



2



原创 Java高并发之线程通信(管道)

2018-04-11 18:04:12 Hu_sin 阅读数 905 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/HUXINY/article/details/79900479>

Java高并发之线程通信(管道)

Java管道原理：

广义上讲，管道就是一个用来在两个实体之间单项数据传输的导管。Java中的管道和linux中的管道是一样的，从本质上说，管道也是一种文件。

实际上原理很简单，如下图所示：



管道流特点：

1. 管道数据流向的单向的，数据只能从一个进程(线程)流向另一个进程(线程)，如果要进行双向通信，必须建立两个管道。
2. 管道的读数据是一次性操作，数据一旦被读，它就从管道中被抛弃，释放空间以便写更多数据。
3. 当管道输出流write()导致管道缓冲区变满时，管道的write()调用将默认的被阻塞，等待缓冲区的数据被读取。同样的读进程也可能工作得比写进程块。当数据被读取时，管道变空。当这种情况发生时，一个随后的read()调用将默认被阻塞，等待缓冲区数据，这解决了read()调用返回文件结束的问题。
4. 管道输出流或者管道输入流的提前关闭，不会影响到对端流。比如输出流提前结束，输入流不会产生异常；输入流的提前结束也不会影响到输出流。

管道使用实例：

注意：管道输出流输出结束后一定要关闭输出流，因为只有关闭输出流之后，对端输入流read()才能返回null，如果没有关闭输出流会发生什么呢？

当输出流使用后关闭时：

```
1 public class test {
2     public static class Write extends Thread{
3         public PipedOutputStream pos;
4         Write(PipedOutputStream pos){
5             this.pos = pos;
6         }
7         public void run(){
8             PrintStream p = new PrintStream(pos);
9             for(int i=1;i<10;i++){
10                 p.println("hello");
11                 p.flush();
12             }
13             p.close();
14         }
15     }
16 }
```



```

14         }
15     }
16
17     public static class Read extends Thread{
18         public PipedInputStream pis;
19         public String line = "null";
20         Read(PipedInputStream pis){
21             this.pis = pis;
22         }
23         public void run(){
24             BufferedReader r = new BufferedReader(new InputStreamReader(pis));
25             try {
26                 while(line!=null){
27                     line = r.readLine();
28                     System.out.println(getName()+": "+line);
29                 }
30                 r.close();
31             } catch (IOException e) {
32                 e.printStackTrace();
33             }
34         }
35     }
36
37     public static void main(String args[]) throws InterruptedException, IOException{
38         // 创建管道通信流
39         PipedOutputStream pos = new PipedOutputStream();
40         PipedInputStream pis = new PipedInputStream(pos);
41         new Write(pos).start();
42         new Read(pis).start();
43     }
44 }

```

输出结果如下:

```

Problems Javadoc Declaration Console
<terminated> test (1) [Java Application] C:\Program Files (x86)
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: hello
Thread-1: null

```

可以看到最后管道输入流返回的是null。

当输出流使用完后不关闭时:

```

1 public class test {
2     public static class Write extends Thread{
3         public PipedOutputStream pos;
4         Write(PipedOutputStream pos){
5             this.pos = pos;
6         }
7         public void run(){
8             PrintStream p = new PrintStream(pos);
9             for(int i=1;i<10;i++){
10                 p.println("hello");
11                 p.flush();
12             }
13             //p.close();
14         }
15     }
16
17     public static class Read extends Thread{
18         public PipedInputStream pis;
19         public String line = "null";
20         Read(PipedInputStream pis){
21             this.pis = pis;
22         }
23         public void run(){

```

```

24         BufferedReader r = new BufferedReader(new InputStreamReader(pis));
25         try {
26             while(line!=null){
27                 line = r.readLine();
28                 System.out.println(getName()+" : "+line);
29             }
30             r.close();
31         } catch (IOException e) {
32             e.printStackTrace();
33         }
34     }
35 }
36
37 public static void main(String args[]) throws InterruptedException, IOException{
38     // 创建管道通信流
39     PipedOutputStream pos = new PipedOutputStream();
40     PipedInputStream pis = new PipedInputStream(pos);
41     new Write(pos).start();
42     new Read(pis).start();
43 }
44 }

```

输出结果如下：

```

Thread-1: hello
Thread-1: hello
Thread-1: hello
java.io.IOException: Write end dead
    at java.io.PipedInputStream.read(Unknown Source)
    at java.io.PipedInputStream.read(Unknown Source)
    at sun.nio.cs.StreamDecoder.readBytes(Unknown Source)
    at sun.nio.cs.StreamDecoder.implRead(Unknown Source)
    at sun.nio.cs.StreamDecoder.read(Unknown Source)
    at java.io.InputStreamReader.read(Unknown Source)
    at java.io.BufferedReader.fill(Unknown Source)
    at java.io.BufferedReader.readLine(Unknown Source)
    at java.io.BufferedReader.readLine(Unknown Source)
    at test$Read.run(test.java:35)

```

可以看到输入流没有返回null，而是返回一个异常：Write end dead。

所以当管道通信结束后一定要关闭管道输出流。输入流可以通过read()方法是否返回null来判断管道通信是否结束。

工程中的管道使用：

上述的管道通信的例子只是用于实验的，在实际工程中我们并不会这样使用管道，把管道的两端(即管道输出流和管道输入流)设置为全局可见的是不安全的道泄漏的问题。

我们看下面的代码：

```

1 public class test {
2     public static class Write extends Thread{
3         public PipedOutputStream pos;
4         Write(PipedOutputStream pos){
5             this.pos = pos;
6         }
7         public void run(){
8             PrintStream p = new PrintStream(pos);
9             for(int i=1;i<1000;i++){
10                 p.println("hello");
11                 p.flush();
12             }
13             p.close();
14         }
15     }
16
17     public static class Read extends Thread{
18         public PipedInputStream pis;
19         public String line = "null";
20         Read(PipedInputStream pis){
21             this.pis = pis;
22         }
23         public void run(){
24             BufferedReader r = new BufferedReader(new InputStreamReader(pis));
25             try {
26                 while(line!=null){
27                     line = r.readLine();

```

```

28         System.out.println("read: "+line);
29     }
30     r.close();
31     } catch (IOException e) {
32         e.printStackTrace();
33     }
34 }
35 }
36
37 public static class Other_Thread extends Thread{
38     public PipedInputStream pis;
39     public String line = "null";
40     Other_Thread(PipedInputStream pis){
41         this.pis = pis;
42     }
43     public void run(){
44         BufferedReader r = new BufferedReader(new InputStreamReader(pis));
45         try {
46             while(line!=null){
47                 line = r.readLine();
48                 System.out.println("Other_Thread: "+line);
49             }
50             r.close();
51         } catch (IOException e) {
52             e.printStackTrace();
53         }
54     }
55 }
56
57 public static void main(String args[]) throws InterruptedException, IOException{
58     // 创建管道通信流
59     PipedOutputStream pos = new PipedOutputStream();
60     PipedInputStream pis = new PipedInputStream(pos);
61     new Write(pos).start();
62     new Read(pis).start();
63     new Other_Thread(pis).start();
64 }
65 }

```

这里我多创建了一个线程Other_Thread，用来模拟write线程和read线程正常通信的情况下，Other_Thread线程获取了全局的管道输入流，并运行，看看会？

运行结果：

```

read: hello
read: hello
read: hello
read: hello
read: hello
Other_Thread: hello
read: hello
read: hello
read: hello
Other_Thread: hello
read: hello
read: hello
read: hello
read: hello

```

这是运行结果的一个片段，但是已经可以说明问题了，write线程的管道数据被Other_Thread线程截取了，这样的设计会产生管道泄漏问题。

在工程中更科学的利用管道应该如下：

```

1 public class test {
2     public static class Write extends Thread{
3         public PipedOutputStream pos = null;
4
5         // 获取线程中的管道输出流
6         public PipedOutputStream getPos(){
7             pos = new PipedOutputStream();
8             return pos;
9         }
10        @Override
11        public void run(){
12            PrintStream p = new PrintStream(pos);
13            for(int i=1;i<1000;i++){
14                p.println("hello");
15                p.flush();
16            }
17        }
18    }
19 }

```

```

17         } p.close();
18     }
19 }
20
21 public static class Read extends Thread{
22     public PipedInputStream pis = null;
23     public String line = "null";
24
25     // 获得线程中的管道输入流
26     public PipedInputStream getPis(){
27         pis = new PipedInputStream();
28         return pis;
29     }
30     @Override
31     public void run(){
32         BufferedReader r = new BufferedReader(new InputStreamReader(pis));
33         try {
34             while(line!=null){
35                 line = r.readLine();
36                 System.out.println("read: "+line);
37             }
38             r.close();
39         } catch (IOException e) {
40             e.printStackTrace();
41         }
42     }
43 }
44 public static void main(String args[]) throws InterruptedException, IOException{
45     Write write = new Write();
46     Read read = new Read();
47     // 连接两个线程的管道流
48     write.getPos().connect(read.getPis());
49     write.start();
50     read.start();
51 }
52 }

```

在实际工程中一个线程往往需要跟多个线程通信，比如，write线程与read线程通信结束后，还需要write线程与其他线程通信，该怎么办呢？请看如下代码：

```

1 public class test {
2     public static class Write extends Thread{
3         public PipedOutputStream pos = null;
4
5         // 获取线程中的管道输出流
6         public PipedOutputStream getPos(){
7             pos = new PipedOutputStream();
8             return pos;
9         }
10        // 把数据通过管道输出流发送出去
11        public void SentData(){
12            PrintStream p = new PrintStream(pos);
13            for(int i=1;i<10;i++){
14                p.println("hello");
15                p.flush();
16            }
17            p.close();
18        }
19        @Override
20        public void run(){
21            while(true); // 模拟耗时工作
22        }
23    }
24
25    public static class Read extends Thread{
26        public PipedInputStream pis = null;
27        public String line = "null";
28
29        // 获得线程中的管道输入流
30        public PipedInputStream getPis(){
31            pis = new PipedInputStream();
32            return pis;

```

```

33     }
34     //利用管道输入流接收管道数据
35     public void ReceiveData(){
36         BufferedReader r = new BufferedReader(new InputStreamReader(pis));
37         try {
38             while(line!=null){
39                 line = r.readLine();
40                 System.out.println("read: "+line);
41             }
42             r.close();
43         } catch (IOException e) {
44             e.printStackTrace();
45         }
46     }
47     @Override
48     public void run(){
49         while(true);        //模拟耗时工作
50     }
51 }
52
53 public static class Other_Thread extends Thread{
54     public PipedInputStream pis = null;
55     public String line = "null";
56
57     //获得线程中的管道输入流
58     public PipedInputStream getPis(){
59         pis = new PipedInputStream();
60         return pis;
61     }
62     //利用管道输入流接收管道数据
63     public void ReceiveData(){
64         BufferedReader r = new BufferedReader(new InputStreamReader(pis));
65         try {
66             while(line!=null){
67                 line = r.readLine();
68                 System.out.println("Other thread: "+line);
69             }
70             r.close();
71         } catch (IOException e) {
72             e.printStackTrace();
73         }
74     }
75     @Override
76     public void run(){
77         while(true);        //模拟耗时操作
78     }
79 }
80 public static void main(String args[]) throws InterruptedException, IOException{
81     Write write = new Write();
82     Read read = new Read();
83     Other_Thread other = new Other_Thread();
84     //连接两个线程的管道流 ---read和write线程
85     write.getPos().connect(read.getPis());
86     write.start();
87     read.start();
88     other.start();
89     write.SentData();
90     read.ReceiveData();
91     Thread.sleep(2000);
92     //重新连接两个线程的管道流 ---Other_Thread和write线程
93     write.getPos().connect(other.getPis());
94     write.SentData();
95     other.ReceiveData();
96 }
97 }

```

运行结果:



```
read: hello
read: hello
read: hello
read: hello
read: null
Other thread: hello
Other thread: hello
Other thread: hello
Other thread: hello
```

👍
2

🔗

💬

🔖

📱

<

>

这只是输出的一个片段，但是已经能够说明问题了。

有 0 个人打赏

文章最后发布于: 2018

2、3年经验的切图仔，如何把开发思维提前 5 年？

每个前端开发都想面试大厂、带团队，可是大厂技术思维领先你至少5年，想要进击，就得把眼光放到前方，工程化正是你需要的！快狗打车Le...



想对作者说点什么

java 管道(pipe)

阅读数 1695

官方地址：<https://docs.oracle.com/javase/7/docs/api/java/nio/channels/Pipe.html>java管道为运行在同一个JV...

博文 来自： Darklov

Java管道通信

阅读数 614

在学习并发编程的时候，对于任务间的通信有了一些新的认识，由于之前把那个没有接触过管道通信，在《Thinkingi...

博文 来自： 一点思考

Java使用管道实现进程间通讯

阅读数 38

转载请注明出处：<http://blog.csdn.net/jmppok/article/details/175007391>.进程通讯大家都知道进程间通讯有三...

博文 来自： xidajiancun

android 线程间通信几种方式

阅读数 2122

1.共享变量（内存）2.管道3.handle机制runOnUiThread(Runnable)view.post(Runnable)

博文 来自： shunjianopo的博客

Java的通过管道来实现线程通信

阅读数 204

前言与一点思考最近使用golang的channel并发编程非常爽，又快又方便还不用考虑什么线程不安全的问题，同时在...

博文 来自： 写点小东西

Java之路：管道流

阅读数 241

在UNIX/Linux中有一个很有用的概念——管道（pipe），它具有将一个程序的输出当作另一个程序的输入的能力。...

博文 来自： 星云999的博客

对java和c语言之中进程间通信的理解

阅读数 1917

进程间通信：1管道：java里是读写管道文件即可。c语言利用mkfifo创建两个管道文件，java读写管道文件即可。C...

博文 来自： xiongjianze的博客

管道和多线程

阅读数 5165

管道是内核维护的一个缓存,它提供两个fd,从一个fd写入数据,从另一个fd读出数据. 所以它是半双工的. 我在这里讨论...

博文 来自： rainychen的博客

Java多线程——通过管道进行线程间通信

阅读数 33

管道流在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，其中管道流（pip...

博文 来自： 等待新娘的僵尸的...

Java高并发之线程通信(八)---管道 - 畅志聪的专栏 - CSDN博客

java线程通信-等待通知:通过管道进行线程通信 - y41992..._CSDN博客

拿下阿里offer的AI应届生，需要具备什么样的能力？

人工智能的火广大程序员应该都有了解，但是进军AI领域的条件是？

学院 讲师： CSDN

java线程通信---pipe管道

阅读数 3260

java线程通信，pipe管道

博文 来自： 彦帅的博客

java线程通信---pipe管道 - StrideBin的博客 - CSDN博客

Java的通过管道来实现线程通信 - 写点小东西 - CSDN博客

【Java】线程管道通讯

很多操作系统对于管道的吹嘘往往是天花龙凤，好心点的就贴段伪代码给你看，为出书而出书的，就直接一堆概念堆...

2

阅读数 2168

博文 来自: [编程记录, 亲测有效](#)













**Darklovy**
185篇文章
[关注](#) 排名:千里之外

**Going_cc**
36篇文章
[关注](#) 排名:千里之外

**iteye_1287**
0篇文章
[关注](#) 排名:千里之外

**名扬--**
4篇文章
[关注](#) 排名:千里之外

Java高并发学习(一) - HUXINY的博客 - CSDN博客

Java多线程——线程通信 - 喃卿的博客 - CSDN博客

java与C之间的通信

最近正在开发一个基于指纹的音乐检索应用，算法部分已经完成，所以尝试做一个Android App。Android与服务器...

赏

阅读数 2268

博文 来自: [点滴路程](#)

java与c通信实现方案

1、源代码下载java2c2、介绍Java与c通信，最大的问题就是Java数据和c数据的转换问题。最近我做的项目就是java...

阅读数 1473

博文 来自: [祈雨v的博客](#)

Java多线程——通过管道进行线程间通信 - 等待新娘的僵..._CSDN博客

Java高并发学习(六) - HUXINY的博客 - CSDN博客

JAVA多线程之间使用管道进行通讯的小问题

- 问答

反转！“只问了1个框架，就给了35K的Python岗”

学Python的程序员建议收藏！

学院 讲师: CSDN

java 多线程通信之管道流 - weixin_33699914的博客 - CSDN博客

java如何实现进程间的通信？

用SOCKET如何? 管道呢?

论坛

【多线程高并发】线程之间通信

github地址: https://github.com/zhaikaishun/concurrent_programming示例都在Multi_002项目下关键字: 线...

阅读数 1864

博文 来自: [翟开顺](#)

Java里的管道输入流 PipedInputStream与管道输出流 PipedOutputStream

Java里的管道输入流PipedInputStream与管道输出流PipedOutputStream实现了类似管道的功能，用于不同线程之...

阅读数 1万+

博文 来自: [zlp1992的专栏](#)

Java NIO深入理解Pipe(管道)

一-JavaNIO概述JavaNIO管道是2个线程之间的单向数据连接。Pipe有一个source通道和一个sink通道。数据会被写...

阅读数 2317

博文 来自: [yhl_jxy的博客](#)

【Java多线程】——通过管道进行线程间通信

在Java中提供了各种各样的输入输出流stream，我们可以用他们对数据进行方便的操作。其中，管道（pipeStream...

阅读数 365

博文 来自: [名侦探张六儿的博客](#)

年度榜单：Python三连冠，碾压Java！你怎么看？

IEEE Spectrum近日发布了2019年度编程语言排行榜，令人些许意外的是，Python连续三年问鼎巅峰，你怎么看？

学院 讲师: CSDN

Java 如何利用命名管道进行通信？

rt!

论坛

java管道通信

原文地址: java管道通信作者: 花开花落Java提供管道功能，实现管道通信的类有两组: PipedInputStream和Piped...

阅读数 500

博文 来自: [dl429的专栏](#)

消息队列和管道的区别

转载自: <http://bbs.chinaunix.net/viewthread.php?tid=265266>作者: beginner-bj请问管道和消息队列有什么不...



阅读数 3154

博文 来自: [自由的天空](#)

java文件流和管道流

文件流是指专门用于操作数据源中的文件的流，主要有FileInputStream、FileOutputStream、FileReader、FileWriter...

阅读数 769

博文 来自: [xiao_ma_CSDN的...](#)

Java管道流的作用及应用

阅读数 4244

packageio;/*作用*管道流的主要作用就是可以进行两个线程间的通信。一个线程作为管道输出流，另一个线程作为...

博文 来自: shallynever的博客

2

羡慕AI高薪岗！为什么这类程序员不建议大家转型？

被众多开发工程师羡慕的AI程序员为啥这么高薪！30w只是白菜价有啥要求？

学院 讲师: CSDN



阅读数 693

java管道使用

由于java语言的stream严格区分为inputstream和outputstream，流数据读写之间转换一般使用临时文件方式来转...

博文 来自: 龙行天下的专栏

java管道操作，超简单的

//packageIjI002.java.io;importjava.io.*;/* * *@authorAdministrator *管道的基本操作 */publicclassPipeDemo{ ...

博文 来自: lijianl的专栏



阅读数 23万+

java系统高并发解决方案（转载）

转载博客地址：http://blog.csdn.net/zxl333/article/details/8454319一个小型的网站，比如个人网站，可以使用...

博文 来自: jimmy609的专栏



阅读数 2万+

Java高并发秒杀API(四)之高并发优化

Java高并发秒杀API(四)之高并发优化1.高并发优化分析关于并发发性上不去是因为当多个线程同时访问一行数据时...

博文 来自: 不积跬步，无以至...

java多线程通信（管道通信）

阅读数 1081

管道流类PipedInputStream类和PipedOutputStream类用于在应用程序中创建管道通信。一个PipedInputStream...

博文 来自: gchichafengyun的...

一本可陪伴一辈子的笔记本

可擦可写，可循环利用，支持OCR识别，让你的笔记本变得智能

学院 讲师: CSDN

java多线程之任务间管道通信

阅读数 3649

packagecom.eshroe.sweetop.concurrency;importjava.io.IOException;importjava.io.PipedReader;importjava...

博文 来自: 独自登高楼 望断天...

java进程间通讯机制代码----RMI、共享内存、Socket、管道

11-20

java进程间通讯机制代码----RMI、共享内存、Socket、管道，等方式，每种方法我都讲了原理和例子程序，很有参考意义...

下载

Java通过管道进行进程间通信

阅读数 3162

Java语言中提供了各种各样的输入输出流Stream，使我们能够很方便地对数据进行操作，其中管道流（pipeStream...

博文 来自: x_i_y_u_e的专栏

JAVA管道流

阅读数 215

JAVA管道流

博文 来自: 雪村一郎的博客

java之管道流实现线程间的通信(图文版)

阅读数 66

管道流实现线程间的通信(文字)/*作用 *管道流的主要作用就是可以进行两个线程间的通信。一个线程作为管道输出流...

博文 来自: weixin_36898943...

反转！BAT编程吸金榜来了，AI程序员刷爆了.....

2019年BAT等大厂积极布局AI领域，程序员转行学AI的门槛是什么？怎么转？

学院 讲师: CSDN

java 多线程通信之管道流

阅读数 8

/*管道流:PipedInputStreamvoidconnect(PipedOutputStreamsrc)使此管道输入流连接到管道输出流srcPipedOut...

博文 来自: weixin_33699914...

Java多线程之通过管道线程间通信（字节流、字符流），类ThreadLocal与类InheritableThreadLocal的使用

阅读数 921

一、通过管道线程间通信（字节流、字符流）在Java语言中提供了各种各样的输入/输出流Stream，使我们能够...

博文 来自: ChangWen的博客

程序员实用工具网站

阅读数 9万+

目录 1、搜索引擎 2、PPT 3、图片操作 4、文件共享 5、应届生招聘 6、程序员面试题库 7、办公、开发软件 8、高...

博文

我花了一夜用数据结构给女朋友写个H5走迷宫游戏

阅读数 4万+

起因 又到深夜了，我按照以往在csdn和公众号写着数据结构！这占用了我大量的时间！我的超越妹妹严重缺乏陪伴...

博文

别再翻了，面试二叉树看这 11 个就够了~

阅读数 1万+

写在前边 数据结构与算法：不知道你有没有这种困惑，虽然刷了很多算法题，当我去面试的时候，面试官让你手写...

博文

AI实训路线推荐：考虑转型/跳槽程序员免费领取，自测专业程度！

如何能够短时间内抓住技术重点，打造属于自己的“offer收割机”？

学院 讲师: CSDN



让程序员崩溃的瞬间（非程序员勿入）

阅读数 10万+

今天给大家带来点快乐，程序员才能看懂。来源：https://zhuanlan.zhihu.com/p/47066521 1. 公司实习生找 Bug...

博文

接私活必备的 10 个开源项目！

点击蓝色“GitHubDaily”关注我加个“星标”，每天下午 18:35，带你逛 GitHub！作者 | SevDot来源 | http://1t.c...

GitHub开源的10个超棒后台管理面板

目录 1、AdminLTE 2、vue-Element-Admin 3、tabler 4、Gentelella 5、ng2-admin 6、ant-design-pro 7、blu...

100 个网络基础知识普及，看完成半个网络高手

欢迎添加华为云小助手微信（微信号：HWCloud002或HWCloud003），输入关键字“加群”，加入华为云线上技...

VS CODE远程开发入门

在我们办公室，通常配置两台电脑，一台 Windows 主机，主要用于办公、即时通讯，一台 Linux 主机，用于开发。...

中国最顶级的一批程序员，从首富到首负！

过去的20年是程序员快意恩仇的江湖时代通过代码，实现梦想和财富有人痴迷于技术，做出一夜成名的产品有人将技...

为什么面向对象槽透了？

又是周末，编程语言“三巨头”Java, Lisp 和C语言在Hello World咖啡馆聚会。服务员送来咖啡的同时还带来了一张...

分享靠写代码赚钱的一些门路

作者 mezod，译者 josephchang10如今，通过自己的代码去赚钱变得越来越简单，不过对很多人来说依然还是很难...

对计算机专业来说学历真的重要吗？

我本科学校是渣渣二本，研究生学校是985，现在毕业五年，校招笔试、面试，社招面试参加了两年了，就我个人的...

世界上最好的学习法：费曼学习法

你是否曾幻想读一遍书就记住所有的内容？是否想学习完一项技能就马上达到巅峰水平？除非你是天才，不然这是不...

学Linux到底学什么

来源：公众号【编程珠玑】作者：守望先生 网站：https://www.yanbinghu.com/2019/09/25/14472.html 前言 ...

深入理解C语言指针

一、指针的概念 要知道指针的概念，要先了解变量在内存中如何存储的。在存储时，内存被分为一块一块的。每一块...

C语言实现推箱子游戏

很早就想过做点小游戏了，但是一直没有机会动手。今天闲来无事，动起手来。过程还是蛮顺利的，代码也不是非常...

面试官：兄弟，说说基本类型和包装类型的区别吧

Java 的每个基本类型都对应了一个包装类型，比如说 int 的包装类型为 Integer，double 的包装类型为 Double。基...

8000字干货：那些很厉害的人是怎么构建知识体系的

本文约8000字，正常阅读需要15~20分钟。读完本文可以获得如下收益： 分辨知识和知识体系的差别 理解如何用八...

Android完整知识体系路线(菜鸟-资深-大牛必进之路)

前言 移动研发火热不停，越来越多人开始学习Android 开发。但很多人感觉入门容易成长很难，对未来比较迷茫，...

网易云音乐你喜欢吗？你自己也可以做一个

【公众号回复“1024”，免费领取程序员赚钱实操经验】今天我章鱼猫给大家带来的这个开源项目，估计很多喜欢...

C语言这么厉害，它自身又是用什么语言写的？

这是来自我的星球的一个提问：“C语言本身用什么语言写的？”换个角度来问，其实是：C语言在运行之前，得编译...

Python 愤怒的小鸟代码实现(1):物理引擎pymunk使用

python 愤怒的小鸟代码实现（1）：物理引擎pymunk使用游戏介绍 游戏介绍 最近比较忙，国庆正好有时间写了pyt...

前端开发必备网站推荐

本人是一个纯正的小白，在学习的过程中搜集了一些关于前端开发的网站，希望对大家能够有所帮助！授人以鱼不如...

中国物联网激荡20年

故事还要从24年前那个夏天说起。 ** 1 ** 1995年的夏天，美国西海岸有一个中年人出版了一本叫《未来之路》的书...

阅读数 2万+

博文



2

阅读数 1万+

博文



阅读数 5万+

博文



阅读数 381

博文



阅读数 7915

博文



阅读数 2万+

博文

阅读数 1万+

博文

阅读数 1万+

博文

阅读数 1万+

博文

阅读数 1万+

博文

阅读数 3万+

博文

阅读数 3万+

博文

阅读数 1万+

博文

阅读数 2万+

博文

阅读数 4591

博文

阅读数 3592

博文

阅读数 9623

博文

阅读数 7161

博文

阅读数 3954

博文



阅读数 6010

博文

Vue 3.0 源码开放，看看都有哪些新特性

当大多数国人还在庆祝国庆节的时候，尤雨溪大大在昨天凌晨发布了 Vue 3.0 源代码，源码地址：<https://github.co...>

用Matplotlib，妈妈再也不担心我没有表情包斗图了

全文共3003字，预计学习时长6分钟 Netflix、Instagram、YouTube、电影还有电视剧，这些都有什么共同之处呢...

十大优秀编程项目，让你的简历金光闪闪

全文共3241字，预计学习时长6分钟 被问到如何学习编程时，最常听到的问题就是：“有没有什么新项目的好点子？ ...

第二弹！python爬虫批量下载高清图

文章目录前言下载免费高清图下载带水印的精选图代码与总结 前言 在上一篇文章没高质量配图？python爬虫绕...

SpringBoot注解梳理

一、注解(annotations)列表 二、注解(annotations)详解 三、JPA注解 四、springMVC相关注解 五、全局异常处理 ...

结合OpenCV与TensorFlow进行人脸识别

作为新手来说，这是一个最简单的人脸识别模型，难度不大，代码量也不算多，下面就逐一讲解，数据集的准备就...

python入门教程

入门教程:菜鸟教程（这个网站可以说是所有语言快速入门的必备的网站了）<https://www.runoob.com/python3/py...>

数据库基础（常用SQL语句）

一、数据库级及SQL语言简介 1、目前主流数据库 微软：sql server、access 瑞典：mysql ibm: db2 sybase:sybase...

云顶之弈阵容助手-基于遗传算法

LOL云顶之弈人工智能助手概述爬取相关网站内容获取英雄信息阵容搭配与得分遗传算法设计运行结果完整代码——...

2019年10月全国程序员工资统计，一半以上的职位5个月没招到人。

我每个月第一天（也许是第二天，第三天），会爬招聘网站，并在CSDN发布。趋势 全国程序员平均工资走势图。...

2019年10月中国编程语言排行榜

2019年10月2日，我统计了某招聘网站，获得有效程序员招聘数据9万条。针对招聘信息，提取编程语言关键字，并...

Java 网络爬虫，就是这么的简单

这是 Java 网络爬虫系列文章的第一篇，如果你还不知道 Java 网络爬虫系列文章，请参看 学 Java 网络爬虫，需要哪...

面试官，不要再问我三次握手和四次挥手

三次握手和四次挥手是各个公司常见的考点，也具有一定的水平区分度，也被一些面试官作为热身题。很多小伙伴说...

告别AV画质：实时把动画变成4k高清，延时仅3毫秒，登上GitHub趋势榜

栗子 发自 凹非寺量子位 出品 | 公众号 QbitAI看动画(特别是里番)的时候，总会觉得画质不够好，就算已经有1080p...

人是怎么有钱的

近发现，各种信息流广告，一水的，“上门窝囊女婿突然暴露真实身份，是超级富二代。”当然，我只是简单陈述一...

程序员因接外包坐牢 456 天！两万字长文揭露心酸真实经历

点击上方“码农突围”，马上关注，每天早上8:50准时推送真爱，请置顶或星标作者：西畔随云 本文经授权转载自西...

c# 应用r包 c# excel格式刷 c# 监控word保存 c#中字符串排序 c# 打印方向 c# 获取根目录 c#语言文档 c#设置开机自启动 c# cpu id 主板 c# timer越来越卡

阅读数 1万+

博文



2

阅读数 4918

博文



阅读数 9438

博文



阅读数 1万+

博文



阅读数 1万+

博文



阅读数 3335

博文

阅读数 6100

博文

阅读数 3342

博文

阅读数 3386

博文

阅读数 4万+

博文

阅读数 5555

博文

阅读数 4940

博文

阅读数 5万+

博文

阅读数 1948

博文

阅读数 3728

博文

阅读数 6121

博文

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客



Hu_sin

[TA的个人主页 >](#)

私信

关注

原创

59

粉丝

48

获赞

33

评论

26



访问: 4万+

周排名: 4万+

积分: 1091

总排名: 6万+

勋章:  

等级: 博客 4






👍
2



最新文章

Tengine 树莓派 安装及使用
树莓派安装opencv
Csapp 第三章阅读以及思考
笔记---Linux连接
OpenPose 安装和使用

分类专栏

	树莓派	2篇
	arduino	
	python	9篇
	微信跳一跳辅助	1篇
	java	15篇

展开

归档

2019年9月	4篇
2019年8月	1篇
2019年5月	2篇
2019年4月	4篇
2019年3月	3篇
2019年2月	2篇
2019年1月	7篇
2018年11月	1篇

展开

热门文章

Java Spring 入门教程
阅读数 12457
利用Beautifulsoup爬取网页图片
阅读数 4720
Java高并发学习(一)
阅读数 4449
Java NIO介绍及使用
阅读数 3250
arduino串口使用print还是write?
阅读数 2425

最新评论

使用ecryptfs 加密文件



weixin_44329790: 根本装不上, 特么的, 哪抄的帖子吧 ...

Java Spring 入门教程

chon01: 所以, AOP和IOC是什么? ...

Java Spring 入门教程

qq_42611205: 博主也是山科大的? ...

Java高并发学习(一)

wl_ang: suspend()因为不会释放锁, 会导致线程产生环路等待, 造成死锁 ...

VGG_face Caffe 微...

qq_41404455: 你好, 我运行上面的程序学习四次之后就自动killed了请问是怎么回事呢 ...



CSDN学院



CSDN企业招聘

QQ客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉



2

