Operating System and System Programming Lab (15B17CI472)

Lab Test 2: Odd 2021 (04/12/2021)

Patil Amit Gurusidhappa

19104004

B11

Q1.

Code

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#include <string.h>

#define MAX\_NUM 1

sem\_t sem1, sem2, sem3, sem4;

int cuur = 0;

// Decleration of globle String

char globalString[10] = "";

void printFunction(int i)

{

    printf("Thread %d is executing\n", i);

    printf("Modiifed global string to -> %s\n",globalString);

}

void \*f1(void \*p)

{

    printf("%s", (char \*)p);

    for (int i = 1; i <= MAX\_NUM; i++)

    {

        sem\_wait(&sem1);

        strncat(p,"f1",1);

        printFunction(1);

        sem\_post(&sem2);

    }

}

void \*f2(void \*p)

{

    for (int i = 1; i <= MAX\_NUM; i++)

    {

        sem\_wait(&sem2);

        printFunction(2);

        strncat(p, "f2", 1);

        // printf("%s", strncat);

        sem\_post(&sem3);

    }

}

void \*f3(void \*p)

{

    for (int i = 1; i <= MAX\_NUM; i++)

    {

        sem\_wait(&sem3);

        printFunction(3);

        strncat(p, "f3", 1);

        // strcnt()

        sem\_post(&sem4);

    }

}

void \*f4(void \*p)

{

    for (int i = 1; i <= MAX\_NUM; i++)

    {

        sem\_wait(&sem4);

        printFunction(4);

        strncat(p, "f4", 1);

        sem\_post(&sem1);

    }

}

int main()

{

    pthread\_t p1, p2, p3, p4;

    sem\_init(&sem1, 0, 0);

    sem\_init(&sem2, 0, 0);

    sem\_init(&sem3, 0, 0);

    sem\_init(&sem4, 0, 1);

    pthread\_create(&p1, NULL, f1, globalString);

    pthread\_create(&p2, NULL, f2, globalString);

    pthread\_create(&p3, NULL, f3, globalString);

    pthread\_create(&p4, NULL, f4, globalString);

    pthread\_join(p1, NULL);

    pthread\_join(p2, NULL);

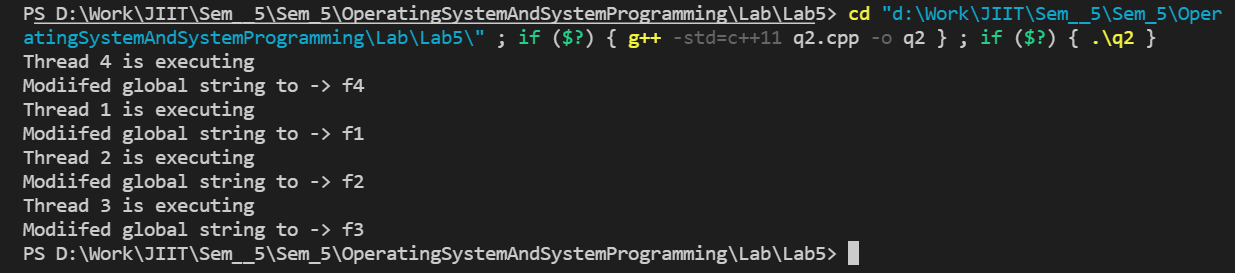
    pthread\_join(p3, NULL);

    pthread\_join(p4, NULL);

    return 0;

}

Output



Explanation

Global string is modified on sequential execution of the threads

All the threads all organized using Semaphores ->Wait and post methods