CS492 Lab 3
Kernel modules and device drivers

This is an individual assignment. Individual assignments, as the word indicate, are to be done **INDIVIDUALLY**. Any sign of collaboration will result in a 0 and being reported to the Honor Board.

Introduction

The objectives of this project is to familiarize yourself with the kernel source code, using the git version control software. Specifically, you will:

- Develop and deploy kernel modules;
- Implement a simple character device driver.

All the project steps will be performed in the supplied Debian virtual machine. The following concepts from the course will be put in practice in the project:

- Kernel module development;
- Kernel source code cross-references;
- Kernel device drivers.

Project Steps

a) You can reuse the same identical virtual machine used in Lab 1. Alternative you may consider to repeat the installation of the virtual machine as described next. Download the Debian virtual machine from Canvas, use the provided instructions to import it in VirtualBox. Eventually, customize the number of CPUs and amount of memory according to your computer. It is recommended to configure the virtual machine with at least 2 CPUs and 4GB of RAM. The user "student" has been created with associated username: "student" and password: "cs492"; the super-user's username is "root" and its password is "cs492". There is a graphical user interface. There are at least the following editors: ***vi***, ***vim***, and ***nano***. The source code of the Linux kernel is in the ***linux-4.9/*** subdirectory of the "student" 's home folder. The kernel code has been compiled at least once. Please don't use this directory for your work, the code for this assignment should be created outside the Linux kernel source tree.

b) Create a git repository in **github classroom** using a different classroom based on the section you are enrolled.

c) Use the git repository to track all your kernel module development, you don't need to track modifications to the Linux kernel source – this is because this assignment doesn't require any modification to the Linux kernel source. That is the big advantage of kernel modules.

d) Using the provided code base as your starting point, develop, deploy and demonstrate working two kernel modules:

   a. Hello: Displays simple greetings when it is loaded and unloaded, with a username that is specified when the module is installed into the kernel.

b. Qmod: Provides a shared queue of string messages via a character device. You are provided with two userspace clients, one that adds a message to the queue and one that removes a message from the queue. You will need to implement the queue as a linked list data structure[1]; remember that you cannot use `malloc` in kernel code!

e) For each module, record a video of the module being deployed and used, and the resulting kernel log output. For the qmod module, you will need to show the client operations being run (several times) to add and remove messages, and the resulting log. You should add several messages to the queue before removing (and continue to add and remove), to demonstate a queue of size greater than one. The queue should be implemented as a linked list, the API does not allow bounded queue size. Your name should be visible in these videos, being output as a result of the modinfo command.

[1] As you might expect, the kernel does include a generic implementation of queues, `kfifo`, but you are better off implementing your own.