11.03.2015

BLG 312E – Computer Operating Systems

Homework 2

Submission Deadline: 01.04.2015, 23:55
(Late Submission Deadline: 02.04.2015, 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 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 additional 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:

In this homework, there will be one process with three threads.

Description: In a hospital, there are 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 *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.

You are required to model ONE *department* in this hospital as a process with the *doctor*s and the *nurse* as threads of this process. The 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 minutes 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.

(**Hint!** To simulate the waiting times during the registration and treatment stages, you can use the "sleep" command.)

Please note: For this homework, you are required to model the *department* as ONE process with THREE threads (one *nurse* thread and two *doctor* threads). Solving the problem with multiple processes and no threads, will NOT get any points.

Test: Your program will be tested in the form:

./program input_file output_file

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

Please preserve the order and meaning of the program arguments.

Please test your program with different order input 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

```
3
6
5
…
```

Output file format: Your program must print the events into a file in their order of occurrence in the format given below.

```
Nurse: Patient 1 is registered
Nurse: Patient 2 is registered
Doctor 1: Patient 1 is treated
Nurse: Patient 3 is registered
Doctor 1: Patient 2 is treated
Doctor 2: Patient 3 is treated
…
```