CS3003D-OPERATING SYSTEMS

ASSIGNMENT-1

NAME: POTHURI LOKESH
ROLL NO: B200839CS
BATCH:B

PROBLEM STATEMENT

Download the latest stable Linux kernel from kernel.org, compile it, and dual boot it with your current Linux version. Your current version as well as the new version should be present in the grub-menu

METHODOLOGY

- We can directly dual boot with host os,but anything goes wrong then os could be corrupted,it creates problems in loading the kernel.
- Obtain the kernel source code from the kernel.org
- Next extract it and configure the features of the kernel.
- Install the dependencies that are required like build-essentials,bison,flex...
- Configure and compile the kernel as well.
- Install the compiled kernel and to the grub entry.
- Reboot the system.

Explanation

Introduction

A kernel is a piece of software that controls the hardware and does some basic functions like file management. Every operating system has one kernel. The Linux kernel is open source, implying a wide variety of coders together contribute to building it rather than just one company or developer team. In this report, we'll be showing the steps taken to upgrade to the latest stable kernel (which is 5.19.7 at the time of writing this report).

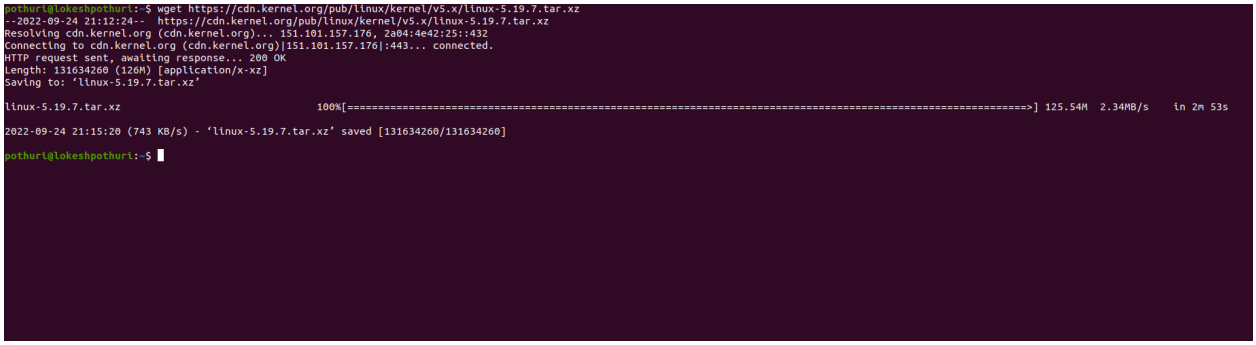
The process of building a Linux kernel takes seven easy steps to complete. However, the procedure requires a significant amount of time to complete, depending on the system speed.

Procedure

1.Getting the latest version of Linux Kernel source code.

$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.7.tar.xz

This will download the kernel code published on https://cdn.kernel.org as a tar file.



2.Extract the tar.xz file and change the directory to linux.5.19.7

$ tar -xf linux.5.19.7.tar.xz

This will extract the downloaded tar file into the folder linux-5.19.7 folder which contains all the source code.Now the change the directory to linux.5.19.7 by using below command.

$ cd linux.5.19.7

```
pothuri@lokeshpothuri:~$ tar -xf linux-5.19.7.tar.xz
pothuri@lokeshpothuri:~$ cd linux-5.19.7
pothuri@lokeshpothuri:~/linux-5.19.7$
```

3.configure the features

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

```
pothuri@lokeshpothuri:~/linux-5.19.7$ cp /boot/config-$(uname -r) .config
pothuri@lokeshpothuri:~/linux-5.19.7$
```

4.Install the required compilers and other tools

$ sudo apt-get install build-essential libncurses-dev bison flex libssl-dev libelf-dev

The above command will install all the required dependencies for the kernel compilation.

```
pothuri@lokeshpothuri:~/linux-5.19.7$ sudo apt-get install build-essential libncurses-dev bison flex libssl-dev libelf-dev
[sudo] password for pothuri:
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8ubuntu1.1).
The following additional packages will be installed:
  libfl-dev libfl2 libsigsegv2 libssl1.1 m4 zlib1g zlib1g-dev
Suggested packages:
  bison-doc flex-doc ncurses-doc libssl-doc m4-doc
The following NEW packages will be installed:
  bison flex libelf-dev libfl-dev libfl2 libncurses-dev libsigsegv2 libssl-dev m4 zlib1g-dev
The following packages will be upgraded:
  libssl1.1 zlib1g
2 upgraded, 10 newly installed, 0 to remove and 459 not upgraded.
Need to get 4,714 kB of archives.
After this operation, 15.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libsigsegv2 amd64 2.12-2 [13.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 m4 amd64 1.4.18-4 [199 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/main amd64 flex amd64 2.6.4-6.2 [317 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 zlib1g amd64 1:1.2.11.dfsg-2ubuntu1.3 [53.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libssl1.1 amd64 1.1.1f-1ubuntu2.16 [1,321 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/main amd64 bison amd64 2:3.5.1+dfsg-1 [657 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 zlib1g-dev amd64 1:1.2.11.dfsg-2ubuntu1.3 [155 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libelf-dev amd64 0.176-1.1build1 [57.0 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libfl2 amd64 2.6.4-6.2 [11.5 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libfl-dev amd64 2.6.4-6.2 [6,316 B]
Get:11 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libncurses-dev amd64 6.2-0ubuntu2 [339 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libssl-dev amd64 1.1.1f-1ubuntu2.16 [1,584 kB]
Fetched 4,714 kB in 51s (91.6 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libsigsegv2:amd64.
(Reading database ... 186349 files and directories currently installed.)
Preparing to unpack .../libsigsegv2_2.12-2_amd64.deb ...
Unpacking libsigsegv2:amd64 (2.12-2)
```

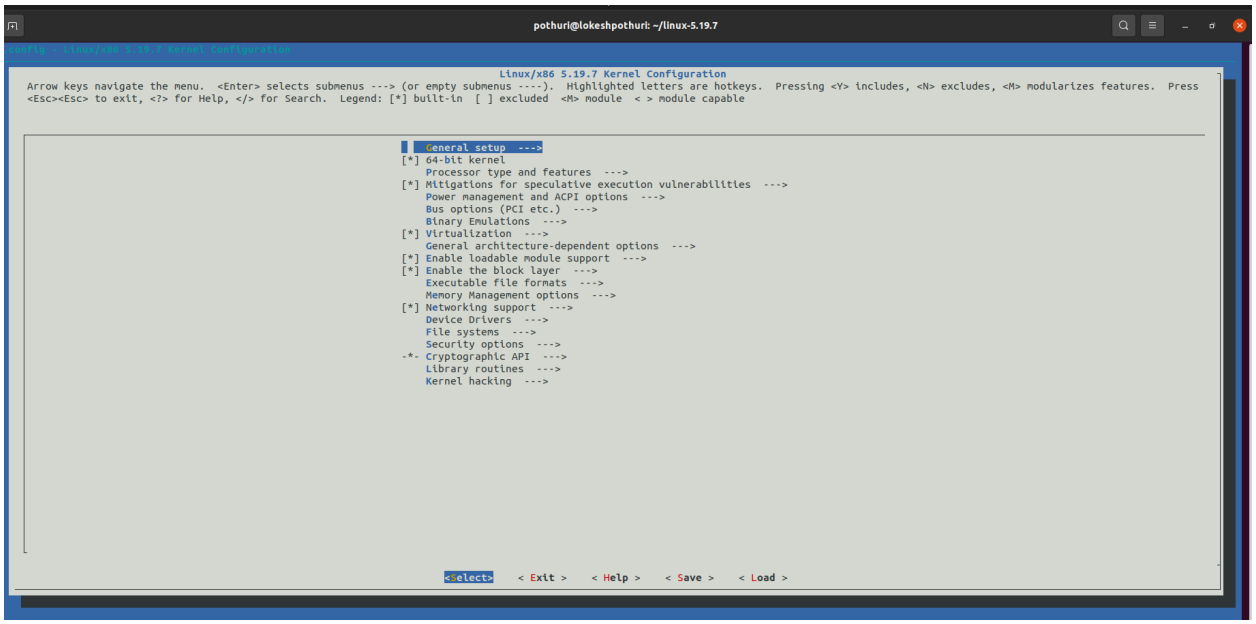
5.Configure the kernel

$ make menuconfig

This is just to make sure we have all the required files in this directory and we are good to compile the code and check the menuconfig.

```
pothuri@lokeshpothuri:~/linux-5.19.7$ make menuconfig
  HOSTCC  scripts/basic/fixdep
  UPD     scripts/kconfig/mconf-cfg
  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
  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/mconf
#
# using defaults found in /boot/config-5.15.0-48-generic
```

This command open up a configuration tool that allows us to go through every module available and enable or disable what we need or don't need.



6.Compile the kernel

$ make

This command will start compiling the kernel.

To speed up compile time, pass the -j as follows:

use 4 core/thread

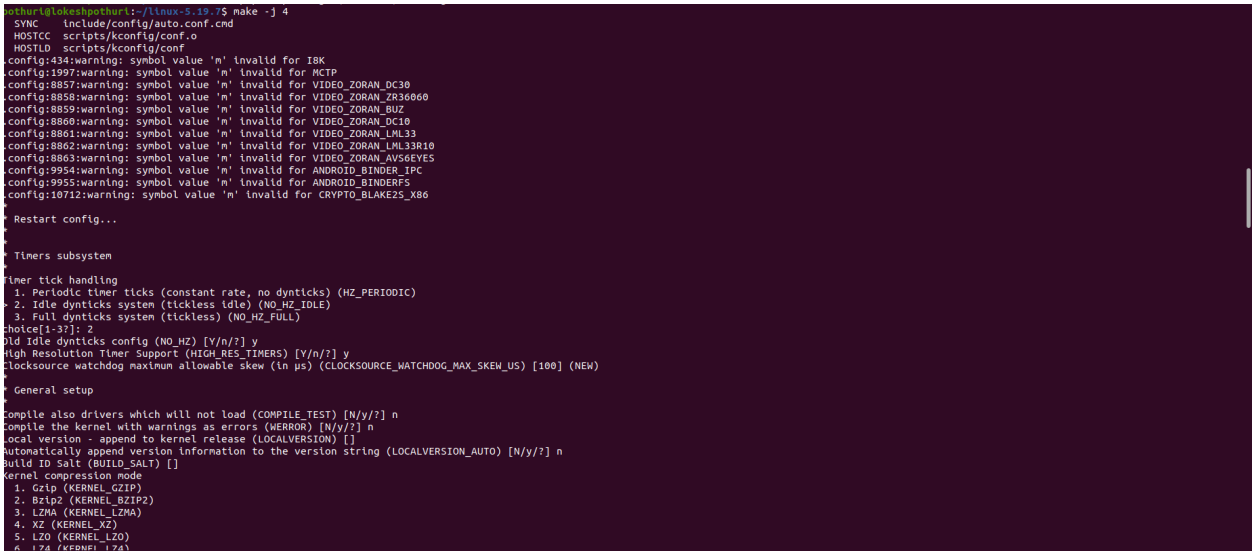
$ make -j 4

get thread or cpu core count using nproc command

$ make -j $(nproc)

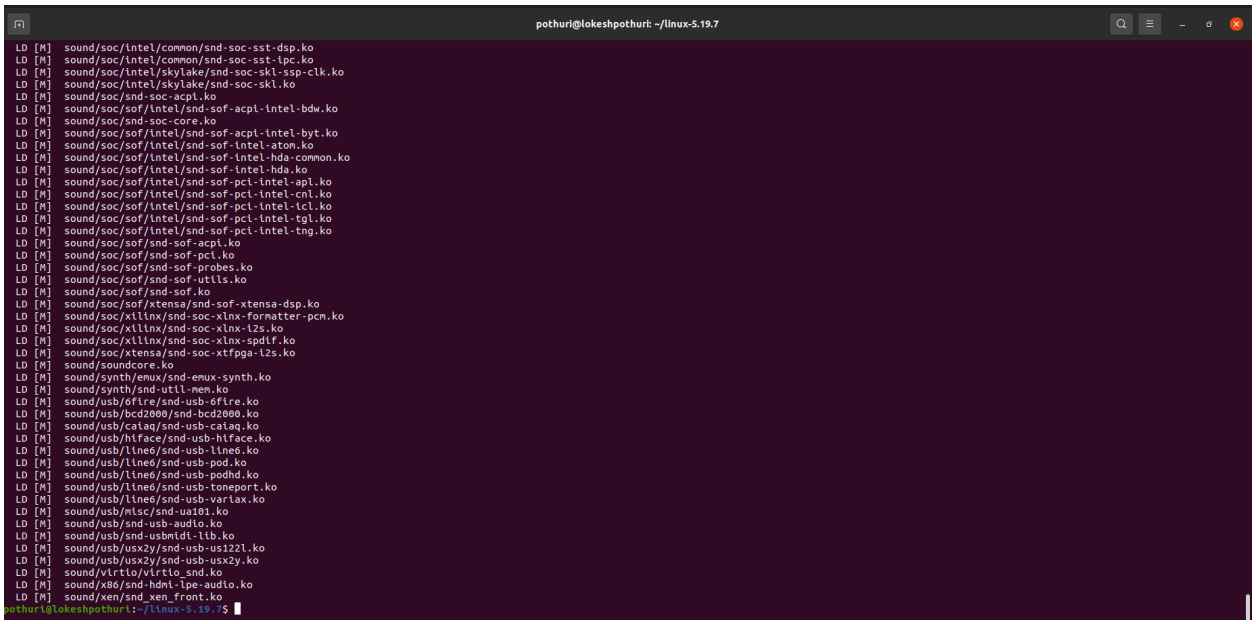
The $ nproc on my system gave 4 using which I performed the make command.

$ make -j 4



Compiling and building the Linux kernel going take a significant amount of time. The build time depends upon your system's resources such as available CPU core and the current system load. So we must have some patience.

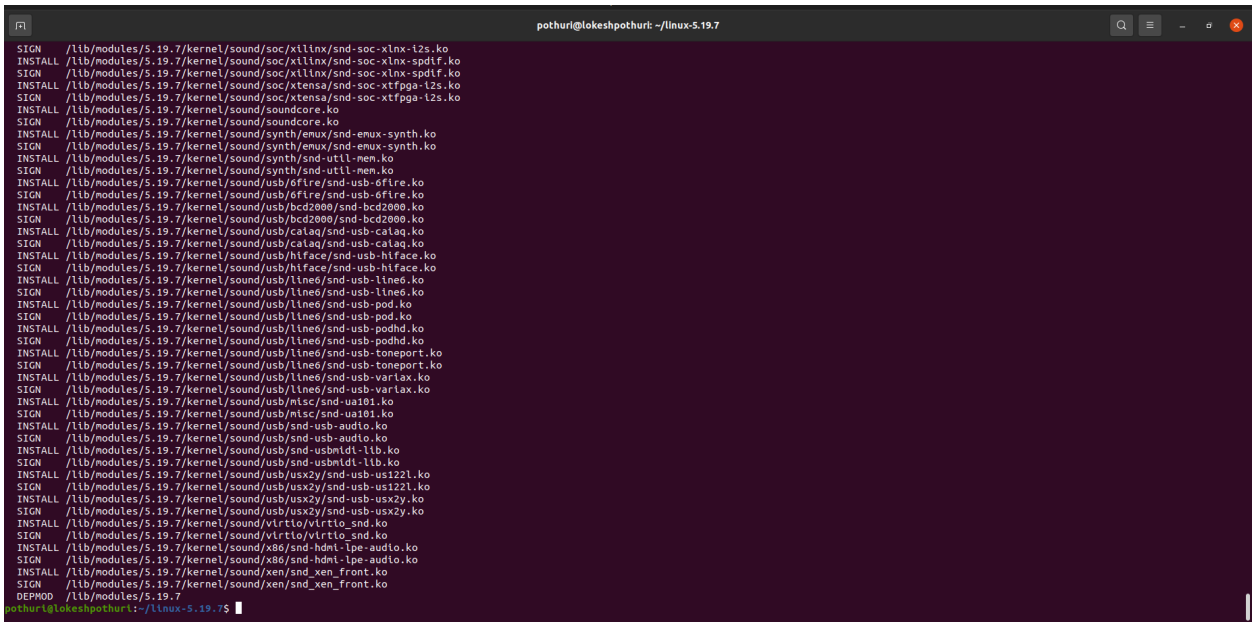
The end of this step looks like this ⇩



7.Install the Linux kernel modules↓

$ sudo make modules_install

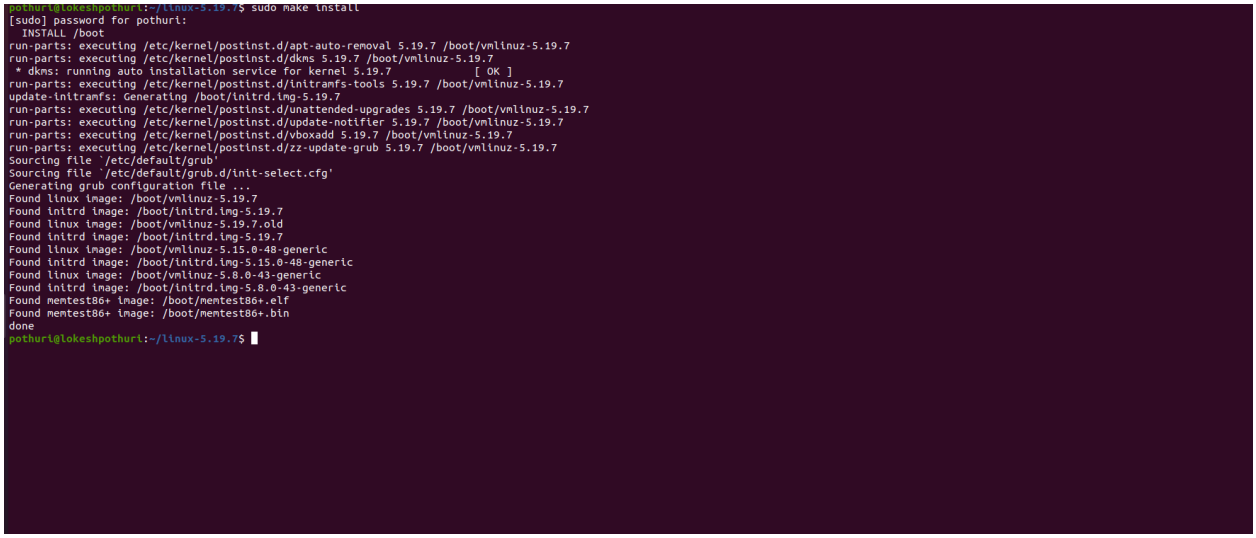
The end looks likes this ⇩



8.Install the Linux kernel

So far we have compiled the Linux kernel and installed kernel modules. It is time to install the kernel itself:

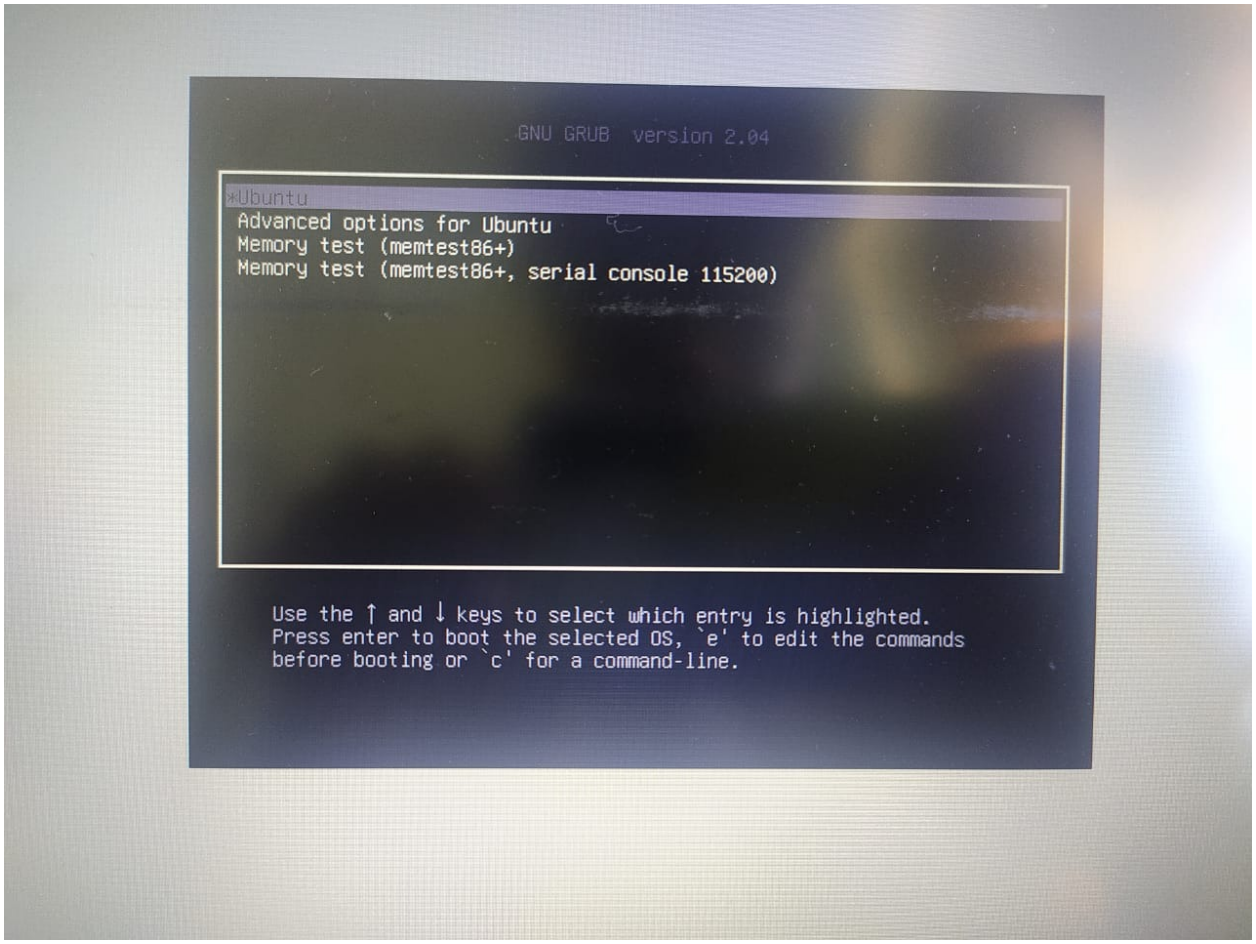
$ sudo make install

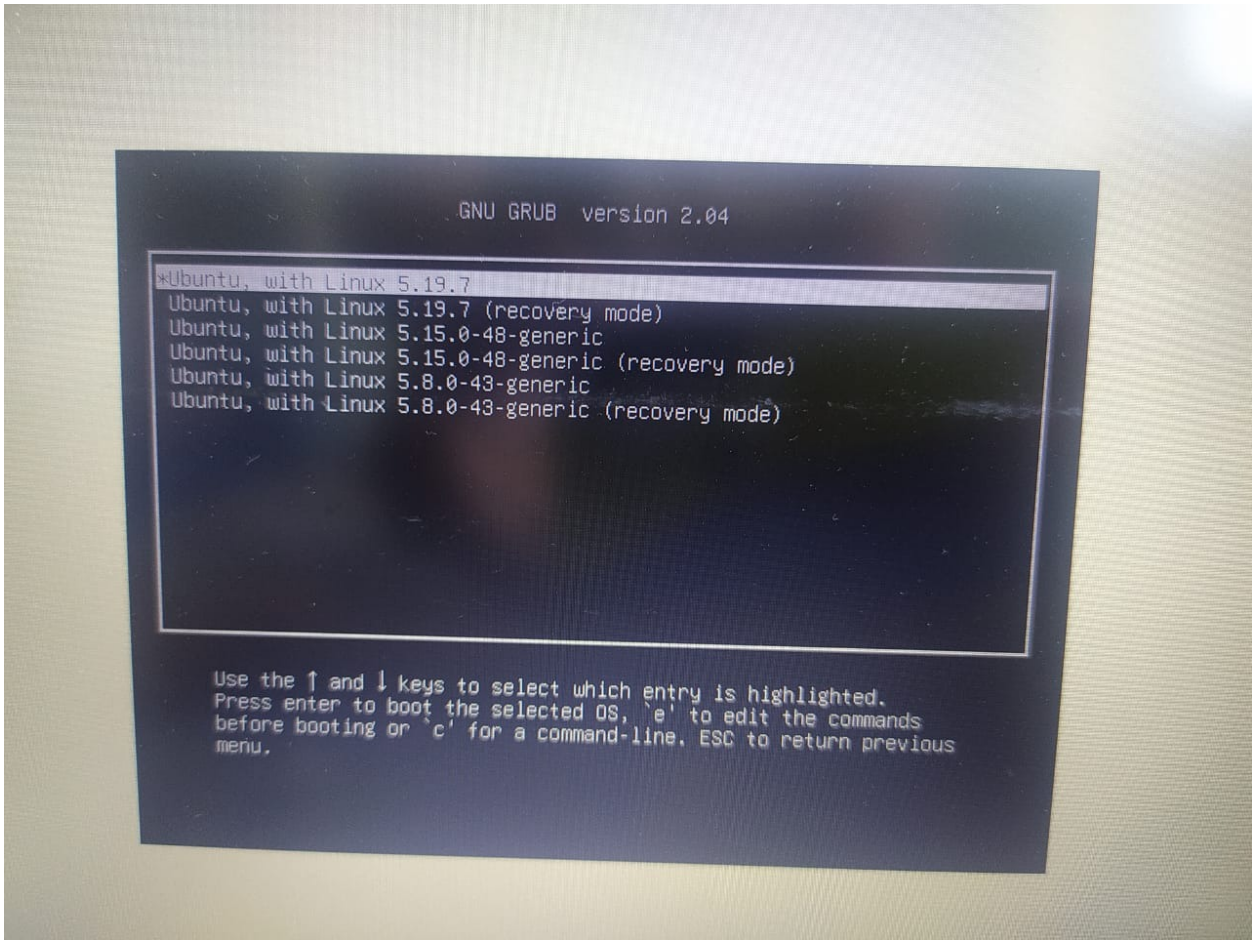


Grub Menu

We should ideally reboot the virtual system after the completion of this stage.

On turning on hold the right shift key during boot up to access the GRUB menu, which looks like this as shown below


GNU GRUB version 2.04
*Ubuntu
Advanced options for Ubuntu
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.



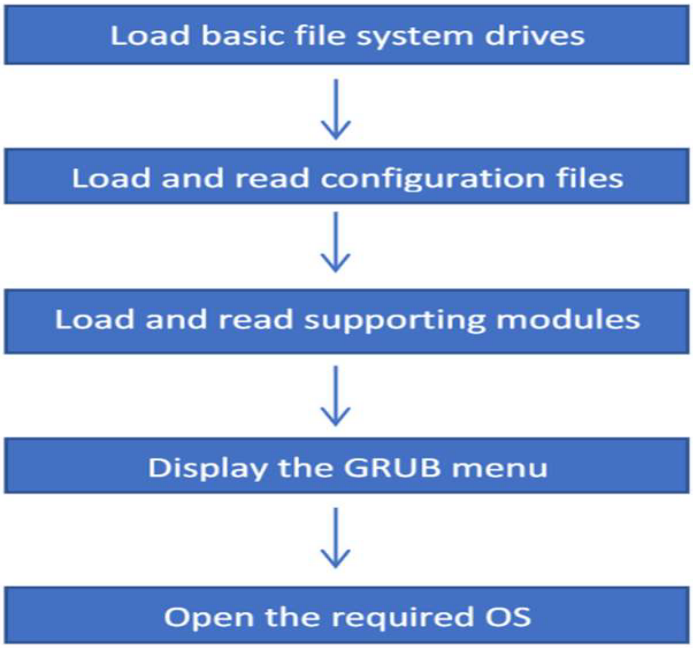
To show that the latest kernel is installed, use the command in the note given below, this will print a more specific string with actual release.

$uname -r

Conclusion

Linux Compile Kernel version 5.19.7
Configurations! We completed various steps to build the Linux kernel from source code and the compiled kernel should be running on our system.

Flow chart


Load basic file system drives
Load and read configuration files
Load and read supporting modules
Display the GRUB menu
Open the required OS