

3.启动线程,每个线程从开始位置读取到结束位置为止

代码如下:

读文件工具类

```
1 import java.io.*;
    import java.nio.ByteBuffer;
    import java.nio.channels.FileChannel;
    import java.util.Observable;
 6
     * Created with IntelliJ IDEA.
     * User: okey
     * Date: 14-4-2
     * Time: 下午3:12
10
     * 读取文件
11
12
     */
13
    public class ReadFile extends Observable {
14
15
        private int bufSize = 1024;
16
       // 换行符
17
       private byte key = "\n".getBytes()[0];
       // 当前行数
18
       private long lineNum = 0;
19
       // 文件编码,默认为gb2312
20
21
       private String encode = "gb2312";
        // 具体业务逻辑监听器
22
23
        private ReaderFileListener readerListener;
24
25
        public void setEncode(String encode) {
26
           this.encode = encode;
27
28
29
        public void setReaderListener(ReaderFileListener readerListener) {
           this.readerListener = readerListener;
30
31
32
        /**
33
34
        * 获取准确开始位置
35
         * @param file
36
         * @param position
37
         * @return
38
         * @throws Exception
         */
39
        public long getStartNum(File file, long position) throws Exception {
40
41
           long startNum = position;
           FileChannel fcin = new RandomAccessFile(file, "r").getChannel();
42
43
           fcin.position(position);
44
           try {
45
               int cache = 1024;
46
               ByteBuffer rBuffer = ByteBuffer.allocate(cache);
47
               // 每次读取的内容
48
               byte[] bs = new byte[cache];
               // 缓存
49
50
               byte[] tempBs = new byte[0];
51
               String line = "";
52
               while (fcin.read(rBuffer) != -1) {
53
                   int rSize = rBuffer.position();
54
                   rBuffer.rewind();
55
                   rBuffer.get(bs);
56
                   rBuffer.clear();
57
                   byte[] newStrByte = bs;
                   // 如果发现有上次未读完的缓存,则将它加到当前读取的内容前面
58
59
                   if (null != tempBs) {
60
                       int tL = tempBs.length;
61
                       newStrByte = new byte[rSize + tL];
                       System.arraycopy(tempBs, 0, newStrByte, 0, tL);
62
63
                       System.arraycopy(bs, 0, newStrByte, tL, rSize);
64
                   }
                   // 获取开始位置之后的第一个换行符
65
66
                   int endIndex = indexOf(newStrByte, 0);
67
                   if (endIndex != -1) {
                       return startNum + endIndex;
68
69
70
                   tempBs = substring(newStrByte, 0, newStrByte.length);
```

```
71
                      联系我们
73
                                  请扫描二维码联系客服
           } catch
74
               e.p
                                  webmaster@csdn.net
75
           } final
                                  2 400-660-0108
76
               fci
                                  ▲ QQ客服 ● 客服论坛
77
78
            return
                     关于 招聘 广告服务 网站地图
79
                     ©2018 CSDN版权所有 京ICP证09002463号
80
                     ☆ 百度提供搜索支持
81
82
         * 从设置的
                                                      智置为负数, 刚读取到文件末尾
                                  经营性网站备案信息
         * @param f
83
                                  网络110报警服务
84
         * @param s
                                  中国互联网举报中心
85
         * @param e
                                  北京互联网违法和不良信息
86
         * @throws
                                  举报中心
87
         */
                       CSDN APP
88
        public void
                                                      tart, long end) throws Exception {
           File fin = new File(fullPath);
89
90
           if (fin.exists()) {
               FileChannel fcin = new RandomAccessFile(fin, "r").getChannel();
91
92
               fcin.position(start);
93
               try {
94
                   ByteBuffer rBuffer = ByteBuffer.allocate(bufSize);
95
                   // 每次读取的内容
96
                   byte[] bs = new byte[bufSize];
97
                   // 缓存
98
                   byte[] tempBs = new byte[0];
                   String line = "";
99
100
                   // 当前读取文件位置
                   long nowCur = start;
101
102
                   while (fcin.read(rBuffer) != -1) {
103
                       nowCur += bufSize;
104
                       int rSize = rBuffer.position();
105
106
                       rBuffer.rewind();
107
                       rBuffer.get(bs);
108
                       rBuffer.clear();
109
                       byte[] newStrByte = bs;
110
                       // 如果发现有上次未读完的缓存,则将它加到当前读取的内容前面
111
                       if (null != tempBs) {
112
                          int tL = tempBs.length;
113
                          newStrByte = new byte[rSize + tL];
114
                          System.arraycopy(tempBs, 0, newStrByte, 0, tL);
115
                          System.arraycopy(bs, 0, newStrByte, tL, rSize);
116
                      }
                       // 是否已经读到最后一位
117
118
                       boolean isEnd = false;
                       // 如果当前读取的位数已经比设置的结束位置大的时候,将读取的内容截取到设置的结束位置
119
120
                       if (end > 0 && nowCur > end) {
121
                          // 缓存长度 - 当前已经读取位数 - 最后位数
122
                          int l = newStrByte.length - (int) (nowCur - end);
                          newStrByte = substring(newStrByte, 0, 1);
123
124
                          isEnd = true;
125
126
                       int fromIndex = 0;
127
                       int endIndex = 0;
128
                       // 每次读一行内容,以 key (默认为\n) 作为结束符
                       while ((endIndex = indexOf(newStrByte, fromIndex)) != -1) {
129
                          byte[] bLine = substring(newStrByte, fromIndex, endIndex);
130
                          line = new String(bLine, 0, bLine.length, encode);
131
                          lineNum++;
132
133
                          // 输出一行内容, 处理方式由调用方提供
134
                          readerListener.outLine(line.trim(), lineNum, false);
                          fromIndex = endIndex + 1;
135
136
                       // 将未读取完成的内容放到缓存中
137
138
                       tempBs = substring(newStrByte, fromIndex, newStrByte.length);
139
                       if (isEnd) {
140
                          break;
141
142
143
                   // 将剩下的最后内容作为一行,输出,并指明这是最后一行
144
                   String lineStr = new String(tempBs, 0, tempBs.length, encode);
                   readerListener.outLine(lineStr.trim(), lineNum, true);
145
               } catch (Exception e) {
146
                   e.printStackTrace();
147
148
               } finally {
149
                   fcin.close();
150
151
152
           } else {
153
               throw new FileNotFoundException("没有找到文件: " + fullPath);
154
           // 通知观察者,当前工作已经完成
155
156
           setChanged();
157
           notifyObservers(start+"-"+end);
158
159
160
161
         * 查找一个byte[]从指定位置之后的一个换行符位置
162
163
         * @param src
         * @param fromIndex
164
165
         * @return
166
         * @throws Exception
167
         */
168
        private int indexOf(byte[] src, int fromIndex) throws Exception {
```

0

8

```
169
     170
                                                     i++) {
                     联系我们
                                                                                                                                            0
               if
171
                                  请扫描二维码联系客服
172
173
                                  webmaster@csdn.net
174
                                  2 400-660-0108
175
           return
                                  ▲ QQ客服 ● 客服论坛
176
177
                     关于 招聘 广告服务 网站地图
178
        /**
                     ©2018 CSDN版权所有 京ICP证09002463号
179
         * 从指定开刻
                                                      发一个全新的byte[]
                     ☆ 百度提供搜索支持
180
                                  经营性网站备案信息
181
         * @param s
         * @param f
182
                                  网络110报警服务
183
         * @param e
                                  中国互联网举报中心
         * @return
184
                                  北京互联网违法和不良信息
185
         * @throws
                                  举报中心
         */
                       CSDN APP
186
187
        private byt
                                                      , int endIndex) throws Exception {
188
           int size = endIndex - fromIndex;
189
           byte[] ret = new byte[size];
190
           System.arraycopy(src, fromIndex, ret, 0, size);
191
           return ret;
192
193
194 }
```

读文件线程

```
1
     * Created with IntelliJ IDEA.
 3
     * User: okey
     * Date: 14-4-2
     * Time: 下午4:50
 5
     * To change this template use File | Settings | File Templates.
 6
 7
    public class ReadFileThread extends Thread {
 9
10
        private ReaderFileListener processPoiDataListeners;
11
        private String filePath;
12
        private long start;
13
        private long end;
14
15
        public ReadFileThread(ReaderFileListener processPoiDataListeners,long start,long end,String file) {
16
            this.setName(this.getName()+"-ReadFileThread");
17
            this.start = start;
18
            this.end = end;
19
            this.filePath = file;
20
            this.processPoiDataListeners = processPoiDataListeners;
21
22
23
        @Override
24
        public void run() {
25
            ReadFile readFile = new ReadFile();
26
            readFile.setReaderListener(processPoiDataListeners);
27
            readFile.setEncode(processPoiDataListeners.getEncode());
28
              readFile.addObserver();
29
            try {
30
                readFile.readFileByLine(filePath, start, end + 1);
            } catch (Exception e) {
31
32
                e.printStackTrace();
33
34
35 }
```

具体业务逻辑监听

```
1 /**
 2
     * Created with Okey
     * User: Okey
     * Date: 13-3-14
     * Time: 下午3:19
     * NIO逐行读数据回调方法
 6
 7
     */
 8
    public abstract class ReaderFileListener {
 9
10
       // 一次读取行数,默认为500
11
       private int readColNum = 500;
12
13
       private String encode;
14
15
       private List<String> list = new ArrayList<String>();
16
        /**
17
18
        * 设置一次读取行数
19
        * @param readColNum
20
        */
21
       protected void setReadColNum(int readColNum) {
22
           this.readColNum = readColNum;
23
24
25
       public String getEncode() {
26
           return encode;
27
```

```
28
    29
            public
                     联系我们
30
           this.enc
                                  请扫描二维码联系客服
31
32
                                  webmaster@csdn.net
33
                                  2 400-660-0108
        * 每读取到-
34
                                  ▲ QQ客服 ● 客服论坛
35
        * @param li
36
        * @param li
                               广告服务 网站地图
                     关于 招聘
37
        * @param ov
                     ©2018 CSDN版权所有 京ICP证09002463号
38
        * @throws E
                     ☆ 百度提供搜索支持
39
                                  经营性网站备案信息
                                                       oolean over) throws Exception {
40
       public void
           if(null
41
                                   网络110报警服务
42
              list
                                   中国互联网举报中心
           if (!ove
43
                                   北京互联网违法和不良信息
44
              outp
                                  举报中心
              list
                       CSDN APP
45
           } else i
46
47
              output(list);
48
              list.clear();
49
50
51
52
        * 批量输出
53
54
        * @param stringList
55
56
        * @throws Exception
57
58
       public abstract void output(List<String> stringList) throws Exception;
59
60 }
```

0

线程调度

```
1 import java.io.File;
 2
    import java.io.FileInputStream;
 3
    import java.io.IOException;
 4
 5
     * Created with IntelliJ IDEA.
 6
     * User: okey
 8
     * Date: 14-4-1
     * Time: 下午6:03
     * To change this template use File | Settings | File Templates.
10
11
12
    public class BuildData {
        public static void main(String[] args) throws Exception {
13
14
            File file = new File("E:\\1396341974289.csv");
15
            FileInputStream fis = null;
16
17
                ReadFile readFile = new ReadFile();
                fis = new FileInputStream(file);
18
                int available = fis.available();
19
                int maxThreadNum = 50;
20
21
                // 线程粗略开始位置
                int i = available / maxThreadNum;
22
                for (int j = 0; j < maxThreadNum; j++) {</pre>
23
                    // 计算精确开始位置
24
                    long startNum = j == 0 ? 0 : readFile.getStartNum(file, i * j);
25
26
                    long endNum = j + 1 < maxThreadNum ? readFile.getStartNum(file, i * (j + 1)) : -2;</pre>
27
                    // 具体监听实现
                    ProcessDataByPostgisListeners listeners = new ProcessDataByPostgisListeners("gbk");
28
                    new ReadFileThread(listeners, startNum, endNum, file.getPath()).start();
29
30
31
            } catch (IOException e) {
32
                e.printStackTrace();
33
            } catch (Exception e) {
34
                e.printStackTrace();
35
36
37 }
```

现在就可以尽情的调整 maxThreadNum来享受风一般的速度吧!

```
想对作者说点什么
```

机器熊技术大杂烩: 想你这样的人真是坑人不眨眼! 以后走路往后悄悄免得被拍黑砖! (08-09 17:15 #8楼)

(1) yang420097009: 草,能不能给全了,浪费感情 (07-23 16:15 #7楼)

愛 爱学习的小猿: 是不是少写了ProcessDataByPostgisListeners类啊?? (02-08 11:21 #6楼)



java大文件复制最高效方法: FileChannel (已测)

② 238

0

话不多说,上代码: package com.yhfund.file.util; import java.io.*; import java.nio.channels.FileChanne...

Java 多线程读取一个文件

 $package\ com. thread. multipl. my solution; \quad import\ java. io. IOException; \quad import\ java. io. Random Access Fi...$

Java多线程读取大文件

前言 今天是五一假期第一天,按理应该是快乐玩耍的日子,但是作为一个北漂到京师的开发人员,实在难想出去那玩耍。好玩的...

JAVA多线程读取同一个文件, 加速对文件内容的获取

♦ ◎ 7918

前几天,朋友托我帮个忙,问我能不能用多线程的方式,读取一个文件,获取里面的内容。他大概想做的事情,就是读取文件里面每一...

Java 多线程并行处理大文件 (淘宝2012校招技术笔试题)

问题:统计一个单词可重复的英文文件(假设4G)中每个单词出现的次数,把结果按照英文排序放入一个文件中。并能够检索特定单...

相关热词 javall 与java java的~ java java和--



Extjs4 颜色选择器 ColorField 阅读量: 3564 java读取超大文件 阅读量: 1389 使用Flash(Flax)post方式跨越请求的Extjs4 代理(Proxy)的实现类 阅读量: 762 Intellij 更改JVM内存 阅读量: 533

最新评论

JAVA使用多线程读取超大文作 majianxiong_lzu: 想你这样的人真以后走路往后悄悄免得被拍黑砖!

JAVA使用多线程读取超大文作 yang420097009:草,能不能给全

JAVA使用多线程读取超大文作 qq_34533072: 是不是少写了Prod gisListeners类啊??

JAVA使用多线程读取超大文化 u013516035: 是不是少写了Proce sListeners类啊??

JAVA使用多线程读取超大文作 lblblb1111: 是不是少写了Process steners类啊??

联系我们



请扫描二维码联系客服

✓ webmaster@csdn.ne

▲ 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

8