**MULTI THREADING AND INTERTHREAD COMMUNICATION**

**Process**

A process is the fundamental unit of execution in a computer's operating system, and it represents the dynamic execution context of a running program.

**Threads**

A thread is the smallest unit of execution within a process.

**Multithreading**

In a multithreaded process, multiple threads exist within the same process and share the same resources, such as memory space and file descriptors.

**pthread**

The POSIX threads library defines a set of functions and data types for creating, manipulating, and synchronizing threads.

**Creating a Pthread**

1. **pthread\_create Function:**

The pthread\_create function is the primary method for creating threads in C using the pthread library. It takes several parameters, including a pointer to the thread ID, thread attributes, a start routine (the function to be executed by the thread), and optional arguments to be passed to the start routine.

**Syntax:**

int pthread\_create(pthread\_t \*thread, const pthread\_attr\_t \*attr,

                   void \*(\*start\_routine) (void \*), void \*arg);

**\*thread (pthread\_t thread):**

This is a pointer to a pthread\_t variable, which will hold the ID of the newly created thread.

**\*attr (const pthread\_attr\_t attr):**

A pointer to a pthread\_attr\_t structure that specifies the attributes for the thread. If you pass NULL, the thread is created with default attributes.

**\*\*start\_routine (void \*(start\_routine) (void )):**

A pointer to the function that the thread will execute once it is created. This function must return a void \* and take a single void \* argument.

\*arg (void arg):

**Important**

A pointer to the argument that will be passed to the start\_routine. If no argument is needed, NULL can be passed.

**Thread Attributes**

* Stack Size
* Scheduling Policy
* Scheduling Priority
* Detached State
* Inheritance of Scheduling Attributes
* Scope

1. **Thread Attributes:**

The pthread\_attr\_init function is used to initialize a thread attribute object, and this object can be passed to pthread\_create to specify various attributes for the created thread, such as stack size, scheduling policy, etc.

1. **Thread Detach**

Threads can be created in a detached state using the pthread\_detach function. A detached thread automatically releases its resources when it completes without requiring an explicit call to pthread\_join.

#include <pthread.h>

#include <stdio.h>

// Function to be executed by the first thread

void \*thread\_function1(void \*arg) {

    printf("Thread 1 is running\n");

    // Thread 1 code here

    return NULL;

}

// Function to be executed by the second thread

void \*thread\_function2(void \*arg) {

    printf("Thread 2 is running\n");

    // Thread 2 code here

    return NULL;

}

int main() {

    pthread\_t thread\_id1, thread\_id2;

    // Create the first thread

    int result1 = pthread\_create(&thread\_id1, NULL, thread\_function1, NULL);

    if (result1 != 0) {

        perror("Thread 1 creation failed");

        return 1;

    }

    // Create the second thread

    int result2 = pthread\_create(&thread\_id2, NULL, thread\_function2, NULL);

    if (result2 != 0) {

        perror("Thread 2 creation failed");

        return 1;

    }

    // Main thread continues execution

    // Wait for the first thread to finish

    pthread\_join(thread\_id1, NULL);

    printf("Thread 1 has finished\n");

    // Wait for the second thread to finish

    pthread\_join(thread\_id2, NULL);

    printf("Thread 2 has finished\n");

    return 0;

}

**Inter Thread Communication**

Inter-thread communication is the process of passing data between threads in a way that allows synchronization and coordination between them.

**1)Mutex and Condition Variables:**

* Mutex (short for mutual exclusion) is used to protect shared resources, ensuring that only one thread can access the critical section at a time.
* Condition variables allow threads to wait for a particular condition to be met before proceeding.

**2) Message Passing:**

Message passing involves sending messages between threads using a shared data structure or a message queue.