Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_13\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **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. |
| 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. |

Submitted On

15-06-2023

(Date: DD/MM/YY)

**Task 01:** Semaphore is one of the concurrency mechanisms available. Find out about more concurrency mechanisms. How do these mechanisms protect critical sections? Compare their implementations with *wait()* and *signal()* operations of semaphores.

**Solution:**

#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;

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

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

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

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

pthread\_join(tidA, NULL);

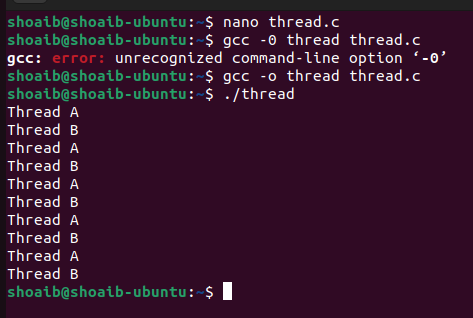
pthread\_join(tidB, NULL);

sem\_destroy(&semA);

sem\_destroy(&semB);

return 0;}

**Output:**



**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.

**Solution:**

#!/bin/bash

read -p "Enter the number of students: " num\_students

for ((i = 1; i <= num\_students; i++)); do

read -p "Enter the grade for student $i: " grade

if ((grade >= 90)); then

letter\_grade="A"

elif ((grade >= 80)); then

letter\_grade="B"

elif ((grade >= 70)); then

letter\_grade="C"

elif ((grade >= 60)); then

letter\_grade="D"

else

letter\_grade="F"

fi

echo "Student $i - Grade: $letter\_grade"

done

**Output:**

A screenshot of a computer program

Description automatically generated with low confidence