Lab 4: Write and Load a Kernel Module

Welcome to the 4th lab of the VE473 Advanced Embedded System course! Today we have two goals. First, we will build and install a kernel module. Next, we will write a kernel module and use it to observe changes in a variable.

Project Description and Exercise

First, on the Raspberry PI, please create a new directory to hold your kernel modules, e.g. /project/your-username/modules and cd into it. Save a copy of simple_module.c (a simple template for writing kernel modules in this course, based on the "Hello, World!" module shown on pages 338 and 339 of Robert Love's Linux Kernel Development, Third Edition) into that directory. Then compile the module. If it successes, it will produce a kernel module file named simple_module.ko.

Second, clear out the contents of the system log using the following command

sudo dmesg --clear

and then use the insmod utility to load your kernel module into the kernel, as in:

sudo insmod simple_module.ko

If you recieved no error messages, then your module has been successfully loaded. To confirm this, you can check the system log by issuing the command

dmesg

which prints out the system log, which now should show the message that was printed when the module loaded.

Third, to confirm your module was loaded, you can also issue the command

lsmod

to see a listing of all currently loaded kernel modules. Verify that your module appears in the list, and as the answer to this exercise, copy the output that was produced by lsmod.

Fourth, the basic utility for removing modules from the kernel is called rmmod. When using this tool you can either specify the module name (as shown in lsmod) or you can specify a .ko file, as in

sudo rmmod simple_module.ko

Fifth, the kernel module is widely used to implement the driver and access the hardware. In this part, we will implement a kernel module to access the hardware jiffies counter. Recall that this variable keeps track of how many timer ticks have occurred since system boot. Your kernel module can be developed based on the simple_module.c. In your kernel module, the value of the jiffies variable (hint: it's an unsigned long) to the system log when the module is loaded and unloaded. Test your kernel module in your Raspberry PI.

Submission to each exercise:

0. List the names of the people who worked together on this lab.
1. Please show the output after you compile the module.
2. Please show the message that appears in the system log.
3. Please show the output that was produced by lsmod.
4. Please show the output of lsmod and the line of the system log, which show that the module was unloaded.
5. Load and unload the kernel module, please show the system log message that shows the values of the jiffies variable when the module was loaded and unloaded.
   Meanwhile, please also submit your kernel module code.

Advanced Topic (not included in the lab grade):

1. Write a normal user program to read the hardware jiffies counter. Compare the time between two continuous reads of the jiffies counter with the implementation in the step 5.