Operating System Lab
CS342

Tanishq Malu Lab:7 1901CS63

Q1.

Pseudo-Code

Writer-code

```
do {
    wait(wrt);             // writer requests for critical section
    // performs the write
    // leaves the critical section
    signal(wrt);
} while(true);
```

Reader-code

```
do {
   wait(mutex);            // Reader wants to enter the critical section
   readcnt++;              // The number of readers has now increased by1
   if (readcnt==1)         // Giving preference to reader even if one reader is present
      wait(wrt);

   signal(mutex);          // other readers can enter while this current reader is inside
                              the critical section

   // current reader performs reading here

   wait(mutex);            // a reader wants to leave
   readcnt--;

   if (readcnt == 0)       // If no reader is left in the critical section
       signal(wrt);        // writers can enter

   signal(mutex); // reader leaves
} while(true);
```

Compilation:

```
g++ -o q1 q1.c -lm -pthread -fopenmp
```

Syntax:

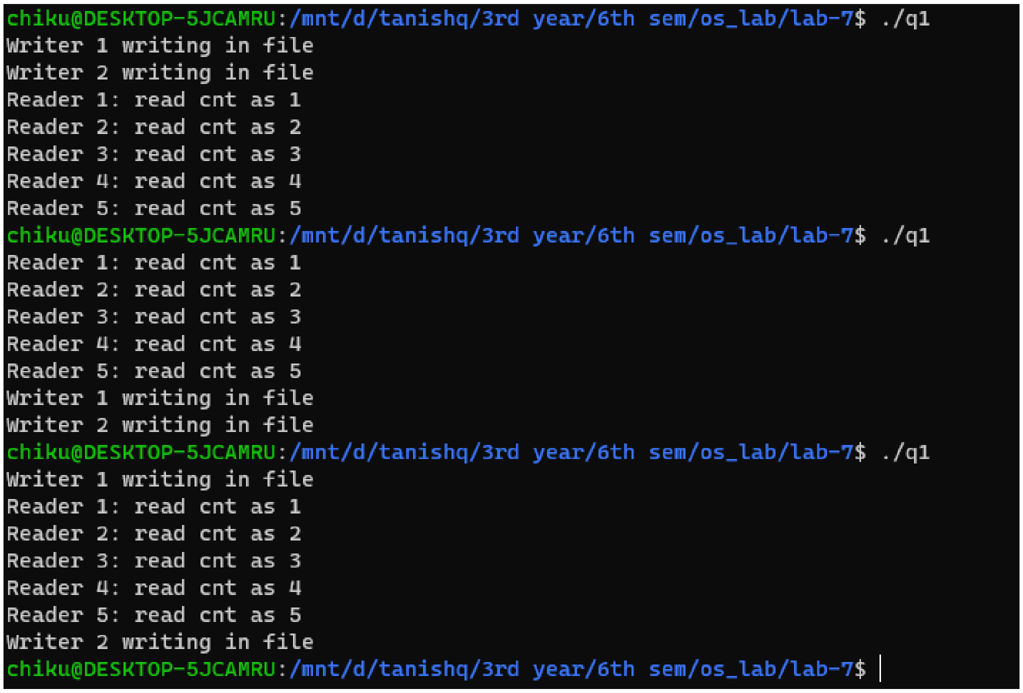
```
./q1
```

Sample Input and Output:

Input:

```
g++ -o q1 q1.c -lm -pthread -fopenmp
./q1
```

Output:



```
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q1
Writer 1 writing in file
Writer 2 writing in file
Reader 1: read cnt as 1
Reader 2: read cnt as 2
Reader 3: read cnt as 3
Reader 4: read cnt as 4
Reader 5: read cnt as 5
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q1
Reader 1: read cnt as 1
Reader 2: read cnt as 2
Reader 3: read cnt as 3
Reader 4: read cnt as 4
Reader 5: read cnt as 5
Writer 1 writing in file
Writer 2 writing in file
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q1
Writer 1 writing in file
Reader 1: read cnt as 1
Reader 2: read cnt as 2
Reader 3: read cnt as 3
Reader 4: read cnt as 4
Reader 5: read cnt as 5
Writer 2 writing in file
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$
```

Q 2.

```
semaphore *mutex, *data;
void A() {
        wait(mutex);
        wait(data);
        print(“process A”)
        signal(mutex);
        signal(data);
}
void B() {
        wait(data);
        wait(mutex);
        print(“process B”)
        signal(data);
        signal(mutex);

}
```

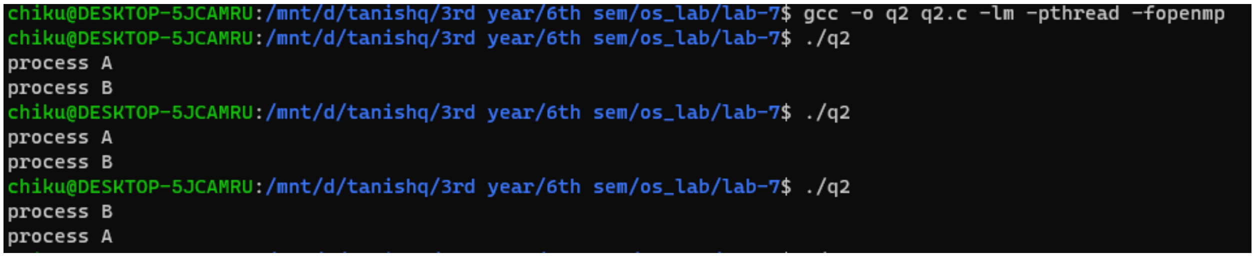
Sequence

1. Suppose, function A starts running and it calls wait(mutex) (A context switch occurs)

2. Then we switch to B and it calls wait(data).

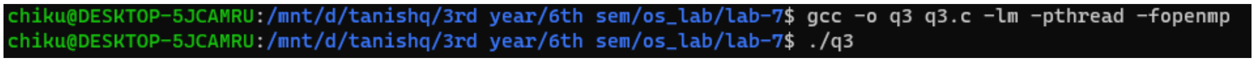
3. Now, function B waits until semaphore - mutex gets freed and function A waits until semaphore – data gets freed.

4. Thus they have entered into a deadlock.

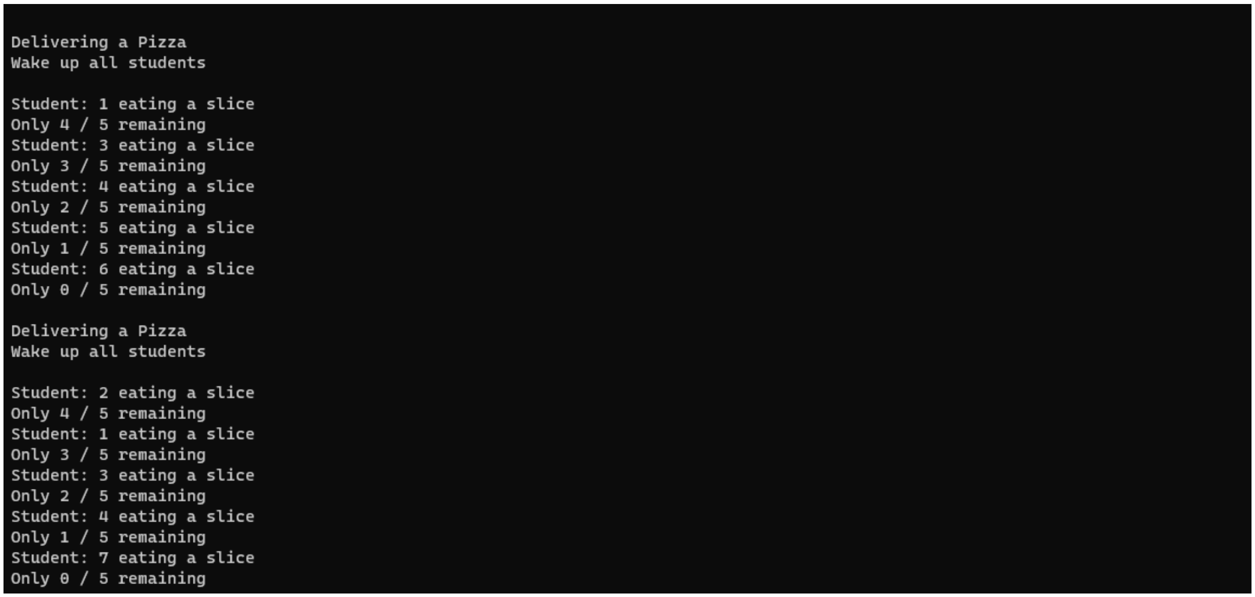


```
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ gcc -o q2 q2.c -lm -pthread -fopenmp
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q2
process A
process B
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q2
process A
process B
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q2
process B
process A
```

Q 3.



```
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ gcc -o q3 q3.c -lm -pthread -fopenmp
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$ ./q3
```



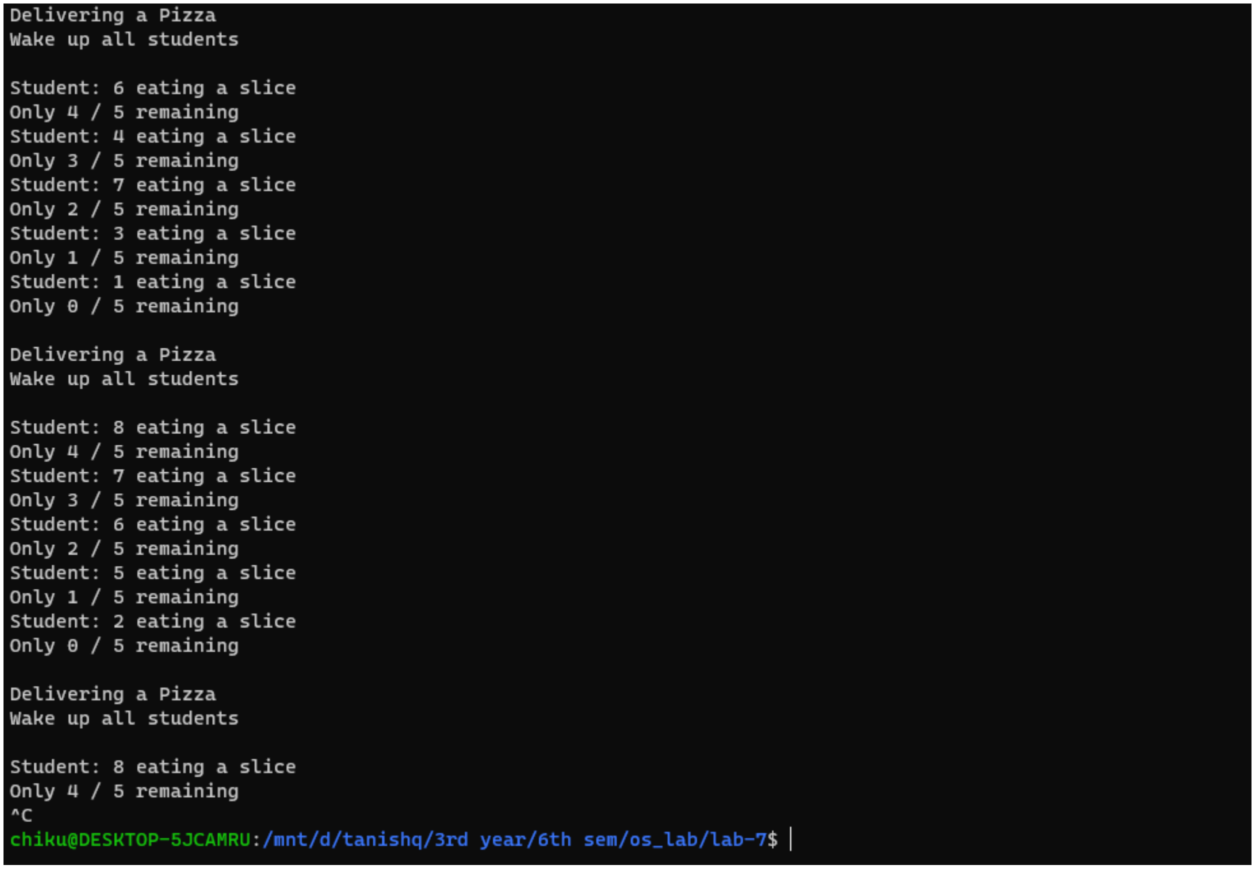
```

Delivering a Pizza
Wake up all students

Student: 1 eating a slice
Only 4 / 5 remaining
Student: 3 eating a slice
Only 3 / 5 remaining
Student: 4 eating a slice
Only 2 / 5 remaining
Student: 5 eating a slice
Only 1 / 5 remaining
Student: 6 eating a slice
Only 0 / 5 remaining

Delivering a Pizza
Wake up all students

Student: 2 eating a slice
Only 4 / 5 remaining
Student: 1 eating a slice
Only 3 / 5 remaining
Student: 3 eating a slice
Only 2 / 5 remaining
Student: 4 eating a slice
Only 1 / 5 remaining
Student: 7 eating a slice
Only 0 / 5 remaining
```



```
Delivering a Pizza
Wake up all students

Student: 6 eating a slice
Only 4 / 5 remaining
Student: 4 eating a slice
Only 3 / 5 remaining
Student: 7 eating a slice
Only 2 / 5 remaining
Student: 3 eating a slice
Only 1 / 5 remaining
Student: 1 eating a slice
Only 0 / 5 remaining

Delivering a Pizza
Wake up all students

Student: 8 eating a slice
Only 4 / 5 remaining
Student: 7 eating a slice
Only 3 / 5 remaining
Student: 6 eating a slice
Only 2 / 5 remaining
Student: 5 eating a slice
Only 1 / 5 remaining
Student: 2 eating a slice
Only 0 / 5 remaining

Delivering a Pizza
Wake up all students

Student: 8 eating a slice
Only 4 / 5 remaining
^C
chiku@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/os_lab/lab-7$
```

-- The End ---