

Operating System Lab 14 Tasks

**Name:** Alishba Waqar

**Sap ID:** 46997

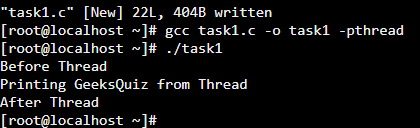
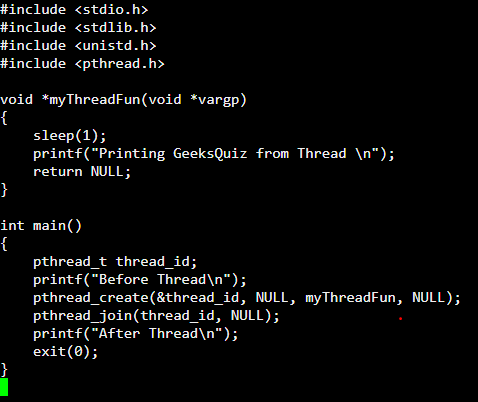
**Batch:** BSCS-5th

# Lab Instructor:

Kausar Nasreen Khattak

# Task 1:

**Type the following and execute it.**



**pthread\_create:**

This function creates a new thread that runs the function myThreadFun. The thread will sleep for 1 second and then print "Printing GeeksQuiz from Thread".

## pthread\_join:

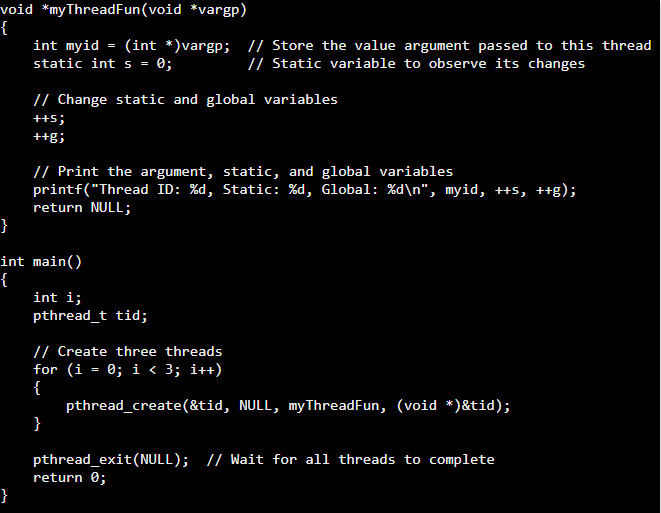
This waits for the newly created thread to finish its execution before moving on with the main function.

## Output:

The program first prints "Before Thread", then the thread runs and prints "Printing GeeksQuiz from Thread", and finally, the main function prints "After Thread".

# Task 2:

**Try to execute the following code:**



This code creates three threads that each modify static and global variables and print their values:

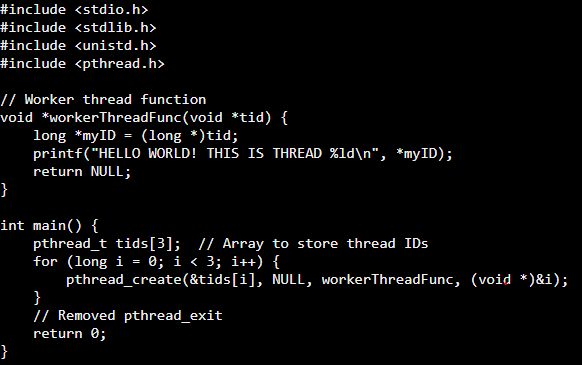
* Global variable (g) is modified by all threads.
* Static variable (s) is local to the thread but retains its value between function calls within that thread.
* Each thread prints its unique ID along with the updated values of the global and static variable

# Task 3:

**Try to execute the following code:**

**Modify pthread\_create and convert into for loop and create three threads and show output. Also removes pthread\_exit and see what happens.**

1. Header Files: The necessary libraries (stdio.h, stdlib.h, unistd.h, pthread.h) are included to handle input/output, memory allocation, threading, and other functions.
2. Worker Function: The workerThreadFunc prints a message identifying the thread's ID.
3. Main Function: A single thread is created using pthread\_create and runs the workerThreadFunc.



# Observation Without pthread\_exit:

* Without pthread\_exit, the main program may terminate before all threads complete, depending on how the OS schedules threads.
* With pthread\_exit, the main thread ensures that all threads complete before the program exits.

# Task 4:

**Define POSIX thread and it’s working in your own words.**

POSIX threads (pthreads) are a standard API for creating and managing threads in C/C++. They enable parallel execution of tasks within a process, utilizing shared memory and efficient communication mechanisms.

## Working of POSIX Threads:

A thread is created using pthread\_create.

Each thread executes independently but shares the memory and resources of the parent process.

Synchronization is achieved using mutexes, semaphores, or condition variables to avoid race conditions.

The main thread can wait for child threads to finish using pthread\_join.