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：

```
void pickUp(int philosopherNumber) {
    /*Your code here*/
    pthread_mutex_lock(&mutex_all);
    pthread_mutex_lock(&chopsticks[philosopherNumber]);
    pthread_mutex_lock(&chopsticks[(philosopherNumber+1)%5]);
    pthread_mutex_unlock(&mutex_all);
}
```

这样保证没有死锁

结果展示：

```
Philosopher 2 will think for 3 seconds
Philosopher 3 will think for 3 seconds
Philosopher 4 will think for 2 seconds
Philosopher 0 will think for 1 seconds
Philosopher 1 will think for 3 seconds
Philosopher 0 will eat for 3 seconds
Philosopher 0 will think for 3 seconds
Philosopher 4 will eat for 1 seconds
Philosopher 4 will think for 1 seconds
Philosopher 3 will eat for 2 seconds
Philosopher 3 will think for 1 seconds
Philosopher 2 will eat for 2 seconds
Philosopher 2 will think for 3 seconds
Philosopher 1 will eat for 1 seconds
Philosopher 4 will eat for 1 seconds
Philosopher 1 will think for 1 seconds
Philosopher 4 will think for 2 seconds
Philosopher 0 will eat for 3 seconds
Philosopher 3 will eat for 1 seconds
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