**Experiment No: 05**

**Experiment Name:** Thread in Linux.

**Aims and Objectives:** We will focus on how a thread is created and identified. We will also represent an example that will explain how to do basic threaded programming.

**Thread Creation:**

Normally when a program starts up and becomes a process, it starts with a default thread. So we can say that every process has at least one thread of control.  A process can create extra threads using the following function:

**Source Code:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

void \*print\_message\_function( void \*ptr );

main()

{

pthread\_t thread1, thread2;

const char \*message1 = "Thread 1";

const char \*message2 = "Thread 2";

int iret1, iret2;

iret1 = pthread\_create( &thread1, NULL, print\_message\_function, (void\*) message1);

if(iret1)

{

fprintf(stderr,"Error - pthread\_create() return code: %d\n",iret1);

exit(EXIT\_FAILURE);

}

iret2 = pthread\_create( &thread2, NULL, print\_message\_function, (void\*) message2);

if(iret2)

{

fprintf(stderr,"Error - pthread\_create() return code: %d\n",iret2);

exit(EXIT\_FAILURE);

}

printf("pthread\_create() for thread 1 returns: %d\n",iret1);

printf("pthread\_create() for thread 2 returns: %d\n",iret2);

pthread\_join( thread1, NULL);

pthread\_join( thread2, NULL);

exit(EXIT\_SUCCESS);

}

void \*print\_message\_function( void \*ptr )

{

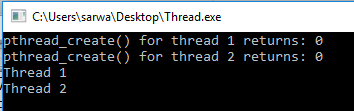
char \*message;

message = (char \*) ptr;

printf("%s \n", message);

}

**Output:**

****

**Conclusion:**

We have learnt how a thread is created and identified. We have also leant example that explain how to do basic threaded programming.

.