

Java的多进程运行模式分析 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E7_9A_84_E5_A4_9A_c104_144662.htm 一般我们在java中运行其它类中的方法时，无论是静态调用，还是动态调用，都是在当前的进程中执行的，也就是说，只有一个java虚拟机实例在运行。而有的时候，我们需要通过java代码启动多个java子进程。这样做虽然占用了一些系统资源，但会使程序更加稳定，因为新启动的程序是在不同的虚拟机进程中运行的，如果一个进程发生异常，并不影响其它的子进程。在Java中我们可以使用两种方法来实现这种要求。最简单的方法就是通过Runtime中的exec方法执行java classname。如果执行成功，这个方法返回一个Process对象，如果执行失败，将抛出一个IOException错误。下面让我们来看一个简单的例子。// Test1.java文件import java.io.*.public class Test{ public static void main(String[] args) { FileOutputStream fOut = new FileOutputStream("c:\\Test1.txt"). fOut.close(). System.out.println("被调用成功!"). }}// Test_Exec.javapublic class Test_Exec{ public static void main(String[] args) { Runtime run = Runtime.getRuntime(). Process p = run.exec("java test1"). }} 通过java Test_Exec运行程序后，发现在C盘多了个Test1.txt文件，但在控制台中并未出现"被调用成功！"的输出信息。因此可以断定，Test已经被执行成功，但因为某种原因，Test的输出信息未在Test_Exec的控制台中输出。这个原因也很简单，因为使用exec建立的是Test_Exec的子进程，这个子进程并没有自己的控制台，因此，它并不会输出任何信息。如果要输

出子进程的输出信息，可以通过Process中的getInputStream得到子进程的输出流（在子进程中输出，在父进程中就是输入），然后将子进程中的输出流从父进程的控制台输出。具体的实现代码如下如示：`// Test_Exec_Out.java`

```
import java.io.*;
public class Test_Exec_Out{
    public static void main(String[] args) {
        Runtime run = Runtime.getRuntime();
        Process p = run.exec("java test1");
        BufferedInputStream in = new BufferedInputStream(p.getInputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(in));
        String s;
        while ((s = br.readLine()) != null)
            System.out.println(s);
    }
}
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com