ASSIGNMENT - 5

SNEHAL KUMAR

2019101003

REPORT

Benchmark.c is used to test the scheduling algorithms. It contains 10 processes with varying I/O, CPU times and priorities.

RESULTS

(The ranking is based on endtime-creationtime-runtime:)

1. Round Robin - 1133
2. PBS - 1148
3. MLFQ - 1302
4. FCFS - 1835

PERFORMANCE ANALYSIS

RR: Has best performance as all processes were given equal priority and yielded after tick. Thus, the I/O intensive processes needn't wait for CPU intensive process to finish (Convoy effect).

PBS: The priorities for the process (lower for I/O) made it suitable for scheduling the processes efficiently. With different priorities, it may have different performance.

MLFQ: Due to changing priorities, CPU intensive process get demoted, while I/O process finish earlier. The overhead of changing queues adds to slower performance.

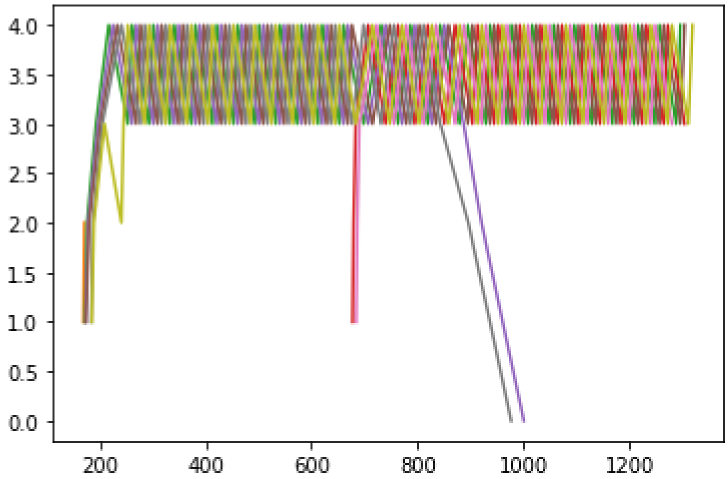
If a process voluntarily relinquishes control of the CPU and returns, it is added to the same queue. This implies that the I/O bound processes always stay in higher priority queues while the CPU intensive processes that are in lower priority queues for too long get promoted (aging) to prevent starvation.

If aging is not implemented, a user process can be given higher priority and thus stay in higher priority queue while the CPU intensive processes will not get a chance to execute. In this way, a process can exploit the scheduling algorithm.

FCFS: It has the worst performance since every process has to wait for previous process to finish. If CPU intensive process comes first, it will even delay I/O processes for long time and reduces efficiency drastically.

BONUS

The timeline of processes (queues vs ticks) in MLFQ: (Here process gets demoted if it exceeds timeslice limit of queue and due to aging, some processes get promoted)


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
200
400
600
800
1000
1200