Mauro Lopez

LAB 9 Report

ECEN-449

Section 502

November 29, 2018

Introduction:

In this lab we are going to learn more about built in kernels for Linux. This built in drives will load during the boot in process, so that you can call them as soon as is done the boot in process.

Procedures:

We start off by first looking at the Linux menu configuration. We do this by typing the following command:

$ make ARCH=blackfin menuconfig

This will open a window and allow you to look through the configurations of different drives.

Once we have an idea of how different configurations depend on different drives. We can start by implementing the multipy IP from our lab 6.

We begin by first creating a new directory for the our multiply drive and call it "multiplier_driver". We then create a Makefile and add the following line to it.

Ob-$(CONFIG_MULTIPLIER_DRIVER) += multiplier.o

We copy all of our source files for our multiplier from our lab 6 modules directory.

NOTE: avoid overriding the Makefile.

The next step is to also create a Kconfig file and add the following to it.

Config MULTIPLIER_DRIVER

Tristate "multiplier_driver"

Depends on ARM

Default y if ARM

Help

Refer to ECEN449TAMU

Once we have those files ready we need to edit the Makefile in the Driver directory as well as the Kconfig file.

Add the following to the makefile:

#ECEN 449

Ob-$(CONFIG_MULTIPLIER_DRIVER) += multiplier_driver/

To the Kconfig:

Source “drivers/multiplier_driver/Kconfig”

Once we have done the above commands we can launch the menuconfig.

We do this by using the same command as before:

$ make ARCH=arm menuconfig

In this menu we should see our driver set to “built in”.

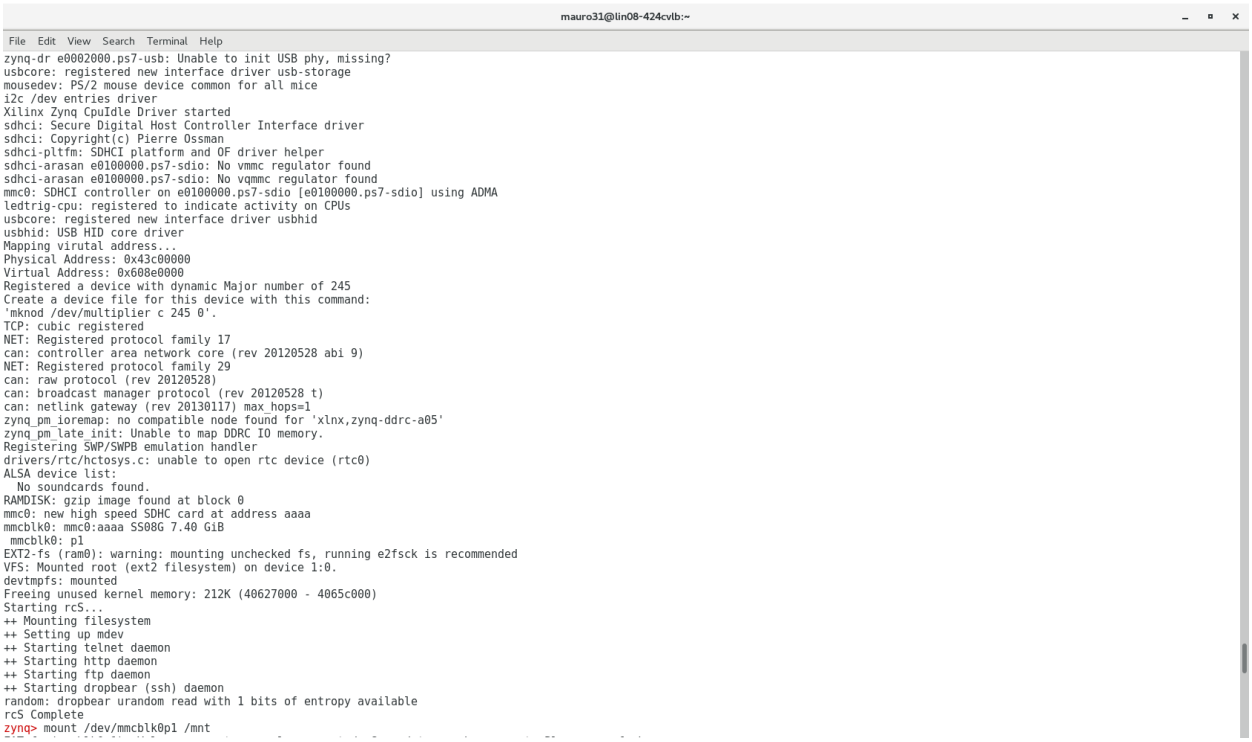
We then need to recompile our linux source files and generate our uImage.

Once we have compiled and unzipped our zImage file, we can use our BOOT.bin from lab 5 as well as our devicetree.dtb to boot Linux in our ZYBO board.

We follow the same steps from above to add our IR_DEMOD driver.

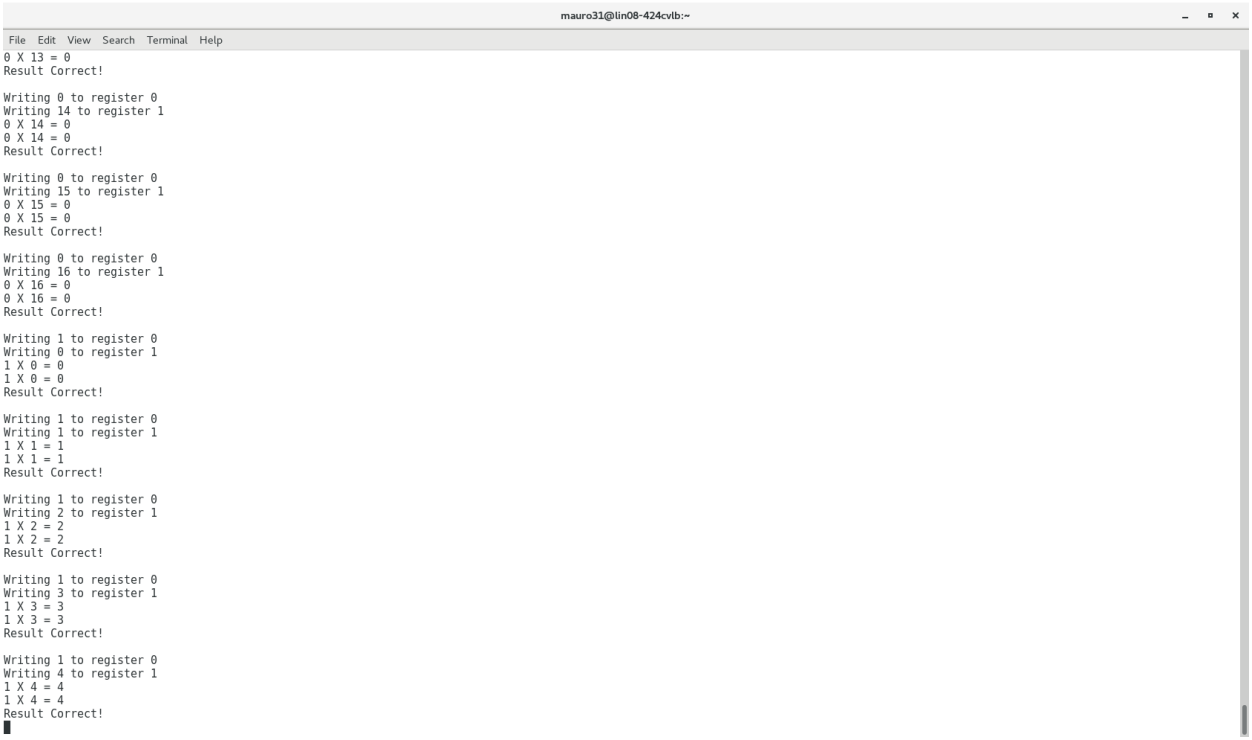
Results:

Once we have successfully added our driver, we should be able to see it as the system is boot it up. Something similar to this:

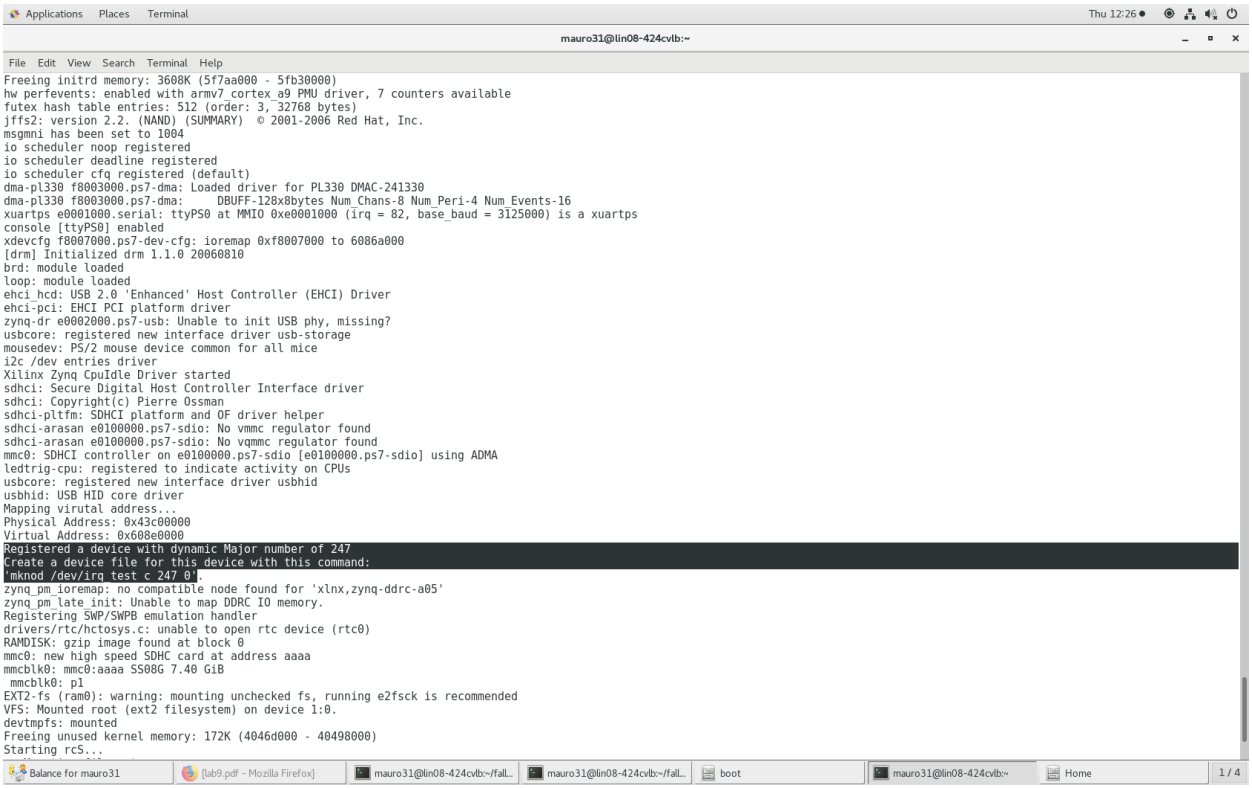


We can see our multiplier drive print out statements.

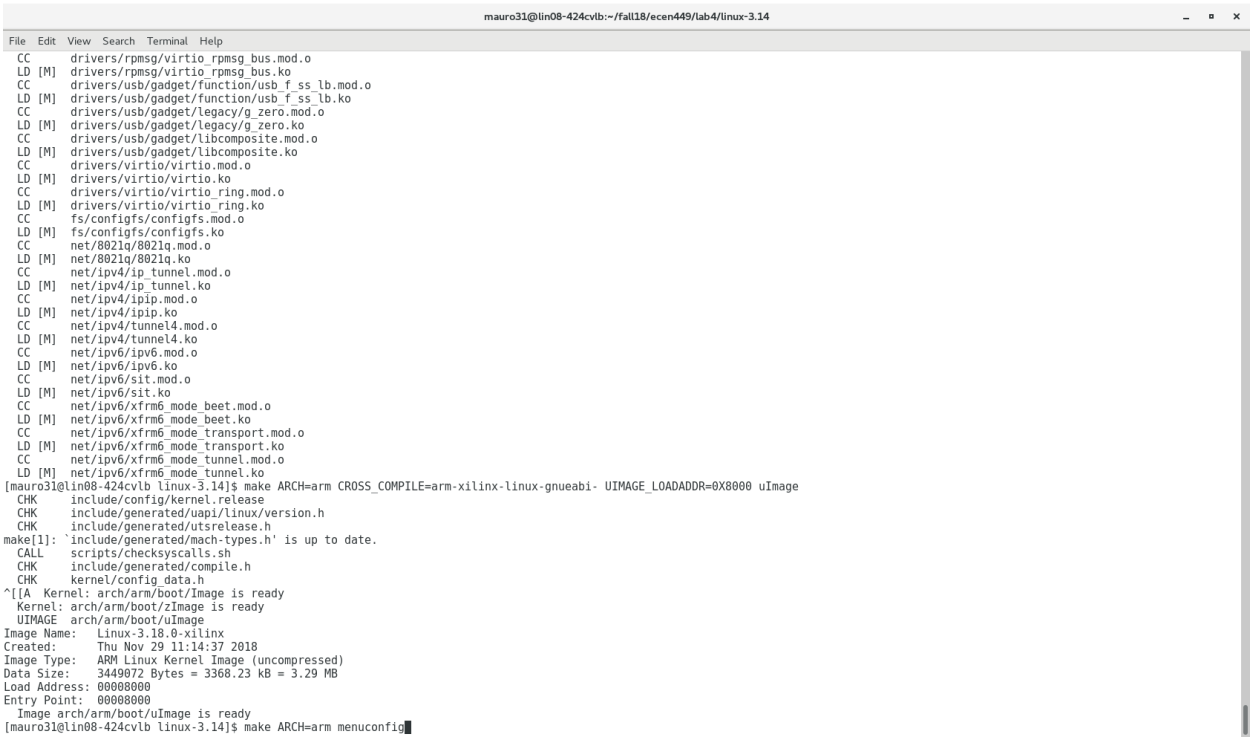
When we run our devtest executable file we should get the following:



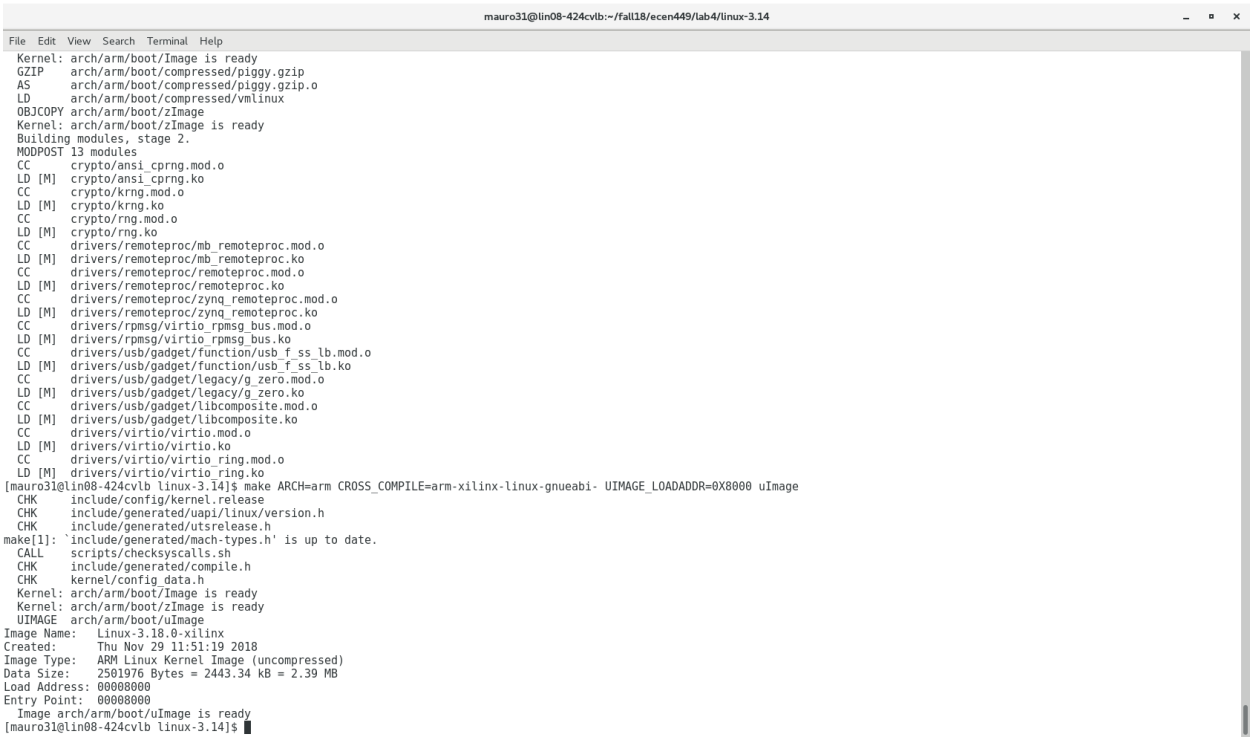
Once we implement the IR_DEMOD driver into linux we should also get the boot in sequence with the following print out statements:



As we add and remove drives from our menuconfig we can see the size of our uImage change. For instance here is the initial size of the uImage with the multiplier_drive added to it:



Once we remove the networking support, Device Drivers/Multimedia support, and Device Drivers/Soundcard support, we get the following. Note: I removed the multiplier_drive and added the IR_DEMOD driver instead, the difference still visible.



Conclusion:

Overall the lab was helpful in teaching another way to implement drives directory into Linux. I also learned more about configuring the menu and how it affects the size of the uImage. We also saw that, built in drives also can run executable files such as the multiplier.

Question:

one of the advantage of loadable kernel modules is that you can add them externally. You can put the drive into the sd card without having to recompile the uImage. With a built-in module you can have the program initialize at the beginning of the boot in system. Meaning there is less room for the program to have errors in managing the memory.

FEEDBACK:

Overall, I think the labs are helpful. They do help learn the material and their design very well with very little errors. Even when there are errors on the lab manuals, the TA is very helpful in providing aid to students.However, I do think that the time frame provided to demo and complete the labs is not the best. For instance, students should be allow to demo their previous labs after

the due date with a percentage taken off. The reason for this is that it encourages students to at least finish the labs and learn what the lab intends them to learn. Or there should be an option that if a student didn’t finish the lab, a solution manual should be provided so that the students that didn’t complete the lab can at least work on the next lab.