

2012-11-23 // FFMPEG: немного про time-base и PTS

В терминах FFMPEG time-base, это параметр, используя который можно перевести PTS (Presentation Timestamp) закодированного пакета (AVPacket) или раскодированного фрейма (AVFrame) в реальное время, выраженное в секундах (это же верно для DTS закодированного пакета). Представляет собой рациональное число типа AVRational.

А вот теперь интересности, связанные с ним.

Общие сведения

Внутри библиотек libavformat и libavcodec, входящих в состав FFMPEG time-base может храниться в двух контекстах:

- контекст потока (AVStream)
- и контекст кодека (AVCodecContext)

При этом временные метки (далее я буду говорить только про PTS, если потребуется сказать про DTS - отдельно обращаю внимание) могут храниться у AVFrame (поле pts) и AVPacket (аналогично, поле pts) тип всех временных меток (включая поле dts у AVPacket) - int64_t.

Отсюда возникает резонный вопрос: как соотносить time-base и временные метки.

Короткий ответ:

- time-base AVStream является основной для временных меток AVPacket
- time-base AVCodecContext является основой для временных меток AVFrame

Далее рассмотрим, когда и как мы должны устанавливать руками, а что и когда установится само.

Чтение и раскодирования файла

После того как создан и открыт AVFormatContext, прочитана информация о потоках картина получается такой:

- поле time_base каждого потока (AVStream) заполнено самой библиотекой, изменять вручную его крайне не желательно, даже не могу представить, какие последствия может принести данная операция.
- поле time_base созданных декодеров для каждого типа потока (AVCodecContext) так же заполняется библиотекой и изменять его не желательно

Логично было бы предположить, что временные метки для потока и его декодировщика должны быть одинаковы, но это не так. Как следствие временные метки пакета и раскодированного фрейма нельзя сравнивать напрямую. И если вы хотите установить, после раскодировки, PTS фрейма, это нужно делать при помощи функции av_rescale_q(...), примерно так:

```
frame.pts = av_rescale_q(packet.pts, packetTimeBase, frameTimeBase);
```

Ясно, что frameTimeBase берётся из AVCodecContext, а packetTimeBase из AVStream.

Стоит отметить так же то, что у некоторых пакетов поле pts имеет значение AV_NOPTS_VALUE, тогда как поле dts имеет корректное значение, тогда его и стоит брать для присваивания значению PTS фрейма. Суть такого поведения для меня не ясна.

Так же стоит отметить для звуковых семплов, после раскодировки значение PTS фрейма будет AV_NOPTS_VALUE и НИ В КОЕМ СЛУЧАЕ НЕ МЕНЯЙТЕ ЕГО НА ЧТО-ТО ЛИБО ДРУГОЕ, это приведёт в смятение микшер, как следствие, звука в результирующем файле вы не услышите. И снова, суть такого поведения для меня не ясна.

Кодировка и запись (микширование потоков)

При кодировании и микшировании мы сами добавляем потоки в выходной формат. При этом time-base мы должны задавать только для кодировщика (AVCodecContext) и делать это **обязательно**.

Поле time-base у потока (AVStream) можно устанавливать, а можно и нет, всё равно, после вызова avformat_write_header(...) оно будет сброшено и установлено в необходимое значение подходящее для данного типа контейнера. При этом, значение time-base из AVCodecContext будет использоваться как подсказка, если контейнер не определяет требований к значению time-base.

Соответственно, важно: даже при записи вы не можете гарантировать, что значения time-base в AVStream и в AVCodecContext будут эквивалентны. Кроме того, вы не можете опираться на значение time-base внутри AVStream пока не будет вызвана avformat_write_header(...). Как следствие, при копировании временной метки из фрейма в пакет нужно использовать av_rescale_q(...)

Особо хочу отметить, что если вы делаете масштабирование поля pts пакета, не забудьте сделать то же для поля dts, иначе получите интересные артефакты при воспроизведении.

Для аудио-фреймов и аудио пакетов поля pts и dts всегда будут AV_NOPTS_VALUE и никаких преобразований делать не нужно.

Случай транскодирования

Пусть мы имеем входной файл формата Format1 с одним видео потоком Stream1, закодированный кодеком Codec1, нам нужно его перекодировать в файл формата Format2, так же с одним видео потоком Stream2, закодированным кодеком Codec2.

Что получается:

1. после открытия входного файла Stream1 имеет time-base InStreamTimeBase, а Coder1 имеет time-base InCoderTimeBase.
2. после настройки выходного формата, и записи заголовка мы имеем: Stream2 с OutStreamTimeBase и Codec2 с OutCoderTimeBase.
3. av_read_frame() возвращает пакеты с pts/dts в масштабе InStreamTimeBase.
4. после декодирования видео-фрагментов, мы получаем фрейм с pts в масштабе InCoderTimeBase
5. если возникает необходимость копировать pts/dts пакеты в pts фрейма необходимо сделать масштабирование: frame.pts = av_rescale_q(packet.pts/dts, InStreamTimeBase, InCoderTimeBase);
6. перед кодировкой фрейма при помощи Coder1 нужно смасштабировать его pts: frame.pts = av_rescale_q(frame.pts, InCoderTimeBase, OutCoderTimeBase);
7. перед записью получившегося пакета, нужно смасштабировать его pts/dts: packet.pts/dts = av_rescale_q(packet.pts/dts, OutCoderTimeBase, OutStreamTimeBase);
8. записать пакет.
- 9.

Масштабирования нужно делать только если значение временной метки не равно AV_NOPTS_VALUE.

За сим всё.