Operating Systems Laboratory
Spring Semester 2017-18

Assignment 3b

Implement a Virtual CPU Scheduler on Top of Linux using Threads

Assignment given on: February 05, 2018
Assignment deadline: February 12, 2018

Implement a virtual CPU scheduler on top of the Linux kernel using multi-threading, and run a synthetic job mix on the scheduler. The details of the specification are as follows.

a) Read an integer *N*, and create *N* concurrent threads T_1, T_2, …, T_N that will be scheduled by the virtual scheduler. Each thread will be generating 1000 random integers, sort them, and wait for a random time interval between 1 and 10 seconds before terminating. These N threads are referred to as **WORKER** threads.

b) Each of the threads will be having 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. The running status of all the threads will be stored in a shared data structure called **STATUS**.

c) Another thread will be created, called the **SCHEDULER** thread, which will be sending appropriate signals to the *N* worker threads. It will implement a round-robin scheduling algorithm with a time quantum of 1 second. It will cyclically run one of the *N* 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 woken up.

d) Another thread will be created, called **REPORTER** thread, which will continuously monitor the **STATUS** data structure, and print relevant messages on the screen whenever a context switch or thread termination takes place.

Hint: *Use the POSIX Pthread library for creating/managing the threads.*

Submission Guideline:

- Create the program as a single file as `Ass3b_<groupno>.c` or `.cpp`, and upload it.