C语言程序设计 课程设计报告

221220144 张路远

**一、选题与组队**

本次课程设计选择设计并制作一个简单的通讯录系统。选择通讯录系统是因为其结构和功能相对比较明确，涉及的知识点较为全面，可以巩固学到的知识。

本次课程设计由本人独立完成，未组队。

**二、程序功能与设计理念**

通过参考手机通讯录系统，确定以下几个基本功能：

1、注册新用户

2、展示我的名片

3、修改我的名片

4、新建联系人

5、修改联系人

6、查找联系人

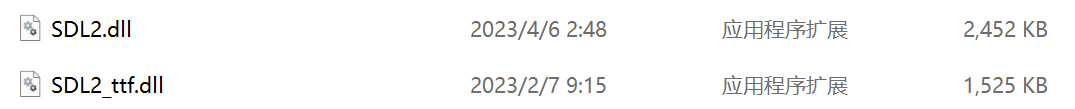
7、删除联系人

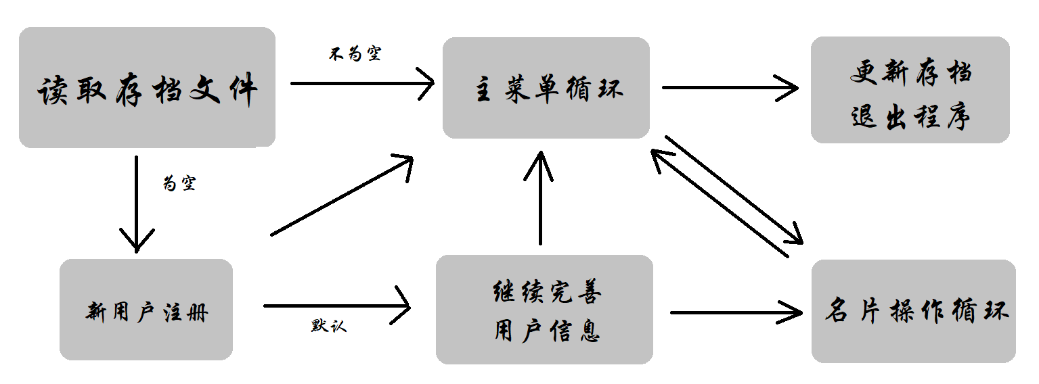
为扩大知识运用范围，额外设计两个偏娱乐性的功能：

8、访问一位随机联系人

9、“听猫叫”（播放音频）

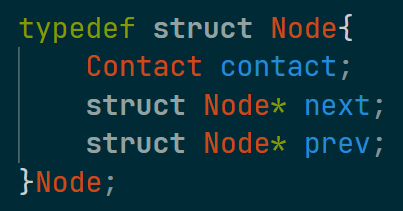
程序主要依托C语言基本库设计后端，处理数据、逻辑和计算等。前端方面，采用SDL2基础库和SDL2\_ttf字体库，不使用SDL\_GUI或其他高级库，训练自己打印文字、图片，设计按钮、窗口等基本能力

链接使用的动态库

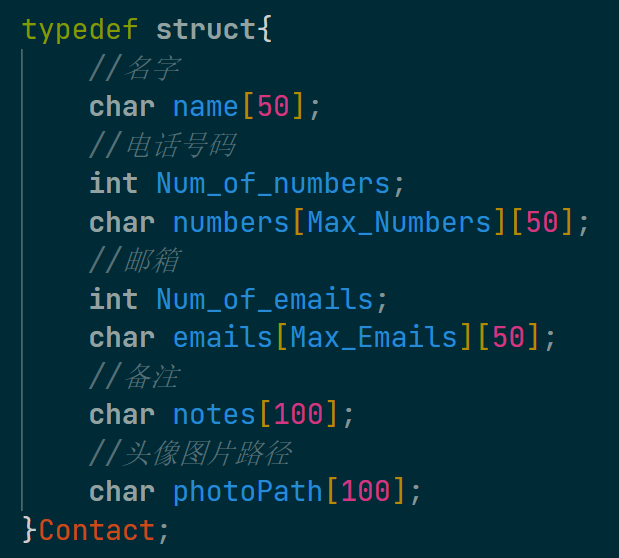
**三、程序框架图**

**四、数据结构**

联系人的保存采用双向链表结构。结构体定义如下：



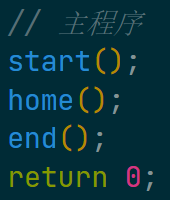
联系人数据结构：



其中，头节点作为全局变量命名为head，用于储存用户的信息

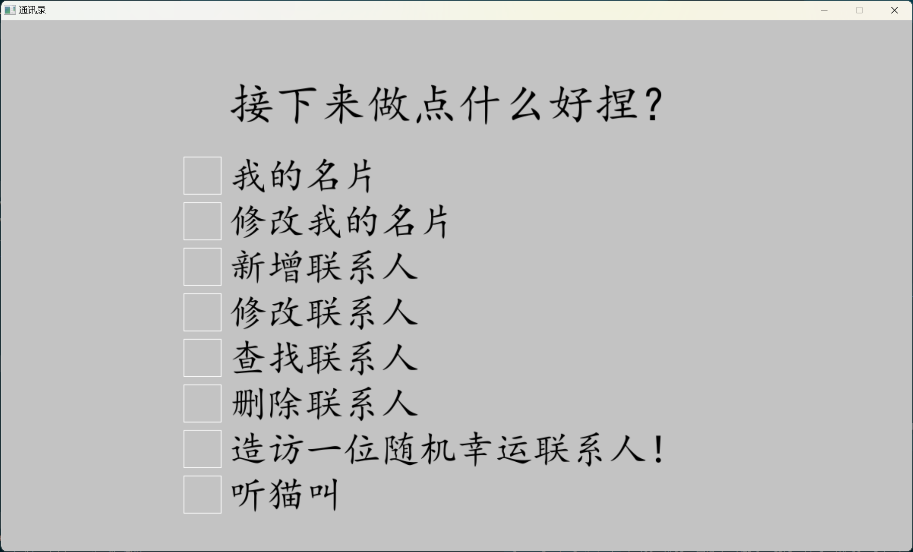
**五、主要模块的功能**

通过封装，除去图形库等的初始化工作外，main函数主体如下：



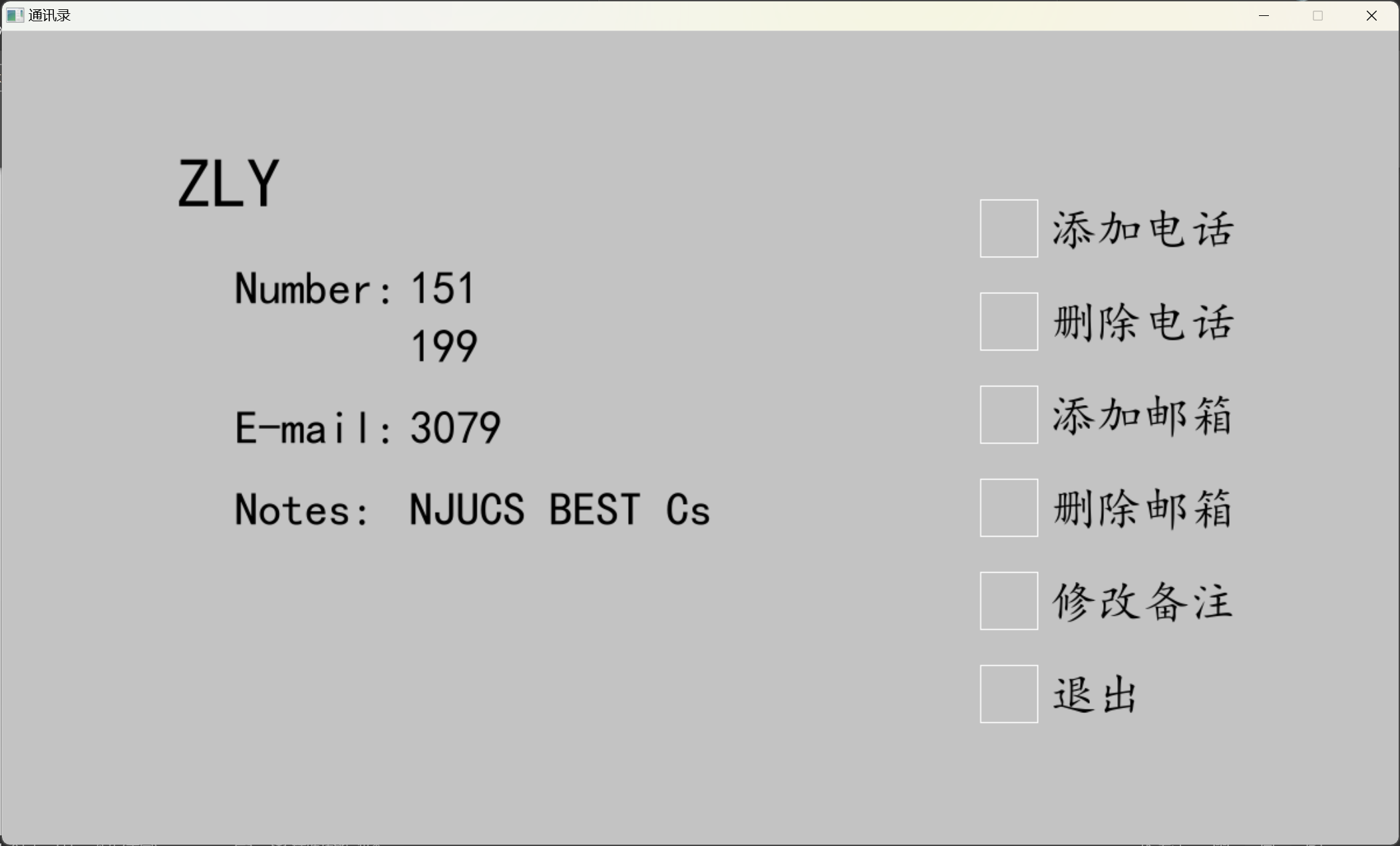
1、start为启动模块。调用后，程序开始从文本文件“contacts.txt”逐行读取联系人信息，并写入双向链表中储存。如果发现文本文件为空，说明用户首次使用系统，将引导用户完成注册，再进入程序主体模块。

2、home为程序主体模块。调用后，进入菜单循环，检测鼠标以确定用户的选项。

 菜单交互界面如图：

其中，“我的名片”，“查找联系人”，“删除联系人”，“造访一位随机幸运联系人！”和“听猫叫”功能为单层功能，对应功能函数被调用后，只留下返回菜单的通路；“修改我的名片”，“新增联系人”和“修改联系人”为二层功能，调用后会进入要处理的联系人的名片操作循环。名片操作循环是菜单循环的子循环和子模块，在退出之前，用户可以反复地进行联系人电话、邮箱的增删和备注的修改。

名片循环如图：



**菜单循环各模块功能介绍如下：**

“我的名片”：打开用户的界面，留下返回菜单的通路。

“修改我的名片”：进入用户的名片操作循环。

“新增联系人”：要求用户输入新联系人的姓名，以此创建一个新联系人。然后，进入该联系人的名片操作循环，引导用户完善新联系人的信息。

“修改联系人”：要求用户输入要修改的联系人的姓名，然后，进入该联系人的名片操作循环，用户可以修改此联系人的信息。如果联系人不存在，提示用户先新建此联系人。

“查找联系人”：要求用户输入要查找的联系人的姓名，然后，打开该联系人的名片。如果联系人不存在，提示用户先新建此联系人。

“删除联系人”：要求用户输入要删除的联系人的姓名。如果用户尝试删除自己，会警告用户。如果用户输入的联系人不存在，会提示用户检查联系人姓名是否正确。正常情况下，用户确认联系人姓名后，程序会再次要求用户确认删除，然后删除该联系人，提示删除成功。

“造访一位随机幸运联系人！”：从用户的所有联系人（包括自己）中随机抽出一位，打开他的名片。“嘿，和你的老伙计打个电话吧！”

“听猫叫”：播放某只“猫”的音频，且留在菜单。

**名片操作循环各模块功能如下：**

“添加电话”：检查该联系人电话数是否超标（标准可由宏定义，默认为3）。若是，警告用户电话数已达最大；若否，允许用户为该联系人添加一个新的电话，然后返回名片操作循环。

“删除电话”：检查该联系人的电话数是否为0。若是，提示一句俏皮话“可怜的家伙一个电话也没有”；若否，列出该用户全部电话，允许用户选择其一删除。删除过程需要用户的二次确认。

“添加邮箱”、“删除邮箱”：同上。（邮箱最大标准默认为2）

“修改备注”：显示用户的备注，且允许用户编辑。用户确认后将备注修改为新的。

“退出”：退出名片操作循环，回到菜单循环。

3、end为结束模块。在home菜单循环中，按下程序窗口右上角的关闭按钮后，菜单循环就会退出，进而调用end（）函数，进入结束模块。该模块会将联系人链表数据写入存档，覆盖原存档；然后清理归申请的、包括SDL2特有的资源；最后销毁渲染器和窗口关闭SDL2，结束程序。

**六、调试过程中出现的主要问题及思考**

在调试过程中出现了不少问题，大概可分为L类（逻辑问题）、F类（前端问题）和T类（txt文件操作问题）。下面列举一些典型的问题：

L类问题：

1、添加新的联系人后，查找不到该联系人；关闭程序后，也没有被写入存档文件。这是因为仅仅新建的联系人节点，忘记写入双向链表中。

2、抽出随机联系人，结果发现总是同一个联系人。这是因为递归逻辑错误，没有正确的把指针后移，而是每次都把指针重定向到第二个节点。

3、按下窗口关闭按钮后，程序没有关闭；多次按下后，程序关闭。这是因为home函数本身具有一层循环，该循环需要检测到关闭按键才会退出。由于其他从其他模块返回菜单的逻辑是直接调用home函数，因此，多次往返后，嵌套了多个home循环。通过在每次调用home后跟上end函数解决了此问题。

F类问题：

1、按SDL2规则设计了在屏幕上打印文字的函数，但是打印出来的中文是乱码。这是因为编码问题，尽管SDL2\_ttf支持UTF-8，但是设计的打印函数调用的基本函数没有选择对应的UTF-8编码适用导致的。

2、程序进入新的函数后，屏幕上出现了不同功能内容同时出现的问题。这是因为部分功能函数没有在被调用时先行清除屏幕导致的。

3、打印指定的文字，但是没有显示。这是因为没有在显示文字后设置延时，导致文字还没看清程序就执行到后面的内容覆盖掉了。

T类问题：

1、在调试时，发现程序启动越来越慢，而且似乎启动等待时间总是2倍增长。这是因为在写档函数尚未验证功能的情况下以外取消注释，导致错误的写档函数每次都把档案内容复制了一边，进而导致启动时间2倍增长。

2、新存的档案总是多出一行换行，导致读档时发生错误。这是写档函数画蛇添足，补充\n导致的。

思考：

1、在写一类函数时，最好先给这一类函数共性的部分做一个模板，以免后面新设计的函数遗忘某些简单但重要的部分，比如首先清除屏幕。

2、应当做好档案的备份操作，否则可能面临重新手敲测试档案的麻烦事。

3、考虑将程序框架独立一个副本出来，当新写函数时，现在副本中验证其功能正常，再放入主程序验证。这样，出现问题后可立即判断是和其他函数发生冲突还是自身问题。

4、由于中文之于编程的特殊性，遇到乱码问题，首先检查同等条件下英文是否乱码；若不是，问题大概率在编码上。

5、函数命名减少重名部分，这样可以利用查找功能更快找到目标函数。

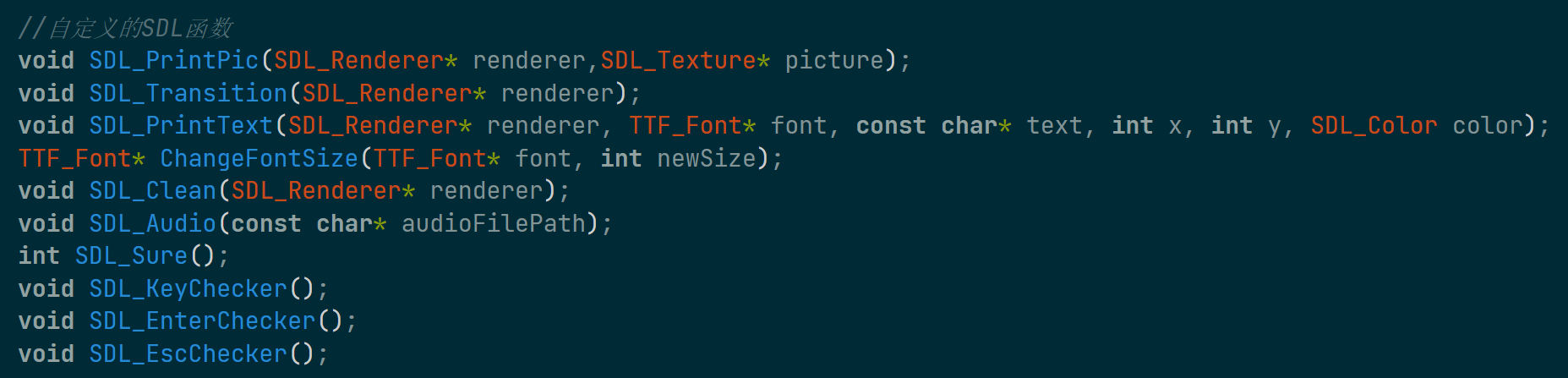
6、未完成的功能先注释掉，且在副本中充分验证功能前最好不要在主工程中解开注释运行。

7、注释并不是越多越好。注释过多，可能导致编辑代码时眼花缭乱。作为单人项目，注释保留必要的精简内容即可。

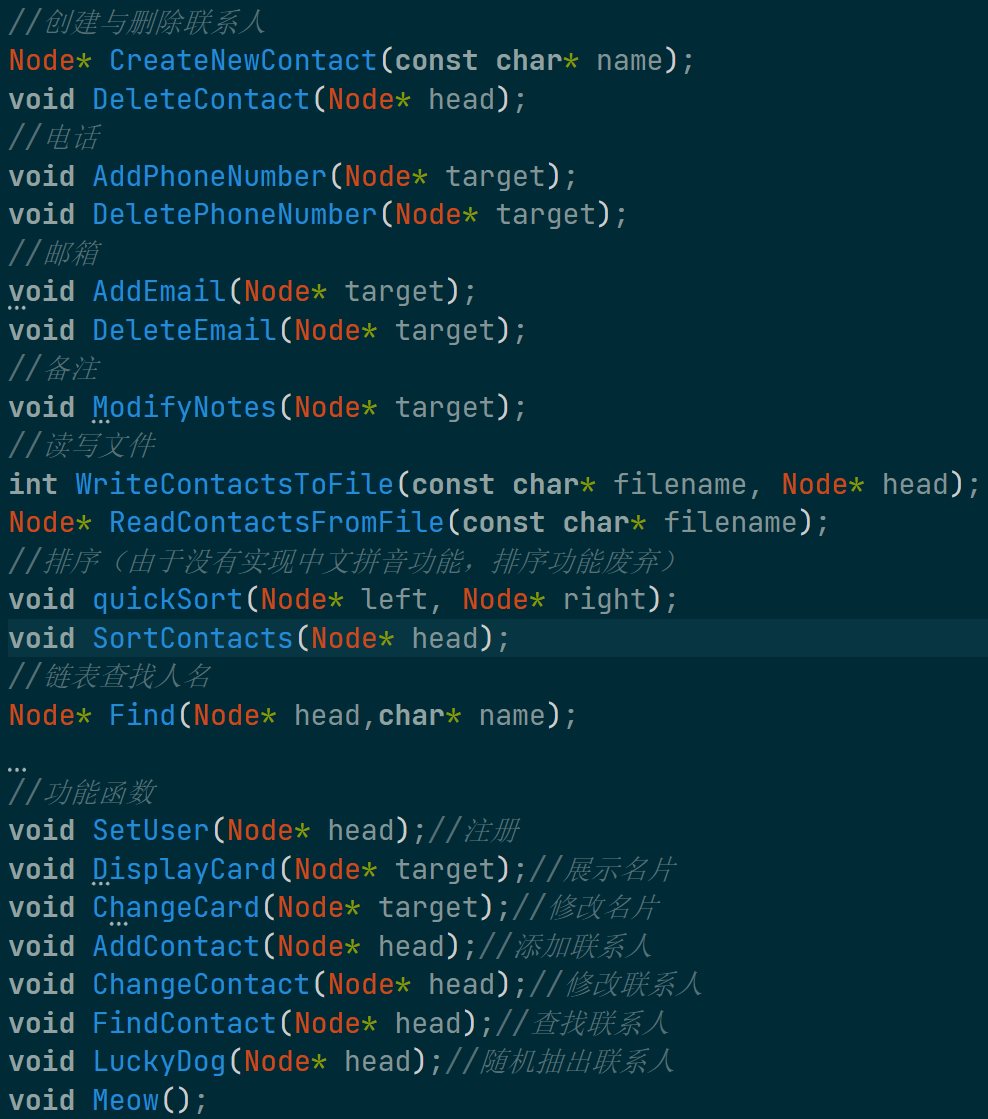
**七、学习体会与收获**

本次课程设计，主要目的在于用实践检查自己所学知识。本次单人项目，合计完成了1832行的代码量，涉及到C语言程序设计课堂上讲到的大量知识点。

由于加入了前端任务，且为了锻炼能力，没有选择使用高级库，课程设计任务量大大增加，但也充分训练了基本功，包括利用SDL2的基础图形库，写了一大批可复用的自定义SDL函数。这些函数或许在高级库中有现成的、更可靠的选择，但这个设计过程涉及到的封装理念：函数的抽象、接口的设计等经验是有价值的。例如，在指定位置打印文字，我采用了先行把文字转化为纹理，然后创建对应的位置矩形，最后复用设计的指定位置打印纹理的函数实现文字打印功能。



程序设计模块化也是一个重要的技巧。通过不断封装实现好的功能，不断简化程序整体，使得新功能的实现逐渐变得像搭建积木一样简单，快速推进进度，避免重复造轮子。



本次设计也有两个遗憾：一是因为尚未掌握拼音库的使用，无法把中文人名转化为拼音进行字典序排序，查找联系人不得不采用遍历算法，写好的排序函数也因此没有发挥上作用(当然，如果用户储存的联系人全是英文名则另说)；二是由于Win11更新，用户在输入中文时无法看到输入法的候选词，导致多音字输入变成了盲选，最后只能依靠退格功能弥补缺陷。

**八、完整的源程序清单**

1、C语言标准库：

stdio.c

stdlib.c

stdbool.c

string.c

time.c

2、SDL2标准库

SDL2.c

SDL2.dll(动态库)

3、SDL2字体库

SDL2\_ttf.c

SDL2\_ttf.dll(动态库)