

Documentation

Problem 2 (70 points)

The Dining Philosophers problem was invented by E. W. Dijkstra, a concurrency pioneer, to clarify the notions of deadlock and starvation freedom. Imagine five philosophers who spend their lives just thinking and feasting. They sit around a circular table with five chairs. The table has a big plate of rice. However, there are only five chopsticks (in the original formulation forks) available (see Figure 1 of Chapter 1, Exercise 1 from the textbook). Each philosopher thinks. When he gets hungry, he sits down and picks up the two chopsticks that are closest to him. If a philosopher can pick up both chopsticks, he can eat for a while. After a philosopher finishes eating, he puts down the chopsticks and again starts to think.

1. Write a program (Version 1) to simulate the behavior of the philosophers, where each philosopher is a thread and chopsticks are shared objects. Notice that you must prevent a situation where two philosophers hold the same chopstick at the same time.
2. Write a program (Version 2) that modifies Version 1 so that it never reaches a state where philosophers are deadlocked, that is, it is never the case that each philosopher holds one chopstick and is stuck waiting for another to get the second chopstick.
3. Write a program (Version 3) so that no philosopher ever starves.
4. Write a program (Version 4) to provide a starvation-free solution for any number of philosophers N .

IMPLEMENTATION DETAILS:

- If a philosopher acquires one chopstick and then waits until the second one is acquired such condition can create a deadlock and to avoid that once one chopstick is acquired and if second is not available, the philosopher goes back to thinking state leaving the acquired chopstick and tries again later. If available, he starts eating.
- This avoids deadlocks, but the second problem arrives of starvation where each philosopher might not get an equal chance.
- Hence now to avoid starvation anytime the philosopher is to start eating he checks the number of times the neighboring philosopher has eaten, if it is greater only then he starts eating else he goes back to thinking.
- This is the basic logic used in the program.

RESULTS:

Version 1:

```
siddhi@siddhi-VirtualBox:~/Downloads$ gcc dp1.c -pthread
siddhi@siddhi-VirtualBox:~/Downloads$ ./a.out
0 is now thinking.
1 is now thinking.
2 is now thinking.
3 is now thinking.
4 is now thinking.
0 is now hungry.
0 is now eating.
0 is now thinking.
1 is now hungry.
1 is now eating.
4 is now hungry.
3 is now hungry.
2 is now hungry.
0 is now hungry.
1 is now thinking.
1 is now hungry.
0 is now eating.
0 is now thinking.
0 is now hungry.
```

Version 2:

```
3 is now hungry.
3 is now eating.
4 is now hungry.
4 is now eating.
0 is now hungry.
0 is now eating.
3 is now thinking.
2 is now thinking.
4 is now thinking.
1 is now thinking.
0 is now thinking.
4 is now hungry.
4 is now eating.
1 is now hungry.
1 is now eating.
3 is now hungry.
3 is now eating.
2 is now hungry.
2 is now eating.
0 is now hungry.
0 is now eating.
0 is now eating.
siddhi@siddhi-VirtualBox:~/Downloads$ gcc dp2.c -pthread
siddhi@siddhi-VirtualBox:~/Downloads$ ./a.out
```

Version 3:

```
2 is now hungry.
2 is now thinking.
1 is now hungry.
1 is now thinking.
0 is now hungry.
0 is now thinking.
3 is now hungry.
3 is now thinking.
4 is now hungry.
4 is now thinking.
2 is now hungry.
2 is now thinking.
1 is now hungry.
1 is now thinking.
0 is now hungry.
0 is now thinking.
3 is now hungry.
3 is now thinking.
4 is now hungry.
4 is now thinking.
2 is now hungry.
2 is now thinking.
siddhi@siddhi-VirtualBox:~/Downloads$ gcc dp3.c -pthread
siddhi@siddhi-VirtualBox:~/Downloads$ ./a.out
```

Version 4:

```
2 is now hungry.
1 is eating 474 times.
1 is now thinking.
7 is now hungry.
0 is eating 475 times.
0 is now thinking.
5 is eating 475 times.
5 is now thinking.
4 is eating 476 times.
4 is now thinking.
2 is eating 474 times.
2 is now thinking.
6 is eating 442 times.
6 is now thinking.
1 is now hungry.
7 is eating 474 times.
7 is now thinking.
3 is now hungry.
3 is eating 474 times.
3 is now thinking.
siddhi@siddhi-VirtualBox:~/Downloads$ gcc dp4.c -pthread
siddhi@siddhi-VirtualBox:~/Downloads$ ./a.out
Enter number of philosophers:
8
```

Hence from the above screenshots its is clear that all the philosophers are getting a fair chance in the starvation free version.