As with all the assignment, you will submit one zip or tar file containing your answers.

# pthreads API

Write a “C” program that will:

1. Accept two parameters from the command line:
   1. The number of threads to start
   2. Length of the string to generate
2. Program will start up the number of threads specified. A thread number (integer) should be passed to each thread to identify its output.
3. Each thread will add characters (of your choice) to a string until it reaches the length specified. Make sure this operation is atomic! When a thread discovers that the string is already of the required length, it will terminate. Thread should print out each character (and the thread number) added.
4. After all threads have terminated, print the final string.

**Submitted with readme.md**

1. What is the maximum number of processes that may present in critical section at any point of time in this binary semaphore problem?

|  |  |
| --- | --- |
| Each process Pi (i=1 to 6) execute the following code | Process P7 execute the following code |
| While(true){  P(mutex);  /\*Critical Section\*/  V(mutex);  /\*Remainder Section\*/  } | While(true){  V(mutex);  /\*Critical Section\*/  V(mutex);  /\*Remainder Section\*/  } |

**Maximum of 7 (all) processes**; *say P1 enters the Critical Section, then P7 enters its Critical Section (but not before signaling the mutex). Now P1 and P7 are both in the Critical Section, and the mutex is left open for another process. P7 may continue to run, continuously reopening the mutex from any lock initiated by P1-P6.*

1. [In a certain application, the initial value of a counting semaphore s is 10. The following operations were completed on the semaphore in the given order 5P, 20P, 2V, 18P, 5V, 10P, 1V. The new value of counting semaphore is \_\_\_?](https://gateoverflow.in/409515/nptel-synchronisation-semaphore)

**-35**