

# WAV 文件格式实例分析

合肥电子工程学院通信与通抗原理教研室(230037) 徐济仁 牛纪海 陈家松

摘 要：分析 WAV 文件格式 ,给出 WAV 文件格式的解析实例。

关键词：WAV 文件 声音文件 格式码 chunk

WAV 文件格式是一种重要的用于存放声音的文件格式。尽管现在有 MP3、RAM 等压缩效率更高的声音文件格式 ,并且广泛被音乐文件所采用 ,但是有很多应用程序仍然采用 WAV 文件格式。由于 WAV 文件没有采用压缩技术 ,因此它的文件很庞大 ,一般都在几 MB 以上。也正因为没有采用压缩技术 ,WAV 文件中声音的采样数据很容易被读出来 ,便于做其它处理。例如 :画出声音的信号波形、作出频谱等。现在的应用程序几乎都支持 WAV 文件格式 ,也有专门软件可以完成从 WAV 文件格式向其它文件格式的转换 ,因此 WAV 文件在目前仍然有着广泛的应用价值。

## 1 RIFF 文件和 WAV 文件格式

### (1)RIFF 文件格式

在 Windows 环境下 ,大部分多媒体文件都按照某种结构来存放信息 ,这种结构称为资源互换文件格式(Resources Interchange File Format ,RIFF)。例如声音的 WAV 文件、视频的 AVI 文件等都是由此结构衍生出来的。RIFF 可以看做是一种树状结构 ,其基本构成单位为 chunk ,犹如树状结构中的节点 ,每个 chunk 由辨别码、数据大小及数据组成。

辨别码由 4 个 ASCII 码构成。数据大小则标示出紧跟其后数据的长度(单位为 Byte) ,而数据大小本身也用掉 4B ,因此实际上 1 个 chunk 的长度为数据大小加 8B。一般 ,chunk 本身并不允许内部再包含 chunk ,但有 2 种例外 ,分别为以 RIFF 及 LIST 为辨别码的 chunk。针对此二种 chunk ,RIFF 又从原先的数据中切出 4B ,此 4B 称为格式辨别码。然而 RIFF 又规定文件中仅能有 1 个以 RIFF 为辨别码的 chunk。

凡是依循此结构的文件 ,称之为 RIFF 文档。该结构提供了系统化的分类。如果和 MS-DOS 文件系统作比较 ,RIFF chunk 就好比是硬盘的根目录 ,其格式辨别码就是此硬盘的逻辑代码(C :或 D :) ,LIST chunk 为其下的子目录 ,其它的 chunk 则为一般的文件。微软为 RIFF 文件的处理提供了相关的函数。视窗下的各种多媒体文件格式

就如同在磁盘机下规定了能放怎样的目录以及在该目录下仅能放何种数据。

### (2)WAV 文件格式

声音文件也称为 WAV 或 WAVEFORM 文件 ,其结构如图 1 所示。RIFF 的格式辨别码为 WAVE。整个文件由 2 个 chunk 组成 :辨别码“fmt ”

(注意 ,最后一个是空白字符)和“data ”。

在 fmt 的 chunk 下包含了一个 PCMWAVEFORMAT 数据结构 ,其定义如下 :

```
typedef struct pcmwaveformat-tag {
    WAVEFORMAT wf ;
    WORD wBitsPerSample ;
} PCMWAVEFORMAT ;
typedef struct waveform-tag {
    WORD wFormatTag ;
    WORD nChannels ;
    DWORD nSamplesPerSec ;
    DWORD nAvgBytesPerSec ;
    WORD nBlockAlign ;
} WAVEFORMAT ;
```

其意义分别为 :

① wFormatTag :记录此声音的格式代号 ,例如 WAVE\_FORMAT\_PCM、WAVE\_FORAM\_ADPCM 等。

② nChannels :记录声音的频道数。

③ nSamplesPerSec :记录每秒取样数。

④ nAvgBytesPerSec :记录每秒的数据量。

⑤ nBlockAlign :记录区块的对齐单位。

⑥ wBitsPerSample :记录每个取样所需的位数。

data chunk 包含真正的声音数据。Windows 目前仅提供 WAVE\_FORMAT\_PCM 一种数据格式 ,所代表的意义是脉冲编码调制(Pulse Code Modulation)。针对此格式 ,Windows 定义了 data chunk 中数据的存放情形。图 2

" RIFF "
xxxx
" WAVE "
" fmt "
sizeof(PCMWAVEFORMAT)
struct of PCMWAVEFORMAT
" data "
xxxx
wave form data

图 1 WAV 文件结构

中列出了 4 种不同频道数、取样所需的位元数以及位元位置的安排。

频道 0	频道 0	频道 0	频道 0
nChannels=1 ,wBitsPerSample=8			
频道 0(左)	频道 1(右)	频道 0(左)	频道 1(右)
nChannels=2 ,wBitsPerSample=8			
频道 0(低位)	频道 0(高位)	频道 1(低位)	频道 1(高位)
nChannels=1 ,wBitsPerSample=16			
频道 0(低位)	频道 0(高位)	频道 1(低位)	频道 1(高位)
nChannels=2 ,wBitsPerSample=16			

图 2 PCM 文件中位元安排方式

第 1 排表示单声道 8 位元 ,第 2 排表示双声道 8 位元 ,第 3 排表示单声道 16 位元 ,第 4 排表示双声道 16 位元。8 位元代表音量大小由 8 个位元所表示 ,16 位元则代表音量大小由 16 个位元所表示。理论上 8 位元可以表示 0~255 ,16 位元可表示 0~65 536 ,但是 Windows 规定 16 位元其值的范围为-32 168~32 167。此外要注意 ,0 并不一定代表无声 ,而是由中间的数值来决定 ,即在 8 位元时为 128 ,16 位元时为 0 才是无声。因此 ,若程序设计时需放入无声的数据 ,要特别注意声音格式是 16 位元还是 8 位元。

## 2 WAV 文件的解析实例

点击“程序/MS-DOS 方式”,进入 MS-DOS 工作方式 ,调用 DOS 应用程序 DEBUG ,查看一个具体的 WAV 文件(例如 :TEST.WAV)。文件中的内容如下。

```
1C6C:0000 52 49 46 46 84 01 00 00 -57 41 56 45 66 6D 74 20
1C6C:0010 12 00 00 00 01 00 01 00 -22 56 00 00 22 56 00 00
1C6C:0020 01 00 08 00 28 00 66 61 -63 74 04 00 00 00 52 01
1C6C:0030 00 00 64 61 74 61 52 01 -00 00 3D 42 46 4A 4D 51
1C6C:0040 55 5A 5D 61 64 68 6B 6E -73 78 7C 81 85 89 8E 94
1C6C:0050 97 9B 9F A3 A6 A9 AD B1 -B4 B8 BD C3 C6 C9 C8 C8
1C6C:0060 C8 C8 C7 C6 C3 C1 BD B9 -B6 B3 AD A8 A1 9B 96 92
1C6C:0070 8C 86 7D 74 6E 68 65 62 -5B 54 4E 49 43 3E 3B 39
1C6C:0080 37 35 34 34 35 36 37 39 -3E 43 48 4D 51 56 59 5D
1C6C:0090 61 65 68 6C 70 74 79 7F -85 8B 8F 94 95 96 9B A1
1C6C:00A0 A3 A6 A9 AC AD AE AF B1 -B6 BB BE C2 C3 C5 C7 CA
1C6C:00B0 C8 C7 C1 BB B7 B3 AF AB -A6 A2 9D 99 94 8F 86 7D
1C6C:00C0 74 6B 64 5D 57 52 4F 4D -47 42 3C 36 36 37 37 37
1C6C:00D0 37 38 39 3B 3C 3D 42 48 -4D 53 57 5C 61 66 6C 73
1C6C:00E0 79 7F 82 85 8D 95 9A A0 -A2 A5 A8 AC AF B3 B6 BA
1C6C:00F0 B9 B9 B5 B2 AE AB A9 A7 -A4 A1 A0 A0 9F 9E 9B 98
1C6C:0100 97 97 94 92 90 8F 8E 8D -8A 88 87 86 83 80 7B 77
1C6C:0101 70 6A 64 5E 5A 57 53 50 -4A 45 41 3E 3E 3E 43 48
1C6C:0102 48 49 4D 51 58 5F 5F 60 -62 65 6C 74 78 7D 81 85
1C6C:0103 8B 91 97 9E A0 A2 A8 AF -B7 BF BF C0 C2 C5 C6 C8
1C6C:0104 C5 C2 BB B5 B1 A0 A4 9C -94 8D 88 83 7E 79 75 71
1C6C:0105 6D 69 67 65 63 62 61 60 -61 63 68 6D 6F 72 71 70
1C6C:0106 70 71 70 6F 6C 69 65 62 -61 60 5E 5D 5D 5B 59
1C6C:0107 5A 5C 62 68 6C 71 74 78 -7D 82 87 8C 8B 8A 90 97
1C6C:0108 9B A0 A1 A2 A3 A4 A4 A5 -A5 A5 A6 A8
```

上述文件长度为 018CH(十六进制)。以此文件为例 ,按照从上到下 ,从左到右的顺序来分析它的含义。

《微型机与应用》2002 年第 3 期

52 49 46 46 :对应的 ASCII 码为 RIFF ,表示 chunk 的辨别码为 RIFF。

00 00 01 84 :表示辨别码为 RIFF 的 chunk 的数据长度为 0000 0184H。

57 41 56 45 :对应的 ASCII 码为 WAVE。

以下为 3 个 chunk ,辨别码分别为 fmt(注意 t 后面有 1 个空格)、fact 和 data。

66 6D 74 20 :对应的 ASCII 码为 fmt(注意 t 后面有 1 个空格) ,表示 chunk 的辨别码为 fmt(注意 t 后面有 1 个空格)。

00 00 00 12 :表示辨别码为 fmt 的 chunk 的数据长度为 0000 0184H。

00 01 :记录此声音的格式代号 ,表示它的格式为 WAVE\_FORMAT\_PCM。

00 01 :记录声音的频道数 ,频道数为 1。

00 00 56 22 :记录每秒取样数。每秒取样数为 0000 5622H。

00 00 56 22 :记录每秒的数据量。每秒的数据量为 0000 5622H。

00 01 :记录区块的对齐单位为 1。

00 08 :记录每个取样所需的位元数 ,它的值为 8 ,表示需要用 1B。

00 28 :为备用字。

66 61 63 74 :对应的 ASCII 码为 fact ,表示 chunk 的辨别码为 fact。

00 00 00 04 :表示辨别码为 fact 的 chunk 的数据长度为 4B。

00 00 01 52 :表示真实的语音采样数据量的大小 ,长度为 0000 0152H。

64 61 74 61 :对应的 ASCII 码为 data ,表示 chunk 的辨别码为 data。

00 00 01 52 :表示辨别码为 data 的 chunk 的数据长度为 0000 0152H 个字节。

3D 42 46 4A 4D 51 55 5A...9B A0 A1 A2 A3 A4 A4 A5-A5 A5 A6 A8(图 3 中最末一行)

从 3D 42 46 至文件结束表示为真实的采样数据。

有的 WAV 文件在实际应用中 ,可能适当增加了一些 chunk ,稍微有一些变化 ,但分析方法基本与上面类似。

## 参考文献

- 1 黄伟伦.MS-WINDOWS 多媒体程序设计实务与范例.武汉 :华中理工大学出版社 ,1993
- 2 林宝成.汉语讲话者识别方法.模式识别与人工智能 ,1998 ;12(1)

(收稿日期 :2001-09-10)