

3.哲学家问题

思路：

让哲学家在请求获取资源时同时获取2个，或者都不获取。

实现方法为加一个最外层的锁mutex_all

拿筷子前需请求锁mutex_all

拿起两个筷子后再释放锁mutex_all

主要代码：

```
1 void pickup(int philosopherNumber) {
2     /*Your code here*/
3     pthread_mutex_lock(&mutex_all);
4     pthread_mutex_lock(&chopsticks[philosopherNumber]);
5     pthread_mutex_lock(&chopsticks[(philosopherNumber+1)%5]);
6     pthread_mutex_unlock(&mutex_all);
7 }
```

这样保证没有死锁

结果展示：

```
1  Philosopher 2 will think for 3 seconds
2  Philosopher 3 will think for 3 seconds
3  Philosopher 4 will think for 2 seconds
4  Philosopher 0 will think for 1 seconds
5  Philosopher 1 will think for 3 seconds
6  Philosopher 0 will eat for 3 seconds
7  Philosopher 0 will think for 3 seconds
8  Philosopher 4 will eat for 1 seconds
9  Philosopher 4 will think for 1 seconds
10 Philosopher 3 will eat for 2 seconds
11 Philosopher 3 will think for 1 seconds
12 Philosopher 2 will eat for 2 seconds
13 Philosopher 2 will think for 3 seconds
14 Philosopher 1 will eat for 1 seconds
15 Philosopher 4 will eat for 1 seconds
16 Philosopher 1 will think for 1 seconds
17 Philosopher 4 will think for 2 seconds
18 Philosopher 0 will eat for 3 seconds
19 Philosopher 3 will eat for 1 seconds
20 Philosopher 3 will think for 3 seconds
```

设定哲学家i的筷子为 $i, (i+1) \% 5$

不能有序号相邻的哲学家一起eat

结果显示没有错误情况

且程序一直运行，没有产生死锁