

CSCE 313 Programming Assignment 5

Hunter Cleary - hncleary - 625001547

March 2019

1 Inter-Process Communications

Communications in this assignment were sent using a FIFO / named pipes, message queues, and shared memory.

FIFO is an extension of the traditional pipe available within unix. A traditional pipe only lasts as long as the process does, while a name pipe will last as long as the system is up and must be deleted when no longer used. Once one of the pipe "files" is created any process can utilize it. Although, the pipe must be open at both ends in order for it to be used.

A message queue is a linked list of messages stored within the kernel and are interpreted with a unique identifier. Messages are added to the queue in order, but when receiving the messages they aren't required to be processed in order. Uses: (`ftok()`, `msgget()`, `msgsnd()`, `msgrcv()`, `msgctl()`)

Communications through shared memory is possible by running two processes that have a common shared memory location. Changes made by one process can be viewed by the other. With shared memory, the data does not have to go through the kernel, significantly speeding up the procedure. Uses: (`ftok()`, `shmget()`, `shmat()`, `shmdt()`, `shmctl()`)

2 Running

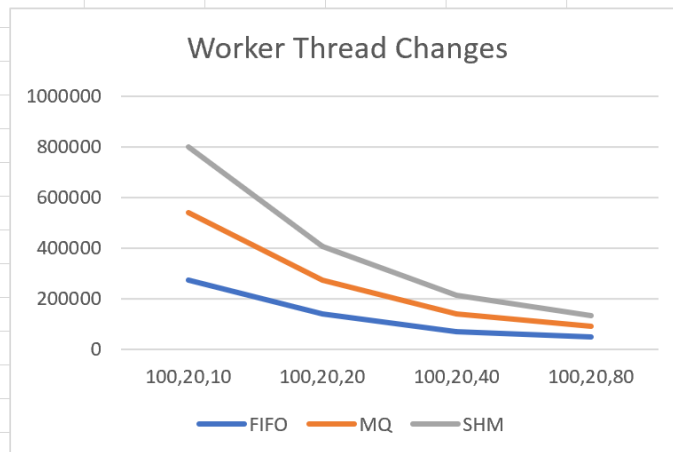
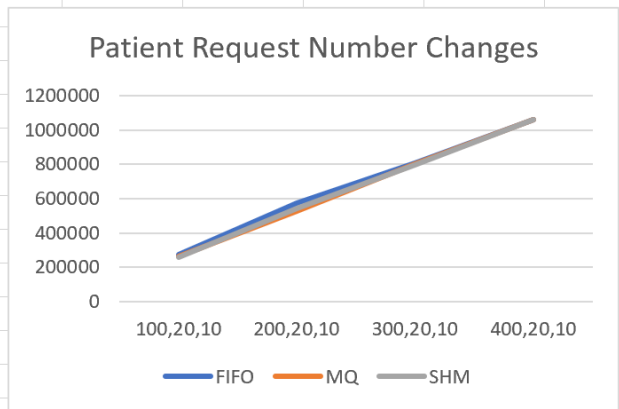
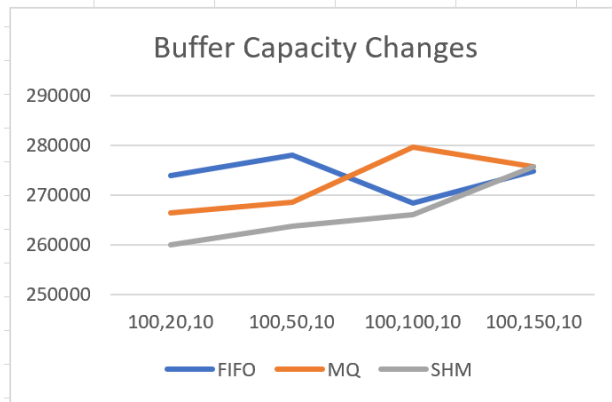
```
M = 256
populating for person 1
populating for person 2
populating for person 3
populating for person 4
populating for person 5
populating for person 6
populating for person 7
populating for person 8
populating for person 9
populating for person 10
populating for person 11
populating for person 12
populating for person 13
populating for person 14
populating for person 15
Pushing quit requests... done.
Took 0 seconds and 278101 micor seconds
All Done!!!
```

[-2.00,-1.60):	0	0	0	0	0	0	0	0	4	0
[-1.60,-1.20):	0	0	4	0	2	0	0	0	2	0
[-1.20,-0.80):	0	8	14	0	10	0	0	0	2	0
[-0.80,-0.40):	28	71	51	2	57	29	65	4	63	2
[-0.40,-0.00):	53	21	31	98	19	64	35	96	12	91
[-0.00,0.40):	8	0	0	0	4	7	0	0	9	7
[0.40,0.80):	11	0	0	0	3	0	0	0	2	0
[0.80,1.20):	0	0	0	0	5	0	0	0	2	0
[1.20,1.60):	0	0	0	0	0	0	0	0	3	0
[1.60,2.00):	0	0	0	0	0	0	0	0	1	0
[-2.00,2.00):	100	100	100	100	100	100	100	100	100	100

3 Performance

The performance of each IPC method was measured with varying numbers of patient requests, buffer capacity, and worker threads.

	n	100 default number of requests per patient		
	b	20 default capacity of the request buffer		
	w	10 default number of worker threads		
b dif	Time in Microseconds			
	100,20,10	100,50,10	100,100,10	100,150,10
FIFO	273964	278101	268464	274824
MQ	266466	268614	279706	275681
SHM	260010	263829	266152	275726
n dif				
	100,20,10	200,20,10	300,20,10	400,20,10
FIFO	273964	572207	805676	1059391
MQ	266466	525494	803698	1057859
SHM	260010	542890	793992	1061277
w dif				
	100,20,10	100,20,20	100,20,40	100,20,80
FIFO	273964	141123	70160	48808
MQ	266466	132760	68574	41430
SHM	260010	131901	75579	44937



Maximum number of worker channels for each IPC:

	Max Channels
FIFO	509
MQ	127
SHM	253

No comparison to PA4 is available due to a lack of data gathered from that assignment.

Destructors / Cleanup:

- FIFO - destructor deletes the pipe "file"
- MQ - destructor uses `mq_close()` and `mq_unlink()` to unlink the queue and delete
- SHM - destructor deletes the buffers for both sides that are accessing the shared memory