7-9-AVIO内存输入模式(07-09-avio_decode_audio)

内存IO模式：avio_alloc_context()

```
/**
* Allocate and initialize an AVIOContext for buffered I/O. It must be later
* freed with avio_context_free().
*
* @param buffer Memory block for input/output operations via AVIOContext.
*        The buffer must be allocated with av_malloc() and friends.
*        It may be freed and replaced with a new buffer by libavformat.
*        AVIOContext.buffer holds the buffer currently in use,
*        which must be later freed with av_free().
* @param buffer_size The buffer size is very important for performance.
*        For protocols with fixed blocksize it should be set to this blocksize.
*        For others a typical size is a cache page, e.g. 4kb.
* @param write_flag Set to 1 if the buffer should be writable, 0 otherwise.
* @param opaque An opaque pointer to user-specific data.
* @param read_packet  A function for refilling the buffer, may be NULL.
*                     For stream protocols, must never return 0 but rather
*                     a proper AVERROR code.
* @param write_packet A function for writing the buffer contents, may be NULL.
*        The function may not change the input buffers content.
* @param seek A function for seeking to specified byte position, may be NULL.
*
* @return Allocated AVIOContext or NULL on failure.
*/
AVIOContext *avio_alloc_context(
                  unsigned char *buffer,
                  int buffer_size,
                  int write_flag,
```

```
                void *opaque,
                int (*read_packet)(void *opaque, uint8_t *buf, int buf_size),
                int (*write_packet)(void *opaque, uint8_t *buf, int buf_size),
                int64_t (*seek)(void *opaque, int64_t offset, int whence));
```

- **opaque**是`read_packet`/`write_packet`的第一个参数，指向用户数据。
- **buffer**和**buffer_size**是`read_packet`/`write_packet`的第二个和第三个参数，是供FFmpeg使用的数据区。
  `buffer`用作FFmpeg输入时，由用户负责向`buffer`中填充数据，FFmpeg取走数据。
  `buffer`用作FFmpeg输出时，由FFmpeg负责向`buffer`中填充数据，用户取走数据。
- **write_flag**是缓冲区读写标志，读写的主语是指FFmpeg。
  `write_flag`为1时，`buffer`用于写，即作为FFmpeg输出。
  `write_flag`为0时，`buffer`用于读，即作为FFmpeg输入。
- **read_packet**和**write_packet**是函数指针，指向用户编写的回调函数。
- seek也是函数指针，需要支持seek时使用。 可以类比fseek的机制