

# BLG 312E – Computer Operating Systems

## Homework 3

**Submission Deadline:** 09.05.2018, 23:55

**(Late Submission Deadline:** 10.05.2018, 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, collaborating with another student, 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 held equally responsible.
- 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.

**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.)

**Note: Please only use English letters in your submission files, even for your names.**

**Program:** Write and test a C program that implements the described behavior below.

**Description:** In an office there is a coffee machine that offers two types of coffee. For preparing each type of coffee different amounts of water are needed as given below.

Coffee Type1: 1 cup  
Coffee Type2: 2 cups

The coffee machine's water container holds  $W$  cups of water. For safe operation it is not allowed for the water level in the coffee machine to drop below 1 cup of water.

At most one person can take coffee from the machine at a time. A person wanting coffee has to select the coffee type he/she needs first. If preparing that type of coffee will bring the water level to below 1 cup, a warning is issued, the requested coffee is not prepared and the person has to wait until more water is added.

There is an office employee whose job is to add water to the coffee machine to its full capacity when the warning is issued. (**Note:** You can assume he has no other duties and sleeps until he is needed to fill the coffee machine).

### **Instructions:**

- You are required to implement each office employee trying to take coffee from the coffee machine and the employee responsible for filling the coffee machine with water as processes.
- To simulate the working of this system, inputs will be provided by way of an input text file. The file will contain two lines. The first line will show the total cups of water the coffee machine can hold (W) and the second line will be a list of numbers consisting of 1s and 2s, showing the types of coffee the employees will request. You can assume that the file format is correct and the second line contains only 1s and 2s. You do not need to check for errors.
- Use the appropriate Linux system calls and IPC primitives (i.e. semaphores and/or shared memory) as needed. For these IPC primitives you are required to use the following libraries (if needed): “semaphore.h” or “sys/sem.h” for semaphore operations and “shm.h” for shared memory operations.
- Please check that when exiting, 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 text files and must be in the format given below. For example, assume that the simulation will be done for 4 employees. The sample input file given below should be interpreted as follows: The coffee machine can hold 4 cups of water. The first employee wants coffee of Type1, the second and third employees want coffee of Type2 and the fourth employee wants coffee of Type1.

sample\_input.txt

4
1 2 2 1

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

`./program_name input_file_name`

Please test your program with different input files and make sure to achieve expected results.

**Output format:** Your program must print the events on the screen in their order of occurrence in the format given below. For the example given above, the output will be printed as follows:

```
SIMULATION BEGINS
Current water level 4 cups
Employee 1234 wants coffee Type 1
Employee 1234 SERVED
Current water level 3 cups
Employee 1235 wants coffee Type 2
Employee 1235 SERVED
Current water level 1 cups
Employee 1236 wants coffee Type 2
Employee 1236 WAITS
Employee 896 wakes up and fills the coffee machine
Current water level 4 cups
Employee 1236 SERVED
Current water level 2 cups
Employee 1237 wants coffee Type 1
Employee 1237 SERVED
Current water level 1 cups
Employee 896 wakes up and fills the coffee machine
SIMULATION ENDS
```