COP 4610 Operating Systems Principles

Summer C 2017 – Programming Assignment 1

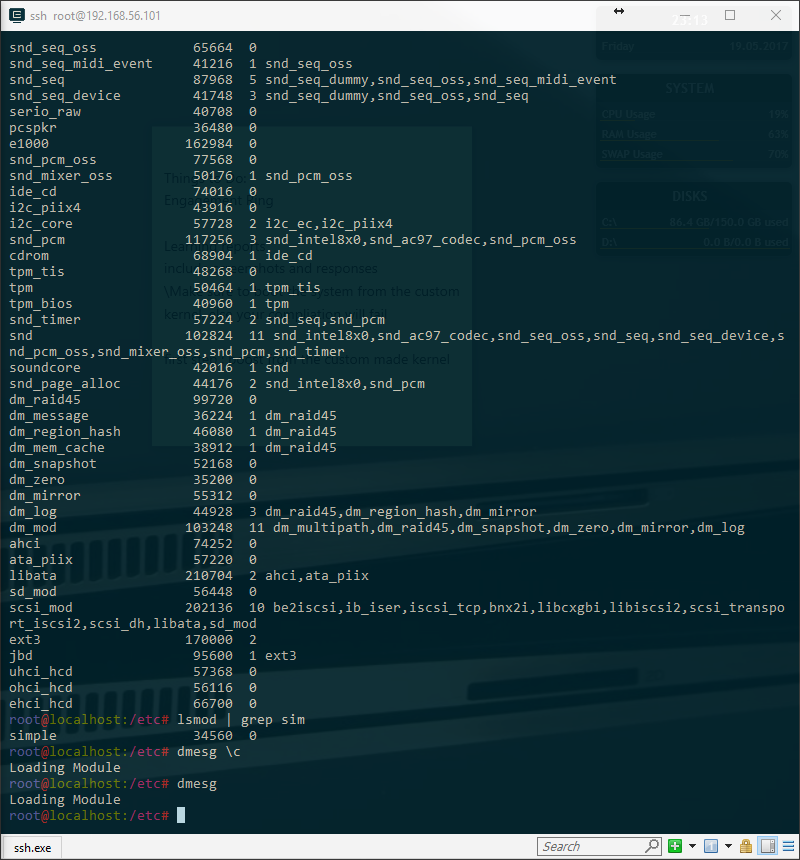
Hamilton Chevez PID: 3350827

Part 2A

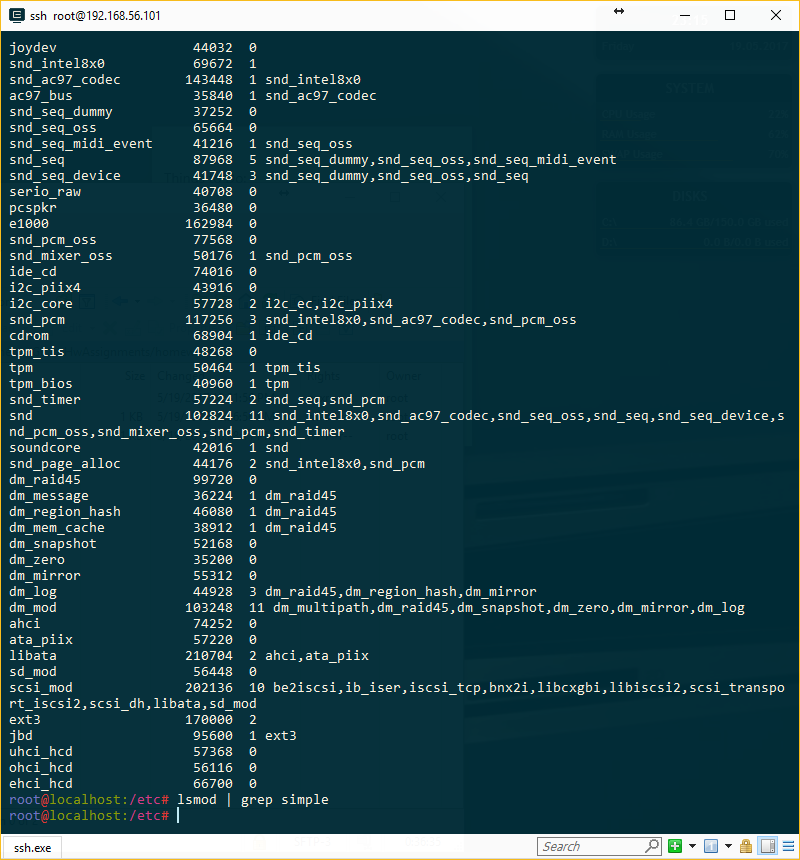
After compiling my module, I ran lsmod and got a listed of all the modules currently loaded in the kernel.



Afterwards, I loaded my module into the kernel by running “insmod simple.ko”. After calling “service syslog restart” and lsmod I did not see anything regarding the module I had loaded. It may be possible that the “simple” module appeared earlier in the list. Hence I ran “lsmod | grep sim” and so I confirmed that my module had been loaded.



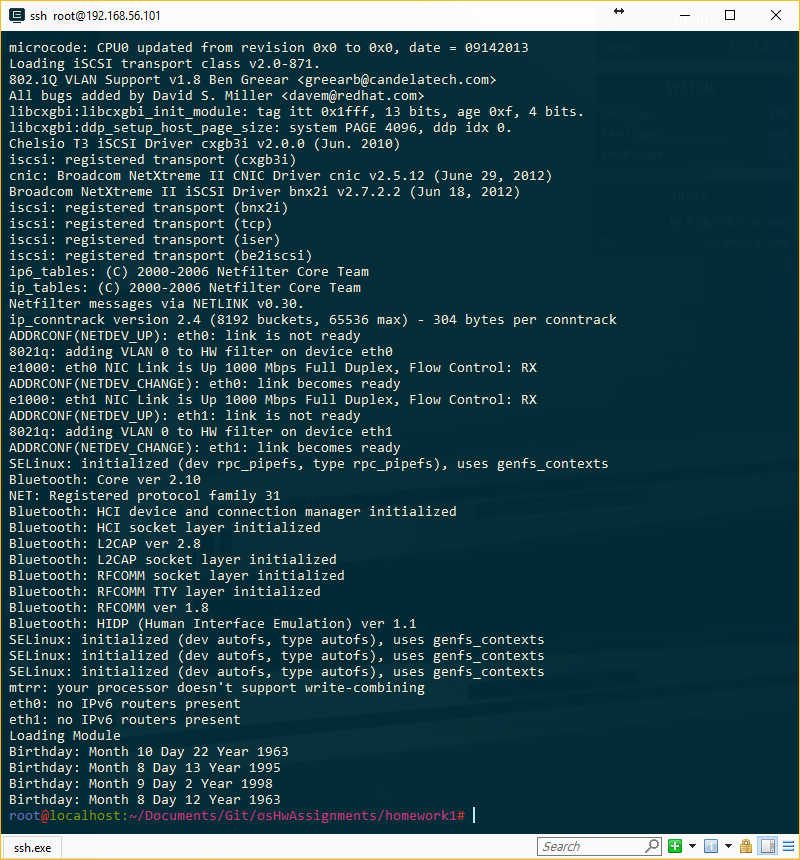
Afterwards I unloaded my simple module from the kernel by running “rmmod simple”



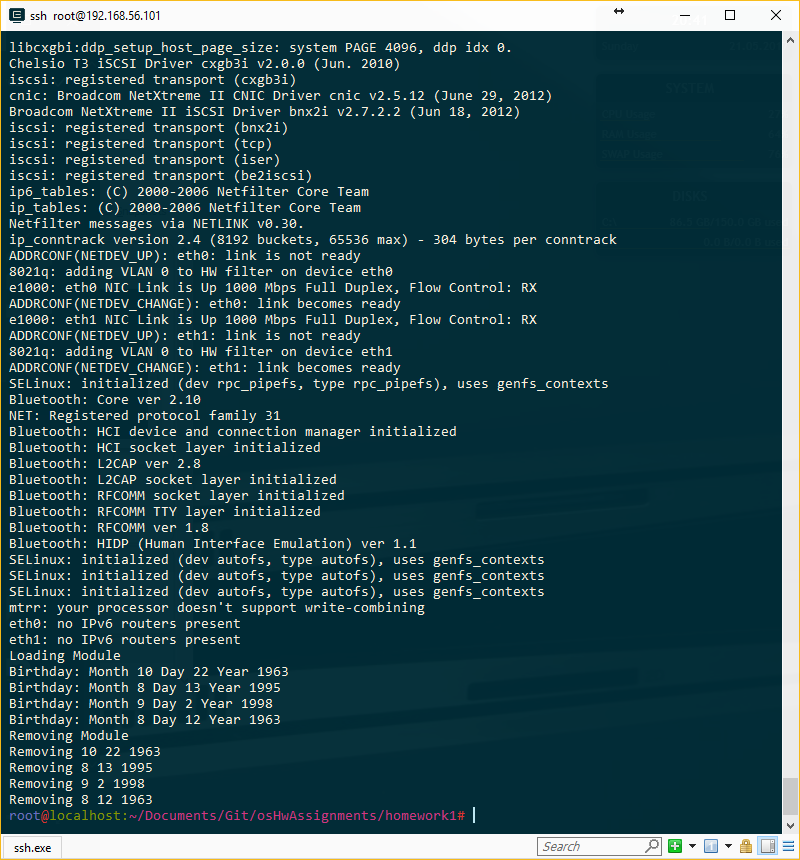
I ran lsmod | grep simple to make sure to list the module I wanted and the list resulted empty, confirming that my module was no longer part of the kernel.

Part 2B

The following are screenshots of my work as I progressed through part 2b of the homework. First of when I loaded the birthdays into the kernel.



And after I removed Birthdays from the Kernel



**Report On What I learned**

All modules should have two functions defined,

Int simple\_init(void) and void simple\_exit(void)

that are called when they are loaded and removed from the kernel. The bash command lsmod is synonymous to the regular ls command with the exception that what it lists are the current modules loaded into the kernel. In order to insert a module into the kernel, it first has to be compiled from its c format to the appropriate file type .ko . Afterwards, it can be inserted by calling insmod “name of file”.ko.

To understand better what the linked list of this homework assignment was doing I explored the github repository of Linus Torvalds to see the implementation.

https://github.com/torvalds/linux/blob/master/include/linux/list.h