Mawlana Bhashani Science and Technology University

Lab-Report

Report No : 03

Experiment name : Threads on Operating System

Course code : ICT-3110

Course title : Operating System Lab.

Date of Performance :

Date of Submission : September 10,2020.

Submitted by

Name: Md Sohanur

ID: IT- IT-18011

3rd year 1st semester

Session: 2017- 18

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

i)What is Thread?

Answer: A thread is a path of execution within a process. A process can contain multiple threads. With respect to computer programming, a thread is a small set of instructions designed to be scheduled and executed by the cpu independently of the parent process. A thread is a single sequential flow of control within a program.

ii)Types of Threads.

Answer: There are two types of threads. These are given below:

i. User level thread.
ii. Kernel level thread.

User level thread (ULT): ULT implemented in the user level library, they are not created using the system calls. Thread switching does not need to call system OS and to cause interrupt to kernel. Kernel does not know about the user level thread and manages them as if they were single-threaded processes.

Advantages of ULT are given below:

- Can be implemented on that OS that doesn't support multiple threading.
- Simple representation since thread has only program counter, register set stack space.
- Simple to create since no interventional of kernel.
- Thread switching is fast since no OS calls need to be made.

Disadvantages of ULT are given below:

- Less co-ordination among the threads and kernel.
- If one thread causes a page fault, the entire process blocks.

Kernel level thread (KLT): Kernel knows and manages the threads. Instead of thread table in each process, the kernel itself has thread table that keeps track of all threads in the system. OS kernel provides system call to create and manage threads.

Advantages of KLT are given below:

- Since kernel has full knowledge about the threads in the system, scheduler may decide to give more time to processes having large number of threads.
- Good for application that frequently block.

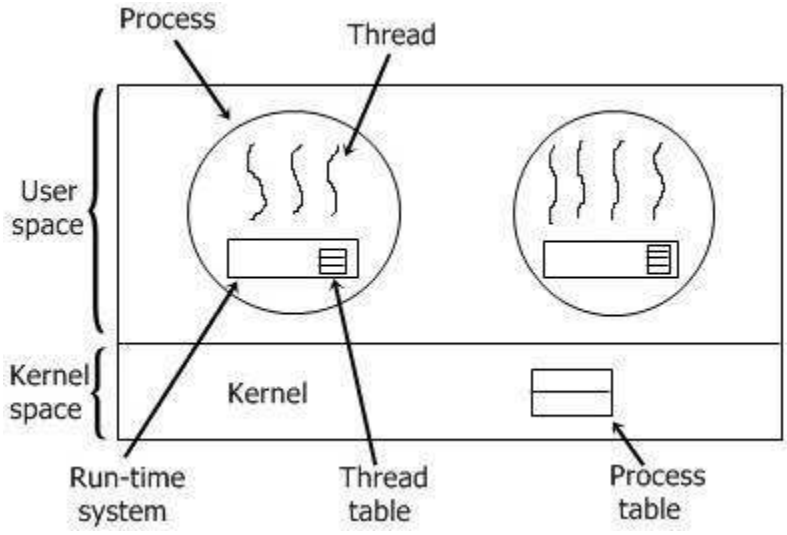
Disadvantages of ULT are given below:

- Slow and inefficient.
- It requires thread control block so it is an overhead.

iii) Implementation of threads.

Answer:

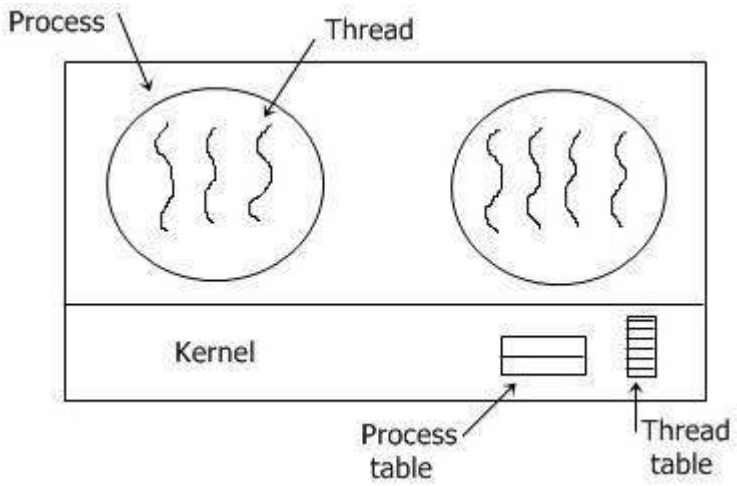
Threads Implementation in User Space: In this model of implementing the threads package completely in user space, the kernel don't know anything about them. The advantage of implementing threads package in user space is that a user-level threads package can be implemented on an OS (Operating System) that doesn't support threads. All of these implementations have the same general structure as illustrated in the figure given below.



Threads Implementation in Kernel : In this, there is no any thread table in each process. But to keep track of all the threads in the system, the kernel has the thread table. In this method of implementation model, the threads package completely in

the kernel. There is no need for any runtime system. To maintain the record of all threads in the system a kernel has a thread table.A call to the kernel is made whenever there is a need to create a new thread or destroy an existing thread. In this, the kernel thread table is updated.

In this method of implementing the threads package entirely in the kernel, no any run-time system is need in each as illustrated in the figure given below.



Now, let's discuss briefly about another two method given here.

- Hybrid implementation
- Scheduler activation

Hybrid Implementation: In this method, each kernel-level thread has some set of user-level threads that take turns using it.

Scheduler Activation: The goal of this scheduler activation work are to mimic the functionality of kernel threads, but with better performance and greater flexibility generally associated with threads packages implemented in user space.

Conclusion: From this lab we come to learn the concept of thread and its types. Finally we learn about the implementation of threads. A thread shares with its peer threads few information like code segment, data segment and open files. When one thread alters a code segment memory item, all other threads see that. User level threads are threads that are visible to the programmer and are unknown to the kernel.