一、 实践内容

（一）实践的主要内容：

①分别实现 ECB、CBC、CFB、OFB 这四种操作模式的 AES（块长 128 位、密钥长 128 位）。每种操作模式都有一组对应的测试数据，以便检查程序的正确性。其中，CFB 操作模式为 8 位 CFB 操作模式，OFB 操作模式为 8 位 OFB 操作模式。

②以命令行的形式，指定明文文件、密钥文件、初始化向量文件的位置和名称、加密的操作模式以及加密完成后密文文件的位置和名称。

③分别实现对每种操作模式下加密及解密速度的测试，要求在程序中生成 5MB 的随机测试数据，连续加密、解密 20 次，记录并报告每种模式的加密和解密的总时间和速度。

（二）相关原理：

①AES 是一种明文分组 128bit,密文长度 128bit,密钥长度可变(128/192/256 等，这里采用 128 位）的分组密码算法。它是面向二进制的密码算法；不是对合运算，加解密算法不同；综合运用多种密码技术（置换、代替、代数）；整体结构是 SP 结构，基本轮函数迭代，迭代轮数可变（>=10）。每一轮由三层构成：非线性层（字节替换）、线性混合层（行移位代换、列混淆）、密钥加层（轮密钥加）。

②关于四种运算：

- 字节替换：是 S 盒变换，将状态中的每一个字节非线性地变换为另一个字节，是 AES 唯一的非线性变换。字节替换是可逆的，它由两个可逆变换复合而成。使用 16 个相同的 S 盒，8 位输入 8 位输出。
- 行移位：状态矩阵中的每一行以字节为单位，循环右移不同的位移量——第 0 行不动，第 1 行循环左移 1 个字节，第 2 行循环左移 2 个字节，依此类推。
- 列混合：将一个状态的每一列视为有限域 GF(2^8)上的一个多项式，逐列进行变换——用一个固定的多项式 a(x)乘以每一列所表示的多项式，得到对应的 4 字节向量，再模多项式，替换原列。
- 轮密钥加：将输入阵列和一个轮密钥进行按位异或（模 2 加）运算（轮密钥按顺序取自扩展密钥，而扩展密钥又是由原始工作密钥经过扩展后得到）。

③分组加密应用于大数据加密，就牵涉到了各种不同的模式。在本次课程实践中，要求实现 ECB\CBC\CFB\OFB **四种模式**的应用。

- ECB：将每块明文加密成相应的密码块，最后一块可能需要用一些任意二进制序列填充。相同明文块会被加密成相同密文块。

特点：最简易；相同密钥下明密文一一对应，易暴露明文固有格式；各密文块间缺乏相关性，易受到块替代攻击。本质上是一个“大的单字母替换”。

- CBC：加入反馈机制，当前明文块先与前面的密文块进行异或，再加密。

特点：同一明文块可产生不同密文块；有误码扩散，可自同步；只有当所有比特块到达后才能开始编解码，不能直接用于交互式终端，否则传输带宽浪费严重。

- CFB：按比分组小得多的单位进行加密（使用 128bit 移位寄存器）。将前一个分组的密文加密，再和当前分组的明文进行异或。

移位寄存器的内容与明文的所有历史有关，需要一个初始向量（寄存器初值）；存在误码扩散，可自同步；与 ECB 和 CBC 相比加密效率较低；将分组密码转换为流密码（序列密码），实现即时加密。

- OFB：与 CFB 类似，但是是在块内部进行反馈，通过将明文分组和密码算法的输出进行异或来产生密文分组，反馈机制不依赖明文和密文流。

特点：没有误码扩散；易受到对消息流的篡改；链接相关性，密文与前面明文无关；应用时要求一次一密方式。

二、 实践环境

pc 操作系统: win10

代码编写及编译 IDE：codeblocks

执行:命令行运行编译产生的 exe 文件

编程语言：C 语言

三、 实践过程与步骤

（一） 实现 ECB、CBC、CFB、OFB 这四种操作模式的 AES，用一组对应的测试数据检查程序的正确性：

①ECB：支持输入 9 个参数或 11 个参数（仍旧给出 iv 文件，但实际不会用到）。

输入 9 个参数：

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -m ECB -c AES_Cipher.txt
DB727AC6624F3699CBFC4F0F890832B8A4B1DCA1F52EF8E4CE0FD12E307476C6
Already written to the file.
```

输入 11 个参数：

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m ECB -c AES_Cipher.txt
DB727AC6624F3699CBFC4F0F890832B8A4B1DCA1F52EF8E4CE0FD12E307476C6
Already written to the file.
```

密文也会输出到指定 txt 文件，如下所示：

AES_Cipher.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

DB727AC6624F3699CBFC4F0F890832B8A4B1DCA1F52EF8E4CE0FD12E307476C6

②CBC：输入 11 个参数。

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m CBC -c AES_Cipher.txt
9B0048990511252F5E1088663F8CB038A21952EAC6D2C27546369FCA0136BF04
Already written to the file.
```

AES_Cipher.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

9B0048990511252F5E1088663F8CB038A21952EAC6D2C27546369FCA0136BF04

③CFB：输入 11 个参数。

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m CFB -c AES_Cipher.txt
EAC2DD6334E8FA07FDAB477ABA1628A93AFAAAD753B7E05CD59548A2927BDA97
Already written to the file.
```

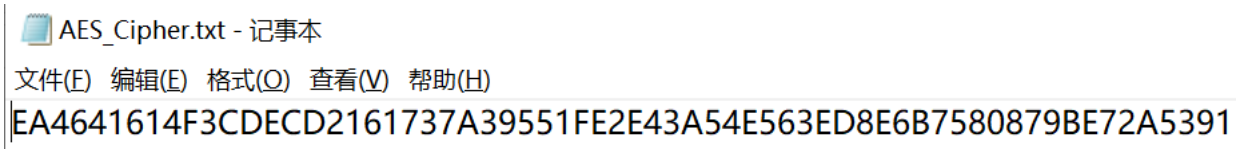
AES_Cipher.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

EAC2DD6334E8FA07FDAB477ABA1628A93AFAAAD753B7E05CD59548A2927BDA97

④OFB：输入 11 个参数。

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m OFB -c AES_Cipher.txt
EA4641614F3CDECD2161737A39551FE2E43A54E563ED8E6B7580879BE72A5391
Already written to the file.
```



（二） 在程序中生成 5MB 的随机测试数据，连续加密、解密 20 次，记录并报告每种模式的加密和解密的总时间和速度：

5MB 的数据即为 5*(10^6)*8bit，由于在程序实现时，一个十六进制字符对应 4bit，因此在程序中需要随机生成的数据（十六进制字符 0x00 到 0x0F）是 10M 个。

在这一部分实践中，代码改动自上一部分即可，增加了加解密模式的自动转换和计时功能。通过程序生成 10M 个字符数据写入明文文件，然后进行加解密，循环 20 次，查看运行时间结果。

①ECB：

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m ECB -c AES_Cipher.txt
ECB 20 rounds Encrypt-Decrypt Speed Test:
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.
20 rounds ALL DONE.
The time spent is: 355295.0000ms

The speed is: 0.5629MByte/s
```

②CBC：

```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m CBC -c AES_Cipher.txt
CBC 20 rounds Encrypt-Decrypt Speed Test:
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.
20 rounds ALL DONE.
The time spent is: 358606.0000ms

The speed is: 0.5577MByte/s
```

③CFB：

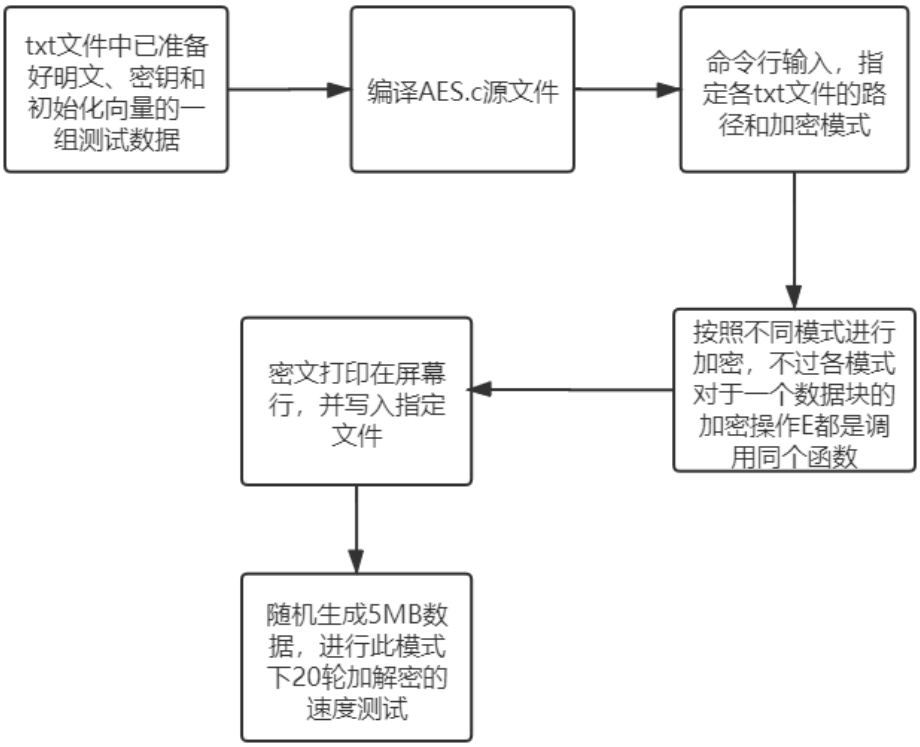
```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m CFB -c AES_Cipher.txt
CFB 20 rounds Encrypt-Decrypt Speed Test:
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.
20 rounds ALL DONE.
The time spent is: 6202629.0000ms

The speed is: 0.0322MByte/s
```

④OFB：

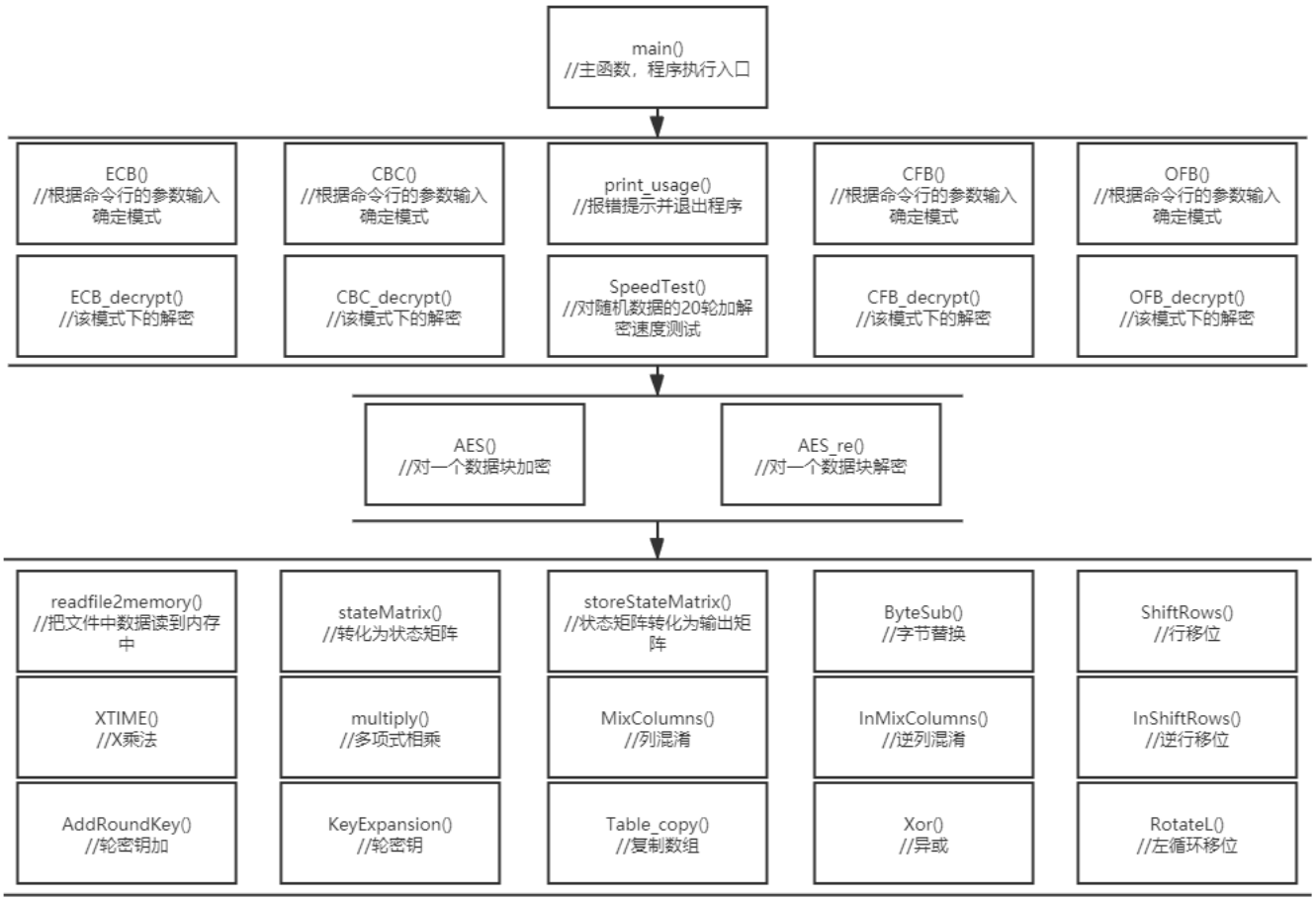
```
D:\A DiskF\coding demo\codeblocksProject\AES>AES.exe -p AES_plain.txt -k
 AES_key.txt -v AES_iv.txt -m OFB -c AES_Cipher.txt
OFB 20 rounds Encrypt-Decrypt Speed Test:
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.
20 rounds ALL DONE.
The time spent is: 6327055.0000ms

The speed is: 0.0316MByte/s
```

四、 程序设计方案

本程序涉及到了文件读写，在同一目录下提前已准备好 AES_plain.txt/AES_key.txt/AES_iv.txt 和 AES_Cipher.txt 四个文件。前三个文件中都是有一组测试数据的，加密结果打印到屏幕上，同时也会写入 AES_Cipher.txt。具体的程序流程为：



此外，在 C 语言设计的程序中，通常都会涉及到众多函数的调用。下面我通过自己定义的函数的调用层次示意图来展现整个程序（不包含库函数，只画出自己写的功能性函数）：



具体代码见 C 源文件 *AES.c*。

五、 实践结果与分析

实验结果截图参见上文第四部分——实践过程与步骤。

对于实践的第一部分，经过验证，对于指定的测试数据，各个模式的加密结果都是正确的，这表明程序编写正确且成功执行了。

对于测试速度的第二部分实践内容，根据结果可见 ECB/CBC 模式的加解密速度相接近，CFB/OFB 的加解密速度相接近；ECB/CBC 的速度大约是 CFB/OFB 的 16~18 倍。这是合理的，因为在本次实践中，AES 的密钥长度是 128 位；CFB 操作模式为 8 位 CFB 操作模式，OFB 操作模式也是 8 位 OFB 操作模式，这意味着明文实际上也是 8 位一组了，分组变小，组数远远增多，加密操作及相关的异或操作次数都变为了 ECB/CBC 模式的 16 倍左右（128/8）。因此，理论情况下，引入反馈的 ECB/CBC 模式的速度会是 CFB/OFB 模式的 16 倍，实际的实践结果基本符合。