

Lab-Report

Report No:

Course code: ICT-3110

Course title: Operating Systems Lab

Date of Performance:

Date of Submission:

Submitted by

Name: Hasibul Islam Imon

ID:IT-18047

3th year 1st semester

Session: 2017-18

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Lab report name: Threads on Operating System

Objectives:

1. A process with multiple threads make a great server for example printer server.
2. Because threads can share common data, they do not need to use interprocess communication.
3. Because of the very nature, threads can take advantage of multiprocessors.

Question no.1. What is Thread?

Answer: A thread otherwise called a lightweight process (LWP) is a basic unit of CPU utilization, it comprises of a thread id, a program counter, a register set and a stack. It shares with other threads belonging to the same process its code section, data section, and operating system resources such as open files and signals.

Question no.2. Types of Threads

Answer: There are two types of threads...

User threads: User threads are supported above the kernel and are implemented by a thread library

at the user level. Thread creation & scheduling are done in the user space, without kernel

intervention. Therefore they are fast to create and manage blocking system call will cause

the entire process to block.

Kernel threads: Kernel threads are supported directly by the operating system .Thread creation,

scheduling and management are done by the operating system. Therefore they are slower to

create & manage compared to user threads. If the thread performs a blocking system call,

the kernel can schedule another thread in the application for execution.

Question no:3 Implementation of Threads.

Kernel level threads are supported and managed directly by the operating system.

- 1.The kernel knows about and manages all threads.
- 2.One process control block (PCP) per process.
- 3.One thread control block (TCB) per thread in the system.
- 4.Provide system calls to create and manage threads from user space.

Conclusion:

In many respect, threads are popular way to improve application through parallelism. The CPU switches rapidly back and forth among the threads giving illusion that the threads are running in parallel. Since thread will generally call different procedures and thus a different execution history. This is why thread needs its own stack. An operating system that has thread facility, the basic unit of CPU utilization is a thread.