01.04.2015

BLG 312E – Computer Operating Systems

Homework 3

Submission Deadline: 24.04.2015, 23:55
(Late Submission Deadline: 25.04.2015, 23:55)

- You are expected to work individually on all exams and homeworks. All forms of collaboration are discouraged and will be treated as cheating. This includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (including Internet resources) even when attributed. Serious offenses will be reported to the administration for disciplinary measures. All parties involved in the act will be treated equally.
- You have to achieve at least 20 out of 100 points on a homework for its submission to be accepted. Homeworks with lower grades will NOT be considered as submitted. Submitting only parts of the codes provided in class will NOT be sufficient to achieve a grade of 20.
- Submission of the 3rd homework is compulsory for being allowed to take the final exam.
- Late submissions will be allowed for only 24 hours after the submission deadline. Regardless of the time of the late submission and its reason, **ALL** late submissions will be graded as 50% of the original grade the homework deserves. (Please note that in the case of late submissions, for a homework to be considered as submitted, its original (unreduced) grade should be at least 40).

What to submit: You should submit your source file(s) via the Ninova system. (No report file is required; however, it is expected that you include comments in your source file)

Program: Write and test a C program that implements the described behavior below.

Description: In a hospital, there are 5 departments that deal with different types of problems. In each *department* there is one *nurse* and two *doctor*s. When a patient arrives at the hospital he/she goes to the appropriate *department* directly.

When a patient arrives at a *department*, the *nurse* registers him/her. After this step, the *nurse* sends the patient into the waiting room. Whenever one of the *doctor*s becomes available, he/she calls the first patient in line and treats him/her. While the *doctor*s treat patients, the *nurse* continues registering incoming patients.

In this hospital there is only one *X-Ray room* with 3 machines and all departments have to use the same room. If a patient has been tagged as needing an X-Ray, after the initial treatment, the doctor takes the patient to the X-Ray room and gets the X-Ray done. If all of the X-Ray machines are full, the doctor has to wait. Assume that taking an X-Ray requires a fixed time unit of 2 seconds. (Note: This 2 seconds is separate from the initial treatment time). After the X-Ray, the patient's treatment is completed and he/she leaves.

You are required to model 5 *departments* in this hospital where each *department is* a **process** with the *doctor*s and the *nurse* as **threads** of the processes. Each waiting room should be modeled as a first-in-first-out queue which can be accessed both by the *nurse* and the *doctor*s. The nurse needs 2 seconds to register each patient (this time is fixed). Treating a patient takes *n* minutes for a doctor and *n* depends on the nature of the problem the patient has. To simulate patients with different types of problems, a *doctor* thread should read the time it will take to treat the current patient from an input file. In this input file, beside the treatment time it will also be stated whether the patient requires an X-Ray or not. Each department will use a **separate input file**. (**Hint!** To simulate the waiting times during registration, treatment and the time it requires to have an X-Ray taken, you can use the "sleep" command.)

Test: Your program will be tested in the form:

```
./program input_file1 input_file2 input_file3 input_file4 input_file5
```

where the names of the input files need to be given as parameters.

Please preserve the order and meaning of the program arguments.

Please test your program with different order files and make sure to achieve expected results.

Please check that your program correctly removes all allocated resources (e.g. shared memory locations, semaphores, and any others you have used).

Input file format: Input files for testing your program must be in the format given below where each line contains the time it will take to treat each patient and whether the patient requires an X-Ray (1) or not (0). Note that you will have 5 input files (one for each department).

```
3 0
2 0
…
```

```
4 1
3 0
…
```

Output format: Your program must print out the events on the screen in their order of occurrence in the format given below.

```
Department 1 Nurse: Patient 1 is registered
Department 2 Nurse: Patient 1 is registered
Department 2 Nurse: Patient 2 is registered
Department 1 Nurse: Patient 2 is registered
Department 1 Doctor 1: Patient 1 is treated
Department 1 Doctor 2: Patient 2 is treated
Department 2 Doctor 1: Patient 2 is treated
Department 2 Doctor 2: Patient 1 is treated (X-Ray taken)
…
```