CS446 Lab 2 - Linux Kernel Modules

Outline:

In this lab, you will learn how to create a kernel module and load it into the Linux kernel. The lab should be completed using the Linux virtual machine that you installed in Lab 1. Although you may use an editor to write these C programs, you will have to use the ***terminal*** application (on the virtual machine) to compile the programs, and you will have to enter commands on the command line to manage the modules in the kernel.

As you'll discover, the advantage of developing kernel modules is that it is a relatively easy method of interacting with the kernel, thus allowing you to write programs that directly invoke kernel functions. You need to keep in mind that you are indeed writing ***kernel code*** that directly interacts with the kernel. That normally means that any errors in the code could crash the system! However, since you will be using a virtual machine, any failures will, at worst, only require rebooting the system.

To complete Lab 2, do the following:

A) Follow the steps in the attached file (file) for:

- Creating kernel modules.
- Loading and removing kernel modules.

B) Save a copy of simple.c file.

C) Edit simple.c file to add the following functionality:

- Declare and initialize an integer `'i=3'` in `simple_exit (void)`.
- Write a `void` function `printint (int *ptri)` that takes the address of `i` as a parameter and prints a message of your choice `(* ptri)` times. Note that `(* ptri = value at the address stored in ptri)`.
- Call the `printint` function in `simple_exit (void)`.
- Compile the file to create the kernel module without errors using the `Makefile`.
- Load the kernel module `'simple'`.
- Use the `dmesg` command.
- If all the above steps were performed correctly, you should see the loading module message and the message you typed `(* ptri)` many times.

- Now follow the steps to remove the kernel module.
- Print the output you see on the terminal in a text file named output.txt.

Note:

- To load and remove kernel modules, you can use the code (simple.c) in the attached file (file).
- To edit (simple.c) file to add the functionalities, you can follow loading and removing kernel modules in the attached file (file).