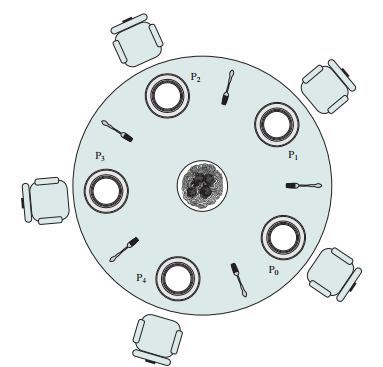
**What is Dining Philosophers Problem?**

There are some Philosophers whose work is just thinking and eating. Let there are 5 (for example) philosophers. They sat at a round table for dinner. To complete dinner each must need two Forks (spoons). But there are only 5 Forks available (Forks always equal to no. of Philosophers) on table. They take in such a manner that, first take left Fork and next right Fork. But problem is they try to take at same time. Since they are trying at same time, Fork 1, 2, 3, 4, 5 taken by Philosopher 1, 2, 3, 4, 5 respectively (since they are left side of each). And each one tries to take right side Fork. But no one found available Fork. And also, that each one thinks that someone will release the Fork and then I can eat. This continuous waiting leads to Dead Lock situation.



**Dining Arrangement**

**Solution:** To solve this Dead Lock situation, Last philosopher (any one can do this) first try to take right side fork and then left side fork. i.e in our example 5th person tries to take 4th Fork instead of 5th one. Since 4th Fork already taken by 4th the person, he gets nothing. But he left 5th Fork. Now the first person will take this 5th Fork and complete dinner and make 1st and 5th available for remaining people. Next 2nd person takes 1st fork and completes and releases 1st and 2nd. This continuous until all finishes dinner.

**Operating System**

In Operating System, this concept used in process synchronization. Same problem but instead of Philosophers processes are there and instead of Forks Resources are there. We follow above solution to avoid dead lock condition.

**Output**

Fork 1 taken by Philosopher 1  
Fork 2 taken by Philosopher 2  
Fork 3 taken by Philosopher 3  
Philosopher 4 is waiting for fork 3

Till now num of philosophers completed dinner are 0

Fork 4 taken by Philosopher 1  
Philosopher 2 is waiting for Fork 1  
Philosopher 3 is waiting for Fork 2  
Philosopher 4 is waiting for fork 3

Till now num of philosophers completed dinner are 0

Philosopher 1 completed his dinner  
Philosopher 1 released fork 1 and fork 4  
Fork 1 taken by Philosopher 2  
Philosopher 3 is waiting for Fork 2  
Philosopher 4 is waiting for fork 3

Till now num of philosophers completed dinner are 1

Philosopher 1 completed his dinner  
Philosopher 2 completed his dinner  
Philosopher 2 released fork 2 and fork 1  
Fork 2 taken by Philosopher 3  
Philosopher 4 is waiting for fork 3

Till now num of philosophers completed dinner are 2

Philosopher 1 completed his dinner  
Philosopher 2 completed his dinner  
Philosopher 3 completed his dinner  
Philosopher 3 released fork 3 and fork 2  
Fork 3 taken by philosopher 4

Till now num of philosophers completed dinner are 3

Philosopher 1 completed his dinner  
Philosopher 2 completed his dinner  
Philosopher 3 completed his dinner  
Fork 4 taken by philosopher 4

Till now num of philosophers completed dinner are 3

Philosopher 1 completed his dinner  
Philosopher 2 completed his dinner  
Philosopher 3 completed his dinner  
Philosopher 4 completed his dinner  
Philosopher 4 released fork 4 and fork 3

Till now num of philosophers completed dinner are 4