**Design of Sender and Receiver**

Members: Emmanuel Khlot, Marco Botello, Sahil Sheth, Cesar Martinez Melgoza

1. **Problem Statement**

We were challenged to use recently acquired knowledge of shared memory and message queues to create an application which transfers files between two processes. A sender program and receiver program in which the sender would send the files to the receiver process. The receiver would implement the process that receives from the sender process in which later would invoke as ./recv where recv is the name of the executable file.

1. **How to use your program**

To use this program, you must first locate the folder titled Sender-Receiver-master. Within this folder you will find the .cpp files which would then need to be compiled using make. After makefile is executed, using ‘,/sender <TEXTFILE> would run the sender on a file and perform the task as described above, sending the file to the receiver.

1. **Design of your program**

What was used the most in this program were variables to transfer the files and parameters to notify the progress of the sender and receiver functions. We used pointers and strings to be able to allow the sender and receiver tasks to be able to forward the files and confirmations to allow us to know and understand the status and outcome of such operations.

1. **Any Limitations**

There were no limitations set in the application.

1. **Any shortcomings**

Not all compilers are created equal, some of the contributors had faced exclusive errors when attempting to compile the application. Another shortcoming was that the recv.o file has a segmentation fault as pictured below.

