# LAB # 13

# Revision Lab

**Task # 01**:

Create a C program to implement thread synchronization, in which two threads, A and B, alternately print their names a certain number of times. Thread A should be the first to print its name, followed by Thread B, and so on. The goal is to use semaphores to synchronize the two threads and ensure the desired alternating behavior.

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#define NUM\_TIMES 5

sem\_t semA, semB;

void\* threadA(void\* arg) {

for (int i = 0; i < NUM\_TIMES; i++) {

sem\_wait(&semA); // Wait for permission to proceed

printf("Thread A\n");

sem\_post(&semB); // Signal Thread B to proceed

}

pthread\_exit(NULL);

}

void\* threadB(void\* arg) {

for (int i = 0; i < NUM\_TIMES; i++) {

sem\_wait(&semB); // Wait for permission to proceed

printf("Thread B\n");

sem\_post(&semA); // Signal Thread A to proceed

}

pthread\_exit(NULL);

}

int main() {

pthread\_t tidA, tidB;

// Initialize semaphores

sem\_init(&semA, 0, 1);

sem\_init(&semB, 0, 0);

// Create threads

pthread\_create(&tidA, NULL, threadA, NULL);

pthread\_create(&tidB, NULL, threadB, NULL);

// Wait for threads to finish

pthread\_join(tidA, NULL);

pthread\_join(tidB, NULL);

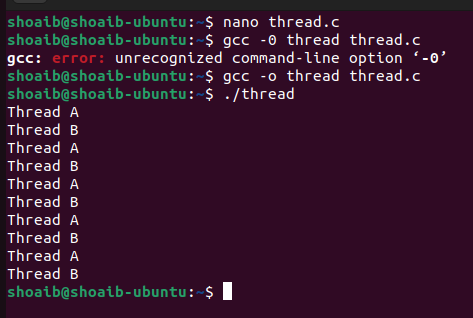
// Destroy semaphores

sem\_destroy(&semA);

sem\_destroy(&semB);

return 0;

}



**Task # 02**:

Create a shell script that accepts input for 5 students' grades and calculates them based on the following criteria:

"A" grade is defined as a score greater than or equal to 90.

"B" grade is defined as a score greater than or equal to 80 but less than 90.

"C" grade is one that is greater than or equal to 70 but less than 80.

"D" grade is earned if your score is greater than or equal to 60 but less than 70.

A score of less than 60 corresponds to a "F" grade.

The script should ask the user for the number of students and their grades before calculating and displaying the corresponding grades for each student.

