Assignment 05

→ **Aim** : Matrix Multiplication using POSIX threads

→ **Objectives** :

To implement C program for Matrix Multiplication using POSIX threads

→ **Theory** :

A] What is thread ?

A thread is simply a semi-process, that has its own stack and executes a given piece of code. The thread normally shares its memory with other threads. A Thread Group is a set of threads all executing inside the same process. They all share the same memory, same heap memory, same set of file descriptors, etc. All these threads execute in parallel.

B] What are pthreads ?

Hardware vendors have implemented their own proprietary versions of threads. The primary motivation for using pthread is to realize potential program performance gains.

- When compared to the cost of creating & managing a process, a thread can be created with much less operating system overhead.
- Inter thread communication is more efficient and easier than IPCs.
- Advantages:

1. Overlapping CPU work with I/O: While one thread is waiting for I/O call to complete, other threads can perform CPU intensive work.
2. Priority / Real time scheduling: Important tasks can be scheduled to supersede or interrupt lower priority tasks.
3. Asynchronous event handling: tasks can be interleaved.

→ The pthreads API:

1. Thread management: The first class of functions work directly on threads - creating, detaching, joining, etc. They include functions to set/query thread attributes.

2. Mutexes: The second class of functions that deal with synchronization. They provide functions for creating, destroying and unlocking mutexes.

3. Condition Variables: This deals with a finer class of synchronization - based on programmer specified conditions.

→ Thread Creation & ~~Destruction~~ Termination

The function pthread_create is used to create a new thread & a ~~thr~~ thread to terminate itself using pthread_exit. A thread to wait for termination of another thread uses the function pthread_join.

1. int pthread_create (pthread_t *threadhandle,
   pthread_attr_t * attribute
   void* (*start_routine)(void*), void *arg);

2. void pthread_exit (void * rtvalue);

3. int pthread_join (pthread_t handle,
   void ** returnvalue);

→ Thread Initialization:

Initially threads are created from within a process. Once created, threads are peers & may create new threads. An "initial thread" is created by default which runs main.

→ Terminating thread execution:

- int pthread_cancel (pthread_t thread):

A cancel is a mechanism by which the thread informs either itself or other thread to terminate. The cancelled can delay processing the cancel after receiving it.

- pthread_exit:

The thread receives a signal that terminates the entire process.

→ int pthread_attr_destroy (pthread_attr_t *attr)

It destroys the attribute object pointed to by attr releasing any resources ~~point~~ associated with it.

→ Conclusion :

Topics covered :

1. Threads
2. Mutex.
3. Functions) calls related to threads.