***Jaypee Institute of Information Technology, Noida***

***Operating System and System Programming lab [15B17CI472]***

***Assignment- 6***

1. Write simple thread program which creates 5 threads with the pthread\_create() routine. Each thread prints a "Hello World!" message, and then terminates with a call to pthread\_exit( ).
2. Write a program where 2 threads are created and each thread print information like (Name,

Hall No., employee ID, branch).

1. Write a program where 2 threads communicate using a single global variable “balance” and initialized to 1000..Thread 1 deposits amount = 50 for 50 times and prints the balance amount and thread 2 withdrawals amount=20 for 20 times and prints the final balance. Execution of thread 1 and thread 2 should not interleave.
2. Write a program where 2 threads operate on a global variable “account” initialized to 1000. There is a deposit function which deposits a given amount in this “account”: int deposit(int amount). There is a withdrawal function which withdraws a given amount from the “account”: int withdrawal(int amount). However there is a condition: withdrawal function should block the calling thread when the amount in the “account” is less than 1000, i.e. you can’t withdraw if the “account” value is less than 1000. Threads calling the deposit function should indicate to the withdrawing threads when the amount is greater than 1000.
3. Write a thread program which demonstrates how to "wait" for thread completions by using the Pthread join routine. Since some implementations of Pthreads may not create threads in a joinable state, therefore explicitly created attribute in a joinable state so that they can be joined later. Created thread should perform the calculation of sum =sum + sin(i) + Tan(i), where i =0 to 1000. Print the out in the following manner

|  |
| --- |
| Main: creating thread 0 |
| Main: creating thread 1 |
| Thread 0 starting... |
| Main: creating thread 2 |
| Thread 1 starting... |
| Main: creating thread 3 |
| Thread 2 starting... |
| Thread 3 starting... |
| Thread 1 done. Result = -3.153838e+06 |
| Thread 0 done. Result = -3.153838e+06 |
| Main: completed join with thread 0 having a status of 0 |
| Main: completed join with thread 1 having a status of 1 |
| Thread 3 done. Result = -3.153838e+06 |
| Thread 2 done. Result = -3.153838e+06 |
| Main: completed join with thread 2 having a status of 2 |
| Main: completed join with thread 3 having a status of 3 |
| Main: program completed. Exiting. |

1. Write a program that computes the square roots of the integers from 0 to 10 in a separate thread and returns an array of doubles containing the results. In the meantime the main thread should display a short message to the user and then display the results of the computation when they are ready.
2. Write a thread program which demonstrates Pthreads condition variables. The main thread creates three threads. Two of those threads increment a "count" variable, while the third thread watches the value of "count". When "count" reaches a predefined limit, the waiting thread is signaled by one of the incrementing threads. The waiting thread "awakens" and then modifies count. The program continues until the incrementing threads reach TCOUNT. The main program prints the final value of count.
3. Demonstration of Race Condition in producer and consumer problem using thread implementation.