

WAVE 文件格式分析实验

一、实验目的

了解 Wave 文件格式并学习 Windows 下用低级函数来播放 Wave 文件。

二、实验原理

1、Windows 中的低级多媒体函数

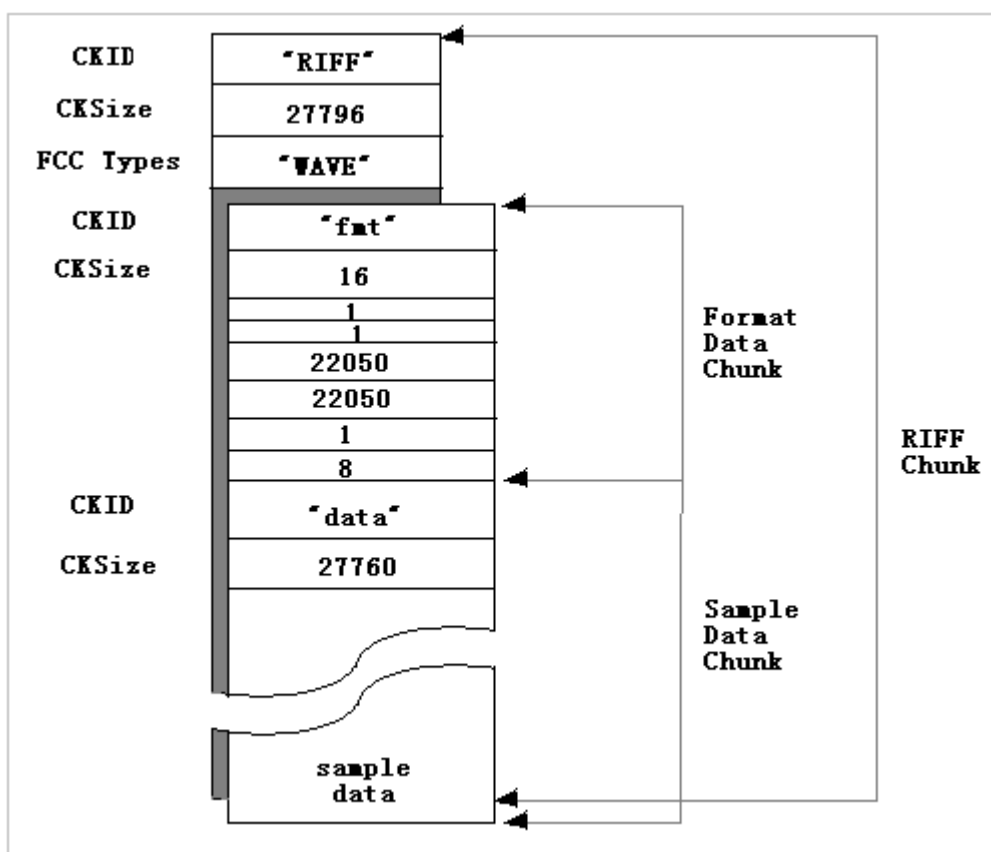
Windows 中的低级多媒体函数都是以 mmio 开头的，在 VC 的联机帮助中，可以找到以下函数的具体含义和使用方法。

mmioAdvance
mmioAscend
mmioClose
mmioCreateChunk
mmioDescend
mmioFlush
mmioGetInfo
mmioInstallIOProc
mmioOpen
mmioRead
mmioRename
mmioSeek
mmioSendMessage
mmioSetBuffer
mmioSetInfo
mmioStringToFOURCC
mmioWrite

事实上我们只用到 5 个函数：mmioAscend、mmioClose、mmioOpen、mmioRead、mmioDescend

2、Wave 文件的结构

Wave 文件的数据以几种方式出现：抽样速率的各种综合、多种声道（单声和立体声）和不同频率（每个样本的位数）。这些信息以一种格式块的形式出现-在 RIFF 专门名词中称为（chunk）-位于每个 WAVE 文件的开头附近。下图是一个 RIFF 文件结构的示意图：



一个 WAVE 文件最少包含 3 个块，RIFF 块是其中最大的。整个 WAVE 文件就是一个 RIFF 块。Cksize 紧跟在“RIFF”CKID 之后出现，它包含一个值，等于文件大小减去 8 个字节-这 8 个字节用来存储 RIFF 块的 CKID 和 CKSIZE。第二和第三块称为子块，包含在 RIFF 块之中。这些子块的第一个是“fmt”块，包含 PCMWAVEFORMAT 结构所需要的信息；第二个子块“data”块是文件的最大部分，紧跟在“fmt”块后，包含所有的数据波形。RIFF 块的 CKSIZE 等于“fmt”块和“data”块占用的字节之和。

为了读写 RIFF 文件，用户使用为多媒体块信息准备的叫 MMCKINFO 的标准数据结构。在 VC 中，这个结构定义为：

```
typedef struct {
    FOURCC ckid;
    DWORD  cksize;
    FOURCC fccType;
    DWORD  dwDataOffset;
```

```
DWORD dwFlags;  
} MMCKINFO;
```

三、使用示例

实验示例 sb2.exe 的主要步骤为：

```
MMCKINFO MMckInfoParent;  
MMCKINFO MMckInfoChild;  
//打开一个 WAVE 文件  
HMMIO hmmio = mmioOpen(FileNameAndPath.GetBuffer(80),NULL,MMIO_READ);  
//定父块，就是 RIFF 块的位置  
MMckInfoParent.fccType = mmioFOURCC('W','A','V','E');  
mmioDescend(hmmio,&MMckInfoParent,NULL,MMIO_FINDRIFF);  
//读如格式块内容  
MMckInfoChild.ckid = mmioFOURCC('f','m','t',' ');  
mmioDescend(hmmio,&MMckInfoChild,&MMckInfoParent,MMIO_FINDCHUNK);  
mmioRead(hmmio,(LPSTR)&PCMWaveFmtRecord,MMckInfoChild.cksize);  
HWAVEOUT hWaveOut;  
errorCode = waveOutOpen(  
    &hWaveOut,  
    WAVE_MAPPER,  
    (WAVEFORMATEX*)&PCMWaveFmtRecord,  
    0L,  
    0L,  
    WAVE_FORMAT_QUERY);  
mmioAscend(hmmio,&MMckInfoChild,0);  
//读如 WAVE 数据块，并返回所分配内存的指针  
MMckInfoChild.ckid = mmioFOURCC('d','a','t','a');  
mmioDescend(hmmio,&MMckInfoChild,&MMckInfoParent,MMIO_FINDCHUNK);  
long lDataSize = MMckInfoChild.cksize;  
HANDLE waveDataBlock = ::GlobalAlloc(GMEM_MOVEABLE,lDataSize);  
LPBYTE pWave = (LPBYTE)::GlobalLock(waveDataBlock);  
WaveHeader.lpData = (LPSTR)pWave;  
WaveHeader.dwBufferLength = lDataSize;  
WaveHeader.dwFlags = 0L;  
WaveHeader.dwLoops = 0L;  
mmioClose(hmmio,0);  
return waveDataBlock;
```

这样，我们从 RIFF 文件中抽取了数据并创建了三个基本数据结构（格式记录、波音频数据、WAVE 数据头）。从内存中播放波音频需要 5 个步骤：打开波音频设备；生成 WAVE 文件头；将数据写入设备；取消 WAVE 文件头；关闭设备。

```
HWAVEOUT hWaveOut;
MMRESULT ReturnCode = waveOutOpen(
    &hWaveOut,
    WAVE_MAPPER,
    (WAVEFORMATEX*)&PCMWaveFmtRecord,
    0l,
    0l,
    0l);
ReturnCode = waveOutPrepareHeader(
    hWaveOut,
    &WaveHeader,
    sizeof(WaveHeader));
ReturnCode = waveOutWrite(
    hWaveOut,
    &WaveHeader,
    sizeof(WaveHeader));
do {}
while(!(WaveHeader.dwFlags & WHDR_DONE));
ReturnCode = waveOutUnprepareHeader(hWaveOut,
    &WaveHeader,
    sizeof(WaveHeader));
WaveHeader.dwFlags = 0l;
ReturnCode = waveOutClose(hWaveOut);
```

四、实验要求

请同学们认真阅读实验原理部分，最好能够提前了解一下 WAVE 文件的格式。然后参考给出的样板程序 sb2.exe 思路编制自己的程序播放 WAVE 文件。

在意思清楚、完整的前提下，思考题的回答越精炼越好。

五、思考题

- 1 WAVE 文件保存有那些信息，以什么格式存储？
- 2 在同一个程序中能不能同时进行播放和录制两项操作？为什么？这对声卡有什么要求？
- 3 资源交换文件格式（RIFF）有什么特点？