Q2.

Unix domain Sockets

There are 2 files attached:

S1.c→sender

S2.c →reciever

First we generate an array of random 50 strings in S1 and than we make use of functions like socket(to create a socket), bind(binds the socket to the address and port number), listen(to wait for the client to approach the server), accept(to extract the first connection request on the queue) in S1.

In S2 we use functions like socket to create socket and then finally connect function to establish a connection.

Then we send of the array with 5 elements every time (with indices) from S1

We read/recieve from S1 in S2, 5 elements every time.

S2 reads the indices of the string also and returns the highest ID received.

FIFO

There are 2 files attached.

P1.c and P2.c .

First we generate an array of random 50 strings in P1 and make a FIFO using mkfifo().

Then we write the array with 5 elements every time (with indices), to that FIFO.

We read from that FIFO in P2, 5 elements every time.

P2 reads the indices of the string also and returns the highest ID received.

Shared Memory

There are 2 files attached.

sh1.c, sh2.c

First we generate an array of random 50 strings in sh1, and then create and open a new, POSIX shared memory object. Then we use ftruncate function. After that we use mmap and give it the required arguments.

Then we write the array with 5 elements every time (with indices), using sprintf.

In sh2.c we use shm _open to open the req memory object and then we use mmap and give it req arguments.

Then we use printf to print the elements of array 5 elements each time. And return the highest element of array too.