**Lab 05 - CIS 452**

**Compile and run sampleProgramOne.**

1. **What is being output by sampleProgramOne (i.e., what is the meaning of the output values) ?**

sampleProgram1 is outputting the range of the memory addresses of the shared memory.

1. **Read the man pages; then describe the meaning/purpose of each argument used by the shmget() function call.**

There are 3 arguments used by the shmget() function call: key, size, and shmflg. The first argument, key, specifies the unique key, and for IPC\_PRIVATE specifically, it generates a unique key that is unique to our specific process. The key is also compared to other keys within the shared memory segment. The second argument, size, specifies the size of the shared memory space. The third argument, shmflg, returns an integer that refers to flags separated by bitwise operators (“|”) that specify the permissions of the shared memory space. Some of the permissions include creating the memory space if it’s not already created (IPC\_CREATE), read permissions for the user (S\_IRUSR), and write permissions for the user (S\_IWUSR).

1. **Describe two specific uses of the shmctl() function call.**

We can use the shmctl() function call to find out which process last made a change to the shared memory. A call that would accomplish this would look like the following: shmctl(shmId, IPC\_STAT, &shmDataStructure); where shmId is the identification number of the shared memory, and shmDataStructure is a struct shmid\_ds where the the information about the shared memory will be placed. We can also use the shmctl() function to set the read and write permissions of the shared memory. A call that would accomplish this would look like the following: shmctl(shmId, IPC\_SET, &shmDataStructure); where shmId is the identification number of the shared memory, and shmDataStructure is a struct shmid\_ds where shm\_perm.mode (type mode\_t) has been set to whatever permissions we want for the memory range (Set the same way we assign in shmget() were we OR multiple flags together).

1. **Read the man pages, then use shmctl() to modify sampleProgramOne so that it prints out the size of the shared memory segment. What changes/lines do you have to add to the program?**

First we need to create a shmid\_ds struct to store the shared memory details in. Then we call shmctl() with shmid, the command we are running (IPC\_STAT for status/information), and a pointer to the shmid\_ds struct. We then need to make sure the function completed properly by checking the return value of the function. After making sure everything worked, we can access shared memory segment size (segsz) in the shmid\_ds struct we created earlier and print it.