

48450 Real Time Operating Systems

Assignment 3 (25 marks)

Deadline for submission: 23:59 PM, 22 May 2018

1. Introduction

This assignment is a topic about CPU scheduling and Memory management, which are key elements in Real Time Operating System. A submission will be marked on its merits and may be awarded a mark that is less than 25 score if it's of modest quality. You need to program as the requirement, complete it including a reflective self-assessment in the conclusion and submit it by the due date.

All programs are implemented in C language.

This assignment is marked out of 25 and comprises 25% of the total score for this course.

2. Assignment details

CPU Scheduling, FIFOs, Memory management and Signals

The assignment will involve some application program developments using CPU scheduling, FIFOs and Signal concepts. You are required to develop a program, namely **Prg_1** and **Prg_2**. The requirements of these two programs are

- (1) **Prg_1 (11 points):** You are required to use **CPU scheduling** and **FIFOs** in the **Prg_1**. It should include two threads, **Threads 1 and 2**.

Thread 1: In this thread, the **Prg_1** is to simulate CPU scheduling by applying **Shortest-remaining-time-first algorithm (SRTF)**. Your program is required to measure the *average waiting time and turn-around time* (See page 6 of Lecture 6) in the **CPU scheduling**. After the CPU scheduling is completed, your program is required to define a FIFOs and write these *average waiting time and turn-around time* to CPU memory through the FIFOs. The input data involving the CPU scheduling are as follows:

Process ID	Arrive time	Bust time
1	8	10
2	10	3
3	14	7
4	9	5
5	16	4
6	21	6
7	26	2

Thread 2: In this thread, your program is required to read the *average waiting time and turn-around time* from the memory through **the FIFOs** as defined in the **Thread 1**. Then, your program is required to write those read data to a text file named "output.txt".

- (2) **Prg_2** (11 points): You need to use **Deadlock detection and Signals**. Your program is required to detect the CPU deadlock and report all the possible deadlocks caused by different processes. Your program is required to read the process information from “Topic2_Prg_2.txt” and run the deadlock detection algorithm by referring to the slide 22 on lecture 5. The data file of “Topic2_Prg_2.txt” are:

<u>Process ID</u>	<u>Allocation</u>	<u>Request</u>	<u>Available</u>
	<i>A B C</i>	<i>A B C</i>	<i>A B C</i>
<i>P0</i>	0 1 0	0 1 2	0 1 2
<i>P1</i>	2 0 0	2 0 2	
<i>P2</i>	3 0 3	0 0 2	
<i>P3</i>	2 1 1	3 2 2	
<i>P4</i>	0 0 2	0 3 5	
<i>P5</i>	2 1 3	0 1 1	
<i>P6</i>	5 2 4	1 6 4	
<i>P7</i>	1 3 1	5 0 3	
<i>P8</i>	2 4 2	1 2 4	

The results are all those process IDs that will caused the CPU deadlock. You need to write the results to “output_topic2.txt”. In addition, you need to set up your own signal (SIGUSR1 or SIGUSR2) to watch the “output_topic2.txt” and give a notification when the writing “output_topic2.txt” is completed.

Furthermore, it is an option that if your program is capable of generating Gantt-chart style graphical outputs on the console.

- (3) **Report** (3 points): You are required to write a report to summarise your observation.

3. Assignment Deadline and Submission

The deadline to submit this assignment is 23:59 PM, 22 May 2018

You are required to submit two formats of the assignment via email:

1. Your full assignment report.
2. Your ‘C’ code file

If you use makefile for compiling your program, you are required to send it to the lecture coordinator as well