

NAME: Nikita Udodenko

DATE: 7/27/2021

YouTube Link: <https://youtu.be/zyREjbdoyk>

REPORT PA3

For this programming assignment we needed to implement the “Request Channel” abstract class, and three channels (FIFO Request Channel, Memory Queue Request Channel, Shared Memory Request Channel) which are the subclasses of the Request Channel. They all share some similar functions, but all are implemented in a different way.

The Message Queue Request Channel allows the exchange of messages between processes. To implement the program, we used function from the “mqueue.h”. The maximum capacity of the buffer is 8K. MQ provide use with a byte-stream where send and receive functions are synchronized by the Kernel. For that reason, the MQ implementation is rather straightforward. For the Shared Memory Request Channel, we had to implement an outside class called “SHMQ” that contained the constructor, the message send, message receive functions and a destructor. Messages are not synchronized in SHM, meaning that we need support synchronization using Kernel Semaphores, otherwise the `cread()` and `cwrite()` function in SHM would be meaningless.

To call the different channels the user has to type in “-i <f|q|m>”. f – stand for FIFO, q – stands for MQ, m – stands for SHM. If the user wants to create multiple communication channels, then he has to type “-c #”. All newly created channels will go into the vector called `mult_chan`, and after we are done getting data the program will iterate through and delete each channel. For a single data point request using multiple channels, the program will output the point for each channel created and time taken in each channel to receive data point. For 1000 data points request, the program will output the result in the `x1.csv` file, but in the terminal, it is going to show time taken by each channel to transfer 1000 data points to the `x1.csv`. For a file transfer request using multiple channels, each channel is going to transfer the $(\text{filesize} / \text{num_channels})$ bytes, if the filesize does not equally divide by the number of channels, then the last channel will have a greater capacity and the rest of the channels will have equal capacity.

1000 Data points transfer using 5 channels

FIFO :

6.335130 sec, 6.299998 sec, 6.284236 sec, 6.321365 sec, 6.265435 sec

MQ :

8.014414 sec, 9.123672 sec, 6.851062 sec, 7.318635 sec, 7.745480 sec

SHM :

8.557967 sec, 8.153792 sec, 7.602794 sec, 7.590538 sec, 7.963879 sec

From the above results we can see that the FIFO channel is the fastest, and the timing is about the same along 5 channels, however in the MQ and SHM the time difference in performance of the channels can be more than 1 sec.

File request using 5 channels

FIFO :

0.002656 sec, 0.004548 sec, 0.004159 sec, 0.004757 sec, 0.011316 sec

MQ :

0.010201 sec, 0.010380 sec, 0.008796 sec, 0.010688 sec, 0.009140 sec

SHM :

0.008366 sec, 0.007488 sec, 0.006873 sec, 0.009496 sec, 0.006885 sec

The last channel had a greater size transfer than the first 4. The SHM was faster than MQ and FIFO at the last transfer where the capacity of the channel was greater. However, at first 4 channels the FIFO channel was the fastest.