

Java中利用管道实现线程间的通讯 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_Java_E4_B8_AD_E5_88_A9_c104_145656.htm 在Java 语言中，提供了各种各样的输入输出流（stream），使我们能够很方便的对数据进行操作，其中，管道（pipe）流是一种特殊的流，用于在不同线程（threads）间直接传送数据。一个线程发送数据到输出管道，另一个线程从输入管道中读数据。通过使用管道，实现不同线程间的通讯。无需求助于类似临时文件之类的东西。本文在简要介绍管道的基本概念后，将以一个具体的实例pipeapp加以详细说明。

1．管道的创建与使用 Java提供了两个特殊的专门的类专门用于处理管道，它们就是PipedInputStream类和PipedOutputStream类。PipedInputStream代表了数据在管道中的输出端，也就是线程向管道读数据的一端；PipedOutputStream代表了数据在管道中的输入端，也就是线程向管道写数据的一端，这两个类一起使用可以提供数据的管道流。为了创建一个管道流，我们必须首先创建一个PipedOutputStream对象，然后，创建PipedInputStream对象，实例如下：

```
pipeout= new PipedOutputStream(); pipein= new PipedInputStream(pipeout);
```

一旦创建了一个管道后，就可以象操作文件一样对管道进行数据的读写。

2．演示程序：pipeapp 应用程序由三个程序组成：主线程（pipeapp.java）及由主线程启动的两个二级线程（ythread.java和zthread.java），它们使用管道来处理数据。程序从一个内容为一行一行"x"字母的"input.txt"文件中读取数据，使用管道传输数据，第一次是利用线程ythread将数据"x"转换为"y"，最后利用线程zthread

将"y"转换为"z",之后,程序在屏幕上显示修改后的数据。主线程 (pipeapp.java) 在main()方法中,程序首先创建一个应用对象: pipeapp pipeapp=new pipeapp().由于程序中流操作都需要使用IOException异常处理,所以设置了一个try块。在try中,为了从源文件中读取数据,程序为"input.txt"文件创建了一个输入流XfileIn,: fileinputstream xfileIn= new fileinputstream("input.txt").新的输入流传递给changetoy()方法,让线程ythread能读取该文件: inputstream yInpipe =pipeapp.changetoy(xfileIn). changetoy()方法创建将输入数据"x"改变到"y"的线程ythread,并返回该线程的输入管道: inputstream zInpipe = pipeapp.changetoz(yInpipe). changetoz()方法启动将数据从"y"改变到"z"的线程zehread,主程序将使用从changetoz()返回的输入管道。得到以修改的数据。然后,程序将管道输入流定位到datainputstream对象,使程序能够使用readline()方法读取数据: datainputstream inputStream = new datainputstream(zInpiepe). 创建了输入流以后,程序就可以以行一行的读取数据病显示在屏幕上。 String str= inputStream.readline(). While(str!=null) { system.out.println(str). str=inputstream.readline(). } 显示完成之后,程序关闭输入流: inputStream.close(). changetoy()方法 changetoy()方法首先通过传递一个参数inputstream给datainputstream对象来定位资源的输入流,使程序能使用readline()方法从流中读取数据: datainputstream xfileIn =new datainutstream(inputStream); 然后, changetoy()创建输出管道和输入管道: pipeostream pipeout = new pipeoutputstream(). pipeinputstream pipeIn = new pipedinputsteam(pipeout). 为了能够使用println()方法输出修改

的后的文本行到管道，程序将输出管道定位到printstream对象：
printstream printstream = new printstream(pipeout). 现在，程序可以创建将数据从x改变到y的线程，该线程是ythread类的一个对象，他传递两个参数：输入文件（xfileIn）和输出管道（调用printstream）
ythread ythread = new thread(xfileIn, printstream). 之后，程序启动线程：changetoz（）方法
changetoz()方法与changetoy()方法很相似，他从changetoy()返回的输入流开始：
datainputstream yfileIn = new datainputstream(inputstream). 程序创建一个新的管道：
pipedoutputstream pipeout2 = new pipedoutputstream().
pipedinputstream pipeIn2 = new pipedinputstream(pipeout2). 该线程通过这个新的管道发出修改后的数据（输入流pipeIn2）给主程序。 源程序如下：
////pipeapp.java-
pipeapp的主应用程序
//import java.io.*
class pipeapp{
public static void main(string[] args){
pipeapp pipeapp = new pipeapp().try{
fileinputstream xfile = new fileinputstream("input.txt").
inputstream yInpipe = pipeapp.changetoy(xfileIn).
inputstream zInpipe = pipeapp.changetoz(yInpipe).
system.out.println().system.out.println("here are the results").
system.out.println().
datainputstream inputStream = new datainputstream(zInpipe).
string str = inputStream.readLine().while (str!=null){
system.out.println(str).str = inputStream.readLine().}
inputstream.close().}
catch(exception e){
system.out.println(e.toString()).}
}
public inputstream changetoy(inputstream inputStream){
try{
datainputstream pipeout = new datainputstream(inputStream).
pipedoutputstream pipeout = new

```
pipedReaderStream().pipeTo(pipedWriterStream)
pipedReaderStream(pipeOut).printStream(pipeOut)
printStream(pipeOut).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
```