DINING PHILOSOPHER'S PROBLEM

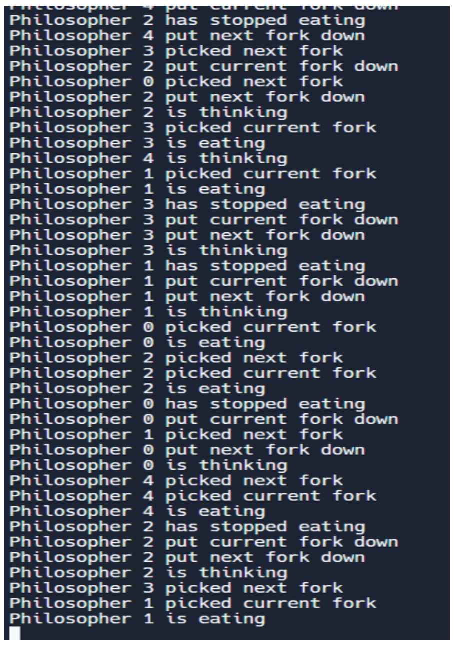
A group pf 5 philosophers are on a table to eat, think, pick fork /bowls and so on. Simultaneously the philosophers must not pick up the same fork since it can lead to what is called a deadlock.

Now, Deadlock describes a condition in which two or more threads are blocked (hung) forever because they are waiting for each other.

PART 1:

In this case since only philosophers , fork , eat and think are involved I used semaphores with ordering of these cases to get the desired output.
Semaphore utilities like sem_wait , sem_post , sem_init were used and after using all the functions an array of threads were created and afterwards joined.

In this case , I implemented for both conditions where philosopher is getting to decide and where forks are in the main role.



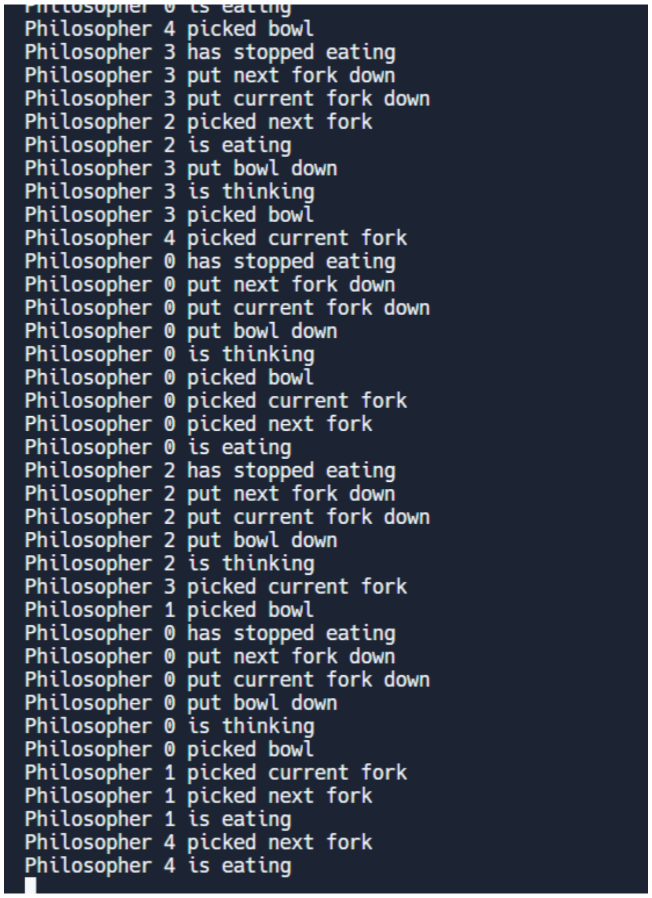
```
Philosopher 4 put current fork down
Philosopher 2 has stopped eating
Philosopher 4 put next fork down
Philosopher 3 picked next fork
Philosopher 2 put current fork down
Philosopher 0 picked next fork
Philosopher 2 put next fork down
Philosopher 2 is thinking
Philosopher 3 picked current fork
Philosopher 3 is eating
Philosopher 4 is thinking
Philosopher 1 picked current fork
Philosopher 1 is eating
Philosopher 3 has stopped eating
Philosopher 3 put current fork down
Philosopher 3 put next fork down
Philosopher 3 is thinking
Philosopher 1 has stopped eating
Philosopher 1 put current fork down
Philosopher 1 put next fork down
Philosopher 1 is thinking
Philosopher 0 picked current fork
Philosopher 0 is eating
Philosopher 2 picked next fork
Philosopher 2 picked current fork
Philosopher 2 is eating
Philosopher 0 has stopped eating
Philosopher 0 put current fork down
Philosopher 1 picked next fork
Philosopher 0 put next fork down
Philosopher 0 is thinking
Philosopher 4 picked next fork
Philosopher 4 picked current fork
Philosopher 4 is eating
Philosopher 2 has stopped eating
Philosopher 2 put current fork down
Philosopher 2 put next fork down
Philosopher 2 is thinking
Philosopher 3 picked next fork
Philosopher 1 picked current fork
Philosopher 1 is eating
```

PART 2:

In this case philosophers, fork, eat , think and bowl are involved and for that we divided into further two cases.

The first case where I took use of a new semaphore bowl created which was initialised in the main along with the existing semaphore in the previous part and after that the function were implemented using sleep and also using the new semaphore.

In second case, I didnt use the bow semaphore and tried to implement it using the only semaphore available like the the first part and also used many sleep calls to make the function wait without going into deadlock.



```
Philosopher 0 is eating
Philosopher 4 picked bowl
Philosopher 3 has stopped eating
Philosopher 3 put next fork down
Philosopher 3 put current fork down
Philosopher 2 picked next fork
Philosopher 2 is eating
Philosopher 3 put bowl down
Philosopher 3 is thinking
Philosopher 3 picked bowl
Philosopher 4 picked current fork
Philosopher 0 has stopped eating
Philosopher 0 put next fork down
Philosopher 0 put current fork down
Philosopher 0 put bowl down
Philosopher 0 is thinking
Philosopher 0 picked bowl
Philosopher 0 picked current fork
Philosopher 0 picked next fork
Philosopher 0 is eating
Philosopher 2 has stopped eating
Philosopher 2 put next fork down
Philosopher 2 put current fork down
Philosopher 2 put bowl down
Philosopher 2 is thinking
Philosopher 3 picked current fork
Philosopher 1 picked bowl
Philosopher 0 has stopped eating
Philosopher 0 put next fork down
Philosopher 0 put current fork down
Philosopher 0 put bowl down
Philosopher 0 is thinking
Philosopher 0 picked bowl
Philosopher 1 picked current fork
Philosopher 1 picked next fork
Philosopher 1 is eating
Philosopher 4 picked next fork
Philosopher 4 is eating
```