

# **Lab Assignment-5**

**Name:** Sadman Sharif

**ID:** 20101107

**Section:** 02

**Course:** CSE321

**Submission Date:** 08/12/2022

# Task-1

```
#include <pthread.h>
#include <semaphore.h>
#include <stdlib.h>
#include <stdio.h>

#define Max 10
#define BufLen 6
#define NUMTHREAD 2

sem_t empty;
sem_t full;

int in= 0,pro=0,con=0,out=0;

int buffer[BufLen];

char item ='a';

pthread_mutex_t mutex;

void *producer(void *pno){

    for(int i=0;i<Max; i++){

        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        buffer[in]= item;
        printf("%d produced %c by thread %d\n",pro,buffer[in],*((int
*)pno));

        in = (in+1) % BufLen;
        pro=pro+1;
        item+= 1;

        pthread_mutex_unlock(&mutex);
        sem_post(&full);
    }
}
```

```

void *consumer(void *cno){

    for(int i=0;i< Max;i++){

        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        int item = buffer[out];

        printf("%d consumed %c by thread %d\n",con,item,*((int
*)cno));

        con=con+1 ;
        out=(out+1)%BufLen;

        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
    }

}

int main(){

    pthread_t p[10],c[10];

    pthread_mutex_init(&mutex, NULL);
    sem_init(&empty,0,BufLen);
    sem_init(&full,0,0);

    int p_ref[1] = {0};
    int c_ref[1] = {1};

    pthread_create(&p[0],NULL, (void *)producer,(void *)&p_ref[0]);

    pthread_create(&c[0], NULL, (void *)consumer,(void *)&c_ref[0]);

    pthread_join(c[0],NULL);
    pthread_join(p[0],NULL);

    pthread_mutex_destroy(&mutex);

    sem_destroy(&empty);
    sem_destroy(&full);

    return 0;
}

```

# Task-1 (Output)

```
sadman@sadman: ~/$ gcc ami.c
cc1: fatal error: ami.c: No such file or directory
compilation terminated.
sadman@sadman: ~/$ pwd
/home/sadman
sadman@sadman: ~/$ cd Desktop
sadman@sadman: ~/Desktop$ gcc ami.c
sadman@sadman: ~/Desktop$ ./a.out
0 produced a by thread 0
1 produced b by thread 0
2 produced c by thread 0
3 produced d by thread 0
4 produced e by thread 0
5 produced f by thread 0
0 consumed a by thread 1
1 consumed b by thread 1
2 consumed c by thread 1
3 consumed d by thread 1
4 consumed e by thread 1
5 consumed f by thread 1
6 produced g by thread 0
7 produced h by thread 0
8 produced i by thread 0
9 produced j by thread 0
6 consumed g by thread 1
7 consumed h by thread 1
8 consumed i by thread 1
9 consumed j by thread 1
sadman@sadman: ~/Desktop$
```

## Task-2

```
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>

#define MaxCrops 5
#define warehouseSize 5

sem_t empty;
sem_t full;

int in=0,out=0;
char
crops[warehouseSize]={'R','W','P','S','M'},warehouse[warehouseSize]={'
N','N','N','N','N'};

pthread_mutex_t mutex;

void *Farmer(void *farm)
{
    for(int i=0;i<MaxCrops;i++){
        char item=crops[in];
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        warehouse[in]=item;
        printf("Farmer %d: Insert crops %c at %d\n",*((int
*)farm),warehouse[in],in);
        in=(in+1)%warehouseSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
    }
}

void *ShopOwner(void *shop)
{
    for(int i = 0; i < MaxCrops; i++){

        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        char item = warehouse[out];
        printf("Shop Owner %d: Removes crops %c from %d\n",*((int
*)shop),item,out);
```

```

        warehouse[out]='N';
        out=(out+1)%warehouseSize;
        printf("ShopOwner%d:",*((int*)shop));
        for(int i=0;i<warehouseSize;i++){
            printf("%c",warehouse[i]);
        }
        printf("\n");

        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
    }

}

int main(){

    pthread_t Far[5],Sho[5];
    pthread_mutex_init(&mutex, NULL);
    sem_init(&empty,0,warehouseSize);
    sem_init(&full,0,0);

    int a[5]={1,2,3,4,5};

    for(int i=0;i<5;i++){
        pthread_create(&Far[i],NULL,(void *)Farmer,(void *)&a[i]);
    }
    for(int i=0;i<5;i++){
        pthread_create(&Sho[i], NULL,(void *)ShopOwner,(void
*)&a[i]);
    }
    for(int i=0;i<5;i++){
        pthread_join(Far[i],NULL);
    }
    for(int i=0;i<5;i++){
        pthread_join(Sho[i],NULL);
    }

    pthread_mutex_destroy(&mutex);
    sem_destroy(&empty);
    sem_destroy(&full);

    return 0;

}

```

## Task-2 (Output)

```
sadman@sadman: ~/Desktop
Shop Owner 4: Removes crops W from 4
ShopOwner4:RSNNN
Farmer 2: Insert crops W at 2
Farmer 2: Insert crops S at 3
Shop Owner 2: Removes crops R from 0
ShopOwner2:NSWSN
Shop Owner 2: Removes crops S from 1
ShopOwner2:NNWSN
Farmer 4: Insert crops P at 4
Shop Owner 1: Removes crops W from 2
ShopOwner1:NNNSP
Farmer 3: Insert crops P at 0
Shop Owner 1: Removes crops S from 3
ShopOwner1:PNNNP
Farmer 3: Insert crops W at 1
Shop Owner 1: Removes crops P from 4
ShopOwner1:PWNNN
Farmer 3: Insert crops P at 2
Farmer 1: Insert crops R at 3
Shop Owner 3: Removes crops P from 0
ShopOwner3:NWPRN
Farmer 5: Insert crops M at 4
Shop Owner 4: Removes crops W from 1
ShopOwner4:NNPRM
Shop Owner 4: Removes crops P from 2
ShopOwner4:NNNRM
Farmer 1: Insert crops M at 0
Farmer 1: Insert crops W at 1
Farmer 5: Insert crops R at 2
Shop Owner 3: Removes crops R from 3
ShopOwner3:MWRNM
Farmer 5: Insert crops S at 3
Shop Owner 2: Removes crops M from 4
ShopOwner2:MWRSN
Farmer 3: Insert crops S at 4
Shop Owner 3: Removes crops M from 0
ShopOwner3:NWRSS
Shop Owner 3: Removes crops W from 1
ShopOwner3:NNRSS
Shop Owner 3: Removes crops R from 2
ShopOwner3:NNNSS
Shop Owner 1: Removes crops S from 3
ShopOwner1:NNNNS
Shop Owner 1: Removes crops S from 4
ShopOwner1:NNNNN
sadman@sadman:~/Desktop$ ./a.out
```