LAB 03 - How to view threads of a process on linux and thread program .

Objectives:

i. What is Thread.
ii. Types of Thread
iii. Implementation of Thread

Theory : A thread is a flow of execution through the process code, with its own program counter that keeps track of which instruction to execute next, system registers which hold its current working variables, and a stack which contains the execution history.

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.

A thread is also called a **lightweight process**. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process.

What are the differences between process and thread?

Threads are not independent of one other like processes as a result threads shares with other threads their code section, data section and OS resources like open files and signals. But, like process, a thread has its own program counter (PC), a register set, and a stack space.

Types of Thread:

Threads are implemented in following two ways –

- **User Level Threads** – User managed threads.
- **Kernel Level Threads** – Operating System managed threads acting on kernel, an operating system core.

Multithreading Models:

Some operating system provide a combined user level thread and Kernel level thread facility. Solaris is a good example of this combined approach. In a combined system, multiple threads within the same application can run in parallel on multiple processors and a blocking system call need not block the entire process. Multithreading models are three types

- Many to many relationship.
- Many to one relationship.
- One to one relationship.

Corresponding Code:

```
#include<stdio.h>
#include<string.h>
#include<pthread.h>
#include<stdlib.h>
#include<unistd.h>

pthread_t tid[2];

void* doSomeThing(void *arg)
{
   unsigned long i = 0;
   pthread_t id = pthread_self();

   if(pthread_equal(id,tid[0]))
   {
      printf("\n First thread processing\n");
   }
   else
   {
      printf("\n Second thread processing\n");
   }

   for(i=0; i<(0xFFFFFFFF);i++);

   return NULL;
}

int main(void)
{
   int i = 0;
   int err;

   while(i < 2)
   {
      err = pthread_create(&(tid[i]), NULL, &doSomeThing, NULL);
      if (err != 0)
         printf("\ncan't create thread :[%s]", strerror(err));
      else
         printf("\n Thread created successfully\n");

      i++;
   }

   sleep(5);
   return 0;
}
```

Output:

```
tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx: ~/CodePractice/C_programming
File  Edit  View  Search  Terminal  Help
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$ gcc OS_multithreading.c -lpthread
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$ ./a.out

 Thread created successfully

 First thread processing

 Thread created successfully

 Second thread processing
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$
```

Thread in command line:

Here are several ways to show threads for a process on Linux.

1: PS

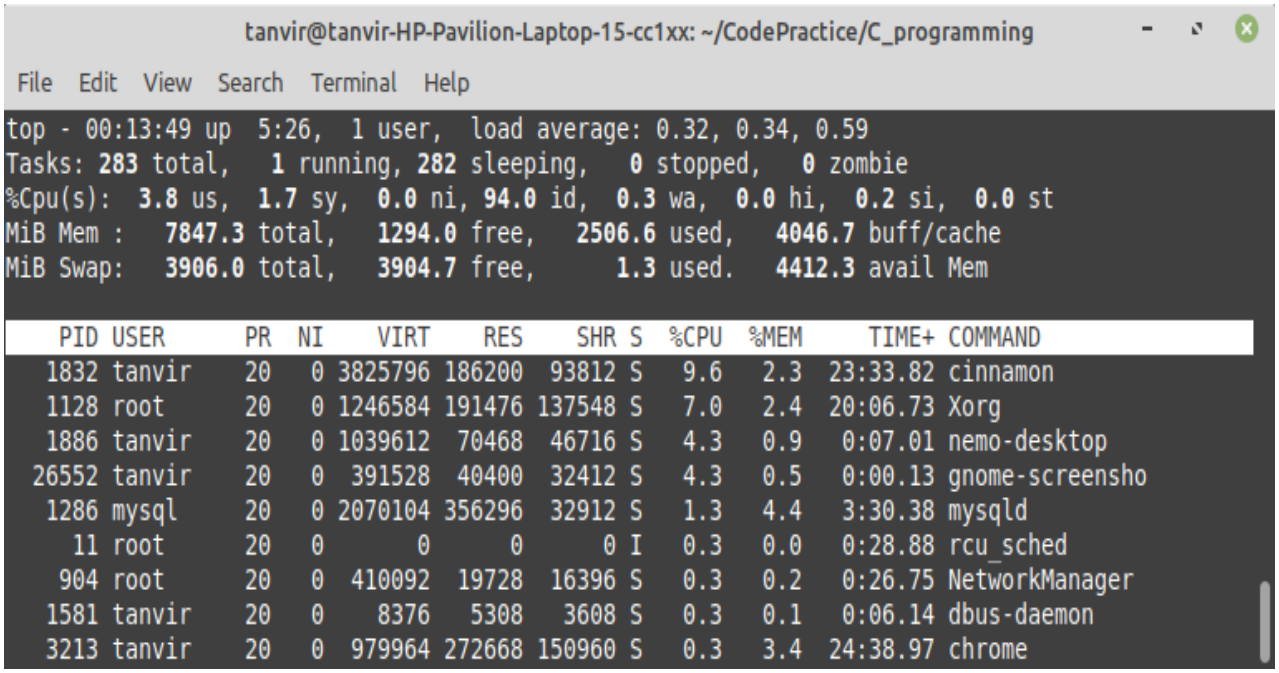
In ps command, "-T" option enables thread views. The following command list all threads created by a process with <pid>

```
tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx: ~
File  Edit  View  Search  Terminal  Help
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~$ ps -T -p 1128
    PID    SPID TTY          TIME CMD
   1128    1128 tty7     00:19:02 Xorg
   1128    1287 tty7     00:00:00 Xorg:disk$0
   1128    1288 tty7     00:00:00 Xorg:disk$1
   1128    1289 tty7     00:00:00 Xorg:disk$2
   1128    1290 tty7     00:00:00 Xorg:disk$3
   1128    1328 tty7     00:00:00 Xorg:disk$0
   1128    1329 tty7     00:00:00 Xorg:disk$1
   1128    1330 tty7     00:00:00 Xorg:disk$2
   1128    1331 tty7     00:00:00 Xorg:disk$3
   1128    1359 tty7     00:01:15 InputThread
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~$
```

The "SID" column represents thread IDs, and "CMD" column shows thread names.

2: Top:

The top command can show a real-time view of individual threads. To enable thread views in the top output, invoke top with "-H" option. This will list all Linux threads.



To restrict the top output to a particular process <pid> and check all threads running inside the process: then we use $ top -H -p <pid>

3: Htop:

A more user-friendly way to view threads per process is via htop, an ncurses-based interactive process viewer. This program allows you to monitor individual threads in tree views.