# Experiment No. 5

## Aim: Study Pthreads and implement the following : write a program which shows a performance.

**Theory:**

#### POSIX Threads in OS :

The POSIX thread libraries are a C/C++ thread API based on standards. It enables the creation of a new concurrent process flow. It works well on multi-processor or multi-core systems, where the process flow may be scheduled to execute on another processor, increasing speed through parallel or distributed processing. Because the system does not create a new system, virtual memory space and environment for the process, threads needless overhead than “forking” or creating a new process. While multiprocessor systems are the most effective, benefits can also be obtained on uniprocessor systems that leverage delay in I/O and other system processes that may impede process execution.

To utilize the PThread interfaces, we must include the header pthread.h at the start of the C script.

#include <pthread.h>

PThreads is a highly concrete multithreading system that is the UNIX system’s default standard. PThreads is an abbreviation for POSIX threads, and POSIX is an abbreviation for Portable Operating System Interface, which is a type of interface that the operating system must implement. PThreads in POSIX outline the threading APIs that the operating system must provide.

Pthread program: #include < pthread.h > #include < stdio.h > #include < stdlib.h >

#define NUM\_THREADS 5 void \*PrintHello(void \*threadid)

{

long tid;

tid = (long)threadid;

printf("Hello World! It's me, thread #%ld!\n", tid); pthread\_exit(NULL);

}

int main(int argc, char \*argv[])

{

pthread\_t threads[NUM\_THREADS]; int rc;

long t;

for(t=0;t< NUM\_THREADS;t++){

printf("In main: creating thread %ld\n", t);

rc = pthread\_create(&threads[t], NULL, PrintHello, (void \*)t); if (rc){

printf("ERROR; return code from pthread\_create() is %d\n", rc); exit(-1);

}

}

/\* Last thing that main() should do \*/ pthread\_exit(NULL);

}

OUTPUT :

Compile & run(linux terminal):

gcc -o Pthread\_Hello\_world pthread\_hello\_world.c -lpthread

./Pthread\_Hello\_world

OUTPUT:

In main: creating thread 0 In main: creating thread 1

Hello World! It's me, thread #0! In main: creating thread 2

Hello World! It's me, thread #1! In main: creating thread 3

Hello World! It's me, thread #2! In main: creating thread 4

Hello World! It's me, thread #3! Hello World! It's me, thread #4!

On terminal you can use the Linux ps command provides several flags for viewing thread information.

Example : - ps –Lf

Linux clusters also provide the top command to monitor processes on a node. If used with the -H flag, the threads contained within a process will be visible.

Example : - top –H