还不会Redis，不会字典树Trie

其余功能基本实现

1. 离线部分
   1. 创建词典 vector<string,int>

把中文语料文件和英文语料文件分别放入两个文件夹中，在配置文件中写好

读取时自动解析文件夹，将这两个文件夹下的所有文件全部按顺序读取

对于英文 先读取一遍将大写改为小写将标点符号改为空格

对于中文直接用cppjieba分词

存入文件时，先写入英文单词的个数，再写入英文，再写入中文

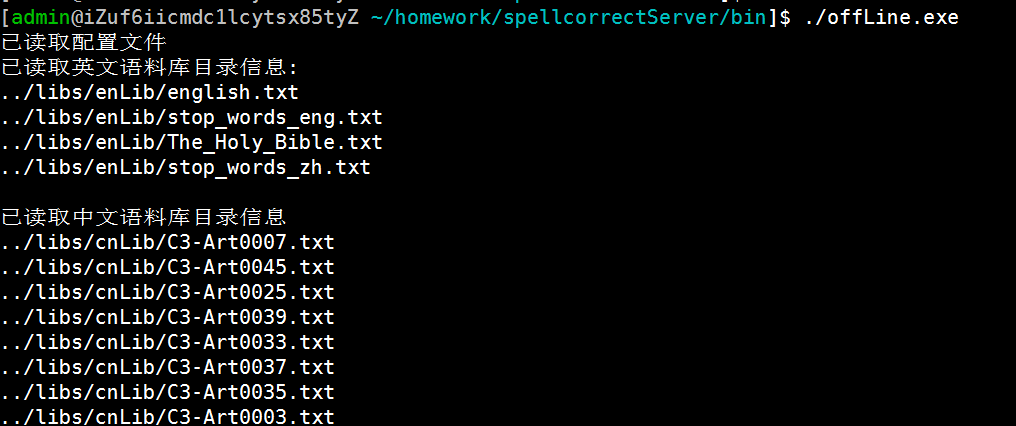
B 创建索引 vector<string,set<int>>

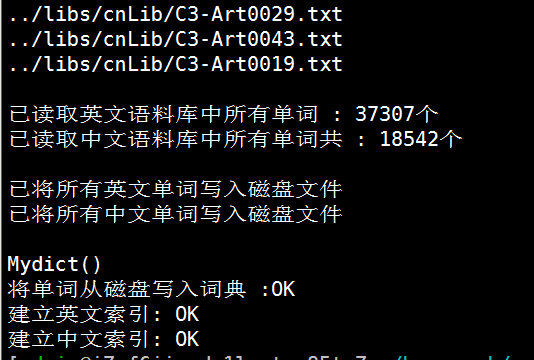
英文索引，英文单词在vector的前一部分，所以直接对于26个字母作为下标建立索引就可以了

中文索引，先遍历每一个中文词，对每个单词中的汉字作为下边建立索引，中文单词在vector的后一部分。

最后再写入文件

效果





1. 搭框架

在6.14的v4版本中

要改Mytask（子线程调用），send也要修改。要加上信息长度，在回调服务器中要加入时间线程用于定期更新（绑缓存管理的更新函数）

增加缓存和缓存管理类，要用到各个子线程的\_\_thread变量。小改一下

1. 业务逻辑

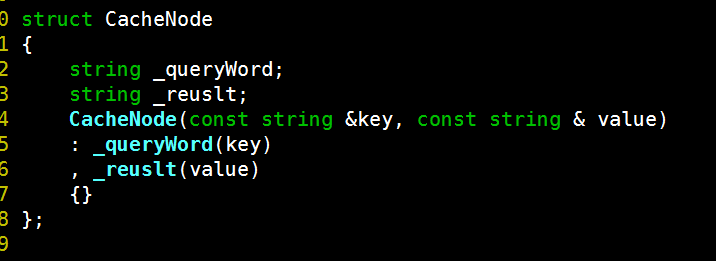
客户端发请求 🡪 EPoll池收到 🡪 交给Tcp连接的Onmesser回调函数执行 🡪 该回调函数读取数据，创建Task对象交给子线程 🡪 子线程执行Task的process的回调函数进行处理，得到想要结果的json字符串 🡪 放入EPOll池的待发送任务链表，唤醒EPOll池 🡪 EPOLL池执行绑定Tcp连接的回调函数 🡪 将JSON字符串发给客户端

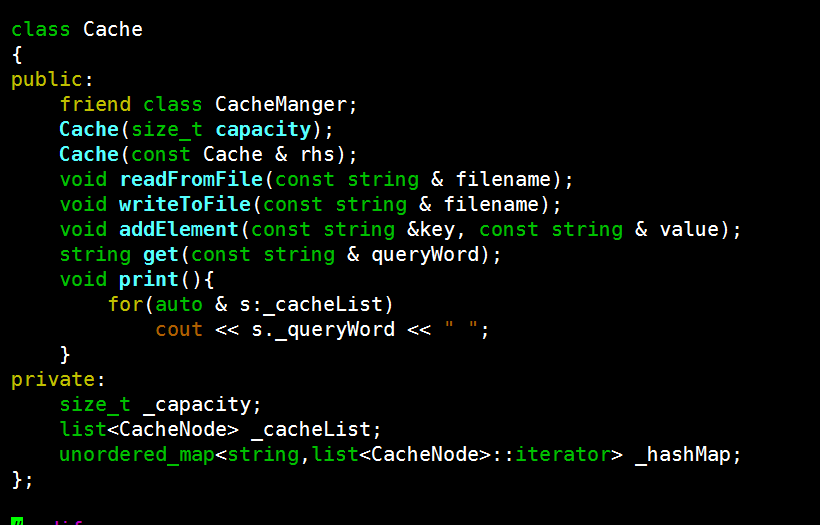
缓存实现：

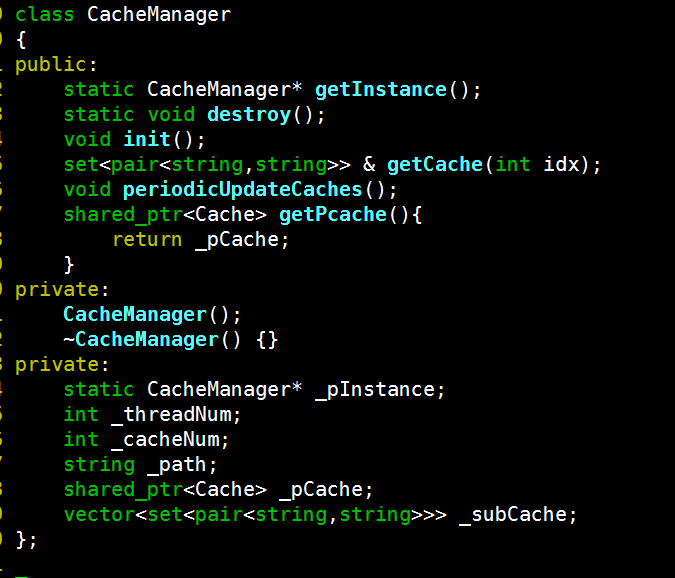
每个线程可用属于自己的set来记录更新间隔内的热点数据

各个线程共用一个cache作为只读查询，这个cache定期更新，

更新时把每个线程的set中的数据插入到公共cache







MyTask中的process逻辑（子线程拿到任务并计算最后发给主线程EPOLL池）

在缓存中查找要查找的数据是否存在，若不存在进行计算来得到数据，存在在缓存中拿数据，将数据写入属于自己线程的set，把数据交给EPoll池

计算：

用百度到的最小编辑距离算法，

判断横纵数组的元素是否相等时，，全按照中文的标准进行。就是讲这两个数组为

Vector<string>。每个元素是一个汉字（英文就是一个字符）

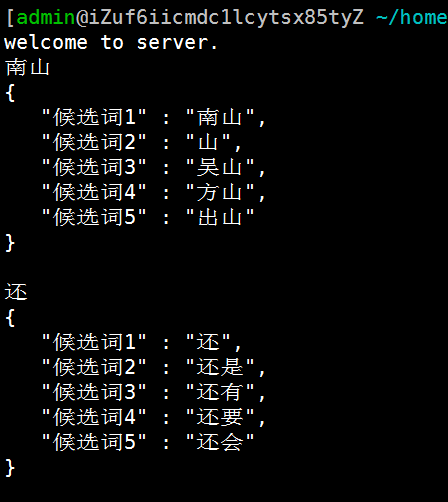
把查找词和候选词转化为这两个数组再用百度到的算法就可以了

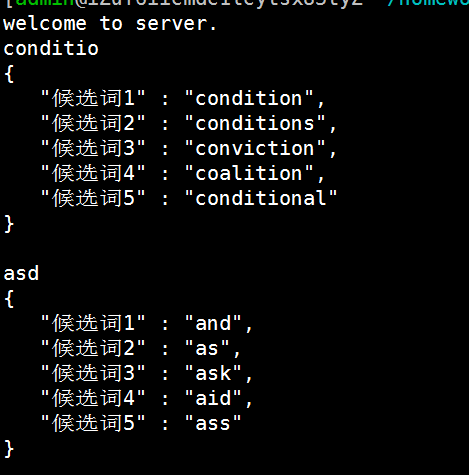
结果放入优先队列要自己写一个比较的函数sturct

然后没了

效果

客户端





服务器：



