Instructions for opening the vm

1) ssh into attu

2) navigate to the folder that contains the vm image

3) run the practice.sh script

    3.1) **chmod +x practice.sh** (only if you don't already have practice.sh set to executable)

    3.2) **./practice.sh**

4) open a new terminal and ssh into the same attu server

5) **ssh -p 6500 suneja@127.0.0.1**

6) **password: suneja**

I used this blog to complete this assignment: https://blog.sourcerer.io/writing-a-simple-linux-kernel-module-d9dc3762c234

Instructions for installing headers

1) **sudo sh** #to get to the root

2) **apt-get install build-essential linux-headers-`uname -r`**

3) **exit** #once everything has finished installing

Instructions for creating a kernel module

1) **mkdir akhila-assignment**
2) copy **hello_world.c** and **Makefile** into this folder
3) **make** #run make
4) **sudo insmod hello_world.ko** #insert the module
5) **sudo dmesg** #opens kernel log, should see Hello World at the bottom
6) **sudo rmmod hello_world** #removes module
7) **sudo dmesg** #opens kernel log, should see Goodbye! at the bottom

Things that went wrong and how I fixed them

The first time I tried running **apt-get install build-essential linux-headers-`uname -r`,** I got an error saying:

```
E: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 4772 (unattended-upgr)

N: Be aware that removing the lock file is not a solution and may break your system.
```

```
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), is another process using it?
```

This is how I resolved it:

1) **ps -eaux** #to find the process 4772

2) **ps -eax | grep 4772** #to see what the process is, it is an old python script

3) **kill -9 4772**

4) **apt-get install build-essential linux-headers-`uname -r`** #run this again

I copied my hello_world.c and Makfile code from the internet, so I had some formatting errors.

I found my Makefile errors by doing a dry run, **make -n**, and googled the errors I received (just some indenting problems).

To find my hello_world.c errors I ran **make > errors.txt 2>&1** to easily view all the errors in a single file. Turns out the code I copied used ASCII quotation marks instead of Unicode.