

Sender

[./sender keyfile.txt]

keyfile.txt

[file from Sender]

N number
of bytes

Shared Memory

Number of
bytes read

Message Queue

Successful
Reception

recvfile

Reciever

[./recv]

EXIT program

if size ==0

if size != 0

SENDER PROGRAM (LABELED IN BLACK)

1. SENDER IS INVOKED WITH ./SENDER KEYFILE.TXT
2. PROGRAM ATTACHES TO SHARED MEMORY AND MESSAGE QUEUE
3. READS PREDEFINED NUMBER OF BYTES FROM FILE AND STORES IT IN SHARED MEMORY
4. SEND MESSAGE TO RECEIVER VIA MESSAGE QUEUE CONTAINING A SIZE VARIABLE INDICATING HOW MANY BYTE WERE READ FROM THE FILE
5. WAIT ON MESSAGE QUEUE TO RECEIVE A MESSAGE FROM RECEIVER CONFIRMING RECEPTION
6. REPEAT STEPS 3-5 UNTIL SIZE VARIABLE IS ZERO.

RECEIVER PROGRAM (LABELED IN GRAY)

1. RECEIVER IS INVOKED AS ./RCV
2. SETS UP CHUNK OF SHARED MEMORY AND A MESSAGE QUEUE
3. WAIT ON MESSAGE QUEUE TO RECEIVE FROM SENDER PROGRAM, MESSAGE WILL CONTAIN A FIELD CALLED "SIZE" DENOTING NUMBER OF BYTES THE SENDER SAVED IN SHARED MEMORY
4. IF SIZE IS NOT ZERO, THE RECEIVER READS NUMBER OF BYTES AND SAVES THEM ON FILE CALLED RECVFILE. SENDS MESSAGE TO SENDER TO CONFIRM RECEPTION
5. OTHERWISE IF SIZE IS ZERO, PROGRAM CLOSES FILE DETACHES SHARED MEMORY AND MESSAGE QUEUES AND EXITS