Operating Systems Lab (CS 470):

Lab 6: Given a multiprocessor system with 3 processors simulate such a computer by executing the processes assigned to each processor using different scheduling algorithms.

Overview

PCB is a data structure holding information about processes. In this specific scenario we need to have a process id, a process state (create, waiting, executing, ready, terminating), a priority and a CPU burst time for each process. All other type of information is optional.

Instructions

Each processor will have originally a certain number of processes. These numbers should be provided through command line arguments. For each process, the status, the CPU burst and the process id should be generated randomly. The original CPU burst time cannot be bigger than 5 seconds and should be bigger than 0. The process id should be unique. For the process state, originally each one should be set to "new". For the priority the initial value should be in the range [0,127], where a priority of 127 is the lowest priority. If the processes from one processor are finished (all of them were executed), that processor should take over some processes from the other processors to mimic load balancing.

Notes

- Processor 1 will implement a Round Robin scheduling mechanism (time quantum 500).
- Processor 2 will implement a FCFC scheduling mechanism (time quantum 1).
- Processor 3 will implement a priority based scheduling (time quantum 1).
- Implement an aging mechanism which is invoked each 3 seconds.
- Implement a status mimicking mechanism which changes randomly the status of each process according to the process state diagram which is invoked every 3 secs.
- The 3 processors, the aging, and the status mimicking process should run in a separate thread in parallel, synchronizing between the processes if necessary.
- The load balancing (taking over processes from other processors) takes place only when one processor is completely finished executing its processes.
- A process cannot be "executed" if not in "running" state. Once the process passed the "new" state it cannot return into that state. Same with the terminate state, which will be set once the process is completely executed.
- An "execution" decreases the CPU burst of a process with 500 milliseconds. Use wait() to mimic the "execution time".
- Each step during the simulation should be visible on the screen.