Waiting Students Problem

OS2 Final Lab Exam Spring 2022

Problem Statement:
A university offers **K** subjects, where a student selects **one** subject at any given time. A student will not choose two subjects. The lab available in the university has **m** systems. Students coming to the lab will use it as per following rules:

1. Only students studying the same subject can use the lab together. Suppose the lab currently has students of subject 1. Then any student of another subject wishing to enter the lab has to wait in a **queue** until they get their turn. But students of subject1 can enter and use the lab.

2. Only m students at a given time can use the LAB as there are **m** systems. If more than m students of the subject currently using the lab arrive, then the remaining students will have to wait in the same queue.

You are given that:
1. The students arrive individually at an exponential rate with the average being **λ** arrivals per millisecond.
2. A student can take any one of the subjects with equal probability, i.e., **1/K**.
3. You can stop your program after **n** students have accessed the lab.

You have to develop programs in C/C++ to solve the following problems using semaphores which are increasing in the order of difficulty. Please start with question (A) and after you complete it, please go to the next question.

(A) Design a solution for synchronization of lab usage among the students using semaphores and where subjects are two, i.e. **K=2**.

(B) Design a solution when the total subjects, **K>=2.**

(C) Assuming that each student entering the lab leaves it in finite time, design a starvation free solution for above scenarios such that a student waiting in the queue will eventually leave.

Input to your program: K m λ n
A sample input can be: 10 10 5 100

Output of your Program:
Your program should output two things:

I. **Average Time to enter the Lab:** the average time taken in milliseconds for a student to enter the lab after they have made a request to enter.
II. **Log:** a log of all the entries as explained below.

Log Format: Each student thread has to log the time they requested entry to the lab, time they got access to the lab and time they have exited the lab.

Student 1 group k2 requested the lab 10:00
Student 1 group k2 entered the lab 10:01
Student 3 group k2 exited the lab 10:05
Student 1 group k2 exited the lab 10:20
.
.
.
Student 4 group k5 requested the lab 10:35
Student 3 group k6 entered the lab 10:32
Student 4 group k6 exited the lab 10:35
Student 3 group k6 entered the lab 10:40

Naming convention: Please name your program as per the problem that you are solving in the following format: WaitStudents-<RollNo>-A.c; WaitStudents-<RollNo>-B.c; WaitStudents-<RollNo>-C.c.

Then zip all your programs with this name: WaitStudents-<RollNo>-All.zip. Then you can upload it on the classroom page.

Note: A trivial solution to this problem is using mutex lock where you allow only one student to enter the lab. Naturally such a solution is very inefficient and hence not desirable. So please note that you will be graded based on the efficiency of your solution.