Name Shehryar

Roll NO Bsf2000745

Class BS IT Evening Fourth semester

Subject Operating System

# Activity one

3.1 Which of the following components of program state are shared across threads in a multithreaded process?

a. Register values

b. Heap memory

c. Global variables

d. Stack memory

**Answer with Reason:**

Heap memory and Global variable are shared among the thread of a multi-thread process. Every thread has its separate set of register values and a separate stack.

3.2 Can a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single processor system? Explain.

**Answer with Reason:**

A multi-thread system consist of multiple user level thread that cannot make use of the different processor in a multiprocessor system concurrently. The operating system perceives only a single processor and will not schedule the different threads of the process on separate processor. Thus, executing multiple user-level thread on a multiprocessor system has no benefit.

# Activity 2

Servers can be designed to limit the number of open connections. For example, a server may wish to have only N socket connections at any point in time. As soon as N connections are made, the server will not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections.

**Answer with Reason:**

Following are the code to this question:

**code:**

```
wait(semaphore *S)  //define method
{
       S-> count++;  //increment value
        if( S -> count = 100)  //define condition that check count is equal to 100
              block; //use block keyword
        else
             add this connection to S -> list;  //add value in list
}
signal(semaphore *S)  //pass the value in method parameter
{
     S -> count--;  //decrement value
     if(S ->count < 100)  //define condition that check value count is less then 100
        removeprocess P from S -> list  //remove from list
    else
        wakeup(P)  //start process
}
```

**Explanation:**

In the given Semaphores code, a count variable is used, which counts from 0 to 100, and a connection is used, that adds is value and increments by one.

- In the next line, a connection is used, that uses the code to decrements its value.
- In the last step, a code value that is equal to 100, and other connections did not enable to decrements to 99.