**CMSC 621**

**Advanced Operating Systems**

**Ajinkya Wakhale**

**ReadMe File**

For part 1 and part 2, there are one program called sender\_test.c

**To compile** programs

**gcc -pthread -o sender\_test sender\_test.c**

**gcc -o sequencer sequencer.c**

**To run the executable-**

./sender\_test 0 messagetobemulticasted

./sender\_test 1 message1

./sender\_test 2 message2

./sender\_test 2 message3

./sequencer

Make file name is **Makefile.mk**. I will compile both sender\_test and sequencer program. Once programs are complied, you can run it as mentioned above.

Number of processes can be of any number. You just have to change rank and a general message which you want to want to send for each individual process. This general message will be changed in the sender thread, if you are sending multiple messages from one process.

**All the processes wait for the master to join, hence master process or rank 0 process should be run at last.**

For first part, it will give Clock synchronized message and will print local clock value of all the processes. For second part, it will print both the messages it is receiving and the order in which it will send the message. Output of message will be in the order which sequencer receives. Sequencer should always be running.

For third part, at the starting of all processes, it will read the value from the file, update it and write it back to the file. It will print (initial value+1) and then the value is again read back from the file and its value is printed. You can run as many processes you want, value will always be consistent.