# Operating Systems (CSL 3030)

Readme file

| Lab Assignment | 04 |
|---|---|
| Roll No. | B20CS018 |
| Name | Harshita Gupta |

## Input of Process Parameters:

### CPU Bound processes-

| Priority | Rounds | Sleep Time | Sleep probability |
|---|---|---|---|
| 5 | 1000 | 1 | 0.3 |
| 5 | 1000 | 1 | 0.3 |

### I/O Bound processes-

| Priority | Rounds | Sleep Time | Sleep probability |
|---|---|---|---|
| 10 | 4000 | 3 | 0.7 |
| 10 | 4000 | 3 | 0.7 |

**All files:** process.c, queue.c, scheduler.c, queue.h, generator.c, prio_queue.h, prio_queue.c, elem.h

**To run the program -**

```
gcc process.c -o process
gcc generator.c -o gen
gcc queue.c -c
gcc scheduler.c -c
gcc prio_queue.c -c
gcc queue.o scheduler.o prio_queue.o -o scheduler
```

**'testout'** file shows how the status of respective processes is displayed in the terminal.

The **results folder** has Response time, Total waiting time, and Turn-around time for each process with their process id and contains the average of these metrics for all 4 processes, for both the algorithms with time quantum = 1000 iterations and 2000 iterations.

## Round-Robin Scheduling Algorithm

1. The scheduler maintains a queue of ready processes and a list of blocked and swapped-out processes.
2. The Process Control Block of the newly created process is added to the end of the ready queue. The Process Control Block of the terminating process is removed from the scheduling data structures.
3. The scheduler always selects the Process Control Block from the head of the ready queue. This is a disadvantage since all processes are given the same priority. Round robin also favors the process with short CPU bursts and penalizes long ones.
4. When a running process finishes its time slice, it is moved to the end of the ready queue. A time slice is the amount of time that each process spends on the processor per iteration of the Round Robin algorithm. All processes are executed in a first come first serve manner but are preempted after a time slice. The process will either finish in the time slice given or the process will be returned to the tail of the ready queue and returned to the processor at a later time.

**Disadvantages of Round Robin Algorithm**

- Larger waiting time and Response time
- Context Switches
- Low throughput

## Priority-Based Scheduling Algorithm

1. Allocate CPU to every process in a round-robin fashion, according to the given priority, for a given time quantum (say k units) only for one time.
2. Processors are arranged in increasing order or their remaining CPU burst time in the ready queue. New priorities are assigned according to the remaining CPU bursts of processes; the process with the shortest remaining CPU burst is assigned with the highest priority
3. The processes are executed according to the new priorities based on the remaining CPU bursts, and each process gets control of the CPU until they finished their execution.

## Comparative analysis of the stated performance metrics

The performance of two algorithms can be compared by considering **the number of context switches, average waiting time, and average turnaround time**.

It is observed that the proposed algorithm performs better over simple round robin for varying time quantum. We see that priority based round robin has less number of context switches, less turnaround, and less average waiting time in comparison to simple round robin for the same value of time quantum.

## Conclusion

We have successfully compared both the algorithm i.e. simple round robin and the priority based round robin that the latter one is more efficient because it has less average waiting time, average turnaround time, and the number of context switches as compared to simple round robin, in turn reducing the operating system overhead and hence dispatch latency. Also, it reduces the problem of starvation as the processes with less remaining CPU burst time are assigned with the higher priorities and are executed first in the second round of the algorithm. The performance of time-sharing systems can be improved with the proposed algorithm and can also be modified to enhance the performance of the real-time systems.