CPSC_351_Assignment_1

CPSC 351- Section 2(13643)
Spring 2019

Group Members:

James I Ku
894841865
thatoneddrguy@csu.fullerton.edu

Esteban Montelongo
888847456
EstebanMontelongo@csu.fullerton.edu

Bony Roy
898161054
broy91@csu.fullerton.edu

Purpose:
Use shared memory, and message queues in order to implement an application which synchronously transfers files between two processes.

Programming Language Used:
C++

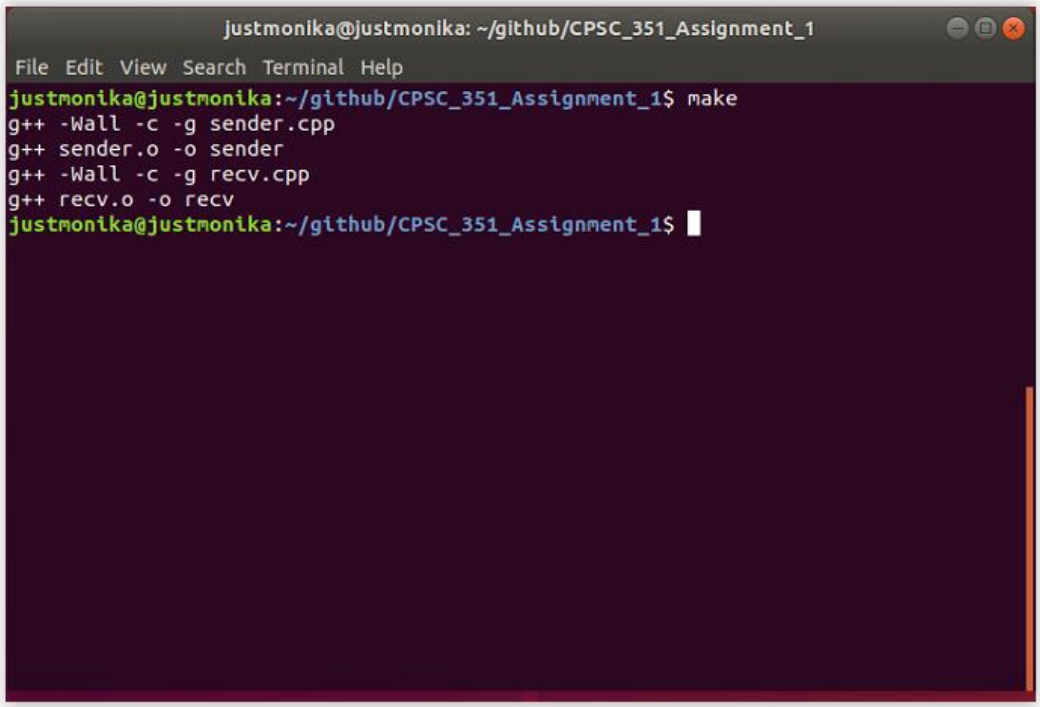
Extra Credit:
Not Implemented

File Names:

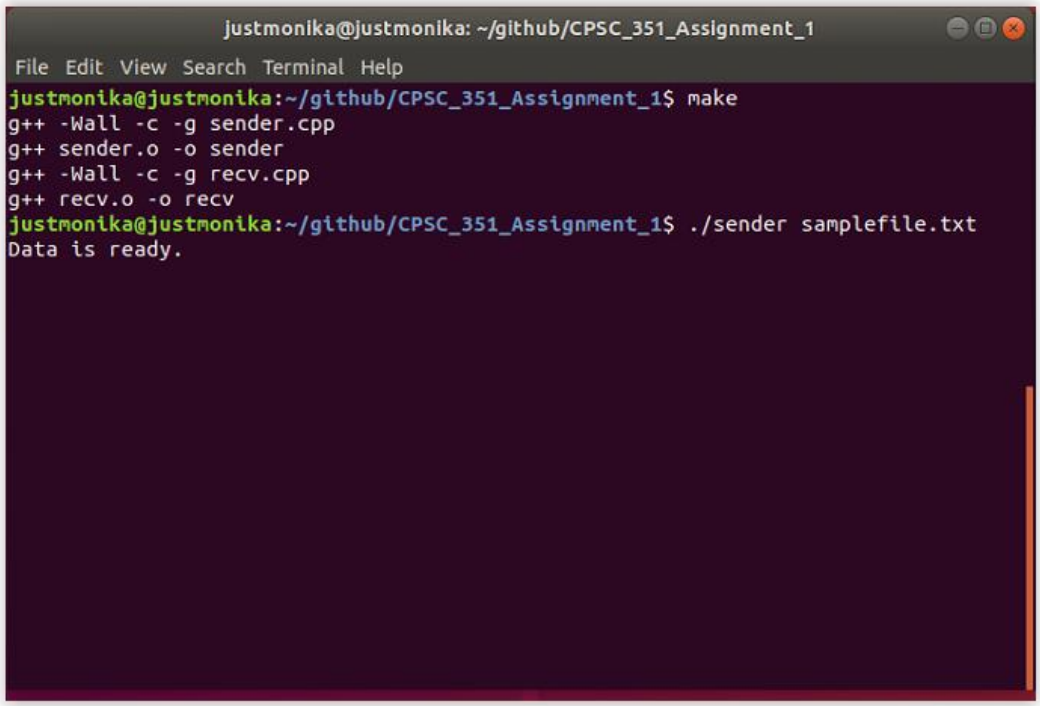
1. p1-roykumont.tar
2. sender.cpp: CPP File that sends the message from the text file to recv.cpp
3. recv.cpp : CPP File the receives the message from the sender.cpp
4. keyfile.txt: Text file that holds a string that is used to generate the same key for both the sender and the receiver.
5. msg.h: Header File that holds a struct of the message relayed through the message queues
6. Makefile: Makefile to build both the sender and receiver files

To run our program:

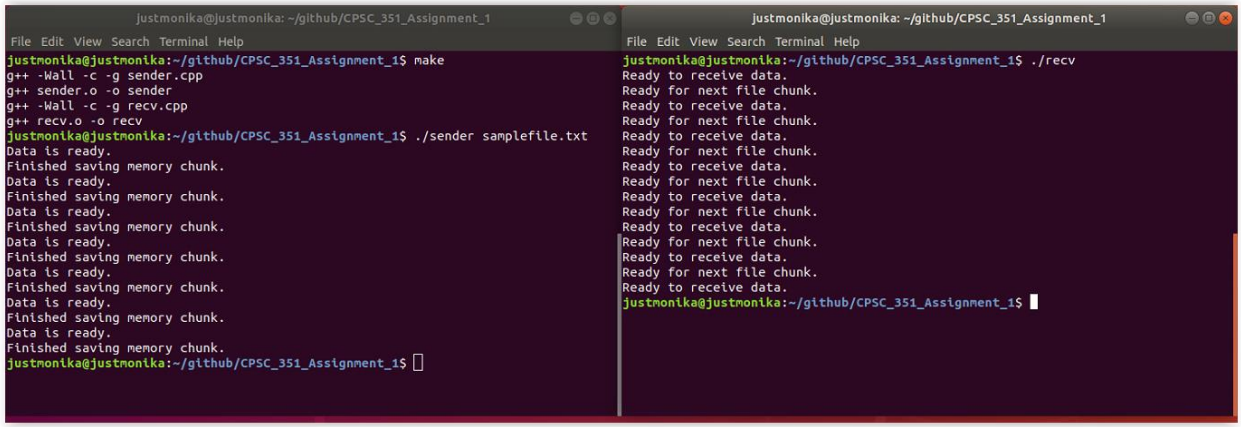
1. Download tar archive.
2. Extract files from tar archive.
3. Open terminal in directory where files are located.
4. In terminal type make and press enter.



5. Sender: open termimal in directory that contains the sender file. run : type ./sender <name of file>,press enter.



6. Receiver: open new terminal in directory containing file. run: type ./recv, press enter. Received file is saved as recvfile.



Team Collaboration:

Our team met up three times at the CSUF library to research the necessary system function calls and collaborate on the code via GitHub.

Sources:

Professor Yun Tian, Skeleton Code posted on Titanium and relevant links posted in document.