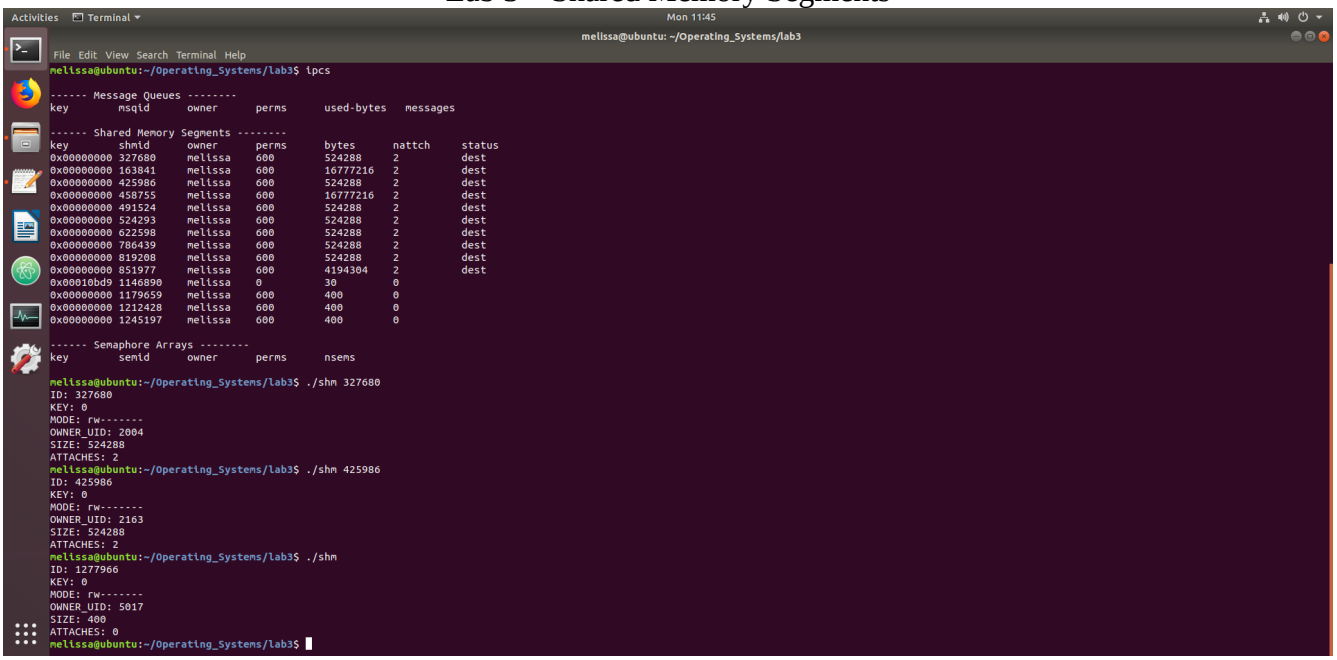
Lab 3 – Shared Memory Segments



In order to run and compile my program, first run “make” to check if any updates to the C file have been made. This will compile the program and create an executable to be run on command. Since this program finds the ID, key, mode, owner_uid, size, and attaches, we want to choose a segment to pull these values from. In order to choose a shared memory segment, run the command “ipcs” to view shared memory segments running on your computer and take the shmid of one segment. We can now execute the program that was written using “./shm” followed by the chosen shmid. As seen above, when inputing the shmid, the program finds the wanted information from the id provided. If no shmid is provided, the program generates a random one and executes the program normally.
This program finds the id by either taking user input or generating its own id by using the function shmget(), which allocates a System V shared memory segment. Assuming the id exists, using the function shmctl(), which checks if the id is a valid segment, the mode is then found by accessing the data structures shmid_ds and ipc_perm. Shmid_ds contains the variables struct ipc_perm shm_perm, size_t shm_segsz, time_t shm_atime, time_t shm_dtime, time_t shm_ctime, pid_t shm_cpid, pid_t shm_lpid, and shmatt_t shm_nattch. In ipc_perm, which is contained in shmid_ds, the variables key_t __key, uid_t uid, gid_t gid, uid_t cuid, gid_t cgid, unsigned short mode, and unsigned short __seq are accessible. When getting the value of mode from ipc_perm, the number obtained is then converted to a readable character such as “r” for “read,” “w” for “write,” or “rw” for “read-write.” Finally, the program then gets the remaining values wanted from their respected variables in the shmid_ds and ipc_perm data structures and displays them to the user.