OS LAB EXP 13

**Q1. Execute and write the output of the following program for *mutual exclusion*.**

**Output:**

The Child sets WAIT signal & doing her job

The Parent waits for WAIT signal

The Child sets WAKE signal & finished her job

Child Over

The Parent WAKED UP & doing her job

Parent Over

**Q2. Write a program to perform process synchronization in producer-consumer problem**

#include <stdio.h>

#include <stdlib.h>

int mutex = 1;

int full = 0;

int empty = 10, x = 0;

void producer()

{

--mutex;

// Increase the number of full slots by 1

++full;

// Decrease the number of empty slots by 1

--empty;

// Item produced

x++;

printf("\nProducer produces item %d",x);

// Increase mutex value by 1

++mutex;

}

void consumer()

{

// Decrease mutex value by 1

--mutex;

// Decrease the number of full slots by 1

--full;

// Increase the number of empty slots by 1

++empty;

printf("\nConsumer consumes item %d",x);

x--;

// Increase mutex value by 1

++mutex;

}

int main()

{

int n, i;

printf("\n1. Press 1 for Producer"

"\n2. Press 2 for Consumer"

"\n3. Press 3 for Exit");

for (i = 1; i > 0; i++) {

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;

}

}

}

**OUTPUT:**

1. Press 1 for Producer

2. Press 2 for Consumer

3. Press 3 for Exit

Enter your choice: 1

Producer produces item 1

Enter your choice: 1

Producer produces 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