Assignment 4

Operating System Lab (**CS341**)

Department of CSE, IIT Patna

Date:- 29-Jan-2019 **Time:-** 3 hours

Instructions:

1. All the assignments should be completed and uploaded by 5 pm. Marks will be deducted for the submissions made after 5 pm.
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments (if necessary) are mandatory.
4. You should zip all the required files and name the zip file as ***roll_no*.zip**, eg. **1501cs11.zip.**
5. Upload your assignment (**the zip file**) in the following link: https://www.dropbox.com/request/Zrducx18qeuD4ihbhjCh

Questions:

1. Write a simple C program to creates 3 threads with the pthread_create() routine. Each thread prints a "CSE OS LAB, by Thread ID: %d" message, and then terminates with a call to pthread_exit().

2. Write a simple C program to creates 5 threads. Define 3 variables (global, static and local) and show how update on each variables are reflected in multiple threads.

3. Write a simple C program to creates 5 threads and demonstrates how to "wait" for thread completions by using the Pthread join routine.
   Define a subroutine **void *Busy (void *tid)** which takes thread id as argument and keeps the thread busy.
   Display the following messages in your code in appropriate positions:
   "Creating thread %ld\n"

"ERROR; return code from pthread_create() is %d\n"

"ERROR; return code from pthread_join() is %d\n"

"Completed join with thread %ld having a status of %ld\n"

"Exiting from program.\n"

4)

Write a C program with the following requirements: -

a) The program should have exactly 2 threads. You should use the standard pthreads library for thread creation and management. This includes the main thread. So you have to spawn 1 additional thread after the program starts.

b) The main thread should print odd numbers on the console with one number on each line Example:

1 3 5 7 …

c) The second thread should print even numbers on the console, again with one number on each line Example:

2 4 6 8 …

d) The program should stop after printing 10000 numbers (from both threads).

Notice that the output is jumbled up and the numbers are not in order.

5) On the above described question use thread synchronization mechanism of semaphores to synchronize the output.

For example, the main thread prints an odd number and passes on the information to the second thread about this. In the mean time, the second thread will be waiting for the particular event to occur and begins its work i.e. printing the number only after the particular event occurs. On a similar note the main thread after printing an odd number will waits for that particular event this time from the second thread to make sure that the following even number is printed before printing the next odd number. Utilize thread synchronization mechanism of semaphores and pthreads to achieve the desired result.