Pre Step:

Log In Virtual Box: Y:\Subject\vm\_image\EmbeddedSystem

Start: StartTheVM\_50gram\_2024

Connect to Putty:

Serial Line:COM3

Connection type: Serial

Speed: 115200

NFS remote directory mounting:

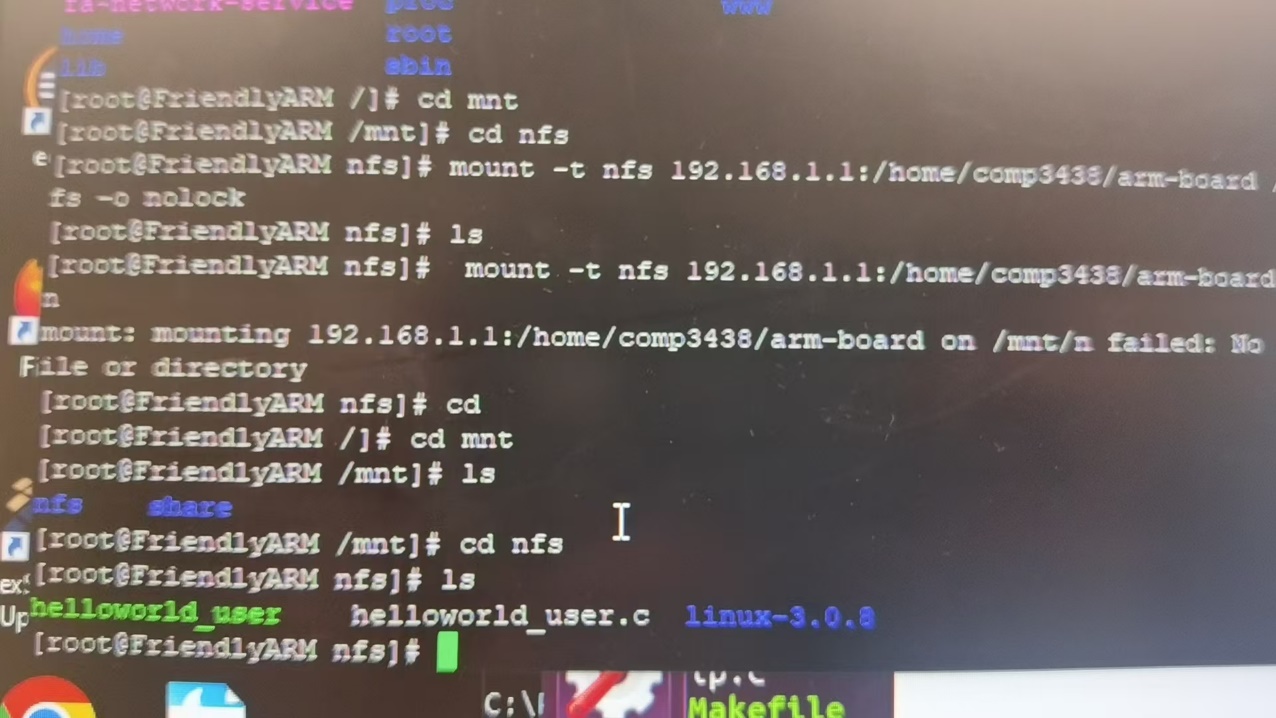
Enter:

cd /mnt

ls nfs

mount -t nfs 192.168.1.1:/home/comp3438/arm-board /mnt/nfs -o nolock

Result should have after compilation:



Open

https://elixir.bootlin.com/

Search

linux-3.0.8

Identifier: MODULE\_LICENSE

Click Into: include/linux/module.h, line 140 (as a macro)

for Linux kernel source code

Step 1 and 2 (Proper download of device and application source code in correct directory ):

In Virtual Embedded Machine:

Go to Firefox browser Blackboard Lecture 9 then Download the helloworld.zip file

OR

Download the helloworld.zip file from original computer blackboard and copy into shared folder:

(Original machine documents/shared)

(Embedded Computer top left Devices-Shared Folder)

Enter the following command in the /home/comp3438/arm-board/linux-3.0.8/drivers/char position

First Command:

sudo bash

COMP3438 Enterpassword:12345

Then following command to copy

cp /media/sf\_Shared/helloworld\_driver.c .

To copy the helloworld\_driver.c file into the embedded machine

cp /media/sf\_Shared/helloworld\_user.c .

To copy the helloworld\_user.c file into the embedded machine

cp /media/sf\_Shared/Kconfig .

cp /media/sf\_Shared/Makefile .

Target:

Put helloworld\_driver.c file into /home/comp3438/arm-board/linux-3.0.8/drivers/char

Put helloworld\_user.c file into /home/comp3438/arm-board

Put replace Makefile and Kconfig in /home/comp3438/arm-board/linux-3.0.8/drivers/char

Remove executable nature of helloworld\_driver.c:

chown comp3438:comp3438 helloworld\_driver.c

chmod a-x helloworld\_driver.c

chown comp3438:comp3438 helloworld\_user.c

chmod a-x helloworld\_user.c

Step 3 (Driver Compilation) :

At /home/comp3438/arm-board/linux-3.0.8/drivers/char

Vim Makefile

Add

obj-$(CONFIG\_HELLO\_WORLD) += helloworld\_driver.o

at the lowest of Makefile

Next

Vim Kconfig

Shift+g to go to end of the file

Enter before last line:

config HELLO\_WORLD

tristate "The hello world char driver"

depends on CPU\_S5PV210

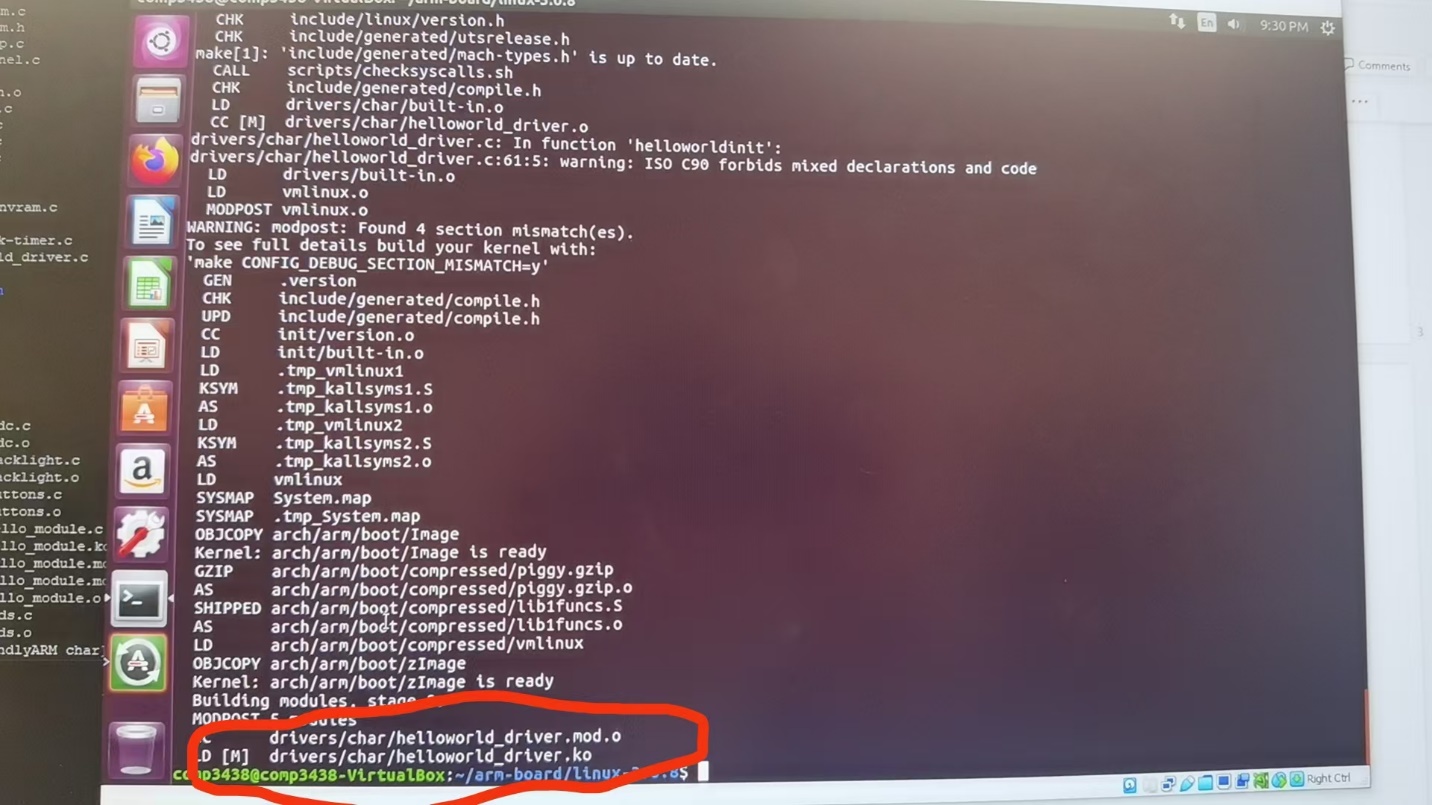
Next return back to linux-3.0.08 (two times cd ..)

Then enter: make menuconfig

Inside the GUI go Device Drivers, then Character Devices, Finally enter “M” for Hello World Driver Configuration

TAB to Exit

Then start compilation by entering:make

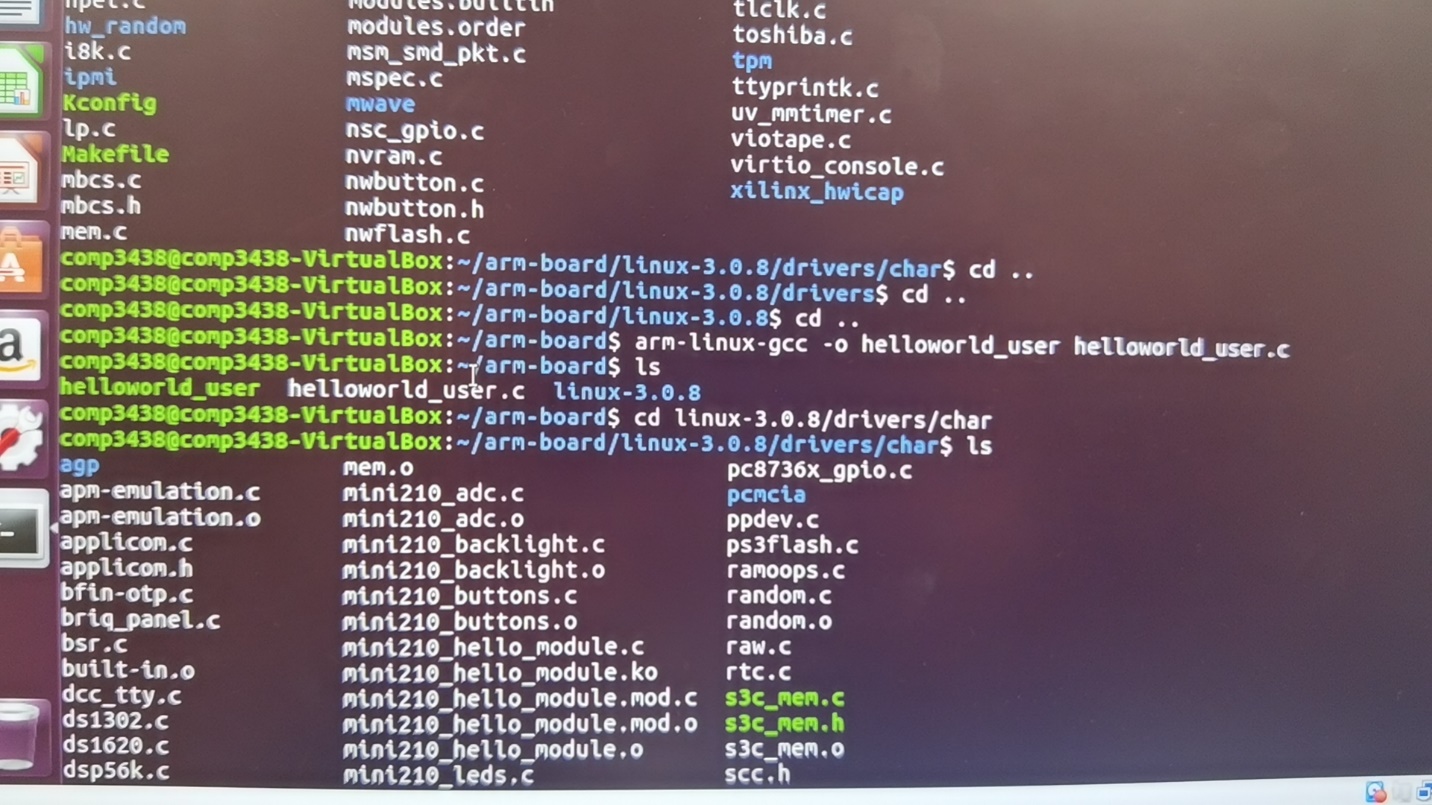
Successful compilation

Step 4 (Driver loading and device file set up):

Enter the following codes to generate a file to run later in directory (/home/comp3438/arm-board):

arm-linux-gcc -o helloworld\_user helloworld\_user.c

Expecting Result:



Step 5 (Application Execution):

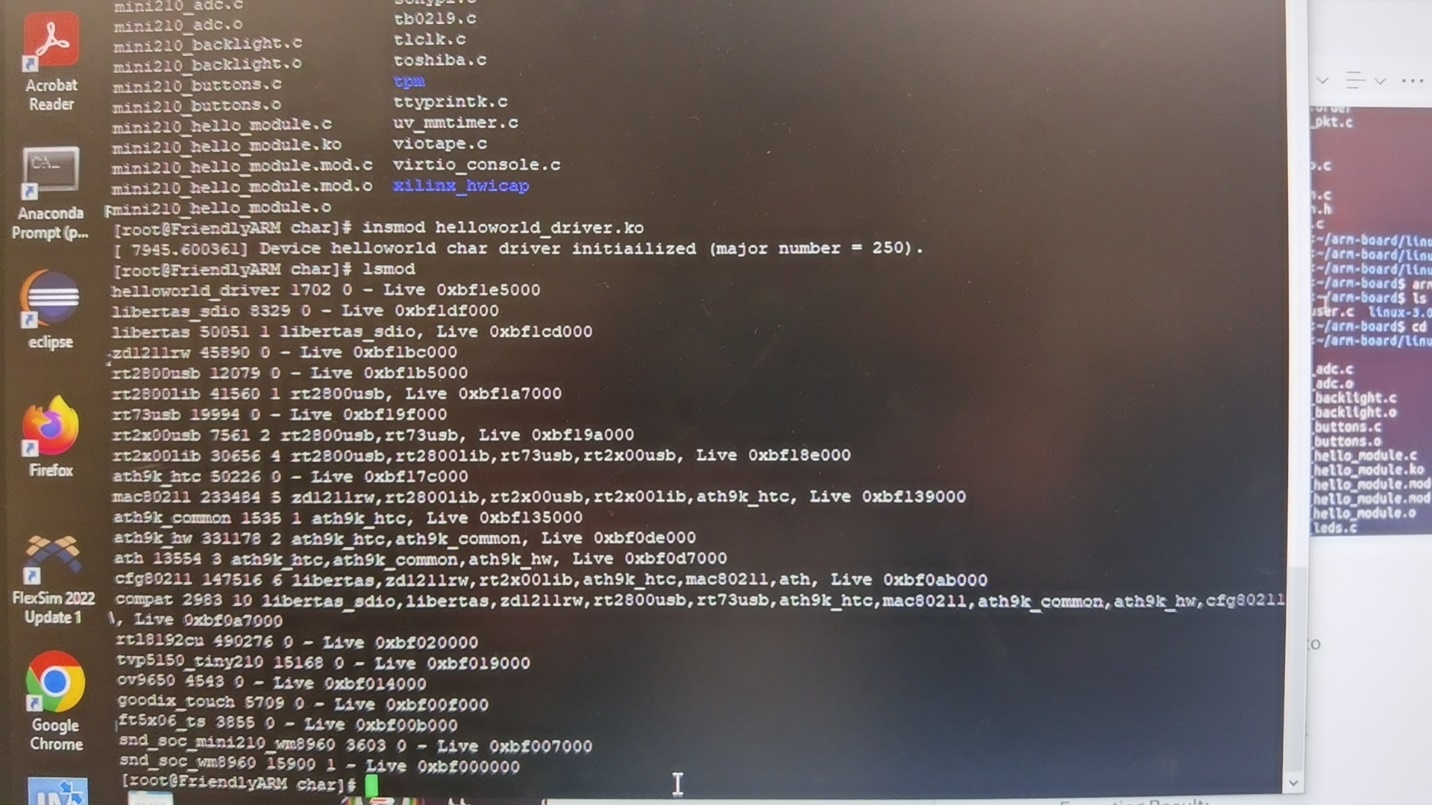
Return back to putty and go to linux-3.0.8/drivers/char

Run code:

insmod helloworld\_driver.ko

Result should be like this :

cat /proc/devices (Check logs about the kernel, faked files, 250-helloworld char driver)



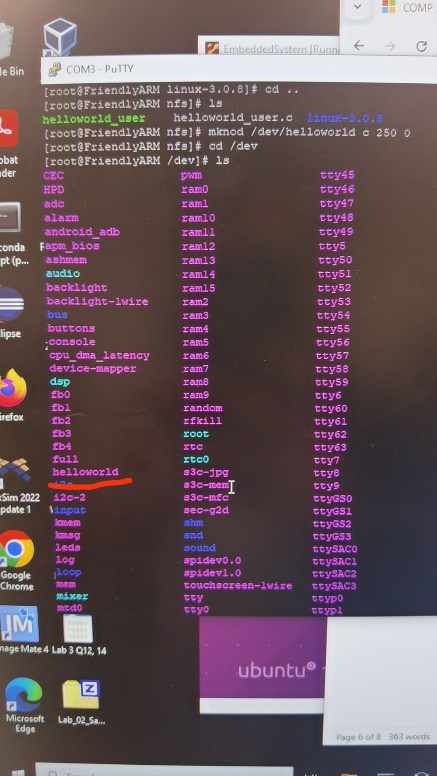
Then create a device file: (go back to nfs three times cd ..)

Enter cd /dev and then ls to check:

Major number 250 and minor number is 1 as listed at S\_N 1 in helloworld\_driver.c

mknod /dev/helloworld c 250 1

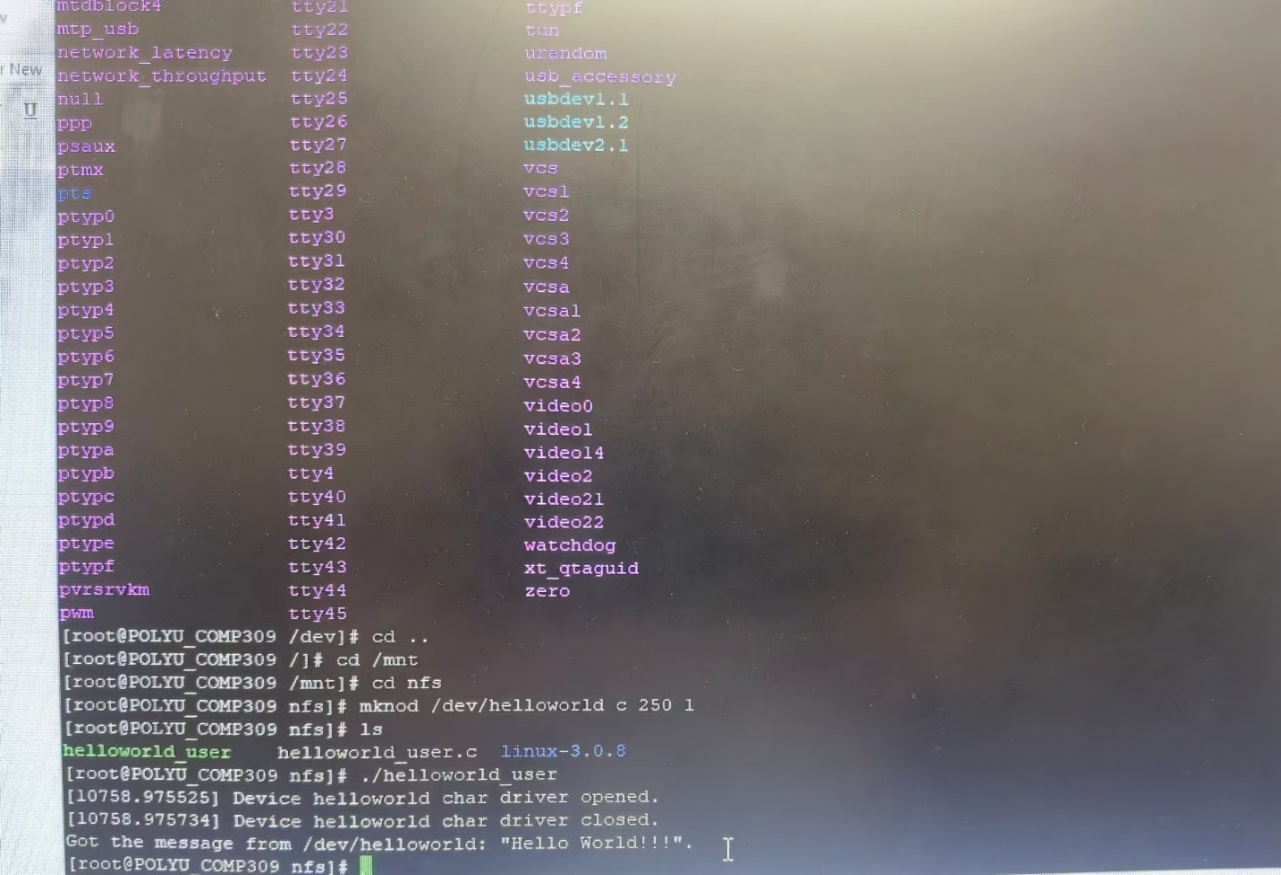
Expected result:



After device code created, we can do execution, return back to nfs and run the code entering :

./helloworld\_user

Expecting Result:

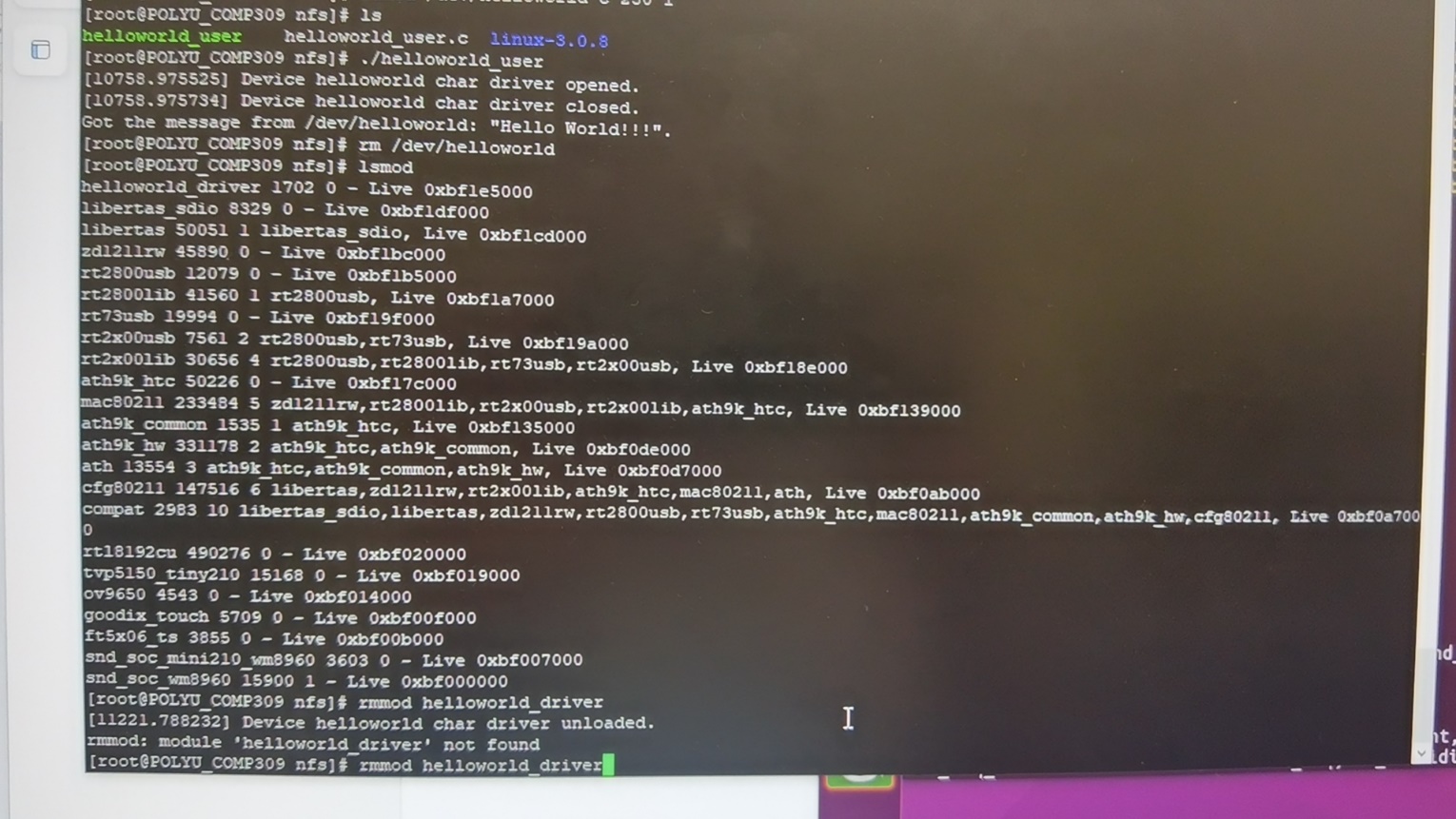


Step 6 (Device file deletion and driver unloading):

Delete Device file call command : rm /dev/helloworld

Driver Unloading call command : rmmod helloworld\_driver

Expecting Result:



Project Rubric:

