Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 03

Course code: ICT- ICT-3110

Course title: Operating Systems Lab

Date of Performance:15-09-20

Date of Submission:27-09-20

Submitted by

Name: Nusrat Jahan Jui

ID:IT-18039

3rd year 1stsemester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment no : 03

Experiment Name : Threads on Operating System.

Theory:

A thread is a path of execution within a process. A process can contain multiple threads. A thread is a basic unit of CPU utilization that shares with other threads belonging to the same process it's code section,data section and other operating system resources such as open files and signals.

Threads are divided into parts.

1. User Threads.
2. Kernel Threads

User Threads: It is implemented in the user level library, they are not created using the system calls. This threads support above the kernel and managed without kernel support. This type of thread library contains code for creating and destroying threads, for passing message and data between threads, for scheduling thread execution and for saving and restoring thread contexts. The main applicationin this threads starts with a single thread.

Advantages:

- Can be implemented on an OS that does't support multithreading.
- Context switch time is less.
- Simple to create since no intervention of kernel.

Disadvantages:

- No or less co-ordination among the threads and Kernel.
- If one thread causes a page fault, the entire process blocks.

Kernel Threads: Kernel knows and manages the threads. Instead of thread table in each process, the kernel itself has thread table (a master one) that keeps track of all the threads in the system. In addition kernel also maintains the traditional process table to keep track of the processes. OS kernel provides system call to create and manage threads.

Advantages:

- Kernel can simultaneously schedule multiple threads .
- 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 applications that frequently block.

Disadvantages:

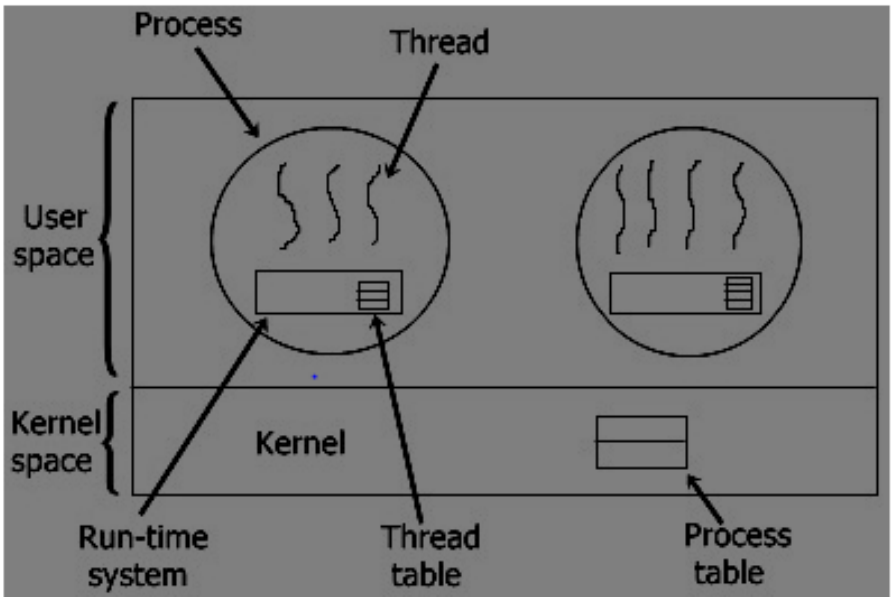
- Slow and inefficient.
- It requires thread control block so it is an overhead.

Implementation:

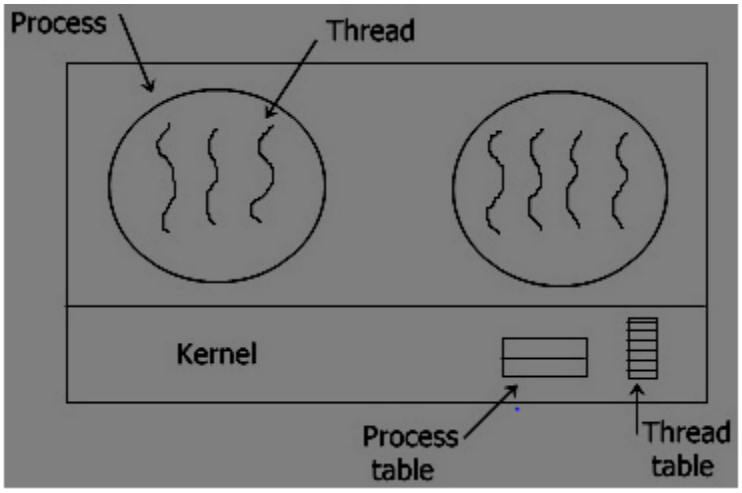
There are two ways of implementing a thread package:

1. In user space
2. In kernel

Threads implementation in the user space: In this model of implementation, the threads package entirely in user space, the kernel has no idea about it. A user-level threads package can be executed on an operating system that doesn't support threads and this is the main advantage of this implementation model i.e. Threads package in user space.



Threads implementation in the kernel: 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.



Other two methods are as follows:

1. Hybrid implementation
2. Scheduler activation

Hybrid implementation: In this implementation, there is some set of user-level threads for each kernel level thread that takes turns by using it.

Scheduler activation: The objective of this scheduler activation work is to replicate the working or function of kernel threads, but with higher performance and better flexibility which are usually related to threads packages which are implemented in userspace.

Discussion :

I have learned a lot doing this lab like the basic concept of thread, thread types, how it can be implemented and also how it works in operating system by using kernel. The main benefit of using thread is that we can do multiple task by dividing a process into multiple threads.Threads are used in case of multiple applications running at the same particular time few activities might block from one point of time to another.