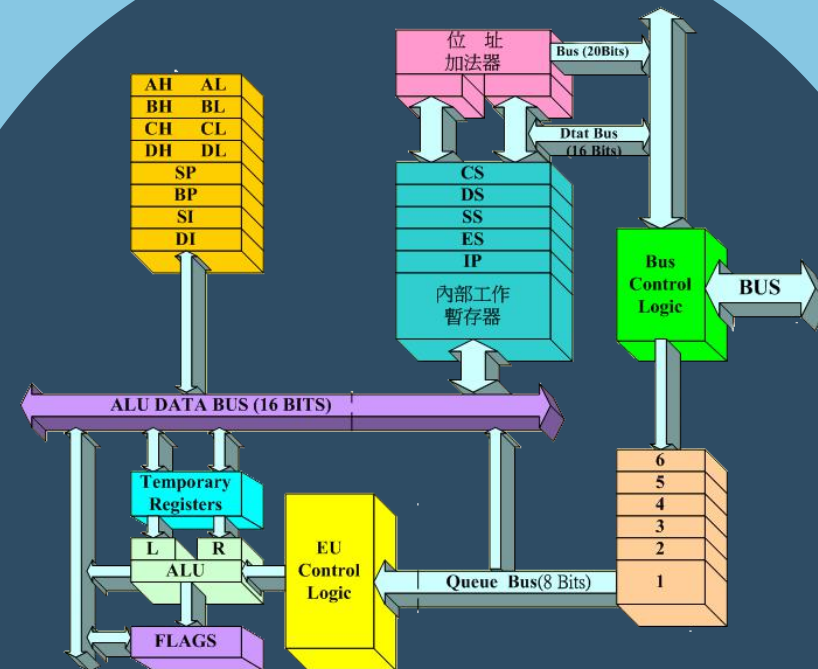


# 文件结构分析

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 学会底层工作方法

## 案例：读懂BMP图像文件

- BMP，全称Bitmap，Windows操作系统中的标准图像文件格式

## BMP图像文件的组成

- 位图头文件头数据结构
- 位图信息头
- 彩色表/调色板
- 位图数据



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEB...
C:\>debug bmp16bit.bmp
-d
073F:0100  42 4D 76 02 00 00 00 00-00 00 76 00 00 00 28 00  BM.....v...C.
073F:0110  00 00 20 00 00 00 20 00-00 00 01 00 04 00 00 00  .. ...
073F:0120  00 00 00 02 00 00 C4 0E-00 00 C4 0E 00 00 00 00  .....
073F:0130  00 00 00 00 00 00 00 00-00 00 00 00 80 00 00 80  .....
073F:0140  00 00 00 80 80 00 80 00-00 00 80 00 80 00 80 80  .....
073F:0150  00 00 80 80 80 00 C0 C0-C0 00 00 00 FF 00 00 FF  .....
073F:0160  00 00 00 FF FF 00 FF 00-00 00 FF 00 FF 00 FF FF  .....
073F:0170  00 00 FF FF FF 00 33 33-33 33 33 33 33 39 99  .....33333339.
-d
073F:0180  99 99 99 99 99 99 33 33-33 33 33 33 33 39 99  .....33333339.
073F:0190  99 99 99 99 99 99 33 CC-CC CC CC CC CC 33 39 99  .....3.....39.
073F:01A0  99 99 99 99 99 99 33 CF-FF FF FF FF FC 33 39 99  .....3.....39.
073F:01B0  99 99 99 99 99 99 33 CF-FF FF FF FF FC 33 39 99  .....3.....39.
073F:01C0  99 99 99 99 99 99 33 CF-FF FF FF FF FC 33 39 99  .....3.....39.
073F:01D0  99 99 99 99 99 99 33 CC-CC CC CC CC CC 33 39 99  .....3.....39.
073F:01E0  99 99 99 99 99 99 33 33-33 33 33 33 33 33 39 99  .....33333339.
073F:01F0  99 99 99 99 99 99 33 33-33 88 88 88 88 88 88 88  .....333.....
-
```

# 一个文件内容查看工具——Binary Viewer

Binary Viewer: D:\000000-Lecture\0 C++语言基础\7 输入输出流\7 案例: bmp文件格式剖析\bmp16bit.bmp

File Edit Search View Tools Window Help

Visualizer

Show numbers as: ☐ Decimal ☒ Hexadecimal

Position	Address	Value	Color
Selected	N/A	N/A	N/A
Hot	275	66	

Logical Pixel Size 2 Screen Pixels  
Image Width 256 Logical Pixels  
Density 1 Byte per Logical  
Mode Linear

Logical Pixel Size  
Magnifying factor of each image pixels.

Size: 630 Dec/276 Hex

Data View

A...	Hexadecimal (1 Byte)	Text (ASCII)
0000	42 4D 76 02 00 00 00 00 00 00 00 76 00 00 00 28 00	B M v . . . . . v . . . . . ( .
0010	00 00 20 00 00 00 20 00 00 00 01 00 04 00 00 00	. . . . . . . . . . . . . . . . .
0020	00 00 00 02 00 00 C4 0E 00 00 C4 0E 00 00 00 00	. . . . . . . . . . . . . . . . .
0030	00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 80	. . . . . . . . . . . . . . . . .
0040	00 00 00 80 80 80 80 00 00 00 80 00 00 80 00 80	. . . . . . . . . . . . . . . . .
0050	00 00 80 80 80 00 C0 C0 C0 00 00 00 FF 00 00 FF	. . . . . . . . . . . . . . . . .
0060	00 00 00 FF FF 00 FF 00 00 00 FF 00 FF 00 FF FF	. . . . . . . . . . . . . . . . .
0070	00 00 FF FF FF 00 33 33 33 33 33 33 33 39 99	. . . . . . . . . . . . . . . . .
0080	99 99 99 99 99 99 33 33 33 33 33 33 33 39 99	. . . . . . . . . . . . . . . . .
0090	99 99 99 99 99 99 33 CC CC CC CC CC 33 39 99	. . . . . . . . . . . . . . . . .
00A0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99	. . . . . . . . . . . . . . . . .
00B0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99	. . . . . . . . . . . . . . . . .
00C0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99	. . . . . . . . . . . . . . . . .
00D0	99 99 99 99 99 99 33 CC CC CC CC CC CC 33 39 99	. . . . . . . . . . . . . . . . .
00E0	99 99 99 99 99 99 33 33 33 33 33 33 33 39 99	. . . . . . . . . . . . . . . . .
00F0	99 99 99 99 99 99 33 33 88 88 88 88 88 88 88	. . . . . . . . . . . . . . . . .
0100	88 88 88 88 99 99 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0110	EE EE EE E8 99 99 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0120	EE EE EE E8 99 99 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0130	EE EE EE E8 99 99 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0140	EE EE EE E8 99 99 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0150	EE EE EE E8 99 99 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0160	EE EE EE E8 99 00 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0170	EE EE EE E8 00 06 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0180	EE EE EE E8 06 66 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .
0190	EE EE E0 08 66 66 33 33 8D DD DD DD DD D9 EE	. . . . . . . . . . . . . . . . .

File Name: bmp16bit.bmp | Size: 630 Bytes | Address: 00000000(Hex)/0(Dec) | Selection Size: 1 Bytes

Histogram

File Statistics

Entropy	Average	Min Frequency	Max Frequency	Sample Size	Uniqu...	Guessed File Type

Bookmarks Structures Histogram

Data Inspector (00Hex/0Dec)

☐ Big Endian ☒ Little Endian

1 Byte

Hexadecimal 00  
Binary 00000000  
Octal 0  
Unsigned Integer 0  
Signed Integer 0  
ASCII

2 Bytes

Hexadecimal N/A  
Binary N/A  
Octal N/A

Hexadecimal  
Hexadecimal ( 1 Byte)

File Properties

Common

SparseFile False  
System False  
Temporary False

File Info

Create Date 5/22/2015, 7:48:46  
Last Access Time 5/22/2015, 7:48:46  
Last Write Time 3/7/2003, 5:23:42  
Name bmp16bit.bmp  
Size 630

Name  
File name.

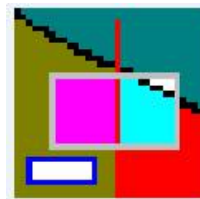


# 位图文件

位图文件头(bitmap-file header)

位图信息数据结构 { 位图信息头(bitmap-information header)  
彩色表(color table)

位图信息(bitmapinfo)



一幅32×32的  
16色bmp图像

/\*位图文件头格式\*/

```
typedef struct tagBITMAPFILEHEADER
```

```
{  
    UINT bfType; /*说明文件的类型*/  
    DWORD bfSize; /*说明文件的大小*/  
    UINT bfReserved1; /*保留，设置为0*/  
    UINT bfReserved2; /*保留，设置为0*/  
    DWORD bfOffBits; /*到图像数据的偏移量*/  
} BITMAPFILEHEADER;
```

```
typedef struct tagBITMAPINFO {  
    BITMAPINFOHEADER bmiHeader; // 位图信息头  
    RGBQUAD bmiColors[16]; // 颜色表  
} BITMAPINFO;
```

/\*位图信息\*/

- 各个像素的颜色的序号（彩色表的下标）
- 从左下到右上的像素
- 与设备操控有关
- 长度：biSizeImage

/\*位图信息头\*/

```
typedef struct tagBITMAPINFOHEADER {
```

```
    DWORD biSize; /*BITMAPINFOHEADER结构所需要的字节数*/  
    LONG biWidth; /*图像的宽度，以像素为单位*/  
    LONG biHeight; /*图像的高度，以像素为单位*/  
    WORD biPlanes; /*为目标设备说明位面数，其值设置为1*/  
    WORD biBitCount; /*位数/像素*/  
    DWORD biCompression; /*图像数据压缩的类型：不压缩，或4/8位RLE*/  
    DWORD biSizeImage; /*图像的大小，以字节为单位。*/  
    LONG biXPelsPerMeter; /*水平分辨率，用像素/米表示*/  
    LONG biYPelsPerMeter; /*垂直分辨率，用像素/米表示*/  
    DWORD biClrUsed; /*位图使用的彩色表中的颜色索引数：2/16/256/224*/  
    DWORD biClrImportant; /*对图像显示有重要影响的颜色索引的数目*/
```

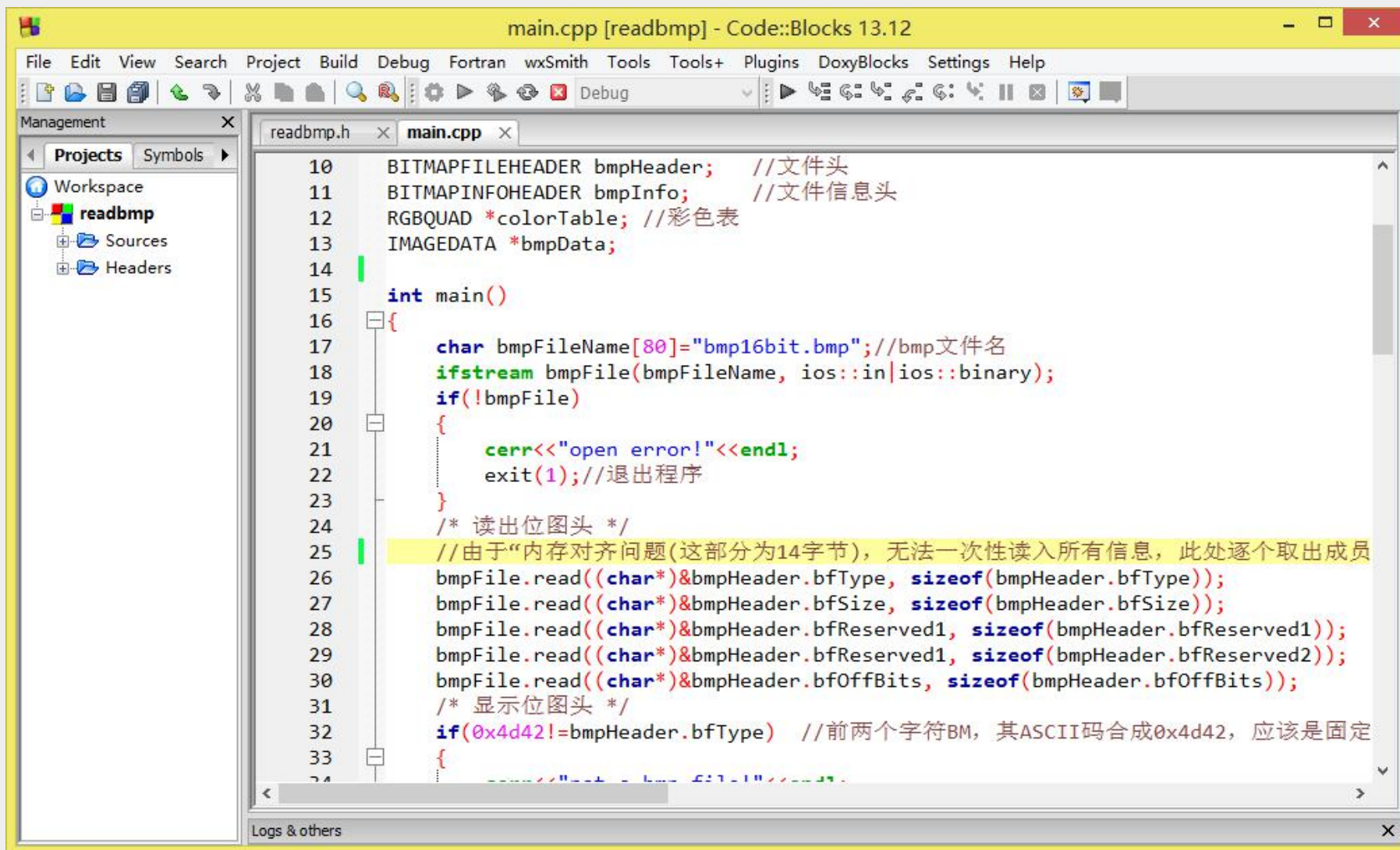
```
} BITMAPINFOHEADER;
```

/\*色彩结构体\*/

```
typedef struct tagRGBQUAD { /* rgbq */  
    BYTE rgbBlue; /*指定蓝色强度*/  
    BYTE rgbGreen; /*指定绿色强度*/  
    BYTE rgbRed; /*指定红色强度*/  
    BYTE rgbReserved; /*保留，设置为0*/  
} RGBQUAD;
```

A...	Hexadecimal (1 Byte)
0000	42 4D 76 02 00 00 00 00 00 00 00 76 00 00 00 28 00
0010	00 00 20 00 00 00 20 00 00 00 01 00 04 00 00 00
0020	00 00 00 02 00 00 C4 0E 00 00 C4 0E 00 00 00 00
0030	00 00 00 00 00 00 00 00 00 00 00 80 00 00 80
0040	00 00 00 80 80 00 80 00 00 00 00 80 00 80 80
0050	00 00 80 80 80 00 C0 C0 C0 00 00 FF 00 00 FF
0060	00 00 00 FF FF 00 FF 00 00 FF 00 00 FF 00 FF
0070	00 00 FF FF FF 00 33 33 33 33 33 33 33 33 33
0080	99 99 99 99 99 99 33 33 33 33 33 33 33 33 33
0090	99 99 99 99 99 99 33 CC CC CC CC CC 33 33 33
00A0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 33
00B0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 33
00C0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 33
00D0	99 99 99 99 99 99 33 CC CC CC CC CC 33 33 33
00E0	99 99 99 99 99 99 33 33 33 33 33 33 33 33 33
00F0	99 99 99 99 99 99 33 33 33 88 88 88 88 88 88
0100	88 88 88 88 99 99 33 33 33 8D DD DD DD DD D9 EE
0110	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0120	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0130	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0140	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE

# 用C程序读出BMP文件信息



The screenshot shows a C++ IDE window titled "main.cpp [readbmp] - Code::Blocks 13.12". The left sidebar displays a project named "readbmp" with subfolders for "Sources" and "Headers". The main editor area shows the following C++ code:

```
10  BITMAPFILEHEADER bmpHeader; //文件头
11  BITMAPINFOHEADER bmpInfo; //文件信息头
12  RGBQUAD *colorTable; //彩色表
13  IMAGEDATA *bmpData;
14
15  int main()
16  {
17      char bmpFileName[80]="bmp16bit.bmp";//bmp文件名
18      ifstream bmpFile(bmpFileName, ios::in|ios::binary);
19      if(!bmpFile)
20      {
21          cerr<<"open error!"<<endl;
22          exit(1);//退出程序
23      }
24      /* 读出位图头 */
25      //由于“内存对齐问题(这部分为14字节)，无法一次性读入所有信息，此处逐个取出成员
26      bmpFile.read((char*)&bmpHeader.bfType, sizeof(bmpHeader.bfType));
27      bmpFile.read((char*)&bmpHeader.bfSize, sizeof(bmpHeader.bfSize));
28      bmpFile.read((char*)&bmpHeader.bfReserved1, sizeof(bmpHeader.bfReserved1));
29      bmpFile.read((char*)&bmpHeader.bfReserved2, sizeof(bmpHeader.bfReserved2));
30      bmpFile.read((char*)&bmpHeader.bfOffBits, sizeof(bmpHeader.bfOffBits));
31      /* 显示位图头 */
32      if(0x4d42!=bmpHeader.bfType) //前两个字符BM，其ASCII码合成0x4d42，应该是固定
33      {
34          //显示错误信息
35      }
```

用汇编语言呢？