25.03.2016

BLG 312E – Computer Operating Systems

Homework 2

Submission Deadline: 06.04.2016, 23:55
(Late Submission Deadline: 07.04.2016, 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 several threads.

Description: In a company new employees are hired after a written exam and an interview. A “Registrar”, a “Written Exams Committee (WEC)” and an “Interviewer” handle the whole hiring process. In this homework, you will only implement the written exam part.

The written exam consists of questions from *T* different topics. Each member of the WEC is responsible for preparing one question for only one topic. Therefore, the WEC consists of *T* members. Each member of the WEC should read the topic assignments from an input file. After learning the topic, each WEC member should prepare a written exam question on the assigned topic and write the question to a question sheet. When all questions are ready, each member of the WEC should check all other questions prepared by the other members and approve them. When all questions are approved by the WEC, the preparation of the written exam is completed.

- You are required to model the *WEC* as a process with the *members of the WEC* as threads of this process.
- The question sheet where the members write their questions should be modeled as an array where each item on the array consists of a question prepared by the corresponding WEC member (e.g. the question at the 5th location on the array is prepared and written by WEC member no. 5) and a number for approvals. You can use the following data structure for representing each question in the array.

```
typedef struct question{
    char text[30];
    int approvals;
} Question;
```

- Preparing a question takes *n* minutes and *n* depends on the topic. To simulate questions for different topics, a WEC member thread should read the time it will take to prepare the question from an input file. **Hint!** To simulate the question preparation times, you can use the "sleep" command and treat minutes as seconds (i.e. sleep for *n* seconds to simulate *n* minutes).
- After all the questions are prepared, the WEC members check all the questions concurrently (i.e. multiple WEC members can examine the same question at the same time). Each WEC member checks the questions prepared by the other members and approves them (i.e. increases the number of approvals). Checking a question takes 1 minute regardless of its topic (sleep for 1 second to simulate this). The last WEC member approving a question (i.e., approvals=T-1) needs to print on the screen that question is ready.
- In the input file which contains the topic assignments, each line consists of three fields, i.e. the WEC member number, the topic assigned to that member and *n* (the time it takes to prepare a question on that topic).

Please note: For this homework, you are required to model the WEC as ONE process with *T* threads, where *T* is the number of topics, i.e. the number of committee members, and is supplied as a command line parameter to the program. Solving this homework problem in any other way than using ONE process with *T* threads will NOT get any points.

Test: Your program will be tested in the form:

```
./program input_file number_of_WEC_members
```

where the name of the input file and the number of WEC members need to be given as input parameters.

Please preserve the order and meaning of the program arguments.

Please test your program with different 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 WEC member number, the topic assigned to that member and the time it takes to prepare a question on that topic.

```
1 computer_operating_systems 5
2 software_engineering 2
3 system_programming 4
…
```

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

```
WEC member 2: A question is prepared on software_engineering
WEC member 3: A question is prepared on system_programming
WEC member 1: A question is prepared on computer_operating_systems
WEC member 3: Question 1 is ready
WEC member 3: Question 2 is ready
WEC member 1: Question 3 is ready
…
```

Contents of the array: For the given input above, the maintained array of questions can be as follows. Each item on the array consists of a question prepared by the corresponding WEC member (e.g. the question at the 2nd location on the array is prepared and written by WEC member no. 2) and the number of approvals for that question.

	text	*approvals*
1st question	computer_operating_systems question	2
2nd question	software_engineering question	2
3rd question	system_programming question	2