多媒体技术基础 PROJECT-3

本实验中，我们对 JPEG 图像的编码标准和实现进行学习，并完成一个 24 位 BMP 转 JPEG 的程序。

实验目标

本次实验，需要大家对其中的几个缺失部分进行填充，以完成一个完整的程序并且成功运行。

代码说明

本程序的源代码包含一下几个文件：

cio.c, cio.h	图像压缩时使用的 IO 接口
rdbmp.c	读取 BMP 图像信息的功能函数
cmarker.c	写入 JPEG 图像信息和标记的功能函数
fdctflt.c	前向 DCT 变换
cjpeg.c, cjpeg.h	图像转换的主题程序，以及所用到的定义和结构等
Makefile	如果你使用 Linux 环境，请将第一行的 MAKE 定义注释掉
test.bmp, test2.bmp	两个测试文件

JPEG 有 4 中操作方式，包括基于 DCT 的顺序模式、基于 DCT 的累进模式、无失真模式和层次模式。这份代码采用的是基于 DCT 的顺序模式。

实验要求

1. 代码部分（60%）

1）cjpeg.c 中的主体编码函数 jepg_encode()中的**转换和压缩的部分**被去掉了，请完成它并且成功运行。对此需要了解编码的操作顺序。（15%）

2）jpeg_encode() 函数会调用 jpeg_compress()函数对数据进行压缩，并调用 write_bits()写入压缩后的数据。同时，在完成全部的数据写入后，write_align_bits()函数对尾部进行填充。请完成 **write_bits()**和 **write_align_bits()**两个函数。其中，write_bits()函数将待写入数据的值去除开始的’0’位，并在一起，然后以字节为单位写入，写入前先判断是否为 0xFF，如果是，就在该字节后面插入 0x00。write_align_bits()

对数据区末尾不满 1 个字节的数据在后面填充’1’位，凑满 1 个字节，并写入。（15%）

3）对于压缩数据值，因为它可能存在负值，所以将其转换为正值（避免起始’1’位）。请完成 set_bits()函数。它对特定值求它的绝对值的位长度，同时如果原值为负值$(-2^x,-2^{x-1}]$，将其加上2^x，映射到$[0,2^{x-1})$。（15%）

4）正确完成以上修改之后，就可以编译并转换图片了，但是会发现转换出来的图像是上下相反的，请改正代码，使之正确完成转换工作。（15%）

2. 文档部分（40%）

完成一份详细的文档，要包含以下内容：

1）JPEG 编码过程的详细说明，要求包含操作流程（20%）

2）Huffman 编码的原理（10%）

3）DCT 变换的原理（10%）

3. 重要说明

要求使用 gcc 编译

参考资料

jpegsrc.v7.tar.gz　　IJG,Independent JPEG Group ，一个很好的开源实现

JPEG 标准及其实现.pdf　　JPEG 标准和实现的详细说明

itu-t81.pdf　　JPEG 官方标准

The JPEG Still Picture Compression Standard.pdf　对 JPEG 标准进行了概括

这次的参考资料内容很多，建议前三个只做简单了解，可以详细读一下最后一篇文档。同学们也可以自己再查找一些相关资料做参考。