ECGR 6181/8181 - Lab4

Objective: Writing kernel modules
Outcomes: After this lab, you will be able to

- Write and install a kernel module
- Pass parameters to a kernel module
- Write a kernel module that accesses kernel variable

Modules are pieces of code that can be loaded and unloaded into the kernel upon demand. They extend the functionality of the kernel without the need to reboot the system. For example, one type of module is the device driver, which allows the kernel to access hardware connected to the system. Without modules, we would have to build monolithic kernels and add new functionality directly into the kernel image. Besides having larger kernels, this has the disadvantage of requiring us to rebuild and reboot the kernel every time we want new functionality. Download the Lab 4 zip file from Canvas. Edit KERN SRC variable in the Makele to set it to your Linux kernel path from Lab 3

```
$ make
```

The kernel module hello world module.ko is generated. Copy it to the _install directory under Busybox. Follow the steps from Lab 3 and launch Linux under QEMU. At the emulated Linux prompt,

```
# mkdir -p /lib/modules/‘uname -r‘
# cp hello_world_module.ko /lib/modules/{your kernel version number}
# modprobe hello_world_module // Loads the module; prints "Hello, world1"
# lsmod // Verify that the Hello World module exists
# modprobe -r hello_world_module // Removes module; prints "Goodbye, cruel world"
```

Note: uname -r Linux command prints the kernel version. The apostrophe used is the one on the upper left hand corner of the keyboard.

Do the following

- Write a kernel module that accepts as command line parameters the name of the person to be greeted and the number of times the greeting is to be printed.
- System timers interrupt the processor at programmable frequencies. This frequency, or the number of timer ticks per second, is contained in the kernel variable HZ. The jiffies variable holds the number of times the system timer popped since the system booted. The kernel increments jiffies HZ times every second. Thus, on a kernel with a HZ value of 100, a "jiffy" is a 10-millisecond duration, while on a kernel with HZ set to 1000, a jiffy is only 1-millisecond. Write a kernel module (jiffies module) that uses the jiffies and HZ kernel variables to write the value of time since bootup to /proc.

Useful links -
https://devarea.com/linux-kernel-development-kernel-module-parameters/
https://devarea.com/linux-kernel-development-creating-a-proc-file-and-interfacing-with-user-space/