

OPERATING SYSTEM ASSIGNMENT 4

NAME: Kaushal Oza

SRN: 201900754

T.Y B.TECH A - 40

ASSIGNMENT NO – 4

Q1) THREADING USING MUTEX

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <time.h>

#define Iterations 12
#define Size 7

int in = 0; int out =
0; int buffer[Size] =
{0};
pthread_mutex_t mutex;

void show()
{
    for (int i=0;i<Size;i++)
    printf("%d ",buffer[i]);
    printf("\n");
}

void* prod(void* _args)
{
    int
    item;
    for(int i=0;i<Iterations;i++)
    {
```

```

        sleep(rand() % 3);    item = 1
+ rand() % 10;
pthread_mutex_lock(&mutex);
buffer[in] = item;
printf("\nProduced :
%d",buffer[in]);
printf("\nBuffer Status : ");
    show();
    in = (in+1)%Size;
pthread_mutex_unlock(&mutex);
}
}
void* cons(void* _args)
{
    for(int i=0;i<Iterations;i++)
    {
        sleep(rand() % 5);
        int item = buffer[out];
pthread_mutex_lock(&mutex);
buffer[out] = 0;
printf("\nConsumed : %d",item);
printf("\nBuffer Status : ");
show();    out = (out+1)%Size;
        pthread_mutex_unlock(&mutex);
    }
}

int main()
{
    pthread_t pro,con;
pthread_mutex_init(&mutex,NULL);
pthread_create(&pro,NULL,prod,NULL);
pthread_create(&con,NULL,cons,NULL);
sleep(10);  pthread_join(pro,NULL);
pthread_join(con,NULL);
    pthread_mutex_destroy(&mutex);
    return 0;
}

```

OUTPUT:

Produced : 8
Buffer Status : 8 0 0 0 0 0

Consumed : 0
Buffer Status : 8 0 0 0 0 0 0

Produced : 1
Buffer Status : 8 1 0 0 0 0 0

Produced : 5
Buffer Status : 8 1 5 0 0 0 0

Produced : 9
Buffer Status : 8 1 5 9 0 0 0

Produced : 5
Buffer Status : 8 1 5 9 5 0 0

Consumed : 5
Buffer Status : 8 1 0 9 5 0 0

Consumed : 9
Buffer Status : 8 1 0 0 5 0 0

Produced : 6
Buffer Status : 8 1 0 0 5 6 0

Produced : 8
Buffer Status : 8 1 0 0 5 6 8

Produced : 2
Buffer Status : 2 1 0 0 5 6 8

Consumed : 5
Buffer Status : 2 1 0 0 0 6 8

Produced : 3
Buffer Status : 2 3 0 0 0 6 8

Produced : 7
Buffer Status : 2 3 7 0 0 6 8

Produced : 5
Buffer Status : 2 3 7 5 0 6 8

Produced : 4
Buffer Status : 2 3 7 5 4 6 8

Consumed : 6
Buffer Status : 2 3 7 5 4 0 8

Consumed : 8
Buffer Status : 2 3 7 5 4 0 0

Consumed : 2
Buffer Status : 0 3 7 5 4 0 0

Consumed : 3
Buffer Status : 0 0 7 5 4 0 0

Consumed : 7
Buffer Status : 0 0 0 5 4 0 0

Consumed : 5
Buffer Status : 0 0 0 0 4 0 0

Consumed : 4
Buffer Status : 0 0 0 0 0 0 0

Q2) THREADING USING SEMAPHORES

CODE:

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

#define Iterations 12
#define Size 7

sem_t empty; sem_t
full;
sem_t lock;
int in = 0; int
out = 0;
int buffer[Size] = {0};
```

```
void show()
```

```
{
```

```
    for (int i=0;i<Size;i++)
```

```
printf("%d ",buffer[i]);
```

```
printf("\n");
```

```
}
```

```
void* prod(void* _args)
```

```
{  int
```

```
item;
```

```
for(int
```

```
i=0;i<Itera
```

```
tions;i++)
```

```
{
```

```
    sleep(rand() % 3);    item = 1 +
```

```
rand() % 10;    sem_wait(&empty);
```

```
sem_wait(&lock);    buffer[in] = item;
```

```
printf("\nProduced : %d",buffer[in]);
```

```
printf("\nBuffer Status : ");
```

```
    show();    in =
```

```
(in+1)%Size;
```

```
sem_post(&full);
```

```
    sem_post(&lock);
```

```
}
```

```
}
```

```
void* cons(void* _args)
```

```
{
```

```
    for(int i=0;i<Iterations;i++)
```

```
{
```

```
    sleep(rand() % 5);
```

```
sem_wait(&full);
```

```
sem_wait(&lock);    int item =
```

```
buffer[out];    buffer[out] = 0;
```

```
printf("\nConsumed : %d",item);
```

```
printf("\nBuffer Status : ");
```

```
show();    out = (out+1)%Size;
```

```
sem_post(&empty);
```

```
    sem_post(&lock);
```

```
}
```

```
}
```

```
int main()
```

```

{
    pthread_t pro,con;
    sem_init(&lock,0,Size);
    sem_init(&empty,0,Size);
    sem_init(&full,0,0);
    pthread_create(&pro,NULL,prod,NULL);
    pthread_create(&con,NULL,cons,NULL);
    sleep(10);  pthread_join(pro,NULL);
    pthread_join(con,NULL);
    sem_destroy(&lock);
    sem_destroy(&empty);
    sem_destroy(&full);
    return 0;
}

```

OUTPUT:

Produced : 8
Buffer Status : 8 0 0 0 0 0 0

Consumed : 8
Buffer Status : 0 0 0 0 0 0 0

Produced : 1
Buffer Status : 0 1 0 0 0 0 0

Consumed : 1
Buffer Status : 0 0 0 0 0 0 0

Produced : 5
Buffer Status : 0 0 5 0 0 0 0

Produced : 9
Buffer Status : 0 0 5 9 0 0 0

Produced : 5
Buffer Status : 0 0 5 9 5 0 0

Consumed : 5
Buffer Status : 0 0 0 9 5 0 0

Consumed : 9
Buffer Status : 0 0 0 0 5 0 0

Produced : 6
Buffer Status : 0 0 0 0 5 6 0

Produced : 8
Buffer Status : 0 0 0 0 5 6 8

Produced : 2
Buffer Status : 2 0 0 0 5 6 8

Produced : 3
Buffer Status : 2 3 0 0 5 6 8

Produced : 7
Buffer Status : 2 3 7 0 5 6 8

Produced : 5
Buffer Status : 2 3 7 5 5 6 8

Consumed : 5
Buffer Status : 2 3 7 5 0 6 8

Produced : 4
Buffer Status : 2 3 7 5 4 6 8

Consumed : 6
Buffer Status : 2 3 7 5 4 0 8

Consumed : 8
Buffer Status : 2 3 7 5 4 0 0

Consumed : 2
Buffer Status : 0 3 7 5 4 0 0

Consumed : 3
Buffer Status : 0 0 7 5 4 0 0

Consumed : 7
Buffer Status : 0 0 0 5 4 0 0

Consumed : 5
Buffer Status : 0 0 0 0 4 0 0

Consumed : 4
Buffer Status : 0 0 0 0 0 0 0

