Assignment 7
Operating Systems (PG)
The 6D-Horror Movie Screening Problem

Objective:

To understand the working of semaphores. This assignment will be done in 2 phases.

Details:

A 6D horror movie attraction is being screened at Cinemax, Inorbit Mall. There are **n** patrons and only **one** screen in the theatre. The patrons **repeatedly** watch the movie (due to its pure awsomeness!).

The hall can accomodate a maximum of **C** patrons, where **C < n**.

However, due to management (financial) reasons, the movie can be screened only when the theatre hall is **full.** After a successful screening of the movie, the patrons **wander** around the mall before **returning** to watch another screening of the movie.

Due to timing issues in Hyderabad, the theatre screens the movie only for **T** times.

Requirements:

1. The movie screens only when the hall is full. i.e (**C** patrons in the hall)
2. No patron can **leave** the hall once the screening starts
3. No patron can **enter** the hall once the screening starts
4. A patron **has** to leave the hall once the screening is over and cannot sit and wait for the next screening to start. (i.e A patron after watching a screening, leaves the hall, wanders around the mall and then start the process all over again to watch the next screening)

To Implement:

Implement a C / C++ POSIX pthread program to realize the above problem scenario. You are required to understand the problem statement, figure the entities involved, number of semaphores required and come up with a solution that realizes the above problem.

Hints and Guidelines:

1. Read the man pages for pthread to become familiar with pthread working. This should have been done as part of your earlier assignment.
2. Have a good understanding of POSIX semaphores, and refer to man pages for sem_init, sem_wait, sem_post, sem_getvalue and sem_destroy.
3. You can walkthrough numerous tutorials on basic producer-consumer problem achieved through POSIX pthreads and semaphores available online to get familiar with POSIX pthread programming.

Deliverables:

PHASE1:

Submit a text file containing the details of

1. Your approach to this problem
2. Identification of the required entities (threads)
3. Semaphores to be used
4. Design decisions regarding each and every semaphore used
5. Other synchronization constructs (if you feel is/are necessary)
6. Potential deadlocks, race conditions that may occur if synchronization was not present
7. Flow chart which depicts how semaphores are handling synchronization issues in the above problem. (http://tinyurl.com/9thf8dz For example, the given link contains flowchart for the typical producer consumer synchronization problem. This image is for reference only.)

Upload format : PDF file. Name the file as “<rollnumber>_semP1.pdf”.
Example : 201106560_semP1.pdf

PHASE2:

You are required to submit the C/C++ code for the above problem along with a ReadMe file which contains the details of how to run the code and other information if required.

1. C/C++ code
2. ReadMe file

```
201106560/
        |----- 201106560_code.c
        |-----201106560_readme.txt
```

Tar.gz the above folder and name it as Assignment7.tar.gz

Deadline:

PHASE1: 6th November 2012 11:59PM
PHASE2: 12th November 2012 11:59PM