Ex. No.: 8
Date: 13.04.2024

PRODUCER CONSUMER USING SEMAPHORES

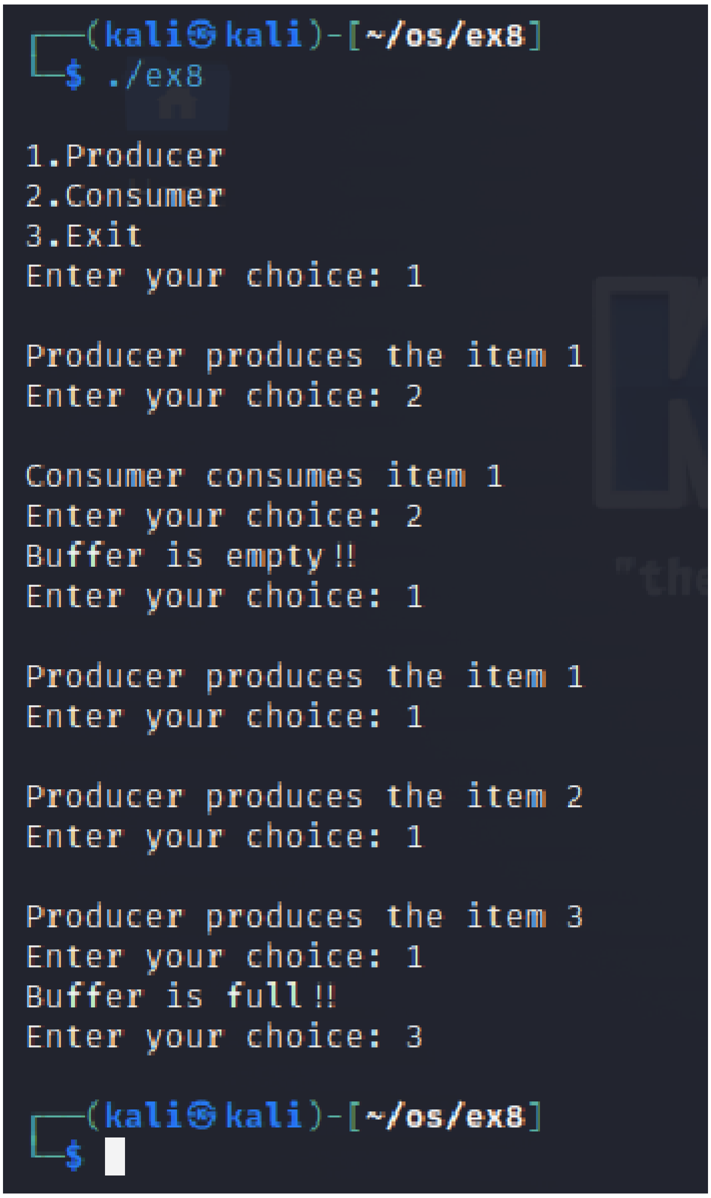
Aim:

To write a program to implement solution to producer consumer problem using semaphores.

Program Code:

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

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

void producer();
void consumer();
int wait(int);
int signal(int);

int main() {
   int n;

   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:



```
┌──(kali㉿kali)-[~/os/ex8]
└─$ ./ex8

1.Producer
2.Consumer
3.Exit
Enter your choice: 1

Producer produces the item 1
Enter your choice: 2

Consumer consumes item 1
Enter your choice: 2
Buffer is empty!!
Enter your choice: 1

Producer produces the item 1
Enter your choice: 1

Producer produces the item 2
Enter your choice: 1

Producer produces the item 3
Enter your choice: 1
Buffer is full!!
Enter your choice: 3

┌──(kali㉿kali)-[~/os/ex8]
└─$ 
```

Result:

The above program executed successfully and output got verified.