Assignment 3 Report

Part A:

- To access our virtual machine, we used Putty and connected to remote.cs and used ssh <ip address> to connect our virtual machine.
- In case of kernel panic we used pulse secure application to reboot/shutdown our virtual machine.

Part B:

- We downloaded “hello.c” and “hellon.c” from the course website. We created a makefile and used “make” command to compile both the files and we got “hello.ko” and “hellon.ko” files.
- To load the kernel module, we used “sudo insmod hello.ko” (sudo allows us to gain root access).
- We used “dmseg” command to check the kernel log if “mymodule: Hello World!” was printed. (Here dmseg refers to diagnostic message which prints the message buffer of the kernel).
- To unload the kernel module, we used “sudo rmmod hello”.
- We used dmseg command again to check the kernel log if “mymodule: Goodbye, cruel world!!” was printed.
- For “hellon.c” we used the insertion command “sudo insmod hellon.ko whom=class howmany=10” and got the result as “Hello Class!” printed ten times. (“Whom” is a static char which was initialized to “world” and “howmany” was a static int which was initialized to “1”). After passing arguments “whom=class” and “howmany=10”, the value of “whom” and “howmany” got updated.

Part C:

- We created the miscellaneous character device driver by following the example linked on the course website
- We modified the example to implement the .read file operation to return “Hello World!” message to a user space program that invokes the read() system call on the miscellaneous device.
- We used “ls /dev/sud*” to check if our device driver was successfully registered.
- To read “Hello World!” from Kernel Module we use the command “sudo ./demo” (Here, “demo.c” is our user space program)

Part D:

- We downloaded the entire linux source code onto our VM using the “wget” command followed by the download link (e.g., wget https://cdn.kernel.org/pub/linux/kernel/v4.15.1/linux-4.15.1.tar.xz).
- We extracted the file using “tar -xvf linux-4.15.1.tar.xz”.
- We installed cscope using the command “sudo apt install cscope”.
- First, we need to go inside the directory where we need to index files.
- We will use the command “export CSCOPE_EDITOR=`which nano`” to change our primary editor as nano.
- We use the command “find . -name "*.c" -o -name "*.h" > cscope.files” to pass all the files with .c and .h extensions to cscope.files
- Now we use the command “cscope -q -R -b -i cscope.files” to index all the files passed.
- To explore the indexed files, we use the command “cscope -d”.