**Lecture 5**

**POSIX Thread:**

In Linux platforms, the C language contains **pthread** standard API for all kinds of thread related functions. It is also known as a POSIX thread that allows users to create many threads for simultaneous processes to flow. It is best used in multi-core systems or processors to implement threads on the kernel to achieve the system.

**Implementation:**

**(Library: #include <pthread.h>)**

It is necessary to include this pthread.h header file in the script initially. This will help in using the functions of the pthread library. To compile and execute the c file consisting of pthread library following commands should be used. (Assume the name of the c file is th.c)

**gcc -o th th.c -lpthread**

**./th**

**Functions of pthread:**

1. **int pthread\_create(pthread\_t \* thread\_id, const pthread\_attributes\_t \* attr, void \* (\*thread\_function), void \*argument);**

**Purpose:** It is used to create a new thread.

**Parameters:**

* **thread\_id:** This acts as a pointer to an unsigned integer value. It returns the thread id of the thread that is formed.
* **attributes:** This parameter acts as a pointer to a structure. It is used to define attributes of a thread that can be the policy of scheduling, and stack address, etc.
* **thread\_function:** This parameter is a pointer to the thread function of a thread.
* **argument:** This parameter is a pointer to void with different arguments to the function pre-defined at the start of the argument.

1. **void pthread\_exit(void \*return\_value);**

**Purpose:** It is used to terminate or end a thread.

**Parameters:**

* **return\_value:** It stores the status of the thread such that the thread terminates. It must be a global variable. This will allow the next thread in line to join the thread if it is available.

1. **int pthread\_join(pthread\_t thread\_id, void \*thread\_return);**

**Purpose:** This is a function used at the time of wait for the termination of the thread.

**Parameters:**

* **thread\_id:** It is the ID of the thread for which the thread in line waits.
* **thread\_return:** It is the parameter that acts as a pointer to the particular location where we have defined the exit status.