Study Tour In A Museum Project

### Md Saiful Islam Sabbir

*Department of computer science and engineering East West University Dhaka,Bangladesh*

*Email:* 2016-3-60-040@std.ewubd.edu

### Tamanna Tabassum

*Department of computer science and engineering East West University Dhaka,Bangladesh*

*Email:* 2018-1-60-082@std.ewubd.edu

### Sk Mohammad Asem

*Department of computer science and engineering East West University Dhaka,Bangladesh*

*Email:2017-3-60-068* @std.ewubd.edu

*Rownak Jahan*

*Department of computer science and engineering East West University Dhaka,Bangladesh*

*Email:* 2020-1-60-014@std.ewubd.edu

***Abstract*—.** A museum arranged a day long program for some school children in which students can participate in various activities. So, all students may not be able to watch it at the same time. Some students have to wait unless they get 3D glasses. Using POSIX threads, mutex locks, and semaphores implement a solution that synchronizes the activities of the students. The total number of students, the number of 3D glasses are passed as command line arguments. Once a student thread finish watching the movie, it should terminate. Once all the student threads are terminated, the main program should be terminated. Program should work for any number of students and 3D glasses.

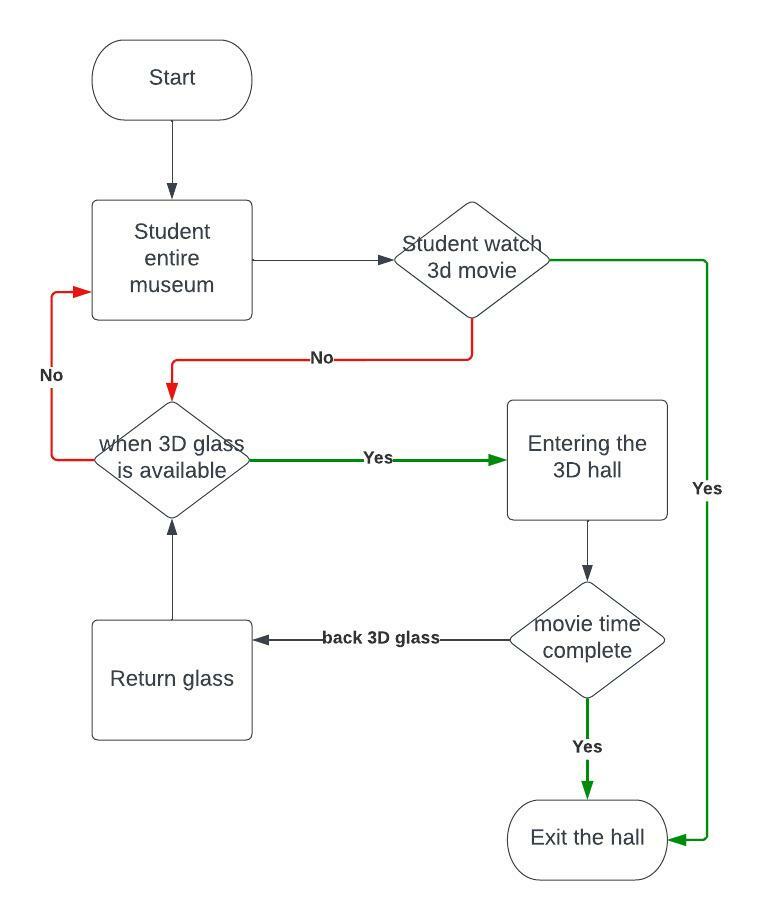
***Keywords—Mutex\_lock, p\_thread, sem\_wait, sem\_post,movie***

# Introduction

A Museum organized a day-long program for some younger students where they can participate in various activities. Among these activities, a 3D movie is included where 3D glasses are required. But unfortunately, All of them cannot watch the movie together because of the shortest 3D glass. So, There are 3 groups of students. The group will be students who have already gotten 3D glasses and started watching the movie. 2nd one will be those who wait for the 3D glasses and the other one appears when the waiting room is full then they have to leave. As a result, The main thing is , A few students have to wait while others are watching it. When they finished their watching, then the others will be able to watch this movie. Besides, If the waiting area is full, the rest of the students have to leave. Here, We incorporate some solutions for this problem, that how can solve it in a specific manner. The purpose of this solution is to make an easy and effective way of handling this kind of problem in other sectors also. Here, We utilized POSIX strings, mutex locks, and semaphores to execute an answer that synchronizes the exercises of the understudies. The absolute number of understudies, and the quantity of 3D glasses are passed as order line contentions. When an understudy string finishes watching the film, it ought to end. When all the understudy strings are ended, the principal program ought to be ended.

# Methodology

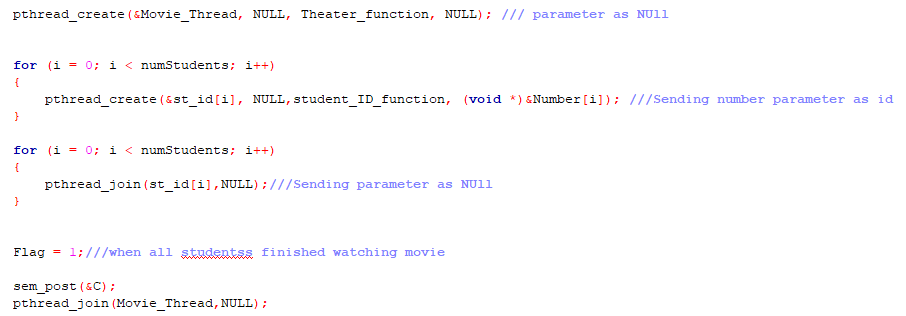
There were specific steps we followed for implementing our proposed methodology shown in Figure-1



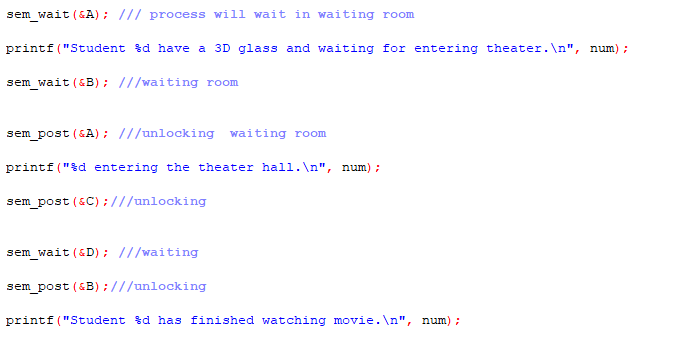
**Figure1**. Overall framework of our proposed approach

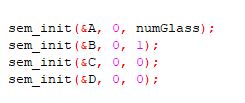
## Implementation

We are creating a project. We are mainly focus posix threads, mutex locks, and semaphores implement a solution that synchronizes the activities of the students. Firstly implement posix thread. Threads are two types : Single threading and Multi-threading. The main purpose of multithreading is to provide simultaneous execution of two or more parts of a program to maximum utilize the CPU time. A multithreaded program contains two or more parts that can run concurrently. In our case we are considering multi-threading. Because there can be two or more students in the waiting area. A mutex lockis a synchronization mechanism. We use it when there are many threads for execution. We have used the mutex lock because it provides mutual exclusion.



Also we are implementing semaphore as well. Semaphores are integer variables that are used to solve the critical section problem. we are using counting semaphore in our code. These semaphores are used to coordinate the resource access. If the resources are added, semaphore count automatically incremented and if the resources are removed, the count is decremented. Semaphores allow only one process into the critical section. They follow the mutual exclusion principle strictly and are much more efficient than some other methods of synchronization.





# Result Analysis & Evaluation

When we are input number of student 5 and number of 3D glass 3 then show figure-2:

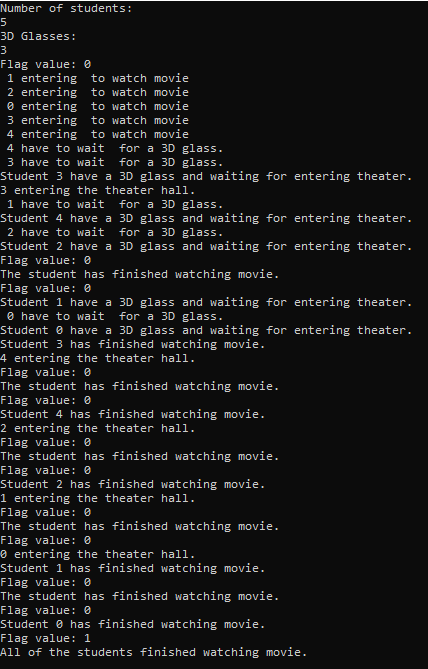


Figure:2

When number of student 3 and number of 3D glasses 5 then shown figure-3:

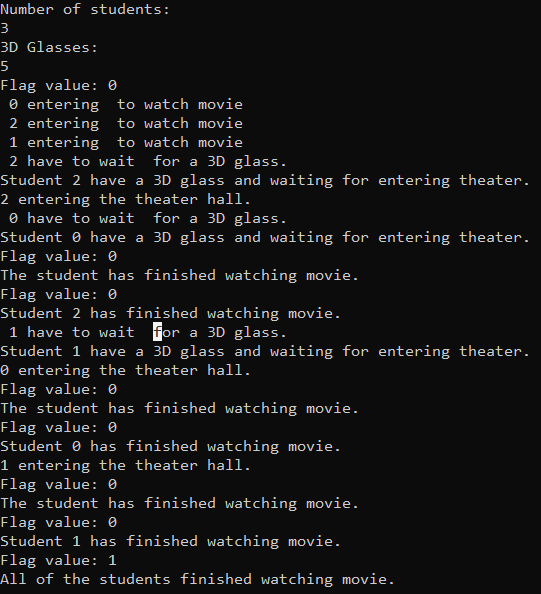


Figure-3

But we don’t need figure-3 .Because our project description mainly focus on number of student is less than number of glasses.

# Limitations

Though we have tried to minimize the limitations, there are still a few limitations. In this study, when The number of students will be more than the number of glasses, then code will run. But in our code, if the number of glasses is more than the number of students, then our code also run successfully. So, it is a limitation for this study.

# Conclusion & Future Work

A museum arranged a day long program for some school children in which students can participate in various activities. In this study, we distributed the 3D glass among the student. For implementing a solution we use posix thread, Single threading and Multi-threading. Then we find the solution for the study.