mp3 每帧均为1152个字节， 则：

frame\_duration = 1152 \* 1000 / sample\_rate

例如：sample\_rate = 44100HZ时，计算出的时长为26.122ms，这就是经常听到的mp3每帧播放时间固定为26ms的由来。

MP3简介

MP3是MPEG-1 Audio Layer 3的简称，是当今比较流行的一种数字音频编码和有损压缩格式（有Layer 3，也必然有Layer1和Layer2，也就是MP1和MP2，但不在本文讨论范围之内）。MP3技术的应该可以用来大幅度的降低音频文件存储所需要的空间。它丢掉脉冲编码调制（PCM）音频数据中对人类听觉不重要得数据，从而达到了较高的压缩比（高达12：1－10：1）。简单地说，MP3在编码时先对音频文件进行频谱分析，然后用过滤器滤掉噪音电平，接着通过量化的方式将剩下的每一位打散排列，最后形成有较高压缩比的MP3文件，并使压缩后的文件在回放时也能够达到比较接近原音源的效果。  
  
MP3的音频质量取决于它的Bitrate和Sampling frequency，以及编码器质量。MP3的典型速度介于每秒128到320kb之间。采样频率也有44.1，48和32 kHz三种频率，比较常见的是采用CD采样频率——44.1kHz。常用的编码器是LAME，它完全遵循LGPL的MP3编码器，有着良好的速度和音质。

MP3文件格式

MP3文件是由帧（frame）构成，帧是MP3文件的最小组成单位。每帧都包含帧头，并可以计算帧的长度。根据帧的性质不同，文件主要分为三个部分，ID3v2标签帧，数据帧和ID3v1标签帧。并非每个MP3文件都有ID3v2，但是数据帧和ID3v1帧是必须的。ID3v2在文件头，以字符串“ID3”为标志，包含了演唱者，作曲，专辑等信息，长度不固定，扩展了ID3V1的信息量。ID3v1在文件结尾，以字符串“TAG”为标记，其长度是固定的128个字节，包含了演唱者、歌名、专辑、年份等信息。

ID3V2

ID3V2到现在一共有四个版本，但流行的播放软件一般只支持第三版，既ID3V2.3。每个ID3V2.3 的标签都一个标签头和若干个标签帧或一个扩展标签头组成。关于曲目的信息如标题、作者等都存放在不同的标签帧中，扩展标签头和标签帧并不是必要的，但每个标签至少要有一个标签帧。标签头和标签帧一起顺序存放在MP3 文件的首部。

标签头

长度为10个字节，位于文件首部，其数据结构如下：

char Header[3]; /\* 字符串 "ID3" \*/

char Ver; /\* 版本号ID3V2.3 就记录3 \*/

char Revision; /\* 副版本号此版本记录为0 \*/

char Flag; /\* 存放标志的字节，这个版本只定义了三位，很少用到，可以忽略 \*/

char Size[4]; /\* 标签大小，除了标签头的10 个字节的标签帧的大小 \*/

标签大小为四个字节，但每个字节只用低7位，最高位不使用，恒为0，其格式如下：  
0xxxxxxx 0xxxxxxx 0xxxxxxx 0xxxxxxx

计算公式如下：

ID3V2\_frame\_size = (int)(Size[0] & 0x7F) << 21  
 | (int)(Size[1] & 0x7F) << 14  
 | (int)(Size[2] & 0x7F) << 7  
 | (int)(Size[3] & 0x7F) + 10;

标签帧

每个标签帧都有一个10字节的帧头和至少一个字节的不固定长度的内容组成。它们是顺序存放在文件中，由各自特定的标签头来标记帧的开始。其帧的结构如下：

char FrameID[4]; /\*用四个字符标识一个帧，说明其内容 \*/

char Size[4]; /\* 帧内容的大小，不包括帧头，不得小于1 \*/

char Flags[2];/\* 存放标志，只定义了6 位，此处不再说明 \*/

常用帧标识：

TIT2：标题  
TPE1：作者  
TALB：专辑  
TRCK： 音轨，格式：N/M，N表示专辑中第几首，M为专辑中歌曲总数  
TYER：年份  
TCON：类型  
COMM：备注，格式：“eng/0备注内容”，其中eng表示所使用的语言  
帧大小为四个字节所表示的整数大小。

ID3V1

char Header[3]; /\* 标签头必须是"TAG"否则认为没有标签 \*/  
char Title[30]; /\* 标题 \*/  
char Artist[30]; /\* 作者 \*/  
char Album[30]; /\* 专集 \*/  
char Year[4]; /\* 出品年代 \*/  
char Comment[28]; /\* 备注 \*/  
char reserve; /\* 保留 \*/  
char track; /\* 音轨 \*/  
char Genre; /\* 类型 \*/

其实，关于最后31个字节还存在另外一个版本，就是30个字节的Comment和一个字节的Genre。

有了上述的这些信息，我们就可以自己写代码，从MP3文件中抓取信息以及修改文件名了。但是，如果真的想写一个播放软件，还是需要读它的数据帧，并进行解码。

数据帧

数据帧往往有多个，至于有多少，由文件大小和帧大小来决定。每个帧都有一个四字节长的帧头，接下来可能有两个字节的CRC校验，其存在由帧头中的具体信息决定。接着就是帧的实体数据，也就是MAIN\_DATA了。

帧头结构如下：

位置 长度 描述  
（BIT） （BITS）  
————————————————————————————  
31－19 12 Frame sync(0xFFF)  
18/17 2 Layer, 00 – reserved, 01 – Layer III  
 10 – Layer II, 11 － Layer I  
16 1 protection\_bit, 0 意味着受CRC保护，帧头后面跟16位的CRC。  
15-12 4 bitrate\_index, 比特率  
11-10 2 sampling\_frequency, 00 – 44.1KHz, 01 – 48KHz  
 10 – 32 KHz,11 – 保留  
9 1 padding\_bit,1 意味着帧里包含padding位，仅当采样频率为44.1KHz时发生。  
8 1 private\_bit  
7－6 2 mode,00－stereo, 01-joint stereo(intensity stereo and/or ms\_stereo)  
 11- dual\_channel, 11 – single\_channel  
5-4 2 mode\_extension,在Layer III中表示使用了哪一种joint stereo编码方式。  
 ntensity\_stereo   ms\_stereo  
 00 off off  
 01 on off  
 10 off on  
 11 on on  
3 1 copyright,1 表示受版权保护。  
2 1 original，0表示该bitstream是一个copy，1表示是original.  
1-0 2 emphasis，表示会使用哪一种de-emphasis。  
 00 － no emphasis,01 – 50/15 microsec. Emphasis  
 10 – reserved,11 – CCITT J.17

1)无论帧长是多少，每帧的播放时间都是26ms

2)数据帧大小：

FrameSize = 144 \* Bitrate / SamplingRate + PaddingBit  
当144 \* Bitrate / SamplingRate不能被8整除，则加上相应的paddingBit.

MAIN\_DATA:

MP3的granule包含18 \* 32个subband采样。每个数据帧含有两个granule的数据，其内容结如下：  
- main\_data\_end pointer  
- side info for both granules (scfsi)  
- side info granule 1  
- side info granule 2  
- scalefactors and Huffman code data granule 1  
- scalefactors and Huffman code data granule 2

主要数据里包含了scalefactors, Huffman encoded data和ancillary information。其内容不再详叙，可以参考MP3 SPEC－IS0 11172-3 AUDIO PART。我们一般用的都是立体声，scfsi的长度为32个字节。

这里要解释的一个概念就是位流――bitstream。我们平常接触到的数据都是整数，最小的单位就是byte后者char。虽然我们也会用一个字节里的不同位来表示不同的含义，但总的来说，我们在出来数据的时候还是把它当作一个个字节看待。但对MP3这种数据格式来说，这是行不通的。在解码时，它的数据输入就是一个个比特流。其中一个或几个比特会是你的采样数据或者信息编码。你需要从整个MAIN\_DATA里提取你所需要的以BIT为单位的参数和输入信号，从而进行解码。所以我们需要一个子程序，getbit(n)，也就是从缓冲中提取所需要的位，并形成一个新的整数，作为我们的输出。

LAME标签帧

可是，当你真的打开一个MP3文件的时候，你会发现，很奇怪，很多时候第一个数据帧的帧头后面的32个字节居然都为0，这是为什么呢，这么奇怪的解码信息该如何解释？找到MP3 INFO TAG REV SPECIFICATION的网站，我才明白，原来第一帧并不是真正的数据帧，而是LAME编码的标志帧。

这里又要牵涉到两个概念：CBR和VBR。CBR表示比特率不变，也就是每帧的长度是一致的，它以字符串“INFO”为标记。VBR是Variable BitRate的简称，也就是每帧的比特率和帧的长度是变化的，它以字符串“Xing”为标记。同时，它还存放了MP3文件里帧的总个数，和100个字节的播放总时间分段的帧的INDEX，还有其他一些参数，这被称为Zone A，传统Xing VBR标签数据，共120个字节。

在二进制文本编辑器里我们还可看到一个字符串“LAME”，并且后面清楚地跟着版本号。这就是20个字节的Zone B初始LAME信息，表示该文件是用LAME编码技术。接下来一直到该帧结束就是Zone C－LAME标签。