Dining Philosopher problem

Entites

1. We have table having five philosopher sitting around it
2. We have noodle in between them
3. And we have only 5 forks to eat
4. There are two states of philosophers –
   1. Eating
   2. Thinking
5. For eating, they need two folks compulsory
6. If everyone eats, then we need 10 folks

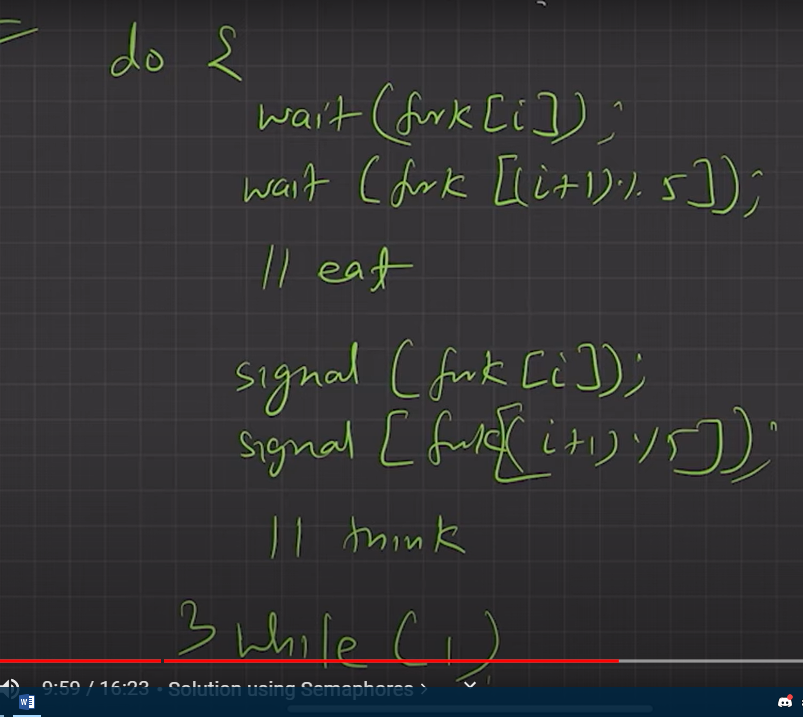
So we need to synchronise these philosphers(process) in such a way they eat in sych with each other without forks(resources)) with each other.

Soln

Using semaphore

1. Each fork – semaphore(binary semaphore)
2. Semaphore fork[5] {1}
3. Wait() -> fork[i] => phi[i] ->acquire
4. Realease() ->fork[i] -> fork -> relaease

Soln for two adjacent is



But if every philosopher gets left fork, there will be a deadlock, p1 will wait for p2, p2-p3, p3-p4,p5-p1. Now this is a deadlock.

We can avoid this by

1. Allow at most 4 ph. To be sitting simultaneously | Problem fifth philosopher is removed
2. Making operation in critiacal section. Allow a ph. To pick up his fork only if both forks are available and to do this, he must pick them up in a critical section (atomically).
3. Odd-even rule. an odd ph. Picks up first his left fork and then his right fork, whereas an even ph. Picks up his right fork then his left fork.