**LAB REPORT NO 8**



Submitted by:  **Muhammad Ali**

Registration No: - **19PWCSE1801**

Class Section: A

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr.Mian Ibad Ali shah**

Data:(22,08,2021)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Task 1: Implement producer/consumer problem using threads.**

**Code: -**

#include <stdio.h>

#include <pthread.h>

void \*Producer(void \*);

void \*Consumer(void \*);

void wait(int \*s){

while(s<=0);

s--;

}

void signal(int \*s){

s++;

}

int empty=1,full=0,sm=1;

int global=100;

int main(){

pthread\_t ptid,ctid;

pthread\_create(&ptid,NULL,Producer,NULL);

pthread\_create(&ctid,NULL,Consumer,NULL);

pthread\_join(ptid,NULL);

pthread\_join(ctid,NULL);

return 0;

}

void \*Producer(void \*param){

int i=0;

while(i<2){

wait(&empty);

wait(&sm);

global+=100;

signal(&sm);

signal(&full);

i++;

printf("Producer Global value: %d\n",global);

}

return NULL;

}

void \*Consumer(void \*param){

int i=0;

while(i<2){

wait(&full);

wait(&sm);

global-=25;

signal(&sm);

signal(&empty);

i++;

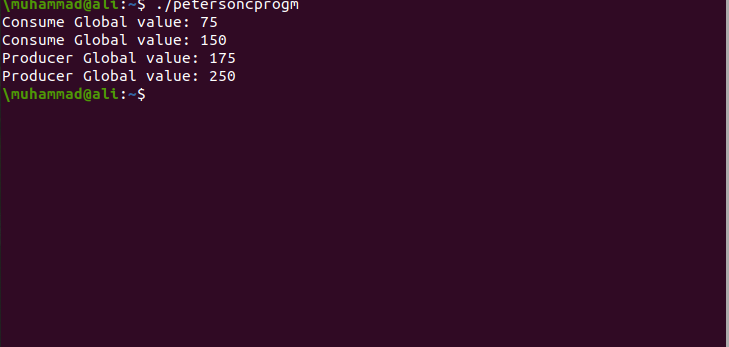
printf("Consume Global value: %d\n",global);

}

return NULL;

}

**Output: -**

****

**Task 2: Solve the critical section problem in task1 using:**

1. **Peterson solution**

**Code: -**

#include<stdio.h>

#include<pthread.h>

void \*function1(void \*);

void \*function2(void \*);

int turn,global=100;

int flag[2];

int turn=0;

int main(){

pthread\_t tid1,tid2;

pthread\_create(&tid1,NULL,function1,NULL);

pthread\_create(&tid2,NULL,function2,NULL);

pthread\_join(tid1,NULL);

pthread\_join(tid2,NULL);

}

void \*function1(void \*param)

{

int i=0;

while(i<2){

flag[0]=1;

turn=1;

while (flag[0]==1 && turn==1){

global+=10;

flag[0]=0;

i++;

printf("first thread is %d\n",global);

}}}

void \*function2(void \*param)

{

int j=0;

while(j<2){

flag[1]=1;

turn=0;

while (flag[1]==1 && turn==0){

global-=75;

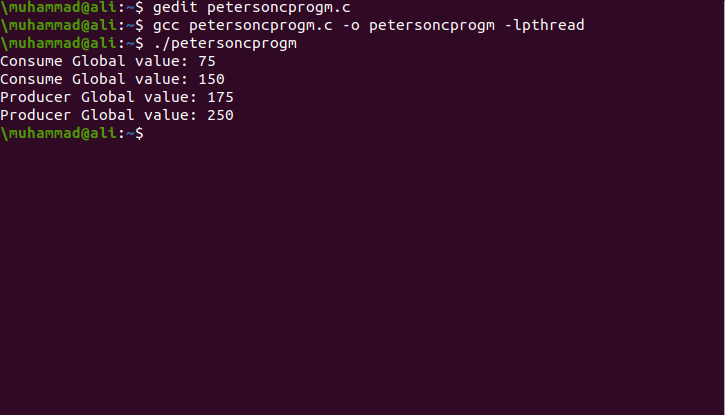
flag[1]=0;

j++;

printf("second thread is %d\n",global);

}}}

**Output: -**

****

1. **Test and set instruction**

**Code: -**

#include <stdio.h>

#include <pthread.h>

void \*func1(void \*);

void \*func2(void \*);

int Lock=0;

int global=100;

int main()

{

pthread\_t tid1,tid2;

pthread\_create(&tid1,NULL,func1,NULL);

pthread\_create(&tid2,NULL,func2,NULL);

pthread\_join(tid1,NULL);

pthread\_join(tid2,NULL);

return 0;

}

int TestAndSet(int \*Lock){

int temp= \*Lock;

\*Lock=1;

return temp;

}

void \*func1(void \*param){

int i=0;

while(i<2) {

while(TestAndSet(&Lock));

global+=100;

Lock=0;

i++;

printf("First Thread: Global: %d\n",global);

}

return NULL;

}

void \*func2(void \*param){

int i=0;

while(i<2){

while(TestAndSet(&Lock));

global-=25;

Lock=0;

i++;

printf("Second Thread: Global: %d\n",global);

}

return NULL;

}

**Output: -**

