

使用说明

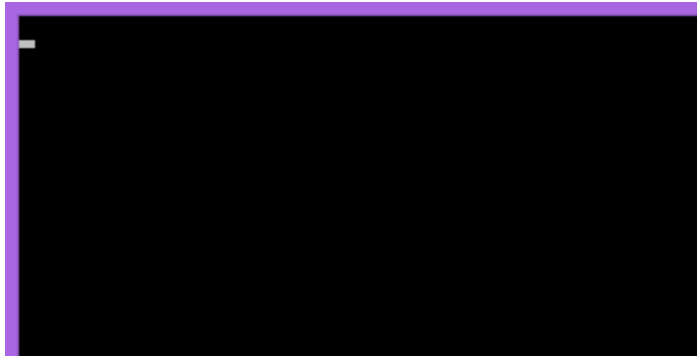
读取文件并解析统计 (C++)

程序: main.cpp

开发环境: win8.1、codeblocks(svn9916)

使用方法:

(1)先运行此代码, 处于等待状态,



(2)一旦运行 start_sh1_play.bat, 解析即开始

```
上海中畅高效多用行情网关V1.4<有效期至2017-07-01>
按 Ctrl+q 退出
=====
上交所Level1行情
=====
数据源:                                     历史数据文件
文件路径:                                     ../../hisfile/20160922
已读取字节:                                4616416? ! 已读取数据包:                19
速率:                                       459.24kB/s ! 最新消息时间:          20161008-16:56:56.129
同步文件路径:                             ../../data/mktdt00.txt
=====
dataBagID : 1  HeadTime : 09:29:35.880
dataBagID : 2  HeadTime : 09:29:41.180
dataBagID : 3  HeadTime : 09:29:45.320
dataBagID : 4  HeadTime : 09:29:50.880
dataBagID : 5  HeadTime : 09:29:56.180
dataBagID : 6  HeadTime : 09:30:01.680
dataBagID : 7  HeadTime : 09:30:06.830
dataBagID : 8  HeadTime : 09:30:11.890
dataBagID : 9  HeadTime : 09:30:16.840
dataBagID : 10 HeadTime : 09:30:21.820
dataBagID : 11 HeadTime : 09:30:26.720
dataBagID : 12 HeadTime : 09:30:31.850
dataBagID : 13 HeadTime : 09:30:36.840
dataBagID : 14 HeadTime : 09:30:41.890
dataBagID : 15 HeadTime : 09:30:46.850
dataBagID : 16 HeadTime : 09:30:51.800
dataBagID : 17 HeadTime : 09:30:56.890
dataBagID : 18 HeadTime : 09:31:01.850
dataBagID : 19 HeadTime : 09:31:06.820
```

(3)运行结束后, 回放程序的“已读取字节”变为 0KB/s, 解析程序输出“finished!”即解析结束, 下一步针对 resOut.txt 文件进行查询

一、读取数据文件并解析 (大致思路)

1、由于 mktdt00.txt 文件处于不断更新的状态, 需要反复读取。

回放工具运行, txt 文件即开始更新, 根据运行界面显示写入数据大约为 500K/s, mktdt00.txt 文件的大小 2.37M, 所以 txt 文本的更新屏幕是 4-5 秒一次, 而 txt 文件

刷写的时间远小于此时间间隔，据此可以设置隔一秒钟读取一次 `mktddt00.txt`，发现文件第一行的时间变量 `HeadTime` 发生变化时，即可认定 `mktddt00.txt` 已经被更新，然后读取，解析。

- 2、将解析到的每一组数据都缓存到 `preData` 数组中，如果检测到新读取的时间是下一分钟的时候，则存储了上一时刻数据的 `preData` 中缓存的数据即为对应分钟片段的数据，存入数组 `resOut` 中，最后将 `resOut` 中的数据输出到 `resOut.txt` 中。
- 3、由于每个时刻读取的交易量是从开始到当前时刻位置的总交易量，所以每次读取到某分钟段的最后一个时刻，则存储此时的交易总量到 `preMinVol` 数组中，当读到下一个分钟段内的最后一个时刻时，用交易总量减去 `preMinVol` 存储的量，即为本分钟段内的交易量。

二、文件的存储

1、存储文件：`resOut.txt`

2、存储格式说明：存储的每一个字符串用“，”隔开，如下图所示

```
600009|上海机场,09:31|27.640|35000,09:32|27.610|32700,09:33|27.590|4000,09:34|27.580|13200,0
600010|宝钢股份,09:31|2.860|945100,09:32|2.860|277300,09:33|2.860|183300,09:34|2.860|135400,
600011|华能国际,09:31|7.180|14700,09:32|7.180|43100,09:33|7.180|8500,09:34|7.180|32000,09:35
600012|皖通高速,09:31|12.700|19200,09:32|12.700|15900,09:33|12.690|24500,09:34|12.680|14200,
600015|华夏银行,09:31|10.150|369520,09:32|10.150|55600,09:33|10.140|276692,09:34|10.140|2787
600016|民生银行,09:31|9.390|265980,09:32|9.380|203020,09:33|9.380|82400,09:34|9.370|71440,09
600017|日照港,09:31|4.070|130000,09:32|4.070|49400,09:33|4.070|106600,09:34|4.070|59300,09:3
```

每一行是一支股票的数据，其中第一列包含的数据为：编号|中文简称。

从第 2~241 列的 240 列数据，表示从 9:30~11:30、13:00~15:00 包含的 240 个分钟段，每一分钟段的数据存放格式为：时间段|交易价格|交易量。

查询（控制台查询（C++））

程序：`search.cc`

开发环境：`ubuntu_LTS_15.04`，`gedit`，`GCC 5.4.0`

使用说明：

(1) 运行程序后，要求输入股票编号以及时间片段

```
ene@ubuntu:~/leetcode$ g++ search.cc -o search;./search
please input SecurityID and Time : (input 'end' to exit !)
such as : 600178 09:35
-----
```

按要求输入后，会返回这一分钟的交易价格和交易量

```

please input SecurityID and Time : (input 'end' to exit !)
such as : 600178 09:35
-----
600123 11:23
SecurityID : 600123
Symbol : 兰花科创
Price : 7.470
Volume : 2200
-----
please input SecurityID and Time :
600600 14:55
SecurityID : 600600
Symbol : 青岛啤酒
Price : 32.140
Volume : 21000
-----
please input SecurityID and Time :

```

(2) 输入“end”可以推出查询

```

600600 14:55
SecurityID : 600600
Symbol : 青岛啤酒
Price : 32.140
Volume : 21000
-----
please input SecurityID and Time :
end
ene@ubuntu:~/leetcode$

```

(3) 补充说明:

- a) 输入支持 09: 35, 9: 35 这种形式
- b) 输入时间不符合要求时（格式错误或者时间不在可查询范围内），会要求重新输入时间，直到输入正确时间或者 end 退出为止

```

-----
600178 14:25:53
timess is not a normal value,input time again.
14:25
SecurityID : 600178
Symbol : 东安动力
Price : 12.080
Volume : 33700
-----
please input SecurityID and Time :
600178 12:24
timess is not a normal value,input time again.

```

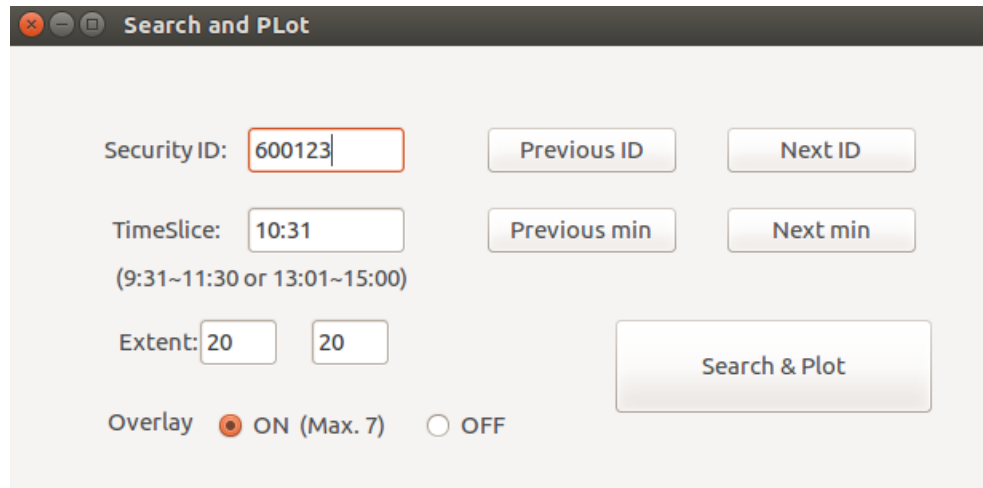
查询（GUI 版本（python））

代码：search_GUI.py、resOutSearch.py（定义部分函数）

开发环境：ubuntu_LTS_15.04, gedit, python2.7, GUI 库（pygtk）

使用说明：

- (1) 运行 search_GUI.py 之后弹出一个窗口



其中：

- (a) Security ID 后面的文本框内输入股票的编号，
- (b) 按键 PreviousID,NextID 分别是让 ID 的值向临近的点变化
- (c) TimeSlice 输入时间片段，
- (d) 按键 Previous min,Next min 的作用与前面类似，变化幅度为 1 分钟
- (e) Extent: 后的两个方框可以设定输出图像的时间范围（比如默认值（20，20），就是指查询时刻向前 20 分钟作为绘图起点，向后 20 分钟作为绘图的终点），当输入的范围导致有的点超出可查询时间范围时，会自动去掉这些点，选取客观存在的点进行绘图。
- (f) Overlay 是控制图像叠加的功能，默认 ON 是每次 plot 开启新窗（可以开启无数个新窗），OFF 时在原有绘图窗口叠加，不再开启新窗。目前 OFF 模式仅能使用 price 图像叠加，功能有待改进，不建议使用。

- (2) 设置好参数之后，点击 Search&Plot 按钮绘制图像

