Name : Bramha Nimbalkar

Roll no : 7

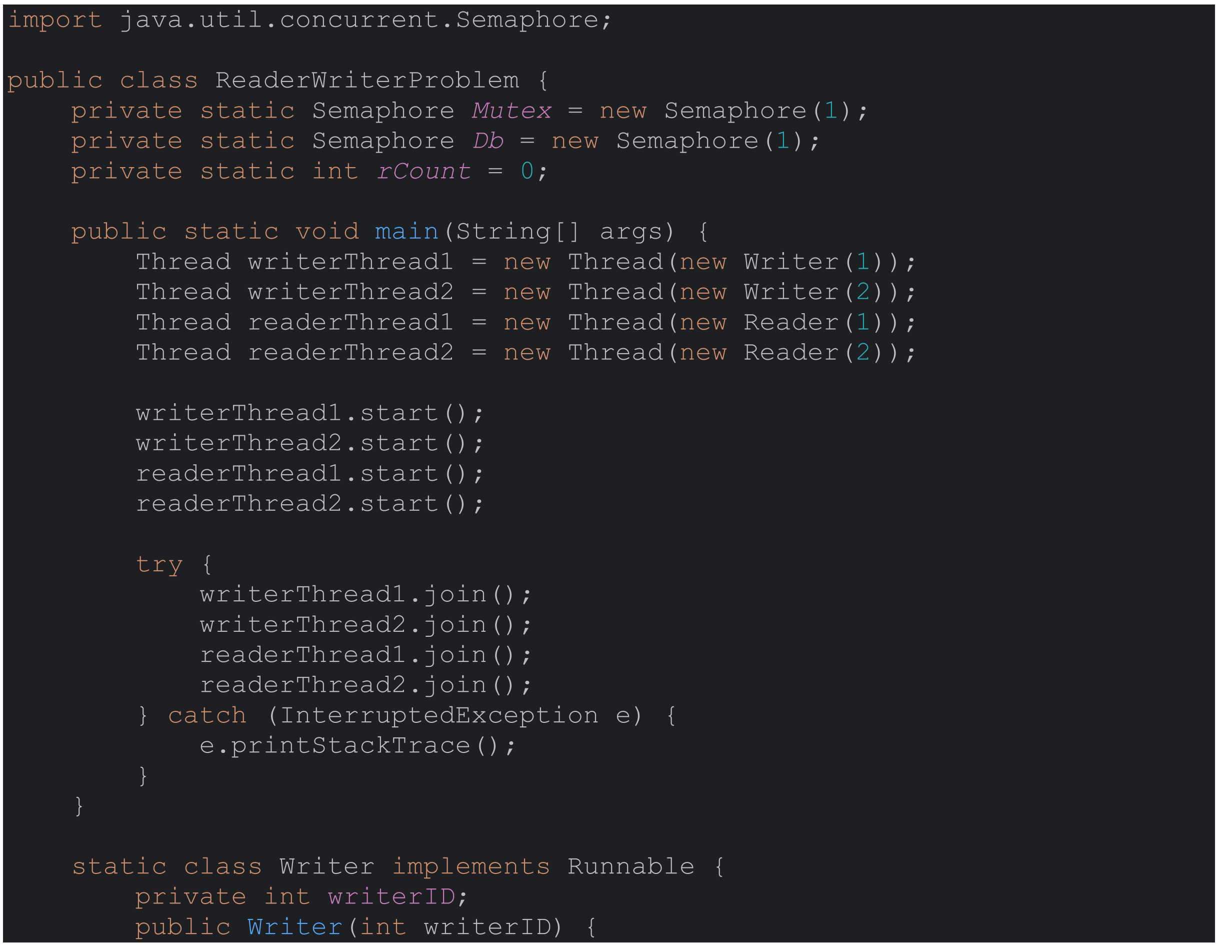
SRN : 202100381

ASSIGNMENT 5

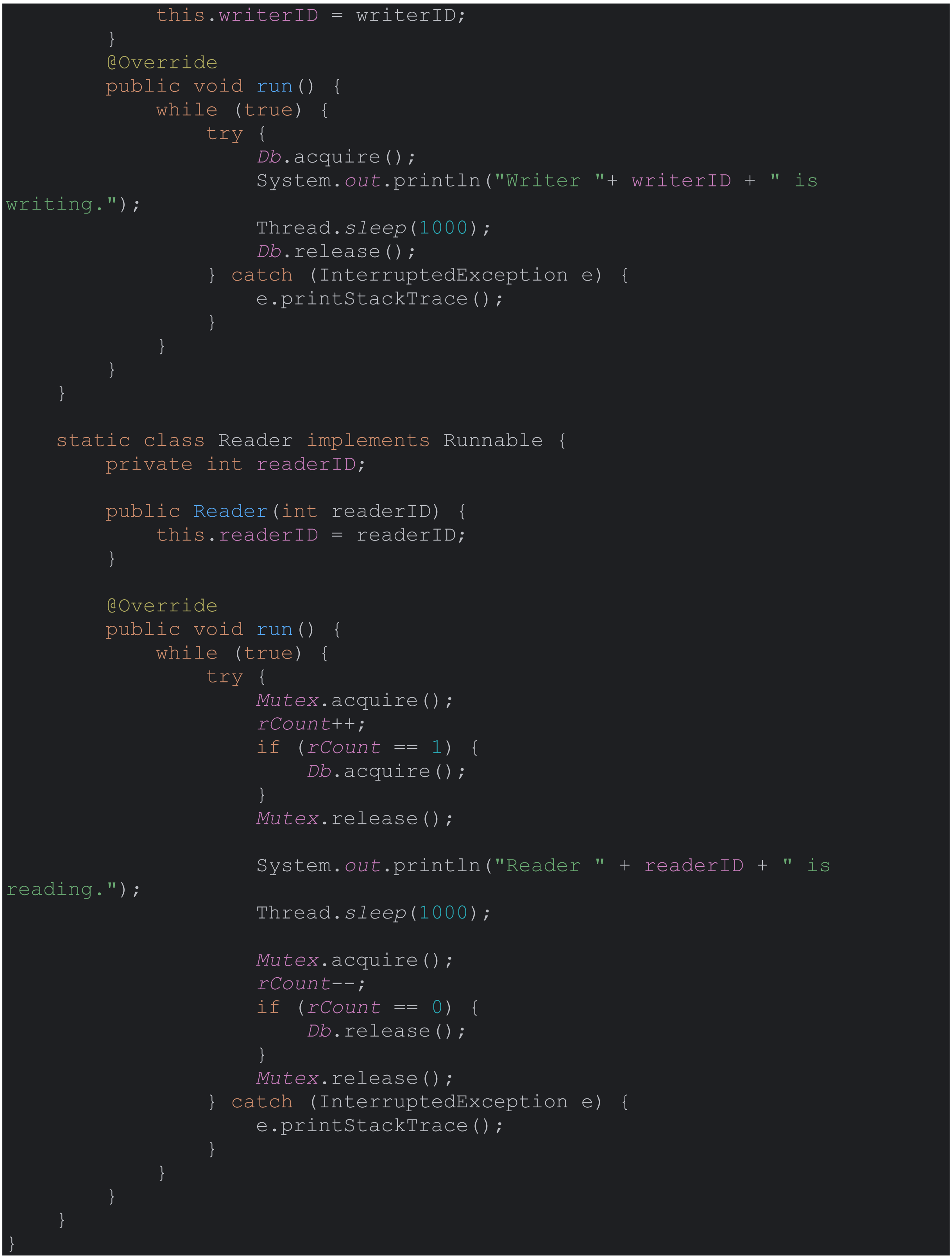
Problem : Implement in C language solution of Reader-Writer problem using

i) Semaphore

ii) Mutex

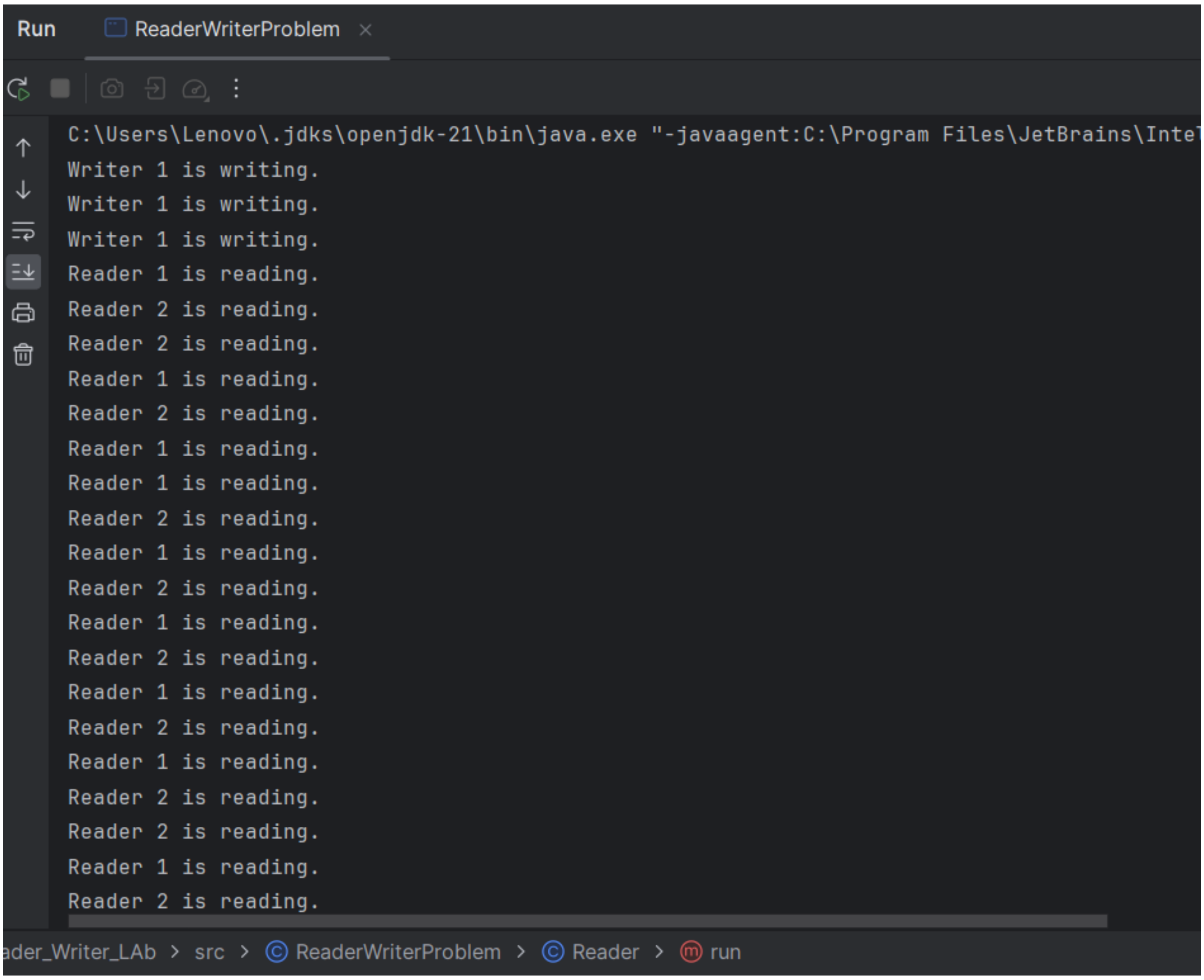


```
import java.util.concurrent.Semaphore;

public class ReaderWriterProblem {
    private static Semaphore Mutex = new Semaphore(1);
    private static Semaphore Db = new Semaphore(1);
    private static int rCount = 0;

    public static void main(String[] args) {
        Thread writerThread1 = new Thread(new Writer(1));
        Thread writerThread2 = new Thread(new Writer(2));
        Thread readerThread1 = new Thread(new Reader(1));
        Thread readerThread2 = new Thread(new Reader(2));

        writerThread1.start();
        writerThread2.start();
        readerThread1.start();
        readerThread2.start();

        try {
            writerThread1.join();
            writerThread2.join();
            readerThread1.join();
            readerThread2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    static class Writer implements Runnable {
        private int writerID;
        public Writer(int writerID) {
```



```
            this.writerID = writerID;
        }
        @Override
        public void run() {
            while (true) {
                try {
                    Db.acquire();
                    System.out.println("Writer "+ writerID + " is 
writing.");
                    Thread.sleep(1000);
                    Db.release();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    static class Reader implements Runnable {
        private int readerID;

        public Reader(int readerID) {
            this.readerID = readerID;
        }

        @Override
        public void run() {
            while (true) {
                try {
                    Mutex.acquire();
                    rCount++;
                    if (rCount == 1) {
                        Db.acquire();
                    }
                    Mutex.release();

                    System.out.println("Reader " + readerID + " is 
reading.");
                    Thread.sleep(1000);

                    Mutex.acquire();
                    rCount--;
                    if (rCount == 0) {
                        Db.release();
                    }
                    Mutex.release();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

Output



```
C:\Users\Lenovo\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Inte
Writer 1 is writing.
Writer 1 is writing.
Writer 1 is writing.
Reader 1 is reading.
Reader 2 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 1 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 2 is reading.
Reader 2 is reading.
Reader 1 is reading.
Reader 2 is reading.
```