ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 312E
COMPUTER OPERATING SYSTEMS

Homework1 Answers

Ali Emre Kaya
150210097
kayaemr21@itu.edu.tr

SPRING 2025

Contents

1 Questions

1.1 Scheduling Fairness Analysis: How does your scheduler ensure fairness among processes? What scheduling strategies could be used to improve fairness?

Round Robin is a fair scheduling approach. Most prior task run firstly which given algorithm want it. Also to obstruct starvation, same job cannot run consecutively.

My code ensure that when a job finish its time slice, other higher priority one comes and run with the implemented algorithm which is just a kind of round robin scheduling

To improve fairness, Multi-Level Feedback Queue (**MLFQ**) can be implemented, this scheduling mechanism arrange jobs priority numbers and with that ability, it optimize turnaround time and minimize response time.

1.2 Edge Cases and Failure Scenarios: Edge Cases and Failure Scenarios: What are some possible failure cases in your scheduler (e.g., a process not responding, starvation, dead- lock)? How does your implementation handle these cases?

I ran my code for many scenarios but I took good response every of them, but my code has some limitation;

```
#define MAX_ITEMS 10000
#define MAX_ITEM_LENGTH 30
#define MAX_JOBS 100
```

This macros define the limitations, an example my sistem cannot take more job more than 100 or cannot take a arrival time like 10^{30}. But this macros can able to change whatever according to users requests.

When two long executive time and high priority level jobs come to the system, other jobs will be stay in starvation situation and for a long time those little jobs cannot be finished. My implementation did not handle this cases, to handle this kind cases, **MLFQ** needed, because that scheduling technique decrease the priority level in processing time.

Dead-lock situation may be occur when only one job occur because according to the algorithm, same job cannot process consecutively. My code handle this, when there is only one job in queue, it don't listen algorithm basically, after finishing a job,continue that job again.

REFERENCES