

CMPE 382 Operating Systems, Spring 2023-2024

Homework 2- Synchronization: Semaphores and critical section implementation

Due Date: May 12, 2024 23:59 (**a delay of up to 3 days is accepted with a penalty of 10% each day**)

You are expected to write a C program using threads and semaphores to implement critical sections.

First, copy the following shell script into a new.sh file. When executed, it will create a new directory and fill it with 20 - 30 files, each containing 1000 random integers.

```
#!/bin/bash
rm -rf myDir
mkdir myDir
cd myDir
for i in $(seq 1 $((21+RANDOM%10))); do
  for j in $(seq 1 $((RANDOM%1000))); do
    echo $RANDOM >> file$i.txt
  done
done
cd ..
```

You must design your program to accept two parameters from command line, the directory containing generated txt files (myDir) and the number simultaneously active threads.

Your program must first open the directory passed to it as a command line argument (for example, “./hw2 directoryName threadNumber”). The program is to use threads to compute the number of prime numbers in each file and print thread by thread. The number of active threads at a time is to be limited to threadNumber.

Hint1: use a semaphore (mutex) to limit the number of simultaneously active threads (threadNumber). Create a threads whenever an active thread terminates. Your program must work for any number of files.

Hint2: you may check number of active threads per process by executing “ps -eLf” command in a second terminal window under NLWP column, after your program starts execution.

Notes about the problem: you may think of this problem as if an unknown number of students are queued in front of an instructor’s office to see their exam papers. But the instructor’s office can only host at most n (threadNumber) students. So, the new student can enter the room, only when one of the students inside leaves the room.

Execute your program observing time data:

```
time ./hw2 myDir 1
time ./hw2 myDir 2
...
```

Note that “time” is a system utility, which displays total execution time details of the executing program given in the command line.

Let the output of one of the runs be similar to the following.

```
Thread 1 has found 6454 primes in file1.txt
...
Thread 5 has found 23935 primes in file2.txt
...
real 0m54.277s
user 1m48.500s
sys 0m0.044s
```

This sample output shows that this execution took 54.277 seconds.

Repeat the execution for at least 8 times with different number of simultaneously active threads. Prepare a table showing how the times change with the number of simultaneous threads, similar to the one as follows:

THREADS	REAL	USER	SYS
1			
2			
3			
4			
...			

As the number of simultaneous threads increase, what changes do you observe for real, user, and sys values? Explain. You may increase number of files and file sizes by making changes in the shell script to observe higher variations.

Submission requirements:

- Well documented source file (yourID_hw2.c)
- Output of each run (yourID_hw2Output.pdf)
- Table and report (yourID_hw2-Report.pdf)

After the submission:

- A quiz will follow the submission day
- The objective of the quiz is to assess your understanding of the assignment and its implementation.