

Nicholas Tann

Lab 5

ECEN 449-503

2/24/2020

### Introduction:

This lab is intended to guide students through the process of running a simple 'Hello World!' kernel module on the FPGA. It will build off the last lab, where first we will need to get Linux running on the board. The Linux kernel will be used to allow dynamic loading of kernel modules.

### Procedure:

We use the contents of lab 4 which includes the Linux kernel source and boot Linux on the board to read and write files to it and test out the mount operation on the SD card. The next step was to cross-compile a kernel module on CentOS and load it onto the SD card. Lastly we created a Makefile listing all of the dependencies of our individual object files. We will take our hello.ko and load it onto the SD to be mounted on the FPGA. The same process was repeated with multiply.c

### Results:

After cross-compiling and instantiating the Hello World kernel module I was able to see the printk statement. Lastly, running lsmod showed that the 'hello' module was running. Multiply.ko also worked as intended.

## Conclusion:

In this lab, I was able to get a simple Linux kernel module up and running on the ZYBO board. I was guided through the process of creating a module whose purpose was to print a statement on the screen by instantiating it using the PICOCOM terminal.

## Questions:

1. Since the mounting directory we created was stored in RAM, we would need to recreate the directory if we were to reset the ZYBO board at that point.
2. The mount point for the SD card on the CentOS machine is media/ntann/1AAD-BD58
3. In the Makefile, everywhere we see hello.o we would need to change to whatever we renamed the file (keeping the .o extension). Unless you did the same procedure the lab 4 directory, you would not be able to see what the SD card has since the kernel support is not present.

```
KAMUISK: gzip image found at block 0
mmc0: new high speed SDHC card at address aaaa
mmcblk0: mmc0:aaaa SL08G 7.40 GiB
  mmcblk0: p1
EXT2-fs (ram0): warning: mounting unchecked fs, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem) on device 1:0.
devtmpfs: mounted
Freeing unused kernel memory: 212K (40627000 - 4065c000)
random: dropbear urandom read with 1 bits of entropy available
mmc0: card aaaa removed
mmc0: new high speed SDHC card at address aaaa
mmcblk0: mmc0:aaaa SL08G 7.40 GiB
  mmcblk0: p1
FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
random: nonblocking pool is initialized
Hello world!
Goodbye world!
zynq> rmdir hello
zynq> ls
BOOT.bin          hello.ko
FOUND.000          test
System Volume Information  uImage
devicetree.dtb     uramdisk.image.gz
zynq> rmdir hello
zynq> lsmod
zynq> ls
BOOT.bin          hello.ko
FOUND.000          test
```