Lab #5

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 variables

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 following two files from Moodle hello_world_module.c and Makefile.

Edit KERN_SRC variable in the Makefile to set it to your kernel path.

$ make

The kernel module hello_world_module.ko is generated. Copy it to the _install/home/your_name 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` (uname -r Linux command prints the kernel version)
cp /home/your_name/hello_world_module.ko /lib/modules/{your kernel version number}
modprobe hello_world_module // Loads the module; prints “hello world”

lsmod // Verify that the Hello World module exists

modprobe -r hello_world_module // Removes module; prints “Goodbye, cruel world”

To do:

1. 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 (For example, “Hello Jane” printed 10 times). See Ch 2. of the book Linux Device Drivers, 3rd Ed. (Available for free at http://lwn.net/Kernel/LDD3/)

2. 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. Download procfs_example.c from Moodle that describes the use of procfs file system from within the Linux kernel. Use this code as a guide to writing the jiffies_module. Compile your module and execute it under QEMU. Examine /proc/jiffies psuedo file to see the output.