

# Project 3

## 实验目标

1. 学习 jpeg 图像的编码标准和实现
2. 完成24位 bmp 转 jpeg 的程序

## 实验要求

1. 完成 `cjpeg.c` 中的主题编码函数 `jpeg_encode()` 中的转化和压缩部分 (tips: 完成这部分需要了解编码的操作顺序) ; (15')
2. `jpeg_encode()` 会调用 `jpeg_compress()` 函数, 后者会对数据进行压缩并调用 `write_bits()` 来写入压缩后的数据。同时, 在完成全部的数据写入之后, `write_align_bits()` 函数会对尾部的数据进行填充, 请完成 `write_bits()` 和 `write_align_bits()` 两个函数。 (tips : `write_bits()` 函数将待写入数据值除去开始的 '0' 位, 并在一起, 然后以字节位单位写入, 写入时, 若值为 `0xff`, 则在该字节后面插入 `0x00`, `write_align_bits()` 对数据区末尾不满 1 个字节的数据补充 '1', 凑满一个字节并写入) ; (20')
3. 对于压缩的数据值, 由于可能存在负值, 因此需要将其转换为正值 (避免起始位为 '1')。请完成 `set_bits()` 函数, 它对特定值求它的绝对值的位长度, 若原值为负值则转为正值, 其映射关系如下: (15')

$$f : (-2^x, -2^{x-1}] \mapsto [0, 2^{x-1})$$

4. 正确完成以上任务之后就可以编译转换图片了, 但是有趣的是转换出来的图片是倒的 (???), 请修改代码使得图像显示正确。 (15')

## 文件说明

本程序的源代码包含以下文件:

`cio.c`, `cio.h` 图像压缩使用的 IO 接口

`rdbmp.c` 读取 bmp 图像信息的功能函数

`cmaker.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` 若使用的是 Linux 环境, 请将的第一行的 `MAKE` 定义注释掉

`test.bmp`, `test2.bmp` 两个测试文件

jpeg 有四种操作方式, 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。

## 文档要求

1. jpeg 编码过程的详细说明 (15')
2. Huffman 编码原理 (10')
3. DCT 变换的原理 (10')

## 截止日期

2019.12.15 23:59:59