OS Assignment #1

Table of Contents

Mohammad Yehya Hayati, K213309, BCS-4F

Intro

This is the first Operating System Assignment of Spring 2023. Roll Number = K213309, Name = Mohammad Yehya Hayati.

Installing all packages

The main command to perform this is:

- **sudo apt install gcc flex bison dwarves libelf-dev build-essential libncurses5-dev libssl-dev**

Here is a screenshot (I performed this prior to any research, and was just blindly following the incomplete/incorrect manual, therefore my work is all over the place):

```
mohammadyehya@K213309: $ sudo apt-get install libssl-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
...
Setting up libssl-dev:amd64 (3.0.2-0ubuntu1.8) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
```

```
mohammadyehya@K213309: $ sudo apt install libelf-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
Setting up libelf-dev:amd64 (0.186-1build1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
```

```
mohammadyehya@K213309: $ sudo apt install flex bison dwarves
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
Preparing to unpack .../2-flex_2.6.4-8build2_amd64.deb ...
```

Downloading and Configuring the Kernel

Download and Extract

First we have to download the kernel using this command:

- **wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.15.1.tar.xz**

Then we extract and open the file using these commands:

- **tar xf linux-5.15.1.tar.xz**
- **cd linux-5.15.1**

Configuring

Now that the kernel is downloaded and extracted we need to configure it.

So first we need to copy our current kernel config, and that is done using these commands:

- **ls /boot | grep config**
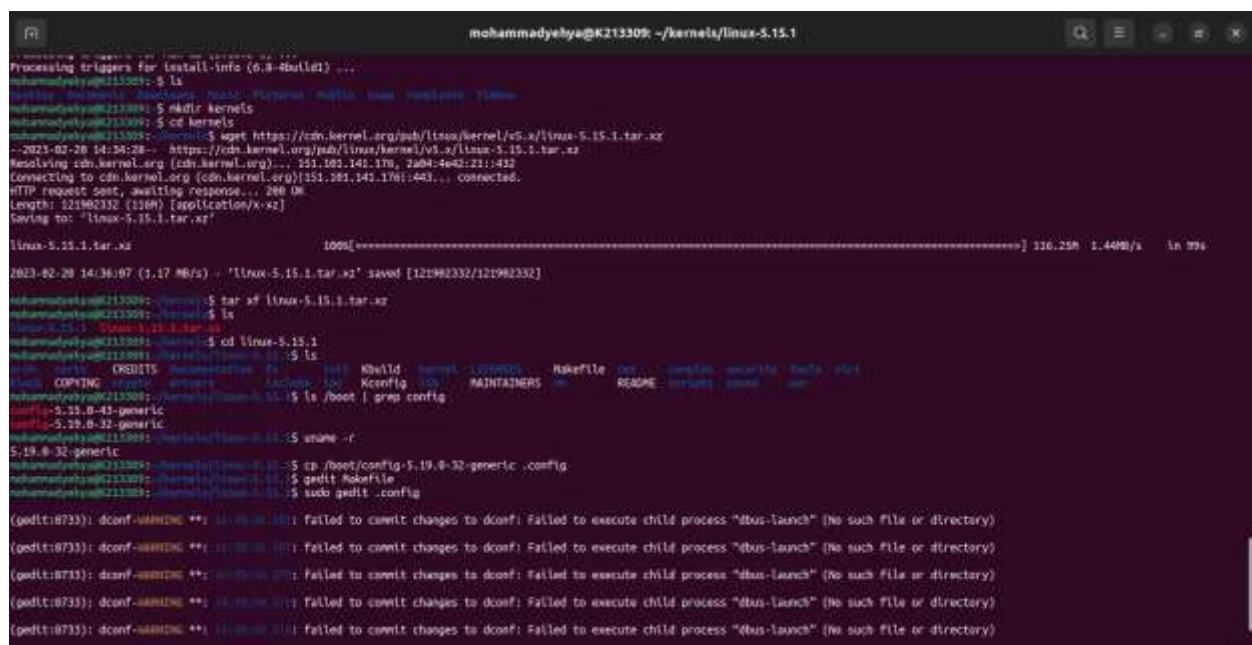- **cp /boot/config-X.X.X-X-generic .config**

(X.X.X-X is the version of the current running kernel)

Next we need to open the Makefile and set our roll number with the kernel version (as requested in the assignment).

We open the Makefile using the command:

- **gedit Makefile**
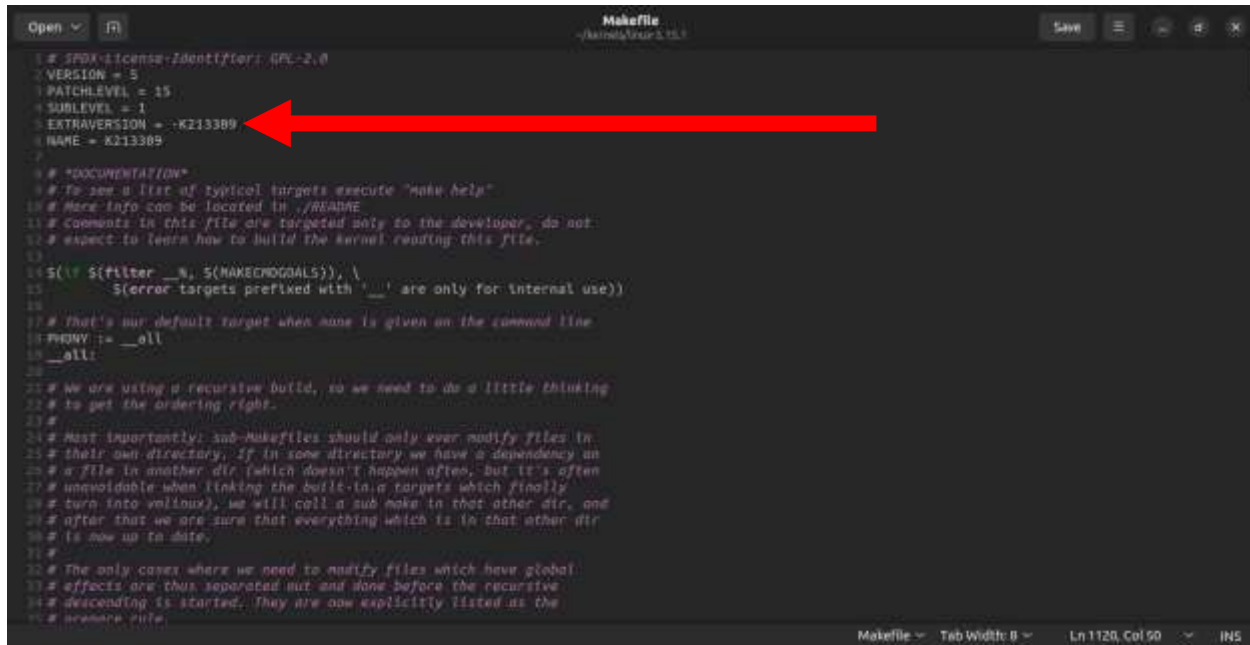
Here are some screenshots:

Mohammad Yehya Hayati, K213309, BCS-4F

Now we have to set the config parameters of our current running kernel to the new one, and to do that we use this command:
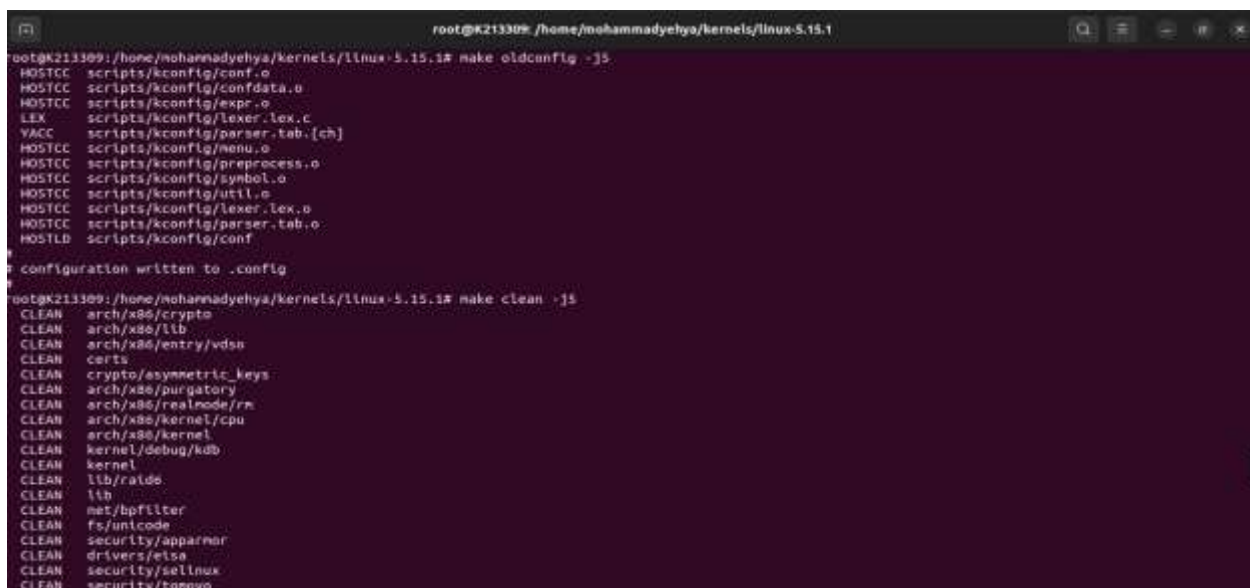
- **yes '' | make oldconfig -j4**

(-jx, where x represents the number of cores available to your pc. To find out how many and available use command nproc)

Extra step is to clean all temp files in the new kernel, to do that we use command:

- **make clean -j4**
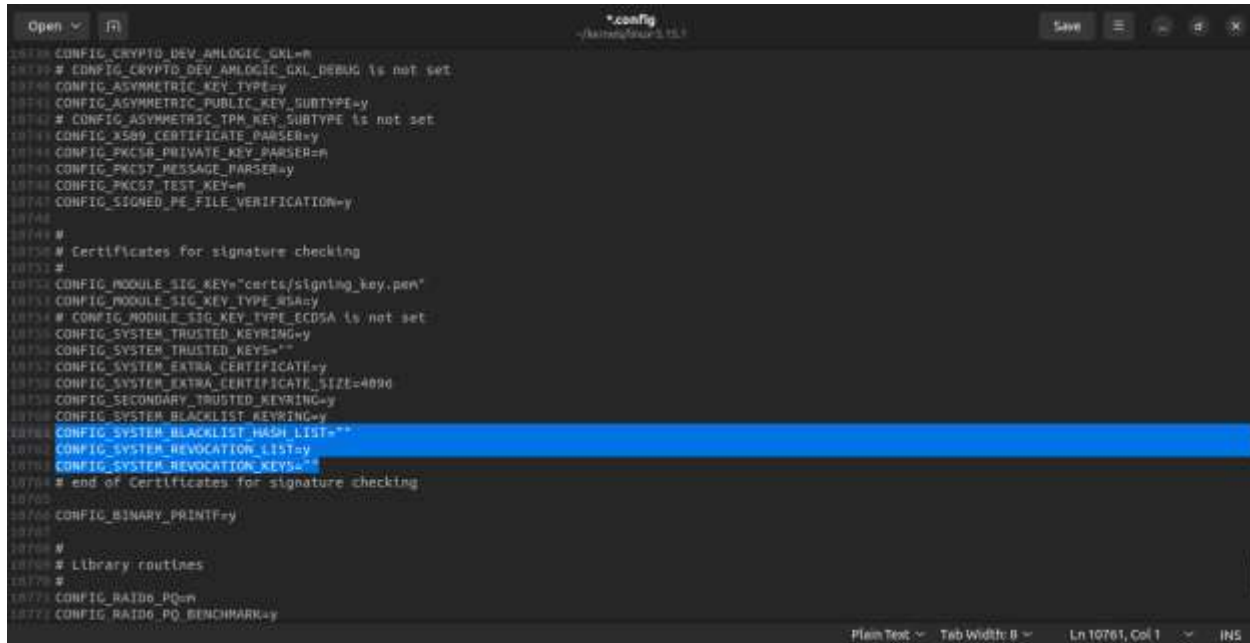
Here are some screenshots:

Mohammad Yehya Hayati, K213309, BCS-4F

Before building the kernel we need to make sure we perform one more step.

We need to open the .config file and change 2 strings to empty strings.

These strings are set to CONFIG_SYSTEM-BLACKLIST_HASH_LIST and CONFIG_SYSTEM_REVOCATION_KEYS.

Here is a screenshot:

Mohammad Yehya Hayati, K213309, BCS-4F

Adding Hello World System Call

First of all we will make a directory in the kernel directory having the name helloworld, this is done using the command :

- **mkdir helloworld**

Next we will make a C file which actually defines what the system call will do:

- **gedit helloworld.c**

Here is a screenshot:
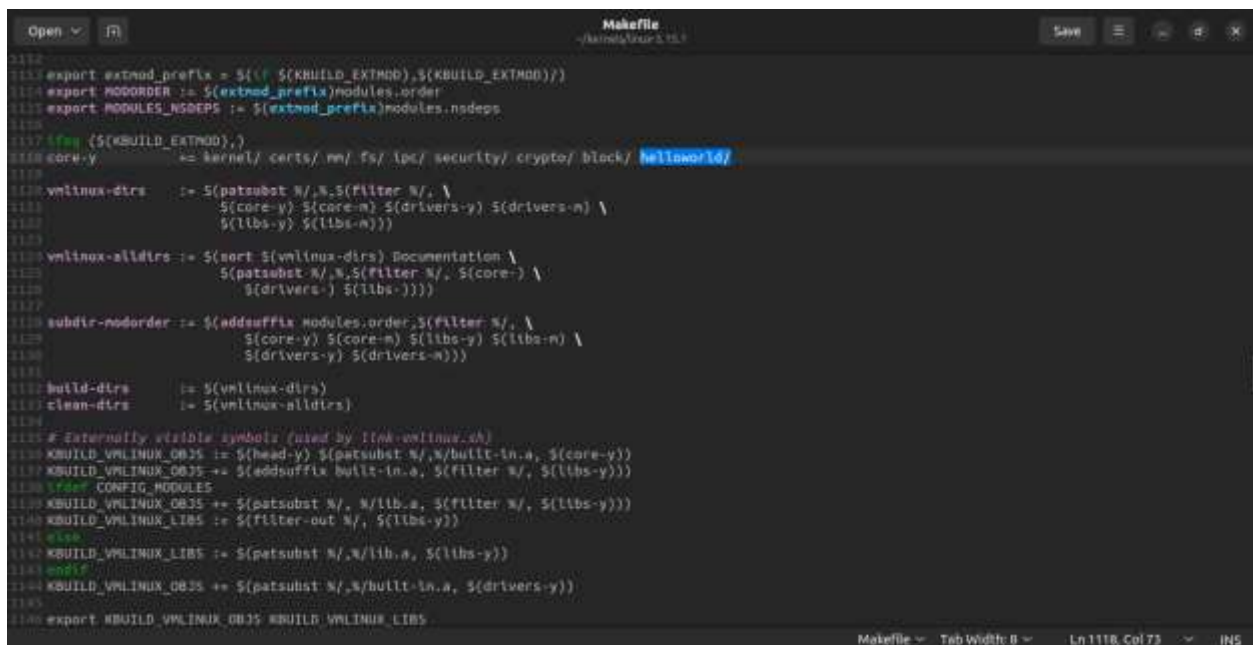
Then we make a Makefile (gedit Makefile) and do the following:

Then we go back to the kernel directory and open the Makefile again, this time we need to add out C file into the list as follows:

Now we have to update the system call library to include our helloworld.c, so we need to open the file as follows:

Mohammad Yehya Hayati, K213309, BCS-4F

- **gedit include/linux/syscalls.h**

And add the following line:

The last step is to add the system call to the kernel's system call table like so

- **gedit arch/x86/entry/syscalls/syscall_64.tbl:**

And do as follows

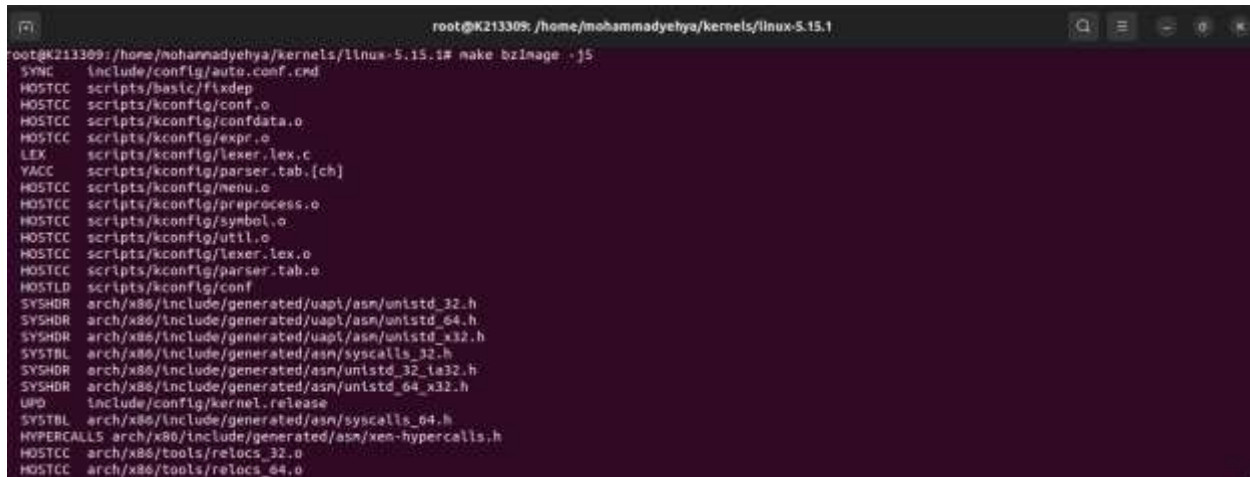Mohammad Yehya Hayati, K213309, BCS-4F

And remember to put it above the x32 system calls, as well as to keep the number of the system call in mind.

Building the Kernel

Now we have to build the kernel into an installable image file which we can do using the command:

- **Sudo make bzImage -j4**

Now we need to make and install the modules, and this process takes an extremely long amount of time since the Linux kernel has a huge number of line of codes (around 3 hours for my pc but can vary from pc to pc);

- **make module –j4**
- **make modules_install –j4**
- **make install –j4**

Here is a screenshot:

Mohammad Yehya Hayati, K213309, BCS-4F

Since the process took a long time to finish, I forgot to take screenshots of the other commands (module_install and install).

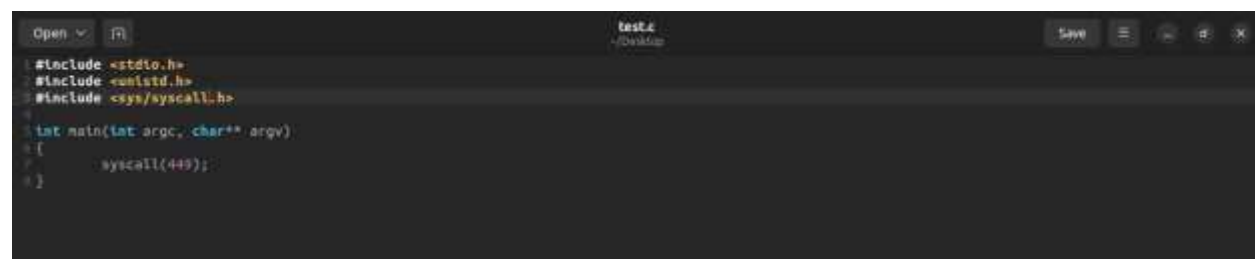The last step is to update the GRUB menu (BIOS) simply using the command:

- **sudo update-grub**

Selecting the Kernel and Checking

During the reboot phase we have to pick our kernel version, as follows:
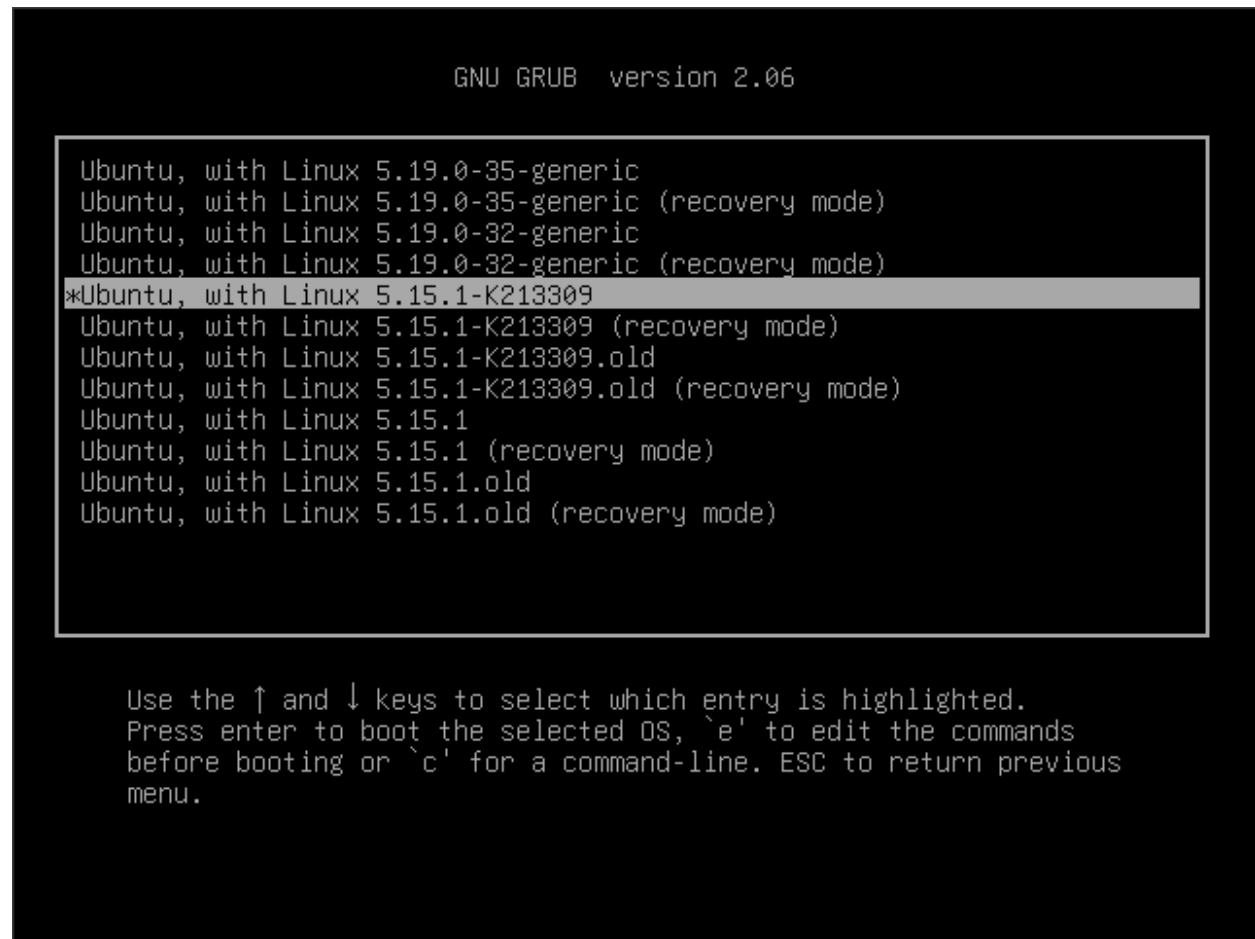
Once we have picked the kernel version, all we need to do is make an example C file and run the helloworld system call in it.

Now that we have created a C file, it is time to execute it.

Here are the screenshots:

Mohammad Yehya Hayati, K213309, BCS-4F