Hasib Ar Rafiul Fahim

2019-1-60-036

Lab Tasks

Lab 6:

Task 1:

#include<stdio.h>

#include<pthread.h>

int arr[50];

pthread\_mutex\_t lock;

pthread\_cond\_t cond;

int count = 0;

void \* store(void \* n){

int i;

pthread\_mutex\_lock(&lock);

while(count>2){

printf("Waiting");

pthread\_cond\_wait(&cond,&lock);

}

printf("Adding the values to array from 1 to n \n");

for(i=0;i<50;i++){

arr[i]=i;

count++;

}

pthread\_mutex\_unlock(&lock);

}

void \* print(void \* n){

int i;

pthread\_mutex\_lock(&lock);

pthread\_cond\_signal(&cond);

printf("Printing the values:\n ");

for(i=0;i<50;i++){

printf("%d ",arr[i]);

count--;

}

printf("\n");

pthread\_mutex\_unlock(&lock);

}

int main()

{

pthread\_t tid,tid2;

pthread\_mutex\_init(&lock,NULL);

pthread\_cond\_init(&cond,NULL);

pthread\_create(&tid,NULL,&store,NULL);

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

pthread\_join(tid,NULL);

pthread\_join(tid2,NULL);

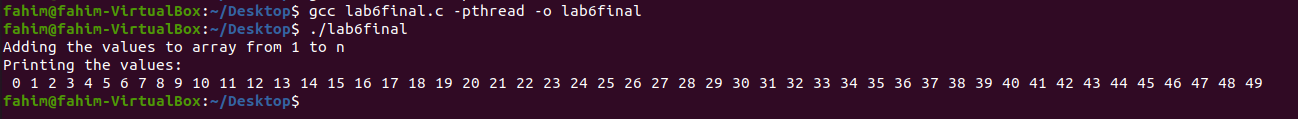
pthread\_mutex\_destroy(&lock);

pthread\_cond\_destroy(&cond);

return 0;

}

Here is the screenshot of the output below:



Task 2:

#include<stdio.h>

#include<pthread.h>

#include<semaphore.h>

int sum=0;

sem\_t lock;

void \* runner(void \* any)

{

int n,i=0;

printf("\nChild Thread\n");

printf("Enter n: ");

scanf("%d",&n);

sem\_wait(&lock);

for(i=0;i<n+1;i++){

sum=sum+i;

}

sem\_post(&lock);

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

}

int main()

{

void \*sum1,\*sum2;

int i=0,num;

sem\_init(&lock,0,2);

printf("Enter how many thread you want: ");

scanf("%d",&num);

pthread\_t tid[num];

printf("\nMain thread\n");

for(i=0;i<num;i++){

pthread\_create(&tid[i],NULL,&runner,NULL);

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

}

return 0;

}

