

随笔档案 (3811) 2021年7月(40) 2021年6月(1) 2021年5月(29) 2021年4月(15) 2021年3月(13) 2021年2月(96) 2021年1月(47)

随略分長(5113)
android(2)
ASP NEF (30)
ASP NEF (30)
ASP NEF MVC(11)
Boost(1(18)
c#(9)
C+++/(C(773)
c++11(15)
cmake/autotool(62)
com/ATI/Activex(75)
Css(16)
Cx/mage(12)
darwin stream server(3)
DataBase(29)
DirectX(16)
Exis(13)
更多

找找看

常用链接 我的随笔 我的评论 我的参与 最新评论 我的标签

随笔分类 (5113) android(2)

谷歌搜索

2020年11月(27 2020年11月(21) 2020年10月(44) 2020年9月(14) 2020年8月(4) 2020年7月(18) 2020年6月(5) 2020年5月(2) 更多

文章分类 (2) SilverLight(1) sql server(1)

sql server(1)
参考博客
com
morewindows
关注DirectX
chenyujing1234
个人开发历史职识库
open、w程
大拨30软件开发
Dean Chen的专栏
Sloan
曾视频杆开发
Dean Chen的专栏
Sloan
自视频杆开发与原理的析
mydin域执开发与原理的析
mydin域执开发与原理的析
mydin域执开发与原理的析
mydin域执开发
更多

服新评论1. ReRTSP协议详解
@陶子用vl足可以播放的,是直接
获取海康设备的rtsp的地址...
---cosine-x
2. Re:RTSP协议详解 @cosine-x 你的rtsp要能够用vlc播放器播放...

--陶子 3. Re:RTSP协议详解 @cosine-x ip和端口对吗... --陶子

ルビ畑到J区个问题... --cosine-x 5. Re:Git 进阶 - 获取 fetch

大佬,流程图中的两个push是不 是手误?应该是pull? --临渊久渡

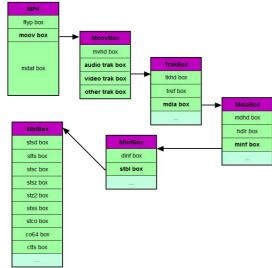
剛速排行榜1. Nginx2location 匹配規則洋解(234821)
2. cmake使用方法洋解(167335)
3. MinGWを接和使用(100038)
4. RTMP、RTSP、HTTP細胞前以洋解(附:直播流地址、播放软件)(96061)
5. C语言字符串操作总结大全個洋細)(90955)

评论排行榜 1. 非IE内核浏览器支持activex插件

37) 2. Nginx之location 匹配规则详解(

推荐排行榜 1. C++中的头文件和源文件(25) 2. Nginx之location 匹配规则详解(22) 3. Javascript中定义类(12) 4. JavaScript中typeof知多少?(11

) 5. MinGW安装和使用(9)



normal_box_file.jpg

3. Mp4文件整体结构

这里,我们按照 MP4 box 的划分来进行相关的阐述。先看一张 MP4 给出的结构图:

typ		_	1			1	4.3	file type and compatibility
pdin							8.43	progressive download information
moov						•	8.1	container for all the metadata
	mvhd					٠	8.3	movie header, overall declarations
	trak					•	8.4	container for an individual track or stream
		tkhd				*	8.5	track header, overall information about the track
		tref					8.6	track reference container
		edts					8.25	edit list container
			elst				8.26	an edit list
		mdia				*	8.7	container for the media information in a track
			mdhd			•	8.8	media header, overall information about the media
	\vdash	_	hdlr	_		•	8.9	handler, declares the media (handler) type
			minf			1	8.10	media information container
				vmhd			8.11.2	video media header, overall information (video track only)
				smhd			8.11.3	sound media header, overall information (soun- track only)
				hmhd			8.11.4	hint media header, overall information (hint traci
				nmhd			8.11.5	Null media header, overall information (some tracks only)
_			_	dinf			8.12	data information box, container
				GIIII	dref	٠	8.13	data reference box, declares source(s) of medi-
_				stbl		+	8.14	data in track sample table box, container for the time/space
					stsd		8.16	map sample descriptions (codec types, initialization
					stts		8.15.2	etc.) (decoding) time-to-sample
					ctts	-	8.15.3	(composition) time to sample
			†		stsc		8.18	sample-to-chunk, partial data-offset information
			_		stsz	-	8.17.2	sample sizes (framing)
					stz2		8.17.3	compact sample sizes (framing)
			1		stco	٠	8.19	chunk offset, partial data-offset information
					co64		8.19	64-bit chunk offset
					stss	-	8.20	sync sample table (random access points)
					stsh	Т	8.21	shadow sync sample table
					padb	-	8.23	sample padding bits
					stdp	Т	8.22	sample degradation priority
					sdtp	-	8.40.2	independent and disposable samples
				-	sbgp		8.40.3.2	sample-to-group
					sgpd			sample group description
					subs		8.42	sub-sample information
	mvex					Т	8.29	movie extends box
		mehd					8.30	movie extends header box
		trex				٠	8.31	track extends defaults
	ipmc						8.45.4	IPMP Control Box
moof							8.32	movie fragment
	mfhd				_	•	8.33	movie fragment header
	traf						8.34	track fragment
	_	tfhd	1		1	1	8.35	track fragment header
	-	trun	1	_	+	₩	8.36	track fragment run
		sdtp	-		+	-	8.40.2	independent and disposable samples
	-	sbgp	1	-	+	+		sample-to-group
		subs			+	-	8.42	sub-sample information
mfra		_	1	-	-	-	8.37	movie fragment random access
	tfra	-			-	L	8.38	track fragment random access
	mfro	-	1	-	-	ľ	8.39	movie fragment random access offset
mdat		-	-	-	+	-	8.2	media data container
free		-	1		+	-	8.24	free space
skip	dt.o.	-	-	-	+	-	8.24	free space
	udta	1	1			1	8.27	user-data

一般来说,解析媒体文件,最美心的部分是视频文件的宽高、时长、码率、编码格式、帧列表、关键帧列表,以及所对应的时戳和在文件中的位置,这些信息,在mp4中,是以特定的算法分开存放在stbl box下属的几个box中的,需要解析stbl下面所有的box,来还原媒体信息。下表是对于以上几个重要的box存放信息的说明

box类型					说明			
ftyp						file type ,表明文件类型		
moov			metadata container, 存放媒体信息的地方					
	mvhd					movie header,文件的总体信息,如时长,创建时间等		
	trak					track or stream container , 存放视频/音频流的容器		
		tkhd				track header, track的总体信息,如时长,宽高等		
		mdia				trak media information container , 不解释		
			mdha	1		media header, 定义TimeScale, trak需要通过TimeScale换算成真实时间		
			hdlr			handler,表明本trak类型,指明是video/audio/还是hint		
			minf			media information container,数据在子box中		
				stbl		sample table box,存放时间/偏移的映射关系表,数据在子box中		
					stsd	sample descriptions		
					stts	(decoding) time-to-sample, "时戳-sample序号"的映射表		
					stsc	sample-to-chunk, sample和chunk的映射表,这里的算法比较巧妙		
					stsz	sample size,每个sample的大小		
					stz2	sample size , 另一种sample size的存储算法 , 更节省空间		
					stss	sync sample table,可随机访问的sample列表 (关键帧列表)		
					stco	chunk offset,每个chunk的偏移,sample的偏移可根据其他box推算出来		
					co64	64-bit chunk offset		
mdat						media data container, 具体的媒体数据		

majorbox.jpg

3.1 File Type Box (ftyp)

通常放在MP4文件的开头,告诉解码器基本的解码版本和兼容格式。

基本格式如下:

```
aligned(8) class FileTypeBox
  extends Box('ftyp') {
   unsigned int(32) major_brand;
   unsigned int(32) minor_version;
   unsigned int(32) compatible_brands[];
```

字段说明:

字段	长度	说明	默认值
major_brand	4	推荐兼容性的版本	iso6
minor_version	4	最低兼容性的版本	1
compatible_brands	列表值	所有的兼容性的版本	'iso6' 'isom' 'dash'

Nginx模块实现

```
ngx_int_t
ngx_rtmp_mp4_write_ftyp(ngx_buf_t *b)
    u_char *pos;
    pos = ngx_rtmp_mp4_start_box(b, "ftyp");
    ngx_rtmp_mp4_box(b, "iso6");
    /* minor version */
ngx_rtmp_mp4_field_32(b, 1);
    /* compatible brands */
ngx_rtmp_mp4_box(b, "isom");
ngx_rtmp_mp4_box(b, "iso6");
ngx_rtmp_mp4_box(b, "dash");
    ngx_rtmp_mp4_update_box_size(b, pos);
    return NGX_OK;
```

3.2 Movie Box (moov)

作为容器盒子,存放相关的trak及meta信息.

• 基本格式如下:

```
aligned(8) class MovieExtendsBox extends Box('mvex'){}
```

3.2.1 Movie Header Box (mvhd)

mvhd 是 moov 下的第一个 box , 用来描述 media 的相关信息:

基本格式如下:

```
aligned(8) class MovieHeaderBox extends FullBox('mvhd', version, 0) {
   if (version=1) {
      unsigned int(64) creation_time;
      unsigned int(64) modification_time;
      unsigned int(32) timescale;
      unsigned int(34) duration;
   } else { // version=0
      unsigned int(32) creation_time;
      unsigned int(32) creation_time;
      unsigned int(32) timescale;
      unsigned int(32) timescale;
      unsigned int(32) duration;
}
                     template int(32) rate = 0x00010000; // typically 1.0
template int(16) volume = 0x0100; // typically, full volume
const bit(16) reserved = 0;
const unsigned int(32)[2] reserved = 0;
template int(32)[9] matrix =
{ (x0x0010000, 0, 0, 0, 0x0010000, 0, 0, 0x40000000 };
    // Unity matrix
bit(32)[6] pre_defined = 0;
    unsigned int(32) next_track_ID;
```

字段说明:

字段	长度	说明	默认值
version	4	版本	0 or 1s
creation_time	4	创建的UTC时间。从1904年开始算起,用秒来表示	无
modification_time	4	最后一次修改时间	无
timescale	4	文件媒体在1秒时间内的刻度值,可以理解为1秒长度的时间单元数	无
duration	4	该track的时间长度,用duration和time scale值可以计算track时长s	实际时间为: duration/timescale = xx 秒
rate	4	推荐播放速率	0x00010000
volume	2	音量大小	0x0100 为最大值
reserved	10	保留字段	0
matrixs	4 * 9	视频变换矩阵	{0x00010000,0,0,0,0x0001s0000,0,0,0,0x40000000}
next_track_ID	4	下一个track使用的id号	

Nginx模块实现

```
static ngx_int_t
ngx_rtmp_mp4_write_mvhd(ngx_buf_t *b)
    u_char *pos;
    pos = ngx_rtmp_mp4_start_box(b, "mvhd");
    /* version */
ngx_rtmp_mp4_field_32(b, 0);
    ngx_rtmp_mp4_field_32(b, 0);
    /* modification time */
ngx_rtmp_mp4_field_32(b, 0);
    /* timescale */
ngx_rtmp_mp4_field_32(b, 1000);
     ngx_rtmp_mp4_field_32(b, 0);
```

```
ngx_rtmp_mp4_field_32(b, 0x0001000)
ngx_rtmp_mp4_field_16(b, 0x0100);
ngx_rtmp_mp4_field_16(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
          ngx_rtmp_mp4_field_32(b, 0);
         ngx rtmp mp4 write matrix(b, 1, 0, 0, 1, 0, 0);
         /* reserved */
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
         ngx_rtmp_mp4_field_32(b, 1);
         ngx_rtmp_mp4_update_box_size(b, pos);
3.2.2 Movie Extends Box (mvex)(fMP4专有)
```

mvex 是 fMP4 的标准盒子。它的作用是告诉解码器这是一个fMP4的文件,具体的 samples 信息内容不再放到 trak 里面,而是在每一个 moof 中。基本格式为

```
aligned(8) class MovieExtendsHeaderBox extends FullBox('mehd', version, 0) { if (version==1) {
    unsigned int(64) fragment_duration;
} else { // version==0
    unsigned int(32) fragment_duration;
```

3.2.2.1 Track Extends Box (trex)(fMP4专有)

trex 是 mvex 的子一级 box 用来给 fMP4 的 sample 设置默认值。基本内容为

```
aligned(8) class TrackExtendsBox extends FullBox('trex', 0, 0){
      ligned(s) class TrackExtendssox extends Fullbox (trex unsigned int(32) track_ID; unsigned int(32) default_sample_description_index; unsigned int(32) default_sample_duration; unsigned int(32) default_sample_flags
unsigned int(32) default_sample_flags
```

3.2.3 Track Box (trak)

trak box 就是主要存放相关 media stream 的内容。

3.2.3.1 Track Header Box (tkhd)

tkhd 是 trak box 的子一级 box 的内容。主要是用来描述该特定 trak 的相关内容信息。其主要内容为:

• 基本格式如下:

```
aligned(8) class TrackHeaderBox
    extends FullBox('tkhd', version, flags){
    if (version==1) {
        unsigned int(64) creation_time;
        unsigned int(64) modification_time;
        unsigned int(32) track_ID;
        const unsigned int(32) reserved = 0;
        unsigned int(64) duration;
    } else (// version==0
        unsigned int(32) creation_time;
        unsigned int(32) modification_time;
        unsigned int(32) track_ID;
        const unsigned int(32) reserved = 0;
        unsigned int(32) duration;
}
       const unsigned int(32)[2] reserved = 0;
 const unsigned int(32)[2] reserved = 0;
template int(16) layer = 0;
template int(16) alternate_group = 0;
template int(16) volume = {if track_is_audio 0x0100 else 0); const unsigned int(16) reserved = 0;
template int(32)[9] matrix=
{ 0x00010000,0,0,0,0x00100000,0,0,0,0x40000000 };
// unity matrix
unsigned int(32) width;
unsigned int(32) height;
}
```

字段说明:

字段	长度	说明	默认值
version	4	版本	
creation_time	4	创建时间,非必须	0
modification_time	4	修改时间,非必须	0
track_ID	4	指明当前描述的 track ID	1
reserved	4	保留	0
duration	4	当前 track 内容持续的时间。通常结合 timescale 进行相关计算	0
reserved	12	保留字段	0
reserved	2	保留字段	0
alternate_group	2	保留字段	0
volume	2	保留字段	if track_is_audio 0x0100 else 0
reserved	2	保留字段	0
matrix	9 * 4	matrix	b, 1, 0, 0, 1, 0, 0 , width , height

3.2.3.2 Media Box (media)

mdia 主要用来包裹相关的 media 信息。

(1) Media Header Box (mdhd)

基本格式如下:

```
aligned(8) class MediaHeaderBox extends FullBox('mdhd', version, 0) { if (version==1) { unsigned int(64) modification_time; unsigned int(64) modification_time; unsigned int(64) timescale; unsigned int(64) duration; } else (// version==0 unsigned int(32) creation_time; unsigned int(32) modification_time; unsigned int(32) modification_time; unsigned int(32) timescale; unsigned int(32) duration; }
  bit(1) pad = 0;
unsigned int(5)[3] language; // ISO-639-2/T language code unsigned int(16) pre_defined = 0;
}
```

```
字段
         长度
                                                             默认值
                                     说明
                  版本
version
creation_time
                 创建时间,非必须
                                                             0
modification_time 4
                 修改时间,非必须
                                                             0
           4
                 文件媒体在1秒时间内的刻度值,可以理解为1秒长度的时间单元数
                                                             无
          4 当前 track 内容持续的时间。通常结合 timescale 进行相关计算
duration
                                                             0
lanuage
     4s 表明当前 trak 的语言。因为该字段总长为 15bit,通常是和 pad 组合成为 2B 的长度。
```

(2) Handler Reference Box(hdlr)

基本格式如下:

```
aligned(8) class HandlerBox extends FullBox('hdlr', version = 0, 0) {
  unsigned int(32) pre_defined = 0;
  unsigned int(32) handler_type;
  const unsigned int(32)[3] reserved = 0;
  string name;
```

字段说明:

字段	长度	说明	默认值
version	4	版本	
pre_defined	4	版本	0
handler_type	4	是代指具体 trak 的处理类型	0
reserved	4 * 3	reserved	0
data	string	reserved	"VideoHandler" or "SoundHandler"

• handler_type 类型如下:

```
vide : Video track
soun : Audio track
hint : Hint track
meta : Timed Metadata track
auxv : Auxiliary Video track
```

3.2.3.3 Media Information Box (minf)

minf 是子属内容中,重要的容器 box,用来存放当前 track 的基本描述信息。

(1) Video Media Header Box (vmhd)

• 基本格式如下:

```
aligned(8) class VideoMediaHeaderBox extends FullBox(Vmhd', version = 0, 1) { template unsigned int(16) graphicsmode = 0; // copy, see below template unsigned int(16)[3] opcolor = (0, 0, 0);
```

(2) Sound Media Header Box (smhd)

• 基本格式如下:

```
aligned(8) class SoundMediaHeaderBox

extends FullBox('smhd', version = 0, 0) {

template int(16) balance = 0;

const unsigned int(16) reserved = 0;
```

(3) Data Information Box (dinf)

dinf 是用来说明在 trak 中,media 描述信息的位置。其实本身就是一个容器,没啥内容

基本格式如下:

```
aligned(8) class SoundMediaHeaderBox
  extends FullBox('smhd', version = 0, 0) {
  template int(16) balance = 0;
  const unsigned int(16) reserved = 0;
```

(4) Data Reference Box (dref)

dref 是用来设置当前Box描述信息的 data_entry。

基本格式如下:

```
aligned(8) class DataReferenceBox
  extends FullBox('dref', version = 0, 0) {
  unsigned int(32) entry_count;
    for (i=1; i <= entry_count; i++) {
   DataEntryBox(entry_version, entry_flags) data_entry; }
}</pre>
```

字段说明:

字段	长度	说明	默认值
version	4	版本	0
entry_count	4	入口数	1
entry_version	4	入口数	0
entry_flags	3	入口数	0

3.3 Moof Box

3.4 Sidxs Box

作者:lo隨风 链接:https://www.jianshu.com/p/dee55d54fd54 来源:简书 著作权归作者所有。商业转载请联系作者获得授权,非商业转载请注明出处。

分类: 音视频文件, 音视频、流媒体







- 無損権等:
 我给鴻星尔克写了一个720°看鞋展厅
 带团队场的日常(E)
 你为什么不想向上汇报?
 传统NET 4x应用容器化体验(4)
 CSS 世界中的方位与顺序 加州大学伯克利哈斯商学院 | 高管教育 大数据与数据科学实战

- 器新統領:
 10月28日上市 | 微吹欠晒 (帝国時代4): 元法、画馬大提升
 10月28日上市 | 微吹欠晒 (帝国時代4): 元法、画馬大提升
 パ水製練耳印から非省支援之格の折沈木寿の雨? 吉方4替
 旧は管省画面世界第一 第 元 大 台界电力の正式技能
 《船崎時保作七》上架時間WeGame: 最高直降30元
 本了風速、空与地后 Intel CEO放话: 还有100多家公司等着我们代工
 更多新闻。