Design of Assignment1

Sender program

Void init{

Create file;

Create key: ftok;

Get the id of the shared memory: shmget;

Attach the id to the shared memory: shmat;

Check if the id attach to the shared memory;

Attach to the message queue: msgget;

Check if the id attach to the message queue;

}

Void Cleanup{

Detach from share memory: msgctl;

}

Void send{

Open file;

If it is not end of file{

Read file and store it in the shared memory;

Send message to receiver;

Check if the message is sent;

Wait for the receiver to receive the message;

}

Tell the receiver there is nothing more to send;

Close file;

}

Int main

If less than two argument {

Error and exit;

}

Init();

Send();

Cleanup();

A close up of a map

Description automatically generated

Receiver program

Void init {

Create file: ftok;

Allocate shared memory: shmid;

Attach it to the shared memory: shmat;

Check for errors;

Create a message queue: msgget;

}

Void mainloop {

Open file;

Read file;

Get message and message size;

If size != 0 {

Save the shared memory to file;

Tell the sender to keep sending;

} else { close the file; }

Void cleanup {

Detach from shared memory: shmdt;

Deallocate the shared memory chunk: shmctl;

Deallocate the message queue: msgctl;

Void ctrlCSignal {

Cleanup();

}

Int main {

ctrlCSignal();

init();

mainLoop();

cleanup();

}

A close up of a map

Description automatically generated

Interaction between the processes:A close up of a map

Description automatically generated