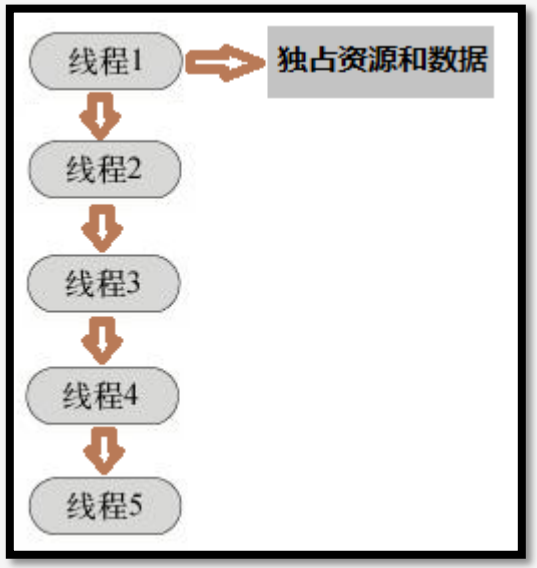
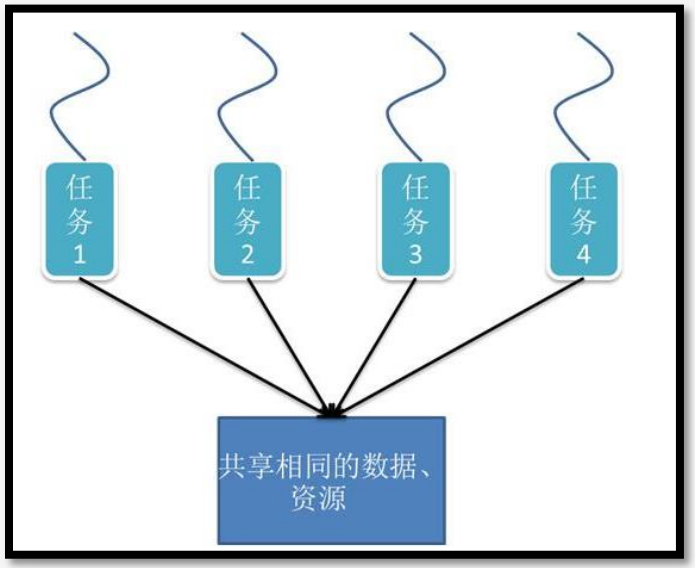
# 学习任务二 开发小游戏

## 子任务2.4 实现线程

# 知识准备：理解多线程

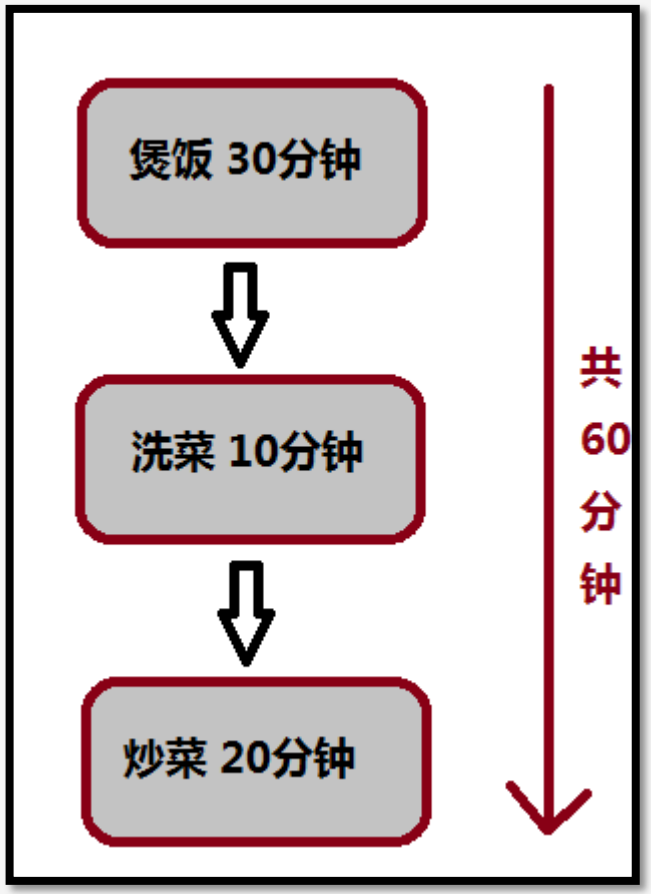



单线程处理事物的方式

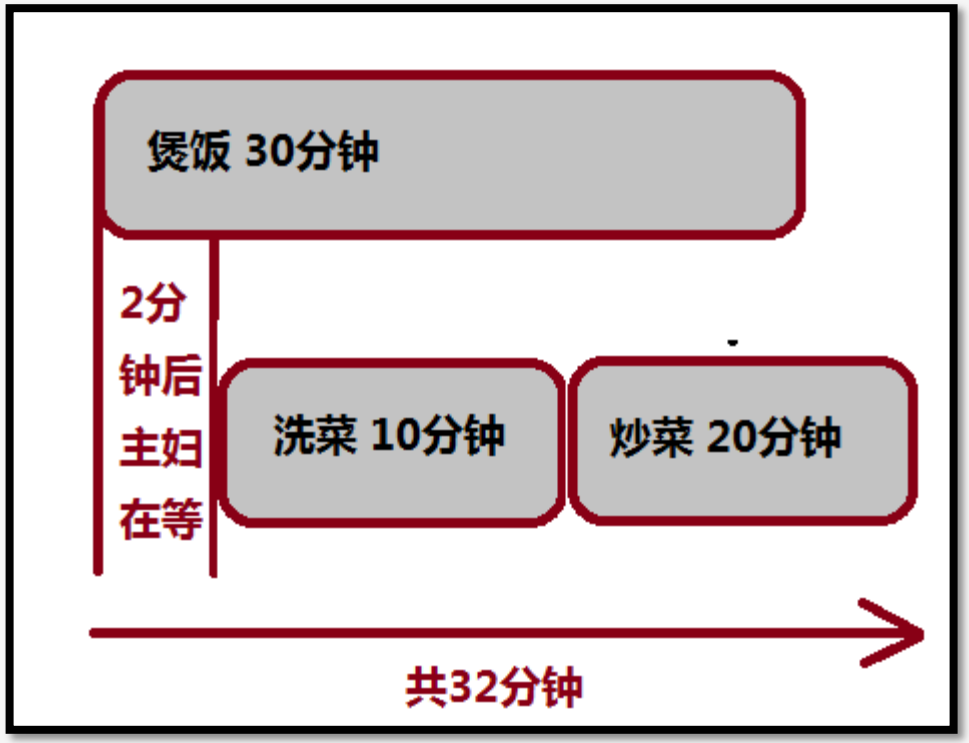



多线程处理事物的方式

# 知识准备：理解多线程

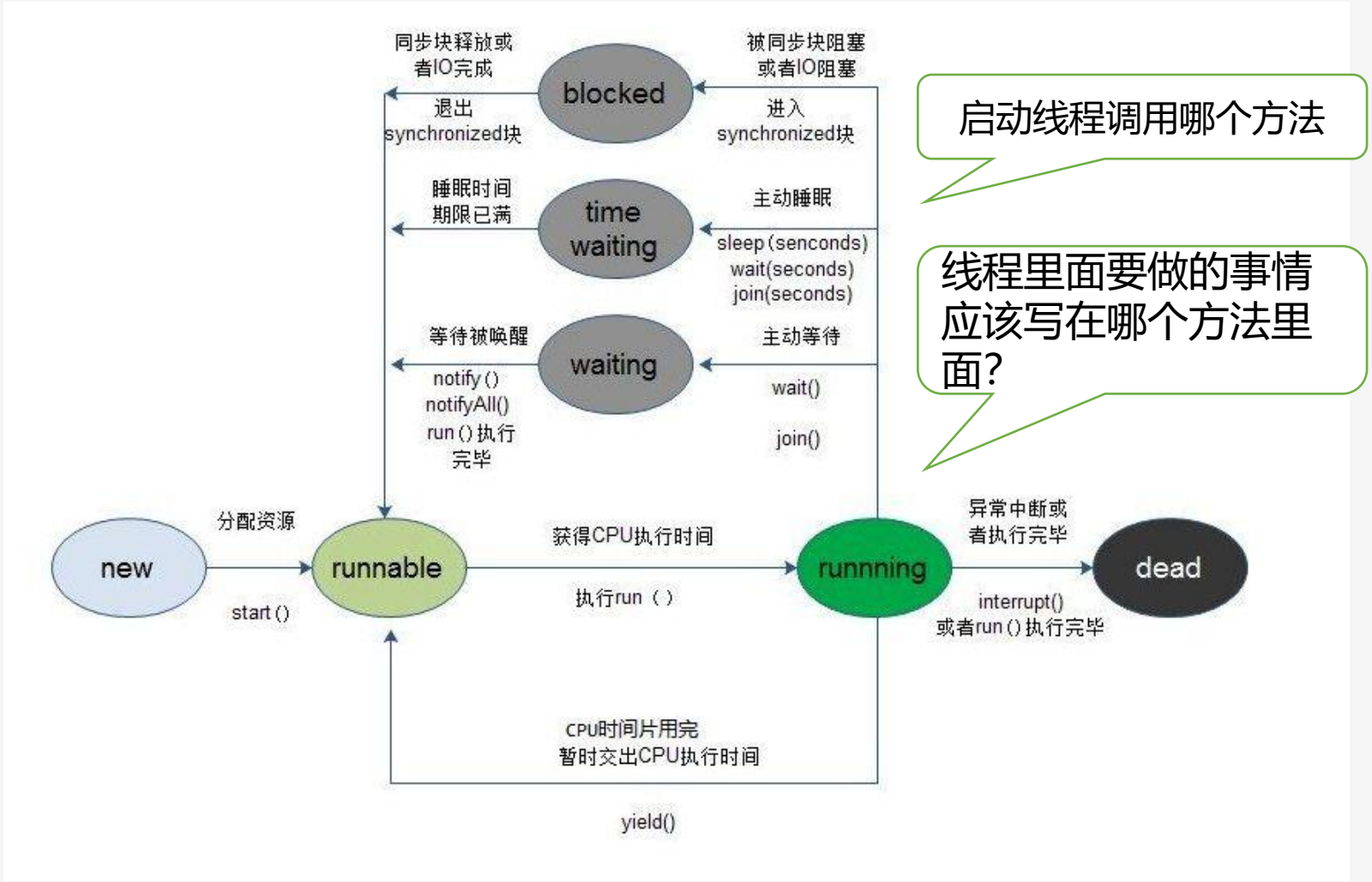



单线程处理事物的方式




多线程处理事物的方式

# 知识准备：理解线程的生命周期




启动线程调用哪个方法

线程里面要做的事情应该写在哪个方法里面?

# 知识准备：线程的实现

- 要么是Thread类的子类
- 要么实现Runnable接口

无论是哪一种实现方法，作为线程的类里面一定要有run方法

# 知识准备：线程的实现 单线程效果演示

```
try {
    System.out.println("听音乐中1");
    Thread.sleep(10);
    System.out.println("听音乐中2");
    Thread.sleep(10);
    System.out.println("听音乐中3");
    Thread.sleep(10);
    System.out.println("听完音乐");

    System.out.println("写论文1");
    Thread.sleep(100);
    System.out.println("写论文2");
    Thread.sleep(100);
    System.out.println("写论文3");
    Thread.sleep(100);
    System.out.println("完成论文");
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

这是调用当前唯一主线程

```
听音乐中1
听音乐中2
听音乐中3
听完音乐
写论文1
写论文2
写论文3
完成论文
```

# 知识准备：线程的实现 线程实现演示

```java
package thread;
public class myThread1 extends Thread{
    public void run(){
        try {
            for(int i=0;i<6;i++){
                System.out.println("听音乐中"+(i+1));
                Thread.sleep(50);
            }
            System.out.println("听完音乐");
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

继承Thread类
是Thread类的子类

实现Runnable接口
拥有线程的核心方法

```java
package thread;
public class myThread2 implements Runnable{
    public void run(){
        try {
            for(int i=0;i<3;i++){
                System.out.println("写论文中"+(i+1));
                Thread.sleep(100);
            }
            System.out.println("完成论文");
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

# 知识准备：线程的实现 多线程效果演示

继承Thread类的可以直接启动

```
package thread;
import java.io.*;
public class myThread {

    public static void main(String args[]){

        Thread t1=new myThread1();
        Thread t2=new Thread(new myThread2());
        t1.start();
        t2.start();

    }
}
```

实现Runnable接口的要先创建一个Thread对象再启动

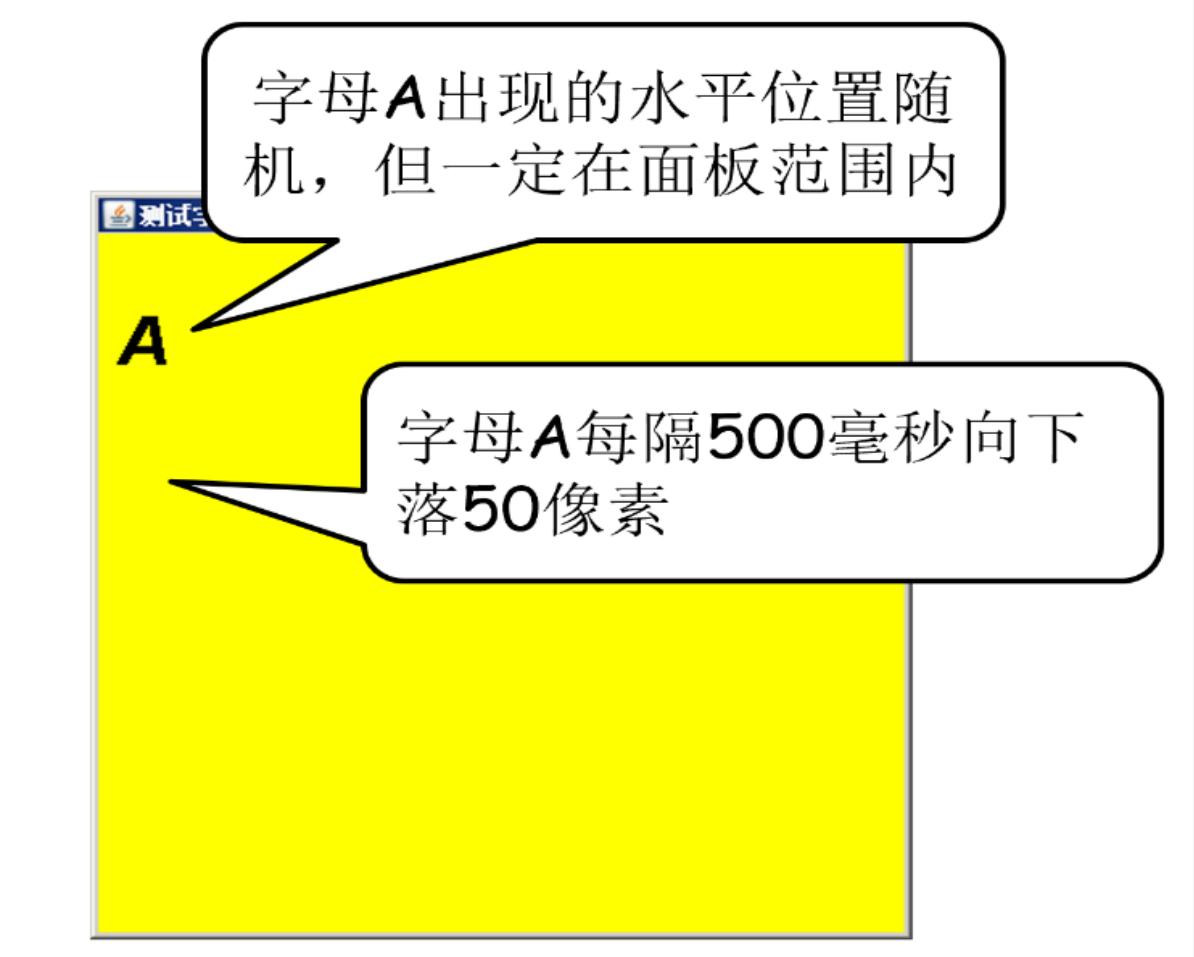
```
写论文中1
听音乐中1
听音乐中2
写论文中2
听音乐中3
听音乐中4
听音乐中5
写论文中3
听音乐中6
完成论文
听完音乐
```

# 课堂训练任务1——完成倒计时线程

# 课堂训练任务2——完成倒计时线程

综合绘图和线程技术
完成一个图形计时器

# 课堂训练任务3——完成游戏界面的一个字母下落

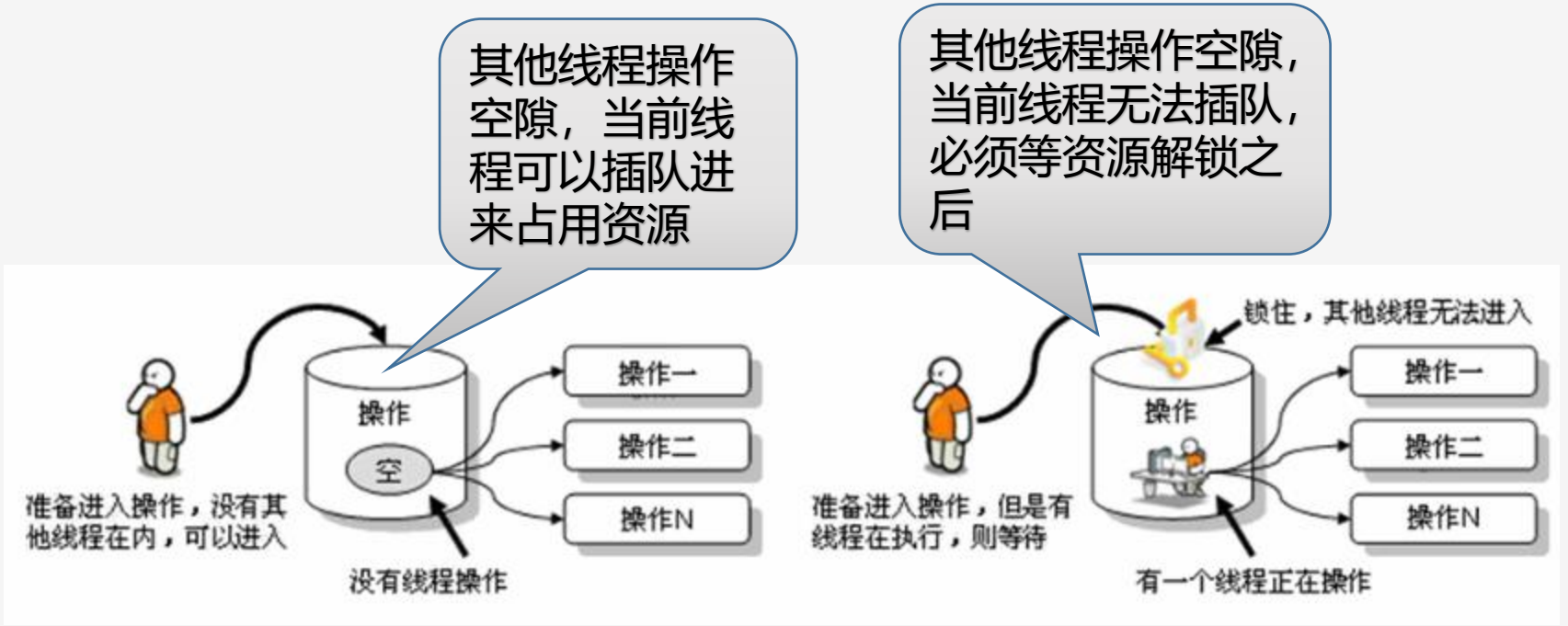


# 其他知识链接——线程的其他状态同步

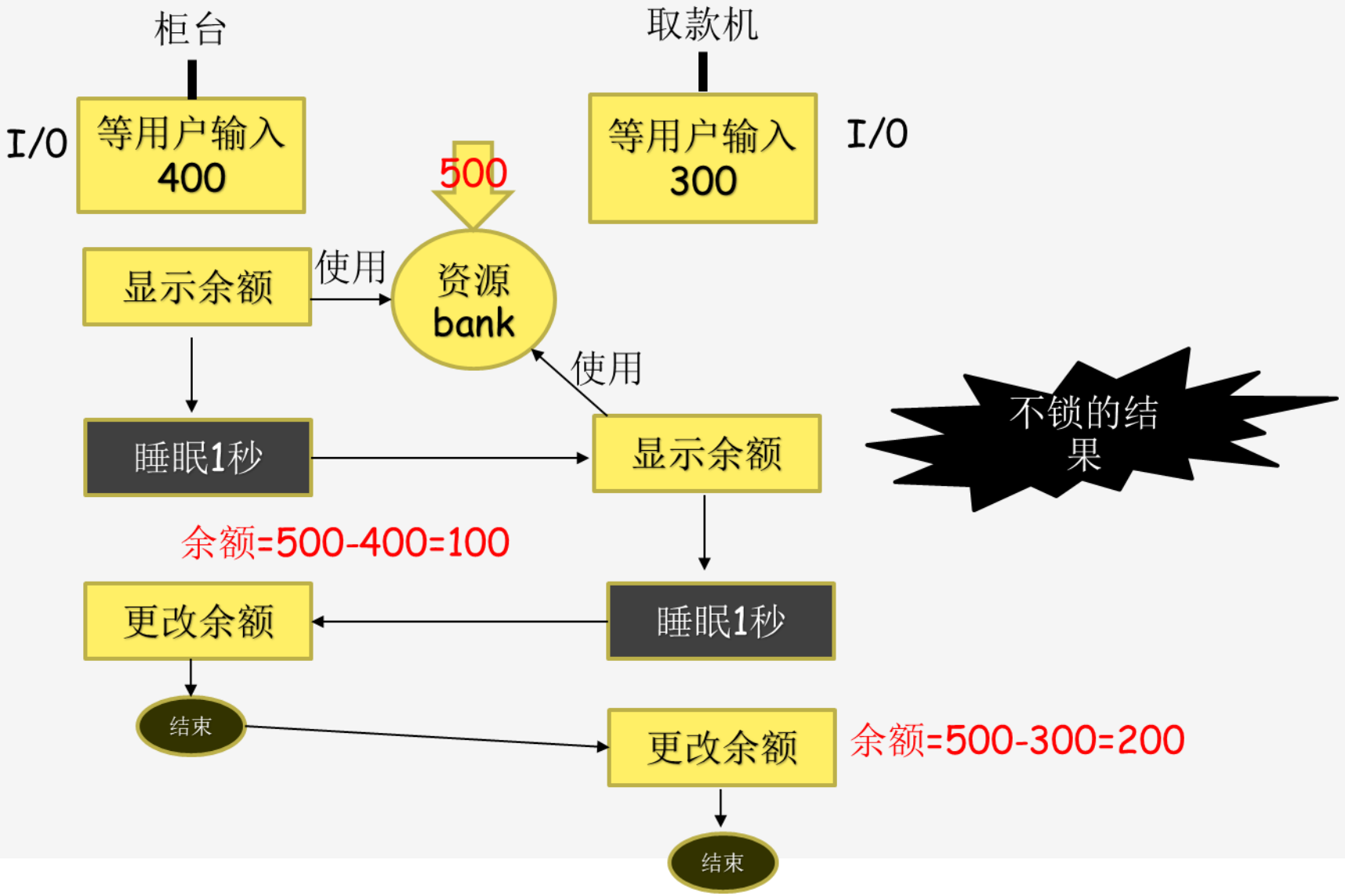
在一般情况下，创建一个线程是不能提高程序的执行效率的，所以要创建多个线程。但是多个线程同时运行的时候可能调用线程函数，在多个线程同时对同一个内存地址进行写入，由于CPU时间调度上的问题，写入数据会被多次的覆盖，所以就要使线程同步。

当有一个线程在对内存进行操作时，其他线程都不可以对这个内存地址进行操作，直到该线程完成操作， 其他线程才能对该内存地址进行操作，而其他线程又处于等待状态，实现线程同步的方法有很多

# 其他知识链接——线程的其他状态同步



# 其他知识链接——线程的其他状态锁机制



# 其他知识链接——线程的其他状态锁机制