

注：写了一系列的结构体的分析的文章，在这里列一个列表：

[FFMPEG结构体分析：AVFrame](#)

[FFMPEG结构体分析：AVFormatContext](#)

[FFMPEG结构体分析：AVCodecContext](#)

[FFMPEG结构体分析：AVIOContext](#)

[FFMPEG结构体分析：AVCodec](#)

[FFMPEG结构体分析：AVStream](#)

[FFMPEG结构体分析：AVPacket](#)

FFMPEG有几个最重要的结构体，包含了解协议，解封装，解码操作，此前已经进行过分析：

[FFMPEG中最关键的结构体之间的关系](#)

在此不再详述，其中AVPacket是存储压缩编码数据相关信息的结构体。本文将会详细分析一下该结构体里重要变量的含义和作用。

首先看一下结构体的定义（位于avcodec.h文件中）：

```

1.  /* 雷霄骅
2.   * 中国传媒大学/数字电视技术
3.   * leixiaohua1020@126.com
4.   *
5.   */
6.  typedef struct AVPacket {
7.      /**
8.       * Presentation timestamp in AVStream->time_base units; the time at which
9.       * the decompressed packet will be presented to the user.
10.      * Can be AV_NOPTS_VALUE if it is not stored in the file.
11.      * pts MUST be larger or equal to dts as presentation cannot happen before
12.      * decompression, unless one wants to view hex dumps. Some formats misuse
13.      * the terms dts and pts/cts to mean something different. Such timestamps
14.      * must be converted to true pts/dts before they are stored in AVPacket.
15.      */
16.      int64_t pts;
17.      /**
18.       * Decompression timestamp in AVStream->time_base units; the time at which
19.       * the packet is decompressed.
20.       * Can be AV_NOPTS_VALUE if it is not stored in the file.
21.       */
22.      int64_t dts;
23.      uint8_t *data;
24.      int size;
25.      int stream_index;
26.      /**
27.       * A combination of AV_PKT_FLAG values
28.       */
29.      int flags;
30.      /**
31.       * Additional packet data that can be provided by the container.
32.       * Packet can contain several types of side information.
33.       */
34.      struct {
35.          uint8_t *data;
36.          int size;
37.          enum AVPacketSideDataType type;
38.      } *side_data;
39.      int side_data_elems;
40.
41.      /**
42.       * Duration of this packet in AVStream->time_base units, 0 if unknown.
43.       * Equals next_pts - this_pts in presentation order.
44.       */
45.      int duration;
46.      void (*destruct)(struct AVPacket *);
47.      void *priv;
48.      int64_t pos;          ///< byte position in stream, -1 if unknown
49.
50.      /**
51.       * Time difference in AVStream->time_base units from the pts of this
52.       * packet to the point at which the output from the decoder has converged
53.       * independent from the availability of previous frames. That is, the
54.       * frames are virtually identical no matter if decoding started from
55.       * the very first frame or from this keyframe.
56.       * Is AV_NOPTS_VALUE if unknown.
57.       * This field is not the display duration of the current packet.
58.       * This field has no meaning if the packet does not have AV_PKT_FLAG_KEY
59.       * set.
60.       *
61.       * The purpose of this field is to allow seeking in streams that have no
62.       * keyframes in the conventional sense. It corresponds to the
63.       * recovery point SEI in H.264 and match_time_delta in NUT. It is also
64.       * essential for some types of subtitle streams to ensure that all
65.       * subtitles are correctly displayed after seeking.
66.       */
67.      int64_t convergence_duration;
68.  } AVPacket;

```

在AVPacket结构体中，重要的变量有以下几个：

`uint8_t *data`：压缩编码的数据。

例如对于H.264来说，1个AVPacket的data通常对应一个NAL。

注意：在这里只是对应，而不是一模一样。他们之间有微小的差别：[使用FFMPEG类库分离出多媒体文件中的H.264码流](#)

因此在使用FFMPEG进行视音频处理的时候，常常可以将得到的AVPacket的data数据直接写成文件，从而得到视音频的码流文件。

`int size`：data的大小

`int64_t pts`：显示时间戳

`int64_t dts`：解码时间戳

`int stream_index`：标识该AVPacket所属的视频/音频流。

这个结构体虽然比较简单，但是非常的常用。

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/14215755>

文章标签：[ffmpeg](#) [avpacket](#) [源代码](#) [解码](#) [结构体](#)

个人分类：[FFMPEG](#)

所属专栏：[FFmpeg](#)

此PDF由[spygg](#)生成,请尊重原作者版权!!!

我的邮箱:[liushidc@163.com](mailto:liushidc@163.com)