COSC 350 System Software (Lab #10)

Task#1

Write complete two C programs “msgQsnd.c” and “msgQrcv.c” to communicate through message queue .

msgQsnd.c:

- Create a message queue with rw-r-r. To create a message queue, use existing file name “msgQsnd.c for creating a key value.
- Ask a message “ two integers value” and send to the message queue.
- Keep asking a message until EOF (Ctr-D)
- Remove the message queue with termination.

msgQrcv.c:

- Receive a message (two integers) from the message queue created by msgQsnd.c.
- Calculate sum of two integers and display result on standard output.
- keep reading a message until EOF

What is producer Consumer Problem

- Two processes share a common, fixed-sized buffer size 10.
- Producer puts information into the buffer, and consumer takes it out.

```
#define N 10
int count = 0;
void producer()
{
    int item
    while (ture)
    {
        item = produce_item();
        if (count == N)
            sleep();
        insert_item(item)
        count = count + 1;
        if (count ==1)
            wakeup(consumer);
    }
}
```

```
void cunsumer()
{
    int item;
    while(true)
    {
        if (count == 0)
            sleep();
        item = remove_item();
        count = count – 1;
        if (count == N – 1)
            wakeup(producer);
        consume_item(item);
    }
}
```

Troubles arises

- When the producer wants to put a new item in the buffer, but it is already full.
- When the consumer tries to take an item from the buffer, but buffer is already empty.

Solutions for each case

- When the producer wants to put a new item in the buffer, but it is already full.
  - Solution – producer is going to sleep, awakened by customer when customer has removed on or more items.
- When the consumer tries to take an item from the buffer, but buffer is already empty.
  - Solution – customer is going to sleep, awakened by the producer when producer puts one or more information into the buffer.

Task #2 Write a complete C program to simulate producer consumer problems without using semaphores. Just simulate previous algorithms with count variables. And shows the race condition problems.

- You need create two threads to simulate: producer and consumer

Task #3: Write a complete C program to simulate producer consumer problems with semaphores.

- You need create two threads: producer and consumer.
- You need use semaphores: two countable semaphore for empty and full and one binary semaphore mutex for mutual exclusion in the algorithm
- You need find out a way to demonstrate your program works properly.
- You need use ftok(), semget(), semctl() and semop() system calls for semaphores.

Algorithms for Solving Producer Customer Problem using semaphores

```
#define N 10
typedef int semaphore;
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;
void producer ()
{
    int item;

    while (true)
    {
        item = produce_item();
        down (&empty);
        down (&mutex);
        insert_item(item);
        up(&mutex);
        up(&full);
    }
}
```

```
void consumer()
{
    int item;

    while (true)
    {
        down(&full)
        down(&mutex)
        item = remove_item();
        up(&mutex);
        up(&empty);
        consume_item(item);
    }
}
```