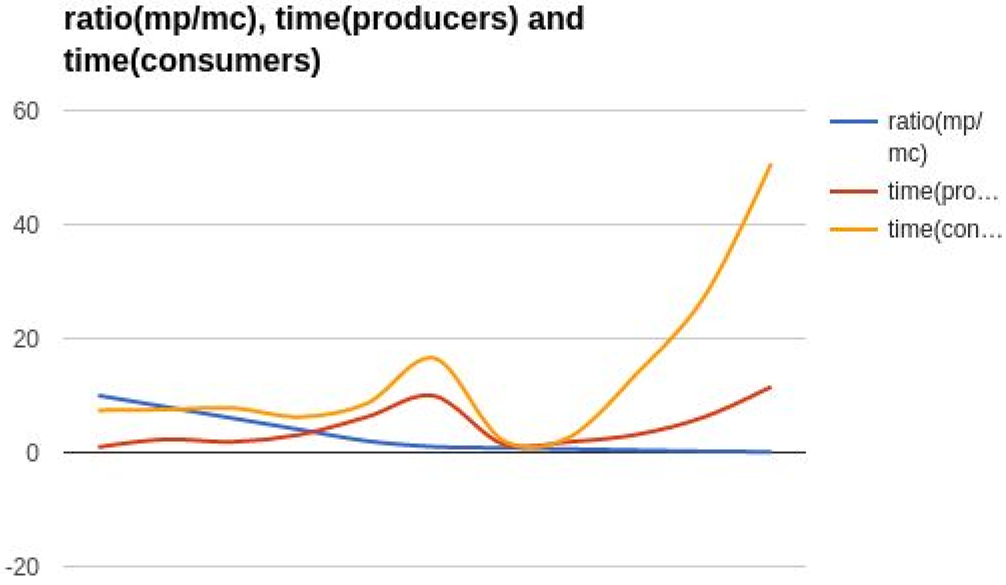
MINIX Project Report

By:- Sahil Yerawar
CS15BTECH11044

AIM:- To implement semaphore services in MINIX 3.2.1 and to implement producer_consumer problem using these semaphores in minix.

Performance analysis of producer and consumer threads



Here x axis denotes the the ratio mp/mc where mp is the sleeping parameter for the producer process and mc is the sleeping parameter for the consumer process.Also, Y-axis depicts the time taken by the processes(in seconds).
Also note that these observations have been taken for np = 10, nc = 15, cnp = 15, cnc = 10, and max number of processes = 100.

Observations:-

We observe that ratio mp/mc > 1, the time taken by the producer processes is lesser than that of consumer processes. This might be due to the fact that while executing, a block of producer processes accessing memory was seen. This might be due to the fact that the method by which the next process is chosen from the pool of sleeping processes is done by using the FIFO algorithm. This makes it easier for the immediate producer processes to access the semaphore as they are logically the next ones to enter the queue. We also see that the time taken by the producer and consumer is almost not changing for mp/mc > 1.

At the value of mp/mc = 1, we see a spike in the time taken by the producer process and an even greater spike in the consumer processes.
This anomaly can be explained as we know that mp = mc, the average time taken by the producer and the consumer processes is same. Hence random order of execution can occur.

After that point(For the values of mp/mc < 1), the time taken by the producers and consumers increases gradually. But the rate of increase for consumers is greater than that of producers. This can be due to the fact that, producer processes execute as a block and as such take less time to complete. While for the consumer processes, some of the processes have to wait for a while, which increases the average waiting time for the consumer processes.