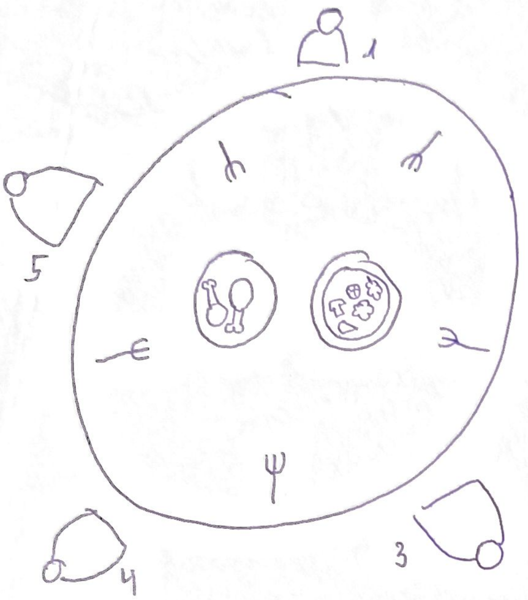
Nguyễn Minh Đức - ITITIU21045

DINING PHILOSOPHERS PROBLEM



Philosopher State

Thinks ↓

- Does not interact with his colleagues

Eats ↓

- Tries to pick up 2 forks closest to him (left & right), but just pick up ~~one~~ 1 at a time.
- Can not pick up the fork that already picked before.
- When a philosopher has 2 forks, he will eat without releasing the forks. After finished, he puts down the forks and starts thinking.

Semaphores:

- A philosopher tries to grab a fork by executing a wait() operation on that semaphore
- He releases his forks by executing the signal() operation on the appropriate semaphores.

→ The shared data are $\boxed{\text{semaphore chopsticks[5];}}$ where all elements of chopstick are initially 1

The structure of ~~sem~~ philosopher

```
do {
    wait (chopsticks[i]);
    wait (chopsticks[(i+1) % 5]);
    ...
    // eat
    signal (chopsticks[i]);
    signal (chopsticks[(i+1) % 5]);
    // think
} while (TRUE);
```

*** Deadlock:** 2 philosophers ~~eat~~ that sits next to each other eat simultaneously because ~~when~~ one chopstick may be held by one and the other cannot grab a chopstick, thus he will not able to eat.

- Suppose all 5 philosophers became hungry and each grabs their left chopstick → All elements of chopstick will now be equal to 0.

When each tries to grab their right chopstick, he will be delayed forever.

⇓

$\boxed{\text{Deadlock}}$

*** Possible remedies:**
- Allow a philosopher to pick up his chopsticks only if both chopsticks are available
- Use asymmetric solution, that is an odd philosopher picks up his first ~~chop~~ left and then right; whereas an even philosopher picks right and then left.