Jonathan Carreiro

Lab 10

Part 1:

a. Information retrieved by ipcs before instruction a:

```
jcarreiro@LAPTOP-Q8L7674C:~/repos/CIS-370/Lab10$ ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
```

This simply tells me that the system has no message queues, shared memory segments, or semaphore arrays.

b. The information retrieved by ipcs after instruction a is exactly the same which means that compiling the two programs did not create any of the above structures.

c. You need to run printLine first because it creates the shared memory.

d. Running the printLine program causes it to print the character ‘a’ 10 times every 4 seconds. This happens infinitely until the program is forced to exit by the user.

Information retrieved by ipcs after instruction d:

```
jcarreiro@LAPTOP-Q8L7674C:~/repos/CIS-370/Lab10$ ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x0000007b 0          jcarreiro  666        8          0

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
```

This tells me that this program created a shared memory segment with a key 123 (represented in hex), an id of 0, that I own, has all permissions by default, is 8 bytes of memory, and has nothing attached to it.

e. The information retrieved by ipcs in the second shell before executing changeLine is the same as the information retrieved in the first shell after executing printLine. In

addition to my previous interpretation, this tells me that the shared memory was created across the entire system.

After running the changeLine program multiple times while running both programs in different shells, I got the following every time:

```
jcarreiro@LAPTOP-Q8L7674C:~/repos/CIS-370/Lab10$ ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x0000007b 0          jcarreiro  666        8          1

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
```

The two things of note here are that there is only one segment of shared memory and that the number for nattch is now 1. The reason that there is only one segment of shared memory despite both programs creating one is that they both use the same key. Since there is already a segment created with that key, trying to create another with that key does nothing. As for the value of nattch being 1, this is because there is now one other program that has attached this segment of memory to itself.