Module 4: Threads in Go
Topic 3.3: Dining Philosophers

Dining Philosophers Problem

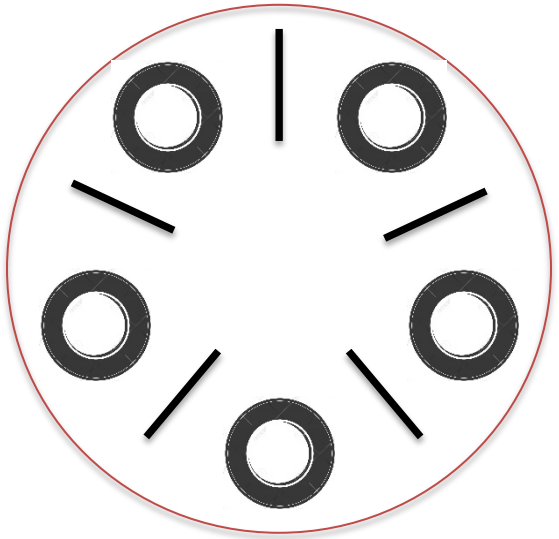
- Classic problem involving concurrency and synchronization

Problem

- 5 philosophers sitting at a round table
- 1 chopstick is placed between each adjacent pair
- Want to eat rice from their plate, but needs two chopsticks
- Only one philosopher can hold a chopstick at a time
- Not enough chopsticks for everyone to eat at once

UCI Division of Continuing Education

Dining Philosopher Issues



- Each chopstick is a mutex
- Each philosopher is associated with a goroutine and two chopsticks

UCI Division of Continuing Education

Chopsticks and Philosophers

```
type ChopS struct{ sync.Mutex }

type Philo struct {
   leftCS, rightCS *ChopS
}
```

UCI Division of Continuing Education

Philosopher Eat Method

```
func (p Philo) eat() {
    for {
        p.leftCS.Lock()
        p.rightCS.Lock()

        fmt.Println("eating")

        p.rightCS.Unlock()
        p.leftCS.Unlock()
    }
}
```

UCI Division of Continuing Education

Initialization in Main

```
CSticks := make([]*ChopS, 5)
for i := 0; i < 5; i++ {
   CSticks[i] = new(ChopS)
}
philos := make([]*Philo, 5)
for i := 0; i < 5; i++ {
   philos[i] = &Philo{Csticks[i],
                      Csticks[(i+1)%5]}
}
```

- Initialize chopsticks and philosophers
- Notice `(i+1)%5`

Start the Dining in Main

```
for i := 0; i < 5; i++ {
    go philos[i].eat()
}
```

- Start each philosopher eating
- Would also need to Wait in the main

UCI Division of Continuing Education

Deadlock Problem

- All philosophers might lock their left chopsticks concurrently
- All chopsticks would be locked
- Noone can lock their right chopsticks

```
p.leftCS.Lock()
p.rightCS.Lock()
fmt.Println("eating")
p.rightCS.Unlock()
p.leftCS.Unlock()
```

UCI Division of Continuing Education

Deadlock Solution

- Each philosopher **picks up lowest numbered chopstick first**

```
philos[i] = &Philo{Csticks[i],
                   Csticks[(i+1)%5]}
```

- Philosopher 4 picks up chopstick 0 before chopstick 4
- Philosopher 4 blocks allowing philosopher 3 to eat
- No deadlock, but Philosopher 4 may starve

UCI Division of Continuing Education