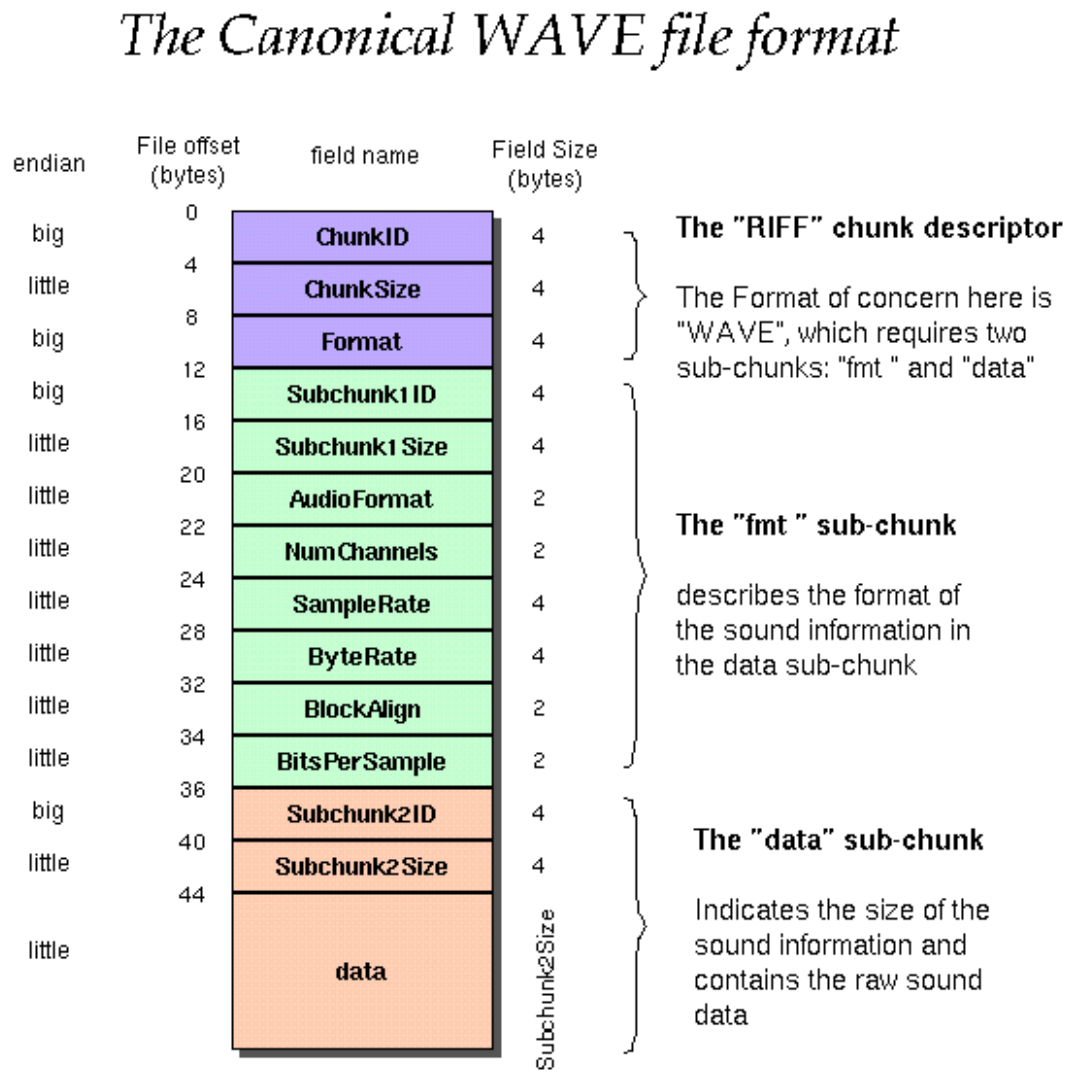
WAV文件格式详解

1. WAV 概述

Waveform Audio File Format （ ***WAV***，又或者是因为 ***WAV*** 后缀而被大众所知的），它采用 ***RIFF*** （Resource Interchange File Format）文件格式结构。通常用来保存 ***PCM*** 格式的原始音频数据，所以通常被称为无损音频。但是严格意义上来讲， ***WAV*** 也可以存储其它压缩格式的音频数据。

2. WAV文件格式解析

WAV 文件遵循 ***RIFF*** 规则用于存储多媒体文件，其内容以区块（`chunk`）为最小单位进行存储。`WAV`文件一般由3个区块组成：`RIFF chunk`、`Format chunk`和`Data chunk`。所有基于压缩编码的`WAV`文件必须含有`fact`块。此外所有其它块都是可选的。




```
Offset  Size  Name             Description
----------------------------------------------------------------------------------------
--------------
```

```
The canonical WAVE format starts with the RIFF header:

0         4   ChunkID          Contains the letters "RIFF" in ASCII form
                               (0x52494646 big-endian form).
4         4   ChunkSize        36 + SubChunk2Size, or more precisely:
                               4 + (8 + SubChunk1Size) + (8 + SubChunk2Size)
                               This is the size of the rest of the chunk
                               following this number.  This is the size of the
                               entire file in bytes minus 8 bytes for the
                               two fields not included in this count:
                               ChunkID and ChunkSize.
8         4   Format           Contains the letters "WAVE"
                               (0x57415645 big-endian form).

The "WAVE" format consists of two subchunks: "fmt " and "data":
The "fmt " subchunk describes the sound data's format:

12        4   Subchunk1ID      Contains the letters "fmt "
                               (0x666d7420 big-endian form).
16        4   Subchunk1Size    16 for PCM.  This is the size of the
                               rest of the Subchunk which follows this number.
20        2   AudioFormat      PCM = 1 (i.e. Linear quantization)
                               Values other than 1 indicate some
                               form of compression.
22        2   NumChannels      Mono = 1, Stereo = 2, etc.
24        4   SampleRate       8000, 44100, etc.
28        4   ByteRate         == SampleRate * NumChannels * BitsPerSample/8
32        2   BlockAlign       == NumChannels * BitsPerSample/8
                               The number of bytes for one sample including
                               all channels. I wonder what happens when
                               this number isn't an integer?
34        2   BitsPerSample    8 bits = 8, 16 bits = 16, etc.
          2   ExtraParamSize   if PCM, then doesn't exist
          X   ExtraParams      space for extra parameters

The "data" subchunk contains the size of the data and the actual sound:

36        4   Subchunk2ID      Contains the letters "data"
                               (0x64617461 big-endian form).
40        4   Subchunk2Size    == NumSamples * NumChannels * BitsPerSample/8
                               This is the number of bytes in the data.
                               You can also think of this as the size
                               of the read of the subchunk following this
                               number.
44        *   Data             The actual sound data.
```

2.1 WAVE 文件结构

WAV 文件采用的是 ***RIFF*** 格式结构。至少是由**3**个块构成**,**分别是 ***RIFF*** 、 ***fmt*** 和 ***Data*** 。所有基于压缩编码的 ***WAV*** 文件必须含有 ***fact*** 块。此外所有其它块都是可选的。块 ***fmt*** , ***Data*** 及 ***fact*** 均为 ***RIFF*** 块的子块。 ***WAV*** 文件的文件格式类型标识符为“ ***WAV*** ”。基本结构如表**2**。

表 2 **WAVE** 文件结构

RIFF 块
文件格式类型“WAVE”
fmt 块
fact 块(压缩编码格式要含有该块) 表**3**。
data 块

表3 常见的压缩编码格式

格式代码	格式名称	**fmt** 块长度	**fact** 块
1(0x0001)	PCM/非压缩格式	16	
2(0x0002	Microsoft ADPCM	18	√
3(0x0003)	IEEE float	18	√
6(0x0006)	ITU G.711 a-law	18	√
7(0x0007)	ITU G.711 μ-law	18	√
49(0x0031)	GSM 6.10	20	√
64(0x0040)	ITU G.721 ADPCM		√
65,534(0xFFFE)	见子格式块中的编码格式	40	

2.2 WAV文件头格式

WAV文件由文件头和数据体两部分组成。其中,文件头是由文件标识字段与格式块两部分组成,后者保存的是编码参数和声音参数,格式如表**4**。

表 4 **WAVE** 文件头格式

偏移地址	字节数	数据类型	字段名称	字段说明
00H	4	字符	文档标识	大写字符串"RIFF",标明该文件为有效的 RIFF 格式文档。
04H	4	长整型数	文件数据长度	从下一个字段首地址开始到文件末尾的总字节数。该字段的数值加 8 为当前文件的实际长度。
08H	4	字符	文件格式类型	所有 WAV 格式的文件此处为字符串"WAVE",标明该文件是 WAV 格式文件。
0CH	4	字符	格式块标识	小写字符串,"fmt "。
10H	4	长整型数	格式块长度。	其数值不确定,取决于编码格式。可以是 16、 18 、20、40 等。(见表 2)
14H	2	整型数	编码格式代码。	常见的 WAV 文件使用 PCM 脉冲编码调制格式,该数值通常为 1。(见表 3)
16H	2	整型数	声道个数	单声道为 1,立体声或双声道为 2
18H	4	长整型数	采样频率	每个声道单位时间采样次数。常用的采样频率有 11025, 22050 和 44100 kHz。
1CH	4	长整型数	数据传输速率,	该数值为:声道数×采样频率×每样本的数据位数/8。播放软件利用此值可以估计缓冲区的大小。
20H	2	整型数	数据块对齐单位	采样帧大小。该数值为:声道数×位数/8。播放软件需要一次处理多个该值大小的字节数据,用该数值调整缓冲区。
22H	2	整型数	采样位数	存储每个采样值所用的二进制数位数。常见的位数有 4、8、12、16、24、32
24H				对基本格式块的扩充部分(详见扩展格式块,格式块的扩充)

2.2.1 扩展格式块

当**WAV**文件采用非**PCM**编码时,使用的是扩展格式块,它是在基本格式块**fmt**之后扩充了一个的数据结构。该结构的前两字节为长度字段,指出后面区域的长度。紧接其后的区域称之为扩展区,含有扩充的格式信息,其长度取决于压缩编码类型。当某种编码格式(如**ITU G.711 a-law**)使扩展区的长度为**0**时,长度字段还必须保留,只是长度字段的数值为**0**。因此,扩展格式块长度的最小值为基本格式块的长度**16**加**2**。

2.2.2 格式块的扩充

当编码格式代码为**0xFFFE**时,为扩充标识码。此时格式块扩展区长度为**24**字节,包含了新增的格式字段和真正的编码格式代吗,格式如表**5**。

表5

偏移	长度	数据类型	字段名称	字段说明
24H	2	整型数	扩展区长度	22
26H	2	整型数	有效采样位数	最大值为每个采样字节数*8
28H	4	长整形数	扬声器位置	声道号与扬声器位置映射的二进制掩码
32H	2	整型数	编码格式	真正的编码格式代码
34H	14			\x00\x00\x00\x00\x10\x00\x80\x00\x00\xAA\x00\x38\x9B\x71

2.2.3 fact块

采用压缩编码(修订版**Rev.3**以后出现的编码格式)的**WAV**文件必须有含有**fact**块。块标识符为“**fact**”。块长度至少**4**个字节。目前**fact**块只有一个数据项,为每个声道采样总数,或采样帧总数。该数值可由**data**块中的数据长度除以数据块对齐单位的数值计算出。虽然基于压缩编码的文件含有**fact**块,然而,实测中发现,将文件转换成**PCM**编码格式后,原**fact**块仍然存在(如表**6**)。

表 6 **fact** 块结构示意图

字段	长度	内容
块标识	4	"fact"
块长度	4	4(最小数值为 4 个字节)
采样总数	4	采样总数 (每个声道)

3 WAV文件语音数据的组织结构

WAV文件的声音数据保存在数据块中。块标识符为“**data**”,块长度值为声音数据的长度。从数据块的第9个字符开始是声音波形采样数据。每个样本按采样的时间先后顺序写入。样本的字节数取决于采样位数。对于多字节样本,低位字节数据放在低地址单元,相邻的高位字节数据放在高地址单元。多声道样本数据采用交替方式存储。例如:立体声(双声道)采样值的存储顺序为:通道1第1采样值,通道2第1采样值;通道1第2采样值,通道2第2采样值;以此类推。基于PCM编码的样本数据排列方式如表**7-9**。

表7 **8**位**PCM**

	样本 **1**		样本 **2**	
8 位单声道	0 声道		0 声道	
8 位立体声	0 声道(左)	1 声道(右)	0 声道(左)	1 声道(右)

表8 **16**位单声道**PCM,**每个采样点占**2**个字节

	样本 **1**		样本 **2**	
16 位单 声道	0 声道 低字节	0 声道 高字节	0 声道 低字节	0 声道 高字节

表9 **16**位立声道**PCM,**每个采样点占**4**个字节

样本 **1**			
0-左声 道低字节	0-左声 道高字节	1-右声 道低字节	1-右声 道高字节

4 实例分析

4.1 采样率16KHz 采用位宽16bit 双通道1KHz正弦波

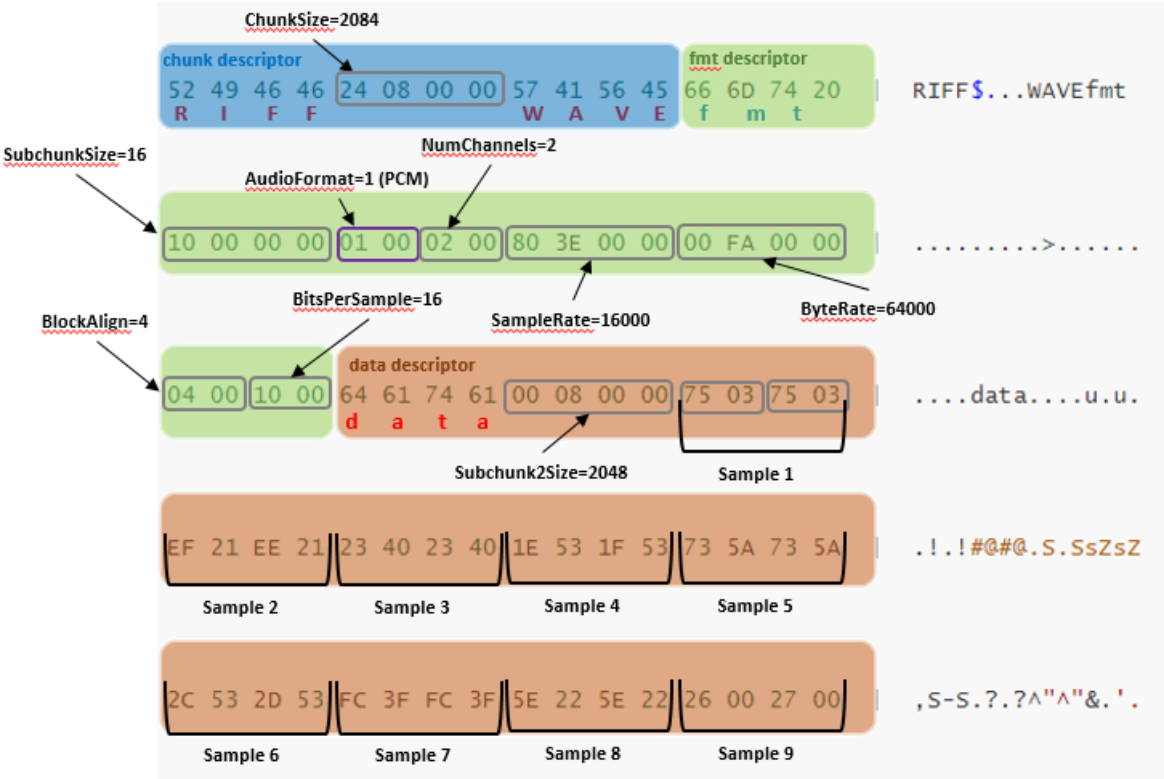
我使用数字音频处理的瑞士军刀工具 ***sox*** 程序创建SampleRate 16000 BitsPerSample 16bit标准WAVE格式：

```
$ sox -n -e signed-integer -r16k -c2 -b16 sine.wav synth 0.032 sine  1000.0
$
$ soxi sine.wav

Input File     : 'sine.wav'
Channels       : 2
Sample Rate    : 16000
Precision      : 16-bit
Duration       : 00:00:00.03 = 512 samples ~ 2.4 CDDA sectors
File Size      : 2.09k
Bit Rate       : 523k
Sample Encoding: 16-bit Signed Integer PCM
```

例如使用hexdump工具显示sine.wav文件的开头256个字节，其中的字节显示为十六进制数字：

```
$ hexdump -e '16/1 "%02X " "  |  "' -e '16/1 "%_p" "\n"' -n256 sine.wav
52 49 46 46 24 08 00 00 57 41 56 45 66 6D 74 20  |  RIFF$...WAVEfmt
10 00 00 00 01 00 02 00 80 3E 00 00 00 FA 00 00  |  .........>......
04 00 10 00 64 61 74 61 00 08 00 00 75 03 75 03  |  ....data....u.u.
EF 21 EE 21 23 40 23 40 1E 53 1F 53 73 5A 73 5A  |  .!.!#@#@.S.SsZsZ
2C 53 2D 53 FC 3F FC 3F 5E 22 5E 22 26 00 27 00  |  ,S-S.?.?^"^"&.'.
54 DD 55 DD 53 C0 52 C0 84 AC 84 AC DE A5 DE A5  |  T.U.S.R.........
8A AC 8B AC 46 C0 46 C0 67 DD 66 DD 0E 00 0F 00  |  ....F.F.g.f.....
7C 22 7D 22 D9 3F D8 3F 57 53 57 53 42 5A 41 5A  |  |"}".?.?WSWSBZAZ
5B 53 5C 53 D0 3F D0 3F 88 22 88 22 FE FF FF FF  |  [S\S.?.?."."....
7A DD 7A DD 2E C0 2E C0 A6 AC A6 AC BF A5 BE A5  |  z.z.............
A6 AC A7 AC 2D C0 2B C0 7D DD 7D DD FB FF FB FF  |  ....-.+.}.}.....
8D 22 8E 22 CA 3F CA 3F 63 53 62 53 39 5A 39 5A  |  ."."."?.?cSbS9Z9Z
62 53 62 53 CC 3F CC 3F 8B 22 8A 22 FD FF FE FF  |  bSbS.?.?."."....
79 DD 79 DD 30 C0 30 C0 A2 AC A2 AC C3 A5 C3 A5  |  y.y.0.0.........
A2 AC A2 AC 31 C0 32 C0 77 DD 78 DD 01 00 00 00  |  ....1.2.w.x.....
88 22 88 22 D0 3F CF 3F 5D 53 5D 53 3D 5A 3D 5A  |  ."."."?.?]S]S=Z=Z
```



（1）“52 49 46 46”这个是Ascii字符“**RIFF**”，这部分是固定格式，表明这是一个**WAVE**文件头。
（2）“24 08 00 00”，这个是我这个**WAVE**文件的数据大小，这个大小包括除了前面4个字节的所有字节，也就等于文件总字节数减去8。16进制的“24 08 00 00”对应是十进制的“2084”。
（3）“57 41 56 45 66 6D 74 20”，也是Ascii字符“**WAVEfmt**”，这部分是固定格式。
以后是**PCMWAVEFORMAT**部分

（4）“10 00 00 00”，这是一个DWORD，对应数字16，这个对应定义中的**PCMWAVEFORMAT**部分的大小，可以看到后面的这个段内容正好是16个字节。
（5）“01 00”，这是一个WORD，对应定义为编码格式（WAVE_FORMAT_PCM格式一般用的是这个）。
（6）“02 00”，这是一个WORD，对应数字2，表示声道数为2，是双声道**Wav**。
（7）“80 3E 00 00”对应数字16000，代表的是采样频率16000，采样率（每秒样本数），表示每个通道的播放速度
（8）“00 FA 00 00”对应数字64000，代表的是每秒的数据量，波形音频数据传送速率，其值为通道数×每秒样本数×每样本的数据位数 / 8（(2*1600*16)/8）。播放软件利用此值可以估计缓冲区的大小。
（9）“04 00”对应数字是4，表示块对齐的内容。数据块的调整数（按字节算的），其值为通道数×每样本的数据位值 / 8。播放软件需要一次处理多个该值大小的字节数据，以便将其值用于缓冲区的调整。
（10）“10 00”数值为16，采样大小为16Bits，每样本的数据位数，表示每个声道中各个样本的数据位数。如果有多个声道，对每个声道而言，样本大小都一样。

4.2 采样率16KHz 采用位宽32bit 双通道1KHz正弦波

例如用 ***sox*** 程序生成一个SampleRate 16000 BitsPerSample 32bit 的1KHz的sine wav文件：

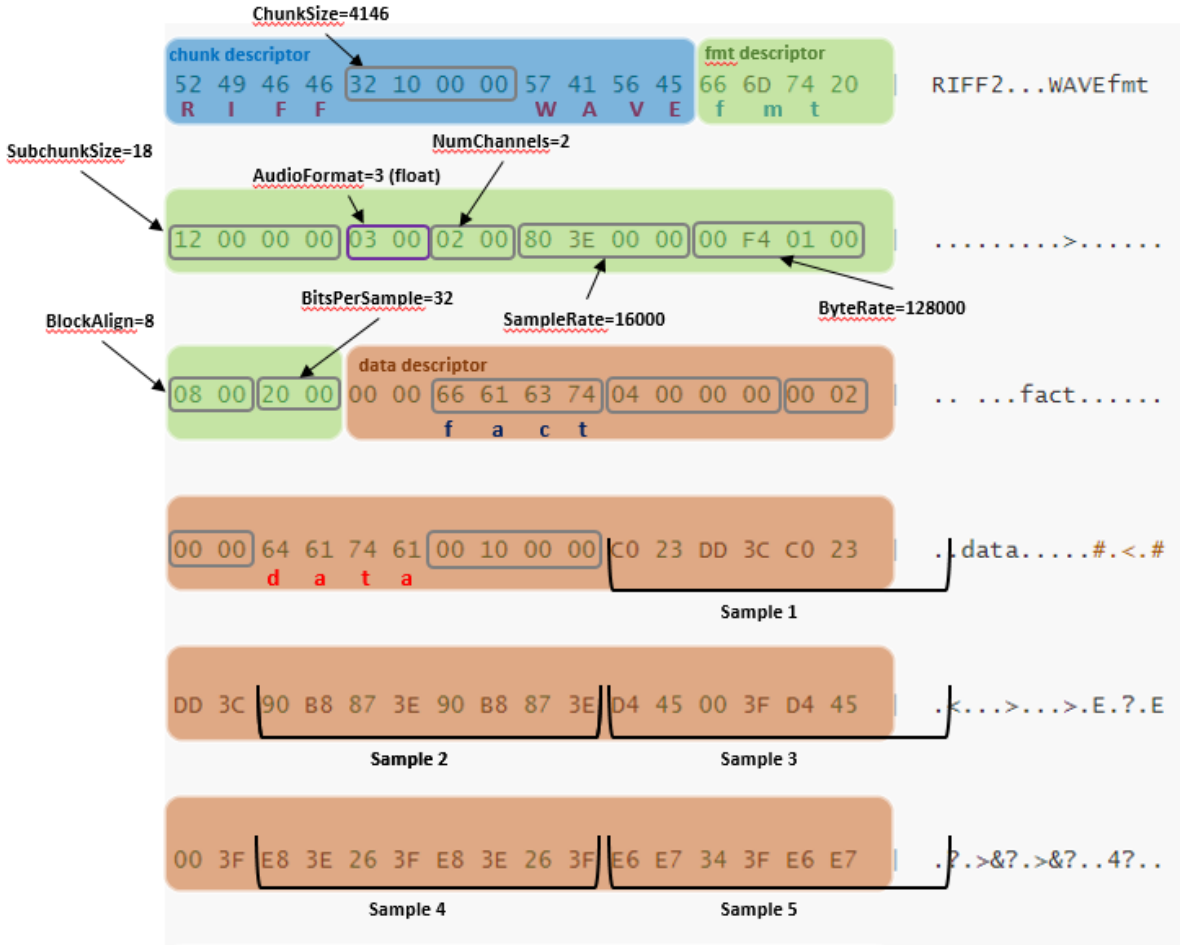
```
$ sox -n -e floating-point -r16k -c2 -b32 sine.wav synth 0.032 sine  1000.0

$ soxi sine.wav

Input File     : 'sine.wav'
Channels       : 2
Sample Rate    : 16000
Precision      : 25-bit
Duration       : 00:00:00.03 = 512 samples ~ 2.4 CDDA sectors
File Size      : 4.15k
Bit Rate       : 1.04M
Sample Encoding: 32-bit Floating Point PCM

$ hexdump -C -n256 sine.wav
00000000  52 49 46 46 32 10 00 00  57 41 56 45 66 6d 74 20  |RIFF2...WAVEfmt |
00000010  12 00 00 00 03 00 02 00  80 3e 00 00 00 f4 01 00  |.........>......|
00000020  08 00 20 00 00 00 66 61  63 74 04 00 00 00 00 02  |.. ...fact......|
00000030  00 00 64 61 74 61 00 10  00 00 c0 23 dd 3c c0 23  |..data.....#.<.#|
00000040  dd 3c 90 b8 87 3e 90 b8  87 3e d4 45 00 3f d4 45  |.<...>...>.E.?.E|
00000050  00 3f e8 3e 26 3f e8 3e  26 3f e6 e7 34 3f e6 e7  |.?.>&?.>&?..4?..|
00000060  34 3f 93 5b 26 3f 93 5b  26 3f 8e f1 ff 3e 8e f1  |4?.[&?.[&?...>..|
00000070  ff 3e dc 7a 89 3e dc 7a  89 3e 00 b2 9a 3a 00 b2  |.>.z.>.z.>...:..|
00000080  9a 3a 94 b0 8a be 94 b0  8a be 82 ba fe be 82 ba  |.:..............|
00000090  fe be e9 f8 26 bf e9 f8  26 bf f4 45 34 bf f4 45  |....&...&..E4..E|
000000a0  34 bf c6 ec 26 bf c6 ec  26 bf e2 ea fe be e2 ea  |4...&...&.......|
000000b0  fe be 62 68 8a be 62 68  8a be 00 00 ec 39 00 00  |..bh..bh.....9..|
000000c0  ec 39 e2 f1 89 3e e2 f1  89 3e e0 62 ff 3e e0 62  |.9...>...>.b.>.b|
000000d0  ff 3e 8c af 26 3f 8c af  26 3f e1 84 34 3f e1 84  |.>..&?..&?..4?..|
000000e0  34 3f d7 b7 26 3f d7 b7  26 3f d0 41 ff 3e d0 41  |4?..&?..&?.A.>.A|
```

```
000000f0  ff 3e 3e 23 8a 3e 3e 23  8a 3e 00 00 4c b8 00 00  |.>>#.>>#.>..L...|
```



（1）“52 49 46 46”这个是Ascii字符“**RIFF**”，这部分是固定格式，表明这是一个**WAVE**文件头。
（2）“32 10 00 00”，这个是我这个**WAVE**文件的数据大小，这个大小包括除了前面4个字节的所有字节，也就等于文件总字节数减去8。16进制的“32 10 00 00”对应是十进制的“4146”。
（3）“57 41 56 45 66 6D 74 20”，也是Ascii字符“**WAVEfmt**”，这部分是固定格式。
以后是**PCMWAVEFORMAT**部分

（4）“12 00 00 00”，这是一个DWORD，对应数字18，这个对应定义中的PCMWAVEFORMAT部分的大小，可以看到后面的这个段内容正好是18个字节。一般情况下大小为16，此时最后附加信息没有，上面这个文件多了两个字节的附加信息。
（5）“03 00”，这是一个WORD，对应定义为编码格式（IEEE float）。
（6）“02 00”，这是一个WORD，对应数字2，表示声道数为2，是双声道**Wav**。
（7）“80 3E 00 00”对应数字16000，代表的是采样频率16000，采样率（每秒样本数），表示每个通道的播放速度
（8）“00 F4 01 00”对应数字128000，代表的是每秒的数据量，波形音频数据传送速率，其值为通道数×每秒样本数×每样本的数据位数 / 8（(2*1600*32)/8）。播放软件利用此值可以估计缓冲区的大小。
（9）“08 00”对应数字是8，表示块对齐的内容。数据块的调整数（按字节算的），其值为通道数×每样本的数据位值 / 8。播放软件需要一次处理多个该值大小的字节数据，以便将其值用于缓冲区的调整。
（10）“20 00”数值为32，采样大小为32Bits，每样本的数据位数，表示每个声道中各个样本的数据位数。如果有多个声道，对每个声道而言，样本大小都一样。

（11）“00 00”此处为附加信息（可选），和（4）中的size对应。

（12）“66 61 73 74” Fact是可选字段，一般当wav文件由某些软件转化而成，则包含该项，“04 00 00 00”Fact字段的大小为4字节，“00 02 00 00”是fact数据。

（13）“64 61 74 61”，这个是Ascii字符“data”，标示头结束，开始数据区域。
（14）“00 10 00 00”十六进制数是“0x00001000”,对应十进制4096，是数据区的开头，以后数据总数，看一下前面正好可以看到，文件大小是4154，从（2）到（13）包括（13）正好是4154-4096=50字节。

参考文档

[1] WAVE PCM soundfile format

[2] wav文件格式分析与详解