**SOFTWARE DESIGN MODEL**

Create file

|

|

🡪 Read file into sender file

|

|

🡪 Send message to recv file

saying it is ready

|

|

🡪 Receive message and get the size

of message

|

|

🡪 Send message back saying ready for more

|

|

🡪 If no more file to read,

Ctrl + C to end messages via signal

handler

**PSEUDOCODE**

**sender.cpp**

init:

key —> filename (ftok)

shared memory id —> key & shared memory chunk size (shmget)

shared memory pointer —> shared memory id (shmat)

message queue id —> get message from key (msgget)

cleanup:

shared memory pointer —> safely delete (shmdt)

shared memory id —> safely delete (shmctl)

message queue id —> safely delete (msgctl)

send:

open file

check size of message to be sent —> read characters from shared memory pointer (fread)

send message —> send message given message queue id and address of sender message (msgsnd)

receive message —> receive next message (msgrcv)

size of message to be sent —> set to 0

send message —> send message given message queue id and address of sender message (msgsnd)

close file

main:

call init

call send

call cleanup

**recv.cpp**

init:

key —> filename (ftok)

shared memory id —> key & shared memory chunk size (shmget)

message queue id —> get message from key (msgget)

shared memory pointer —> shared memory id (shmat)

mainLoop:

open file

receive message —> receive message given message queue id and address of receiver message (msgrcv)

message size —> set to the size of receiver message

shared memory pointer —> save the shared memory to file (fwrite)

receiver message —> set its message type to “recieved” (RECV\_DONE\_TYPE)

send message —> send message given message queue id and address of receiver (msgsnd)

receive message —> receive next message given message queue id and address of receiver message (msgrcv)

message size —> set to the size of receiver message

cleanup:

shared memory pointer —> safely delete (shmdt)

shared memory id —> safely delete (shmctl)

message queue id —> safely delete (msgctl)

ctrlCSignal:

call cleanup

main:

call signal handler—> set to ctrlCSignal

call init

call mainLoop