华南师范大学本科生实验报告

姓名____蓝昊荣_____ 学号 20202121002__

院系__计算机学院_____ 专业 计算机科学与技术师范

年级__2020_____ 班级__一班__

实验时间 2022__年 9__月 27__日

实验名称_____编译原理实验 1__

课程名称_____编译原理_____

华南师范大学教务处编印

华南师范大学本科生实验报告

实验课程：编译原理

实验名称：编译原理实验 1（C++源程序的压缩和解压）

一、实验内容

（一）压缩器

1、为了提高 C++源程序的可读性，C++程序在书写过程中加入了空行、空格、缩进、注释等。假设你想牺牲可读性，以节省磁盘空间，那么你可以存贮一个删除了所有不必要空格和注释的 C++源程序的压缩文本。

2、由于 C++源程序是由一些具有特定功能的单词组成的，因此我们可以采用转换为编码的方法来进一步实现减小源文件大小的压缩功能。

为了实现这一效果，我们可以先把 C++源代码中的各类单词（记号）进行拼装分离，并进行编码的转换。

说明： C++语言包含了几种类型的单词（记号）：标识符，关键字，数（包括整数、浮点数），字符串、注释、特殊符号（分界符）和运算符号等【详细的单词类别及单词组成规则见另外的文件说明】。

3、把具有功能 1 和功能 2 的 C++源程序压缩器实现，并得到压缩文本。

（二）解压器

1、能够把通过（一）压缩器压缩得到的 C++源程序压缩文本进行重新解压并还原出除了被删除的所有不必要空格和注释的 C++源程序。

（三）要求应用程序应为 Windows 界面，在一个界面中可以实现压缩与解压两个功能。

该界面应该具备的详细功能有：

1、打开一个 C++源文件，并可以浏览该源程序。

2、压缩所打开的 C++源程序。

3、查看或浏览得到的压缩文本。

4、打开一个 C++源程序的压缩文本，并可以浏览该压缩文本。

5、解压该压缩文本，并可以浏览所解压出来的源程序。

（四）应该书写完善的软件设计文档。

二、实验目的

1、掌握手工生成词法分析器的方法，了解词法分析器的内部工作原理。

2、通过设计、调试程序，加深对词法分析原理的理解，并掌握在对程序设计语言源程序进行扫描过程中，将其分解成各类单词的词法分析方法。

3、掌握使用自定义编码规则的，对 C/C++源程序中出现频率较高的单词，如关键字等进行压缩，并解压还原。

4、掌握自己编写程序实现将源程序中的注释、不必要的换行、缩进删除，以降低存储文件所需的内存空间。

5、熟悉 MFC 中常用控件，比如 Button、Edit Control、Static Text 的使用，并尝试用 MFC 为 C++源程序压缩程序设计 windows 界面。

6、了解 MFC 生成桌面应用安装包的原理及过程。

三、实验文档：

（一）实验思路

1、如何删除代码中所有注释？

首先我们要明确 C++的注释有两种

1、//:一般用于单行注释

2、/* ...*/：一般用于多行注释

对于单行注释//，当我们连续读到两个’/’，可以认为接下来的内容是程序中的注释，那我们只要把本行所有注释内容读取完毕即可，怎么判断本行是否全部读取完毕呢？只要读到换行符，说明要进入到下一行了。

对于多行注释/*...*/，如果我们读到"/*"，说明多行注释开始了，那我们只要一直读入即可，除非我们读到“*/”，我们才停止。

但是需要注意的是，如果读到“，说明接下来的是字符串，此时要特别处理，因为字符串中的//并不是注释，例如 string abc=”123//123”。同理，读到’也要注意这一点。

综上，可以把代码中的所有注释给删除掉。

2、如何删除代码中不必要的缩进和空行呢？

缩进和空行的作用是为了使代码整体框架更清晰，让程序员阅读起来更方便。但实际并不影响代码的编译运行。例如：

```
bool operator<(struct cards a, struct cards b)
{
        return a.length <= b.length;
}
```

return 前面的缩进就可以删除。我们可以将文件一行一行的读取，存储到字符串，然后从头遍历字符串，只要遇到空格或’\t’则删除，如果没遇到则退出遍历。

将一行的缩进删除完之后我们就可以判断该行是否为空行了，注意，如果文件采用 CRLF 的格式存储，意味着行结尾是"\r\n"，如果采用 LF 的格式存储，则行结尾是"\n"。在 linux 环境下使用 getline(ifs,str)函数读取一行数据时，会将一行数据读入，遇到\n 停止，且将\n 抛弃。所以如果文本格式为 CRLF，则最后为"\r\n"，所以即使为空行 str 的长度也不为 0，而是 1；而在 windows 环境下，则不会有这种不一致问题。如果判断 str 的长度为 0，说明是空行，可以删除。

3、如何实现源程序中单词的编码？

首先，要明确的是，我们并非将所有的单词都进行编码，而是将部分常用的单词进行编码，如关键字等。如果盲目追求编码的完整性，压缩的效果反而会下降。因此，我们可以将每一个关键字、符号等均安排一个编号与之对应，这种一一对应的结构，可以借助 STL 中的 map 容器，其底层是红黑树，拥有丰富的接口，使用起来很友好。

源程序中主要有四类单词：

1、 关键字

2、 符号

3、 标识符

4、 常量：数字、字符、字符串

对于前两类单词，我们选择编码压缩再存储，对于后两类单词，我们不进行编码压缩。

4、将压缩器和解压器实现后，如何为我们的程序设计一个 windows 界面呢？

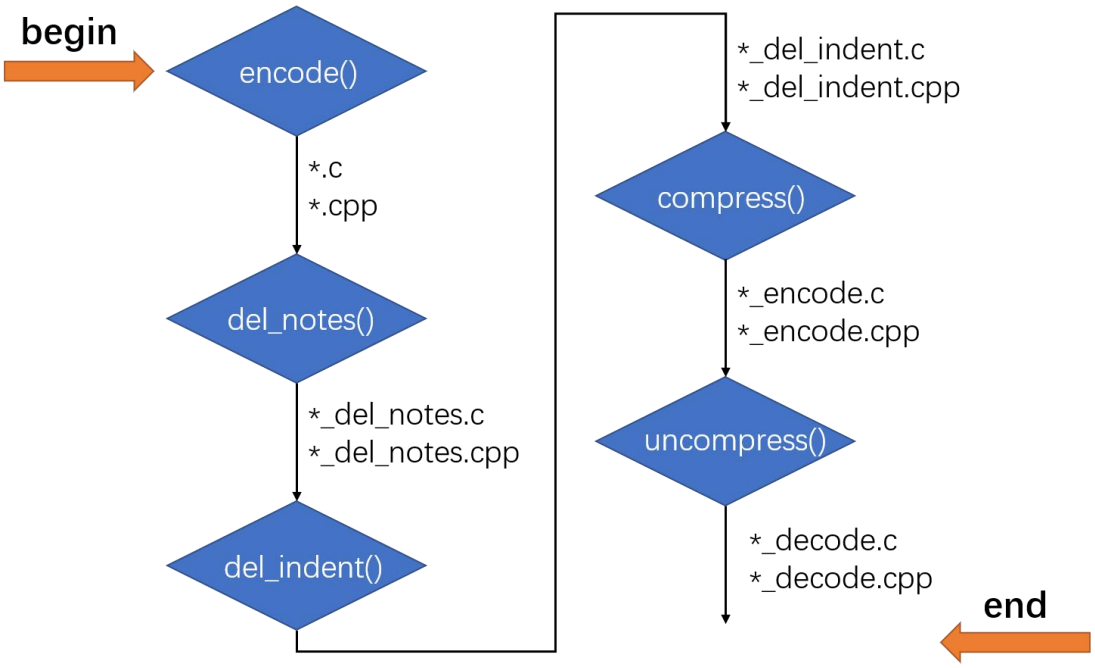
我们可以使用 MFC，创建对话框模板，在对话框中加入一些基础的控件，如按钮、编辑框、静态文本框，然后为控件添加变量、消息处理函数与我们的程序连接起来，从而将我们的压缩器和解压器嵌入。最后打包生成.exe 可执行文件。

（二）、程序结构

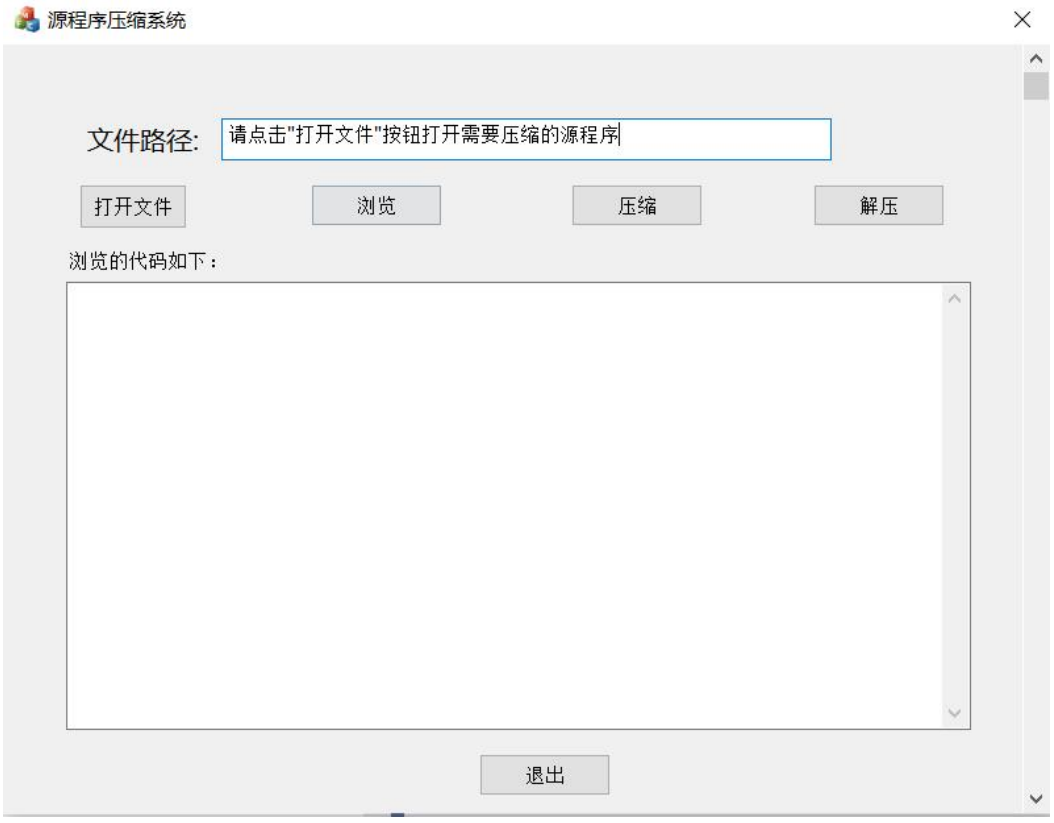
变量名	数据类型	备注
map_	map<string,unsigned char>	单词及编码对应关系的存储结构
num	int	编码的个数
file_path	string	需要解压的源程序的文件路径

函数名	返回值	备注
encode()	void	对指定的单词进行编码
del_notes()	string	将文件中的注释删除，并返回新生成文件的路径
del_indent()	string	将文件中的缩进、空行删除，并返回新生成文件的路径
compress()	string	对文件进行压缩，并返回压缩文件的路径
umcompress()	string	对文件进行解压，并返回解压文件的路径

（三）程序流程图



（四）windows 界面设计

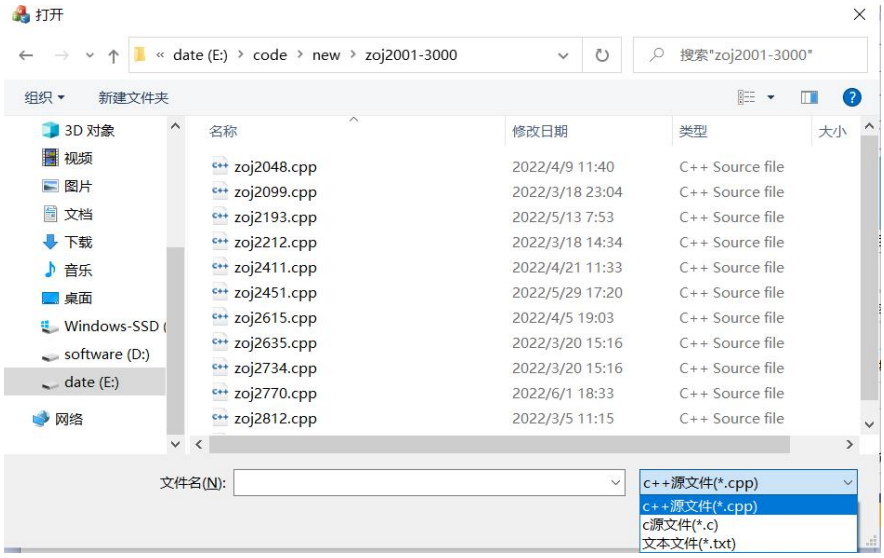


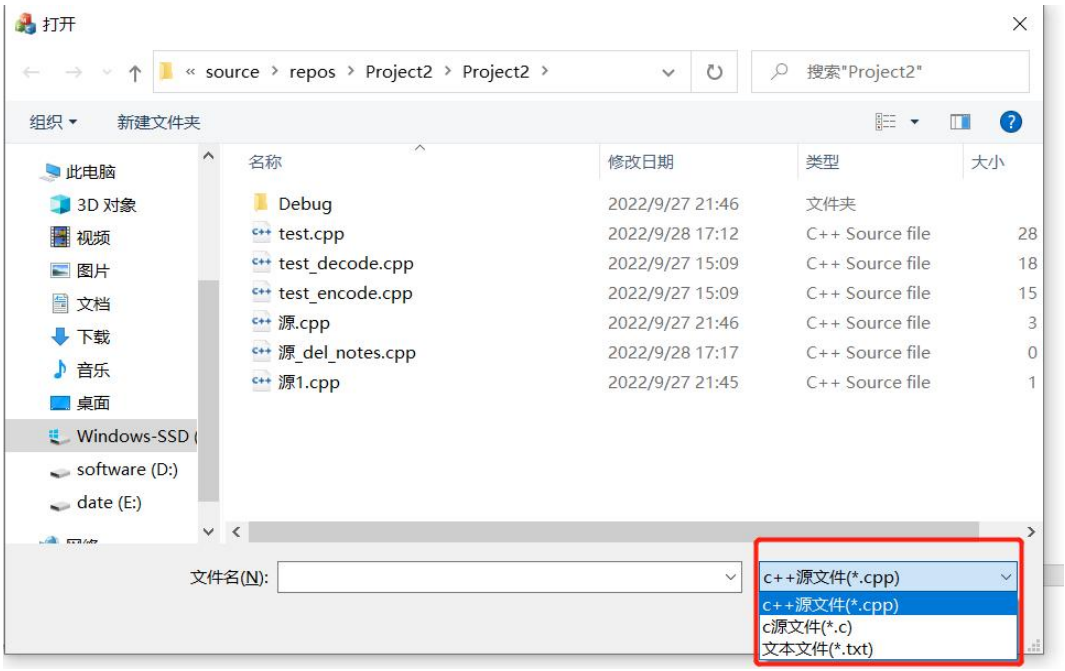
（五）软件使用说明

1、软件名称：C++源程序压缩解压系统.exe，双击软件即可打开运行。软件页面如上图所示。

2、软件有五个按钮，分别为“打开文件”、“浏览”、“压缩”、“解压”、“退出”。

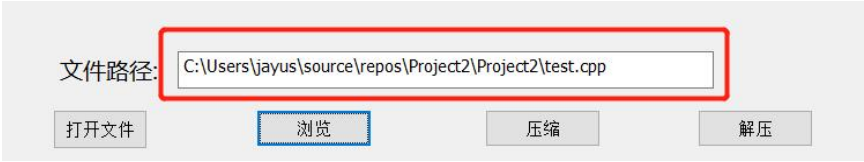
3、点击按钮“打开文件”，会跳转如下界面：



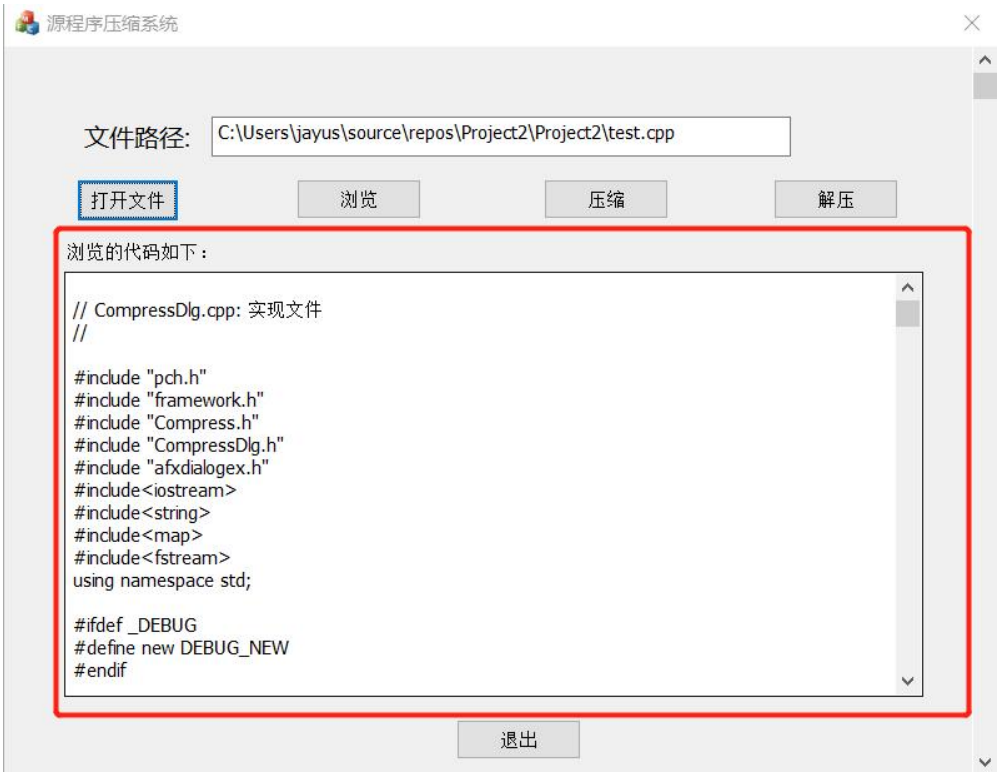


如右下角所示，只支持打开后缀为.cpp/.c/.txt 的文件。

4、选择好文件后，软件会显示出文件路径，如下图所示：



5、点击按钮“浏览”，即可在软件中浏览文件中的内容，如下图所示：

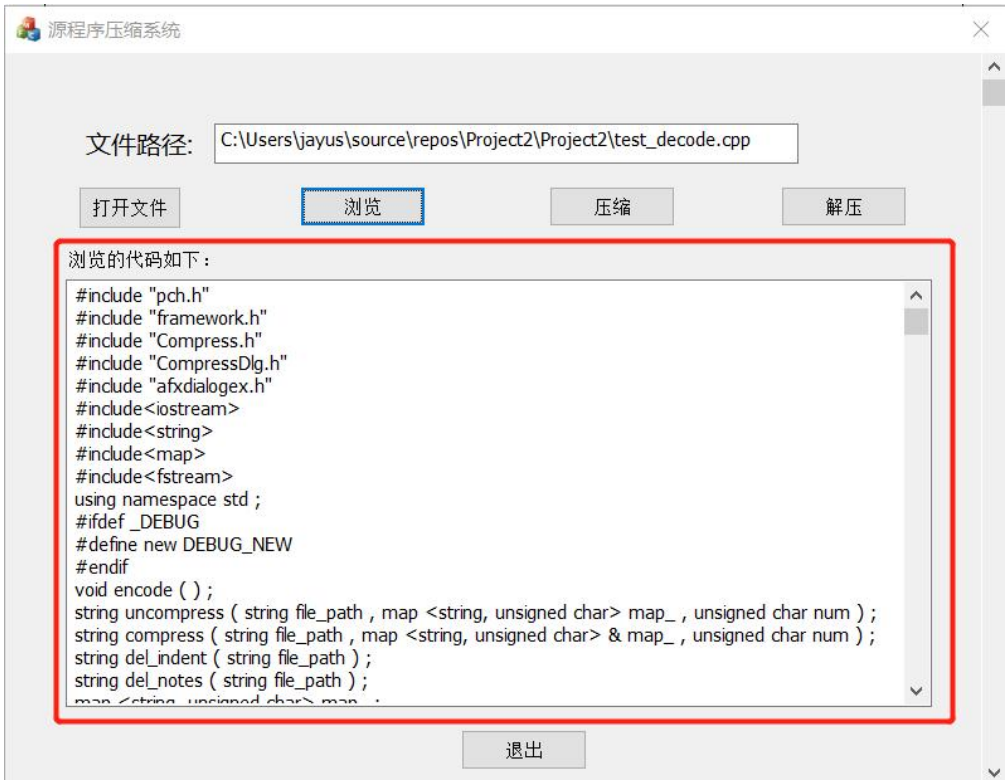


6、点击按钮“压缩”，即可对打开的文件（*.cpp/.c/.txt）进行压缩，压缩成功后会在同目录下生成压缩文件（*_encode.cpp/.c/.txt）。

7、如果想要浏览压缩后的文件内容，可以点击按钮“打开文件”，选中上一步生成的压缩文件（*_encode.cpp/.c/.txt），然后再点击按钮“浏览”。

8、点击按钮“解压”，即可对打开的文件（*_encode.cpp/.c/.txt）进行解压，解压成功后会在同目录下生成解压文件（*_decode.cpp/.c/.txt）。

9、如果想要浏览解压后的文件内容，可以点击按钮“打开文件”，选中上一步生成的解压文件（*_decode.cpp/.c/.txt），然后再点击按钮“浏览”。



10、点击按钮“退出”，即可退出软件。

（五）测试结果

数据	压缩前 文件大小	压缩文件 大小	解压缩后 文件大小
test1.cpp （无注释）	1156 字节	407 字节	1005 字节
test2.cpp (有注释)	31105 字节	16302 字节	19585 字节
test3.cpp (有注释)	13943 字节	6921 字节	8625 字节

（六）实验重难点

1、使用什么存储结构存储单词和编码的对应关系？

单词和编码实际是一一对应的关系，因此我们可以选用的存储结构很多，例如，数组、树形结构等等。但是我们在程序中需要反复查找，所以还要考虑查找的效率。综合以上因素，我决定采用 STL 封装好的 map 容器，map 容器是基于红黑树实现的，因此查找的时间复杂度是 nlogn，同时容器中封装了很多函数接口，方便用户调用，因此是个合适的存储结构。

2、使用什么存储结构存储编码呢？

编码可以从0,1,2,…,往后一直编，一个单词对应一个编码，那我们直观的可以使用int存储我们的编码，然后压缩时，再把编码写进压缩文件中。但是经过测试会发现一个很奇怪的现象，我们经过压缩后的文件的大小会比压缩前要大。经过分析，由于我们压缩文件时主要写入的是编码，而这个编码是用int存储的，一个int数据占用4B，但我们考虑一种情况，如果我们需要压缩字符’+’，未压缩前实际只需要1B的空间存储，而压缩后用编码去存储’+’，实际需要4B，反而增大了占用的空间。

那么我们如何解决这种情况呢？实际上，我们可以使用unsigned char数据类型存储编码，该数据类型占用1B，由8位二进制构成，因此可以表示0-255的整数。

3、在删除注释的时候，我们需要注意一种特殊情况，如：“abc//”

删除注释的思路就是，按字符读取文件，如果连续遇到两个’/’，说明接下来的数据是注释内容，但是考虑一种情况：有一个字符串，而字符串中刚好有连续两个’/’，那么字符串后面的内容就会被当成是注释，从而把我们的非注释内容删除掉了。

4、在压缩过程中，按字符读取文件时，可能会读到多余的字符，比方说，标识符的结束读入条件是：(!isalpha(read_c)) && (!isalnum(read_c)) && (read_c != ’_’)，那程序会往后多读一个其余字符再停止，如何处理这个其余的字符呢？

其实我们可以将文件指针进行前移，只要我们的程序多读入了一个其余的字符，我们就跳出循环并且将文件指针前移一位。让下次循环来处理这个字符即可。

四、实验总结

1、在真正编写代码前，我们需要先认真反复地阅读实验要求，弄清楚每一个实验任务，然后重点是要花较长的时间构思我们的实验思路，而不能写一行想一会，如果没有想清楚就动手写代码，我们的代码难免会有考虑不周全的情况，这样我们到时运行可能就会有很多意想不到的错误。与其花时间在运行调试上，不如我们在写代码前就考虑清楚，构建思路，这样一定程度上可以减少我们出错的可能性。

2、在设计 windows 界面的时候，我们可以参考一些 MFC 中类似的实战项目，参考一下别人的设计思路，以及别人是如何组织各控件的关系的，然后再构思我们的设计思路。

3、在遇到问题的时候，我们应该先自己动脑思考，如果暂时想不到可行的解决方法，我们可以先放一下，过一段时间在思考，如果我们在实验过程中实在遇到无法解决的问题，我们应该积极主动寻求外界帮助，在本次实验过程中我遇到了很多问题，通过和同学的交流，思维碰撞能够一起讨论出可行的解决方案，或者可以跟老师交流，询问老师的建议。

五、参考文献：

1、《编译原理》机械工业出版社 第二版

2、unsigned char 类型变量的读写操作

3、CString 与 utf-8 互转及其他类型转换

4、MFC 最详细入门教程

5、Visual Studio 2019 制作 MFC 图标和生成桌面应用安装包【超详细】

六、附录（代码见源程序文件夹）