

Indian Institute of Technology Jodhpur
Operating Systems Lab (CS330)
Assignment 4

Dated 31st March, 2021

Total marks: 40

Implement a virtual CPU scheduler on top of the Linux kernel using POSIX threads. The details of the specification are as follows.

A) Create N concurrent threads that will be scheduled by the virtual scheduler. Each thread will be either a producer (P) or a consumer (C), with equal probability of occurrence. Each of the P threads will be generating 1000 pseudo-random integers, and storing them in a shared BUFFER of maximum capacity M; if the buffer is full, the thread will wait. Each of the C threads will be repeatedly removing an element from the BUFFER; if the buffer is empty, the thread will wait. The P and C threads are referred to as WORKER threads.

B) Each of the WORKER threads will have signal handlers installed for handling the user-defined signals SIGUSR1 and SIGUSR2. SIGUSR1 will be used to put the thread to sleep, while SIGUSR2 will be used to wake up the thread to resume execution.

C) Another thread, called the SCHEDULER thread, will be created that will be sending sleep/wakeup signals to the WORKER threads. It will implement a round-robin scheduling algorithm with a specified time quantum (say, 1 second). It will run one of the N WORKER threads at a time, while the other N-1 threads will be put to sleep. During context switch, the currently running thread will be put to sleep, while the next thread in the READY queue will be activated.

Print relevant messages on the screen whenever a context switch or thread termination takes place. Also display the number of elements in BUFFER as context switching takes place.

References:

<http://www.csc.villanova.edu/~mdamian/threads/posixthreads.html>

<http://www.cs.kent.edu/~ruttan/sysprog/lectures/multi-thread/multi-thread.html>

<https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>

Evaluation Guidelines:

- (a) Creation of concurrent threads 5
- (b) Storing data in a shared buffer 5
- (c) Correctly implementation of signal handlers 6
- (d) Correctly synchronised accesses to buffers 6
- (e) Implementation of scheduling policy 4
- (f) Printing of events correctly for each context switch 4
- (g) Overall correctness 10

Deadline : 10th April, 2021