

## Thread-safe Shared Counter and File Logger using Mutexes (Time:1 Hours)

CO3

### Objective

Implement a multithreaded C program that:

- Spawns multiple worker threads.
- Each thread updates a shared counter and writes to a shared log file.
- Uses `pthread_mutex_t` to protect both the counter and the file.
- Demonstrates what happens with and without mutex protection.

### Requirements

#### 1. Shared Counter

- Global integer counter = 0.
- Each worker thread increments the counter 10 000 times.

#### 2. Shared Log File

- All threads write a line to the same file when they finish their increments:  
"Thread X finished, counter = Y\n"

#### 3. Use a Mutex

- Protect the counter increment section.
- Protect file writes separately (or use one mutex for both).

#### 4. Show Race Condition

- Add a command-line flag `--no-mutex` so the program runs without locking and shows the wrong counter at the end.

#### 5. Number of Threads

- Pass as command-line argument (e.g., `./prog 5`).

#### 6. Output

- Print final counter value.
- Compare runs with/without mutex.