**K-V数据库函数接口整理**

**KVDBHandler**

数据成员：string KVDB：存储K-V数据库文件名； Hash index：作为索引快速查找K-V数据库； MinHeap minHeap：以最小堆为存储方式，存储设置过期时间的key；int offset：当前存储文件末与文件头偏移值

使用结构体KVDBData；数据成员包括int keyLen、int valueLen、string key、string value；同时设置set函数可对所有数据成员设值

void resetIndex()：首先将索引清空，offset偏移值归零；将文件长度存为length用以遍历文件，在while循环中调用readKVDBData函数从文件中读取每一组K-V表，增加offset值，存入索引

void readKVDBData(fstream& file, KVDBData& s)：执行程序前，文件读指针已定位；利用read函数按照keyLen，valueLen，key，value顺序从文件中读取，若valueLen为-1，则说明该key已被删除，不需读取value，之间将value设为空，之后将所有读取到的结果放入参数s中传回去

void writeKVDBData(fstream& file, KVDBData& s)：执行程序前，文件写指针已定位；利用write函数按照keyLen、valueLen、key，value顺序将参数s写入文件

**KVDBHandler友元函数**

int set(KVDBHandler\* handler, const string& key, const string& value)：调用writeKVDBData函数；在索引中新增key和当前偏移值；偏移值加上新增长度；在最小堆中新增key，存为KEY\_NOT\_EXIST\_IN\_MINHEAP

int get(KVDBHandler\* handler, const string& key, string& value)：调用最小堆的del函数，将当前时间已过期的key从最小堆中删除；从索引中得到key所对应的偏移值；如果偏移值为KEY\_NOT\_EXIST，说明key不存在；调用最小堆的get函数，若为false则此key已过期；借助偏移值定位文件读指针，从文件中读出key对应的K-V组；若valueLen为-1则说明此key已被删除

int del(KVDBHandler\* handler, const std::string& key)：调用get函数，若得到结果不为SUCCESS，则说明此key因为各种各样的原因已失效，删除失败；文件尾加key，同时valueLen设置为-1；索引中更新key的偏移值；offset更新

int purge(KVDBHandler\* handler)：定义副本文件，调用两次purge子函数

int purgeSubfunction(KVDBHandler\* handler, KVDBHandler\* save\_handler)：主要功能为将源文件中内容去除已失效的值写入副本文件；主要方式为遍历源文件索引，若源文件中此key可以得到但是副本文件中无法得到，则写入副本文件当中；源文件遍历完毕之后麻将源文件中内容删除

int expires(KVDBHandler\* handler, const string key, int n)：调用get函数，若得到结果不为SUCCESS，则说明此key因为各种各样的原因已失效，设置过期失败；若成功，则调用最小堆set函数设置过期时间

**MinHeap**

数据成员：string MinHeap\_filename存储最小堆文件名；Hash index作为历史记录，存储曾存在于最小堆的key；priority\_queue<TimeNode, vector<TimeNode>, cmp> minHeap最小堆存储设置过期时间的key

使用结构体TimeNode；数据成员包括string key、int time；同时重载比较函数

void reset()：清空历史记录和最小堆；遍历最小堆文件，若从文件中读到的key的过期时间大于等于当前时间则推入最小堆，并在历史记录中存为KEY\_EXIST\_IN\_MINHEAP；

void set(string key, int time)：将key和key的过期时间写入文件；推入最小堆；推入历史记录；

bool get(string key)：从历史记录中得到key，若返回值为EY\_NOT\_EXIST或KEY\_NOT\_EXIST\_IN\_MINHEAP，说明此key未过期，返回true；遍历最小堆，从最小堆中得到key；若能得到则说明未过期；若存在于历史记录但是不存在于最小堆说明此key已过期，返回false

void del()：遍历最小堆，将最小堆中过期时间小于当前时间的key全部从最小堆中删除

**Hash**

数据成员：以链表形式存储哈希表

使用结构体；数据成员包括string key、int offset；

int GetHashCode(string str)：哈希函数，传入的字符串按照一定规律设置为hash值

void set(string key, int offset)：得到此key对应的hash值；遍历对应的链表，若找到key则更新offset；若无法找到，则新增；

int get(string key)：得到此key对应的hash值；遍历对应的链表，若找到key则返回对应的offset，若找不到则返回KEY\_NOT\_EXIST

void clear()：将哈希表清空

**K-V数据库函数返回值情况分析**