

D.GUNA VARDHAN - 192372041

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

Aim:

To illustrate the concept of multithreading in C, where multiple threads are executed concurrently.

Algorithm:

1. Initialize the main thread.
2. Create additional threads using `pthread_create()`.
3. Each thread executes a function.
4. The main thread waits for all threads to finish using `pthread_join()`.
5. The threads perform a task, and the main thread handles the synchronization.

Procedure:

1. Include the necessary header for pthreads: `<pthread.h>`.
2. Define a function that will be executed by each thread.
3. Use `pthread_create()` to create new threads.
4. Use `pthread_join()` to ensure the main thread waits for the other threads to finish.
5. Display a message from each thread and the main thread to show parallel execution.

Code:

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

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

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

```
    long tid = (long)thread_id;
```

```
    printf("Hello from thread %ld\n", tid);
```

```

        return NULL;
    }

int main() {
    pthread_t threads[3];
    long t;

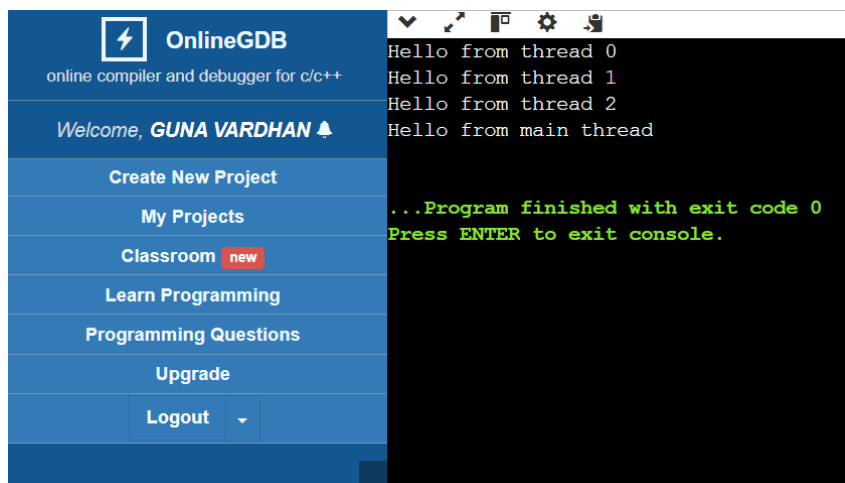
    for (t = 0; t < 3; t++) {
        pthread_create(&threads[t], NULL, print_message, (void*)t);
    }

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

    printf("Hello from main thread\n");
    return 0;
}

```

Output:



The screenshot shows the OnlineGDB web interface. On the left is a blue sidebar with the OnlineGDB logo and navigation links: 'Create New Project', 'My Projects', 'Classroom' (with a 'new' badge), 'Learn Programming', 'Programming Questions', 'Upgrade', and a 'Logout' button with a dropdown arrow. The main area on the right has a black background with white text showing the program's output: 'Hello from thread 0', 'Hello from thread 1', 'Hello from thread 2', and 'Hello from main thread'. Below this, green text indicates '...Program finished with exit code 0' and 'Press ENTER to exit console.'.