



Lab1 : 死锁



TAs :

郭叙森 宋日辉 古剑锋 赵治豪

管卓群 周晓梅 刘林林 陈胜杰

嵌入式课程群 : 608948783



死锁就是两个或者多个进程，互相请求对方占有的资源。

- 互斥条件：一个资源每次只能被一个进程使用
- 请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放
- 不剥夺条件：进程已获得的资源，在未使用完之前，不能强行剥夺
- 循环等待条件：若干进程之间形成一种头尾相接的循环等待资源关系



```
1 ▼ class A {  
2     synchronized void methodA(B b) {  
3         b.last();  
4     }  
5  
6     synchronized void last() {  
7         System.out.println("Inside A.last()");  
8     }  
9  
10 }  
11  
12 ▼ class B {  
13     synchronized void methodB(A a) {  
14         a.last();  
15     }  
16  
17     synchronized void last() {  
18         System.out.println("Inside B.last()");  
19     }  
20  
21 }  
22
```

关键字 synchronized:

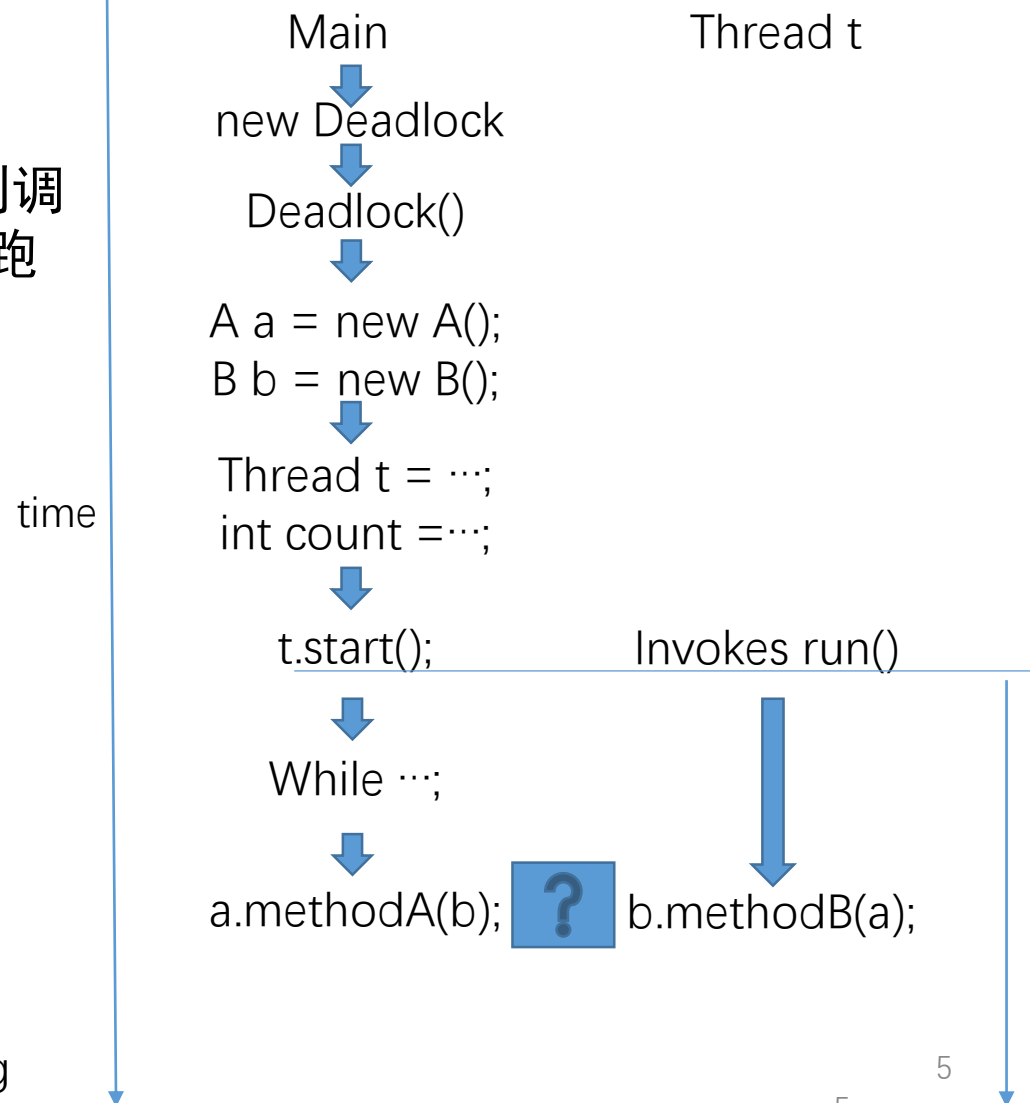
- 当它用来修饰一个方法或者一个代码块的时候，能够保证在同一时刻最多只有一个线程执行该段代码。
- 当一个线程访问object的一个synchronized同步代码块或同步方法时，其他线程对object中所有其它synchronized同步代码块或同步方法的访问将被阻塞。



//关于runnable，你们安卓开发可能已经或将用到
//就是一个线程，会在后台默默的运行
//每次调度到它执行的时候，它就运行run()中的语句

```
class Deadlock implements Runnable{  
    A a=new A();  
    B b=new B();  
  
    //构造函数  
    Deadlock(){  
        Thread t = new Thread(this);  
        int count = 20000;  
  
        t.start();//线程t开始，  
        while(count-->0);//等待20000  
        a.methodA(b);  
    }  
    //runnable运行时调用的方法  
    public void run(){  
        b.methodB(a);  
    }  
    public static void main(String args[]){  
        new Deadlock();  
    }  
}
```

->这里是主函数的时间轴，当
t.start()，之后，线程t就被插入到调
度队列里，当调度到他的时候，就跑
run()里面的代码





流程：

- 1 把上面的代码抄到 Deadlock.java 里面, 保存
- 2 javac Deadlock.java
- 3 Linux 系统 Windows 系统 (把下面这段到记事本里, 然后保存为 .bat, 然后将批处理文件放在 java 程序 (Deadlock.class) 目录下, 双击运行, 观察结果)

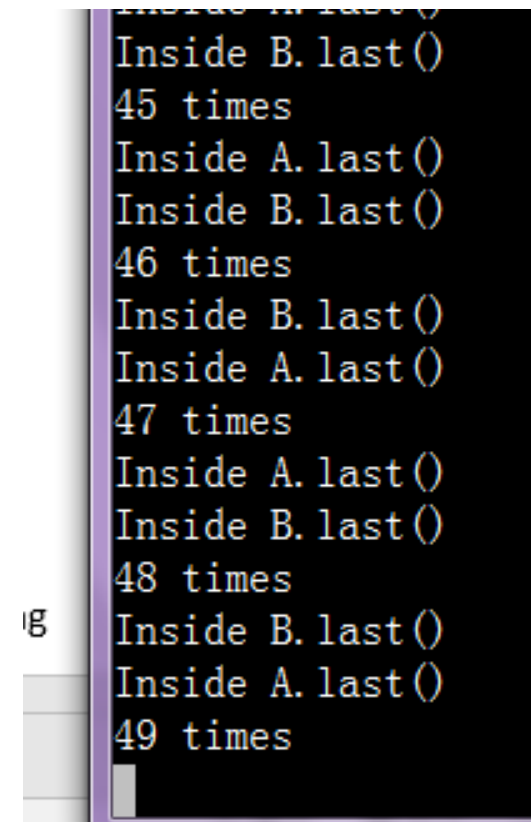
```
#!/bin/bash  
  
for (( c=1; c<=100; c++ ))  
do  
    echo "$c times"  
    java Deadlock  
done
```

```
Deadlock.bat - 记事本  
文件(F) 编辑(E) 格式(O) 查看(V)  
  
cd /d %~dp0  
@echo off  
:start  
set /a var+=1  
echo %var%  
java Deadlock  
if %var% leq 1000 GOTO start  
pause
```



流程：

4. 上一步的3就是让这个程序跑一百遍，通常情况下，跑个几十遍就会停的像这样：跑到第49次就停了，多少次停是随机的，调节程序中count值，使其发生死锁。
5. 截图， 写实验报告，要求写清楚为什么这就产生了死锁。



```
Inside B.last()  
45 times  
Inside A.last()  
Inside B.last()  
46 times  
Inside B.last()  
Inside A.last()  
47 times  
Inside A.last()  
Inside B.last()  
48 times  
Inside B.last()  
Inside A.last()  
49 times
```



应LN老师和RYH老师的要求（hon do ni??），在实验报告，写上死锁产生的四个必要条件。





实验报告要求：

1. 死锁停在第几次的截图
2. 产生死锁的4个必要条件
3. 对上述程序产生死锁的解释

提交日期：待定