2021-CS211: Programming and Operating Systems

Homework Assignment

This is an "Open Book" assignment. You can use any lecture notes, or books, or resources online. But you are required to acknowledge them in your solutions. You are ***not*** *permitted to consult with another person.*

You are expected to adhere to the highest standards of integrity and honesty when completing this assessment.
If there is any concern in relation to the integrity of the assessment, you may be required to participate in an interview to verify the originality of your work. NUI Galway's policies on ***Academic Integrity*** *will be enforced. For more details, see* `http://www.nuigalway.ie/centre-excellence-learning-teaching/teachinglearning/academicintegrity`

Submit your work as a typed (preferably) or hand-written document, converted to PDF. It should be uploaded to the "Assignment" section of the CS211 Blackboard module. If you have any problems with that, you must make alternative arrangements with Niall Madden at least 24 hours in advance of the deadline.

Deadline: 5pm, Friday 30 April, 2021.

.....

Q1. (a) [**2 Marks**] What is a *process*? Name the five possible states of process activity.

(b) [**8 Marks**] Explain the following CPU scheduling algorithms, mentioning an advantage and a disadvantage of each:

(i) *First-Come-First-Served* (FCFS),

(ii) *Shortest-Job-First* (SJF),

(iii) *Round-Robin* (RR).

The table below shows the CPU burst times (in seconds) of five processes submitted in the given order, all at time $t = 0$. Calculate the average wait and response times for each of FCFS, SJF, and RR with a time quantum of $q = 4$ seconds.

Process	P_1	P_2	P_3	P_4	P_5
Burst Time	20	8	4	4	4

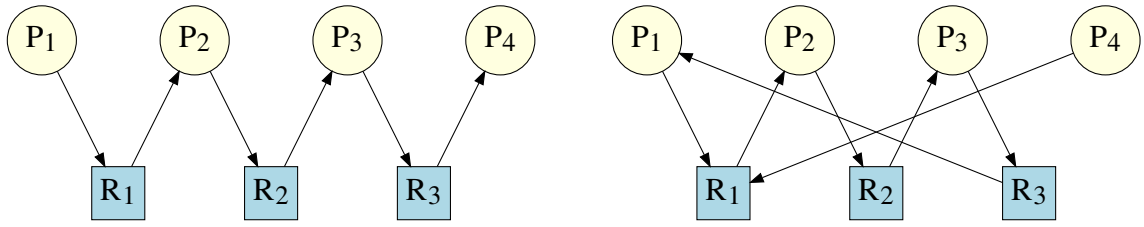
. . . . . .

Q2. (a) [**6 Marks**] Consider the two *Resource Allocation Graphs* in Figure 1. In both scenarios presented there are four processes, P_1, P_2, P_3 and P_4, and three resources, R_1, R_2, and R_3, of which there is one instance each.

i. Determine if it represents a deadlocked state.

ii. If it is deadlocked, explain why.

iii. If it is not deadlocked, explain how the resource allocation would proceed so that, eventually, all processes received their desired resource allocation. (You may assume that if a process has its desired resource allocation it will eventually terminate and release that allocation).



(a) Example 1 (b) Example 2

Figure 1: Two resource allocation graphs for Q2-(a)

(b) [**4 Marks**] Suppose a system has three running processes, each of which requires three resources of a particular type. Furthermore the system has five instances of that resource. Explain, with the aid or a resources allocation graph, how the system could reach deadlock.

What is the smallest number of extra resources that could ne added the ensure deadlock is not possible?

. . . . . .

List all sources (e.g., websites, text books, notes) you used in answering these questions.