

第5章

课程设计高级实例

本章通过五个综合实例,展示 C 程序设计的高级应用,通过练习使用函数的声明、定义与调用,要求读者掌握模块化程序设计的基本方法,能够综合利用所学知识,完成复杂的 C 程序设计。

5.1 小型数据库实例 1(通信录)

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 3
#define LEN sizeof(struct friend)
struct friend
{
    int num; /* 编号 */
    char name[8]; /* 姓名 */
    char telephone[13]; /* 电话 */
    char email[20]; /* E-mail 地址 */
    char oicq[12]; /* oicq */
    char address[30]; /* 住址 */
}friends[MAX];

void menu(); /* 功能:主菜单选择函数 */
void addfriend(); /* 功能:添加好友信息 */
void showall(); /* 功能:浏览并显示通信录所有好友信息 */
void delfriend(); /* 功能:根据姓名删除好友信息 */
void editfriend(); /* 功能:根据姓名修改好友信息 */
void menu_search(); /* 功能:查找菜单 */
void searchbyname(); /* 功能:按姓名查找好友信息 */
void searchbytel(); /* 功能:按电话号码查找好友信息 */
int jiemi(char * filename,char * psw); /* 功能:解密文件 */
void check(char filename[20]); /* 检测文件是否需要解密 */
void jiami(char filename[20]); /* 功能:加密文件 */
void fun_quit(); /* 功能:检测退出时是否需要加密文件函数 */
void main() /* 主函数 */
```

```

{ system("cls"); /* 清屏 */
  check("tongxl"); /* 调用检测文件是否需要解密函数 */
  menu(); /* 调用主菜单 */
}

void menu() /* 功能：主菜单选择函数 */
{ char *menu[]={"#^#^#^#^#^#^#通信录#^#^#^#^#^#^#^#",
  " a.添加好友信息",
  " b.删除好友信息",
  " c.修改好友信息",
  " d.查询好友信息",
  " e.显示所有好友信息",
  " f.退出",
  "#^#^#^#^#^#^#^#^#^#^#^#^#^#^#^#", " "};
  /* 定义菜单指针数组 */

  int i;
  char sel;
  int quit=0;
  do
  { system("cls"); /* 清屏 */
    for(i=0;i<9;i++) /* 输出显示菜单 */
      printf("\n%s",menu[i]);
    printf("\n 请输入 a,b,c,d,e 或 f: \n");
    sel=getchar();
    switch(sel)
    { case 'a': addfriend();break;
      case 'b': delfriend();break;
      case 'c': editfriend();break;
      case 'd': menu_search();break;
      case 'e': showall();break;
      case 'f': quit=1;fun_quit();
    }
  }while(!quit);
}

void addfriend() /* 功能：添加好友信息 */
{ int total=0; /* 用于存放已有好友人数 */
  int x;
  FILE *fp;
  int i=0;
  struct friend *p_friend=friends; /* 定义一指针 p_friend 指向数组 friends */
  struct friend friend_add;
  if((fp=fopen("tongxl", "rb"))==NULL) /* 打开文件 */
  { printf("文件有错误不能打开!");
    exit(0);
  }
  while(fread(p_friend, LEN, 1, fp))

```

```

/* 将 fp 所指文件的数据读入指针 p_friend 所指的地址中,并统计好友人数 */
{
    p_friend++;
    total=total+1;
}
fclose(fp); /* 关闭文件 */
if(total==MAX)
{
    printf("\n 通信录已满,无法再增加好友记录!");
    system("pause");
    menu();
}
else
{
    printf("\n=====");
    printf("\n 请输入好友的编号:");
    scanf("%d",&friend_add.num);
    printf("\n 请输入好友的姓名:");
    scanf("%s",friend_add.name);
    printf("\n 请输入好友的电话号码:");
    scanf("%s",friend_add.telephone);
    printf("\n 请输入好友的 email 地址:");
    scanf("%s",friend_add.email);
    printf("\n 请输入好友的 oicq 号码:");
    scanf("%s",friend_add.oicq);
    printf("\n 请输入好友的住址:");
    scanf("%s",friend_add.address);
    if((fp=fopen("tongxl", "ab+"))==NULL)
    {
        printf("文件有错误不能打开!");
        exit(0);
    }
    fseek(fp,LEN,SEEK_END); /* 将文件指针移动到文件末尾 */
    fwrite(&friend_add,LEN,1,fp); /* 将添加的数据写入到 fp 所指的文件中 */
    fclose(fp);
    total=total+1; /* 好友人数加 1 */
    printf("\n 增加好友记录成功!\n");
    printf("\n 是否增加好友记录? 继续请按 1, 否则, 请按 0 返回主菜单");
    scanf("%d",&x);
    while(x!=0&&x!=1)
    {
        printf("\n 提示: 您的输入有误, 请重新输入 0 或 1!");
        scanf("%d",&x);
    }
    if(x==0) menu ();
    else addfriend();
}
}

void showall() /* 功能: 浏览并显示通信录所有好友信息 */
{
    struct friend friend;

```

```

FILE * fp;
int len;
if((fp=fopen("tongxl", "rb"))==NULL) /* 打开文件 */
{ printf("文件有错误不能打开!");
  exit(0);
}
fseek(fp, 0, SEEK_END); /* 将指针 fp 指向文件的末尾 */
len=ftell(fp); /* len 值为 0 表示文件空, 无须解密, 否则需要密码后进入通信录系统 */
if(len==0)
{ printf("\n 通信录为空, 没有信息可显示!");
  printf("\n 按任意键返回主菜单\n");
  getch();
  menu();
}
rewind(fp); /* 将 fp 指向文件头 */
printf("\n 编号 姓名 电话 email iocq 住址 \n");
while(fread(&friend, LEN, 1, fp))
printf("\n%3d%5s%10s%10s%10s", friend.num, friend.name, friend.telephone,
      friend.email, friend.oicq, friend.address);
fclose(fp); /* 关闭文件 */
system("pause");
}

void delfriend() /* 功能: 按姓名删除好友 */
{ struct friend * p_friend=friends;
  struct friend kong={0, ""};
  FILE * fp;
  int total=0;
  int i;
  int flag=0; /* 标记通信录中是否存在要删除的好友, 0 表示不存在, 1 表示存在 */
  char name_del[8]; /* 要删除的好友的姓名 */
  if((fp=fopen("tongxl", "rb"))==NULL)
  { printf("文件有错误不能打开!");
    exit(0);
  }
  while(fread(p_friend, LEN, 1, fp))
  { p_friend++;
    total++;
  }
  fclose(fp);
  p_friend=friends;
  printf("\n 请输入要删除的好友的姓名: ");
  scanf("%s", name_del);
  for(i=0; i<total; i++)
  { if(strcmp(p_friend->name, name_del)==0)
    { p_friend[i]=friends[total-1];

```

```

        friends[total-1]=kong;
        flag=1;
        total=total-1;
        break;
    }
    else p_friend++;
}
if(flag==0)
{ printf("\n 通信录中不存在该姓名的好友!\n");
  system("pause");
  return;
}
if(flag==1)
{ if((fp=fopen("tongxl", "wb+ "))==NULL)
  { printf("文件有错误不能打开!");
    exit(0);
  }
  p_friend=friends;
  for(i=0;i<total;i++)
  { fwrite(p_friend, LEN, 1, fp);
    p_friend=p_friend+1;
  }
  fclose(fp);
  printf("\n 成功删除该好友信息!\n");
  system("pause");
  menu();
}
}

void editfriend() /* 功能：修改好友信息 */
{ struct friend * p_friend=friends;
  struct friend friend_edit; /* 记录修改后的信息 */
  FILE * fp;
  int total=0;
  int i;
  int locate; /* 用于标记要修改信息的好友在文件中的位置 */
  int flag=0; /* 标记通信录中是否存在要修改信息的好友,0表示不存在,1表示存在 */
  char name_edit[8]; /* 要修改信息的好友的姓名 */
  if((fp=fopen("tongxl", "rb+ "))==NULL) /* 以 r+方式打开,从而对文件进行修改 */
  { printf("文件有错误不能打开!");
    exit(0);
  }
  while(fread(p_friend, LEN, 1, fp))
  { p_friend++;
    total++;
  }
}

```

```

    p_friend=friends;          /* 将指针 p_friend 重新定位到数组 friends 的首地址处 */
    printf("\n 请输入要修改信息的好友的姓名: ");
    scanf("%s",name_edit);
    for(i=0;i<total;i++)
    { if(strcmp(p_friend->name,name_edit)==0)
      { locate=i;                /* 标记要修改信息的好友的位置 */
        flag=1;
        break;
      }
      else p_friend++;
    }
    if(flag==0)
    { printf("\n 通讯录中不存在该姓名的好友!\n");
      system("pause");
      menu();
    }
    if(flag==1)
    { printf("\n=====");
      printf("\n 请输入好友的编号(原编号为%d): ",p_friend->num);
      scanf("%d",&friend_edit.num);
      printf("\n 请输入好友的姓名(原姓名为%s): ",p_friend->name);
      scanf("%s",friend_edit.name);
      printf("\n 请输入好友的电话号码(原电话号码为%s): ",p_friend->telephone);
      scanf("%s",friend_edit.telephone);
      printf("\n 请输入好友的 email 地址(原 email 地址为%s): ",p_friend->email);
      scanf("%s",friend_edit.email);
      printf("\n 请输入好友的 oicq 号码(原 oicq 号码为%s): ",p_friend->oicq);
      scanf("%s",friend_edit.oicq);
      printf("\n 请输入好友的住址(原住址为%s): ",p_friend->address);
      scanf("%s",friend_edit.address);
      fseek(fp,locate * LEN,SEEK_SET);    /* 将文件指针定位到文件中原信息的位置 */
      fwrite(&friend_edit,LEN,1,fp);
      fclose(fp);
      printf("\n 成功修改该好友信息!!\n");
      system("pause");
      menu();
    }
}

void menu_search()                /* 功能: 查询菜单 */
{ int sel;
  system("cls");
  { char * menu_search[]={ "#^#^#^#^#^#^#^#^#^#^#^#^#^#^#^#",
    "    1.按姓名查找",
    "    2.按电话号码查找",
    "#^#^#^#^#^#^#^#^#^#^#^#^#^#^#^#",

```

```

        " ";
    };
    int i;
    for(i=0;i<5;i++)
        printf("\n%s",menu_search[i]);
    printf("\n 请输入 1 或 2: \n");
    scanf("%d",&sel);
    while(sel<1||sel>2)
    {   printf("输入有误,请重新输入!\n");
        printf("\n 请输入 1 或 2: \n");
        scanf("%d",&sel);
    }
    switch(sel)
    {   case 1: searchbyname();break;
        case 2: searchbytel();break;
    }
}

void searchbyname()                /* 功能: 按姓名查找好友信息 */
{   struct friend friend_search;    /* 记录查询到的信息 */
    FILE * fp;
    int flag=0;                    /* 标记通信录中是否存在要查询的好友,0 表示不存在,1 表示存在 */
    char name_search[8];            /* 要查询的好友的姓名 */
    if((fp=fopen("tongx1", "rb"))==NULL) /* 以只读方式打开的文件 */
    {   printf("文件有错误不能打开!");
        exit(0);
    }
    printf("\n 请输入要修改查找的好友的姓名: ");
    scanf("%s",name_search);
    while(fread(&friend_search,LEN,1,fp))
    if(strcmp(friend_search.name,name_search)==0)
    {   flag=1;
        break;
    }
    fclose(fp);
    if(flag==0)
    {   printf("\n 通信录中不存在名字为%s 的好友!\n",name_search);
        system("pause");
        menu();
    }
    if(flag==1)
    {   printf("\n=====");
        printf("\n 查找成功!\n");
        printf("姓名为%s 的好友的个人信息如下: \n",name_search);
        printf("\n 编号: %d",friend_search.num);
        printf("\n 电话号码: %s",friend_search.telephone);
    }
}

```

```

        printf("\nemail: %s", friend_search.email);
        printf("\noicq: %s", friend_search.oicq);
        printf("\n 住址: %s\n", friend_search.address);
        system("pause");
        menu();
    }
}

void searchbytel()                                /* 功能: 按电话号码查找好友信息 */
{
    struct friend friend_search;                    /* 记录查询到的信息 */
    FILE * fp;
    int flag=0;                                     /* 标记通信录中是否存在要查询的好友, 0 表示不存在, 1 表示存在 */
    char tel_search[13];                           /* 要查询信息的好友的电话号码 */
    if((fp=fopen("tongxl", "rb"))==NULL) /* 以只读方式打开的文件 */
    {
        printf("文件有错误不能打开!");
        exit(0);
    }
    printf("\n 请输入要修改查找的电话号码: ");
    scanf("%s", tel_search);
    while(fread(&friend_search, LEN, 1, fp))
    if(strcmp(friend_search.telephone, tel_search)==0)
    {
        flag=1;
        break;
    }
    fclose(fp);
    if(flag==0)
    {
        printf("\n 通信录中不存在电话号码为%s的好友!\n", tel_search);
        system("pause");
        menu();
    }
    if(flag==1)
    {
        printf("\n=====");
        printf("\n 查找成功!\n");
        printf("电话号码为%s的好友的个人信息如下: \n", tel_search);
        printf("\n 编号: %d", friend_search.num);
        printf("\n 姓名: %s", friend_search.name);
        printf("\nemail: %s", friend_search.email);
        printf("\noicq: %s", friend_search.oicq);
        printf("\n 住址: %s\n", friend_search.address);
        system("pause");
        menu();
    }
}

void fun_quit()
{
    FILE * fp;
    int len;

```

```

if((fp=fopen("tongxl", "rb"))==NULL) /* 打开文件 */
{ printf("文件有错误不能打开!");
  exit(0);
}
fseek(fp, 0, SEEK_END); /* 将指针 fp 指向文件的末尾 */
len=ftell(fp); /* 值为 0 表示文件为空, 无需解密, 否则需要解密后进入通信录系统 */
if(len==0)
{ printf("\n 文件为空, 无需加密!\n 按任意键退出系统……");
  exit(1);
}
else
{ jiemi("tongxl");
  exit(1);
}
}

void check(char filename[20]) /* 检测文件是否需要解密 */
{ FILE * fp, * fp_psw;
  int len, i=0, j=0;
  int k=0;
  char ch;
  char psw[9], pswold[9];
  if((fp=fopen(filename, "rb"))==NULL) /* 以只读方式打开的文件 */
  { printf("文件有错误不能打开!");
    exit(0);
  }
  fseek(fp, 0, SEEK_END); /* 将指针 fp 指向文件的末尾 */
  len=ftell(fp); /* 0 表示文件为空, 无需解密, 否则需要解密后进入通信录系统 */
  if(len>0)
  {
    if((fp_psw=fopen("password", "rb"))==NULL) /* 以只读方式打开密码文件 */
    { printf("密码文件有错误不能打开!");
      exit(0);
    }
    ch=fgetc(fp_psw);
    while(!feof(fp_psw)) /* 从密码文件中读出原来原加密文件密码到 pswold 中 */
    { pswold[i++]=ch;
      ch=fgetc(fp_psw);
    }
    pswold[i]='\0';
    printf("\n 文件已加密, 需解密!\n 请输入密码: ");
    scanf("%s", psw);
    if(strcmp(psw, pswold)==0)
    { k=jiemi("tongxl", psw); /* 调用解密函数 */
      if(k==1)
        printf("\n 文件解密完成!\n 按任意键进入通信录系统!\n");
    }
  }
}

```

```

        else
        { printf("\n 文件解密没有完成!\n 按任意键退出通信录系统!\n");
          getch();
          exit(0);
        }
    }
    else
    { printf("\n 密码错误,无法继续操作,即将退出系统...\n");
      system("pause");
      exit(0);
    }
}

else
printf("\n 原文件为空,无须解密!\n");
printf("\n 按任意键进入通信录!\n");
getch();
/* 进入菜单调用 */
}

int jiemi(char * filename, char * psw)      /* 功能: 解密函数 */
{ FILE * fp1, * fp2;
  int len=0;
  int j=0;
  char ch;
  if((fp1=fopen(filename, "rb"))==NULL) /* 以只读方式打开的文件 */
  { printf("文件有错误不能打开!");
    return 0;
  }
  if((fp2=fopen("new", "wb"))==NULL)
  { printf("文件有错误不能打开!");
    return 0;
  }
  ch=getc(fp1);
  while(!feof(fp1)) /* 将 fp1 中数据与密码异或后写入 fp2 文件中 */
  { fputc(ch^psw[j]>=len? j=0: j++], fp2);
    ch=getc(fp1);
  }
  fcloseall(); /* 关闭已打开的所有文件 */
  remove(filename); /* 删除加密前的文件 */
  rename("new", filename); /* 将文件"new"重命名为解密前的文件名 */
  return 1;
}

void jiami(char * filename)                /* 功能: 加密函数 */
{ char new_psw[9], ch;
  FILE * fp_psw, * fp;
  int i=0, j;

```