Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 03

Course code: ICT-3110

Course title: Operating Systems Lab

Date of Performance:

Date of Submission:15/09/2020

Submitted by

Name: Md Ahadul Haque

ID:IT-18045

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 : Threads on Operating System.

Theory :

Thread is a single sequence stream within a process. Threads have same properties as of the process so they are called as light weight processes. Threads are executed one after another but gives the illusion as if they are executing in parallel. Each thread has different states. Each thread has

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

Types of Threads:

User Level thread (ULT) –
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.

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.

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.

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:

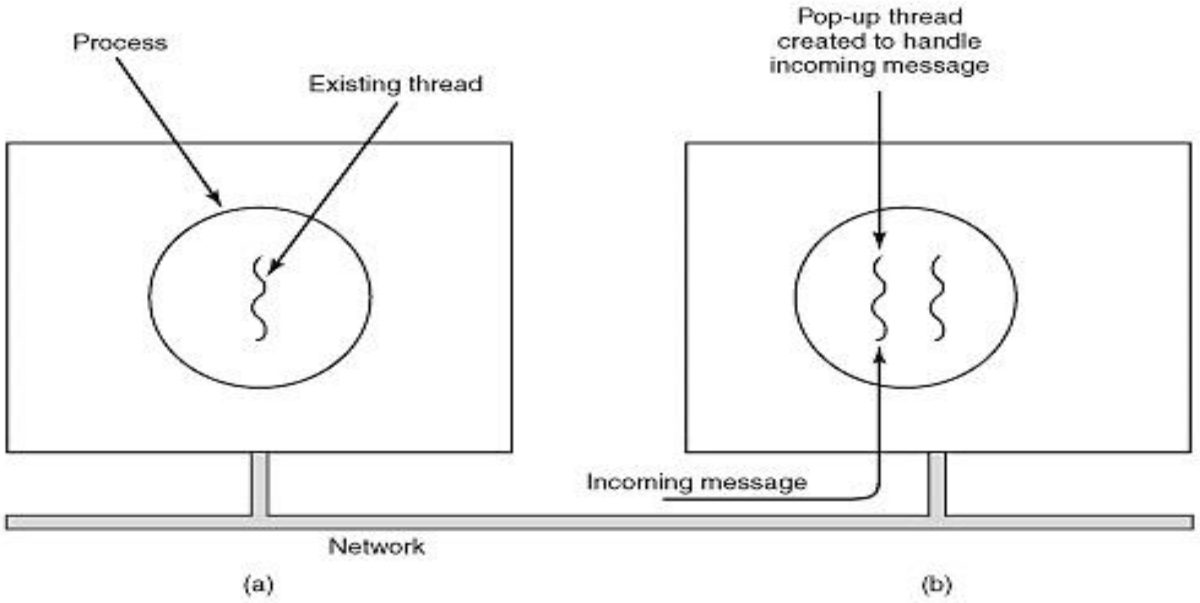
- Hybrid implementation
- Scheduler activation

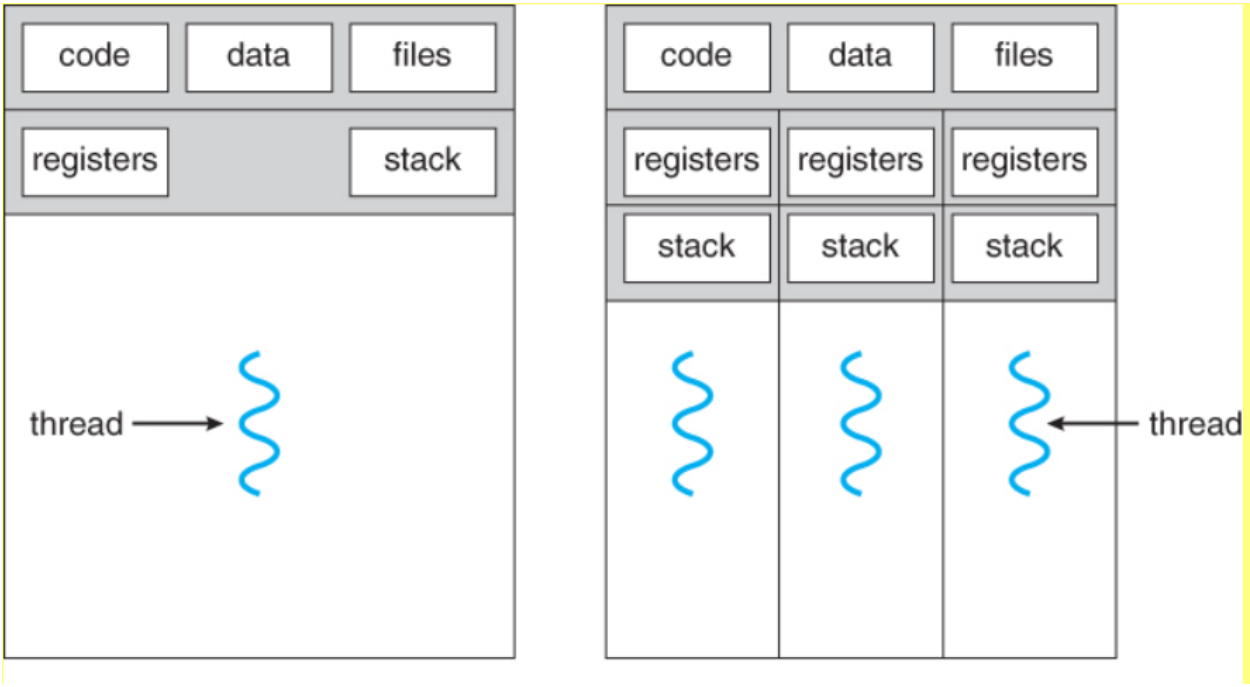
Hybrid implementation

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.


Process
Existing thread
Pop-up thread
created to handle
incoming message
Incoming message
Network
(a)
(b)


code
data
files
registers
stack
thread
code
data
files
registers
registers
registers
stack
stack
stack
thread

Discussion :

Thread are very useful in modern programming whenever a process has multiple task to perform independently from others.This is particularly true when one of the tasks may block ,and it is desired to allow the other tasks to proceed without blocking.By default threads share common code,data,and others resources,which allows multiple tasks to be performed simultaneously in a single address space.