Programming Assignment 2

Submit Assignment

Due Mar 29 by 6pm **Points** 100 **Submitting** a file upload **Available** after Mar 1 at 7:15pm

Learning Objective: To gain a fuller understanding of kernel modules by creating one.

Development: Use gcc and the C programming language, on the version of Linux we have in our VM.

Assignment: Write a character-mode Linux device driver as a kernel module.

Your driver must:

- Store bytes written to it up to a constant buffer size (at least 1KB)
- Allow them to be read back out in FIFO fashion
- Remove them from the buffer as they are read back out

Additionally, the driver must handle the following overflow conditions:

- If not enough buffer is available to store a write request, the driver must store only up to the amount available
- If not enough data is available to service a read request, the driver must respond with only the amount available (including 0 bytes)

The driver will need to:

- Successfully initialize and de-initialize itself, including registering itself and obtaining a new major device number.
- Report using **printk** each time its character device is opened, closed, read or written.

To do this, you will need to implement the expected device driver functions to:

- Initialize the kernel module.
  - …including registering the device.
  - You do not need to create the device file, but you do need to log the device number you are assigned so that **mknod** can be called to do so after your kernel module is installed.
- De-initialize the kernel module.
  - …including de-registering the device.
- Open the device.
- Close the device.
- Read from the device.
- Write to the device.

Below are some resources that will help you structure a device driver. Feel free to ask the professor and the teaching assistants for advice, but you should read over the relevant bits of these resources first.

Peter Jay Salzman, Michael Burian, Ori Pomerantz. *The Linux Kernel Module Programming Guide.* **http://www.tldp.org/LDP/lkmpg/2.6/html/ (http://www.tldp.org/LDP/lkmpg/2.6/html/)**

Derek Molloy. *Writing a Linux Kernel Module - Part 2: A Character Device*. **http://derekmolloy.ie/writing-a-linux-kernel-module-part-2-a-character-device/ (http://derekmolloy.ie/writing-a-linux-kernel-module-part-2-a-character-device/)**

Submitting

Zip up your code, and upload it to Webcourses. Include the makefile.