**华中科技大学计算机科学与技术学院**

**C语言课程设计报告**

题目: 科研项目信息管理系统

专 业： 计算机科学与技术

班 级： 校交1601班

学 号： U201610270

姓 名： 蒋苡杭

成 绩：

指导教师： 甘早斌

**完成日期：** 2017**年 9 月 8 日**

**目 录**

[一、系统需求分析 1](#_Toc463605737)

[1.1数据维护功能 1](#_Toc463605738)

[1.2数据查询功能 1](#_Toc463605739)

[1.3数据统计功能 2](#_Toc463605740)

[1.4数据存取功能 2](#_Toc463605741)

[1.5辅助功能 2](#_Toc463605742)

[二、总体设计 2](#_Toc463605743)

[2.1文件模块 3](#_Toc463605744)

[2.2数据维护模块 4](#_Toc463605745)

[2.3数据查询模块 5](#_Toc463605746)

[2.4数据统计模块 6](#_Toc463605747)

[2.5帮助 7](#_Toc463605748)

[三、数据结构设计 7](#_Toc463605749)

[四、详细设计与系统实现 13](#_Toc463605750)

[4.1数据维护 13](#_Toc463605752)

[4.1.1院系信息维护 14](#_Toc463605753)

[4.1.2团队信息维护 17](#_Toc463605754)

[4.1.3项目信息维护 20](#_Toc463605755)

[4.2数据查询 22](#_Toc463605756)

[4.2.1院系信息查询 22](#_Toc463605757)

[4.2.2团队信息查询： 25](#_Toc463605758)

[4.2.3项目信息查询 26](#_Toc463605759)

[4.3数据统计 29](#_Toc463605760)

[4.3.1按院系信息统计：](#_Toc463605761) 29

[4.3.2按团队信息统计 3](#_Toc463605762)4

[4.3.3按项目信息统计 36](#_Toc463605763)

[五、系统实现](#_Toc463605764) 36

[六、运行测试与结果分析 182](#_Toc463605764)

[6.1输入数据 182](#_Toc463605765)

[6.2系统初始化功能展示 184](#_Toc463605766)

[6.3数据维护功能展示 185](#_Toc463605767)

[6.4数据查询功能展示 195](#_Toc463605768)

[6.5数据统计功能展示 198](#_Toc463605769)

[6.6帮助功能展示 200](#_Toc463605770)

[六、总结 201](#_Toc463605772)

[七、参考文献 204](#_Toc463605773)

# 一、系统需求分析

科研项目信息管理系统用于大学管理各个院系的独立研究团队，主要包括院系基本信息，科研团队基本信息和科研项目基本信息，以帮助学校管理人员对各个院系的团队信息以及团队的科研项目信息进行有效地管理，以期提高项目管理的效率。

科研项目信息管理系统要求实现以下几方面的基本功能：

1.1数据维护功能

科研项目信息管理系统的基本信息主要包括以下三类：

（1）院系基本信息：院系名称、负责人姓名、院系联系电话等数据项。

（2）科研团队基本信息：团队名称、负责人姓名、教师人数、研究生人数、所属院系等数据项。

（3）科研项目基本信息：项目编号、项目类型、起始时间、项目经费、项目负责人、所属团队等数据项。

系统应实现以上三种数据的录入、修改、和删除功能。信息录入时，系统应提供快捷方便的数据录入方式；同时应进行自动数据校验，满足数据的正确性，合理性，有效性和依赖性要求。

1.2数据查询功能

系统应实现对三种基础数据的查询功能，提供按多种条件进行查询的方式，具体包括：

i. 院系信息查询功能

（1）以院系负责人为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系负责人为“张三”的院系基本信息。

（2）以院系名称的全部或一部分为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系名称中包含“学院”的所有院系基本信息。

ii. 科研团队信息查询功能

（1）团队名称的全部或一部分为条件查找并显示满足条件的科研团队基本信息。例如，查找并显示团队名称中包含“信息”字样的科研团队基本信息。

（2）以教师人数为条件查找并显示满足条件的科研团队基本信息。例如，查找并显示教师人数>5的所有科研团队基本信息。

iii. 科研项目基本信息查询功能

（1）以项目编号为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示项目编号为 “01217050102”的科研项目基本信息。

（2）以所属团队为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示所属团队为 “信息安全”的所有科研项目基本信息。

上述查询结果中，如果有多条信息被查中，查询结果应生成表格并逐条显示。

1.3数据统计功能

在以上三种基础数据信息的基础上，提供多方面的数据统计功能，具体包括：

（1）统计各院系教师总数，研究生总数，及研究生与教师的人数比（保留 2 位小数），按学生教师人数比值降序排序后，输出统计结果。

（2）统计某年度各院系科研项目数，973 项目数，863 项目数，及科研总经费，按科研项目数降序排序后输出。

（3）统计历年来类别为国家自然科学基金的科研项目数最多的 10 个科研团队，按项目数降序排序后输出科研团队名称，类别为国家自然科学基金的科研项目数，以及项目经费总数。

（4）统计科研项目数和教师人数的比值最高的5个科研团队，按比值（保留两位小数）降序排序后，输出科研团队名称，教师人数，科研项目数，项目数和教师人数比值。

（5）统计科研项目经费最为充足的10个科研项目，按经费降序排序后，输出科研项目名称，经费，以及负责人姓名。

1.4数据存取功能

以上三种信息在程序运行时，以链表结构形式存在于内存中，并且数据的存储采用动态存储的分配方式。同时，在外存上以数据文件形式对数据进行存储，且保证数据在内存和外存两种存储介质上内容的一致性。

1.5辅助功能

帮助功能，主要是对系统的操作方式进行介绍，提供良好人机交互界面的辅助功能。

# 二、总体设计

科研项目信息管理系统需要建立五大功能模块：文件、数据维护、数据查询、数据统计、帮助，功能结构图如图2.1所示：

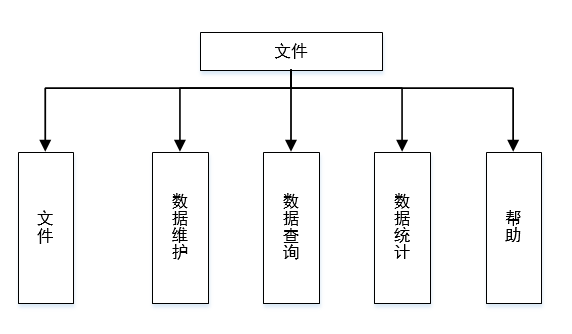


图2.1科研项目管理系统的功能模块

下面为这五个模块及其子模块的功能的介绍。

2.1文件模块

文件模块的功能包括一系列与系统启动和系统结束运行相关的环境维护和数据保障操作。它进一步包括6个子模块：界面初始化、数据保存、退出系统。其功能结构如图2.2所示。

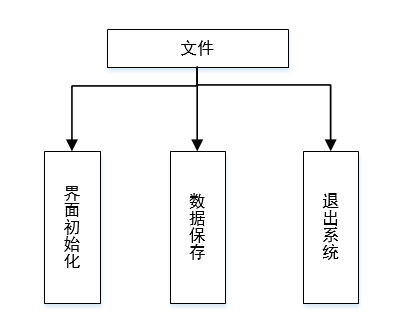


图2.2 文件模块的子模块划分

（1）界面初始化子模块：用于设置控制台窗口显示模式，将屏幕窗口设置为80列和25行文本字符界面，设置窗口标题栏，清屏并显示系统菜单栏和系统状态栏。

（2）数据加载子模块：用于将分别放在多个数据文件中的基础数据读入内存，并构造数据链表，同时输出数据加载相关提示信息。

（3）数据保存模块：用于将链表数据按缺省路径分别保存到各个数据文件。

（4）数据备份子模块：用于将存放在多个数据文件中的基础数据按缺省路径转储到一个数据文件中，供用户在系统数据被破坏或丢失后用来恢复系统。

（5）数据恢复子模块：与数据备份子模块在功能上相对应，能从一个备份数据文件恢复得到备份时间点的系统数据，并将恢复出来的数据加载到内存中，用于提高系统的安全性和可靠性。

（6）退出系统子模块：释放程序运行过程中申请的动态存储区，关闭控制台标准输入和输出设备句柄。清除屏幕窗口信息，结束系统运行。

2.2数据维护模块

数据维护模块完成对三种基础数据信息的录入、修改和删除功能，保证数据的准确性，完整性和有效性。该模块按信息种类划分为院系基本信息代码维护、科研团队基本信息代码维护、科研项目基本信息代码维护三个子模块，如图2.3所示。

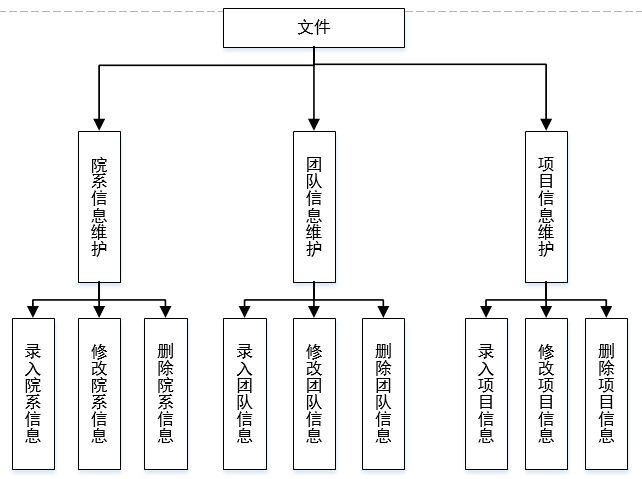


图2.3 数据维护模块的子模块划分

院系信息维护子模块、科研团队信息维护子模块、科研项目信息维护子模块：它们分别用于录入、删除和修改院系信息、科研团队信息、科研项目信息，以保证这三种基础数据的正确性。同时，将这三种数据存入数据链表，并分别保存到院系信息数据文件、科研团队信息数据文件和科研项目信息数据文件，保持内存和外存两种储存介质上数据内容的一致性。

2.3数据查询模块

数据查询模块提供对系统三类基础数据信息按多种条件查询的功能，按信息种类分为院系信息查询，科研团队信息查询和科研项目信息查询三个子模块，如图2.4所示。

（1）院系信息查询子模块：用于提供以院系负责人和院系名称两种条件分别查询院系信息的功能。

（2）科研团队信息查询子模块：用于提供以项目编号和教师人数两种条件分别查询科研团队信息的功能。

（3）科研项目信息查询子模块：用于提供以团队名称的全部或一部分和所属团队两种条件分别查询科研团队信息的功能。

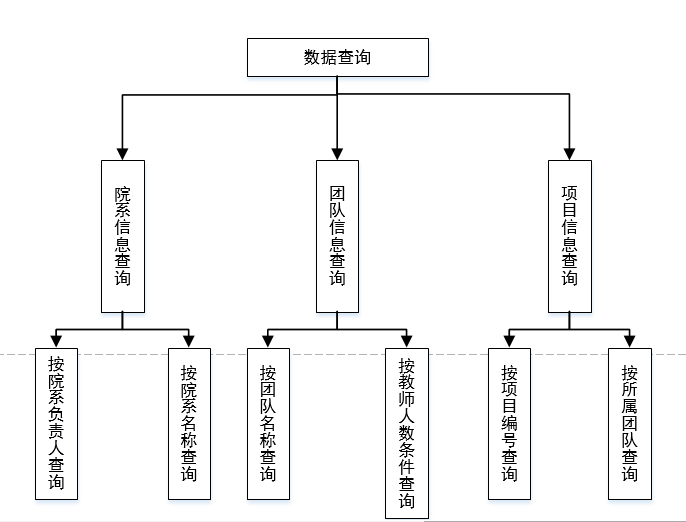


图2.4 数据查询模块的子模块划分

2.4数据统计模块

数据统计模块提供对三种基础数据进行多方面统计的功能。按统计条件，该模块划分为教师和研究生相关信息统计、科研项目和经费统计、科研团队相关信息统计、科研项目和教师人数比统计和科研项目经费统计5个子模块，如图2.5所示。

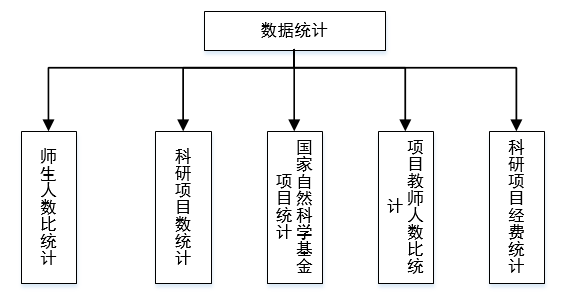


图2.5 数据统计模块子模块划分

（1）师生数量相关信息统计子模块：

统计各院系教师总数，研究生总数，及研究生与教师的人数比（保留 2 位小数），按学生教师人数比值降序排序后，输出统计结果。

（2）科研项目和经费统计子模块：

统计某年度各院系科研项目数，973项目数，863项目数，及科研总经费，科研项目数降序排序后输出。

（3）科研项目最多的团队信息统计子模块：

统计历年来类别为国家自然科学基金的科研项目数最多的 10 个科研团队，按项目数降序排序后输出科研团队名称，类别为国家自然科学基金的科研项目数，以及项目经费总数。

（4）科研项目和教师人数比统计子模块：

统计科研项目数和教师人数的比值最高的 5 个科研团队，按比值（保留两位小数）降序排序后，输出科研团队名称，教师人数，科研项目数，项目数和教师人数比值。

（5）各项目经费统计子模块：

统计科研项目经费最为充足的5个科研项目，按经费降序排序后，输出科研项目名称，经费，以及负责人姓名。

2.5帮助

帮助模块为用户使用系统提供操作帮助信息。分为帮助主题子模块和关于科研项目两个子模块，如图2.6所示。

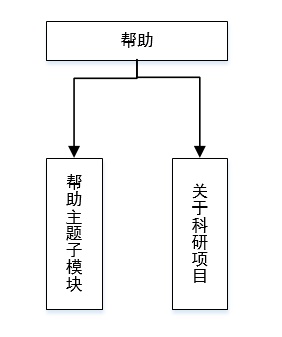


图2.6 帮助模块子模块划分

（1）帮助主题子模块：用于按关键词为用户提供帮助信息。

（2）国家科研项目类型说明子模块：用于显示5种科研项目的详细信息，帮助用户对不同科研项目有更加深入的了解。

# 三、数据结构设计

按照任务要求，系统需要处理的基础信息有三种：院系基本信息、科研团队基本信息和科研项目基本信息。这三种数据信息间存在这样的关联：院系基本信息中的院系名称与科研团队基本信息中的所属院系存在关联，科研团队基本信息中的团队名称与科研项目基本信息表中的所属团队存在关联。院系基本信息、科研团队基本信息和科研项目基本信息在内存中构成一个三向的十字交叉链表，如图3.1所示。

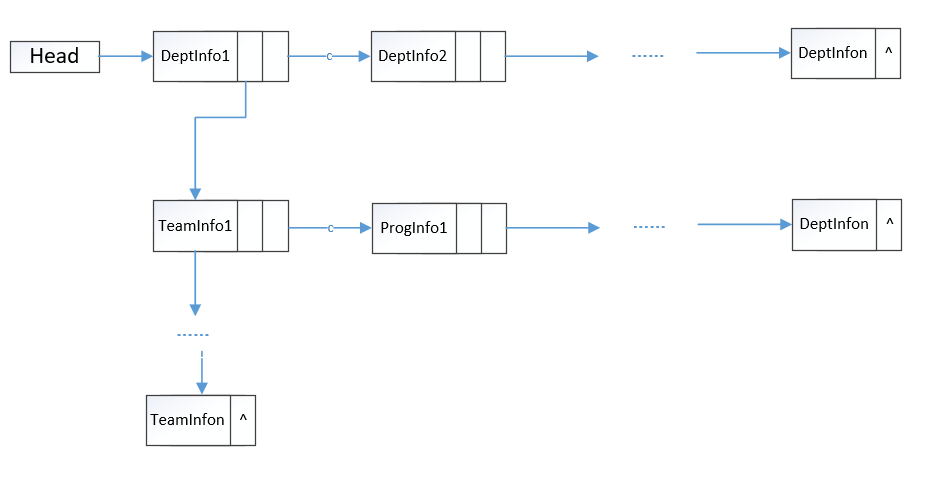


图3.1 三向十字链表存储结构示意图

下面为对本系统所涉及的基础数据和生成数据的数据结构，以及数据在内存和外存中的存储结构的介绍。

1. 院系基本信息表：

typedef struct dept\_node {

char dept\_name[20]; /\*\*< 院系名称\*/

char dmgr\_name[12]; /\*\*< 院系负责人\*/

char telno[15]; /\*\*< 联系电话\*/

struct dept\_node \*next; /\*\*< 指向下一结点的指针\*/

struct team\_node \*tnext; /\*\*< 指向科研团队基本信息支链的指针\*/

} DEPT\_NODE;

院系基本信息表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据结构名称：院系基本信息表 | | | 数据结构标识：DeptInfo | |
| 数据项名称 | 数据项标识 | 类型及长度 | | 示例 |
| 院系名称 | dept\_name | char[20] | | “计算机学院” |
| 院系负责人 | dmgr\_name | char[12] | | “冯丹” |
| 联系电话 | telno | char[15] | | “027-85693623” |

* 内存中的存储结构：存放在十字交叉链表的主链节点上。每个主链节点除了保存下一个结点的地址外，还保存该分类对应的科研团队基本信息链表的头结点地址，如图3.1所示。
* 数据文件中的存储结构：每条信息作为一条记录放到二进制文件。

1. 科研团队基本信息表：

typedef struct team\_node {

char team\_name[30]; /\*\*< 团队名称\*/

char tmgr\_name[12]; /\*\*< 团队负责人姓名\*/

int team\_prof\_amount; /\*\*< 团队教师数\*/

int team\_grd\_amount; /\*\*< 团队研究生数\*/

struct prog\_node \*pnext; /\*\*< 指向科研项目基本信息支链的指针\*/

struct team\_node \*next; /\*\*< 指向下一结点的指针\*/

char dept\_name[20]; /\*\*< 所属院系名称\*/

} TEAM\_NODE;

科研团队基本信息表如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称：科研团队基本信息表 | | 数据结构标识：TeamInfo | |
| 数据项名称 | 数据项标识 | 类型及长度 | 示例 |
| 团队名称 | team\_name | char[30] | “Computer” |
| 团队负责人 | tmgr\_name | char[12] | “李四” |
| 教师人数 | team\_prof\_amount | int | 5 |
| 研究生人数 | team\_grd\_amount | int | 20 |
| 所属院系 | dept\_name | char[20] | “Computer” |

* 在内存中的存储结构：存放在十字交叉链表中相应院系信息结点的科研团队基本信息结点上，每个结点除了保存下一个科研团队基本信息的结点的地址外，还保存该团队对应的科研项目基本信息链表的头结点地址，如图3.1所示。
* 数据文件中的存储结构：每条信息作为一条记录放到二进制文件。

1. 科研项目基本信息表

typedef struct prog\_node {

char prog\_num[15]; /\*\*< 项目编号\*/

char type; /\*\*< 项目类别\*/

char st\_date[8]; /\*\*< 起始时间\*/

char pmgr\_name[12]; /\*\*< 项目负责人姓名\*/

char team\_name[30]; /\*\*< 所属团队\*/

float prog\_fee; /\*\*< 项目经费\*/

struct prog\_node \*next; /\*\*< 指向下一结点的指针\*/

} PROG\_NODE;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据结构名称：科研项目基本信息表 | | | 数据结构标识：ProgInfo | |
| 数据项名称 | 数据项标识 | 类型及长度 | | 示例 |
| 项目编号 | prog\_num | char[15] | | “0121210044” |
| 项目类别 | type | char | | '1' |
| 起始时间 | st\_date | char[8] | | “2015/03” |
| 项目经费 | prog\_fee | float | | 23.5 单位:万元 |
| 项目负责人 | pmgr\_name | char[12] | | “王五” |
| 所属团队 | team\_name | char[30] | | “网络安全” |

* 在内存中的存储结构：存放在十字交叉链表中相应科研团队信息结点的科研项目基本信息结点上，每个结点保存下一个项目信息的结点，如图3.1所示。
* 数据文件的存储结构：每条信息作为一条记录放到二进制文件。

1. 各院系学生教师人数比表如下:

typedef struct rate\_node {

char dept\_name[20]; /\*\*< 院系名称\*/

int dept\_prof\_amount; /\*\*< 团队教师数\*/

int dept\_grd\_amount; /\*\*< 团队研究生数\*/

float proportion; /\*\*< 学生教师人数比\*/

struct rate\_node \*next; /\*\*< 指向下一结点的指针\*/

} RATE\_NODE;

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称：院系学生教师人数比信息表 数据结构标识：StuProfRate | | | |
| 数据项名称 | 数据项标识 | 类型及长度 | 示例 |
| 院系名称 | dept\_name | char[20] | “Computer” |
| 研究生人数 | dept\_grd\_amount | int | 20 |
| 教师人数 | dept\_prof\_amount | int | 5 |
| 学生教师人数比 | stu\_prof\_rate | float | 4.00 |

* 内存中的存储结构：存放在各院系师生人数情况单向链表的结点上。每个结点保存下一个院系人数结点的地址。
* 数据文件中的存储结构：不存入外存。

1. 院系科研项目情况表

typedef struct dept\_prog\_node {

int dept\_prog\_amount1; /\*\*< 973 项目数\*/

int dept\_prog\_amount3; /\*\*< 863 项目数\*/

float dept\_fee; /\*\*< 科研总经费\*/

int dept\_prog\_amount; /\*\*< 科研项目数\*/

char dept\_name[20]; /\*\*< 院系名称\*/

struct dept\_prog\_node \*next; /\*\*< 指向下一结点的指针\*/

} DEPT\_PROG\_NODE;

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称：科研项目和经费统计信息表 数据结构标识：DeptProgInfo | | | |
| 数据项名称 | 数据项标识 | 类型及长度 | 示例 |
| 973 项目数 | dept\_prog\_amount1 | int | 15 |
| 863 项目数 | dept\_prog\_amount3 | int | 20 |
| 科研总经费 | dept\_fee | float | 2.18 单位:万元 |
| 科研项目数 | dept\_prog\_amount | int | 30 |
| 院系名称 | dept\_name | char[20] | “Computer” |

* 内存中的存储结构：存放在各院系科研项目单向链表的结点上。每个结点保存下一个科研项目信息结点的地址。
* 数据文件中的存储结构：不存入外存。

1. 各团队的国家基金项目表

typedef struct most\_prog\_node {

char team\_name[30]; /\*\*< 团队名称\*/

int team\_prog\_amount2; /\*\*< 国家基金项目数\*/

float prog\_fee2; /\*\*< 国家基金科研经费\*/

struct income\_node \*next; /\*\*< 指向下一结点的指针\*/

} MOST\_PROG\_NODE;

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称：团队科研项目信息表 数据结构标识：TeamProgInfo | | | |
| 数据项名称 | 数据项标识 | 类型及长度 | 示例 |
| 团队名称 | team\_name | char[30] | “网络安全” |
| 国家基金项目数 | team\_prog\_amount2 | int | 25 |
| 国家基金科研经费 | prog2\_fee | float | 5.23 单位:万元 |

* 内存中的存储结构：存放在各团队国家基金项目单向链表的结点上。每个结点保存下一个团队相关信息结点的地址。
* 数据文件中的存储结构：不存入外存。

1. 团队项目和教师人数情况统计

typedef struct prog\_prof\_rate\_node {

char team\_name[30]; /\*\*< 团队名称\*/

int team\_prof\_amount; /\*\*< 团队教师人数\*/

int team\_prog\_amount; /\*\*< 团队科研项目数\*/

float team\_prog\_prof\_rate; /\*\*< 项目教师人数比\*/

struct prog\_prof\_rate\_node \*next; /\*\*< 指向下一结点的指针\*/

} PROG\_PROF\_RATE\_NODE;

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称：项目教师人数比高的团队信息表 数据结构标识：ProgProfRate | | | |
| 数据项名称 | 数据项标识 | 类型及长度 | 示例 |
| 团队名称 | team\_name | char[30] | “网络安全” |
| 教师人数 | team\_prof\_amount | int | 15 |
| 科研项目数 | team\_prog\_amount | int | 30 |
| 项目人数比 | team\_prog\_prof\_rate | float | 2.00 |

* 内存中的存储结构：存放在各项目教师人数比的单向链表的结点上。每个结点保存下一个信息结点的地址。
* 数据文件中的存储结构：不存入外存。

1. 科研项目经费统计信息表

typedef struct prog\_fee\_node {

char team\_name[30]; /\*\*< 所属团队名称\*/

char pmgr\_name[12]; /\*\*< 项目负责人姓名\*/

char prog\_num[15]; /\*\*< 项目编号\*/

float prog\_fee; /\*\*< 项目经费\*/

struct prog\_fee\_node \*next; /\*\*< 指向下一结点的指针\*/

} PROG\_FEE\_NODE;

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名称：单个项目经费统计信息表 数据结构标识：ProgFeeInfo | | | |
| 数据项名称 | 数据项标识 | 类型及长度 | 示例 |
| 项目编号 | prog\_num | char[15] | "10001" |
| 所属团队 | team\_name | char[30] | "网络安全" |
| 团队科研经费数 | prog\_fee | float | 2.10 |
| 项目负责人 | pmgr\_name | char[12] | “张三” |

* 内存中的存储结构：存放在各项目及经费信息的单向链表的结点上。每个结点保存下一个信息结点的地址。
* 数据文件中的存储结构：不存入外存。

# 四、详细设计与系统实现

4.1数据维护

函数原型：

BOOL MaintainDeptInfo(void) /\*维护院系基本信息\*/

BOOL MaintainTeamInfo(void) /\*维护团队基本信息\*/

BOOL MaintainProgInfo(void) /\*维护项目基本信息\*/

函数功能：进入三类信息维护的子菜单，选择要执行的操作。

详细设计：定义整型变量承接菜单编号的值，赋初值为0，获取用户选择的菜单编号。利用switch-case语句选择菜单编号相应的操作，进行对应的主函数操作后跳出循环，返回值为TRUE。（数据维护，数据查询，数据统计，与这个函数类似，都以它为例子）如图4.1所示。

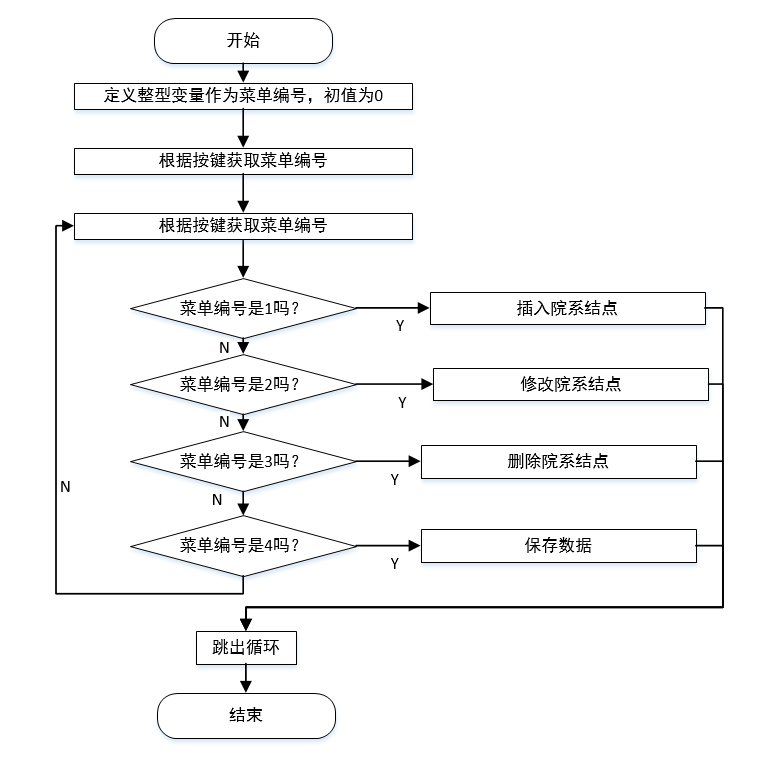


图4.1.1 数据维护模块函数流程图

4.1.1分类信息维护

（1）插入院系信息：

函数原型：void InsertDeptNode(DEPT\_NODE\* hd)

函数功能：插入一个院系信息结点。

详细设计：新建院系信息结点指针pdept\_node，用malloc函数为新的院系结点动态分配新的内存空间，并且把返回值保存在新院系结点。现在pdept\_node指向新结点的内存块，输入新院系结点的成员dept\_name作为院系名称。判断结点成员的院系名称长度是否大于20，如果是，就要输出插入失败的提示信息并退出函数。函数若继续运行，调用SeekDeptNode函数，查找院系信息链表里有没有重名的院系，返回值不是NULL说明有重名院系。判断返回值是不是NULL，如果是，就要输出失败信息并退出函数。函数若继续运行，就输入完新结点剩下的成员数据。将新结点指向团队链头结点的指针置空，将新结点的next指针域指向现在的头结点，再将头指针指向新的结点，这样新结点就成为了头结点，而新结点的next指针域指向原来的头结点，也就成为第二个结点。插入完毕。过程如流程图4.1.2所示。

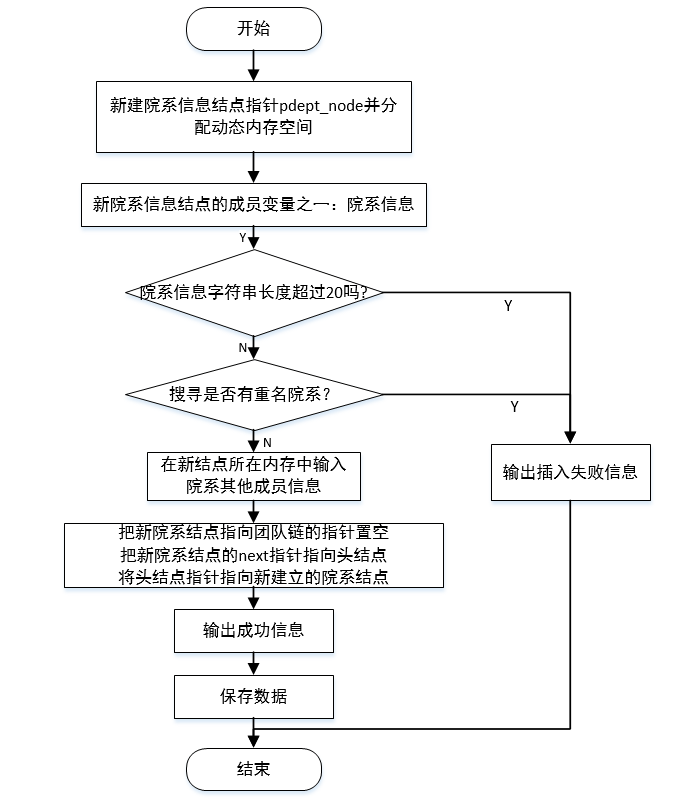


图4.1.2插入院系结点函数流程图

函数原型：DEPT\_NODE \*SeekDeptNode(DEPT\_NODE \*hd, char \*dept\_name)

函数功能：查询指定院系名称的院系信息结点，并返回地址。

详细设计：创建院系指针结点pdept\_node和标记常数find，赋初值为0。使pdept\_node指向院系信息头结点。遍历院系信息链表，若遇到某结点的成员变量之一院系名称与函数参数中的dept\_name(院系名称)相同，则将标记常数置为1，后跳出循环。此时若标记常数find不为0，则说明找到了一个符合条件的结点，则返回此时pdept\_node指向的院系结点的地址，否则说明一个结点都没有找到，返回NULL。如流程图4.1.3所示。

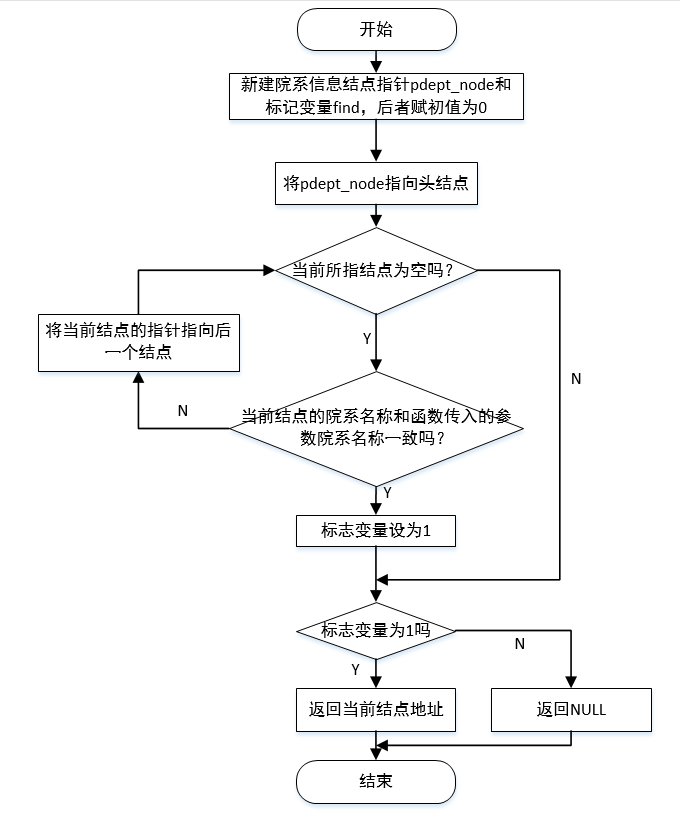


图4.1.3 根据已有院系名称查找院系结点的函数流程图

（2）修改院系信息：

函数原型：BOOL ModifDeptNode(DEPT\_NODE \*hd)

函数功能：修改指定的院系信息结点的成员

详细设计：与4.1.3中修改指定团队信息的函数ModifProgNode类似，不重复叙述。流程图类似于图4.1.3，不重复作图。

（3）删除院系信息：

函数原型：BOOL DelDeptNode(DEPT\_NODE \*hd)

函数功能：删除指定的院系信息结点及其对应的子链.

详细设计：与4.1.2中删除指定团队结点信息的函数DelTeamNode类似，不重复叙述。流程图类似于图4.1.4，不重复作图。

4.1.2基本信息维护

（1）插入团队基本信息：

函数原型：void InsertTeamNode(DEPT\_NODE\* hd)

函数功能：插入一个科研团队信息结点.

详细设计：与4.1.1中插入院系信息结点的函数InsertDeptNode类似，不重复叙述。流程类似于图4.1.2，不重复作图。

（2）修改团队基本信息：

函数原型：BOOL ModifTeamNode(DEPT\_NODE \*hd)

函数功能：修改指定的基本科研团队信息结点

详细设计：与4.1.3中修改指定项目信息结点的函数ModifProgNode类似，不再重复叙述。过程类似于流程图4.1.5，不重复作图。

1. 删除团队基本信息：

函数原型：BOOL DelTeamNode(DEPT\_NODE \*hd)

函数功能：删除指定科研团队基本信息结点.

详细设计：新建院系信息结点指针pdept\_node，输入要删除的团队名称贺团队所属的院系名称，调用SeekDeptNode函数(具体描述见4.1.1插入院系信息部分的第二个函数)使该指针指向具有相同院系名称的团队结点内存空间。现在pdept\_node指向需要删除的团队结点的内存块或者是空结点(当没有找到对应名称的院系结点时，SeekDeptNode会返回NULL)，如果pdept\_node指向空结点，那么就是没有找到对应院系结点，无法删除，输出提示失败信息并且返回FALSE。如果找到对应院系，新建团队信息结点指针pteam\_node\_prior作为前驱指针赋初值为NULL，pteam\_node\_current作为指向当前结点的指针，指向院系链表的团队子链的头结点。

下面搜索链表，定位要删除的结点：如果指向当前结点的指针不为空指针，并且指向当前结点指针的成员之一——团队名称与要查找的“团队名称”不相符合，那么就要进入循环使前驱指针指向当前结点，当前指针指向下一个结点。跳出循环后，要么是找到了要搜索的结点并且被指向当前结点的指针所指向，要么是搜索到链表尾还没有对应的结点。如果是后者，就要输出失败信息并返回FALSE，如果是前者就要进入下一个环节。

下面改变前驱结点，使其“绕过”要删除的当前结点。但要注意，“绕过”的过程根据要删除结点的位置而有所不同。如果要删除结点正好是头结点，那么只要将院系链指向团队子链的头指针指向团队链的下一个结点即可。如果不是头结点，就拿前驱结点的next指针域指向当前结点的下一个结点，是不是头结点则可以根据前驱结点是否为NULL来判断。

最后一步要删除当前指针指向的要删除的结点。但要注意，删除团队结点的同时也要注意删除对应的项目子链。与上面类似，创建指向项目链头结点的当前结点指针以及指向当前结点前面一个结点的前驱指针。当前指针指向的不是空结点的情况下，进入循环将前驱结点指向当前指向结点，将指向当前指向结点的指针指向后一个结点，调用free函数清除前驱结点此时指向结点的信息。跳出循环后，项目子链的结点已经清除，最后再清除要删除的团队链结点(正有指向团队当前结点的指针所指向)即可以完成删除任务。流程图见4.1.4

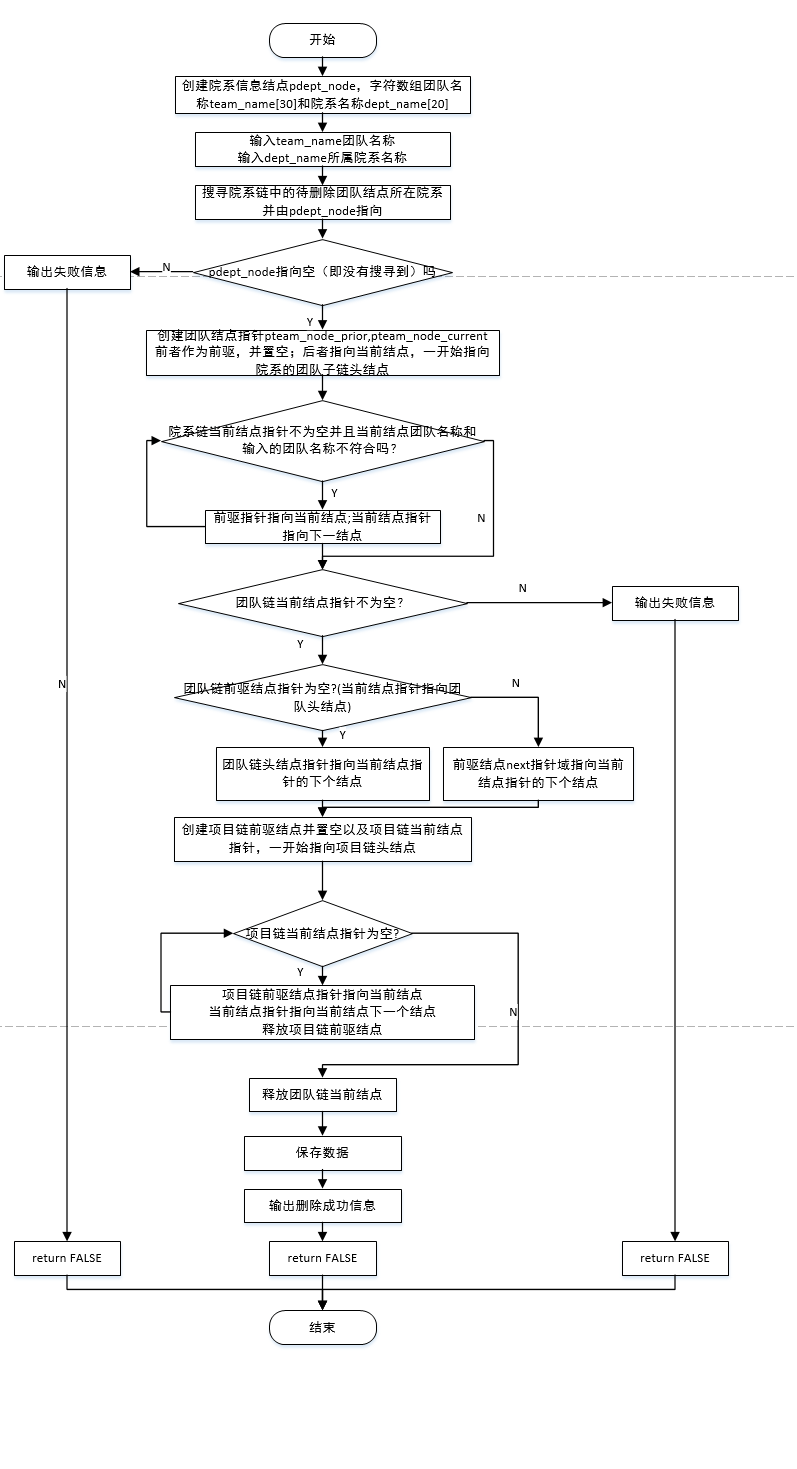


图4.1.4 删除指定科研团队结点的函数流程图

4.1.3项目信息维护

（1）插入项目信息

函数原型：void InsertProgNode(DEPT\_NODE\* hd)

函数功能：插入科研项目信息结点

详细设计：类似于报告4.1.1部分插入院系信息结点函数InsertDeptNode，不重复叙述。流程类似于图4.1.2，不重复作图。

（2）修改项目信息：

函数原型：BOOL ModifProgNode(DEPT\_NODE \*hd)

函数功能：修改指定的科研项目基本信息结点

详细设计：新建项目结点指针pprog\_node\_temp，输入需要修改的项目信息的项目编号和所属团队，调用SeekProgNode函数（描述见下）返回满足条件的项目结点的地址，使得pdept\_node\_temp指向这个结点。由于搜寻满足条件的项目结点可能失败，即pprog\_node可能指向空结点。那么此时就要输出失败信息并 返回FALSE。如果搜寻到了满足项目编号和所属团队的结点，就要对结点成员进行修改。修改方式是创建另一个项目结点指针pprog\_node，分配一块新的内存空间，输入该项目结点的成员变量内容，储存在新内存块中。保存将要被替代的pprog\_node\_temp的指针域到新建院系结点指针pprog\_node\_next。将新内存结点pprog\_node的数据域赋值给将要被修改的结点的数据域。将经过数据域替换的结点指针的next指针域指向保存着被修改的院系结点指针域的指针所指内容。这样被修改的结点又可以与后面的结点形成链表。最后释放新项目结点的内存块。

具体过程如流程图4.1.5所示。

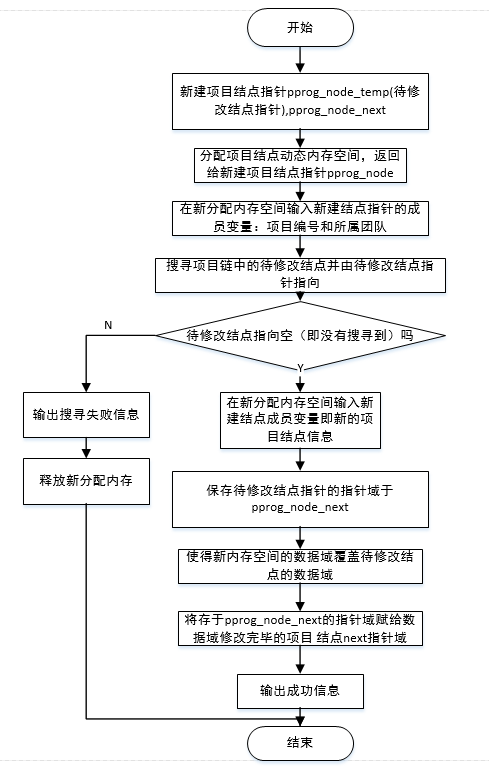


图4.1.5 修改项目结点信息的函数流程图

函数原型：PROG\_NODE \*SeekProgNode(DEPT\_NODE \*hd, char \*team\_name, char \*prog\_num)

函数功能：查询指定项目编号和所属团队的项目信息结点，并返回地址。

详细设计：创建团队指针结点pteam\_node、项目指针结点pprog\_node和标记常数find，赋初值为0。pteam\_node指向对应团队信息结点，pprog\_node指向对应项目信息头结点。遍历项目信息链表，若遇到某结点的成员项目编号与输入的项目编号相同，则将标记常数置为1，后跳出循环。此时若标记常数find不为0，则说明找到了一个符合条件的结点，则返回此时pprog\_node指向的项目结点的地址，否则说明一个结点都没有找到，返回NULL。过程类似4.1.2中查找指定院系结点的函数SeekDeptNode，流程图如4.1.2所示，不再重复作图。

（3）删除项目信息：

函数原型：BOOL DelProgNode(DEPT\_NODE \*hd)

函数功能：删除指定的科研项目基本信息结点

详细设计：类似于4.1.2中删除团队结点的函数DelTeamNode，不重复叙述。

流程类似于图4.1.4，不重复作图。

4.2数据查询

函数原型：

BOOL QueryDeptInfo(void); /\*查询院系基本信息\*/

BOOL QueryProgInfo(void); /\*查询项目基本信息\*/

BOOL QueryTeamInfo(void); /\*查询团队基本信息\*/

函数功能：进入三类信息查询的子菜单，选择要执行的操作。

详细设计：三个函数均类似于4.1部分的MaintainDeptInfo函数，不再重复叙述。具体过程类似于图4.1.1，不再重复作图。

4.2.1院系信息查询

（1）按院系负责人查询：

函数原型：void SeekDmgr(void)

函数功能：根据输入的院系负责人姓名查找院系基本信息

详细设计：创建字符数组dmgr\_name用于输入要查询的院负责人姓名作为查询条件，创建院系信息结点指针find用于遍历链表结构，创建标记变量flag，赋初值为0。输入院系负责人姓名作为查询条件。让遍历指针指向头结点，当遍历指针不为空时，进入循环，判断院系信息遍历指针的成员院系负责人姓名是否与输入的院系负责人姓名相一致，如果是，将标记变量赋值为1，输出此时的遍历结点的成员如院系名称等院系基本信息。将遍历结点指针指向下一个结点后，进行判断，进入下一轮循环。当遍历结点指向空从而跳出循环后，如果flag为0，则说明一个符合的院系结点都没有找到，输出失败提示信息。否则，输出成功信息，结束函数运行。如流程图4.2.1所示。

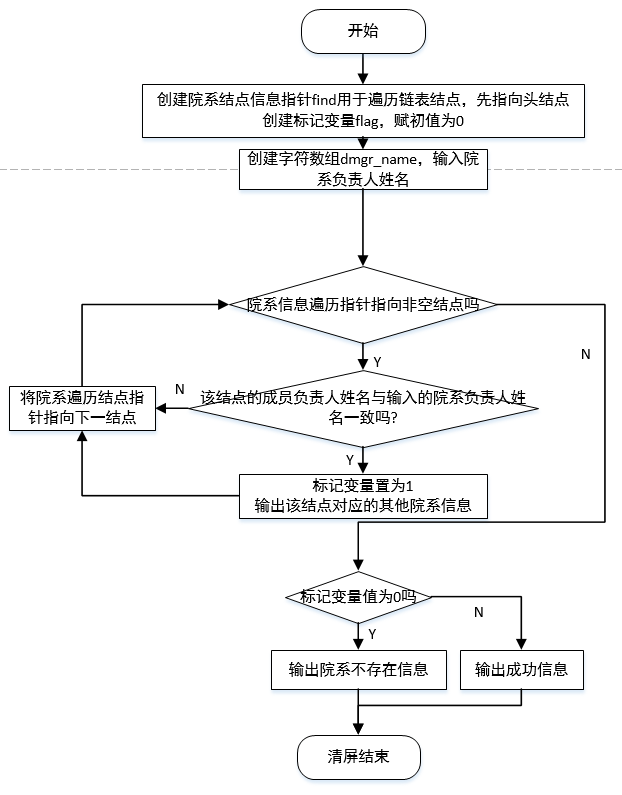


图4.2.1 根据院系负责人姓名查询对应院系的函数的流程图

（2）按院系名称模糊查询：

函数原型：void SeekDeptPart(void)

函数功能：以院系名称的全部或一部分为条件来查找院系基本信息

详细设计：创建字符数组dept\_name用于输入要查询的院系的名称作为查询条件，创建院系信息结点指针find用于遍历链表结构，创建标记变量flag，赋初值为0。输入院系名称作为查询条件。让遍历指针指向头结点，当遍历指针不为空时，进入循环，判断院系信息遍历指针的成员院系名称是否与输入的院系名称相一致，如果是，将标记变量赋值为1，输出此时的遍历结点的成员如院系负责人等院系基本信息。将遍历结点指针指向下一个结点后，进行判断，进入下一轮循环。当遍历结点指向空从而跳出循环后，如果标志变量为0，则说明一个符合的院系结点都没有找到，输出失败提示信息。否则，输出成功信息，结束函数运行。如流程图4.2.2所示。

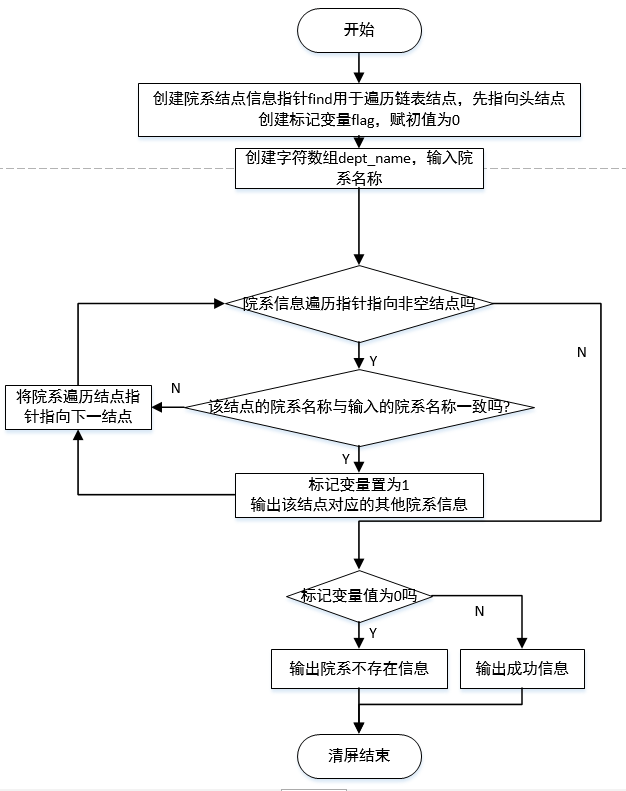


图4.2.2 根据输入的院系名称模糊查找对应院系的函数的流程图

4.2.2科研团队信息查询：

（1）按团队名称模糊查询：

函数原型：void SeekTeamPart(void)

函数功能：以团队名称的全部或一部分为条件来查找团队基本信息

详细设计：设计与4.2.1中以院系名称模糊查询院系基本信息的函数SeekDeptPart类似，不再重复叙述。过程与流程图4.2.2类似，不再重复作图。

（2）按教师人数查询

函数原型：void SeekTeamProf(void)

函数功能：以教师人数为条件查找满足条件的科研团队基本信息

详细设计：创建院系结点类型指针dept\_sech用于遍历院系信息链表，创建标志变量flag赋初值为0，字符数组form用于储存输入的表达式的数字字符串，char类型变量form\_sign用于储存输入的符号。正确输入查询公式后，利用atoi函数将数字字符串转化为int型数字form\_num。将dept\_sech指向头结点，当遍历院系信息指针指向非空结点时，进入第一层循环，创建团队信息遍历结点指针team\_sech指向院系信息链表指针的团队子链头结点。当遍历团队信息指针指向非空结点时，进入第二层循环，根据输入的符号利用switch-case语句编写相应判断条件。例如：当输入的符号form\_sign为'>'时，判断当前结点对应的教师人数是否大于输入的数字form\_sign。如果是，将标记变量设置为1，输出对应的团队结点的其他信息。跳出switch-case语句，将遍历团队信息指针指向下一个结点。进入下一次循环，直到跳出第二层循环，将遍历院系信息结点指针指向下一个院系结点，进行下一次第一层循环，直到跳出第一层循环。判断标记变量是否为0，如果是，输出查找满足条件的团队结点失败的信息，否则输出成功查找的信息，函数结束运行。过程类似于流程图4.2.3，不重复作图。。

4.2.3科研项目信息查询

（1）按项目编号查询：

函数原型：void SeekProgNum(void)

函数功能：按项目编号为条件查找科研项目信息

详细设计：与4.2.1中按照院系负责人姓名查找院系信息的函数SeekDmgr类似，只是将院系负责人姓名换成了项目编号进行比对，不再重复叙述。过程类似流程图4.2.1，不再重复作图。

（2）按所属团队查询

函数原型：void SeekProgTeam(void)

函数功能：按所属团队为条件查找满足条件的科研项目信息

详细设计：创建院系结点类型指针dept\_sech用于遍历院系信息链表，创建标志变量flag赋初值为0，字符型指针prog\_type用于存放项目类型名字符串，字符数组team\_name用于输入团队名称作为查询条件。输入项目所属团队名称，将遍历院系信息链表指针dept\_sech指向头结点。当遍历院系信息指针指向非空结点时，进入第一层循环，创建团队信息遍历结点指针team\_sech指向院系信息链表指针的团队子链头结点。当遍历团队信息指针指向非空结点时，进入第二层循环，创建项目信息遍历结点指针prog\_sech指向团队链表的项目子链头结点。当遍历项目信息指针指向非空结点时，进入第三层循环。利用switch-case语句匹配与项目类别所对应的说明项目类型的字符串常量（如“973计划”），使prog\_type指向该字符串。判断遍历项目信息结点指针的成员之一——所属团队是否与查询条件团队信息相同，如果是，则输出此项目结点对应的剩余的项目信息并且将flag赋值为1（其中项目结点成员之一——类型字符串的输出就是由上面switch-case语句匹配而来）。将项目遍历结点指针指向下一个项目结点后进入下一次循环。跳出第三层循环后，将团队遍历结点指针指向下一个团队结点后进入下一次循环。跳出第二层循环后，将院系遍历结点指针指向下一个院系结点后进入下一次循环。跳出第一层循环后，如果flag为0，则说明一个符合的都没有找到，输出失败提示信息。否则，输出成功信息，结束函数运行。如流程图4.2.3所示。

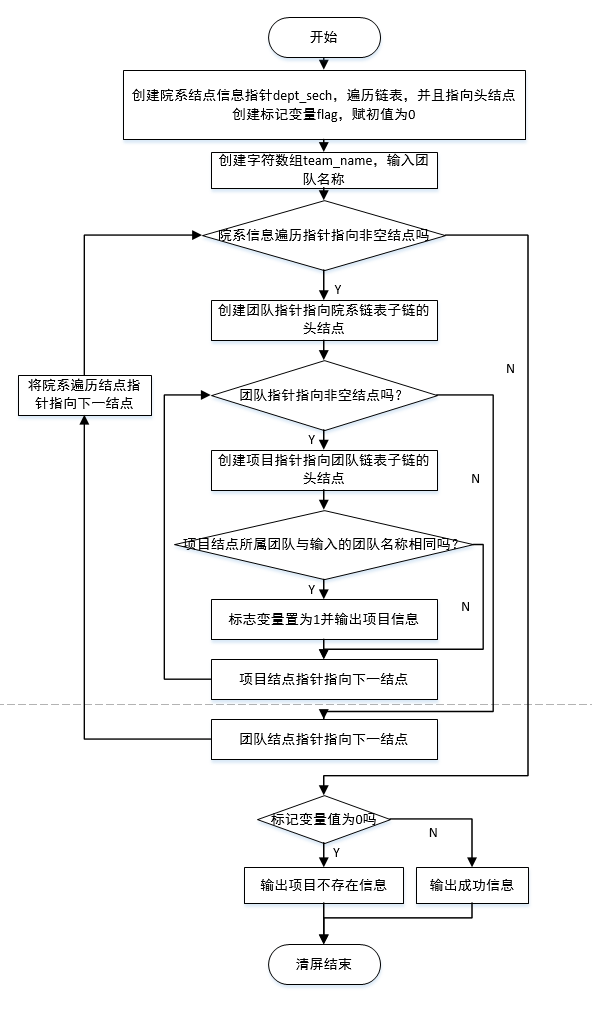


图4.2.3 根据所属团队查找相应项目信息的函数的流程图

4.3数据统计

BOOL StatDept(void) /\*统计院系相关信息\*/

BOOL StatTeam(void) /\*统计团队相关信息\*/

BOOL StatProg(void) /\*统计项目基相关信息\*/

函数功能：进入三类信息统计的子菜单，让用户选择要执行的操作。

详细设计：与4.1数据维护部分函数几乎一模一样，不再重复叙述。过程类似于流程图4.1.1，不再重复作图。

4.3.1按院系统计：

（1）统计学生教师人数比：

函数原型：void StatRateInfo(void)

函数功能：按院系统计学生教师人数比并输出相应结果

详细设计：将为了用于统计这一信息而定义的新的结构体命名为师生比率信息结点。创建遍历院系信息结点指针pdept\_node，用于遍历院系信息链。创建遍历团队信息结点类型指针pteam\_node，用于遍历团队信息链。创建项目信息结点指针prate\_node\_temp和prate\_node\_ret用于分别指向暂存数据的新建结点以及新链表的头结点，并且先使得比率链头结点指向NULL。将遍历院系信息链表指针pdept\_node指向头结点。当遍历院系信息指针指向非空结点时，进入第一层循环，定义两个整型变量分别用于储存该院系的教师人数和学生人数。将比率信息暂存数据结点指针指向新分配的一块内存。将遍历团队信息结点指向此时院系结点的头指针，当遍历团队信息指针指向非空结点时，进入第二层循环，将团队信息结点对应的教师人数和学生人数加到相应的院系教师人数和学生人数变量上。将团队遍历结点指针指向下一个团队结点后进入下一次循环。跳出第二层循环后，将该院系的教师学生人数分别赋给比率信息暂存数据结点相应的成员变量，然后计算出该成员结点的学生人数比，赋给它相应的成员变量。让此时数据域输入完成的比率信息暂存数据结点的指针域指向头结点，再让头结点指向暂存数据结点，这样完成一次循环。

跳出第一层循环后，调用SortRateInfo函数(详细描述见下面)对比率链信息进行排序。然后让暂存数据结点指针作为遍历链表指向比率链头结点，输出相应信息。结束输出后输出提示信息，退出函数。如流程图4.3.1所示

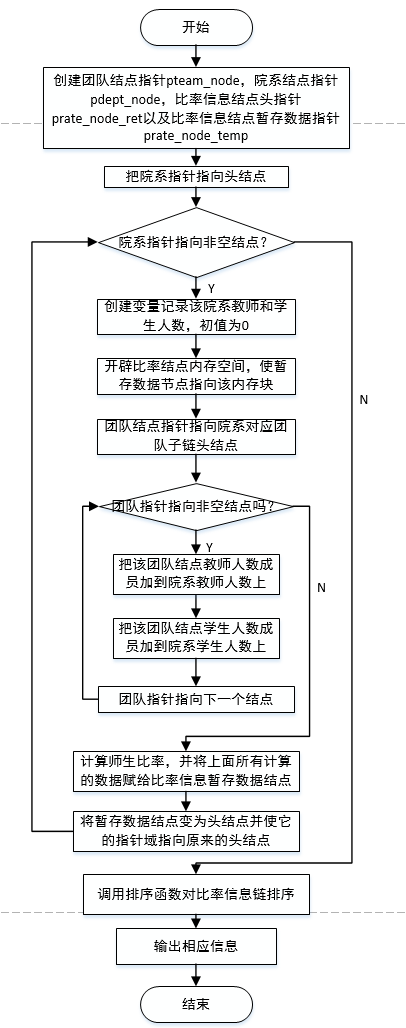


图4.3.1 统计学生教师人数比信息的函数的流程图

函数原型：void SortRateInfo(RATE\_NODE \*rate\_hd)

函数功能：对院系学生教师人数比信息链进行排序

详细设计：创建4个指针，prate\_node\_prior定位指针用于每一趟排序时指向序列的第1个结点，prate\_node\_after遍历指针用于指向序列的第2个结点并随着排序的进行沿链表向后移动，指针prate\_node\_cur最大指针用于指向这一趟排序中成员之一——学生教师人数比最大的那个结点，暂存数据指针prate\_node\_temp指向一块备用的动态储存区，用于每一趟排序后结点的数据域交换和指针域的保存。先把定位指针指向头结点，当定位指针的下一个结点不是空结点时，进入第一层循环，将cur指针指向prior指针，作为一开始的“最大结点”。after指针指向prior后一个结点。当after指针不为空时，判断最大指针所指结点的比值是否小于遍历指针所指结点的比值，如果是，“最大指针”要指向遍历结点，如果不是，最大指针依然指向原结点。然后遍历指针后移一位，进入下一次循环再比较。经过这样第一次循环，最大值就找到了。下面如果定位指针和最大结点指针指向的结点不同，就意味着最大的那个结点并不在定位指针的指向的“头部”（此处头部不一定是整个链的头部，而是代表从定位指针开始往后的头部，即数据最大的位置），就要进行数据域的交换。利用暂存数据指针作为中间量，再使用交换算法分别交换数据域，保留指针域。完成后，定位指针后移一位，再次循环，寻找次大值。完成排序后，释放暂存数据结点所占内存空间。如流程图4.3.2所示。

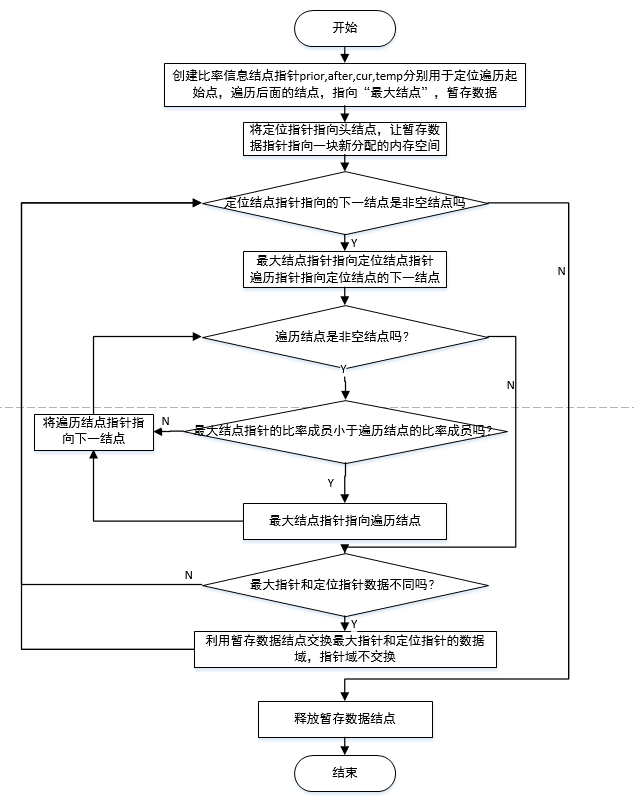


图4.3.2 对学生教师人数比结点链排序的函数的流程图

（2）某年度院系科研项目数统计：

函数原型：void StatDeptInfoPre(void);

函数功能：按院系和输入年份统计各院系科研项目情况

详细设计：创建字符型数组year，用于存放输入的年份字符串。判断年份输入的长度是否符合规范。创建项目统计类型指针head，用于指向项目类型统计链的头结点。创建项目类型统计类型指针temp，用于遍历新建列表，输出信息。调用StatDeptInfo函数，返回头结点地址给head指针。调用SortDpInfo函数对链表进行排序。当temp指向非空结点时，输出该结点对应的信息，将该结点指针指向下一个结点后，进入下一次循环。结束输出后清屏退出函数。

函数原型：DEPT\_PROG\_NODE\* StatDeptInfo(DEPT\_NODE \*hd, char year[5])函数功能：根据输入年份建立院系项目统计信息链并返回头结点

详细设计：创建院系结点类型指针pdept\_node用于遍历院系链，创建团队结点类型指针pteam\_node用于遍历团队链，创建项目结点类型指针pprog\_node用于遍历项目链，创建项目统计类型指针pdp\_node\_temp用于指向新开辟的内存空间并承接数据，成为新链表中的新建结点，创建项目统计类型头指针pdp\_node\_ret 指向NULL，头结点的地址最后需要被返回。

进入第一层循环，用于遍历院系链，创建int类型变量 prog973，prog863，progsum分别代表某年都973项目数，863项目数，和科研项目总数，创建float类型变量dept\_fee用于储存该年度的项目经费，均赋初值为0。

为pdp\_node\_temp动态分配空间，团队遍历指针指向pdept\_node的团队子链的头结点。当pteam\_node不为空时，进入第二层循环，将pprog\_node指向pteam\_node的prog子链。当pprog\_node不为空时，进入第三层循环，先用strncpy函数把pprog\_node的st\_date的前4位复制到prog\_year数组中，末尾加上'\0',然后检查prog\_year是不是那个输入的年份，如果不是就指针后移，进入下一次循环。如果是，就将这个项目的科研项目经费加到dept\_fee（院系这年的经费）上，并且将progsum++,并且检查该项目属于哪一个类型的，如果是对应的973，863类型就要相应的int变量加1。接着pprog\_node指针后移，跳出第三层循环后，pteam\_node指针后移。

跳出第二层循环后，将统计得到的dept\_fee,prog973,dept\_name等常量值赋值给pdp\_node\_temp对应的数据域。这时，pdp\_node\_temp的数据已经完备，将它的next指针域指向pdp\_node\_ret，然后将pdp\_node\_ret指向pdp\_node\_temp（这样下一次循环的时候结点的next指针域就会指向已经创建好的旧结点），直到跳出循环。最后，返回pdp\_node\_ret作为头结点。流程与图4.3.3类似，不再重复作图。

函数原型：void SortDpInfo(DEPT\_PROG\_NODE\* dp\_hd)

函数功能：将院系项目信息统计链按照科研项目数排序

详细设计：与4.3.1中的SortRateInfo函数相类似，不再重复叙述。过程类似于流程图4.3.2，不再重复作图。

4.3.2按科研团队统计

（2）团队的国家自然科学基金的科研项目数统计：

函数原型：void StatMostProg(void)

函数功能：按团队统计国家自然基金项目数最多的十个科研团队 。

详细设计：创建int类型变量count，赋初值为0，用于计数输出信息的条数。创建DEPT\_NODE类型指针pdept\_node，用于作为头指针遍历链表。创建TEAM\_NODE类型指针pteam\_node，用于指向每个院系的团队子链的头结点，遍历团队链。创建PROG\_NODE类型指针pprog\_node，用于遍历prog链。创建“最多项目”类型指针pmp\_node\_temp和pmp\_node\_ret分别指向暂存数据结点和最多项目链的头结点。第一层循环，遍历院系链。每次循环中遍历团队子链，每次循环过程中创建统计国家自然科学基金项目数量的变量progamount2和该团队国家自然科学基金项目经费总量progfee2，赋初值为0。遍历对应的项目子链，利用项目类别判断其项目类型，如果项目确实是国家自然科学基金项目，则加上相应的项目经费和项目数目。将得到的progfee2，progamount2等常量值赋值给pmp\_node\_temp对应的项目经费和项目数量成员。这时，pmp\_node\_temp的数据已经完备，将它的next指针域指向pmp\_node\_ret，然后将pmp\_node\_ret指向pmp\_node\_temp（这样下一次循环的时候结点的next指针域就会指向已经创建好的旧结点），直到跳出循环。最后，pmp\_node\_ret就可以作为链表的头结点。调用SortMpInfo函数对链表进行排序。当pmp\_node\_temp不为NULL且count<10时，输出该结点对应的信息，并将count加一，将该结点后移后，进入下一次循环。结束输出后清屏退出函数。过程类似于流程图4.3.3。

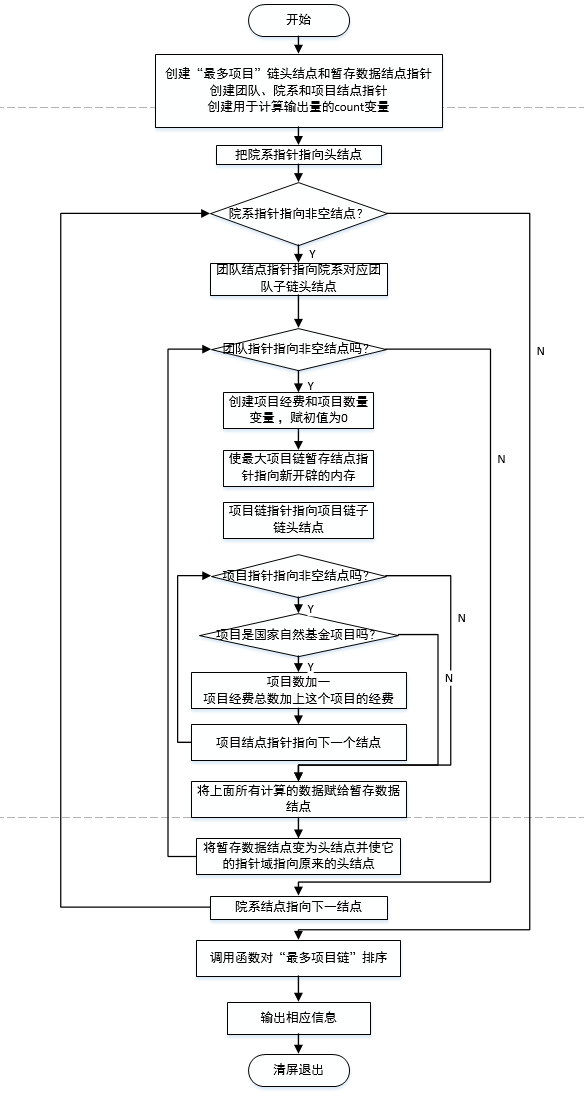


图4.3.3 统计自然基金项目数最多的团队的函数的流程图

函数原型：void SortMpInfo(MOST\_PROG\_NODE \*mp\_hd);函

数功能：按国家自然基金项目数将团队项目链排序

详细设计：与4.3.1中的统计学生教师人数比的SortRateInfo函数几乎一样，不再重复叙述。过程类似于流程图4.3.1，不再重复作图。

（2）团队项目教师人数比值统计：

函数原型：void StatProgRate(void)

函数功能：按团队统计项目教师人数比值情况，并降序输出

详细设计：与4.3.1中的统计学生教师人数比的StatRateInfo函数几乎一样，不再重复叙述。过程类似于流程图4.3.1，不再重复作图。

函数原型：void SortPpInfo(PROG\_PROF\_RATE\_NODE \*pp\_hd

函数功能：按照团队项目教师人数比将团队项目信息链中的结点降序排序

详细设计：与4.3.1中的SortRateInfo函数几乎一样，不再重复叙述。过程类似于流程图4.3.2，不再重复作图。

4.3.3按项目信息统计

（1）项目经费丰厚程度：

函数原型：void StatRichProg(void)

函数功能：统计科研经费最高的项目，输出科研经费最高的10个项目

详细设计：与4.3.2部分统计团队自然基金项目数最多的StatMostProg函数类似，流程类似于图4.3.3，不再重复作图。

# 五、系统实现

**1. 头文件：research.h**

#include <stdio.h>

#include <stdlib.h>

#include <windows.h>

#include <wincon.h>

#include <conio.h>

#include <string.h>

#include <io.h>

#include <fcntl.h>

#include <sys\stat.h>

#include <ctype.h>

#include <time.h>

#ifndef TYPE\_H\_INCLUDED

#define TYPE\_H\_INCLUDED

#define SCR\_ROW 25 /\*屏幕行数\*/

#define SCR\_COL 80 /\*屏幕列数\*/

/\*\*

\*科研项目信息链结点结构

\*/

typedef struct prog\_node {

char prog\_num[15]; /\*\*< 项目编号\*/

char type; /\*\*< 项目类别\*/

char st\_date[8]; /\*\*< 起始时间\*/

char pmgr\_name[12]; /\*\*< 项目负责人姓名\*/

char team\_name[30]; /\*\*< 所属团队\*/

float prog\_fee; /\*\*< 项目经费\*/

struct prog\_node \*next; /\*\*< 指向下一结点的指针\*/

} PROG\_NODE;

/\*\*

科研团队信息链结点结构

\*/

typedef struct team\_node {

char team\_name[30]; /\*\*< 团队名称\*/

char tmgr\_name[12]; /\*\*< 团队负责人姓名\*/

int team\_prof\_amount; /\*\*< 团队教师数\*/

int team\_grd\_amount; /\*\*< 团队研究生数\*/

struct prog\_node \*pnext; /\*\*< 指向科研项目基本信息支链的指针\*/

struct team\_node \*next; /\*\*< 指向下一结点的指针\*/

char dept\_name[20]; /\*\*< 所属院系名称\*/

} TEAM\_NODE;

/\*\*

\*院系信息链结点结构

\*/

typedef struct dept\_node {

char dept\_name[20]; /\*\*< 院系名称\*/

char dmgr\_name[12]; /\*\*< 院系负责人\*/

char telno[15]; /\*\*< 联系电话\*/

struct dept\_node \*next; /\*\*< 指向下一结点的指针\*/

struct team\_node \*tnext; /\*\*< 指向科研团队基本信息支链的指针\*/

} DEPT\_NODE;

/\*\*

\*院系学生教师人数比统计信息链结点结构

\*/

typedef struct rate\_node {

char dept\_name[20]; /\*\*< 院系名称\*/

int dept\_prof\_amount; /\*\*< 团队教师数\*/

int dept\_grd\_amount; /\*\*< 团队研究生数\*/

float proportion; /\*\*< 学生教师人数比\*/

struct rate\_node \*next; /\*\*< 指向下一结点的指针\*/

} RATE\_NODE;

/\*\*

\*科研项目和经费信息链结点结构

\*/

typedef struct dept\_prog\_node {

int dept\_prog\_amount1; /\*\*< 973 项目数\*/

int dept\_prog\_amount3; /\*\*< 863 项目数\*/

float dept\_fee; /\*\*< 科研总经费\*/

int dept\_prog\_amount; /\*\*< 科研项目数\*/

char dept\_name[20]; /\*\*< 院系名称\*/

struct dept\_prog\_node \*next; /\*\*< 指向下一结点的指针\*/

} DEPT\_PROG\_NODE;

/\*\*

\*国家自然基金科研项目最多的团队信息链结点结构

\*/

typedef struct most\_prog\_node {

char team\_name[30]; /\*\*< 团队名称\*/

int team\_prog\_amount2; /\*\*< 国家基金项目数\*/

float prog\_fee2; /\*\*< 国家基金科研经费\*/

struct most\_prog\_node \*next; /\*\*< 指向下一结点的指针\*/

} MOST\_PROG\_NODE;

/\*\*

\*项目教师人数比高的的团队信息链结点结构

\*/

typedef struct prog\_prof\_rate\_node {

char team\_name[30];. /\*\*< 团队名称\*/

int team\_prof\_amount; /\*\*< 团队教师人数\*/

int team\_prog\_amount; /\*\*< 团队科研项目数\*/

float team\_prog\_prof\_rate; /\*\*< 项目教师人数比\*/

struct prog\_prof\_rate\_node \*next; /\*\*< 指向下一结点的指针\*/

} PROG\_PROF\_RATE\_NODE;

/\*\*

\*经费最充足的信息链结点结构

\*/

typedef struct prog\_fee\_node {

char team\_name[30]; /\*\*< 所属团队名称\*/

char pmgr\_name[12]; /\*\*< 项目负责人姓名\*/

char prog\_num[15]; /\*\*< 项目编号\*/

float prog\_fee; /\*\*< 项目经费\*/

struct prog\_fee\_node \*next; /\*\*< 指向下一结点的指针\*/

} PROG\_FEE\_NODE;

/\*\*

\*屏幕窗口信息链结点结点结构

\*/

typedef struct layer\_node {

char LayerNo; /\*\*< 弹出窗口层数\*/

SMALL\_RECT rcArea; /\*\*< 弹出窗口区域坐标\*/

CHAR\_INFO \*pContent; /\*\*< 弹出窗口区域字符单元原信息存储缓冲区\*/

char \*pScrAtt; /\*\*< 弹出窗口区域字符单元原属性值存储缓冲区\*/

struct layer\_node \*next; /\*\*< 指向下一结点的指针\*/

} LAYER\_NODE;

/\*\*

\*标签束结构

\*/

typedef struct labe1\_bundle {

char \*\*ppLabel; /\*\*< 标签字符串数组首地址\*/

COORD \*pLoc; /\*\*< 标签定位数组首地址\*/

int num; /\*\*< 标签个数\*/

} LABEL\_BUNDLE;

/\*\*

\*热区结构

\*/

typedef struct hot\_area {

SMALL\_RECT \*pArea; /\*\*< 热区定位数组首地址\*/

char \*pSort; /\*\*< 热区类别数组首地址\*/

char \*pTag; /\*\*< 热区序号数组首地址\*/

int num; /\*\*< 热区个数\*/

} HOT\_AREA;

LAYER\_NODE \*gp\_top\_layer = NULL; /\*弹出窗口信息链链头\*/

DEPT\_NODE \*gp\_head = NULL; /\*主链头指针\*/

char \*gp\_sys\_name = "科研项目信息管理系统"; /\*系统名称\*/

char \*gp\_team\_info\_filename = "Team.dat"; /\*科研团队基本信息数据文件\*/

char \*gp\_prog\_info\_filename = "Prog.dat"; /\*科研项目信息数据文件\*/

char \*gp\_dept\_info\_filename = "Dept.dat"; /\*院系信息数据文件\*/

char \*gp\_type\_code\_filename = "type.dat"; /\*项目类别代码数据文件\*/

char \*ga\_main\_menu[] = {"文件(F)", /\*系统主菜单名\*/

"数据维护(M)",

"数据查询(Q)",

"数据统计(S)",

"帮助(H)"

};

char \*ga\_sub\_menu[] = {"[S] 数据保存", /\*系统子菜单名\*/

"[X] 退出 Alt+X",

"[D] 院系信息",

"[P] 科研团队信息",

"[C] 科研项目信息",

"[D] 院系信息",

"[P] 科研团队信息",

"[C] 科研项目信息",

"[I] 院系信息统计",

"[T] 科研团队信息统计",

"[C] 科研项目方面统计",

"[T] 帮助主题",

"",

"[A] 科研项目简介"

};

int ga\_sub\_menu\_count[] = {2, 3, 3, 3, 3}; /\*各主菜单项下子菜单的个数\*/

int gi\_sel\_menu = 1; /\*被选中的主菜单项号,初始为1\*/

int gi\_sel\_sub\_menu = 0; /\*被选中的子菜单项号,初始为0,表示未选中\*/

CHAR\_INFO \*gp\_buff\_menubar\_info = NULL;

/\*存放菜单条屏幕区字符信息的缓冲区\*/

CHAR\_INFO \*gp\_buff\_stateBar\_info = NULL;

/\*存放状态条屏幕区字符信息的缓冲区\*/

char \*gp\_scr\_att = NULL; /\*存放屏幕上字符单元属性值的缓冲区\*/

char \*gp\_dept\_code = NULL; /\*存放院系基本信息表的数据缓冲区\*/

char \*gp\_prog\_code = NULL; /\*存放科研项目信息表的数据缓冲区\*/

char \*gp\_team\_code = NULL; /\*存放科研团队信息表的数据缓冲区\*/

char gc\_sys\_state = '\0'; /\*用来保存系统状态的字符\*/

unsigned long gul\_dept\_code\_len = 0; /\*院系信息代码表长度\*/

unsigned long gul\_prog\_code\_len = 0; /\*科研项目信息代码表长度\*/

unsigned long gul\_team\_code\_len = 0; /\*科研团队信息代码表长度\*/

HANDLE gh\_std\_out; /\*标准输出设备句柄\*/

HANDLE gh\_std\_in; /\*标准输入设备句柄\*/

int CreatList(DEPT\_NODE \*\*pphead);

void InitInterface(void); /\*系统界面初始化\*/

void ClearScreen(void); /\*清屏\*/

void ShowMenu(void); /\*显示菜单栏\*/

void PopMenu(int num); /\*显示下拉菜单\*/

void PopUp(SMALL\_RECT \*, WORD, LABEL\_BUNDLE \*, HOT\_AREA \*);

/\*弹出窗口屏幕信息维护\*/

void PopOff(void); /\*关闭顶层弹出窗口\*/

void DrawBox(SMALL\_RECT \*parea); /\*绘制边框\*/

void LocSubMenu(int num, SMALL\_RECT \*parea); /\*主菜单下拉菜单定位\*/

void ShowState(void); /\*显示状态栏\*/

void TagMainMenu(int num); /\*标记被选中的主菜单项\*/

void TagSubMenu(int num); /\*标记被选中的子菜单项\*/

int DealInput(HOT\_AREA \*phot\_area, int \*pihot\_num); /\*控制台输入处理\*/

void SetHotPoint(HOT\_AREA \*phot\_area, int hot\_num); /\*设置热区\*/

void RunSys(DEPT\_NODE \*\*pphd);

/\*系统功能模块的选择和运行\*/

BOOL ExeFunction(int main\_menu\_num, int sub\_menu\_num);

/\*功能模块的调用\*/

unsigned long num\_written;

/\* 该变量用于控制台输出函数 \*/

void CloseSys(DEPT\_NODE \*phd); /\*退出系统\*/

BOOL ShowModule(char \*\*pString, int n);

void gotoxy(int x, int y);

BOOL LoadData(void); /\*数据加载\*/

BOOL SaveData(void); /\*保存数据\*/

BOOL BackupData(void); /\*备份数据\*/

BOOL RemoveData(void); /\*恢复数据\*/

BOOL ExitSys(void); /\*退出系统\*/

BOOL HelpTopic(void); /\*帮助主题\*/

BOOL AboutResearch(void); /\*关于科研项目的介绍\*/

BOOL MaintainDeptInfo(void); /\*维护院系信息\*/

BOOL MaintainProgInfo(void); /\*维护科研项目基本信息\*/

BOOL MaintainTeamInfo(void); /\*维护科研团队基本信息\*/

BOOL QueryDeptInfo(void); /\*查询院系基本信息\*/

BOOL QueryProgInfo(void); /\*查询项目基本信息\*/

BOOL QueryTeamInfo(void); /\*查询团队基本信息\*/

BOOL StatDept(void); /\*统计院系有关信息\*/

BOOL StatTeam(void); /\*统计团队有关信息\*/

BOOL StatProg(void); /\*统计项目有关信息\*/

void InsertDeptNode(DEPT\_NODE\* hd); /\*插入院系信息结点\*/

void InsertTeamNode(DEPT\_NODE\* hd); /\*插入团队信息结点\*/

void InsertProgNode(DEPT\_NODE\* hd); /\*插入项目信息结点\*/

BOOL DelDeptNode(DEPT\_NODE \*hd); /\*删除院系信息结点\*/

BOOL DelTeamNode(DEPT\_NODE \*hd); /\*删除团队信息结点\*/

BOOL DelProgNode(DEPT\_NODE \*hd); /\*删除项目信息结点\*/

BOOL ModifDeptNode(DEPT\_NODE \*hd); /\*修改院系信息结点\*/

BOOL ModifTeamNode(DEPT\_NODE \*hd); /\*修改院系信息结点\*/

BOOL ModifProgNode(DEPT\_NODE \*hd); /\*修改院系信息结点\*/

void StatRateInfo(void);

void SortRateInfo(RATE\_NODE \*rate\_hd); /\*统计院系师生数目有关信息\*/

void StatDeptInfoPre(void);

void SortDeptProgInfo(DEPT\_PROG\_NODE\* dp\_hd);

DEPT\_PROG\_NODE\* StatDeptInfo(DEPT\_NODE \*hd, char year[5]);

/\*统计院系项目数有关信息\*/

void StatMostProg(void);

void SortMostProgInfo(MOST\_PROG\_NODE \*mp\_hd);

/\*统计国家自然基金最多的十个科研团队\*/

void StatProgRate(void);

void SortProgProfInfo(PROG\_PROF\_RATE\_NODE \*pp\_hd);

/\*统计项目教师人数比\*/

void StatRichProg(void);

void SortRichProg(PROG\_FEE\_NODE \*pf\_hd);

/\*统计科研经费最为充足的项目\*/

/\*test部分\*/

void SeekDept\_All(DEPT\_NODE \* head);

void SeekTeam\_All(DEPT\_NODE \* head);

void SeekProg\_All(DEPT\_NODE \* head);

/\*根据输入的院系负责人姓名查找院系基本信息\*/

void SeekDmgr(void);

/\*以院系名称模糊查找院系基本信息\*/

void SeekDeptPart(void);

/\*以团队名称模糊查找团队基本信息\*/

void SeekTeamPart(void);

/\*以教师人数为条件查找满足条件的科研团队基本信息\*/

void SeekTeamProf(void);

/\*按项目编号为条件查找科研项目信息\*/

void SeekProgNum(void);

/\*按所属团队为条件查找满足条件的科研项目信息\*/

void SeekProgTeam(void);

/\*保存系统数据\*/

BOOL SaveSysData(DEPT\_NODE \*phd);

/\*介绍科研项目内容\*/

BOOL AboutResearch(void);

void Intro973(void);

void Intro863(void);

void IntroNationSci(void);

int MyMenu4(char \*\*pString, int n);

int MyMenu3(char \*\*pString, int n);

void MakeDel(); /\*提醒用户删除信息\*/

void EscExit(); /\*按下Esc键后清屏回主菜单\*/

#endif /\*\*< TYPE\_H\_INCLUDED\*/

**2. main.c文件**

#include "research.h"

unsigned long ul;

int main()

{

COORD size = {SCR\_COL, SCR\_ROW}; /\*窗口缓冲区大小\*/

gh\_std\_out = GetStdHandle(STD\_OUTPUT\_HANDLE); /\* 获取标准输出设备句柄\*/

gh\_std\_in = GetStdHandle(STD\_INPUT\_HANDLE); /\* 获取标准输入设备句柄\*/

SetConsoleTitle(gp\_sys\_name); /\*设置窗口标题\*/

SetConsoleScreenBufferSize(gh\_std\_out, size); /\*设置窗口缓冲区大小80\*25\*/

LoadData(); /\*数据加载\*/

InitInterface(); /\*界面初始化\*/

RunSys(&gp\_head); /\*系统功能模块的选择及运行\*/

CloseSys(gp\_head); /\*退出系统\*/

return 0;

}

/\*\*

\* 函数名称: LoadData

\* 函数功能: 将三类基础数据从数据文件载入到内存缓冲区和十字链表中.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: BOOL类型, 功能函数中除了函数ExitSys的返回值可以为FALSE外,

\* 其他函数的返回值必须为TRUE.

\*

\* 调用说明: 为了能够以统一的方式调用各功能函数, 将这些功能函数的原型设为

\* 一致, 即无参数且返回值为BOOL. 返回值为FALSE时, 结束程序运行.

\*/

BOOL LoadData()

{

int Re = 0;

Re = CreatList(&gp\_head);

if (Re < 4 + 8 + 16)

{

/\*数据加载提示信息\*/

printf("\n系统基础数据不完整!\n");

}

printf("\n按任意键进入用户界面\n");

getch();

return TRUE;

}

/\*\*

\* 函数名称: InitInterface

\* 函数功能: 初始化界面.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void InitInterface()

{

WORD att = FOREGROUND\_INTENSITY | FOREGROUND\_BLUE | BACKGROUND\_INTENSITY | BACKGROUND\_RED | BACKGROUND\_GREEN | BACKGROUND\_BLUE; /\*蓝色前景和白色背景\*/

/\*设置控制台屏幕缓冲区字符属性\*/

SetConsoleTextAttribute(gh\_std\_out, att);

ClearScreen(); /\* 清屏\*/

/\*创建弹出窗口信息堆栈，将初始化后的屏幕窗口当作第一层弹出窗口\*/

gp\_scr\_att = (char \*)calloc(SCR\_COL \* SCR\_ROW, sizeof(char));/\*屏幕字符属性\*/

gp\_top\_layer = (LAYER\_NODE \*)malloc(sizeof(LAYER\_NODE));

gp\_top\_layer->LayerNo = 0; /\*弹出窗口的层号为0\*/

gp\_top\_layer->rcArea.Left = 0; /\*弹出窗口的区域为整个屏幕窗口\*/

gp\_top\_layer->rcArea.Top = 0;

gp\_top\_layer->rcArea.Right = SCR\_COL - 1;

gp\_top\_layer->rcArea.Bottom = SCR\_ROW - 1;

gp\_top\_layer->pContent = NULL;

gp\_top\_layer->pScrAtt = gp\_scr\_att;

gp\_top\_layer->next = NULL;

ShowMenu(); /\*显示菜单栏\*/

ShowState(); /\*显示状态栏\*/

return;

}

/\*\*

\* 函数名称: ClearScreen

\* 函数功能: 清除屏幕信息.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void ClearScreen()

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

COORD home = {0, 0};

unsigned long size;

GetConsoleScreenBufferInfo(gh\_std\_out, &bInfo ); /\*取屏幕缓冲区信息\*/

size = bInfo.dwSize.X \* bInfo.dwSize.Y; /\*计算屏幕缓冲区字符单元数\*/

/\*将屏幕缓冲区所有单元的字符属性设置为当前屏幕缓冲区字符属性\*/

FillConsoleOutputAttribute(gh\_std\_out, bInfo.wAttributes, size, home, &ul);

/\*将屏幕缓冲区所有单元填充为空格字符\*/

FillConsoleOutputCharacter(gh\_std\_out, ' ', size, home, &ul);

return;

}

/\*\*

\* 函数名称: ShowMenu

\* 函数功能: 在屏幕上显示主菜单, 并设置热区, 在主菜单第一项上置选中标记.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void ShowMenu()

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

CONSOLE\_CURSOR\_INFO lpCur;

COORD size;

COORD pos = {0, 0};

int i, j;

int PosA = 2, PosB;

char ch;

GetConsoleScreenBufferInfo( gh\_std\_out, &bInfo );

size.X = bInfo.dwSize.X;

size.Y = 1;

SetConsoleCursorPosition(gh\_std\_out, pos);

for (i=0; i < 5; i++) /\*在窗口第一行第一列处输出主菜单项\*/

{

printf(" %s ", ga\_main\_menu[i]);

}

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = FALSE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur); /\*隐藏光标\*/

/\*申请动态存储区作为存放菜单条屏幕区字符信息的缓冲区\*/

gp\_buff\_menubar\_info = (CHAR\_INFO \*)malloc(size.X \* size.Y \* sizeof(CHAR\_INFO));

SMALL\_RECT rcMenu = {0, 0, size.X-1, 0} ;

/\*将窗口第一行的内容读入到存放菜单条屏幕区字符信息的缓冲区中\*/

ReadConsoleOutput(gh\_std\_out, gp\_buff\_menubar\_info, size, pos, &rcMenu);

/\*将这一行中英文字母置为红色，其他字符单元置为白底黑字\*/

for (i=0; i<size.X; i++)

{

(gp\_buff\_menubar\_info+i)->Attributes=BACKGROUND\_BLUE|BACKGROUND\_GREEN | BACKGROUND\_RED;

ch = (char)((gp\_buff\_menubar\_info+i)->Char.AsciiChar);

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_menubar\_info+i)->Attributes |= FOREGROUND\_RED;

}

}

/\*修改后的菜单条字符信息回写到窗口的第一行\*/

WriteConsoleOutput(gh\_std\_out, gp\_buff\_menubar\_info, size, pos, &rcMenu);

COORD endPos = {0, 1};

SetConsoleCursorPosition(gh\_std\_out, endPos); /\*将光标位置设置在第2行第1列\*/

/\*将菜单项置为热区，热区编号为菜单项号，热区类型为0(按钮型)\*/

i = 0;

do

{

PosB = PosA + strlen(ga\_main\_menu[i]); /\*定位第i+1号菜单项的起止位置\*/

for (j=PosA; j<PosB; j++)

{

gp\_scr\_att[j] |= (i+1) << 2; /\*设置菜单项所在字符单元的属性值\*/

}

PosA = PosB + 4;

i++;

}

while (i<5);

TagMainMenu(gi\_sel\_menu); /\*在选中主菜单项上做标记，gi\_sel\_menu初值为1\*/

return;

}

/\*\*

\* 函数名称: ShowState

\* 函数功能: 显示状态条.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明: 状态条字符属性为白底黑字, 初始状态无状态信息.

\*/

void ShowState()

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

COORD size;

COORD pos = {0, 0};

int i;

GetConsoleScreenBufferInfo( gh\_std\_out, &bInfo );

size.X = bInfo.dwSize.X;

size.Y = 1;

SMALL\_RECT rcMenu = {0, bInfo.dwSize.Y-1, size.X-1, bInfo.dwSize.Y-1};

if (gp\_buff\_stateBar\_info == NULL)

{

gp\_buff\_stateBar\_info=(CHAR\_INFO \*)malloc(size.X \* size.Y \* sizeof(CHAR\_INFO));

ReadConsoleOutput(gh\_std\_out, gp\_buff\_stateBar\_info, size, pos, &rcMenu);

}

for (i=0; i<size.X; i++)

{

(gp\_buff\_stateBar\_info+i)->Attributes=BACKGROUND\_BLUE|BACKGROUND\_GREEN|BACKGROUND\_RED;

}

WriteConsoleOutput(gh\_std\_out, gp\_buff\_stateBar\_info, size, pos, &rcMenu);

return;

}

/\*\*

\* 函数名称: TagMainMenu

\* 函数功能: 在指定主菜单项上置选中标志.

\* 输入参数: num 选中的主菜单项号

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void TagMainMenu(int num)

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

COORD size;

COORD pos = {0, 0};

int PosA = 2, PosB;

char ch;

int i;

if (num == 0) /\*num为0时，将会去除主菜单项选中标记\*/

{

PosA = 0;

PosB = 0;

}

else /\*否则，定位选中主菜单项的起止位置: PosA为起始位置, PosB为截止位置\*/

{

for (i=1; i<num; i++)

{

PosA += strlen(ga\_main\_menu[i-1]) + 4;

}

PosB = PosA + strlen(ga\_main\_menu[num-1]);

}

GetConsoleScreenBufferInfo( gh\_std\_out, &bInfo );

size.X = bInfo.dwSize.X;

size.Y = 1;

/\*去除选中菜单项前面的菜单项选中标记\*/

for (i=0; i<PosA; i++)

{

(gp\_buff\_menubar\_info+i)->Attributes=BACKGROUND\_BLUE|BACKGROUND\_GREEN|BACKGROUND\_RED;

ch = (gp\_buff\_menubar\_info+i)->Char.AsciiChar;

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_menubar\_info+i)->Attributes |= FOREGROUND\_RED;

}

}

/\*在选中菜单项上做标记，黑底白字\*/

for (i=PosA; i<PosB; i++)

{

(gp\_buff\_menubar\_info+i)->Attributes=FOREGROUND\_BLUE|FOREGROUND\_GREEN | FOREGROUND\_RED;

}

/\*去除选中菜单项后面的菜单项选中标记\*/

for (i=PosB; i<bInfo.dwSize.X; i++)

{

(gp\_buff\_menubar\_info+i)->Attributes=BACKGROUND\_BLUE|BACKGROUND\_GREEN | BACKGROUND\_RED;

ch = (char)((gp\_buff\_menubar\_info+i)->Char.AsciiChar);

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_menubar\_info+i)->Attributes |= FOREGROUND\_RED;

}

}

/\*将做好标记的菜单条信息写到窗口第一行\*/

SMALL\_RECT rcMenu = {0, 0, size.X-1, 0};

WriteConsoleOutput(gh\_std\_out, gp\_buff\_menubar\_info, size, pos, &rcMenu);

return;

}

/\*\*

\* 函数名称: CloseSys

\* 函数功能: 关闭系统.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void CloseSys(DEPT\_NODE \*hd)

{

DEPT\_NODE \*pDeptNode1 = hd, \*pDeptNode2;

TEAM\_NODE \*pTeamNode1, \*pTeamNode2;

PROG\_NODE \*pProgNode1, \*pProgNode2;

while (pDeptNode1 != NULL) /\*释放十字交叉链表的动态存储区\*/

{

pDeptNode2 = pDeptNode1->next;

pTeamNode1 = pDeptNode1->tnext;

while (pTeamNode1 != NULL) /\*释放科研团队信息支链的动态存储区\*/

{

pTeamNode2 = pTeamNode1->next;

pProgNode1 = pTeamNode1->pnext;

while (pProgNode1 != NULL) /\*释放科研项目信息支链的动态存储区\*/

{

pProgNode2 = pProgNode1->next;

free(pProgNode1);

pProgNode1 = pProgNode2;

}

free(pTeamNode1);

pTeamNode1 = pTeamNode2;

}

free(pDeptNode1); /\*释放主链结点的动态存储区\*/

pDeptNode1 = pDeptNode2;

}

ClearScreen(); /\*清屏\*/

/\*释放存放菜单条、状态条、项目类别代码等信息动态存储区\*/

free(gp\_buff\_menubar\_info);

free(gp\_buff\_stateBar\_info);

/\*关闭标准输入和输出设备句柄\*/

CloseHandle(gh\_std\_out);

CloseHandle(gh\_std\_in);

/\*将窗口标题栏置为运行结束\*/

SetConsoleTitle("运行结束");

return;

}

/\*\*

\* 函数名称: RunSys

\* 函数功能: 运行系统, 在系统主界面下运行用户所选择的功能模块.

\* 输入参数: 无

\* 输出参数: phead 主链头指针的地址

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void RunSys(DEPT\_NODE \*\*phead)

{

INPUT\_RECORD inRec;

DWORD res;

COORD pos = {0, 0};

BOOL bRet = TRUE;

int i, loc, num;

int cNo, cAtt; /\*cNo:字符单元层号, cAtt:字符单元属性\*/

char vkc, asc; /\*vkc:虚拟键代码, asc:字符的ASCII码值\*/

while (bRet)

{

/\*从控制台输入缓冲区中读一条记录\*/

ReadConsoleInput(gh\_std\_in, &inRec, 1, &res);

if (inRec.EventType == MOUSE\_EVENT) /\*如果记录由鼠标事件产生\*/

{

pos = inRec.Event.MouseEvent.dwMousePosition; /\*获取鼠标坐标位置\*/

cNo = gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] & 3; /\*取该位置的层号\*/

cAtt = gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] >> 2;/\*取该字符单元属性\*/

if (cNo == 0) /\*层号为0，表明该位置未被弹出子菜单覆盖\*/

{

if (cAtt > 0 && cAtt != gi\_sel\_menu && gp\_top\_layer->LayerNo > 0)

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

PopMenu(cAtt); /\*弹出鼠标所在主菜单项对应的子菜单\*/

}

}

else if (cAtt > 0) /\*鼠标所在位置为弹出子菜单的菜单项字符单元\*/

{

TagSubMenu(cAtt); /\*在该子菜单项上做选中标记\*/

}

if (inRec.Event.MouseEvent.dwButtonState

== FROM\_LEFT\_1ST\_BUTTON\_PRESSED)

/\*如果按下鼠标左边第一键\*/

{

if (cNo == 0) /\*层号为0，表明该位置未被弹出子菜单覆盖\*/

{

if (cAtt > 0) /\*如果该位置处于热区(主菜单项字符单元)\*/

{

PopMenu(cAtt); /\*弹出鼠标所在主菜单项对应的子菜单\*/

}

/\*如果该位置不属于主菜单项字符单元，且有子菜单弹出\*/

else if (gp\_top\_layer->LayerNo > 0)

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

}

}

else /\*层号不为0，表明该位置被弹出子菜单覆盖\*/

{

if (cAtt > 0) /\*如果该位置处于热区(子菜单项字符单元)\*/

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

/\*执行对应功能函数:gi\_sel\_menu主菜单项号,cAtt子菜单项号\*/

bRet = ExeFunction(gi\_sel\_menu, cAtt);

}

}

}

else if (inRec.Event.MouseEvent.dwButtonState

== RIGHTMOST\_BUTTON\_PRESSED) /\*如果按下鼠标右键\*/

{

if (cNo == 0) /\*层号为0，表明该位置未被弹出子菜单覆盖\*/

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

}

}

}

else if (inRec.EventType == KEY\_EVENT /\*如果记录由按键产生\*/

&& inRec.Event.KeyEvent.bKeyDown) /\*且键被按下\*/

{

vkc = inRec.Event.KeyEvent.wVirtualKeyCode; /\*获取按键的虚拟键码\*/

asc = inRec.Event.KeyEvent.uChar.AsciiChar; /\*获取按键的ASC码\*/

/\*系统快捷键的处理\*/

if (vkc == 112) /\*如果按下F1键\*/

{

if (gp\_top\_layer->LayerNo != 0) /\*如果当前有子菜单弹出\*/

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

}

bRet = ExeFunction(5, 1); /\*运行帮助主题功能函数\*/

}

else if (inRec.Event.KeyEvent.dwControlKeyState

& (LEFT\_ALT\_PRESSED | RIGHT\_ALT\_PRESSED))

{

/\*如果按下左或右Alt键\*/

switch (vkc) /\*判断组合键Alt+字母\*/

{

case 88: /\*Alt+X 退出\*/

if (gp\_top\_layer->LayerNo != 0)

{

PopOff();

gi\_sel\_sub\_menu = 0;

}

bRet = ExeFunction(1, 4);

break;

case 70: /\*Alt+F\*/

PopMenu(1);

break;

case 77: /\*Alt+M\*/

PopMenu(2);

break;

case 81: /\*Alt+Q\*/

PopMenu(3);

break;

case 83: /\*Alt+S\*/

PopMenu(4);

break;

case 72: /\*Alt+H\*/

PopMenu(5);

break;

}

}

else if (asc == 0) /\*其他控制键的处理\*/

{

if (gp\_top\_layer->LayerNo == 0) /\*如果未弹出子菜单\*/

{

switch (vkc) /\*处理方向键(左、右、下)，不响应其他控制键\*/

{

case 37:

gi\_sel\_menu--;

if (gi\_sel\_menu == 0)

{

gi\_sel\_menu = 5;

}

TagMainMenu(gi\_sel\_menu);

break;

case 39:

gi\_sel\_menu++;

if (gi\_sel\_menu == 6)

{

gi\_sel\_menu = 1;

}

TagMainMenu(gi\_sel\_menu);

break;

case 40:

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

}

}

else /\*已弹出子菜单时\*/

{

for (loc=0,i=1; i<gi\_sel\_menu; i++)

{

loc += ga\_sub\_menu\_count[i-1];

}

switch (vkc) /\*方向键(左、右、上、下)的处理\*/

{

case 37:

gi\_sel\_menu--;

if (gi\_sel\_menu < 1)

{

gi\_sel\_menu = 5;

}

TagMainMenu(gi\_sel\_menu);

PopOff();

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

case 38:

num = gi\_sel\_sub\_menu - 1;

if (num < 1)

{

num = ga\_sub\_menu\_count[gi\_sel\_menu-1];

}

if (strlen(ga\_sub\_menu[loc+num-1]) == 0)

{

num--;

}

TagSubMenu(num);

break;

case 39:

gi\_sel\_menu++;

if (gi\_sel\_menu > 5)

{

gi\_sel\_menu = 1;

}

TagMainMenu(gi\_sel\_menu);

PopOff();

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

case 40:

num = gi\_sel\_sub\_menu + 1;

if (num > ga\_sub\_menu\_count[gi\_sel\_menu-1])

{

num = 1;

}

if (strlen(ga\_sub\_menu[loc+num-1]) == 0)

{

num++;

}

TagSubMenu(num);

break;

}

}

}

else if ((asc-vkc == 0) || (asc-vkc == 32)) /\*按下普通键\*/

{

if (gp\_top\_layer->LayerNo == 0) /\*如果未弹出子菜单\*/

{

switch (vkc)

{

case 70: /\*f或F\*/

PopMenu(1);

break;

case 77: /\*m或M\*/

PopMenu(2);

break;

case 81: /\*q或Q\*/

PopMenu(3);

break;

case 83: /\*s或S\*/

PopMenu(4);

break;

case 72: /\*h或H\*/

PopMenu(5);

break;

case 13: /\*回车\*/

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

}

}

else /\*已弹出子菜单时的键盘输入处理\*/

{

if (vkc == 27) /\*如果按下ESC键\*/

{

PopOff();

gi\_sel\_sub\_menu = 0;

}

else if(vkc == 13) /\*如果按下回车键\*/

{

num = gi\_sel\_sub\_menu;

PopOff();

gi\_sel\_sub\_menu = 0;

bRet = ExeFunction(gi\_sel\_menu, num);

}

else /\*其他普通键的处理\*/

{

/\*计算该子菜单中的第一项在子菜单字符串数组中的位置(下标)\*/

for (loc=0,i=1; i<gi\_sel\_menu; i++)

{

loc += ga\_sub\_menu\_count[i-1];

}

/\*依次与当前子菜单中每一项的代表字符进行比较\*/

for (i=loc; i<loc+ga\_sub\_menu\_count[gi\_sel\_menu-1]; i++)

{

if (strlen(ga\_sub\_menu[i])>0 && vkc==ga\_sub\_menu[i][1])

{

/\*如果匹配成功\*/

PopOff();

gi\_sel\_sub\_menu = 0;

bRet = ExeFunction(gi\_sel\_menu, i-loc+1);

}

}

}

}

}

}

}

}

/\*\*

\* 函数名称: PopMenu

\* 函数功能: 弹出指定主菜单项对应的子菜单.

\* 输入参数: num 指定的主菜单项号

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void PopMenu(int num)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

char \*pCh;

int i, j, loc = 0;

if (num != gi\_sel\_menu) /\*如果指定主菜单不是已选中菜单\*/

{

if (gp\_top\_layer->LayerNo != 0) /\*如果此前已有子菜单弹出\*/

{

PopOff();

gi\_sel\_sub\_menu = 0;

}

}

else if (gp\_top\_layer->LayerNo != 0) /\*若已弹出该子菜单，则返回\*/

{

return;

}

gi\_sel\_menu = num; /\*将选中主菜单项置为指定的主菜单项\*/

TagMainMenu(gi\_sel\_menu); /\*在选中的主菜单项上做标记\*/

LocSubMenu(gi\_sel\_menu, &rcPop); /\*计算弹出子菜单的区域位置, 存放在rcPop中\*/

/\*计算该子菜单中的第一项在子菜单字符串数组中的位置(下标)\*/

for (i=1; i<gi\_sel\_menu; i++)

{

loc += ga\_sub\_menu\_count[i-1];

}

/\*将该组子菜单项项名存入标签束结构变量\*/

labels.ppLabel = ga\_sub\_menu + loc; /\*标签束第一个标签字符串的地址\*/

labels.num = ga\_sub\_menu\_count[gi\_sel\_menu-1]; /\*标签束中标签字符串的个数\*/

COORD aLoc[labels.num];/\*定义一个坐标数组，存放每个标签字符串输出位置的坐标\*/

for (i=0; i<labels.num; i++) /\*确定标签字符串的输出位置，存放在坐标数组中\*/

{

aLoc[i].X = rcPop.Left + 2;

aLoc[i].Y = rcPop.Top + i + 1;

}

labels.pLoc = aLoc; /\*使标签束结构变量labels的成员pLoc指向坐标数组的首元素\*/

/\*设置热区信息\*/

areas.num = labels.num; /\*热区的个数，等于标签的个数，即子菜单的项数\*/

SMALL\_RECT aArea[areas.num]; /\*定义数组存放所有热区位置\*/

char aSort[areas.num]; /\*定义数组存放所有热区对应类别\*/

char aTag[areas.num]; /\*定义数组存放每个热区的编号\*/

for (i=0; i<areas.num; i++)

{

aArea[i].Left = rcPop.Left + 2; /\*热区定位\*/

aArea[i].Top = rcPop.Top + i + 1;

aArea[i].Right = rcPop.Right - 2;

aArea[i].Bottom = aArea[i].Top;

aSort[i] = 0; /\*热区类别都为0(按钮型)\*/

aTag[i] = i + 1; /\*热区按顺序编号\*/

}

areas.pArea = aArea;/\*使热区结构变量areas的成员pArea指向热区位置数组首元素\*/

areas.pSort = aSort;/\*使热区结构变量areas的成员pSort指向热区类别数组首元素\*/

areas.pTag = aTag; /\*使热区结构变量areas的成员pTag指向热区编号数组首元素\*/

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

PopUp(&rcPop, att, &labels, &areas);

DrawBox(&rcPop); /\*给弹出窗口画边框\*/

pos.X = rcPop.Left + 2;

for (pos.Y=rcPop.Top+1; pos.Y<rcPop.Bottom; pos.Y++)

{

/\*此循环用来在空串子菜项位置画线形成分隔，并取消此菜单项的热区属性\*/

pCh = ga\_sub\_menu[loc+pos.Y-rcPop.Top-1];

if (strlen(pCh)==0) /\*串长为0，表明为空串\*/

{

/\*首先画横线\*/

FillConsoleOutputCharacter(gh\_std\_out,'-',rcPop.Right-rcPop.Left-3,pos, &num\_written);

for (j=rcPop.Left+2; j<rcPop.Right-1; j++)

{

/\*取消该区域字符单元的热区属性\*/

gp\_scr\_att[pos.Y\*SCR\_COL+j] &= 3; /\*按位与的结果保留了低两位\*/

}

}

}

/\*将子菜单项的功能键设为白底红字\*/

pos.X = rcPop.Left + 3;

att = FOREGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED;

for (pos.Y=rcPop.Top+1; pos.Y<rcPop.Bottom; pos.Y++)

{

if (strlen(ga\_sub\_menu[loc+pos.Y-rcPop.Top-1])==0)

{

continue; /\*跳过空串\*/

}

FillConsoleOutputAttribute(gh\_std\_out, att, 1, pos, &num\_written);

}

return;

}

/\*\*

\* 函数名称: PopUp

\* 函数功能: 在指定区域输出弹出窗口信息, 同时设置热区, 将弹出窗口位置信息入栈.

\* 输入参数: pRc 弹出窗口位置数据存放的地址

\* att 弹出窗口区域字符属性

\* pLabel 弹出窗口中标签束信息存放的地址

pHotArea 弹出窗口中热区信息存放的地址

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void PopUp(SMALL\_RECT \*pRc, WORD att, LABEL\_BUNDLE \*pLabel, HOT\_AREA \*pHotArea)

{

LAYER\_NODE \*nextLayer;

COORD size;

COORD pos = {0, 0};

char \*pCh;

int i, j, row;

/\*弹出窗口所在位置字符单元信息入栈\*/

size.X = pRc->Right - pRc->Left + 1; /\*弹出窗口的宽度\*/

size.Y = pRc->Bottom - pRc->Top + 1; /\*弹出窗口的高度\*/

/\*申请存放弹出窗口相关信息的动态存储区\*/

nextLayer = (LAYER\_NODE \*)malloc(sizeof(LAYER\_NODE));

nextLayer->next = gp\_top\_layer;

nextLayer->LayerNo = gp\_top\_layer->LayerNo + 1;

nextLayer->rcArea = \*pRc;

nextLayer->pContent = (CHAR\_INFO \*)malloc(size.X\*size.Y\*sizeof(CHAR\_INFO));

nextLayer->pScrAtt = (char \*)malloc(size.X\*size.Y\*sizeof(char));

pCh = nextLayer->pScrAtt;

/\*将弹出窗口覆盖区域的字符信息保存，用于在关闭弹出窗口时恢复原样\*/

ReadConsoleOutput(gh\_std\_out, nextLayer->pContent, size, pos, pRc);

for (i=pRc->Top; i<=pRc->Bottom; i++)

{

/\*此二重循环将所覆盖字符单元的原先属性值存入动态存储区，便于以后恢复\*/

for (j=pRc->Left; j<=pRc->Right; j++)

{

\*pCh = gp\_scr\_att[i\*SCR\_COL+j];

pCh++;

}

}

gp\_top\_layer = nextLayer; /\*完成弹出窗口相关信息入栈操作\*/

/\*设置弹出窗口区域字符的新属性\*/

pos.X = pRc->Left;

pos.Y = pRc->Top;

for (i=pRc->Top; i<=pRc->Bottom; i++)

{

FillConsoleOutputAttribute(gh\_std\_out, att, size.X, pos, &ul);

pos.Y++;

}

/\*将标签束中的标签字符串在设定的位置输出\*/

for (i=0; i<pLabel->num; i++)

{

pCh = pLabel->ppLabel[i];

if (strlen(pCh) != 0)

{

WriteConsoleOutputCharacter(gh\_std\_out, pCh, strlen(pCh),

pLabel->pLoc[i], &ul);

}

}

/\*设置弹出窗口区域字符单元的新属性\*/

for (i=pRc->Top; i<=pRc->Bottom; i++)

{

/\*此二重循环设置字符单元的层号\*/

for (j=pRc->Left; j<=pRc->Right; j++)

{

gp\_scr\_att[i\*SCR\_COL+j] = gp\_top\_layer->LayerNo;

}

}

for (i=0; i<pHotArea->num; i++)

{

/\*此二重循环设置所有热区中字符单元的热区类型和热区编号\*/

row = pHotArea->pArea[i].Top;

for (j=pHotArea->pArea[i].Left; j<=pHotArea->pArea[i].Right; j++)

{

gp\_scr\_att[row\*SCR\_COL+j] |= (pHotArea->pSort[i] << 6)

| (pHotArea->pTag[i] << 2);

}

}

return;

}

/\*\*

\* 函数名称: PopOff

\* 函数功能: 关闭顶层弹出窗口, 恢复覆盖区域原外观和字符单元原属性.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void PopOff(void)

{

LAYER\_NODE \*nextLayer;

COORD size;

COORD pos = {0, 0};

char \*pCh;

int i, j;

if ((gp\_top\_layer->next==NULL) || (gp\_top\_layer->pContent==NULL))

{

/\*栈底存放的主界面屏幕信息，不用关闭\*/

return;

}

nextLayer = gp\_top\_layer->next;

/\*恢复弹出窗口区域原外观\*/

size.X = gp\_top\_layer->rcArea.Right - gp\_top\_layer->rcArea.Left + 1;

size.Y = gp\_top\_layer->rcArea.Bottom - gp\_top\_layer->rcArea.Top + 1;

WriteConsoleOutput(gh\_std\_out, gp\_top\_layer->pContent, size, pos, &(gp\_top\_layer->rcArea));

/\*恢复字符单元原属性\*/

pCh = gp\_top\_layer->pScrAtt;

for (i=gp\_top\_layer->rcArea.Top; i<=gp\_top\_layer->rcArea.Bottom; i++)

{

for (j=gp\_top\_layer->rcArea.Left; j<=gp\_top\_layer->rcArea.Right; j++)

{

gp\_scr\_att[i\*SCR\_COL+j] = \*pCh;

pCh++;

}

}

free(gp\_top\_layer->pContent); /\*释放动态存储区\*/

free(gp\_top\_layer->pScrAtt);

free(gp\_top\_layer);

gp\_top\_layer = nextLayer;

gi\_sel\_sub\_menu = 0;

return;

}

/\*\*

\* 函数名称: DrawBox

\* 函数功能: 在指定区域画边框.

\* 输入参数: pRc 存放区域位置信息的地址

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void DrawBox(SMALL\_RECT \*pRc)

{

char chBox[] = {'+', '-', '|'}; /\*画框用的字符\*/

COORD pos = {pRc->Left, pRc->Top}; /\*定位在区域的左上角\*/

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框左上角\*/

for (pos.X = pRc->Left + 1; pos.X < pRc->Right; pos.X++)

{

/\*此循环画上边框横线\*/

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[1], 1, pos, &ul);

}

pos.X = pRc->Right;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框右上角\*/

for (pos.Y = pRc->Top+1; pos.Y < pRc->Bottom; pos.Y++)

{

/\*此循环画边框左边线和右边线\*/

pos.X = pRc->Left;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[2], 1, pos, &ul);

pos.X = pRc->Right;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[2], 1, pos, &ul);

}

pos.X = pRc->Left;

pos.Y = pRc->Bottom;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框左下角\*/

for (pos.X = pRc->Left + 1; pos.X < pRc->Right; pos.X++)

{

/\*画下边框横线\*/

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[1], 1, pos, &ul);

}

pos.X = pRc->Right;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框右下角\*/

return;

}

/\*\*

\* 函数名称: TagSubMenu

\* 函数功能: 在指定子菜单项上做选中标记.

\* 输入参数: num 选中的子菜单项号

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void TagSubMenu(int num)

{

SMALL\_RECT rcPop;

COORD pos;

WORD att;

int width;

LocSubMenu(gi\_sel\_menu, &rcPop); /\*计算弹出子菜单的区域位置, 存放在rcPop中\*/

if ((num<1) || (num == gi\_sel\_sub\_menu) || (num>rcPop.Bottom-rcPop.Top-1))

{

/\*如果子菜单项号越界，或该项子菜单已被选中，则返回\*/

return;

}

pos.X = rcPop.Left + 2;

width = rcPop.Right - rcPop.Left - 3;

if (gi\_sel\_sub\_menu != 0) /\*首先取消原选中子菜单项上的标记\*/

{

pos.Y = rcPop.Top + gi\_sel\_sub\_menu;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

FillConsoleOutputAttribute(gh\_std\_out, att, width, pos, &ul);

pos.X += 1;

att |= FOREGROUND\_RED;/\*白底红字\*/

FillConsoleOutputAttribute(gh\_std\_out, att, 1, pos, &ul);

}

/\*在制定子菜单项上做选中标记\*/

pos.X = rcPop.Left + 2;

pos.Y = rcPop.Top + num;

att = FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_RED; /\*黑底白字\*/

FillConsoleOutputAttribute(gh\_std\_out, att, width, pos, &ul);

gi\_sel\_sub\_menu = num; /\*修改选中子菜单项号\*/

return;

}

/\*\*

\* 函数名称: LocSubMenu

\* 函数功能: 计算弹出子菜单区域左上角和右下角的位置.

\* 输入参数: num 选中的主菜单项号

\* 输出参数: rc 存放区域位置信息的地址

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void LocSubMenu(int num, SMALL\_RECT \*rc)

{

int i, len, loc = 0;

rc->Top = 1; /\*区域的上边定在第2行，行号为1\*/

rc->Left = 1;

for (i=1; i<num; i++)

{

/\*计算区域左边界位置, 同时计算第一个子菜单项在子菜单字符串数组中的位置\*/

rc->Left += strlen(ga\_main\_menu[i-1]) + 4;

loc += ga\_sub\_menu\_count[i-1];

}

rc->Right = strlen(ga\_sub\_menu[loc]);/\*暂时存放第一个子菜单项字符串长度\*/

for (i=1; i<ga\_sub\_menu\_count[num-1]; i++)

{

/\*查找最长子菜单字符串，将其长度存放在rc->Right\*/

len = strlen(ga\_sub\_menu[loc+i]);

if (rc->Right < len)

{

rc->Right = len;

}

}

rc->Right += rc->Left + 3; /\*计算区域的右边界\*/

rc->Bottom = rc->Top + ga\_sub\_menu\_count[num-1] + 1;/\*计算区域下边的行号\*/

if (rc->Right >= SCR\_COL) /\*右边界越界的处理\*/

{

len = rc->Right - SCR\_COL + 1;

rc->Left -= len;

rc->Right = SCR\_COL - 1;

}

return;

}

/\*\*

\* 函数名称: DealInput

\* 函数功能: 在弹出窗口区域设置热区, 等待并相应用户输入.

\* 输入参数: pHotArea

\* piHot 焦点热区编号的存放地址, 即指向焦点热区编号的指针

\* 输出参数: piHot 用鼠标单击、按回车或空格时返回当前热区编号

\* 返 回 值:

\*

\* 调用说明:

\*/

int DealInput(HOT\_AREA \*pHotArea, int \*piHot)

{

INPUT\_RECORD inRec;

CONSOLE\_CURSOR\_INFO lpCur;

DWORD res;

COORD pos = {0, 0};

int iRet = 0;

int cNo, cTag, cSort;/\*cNo:层号, cTag:热区编号, cSort: 热区类型\*/

char vkc; /\*vkc:虚拟键代码\*/

SetHotPoint(pHotArea, \*piHot);

while (TRUE)

{

ReadConsoleInput(gh\_std\_in, &inRec, 1, &res);

if ((inRec.EventType == MOUSE\_EVENT) &&

(inRec.Event.MouseEvent.dwButtonState

== FROM\_LEFT\_1ST\_BUTTON\_PRESSED))

{

pos = inRec.Event.MouseEvent.dwMousePosition;

cNo = gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] & 3;

cTag = (gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] >> 2) & 15;

cSort = (gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] >> 6) & 3;

if ((cNo == gp\_top\_layer->LayerNo) && cTag > 0)

{

/\*如果点在热区上面\*/

\*piHot = cTag;

SetHotPoint(pHotArea, \*piHot);

if (cSort == 0)

{

iRet = cTag;

return iRet;

}

}

else if(cTag == 0)

{

/\* 如果点在非热点区域 \*/

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = FALSE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur); /\* 隐藏光标 \*/

}

}

/\*如果是键盘输入\*/

else if (inRec.EventType == KEY\_EVENT && inRec.Event.KeyEvent.bKeyDown)

{

vkc = inRec.Event.KeyEvent.wVirtualKeyCode;

switch (vkc)

{

case 37 :/\* 处理上、左键 \*/

case 38 :

--\*piHot;

if(\*piHot < 1)

{

\*piHot = pHotArea->num;

}

SetHotPoint(pHotArea, \*piHot);

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

break;

case 13 :/\*回车键或空格表示按下当前按钮\*/

case 32 :

if(pHotArea->pSort[\*piHot-1] == 0)

{

//若为按钮

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = FALSE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur); /\* 隐藏光标 \*/

iRet = \*piHot;

return iRet;

}

case 39 :/\* 处理右、下键 \*/

case 40 :

++\*piHot;

if(\*piHot > pHotArea->num)

{

\*piHot = 1;

}

SetHotPoint(pHotArea, \*piHot);

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

break;

case 27 :/\* 按ESC键 \*/

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = FALSE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur); /\* 隐藏光标 \*/

iRet = 27;

return iRet;

}

}

}

return iRet;

}

void SetHotPoint(HOT\_AREA \*pHotArea, int iHot)

{

CONSOLE\_CURSOR\_INFO lpCur;

COORD pos = {0, 0};

WORD att1, att2;

int i, width;

att1 = FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_RED; /\*黑底白字\*/

att2 = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

for (i=0; i<pHotArea->num; i++)

{

/\*将按钮类热区置为白底黑字\*/

pos.X = pHotArea->pArea[i].Left;

pos.Y = pHotArea->pArea[i].Top;

width = pHotArea->pArea[i].Right - pHotArea->pArea[i].Left + 1;

if (pHotArea->pSort[i] == 0)

{

/\*热区是按钮类\*/

FillConsoleOutputAttribute(gh\_std\_out, att2, width, pos, &ul);

}

}

pos.X = pHotArea->pArea[iHot-1].Left;

pos.Y = pHotArea->pArea[iHot-1].Top;

width = pHotArea->pArea[iHot-1].Right - pHotArea->pArea[iHot-1].Left + 1;

if (pHotArea->pSort[iHot-1] == 0)

{

/\*被激活热区是按钮类\*/

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = FALSE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur); /\*隐藏光标\*/

FillConsoleOutputAttribute(gh\_std\_out, att1, width, pos, &ul);

}

else if (pHotArea->pSort[iHot-1] == 1)

{

/\*被激活热区是文本框类\*/

SetConsoleCursorPosition(gh\_std\_out, pos);

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = TRUE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur);

}

}

/\*\*

\* 函数名称: ExeFunction

\* 函数功能: 执行由主菜单号和子菜单号确定的功能函数.

\* 输入参数: m 主菜单项号

\* s 子菜单项号

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE 或 FALSE

\*

\* 调用说明: 仅在执行函数ExitSys时, 才可能返回FALSE, 其他情况下总是返回TRUE

\*/

BOOL ExeFunction(int m, int s)

{

BOOL bRet = TRUE;

/\*函数指针数组，用来存放所有功能函数的入口地址\*/

BOOL (\*pFunction[ga\_sub\_menu\_count[0]+ga\_sub\_menu\_count[1]+ga\_sub\_menu\_count[2]+ga\_sub\_menu\_count[3]+ga\_sub\_menu\_count[4]])(void);

int i, loc;

/\*将功能函数入口地址存入与功能函数所在主菜单号和子菜单号对应下标的数组元素\*/

pFunction[0] = SaveData;

pFunction[1] = ExitSys;

pFunction[2] = MaintainDeptInfo;

pFunction[3] = MaintainTeamInfo;

pFunction[4] = MaintainProgInfo;

pFunction[5] = QueryDeptInfo;

pFunction[6] = QueryTeamInfo;

pFunction[7] = QueryProgInfo;

pFunction[8] = StatDept;

pFunction[9] = StatTeam;

pFunction[10] = StatProg;

pFunction[11] = HelpTopic;

pFunction[12] = NULL;

pFunction[13] = AboutResearch;

for (i=1,loc=0; i<m; i++) /\*根据主菜单号和子菜单号计算对应下标\*/

{

loc += ga\_sub\_menu\_count[i-1];

}

loc += s - 1;

if (pFunction[loc] != NULL)

{

bRet = (\*pFunction[loc])(); /\*用函数指针调用所指向的功能函数\*/

}

return bRet;

}

/\*\*

\* 函数名称: SaveSysData

\* 函数功能: 保存系统代码表和三类基础数据.

\* 输入参数: hd 主链头结点指针

\* 输出参数:

\* 返 回 值: BOOL类型

\*

\* 调用说明:

\*/

BOOL SaveSysData(DEPT\_NODE \*hd)

{

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

FILE \*pfout;

pfout = fopen(gp\_dept\_info\_filename, "wb");

if(!pfout)

{

printf("院系文件保存失败！\n");

return FALSE;

}

for (pdept\_node = hd; pdept\_node != NULL; pdept\_node = pdept\_node->next)

{

/\*保存院系信息\*/

fwrite(pdept\_node, sizeof(DEPT\_NODE), 1, pfout);

}

fclose(pfout);

pfout = fopen(gp\_team\_info\_filename, "wb");

if(!pfout)

{

printf("科研团队文件保存失败！\n");

return FALSE;

}

for (pdept\_node = hd; pdept\_node != NULL; pdept\_node = pdept\_node->next)

{

/\*保存科研团队信息\*/

pteam\_node = pdept\_node->tnext;

while(pteam\_node != NULL)

{

fwrite(pteam\_node, sizeof(TEAM\_NODE), 1, pfout);

pteam\_node = pteam\_node->next;

}

}

fclose(pfout);

pfout = fopen(gp\_prog\_info\_filename, "wb");

if(!pfout)

{

printf("科研项目文件保存失败！\n");

return FALSE;

}

for (pdept\_node = hd; pdept\_node != NULL; pdept\_node = pdept\_node->next)

{

/\*保存科研项目信息\*/

pteam\_node = pdept\_node->tnext;

while(pteam\_node != NULL)

{

pprog\_node = pteam\_node->pnext;

while(pprog\_node != NULL)

{

fwrite(pprog\_node, sizeof(PROG\_NODE), 1, pfout);

pprog\_node = pprog\_node->next;

}

pteam\_node = pteam\_node->next;

}

}

fclose(pfout);

return TRUE;

}

/\*\*

\* 函数名称: CreatList

\* 函数功能: 从数据文件读取基础数据, 并存放到所创建的十字链表中.

\* 输入参数: 无

\* 输出参数: phead 主链头指针的地址, 用来返回所创建的十字链.

\* 返 回 值: int型数值, 表示链表创建的情况.

\* 0 空链, 无数据

\* 4 已加载院系信息数据，无科研团队信息和科研项目信息数据

\* 12 已加载院系信息和科研团队信息数据，无科研项目信息数据

\* 28 三类基础数据都已加载

\*

\* 调用说明:

\*/

int CreatList(DEPT\_NODE \*\*phead)

{

DEPT\_NODE \*hd = NULL, \*pDeptNode, tmp1;

TEAM\_NODE \*pTeamNode, tmp2;

PROG\_NODE \*pProgNode, tmp3;

FILE \*pFile;

int find;

int re = 0;

if ((pFile = fopen(gp\_dept\_info\_filename, "rb")) == NULL)

{

printf("院系信息数据文件打开失败!\n");

return re;

}

printf("院系信息数据文件打开成功!\n");

/\*每次读取大小为一个DEPT\_NODE所占内存的院系信息结点\*/

while (fread(&tmp1, sizeof(DEPT\_NODE), 1, pFile) == 1)

{

pDeptNode = (DEPT\_NODE \*)malloc(sizeof(DEPT\_NODE));

\*pDeptNode = tmp1;

pDeptNode->tnext = NULL;

pDeptNode->next = hd;

hd = pDeptNode;

}

fclose(pFile);

if (hd == NULL)

{

printf("院系信息数据文件加载失败!\n");

return re;

}

printf("院系信息数据文件加载成功!\n");

\*phead = hd;

re = re + 4;

if ((pFile = fopen(gp\_team\_info\_filename, "rb")) == NULL)

{

printf("科研团队基本信息数据文件打开失败!\n");

return re;

}

printf("科研团队基本信息数据文件打开成功!\n");

re = re + 8;

/\*从数据文件中读取科研团队基本信息数据，存入主链对应结点的科研团队基本信息支链中\*/

while (fread(&tmp2, sizeof(TEAM\_NODE), 1, pFile) == 1)

{

/\*创建结点，存放从数据文件中读出的科研团队基本信息\*/

pTeamNode = (TEAM\_NODE \*)malloc(sizeof(TEAM\_NODE));

\*pTeamNode = tmp2;

pTeamNode->pnext = NULL;

/\*在主链上查找该团队所在院系对应的主链结点\*/

pDeptNode = hd; /\*位移指针pDeptNode定位到链表头部\*/

while (pDeptNode != NULL

&& strcmp(pDeptNode->dept\_name, pTeamNode->dept\_name) != 0)

{

pDeptNode = pDeptNode->next;

}

if (pDeptNode != NULL) /\*如果找到，则将结点以后进先出方式插入团队信息支链\*/

{

pTeamNode->next = pDeptNode->tnext;

pDeptNode->tnext = pTeamNode;

}

else /\*如果未找到，则释放所创建结点的内存空间\*/

{

free(pTeamNode);

}

}

fclose(pFile);

if ((pFile = fopen(gp\_prog\_info\_filename, "rb")) == NULL)

{

printf("科研项目信息数据文件打开失败!\n");

return re;

}

printf("科研项目信息数据文件打开成功!\n");

re = re + 16;

/\*从数据文件中读取科研项目信息数据，存入科研团队基本信息支链对应结点的缴费支链中\*/

while (fread(&tmp3, sizeof(PROG\_NODE), 1, pFile) == 1)

{

/\*创建结点，存放从数据文件中读出的科研项目信息\*/

pProgNode = (PROG\_NODE \*)malloc(sizeof(PROG\_NODE));

\*pProgNode = tmp3;

/\*查找学生信息支链上对应院系信息结点\*/

pDeptNode = hd;

find = 0;

while (pDeptNode != NULL && find == 0)

{

pTeamNode = pDeptNode->tnext;

while (pTeamNode != NULL && find == 0)

{

if (strcmp(pTeamNode->team\_name, pProgNode->team\_name) == 0)

{

find = 1;

break;

}

pTeamNode = pTeamNode->next;

}

pDeptNode = pDeptNode->next;

}

if (find) /\*如果找到，则将结点以后进先出方式插入科研项目信息支链中\*/

{

pProgNode->next = pTeamNode->pnext;

pTeamNode->pnext = pProgNode;

}

else /\*如果未找到，则释放所创建结点的内存空间\*/

{

free(pProgNode);

}

}

fclose(pFile);

return re;

}

/\*\*

\* 函数名称: SeekTeamNode

\* 函数功能: 按团队名称(无重名)查找学生基本信息结点.

\* 输入参数: hd 主链头指针

\* team\_name 团队名称

\* 输出参数: 无

\* 返 回 值: 查中时返回结点的地址, 否则返回NULL

\*

\* 调用说明:

\*/

TEAM\_NODE \*SeekTeamNode(DEPT\_NODE \*hd, char \*team\_name)

{

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node;

int find = 0;

for (pdept\_node=hd; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

pteam\_node = pdept\_node->tnext;

for(; pteam\_node!=NULL; pteam\_node=pteam\_node->next)

{

if (strcmp(team\_name, pteam\_node->team\_name) == 0)

{

find = 1;

break;

}

}

if (find)

{

break;

}

}

if (find)

{

return pteam\_node;

}

else

{

return NULL;

}

}

/\*\*

\* 函数名称: SeekDeptNode

\* 函数功能: 按院系名称(无重名)查找学生基本信息结点.

\* 输入参数: hd 主链头指针

\* dept\_name 院系名称

\* 输出参数: 无

\* 返 回 值: 查中时返回结点的地址, 否则返回NULL

\*

\* 调用说明: 作为InsertDeptNode的子函数调用

\*/

DEPT\_NODE \*SeekDeptNode(DEPT\_NODE \*hd, char \*dept\_name)

{

DEPT\_NODE \*pdept\_node;

int find = 0;

for (pdept\_node=hd; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

if (strcmp(dept\_name, pdept\_node->dept\_name) == 0)

{

find = 1;

break;

}

}

if (find)

{

return pdept\_node;

}

else

{

return NULL;

}

}

/\*\*

\* 函数名称: SeekProgNode

\* 函数功能: 按和缴费日期查找缴费信息结点.

\* 输入参数: hd 主链头指针

\* team\_name 所属团队

\* prog\_num 项目编号

\* 输出参数: 无

\* 返 回 值: 查中时返回结点的地址, 否则返回NULL

\*

\* 调用说明:

\*/

PROG\_NODE \*SeekProgNode(DEPT\_NODE \*hd, char \*team\_name, char \*prog\_num)

{

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

int find = 0;

pteam\_node = SeekTeamNode(hd, team\_name);/\*首先查找满足团队信息的支链\*/

/\*如果找到，则在科研项目支链上继续查找指定的项目结点\*/

if (pteam\_node != NULL)

{

pprog\_node = pteam\_node->pnext;

while (pprog\_node != NULL)

{

if (strcmp(pprog\_node->prog\_num, prog\_num)==0)

{

find = 1;

break;

}

pprog\_node = pprog\_node->next;

}

if (find)

{

return pprog\_node;

}

else

{

return NULL;

}

}

return NULL;

}

/\*\*

\* 函数名称: DelProgNode

\* 函数功能: 从十字链表中删除指定的科研项目信息结点.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示删除成功, FALSE表示删除失败

\*

\* 调用说明: 根据项目编号和所属团队可以确定唯一的缴费信息

\*/

BOOL DelProgNode(DEPT\_NODE \*hd)

{

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node\_prior;

PROG\_NODE \*pprog\_node\_current;

char prog\_num[15];

char team\_name[30];

gotoxy(28, 6);

printf("请输入您要删除的项目信息的项目编号：");

gotoxy(64, 6);

scanf("%s", prog\_num);

getchar();

gotoxy(28, 7);

printf("请输入您要删除的项目信息所属团队：");

gotoxy(62, 7);

scanf("%s", team\_name);

getchar();

pteam\_node = SeekTeamNode(hd, team\_name);

if (pteam\_node != NULL)/\*查找科研项目支链上给定项目编号的科研项目结点\*/

{

pprog\_node\_prior = NULL;

pprog\_node\_current = pteam\_node->pnext;

while (pprog\_node\_current != NULL

&& strcmp(pprog\_node\_current->prog\_num, prog\_num) != 0)

{

pprog\_node\_prior = pprog\_node\_current;

pprog\_node\_current = pprog\_node\_current->next;

}

if (pprog\_node\_current != NULL)

{

if (pprog\_node\_prior == NULL) /\*位于第一个结点的位置\*/

{

pteam\_node->pnext = pprog\_node\_current->next;

}

else

{

pprog\_node\_prior->next = pprog\_node\_current->next;

}

free(pprog\_node\_current);

}

else

{

char \*warn0[] = {"该团队没有项目 ",

"删除基本信息失败 ",

"确认"

};

ClearScreen();

ShowModule(warn0, 3);

ShowMenu();

return FALSE;

}

}

else

{

char \*warn1[] = {"不存在这样的结点 ",

"删除基本信息失败 ",

"确认"

};

ClearScreen();

ShowModule(warn1, 3);

ShowMenu();

return FALSE;

}

char \*sucinfo[] = {"删除成功,信息已保存",

"确认"

};

SaveSysData(gp\_head);

ShowModule(sucinfo, 2);

ClearScreen();

ShowMenu();

return TRUE;

}

/\*\*

\* 函数名称: DelTeamNode

\* 函数功能: 从十字链表中删除指定的科研团队信息结点.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示删除成功, FALSE表示删除失败

\*

\* 调用说明: 根据团队名称和所属院系可以确定唯一的缴费信息

\*/

BOOL DelTeamNode(DEPT\_NODE \*hd)

{

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node\_prior;

TEAM\_NODE \*pteam\_node\_current;

char team\_name[30];

char dept\_name[20];

PROG\_NODE \*pprog\_node\_prior;

PROG\_NODE \*pprog\_node\_current;

char \*notice[] = {"删除成功，信息已保存",

"确认"

};

gotoxy(28, 6);

printf("请输入您要删除的团队信息的名称：");

gotoxy(60, 6);

scanf("%s", team\_name);

getchar();

gotoxy(28, 7);

printf("请输入您要删除的团队信息所属院系：");

gotoxy(62, 7);

scanf("%s", dept\_name);

getchar();

pdept\_node = SeekDeptNode(hd, dept\_name);

if (pdept\_node != NULL)/\*查找科研项目支链上给定项目编号的科研项目结点\*/

{

pteam\_node\_prior = NULL;

pteam\_node\_current = pdept\_node->tnext;

while (pteam\_node\_current != NULL

&& strcmp(pteam\_node\_current->team\_name, team\_name) != 0)

{

pteam\_node\_prior = pteam\_node\_current;

pteam\_node\_current = pteam\_node\_current->next;

}

if (pteam\_node\_current != NULL)

{

if (pteam\_node\_prior == NULL)/\*位于第一个结点的位置\*/

{

pdept\_node->tnext = pteam\_node\_current->next;

}

else

{

pteam\_node\_prior->next = pteam\_node\_current->next;

}

pprog\_node\_prior = NULL;

pprog\_node\_current = pteam\_node\_current->pnext;

MakeDel();/\*使用户再次确认\*/

while (pprog\_node\_current != NULL) /\*删除团队链的子链\*/

{

pprog\_node\_prior = pprog\_node\_current;

pprog\_node\_current = pprog\_node\_current->next;

free(pprog\_node\_prior);

}

free(pteam\_node\_current);

}

else

{

char \*warn1[] = {"不存在这样的结点 ",

"删除基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

ShowMenu();

return FALSE;

}

}

else

{

char \*warn1[] = {"不存在这样的结点 ",

"删除基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

ShowMenu();

return FALSE;

}

SaveSysData(gp\_head);

ShowModule(notice, 2);

ClearScreen();

ShowMenu();

return TRUE;

}

/\*\*

\* 函数名称: DelDeptNode

\* 函数功能: 从十字链表中删除指定的科研团队信息结点.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示删除成功, FALSE表示删除失败

\*

\* 调用说明: 根据院系名称可以确定唯一的院系信息

\*/

BOOL DelDeptNode(DEPT\_NODE \*hd)

{

DEPT\_NODE \*pdept\_node\_current;

DEPT\_NODE \*pdept\_node\_prior;

TEAM\_NODE \*pteam\_node\_prior;

TEAM\_NODE \*pteam\_node\_current;

char dept\_name[20];

PROG\_NODE \*pprog\_node\_prior;

PROG\_NODE \*pprog\_node\_current;

char \*notice[] = {"删除成功，信息已保存",

"确认"

};

gotoxy(28, 6);

printf("请输入您要删除的院系的名称：");

gotoxy(56, 6);

scanf("%s", dept\_name);

getchar();

pdept\_node\_prior = NULL;

pdept\_node\_current = hd;

while (pdept\_node\_current != NULL

&& strcmp(pdept\_node\_current->dept\_name, dept\_name) != 0)

{

pdept\_node\_prior = pdept\_node\_current;

pdept\_node\_current = pdept\_node\_current->next;

}

if (pdept\_node\_current != NULL)/\*查找科研项目支链上给定项目编号的科研项目结点\*/

{

pteam\_node\_prior = NULL;

pteam\_node\_current = pdept\_node\_current->tnext;

if (pdept\_node\_prior == NULL) /\*位于第一个结点的位置\*/

{

gp\_head = pdept\_node\_current -> next;

}

else

{

pdept\_node\_prior->next = pdept\_node\_current->next;

}

MakeDel();/\*使用户再次确认\*/

while (pteam\_node\_current != NULL)

{

pteam\_node\_prior = pteam\_node\_current;

pteam\_node\_current = pteam\_node\_current->next;

pprog\_node\_prior = NULL;

pprog\_node\_current = pteam\_node\_prior->pnext;

while (pprog\_node\_current != NULL)

{

pprog\_node\_prior = pprog\_node\_current;

pprog\_node\_current = pprog\_node\_current->next;

free(pprog\_node\_prior);

}

free(pteam\_node\_prior);

}

free(pdept\_node\_current);

}

else

{

char \*warn1[] = {"不存在这样的院系 ",

"删除基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

ShowMenu();

return FALSE;

}

SaveSysData(gp\_head);

ShowModule(notice, 2);

ClearScreen();

ShowMenu();

return TRUE;

}

/\*\*

\* 函数名称: MakeDel

\* 函数功能: 提醒用户是否需要删除

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示确认删除, FALSE表示取消删除

\*

\* 调用说明:

\*/

void MakeDel()

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

char \*pCh[] = {"是否要删除当前数据？",

" 确定 取消"

};

int iHot = 1;

pos.X = strlen(pCh[0]) + 6; /\*与矩形区域左右边框位置相关\*/

pos.Y = 7; /\*与矩形区域上下边框位置相关\*/

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = 2;

labels.ppLabel = pCh;

COORD aLoc[] = {{rcPop.Left+3, rcPop.Top+2},

{rcPop.Left+5, rcPop.Top+5}

};

labels.pLoc = aLoc;/\*pCh的位置\*/

areas.num = 2;

SMALL\_RECT aArea[] = {{

rcPop.Left + 7, rcPop.Top + 5,

rcPop.Left + 10, rcPop.Top + 5

},

{

rcPop.Left + 14, rcPop.Top + 5,

rcPop.Left + 17, rcPop.Top + 5

}

}; /\*黑色选中框大小及位置\*/

char aSort[] = {0, 0};

char aTag[] = {1, 2};

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 4;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right-rcPop.Left-1, pos, &ul); /\*分割线\*/

if (DealInput(&areas, &iHot) == 1 && iHot == 1)

{

PopOff();

char \*plabel\_name1[] = {"数据已经删除！",

"确认"

};

ShowModule(plabel\_name1, 2);

}

else

{

PopOff();

}

ClearScreen();

ShowMenu();

return;

}

/\*\*

\* 函数名称: ModifProgNode

\* 函数功能: 对指定的项目信息结点内容进行修改.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示修改成功, FALSE表示修改失败

\*

\* 调用说明:

\*/

BOOL ModifProgNode(DEPT\_NODE \*hd)

{

PROG\_NODE \*pprog\_node\_temp;/\*用于作为原来的被修改结点，被pprog\_node替代\*/

PROG\_NODE \*pprog\_node\_next;/\*用于暂时存放指针域\*/

PROG\_NODE \*pprog\_node = (PROG\_NODE \*)malloc(sizeof(PROG\_NODE));

gotoxy(28, 6);

printf("请输入您要修改的项目信息的项目编号：");

gotoxy(32, 8);

scanf("%s", pprog\_node->prog\_num);

getchar();

gotoxy(28, 10);

printf("请输入您要修改的项目信息所属团队：");

gotoxy(32, 12);

scanf("%s", pprog\_node->team\_name);

getchar();

pprog\_node\_temp = SeekProgNode(hd, pprog\_node->team\_name, pprog\_node->prog\_num);/\*查找指定科研项目结点\*/

if (pprog\_node\_temp != NULL)

{

gotoxy(28, 14);

printf("负责人姓名：");

gotoxy(40, 14);

scanf("%s", pprog\_node->pmgr\_name);

getchar();

gotoxy(28, 16);

printf("项目类别：");

gotoxy(38, 16);

scanf("%c", &pprog\_node->type);

getchar();

gotoxy(28, 18);

printf("项目经费: ");

gotoxy(38, 18);

scanf("%f", &pprog\_node->prog\_fee);

getchar();

gotoxy(28, 20);

printf("起始时间：");

gotoxy(38, 20);

scanf("%s", pprog\_node->st\_date);

getchar();

gotoxy(28, 22);

printf("项目编号：");

gotoxy(38, 22);

scanf("%s", pprog\_node->prog\_num);

getchar();

pprog\_node\_next = pprog\_node\_temp->next;/\*保存指针域\*/

\*pprog\_node\_temp = \*pprog\_node;

pprog\_node\_temp->next = pprog\_node\_next;

free(pprog\_node);

char \*sucinfo[] = {"修改成功，信息已保存",

"确认"

};

SaveSysData(gp\_head);

ShowModule(sucinfo, 2);

ClearScreen();

ShowMenu();

return TRUE;

}

else

{

char \*warn1[] = {"此项目不存在 ",

"插入基本信息失败 ",

"确认"

};

ClearScreen();

ShowModule(warn1, 3);

free(pprog\_node);

ShowMenu();

return FALSE;

}

}

/\*\*

\* 函数名称: ModifTeamNode

\* 函数功能: 对指定的团队信息结点内容进行修改.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示修改成功, FALSE表示修改失败

\*

\* 调用说明:

\*/

BOOL ModifTeamNode(DEPT\_NODE \*hd)

{

TEAM\_NODE \*pteam\_node\_temp;/\*用于作为pteam\_node的替代\*/

TEAM\_NODE \*pteam\_node\_next;/\*用于暂时存放指针域\*/

TEAM\_NODE \*pteam\_node = (TEAM\_NODE \*)malloc(sizeof(TEAM\_NODE));

PROG\_NODE \*pteam\_node\_pnext;

gotoxy(29, 6);

printf("请输入您要修改的团队信息的团队名称：");

gotoxy(32, 8);

scanf("%s", pteam\_node->team\_name);

getchar();

pteam\_node\_temp = SeekTeamNode(hd, pteam\_node->team\_name);/\*查找指定科研项目结点\*/

if (pteam\_node\_temp != NULL)

{

gotoxy(29, 10);

printf("团队名称：");

gotoxy(40, 10);

scanf("%s", pteam\_node->team\_name);

getchar();

gotoxy(29, 12);

printf("教师人数：");

gotoxy(40, 12);

scanf("%d", &pteam\_node->team\_prof\_amount);

getchar();

gotoxy(29, 14);

printf("研究生人数: ");

gotoxy(42, 14);

scanf("%d", &pteam\_node->team\_grd\_amount);

getchar();

gotoxy(29, 16);

printf("团队负责人: ");

gotoxy(41, 16);

scanf("%s", pteam\_node->tmgr\_name);

getchar();

gotoxy(29, 18);

printf("所属院系：");

gotoxy(38, 18);

scanf("%s", pteam\_node->dept\_name);

getchar();

pteam\_node\_next = pteam\_node\_temp->next;/\*保存指针域\*/

pteam\_node\_pnext = pteam\_node\_temp->pnext;/\*保存指向项目链的指针域\*/

\*pteam\_node\_temp = \*pteam\_node;

pteam\_node\_temp->next = pteam\_node\_next;

pteam\_node\_temp->pnext = pteam\_node\_pnext;

free(pteam\_node);

char \*sucinfo[] = {"修改成功，信息已保存",

"确认 "

};

SaveSysData(gp\_head);

ShowModule(sucinfo, 2);

ClearScreen();

ShowMenu();

return TRUE;

}

else

{

char \*warn1[] = {"此团队不存在 ",

"插入基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

free(pteam\_node);

ShowMenu();

return FALSE;

}

}

/\*\*

\* 函数名称: ModifDeptNode

\* 函数功能: 对指定的团队信息结点内容进行修改.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示修改成功, FALSE表示修改失败

\*

\* 调用说明:

\*/

BOOL ModifDeptNode(DEPT\_NODE \*hd)

{

DEPT\_NODE \*pdept\_node\_temp;/\*用于作为pteam\_node的替代\*/

DEPT\_NODE \*pdept\_node\_next;/\*用于暂时存放指针域\*/

DEPT\_NODE \*pdept\_node = (DEPT\_NODE \*)malloc(sizeof(DEPT\_NODE));

TEAM\_NODE \*pdept\_node\_tnext;

gotoxy(28, 6);

printf("请输入您要修改的院系的名称：");

gotoxy(56, 6);

scanf("%s", pdept\_node->dept\_name);

getchar();

pdept\_node\_temp = SeekDeptNode(hd, pdept\_node->dept\_name);/\*查找指定院系结点\*/

if (pdept\_node\_temp != NULL)

{

gotoxy(28, 8);

printf("负责人姓名：");

gotoxy(40, 8);

scanf("%s", pdept\_node->dmgr\_name);/\*输入负责人姓名\*/

getchar();

gotoxy(28, 10);

printf("联系电话：");

gotoxy(40, 10);

scanf("%s", pdept\_node->telno);/\*输入联系电话\*/

getchar();

pdept\_node\_next = pdept\_node\_temp->next;/\*保存指针域\*/

pdept\_node\_tnext = pdept\_node\_temp->tnext;/\*保存指向团队链的指针域\*/

\*pdept\_node\_temp = \*pdept\_node;

pdept\_node\_temp->next = pdept\_node\_next;

pdept\_node\_temp->tnext = pdept\_node\_tnext;

char \*sucinfo[] = {"修改成功，信息已保存",

"确认 "

};

SaveSysData(gp\_head);

ShowModule(sucinfo, 2);

ClearScreen();

ShowMenu();

free(pdept\_node);

return TRUE;

}

else

{

char \*warn1[] = {"此院系不存在 ",

"插入基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

free(pdept\_node);

ShowMenu();

return FALSE;

}

}

/\*\*

\* 函数名称: RemoveSysData

\* 函数功能: 从指定数据文件中恢复三类基础数据.

\* 输入参数: phead 主链头结点指针的地址

\* filename 存放备份数据的数据文件名

\* 输出参数:

\* 返 回 值: BOOL类型, 总是为TRUE

\*

\* 调用说明:

\*/

BOOL RemoveSysData(DEPT\_NODE \*\*phead, char \*filename)

{

DEPT\_NODE \*hd = NULL;

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

unsigned long dept\_node\_num = 0;

unsigned long team\_node\_num = 0;

unsigned long prog\_node\_num = 0;

unsigned long ulloop;

int handle;

int find;

if((handle = open(filename,O\_RDONLY | O\_BINARY)) == -1)

{

handle = open(filename,O\_CREAT | O\_BINARY,S\_IREAD);

}

/\*读取三种基本数据信息的记录数\*/

read(handle, (char \*)&dept\_node\_num, sizeof(dept\_node\_num));

read(handle, (char \*)&team\_node\_num, sizeof(team\_node\_num));

read(handle, (char \*)&prog\_node\_num, sizeof(prog\_node\_num));

/\*读取院系信息，建立十字主链\*/

for(ulloop=1; ulloop<=dept\_node\_num; ulloop++)

{

pdept\_node = (DEPT\_NODE \*)malloc(sizeof(DEPT\_NODE));

read(handle, (char \*)pdept\_node, sizeof(DEPT\_NODE));

pdept\_node -> tnext = NULL;

pdept\_node -> next = hd;

hd = pdept\_node;

}

\*phead = hd;

/\*读取科研团队基本信息，建立科研团队支链\*/

for(ulloop=1; ulloop<=team\_node\_num; ulloop++)

{

pteam\_node = (TEAM\_NODE \*)malloc(sizeof(TEAM\_NODE));

read(handle, (char \*)pteam\_node, sizeof(TEAM\_NODE));

pteam\_node -> pnext = NULL;

pdept\_node = hd;

while(pdept\_node != NULL

&& strcmp(pdept\_node -> dept\_name, pteam\_node->dept\_name))

{

pdept\_node = pdept\_node -> next;

}

if(pdept\_node != NULL)

{

pteam\_node -> next = pdept\_node -> tnext;

pdept\_node -> tnext = pteam\_node;

}

else

{

free(pteam\_node);

}

}

/\*读取科研项目信息，建立科研项目信息链表\*/

for(ulloop=1; ulloop<=prog\_node\_num; ulloop++)

{

pprog\_node = (PROG\_NODE \*)malloc(sizeof(PROG\_NODE));

read(handle, (char \*)pprog\_node, sizeof(PROG\_NODE));

pdept\_node = hd;

find = 0;

while(pdept\_node != NULL && find == 0)

{

pteam\_node = pdept\_node -> tnext;

while(pteam\_node != NULL && find == 0)

{

if(strcmp(pteam\_node->team\_name, pprog\_node->team\_name) == 0)

{

find = 1;

break;

}

pteam\_node = pteam\_node -> next;

}

pdept\_node = pdept\_node -> next;

}

if(find)

{

pprog\_node -> next = pteam\_node -> pnext;

pteam\_node -> pnext = pprog\_node;

}

else

{

free(pprog\_node);

}

}

close(handle);

SaveSysData(hd); /\*将内存中数据保存到数据文件\*/

return TRUE;

}

BOOL ShowModule(char \*\*pString, int n)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

BOOL bRet = TRUE;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

int iHot = 1;

int i, maxlen, str\_len;

for (i=0,maxlen=0; i<n; i++)

{

str\_len = strlen(pString[i]);

if (maxlen < str\_len)

{

maxlen = str\_len;

}

}

pos.X = maxlen + 6;

pos.Y = n + 5;

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = n;

labels.ppLabel = pString;

COORD aLoc[n];

for (i=0; i<n; i++)

{

aLoc[i].X = rcPop.Left + 2;

aLoc[i].Y = rcPop.Top + 2 + i;

}

str\_len = strlen(pString[n-1]);

aLoc[n-1].X = rcPop.Left + 3 + (maxlen-str\_len)/2;

aLoc[n-1].Y = aLoc[n-1].Y + 2;

labels.pLoc = aLoc;

areas.num = 1;

SMALL\_RECT aArea[] = {{

aLoc[n-1].X, aLoc[n-1].Y,

aLoc[n-1].X + 3, aLoc[n-1].Y

}

};

char aSort[] = {0};

char aTag[] = {1};

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

DrawBox(&rcPop);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 2 + n;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right-rcPop.Left-1, pos, &ul);

DealInput(&areas, &iHot);

PopOff();

return bRet;

}

/\*\*

\* 函数名称: SaveData

\* 函数功能:

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: BOOL类型, 总是为TRUE

\*

\* 调用说明:

\*/

BOOL SaveData(void)

{

BOOL bRet = TRUE;

char \*plabel\_name[] = {"按确认键保存数据",

"确认"

};

ShowModule(plabel\_name, 2);

SaveSysData(gp\_head);

char \*plabel\_name1[] = {"数据已保存",

"返回"

};

ShowModule(plabel\_name1, 2);

return bRet;

}

/\*\*

\* 函数名称: ExitSys

\* 函数功能: 清屏后退出系统

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示退出成功, FALSE表示退出失败

\*

\* 调用说明:

\*/

BOOL ExitSys(void)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

BOOL bRet = TRUE;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

char \*pCh[] = {"是否需要退出系统？", "确定 取消"};

int iHot = 1;

pos.X = strlen(pCh[0]) + 6;

pos.Y = 7;

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = 2;

labels.ppLabel = pCh;

COORD aLoc[] = {{rcPop.Left+3, rcPop.Top+2},

{rcPop.Left+5, rcPop.Top+5}

};

labels.pLoc = aLoc;

areas.num = 2;

SMALL\_RECT aArea[] = {{

rcPop.Left + 5, rcPop.Top + 5,

rcPop.Left + 8, rcPop.Top + 5

},

{

rcPop.Left + 13, rcPop.Top + 5,

rcPop.Left + 16, rcPop.Top + 5

}

};

char aSort[] = {0, 0};

char aTag[] = {1, 2};

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 4;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right-rcPop.Left-1, pos, &ul);

if (DealInput(&areas, &iHot) == 1 && iHot == 1)

{

bRet = FALSE;

PopOff();

}

else

{

PopOff();

bRet = TRUE;

}

return bRet;

}

BOOL MaintainDeptInfo(void)

{

char \*pString[] = {"插入院系基本信息",

"修改院系基本信息",

"删除院系基本信息",

"返回"

};

int hotnum = 0;

hotnum = MyMenu4(pString, 4);/\*选择要进行的操作\*/

switch (hotnum)

{

case 1:

InsertDeptNode(gp\_head);

break;

case 2:

ModifDeptNode(gp\_head);

break;

case 3:

DelDeptNode(gp\_head);

break;

case 4:

PopOff();

break;

}

return TRUE;

}

BOOL MaintainTeamInfo(void)

{

char \*pString[] = {"插入团队基本信息",

"修改团队基本信息",

"删除团队基本信息",

"返回"

};

int hotnum = 0;

hotnum = MyMenu4(pString, 4);/\*选择要进行的操作\*/

switch (hotnum)

{

case 1:

InsertTeamNode(gp\_head);

break;

case 2:

ModifTeamNode(gp\_head);

break;

case 3:

DelTeamNode(gp\_head);

break;

case 4:

PopOff();

break;

}

return TRUE;

}

BOOL MaintainProgInfo(void)

{

char \*pString[] = {"插入项目基本信息",

"修改项目基本信息",

"删除项目基本信息",

"返回"

};

int hotnum = 0;

hotnum = MyMenu4(pString, 4);/\*选择要进行的操作\*/

switch (hotnum)

{

case 1:

InsertProgNode(gp\_head);

break;

case 2:

ModifProgNode(gp\_head);

break;

case 3:

DelProgNode(gp\_head);

break;

case 4:

PopOff();

break;

}

return TRUE;

}

/\*\*

\* 函数名称: MyMenu4

\* 函数功能: 将含有3个字符串与1个"返回"字符串的标签束设置成热区并等待用户输入

\* 输入参数: pString 指针，n标签个数

\* 输出参数: 无

\* 返 回 值: 用鼠标单击、按回车或空格时返回当前热区编号

\*

\* 调用说明:

\*/

int MyMenu4(char \*\*pString, int n)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

int iHot = 1;

int i, maxlen, str\_len, x;

/\*找到最长的字符串以便于确定pop的大小\*/

for (i=0, maxlen=0; i<n; i++)

{

str\_len = strlen(pString[i]);

if (maxlen < str\_len)

{

maxlen = str\_len;

}

}

/\*确定pop位置和大小\*/

pos.X = maxlen + 6;

pos.Y = n + 5;

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = n;

labels.ppLabel = pString;

COORD aLoc[n];

for (i=0; i<n; i++)

{

aLoc[i].X = rcPop.Left + 3;

aLoc[i].Y = rcPop.Top + 2 + i;

}

/\*确定 取消的位置\*/

str\_len = strlen(pString[n-1]);

aLoc[n-1].X = rcPop.Left + 3 + (maxlen-str\_len)/2;

aLoc[n-1].Y = aLoc[n-1].Y + 2;

labels.pLoc = aLoc;

areas.num = 4;

SMALL\_RECT aArea[] = {{

aLoc[0].X, aLoc[0].Y,

aLoc[0].X + maxlen - 1, aLoc[0].Y

},

{

aLoc[1].X, aLoc[1].Y,

aLoc[1].X + maxlen - 1, aLoc[1].Y

},

{

aLoc[2].X, aLoc[2].Y,

aLoc[2].X + maxlen - 1, aLoc[2].Y

},

{

aLoc[3].X, aLoc[3].Y,

aLoc[3].X + 3, aLoc[3].Y

}

};

char aSort[] = {0, 0, 0, 0};

char aTag[] = {1, 2, 3, 4};

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 2 + n;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right-rcPop.Left-1, pos, &num\_written);

x = DealInput(&areas, &iHot);

PopOff();

return x;

}

/\*\*

\* 函数名称: MyMenu3

\* 函数功能: 将含有2个字符串与1个"返回"字符串的标签束设置成热区并等待用户输入

\* 输入参数: pString 指针，n标签个数

\* 输出参数: 无

\* 返 回 值: 用鼠标单击、按回车或空格时返回当前热区编号

\*

\* 调用说明:

\*/

int MyMenu3(char \*\*pString, int n)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

int iHot = 1;

int i, maxlen, str\_len,x;

for (i=0, maxlen=0; i<n; i++)

{

str\_len = strlen(pString[i]);

if (maxlen < str\_len)

{

maxlen = str\_len;

}

}

pos.X = maxlen + 6;

pos.Y = n + 5;

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = n;

labels.ppLabel = pString;

COORD aLoc[n];

for (i=0; i<n; i++)

{

aLoc[i].X = rcPop.Left + 3;

aLoc[i].Y = rcPop.Top + 2 + i;

}

str\_len = strlen(pString[n-1]);

aLoc[n-1].X = rcPop.Left + 3 + (maxlen-str\_len)/2;

aLoc[n-1].Y = aLoc[n-1].Y + 2;

labels.pLoc = aLoc;

areas.num = 3;

SMALL\_RECT aArea[] = {{

aLoc[0].X, aLoc[0].Y,

aLoc[0].X + maxlen - 1, aLoc[0].Y

},

{

aLoc[1].X, aLoc[1].Y,

aLoc[1].X + maxlen - 1, aLoc[1].Y

},

{

aLoc[2].X, aLoc[2].Y,

aLoc[2].X + 3, aLoc[2].Y

}

};

char aSort[] = {0, 0, 0};

char aTag[] = {1, 2, 3};

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 2 + n;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right-rcPop.Left-1, pos, &num\_written);

x = DealInput(&areas, &iHot);

PopOff();

return x;

}

/\*\*

\* 函数名称: gotoxy

\* 函数功能: 使光标定位到屏幕的某个点

\* 输入参数: x为横坐标，y为纵坐标

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void gotoxy(int x, int y)

{

COORD pos;

pos.X = x - 1;

pos.Y = y - 1;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),pos);

}

/\*\*

\* 函数名称: InsertDeptNode

\* 函数功能: 在三向十字链表中插入一个院系信息结点.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void InsertDeptNode(DEPT\_NODE\* hd)

{

DEPT\_NODE \*pdept\_node;

pdept\_node = (DEPT\_NODE\*)malloc(sizeof(DEPT\_NODE));

gotoxy(25, 4);

printf("请输入您所要添加的院系的基本信息：");

gotoxy(29, 6);

printf("院系名称：");

gotoxy(39, 6);

scanf("%s", pdept\_node -> dept\_name);/\*输入院系名称\*/

getchar();

if (strlen(pdept\_node -> dept\_name) > 20)

{

char \*warn1[] = {"输入院系名称过长",

"插入院系信息失败",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

ShowMenu();

free(pdept\_node);

return;

}

/\*如果院系已经存在，则做出提示\*/

if (SeekDeptNode(hd, pdept\_node->dept\_name)!=NULL)

{

char \*warn2[] = {"该院系已存在",

"如欲修改院系信息，请选择修改院系信息",

"确认"

};

ShowModule(warn2, 3);

ClearScreen();

ShowMenu();

free(pdept\_node);

return;

}

gotoxy(29, 8);

printf("负责人姓名：");

gotoxy(41, 8);

scanf("%s", pdept\_node->dmgr\_name);/\*输入负责人姓名\*/

getchar();

gotoxy(29, 10);

printf("联系电话：");

gotoxy(39, 10);

scanf("%s", pdept\_node->telno);/\*输入联系电话\*/

getchar();

pdept\_node->tnext = NULL; /\*将子链头结点置空\*/

pdept\_node->next = hd;

gp\_head = pdept\_node; /\*新建结点作为新的主链的头结点\*/

char \*sucinfo[] = {"插入成功，信息已保存",

"确认"

};

SaveSysData(gp\_head);

ShowModule(sucinfo, 2);

ClearScreen();

ShowMenu();

return;

}

/\*\*

\* 函数名称: InsertTeamNode

\* 函数功能: 在院系信息主链十字链表中插入团队基本信息结点.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void InsertTeamNode(DEPT\_NODE\* hd)

{

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node;

pteam\_node = (TEAM\_NODE\*)malloc(sizeof(TEAM\_NODE));

gotoxy(26, 4);

printf("请输入您所要添加的团队基本信息：");

gotoxy(30, 6);

printf("所属院系: ");

gotoxy(40, 6);

scanf("%s", pteam\_node -> dept\_name);

getchar();

/\*判断院系名称输入是否符合规范\*/

if(strlen(pteam\_node->dept\_name)>20)

{

char \*warn2[] = {"院系名称过长 ",

"插入基本信息失败 ",

"确认"

};

ShowModule(warn2, 3);

free(pteam\_node);

ClearScreen();

ShowMenu();

return;

}

/\*在十字链中找到对应的院系信息结点\*/

pdept\_node = SeekDeptNode(hd, pteam\_node->dept\_name);

if(pdept\_node == NULL)

{

char \*warn1[] = {"此院系不存在 ",

"插入基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

free(pteam\_node);

ShowMenu();

return;

}

else

{

gotoxy(30, 8);

printf("团队名称：");

gotoxy(42, 8);

scanf("%s", pteam\_node->team\_name);

getchar();

gotoxy(30, 10);

printf("教师人数：");

gotoxy(40, 10);

scanf("%d", &pteam\_node->team\_prof\_amount);

getchar();

gotoxy(30, 12);

printf("研究生人数: ");

gotoxy(42, 12);

scanf("%d", &pteam\_node->team\_grd\_amount);

getchar();

gotoxy(30, 14);

printf("团队负责人: ");

gotoxy(44, 14);

scanf("%s", pteam\_node->tmgr\_name);

getchar();

pteam\_node->pnext = NULL;

pteam\_node->next = pdept\_node -> tnext;

pdept\_node->tnext = pteam\_node;

gotoxy(30, 16);

printf("按任意键继续……\n");

getch();

char \*sucinfo[] = {"插入成功，信息已保存",

"确认"

};

SaveSysData(gp\_head);

ClearScreen();

ShowMenu();

ShowModule(sucinfo, 2);

return;

}

}

/\*\*

\* 函数名称: InsertProgNode

\* 函数功能: 在团队信息链十字链表中插入项目基本信息结点.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void InsertProgNode(DEPT\_NODE\* hd)

{

PROG\_NODE \*pprog\_node;

TEAM\_NODE \*pteam\_node;

pprog\_node = (PROG\_NODE\*)malloc(sizeof(PROG\_NODE));

gotoxy(26, 4);

printf("请输入您所要添加的项目基本信息：");

gotoxy(30, 6);

printf("所属团队: ");

gotoxy(40, 6);

scanf("%s", pprog\_node->team\_name);

getchar();

/\*判断团队名称输入是否符合规范\*/

if(strlen(pprog\_node->team\_name) > 30)

{

char \*warn2[] = {"科研团队名称过长 ",

"插入基本信息失败 ",

"确认"

};

ShowModule(warn2, 3);

free(pprog\_node);

ClearScreen();

ShowMenu();

return;

}

/\*在十字链中找到对应的团队信息结点\*/

pteam\_node = SeekTeamNode(hd, pprog\_node->team\_name);

if(pteam\_node == NULL)

{

char \*warn1[] = {"此团队不存在 ",

"插入基本信息失败 ",

"确认"

};

ShowModule(warn1, 3);

ClearScreen();

free(pprog\_node);

ShowMenu();

return;

}

else

{

gotoxy(30, 8);

printf("负责人姓名：");

gotoxy(42, 8);

scanf("%s", pprog\_node->pmgr\_name);

getchar();

gotoxy(30, 10);

printf("项目编号：");

gotoxy(40, 10);

scanf("%s", pprog\_node->prog\_num);

getchar();

gotoxy(30, 12);

printf("项目经费: ");

gotoxy(40, 12);

scanf("%f", &pprog\_node->prog\_fee);

getchar();

gotoxy(30, 14);

printf("项目类别: ");

gotoxy(40, 14);

scanf("%c", &pprog\_node->type);

getchar();

/\*判断项目类别名称输入是否符合规范\*/

if(pprog\_node->type!='1' && pprog\_node->type!='2' && pprog\_node->type!='3' && pprog\_node->type!='4' && pprog\_node->type!='5')

{

char \*warn3[] = {"没有此项目类别 插入基本信息失败 ",

"确认"

};

ShowModule(warn3, 2);

free(pprog\_node);

ClearScreen();

ShowMenu();

return;

}

gotoxy(30, 16);

printf("起始时间: ");

gotoxy(40, 16);

scanf("%s", pprog\_node->st\_date);

getchar();

pprog\_node->next = pteam\_node->pnext;

pteam\_node->pnext = pprog\_node;

gotoxy(30, 18);

printf("按任意键继续……\n");

getch();

char \*sucinfo[] = {"插入成功，信息已保存",

"确认"

};

SaveSysData(gp\_head);

ClearScreen();

ShowMenu();

ShowModule(sucinfo, 2);

return;

}

}

BOOL QueryDeptInfo(void)

{

char \*pString[] = {"按院负责人查询",

"按院系名称关键字查询",

"返回"

};

int hotnum = 0;

hotnum = MyMenu3(pString,3);

switch (hotnum)

{

case 1:

SeekDmgr(); /\*按院负责人查询\*/

break;

case 2:

SeekDeptPart(); /\*以院系名称的部分或全部为条件查找\*/

break;

case 3:

break; /\*返回\*/

}

return TRUE;

}

BOOL QueryTeamInfo(void)

{

char \*pString[] = {"按团队关键字查询",

"按教师人数查询",

"返回"

};

int hotnum = 0;

hotnum = MyMenu3(pString,3);

switch (hotnum)

{

case 1:

SeekTeamPart(); /\*按团队关键字查询团队\*/

break;

case 2:

SeekTeamProf(); /\*根据教师人数查询满足条件的团队信息\*/

break;

case 3:

break; /\*返回\*/

}

return TRUE;

}

BOOL QueryProgInfo(void)

{

char \*pString[] = {"按项目编号查询",

"按所属团队查询",

"返回"

};

int hotnum = 0;

hotnum = MyMenu3(pString,3);

switch (hotnum)

{

case 1:

SeekProgNum();/\*按项目编号查询\*/

break;

case 2:

SeekProgTeam();/\*按所属团队查询\*/

break;

case 3:

break;/\*返回\*/

}

return TRUE;

}

/\*\*

\* 函数名称: SeekDmgr

\* 函数功能: 根据输入的院负责人姓名（无重名）查找院系信息

\* 输入参数: head 主链头指针

\* 输出参数: 满足条件的院系信息

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SeekDmgr()

{

char dmgr\_name[12];

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*find = head;

int flag = 0;

gotoxy(27, 2);

printf("请输入院负责人姓名：");

gotoxy(47, 2);

scanf("%s", dmgr\_name);

getchar();

printf("\n\t 院系名称\t\t院负责人\t联系电话");

for(find=head; find!=NULL; find=find->next)

{

if(strcmp(find->dmgr\_name, dmgr\_name) == 0)

{

printf("\n");

flag = 1;

printf("\t %-20s\t%-12s\t%-15s", find->dept\_name, find->dmgr\_name, find->telno);

}

}

if(flag == 0)

{

printf("\n\t\t\t不存在满足条件的院系 ");

}

printf("\n\t\t\t\t按任意键继续……");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekDeptPart

\* 函数功能: 根据输入的院系部分或全部名称查找院系信息

\* 输入参数: 无

\* 输出参数: 满足条件的院系信息

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SeekDeptPart()

{

char dept\_name[20];

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*find = head;

int flag = 0; /\*标记是否找到至少一个符合条件的院系\*/

gotoxy(27, 2);

printf("输入院系名称关键字：");

gotoxy(47, 2);

scanf("%s", dept\_name);

getchar();

printf("\n\t 院系名称\t\t院负责人\t联系电话");

for(find=head; find!=NULL; find=find->next)

{

if(strstr(find->dept\_name, dept\_name))

{

printf("\n");

flag = 1;

printf("\t %-20s\t%-12s\t%-15s", find->dept\_name, find->dmgr\_name, find->telno);

}

}

if(!flag)

{

printf("\n\t\t\t不存在满足条件的院系……");

}

printf("\n\t\t\t按任意键继续…… ");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekTeamPart

\* 函数功能: 根据输团队部分或全部名称查找院系信息

\* 输入参数: 无

\* 输出参数: 满足条件的团队信息

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SeekTeamPart()

{

char team\_name[20];

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*dept\_sech = head;

int flag = 0; /\*标记是否找到至少一个符合条件的团队\*/

gotoxy(27, 2);

printf("输入团队名称关键字：");

gotoxy(47, 2);

scanf("%s", team\_name);

getchar();

printf("\n\t团队名称\t\t队负责人\t所属院系\t教师人数\t研究生人数");

for(dept\_sech=head; dept\_sech!=NULL; dept\_sech=dept\_sech->next)

{

TEAM\_NODE \*team\_sech = dept\_sech->tnext;

while (team\_sech != NULL)

{

if(strstr(team\_sech->team\_name, team\_name))

{

printf("\n");

flag = 1;

printf("\t%-20s\t%-12s\t%-16s%-16d%-8d",team\_sech->team\_name, team\_sech->tmgr\_name,team\_sech->dept\_name,team\_sech->team\_prof\_amount,team\_sech->team\_grd\_amount);

}

team\_sech = team\_sech->next;

}

}

if(!flag)

{

printf("\n\t\t\t不存在满足条件的团队");

}

printf("\n\t\t\t\t按任意键继续…… ");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekProgNum

\* 函数功能: 根据所输入的项目编号查找项目信息

\* 输入参数: 无

\* 输出参数: 满足条件的团队信息

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SeekProgNum()

{

char prog\_num[15];

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*dept\_sech = head;

int flag = 0; /\*标记是否找到至少一个符合条件的团队\*/

char \*prog\_type;

gotoxy(27, 2);

printf("输入团队项目编号：");

gotoxy(45, 2);

scanf("%s", prog\_num);

getchar();

printf("\n\t所属团队\t起始时间\t项目编号\t项目经费\t项目负责人\t项目类别\t\t");

for(dept\_sech=head; dept\_sech!=NULL; dept\_sech=dept\_sech->next)

{

TEAM\_NODE \*team\_sech = dept\_sech->tnext;

while (team\_sech != NULL)

{

PROG\_NODE \*prog\_sech = team\_sech->pnext;

while (prog\_sech != NULL)

{

switch(prog\_sech->type)

{

case '1':

prog\_type = "973计划";

break;

case '2':

prog\_type = "国家自然科学基金项目";

break;

case '3':

prog\_type = "863计划";

break;

case '4':

prog\_type = "国际合作项目";

break;

case '5':

prog\_type = "横向项目";

break;

}

if(!strcmp(prog\_sech->prog\_num, prog\_num))

{

printf("\n");

flag = 1;

printf("\t%-12s\t%-8s\t%-16s%-16.2f%-16s%-21s", prog\_sech->team\_name, prog\_sech->st\_date, prog\_sech->prog\_num, prog\_sech->prog\_fee, prog\_sech->pmgr\_name, prog\_type);

}

prog\_sech = prog\_sech->next;

}

team\_sech = team\_sech->next;

}

}

if(!flag)

{

printf("\n\t\t\t\t不存在满足条件的项目");

}

printf("\n\t\t\t\t按任意键继续…… ");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekProgTeam

\* 函数功能: 根据所属团队查找项目信息

\* 输入参数: 无

\* 输出参数: 满足条件的项目信息

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SeekProgTeam()

{

char team\_name[15];

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*dept\_sech = head;

int flag = 0; /\*标记是否找到至少一个符合条件的项目\*/

char \*prog\_type;

gotoxy(27, 2);

printf("输入项目所属团队：");

gotoxy(45, 2);

scanf("%s", team\_name);

getchar();

printf("\n\t所属团队\t起始时间\t项目编号\t项目经费\t项目负责人\t项目类别\t\t");

for(dept\_sech=head; dept\_sech!=NULL; dept\_sech=dept\_sech->next)

{

TEAM\_NODE \*team\_sech = dept\_sech->tnext;

while (team\_sech != NULL)

{

PROG\_NODE \*prog\_sech = team\_sech->pnext;

while (prog\_sech != NULL)

{

switch(prog\_sech->type)

{

case '1':

prog\_type = "973计划";

break;

case '2':

prog\_type = "国家自然科学基金项目";

break;

case '3':

prog\_type = "863计划";

break;

case '4':

prog\_type = "国际合作项目";

break;

case '5':

prog\_type = "横向项目";

break;

}

if(!strcmp(prog\_sech->team\_name, team\_name))

{

printf("\n");

flag = 1;

printf("\t%-12s\t%-8s\t%-16s%-16.2f%-16s%-21s", prog\_sech->team\_name, prog\_sech->st\_date, prog\_sech->prog\_num, prog\_sech->prog\_fee, prog\_sech->pmgr\_name, prog\_type);

}

prog\_sech = prog\_sech->next;

}

team\_sech = team\_sech->next;

}

}

if(!flag)

{

printf("\n\t\t\t\t……不存在满足条件的项目");

}

printf("\n\t\t\t按任意键继续…… ");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekTeamProf

\* 函数功能: 根据教师人数查找团队信息

\* 输入参数: 无

\* 输出参数: 满足条件的团队信息

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SeekTeamProf(void)

{

char form\_sign;

int form\_num;

char form[10];

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*dept\_sech = head;

int flag = 0; /\*标记是否找到至少一个符合条件的团队\*/

gotoxy(27, 2);

printf("请输入教师人数查询公式，格式为‘>’,'<','='加上非负整数：");

gotoxy(31, 3);

scanf("%c%s", &form\_sign, form);

getchar();

printf("\n\t团队名称\t\t队负责人\t所属院系\t教师人数\t研究生人数");

form\_num = atoi(form);

if (form\_sign!='>'&&form\_sign!='<'&&form\_sign!='=')

{

printf("\n");

printf("\t\t\t\t输入符号有误\n");

printf("\n\t\t\t\t按任意键继续…… ");

getch();

ClearScreen();

ShowMenu();

return;

}

if (form\_num<0)

{

printf("\n");

printf("\t\t\t\t输入数字有误，请输入非负整数\n");

getch();

ClearScreen();

ShowMenu();

return;

}

for(dept\_sech=head; dept\_sech!=NULL; dept\_sech=dept\_sech->next)

{

TEAM\_NODE \*team\_sech = dept\_sech->tnext;

while (team\_sech != NULL)

{

switch (form\_sign)

{

case '>':

if (team\_sech->team\_prof\_amount>form\_num)

{

printf("\n");

flag = 1;

printf("\t%-20s\t%-12s\t%-16s%-16d%-8d", team\_sech->team\_name, team\_sech->tmgr\_name, team\_sech->dept\_name, team\_sech->team\_prof\_amount, team\_sech->team\_grd\_amount);

}

break;

case '<':

if (team\_sech->team\_prof\_amount<form\_num)

{

printf("\n");

flag = 1;

printf("\t%-20s\t%-12s\t%-16s%-16d%-8d", team\_sech->team\_name, team\_sech->tmgr\_name, team\_sech->dept\_name, team\_sech->team\_prof\_amount, team\_sech->team\_grd\_amount);

}

break;

case '=':

if (team\_sech->team\_prof\_amount==form\_num)

{

printf("\n");

flag = 1;

printf("\t%-20s\t%-12s\t%-16s%-16d%-8d", team\_sech->team\_name, team\_sech->tmgr\_name, team\_sech->dept\_name, team\_sech->team\_prof\_amount, team\_sech->team\_grd\_amount);

}

break;

}

team\_sech = team\_sech->next;

}

}

if(!flag)

{

printf("\n\t\t\t不存在满足条件的团队……");

}

printf("\n\t\t\t\t按任意键继续…… ");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: StatRateInfo

\* 函数功能: 按类别统计所有院系的师生人数

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 按要求输出统计结果

\*

\* 调用说明:

\*/

void StatRateInfo(void)

{

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node;

RATE\_NODE \*prate\_node\_temp;

RATE\_NODE \*prate\_node\_ret = NULL;

for(pdept\_node=head; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

int dept\_prof\_amount = 0, dept\_grd\_amount = 0;

prate\_node\_temp = (RATE\_NODE \*)malloc(sizeof(RATE\_NODE));

pteam\_node = pdept\_node->tnext;

while (pteam\_node != NULL)

{

dept\_grd\_amount = dept\_grd\_amount + pteam\_node->team\_grd\_amount;

dept\_prof\_amount = dept\_prof\_amount + pteam\_node->team\_prof\_amount;

pteam\_node = pteam\_node->next;

}

strcpy(prate\_node\_temp->dept\_name, pdept\_node->dept\_name);

prate\_node\_temp->dept\_grd\_amount = dept\_grd\_amount;

prate\_node\_temp->dept\_prof\_amount = dept\_prof\_amount;

prate\_node\_temp->proportion = dept\_grd\_amount/1.0/dept\_prof\_amount;

prate\_node\_temp->next = prate\_node\_ret;

prate\_node\_ret = prate\_node\_temp;

}

SortRateInfo(prate\_node\_ret);

gotoxy(28, 3);

printf("下面按学生教师人数比输出院系信息：");

printf("\n\t 院系名称\t研究生人数\t教师人数\t学生教师人数比");

prate\_node\_temp = prate\_node\_ret;

while(prate\_node\_temp!=NULL)

{

if(prate\_node\_temp->dept\_prof\_amount==0)

{

printf("\n");

printf("\t %-20s\t%-8d\t%-8d\t教师人数不可为0", prate\_node\_temp->dept\_name,prate\_node\_temp->dept\_grd\_amount,prate\_node\_temp->dept\_prof\_amount);

prate\_node\_temp = prate\_node\_temp->next;

continue;

}

printf("\n");

printf("\t %-20s\t%-8d\t%-8d\t%-8.2f", prate\_node\_temp->dept\_name, prate\_node\_temp->dept\_grd\_amount, prate\_node\_temp->dept\_prof\_amount, prate\_node\_temp->proportion);

prate\_node\_temp = prate\_node\_temp->next;

}

printf("\n\t\t\t 按任意键继续……\n");

getch();

ClearScreen();

ShowMenu();

return;

}

/\*\*

\* 函数名称: StatRichProg

\* 函数功能: 统计科研经费最高的项目，输出科研经费最高的10个项目

\* 输入参数: 无

\* 输出参数: 按要求输出统计结果

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void StatRichProg(void)

{

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*pdept\_node;

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

PROG\_FEE\_NODE \*pprog\_fee\_node\_temp;

PROG\_FEE\_NODE \*pprog\_fee\_node\_ret = NULL;

int count = 0;

for(pdept\_node=head; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

pteam\_node = pdept\_node->tnext;

while (pteam\_node != NULL)

{

pprog\_node = pteam\_node->pnext;

while (pprog\_node != NULL)

{

pprog\_fee\_node\_temp=(PROG\_FEE\_NODE\*)malloc(sizeof(PROG\_FEE\_NODE));

strcpy(pprog\_fee\_node\_temp->pmgr\_name, pprog\_node->pmgr\_name);

strcpy(pprog\_fee\_node\_temp->prog\_num, pprog\_node->prog\_num);

strcpy(pprog\_fee\_node\_temp->team\_name, pprog\_node->team\_name);

pprog\_fee\_node\_temp->prog\_fee = pprog\_node->prog\_fee;

pprog\_node = pprog\_node->next;

pprog\_fee\_node\_temp->next = pprog\_fee\_node\_ret;

pprog\_fee\_node\_ret = pprog\_fee\_node\_temp;

}

pteam\_node = pteam\_node->next;

}

}

SortRichProg(pprog\_fee\_node\_ret);

gotoxy(28, 3);

printf("下面按科研经费充足程度输出20条项目信息：\n");

printf("\t所属团队\t项目经费\t项目负责人姓名\t项目编号");

pprog\_fee\_node\_temp = pprog\_fee\_node\_ret;

while(pprog\_fee\_node\_temp!=NULL&&count<10)

{

count++;

printf("\n");

printf("\t%-12s\t%-8.2f\t%-14s\t%-15s",pprog\_fee\_node\_temp->team\_name,pprog\_fee\_node\_temp->prog\_fee,pprog\_fee\_node\_temp->pmgr\_name,pprog\_fee\_node\_temp->prog\_num);

pprog\_fee\_node\_temp = pprog\_fee\_node\_temp->next;

}

printf("\n\t\t\t 按任意键继续……\n");

getch();

ClearScreen();

ShowMenu();

return;

}

/\*\*

\* 函数名称: SortRichProg

\* 函数功能: 对科研经费充足项目信息链进行排序

\* 输入参数: pf\_hd 科研经费充足项目信息链头指针

\* 输出参数: pf\_hd 排序结果通过头指针返回

\* 返 回 值: 无

\*

\* 调用说明:仅在StatRichProg母函数中作为子函数调用

\*/

void SortRichProg(PROG\_FEE\_NODE \*pf\_hd)

{

PROG\_FEE\_NODE \*ppf\_node\_prior;

PROG\_FEE\_NODE \*ppf\_node\_after;

PROG\_FEE\_NODE \*ppf\_node\_cur;

PROG\_FEE\_NODE \*ppf\_node\_temp;

ppf\_node\_prior = pf\_hd;

if (ppf\_node\_prior == NULL)

{

return;

}

/\*排序时，用于交换结点信息的临时结点\*/

ppf\_node\_temp = (PROG\_FEE\_NODE \*) malloc(sizeof(PROG\_FEE\_NODE));

while (ppf\_node\_prior->next != NULL) /\*选择法排序\*/

{

ppf\_node\_cur = ppf\_node\_prior;

ppf\_node\_after = ppf\_node\_prior->next;

while (ppf\_node\_after != NULL)

{

if (ppf\_node\_cur->prog\_fee < ppf\_node\_after->prog\_fee)

{

ppf\_node\_cur = ppf\_node\_after;

}

ppf\_node\_after = ppf\_node\_after->next;

}

/\*交换数据域，指针域不交换\*/

if (ppf\_node\_cur != ppf\_node\_prior)

{

\*ppf\_node\_temp = \*ppf\_node\_prior;

\*ppf\_node\_prior = \*ppf\_node\_cur;

ppf\_node\_prior->next = ppf\_node\_temp->next;

ppf\_node\_temp->next = ppf\_node\_cur->next;

\*ppf\_node\_cur = \*ppf\_node\_temp;

}

ppf\_node\_prior = ppf\_node\_prior->next;

}

free(ppf\_node\_temp);

return;

}

/\*\*

\* 函数名称: SortRateInfo

\* 函数功能: 对院系师生信息链进行排序

\* 输入参数: rate\_hd 院系师生信息链头指针

\* 输出参数: rate\_hd 排序结果通过头指针返回

\* 返 回 值: 无

\*

\* 调用说明:仅在StatRateInfo母函数中作为子函数调用

\*/

void SortRateInfo(RATE\_NODE \*rate\_hd)

{

RATE\_NODE \*prate\_node\_prior;

RATE\_NODE \*prate\_node\_after;

RATE\_NODE \*prate\_node\_cur;

RATE\_NODE \*prate\_node\_temp;

prate\_node\_prior = rate\_hd;

if (prate\_node\_prior == NULL)

{

return;

}

/\*排序时，用于交换结点信息的临时结点\*/

prate\_node\_temp = (RATE\_NODE \*) malloc(sizeof(RATE\_NODE));

while (prate\_node\_prior->next != NULL) /\*选择法排序\*/

{

prate\_node\_cur = prate\_node\_prior;

prate\_node\_after = prate\_node\_prior->next;

while (prate\_node\_after != NULL)

{

if (prate\_node\_cur->proportion < prate\_node\_after->proportion)

{

prate\_node\_cur = prate\_node\_after;

}

prate\_node\_after = prate\_node\_after->next;

}

/\*交换数据域，指针域不交换\*/

if (prate\_node\_cur != prate\_node\_prior)

{

\*prate\_node\_temp = \*prate\_node\_prior;

\*prate\_node\_prior = \*prate\_node\_cur;

prate\_node\_prior->next = prate\_node\_temp->next;

prate\_node\_temp->next = prate\_node\_cur->next;

\*prate\_node\_cur = \*prate\_node\_temp;

}

prate\_node\_prior = prate\_node\_prior->next;

}

free(prate\_node\_temp);

return;

}

/\*\*

\* 函数名称: StatDeptInfoPre

\* 函数功能: 按院系统计某年度的科研项目信息

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 按要求输出统计结果

\*

\* 调用说明:

\*/

void StatDeptInfoPre(void)

{

char year[5];

printf("\n\t\t 请输入需要查询的年份（例2017）：");

scanf("%s",year);

getchar();

if(strlen(year) != 4)

{

printf("\n\t\t 年份格式输入错误 \n\n\t\t 按任意键继续……");

getch();

ClearScreen();

ShowMenu();

return;

}

DEPT\_NODE \*hd = gp\_head;

DEPT\_PROG\_NODE \*head;

DEPT\_PROG\_NODE \*pdp\_node\_temp;

head = StatDeptInfo(hd, year);

SortDeptProgInfo(head);

gotoxy(28, 3);

printf("下面按院系项目数输出院系信息：\n");

printf("\t院系名称\t973项目数\t863项目数\t该年项目总数\t科研总经费");

pdp\_node\_temp = head;

while(pdp\_node\_temp!=NULL)

{

printf("\n");

printf("\t%-12s\t%-8d\t%-8d\t%-8d\t%-.2f", pdp\_node\_temp->dept\_name, pdp\_node\_temp->dept\_prog\_amount1, pdp\_node\_temp->dept\_prog\_amount3, pdp\_node\_temp->dept\_prog\_amount, pdp\_node\_temp->dept\_fee);

pdp\_node\_temp = pdp\_node\_temp->next;

}

printf("\n\t\t\t 按任意键继续……\n");

getch();

ClearScreen();

ShowMenu();

return;

}

/\*\*

\* 函数名称: StatDeptInfo

\* 函数功能: 按院系统计某年度的科研项目信息

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: DEPT\_PROG\_NODE \*类型的头结点

\*

\* 调用说明:

\*/

DEPT\_PROG\_NODE\* StatDeptInfo(DEPT\_NODE \*hd, char year[5])

{

DEPT\_NODE \*pdept\_node = hd;

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

DEPT\_PROG\_NODE \*pdept\_prog\_node\_temp;

DEPT\_PROG\_NODE \*pdept\_prog\_node\_ret = NULL;

char prog\_year[5];

for(pdept\_node=hd; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

int prog973 = 0, prog863 = 0;

int progsum = 0;

float dept\_fee = 0;

pdept\_prog\_node\_temp=(DEPT\_PROG\_NODE\*)malloc(sizeof(DEPT\_PROG\_NODE));

pteam\_node = pdept\_node->tnext;

while (pteam\_node != NULL)

{

pprog\_node = pteam\_node->pnext;

while (pprog\_node != NULL)

{

strncpy(prog\_year, pprog\_node->st\_date, 4);

prog\_year[4] = '\0';

if (strcmp(prog\_year, year))

{

pprog\_node = pprog\_node->next;

continue;

}

dept\_fee += pprog\_node->prog\_fee;

if (pprog\_node->type == '1')

{

prog973++;

}

if (pprog\_node->type == '3')

{

prog863++;

}

progsum++;

pprog\_node = pprog\_node->next;

}

pteam\_node = pteam\_node->next;

}

pdept\_prog\_node\_temp->dept\_fee = dept\_fee;

pdept\_prog\_node\_temp->dept\_prog\_amount1 = prog973;

pdept\_prog\_node\_temp->dept\_prog\_amount3 = prog863;

strcpy(pdept\_prog\_node\_temp->dept\_name, pdept\_node->dept\_name);

pdept\_prog\_node\_temp->dept\_prog\_amount = progsum;

pdept\_prog\_node\_temp->next = pdept\_prog\_node\_ret;

pdept\_prog\_node\_ret = pdept\_prog\_node\_temp;

}

return pdept\_prog\_node\_ret;

}

/\*\*

\* 函数名称: SortDeptProgInfo

\* 函数功能: 对院系项目信息链进行排序

\* 输入参数: dp\_hd 院系项目信息链头指针

\* 输出参数: dp\_hd 排序结果通过头指针返回

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SortDeptProgInfo(DEPT\_PROG\_NODE \*dp\_hd)

{

DEPT\_PROG\_NODE \*pdept\_prog\_node\_prior;

DEPT\_PROG\_NODE \*pdept\_prog\_node\_after;

DEPT\_PROG\_NODE \*pdept\_prog\_node\_cur;

DEPT\_PROG\_NODE \*pdept\_prog\_node\_temp;

pdept\_prog\_node\_prior = dp\_hd;

if (pdept\_prog\_node\_prior == NULL)

{

return;

}

/\*排序时，用于交换结点信息的临时结点\*/

pdept\_prog\_node\_temp = (DEPT\_PROG\_NODE \*) malloc(sizeof(DEPT\_PROG\_NODE));

while (pdept\_prog\_node\_prior->next != NULL) /\*选择法排序\*/

{

pdept\_prog\_node\_cur = pdept\_prog\_node\_prior;

pdept\_prog\_node\_after = pdept\_prog\_node\_prior->next;

while (pdept\_prog\_node\_after != NULL)

{

if(pdept\_prog\_node\_cur->dept\_prog\_amount< pdept\_prog\_node\_after->dept\_prog\_amount)

{

pdept\_prog\_node\_cur = pdept\_prog\_node\_after;

}

pdept\_prog\_node\_after = pdept\_prog\_node\_after->next;

}

/\*交换数据域，指针域不交换\*/

if (pdept\_prog\_node\_cur != pdept\_prog\_node\_prior)

{

\*pdept\_prog\_node\_temp = \*pdept\_prog\_node\_prior;

\*pdept\_prog\_node\_prior = \*pdept\_prog\_node\_cur;

pdept\_prog\_node\_prior->next = pdept\_prog\_node\_temp->next;

pdept\_prog\_node\_temp->next = pdept\_prog\_node\_cur->next;

\*pdept\_prog\_node\_cur = \*pdept\_prog\_node\_temp;

}

pdept\_prog\_node\_prior = pdept\_prog\_node\_prior->next;

}

free(pdept\_prog\_node\_temp);

return;

}

BOOL StatDept(void)

{

int hotnum = 0;

char \*pString[] = {"院系师生数目情况",

"院系项目数量情况",

"返回"

};

hotnum = MyMenu3(pString, 3);/\*选择统计对象的范围\*/

switch (hotnum)

{

case 1:

StatRateInfo(); /\*统计院系师生数目情况\*/

break;

case 2:

StatDeptInfoPre();/\*统计院系项目数情况\*/

break;

case 3:

break;

}

return TRUE;

}

BOOL StatTeam(void)

{

int hotnum = 0;

char \*pString[] = {"自然基金项目情况",

"项目教师人数比情况",

"返回"

};

hotnum = MyMenu3(pString, 3);/\*选择统计对象的范围\*/

switch (hotnum)

{

case 1:

StatMostProg();/\*统计国家自然基金项目团队情况\*/

break;

case 2:

StatProgRate();/\*统计团队科研项目数和教师人数的比值情况\*/

break;

case 3:

break;

}

return TRUE;

}

BOOL StatProg(void)

{

int hotnum = 0;

char \*pString[] = {"--------------",

"经费最高的项目",

"返回"

};

hotnum = MyMenu3(pString, 3);/\*选择统计对象的范围\*/

switch (hotnum)

{

case 1:

case 2:

StatRichProg();/\*统计经费较高的科研项目情况\*/

break;

case 3:

break;

}

return TRUE;

}

/\*\*

\* 函数名称: StatMostProg

\* 函数功能: 按国家自然基金项目数量排序后输出

\* 输入参数: 无

\* 输出参数: 按照要求按序输出

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void StatMostProg(void)

{

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*pdept\_node = head;

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

MOST\_PROG\_NODE \*pmost\_prog\_node\_temp;

MOST\_PROG\_NODE \*pmost\_prog\_node\_ret = NULL;

int count = 0;

for(pdept\_node=head; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

pteam\_node = pdept\_node->tnext;

while (pteam\_node != NULL)

{

int progamount2 = 0;

float progfee2 = 0;

pmost\_prog\_node\_temp=(MOST\_PROG\_NODE\*)malloc(sizeof(MOST\_PROG\_NODE));

pprog\_node = pteam\_node->pnext;

while(pprog\_node != NULL)

{

if (pprog\_node->type == '2')

{

progfee2 += pprog\_node->prog\_fee;

progamount2++;

}

pprog\_node = pprog\_node->next;

}

pmost\_prog\_node\_temp->prog\_fee2 = progfee2;

pmost\_prog\_node\_temp->team\_prog\_amount2 = progamount2;

strcpy(pmost\_prog\_node\_temp->team\_name, pteam\_node->team\_name);

pteam\_node = pteam\_node->next;

pmost\_prog\_node\_temp->next = pmost\_prog\_node\_ret;

pmost\_prog\_node\_ret = pmost\_prog\_node\_temp;

}

}

SortMostProgInfo(pmost\_prog\_node\_ret);

gotoxy(28, 3);

printf("下面按国家自然基金项目总数输出团队信息：");

printf("\n\t团队名称\t\t国家自然基金项目总数\t国家自然基金项目总经费");

pmost\_prog\_node\_temp = pmost\_prog\_node\_ret;

while (pmost\_prog\_node\_temp!=NULL && count<10)

{

count++;

printf("\n");

printf("\t %-20s\t%-8d\t%-8.2f", pmost\_prog\_node\_temp->team\_name, pmost\_prog\_node\_temp->team\_prog\_amount2, pmost\_prog\_node\_temp->prog\_fee2);

pmost\_prog\_node\_temp = pmost\_prog\_node\_temp->next;

}

printf("\n\t\t\t按任意键继续……");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: StatProgRate

\* 函数功能: 按团队项目教师人数比排序后输出

\* 输入参数: 无

\* 输出参数: 按照要求按序输出

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void StatProgRate(void)

{

DEPT\_NODE \*head = gp\_head;

DEPT\_NODE \*pdept\_node = head;

TEAM\_NODE \*pteam\_node;

PROG\_NODE \*pprog\_node;

PROG\_PROF\_RATE\_NODE \*prog\_prof\_node\_temp;

PROG\_PROF\_RATE\_NODE \*prog\_prof\_node\_ret = NULL;

int count = 0;

for(pdept\_node=head; pdept\_node!=NULL; pdept\_node=pdept\_node->next)

{

pteam\_node = pdept\_node->tnext;

while (pteam\_node != NULL)

{

int prog\_num = 0;

float pprate;

prog\_prof\_node\_temp=(PROG\_PROF\_RATE\_NODE\*)malloc(sizeof(PROG\_PROF\_RATE\_NODE));

pprog\_node = pteam\_node->pnext;

while(pprog\_node != NULL)

{

prog\_num++;

pprog\_node = pprog\_node->next;

}

pprate = prog\_num\*1.0/pteam\_node->team\_prof\_amount;

prog\_prof\_node\_temp->team\_prof\_amount = pteam\_node->team\_prof\_amount;

prog\_prof\_node\_temp->team\_prog\_amount = prog\_num;

prog\_prof\_node\_temp->team\_prog\_prof\_rate = pprate;

strcpy(prog\_prof\_node\_temp->team\_name, pteam\_node->team\_name);

pteam\_node = pteam\_node->next;

prog\_prof\_node\_temp->next = prog\_prof\_node\_ret;

prog\_prof\_node\_ret = prog\_prof\_node\_temp;

}

}

SortProgProfInfo(prog\_prof\_node\_ret);

gotoxy(28, 3);

printf("下面按团队项目教师人数比输出5条团队信息：");

printf("\n\t团队名称\t科研项目数\t教师人数\t项目教师人数比");

prog\_prof\_node\_temp = prog\_prof\_node\_ret;

while (prog\_prof\_node\_temp!=NULL && count<5)

{

count++;

if(prog\_prof\_node\_temp->team\_prof\_amount==0)

{

printf("\n");

printf("\t%-12s\t%-8d\t%-8d\t教师人数不可为0", prog\_prof\_node\_temp->team\_name, prog\_prof\_node\_temp->team\_prog\_amount, prog\_prof\_node\_temp->team\_prof\_amount);

continue;

}

printf("\n");

printf("\t%-12s\t%-8d\t%-8d\t%-8.2f", prog\_prof\_node\_temp->team\_name, prog\_prof\_node\_temp->team\_prog\_amount, prog\_prof\_node\_temp->team\_prof\_amount, prog\_prof\_node\_temp->team\_prog\_prof\_rate);

prog\_prof\_node\_temp = prog\_prof\_node\_temp->next;

}

printf("\n\t\t\t\t按任意键继续……");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SortProgProfInfo

\* 函数功能: 对团队项目教师人数比信息链进行排序

\* 输入参数: pp\_hd 团队项目教师人数比信息链头指针

\* 输出参数: pp\_hd 排序结果通过头指针返回

\* 返 回 值: pp\_hd 排序结果通过头指针返回

\*

\* 调用说明:

\*/

void SortProgProfInfo(PROG\_PROF\_RATE\_NODE \*pp\_hd)

{

PROG\_PROF\_RATE\_NODE \*prog\_prof\_node\_prior;

PROG\_PROF\_RATE\_NODE \*prog\_prof\_node\_after;

PROG\_PROF\_RATE\_NODE \*prog\_prof\_node\_cur;

PROG\_PROF\_RATE\_NODE \*prog\_prof\_node\_temp;

prog\_prof\_node\_prior = pp\_hd;

if (prog\_prof\_node\_prior == NULL)

{

return;

}

/\*排序时，用于交换结点信息的临时结点\*/

prog\_prof\_node\_temp = (PROG\_PROF\_RATE\_NODE \*) malloc(sizeof(PROG\_PROF\_RATE\_NODE));

while (prog\_prof\_node\_prior->next != NULL) /\*选择法排序\*/

{

prog\_prof\_node\_cur = prog\_prof\_node\_prior;

prog\_prof\_node\_after = prog\_prof\_node\_prior->next;

while (prog\_prof\_node\_after != NULL)

{

if (prog\_prof\_node\_cur->team\_prog\_prof\_rate < prog\_prof\_node\_after->team\_prog\_prof\_rate)

{

prog\_prof\_node\_cur = prog\_prof\_node\_after;

}

prog\_prof\_node\_after = prog\_prof\_node\_after->next;

}

/\*交换数据域，指针域不交换\*/

if (prog\_prof\_node\_cur != prog\_prof\_node\_prior)

{

\*prog\_prof\_node\_temp = \*prog\_prof\_node\_prior;

\*prog\_prof\_node\_prior = \*prog\_prof\_node\_cur;

prog\_prof\_node\_prior->next = prog\_prof\_node\_temp->next;

prog\_prof\_node\_temp->next = prog\_prof\_node\_cur->next;

\*prog\_prof\_node\_cur = \*prog\_prof\_node\_temp;

}

prog\_prof\_node\_prior = prog\_prof\_node\_prior->next;

}

free(prog\_prof\_node\_temp);

return;

}

/\*\*

\* 函数名称: SortMostProgInfo

\* 函数功能: 对科研团队国家基金项目信息链进行排序

\* 输入参数: mp\_hd 科研团队国家基金项目信息链头指针

\* 输出参数: mp\_hd 排序结果通过头指针返回

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void SortMostProgInfo(MOST\_PROG\_NODE \*mp\_hd)

{

MOST\_PROG\_NODE \*pmost\_prog\_prior;

MOST\_PROG\_NODE \*pmost\_prog\_after;

MOST\_PROG\_NODE \*pmost\_prog\_cur;

MOST\_PROG\_NODE \*pmost\_prog\_temp;

pmost\_prog\_prior = mp\_hd;

if (pmost\_prog\_prior == NULL)

{

return;

}

/\*排序时，用于交换结点信息的临时结点\*/

pmost\_prog\_temp=(MOST\_PROG\_NODE\*)malloc(sizeof(MOST\_PROG\_NODE));

while (pmost\_prog\_prior->next != NULL) /\*选择法排序\*/

{

pmost\_prog\_cur = pmost\_prog\_prior;

pmost\_prog\_after = pmost\_prog\_prior->next;

while (pmost\_prog\_after != NULL)

{

if (pmost\_prog\_cur->team\_prog\_amount2 < pmost\_prog\_after->team\_prog\_amount2)

{

pmost\_prog\_cur = pmost\_prog\_after;

}

pmost\_prog\_after = pmost\_prog\_after->next;

}

/\*交换数据域，指针域不交换\*/

if (pmost\_prog\_cur != pmost\_prog\_prior)

{

\*pmost\_prog\_temp = \*pmost\_prog\_prior;

\*pmost\_prog\_prior = \*pmost\_prog\_cur;

pmost\_prog\_prior->next = pmost\_prog\_temp->next;

pmost\_prog\_temp->next = pmost\_prog\_cur->next;

\*pmost\_prog\_cur = \*pmost\_prog\_temp;

}

pmost\_prog\_prior = pmost\_prog\_prior->next;

}

free(pmost\_prog\_temp);

return;

}

BOOL AboutResearch(void)

{

int hotnum = 0;

char \*pString[] = {"973项目特点介绍",

"863项目特点介绍",

"国家自然科学基金项目介绍",

"返回"

};

while(hotnum != 4)

{

hotnum = MyMenu4(pString, 4);/\*选择统计对象的范围\*/

switch (hotnum)

{

case 1:

Intro973();/\*介绍973项目\*/

break;

case 2:

Intro863();/\*介绍863项目\*/

break;

case 3:

IntroNationSci();/\*介绍国家自然科学基金项目\*/

break;

}

}

return TRUE;

}

void Intro973(void)

{

char \*info973[] = {"决策模式科学：专家咨询与政府决策相结合",

"管理模式科学：项目管理与经费管理有机结合",

"组织模式完善：多部门行业共同推进计划发展",

"返回"

};

ShowModule(info973, 4);

return;

}

void Intro863(void)

{

char \*info973[] = {"1.提高我国自主创新能力",

"2.坚持战略性、前沿性和前瞻性",

"3.促进基础研究与国家目标结合",

"返回"

};

ShowModule(info973, 4);

return;

}

void IntroNationSci(void)

{

char \*infonationsci[] = {"战略定位：支持基础研究，发挥导向作用",

"工作方针：激励创新、引领未来",

"战略：源头创新，打造创新环境",

"返回"

};

ShowModule(infonationsci, 4);

return;

}

BOOL HelpTopic(void)

{

char \*plabel\_name[] = {"按ALT+X可以直接退出程序",

"按ALT+字母可以快捷选择主菜单。",

"按F1键可以获取帮助菜单",

"在子菜单弹出的情况按esc键可退出子菜单",

"确认"

};

ShowModule(plabel\_name, 5);

return TRUE;

}

/\*\*

\* 函数名称: SeekDept\_All

\* 函数功能: 遍历院系信息链表，输出所有院系信息

\* 输入参数: head 主链头指针

\* 输出参数: 所有的院系基本信息

\* 返 回 值: 无

\*

\* 调用说明:测试函数，非测试状态不调用

\*/

void SeekDept\_All(DEPT\_NODE \*head)

{

DEPT\_NODE \*pdept = head;

while(pdept != NULL)

{

printf("\t\t\t %-15s\t%-10s\n", pdept->dept\_name, pdept->dmgr\_name);

pdept = pdept->next;

}

printf("\n\t\t\t\t按任意键继续。。。");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekTeam\_All

\* 函数功能: 无条件输出所有基本信息

\* 输入参数: hd 主链头指针

\* 输出参数: 所有团队基本信息

\* 返 回 值: 无

\*

\* 调用说明:测试函数，非测试状态不调用

\*/

void SeekTeam\_All(DEPT\_NODE \*head)

{

DEPT\_NODE \*pd = head;

TEAM\_NODE \*pt = NULL;

while(pd != NULL)

{

pt = pd->tnext;

while(pt != NULL)

{

/\*遍历链表输出所有基本信息\*/

printf("%-15s\t%-15s\n", pt->dept\_name, pt->team\_name);

pt = pt->next;

}

pd = pd->next;

}

printf("\n\t\t\t 按任意键继续。。。\n");

getch();

ClearScreen();

ShowMenu();

}

/\*\*

\* 函数名称: SeekProg\_ALL

\* 函数功能: 输出所有项目基本信息，这是测试函数

\* 输入参数: head 主链头指针

\* 输出参数: 输出所有项目基本信息

\* 返 回 值: 无

\*

\* 调用说明:测试函数，非测试状态不调用

\*/

void SeekProg\_All(DEPT\_NODE \*head)

{

DEPT\_NODE \*pdept = head;

TEAM\_NODE \*pteam = NULL;

PROG\_NODE \*pprog = NULL;

while(pdept != NULL)

{

pteam = pdept->tnext;

while(pteam != NULL)

{

pprog = pteam->pnext;

while(pprog != NULL)

{

printf("%-12s\t%-20s\n",pprog->pmgr\_name,pprog->team\_name);

pprog = pprog->next;

}

pteam = pteam->next;

}

pdept = pdept->next;

}

printf("\n\t\t\t 按任意键继续。。。\n");

getch();

ClearScreen();

ShowMenu();

}

# 六、运行测试与结果分析

6.1输入数据

1.院系基本信息：

|  |  |  |
| --- | --- | --- |
| 院系名称 | 院系负责人 | 联系电话 |
| 光电学院 | 张新亮 | 027-87793416 |
| 物理学院 | 陈相松 | 027-87543881 |
| 计算机学院 | 冯丹 | 027-87792450 |

2.团队基本信息：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 团队名称 | 负责人姓名 | 研究生人数 | 教师人数 | 所属院系 |
| 信息安全 | 李瑞轩 | 7 | 15 | 计算机学院 |
| 数据存储 | 王芳 | 12 | 7 | 计算机学院 |
| 数字媒体研究 | 凌贺飞 | 37 | 22 | 计算机学院 |
| 凝聚态物理 | 王顺 | 18 | 26 | 物理学院 |
| 引力中心 | 罗俊 | 56 | 32 | 物理学院 |
| 超快光学 | 兰鹏飞 | 20 | 26 | 物理学院 |
| 激光雷达 | 杨克成 | 25 | 17 | 光电学院 |
| 微波工程 | 龚荣洲 | 39 | 23 | 光电学院 |

3.项目基本信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目编号 | 所属团队 | 经费(万元) | 项目类别 | 起始时间 | 项目负责人 |
| 01217030201 | 凝聚态物理 | 7.90 | 2 | 2017/03 | 王顺 |
| 01217080102 | 凝聚态物理 | 12.30 | 1 | 2017/08 | 吴梦昊 |
| 01217050303 | 凝聚态物理 | 2.60 | 3 | 2017/05 | 高国营 |
| 01217030404 | 凝聚态物理 | 5.20 | 4 | 2017/03 | 姚凯伦 |
| 01217020101 | 引力中心 | 27.60 | 1 | 2017/02 | 罗俊 |
| 01217010302 | 引力中心 | 10.20 | 3 | 2017/01 | 杨山清 |
| 01217040203 | 引力中心 | 7.50 | 2 | 2017/04 | 王炎 |
| 01217060504 | 引力中心 | 6.70 | 5 | 2017/06 | 涂良成 |
| 01217020105 | 引力中心 | 10.20 | 1 | 2017/02 | 罗鹏顺 |
| 01217050101 | 超快光学 | 9.40 | 1 | 2017/05 | 李智华 |
| 01217080102 | 超快光学 | 7.60 | 1 | 2017/08 | 祝雪丰 |
| 01217060203 | 超快光学 | 6.80 | 2 | 2017/06 | 兰鹏飞 |
| 01217030204 | 超快光学 | 3.20 | 2 | 2017/03 | 周月明 |
| 01217090205 | 超快光学 | 10.10 | 2 | 2017/09 | 兰鹏飞 |
| 01217060206 | 超快光学 | 6.30 | 2 | 2017/06 | 祝雪丰 |
| 01217090109 | 信息安全 | 3.10 | 1 | 2017/09 | 李之棠 |
| 01217080308 | 信息安全 | 2.30 | 3 | 2017/08 | 张云鹤 |
| 01217060407 | 信息安全 | 9.60 | 4 | 2017/06 | 金海 |
| 01217010306 | 信息安全 | 10.90 | 3 | 2017/01 | 骆松峰 |
| 01217090304 | 信息安全 | 5.90 | 3 | 2017/09 | 鲁宏伟 |
| 01217040203 | 信息安全 | 4.10 | 2 | 2017/04 | 赵贻竹 |
| 01217050202 | 信息安全 | 2.40 | 2 | 2017/05 | 廖小飞 |
| 01217080201 | 信息安全 | 7.60 | 2 | 2017/08 | 韩兰胜 |
| 01217030301 | 微波工程 | 6.90 | 3 | 2017/03 | 杨清华 |
| 01217050302 | 微波工程 | 8.40 | 3 | 2017/05 | 占腊民 |
| 01217090403 | 微波工程 | 5.60 | 4 | 2017/09 | 徐勤芬 |
| 01217050101 | 激光雷达 | 14.30 | 1 | 2017/05 | 杨克成 |
| 01217030202 | 激光雷达 | 5.60 | 2 | 2017/03 | 徐智谋 |
| 01217060203 | 激光雷达 | 7.90 | 2 | 2017/06 | 元秀华 |
| 01217010104 | 激光雷达 | 0.40 | 1 | 2017/01 | 费鹏 |
| 01217040207 | 数字媒体研究 | 5.00 | 2 | 2017/04 | 管涛 |
| 01217060206 | 数字媒体研究 | 6.20 | 2 | 2017/06 | 邹复好 |
| 01216120205 | 数字媒体研究 | 3.80 | 2 | 2016/12 | 于俊清 |
| 01216030504 | 数字媒体研究 | 4.20 | 5 | 2016/03 | 王天江 |
| 01216110303 | 数字媒体研究 | 4.00 | 3 | 2016/11 | 金人超 |
| 01216080102 | 数字媒体研究 | 7.70 | 1 | 2016/08 | 凌贺飞 |
| 01215010101 | 数字媒体研究 | 10.30 | 1 | 2015/01 | 管涛 |
| 01217060101 | 数据存储 | 8.80 | 1 | 2017/06 | 冯丹 |
| 02117060102 | 数据存储 | 11.20 | 1 | 2017/06 | 秦磊华 |
| 01217090203 | 数据存储 | 9.80 | 2 | 2017/09 | 王芳 |
| 01217060404 | 数据存储 | 4.20 | 4 | 2017/06 | 陈进才 |
| 01217090205 | 数据存储 | 10.60 | 2 | 2017/09 | 胡燏翀 |

6.2 系统初始化功能展示

运行结果：输出各类文件加载情况，如图

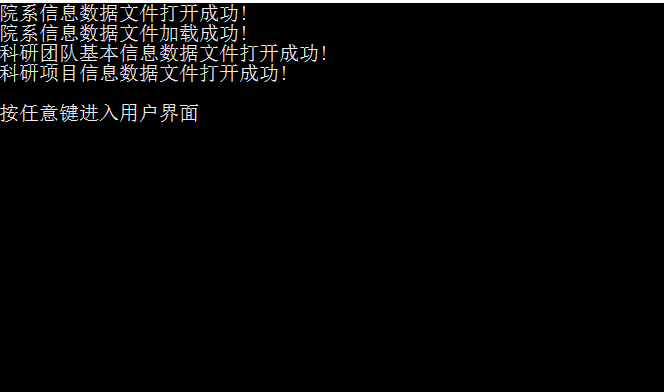


图 6.2.1 初始化界面

6.3 数据维护功能展示

**6.3.1 插入院系、团队、项目结点**

运行结果：将新建院系结点插入链表，并且自动保存信息，如图6.3.1所示

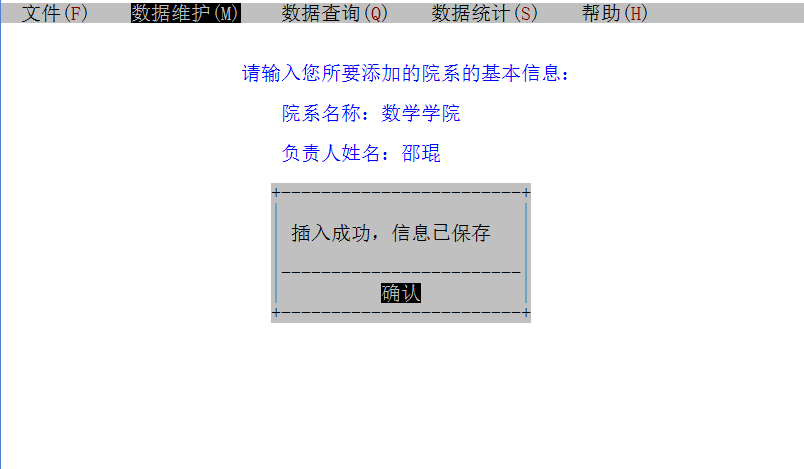


图6.3.1 插入院系结点功能展示

运行结果：将新建团队结点插入链表，并且自动保存信息，如图6.3.2所示

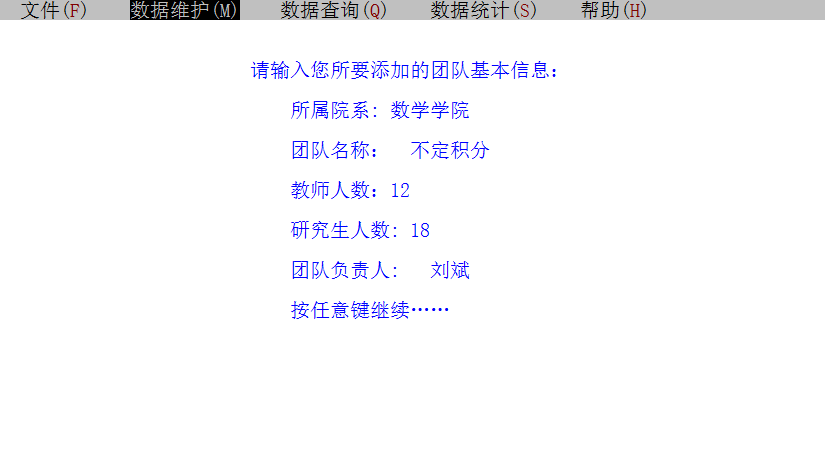


图6.3.2 插入团队结点功能展示

运行结果：将新建项目结点插入链表，并且自动保存信息，如图6.3.3所示

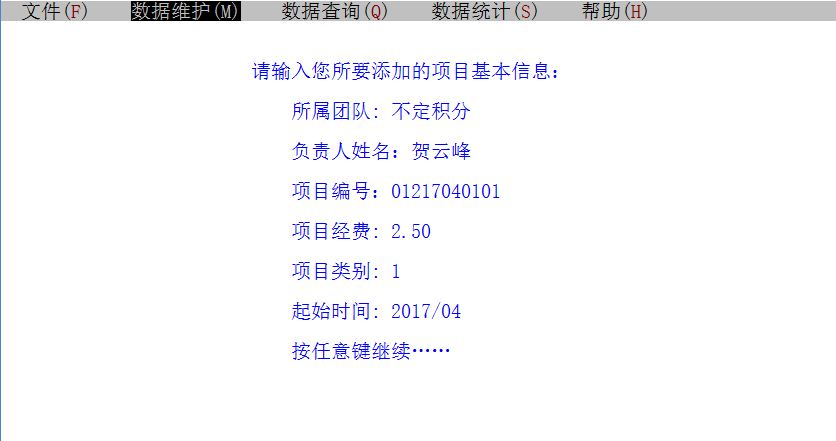


图6.3.3 插入项目结点功能展示

运行分析：为证明插入成功，下面调用数据查询三项功能，如图6.3.4所示：

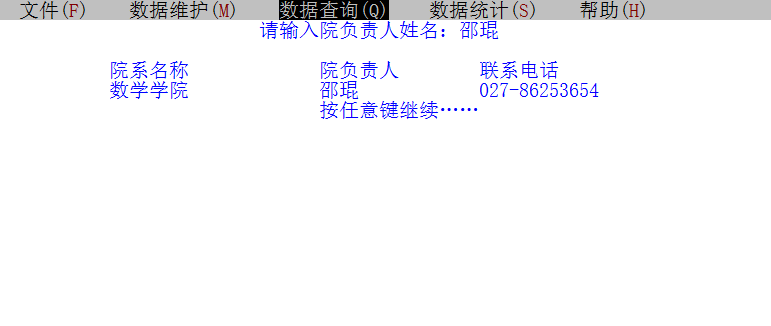


图6.3.4 验证插入院系信息成功

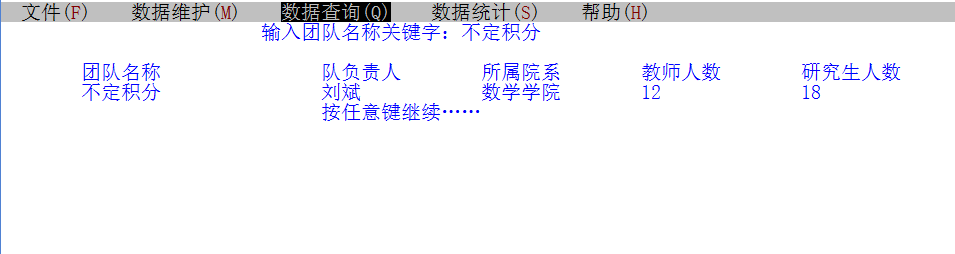


图6.3.5 验证插入团队信息成功

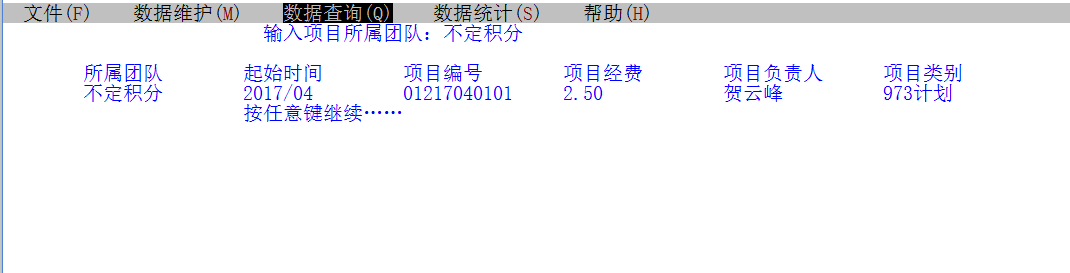


图6.3.6 验证插入项目信息成功

**6.3.2 修改院系、团队、项目结点**

运行结果：修改已存在的院系、团队、项目结点信息，并自动保存相应结果如图6.3.7~6.3.9所示

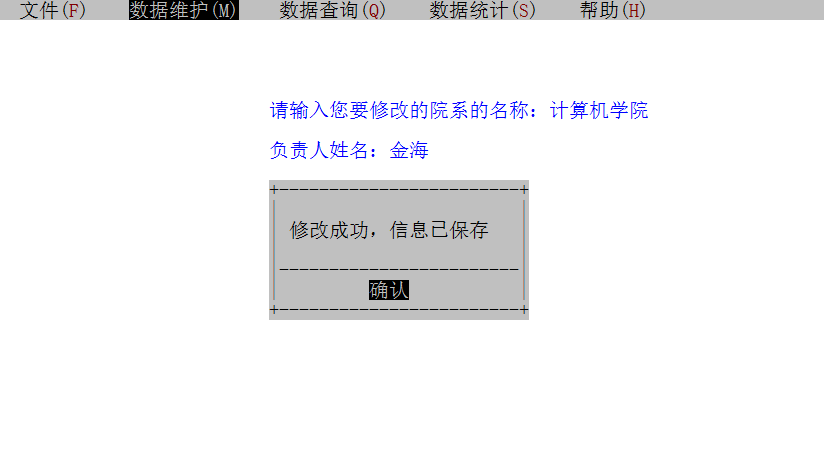


图6.3.7 修改院系结点信息功能展示

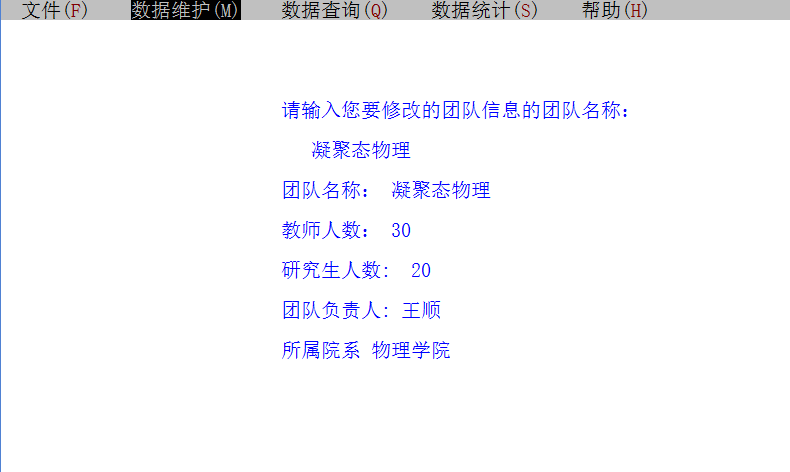


图6.3.8 修改团队信息结点功能展示

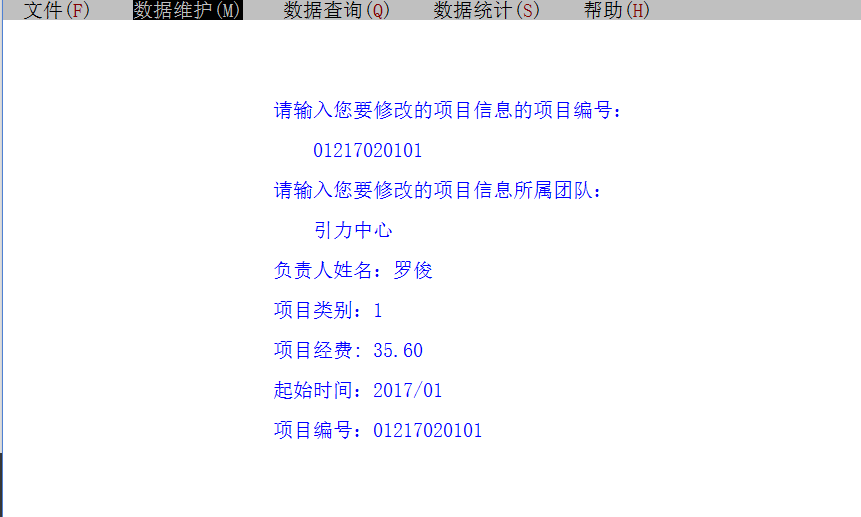


图6.3.9 修改项目信息结点功能展示

运行分析：为证明修改成功，下面调用数据查询功能展示修改结果，如图6.3.10~6.3.12所示。

将院负责人由“冯丹”改为“金海”

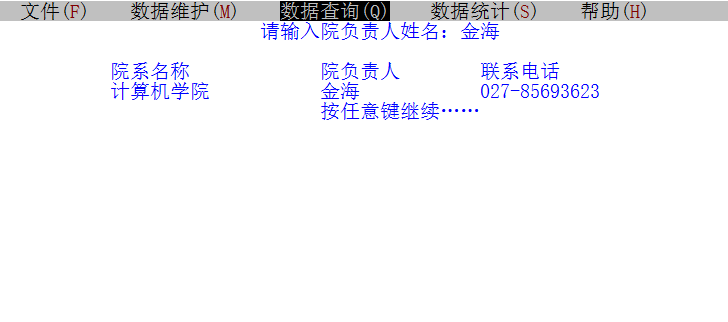


图6.3.10 验证修改院系结点信息功能

将“凝聚态物理”的教师人数和研究生人数进行更改：

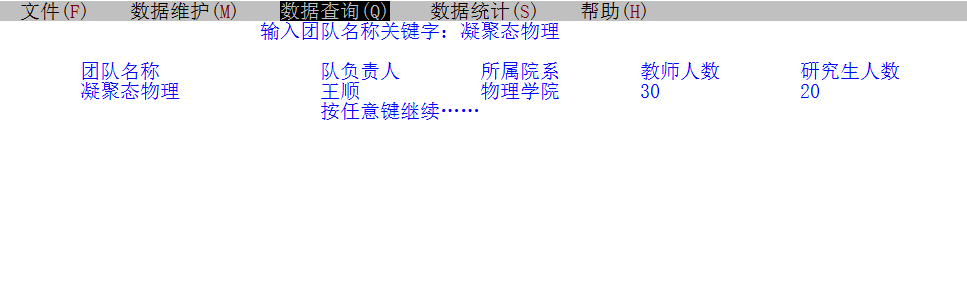


图6.3.11 验证修改团队信息功能

将引力中心01217020101项目的项目经费和起始时间进行更改：

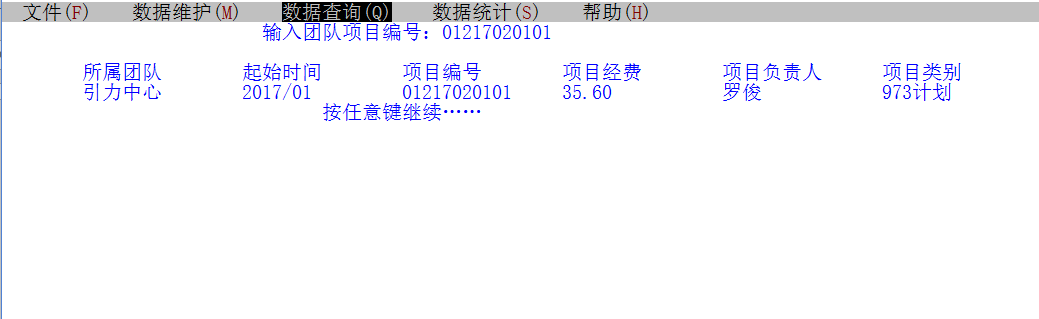


图6.3.12 验证修改项目信息功能

**6.3.3 修改院系、团队、项目结点**

运行结果：修改已存在的院系、团队、项目结点信息，并自动保存相应结果如图6.3.7~6.3.9所示

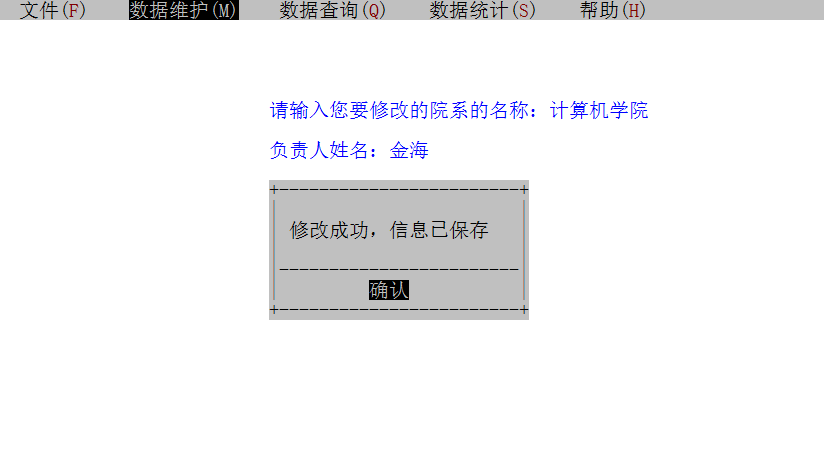


图6.3.7 修改院系结点信息功能展示

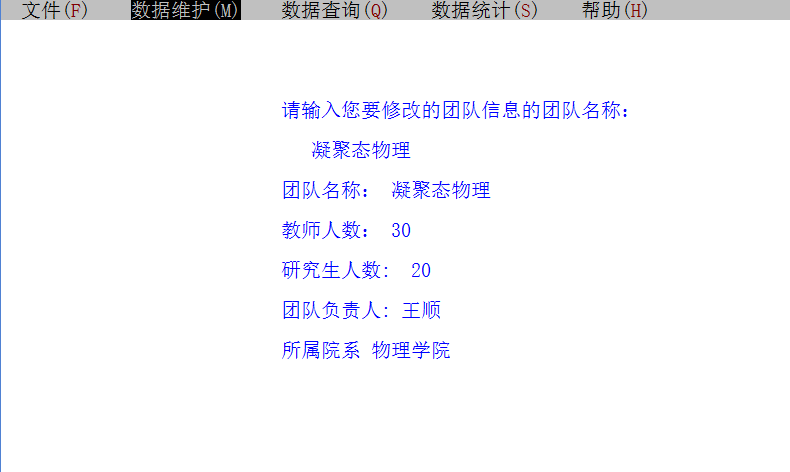


图6.3.8 修改团队信息结点功能展示

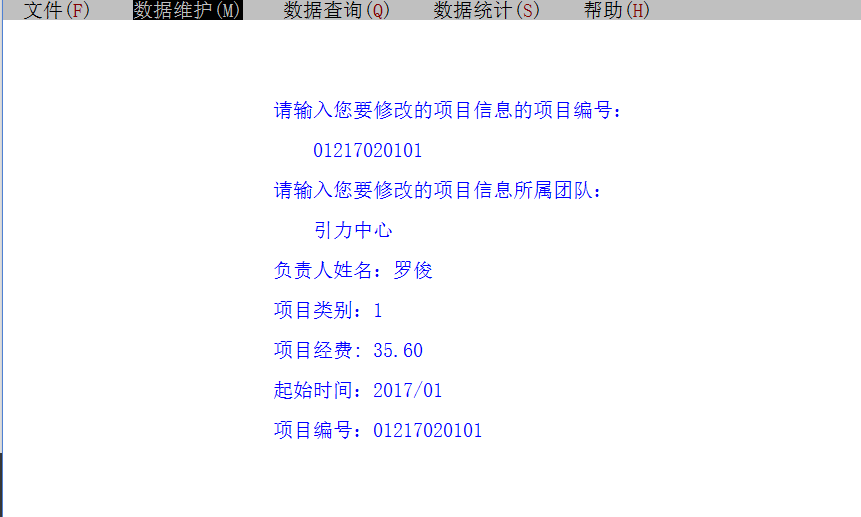


图6.3.9 修改项目信息结点功能展示

运行分析：为证明修改成功，下面调用数据查询功能展示修改结果，如图6.3.10~6.3.12所示。

将院负责人由“冯丹”改为“金海”

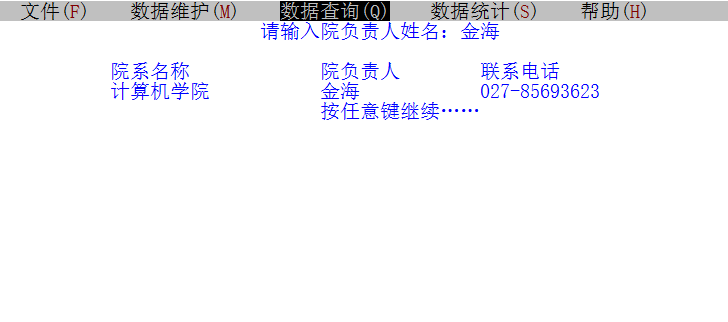


图6.3.10 验证修改院系结点信息功能

将“凝聚态物理”的教师人数和研究生人数进行更改：

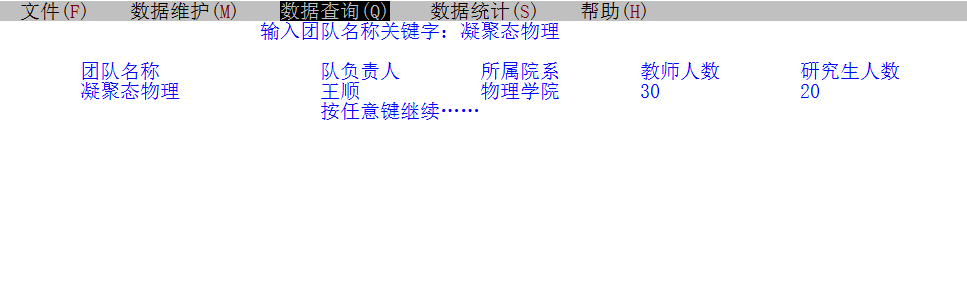


图6.3.11 验证修改团队信息功能

将引力中心01217020101项目的项目经费和起始时间进行更改：

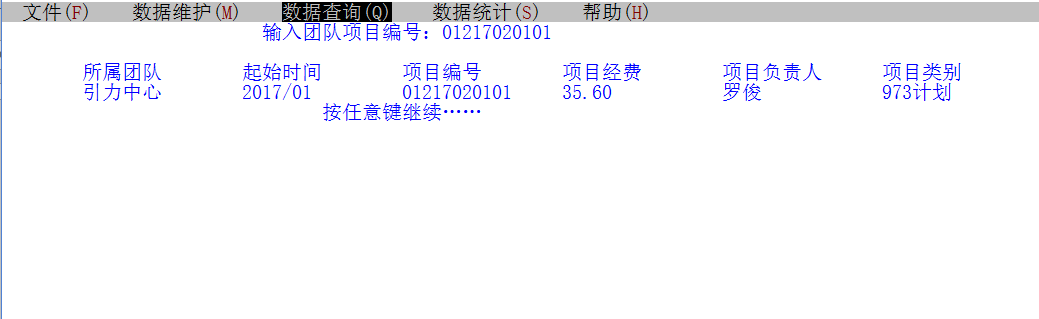


图6.3.12 验证修改项目信息功能

**6.3.3 删除院系、团队、项目结点**

运行结果：删除已存在的项目、团队、院系结点信息，并自动保存相应结果如图6.3.13~6.3.15所示

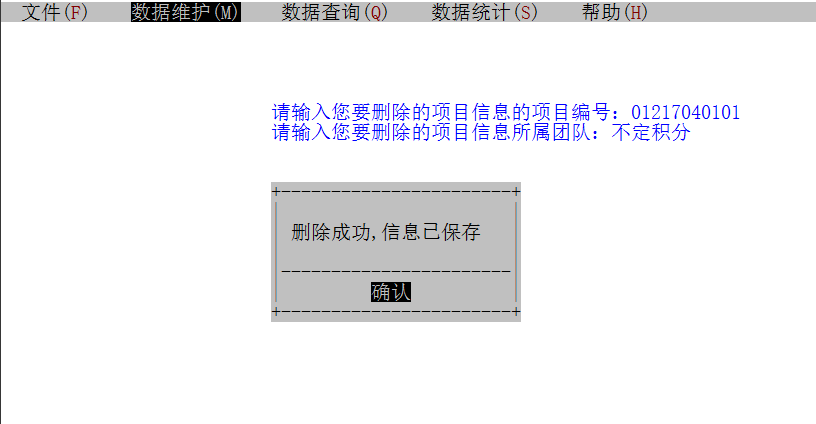


图6.3.13 删除项目结点信息功能展示

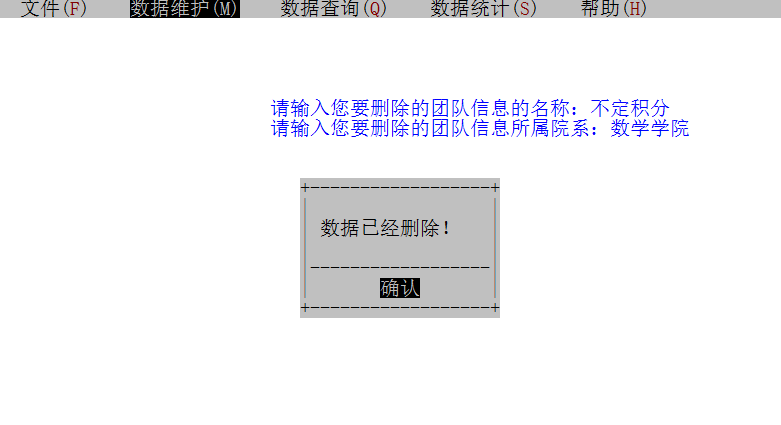


图6.3.14 删除团队信息结点功能展示

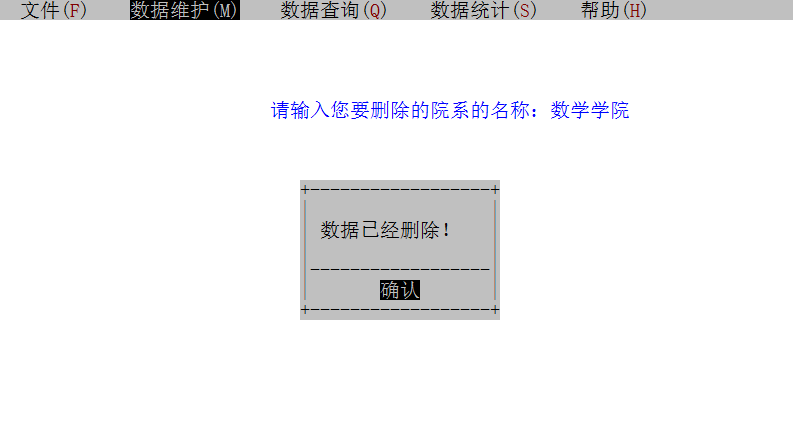


图6.3.15 删除院系信息结点功能展示

运行分析：为证明删除成功，下面调用数据查询功能展示修改结果，如图6.3.16~6.3.18所示。

如图6.3.16，“不定积分团队”中间已经没有项目了

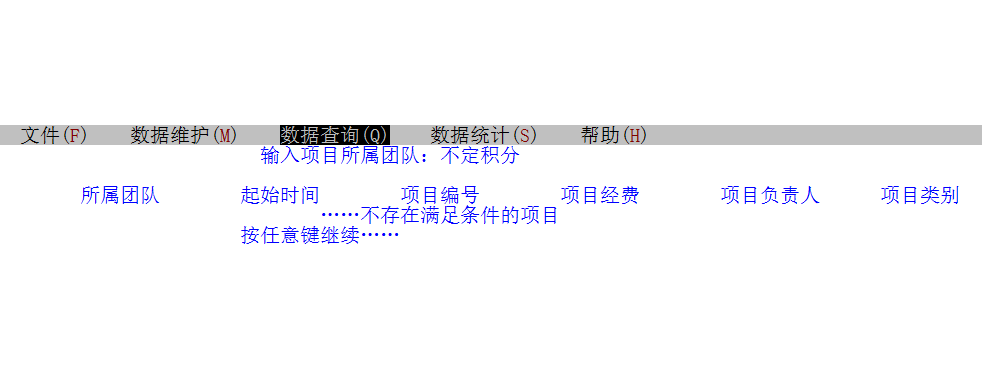


图6.3.16 验证删除项目结点信息功能

如图6.3.17，模糊查找含有"积分"字样的团队，已经不存在这样的团队了

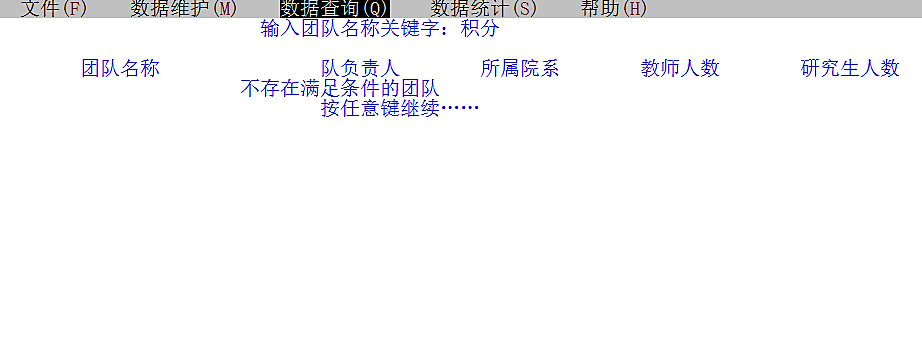


图6.3.17 验证删除团队信息功能

如图6.3.18，删除院系结点后已经不存在数学学院了：

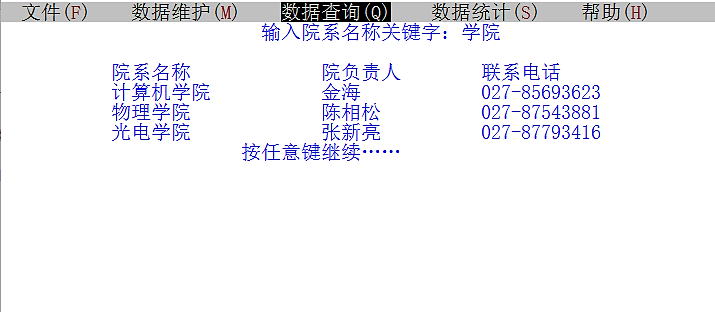


图6.3.18 验证删除院系信息功能

6.4 数据查询功能展示

**6.4.1 根据负责人精确查询院系信息**

**运行结果：如图6.4.1所示，输入光电学院负责人“张新亮”后输出光电学院相关信息。**

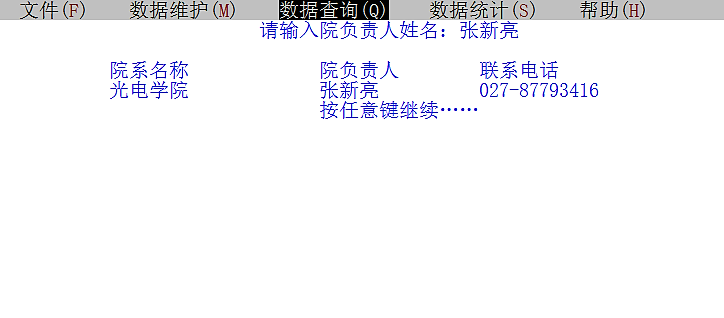


图 6.4.1 根据院系负责人精确查询院系信息

**6.4.2 根据院系关键字模糊查询院系信息**

**运行结果：如图6.4.2所示，输入“学院”后输出院系名称中带有“学院”两字的院系的相关信息。**

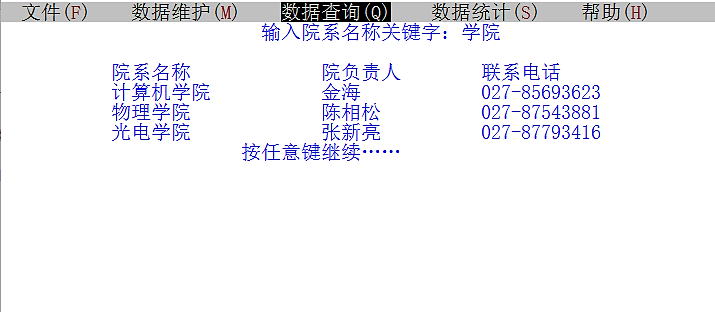


图 6.4.2 根据院系关键字模糊查询院系信息

**6.4.3 根据团队关键字模糊查询团队信息**

**运行结果：如图6.4.3所示，输入“光”后输出团队名称中带有“光”字的院系的相关信息。**

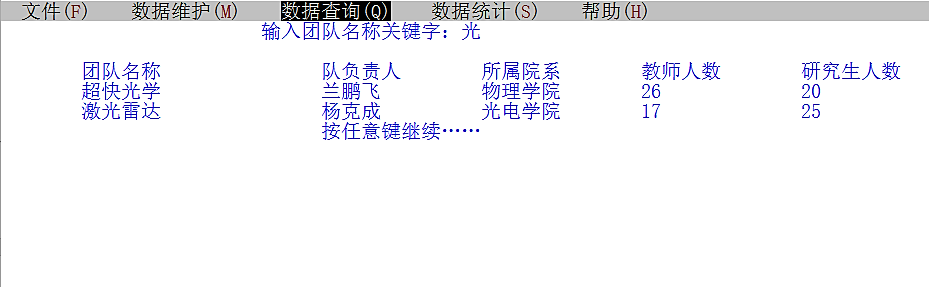
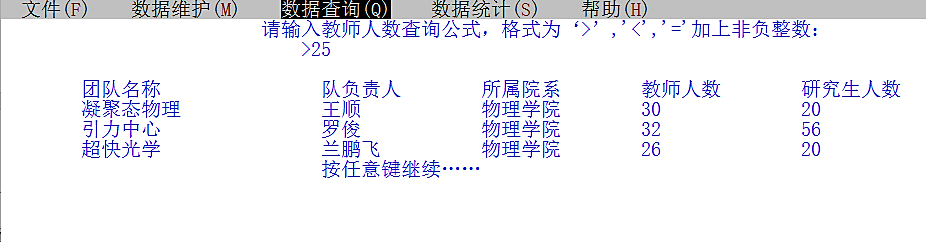


图 6.4.3 根据团队关键字模糊查询团队信息

**6.4.4 根据团队教师人数查询公式查询团队信息**

**运行结果：如图6.4.4所示，查询团队教师人数的公式后输出满足条件的团队信息。**



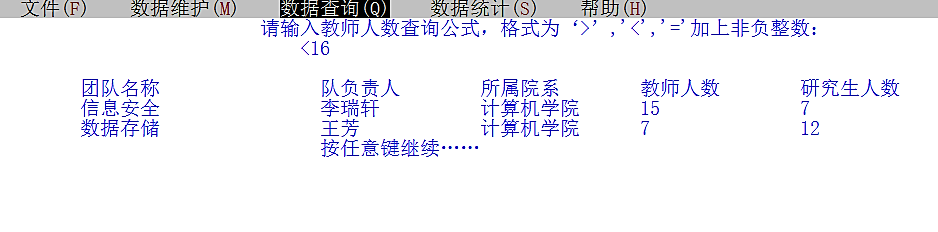


图 6.4.4 根据团队教师人数查询公式查询团队信息

**6.4.5 根据项目编号查询院系信息**

**运行结果：如图6.4.5所示，输入项目编号后输出符合该项目编号的项目的相关信息。**

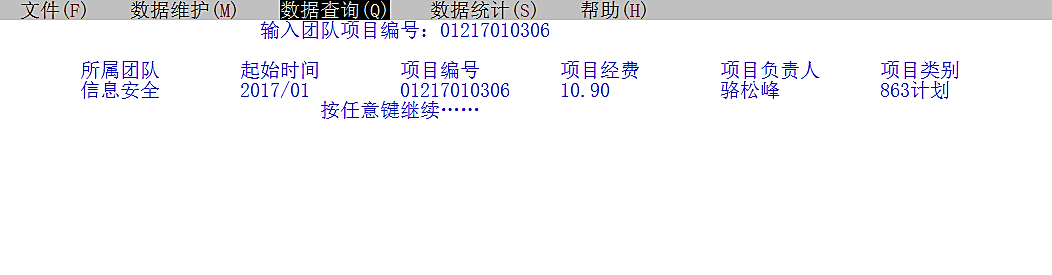


图 6.4.5 根据项目编号查询项目信息

**6.4.6 根据所属团队查询院系信息**

**运行结果：如图6.4.6所示，输入所属团队后输出该团队下属所有项目的相关信息。**



图 6.4.6 根据所属团队查询项目信息

6.5 数据统计功能展示

**6.5.1 根据学生教师人数比输出院系信息**

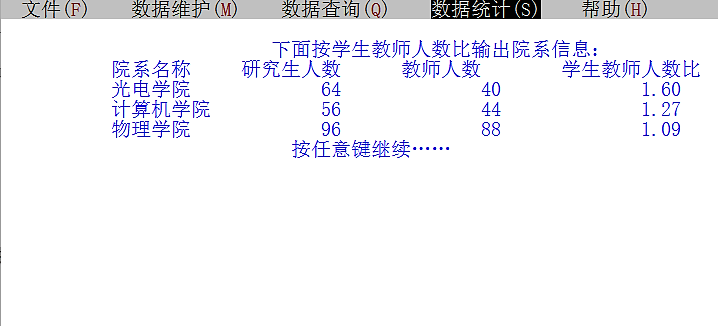
**运行结果：如图6.5.1所示，按学生教师人数比值降序排序后，输出统计结果**。

图 6.5.1 根据学生教师人数比统计院系信息

**6.5.2 科研项目数降序排序后输出院系信息**

**运行结果：如图6.5.2所示，输入年度后按科研项目数降序排序后输出。**

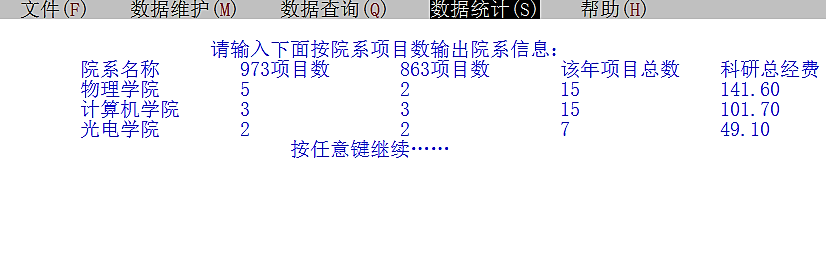


图 6.5.2 **按科研项目数统计院系**信息

**6.5.3 按照国家自然基金项目数统计团队信息**

**运行结果：如图6.5.3所示，输入年度后按照国家自然基金项目数排序后输出团队信息。**

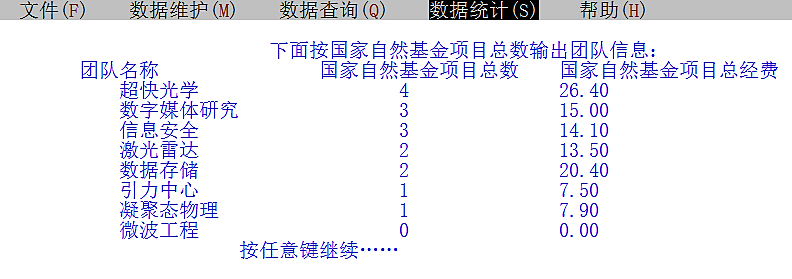


图 6.5.3 **按照国家自然基金项目数统计团队信息**

**6.5.4 按项目教师人数比值统计团队信息**

**运行结果：如图6.5.4所示，按项目教师人数比值降序排序后，输出团队信息。**

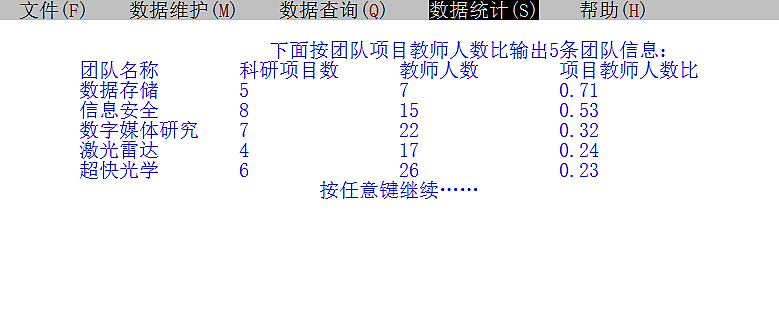


图 6.5.4 **按项目教师人数比值统计团队信息**

**6.5.6 根据项目经费统计项目信息**

**运行结果：如图6.5.6所示，按照项目经费数量统计输出项目的相关信息。**

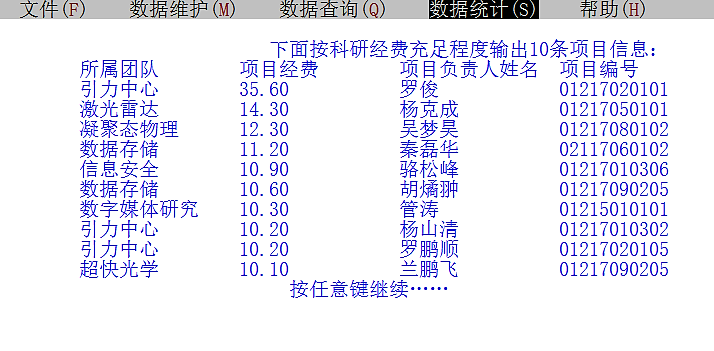


图 6.5.6 根据项目经费统计项目信息

6.6 帮助功能展示

**6.6.1 提供快捷键信息**

**运行结果：如图6.6.1所示，提供系统快捷键的操作。**

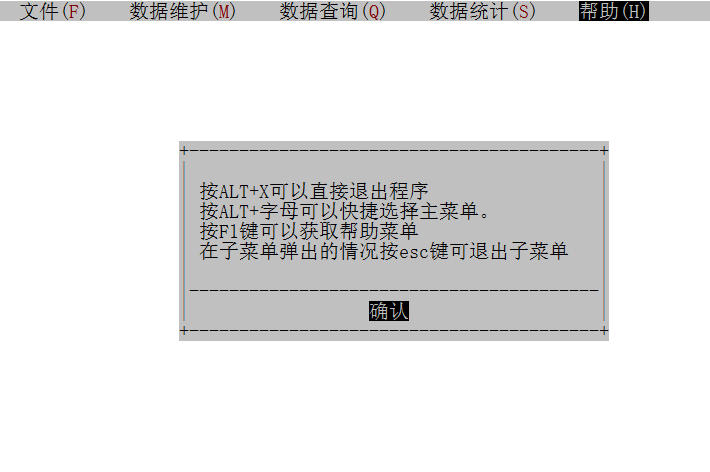
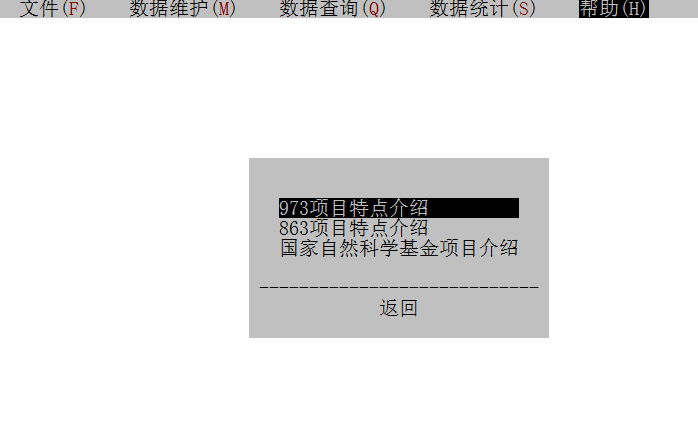


图 6.6.1 帮助功能展示

**6.6.2 提供各个项目的特点介绍**

**运行结果：如图6.6.2所示，介绍各个项目的特点**



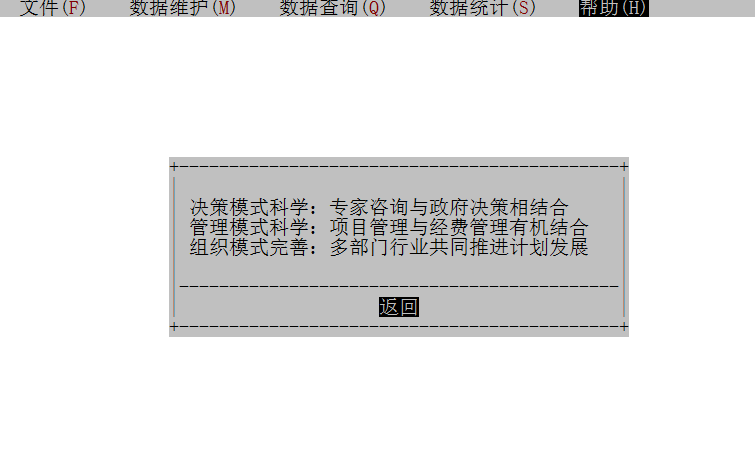


图 6.6.2 介绍项目特点的功能展示

# 七、总结

**代码命名规范性的提升与习惯的培养**

这次课程设计开发难度大，所花时间长，这样的特点使我不得不采取C语言课上老师的一些中肯的编程建议（这些建议在平时简单的作业中我没有采纳，因为我当时没有体会到他们的重要性）。正是这种“不得已而为之”，使我的编程规范性得到了提升。例如，由于函数众多，开发周期长，我不可能像平时做作业一样每个函数就起一个"test1"，"test2"这样的函数名，所以我尽量按照课程设计书上的要求，将每一个函数名都尽量起的更有意义，用英文单词组合的形式表现出它的功能。这显然是一个正确的良好的习惯，我觉得没有这种大批量的练习我是不可能养成这种习惯的，这在编程的时候大大提升了我的效率，程序的可读性也进一步提升。

当我完成课设的时候，我发现自己写了4000多行代码，周围同学一般都是3000多行，我就有点疑虑：会不会是自己的代码风格不够简洁导致的。回头仔细比较才发现虽然我的程序行数多，但是我函数的注释十分完备，并且我在同一个函数不同功能区段处还用了空行将其分隔开来，使得程序逻辑更为清晰。这些良好的习惯都是被上文我说的“不得已而为之”逼出来的，由此可以一窥这次课程设计的功效。

**2. 测试函数的运用**

整个课程设计可以分成不同的模块，但插入链表的模块可以说是最重要的一环，没有它，后面的查找和统计都将失去用途，因此，我在写完了插入链表的模块后写了三个测试函数，SeekProg\_All，SeekDept\_All，SeekTeam\_All（依然在代码文件中有展示），保证我能够顺利完成插入，修改，删除的基本操作后再进入下一步，我认为测试函数是在大规模编程中一个很实用的手段，它可以让我在写了一小段代码的情况下就对我那一小段代码进行测试，这样就可以十分便捷地确定bug的位置。

**3.debug能力的提升**

从课程设计中，我得到的最大收获是课程设计作为一个大项目大大锻炼了我的debug能力。记得网上一篇文章曾指出：迅速发现问题并且定位问题所在是区分普通程序员和优秀程序员的一个重要标志！平时做上机作业遇到bug，我总是会依赖于调试器，一步步跳下去，小程序的bug很容易解决。现在项目大了，没法一步步调试了，我只好逼着自己推断、猜测问题发生点，然后缩小断点范围再去调，一开始判断会出错，调试根本没有效果。但到后来，我对这个项目结构熟悉了以后就可以判断出大部分bug的所在区间，然后去改进程序，我明显感觉到自己这方面水平的提高。

**4.链表操作的再次回味与熟练**

关于链表的操作我上学期末接触了，但自认为不够熟练。不出我所料，这次课程设计的链表操作耗费了我一段时间重新回顾上学期学的内容，才能灵活运用相关知识。本次课程设计插入、查找、删除、遍历、排序这些链表的经典算法都用了至少三次，经过这个过程，我可以说，对于链表的操作我已经十分熟练，能够做到规范地完成链表地设计和相关操作。用熟了关于链表的操作，我发现这些操作并不难，只要把链表操作过程中几个核心步骤熟记于心即可，这一点验证了程序员领域熟能生巧的同样适用。

**5.代码重复性的不足**

这次课程设计让我进一步意识到了自己现有水平的不足，尽管费了很大力气才完成这个系统，链表结构也使用得较为熟练，系统内部处理方面完成得不错。但是输出结果的通用性完成得不太好，代码的重复率太高。代码重复的地方总是出现在需要确定坐标的函数中，每一次输出我都要重新设计输出格式，以便使得各类基本信息有条理地输出，这不符合优秀程序员的标准，大概浪费了三四百行的程序吧。以我现在的水平，要解决这个问题只能通过分割各个函数，传递不同参数。由于确定位置的参数较多，因此就要将原来的函数分割为许多部分，这样会显得代码杂乱无条理，因此我就放弃了这种做法。

**6.自学能力的增强**

本次课设最能体现自学能力的地方就在于界面部分。书上虽然有界面的完整代码，但是要想将它运用，从而完善各个模块的弹出功能还是需要自己对它有一定的认识才行的。我通过翻阅教材后的附录、自己写些小程序验证的方式把每一个结构体的用处都搞明白了。这样才把里面定位，弹出窗口的功能给成功移植到了该用的地方，虽然移植的过程中出现了代码重复性的问题，但是总体效果还是成功的。

**7.学会交流与沟通，严格遵守规范**

写报告本被认为是无趣的一个部分，但从本次报告中，我同样学到了许多。写报告的过程中，我与老师两次探讨了流程图的规范性问题，从中改进了我的流程图，使我的流程图作图变得规范而思路清晰，这是交流的成果。同时重新画流程图的过程也是我表达思路的方式，其实是一种隐性的沟通方式，因为我的思路会通过流程图传递到老师那里。我们不能只有思路，更要学会交流与沟通，才能按照规范严格地把自己的思路清晰地体现出来。这也是作为一个沉默地工科生所应该提升的技能。

**8.程序容错性与健壮性的考虑**

之前完成课堂的小程序的时候没有考虑过容错性和健壮性，所有的程序只要按照样例来输出就完事。但是在实际应用中，这一点是不能不考虑的。我在完成课设以后把自己当作用户去输入一些不同的数据，有些数据并不符合规范要求，以这种方式来检查最后的结果是否有误，这种方式效果良好，每次发现最后的结果有误，我都会在容易输错的地方增家判断语句，一旦用户出错就不执行下面的程序，而是输出提示信息后，重新让用户输入。比如输入项目类别的时候，一旦用户输入1~5之外的项目类别，我就会让系统提示报错。再比如求学生教师人数比的时候，一旦用户教师人数输入为0，那么最后统计时候，比值处就会出现一长串错误数字，最后我就把那一长串数字改成为字符串“教师人数不可为0”，这样可以轻松地让用户理解我的用意。最后，为了让用户时刻不忘记保存，我干脆采用了自动保存的方法，每次插入、修改、删除数据后数据都自动保存。我对自己考虑到那么多的错误，从而进一步完善程序的过程还是很满意的，一方面我的技术进步不少，更重要的是我培养了一种精益求精的精神。

# 七、参考文献

《C语言实验与课程设计》 李开，卢萍，曹计昌编著 科学出版社

《C语言程序设计》曹计昌，卢萍，李开编著. 科学出版社