

SYSC 4001 – Assignment #2

Yuzhou Liu, 100853392

Part B. Verification of the PC problem without semaphore S

Observation:

The program runs successfully when semaphore S was removed. The program was run with only **one producer** and **one consumer**.

Explanation:

Semaphore S acts as a mutual exclusion semaphore that prevents the producer and consumer from accessing the same segment of memory. In the program, the producer continuously writes to separate buffer blocks, and the consumer continuously reads from the buffer blocks. The only instance when the producer and consumer may access the same buffer block is when there is nothing stored in the buffer. At this point, the producer must write to a buffer block while the consumer must read from the same buffer block.

In the case of a single producer and consumer, this scenario is handled by semaphore N so semaphore S is not needed. Semaphore N acts as a synchronization mechanism that guarantees the producer always goes before the consumer. Thus in this scenario, semaphore N guarantees the producer gets access to the buffer block before the consumer, so both are never in the critical section at the same time.

Semaphore N is the reason why semaphore S is not needed.

Bonus

The performance of the program was evaluated by varying parameters to achieve different configurations.

The **reference** configuration used was:

File Size	Number of Buffers	Buffer Size	Semaphore S
1 Mbyte	100	BUFSIZ (8192)	With Semaphore S

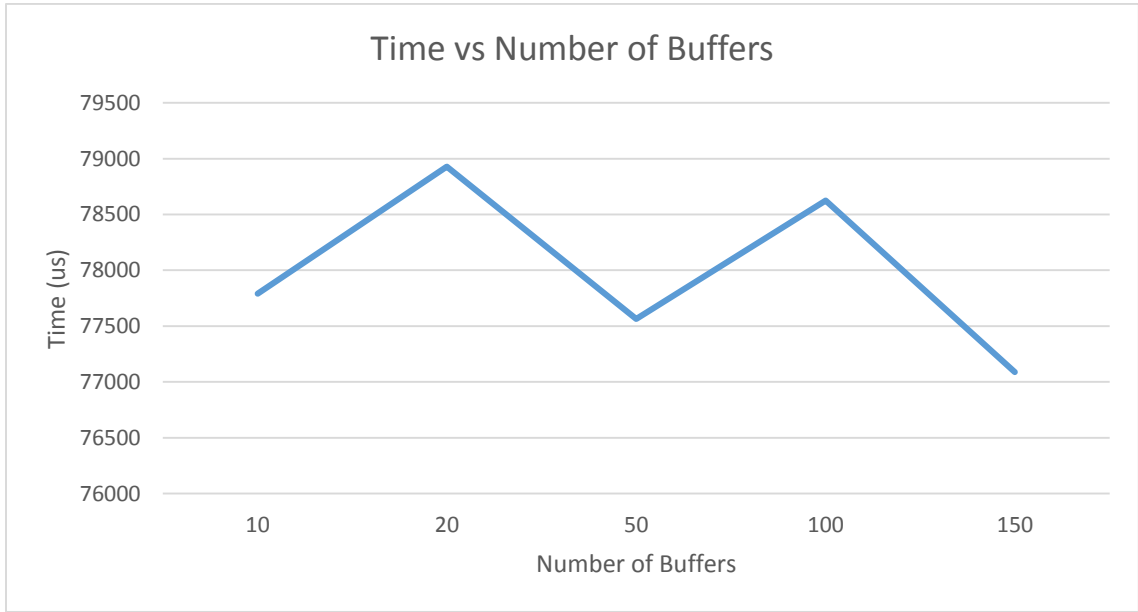
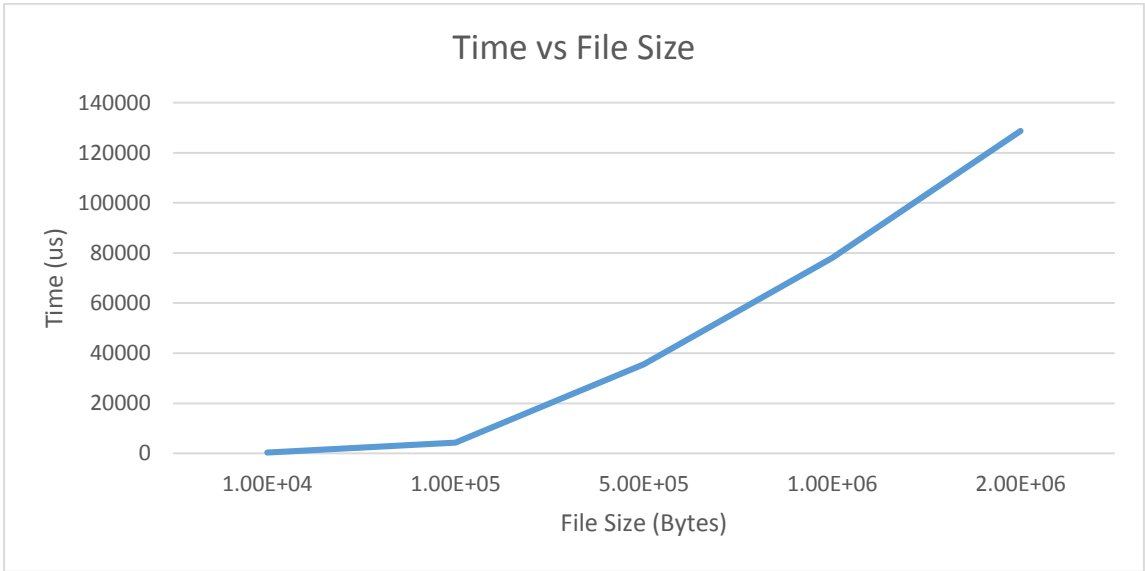
Each configuration was tested by varying a single parameter from the reference configuration. The results are shown below:

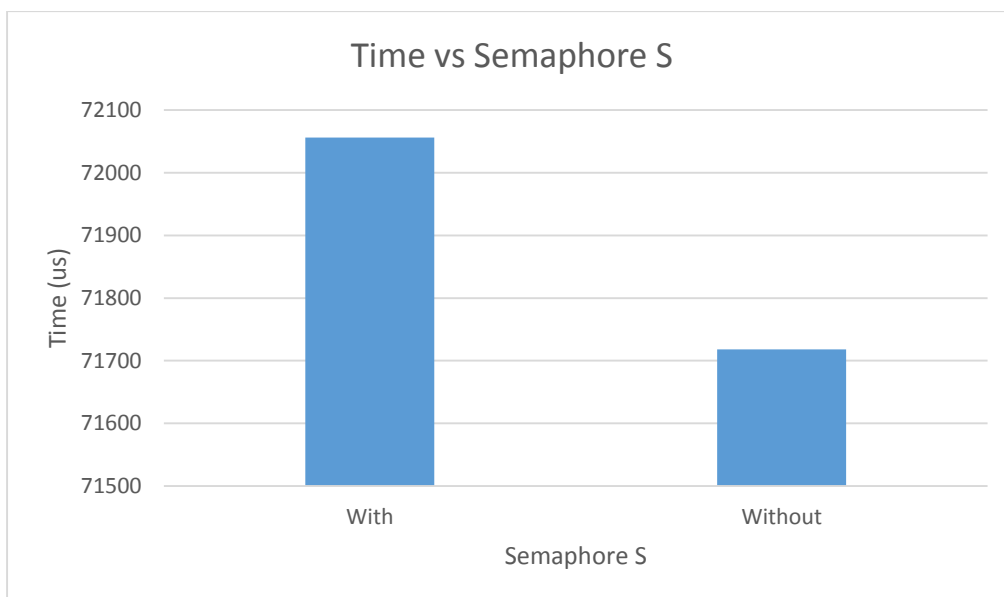
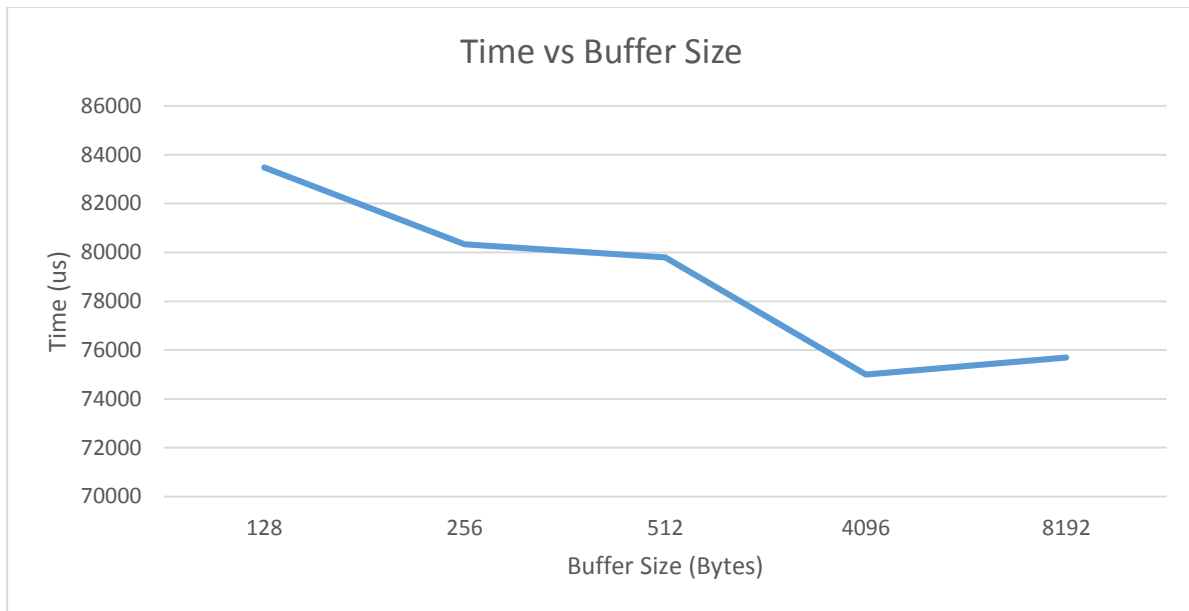
File Size			Number of Buffers			Buffer Size	
File Size	Time (μ s)		# Buffers	Time (μ s)		Buffer size	Time (μ s)
10 kbyte	272		10	77789		128	83474
100 kbyte	4300		20	78929		256	80338
500 kbyte	35571		50	77564		512	79801
1 Mbyte	77994		100	78624		4096	75000
2 Mbyte	128761		150	77087		8192 (BUFSIZ)	75695

Semaphore S	
With/Without	Time (μs)
With	72056
Without	71718

No real difference between with/without semaphore S.

Charts





So the results show that there is almost a linear relationship between completion time and file size. As file size gets bigger, completion time becomes longer.

Completion time vs. number of buffers has no real distinct difference. Completion time does not really have any correlation with number of buffers.

Completion time vs. buffer size shows a small correlation. As buffer size grows, completion time decreases but very slightly (1-2ms every 2x increase in buffer size).

As for the completion time vs. Semaphore S, there is a small difference. Without semaphore S, the program runs 0.3ms faster, but that may not be attributed to the semaphore.