

异步运算

1. 什么是异步计算?

异步计算是分布式计算的一种实现方式，开发人员不必花费太多的硬件成本，即可通过软件方式以更低的成本实现大规模运算需要的处理能力。

2. 为什么我们要使用异步计算?

顾名思义就是不通的线程去做不同的事最后在进行数据汇总并返回结果。在一些业务比较复杂，某些方法计算比较耗时的时候单线程就无法快速返回结果用户响应时间变长，这个时候就要使用异步计算来加快数据处理速度传统的 `Runnable` 接口不能返回结果，在JDK1.5之后增加了 `Callable` 接口可以在线程执行完后返回结果。

3、使用Callable+FutureTask实现异步运算

```
public static void main(String[] args) {
    //创建线程池
    ExecutorService es = Executors.newSingleThreadExecutor();
    //创建Callable对象任务
    CallableDemo calTask=new CallableDemo();
    //提交任务并获取执行结果
    Future<Integer> future =es.submit(calTask);
    //关闭线程池
    es.shutdown();
    try {
        Thread.sleep(2000);
        System.out.println("主线程在执行其他任务");

        if(future.get()!=null){
            //输出获取到的结果
            System.out.println("future.get()-->" +future.get());
        }else{
            //输出获取到的结果
            System.out.println("future.get()未获取到结果");
        }
    }
}
```

```
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    System.out.println("主线程在执行完成");  
}
```

```
public class CallableDemo implements Callable<Integer> {  
    private int sum;  
    @Override  
    public Integer call() throws Exception {  
        System.out.println("Callable子线程开始计算啦! ");  
        Thread.sleep(2000);  
  
        for(int i=0 ;i<5000;i++){  
            sum=sum+i;  
        }  
        System.out.println("Callable子线程计算结束! ");  
        return sum;  
    }  
}
```