LUT University

LUT Computer Vision and Pattern Recognition Laboratory 2022-03-09

BM40A0202 Foundations of Computer Science

Olli-Pekka Hämäläinen

Exercise 8 (week 11): Scheduling and Operating Systems.

Tasks (1 p/task)

1. Task in Moodle.

2. Table 1 shows the arrival time of five processes to a scheduler and an estimate of the execution time required for each of them.

Table 1: Processes.

Process	Arrival time	Estimated duration
1	0	3
2	2	7
3	5	2
4	6	5
5	9	3

Show the operation of the scheduler (timing of the processes) when using the following time allocation algorithms:

(a) FIFO (First-In-First-Out)

(b) LIFO (Last-In-First-Out)

(c) Round Robin (time slice size 1 time unit)

Make a table that has processes as rows and time as columns so that there is H when the process is on hold and E when it is being executed.

TIP: there should be 20 time columns, 00...19 (why?)

3. Let us continue working with the scheduling of the previous task. Add the following time allocation algorithms:

(d) SJF (Shortest Job First)

(e) SRT (Shortest Remaining Time)

(f) HRRN (Highest Response Ratio Next)

4. a) What are the 3 conditions that must be met so that deadlocking is possible?

   b) Students who want to enroll in "Model Railroading II" course at the local university are required to obtain permission from the instructor and pay a laboratory fee. These two requirements are fulfilled independently in either order and at different locations on campus. Enrollment is limited to 20 students; this limit is maintained by both the instructor (who will grant permissions to only 20 students) and the financial office (which will allow only 20 students to pay the laboratory fee). Suppose the registration system has resulted in 19 students having successfully registered for the course, but with the final space being claimed by two students – one who has only obtained permission from the instructor, and another,

PO BOX 20, FI-53851 LAPPEENRANTA, FINLAND
E-mail: olli-pekka.hamalainen@lut.fi

who has only paid the fee. This results in a deadlock situation, because both registers now have reached the 20-student limit. Which condition for deadlock is removed by each of the following solutions to the problem?

i) Both students are allowed in the course.
ii) The class size is reduced to 19, so neither of the two students is allowed to register for the course.
iii) The competing students are both denied entry to the class and a third student is given the twentieth space.
iv) It is decided that the only requirement for entry into the course is the payment of the fee. Thus the student who has paid the fee gets into the course, and the other student is denied entry.

c) How would you suggest the registration process to be handled for the next semester in order to avoid situations like this? (Provided that the organizers want to keep both requirements.) Which condition does your solution remove?

5. There is a T-intersection where a smaller side road B from south connects to a larger road A, which is in east-west orientation. The intersection is controlled by traffic lights which are equipped with sensors that record three variables (num-A-east, num-A-west and num-B) as well as three lists time-A-east, time-A-west and time-B). The "num" variable represents the number of cars that are present in the detection area (say, from 0 to 200 meters behind the respective traffic light post) and "time" list includes the times how long each one of them has spent in the detection area. When a car passes the traffic light post, its "time" value is dropped off from the list (new entries to list go to end of list, oldest ones are at the top – so, the 1st element is always the one that is next dropped off). These variables and lists are updated automatically by the sensor program at an interval of, say, 50 milliseconds. The traffic light steering is easy: there is just one control variable X. If X = 1, the cars travelling along road A have green light, and if X = 0, the cars travelling along road B have green light. (So, no arrow lights or such.)

a) Using the available variables and lists, write a steering algorithm that can control the traffic lights in such a manner that traffic is smooth and "fair" (no deadlocks, no starvation). Prepare to explain your reasoning in the exercise session. (Note: There are infinitely many solutions, you can be creative. Also, don't go overboard polishing your algorithm; remember that done is better than perfect.)

b) In this kind of applications, safety must always be a top priority. What kind of failsafe mechanisms or check-ups would be good to include in the algorithm? (You can include these in the algorithm or just explain in general.)

BONUS: 1 extra point if you are able to demonstrate how your algorithm works by simulation! (In this case, modify your submission file name by adding the extension "-BONUS" after the task numbers.)