27.03.2014

BLG 312E – Computer Operating Systems

Homework 3

Submission Deadline: 27.04.2014, Sunday, 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 multiple processes with two threads per process.

Description: In the restaurant described in the 2nd homework, there are multiple s*ervice units*. Each *service unit* again consists of a *cashier* and a *server* both of which work as explained in the 2nd homework. The *server*s prepare the orders using the *cooking area*. However, the *cooking area* has a capacity of C, which means that at any given time, only C *server*s can prepare an order in parallel, while the others have to wait until at least one of the *servers* finishes preparing its order and leaves the *cooking area*.

Please note: For this homework, there are multiple *service units*. You are required to model each *service unit* as a process with its *cashier* and its *server* as threads. Solving the problem in any other way 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. However, different than from the 2nd homework, the input file contains two entries per line as shown in the example below. The first number shows the index of the *service unit* the order is placed at, the second number shows the order type. For any *cashier*, taking an order lasts for 1 second regardless of the type of the order. Thus, the *cashier* should read the given file line by line, take the orders with the index of its *service unit* and sleep for a second after taking each 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 num C
```

where the names of the input and the output files, the number of service units (num) and the capacity of the cooking area (C) 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:

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

Output file format:

```
Cashier 1: Type 1 is ordered
Cashier 2: Type 3 is ordered
Server 1: Type 1 is ready to be served
…
```