AVI是音频视频交错(Audio Video Interleaved)的英文缩写，它是Microsoft公司开发的一种符合RIFF文件规范的数字音频与视频文件格式，原先用于Microsoft Video for Windows (简称VFW)环境，现在已被Windows 95/98、OS/2 等多数操作系统直接支持。AVI格式允许视频和音频交错在一起同步播放，支持 256 色和RLE压缩，但AVI文件并未限定压缩标准， 因此，AVI文件格式只是作为控制界面上的标准，不具有兼容性，用不同压缩算法生成的AVI文件，必须使用相应的解压缩算法才能播放出来。常用的AVI播 放驱动程序，主要是Microsoft Video for Windows或Windows 95/98 中的Video 1，以及Intel公司的Indeo Video。

在介绍AVI文件前，我们要先来看看RIFF文件结构。AVI文件采用的是RIFF文件结构方式，RIFF（Resource Interchange File Format，资源互换文件格式）是微软公司定义的一种用于管理windows环境中多媒体数据的文件格式，波形音频wave，MIDI和数字视频AVI 都采用这种格式存储。构造RIFF文件的基本单元叫做数据块（Chunk），每个数据块包含 3 个部分，

1、4 字节的数据块标记（或者叫做数据块的ID）

2、数据块的大小

3、数据

整个RIFF文件可以看成一个数据块，其数据块ID为RIFF，称为RIFF块。一个RIFF文件中只允许存在一个RIFF块。RIFF块中包含一系列 的子块，其中有一种字块的ID为"LIST"，称为LIST，LIST块中可以再包含一系列的子块，但除了LIST块外的其他所有的子块都不能再包含子 块。

RIFF和LIST块分别比普通的数据块多一个被称为形式类型（Form Type）和列表类型（List Type）的数据域，其组成如下：

1、4 字节的数据块标记（Chunk ID）

2、数据块的大小

3、4 字节的形式类型或者列表类型

4、数据

下面我们看看AVI文件的结构。AVI文件是目前使用的最复杂的RIFF文件，它能同时存储同步表现的音频视频数据。AVI的RIFF块的形式类型是AVI，它包含 3 个子块，如下所述：

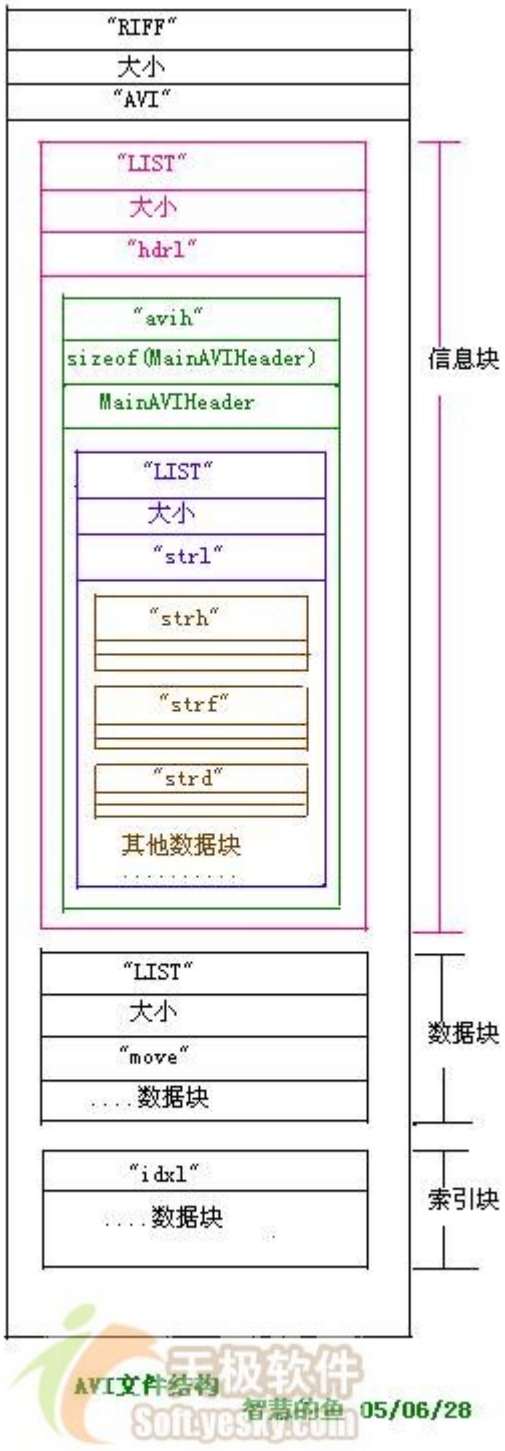
1、信息块，一个ID为"hdrl"的LIST块，定义AVI文件的数据格式。

2、数据块，一个ID为 "movi"的LIST块，包含AVI的音视频序列数据。

3、索引块，ID为 "idxl"的子块，定义 "movi"LIST块的索引数据，是可选块。

AVI文件的结构如下图所示，下面将具体介绍AVI文件的各子块构造。

1、信息块，信息块包含两个子块，即一个ID为 avih 的子块和一个ID 为 strl 的LIST块。



"avih"子块的内容可由如下的结构定义：

```
typedef struct
{
```

```
  DWORD dwMicroSecPerFrame ; //显示每桢所需的时间 ns，定义 avi 的显示速率
  DWORD dwMaxBytesPerSec; // 最大的数据传输率
  DWORD dwPaddingGranularity; //记录块的长度需为此值的倍数，通常是 2048
  DWORD dwFlages; //AVI 文件的特殊属性，如是否包含索引块，音视频数据是否交叉存储
  DWORD dwTotalFrame; //文件中的总桢数
  DWORD dwInitialFrames; //说明在开始播放前需要多少桢
  DWORD dwStreams; //文件中包含的数据流种类
  DWORD dwSuggestedBufferSize; //建议使用的缓冲区的大小，
  //通常为存储一桢图像以及同步声音所需要的数据之和
  DWORD dwWidth; //图像宽
  DWORD dwHeight; //图像高
  DWORD dwReserved[4]; //保留值
}MainAVIHeader;
```

"strl" LIST 块用于记录 AVI 数据流，每一种数据流都在该 LIST 块中占有 3 个子块，他们的 ID 分别是"strh","strf", "strd"；

"strh"子块由如下结构定义。

```
typedef struct
{
  FOURCC fccType; //4 字节，表示数据流的种类 vids 表示视频数据流
  //auds 音频数据流
  FOURCC fccHandler;//4 字节 ，表示数据流解压缩的驱动程序代号
  DWORD dwFlags; //数据流属性
  WORD wPriority; //此数据流的播放优先级
  WORD wLanguage; //音频的语言代号
  DWORD dwInitalFrames;//说明在开始播放前需要多少桢
  DWORD dwScale; //数据量，视频每桢的大小或者音频的采样大小
  DWORD dwRate; //dwScale /dwRate = 每秒的采样数
```

```
  DWORD dwStart; //数据流开始播放的位置，以 dwScale 为单位
  DWORD dwLength; //数据流的数据量，以 dwScale 为单位
  DWORD dwSuggestedBufferSize; //建议缓冲区的大小
  DWORD dwQuality; //解压缩质量参数，值越大，质量越好
  DWORD dwSampleSize; //音频的采样大小
  RECT rcFrame; //视频图像所占的矩形
}AVIStreamHeader;
```

"strf"子块紧跟在"strh"子块之后，其结构视"strh"子块的类型而定，如下所述；如果 strh 子块是视频数据流，则 strf 子块的内容是一个与 windows 设备无关位图的 BIMAPINFO 结构，如下：

```
typedef struct tagBITMAPINFO
{
  BITMAPINFOHEADER bmiHeader;
  RGBQUAD bmiColors[1]; //颜色表
}BITMAPINFO;

typedef struct tagBITMAPINFOHEADER
{
  DWORD biSize;
  LONG biWidth;
  LONG biHeight;
  WORD biPlanes;
  WORD biBitCount;
  DWORD biCompression;
  DWORD biSizeImage;
  LONG biXPelsPerMeter;
  LONG biYPelsPerMeter;
```

```
  DWORD biClrUsed;
  DWORD biClrImportant;
}BITMAPINFOHEADER;
```

如果 strh 子块是音频数据流，则 strf 子块的内容是一个 WAVEFORMAT 结构，如下：

```
typedef struct
{
  WORD wFormatTag;
  WORD nChannels; //声道数
  DWORD nSamplesPerSec; //采样率
  DWORD nAvgBytesPerSec; //WAVE 声音中每秒的数据量
  WORD nBlockAlign; //数据块的对齐标志
  WORD biSize; //此结构的大小
}WAVEFORMAT
```

"strd"子块紧跟在 strf 子块后，存储供压缩驱动程序使用的参数，不一定存在，也没有固定的结构。

"strl" LIST 块定义的 AVI 数据流依次将 "hdrl " LIST 块中的数据流头结构与"movi" LIST 块中的数据联系在一起，第一个数据流头结构用于数据流 0，第二个用于数据流 1，依次类推。

数据块中存储视频和音频数据流，数据可直接存于 "movi" LIST 块中。数据块中音视频数据按不同的字块存放，其结构如下所述，

音频字块

"##wb"

Wave 数据流

视频子块中存储 DIB 数据，又分为压缩或者未压缩 DIB，

"##db"

RGB 数据流

"##dc"

压缩的图像数据流

看到了吧，avi 文件的图像数据可以是压缩的，和非压缩格式的。对于压缩格式来说，也可采用不同的编码，也许你曾经遇到有些 avi 没法识别，就是因为编 码方式不一样，如果没有相应的解码，你就没法识别视频数据。AVI 的编码方式有很多种，比较常见的有 mpeg2，mpeg4，divx 等。

索引块，索引快包含数据块在文件中的位置索引，能提高 avi 文件的读写速度，其中存放着一组 AVIINDEXENTRY 结构数据。如下，这个块并不是必需的，也许不存在。

```
typedef struct
{
  DWORD ckid; //记录数据块中子块的标记
  DWORD dwFlags; //表示 chid 所指子块的属性
  DWORD dwChunkOffset; //子块的相对位置
  DWORD dwChunkLength; //子块长度
};
```