

字映射实现读取文件行(readline)

9:40

jimodeyangguang

阅读数：1274

标签：

java nio io

更多

码

```
io.BufferedReader;
io.File;
io.FileInputStream;
io.FileNotFoundException;
io.IOException;
io.InputStreamReader;
nio.CharBuffer;
nio.MappedByteBuffer;
nio.channels.FileChannel;
nio.charset.Charset;
nio.charset.CharsetDecoder;
util.Scanner;
util.zip.GZIPInputStream;

Test {
    String path = "D:\\log\\proclog\\loganalysis\\0108161331_CHN-MY-1_1021235501.log";

    static void main(String[] s) throws IOException {
        am();
    }

    static void stream() throws FileNotFoundException, IOException {
        startTime = System.currentTimeMillis();
        BufferedReader reader = getReader(new File(path));

        String line;
        while ((line = reader.readLine()) != null) {
            // 空转

            estimatedTime = System.currentTimeMillis() - startTime;
            System.out.printf("stream Diff: %d ms\n", estimatedTime);
        }

        static BufferedReader getReader(File f) throws FileNotFoundException, IOException {
            BufferedReader reader = null;
            if (f.getName().endsWith(".gz")) {
                reader = new BufferedReader(new InputStreamReader(new GZIPInputStream(new FileInputStream(f))));
            } else {
                reader = new BufferedReader(new InputStreamReader(new FileInputStream(f)));
            }

            return reader;
        }

        static void mem() throws IOException {
            startTime = System.currentTimeMillis();
            FileChannel fc = new FileInputStream(path).getChannel();
            MappedByteBuffer byteBuffer = fc.map(FileChannel.MapMode.READ_ONLY, 0, fc.size());
            Charset charset = Charset.forName("US-ASCII");
            CharsetDecoder decoder = charset.newDecoder();
            CharBuffer charBuffer = decoder.decode(byteBuffer);
            Scanner sc = new Scanner(charBuffer).useDelimiter(System.getProperty("line.separator"));
            while (sc.hasNext()) {
                sc.next();
            }

            estimatedTime = System.currentTimeMillis() - startTime;
            System.out.printf("mem Diff: %d ms", estimatedTime);
        }
    }
}
```



0



	0	

远快于内存映射读取，看来逐行读取文本还是继续使用steam api吧。

B，使用一个100MB的文件，mem方式报内存溢出，有点尴尬，先做个记号吧。

学习路线，你一定不知道的薪资翻倍秘诀

薪，你要掌握学习那些技能

么

MappedByteBuffer，高效文件/内存映射

  9058

java.nio引入的文件内存映射方案，读写性能极高。NIO最主要的就是实现了对异步操作的支持。其中一种通过把...

内存映射读取文件后 按行进行输出

11-22

文件。并且按行输出。

区块链以太坊开发必须会

路线图，月薪4万很轻松

内存映射文件原理与DirectMemory

  1.5万

对于IO 包来说有一个新功能是内存映射文件，日常编程中并不是经常用到，但是在处理大文件时是比较理想的提高效率的手段。本文我主要想结合操作系统中（OS）相关方面的知识介绍一下原...

从内存映射

  444

下一个Java程序是如何在机器上执行的：Java源程序（.java文件）--> Java编译器（如Eclipse）--> 字节码（.class文件）-...

快速处理大文件

  3.5万

利用普通的FileInputStream 或者FileOutputStream 抑或RandomAccessFile 来进行频繁的读写操作，都将导致进程因频繁读写外存而降低速度.如...

快速实现大文件的上传和下载：内存映射文件及文件通道到通道批量传输数据

  5207

文件

快速实现大文件的上传和下载：内存映射文件（Memory Mapped File）或者MappedByteBuffer

  1.4万

始，Java内存映射文件（Memory Mapped Files）就已经在java.nio包中，但它对很多程序开发者来说仍然是一个相当新的概念。引入NIO后，Java IO已...

快速实现大文件的上传和下载：内存映射文件---上G大文件处理

  9881

内存映射的原理及过程，与传统IO进行了对比，最后，用实例说明了结果。...

快速实现大文件的上传和下载：内存映射文件---上G大文件处理

  1289

利用普通的FileInputStream 或者FileOutputStream 抑或RandomAccessFile 来进行频繁的读写操作，都将导致进程因频繁读写外存而降低速度.如...

快速实现大文件的上传和下载：内存映射文件readLine、writeLine

  5367

```
import java.io.BufferedReader; import java.io.File; import java.io.FileInputStream; import java.i...
```

学习java java的~ java java和--

个人资料



jimodeyangguang

关注

原创	粉丝	喜欢	评论
18	20	2	1

等级：

博客 3

访问：2万+

积分：539

排名：10万+

最新文章

Linux下安装Python3.3.0

C语言中，当计算字符数组长度时，用sizeof和strlen 的原理及两者的区别

GCC -lm 是什么意思

gdb调试时No symbol "var" defined in current context，局部变量不可查看问题

shell获取进程ID的方法

个人分类

Linux21篇

virtualbox1篇

分区类型2篇

工作1篇

Java5篇

展开

归档

2013年11月4篇

2013年9月1篇

2013年8月1篇

2013年7月5篇

2013年6月11篇

展开

热门文章

linux下解压分卷压缩的zip文件
阅读量：3127

eclipse中的java工程找不到src下的jdbc.properties文件
阅读量：2322

Solution to error: ‘_1’ was not declared in this scope under C++11 standard
阅读量：2102

sed不能删除特定行字符开头的行
阅读量：1411

为什么一般情况下ln不能硬链接目录
阅读量：1355

最新评论

为什么一般情况下ln不能硬链接目录

Otagao：第一句话就不妥吧，转载之前最起码先读读看意思通不通吧。博主开篇就提出一个封闭式问题，“那么在d1...

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服

🗣 客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

 百度提供搜索支持



经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

CSDN APP

0

