MD5算法程序设计与实现

院系：数据科学与计算机学院软件工程系

学号：16341023

姓名：周芙蓉

1. 算法原理概述

MD5消息摘要算法，是一种被广泛使用的密码散列函数，可以产生出一个128为的散列值，用于确保信息传输完整一致。MD5在MD4的基础上增加了“安全-带子（safety-belts）”的概率。虽然MD5比MD4复杂度大一些，单却更为安全。

对MD5算法简要的叙述可以为：

1、数据填充

对消息进行数据填充，使消息的长度对512取模得448，设消息长度为X，即满足X mod 512=448。根据此公式得出需要填充的数据长度。填充方法：在消息后面进行填充，填充第一位为1，其余为0。

1. 添加消息长度

在第一步结果之后再填充上原消息的长度，可用来进行的存储长度为64位。如果消息长度大于264，则只使用其低64位的值，即（消息长度 对 264取模）。

1. 数据处理

准备需要用到的数据：

4个常数：

**A** = 0x67452301, **B** = 0x0EFCDAB89, **C** = 0x98BADCFE, **D** = 0x10325476;

4个函数：

**F(X,Y,Z)**=(X & Y) | ((~X) & Z);

**G(X,Y,Z)**=(X & Z) | (Y & (~Z));

**H(X,Y,Z)**=X ^ Y ^ Z;

**I(X,Y,Z)**=Y ^ (X | (~Z));

把消息分以512位为一分组进行处理，每一个分组进行4轮变换，以上面所说4个常数为起始变量进行计算，重新输出4个变量，以这4个变量再进行下一分组的运算，如果已经是最后一个分组，则这4个变量为最后的结果，即MD5值。

MD5以512位分组来处理输入的信息，且每一分组又被划分为16个32位子分组，经过了一系列的处理后，算法的输出由四个32位分组组成，将这四个32位分组级联后将生成一个128位散列值。

总体流程如下图所示， 表示第i个分组，每次的运算都由前一轮的128位结果值和第i块512bit值进行运算。

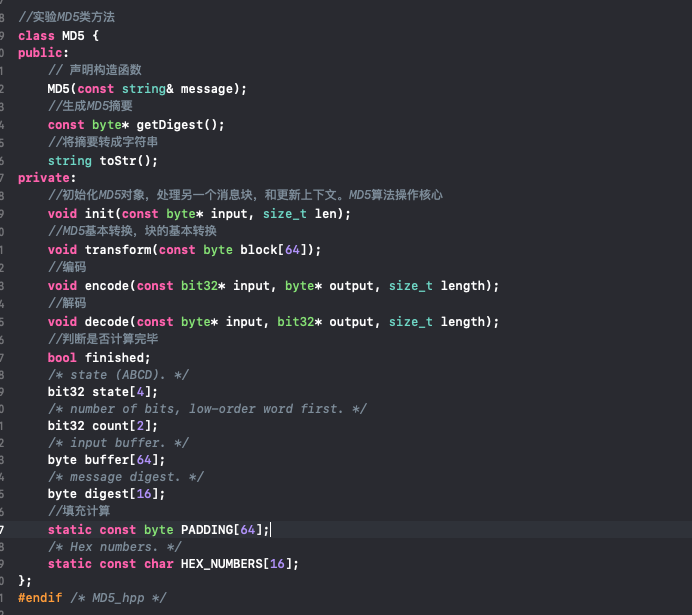


1. 总体结构

main.cpp：测试MD5算法有效性文件，来源于RCF 1321

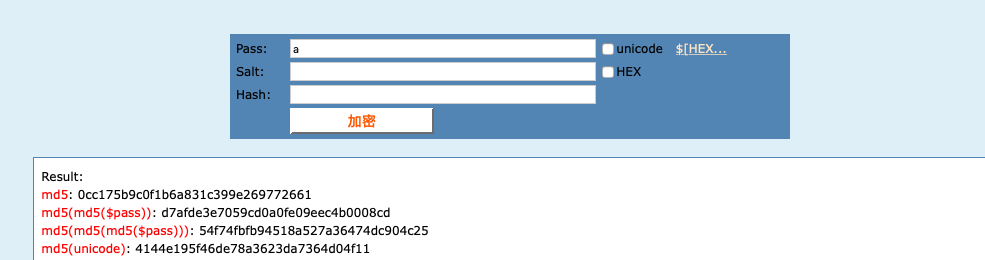
MD5.h：定义了MD5类以及宏定义了一些计算方法

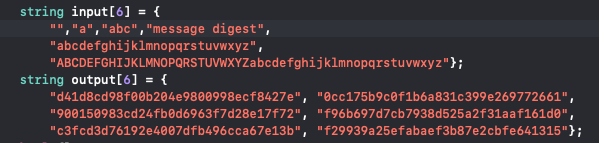
MD5.cpp：实现了MD5中的类方法



1. 模块分解
2. 测试模块

在主函数中，先对程序进行了正确性的测试。首先通过在线MD5加密工具获取一些标准加密结果，再与程序运行结果进行比较，就可以判断程序运行是否正确。





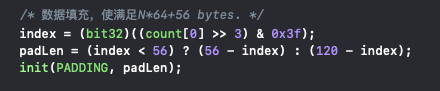
1. 初始化模块

由于在初始化模块中需要多次建立新的对象，所以我们需要对上一次的数据和运算结果进行清除。初始化模块需要做的就是将上一次过程中的运算结果进行清除，保证下一次运算结果的准确性。



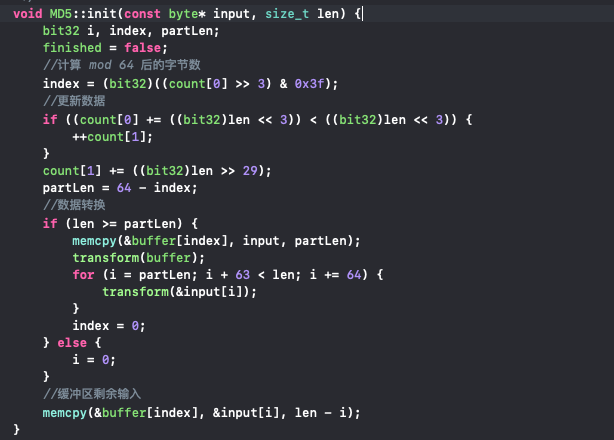
1. 数据填充模块

MD5算法要求输入信息的长度对512求余的结果等于448，如果不满足该要求需要对输入的信息进行部位。部位的原则是填充一个1和n个0，使得信息的长度满足N\*512+448的要求。



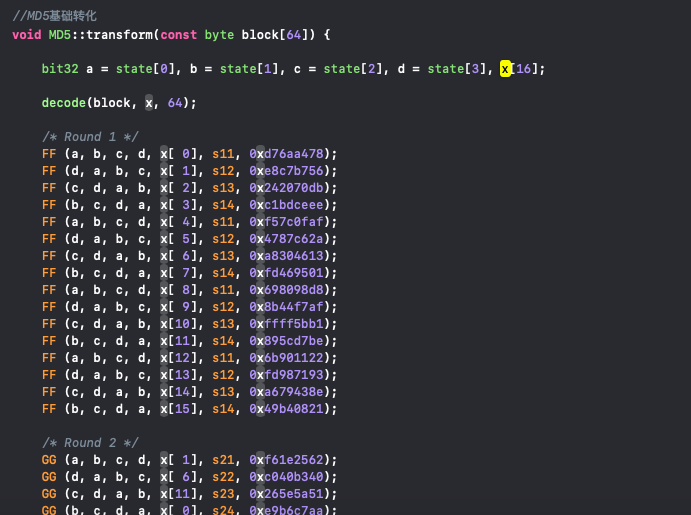
1. append length模块

在 append Length 模块中，对于原始信息的位数长度b，化成二进制表示后 选取低64位，每个word按照小端规则添加到第一步处理后的消息数据的尾部。



1. transform模块

在 transform 模块中，使用之前的宏定义完成四轮轮转运算，随后在末尾分别再加上原始数据，即可得到初步加密的运算结果A B C D 。



1. 输出模块

在最后输出阶段，对加密完成的数据进行格式规范与处理，将数据从小端存储转移到大端存储。

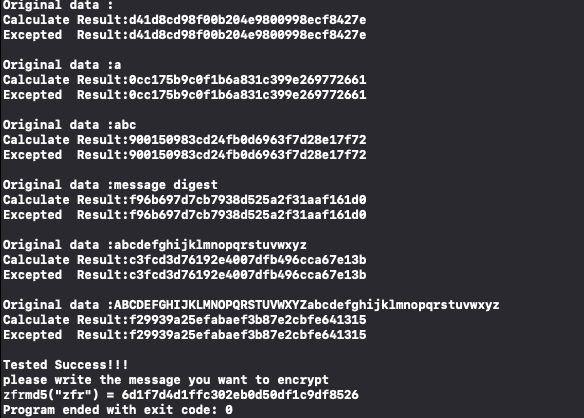


1. 数据结构

为了方便检错和程序的便利地进行多次加密测试，在这里为创建了一个MD5类，用于编写具体实现方法和数据类型。在选用数据类型时选用了bit32和byte，一方面可以保证程序地严整性，一方面在进行数据初始化的过程中操作会比较便利。Main函数提供了程序正确性检验，所用数据均为在线加密后的MD5数据，用于判断程序是否满足要求。同时提供了加密操作，用于用户输入消息，进行MD5加密。



1. 编译运行结果



正确性检测，与标准输入一致，程序通过