FIT2100 Tutorial #5
Uniprocessor Scheduling
Week 8 Semester 2 2019

September 11, 2019

Revision Status:

`$Id: FIT2100-Tutorial-05.tex, Version 2.0 2018/09/26 15:15 Jojo $`

`Updated by Daniel Kos, Sep 2019.`

Acknowledgement

The majority of the content presented in this tutorial was adapted from William Stallings (2017). *Operating Systems: Internals and Design Principles*, Pearson.

Contents

1 Background

This tutorial provides students with the opportunity to explore further on the concepts of uniprocessor scheduling discussed in the lecture.

You should complete the suggested reading in Section 2 before attending the tutorial. You should also prepare the solutions for the two sets of practice tasks given in Section 3.1 and Section 3.2 respectively.

2 Pre-tutorial Reading

You should complete the following two sets of reading:

- Week 6 Lecture Notes on "Uniprocessor Scheduling"
- Stallings' textbook: Chapter 9

3 Practice Tasks

3.1 Review Questions (7 marks)

Question 1

What are the three types of processor scheduling?

Question 2 (3 marks)

What is the difference between **turnaround time** and **response time**?

Question 3

What is the difference between **preemptive** and **non-preemptive** scheduling?

Question 4 (4 marks)

Is a non-preemptive scheduling approach a good choice for interactive systems? Why?

Question 5

What is the meaning of the term: **feedback scheduling**?

3.2 Problem-Solving Tasks (3 marks)

3.2.1 Task 1 (3 marks)

On a system with `n` CPUs, what is the maximum number of processes that can be in the READY, RUN, and BLOCKED states?

3.2.2 Task 2

Consider the following table, which shows when each of the processes arrives to the system and the CPU time (processing time) required for its execution. Assume that no I/O operations are involved in these processes.

Process	Arrival Time (in seconds)	Processing Time (in seconds)
A	0	3
B	1	6
C	4	4
D	6	2

Draw a chart (or sequence) of process execution under the following process scheduling:

(a) First-Come-First-Served (First-In-First-Out)

(b) Shortest Process Next

(c) Round Robin with the CPU time slice quantum of 2 seconds

3.2.3 Task 3

For the processes listed in the table (in Task 2), what is the **average turnaround time** for those three scheduling methods?

3.2.4 Task 4

For the processes listed in the table (in Task 2), what is the **average throughput time** for those three scheduling methods?

3.2.5 Task 5

Considering a variant of the Round Robin scheduling algorithm where the entries in the READY queue are *pointers* to the Process Control Blocks (PCBs).

(a) What would be the effect of putting **two pointers** to the same process in the READY queue?

(b) What would be the major advantage of this scheme?

(c) How could you modify the basic Round Robin algorithm to achieve the same effect without having the duplicate pointers?