CSE 314: Offline on IPC

Not Another Completely "Copiable" Offline

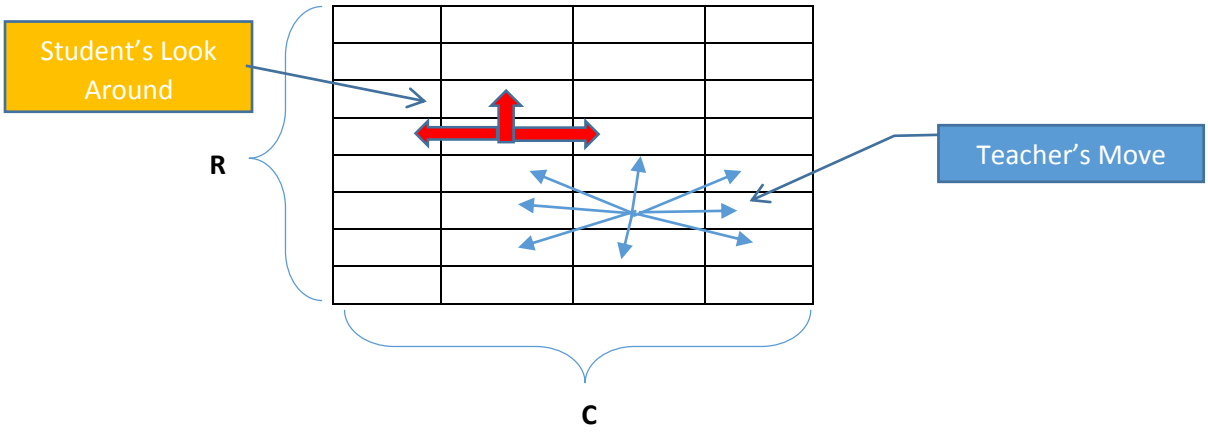

*Figure 1: A lab layout as **R×C** grid*

N students will appear an online test in a lab which is arranged as **R×C** grid, where **R×C = N**. Each grid contains a PC for one students. The PCs will be assigned to the students randomly. Each student has an ID between 1-N. The duration of the online is **D** seconds.

Among the **N** students, **M** students did not take any preparation for the test. After every **K** seconds they make a random decision whether to look around for hacking some solution idea. If they decided to look around, they randomly peep at the PC located immediately at the left, right or front side for **L** seconds and then look back to their own PC again to avoid getting caught by the teachers.

There are **T** teachers as invigilators. Each teacher has an ID between 1-T. Each teacher will start at a random grid and inspect the student seated at his current position. After every **W** seconds they make a random move to one of the adjacent grids. Each grid can contain at most one teacher. If any teacher finds the student at his current position looking at other, he will immediately expel that student and add an entry in the hall of shame shared by all teachers. Each entry will be of the form: "Teacher # has expelled Student # for looking at Student #".

Task list

- Use POSIX threads, semaphores and mutex locks to simulate the above scenario
- You have to define two threads
    - Teacher thread: simulates the behavior of a teacher
        - void * Teacher(void * ID)
    - Student thread: simulates the behavior of both types of students
        - void * Student(void * ID)
        - Each student would check whether he is cunning or not from a global data
- Use **sleep()** for simulating all kinds of time related event
    - #include <unistd.h>

CSE 314: Offline on IPC

    - unsigned int sleep(unsigned int seconds);
- Your program needs to store the state of **M** cunning students and **T** teachers in a file in **append** mode at every **V** seconds
- Your program must continuously print on terminal
    - Which cunning student is currently looking at which student
    - Which teacher is currently at which grid
    - Which teacher has just expelled which cunning student for looking at which student
- Your program must declare all the user defined parameters as defined constant using **#define** like:
    - #define SEED time(0)
    - #define N 30 /* Number of total students */
    - #define M 5 /* Number of cunning students */
    - #define T 3 * Number of total teachers */
    - …
- Make the use of main thread wisely
    - Create N student threads each one having unique ID from 1-N (use array of threads)
    - Randomly choose M cunning students among the N students and keep this information global
        - #include <stdlib.h> /* srand, rand */
        - #include <time.h> /* time */
        - srand (SEED); /* initialize random seed: */
        - int iSecret = rand() % 10 + 1; /* generate a random number between 1 and 10 */
    - Randomly assign PCs to students
    - Create T teacher threads each one having unique ID from 1-T
    - Calculate remaining seconds
    - Save data in file
    - Print the hall of shame at the end
- You have to submit a **report** in pdf format containing
    - List of all shared data items
    - Critical sections of each thread (Student, Teacher and main)
    - List of mutex locks and semaphores with their purpose
- You have to bring the **hard copy** of your report during evaluation

A sample code format:

//----------------necessary headers---------------------

//---------------- user defined parameters ------------

//--------------------global data----------------------------

CSE 314: Offline on IPC

```
//--------------------function prototype-----------------

int main()
{
…

}

//--------------------function definition-----------------
```

Marking Scheme

Completeness	40%
Report contents	40%
Code Quality (algorithm efficiency, readability, robustness, comments etc.)	20%

Submission deadline

- Within 7 days