

5. Write a C program to simulate producer-consumer problem using Semaphores.

Code:

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

int mutex=1,full=0,empty=3,x=0;

int main()
{
    int n;
    void producer();
    void consumer();
    int wait(int);
    int signal(int);
    printf("\n1.Producer\n2.Consumer\n3.Exit");
    while(1)
    {
        printf("\nEnter your choice:");
        scanf("%d",&n);
        switch(n)
        {
            case 1:  if((mutex==1)&&(empty!=0))
                      producer();
                    else
                      printf("Buffer is full!!");
                      break;
            case 2:  if((mutex==1)&&(full!=0))
                      consumer();
                    else
                      printf("Buffer is empty!!");
                      break;
            case 3:
                      exit(0);
                      break;
        }
    }

    return 0;
}

int wait(int s)
```

```

{
    return (--s);
}

int signal(int s)
{
    return(++s);
}

void producer()
{
    mutex=wait(mutex);
    full=signal(full);
    empty=wait(empty);
    x++;
    printf("\nProducer produces the item %d",x);
    mutex=signal(mutex);
}

void consumer()
{
    mutex=wait(mutex);
    full=wait(full);
    empty=signal(empty);
    printf("\nConsumer consumes item %d",x);
    x--;
    mutex=signal(mutex);
}

```

Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc podcon.c -o podcon } ; if ($?) { .\podcon }
1.Producer
2.Consumer
3.Exit
Enter your choice:1

Producer produces the item 1
Enter your choice:1

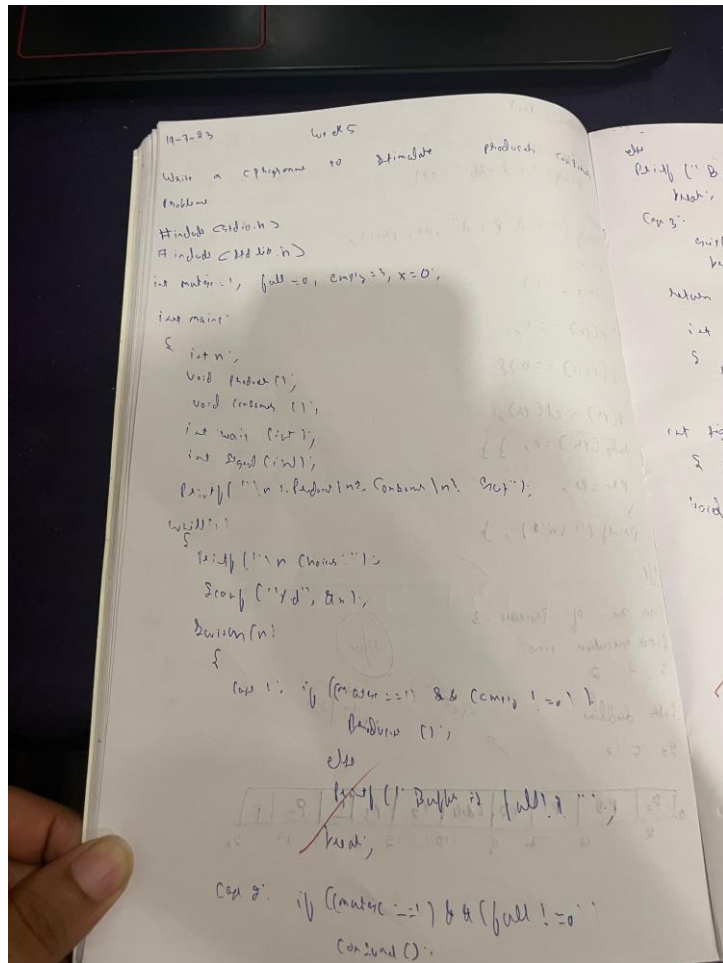
Producer produces the item 2
Enter your choice:2

Consumer consumes item 2
Enter your choice:2

Consumer consumes item 1
Enter your choice:2
Buffer is empty!!
Enter your choice:3
PS D:\VS Code\OS>

```

Observation:



```

// Producer-Consumer Problem
// Using semaphores

#include <iostream>
using namespace std;

const int N = 10; // Buffer size
int buffer[N];
int in = 0, out = 0;

sem_t mutex;
sem_t empty;
sem_t full;

// Producer function
void producer() {
    while (1) {
        // Wait for an empty slot
        wait(empty);

        // Produce an item
        int item = rand() % 100;

        // Lock the buffer
        wait(mutex);

        // Add item to buffer
        buffer[in] = item;
        in = (in + 1) % N;

        // Signal that a new item is added
        signal(full);
    }
}

// Consumer function
void consumer() {
    while (1) {
        // Wait for a full buffer
        wait(full);

        // Lock the buffer
        wait(mutex);

        // Remove item from buffer
        int item = buffer[out];
        out = (out + 1) % N;

        // Signal that an item is removed
        signal(empty);

        // Consume the item
        cout << "Consumed: " << item << endl;
    }
}

int main() {
    // Initialize semaphores
    sem_init(&mutex, 0, 1);
    sem_init(&empty, 0, N);
    sem_init(&full, 0, 0);

    // Create producer and consumer threads
    pthread_t p, c;
    pthread_create(&p, NULL, producer, NULL);
    pthread_create(&c, NULL, consumer, NULL);

    // Wait for threads to finish
    pthread_join(p, NULL);
    pthread_join(c, NULL);

    return 0;
}

```

printf("In Consumer (consume item %d", x);

x--;

return signal (mutex);

Output

1) Producer

2) Consumer

3) Exit

Enter your choice: 1

Producer produces fu item)

Enter your choice: 2

Consumer consumes item 1

10/10

19/7/23

