专题五 文件进阶



要点

- 文件操作基础回顾文件、存储、文件缓冲 系统及其工作原理
- ■理解二进制文件及基本操作: 打开、关闭、 读写函数
- ■有哪些高级应用:二进制文件类型识别、 CAD图形编辑器、BMP图像文件等

2

专题五 文件进阶

- 文件操作基础回顾
 - □文件基本概念
 - □文本文件与二进制文件的区别
 - □缓冲文件系统与文件操作基本原理
 - □文件类型的数据结构
- ■基本二进制文件读写
- ■二进制文件的高级应用
 - □文件类型识别(图片、声音等)
 - □CAD图形编辑器
 - □BMP图像文件

文件的概念

对比:程序运行过程中的数据数据 → 变量 → 内存单元程序结束,内存单元释放

- ■操作系统中的文件: 驻留在外部介质(如磁盘等)中的一个有序数据集
- ■各种类型的文件
 - □程序文件:源文件、目标程序、可执行程序等
 - □数据文件(输入/输出): 文本文件、图像文件、 声音文件、可执行文件等
- ■文件的特点
 - □数据永久保存
 - □数据长度不定
 - 口数据按顺序存取



文本文件和二进制文件的区别

字节 字节 字节

C语言中的文件是数据流(由一个个的字节数据组成) 文件的两种数据形式

- □ ASCII码 (文本文件 text stream) 字符流
- □二进制码 (二进制文件 binary stream) 二进制流
 - 二进制文件是直接把内存数据以二进制形式保存

例如,整数12345

- □文本文件: 49 50 51 52 53 (5个字符'1' '2' '3' '4' '5')
- □二进制文件: 00000000 00000000 00110000 00111001 (4个 字节) 不同整数/浮点数在文本文件中字节数不同,二进制文件表示相同(内存格式)

缓冲文件系统

由于磁盘速度慢 直接把数据写到磁盘效率很低

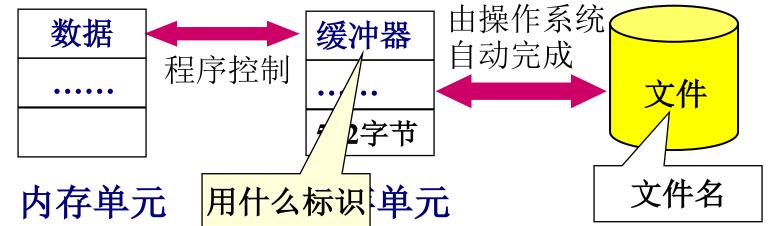


内存单元

内存单元

缓冲文件系统

- ■向磁盘输出数据
 - □数据 → 缓冲区
 - □装满缓冲区后 → 磁盘文件 (由操作系统自动完成)
- ■从磁盘读入数据
 - □先一次性从磁盘文件将一批数据输入到缓冲区
 - □然后再从缓冲区逐个读入数据到变量

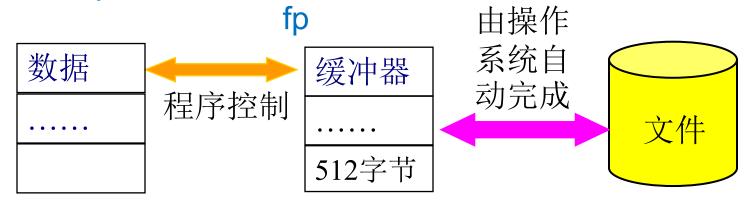


20

缓冲文件与文件类型指针

用文件指针指示文件缓冲区中具体读写的位置

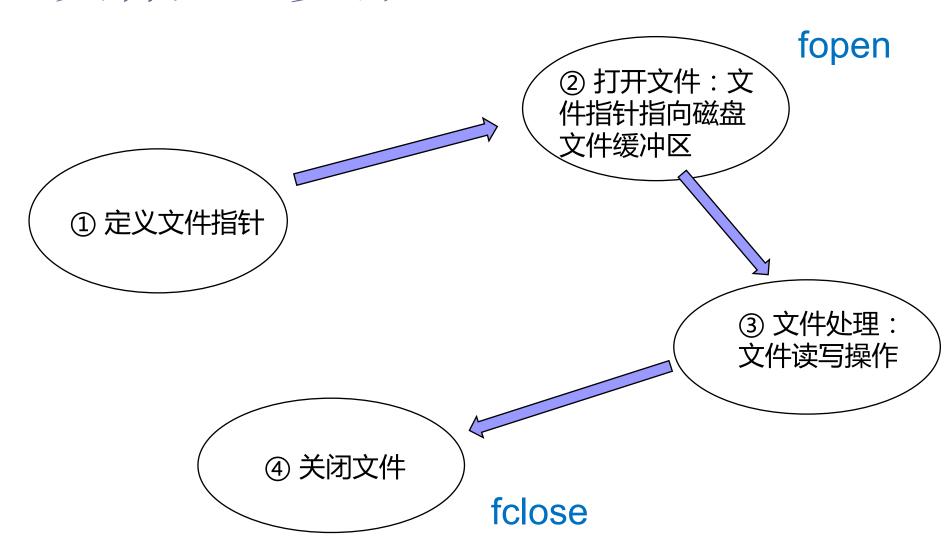
FILE *fp;



fp指向文件缓冲区,通过移动指针实现对文件的操作同时使用多个文件时,每个文件都有缓冲区,用不同的文件指针分别指示



文件处理步骤





二进制文件打开方式

r readw writea appendb binary

- ■文件打开方式参数表
 - \Box fp = fopen("f12-2.bin", "wb");
 - □读、写;文本方式、二进制;添加、创建

文本文	文本文件 (ASCII)		二进	制 文 件(Binary)
使用方式	含义		使用方式	含义
" r "	打开只读文件		" rb "	打开只读文件
66 W 33	建立只写新文件		" wb "	建立只写新文件
" a "	打开添加写文件		" ab "	打开添加写文件
" r+ "	打开读/写文件		" rb+"	打开读/写文件
" w +"	建立读/写新文件		" wb+ "	建立读/写新文件
" a +"	打开读/写文件		" ab+ "	打开读/写文件

a/a+ 打开文件时,文件指针位置位于文件尾

二进制文件读写与打开方式

if 读文件

指定的文件必须存在,否则出错;

if 写文件 (指定的文件可以存在,也可以不存在)

if 以 "wb" 方式写

if该文件已经存在

原文件将被删去重新建立;

else

按指定的名字新建一个文件;

else if 以 "ab" 方式写

if该文件已经存在

写入的数据将被添加到指定文件原有数据的后面,不会删去原来的内容;

else

按指定的名字新建一个文件 (与"wb"相同);

if文件同时读和写

使用 "rb+"、"wb+" 或 "ab+" 打开文件

ve.

打开文件和关闭文件

```
if((fp = fopen("f12-2.data", "wb")) == NULL) {
    printf("File open error!\n"); exit(0);
}
```

fopen("文件名", "文件打开方式")

- □使文件指针与相应文件实体对应起来
- □程序对文件指针进行操作,即fp代表磁盘文件
- 函数fopen()的返回值
 - □执行成功,则返回包含文件缓冲区等信息的FILE型结构 地址,赋给文件指针fp
 - □不成功,则返回一个NULL(空值)

exit(0): 关闭所有打开的文件,并终止程序的执行 参数0表示程序正常结束;非0参数通常表示不正常的程序结束

100

关闭文件

```
if( fclose(fp) ) {
    printf("Can not close the file!\n");
    exit(0);
}
```

fclose(文件指针)

- □把缓冲区中的数据写入磁盘扇区,确保写文件的正常完成
- □ 释放文件缓冲区单元和FILE结构体,使文件指针与具体文件脱钩

函数fclose()的返回值

- □返回0:正常关闭文件
- □返回非0:无法正常关闭文件

10

文件读写函数

- 字符读写函数: fgetc() / fputc()
- ■字符串读写函数: fputs() / fgets()
- 格式化读写函数: fscanf() / fprintf()
- 二进制读写函数: fread () / fwrite()
- ■其他相关函数
 - □检测文件结尾函数feof()
 - □检测文件读写出错函数ferror()
 - □清除末尾标志和出错标志函数clearerr()
 - □文件定位的函数fseek(), rewind(), ftell()

字符串数据流: sprintf, sscanf

数据流 (数组)	标准输入输出	文件输入输出	
打开/关闭	程序自动创建/关闭 (键盘输入,窗口输出) (stdin, stdout, stderr)	通过函数调用手动打开/关闭文件 fopen("文件名", "文件打开方式") fclose(文件指针)	
数目	通常只有一个	不同文件指针指示多个文件	
格式化读写	scanf(格式字符串, 输入表) printf(格式字符串, 输出表)	fscanf(文件指针,格式字符串,输入表) fprintf(文件指针,格式字符串,输出表)	
字符读写	ch = getchar() (失败EOF) putchar(ch)	ch = fgetc(文件指针) (失败EOF) fputc(ch, 文件指针)	
字符串读写	gets(str) (\n或EOF, \n->\0) puts(str) (\0 -> \n) (未指定str大小->缓冲区溢出)	fgets(str, n, 文件指针) (n-1个字符或\n 或EOF, 保留\n, 最后加\0) fputs(str, 文件指针) (不输出\0)	
二进制读写		fread(buffer, size, count, 文件指针) fwrite(buffer, size, count, 文件指针)	
判断是否读完	(ch = getchar()) != EOF	feof(文件指针) (可以存储EOF字符)	
当前读写位置		fseek(文件指针, offset, from) rewind(文件指针) / ftell(文件指针)	
错误处理		ferror(文件指针) / clearerr(文件指针)	

文件函数返回值

■通过返回值检查是否成功 $if((fp = fopen("f12-2.txt", "w")) == NULL) {$ printf("File open error!\n"); exit(0); while(!feof(fp)) if ((ch = fgetc(fp)) == EOF)break; if (fscanf(fp, "%d", &n) == EOF)break; while ((readBlocks = fread(buffer, size, count, fp)) != 0)

. . .

м.

专题五 文件进阶

- 文件操作基础回顾
- ■基本二进制文件读写
 - □二进制文件及读写操作
 - □举例:二进制文件的复制
- ■二进制文件的高级应用
 - □文件类型识别(图片、音频等)
 - □CAD图形编辑器
 - □BMP图像文件

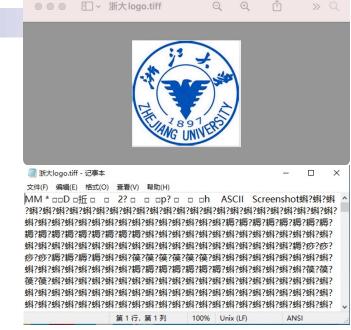


二进制文件

- ■如何理解二进制文件
 - □二进制文件是用特定软件以
 - 二进制方式(非文本方式)生成

的文件,比如:图片文件、视频文件、音乐文件、doc文件、excel文件、ppt文件、可执行文件等

■ 二进制文件无法用文本查看器打开 (显示乱码), 但可以直接看文件的二进制信息



查看二进制

■有多种工具,如文本编辑工具sublime、notepad++或者用vim命令,如下

\$ vim test.bmp

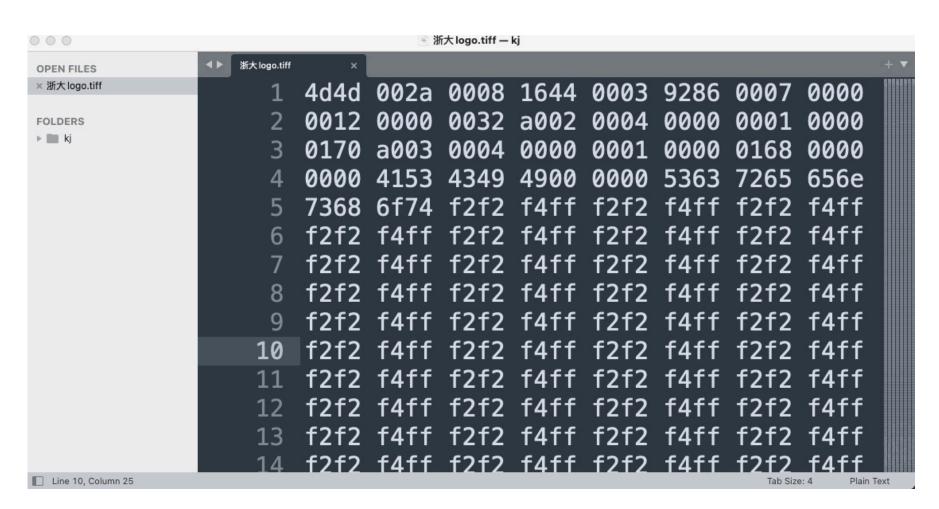
#:%!xxd

#:%!xxd -r

■ 或用C语言实现二进 制的查看输出 d<86>\N\@\@\@\@\@\@(\X\N\@\\@\@\@\@\@\@\D\@.text\@\@\@\@\@\@\@\ @\@\@\@\@\@\@\@\\$O>)\@\@D\B\@\@I÷\M\@\@\@\@\@\F\B\@\@_\@P` ^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@\@<mark>``À`rdata</mark>^@^@^@^@^@^@^@^@^ @^@^@**À**^A^@^@ä+^@^@"^K^N^@^@^@^@^@\@\@\@@^@**@**^@**P@**.xdata^@^@ @<mark>0</mark>@.pdata^@^@^@^@^@^@^@^@^@^@^@^@P^A^@@ô.^@@@^O^N^@^@@@ ^@T^@^@^@@^@**0@/4**^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^ @^@<88>^R^N^@^@^@^@^@T^@^@^@@^@^PB/16^@^@^@^@^@@@ ^@^@^@^@6^B^@^@XF^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@@^@@^@ @^B^@^@^@^@^@^PB/45^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@_<99>^M^@¾F ^@^@<mark>ä</mark>^U^N^@^@^@^@^@^A^A@^@^@@^@^PB/58^@^@^@^@^@^@^@^@^@ @Λ@Λ@Λ@ΔΛΡΛ@Λ@ὶάΛΜΛ@ῖΛIJΛΝΛ@Λ@Λ@Λ@Λ@ΛΦΛΦΛ@Λ@Λ@Λ@Α

:%!xxd





浙大logo.tiff用sublime打开看的结果

二进制数据块读fread和写fwrite

fread(buffer, size, count, fp);

从二进制文件中读入一个数据块到变量 fread的返回值为实际读入的元素个数

fwrite(buffer, size, count, fp);

向二进制文件中写入一个数据块 fwrite的返回值为实际写入的元素个数

- □buffer: 指针,表示存放数据的首地址
- □size: 数据块的字节数
- □count: 要读写的数据块块数
- □fp: 文件指针

70

命令行方式实现的二进制文件复制

- ■命令行方式
 - □ int main(int argc, char *argv[])
 - □ FILE *infile, *outfile;
- 二进制读: "rb"
 - □infile = fopen(argv[1], "rb"); //二进制打开读方式
- 二进制写: "wb"
 - □outfile = fopen(argv[2], "wb"); //二进制打开写方式

```
#include <stdio.h>
                                          copy命令源程序
#include <stdlib.h>
#define MAXLEN 1024
int main(int argc, char *argv[]) {
  int rc;
  if( argc < 3 ) {
     printf("usage: %s %s\n", argv[0], "infile outfile");
    exit(1);
                                                   命令名: mcopy
  unsigned char buf[MAXLEN];
                                                   格式: mcopy file1 file2
  FILE *infile = fopen(argv[1], "rb");
                                                   mcopy a.c b.c
  FILE *outfile = fopen(argv[2], "wb");
                                                   mcopy a.jpg b.jpg
  if( outfile == NULL || infile == NULL ) {
     printf("%s %s", argv[1], "does not exist\n");
    exit(1);
  while( (rc = fread(buf, sizeof(unsigned char), MAXLEN, infile)) != 0 )
    fwrite(buf, sizeof(unsigned char), rc, outfile);
  fclose(infile);
  fclose(outfile);
  return 0;
                                                                         23
```



说明

- fread和fwrite的参数完全一样
- 在实际进行文件读写之前,应该先弄清文件数据的数据结构,这样就可以根据数据结构来设计与文件数据一致的结构体,一次性读入或写入数据
 - 口设计数据存储的struct,以struct为单位读或写
- 而不需要每次从二进制文件中读写一小块 儿数据,从而避免出错,同时也避免多次 的循环读写,加快读写速度

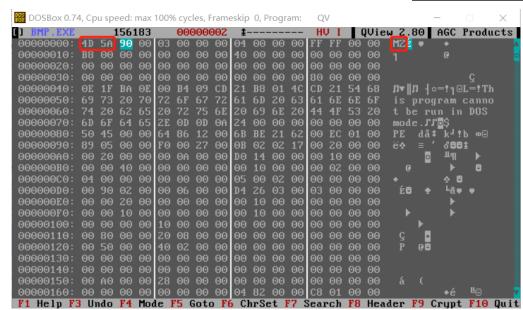


专题五 文件进阶

- 文件操作基础回顾
- ■基本二进制文件读写
- ■二进制文件的高级应用
 - □文件类型识别(图片、音频等)
 - □CAD图形编辑器
 - □BMP图像文件

应用1: 文件类型识别

- ■通过分析文件二进制识别一个文件的类型
 - □Windows PE文件 (可执行的文件,如exe、dll、ocx、sys、com)的头文件必定以0x4D5A(
 - 即字符MZ) 开头
 - □JPEG文件则以FFD8开头



```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAXLEN 100
int DISPLAY = 80;
int PAGE LENGTH = 20;
int main(int argc, char *argv[])
  char filename[MAXLEN];
  FILE *pfile = NULL;
  unsigned char buffer[DISPLAY/4-1];
  int count = 0;
  int lines = 0;
  int i, size;
```

二进制查看源程序

```
# 要求输入文件名
if(argc == 1) {
  printf("请输入文件名:\n");
  scanf("%s", filename);
} else {
  strcpy(filename, argv[1]);
  printf("执行程序:%s\n", argv[0]);
// 打开文件
if(!(pfile = fopen(filename, "rb"))) {
  printf("未能打开该文件.\n");
  return -1;
```

11 读取并显示文件内容

二进制查看源程序

```
while(!feof(pfile)) {
  size = sizeof(buffer);
  if (count < size)
     buffer[count++] = (unsigned char) fgetc(pfile);
  else {
     for (count = 0; count < size; count++)
        printf("%02X ", buffer[count]);
     printf("| ");
     for (count = 0; count < size; count++)
        printf("%c", isprint(buffer[count]) ? buffer[count] : '.');
     printf("\n");
     count = 0;
     if (!(++lines%PAGE_LENGTH))
        if (getchar() == 'E')
          return 0;
```

// 显示最后一行

printf(" ");

```
for (i = 0;i < sizeof(buffer); i++) {
    if (i < count)
        printf("%02X ", buffer[i]);
    else
```

二进制查看源程序

说明:编译后生成可执行文件seefile 使用方法:在命令行方式下输入:seefile <某个文件名>

```
printf("| ");
for (i = 0;i < count; i++)
    printf("%c", isprint(buffer[i]) ? buffer[i] : '.');
printf("\n");

// 结束
fclose(pfile);
return 0;
```



应用2: CAD图形编辑器

- ■问题:基于libgraphics实现矢量图的绘制、输出和输入
- 基本图像的表示:线、矩形、椭圆、文本 等数据结构
- ■实现线、矩形等几何图形的保存、读入
- 举例: CAD.c模块增加保存对象功能

10

CAD分析(以画线为例)

- cad.rar,解压缩cad目录,主要代码见cad.c
 - □ 1) void DrawLineD(void *line);
 - □2) 增加函数void DrawLineDSave(void *line, FILE *fp);
 - □3) 键盘事件函数,增加快捷键F6,实现保存功能
 - ■参照绘制函数: void TraverseAllList(); /*刷新所有图形对象*/
 - 增加函数: void TraverseAllListSave(); /*保存所有图形对象*/
 - □4) 思考:增加F7,实现打开文件功能,从文件读取信息,绘制到图形界面中

思考:二进制文件?

关键函数

```
/*for Line*/
void Traverse void Keyboa void DrawLineD(void *line)
//cjh added
                               LineT lptr = (LineT)line;
                     LineT
void Traverse
                               int pensize = GetPenSize();/*保存当前系统笔画粗细*
                     RectT
                               string color = GetPenColor();/*保存当前系统颜色*/
void Main()
                     Ellips
                    double
                               SetPenSize(lptr->PenSize);/*设置粗细*/
    int i;
                               SetPenColor(lptr->color);/*设置颜色*/
                               MovePen(lptr->x1, lptr->y1);
                     switch
    InitGraph
                               DrawLine(lptr->x2-lptr->x1,lptr->y2-lptr->y1);
                        cas
    Randomize
                               SetPenSize(pensize); /*恢复粗细*/
                               SetPenColor(color);/*恢复颜色*/
    registerK
    registerC
    registerM
                            //cihadded
    registerT
                           void DrawLineDSave(void *line, FILE *fp, int i)
    SetPenCol
                               LineT lptr = (LineT)line;
    SetPenSiz
                               unsigned char buf[1024];
                               sprintf(buf, "%d, %f, %f, %f, %f\n", i, lptr->x1, lptr->y1
    for (i = 0; i < NLIST;</pre>
                               fwrite(buf, strlen(buf), 1, fp);
```

M

应用3: BMP图像文件

- BMP (全称Bitmap) 是Window操作系统中的标准图像文件格式
- ■BMP分成两类:设备相关位图(DDB)和设备 无关位图(DIB)
 - □采用位映射存储格式,除图像深度可选以外, 不采用其他任何压缩,BMP文件所占用的空间 很大
 - □BMP文件的图像深度,可选1bit、4bit、8bit及 24bit
 - □文件存储时,图像扫描方式按从左到右、从下 到上顺序



应用3: BMP图像文件

- ■BMP格式分成头部和数据部等
 - □头部是描述这幅图片的具体参数,如宽、高、通道等,占54个字节
 - □从第55个字节开始,才正式开始图像数据流。 55就是所谓的BMP数据偏移量 (date offset)
- BMP文件格式是Windows环境中交换与图有关的数据的一种标准, Windows的图形图像软件都支持BMP图像格式

```
#include <stdio.h>
                               BMP图片生成源程序
#include <stdlib.h>
#define w 200
#define h 200
void WriteBMP(char*img,const char* filename);
int main() {
  char img[w*h*3];
  char *filename = "test.bmp";
  int i;
  for (i = 0; i < w^*h^*3; i++)
                                     代码生成一幅宽和高均为200的BMP随机位图
    img[i] = rand()\%256;
  WriteBMP(img, filename);
  system("test.bmp");
                 void WriteBMP(char*img, const char* filename) {
  return 0;
                    int i=(w*3+3) / 4 * 4;
                    int bmi[]= \{i*h+54,0,54,40,w,h,1|3*8 << 16, 0,i*h,0,0,100,0\};
                    FILE *fp = fopen(filename, "wb");
                    fprintf(fp,"BM");
                    fwrite(&bmi, 52, 1, fp);
                    fwrite(img, 1, i * h, fp);
                    fclose(fp);
                                                                      35
```



查看BMP图片文件的二进制

bmp文件头(14位)

变量名	地址偏移	大小	作用
bf Type	000 0 h	2 bytes	说明文件的类型,可取值为: • Im - Windows 3.1x, 95, MT, • BA - OS/2 Bitmap Array • CI - OS/2 Color Icon • CP - OS/2 Color Pointer • IC - OS/2 Icon • PT - OS/2 Pointer
bfSize	0002h	4 bytes	说明该位图文件的大小,用字节为单位
bfReserved1	0006h	2 bytes	保留,必须设置为0
bfReserved2	0008h	2 bytes	保留,必须设置为0
bfOffBits	000Ah	4 bytes	说明从文件头开始到实际的图象数据之间的字节的偏移量。 这个参数是非常有用的,因为位图信息头和 调色板的长度会根据不同情况而变化, 所以我们可以用这个偏移值迅速的从文件中 读取到位图数据。

查看BMP图片文件的二进制

位图信息头(40位)

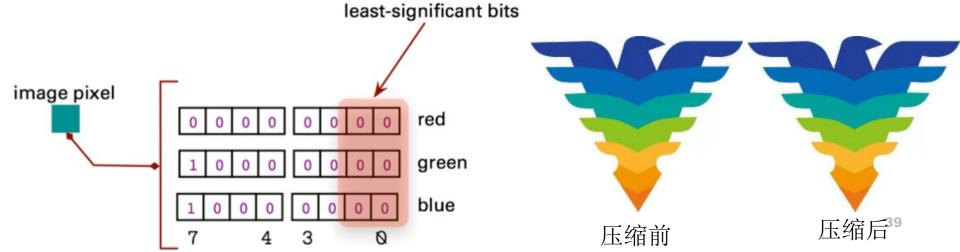
· -	地址偏移	大小	作用
文里石 hiSize	198.4E Marks	4 hertes	TEM BITMAPINFOHEADER结构所需要的字数。
biTidth	0012Ъ	4 bytes	说明图像的宽度,用像素为单位
			说明图像的高度,以像素为单位。
			注:这个值除了用于描述图像的高度之外,它还有另一
			个用处,就是指明该图像是倒向的位图,还是正向的位
biHeight	0016Ъ	4 1-4	图.
Dineight	00101	4 bytes	如果该值是一个正数,说明整像是倒角的,如果该值是
			一个负数,则说明图像是正向的。
			大多数的mp文件都是倒向的位图,也就是高度值是一个
			正数·
			II NA
			为目标设备说明额色平面数,
biPlanes	001Ah	2 bytes	其值将总是被设为1。
			Alentaria (Z/)1 ·
biBitCount	001СЪ	0.1-1	说明比特数/像素,其值为1、4、8、16、24或32。
D1D1 (Lount	OOICE	2 bytes	MACHINE HAR ALLIAN & O. 10. 2000
			说明图像数据压缩的类型。取值范围:
	001ЕЬ	4 bytes	0 BI RGB 不压缩(最常用)
			1 BI_BLES 8比特游程编码(BLE),只用于8位位图
biCompression			2 BI_BLE4 4比特游程编码(BLE),只用于4位位图
			3 BI_BITPIELDS 比特域,用于16/32位位图
			4 BI_JPEG JPEG 位图含JPEG图像(仅用于打印机)
			5 BI_PBG PBG 位图含PBG图像(仅用于打印机)
biSizeImage	0022h	4 bytes	说明图像的大小,
DISTZELBAGE	UUZZII	4 Dytes	以字节为单位。当用BI BGB格式时,可设置为O。
bi I PelsPer l eter	0026Ъ	4 bytes	说明水平分辨率,用像素/米表示,有符号整数
biTPelsPerTeter	002Ah	4 bytes	说明垂直分辨率,用像素/米表示,有符号整数
1 : m . r	0000		说明位图实际使用的彩色表中的颜色索引数
bittrUsed	biClrUsed 002Eh 4 bytes	(设为中的话,则说明使用所有调色板项)	
			说明对图像显示有重要影响的颜色索引的数目
biClrImportant	0032Ъ	4 bytes	如果是0,表示都重要。
			PARTO AVINDA X

```
#include <stdio.h>
                         获取BMP图片宽高源程
int main()
                                      】Z:\Mac 上的 lxx\zjutest\examples\ex33\getbmpwh.exe
  unsigned char data[4];
                                        度为0000009f像素
                                      片高度为ffffffc像素
  FILE *fp;
  fp = fopen("zju.bmp", "rb");
                                    rocess exited after 4.739 seconds with return value 0
                                     按任意键继续.
  fseek(fp, 0x12, SEEK SET);
  fread(data, 4, 1, fp);
  printf("图片宽度为%02x%02x%02x%02x像素\n",
         data[3], data[2], data[1], data[0]);
                                                           分辨率
                                                                159 x 132
  fseek(fp, 0x16, SEEK SET);
                                                                159 像素
                                                           位深度
  fread(data, 4, 1, fp);
                                                           名称
                                                                zju.bmp
  printf("图片高度为%02x%02x%02x%02x像素\n",
                                                           文件夹路径 Z:\Mac 上的 lxx\zjutest\examples\ex33
          data[3], data[2], data[1], data[0]);
                                                                2022/4/1 19:35
                                                                2022/4/1 19:35
                                                                82.0 KB
                                                           大小
  return 0;
                                                           属性
                                                           可用性
                                                           脱机状态
                                                           共享设备
                                                                S-1-1-0
                                                           计算机.
                                                                vmware-host
```

删除属性和个人信息



■对于BMP文件简易的图片压缩算法:由于BMP文件对图片储存方式为直接存储每个像素的RGB,因此可以通过将最低有效位(LSB, Least Significant Bit)置零进行压缩(存储时不存储这些0,读取时将这些0恢复),肉眼很难看出该压缩前后图片的变化



查看图片文件的二进制

zju.bmp

文件类型: BMP 文件 (.bmp)

打开方式: 照片

更改(<u>C</u>)...

位置:

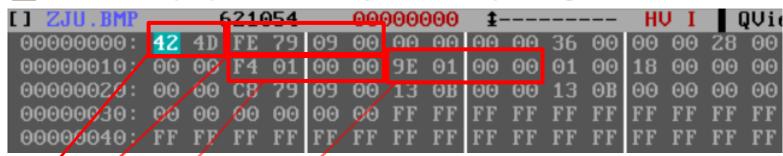
C:\Users\49437\Desktop

大小:

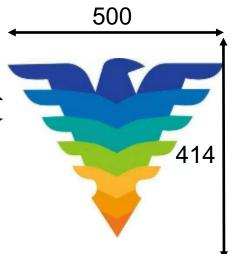
QV

606 KB (621,054 字节)

BOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program:



- 4D即ASCII码BM,表示为Windows文件
- FE 79 09 00表示图片大小,Windows文件遵循小端规则:即整数的低位在前,高位在后,故文件大小实际为00 09 79 FE字节,即621054字节
- F4 01 00 00表示图像宽度,同样根据小端规则,实际为0x1F4像素,即500像素
- 9E 01 00 00表示图像高度,即0x19E像素,414 像素



专题五 文件进阶总结

- ■文件的基本概念
 - □文本文件和二进制文件、文件缓冲系统
 - □文件结构,文件类型指针,自定义类型
 - □文件的打开与关闭、文件处理实现过程
- ■基本二进制文件读写
 - □二进制文件及读写操作
 - □举例:二进制文件的复制
- ■二进制文件的高级应用
 - □文件类型识别(图片、音频等)
 - □CAD图形编辑器
 - □BMP图像文件