

Name: P.MALREDDY

Reg-No: 1923720015

11. Illustrate the concept of multithreading using a C program.

Aim:

To demonstrate the concept of multithreading in C by creating multiple threads that execute concurrently.

Algorithm:

1. **Start.**
2. Initialize the program and include the necessary libraries.
3. Define the functions that will be executed by the threads.
4. Create threads using the `pthread_create` function.
5. Execute the threads concurrently.
6. Use `pthread_join` to wait for threads to finish execution.
7. Print the results from each thread to demonstrate multithreading.
8. **End.**

Procedure:

1. Import `pthread.h` and `stdio.h` libraries.
2. Define the function for thread execution logic.
3. Use `pthread_create` to create multiple threads and pass the function as an argument.
4. Use `pthread_join` to ensure main program waits for all threads to finish.
5. Compile and run the program to observe concurrent thread execution.

Code:

```
#include <stdio.h>
```

```
#include <pthread.h>
```

```
#include <unistd.h>
```

```
void *print_message(void *thread_id) {
```

```
    int tid = *(int *)thread_id;
```

```
    printf("Thread %d is running\n", tid);
```

```
    sleep(1); // Simulate work
```

```

    printf("Thread %d has finished\n", tid);

    return NULL;
}

int main() {
    pthread_t threads[3];
    int thread_ids[3];

    for (int i = 0; i < 3; i++) {
        thread_ids[i] = i + 1;
        pthread_create(&threads[i], NULL, print_message, &thread_ids[i]);
    }

    for (int i = 0; i < 3; i++) {
        pthread_join(threads[i], NULL);
    }

    printf("All threads have completed execution.\n");

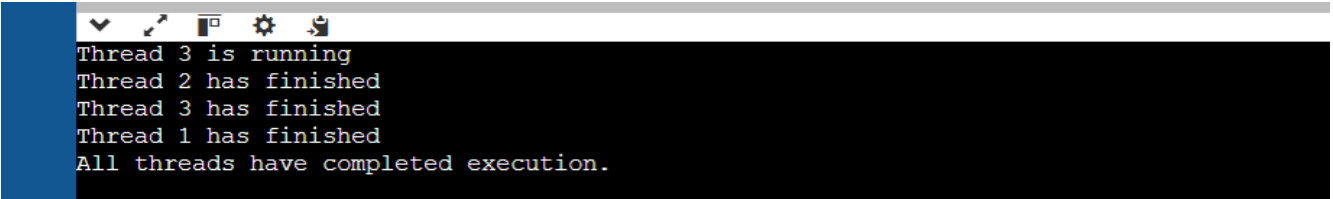
    return 0;
}

```

Result:

When executed, the program creates three threads. Each thread prints its start and end message, demonstrating concurrent execution

Output:

A screenshot of a terminal window with a dark background and a light blue sidebar on the left. The terminal displays the following output: Thread 3 is running, Thread 2 has finished, Thread 3 has finished, Thread 1 has finished, and All threads have completed execution. The window has a standard macOS-style title bar with icons for window control, zoom, and search.

```
Thread 3 is running
Thread 2 has finished
Thread 3 has finished
Thread 1 has finished
All threads have completed execution.
```