26.03.2014

BLG 312E – Computer Operating Systems

Homework 2

Submission Deadline: 11.04.2014, Friday, until 23.00.

- Homeworks submitted after the deadline and/or not submitted via Ninova system will NOT be accepted.
- You are required to submit the 3rd homework. Please note that 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.
- You are expected to work individually on all exams and homeworks. All forms of collaboration are discouraged and will be treated as cheating.

Where to submit: You should submit your homework program code via the Ninova system: ninova.itu.edu.tr

What to submit: You should submit your source file(s) via the Ninova system. (No additional report file is required; however, it is expected from you to include code 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 two threads.

Description: In a restaurant, a *service unit* consists of a *cashier* and a *server*. The *cashier* takes the orders and gets the payment while the *server* prepares the order. The *cashier* puts the order on a queue. The *server* accesses this queue, removes the order and prepares it. They can work in parallel, i.e. while the order of a customer is being prepared by the *server*, the *cashier* can continue taking other orders.

You are required to model the *service unit* as a process with the *cashier* and the *server* as threads. The queue where the orders are placed is a shared location between the *cashier* and the *server*. **Please note:** For this homework, there is only ONE *service unit*. Solving the problem with multiple processes and no threads, will NOT get any points.

There are three types of orders available in this restaurant:

- Orders of Type 1 take 1 second to prepare,
- Orders of Type 2 take 2 seconds to prepare,
- Orders of Type 3 take 3 seconds to prepare.

To simulate the arrival of different types of orders, the *cashier* should read the orders from an input file. For the *cashier*, taking an order lasts for 1 second regardless of the type of the order.

(**Hint!** To simulate the waiting times during the order taking and order preparing stages, you can use the “sleep” command.)

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: Order input files for testing your program must be in the format given below where each line contains the type of the order as 1, 2 or 3.

```
1
2
1
1
1
2
3
2
1
3
```

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

```
Cashier: Type 1 is ordered
Cashier: Type 2 is ordered
Server: Type 1 is ready to be served
Cashier: Type 1 is ordered
Server: Type 2 is ready to be served
Server: Type 1 is ready to be served
…
```