**EXPERIMENT 9**

**OBJECTIVE**:-

Program to design consumer producer problem using semaphores.

**CODE**:-

//------------------------------------------------------------------------------

//

// Desc: consumer\_producer.c

//

// Date: 29/3/2018

//

// Author: VU DUY DU - 2k16/CO/364

//

//------------------------------------------------------------------------------

#include<stdio.h>

#include<stdlib.h>

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

int main(){

printf("Hello Consumer Producer Problem using Semaphores\n");

int n;

//function prototypes

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!\n");

break;

case 2:

if(mutex == 1 && full!=0)

consumer();

else

printf("Buffer is empty!\n");

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:-**

Hello Consumer Producer Problem using Semaphores

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

**DISCUSSION:-**

Producer consumer problem is also known as bounded buffer problem. In this problem we have two processes, producer and consumer, who share a fixed size buffer. Producer work is to produce data or items and put in buffer. Consumer work is to remove data from buffer and consume it. We have to make sure that producer do not produce data when buffer is full and consumer do not remove data when buffer is empty.

*Source code of this experiment can be found here:*

*https://github.com/Sieunguoimay/OSLab-4thsem-2018/tree/master/Exp9*