8轮DES的编程实现

## 算法原理

DES算法是一种按照分组密码工作的对称密码体制。需要加密的明文信息先编为二进制代码，然后对明文按64bit为一组进行加密。有可能最后一组不足位，一般可以随机选取数字填充，但最后8位作为填充指示符。CBC模式下，每个明文组在加密之前先和前一组密文按位模二相加后再进行加密。DES解密算法则为加密算法的逆过程。

算法的入口参数有三个：密钥（64bit）、明文或密文（划分成64bit长的若干组）和处理模式（加密或解密）。CBC模式下，还需要一个初始向量IV。算法的出口参数为相应的明文或密文。

## 整体框架

本程序使用C++语言编写，利用Qt的图形用户界面编辑模式设计了界面。

开发环境为Windows 10, MinGW 32 bit, Qt Creator4.4.1(community)

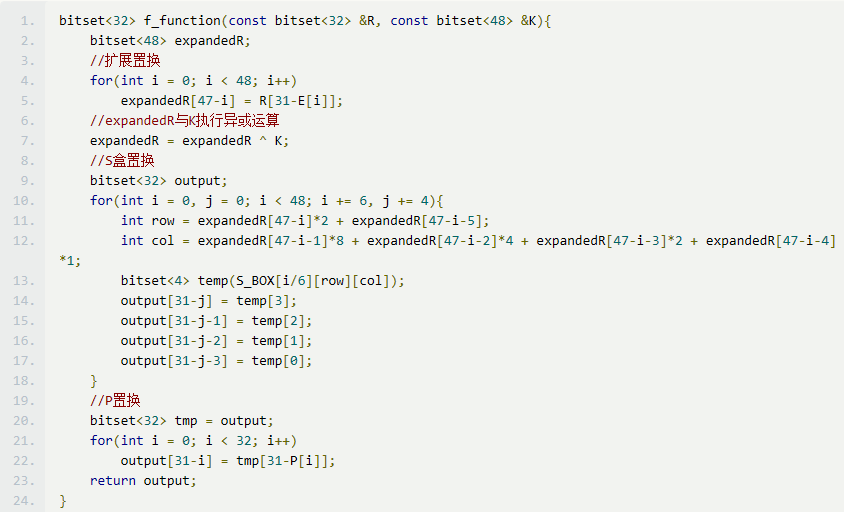
考虑到DES算法的分组加解密模式以bit为单位，选择C++的类<bitset>作为数据结构，节省内存空间，加快运算速度。

**main.cpp**中定义了算法使用的各个置换表和一些全局变量；

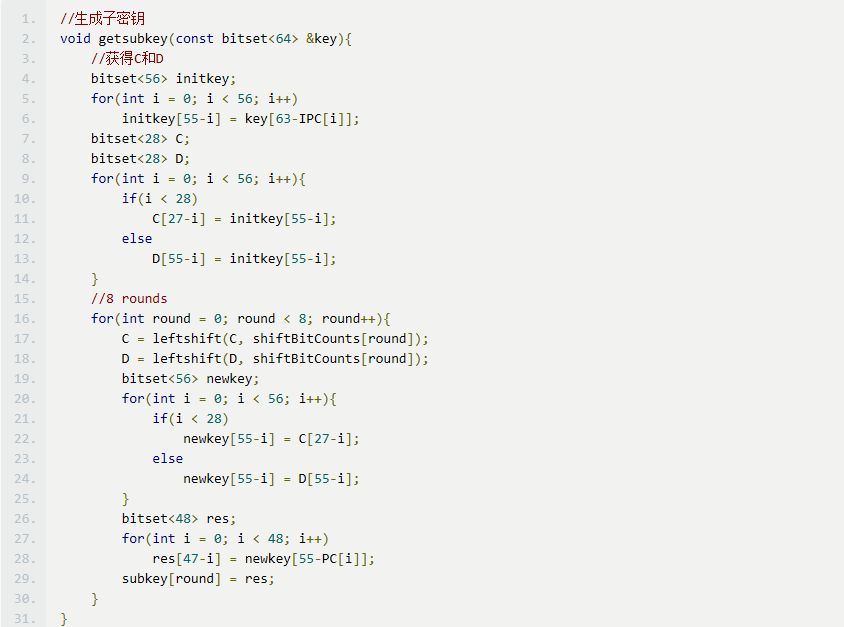
**mainwindow.cpp**中为一些与GUI相关的函数，包括文件输入输出和特定按钮功能的实现；

**des.cpp**实现了算法需要的各个函数。主要的函数有：

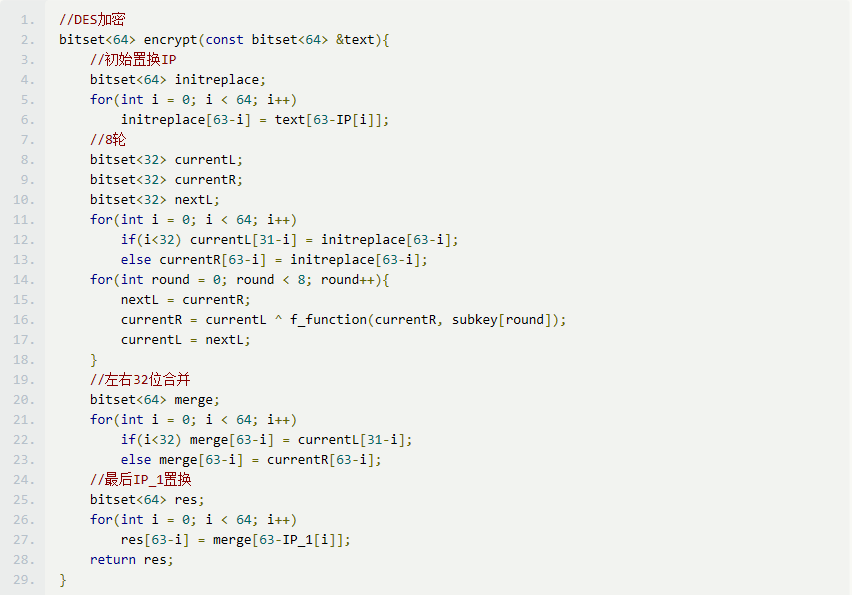
* 轮函数F



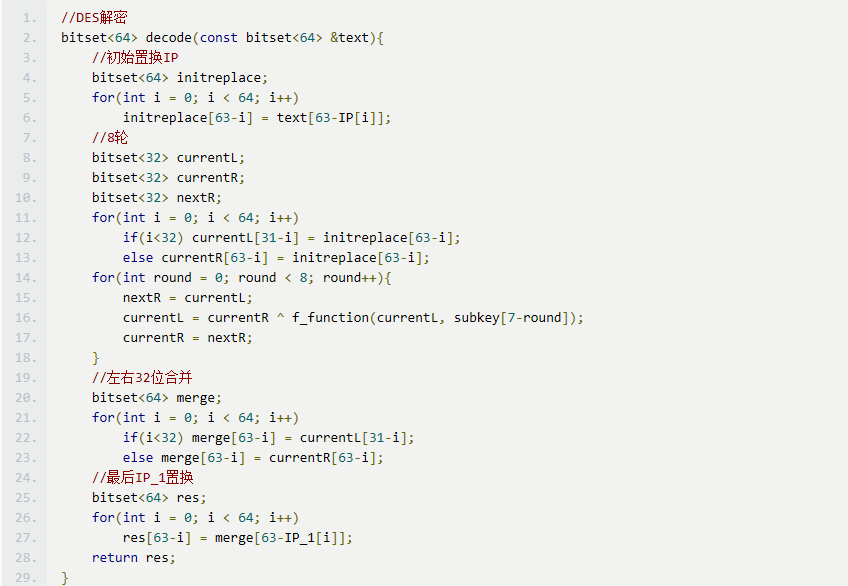
* 生成子密钥



* 加密过程



* 解密过程



* 整体加密函数：

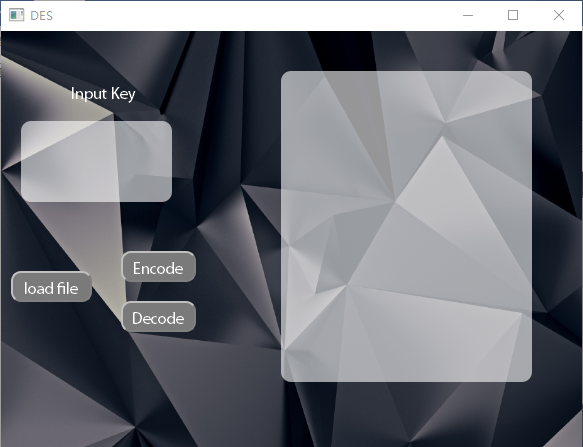


* 整体解密函数

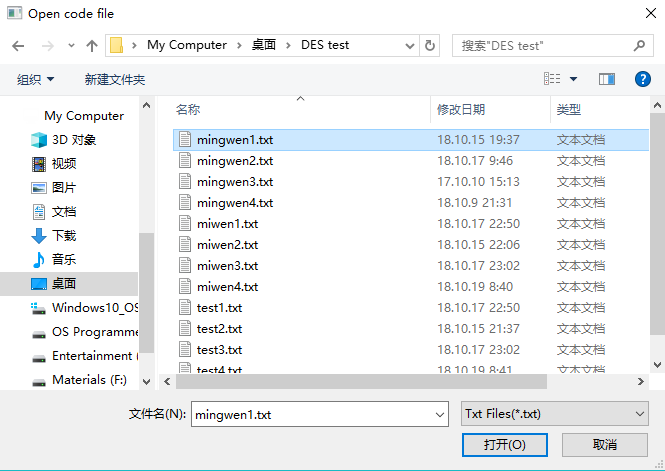


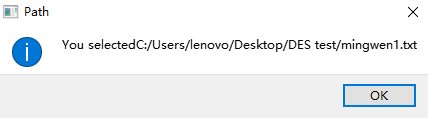
## 操作流程

1. 打开主界面，如图

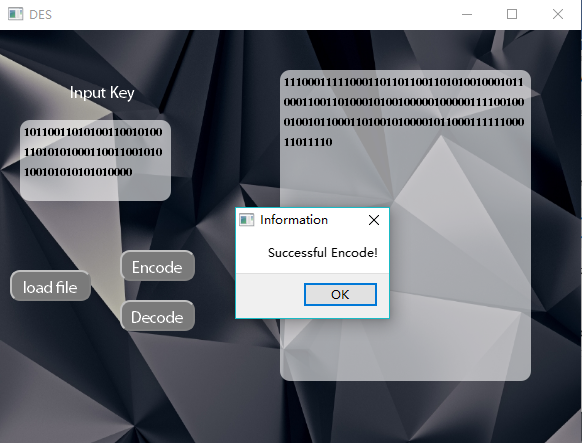


1. 输入Key之后，点击load file读入文件，可在本地任意选择txt文件，确认打开之后弹出提示信息

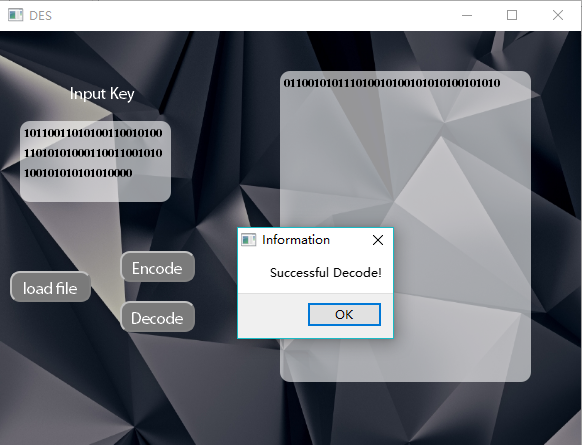




1. 点击Encode，开始加密，加密后的密文以01串的形式显示在右边的框内，选择路径存储对应密文，弹出窗口提示加密完成



1. 解密过程类似，输入对应的密钥，点击load file从本地选择密文文件，再点击Decode开始解密，解密后的明文以01串的形式显示在右边的框内，选择路径存储对应明文，弹出窗口提示解密完成



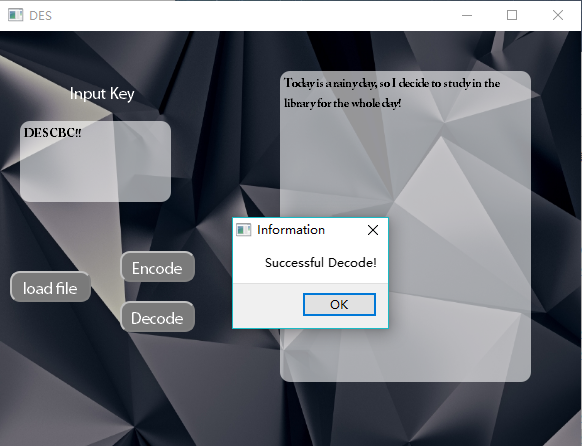
## 进一步思考

由于我们平时需要加密的内容大多数是字符形式而非二进制，故在此程序中额外设计了对英文字符加解密的功能。主要的不同是文件输入、输出时需要先把字符串转成二进制，填充无用字符也需要一些改动。

把mainwindow.cpp中01串版本的push\_encrypt()和push\_decode()注释掉，换成字符串版本的push\_encrypt()和push\_decode()。

由于编制为密文时，若按照每个字节转换成字符，有很多非ASCII字符容易导致显示乱码，故保存密文时仍然按照二进制格式。解密之后再转换为原来的英文字符。

其中使用到的Key是八个字符的字符串，方便记忆。



## 实验结果

本实验主要编制了四个明文文件，其中mingwen1.txt是长度为36的随机01串，mingwen2.txt是长度为60的随机01串，mingwen3.txt是长度为131KB的随机01串，mingwen4.txt是长度为76的英文句子。通过对其分别加密得到密文（miwen1/2/3/4），再对密文重新解密得到明文（test1/2/3/4），将mingwen与test进行比较，若完全相同则说明通过加密实验。其中比较两个01串是否完全相同的函数是：



经过测试，在不同密钥和IV作用下，四个明文均通过了加密实验。实验所用文件在DES test文件夹中。