B.BHANUTEJA REDDY-192325016

30. Write C programs to demonstrate the following thread related concepts.
(i)create (ii) join (iii) equal (iv) exit

AIM

To demonstrate the thread-related concepts: **create**, **join**, **equal**, and **exit** using C programming.

ALGORITHM

1. **Start**
2. Include necessary libraries (pthread.h for threading).
3. Implement each concept as follows:
   - **Create:** Use pthread_create() to create a thread.
   - **Join:** Use pthread_join() to wait for a thread to complete execution.
   - **Equal:** Use pthread_equal() to check if two thread identifiers are equal.
   - **Exit:** Use pthread_exit() to terminate a thread.
4. Display appropriate messages to demonstrate the functionality of each operation.
5. Compile and run the program.

PROCEDURE

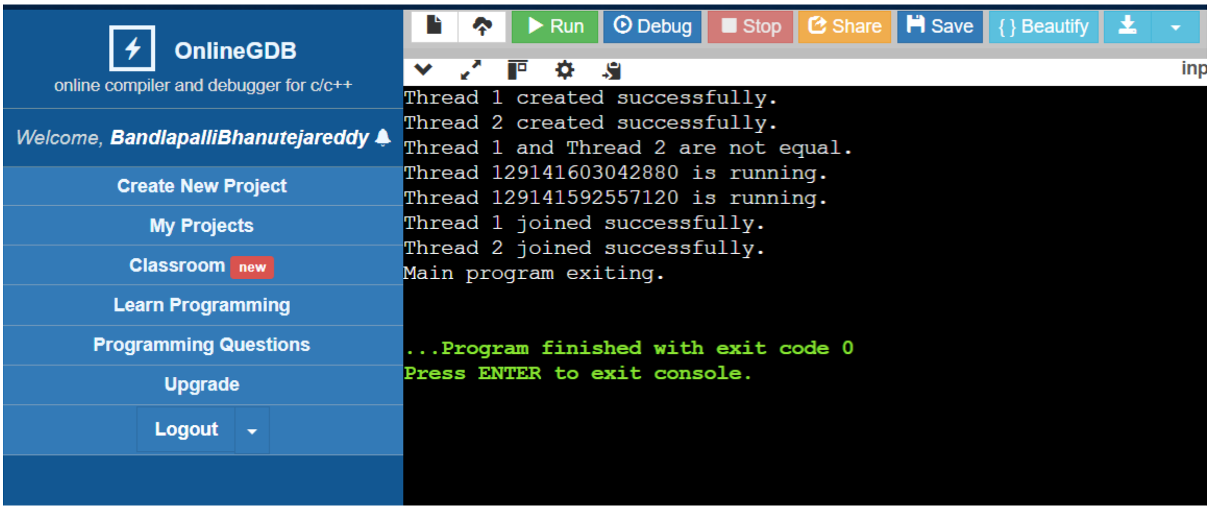
1. Include the pthread.h and stdio.h libraries for thread handling and input/output.
2. Define a thread function that executes specific operations and exits using pthread_exit().
3. Use pthread_create() to create threads.
4. Use pthread_equal() to compare thread identifiers and determine if two threads are the same.
5. Use pthread_join() to wait for threads to complete their execution.
6. Print messages to demonstrate the functionality of each operation.

CODE:

```
#include <stdio.h>

#include <pthread.h>
```

```
#include <unistd.h>

void *threadFunction(void *arg) {
    printf("Thread %ld is running.\n", pthread_self());
    pthread_exit(NULL);
}

int main() {
    pthread_t thread1, thread2;
    int status;

    status = pthread_create(&thread1, NULL, threadFunction, NULL);
    if (status == 0) {
        printf("Thread 1 created successfully.\n");
    }

    status = pthread_create(&thread2, NULL, threadFunction, NULL);
    if (status == 0) {
        printf("Thread 2 created successfully.\n");
    }

    if (pthread_equal(thread1, thread2)) {
        printf("Thread 1 and Thread 2 are equal.\n");
    } else {
        printf("Thread 1 and Thread 2 are not equal.\n");
    }

    pthread_join(thread1, NULL);
    printf("Thread 1 joined successfully.\n");
    pthread_join(thread2, NULL);
    printf("Thread 2 joined successfully.\n");

    printf("Main program exiting.\n");
    return 0;
}
```

OUTPUT:



```
Thread 1 created successfully.
Thread 2 created successfully.
Thread 1 and Thread 2 are not equal.
Thread 129141603042880 is running.
Thread 129141592557120 is running.
Thread 1 joined successfully.
Thread 2 joined successfully.
Main program exiting.


...Program finished with exit code 0
Press ENTER to exit console.
```