**第4\_1题**

*模拟KTV点歌系统。用户可按歌名查找某首歌曲或按歌手名查找其所有歌曲，点歌后显示所点歌曲歌词。管理员可添加和删除歌曲，每个歌曲的歌词用一个单独的文件存储。*

**算法描述：**

歌曲信息保存在songlib.csv 中，管理员密码保存在admin.db中，歌词文件为对应歌曲名.lrc 的纯文本。

**源程序：**

no2\_2.c

**void** Printlrc(**char** \*song) {  
 /\*\*  
 \* @brief 打印歌词  
 \* @param song 歌曲名  
 \* @author 张俊华 16030199025  
 \*/  
 **if**(isspace(song[strlen(song) - 1]))  
 song[strlen(song) - 1]='\0';  
 //去掉字符串结尾的换行符  
 strcat(song, ".lrc");  
 //strcat连接字符串，给字符串加上.lrc后缀  
 FILE \*fp;  
 fp = fopen(song, "r");  
 **if** (fp == NULL) **return**;  
 //初始化文件指针，只读模式  
 **char** string[1000];  
 **while** (fgets(string, 1000, fp)) {  
 puts(string);//每读到一行就在屏幕上显示一行  
 sleep(1);  
  
 }  
 puts("\n\n按任意键继续...");  
 fflush(stdin);  
 getchar();  
}  
  
**void** ChooseSong() {  
 /\*\*  
 \* @brief 检索歌曲  
 \* @author 张俊华 16030199025  
 \*/  
 FILE \*fp;  
 fp = fopen("songlib.csv", "r");  
 **if** (fp == NULL) **return**;  
 //初始化文件指针，只读  
 **char** keyword[100];  
 printf("请输入歌手名或歌曲名：\n");  
 scanf("%s", keyword);  
 //检索关键字  
 **int** index = 0;  
 //结果集下标序号  
 **char** result[1000][100];  
 //结果集  
 **char** string[1000];  
 //用来保存从文件中读到的字符串  
 **while** (fgets(string, 1000, fp)) {  
 //逐行处理，检索关键字  
 **for** (**int** j = 0; j < strlen(string); j++) {  
 //检索是否包含keyword  
 **int** findflag = 1;  
 //检索状态  
 **for** (**int** i = 0; i < strlen(keyword); i++) {  
 **if** (\*(string + j + i) != keyword[i]) {  
 findflag = 0;//发现不匹配  
 **break**;  
 }  
 }  
 **if** (findflag) {  
 //如果匹配  
 **int** i = 0;  
 strcpy(result[++index], string);  
 //把匹配的歌曲信息保存到数组中  
 printf("检索到以下歌曲条目包含关键字：\n");  
 printf("\t%d\t--\t", index);//输出序号  
 **for** (i = 0; i < strlen(string); ++i) {  
 **if** (\*(string + i) == ',') **break**;//输出歌手名，遇到逗号截止  
 printf("%c", \*(string + i));  
 }  
 printf("\t---%s\n", string + i + 1);//输出歌曲名  
 }  
 }  
  
 }  
 **if** (index == 0) {  
 printf("未检索到\n\n");  
 **return**;  
 }  
 printf("请输入选择的序号：");  
 **int** choose;  
 scanf("%d", &choose);  
 //让用户选择一个下标  
 **if** (choose <= index) {  
 printf("点歌成功！\n\n");  
 Printlrc(result[index]);  
 //调用printlic，打印对应歌曲的歌词  
 printf("\n\n");  
 } **else** printf("请检查输入的内容!!\n");  
  
}  
  
**void** AddSong() {  
 /\*  
 \* 管理员函数，追加一首歌  
 \*/  
 FILE \*fp;  
 fp = fopen("songlib.csv", "a"); //追加打开文件  
 **if** (fp == NULL) **return**;  
  
 printf("请输入歌手名：");  
 **char** artist[100];  
 scanf("%s", artist);  
 printf("请输入歌曲名：");  
 **char** songName[100];  
 scanf("%s", songName);  
  
 fprintf(fp, "%s,%s\n", artist, songName);  
 //格式化追加  
 printf("增加歌曲成功\n");  
 printf("请手动导入歌词文件，文件名：“歌手名,歌曲名.lrc”（注意半角逗号）\n\n");  
 fclose(fp);  
}  
  
**void** DelSong() {  
 /\*  
 \* 管理员函数，删除歌曲  
 \*/  
 FILE \*fp;  
 FILE \*fptemp; //中转保存，临时文件  
 fptemp = fopen("temp.txt", "wt");  
 fp = fopen("songlib.csv", "r");  
 **if** (fptemp == NULL) **return**;  
 **if** (fp == NULL) **return**;  
 //初始化文件指针  
  
 **char** string[100];  
 **int** index = 0;  
 printf("请输入歌手名：");  
 **char** artist[100];  
 scanf("%s", artist);  
 printf("请输入歌曲名：");  
 **char** songName[100];  
 scanf("%s", songName);  
  
 **while** (fgets(string, 1000, fp)) {  
 **int** findArtist = 0;  
 **int** findSongName = 0;  
 **for** (**int** j = 0; j < strlen(string); j++) {  
 //检索是否包含artist  
 **int** findArtistflag = 1;  
 //检索状态  
 **for** (**int** i = 0; i < strlen(artist); i++) {  
 **if** (\*(string + j + i) != artist[i]) {  
 findArtistflag = 0;  
 **continue**;  
 }  
 }  
 **if** (findArtistflag == 1) {  
 findArtist = 1;  
 **continue**;  
 }  
 **int** findSongNameFlag = 1;  
 //检索是否包含Songname  
 **for** (**int** i = 0; i < strlen(songName); i++) {  
 **if** (\*(string + j + i) != songName[i]) {  
 findSongNameFlag = 0;//检索到  
 **continue**;  
 }  
 }  
 **if** (findSongNameFlag == 1) {  
 findSongName = 1;  
 **continue**;  
 }  
 }  
 **if** (findArtist && findSongName) {  
 **int** i = 0;  
 printf("检索到以下歌曲条目包含关键字：\n");  
 printf("\t%d\t--\t", ++index);  
 **for** (i = 0; i < strlen(string); ++i) {  
 **if** (\*(string + i) == ',') **break**;  
 printf("%c", \*(string + i));  
 }  
 printf("\t---%s\n", string + i + 1);  
 printf("删除吗(y/n)");  
 fflush(stdin);  
 **char** choice;  
 scanf("%c", &choice);  
 **if** (choice == 'y') **continue**;  
 }  
  
 fputs(string, fptemp);  
 }  
 fclose(fp);  
 fclose(fptemp);  
 fptemp = fopen("temp.txt", "r");  
 fp = fopen("songlib.csv", "wt"); //  
 **if** (fptemp == NULL) **return**;  
 **if** (fp == NULL) **return**;  
 **while** (fgets(string, 1000, fptemp)) {  
 fputs(string, fp);  
 }  
  
 fclose(fptemp);  
 fclose(fp);  
 printf("\n");  
  
}  
  
**void** Admin() {  
 /\*  
 \* 管理员后台  
 \*/  
 **char** psw[100];  
 **char** correctPsw[100];  
 printf("请输入管理员密码：");  
 scanf("%s", psw);  
  
 FILE \*fp;  
 fp = fopen("admin.db", "r");  
 **if** (fp == NULL) **return**;  
  
 fgets(correctPsw, 100, fp);  
 //从文件中读取密码  
 **for** (**int** i = 0; i < strlen(correctPsw); ++i) {  
 //比较是否相同，发现任何一位不同直接退出  
 **if** (psw[i] != correctPsw[i]) {  
 printf("管理员密码错误！\n");  
 **return**;  
 }  
 }  
 printf("管理员身份确认成功！\n");  
 printf("\t序号\t--------\t功能\n");  
 printf("\t1\t--------\t增加歌曲\n");  
 printf("\t2\t--------\t删除歌曲\n");  
 printf("\t0\t--------\t退出系统\n");  
 **int** choose;  
 scanf("%d", &choose);  
 **switch** (choose) {  
 **case** 1:  
 AddSong();  
 **break**;  
 **case** 2:  
 DelSong();  
 **break**;  
 **case** 0:  
 **return**;  
 **default**:  
 puts("请检查输入...");  
 getchar();  
 }  
}  
  
**int** main(**void**) {  
 /\*  
 \* 主入口，很简单的  
 \*   
 \*/  
 **int** choose = 0;  
  
 **while** (1) {  
  
 printf("\t==================\n");  
 printf("\tktv点歌系统\n");  
 printf("\t==================\n");  
 printf("\t序号\t--------\t功能\n");  
 printf("\t1\t--------\t开始点歌\n");  
 printf("\t2\t--------\t管理员登录\n");  
 printf("\t0\t--------\t退出系统\n");  
 scanf("%d", &choose);  
 **switch** (choose) {  
  
 **case** 1:  
 ChooseSong();  
 **break**;  
 **case** 2:  
 Admin();  
 **break**;  
 **case** 0:  
 **return** 0;  
 **default**:  
 puts("请检查输入...");  
 getchar();  
 }  
 }  
}

**测试数据（输入、输出）：**



