

Instalação GNU/Linux e compilação do núcleo

Trabalho da disciplina de Sistemas
Operacionais

**Caio Miglioli, Caio Eduardo Theodoro, Alexandre
Scrocaro**



Coordenação do Curso de Bacharelado em Ciência da Computação -
COCIC

Universidade Tecnológica Federal do Paraná - UTFPR
Campo Mourão, Paraná, Brasil

1 INTRODUÇÃO

O trabalho tem como objetivo fazer um tutorial passo a passo para a realização da instalação do GNU/Linux da distribuição Debian, como também para a compilação do kernel do linux, no caso a versão 5.12.12.

2 CONFIGURAÇÕES UTILIZADAS NO PROCEDIMENTO

Hardware:

Memoria RAM: 8 GB

Processador: Intel® Core™ i5-7200U CPU @ 2.50GHz × 4

Sistema 64 bits

Distribuição:

Debian 10.9.0

Kernel compilado: linux-5.12.12

3 PROCEDIMENTOS DE INICIALIZAÇÃO PARA A VM

Primeiramente, foi usado o Virtual Box para virtualizar o SO usado. Após instalado o Virtual Box, foi inicializada a Virtual Machine com a configuração ilustrada na Figura 1.

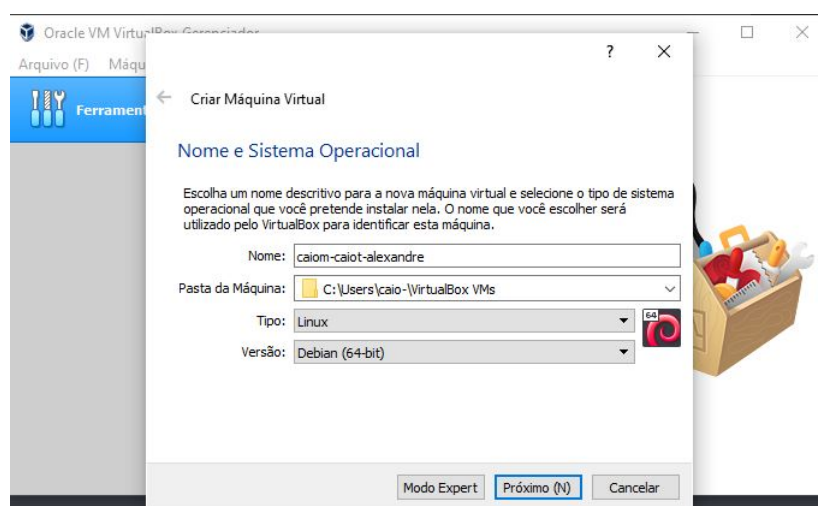


Figura 1 – Configuração de criação da VM

A VM utilizada foi configurada utilizando 2GB de memória RAM, 50GB de espaço no disco rígido e utilizando 2 núcleos do processador.

Após criada a VM, no **Seletor de Discos Ópticos** foi inserido a imagem ISO do sistema Debian 10.9.0, como ilustrada da Figura 2.

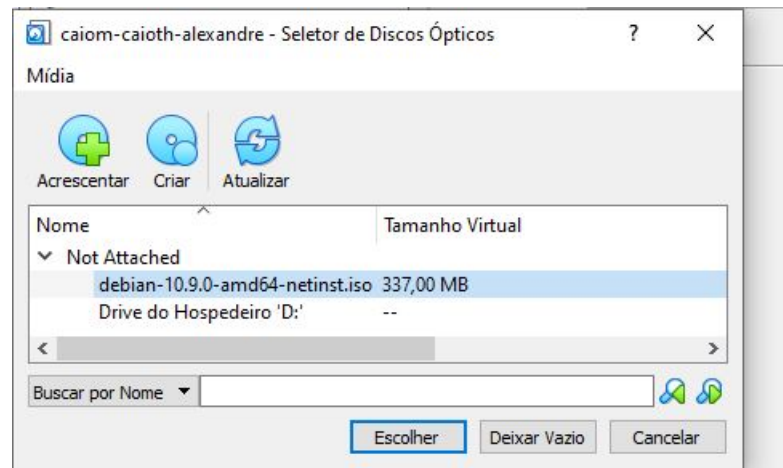


Figura 2 – ISO Sistema Debian

Feito isso, foi iniciada a máquina. Usamos a instalação gráfica para contemplar o ambiente gráfico e outros pacotes. No geral foram escolhidas as linguagens, horario, hostname, username e etc, que não convém serem detalhadas para nosso objetivo. Deixamos por padrão as escolhidas pelo sistema, contemplando somente linguagem e fuso-horario Português-BR.

4 PARTICIONAMENTO

Para o particionamento, foram seguidas algumas instruções recomendadas pelo próprio site do Debian, combinadas com sugestões fornecidas pelo professor.

/ : Também chamada de partição root, é onde /etc, /bin, /sbin, /lib e /dev são alocados, no nosso caso foram usados 10GB.

/usr: Partição que contém todos os programas dos utilizadores (/usr/bin), bibliotecas (/usr/lib), documentação (/usr/share/doc) e etc. Um tamanho razoável para esta partição seria de 5 a 6GB, como em nosso caso temos bastante espaço de disco, foi utilizado 30GB.

/var: Partição que armazena os dados temporários, tais como artigos de news, e-mails, web sites, bases de dados, cache do sistema de pacotes, etc. Um tamanho razoável para esse espaço seria de 2 a 4GB, optamos por utilizar 2GB.

/home: Partição que tem por finalidade armazenar os arquivos pessoais de cada usuário da máquina. O aconselhável para essa partição seria receber o maior tamanho possível, mas como esta máquina está configurada para fins de testes e aprendizado, optamos por, para evitar quaisquer problemas, alocar 5GB.

/swap: Também chamada de partição de troca, é onde a memoria RAM aloca seus dados quando está sem espaço. No nosso caso foram utilizados 2GB, mesma quantidade usada para a RAM.

SCSI3 (0,0,0) (sda) - 53.7 GB ATA VBOX HARDDISK						
>	#1	primária	10.0 GB	f	ext4	/
>	#5	lógica	30.0 GB	f	ext4	/usr
>	#6	lógica	2.0 GB	f	ext4	/var
>	#7	lógica	5.0 GB	f	ext4	/home
>	#8	lógica	2.0 GB	f	swap	swap
>		pri/lóg	4.7 GB		ESPAÇO LIVRE	

Figura 3 – Tela de particionamento de disco.

5 EXECUÇÃO E SAÍDA DOS COMANDOS DE VERIFICAÇÃO DA PARTE 1 DAS ATIVIDADES

5.1 Comando "ps aux":

Lista todos os processos em execução na máquina, como mostrado na Figura 4.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.3	1.0	104828	10184	?	Ss	17:51	0:01	/sbin/init
root	2	0.0	0.0	0	0	?	S	17:51	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	17:51	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	17:51	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	17:51	0:00	[kworker/0:0H-k
root	8	0.0	0.0	0	0	?	I<	17:51	0:00	[mm_percpu_wq]
root	9	0.0	0.0	0	0	?	S	17:51	0:00	[ksoftirqd/0]
root	10	0.0	0.0	0	0	?	I	17:51	0:00	[rcu_sched]
root	11	0.0	0.0	0	0	?	I	17:51	0:00	[rcu_bh]
root	12	0.0	0.0	0	0	?	S	17:51	0:00	[migration/0]
root	13	0.0	0.0	0	0	?	I	17:51	0:00	[kworker/0:1-at
root	14	0.0	0.0	0	0	?	S	17:51	0:00	[cpuhp/0]
root	15	0.0	0.0	0	0	?	S	17:51	0:00	[kdevtmpfs]
root	16	0.0	0.0	0	0	?	I<	17:51	0:00	[netns]
root	17	0.0	0.0	0	0	?	S	17:51	0:00	[kauditd]
root	18	0.0	0.0	0	0	?	S	17:51	0:00	[khungtaskd]
root	19	0.0	0.0	0	0	?	S	17:51	0:00	[oom_reaper]
root	20	0.0	0.0	0	0	?	I<	17:51	0:00	[writeback]
root	21	0.0	0.0	0	0	?	S	17:51	0:00	[kcompactd0]
root	22	0.0	0.0	0	0	?	SN	17:51	0:00	[ksmd]
root	23	0.0	0.0	0	0	?	SN	17:51	0:00	[khugepaged]
root	24	0.0	0.0	0	0	?	I<	17:51	0:00	[crypto]
root	25	0.0	0.0	0	0	?	I<	17:51	0:00	[kintegrityd]
root	26	0.0	0.0	0	0	?	I<	17:51	0:00	[kblockd]
root	27	0.0	0.0	0	0	?	I<	17:51	0:00	[edac-poller]
root	28	0.0	0.0	0	0	?	I<	17:51	0:00	[devfreq_wq]
root	29	0.0	0.0	0	0	?	S	17:51	0:00	[watchdog]
root	30	0.0	0.0	0	0	?	S	17:51	0:00	[kswapd0]
root	48	0.0	0.0	0	0	?	I<	17:51	0:00	[kthrotld]
root	49	0.0	0.0	0	0	?	I<	17:51	0:00	[ipvs_addrconf]
root	59	0.0	0.0	0	0	?	I<	17:51	0:00	[kstrp]
root	104	0.0	0.0	0	0	?	I<	17:51	0:00	[ata_sff]
root	107	0.0	0.0	0	0	?	S	17:51	0:00	[scsi_eh_0]
root	108	0.0	0.0	0	0	?	I<	17:51	0:00	[scsi_tmf_0]
root	110	0.0	0.0	0	0	?	S	17:51	0:00	[scsi_eh_1]
root	112	0.0	0.0	0	0	?	I<	17:51	0:00	[scsi_tmf_1]
root	113	0.0	0.0	0	0	?	I	17:51	0:00	[kworker/u2:2-e
root	117	0.0	0.0	0	0	?	S	17:51	0:00	[scsi_eh_2]
root	118	0.0	0.0	0	0	?	I	17:51	0:00	[kworker/u2:3-e
root	134	0.0	0.0	0	0	?	I	17:51	0:00	[kworker/0:3-at
root	135	0.0	0.0	0	0	?	I	17:51	0:00	[tts_swap]
root	137	0.0	0.0	0	0	?	S	17:51	0:00	[irq/18-vmwgfx]
root	156	0.0	0.0	0	0	?	I<	17:51	0:00	[kworker/0:1H-k
root	193	0.0	0.0	0	0	?	I<	17:51	0:00	[kworker/u3:0]
root	195	0.0	0.0	0	0	?	S	17:51	0:00	[jbd2/sda1-8]
root	196	0.0	0.0	0	0	?	I<	17:51	0:00	[ext4-rsv-conve
root	208	0.0	0.0	0	0	?	S	17:52	0:00	[jbd2/sda6-8]
root	209	0.0	0.0	0	0	?	I<	17:52	0:00	[ext4-rsv-conve

Figura 4 – ps aux

Nota-se a primeira coluna mostrando o usuário (root para sistema e caiot-c+ para nosso usuário), a segunda coluna mostrando o ID do processo, seguido da porcentagem de uso de CPU e de memória RAM.

5.2 Comando "df -h":

Lista as partições do disco rígido. Seus tamanhos foram explicados na sessão 4. Sua saída está mostrada na Figura 5.

```
caiot-caiom-alexandre@debian:/$ df
Sist. Arq.      Blocos de 1K  Usado  Disponível  Uso%  Montado em
udev            485636      0    485636     0%  /dev
tmpfs           101108    4656    96452     5%  /run
/dev/sda1       28704676  257692  26965820   1%  /
/dev/sda6       9545920  6415576  2625720   71%  /usr
tmpfs           505528      0    505528     0%  /dev/shm
tmpfs           5120        4     5116     1%  /run/lock
tmpfs           505528      0    505528     0%  /sys/fs/cgroup
/dev/sda8       40466688  138084  38243280   1%  /home
/dev/sda7       9545920  357152  8684144    4%  /var
tmpfs           101104     12    101092     1%  /run/user/1000
```

Figura 5 – df -h

5.3 Comando "free -b":

Descreve a memória RAM disponível, na Figura 6 é mostrado a memória total e a memória de SWAP.

```
caiot-caiom-alexandre@debian:/$ free -b
              total          used         free       shared  buff/cache   available
Mem:      1035325440  292392960  408670208    6496256   334262272   593031168
Swap:      14998827008      0  14998827008
```

Figura 6 – free -b

5.4 Comando "cat /proc/meminfo":

Descreve a memória RAM disponível, porém de forma mais detalhada. A Figura 7 ilustra todas as divisões de memória do sistema.

```
caiot-caiom-alexandre@debian:/$ cat /proc/meminfo
MemTotal:      1011060 kB
MemFree:       399236 kB
MemAvailable:  579236 kB
Buffers:       28544 kB
Cached:        272796 kB
SwapCached:    0 kB
Active:        318796 kB
Inactive:      195680 kB
Active(anon):  213736 kB
Inactive(anon): 5728 kB
Inactive(file): 105060 kB
Unevictable:   96 kB
Mlocked:       96 kB
SwapTotal:     14647292 kB
SwapFree:      14647292 kB
Dirty:         0 kB
Writeback:     0 kB
AnonPages:     203528 kB
Mapped:        118956 kB
Shmem:         6344 kB
Slab:          48696 kB
SReclaimable:  25040 kB
SUnreclaim:    23656 kB
KernelStack:   3724 kB
PageTables:     6188 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:   15152820 kB
Committed_AS:  1492276 kB
VmallocTotal:  34359738367 kB
VmallocUsed:   0 kB
VmallocChunk:  0 kB
Percpu:        324 kB
HardwareCorrupted: 0 kB
AnonHugePages: 81920 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize:  2048 kB
Hugetlb:       0 kB
DirectMap4k:   83904 kB
DirectMap2M:   964608 kB
```

Figura 7 – cat /proc/meminfo

5.5 Verificações de rede

5.5.1 Comando "ip address show":

Mostra as informações relacionadas ao IP, na Figura 8 podemos ver o IP localhost (127.0.0.1/8) e o IP privado (10.0.2.255).

```
caiot-caiom-alexandre@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:d3:5d:d7 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 84259sec preferred_lft 84259sec
    inet6 fe80::a00:27ff:fed3:5dd7/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
caiot-caiom-alexandre@debian:/$
```

Figura 8 – ip address show

5.5.2 Comando "ip route":

Mostra a rota do IP, na Figura 9 podemos ver IPs privados da máquina para esta função.

```
caiot-caiom-alexandre@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
169.254.0.0/16 dev enp0s3 scope link metric 1000
```

Figura 9 – ip route

5.5.3 Comando "cat /etc/resolv.conf":

Mostra o conteúdo do arquivo **resolv.conf** (servidor de nomes, no caso 198.168.100.1). Ilustrado na Figura 10.

```
caiot-caiom-alexandre@debian:/$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.100.1
```

Figura 10 – cat /etc/resolv.conf

5.5.4 Comando "cat /etc/network/interfaces":

Mostra o conteúdo do arquivo **interfaces**, que descreve as interfaces de rede do sistema. Este arquivo descreve os parâmetros de rede como endereço IP, Máscara de Sub-Rede, DHCP, etc, e também é usado para defini-los. Sua saída está mostrada na Figura 11.


```

caiot-caiom-alexandre@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

```

Figura 11 – cat /etc/network/interface

5.5.5 Comando "ping www.google.com.br":

Serve para calcular o tempo de resposta para um servidor, neste caso o **www.google.com.br**. É utilizado também para testar a conexão a fim de saber se a rede local está conectada à internet. Sua saída está mostrada na Figura 12.

```

caiot-caiom-alexandre@debian:~$ ping www.google.com.br
PING www.google.com.br (172.217.30.163) 56(84) bytes of data.
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=1 ttl=118 time=22.2 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=2 ttl=118 time=22.6 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=3 ttl=118 time=22.3 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=4 ttl=118 time=21.3 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=5 ttl=118 time=22.9 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=6 ttl=118 time=21.2 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=7 ttl=118 time=20.8 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=8 ttl=118 time=21.1 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=9 ttl=118 time=22.0 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=10 ttl=118 time=22.3 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=11 ttl=118 time=22.1 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=12 ttl=118 time=21.4 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=13 ttl=118 time=21.8 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=14 ttl=118 time=22.1 ms
64 bytes from eze03s36-in-f3.1e100.net (172.217.30.163): icmp_seq=15 ttl=118 time=21.3 ms

```

Figura 12 – ping www.google.com.br

5.6 Comando "cd /etc/apt || nano sources.list":

Área de edição de arquivos no terminal, seu funcionamento é semelhante ao do Bloco de Notas do Windows (no caso, do **arquivosource.list**). Sua saída está mostrada na Figura 13.

```

# buster-updates, previously known as 'volatile'
deb http://ftp.br.debian.org/debian/ buster-updates main
deb-src http://ftp.br.debian.org/debian/ buster-updates main

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
[ 21 linhas lidas ]
^G Ajuda      ^O Gravar     ^W Onde está? ^K Recort txt  ^J Justificar
^X Sair       ^R Ler o arq  ^\ Substituir  ^U Colar txt  ^T Verif0rtog

```

Figura 13 – nano sources.list

5.7 Comando "apt-get update":

Verifica os pacotes do sistema a serem atualizados. Este comando é normalmente utilizado em conjunto com o comando "apt-get upgrade", que instala os pacotes encontrados pelo "apt-get update". Sua saída está mostrada na Figura 14.

```
root@debian:/etc/apt# apt-get update
Atingido:1 http://ftp.br.debian.org/debian buster InRelease
Atingido:2 http://security.debian.org/debian-security buster/updates InRelease
Atingido:3 http://ftp.br.debian.org/debian buster-updates InRelease
Lendo listas de pacotes... Pronto
root@debian:/etc/apt#
```

Figura 14 – nano sources.list

5.8 Comando "uname -a":

Verifica o sistema operacional, versão do kernel, arquitetura da máquina e etc, como ilustrado na Figura 15.

```
caiom-caiot-alexandre@debian:~$ uname -a
Linux debian 4.19.0-17-amd64 #1 SMP Debian 4.19.194-1 (2021-06-10) x86_64 GNU/
Linux
```

Figura 15 – uname -a

6 EXECUÇÃO E SAÍDA DOS COMANDOS DE TESTE DA PARTE 1 DAS ATIVIDADES

6.1 Comando "clear":

Limpa o terminal, removendo informação digitada ou mostrada anteriormente.

6.2 Comando "ls -l":

Lista por extenso todos os itens guardados no diretório em questão. Sua saída está mostrada na Figura 16.

```
caiom-caiot-alexandre@debian:~$ ls -l
total 32
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 'Área de trabalho'
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Documentos
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Downloads
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Imagens
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Modelos
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Música
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Público
drwxr-xr-x 2 caiom-caiot-alexandre caiom-caiot-alexandre 4096 jun 21 19:08 Vídeos
```

Figura 16 – ls -l

6.3 Comando "cd <diretorio>":

Entra para o diretório selecionado através do argumento enviado junto do comando. No caso da Figura 17, o terminal executará comandos no diretório **Área de trabalho**.

```
caiom-caiot-alexandre@debian:~$ cd Área\ de\ trabalho/  
caiom-caiot-alexandre@debian:~/Área de trabalho$
```

Figura 17 – cd Área de trabalho

6.4 Comando "cat <nomeArquivo>":

Lê um arquivo e o mostra no terminal. Sua saída está mostrada na Figura 18.

```
caiom-caiot-alexandre@debian:~/Área de trabalho$ cat teste  
testando 123caiom-caiot-alexandre@debian:~/Área de trabalho$
```

Figura 18 – cat

6.5 Comando "rm <nomeArquivo>":

Remove um arquivo cujo nome é especificado pelo argumento do comando.

6.6 Comando "pico"ou "nano":

Abre o editor de arquivos no terminal, é o mesmo comando usado na Figura 13, ou seja, tem as mesmas funções. Sua saída está ilustrada na Figura 19.

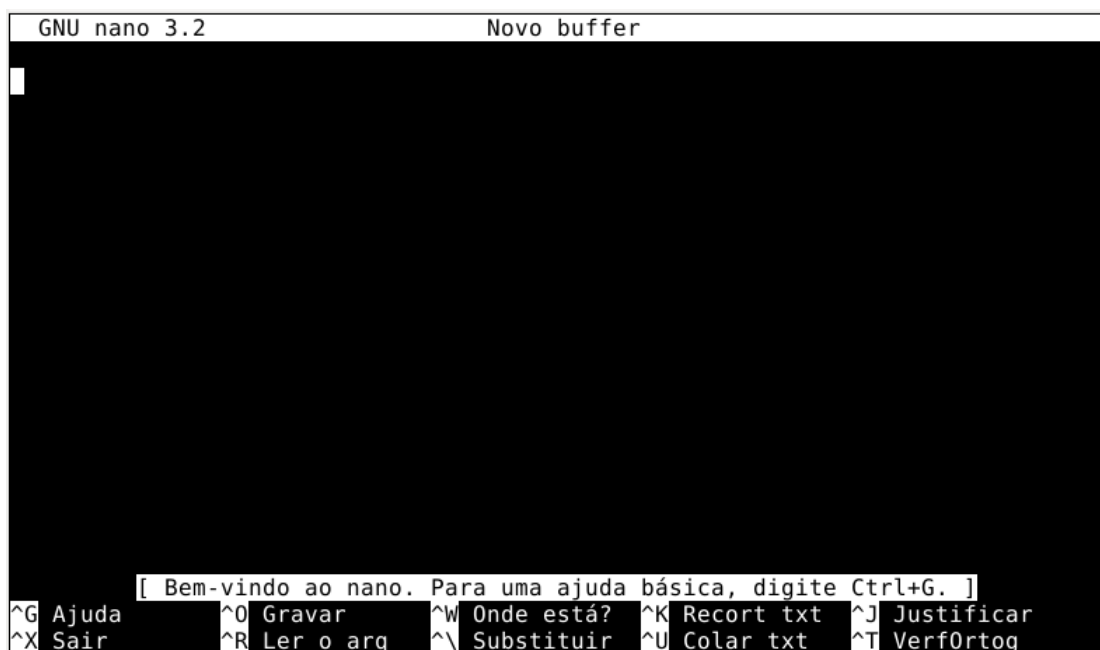


Figura 19 – pico

6.7 Comando "cp <arquivoOrigem> <destinoCopiaArquivo>":

Realiza uma cópia de um arquivo. No caso da Figura 20, foi realizado uma cópia do arquivo teste cujo nome é teste2.

```
caiom-caiot-alexandre@debian:~/Área de trabalho$ cp teste teste2
caiom-caiot-alexandre@debian:~/Área de trabalho$
```

Figura 20 – cp

6.8 Comando "grep <string> <nomeArquivo>":

Procura padrões em um arquivo, ou seja, informa se determinado conteúdo existe dentro de um arquivo, como por exemplo, uma palavra. Sua saída está mostrada na Figura 21.

```
caiom-caiot-alexandre@debian:~$ grep "teste" t.txt
oi teste oi oi oi
caiom-caiot-alexandre@debian:~$ grep "tde" t.txt
caiom-caiot-alexandre@debian:~$
```

Figura 21 – grep

6.9 Comando "head <nomeArquivo>":

Exibe a head (primeira linha) de um arquivo. No caso da Figura 22, a primeira linha do arquivo é "testando 1 2 3".

```
caiot-caiom-alexandre@debian:~/Área de trabalho$ head teste.txt
testando 1 2 3
caiot-caiom-alexandre@debian:~/Área de trabalho$
```

Figura 22 – head

6.10 Comando "tail <nomeArquivo>":

Exibe a tail (últimas linhas) de um arquivo. No caso do exemplo da Figura 23, as últimas linhas são "arquivo 1".

```
caiom-caiot-alexandre@debian:~/Área de trabalho$ tail teste
arquivo 1
```

Figura 23 – tail

6.11 Comando "mv <arquivoOrigem> <novoDestinoOuNome>":

Move ou renomeia arquivos. No exemplo da Figura 24, foi movido o arquivo "teste" para o diretório "testemv".

```
caiom-caiot-alexandre@debian:~/Área de trabalho$ mv teste testemv
```

Figura 24 – mv

7 CONFIGURAÇÃO, COMPILAÇÃO E INSTALAÇÃO DO KERNEL

5.12.12

7.1 Parte 1: Procedimentos Iniciais

Para a instalação do novo kernel, foi necessário realizar alguns passos iniciais, o primeiro deles foi entrar como um Super Usuário para se ter permissão de alterar os arquivos necessários. Para isto foi utilizado o comando **'sudo su'**.

Com as permissões confirmadas, foi instalado alguns pacotes que foram necessários para a instalação do novo Kernel. Com o comando **'cd /root'** acessamos o diretório root e instalamos os pacotes necessários com o comando **'apt-get install build-essential libncurses5-dev xz-utils libssl-dev bc bison libelf-dev libncurses5-dev flex'**. Em uma das tentativas surgiu a necessidade de instalar o pacote **flex**, por isso ele foi adicionado a esse comando.

```
root@debian:/home/caiom-caiot-alexandre# apt-get install build-essential libncurses5-dev xz-utils libssl-dev bc bison libelf-dev li
bncurses5-dev flex
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
xz-utils is already the newest version (5.2.4-1).
xz-utils configurado para instalar manualmente.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-8 gcc gcc-8
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libbinutils
  libbison-dev libc-dev-bin libc6-dev libcc1-0 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfl-dev libfl2 libgcc-8-dev libitm1 liblsan0 libmpx2 libncurses-dev libsigsegv2
  libstdc++-8-dev libtsan0 libubsan1 linux-libc-dev m4 make manpages-dev patch zlibg-dev
Pacotes sugeridos:
  binutils-doc bison-doc debian-keyring flex-doc g++-multilib g++-8-multilib gcc-8-doc
  libstdc++6-8-dbg gcc-multilib autoconf automake libtool gdb gcc-doc gcc-8-multilib
  gcc-8-locales libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan5-dbg liblsan0-dbg
  libtsan0-dbg libubsan1-dbg libmpx2-dbg libquadmath0-dbg glibc-doc git bzr ncurses-doc
  libssl-doc libstdc++-8-doc m4-doc make-doc ed diffutils-doc
Os NOVOS pacotes a seguir serão instalados:
  bc binutils binutils-common binutils-x86-64-linux-gnu bison build-essential dpkg-dev fakeroot
  flex g++ g++-8 gcc gcc-8 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan5 libbinutils libbison-dev libc-dev-bin libc6-dev libcc1-0
  libdpkg-perl libelf-dev libfakeroot libfile-fcntllock-perl libfl-dev libfl2 libgcc-8-dev
  libitm1 liblsan0 libmpx2 libncurses-dev libncurses5-dev libsigsegv2 libssl-dev libstdc++-8-dev
  libtsan0 libubsan1 linux-libc-dev m4 make manpages-dev patch zlibg-dev
0 pacotes atualizados, 45 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 43,8 MB de arquivos.
Depois desta operação, 172 MB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] s
Obter:1 http://ftp.br.debian.org/debian buster/main amd64 libsigsegv2 amd64 2.12-2 [32,8 kB]
Obter:2 http://ftp.br.debian.org/debian buster/main amd64 m4 amd64 1.4.18-2 [203 kB]
Obter:3 http://ftp.br.debian.org/debian buster/main amd64 flex amd64 2.6.4-6.2 [456 kB]
Obter:4 http://ftp.br.debian.org/debian buster/main amd64 bc amd64 1.07.1-2+h1 [189 kB]
```

Figura 25 – Instalação de pacotes necessários.

7.2 Parte 2: Obtendo os Arquivos do Kernel

Para realizar o download da nova versão do Kernel (5.12.12), foi utilizado o comando **'wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.12.tar.xz'**.

```

root@debian:/home/caiom-caiot-alexandre# wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.12.tar.xz
--2021-06-23 17:18:45-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.12.tar.xz
Resolvendo cdn.kernel.org (cdn.kernel.org)... 151.101.1.176, 151.101.65.176, 151.101.129.176, ...
Conectando-se a cdn.kernel.org (cdn.kernel.org)[151.101.1.176]:443... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 118165564 (113M) [application/x-xz]
Salvando em: "linux-5.12.12.tar.xz"

linux-5.12.12.tar.xz          100%[=====] 112,69M  3,20MB/s   em 21s
2021-06-23 17:19:07 (5,33 MB/s) - "linux-5.12.12.tar.xz" salvo [118165564/118165564]

```

Figura 26 – Instalação da nova versão do Kernel (5.12.12).

Por estar em formato compactado (tar.xz), foi necessário primeiramente descompactar, para isso foi utilizado o comando `'tar xvf linux-5.12.12.tar.xz -C /usr/src'`, que enviou os arquivos descompactados para a pasta `'/usr/src'`.

```

linux-5.12.12/tools/vm/slabinfo.c
linux-5.12.12/tools/wmi/
linux-5.12.12/tools/wmi/Makefile
linux-5.12.12/tools/wmi/dell-smbios-example.c
linux-5.12.12/usr/
linux-5.12.12/usr/.gitignore
linux-5.12.12/usr/Kconfig
linux-5.12.12/usr/Makefile
linux-5.12.12/usr/default_cpio_list
linux-5.12.12/usr/gen_init_cpio.c
linux-5.12.12/usr/gen_initramfs.sh
linux-5.12.12/usr/include/
linux-5.12.12/usr/include/.gitignore
linux-5.12.12/usr/include/Makefile
linux-5.12.12/usr/initramfs_data.S
linux-5.12.12/virt/
linux-5.12.12/virt/Makefile
linux-5.12.12/virt/kvm/
linux-5.12.12/virt/kvm/Kconfig
linux-5.12.12/virt/kvm/async_pf.c
linux-5.12.12/virt/kvm/async_pf.h
linux-5.12.12/virt/kvm/coalesced_mmio.c
linux-5.12.12/virt/kvm/coalesced_mmio.h
linux-5.12.12/virt/kvm/dirty_ring.c
linux-5.12.12/virt/kvm/eventfd.c
linux-5.12.12/virt/kvm/irqchip.c
linux-5.12.12/virt/kvm/kvm_main.c
linux-5.12.12/virt/kvm/mmu_lock.h
linux-5.12.12/virt/kvm/vfio.c
linux-5.12.12/virt/kvm/vfio.h
linux-5.12.12/virt/lib/
linux-5.12.12/virt/lib/Kconfig
linux-5.12.12/virt/lib/Makefile
linux-5.12.12/virt/lib/irqbypass.c
root@debian:/home/caiom-caiot-alexandre#

```

Figura 27 – Descompactação do arquivo.

7.3 Parte 3: Configurando os Módulos do Novo Kernel

Para iniciar a configuração, primeiro fomos até o diretório que os arquivos estão localizados utilizando o comando `'cd /usr/src/linux-5.12.12'`. A fim de utilizar somente os módulos mínimos para a utilização do sistema, utilizamos o comando `'make localmodconfig'` onde apertando a tecla **enter** para todas as solicitações inserimos as configurações mínimas recomendadas pelo instalador. Nesta instalação não foi utilizado o comando `'make menuconfig'`.

```

root@debian:/usr/src/linux-5.12.12# 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/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-4.19.0-17-amd64'
aes_x86_64 config not found!!
glue_helper config not found!!
System keyring enabled but keys "debian/certs/debian-uefi-certs.pem" not found. Resetting keys to default value.
.config:7683:warning: symbol value 'm' invalid for FS_ENCRYPTION
*
* Restart config...
*
*
* General setup
*
Compile also drivers which will not load (COMPILE_TEST) [N/y/?] n
Local version - append to kernel release (LOCALVERSION) []
Automatically append version information to the version string (LOCALVERSION_AUTO) [N/y/?] n
Build ID Salt (BUILD_SALT) [4.19.0-17-amd64] 4.19.0-17-amd64
Kernel compression mode
  1. Gzip (KERNEL_GZIP)
  2. Bzip2 (KERNEL_BZIP2)
  3. LZMA (KERNEL_LZMA)
> 4. XZ (KERNEL_XZ)
  5. LZ0 (KERNEL_LZ0)
  6. LZ4 (KERNEL_LZ4)
  7. ZSTD (KERNEL_ZSTD)

```

Figura 28 – Comando make localmodconfig.

7.4 Parte 4: Compilando e Instalando

Após realizarmos todas as configurações necessárias, compilamos os módulos utilizando os comandos 'make -j4' e 'make modules'.

```

root@debian:/usr/src/linux-5.12.12# make -j4
SYSHDR arch/x86/include/generated/uapi/asm/unistd_32.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_64.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_x32.h
WRAP arch/x86/include/generated/uapi/asm/bpf_perf_event.h
WRAP arch/x86/include/generated/uapi/asm/errno.h
WRAP arch/x86/include/generated/uapi/asm/fcntl.h
WRAP arch/x86/include/generated/uapi/asm/ioctl.h
WRAP arch/x86/include/generated/uapi/asm/ioctls.h
WRAP arch/x86/include/generated/uapi/asm/ipcbuf.h
WRAP arch/x86/include/generated/uapi/asm/param.h
WRAP arch/x86/include/generated/uapi/asm/poll.h
WRAP arch/x86/include/generated/uapi/asm/resource.h
WRAP arch/x86/include/generated/uapi/asm/socket.h
WRAP arch/x86/include/generated/uapi/asm/sockios.h
WRAP arch/x86/include/generated/uapi/asm/termbits.h
WRAP arch/x86/include/generated/uapi/asm/termios.h
WRAP arch/x86/include/generated/uapi/asm/types.h
HOSTCC arch/x86/tools/relocs_32.o
SYSTBL arch/x86/include/generated/asm/syscalls_32.h
SYSHDR arch/x86/include/generated/asm/unistd_32_ia32.h
SYSHDR arch/x86/include/generated/asm/unistd_64_x32.h
SYSTBL arch/x86/include/generated/asm/syscalls_64.h

```

Figura 29 – Comando make

E por fim instalamos o novo Kernel utilizando os comandos 'make modules_install' e 'make install'.

```

root@debian:/usr/src/linux-5.12.12# make modules_install
INSTALL arch/x86/crypto/aesni-intel.ko
INSTALL arch/x86/crypto/crc32-pclmul.ko
INSTALL arch/x86/crypto/crc32c-intel.ko
INSTALL arch/x86/crypto/crct10dif-pclmul.ko
INSTALL arch/x86/crypto/ghash-clmulni-intel.ko
INSTALL block/t10-pi.ko
INSTALL crypto/crc32c-generic.ko
INSTALL crypto/cryptd.ko
INSTALL crypto/crypto-simd.ko
INSTALL crypto/ecb.ko
INSTALL drivers/acpi/ac.ko
INSTALL drivers/acpi/battery.ko
INSTALL drivers/acpi/button.ko
INSTALL drivers/acpi/video.ko
INSTALL drivers/ata/acard-ahci.ko
INSTALL drivers/ata/ahci.ko
INSTALL drivers/ata/ata-generic.ko
INSTALL drivers/ata/ata_piix.ko
INSTALL drivers/ata/libahci.ko
INSTALL drivers/ata/libata.ko
INSTALL drivers/cdrom/cdrom.ko
INSTALL drivers/gpu/drm/drm.ko
INSTALL drivers/gpu/drm/drm_kms_helper.ko
INSTALL drivers/gpu/drm/ttm/ttm.ko
INSTALL drivers/gpu/drm/vmwgfx/vmwgfx.ko
INSTALL drivers/hid/hid-generic.ko
INSTALL drivers/hid/hid.ko
INSTALL drivers/hid/usbhid/usbhid.ko
INSTALL drivers/i2c/algos/i2c-algo-bit.ko
INSTALL drivers/i2c/busses/i2c-piix4.ko
INSTALL drivers/input/evdev.ko
INSTALL drivers/input/joydev.ko

```

Figura 30 – Comando make modules_install.

```

root@debian:/usr/src/linux-5.12.12# make install
sh ./arch/x86/boot/install.sh 5.12.12 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.12.12 /boot/vmlinuz-5.12.12
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.12.12 /boot/vmlinuz-5.12.12
update-initramfs: Generating /boot/initrd.img-5.12.12
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.12.12 /boot/vmlinuz-5.12.12
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Imagem Linux encontrada: /boot/vmlinuz-5.12.12
Imagem initrd encontrada: /boot/initrd.img-5.12.12
Imagem Linux encontrada: /boot/vmlinuz-4.19.0-17-amd64
Imagem initrd encontrada: /boot/initrd.img-4.19.0-17-amd64
Imagem Linux encontrada: /boot/vmlinuz-4.19.0-16-amd64
Imagem initrd encontrada: /boot/initrd.img-4.19.0-16-amd64
concluido

```

Figura 31 – Comando make install.

8 VERIFICANDO OS ARQUIVOS APÓS A INSTALAÇÃO

8.1 Verificando os arquivos gerados na compilação a partir do fonte:

Para verificar o tamanho dos arquivos que foram gerados no diretório boot após a compilação do novo Kernel utilizamos o comando 'ls -lh', que gerou a saída mostrada na Figura 32. O tamanho total é de 134MB.


```

root@caiom-caiot-alexandre:/boot# ls -lh
total 134M
-rw-r--r-- 1 root root 202K mar 19 11:29 config-4.19.0-16-amd64
-rw-r--r-- 1 root root 202K jun 10 15:49 config-4.19.0-17-amd64
-rw-r--r-- 1 root root 139K jun 22 03:28 config-5.12.12
drwxr-xr-x 5 root root 4,0K jun 22 03:28 grub
-rw-r--r-- 1 root root 25M jun 22 01:33 initrd.img-4.19.0-16-amd64
-rw-r--r-- 1 root root 35M jun 22 02:10 initrd.img-4.19.0-17-amd64
-rw-r--r-- 1 root root 48M jun 22 03:28 initrd.img-5.12.12
-rw-r--r-- 1 root root 3,3M mar 19 11:29 System.map-4.19.0-16-amd64
-rw-r--r-- 1 root root 3,3M jun 10 15:49 System.map-4.19.0-17-amd64
-rw-r--r-- 1 root root 4,2M jun 22 03:28 System.map-5.12.12
-rw-r--r-- 1 root root 5,1M mar 19 11:29 vmlinuz-4.19.0-16-amd64
-rw-r--r-- 1 root root 5,1M jun 10 15:49 vmlinuz-4.19.0-17-amd64
-rw-r--r-- 1 root root 5,5M jun 22 03:28 vmlinuz-5.12.12

```

Figura 32 – ls -lh

8.2 Verificando o tamanho da pasta com os módulos compilados:

Para verificar o tamanho da pasta contendo os módulos que foram compilados e salvos no diretório /lib/modules, utilizamos o comando 'du -sh', que gerou a saída mostrada na Figura 33. O tamanho total é de 157 MB.

```

root@caiom-caiot-alexandre:/lib/modules# du -sh 5.12.12
157M    5.12.12

```

Figura 33 – du -sh

9 CONCLUSÃO

Nessa atividade nós realizamos a instalação do GNU/Linux e compilamos seu núcleo. Para tanto, foi necessário configurar o Debian em uma máquina virtual e então configurar e instalar o kernel do linux a partir de seu código-fonte. Como foi exposto no relatório, as configurações utilizadas nesse tutorial (uma ressalva: é possível realizar modificações de acordo com o hardware utilizado) possibilitou obtermos um sistema com espaço utilizado de memória bem reduzido.

10 REFERÊNCIAS

- Apêndice C - Particionar para Debian. DEBIAN. Disponível em:
<https://www.debian.org/releases/stable/amd64/apc.pt.html>. Acesso em: 20/06/2021.
- WALLEN, Jack. How to Compile a Linux Kernel. LINUX.COM, 2018. Disponível em:
<https://www.linux.com/topic/desktop/how-compile-linux-kernel-0/>. Acesso em: 20/06/2021.