

INTER PROCESS COMMUNICATION

In this project, I have generated 50 random strings using rand() function of length 4 each and sent 5 strings at a time. Then, I have sent last received index from second process to the first. Then, first process sends strings from the index next to that received index.

1. Using FIFO: FIFO(also known as named pipe) is a very commonly used IPC mechanism. I have first opened 1st process using write mode(second in read mode) , wrote 5 strings into it and then closed it. Then , I opened it in read mode (second in write mode) and read the received index from it. I have used read() and write() functions for it. I have repeated this same process 10 times in a loop. I have calculated the time which was needed to do this whole process using clock_gettime() function.
2. Using Shared Memory : I have used POSIX shared memory primitive to do this process. I have opened it using shm_open() and then wrote data into it. Then the other process read data from it and sent the last index from it. I was facing the problem of synchronisation in these two processes. So, to synchronize these processes , I have used named semaphores.
3. Using UNIX Domain Sockets: I have used Unix domain sockets in this part. First, I have bind the server socket to a specific port no. Then it accepts request from the client socket. Then, it listens to the request from the client side socket . It accepts strings and sends highest ID to the client. In this process also, I was facing the problem of synchronisation in these two processes. So, to synchronize these processes , I have used named semaphores.

I have also submitted one Makefile along with the question which compiles all the processes.

To run Q1:

Run parallely

`./p1_f` and `./p2_f`

Q2:

`./p1_s` and `./p2_s`

Q3:

`./p1_u` and `./p2_u`

Resources referred:

<https://www.ibm.com/docs/en/ztpf/1.1.0.15?topic=considerations-unix-domain-sockets>

<https://www.geeksforgeeks.org/named-pipe-fifo-example-c-program/>

<https://www.geeksforgeeks.org/posix-shared-memory-api/>