



Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 03

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission:

Submitted by

Name: Rafiul Hasan

ID: IT-18016

3rd year 1st semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment no : 03

Experiment Name: Thread on Operating Systems.

Objective:-To introduce the notion of a thread- a fundamental unit of CPU utilization that forms the basis of multithreaded computer systems. To discuss the APIs for the Pthreads, Windows and java thread libraries. To cover operating system support for threads in Windows and Linux.

Thread:-A thread is the smallest unit of processing that can be performed in an OS. In most modern operating systems, a thread exists within a process – that is, a single process may contain multiple threads.

- A thread is a stream of execution throughout the process code having its program counter which keeps track of lists of instruction to execute next, system registers which bind its current working variables. Threads are also termed as lightweight process. Each thread has different states. Each thread has

1. A program counter
2. A register set
3. A stack space

Threads are independent of each other as they share the code, data, OS resources etc.

Types Of Threads:-Threads are implemented in following two ways.

1. User Level Threads- User managed threads
2. Kernel Level Threads- Operating System managed threads acting on kernel, an operating system core.
3. **User Level thread (ULT)** –User Level Thread is implemented in the user level library, they are not created using the system calls. Thread switching does not need to call OS and to cause interrupt to Kernel. Kernel doesn't know about the user level thread and manages them as if they were single-threaded processes.

Advantages of ULT –

- Can be implemented on an OS that does not support multithreading.
- Simple representation since thread has only program counter, register set, stack space.
- Simple to create since no intervention of kernel.
- Thread switching is fast since no OS calls need to be made.

Disadvantages of ULT –

- No or 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 (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 of KLT –

- 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 of KLT –

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

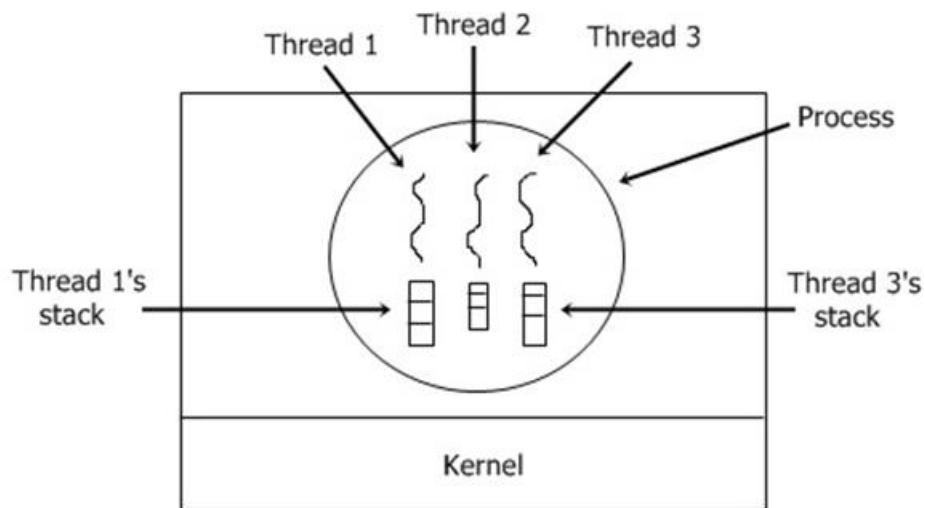
Implementation of Threads:-The new element can only be added with threads. This ability to share the same address space and data is essential for some applications. There are no resources attached to threads. Processes are difficult to create and destroy but threads, on the other hand, can be easily created and destroyed. Creating a thread is about 100x faster than creating a process. The thread has program counter(pc) to keep the track of the instruction to be executed next. It also has registers to hold the presently working variables. There is a stack to store the execution history there is one frame for one procedure called but not still returned from.

Threads are scheduled for the implementation or execution on CPU.

There are four states of a thread:

1. Running
2. Blocked
3. Read
4. Terminated

The stack of each thread is as follows:



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. 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.

Other two methods are as follows:

- Hybrid implementation
- Scheduler activation

Hybrid implementation

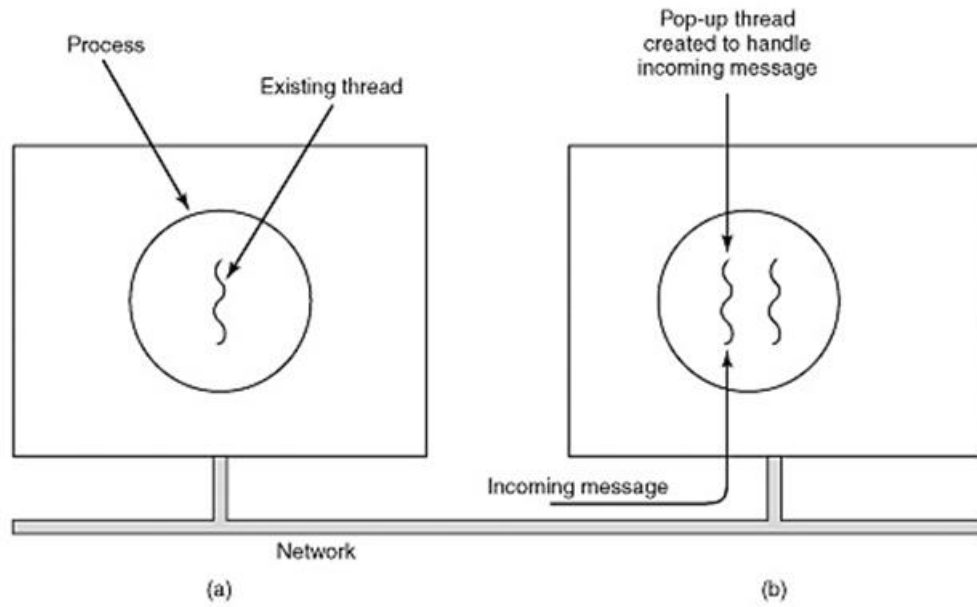
- In this implementation, there is some set of user-level threads for each kernel level thread that takes turns by using it.
-

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.

Pop-up threads

- In this system, a new thread is created just to handle the message as soon as the arrival of a message takes place and thus it is called pop up thread.



Conclusion:-An operating system is needed in order to use various applications on your computer and wireless devices. Without iOS and its continual upgrades there would be no way to operate today's trend setting wireless Apple devices.