

SUBJECT:- OS LAB

DONE BY

NAME:-lokesh

Roll no:-64

Req no:-11803712

1.Gethulb link :- https://github.com/lokesh1314/-oslab-qution-no-1.git

#include<iostream>

#include<thread>

#include<mutex>

using namespace std;

std::mutex m1;

std::mutex m2;

std::mutex m3;

void thread1() {

m1.lock();

m2.lock();

m3.lock();

cout<<"Critical section of Thread Thread One\n";

m1.unlock();

m2.unlock();

m3.unlock();

}

void thread2() {

m2.lock();

m1.lock();

m3.lock();

cout<<"Critical section of Thread Thread Two\n";

m2.unlock();

m1.unlock();

m3.unlock();

}

void thread3() {

m3.lock();

m1.lock();

m2.lock();

cout<<"Critical section of Thread Thread Three\n";

m3.unlock();

m1.unlock();

m2.unlock();

}

int main()

{

thread t1(thread1);

thread t2(thread2);

thread t3(thread3);

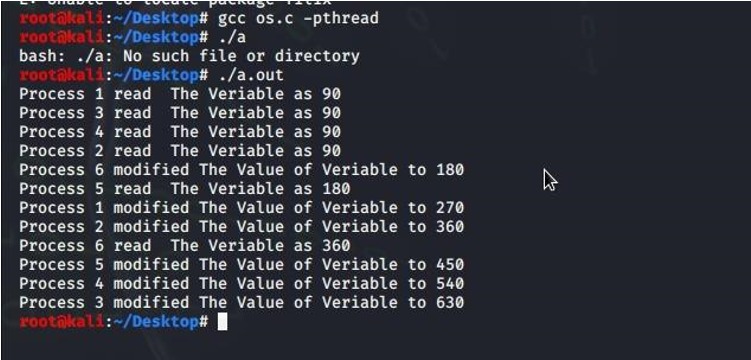
t1.join();

t2.join();

t3.join();

return 0;

}



2 ans

Github link:- https://github.com/lokesh1314/os-partb.git

Code:-

#include<stdio.h>

#include<pthread.h>

int global[2];

void \*sum\_thread(void \*arg)

{

int \*args\_array;

args\_array = arg;

int n1,n2,sum;

n1=args\_array[0];

n2=args\_array[1];

sum = n1+n2;

printf("Hello!!! Welcome to LPU\n");

printf("Sum = %d\n",sum);

return NULL;

}

int main()

{

printf("First number: ");

scanf("%d",&global[0]);

printf("Second number: ");

scanf("%d",&global[1]);

pthread\_t tid\_sum, tid1;

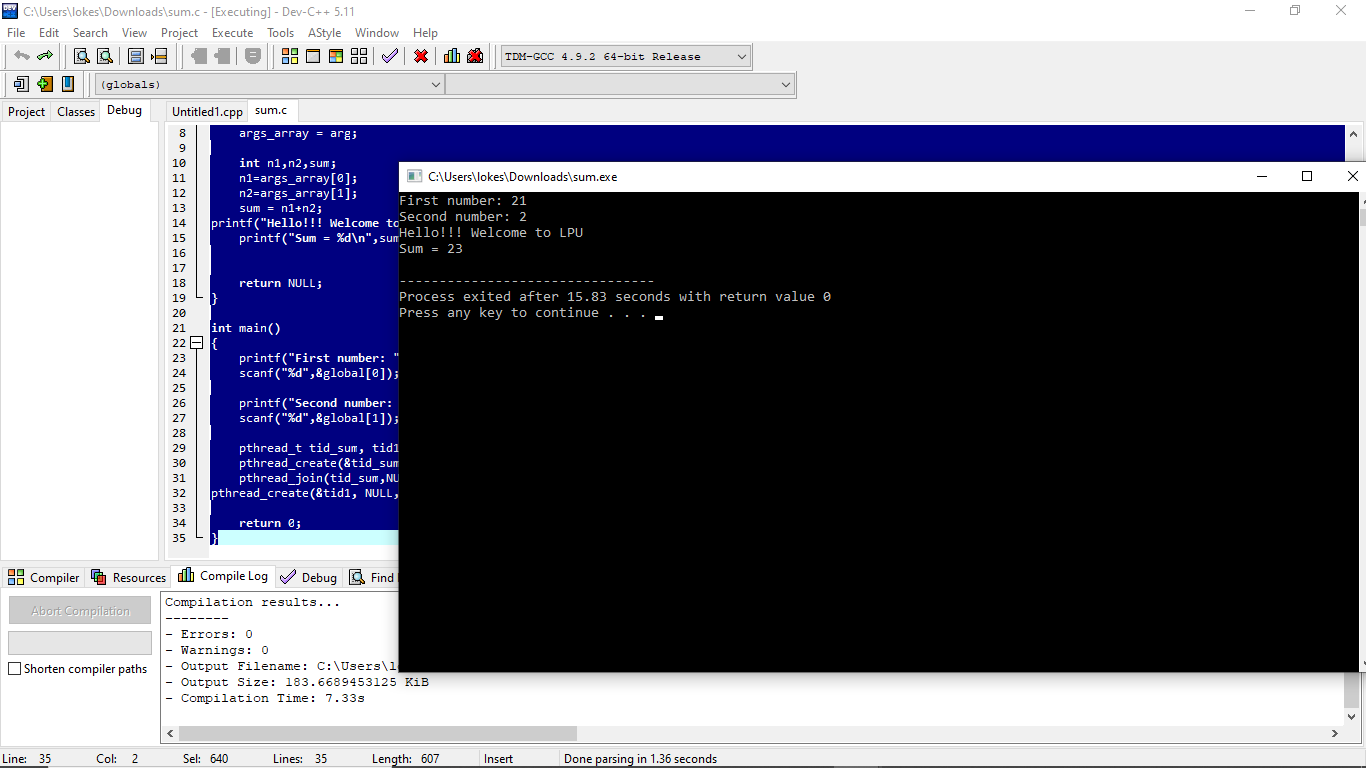
pthread\_create(&tid\_sum,NULL,sum\_thread,global);

pthread\_join(tid\_sum,NULL);

pthread\_create(&tid1, NULL, sum\_thread,NULL);

return 0;

}



3 ans

Github link:- https://github.com/lokesh1314/os-partc.git

Code:-

#include <stdio.h>;

int main()

{

int found,flag,l,p[4][5],tp,tr,c[4][5],i,j,k=1,m[5],r[5],a[5],temp[5],sum=0;

printf("Enter total no of processes");

scanf("%d",&tp);

printf("Enter total no of resources");

scanf("%d",&tr);

printf("Enter claim (Max. Need) matrix\n");

for(i=1;i<=tp;i++)

{

printf("process %d:\n",i);

for(j=1;j<=tr;j++)

scanf("%d",&c[i][j]);

}

printf("Enter allocation matrix\n");

for(i=1;i<=tp;i++)

{

printf("process %d:\n",i);

for(j=1;j<=tr;j++)

scanf("%d",&p[i][j]);

}

printf("Enter resource vector (Total resources):\n");

for(i=1;i<=tr;i++)

{

scanf("%d",&r[i]);

}

printf("Enter availability vector (available resources):\n");

for(i=1;i<=tr;i++)

{

scanf("%d",&a[i]);

temp[i]=a[i];

}

for(i=1;i<=tp;i++)

{

sum=0;

for(j=1;j<=tr;j++)

{

sum+=p[i][j];

}

if(sum==0)

{

m[k]=i;

k++;

}

}

for(i=1;i<=tp;i++)

{

for(l=1;l<k;l++)

if(i!=m[l])

{

flag=1;

for(j=1;j<=tr;j++)

if(c[i][j]<temp[j])

{

flag=0;

break;

}

}

if(flag==1)

{

m[k]=i;

k++;

for(j=1;j<=tr;j++)

temp[j]+=p[i][j];

}

}

printf("deadlock causing processes are:");

for(j=1;j<=tp;j++)

{

found=0;

for(i=1;i<k;i++)

{

if(j==m[i])

found=1;

}

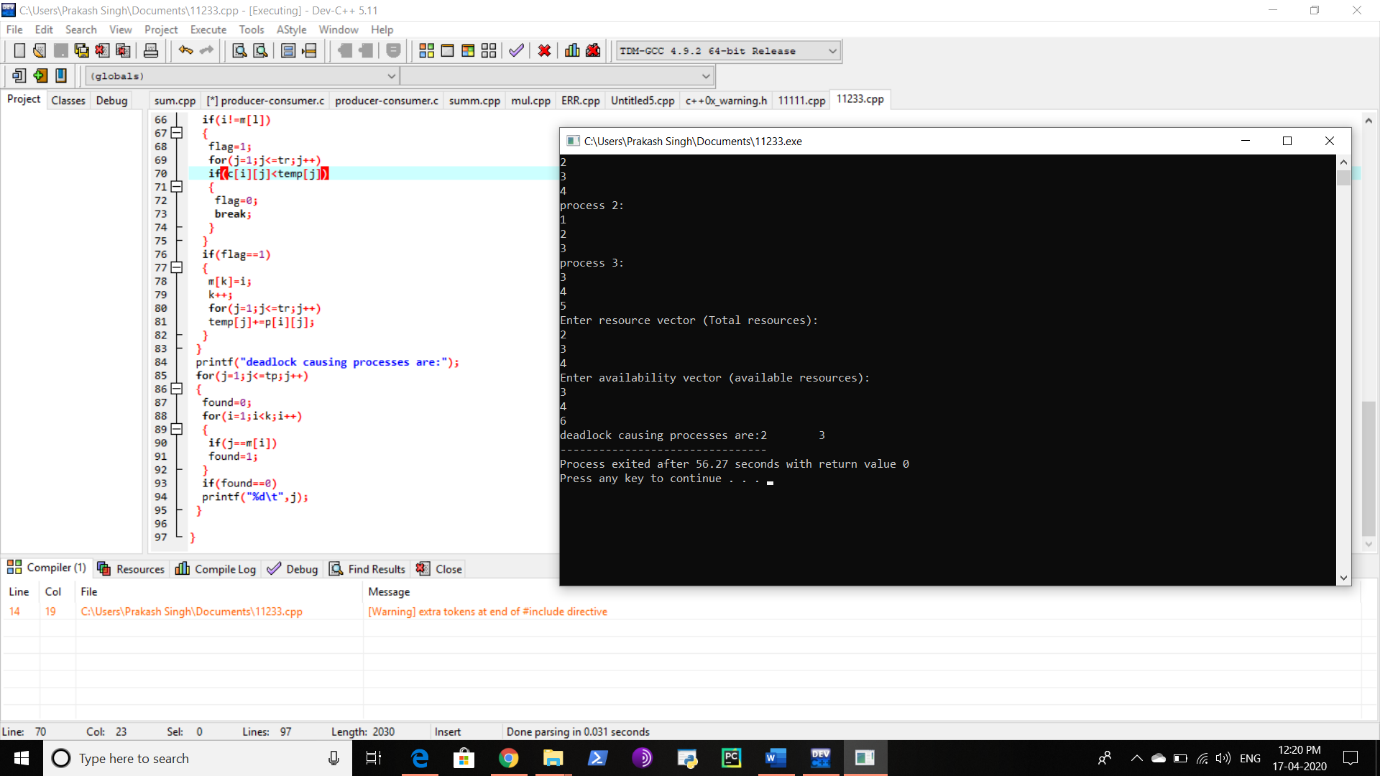
if(found==0)

printf("%d\t",j);

}

}

**OUTPUT-**



4ans

Github link :- https://github.com/lokesh1314/os-lab-part-d.git

Code :-

#include<unistd.h>

#include<stdio.h>

#include<fcntl.h>

int main()

{

int fd , n , p ;

char arr[100];

fd = open("SEEK\_END.txt", O\_CREAT|O\_RDWR, 0777);

n = read(0, arr, 100);

write(fd, arr, n);

p = lseek(fd, -5, SEEK\_END);

read(fd, arrr, 5);

write(1, arr, 5);

printf("\n");

}