

CMPE 382 Operating Systems, Spring 2023-2024

Homework 2- Synchronization: Semaphores and Critical Section Implementation Report

TXT FILES	THREADS	REAL	USER	SYS
29	1	0m0, 039s	0m0, 035s	0m0, 004s
29	4	0m0, 019s	0m0, 029s	0m0, 015s
29	10	0m0, 017s	0m0, 031s	0m0, 012s
29	32	0m0, 040s	0m0, 035s	0m0, 009s
40	1	0m0, 045s	0m0, 044s	0m0, 000s
40	4	0m0, 016s	0m0, 051s	0m0, 000s
40	10	0m0, 016s	0m0, 037s	0m0, 011s
40	32	0m0, 018s	0m0, 053s	0m0, 005s

First, the ‘real’ time represents the actual elapsed time, while ‘user’ and ‘sys’ represent the CPU time used by the system. It can be observed that as the number of threads increases, the ‘real’ time initially decreases, which means that the system is able to do more work simultaneously, which results in faster completion of the task. However, after a point, the ‘real’ time starts to increase again. This is due to the **overhead** of threads, which occurs when the number of threads is unnecessarily large.

Second, the ‘user’ time also increases with the number of threads. This is because as more threads are used, more CPU time is spent on the actual computation. However, like ‘real’ time, ‘user’ time also starts to lose efficiency or even decrease after a certain point. Once reached to that point, adding more threads doesn’t increase performance, and may even degrade, due to the overhead of context switching and overhead.

Lastly, the ‘sys’ time, which represents the time CPU spends on system tasks on behalf of the process, such as memory allocation, tends to decrease as the number of threads increases. This is because these system tasks are usually fixed, and as the number of threads increases, these tasks are shared across more threads, consequently reducing the ‘sys’ time per thread.

In summary, increasing the number of threads can improve performance up to a point, but beyond that, the overhead of managing more threads can degrade performance. It’s important to find the right balance for optimal performance.