**OS LAB WEEK 4**

**ADIT LUHADIA**

**190911112**

**IT A**

**LAB 7 - CLASSICAL PROBLEMS OF SYNCHRONIZATION**

1. Write a C program to solve producer consumer problem with bounded buffer using semaphores.

// 190911112

// Adit Luhadia

#include <stdio.h>

#include <semaphore.h>

#include <pthread.h>

#include <unistd.h>

#include <sys/wait.h>

#include <signal.h>

#include <stdlib.h>

#define MAXCAPACITY 100

int buff[MAXCAPACITY];

sem\_t pmutex, cmutex, empty, full;

int in = 1, out = 1;

int item = 0;

void \*produce(void \*);

void \*consume(void \*);

int status = 0;

void \*produce(void \*arg)

{

    sem\_wait(&empty);

    item++;

    sem\_wait(&pmutex);

    buff[in] = item;

    printf("Produced data %d\n", buff[in]);

    in = in % MAXCAPACITY + 1;

    sem\_post(&pmutex);

    sem\_post(&full);

    return 0;

}

void \*consume(void \*arg)

{

    int citem = 0;

    sem\_wait(&full);

    sem\_wait(&cmutex);

    citem = buff[out];

    printf("Consumed Data = %d\t %d\n", buff[out], citem);

    out = out % MAXCAPACITY + 1;

    sem\_post(&cmutex);

    sem\_post(&empty);

    return 0;

}

int main(int argc, const char \*argv[])

{

    printf("190911112\nAdit Luhadia\n");

    in = 1, out = 1;

    int i, NumThreads;

    sem\_post(&pmutex);

    sem\_post(&cmutex);

    sem\_init(&full, 0, 0);

    sem\_init(&empty, 0, MAXCAPACITY);

    pthread\_t \*producers, \*consumers;

    NumThreads = 5;

    producers = (pthread\_t \*)malloc(sizeof(pthread\_t) \* NumThreads);

    consumers = (pthread\_t \*)malloc(sizeof(pthread\_t) \* NumThreads);

    for (i = 0; i < NumThreads; i++)

    {

        pthread\_create(&consumers[i], NULL, &consume, NULL);

        pthread\_create(&producers[i], NULL, &produce, NULL);

    }

    for (i = 0; i < NumThreads; i++)

    {

        pthread\_join(producers[i], NULL);

        pthread\_join(consumers[i], NULL);

    }

    return 0;

}

Text

Description automatically generated

2. Write a C program to solve the readers and writers Problem.

// 190911112

// Adit Luhadia

#include <stdio.h>

#include <semaphore.h>

#include <pthread.h>

#include <unistd.h>

#include <sys/wait.h>

#include <signal.h>

#include <stdlib.h>

sem\_t mutex, writeblock;

int data = 0, rcount = 0;

void \*reader(void \*arg)

{

    int f;

    f = ((int)arg);

    sem\_wait(&mutex);

    rcount = rcount + 1;

    if (rcount == 1)

        sem\_wait(&writeblock);

    sem\_post(&mutex);

    printf("Data read by the reader%d is %d\n", f, data);

    sleep(1);

    sem\_wait(&mutex);

    rcount = rcount - 1;

    if (rcount == 0)

        sem\_post(&writeblock);

    sem\_post(&mutex);

}

void \*writer(void \*arg)

{

    int f;

    f = ((int)arg);

    sem\_wait(&writeblock);

    data++;

    printf("Data writen by the writer%d is %d\n", f, data);

    sleep(1);

    sem\_post(&writeblock);

}

int main()

{

    printf("190911112\nAdit Luhadia\n");

    int i, b;

    pthread\_t rtid[5], wtid[5];

    sem\_init(&mutex, 0, 1);

    sem\_init(&writeblock, 0, 1);

    for (i = 0; i <= 3; i++)

    {

        pthread\_create(&wtid[i], NULL, writer, (void \*)i);

        pthread\_create(&rtid[i], NULL, reader, (void \*)i);

    }

    for (i = 0; i <= 2; i++)

    {

        pthread\_join(wtid[i], NULL);

        pthread\_join(rtid[i], NULL);

    }

    return 0;

}

Text

Description automatically generated