Zehra Moğulkoç
110510223

The Dining Philosophers Problem

In this assignment, it is asked to find a deadlock-free solution to the Dining Philosophers problem. Monitor-based solution is used in the code that ensures if both chopsticks are free, then take them up, and the other two philosophers should wait.

- Initially, all philosophers begin in a state of thinking.
- When a philosopher wants to enter the critical section, it checks left and right philosophers in test, if they are thinking, it switches to hungry state.
- It locks the two chopsticks and critical section mutexes, and switches to eating state.
- After eating for a while, it releases mutexes and switches to thinking state.
- According to the calculated dining time, it continues to enter the critical section and exits when it is completely finished.
- Meanwhile, while one philosopher thinks, another philosopher continues to queue up hungry. The code terminates when they are all done.

If the distribution type is “exponential”, time is calculated as:

```
exp((double) (min + max)/ 2);
```

If the distribution type is “uniform”, time is calculated as:

```
Randd = ((double)rand() / RAND_MAX);

rand_time = max * randd;
```

When it terminates, their dining times and hungry times are as follows:

Philosopher 2 waits 133 time in total for hungry time

Philosopher 2 finished its 72 dining time

Philosopher 0 waits 0 time in total for hungry time

Philosopher 0 finished its 84 dining time

Philosopher 3 waits 167 time in total for hungry time

Philosopher 3 finished its 60 dining time

Philosopher 4 waits 162 time in total for hungry time

Philosopher 4 finished its 65 dining time

Philosopher 1 waits 116 time in total for hungry time

Philosopher 1 finished its 50 dining time

Zehra Moğulkoç
110510223

Average Hungry State: 133+0+167+162+116/5 = 115,6

Standard Deviation: sqrt(302,76+13.363+2.641+2.152+0.16/4)= 67.92