

Name : Bramha Nimbalkar

Roll no : 7

SRN : 202100381

## ASSIGNMENT 6

**Problem :** Implement in C language solution of **Dining Philosopher Problem** Using Semaphore

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

public class DiningPhilosopherProblem {
    private static final int num = 5;
    private static final Semaphore[] chopsticks = new Semaphore[num];
    private static final Semaphore limit = new Semaphore(num / 2);

    public static void main(String[] args) {
        for (int i = 0; i < num; i++) {
            chopsticks[i] = new Semaphore(1);
        }

        Thread[] philosophers = new Thread[num];
        for (int i = 0; i < num; i++) {
            philosophers[i] = new Thread(new Philosopher(i));
            philosophers[i].start();
        }
    }

    static class Philosopher implements Runnable {
        private final int id;

        public Philosopher(int id) {
            this.id = id;
        }

        private void think() {
            System.out.println("Philosopher " + id + " is thinking");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        private void eat() {
```

```

        System.out.println("Philosopher " + id + " is eating");
    }
    try {
        Thread.sleep(3000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

@Override
public void run() {
    for (int i = 0; i < 3; i++) {
        think();
        try {
            limit.acquire();
            chopsticks[id].acquire();
            chopsticks[(id + 1) % num].acquire();
            eat();
            chopsticks[(id + 1) % num].release();
            chopsticks[id].release();
            limit.release();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
}
}
}
}

```

## Output

```

Run DiningPhilosopherProblem x
C:\Users\Lenovo\.jdk\openjdk-21\bin\java.exe "-javaagent:C:\Program Fi
Philosopher 3 is thinking
Philosopher 1 is thinking
Philosopher 4 is thinking
Philosopher 2 is thinking
Philosopher 0 is thinking
Philosopher 3 is eating
Philosopher 0 is eating

Process finished with exit code 130

```