

**T.C.
DOKUZ EYLUL UNIVERSITY**

**FACULTY OF
ENGINEERING**

**DEPARTMENT OF
COMPUTER ENGINEERING**

**2022 - 2023
FALL SEMESTER**

**CME 3205
OPERATING SYSTEMS**

**ASSIGNMENT 1
DEU CAFE**

**DUE DATE
23:55 – 15.12.2022**

In this assignment, you are asked to create a simulation of a cafe using POSIX threads, mutexes and semaphores. Your program should have the following functionalities and operations.

Your cafe should only sell one product, a normal coffee to make it simpler. The program scenario is, consumers come to the cafe and tries to find an empty register to buy a coffee. Registers take the order, prepare it and give it to the customer and the customer leaves.

Your program should contain global static variables to manage your code. You must use the given global static variables below with specified name and value. This will allow you to experiment with different values easily and allow us to evaluate your assignment easily.

#define CUSTOMER_NUMBER 25

The definition above will be used for how many customers your program should work for. After this many customers are created and serviced, the program should terminate. This will also be used to identify customers by given them IDs from 0 to 24.

#define CUSTOMER_ARRIVAL_TIME_MIN 1

The definition above will be used for how much time, at minimum, should pass between customers arriving.

#define CUSTOMER_ARRIVAL_TIME_MAX 3

The definition above will be used for how much time, at maximum, should pass between customers arriving.

#define REGISTER_NUMBER 5

The definition above will be used for how many registers your program should work for. This will also be used to identify registers by given them IDs from 0 to 4.

#define COFFEE_TIME_MIN 2

The definition above will be used for minimum waiting time for ordering and receiving a coffee by customers. This represents that minimum waiting time should be 1 second.

#define COFFEE_TIME_MAX 5

The definition above will be used for minimum waiting time for ordering and receiving a coffee by customers. This represents that maximum waiting time should be 5 seconds.

The program algorithm of this assignment should be similar to steps given below:

Customer arrives after a random number of seconds passed defined by CUSTOMER_ARRIVAL_TIME_MIN and CUSTOMER_ARRIVAL_TIME_MAX variables.

Customer tries to find an open register and waits if there is none. When he/she finds an empty register, he/she gives his/her order and get his/her coffee. This process should take a random amount of time, defined by COFFEE_TIME_MIN and COFFEE_TIME_MAX variables.

All of this process should be outputted in terminal (CLI) to keep the user informed. Your outputs should be similar to below examples.

CUSTOMER <CUSTOMER_ID> IS CREATED AFTER <TIME> SECONDS.

This output should be given when you create a customer with given <CUSTOMER_ID> and after waiting a random time given in <TIME>.

CUSTOMER <CUSTOMER_ID> GOES TO REGISTER <REGISTER_ID>.

This output should be given when a customers (with <CUSTOMER_ID>) finds an empty register (with <REGISTER_ID>) and starts the ordering process.

CUSTOMER <CUSTOMER_ID> FINISHED BUYING FROM REGISTER <REGISTER_ID> AFTER <TIME> SECONDS.

This output should be given when a customers (with <CUSTOMER_ID>) finishes buying coffee from register (with <REGISTER_ID>) and prints out how much time did it took (with <TIME>), which was randomly selected.

You are free to use mutexes or semaphores for this assignment, but as you can see from the requirements, you need to know which register is used by which customer. Consider this requirement when you are writing your code.

Try to write your assignment as simple as possible, without using specialized secondary libraries, tools or etc. You may have created the best assignment, adding a lot functionality (such as advanced string formatting for terminal) that was not requested. However, this increases the chance of your assignment not working on our computers, therefore getting a significantly reduced grade for your top tier assignment.

A large portion of your code will be graded based on your console output and for an assignment that does not compile or work on our computers will get an automatic zero from code execution and output criteria given below. We will not change your code or try to make it work because it is not feasible nor it is fair to other students. Because of this, make sure your program compiles and runs on different Linux machines, otherwise your grade will be reduced dramatically.

UPLOAD REQUIREMENTS:

You are required to upload the C language code file you have written to the SAKAI. You should compile and test it to make sure it works before upload. You can use a IDE during development but your code must work correctly using console compilation and execution commands (using “gcc” and “./”). If we cannot correctly compile and execute your code, your grade will be reduced according to grading criteria. You do not need to upload a complied version of your code, just your C language code, because we cannot prove if it is a original compilation or not. For this reason, uploading a complied file is not necessary nor it is requested.

The file you are required to upload are given below with an explanation and an example.

(GROUP_ID)_(STUDENT1_ID)_(STUDENT2_ID).c

(Source code you have written in C language)

Example = Group5_2022510123_2022510124.c

Late or no submissions will be graded zero. You can see the basic grading table of this assignment below.

CRITERIA	GRADE
Correct naming of upload files	10
Correct English variable names and English comments.	10
Correct Code Quality	20
Correct Code Execution	30
Correct Console Output	30
TOTAL GRADE	100
CHEATING OR ANY OTHER FORM OF PLAGIARISM	—∞

If you have any questions or problems regarding this lab paper, you can ask about it in our lab sessions.

If you wish, you can also ask it in class forums or assignment page comments.

However, please do not send emails because that would lead to asking the same questions over and over again.

GOOD LUCK TO YOU ALL!