0     0 ?        I<  ago23  0:00 [mm_percpu
root        11  0.0  0.0     0     0 ?        I  ago23  0:00 [rcu_tasks
root        12  0.0  0.0     0     0 ?        I  ago23  0:00 [rcu_tasks
root        13  0.0  0.0     0     0 ?        I  ago23  0:00 [rcu_tasks
root        14  0.0  0.0     0     0 ?        S  ago23  0:06 [ksoftirqd
root        15  0.0  0.0     0     0 ?        I  ago23  0:05 [rcu_prem
root        16  0.0  0.0     0     0 ?        S  ago23  0:00 [migration
root        18  0.0  0.0     0     0 ?        S  ago23  0:00 [cpuhp/0]
root        20  0.0  0.0     0     0 ?        S  ago23  0:00 [kdevtmpfs
root        21  0.0  0.0     0     0 ?        I<  ago23  0:00 [inet_frag
root        22  0.0  0.0     0     0 ?        S  ago23  0:00 [kaudit]
root        23  0.0  0.0     0     0 ?        S  ago23  0:00 [khungtask
root        24  0.0  0.0     0     0 ?        S  ago23  0:00 [oom_reape
root        27  0.0  0.0     0     0 ?        I<  ago23  0:00 [writeback
root        28  0.0  0.0     0     0 ?        S  ago23  0:01 [kcompactd

```

Figura 2 – Execução e retorno do comando ps aux.

Para verificar o espaço em disco em cada partição, foi executado o comando df -h. O particionamento foi realizado com a finalidade de compilar kernel, por isso a partição /usr possuir parte da memória, visto que nele iríamos realizar desempacotamento de arquivos do linux-6.4.11. O /boot, por ser um programa que carrega o kernel na memória foi utilizado o seu valor usual, também devemos destacar não criamos a partição /boot/efi, visto que o padrão utilizado pela máquina na atividade usava BIOS. Deixamos uma quantidade considerável / visto que é o módulo mais importante do sistema, pois apresenta papel fundamental para compilar o mesmo, e demais partições divididas pelo sua importancia para realização da tarefa.



Lab 01 SO [Executando] - Oracle VM VirtualBox

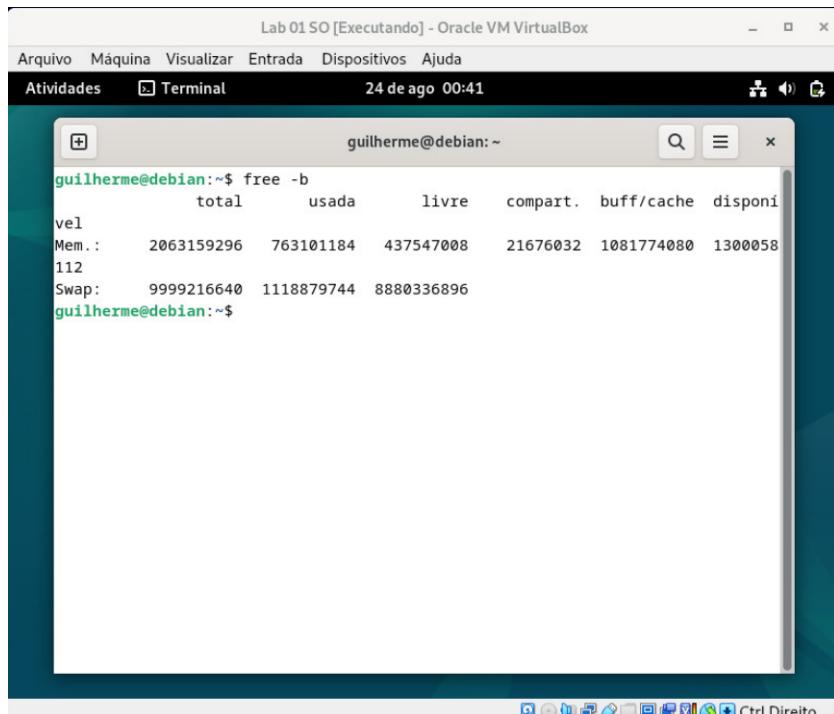
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 24 de ago 00:40

```
guilherme@debian:~$ df -h
Sist. Arq.      Tam. Usado Disp. Uso% Montado em
udev            953M    0  953M  0% /dev
tmpfs           197M  1,2M  196M  1% /run
/dev/sda1        9,1G 143M  8,5G  2% /
/dev/sda5        466   9,7G  34G  23% /usr
tmpfs           984M    0  984M  0% /dev/shm
tmpfs           5,0M  8,0K  5,0M  1% /run/lock
/dev/sda2        9,1G 130M  8,5G  2% /home
/dev/sda3        437M 190M  220M  47% /boot
/dev/sda6        9,1G 466M  8,2G  6% /var
tmpfs           197M  152K  197M  1% /run/user/1000
guilherme@debian:~$
```

Figura 3 – Execução e retorno do comando df -h.

Os comandos cat /proc/meminfo e free -b exibem a quantidade de memória total e disponível do sistema. O free -b mostra, também, informações a respeito do swap e dos buffers.



Lab 01 SO [Executando] - Oracle VM VirtualBox

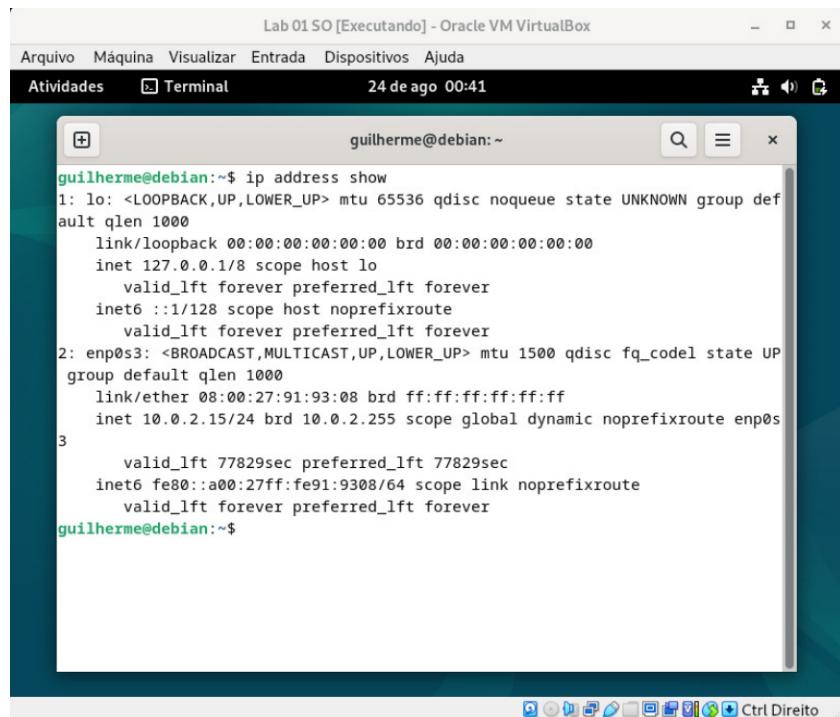
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 24 de ago 00:41

```
guilherme@debian:~$ free -b
              total        usada       livre     compart.  buff/cache disponí
vel
Mem.:    2063159296    763101184    437547008    21676032  1081774080  1300058
112
Swap:    9999216640   1118879744   8880336896
guilherme@debian:~$
```

Figura 4 – Execução e retorno do comando free -b.

Para verificar a rede, primeiramente, deve-se exibir as interfaces de rede disponíveis, ou seja, seus IPs, e também exibir uma tabela de roteamento. Os comandos ip address show (Figura 5) e ip route (Figura 6) foram executados, respectivamente, para retornar tais informações descritas. Por fim, para averiguar o funcionamento da rede, basta executar um ping do google (ping www.google.com.br). Como não houve nenhuma falha ao executar o ping, é confirmado que a rede está em pleno funcionamento e suas configurações estão corretas.



The screenshot shows a terminal window titled "Lab 01 SO [Executando] - Oracle VM VirtualBox". The window contains the output of the "ip address show" command. The output details three network interfaces: "lo" (loopback), "enp0s3" (ethernet), and "enp0s3:3" (loopback). The "lo" interface has an IPv4 address of 127.0.0.1 and an IPv6 address of ::1. The "enp0s3" interface has an IPv4 address of 10.0.2.15 and an IPv6 address of fe80::a00:27ff:fe91:9308. The "enp0s3:3" interface is a link-local loopback interface with an IPv6 address of fe80::a00:27ff:fe91:9308/64. The terminal window also shows the user's name "guilherme" and the prompt "guilherme@debian:~\$".

```
guilherme@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 noprefixroute
            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 08:00:27:91:93:08 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 77829sec preferred_lft 77829sec
        inet6 fe80::a00:27ff:fe91:9308/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
guilherme@debian:~$
```

Figura 5 – Execução e retorno do comando ip address show.

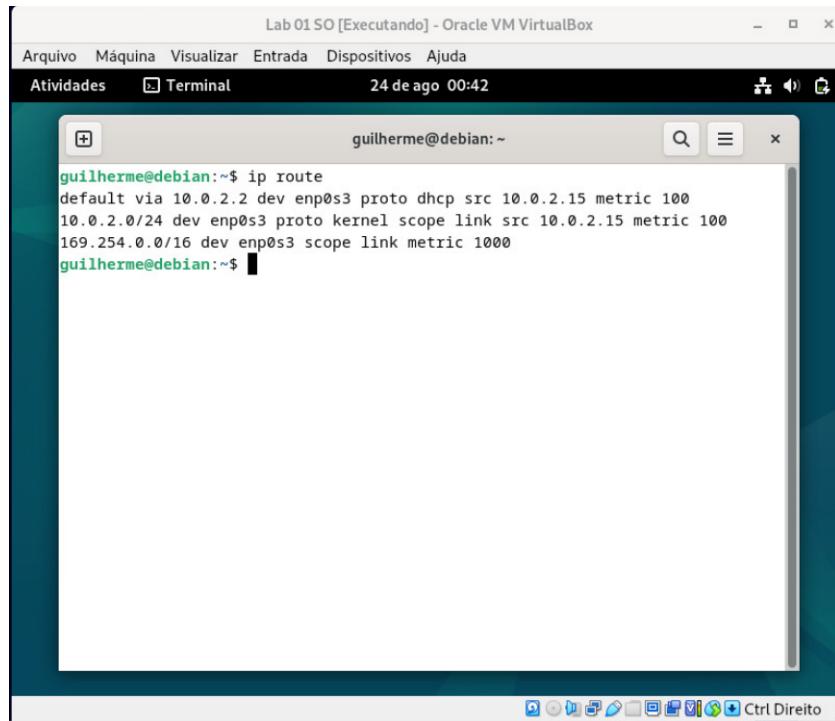


Figura 6 – Execução e retorno do comando ip route.

The screenshot shows a terminal window titled 'Terminal' running on a Linux system. The window title bar includes the text 'Lab 01 SO [Executando] - Oracle VM VirtualBox', 'Arquivo', 'Máquina', 'Visualizar', 'Entrada', 'Dispositivos', 'Ajuda', 'Atividades', and the date and time '24 de ago 00:42'. The terminal itself has a dark background and displays the following command and its output:

```
guilherme@debian:~$ ping www.google.com
PING www.google.com (142.251.128.36) 56(84) bytes of data.
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=1 ttl=63 time=
17.7 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=2 ttl=63 time=
18.5 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=3 ttl=63 time=
20.6 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=4 ttl=63 time=
21.1 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=5 ttl=63 time=
18.6 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=6 ttl=63 time=
17.5 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=7 ttl=63 time=
18.6 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=8 ttl=63 time=
21.7 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=9 ttl=63 time=
18.9 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=10 ttl=63 time=
18.6 ms
64 bytes from gru06s68-in-f4.1e100.net (142.251.128.36): icmp_seq=11 ttl=63 time=
20.4 ms
```

Figura 7 – Execução e saída do comando ping www.google.com.br.

Então, para configurar os repositórios do apt-get e adicionar os repositórios de segurança, foi utilizado o cd /etc/apt e o nano sources.list.

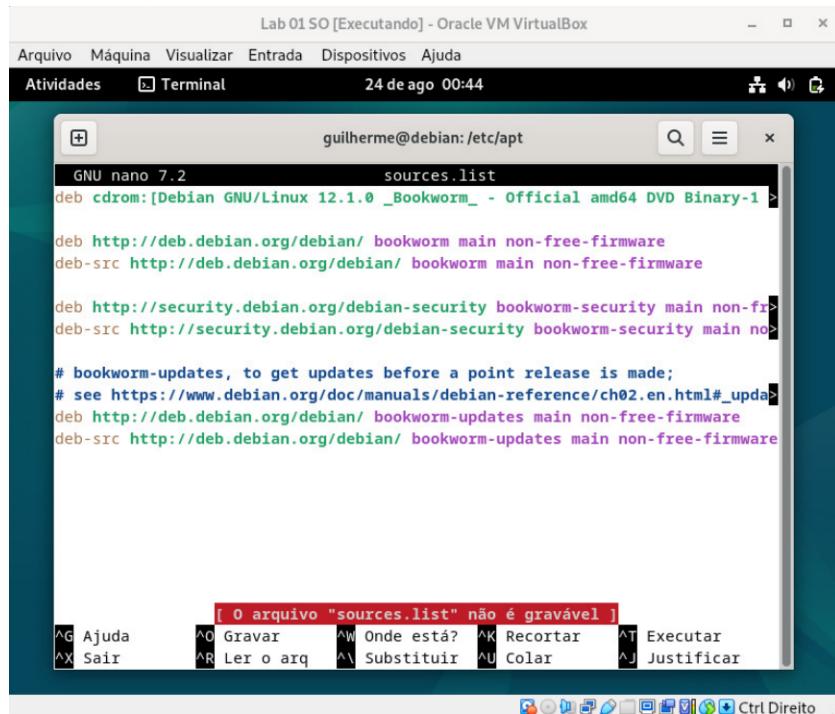


Figura 8 – Interface de execução do comando nano sources.list.

Pode-se, também, visualizar a versão do kernel atual executando `uname -a`, que mostra a versão 6.1.0.

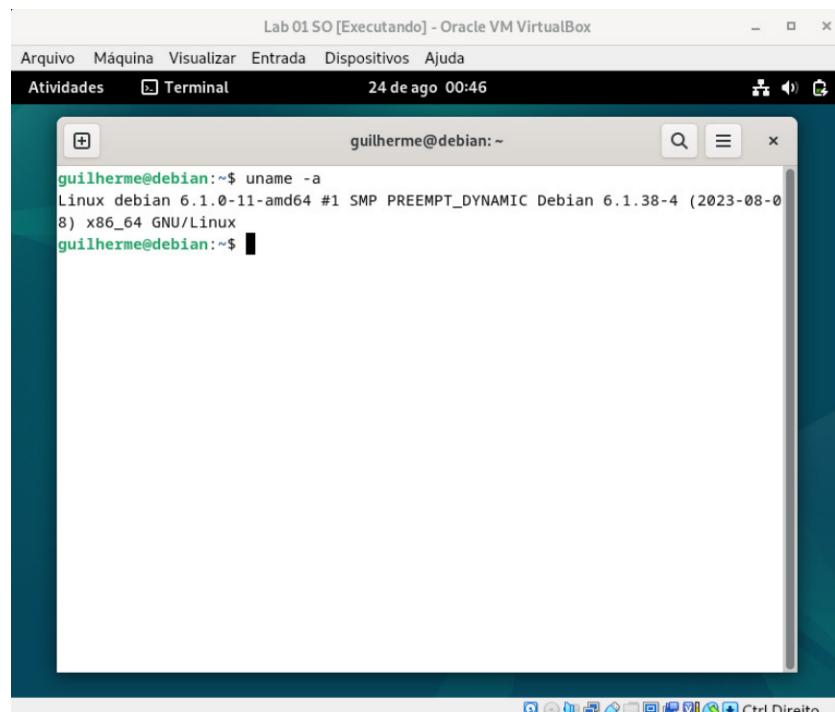
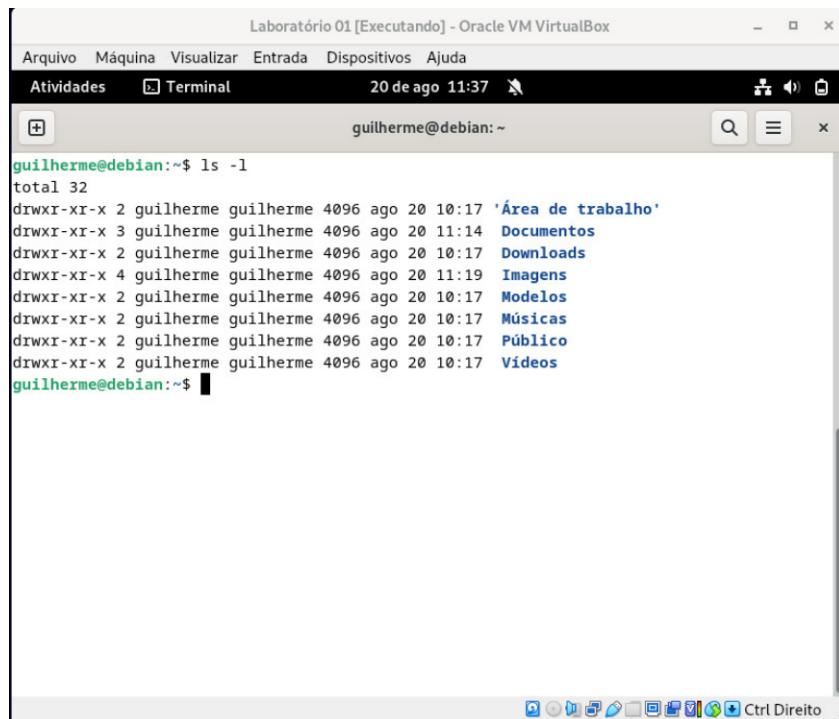


Figura 9 – Execução e retorno do comando `uname -a`.

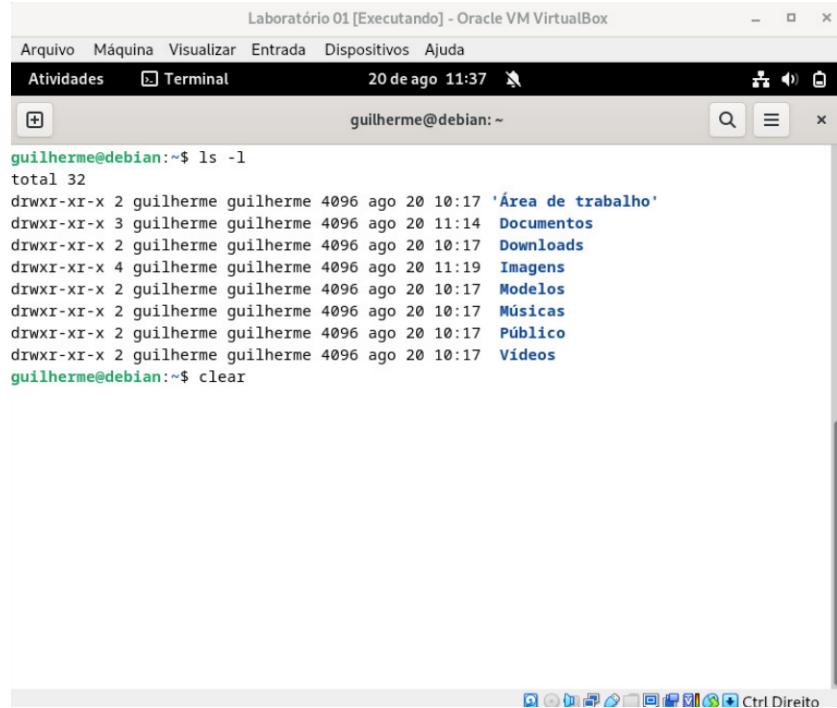
5.2.1 Teste de Comandos

Para finalizar, é necessário executar alguns comandos a fins de testes. Em primeiro plano, executando o comando ls -l (Figura 10), é retornado uma lista de conteúdos de um diretório, além de exibir, na primeira coluna, as permissões, na segunda, o número de entidades, na terceira o dono, seguido do grupo, depois o tamanho, a data e a hora em que foi atualizado pela última vez e o seu respectivo nome. Caso queira limpar o terminal, basta executar o comando clear (Figuras 11 e 12).



The screenshot shows a terminal window titled "Laboratório 01 [Executando] - Oracle VM VirtualBox". The window has a menu bar with "Arquivo", "Máquina", "Visualizar", "Entrada", "Dispositivos", and "Ajuda". Below the menu is a toolbar with icons for "Atividades", "Terminal", and a date/time indicator "20 de ago 11:37". The main area of the window is a terminal session. The user is logged in as "guilherme@debian:~". They have run the command "ls -l", which lists the contents of the current directory. The output shows a total of 32 files and directories, including "Área de trabalho", "Documentos", "Downloads", "Imagens", "Modelos", "Músicas", "Público", and "Vídeos". Each entry includes permissions (e.g., drwxr-xr-x), file size (e.g., 4096), last modified date (e.g., ago 20 10:17), and name (e.g., Área de trabalho). The terminal prompt ends with "guilherme@debian:~\$".

Figura 10 – Execução e retorno do comando ls -l.



Laboratório 01 [Executando] - Oracle VM VirtualBox

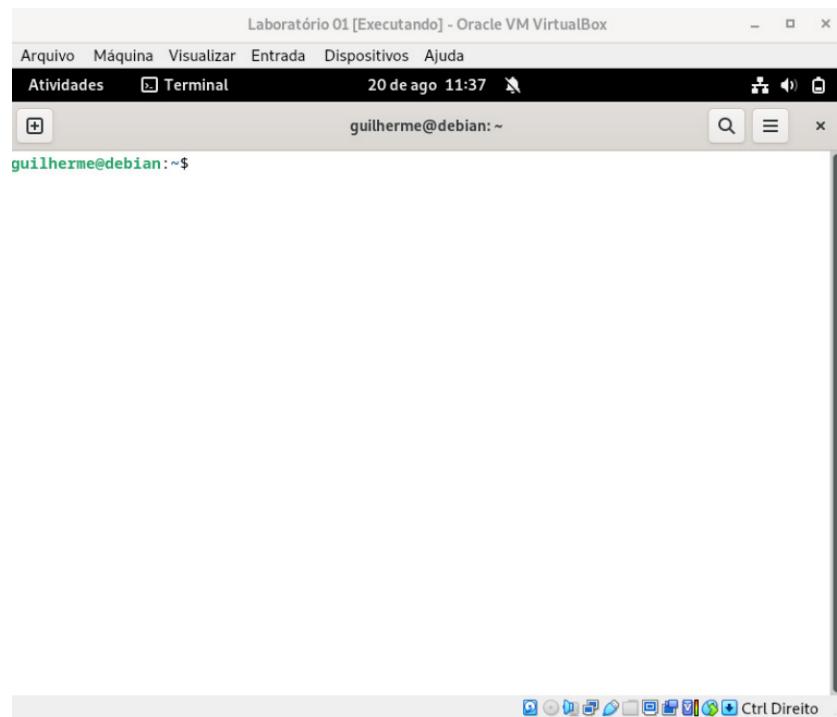
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 20 de ago 11:37

guilherme@debian:~

```
guilherme@debian:~$ ls -l
total 32
drwxr-xr-x 2 guilherme guilherme 4096 ago 20 10:17 'Área de trabalho'
drwxr-xr-x 3 guilherme guilherme 4096 ago 20 11:14 Documentos
drwxr-xr-x 2 guilherme guilherme 4096 ago 20 10:17 Downloads
drwxr-xr-x 4 guilherme guilherme 4096 ago 20 11:19 Imagens
drwxr-xr-x 2 guilherme guilherme 4096 ago 20 10:17 Modelos
drwxr-xr-x 2 guilherme guilherme 4096 ago 20 10:17 Músicas
drwxr-xr-x 2 guilherme guilherme 4096 ago 20 10:17 PÚblico
drwxr-xr-x 2 guilherme guilherme 4096 ago 20 10:17 Vídeos
guilherme@debian:~$ clear
```

Figura 11 – Execução do comando clear.



Laboratório 01 [Executando] - Oracle VM VirtualBox

Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 20 de ago 11:37

guilherme@debian:~

```
guilherme@debian:~$
```

Figura 12 – Terminal limpo após a execução do comando clear.

O comando cd serve para a mudança de diretório de trabalho. Dito isso, basta executar cd seguido de um diretório. Por exemplo, cd / permite o acesso ao diretório raiz do sistema operacional, enquanto cd /home permite que você acesse o diretório /home.

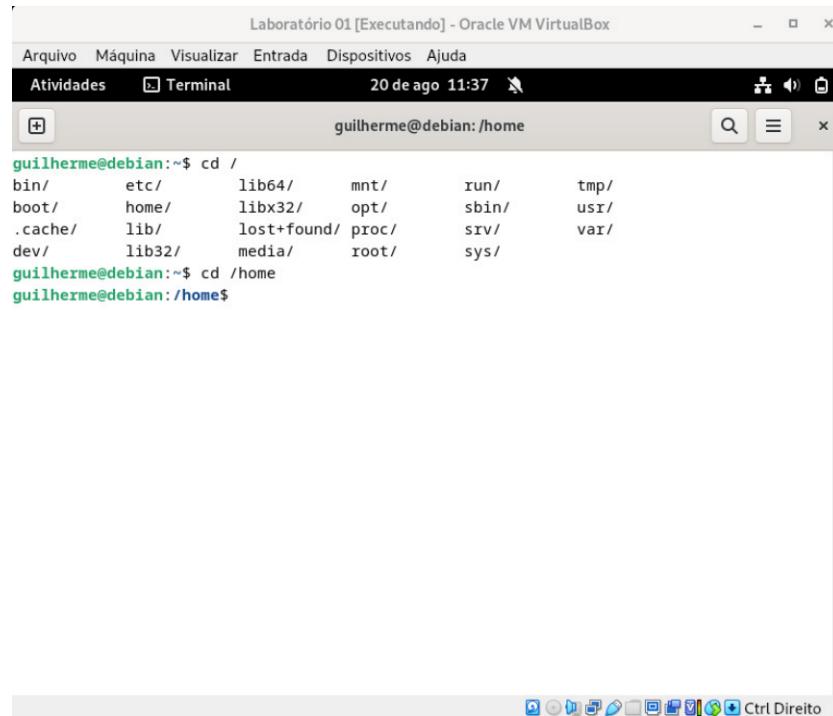


Figura 13 – Execução e retorno do comando cd

Para editar arquivos, pode-se utilizar o comando pico, que ao executado, abre uma interface de edição de textos simples. Um arquivo testelab01 (Figura 14) foi criado - com o conteúdo descrito - a fim de exemplificar melhor esse e outros comandos a seguir.

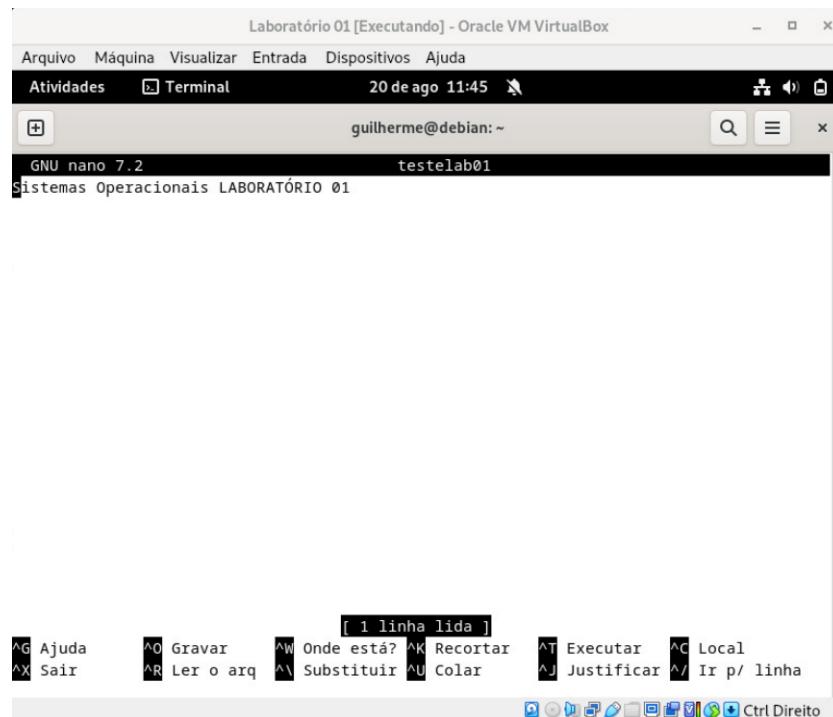


Figura 14 – Execução do comando pico.

O comando cat, derivado de concatenar, como o próprio nome diz, junta arquivos. Porém, no exemplo, tal comando foi utilizado para exibir o conteúdo do arquivo testelab01 (Figura 15). Já para excluir algum arquivo, é simples, basta executar o comando rm seguido do arquivo que deseja excluir (Figura 16). A fim de exemplificar melhor, é possível observar, a partir do ls, o arquivo testelab01 antes e depois de ser excluído.

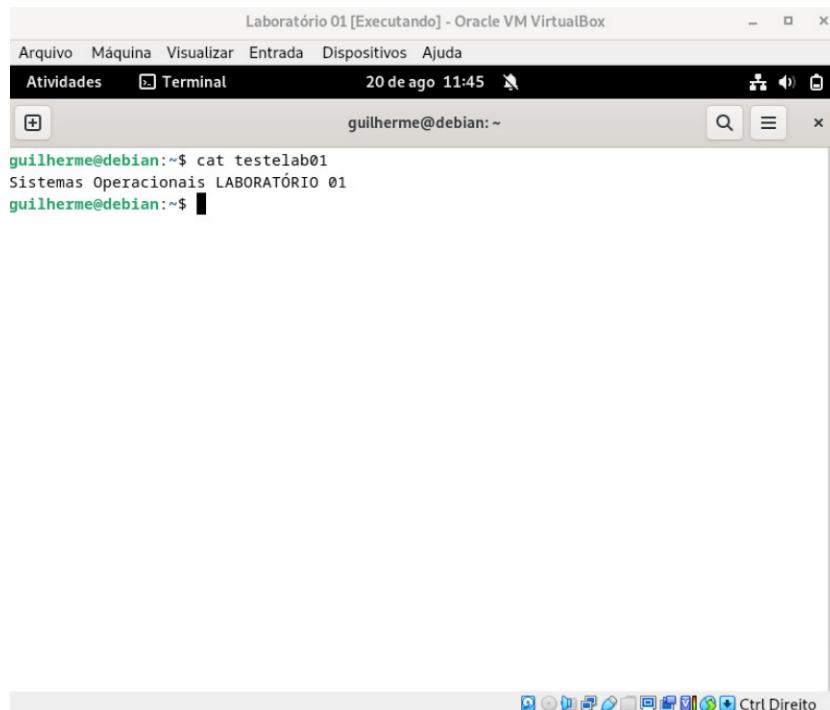
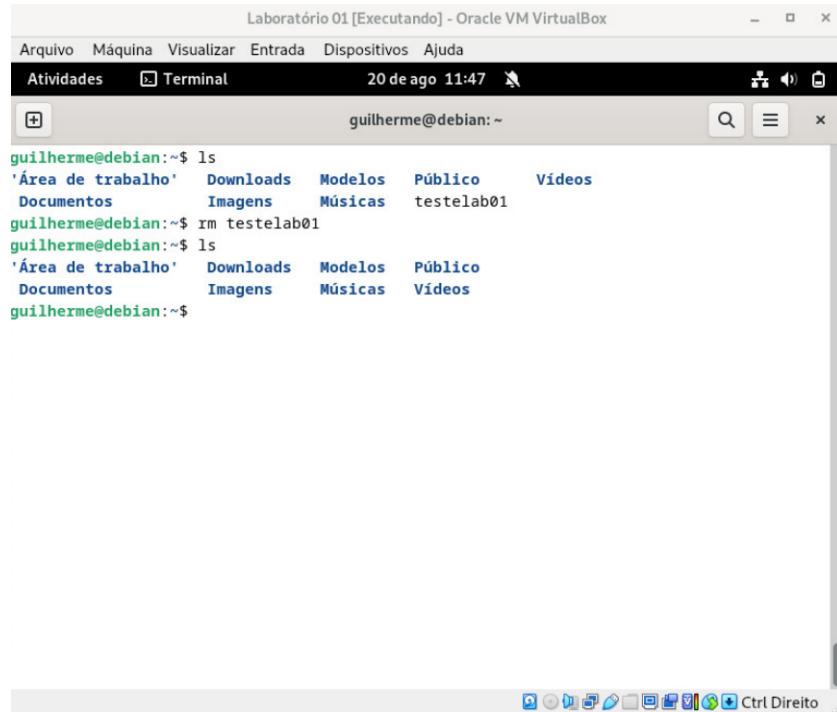


Figura 15 – Execução e retorno do comando cat.



Laboratório 01 [Executando] - Oracle VM VirtualBox

Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 20 de ago 11:47

guilherme@debian:~

```
guilherme@debian:~$ ls
'Área de trabalho' Downloads Modelos Público Vídeos
Documentos Imagens Músicas testelab01
guilherme@debian:~$ rm testelab01
guilherme@debian:~$ ls
'Área de trabalho' Downloads Modelos Público
Documentos Imagens Músicas Vídeos
guilherme@debian:~$
```

Figura 16 – Verificação da exclusão do arquivo testelab01 após a execução do comando rm.

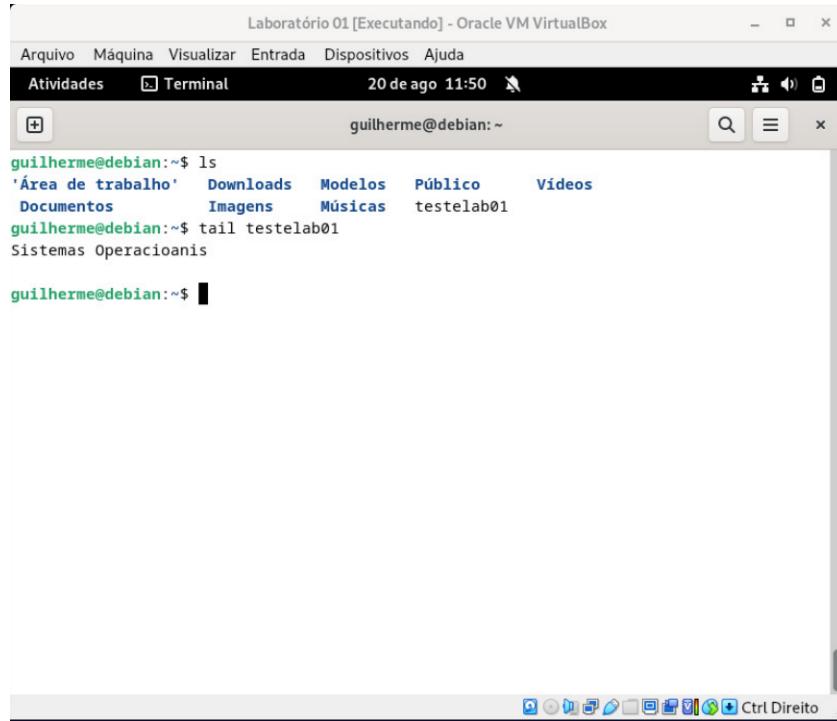
Os comandos head e tail mostram, respectivamente, as primeiras e as últimas linhas de um arquivo texto.



guilherme@debian:~

```
guilherme@debian:~$ head 'Lab 01'
Sistema Operacionais
guilherme@debian:~$
```

Figura 17 – Execução e retorno do comando head.

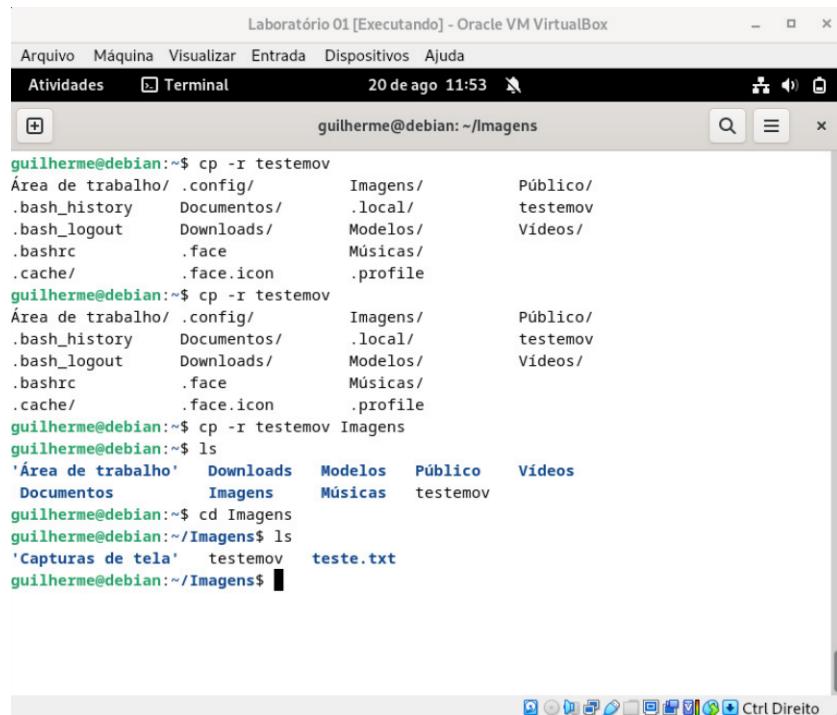


```
guilherme@debian:~$ ls
'Área de trabalho' Downloads Modelos Público Videos
Documentos Imagens Músicas testelab01
guilherme@debian:~$ tail testelab01
Sistemas Operacionais

guilherme@debian:~$
```

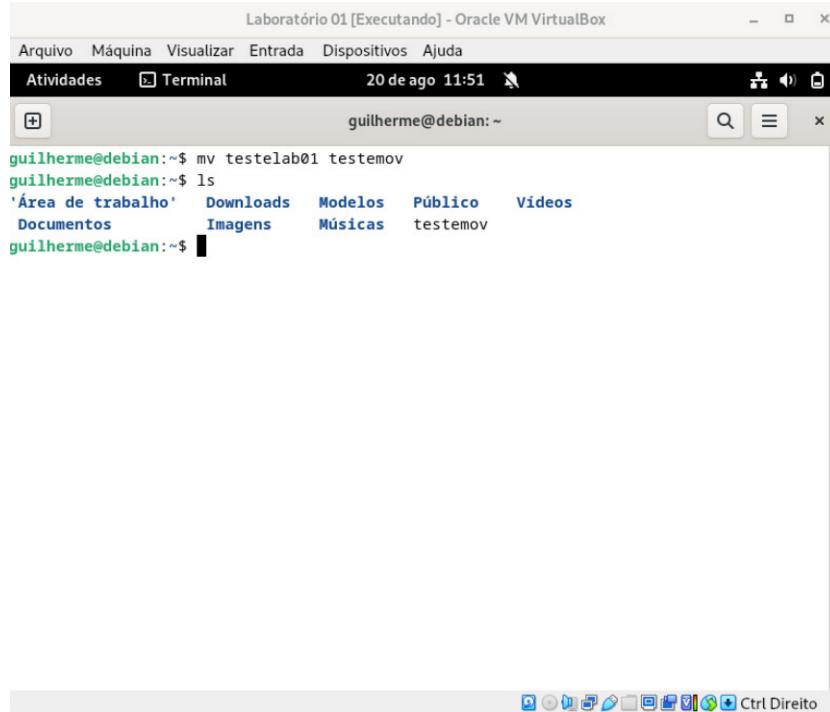
Figura 18 – Execução e retorno do comando tail.

Para fazer cópias de arquivos basta utilizar o comando cp. Utilizando cp -r é possível copiar arquivos recursivamente. No exemplo é mostrado como o arquivo testemov foi copiado para o diretório /Imagens (Figura 19). Já o comando mv move arquivos, como por exemplo, mover o arquivo testelab01 para o testemov (Figura 20).



```
guilherme@debian:~$ cp -r testemov
Área de trabalho/.config/ Imagens/ Público/
.bash_history Documentos/.local/ testemov
.bash_logout Downloads/ Modelos/ Vídeos/
.bashrc .face Músicas/.profile
.cache/.face.icon .profile
guilherme@debian:~$ cp -r testemov Imagens
Área de trabalho/.config/ Imagens/ Público/
.bash_history Documentos/.local/ testemov
.bash_logout Downloads/ Modelos/ Vídeos/
.bashrc .face Músicas/.profile
.cache/.face.icon .profile
guilherme@debian:~$ cp -r testemov Imagens
guilherme@debian:~$ ls
'Área de trabalho' Downloads Modelos Público Videos
Documentos Imagens Músicas testemov
guilherme@debian:~$ cd Imagens
guilherme@debian:~/Imagens$ ls
'Capturas de tela' testemov teste.txt
guilherme@debian:~/Imagens$
```

Figura 19 – Execução e retorno do comando cp.



Laboratório 01 [Executando] - Oracle VM VirtualBox

Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 20 de ago 11:51

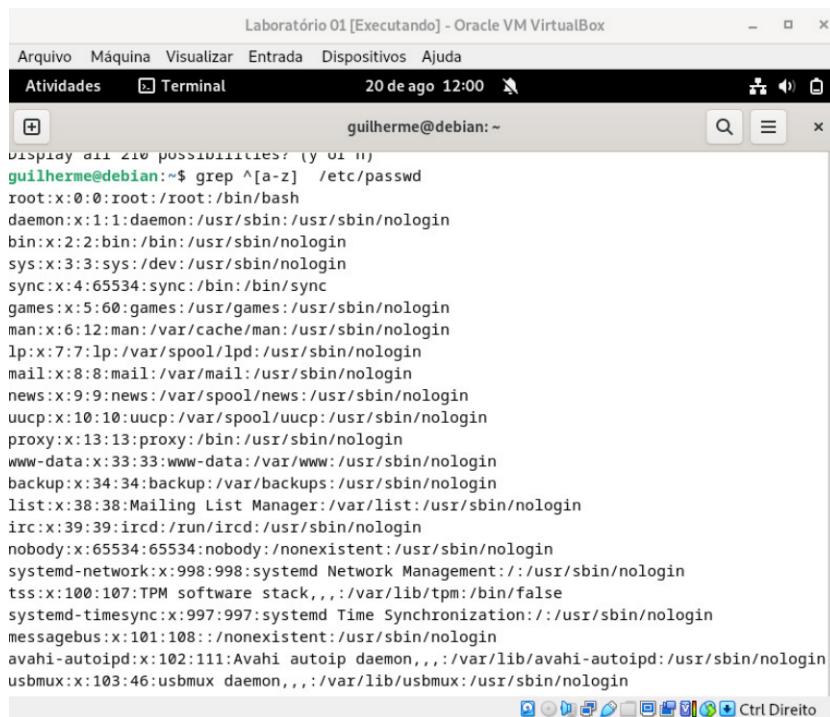
guilherme@debian:~

```
guilherme@debian:~$ mv testelab01 testemov
guilherme@debian:~$ ls
'Área de trabalho' Downloads Modelos Público Videos
Documentos Imagens Músicas testemov
guilherme@debian:~$
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal title is "Laboratório 01 [Executando] - Oracle VM VirtualBox". The window menu bar includes "Arquivo", "Máquina", "Visualizar", "Entrada", "Dispositivos", and "Ajuda". The title bar shows "Atividades" and "Terminal" with the date and time "20 de ago 11:51". The terminal window has a search bar and a toolbar with icons for copy, paste, and others. The user's prompt is "guilherme@debian:~". The user runs the command "mv testelab01 testemov", which renames the file "testelab01" to "testemov". Then, the user runs "ls" to list the contents of the directory, which now includes "testemov".

Figura 20 – Execução e retorno do comando mv.

O comando grep basicamente procura padrões em arquivos. Por exemplo, ao executar grep ^[a-z] /etc/passwd, é buscado, em /etc/passwd, linhas que começam com os caracteres que vão de a à z.



Laboratório 01 [Executando] - Oracle VM VirtualBox

Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

Atividades Terminal 20 de ago 12:00

guilherme@debian:~

```
Display all 216 possibilities? (y or n)
guilherme@debian:~$ grep ^[a-z] /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
tss:x:100:107:TPM software stack,,,:/var/lib/tpm:/bin/false
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:101:108::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:102:111:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:103:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal title is "Laboratório 01 [Executando] - Oracle VM VirtualBox". The window menu bar includes "Arquivo", "Máquina", "Visualizar", "Entrada", "Dispositivos", and "Ajuda". The title bar shows "Atividades" and "Terminal" with the date and time "20 de ago 12:00". The terminal window has a search bar and a toolbar with icons for copy, paste, and others. The user's prompt is "guilherme@debian:~". The user runs the command "grep ^[a-z] /etc/passwd", which prints all lines from the "/etc/passwd" file that start with a lowercase letter from a to z. The output shows various system accounts like root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, nobody, systemd-network, tss, messagebus, avahi-autoipd, and usbmux.

Figura 21 – Execução e retorno do comando grep.

5.3 Configuração do Kernel Linux

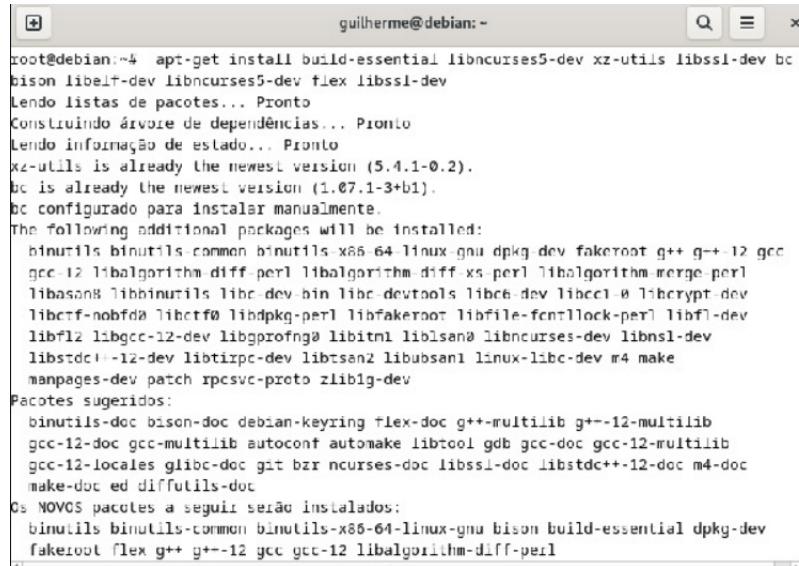
Para instalar a versão mais recente e estável do Kernel Linux e configurá-la, primeiramente precisa-se entrar como super usuário (Figura 22).



```
guilherme@debian:~$ su
Senha:
root@debian:/home/guilherme# cd /
root@debian:/#
```

Figura 22 – Comando **su**.

Após isso, precisa-se instalar os pacotes básicos para a compilação (Figura 23).



```
root@debian:~# apt-get install build-essential libncurses5-dev xz-utils libssl-dev bc
bison libelf-dev libncurses5-dev flex libssl-dev
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
xz-utils is already the newest version (5.4.1-0.2).
bc is already the newest version (1.07.1-3+b1).
bc configurado para instalar manualmente.
The following additional packages will be installed:
binutils binutils-common binutils-x86_64-linux-gnu dpkg-dev fakeroot g++-g++-12 gcc
gcc-12 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
libasan8 libbinutils libc-dev bin libc-devtools libc6-dev libcc1-0 libcrypt-dev
libctf-nobfd8 libctf8 libdpkg-perl libfakeroot libfile-fcntllock-perl libfl-dev
libfl2 libgcc-12-dev libgprofng8 libibtn8 libltsan8 libncurses-dev libns1-dev
libstdc++-12-dev libtirpc-dev libtsan8 libubsan8 linux-libc-dev m4 make
manpages-dev patch rpcsvc-proto zlib1g-dev
Pacotes sugeridos:
binutils-doc bison-doc debian-keyring flex-doc g++-multilib g++-12-multilib
gcc-12-doc gcc-multilib autoconf automake libtool gdb gcc-doc gcc-12-multilib
gcc-12-locales glibc-doc git bzr ncurses-doc libssl-doc libstdc++-12-doc m4-doc
make-doc ed diffutils-doc
Os NOVOS pacotes a seguir serão instalados:
binutils binutils-common binutils-x86_64-linux-gnu bison build-essential dpkg-dev
fakeroot flex g++-g++-12 gcc gcc-12 libalgorithm-diff-perl
```

Figura 23 – Comando **apt-get install make build-essential libncurses5-dev xz-utils libssl-dev bc bison libelf-dev flex dwarves**.

Desse modo, utilizando site oficial ([ORGANIZATION, 2023](#)), obtém-se a versão atual mais estável do Kernel Linux, atualmente a 6.4.11, que será utilizada para aplicar e testar os comandos sugeridos em aula (Figura 24).



```

Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Atividades Terminal 21 de ago 18:52
guilherme@debian:~#
root@debian:~# wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.4.11.tar.xz
--2023-08-21 18:51:57-- https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.4.11.tar.xz
KZ
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... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 137836952 (131M) [application/x-xz]
Salvando em: "linux-6.4.11.tar.xz"

linux-6.4.11.tar.xz 100%[=====] 131,45M 5,54MB/s em 42s

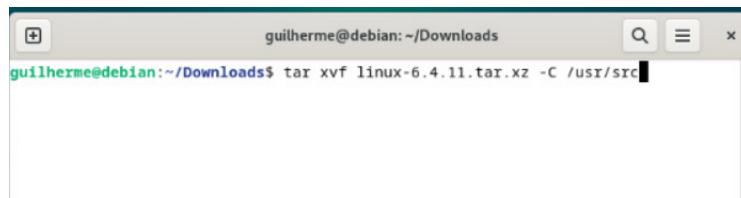
2023-08-21 18:52:45 (3,12 MB/s) - "linux-6.4.11.tar.xz" salvo [137836952/137836952]

root@debian:~# 

```

Figura 24 – Comando wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.4.11.tar.xz.

Ao terminar, os arquivos do SO são descompactados no diretório /usr/src (Figura 25)



```

guilherme@debian:~/Downloads$ tar xvf linux-6.4.11.tar.xz -C /usr/src

```

Figura 25 – Comando tar xvf linux-6.4.11.tar.xz - C /usr/src.

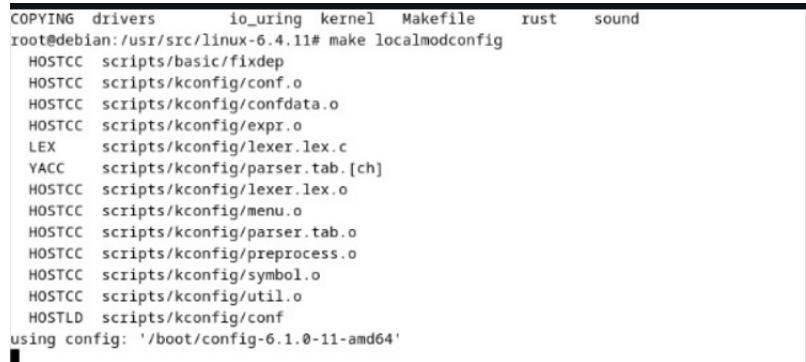
```

linux-6.4.11/arch/arm64/boot/dts/freescale/mba8mx.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-bman-portals.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-10g-0.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-10g-1.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-1g-0.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-1g-1.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-1g-2.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-1g-3.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-1g-4.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0-1g-5.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-fman3-0.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/qoriq-qman-portals.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/s32g2.dtsi
linux-6.4.11/arch/arm64/boot/dts/freescale/s32g274a-evb.dts
linux-6.4.11/arch/arm64/boot/dts/freescale/s32g274a-rdb2.dts
linux-6.4.11/arch/arm64/boot/dts/freescale/s32v234-evb.dts
linux-6.4.11/arch/arm64/boot/dts/freescale/s32v234.dtsi
linux-6.4.11/arch/arm64/boot/dts/hisilicon/
linux-6.4.11/arch/arm64/boot/dts/hisilicon/Makefile
linux-6.4.11/arch/arm64/boot/dts/hisilicon/hi3660-coresight.dtsi
linux-6.4.11/arch/arm64/boot/dts/hisilicon/hi3660-hikey960.dts
linux-6.4.11/arch/arm64/boot/dts/hisilicon/hi3660.dtsi
linux-6.4.11/arch/arm64/boot/dts/hisilicon/hi3670-hikey970.dts
linux-6.4.11/arch/arm64/boot/dts/hisilicon/hi3670.dtsi

```

Figura 26 – Comando tar xvf linux-6.4.11.tar.xz - C /usr/src.

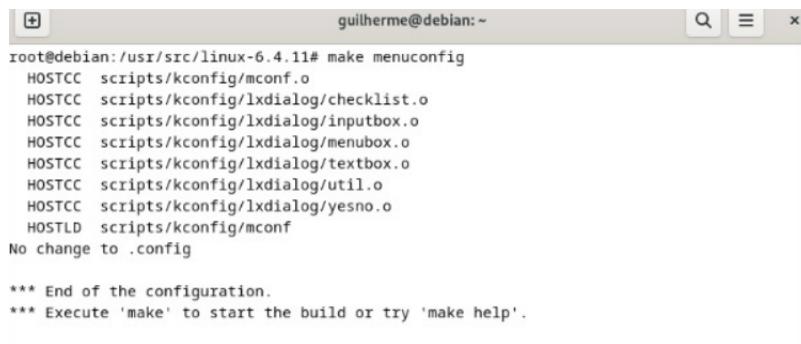
Após os passos anteriores, acessa-se o diretório raiz do kernel utilizando o comando `cd /usr/src/linux-6.4.11`. Nesse diretório o usuário deve salvar as configurações atuais do Kernel (Figura 27).



```
COPYING drivers io_uring kernel Makefile rust sound
root@debian:/usr/src/linux-6.4.11# make localmodconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/menu.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
using config: '/boot/config-6.1.0-11-amd64'
```

Figura 27 – Comando `make localmodconfig`.

Com isso, pode-se configurar o Kernel Linux (Figura 28), conforme as necessidades do usuário (Figuras 29, criando um novo arquivo .config (Figura 30)).



```
root@debian:/usr/src/linux-6.4.11# make menuconfig
HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTLD scripts/kconfig/mconf
No change to .config

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

Figura 28 – Comando `make localmodconfig`.

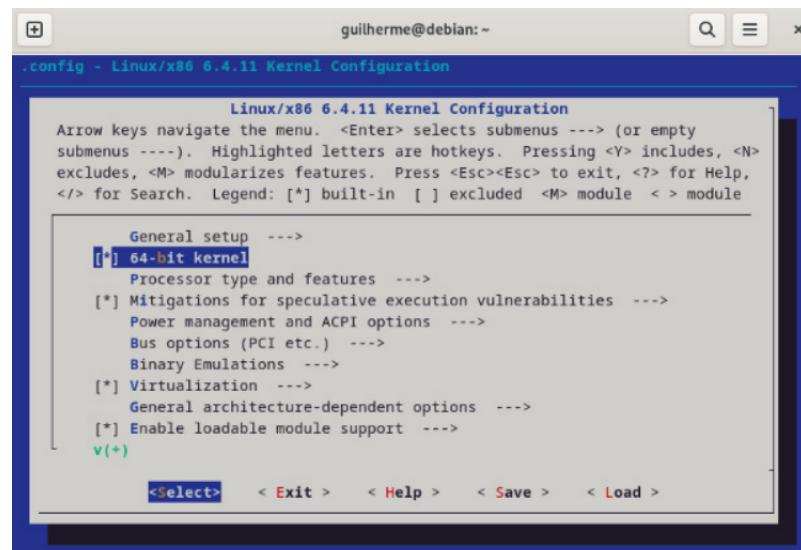


Figura 29 – Interface de configuração.

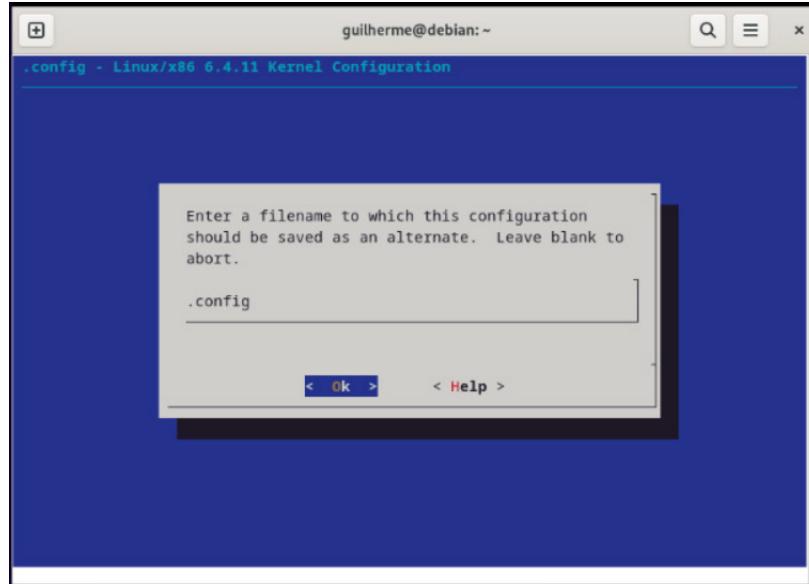
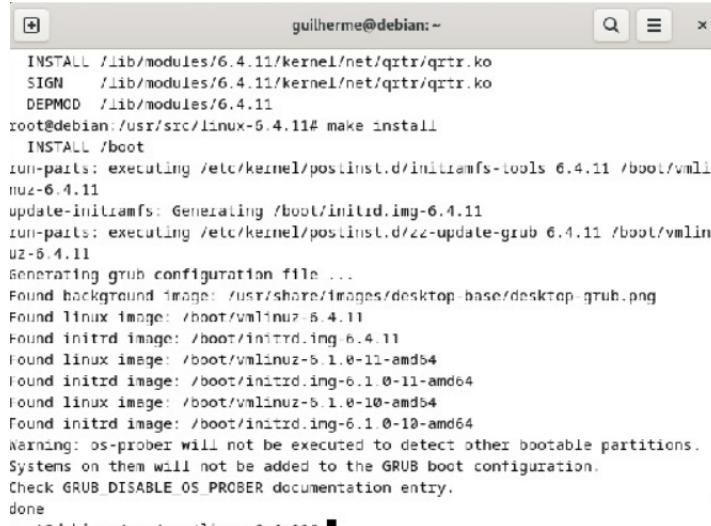


Figura 30 – Salvando as configurações realizadas.

Logo em seguida, necessita-se instalar alguns módulos, além disso, as configurações são compiladas e instaladas (Figura 31).

```
root@debian:/usr/src/linux-5.4.11# make modules_install
INSTALL /lib/modules/5.4.11/kernel/arch/x86/events/rapl.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/events/rapl.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/kernel/msr.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/kernel/msr.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/crypto/aesni-intel.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/crypto/aesni-intel.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/crypto/sha512-ssse3.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/crypto/sha512-ssse3.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/crypto/ghash-clmulni-intel.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/crypto/ghash-clmulni-intel.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/crypto/crc32c-intel.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/crypto/crc32c_intel.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/crypto/crc32_pclmul.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/crypto/crc32_pclmul.ko
INSTALL /lib/modules/5.4.11/kernel/arch/x86/crypto/crc32dif-pclmul.ko
SIGN  /lib/modules/5.4.11/kernel/arch/x86/crypto/crc32dif-pclmul.ko
INSTALL /lib/modules/5.4.11/kernel/kernel/configs.ko
SIGN  /lib/modules/5.4.11/kernel/kernel/configs.ko
INSTALL /lib/modules/5.4.11/kernel/fs/bintfm_misc.ko
SIGN  /lib/modules/5.4.11/kernel/fs/bintfm_misc.ko
INSTALL /lib/modules/5.4.11/kernel/fs/mbcache.ko
SIGN  /lib/modules/5.4.11/kernel/fs/mbcache.ko
```

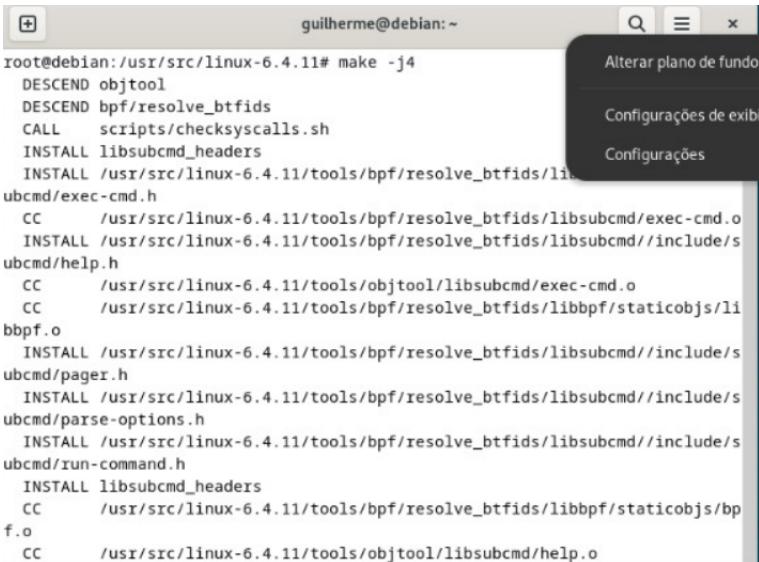
Figura 31 – Comando `make modules_install`.



```
guilherme@debian:~ guilherme@debian:~ 
INSTALL /lib/modules/6.4.11/kernel/net/qrtr/qrtr.ko
SIGN  /lib/modules/6.4.11/kernel/net/qrtr/qrtr.ko
DEPMOD /lib/modules/6.4.11
root@debian:/usr/src/linux-6.4.11# make install
  INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.4.11 /boot/vmlinuz-6.4.11
update-initramfs: Generating /boot/initrd.img-6.4.11
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.4.11 /boot/vmlinuz-6.4.11
Generating grub configuration file ...
Found background image: /usr/share/images/desktop_base/desktop-grub.png
Found linux image: /boot/vmlinuz-6.4.11
Found initrd image: /boot/initrd.img-6.4.11
Found linux image: /boot/vmlinuz-5.1.0-11-amd64
Found initrd image: /boot/initrd.img-5.1.0-11-amd64
Found linux image: /boot/vmlinuz-5.1.0-10-amd64
Found initrd image: /boot/initrd.img-5.1.0-10-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
```

Figura 32 – Comando `make modules_install make install`.

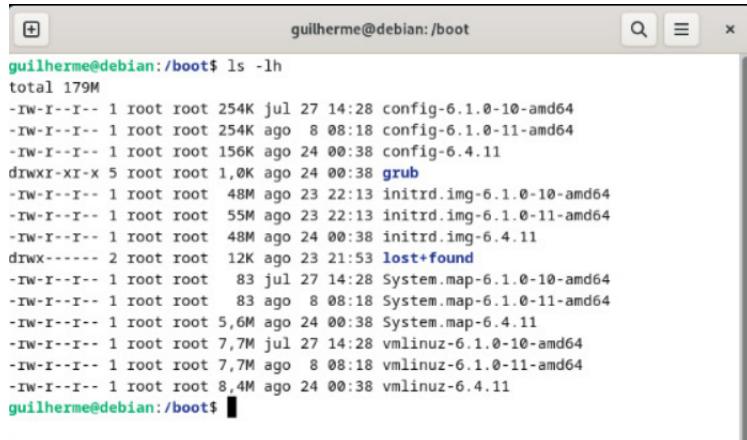
Ao finalizar, deve-se inicializar a compilação e a instalação considerando a existência de 4 núcleos disponíveis para a Máquina Virtual(Figura 33).



```
guilherme@debian:~ guilherme@debian:~ 
root@debian:/usr/src/linux-6.4.11# make -j4
  DESCEND objtool
  DESCEND bpf/resolve_btfids
  CALL scripts/checksyscalls.sh
  INSTALL libsubcmd_headers
  INSTALL /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libsubcmd/exec-cmd.h
  CC      /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libsubcmd/exec-cmd.o
  INSTALL /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libsubcmd/include/subcmd/help.h
  CC      /usr/src/linux-6.4.11/tools/objtool/libsubcmd/exec-cmd.o
  CC      /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libbpf/staticobjs/libbpf.o
  INSTALL /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libsubcmd/include/subcmd/pager.h
  INSTALL /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libsubcmd/include/subcmd/parse-options.h
  INSTALL /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libsubcmd/include/subcmd/run-command.h
  INSTALL libsubcmd_headers
  CC      /usr/src/linux-6.4.11/tools/bpf/resolve_btfids/libbpf/staticobjs/bpf.o
  CC      /usr/src/linux-6.4.11/tools/objtool/libsubcmd/help.o
```

Figura 33 – Comando `make -j4`.

Para verificar o tamanho dos arquivos deve-se utilizar o comando `ls -lh` no caminho `/boot` (Figura 34).



```
guilherme@debian:/boot$ ls -lh
total 179M
-rw-r--r-- 1 root root 254K jul 27 14:28 config-6.1.0-10-amd64
-rw-r--r-- 1 root root 254K ago  8 08:18 config-6.1.0-11-amd64
-rw-r--r-- 1 root root 156K ago 24 00:38 config-6.4.11
drwxr-xr-x 5 root root 1,0K ago 24 00:38 grub
-rw-r--r-- 1 root root 48M ago 23 22:13 initrd.img-6.1.0-10-amd64
-rw-r--r-- 1 root root 55M ago 23 22:13 initrd.img-6.1.0-11-amd64
-rw-r--r-- 1 root root 48M ago 24 00:38 initrd.img-6.4.11
drwx----- 2 root root 12K ago 23 21:53 lost+found
-rw-r--r-- 1 root root  83 jul 27 14:28 System.map-6.1.0-10-amd64
-rw-r--r-- 1 root root  83 ago  8 08:18 System.map-6.1.0-11-amd64
-rw-r--r-- 1 root root 5,6M ago 24 00:38 System.map-6.4.11
-rw-r--r-- 1 root root 7,7M jul 27 14:28 vmlinuz-6.1.0-10-amd64
-rw-r--r-- 1 root root 7,7M ago  8 08:18 vmlinuz-6.1.0-11-amd64
-rw-r--r-- 1 root root 8,4M ago 24 00:38 vmlinuz-6.4.11
guilherme@debian:/boot$
```

Figura 34 – Comando **ls -lh**.

Além disso, para verificar o tamanho do diretório deve-se utilizar o comando **du -sh** no caminho `/lib/modules`(Figura 35).



```
guilherme@debian:/lib/modules$ du . -sh
933M .
guilherme@debian:/lib/modules$
```

Figura 35 – Comando **du -sh**. Tamanho da pasta com os módulos compilados.

6 Conclusões

Em suma, o SO GNU/Linux oferece diversas vantagens para o usuário, sendo elas a flexibilidade, a segurança, a estabilidade e a compatibilidade com vários tipos de hardware e software. Além disso, o GNU/Linux proporciona uma oportunidade de aprendizado e diversas ferramentas que facilitam na criação, edição e testes para diversos sistemas, software e também facilita no gerenciamento de redes. Para usufruir de suas vantagens, o usuário precisa ter conhecimentos básicos sobre os comandos e as ferramentas disponíveis no terminal.

7 Referências

MAZIERO, C. *Sistemas Operacionais: Conceitos e Mecanismos*. 2019. Disponível em: <<https://wiki.inf.ufpr.br/maziero/doku>>

.php?id=socm:start><https://wiki.inf.ufpr.br/maziero/doku.php?id=socm:start>.
Citado na página 4.

ORGANIZATION, I. L. K. *The Linux Kernel Archives*. 2023. Disponível em:
<<https://www.kernel.org/>><https://www.kernel.org/>. Citado na página 18.