

实验二报告

一实验目标

本实验实现一个基于文本的音乐检索与推荐系统。最后的结果是实现三个 exe 文件:分别为无界面版的检索,推荐功能和结合两个功能的界面。最基本目标是在整个实验过程中利用上课所学的知识和数据结构,加强对所学知识的巩固和认识,比如栈,链表,B 树等等数据的实现。

二实验环境

Window10 平台, visual studio 2012, Qt 5.2.0;

三抽象数据结构的说明

实验一有自定义链表,字符串,存储音乐信息结构等,实验二有存储结构的几个数据定义,比如词典,文档链表,B 树索引等等,如下一些示例:

文档存储结构:

```
struct INVDOC
{
    int DocID;    //文档 ID
    int times;    //单词出现次数
    struct INVDOC* next;
};
```

文档链表:

```
class INVDIC
{
public:
    CharString term; //单词
    int DF;          //单词出现的文档数
    int occur;       //单词出现的总次数
    struct INVDOC* head; //文档链表
```

```

public:
    INVDIC():DF(0), occur(0), head(NULL) {term.StrAssign("");}
    ~INVDIC()
    {
        delete head;
    }
    void sortDoc();           //根据文档中的关键词出现次数排序
    //文档的创建, 添加, 查找均在 createB_Tree 这个函数内实现
};

//B 树结构
typedef struct BTreeNode{
    int keynum;              //关键字个数
    struct BTreeNode* parent; //指向双亲
    KeyType key[M+1];        //关键字, 0 号不用
    struct BTreeNode* ptr[M+1]; //子树指针
}BTreeNode, *BTree;

```

这其中最关键的是 `KeyType` 这个关键字的结构, B 树就是用这个进行索引的。

四算法说明

实验二最主要的算法就是 B 树, B 树写完, 那么搜索功能就基本实现了, 而 B 树的算法, 书上有, 这里就不说了, 就说说搜索具体的做法。而推荐算法则是一个比较简单的, 自己想的一个方法, 具体做法如下(截取自我的程序注释)

1.搜索

```

/*
* 实现简单的查寻关键字的音乐查寻功能
* 查寻规则:
* 查寻, 通过找到关键字出现次数, 按从高到低进行显示
* 其中关键字出现在标题, 出现次数+60, 关键字出现在专辑, 出现次数+50,
关键字出现在歌手中, 出现次数+30, 在歌词中+1
* 在关键字出现在标题和专辑中时, 若关键字为“的”等无意义的词, 则按+1
算
* 根据出现的多少排序输出结果
*/

```

2.推荐

```
/*  
*实现简单的音乐的推荐  
*推荐规则：  
*   若输入的是歌名：  
*       标题中含同样的词汇最先(除去“的”这样的词汇，标题很好表达了  
歌的内容等等)  
*       歌手，曲词作者，专辑都相同的第二  
*       歌手，曲词作者相同的次于前面  
*       仅歌手相同的最后(后面三者都是为了保证风格大致一样)  
*       比较直到得到 10 首歌为止，多于 10 首的输出 10 首歌曲，少于 10 首  
的不补  
*   否则：  
*       给出提示”请输入歌名“  
*   且一次仅检索一个词，也就是只能输入一个歌名  
*/
```

五流程概述

基本流程为：

载入词典—>读取网页, 存储音乐信息-->中文分词-->制作 B 树搜索树-->

输入结果-->查寻(或者推荐)-->输出结果(若在 gui 上，则显示在页面上)

六输入输出操作说明

在 exe 文件夹下新建 query1.txt, query2.txt，具体格式为 query1.txt 每一行为待搜索的词，词与词之间空格隔开，当然不隔开也没问题，那么就会先分词，这样也行，query2.txt 具体格式为每行为一个歌曲名，若是多了将得不到推荐结果，至于 gui，则综合两者功能，就是在输入框中输入关键词，有一定的健壮性。

七结果

皆达预期，无界面搜索将输出 result1 文件，无界面的推荐将给出 result2

文件，界面程序将显示在界面上，输出的结果按我的算法进行输出。

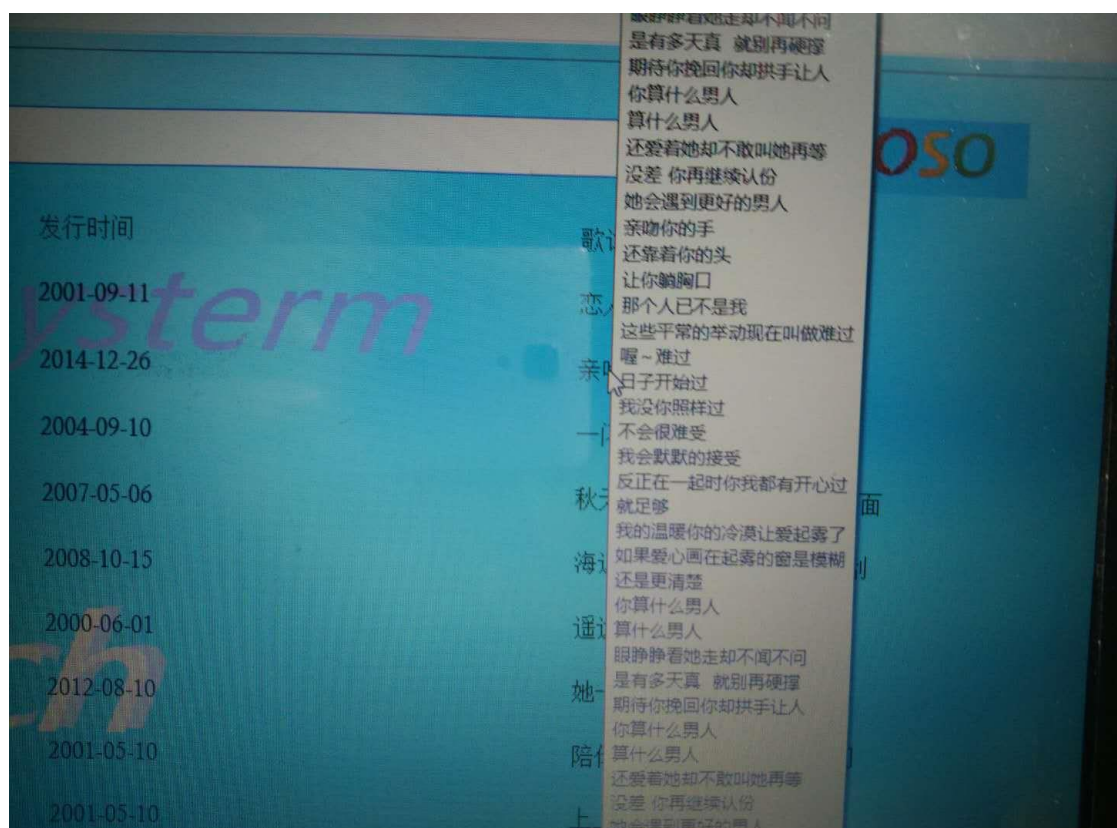
八亮点介绍

1. 精心设计的界面

多瞧瞧，多看看，那个 search 和 recommend 字是不是很有立体感啊，而且界面也挺好看，还有 soso 那个图标还挺带感的。

2. 全部歌词的查看方法

歌词的全部查看功能，利用 qt 的 tooltip 工具，只要你把鼠标久按在歌词上就会出现，一松开就没了，见下图：



3. 上下翻页功能

当你搜索显示的结果较多时，界面下会出现上翻下翻的功能，且到最后
一页下翻键消失，第一页没有上翻键。相应图标在右下角显示。

4. 英文和中文搜索功能分开

中文歌曲的搜索完全靠关键词，具体做法如上。

英文歌曲的搜索，首先是在标题，专辑和歌手中看是否找到输入词，找到即直接输出，然后再找与之相同的歌手专辑等等的歌曲进行输出。在进行英文歌曲的搜索时，还有一个优点，即若是在数据库中存在 Kiss Goodbye 这首歌，则你输入 Kiss Goodbye 或 Kiss 或者 Goodbye 或者 Good 等等均会搜索得到。

5 回车键实现搜索推荐功能

为了翻便喜欢用回车功能键的朋友，已经增添了回车键的事件，即你可以简简单单回一下车就可以进行相应的功能，当然你也可以点击那个 soso 的图标。

九实验体会

虽然刚接手感觉比较简单，但只有自己亲自做了，才会发现这样那样的坑，以及自己这样那样的程序错误，总之只有自己做出来才算自己真正的掌握。所以我觉得这次作业很是提高我们的编程水平和自己解决各种 bug 的能力。