**CECS 326-09**

Operating Systems

Jessie Lazo ID:017165640

Assignment 2

Due Date: 9-28-2020

Submission Date: 9-29-2020

Program Description

The purpose of this assignment is to understand and implement Linux’s mechanism of interprocess communication, the message queue. This was achieved in this assignment by creating the three files master.cpp, sender.cpp, and receiver.cpp which handled their respective roles utilizing the Linux system calls. In master.cpp a message queue is created via msgget() in which the key\_t type “IPC\_PRIVATE” and flag field “IPC\_CREAT” return the identifier associated with the value of the key argument. Within the master class two subprocesses are created by using the fork() system call and executed with execlp() which executes the file at the specified pathname which in this case are /sender and /receiver. In sender.cpp the struct msgbuf{..} is defined, as well as within receiver.cpp, which acts as the pointer argument for both msgsnd() and msgrvc(). msgsnd() identifies which message queue by the identifier that was passed in via a commandline argument and then appends a copy of the user’s inputted message that is pointed to by the message buffer. Once the message is sent the sender can then terminate and within receiver.cpp the message is taken out of the message queue by the system call msgrcv() as it identifies which message queue by the identifier argument and then the message is placed in the buffer. Once this interprocess communication is done the message queue is then removed by system call msgctl() which performs control operations on a given message queue if the given argument is “IPC\_RMID”. The program is complete after those processes run successfully.