Mualla Argin
UIN : 728003004
CSCE 313 - 910

Programming Assignment 4

Design of Code

Within my code the user has the option of request a single datapoint, 1000 data points, transferring files (currently only tested binary and csv files) of varying size using any one of the interpocess communication mechanisms (FIFO,Shared Memory, and Message Queue). On server.cpp, the server gets the buffer capacity,the chosen IPC method, file size (possibly) and processes requests from the client.cpp.

General

- FIFO,Shared Memory, and Message Queue are all subclasses of the request channel and inherit functions and variables from the request channel.
- Each subclass has its own open ipc(mq,shm,fifo) function
- Each subclass has its own cread function
- Each subclass has its own cwrite function
- I added the -i flag to getopt in client.cpp in order to be able to get the number of channels
- I added the -c flag to getopt in client.cpp in order to be able to get the chosen IPC (Interprocess Communication Mechanism
- If user input buffer size to terminal, I change the value of buffer capacity within the client
- Code defaults to FIFO if no interprocess mechanism is entered.

Mualla Argin
UIN : 728003004
CSCE 313 - 910

- Error handling implemented for the memory management functions (i.e. munmap,etc)

FIFO

- FIFO operations are almost identical to my implementation in PA1.
- Additional if statement to account for the new ipc argument in the client and server
- Within if statement I have additional channel checks

Message Queue

- In the message queue request channel I create two message queue objects, two message queue strings, a constructor, a destructor , a send function , a receive function, and an open message queue function.

Shared Memory

- I utilized two semaphores for sychronization and to make sure the send and receive functions work as intended

Data Analysis

IPC Method	**Time to get 1k data points (s)**	**Time to transfer 10MB**
FIFO	31.118 seconds	0.372 seconds
Message Queue	30.263 seconds	0.429 seconds
Shared Memory	31.957 seconds	0.390 seconds

Mualla Argin
UIN : 728003004
CSCE 313 - 910

Time to get 1k data points (s)

Shared Memory > FIFO > Message Queue

Time to transfer 10MB

Message Queue > Shared Memory > FIFO

Overall, the three interpocess communication mechanisms implemented process 1000 data points in almost the same amount of time. The fastest method was Message Queue which took about 30.263 seconds.The slowest method was Shared Memory which took about 31.956 seconds. With larger sets of data that 2 second difference may be a 2 day difference hence with larger datasets message queue would be the most efficient approach. Additionally, FIFO was second to finish in terms of time it took to process the 1000 data points.

Overall, the three interpocess communication mechanisms implemented trasnfer 10 MB in almost the same amount of time. The fastest method was FIFO Queue which took about 0.372 seconds.The slowest method was message queue which took about 0.429 seconds. With larger sets of data that ~ 1 millisecond difference may be a 2 day difference hence with larger files FIFO would be the most efficient approach. Additionally, shared memory was second to finish in terms of time it took to transfer 10 MB. From a quick glance of the data, we see that the message queue is most efficient with data points and least efficient with file transfers.

Mualla Argin
UIN : 728003004
CSCE 313 - 910

Commands Run

- [3 pts] Gather the time to collect 1K data points across $c = 5$ channels for each instance of IPC method and present the results to compare their performance.

```
time ./client -i q -c 5 -p 9
```

I randomly selected patient 9

0m30.263s

```
time ./client -i f -c 5 -p 9
```

I randomly selected patient 9

0m31.118s

```
time ./client -i s -c 5 -p 9
```

I randomly selected patient 9

0m31.957s

Mualla Argin
UIN : 728003004
CSCE 313 - 910

[3 pts] Repeat the previous with file transfers for a 10MB file across $c = 50$ channels and again compare performance.

```
truncate -s 10M BIMDC/test.bin;time ./client -i q -f test.bin -m 4000 -c 50
```

```
0m0.429s
```

```
truncate -s 10M BIMDC/test.bin;time ./client -i f -f test.bin -m 4000 -c 50
```

```
0m0.372s
```

```
truncate -s 10M BIMDC/test.bin;time ./client -i s -f test.bin -m 4000 -c 50
```

```
0m0.390s
```