

## BLG 312E – Computer Operating Systems

### Homework 3

**Submission Deadline:** 02.05.2016, 23:55

**(Late Submission Deadline:** 03.05.2016, 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 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 additional report 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 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. There is only 1 Registrar and only 1 Interviewer in the office. In this homework, you are asked to implement the program simulating the whole hiring process.

A “Registrar” takes an application form from each applicant and places it in the “Applications Box”. Assume that the whole application process takes 3 minutes. **Hint!** To simulate the time, you can use the “sleep” command and treat minutes as seconds (i.e. sleep for 3 seconds to simulate 3 minutes).

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 5<sup>th</sup> location on the array is prepared and written by WEC member no. 5) and a number for approvals. 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 (sleep for  $n$  seconds to simulate this) 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.
- 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).

When all the questions are ready (i.e. approvals are also completed), the Registrar makes the written exam. The exam takes  $M$  minutes.  $M$  is fixed and is given to the program as a command line parameter. At the end of the exam, the Registrar writes the scores down in the application forms by generating random integers between 0 and 50. Use the following data structure for representing each application form and assume that there are 3 applicants.

```
typedef struct application{  
    int exam_score;  
    int interview_score;  
    int total_score;  
} ApplicationForm;
```

When the written exam is completed, the “Interviewer” calls in each applicant for an interview. The Interviewer takes the application forms from the Applications Box. The interview for each applicant lasts for  $I$  minutes.  $I$  is fixed and is given to the program as a command line parameter. The Interviewer scores each applicant after the interview by generating a random integer between 0 and 50 and writes the score down in the applicant's application form.

When all interviews are finished, the Registrar calculates each applicant's total score based on his written exam score and his interview score. Assume that the whole scoring process takes 3 minutes (sleep for 3 seconds to simulate this).

In this homework the Registrar, the WEC and the Interviewer should each be modeled as a process (i.e. there are 3 processes). The WEC should be modeled as a 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. The other two processes do not have multiple threads. You are not required to model applicants and the Applications Box.

**Test:** Your program will be tested in the form:

```
./program input_file number_of_WEC_members written_exam_duration interview_duration
```

where the name of the input file, the number of WEC members ( $T$ ), duration for the written exam ( $M$ ) and the duration for each interview ( $I$ ) 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. An example input file is given below.

1	computer_operating_systems	5
2	software_engineering	2
3	system_programming	4

**Contents of the array (used for the written exam):** 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 2<sup>nd</sup> location on the array is prepared and written by WEC member no. 2) and the number of approvals for that question.

	<i>text</i>	<i>approvals</i>
<i>1<sup>st</sup> question</i>	computer_operating_systems question	2
<i>2<sup>nd</sup> question</i>	software_engineering question	2
<i>3<sup>rd</sup> question</i>	system_programming question	2

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

The Registrar started taking applications.  
The Registrar finished taking applications.  
The Written Exams Committee started preparing questions.  
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  
The Written Exams Committee finished preparing questions.  
The Registrar started the written exam.  
Written exam score of applicant 1 is 33.  
Written exam score of applicant 2 is 17.  
Written exam score of applicant 3 is 42.  
The Registrar finished the written exam.  
The Interviewer started interviews.  
Interview score of applicant 1 is 37.  
Interview score of applicant 2 is 22.  
Interview score of applicant 3 is 15.  
The Interviewer finished interviews.  
The Registrar started calculating total scores.  
Total score of applicant 1 is 70.  
Total score of applicant 2 is 39.  
Total score of applicant 3 is 57.  
The Registrar finished calculating total scores.