**Design of Assignment 1**

**Abstract**

For this assignment, our objective is to analyze the whole concept of process synchronization, where we simulate on how two processes communicate with each other by constructing two simple programs where they both communicate with each other. Our method of how both programs will communicate with each other is by utilizing sender and message queues, with the use of signals to ensure non-conflicting synchronization.

**Design**

We will have 2 programs, sender and receiver with the following pseudo code

Sender:

Void init(int& shmid, int& msqid, void\*& sharedMemPtr){

//Create keyfile.txt contain “Hello World”

Print “making keyfile”;

Generate key;

Get id of memory segment;

Attach to the shared memory;

Attach to the message queue;

}

//Performing cleanup

void cleanUp(const int& shmid, const int& msqid, void\* sharedMemPtr)

{

Detach from shared memory;

exit

}

//Main function

void send(const char\* fileName){

Open file for reading;

Store buffer to the receiver;

Store buffer from the receiver;

Condition error if the file is open correctly;

while(!feof(fp)){

Loop reading the whole file

Wait until receiver sends back a message stating memory chunk is saved;

After no more data to send, send SENDER\_DATA\_TYPE with size field 0;

Exit

}

**Receiver:**

void init(int& shmid, int& msqid, void\*& sharedMemPtr){

//Create keyfile.txt contain “Hello World”, similar with sender file

Print “making keyfile”;

Generate key;

Get id of memory segment;

Attach to the shared memory;

Attach to the message queue;

}

//Receive the message

void mainLoop(){

Open file for writing;

Error check;

Receive the message and size. If SENDER\_DATA\_TYPE not 0, copy

while (msgSize != 0){

if(msgSize != 0)

{

Save shared memory to file;

Tell sender ready for next file chunk;

}

}

//Cleanup

void cleanUp(const int& shmid, const int& msqid, void\* sharedMemPtr){

Detach from shared memory;

Deallocate the shared memory chunk;

Deallocate message queue;

}

//Exit Signal

void ctrlCSignal(int signal){

Call cleanup to free system resource;

}

int main(int argc, char\*\* argv){

//Install Signal handler

Call signal;

Initialize;

Go to mainloop;

Call cleanup function to detach memory segment;

Return;

}