原 ffmpeg 源代码简单分析 ： avcodec_register_all()

2013年10月13日 13:34:01　阅读数：26452

==========

FFmpeg的库函数源代码分析文章列表：

【架构图】

FFmpeg 源代码结构图 - 解码

FFmpeg 源代码结构图 - 编码

【通用】

FFmpeg 源代码简单分析： av_register_all()

FFmpeg 源代码简单分析： avcodec_register_all()

FFmpeg 源代码简单分析：内存的分配和释放（ av_malloc()、 av_free() 等）

FFmpeg 源代码简单分析：常见结构体的初始化和销毁（ AVFormatContext， AVFrame 等）

FFmpeg 源代码简单分析： avio_open2()

FFmpeg 源代码简单分析： av_find_decoder() 和 av_find_encoder()

FFmpeg 源代码简单分析： avcodec_open2()

FFmpeg 源代码简单分析： avcodec_close()

【解码】

图解 FFMPEG 打开媒体的函数 avformat_open_input

FFmpeg 源代码简单分析： avformat_open_input()

FFmpeg 源代码简单分析： avformat_find_stream_info()

FFmpeg 源代码简单分析： av_read_frame()

FFmpeg 源代码简单分析： avcodec_decode_video2()

FFmpeg 源代码简单分析： avformat_close_input()

【编码】

FFmpeg 源代码简单分析： avformat_alloc_output_context2()

FFmpeg 源代码简单分析： avformat_write_header()

FFmpeg 源代码简单分析： avcodec_encode_video()

FFmpeg 源代码简单分析： av_write_frame()

FFmpeg 源代码简单分析： av_write_trailer()

【其它】

FFmpeg 源代码简单分析：日志输出系统（ av_log() 等）

FFmpeg 源代码简单分析：结构体成员管理系统 -AVClass

FFmpeg 源代码简单分析：结构体成员管理系统 -AVOption

FFmpeg 源代码简单分析： libswscale 的 sws_getContext()

FFmpeg 源代码简单分析： libswscale 的 sws_scale()

FFmpeg 源代码简单分析： libavdevice 的 avdevice_register_all()

FFmpeg 源代码简单分析： libavdevice 的 gdigrab

【脚本】

FFmpeg 源代码简单分析： makefile

FFmpeg 源代码简单分析： configure

【H.264】

FFmpeg 的 H.264 解码器源代码简单分析：概述

==========

本文分析一下ffmpeg注册编解码器等的函数avcodec_register_all()（注意不是av_register_all()，那是注册所有东西的）。该函数在所有基于ffmpeg的应用程序中几乎都是第一个被调用的。只有调用了该函数，才能使用编解码器等。

之前已经写过一篇文章了：

ffmpeg 源代码简单分析 ： av_register_all()

其实 注册编解码器和注册复用器解复用器道理是差不多的，重复的内容不再多说。

编码器 的注册是：

REGISTER_ENCODER (X,x);

例如：

REGISTER_ENCODER (LJPEG, ljpeg);

解码器 的注册是：

REGISTER_DECODER (X,x);

例如：

REGISTER_DECODER (H264, h264);

既包含编码器有包含解码器 的注册是：

REGISTER_ENCDEC (X,x);

例如：

REGISTER_ENCDEC (BMP, bmp);

此外还有几种注册：

Parser：

REGISTER_PARSER (X,x);

例如：

REGISTER_PARSER (H264, h264);

BSF（bitstream filters，比特流滤镜，有一个常用：h264_mp4toannexb）：

REGISTER_BSF   (X,x);

例如：

REGISTER_BSF   (H264_MP4TOANNEXB, h264_mp4toannexb);

HWACCEL（hardware accelerators，硬件加速器）：

REGISTER_HWACCEL (X,x);

例如：

REGISTER_HWACCEL (H264_DXVA2, h264_dxva2);

我们来看一下宏的定义，这里以 编解码器 为例：

```cpp
#define REGISTER_ENCODER(X,x) { \
          extern AVCodec ff_##x##_encoder; \
          if(CONFIG_##X##_ENCODER)  avcodec_register(&ff_##x##_encoder); }
#define REGISTER_DECODER(X,x) { \
          extern AVCodec ff_##x##_decoder; \
          if(CONFIG_##X##_DECODER)  avcodec_register(&ff_##x##_decoder); }
#define REGISTER_ENCDEC(X,x)  REGISTER_ENCODER(X,x); REGISTER_DECODER(X,x)
```

在这里，我发现其实编码器和解码器用的注册函数都是一样的：avcodec_register()

以REGISTER_DECODER (H264, h264)为例，就是等效于

```cpp
extern AVCodec ff_h264_decoder;
if(CONFIG_H264_DECODER)  avcodec_register(&ff_h264_decoder);
```

下面看一下 avcodec_register()的源代码：

```cpp
//注册所有的AVCodec
void avcodec_register(AVCodec *codec)
{
    AVCodec **p;
    //初始化
    avcodec_init();
    //从第一个开始
    p = &first_avcodec;
    while (*p != NULL) p = &(*p)->next;
    *p = codec;
    codec->next = NULL;

    if (codec->init_static_data)
        codec->init_static_data(codec);
}
```

这段代码是比较容易理解的。首先先提一点，first_avcdec是就是AVCodec链表的头部地址，是一个全局静态变量，定义如下：

```cpp
/* encoder management */
static AVCodec *first_avcodec = NULL;
```

由此我们可以分析出 avcodec_register() 的含义，一句话概括就是：遍历链表并把当前的AVCodec加到链表的尾部。

同理， **Parser ， BSF（bitstream filters，比特流滤镜） ， HWACCEL（hardware accelerators，硬件加速器）** 的注册方式都是类似的。不再详述。

下面贴出它的原代码：

```cpp
/*
 *雷霄骅
 *leixiaohua1020@126.com
 *中国传媒大学/数字电视技术
 */
/*
 * Provide registration of all codecs, parsers and bitstream filters for libavcodec.
 * Copyright (c) 2002 Fabrice Bellard
 *
 * This file is part of FFmpeg.
 *
 * FFmpeg is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * FFmpeg is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with FFmpeg; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 */

/**
 * @file
 * Provide registration of all codecs, parsers and bitstream filters for libavcodec.
 */

```

```
#include "avcodec.h"
//硬件加速
#define REGISTER_HWACCEL(X,x) { \
          extern AVHWAccel ff_##x##_hwaccel; \
          if(CONFIG_##X##_HWACCEL) av_register_hwaccel(&ff_##x##_hwaccel); }

#define REGISTER_ENCODER(X,x) { \
          extern AVCodec ff_##x##_encoder; \
          if(CONFIG_##X##_ENCODER)  avcodec_register(&ff_##x##_encoder); }
//定义的宏？宏的速度会快一点？注册AVCodec
//extern AVCodec ff_##x##_decoder;
//注意：extern表明全局唯一
//在h264中，对应的就是ff_h264_decoder
//由此可见AVCodecParser的名字是固定的
#define REGISTER_DECODER(X,x) { \
          extern AVCodec ff_##x##_decoder; \
          if(CONFIG_##X##_DECODER)  avcodec_register(&ff_##x##_decoder); }
#define REGISTER_ENCDEC(X,x)  REGISTER_ENCODER(X,x); REGISTER_DECODER(X,x)
//定义的宏？宏的速度会快一点？注册AVCodecParser
//extern AVCodecParser ff_##x##_parser;
//在h264中，对应的就是ff_h264_parser
//由此可见AVCodecParser的名字是固定的
#define REGISTER_PARSER(X,x) { \
          extern AVCodecParser ff_##x##_parser; \
          if(CONFIG_##X##_PARSER)  av_register_codec_parser(&ff_##x##_parser); }
#define REGISTER_BSF(X,x) { \
          extern AVBitStreamFilter ff_##x##_bsf; \
          if(CONFIG_##X##_BSF)     av_register_bitstream_filter(&ff_##x##_bsf); }

void avcodec_register_all(void)
{
    static int initialized;

    if (initialized)
        return;
    initialized = 1;

    /* hardware accelerators */
    REGISTER_HWACCEL (H263_VAAPI, h263_vaapi);
    REGISTER_HWACCEL (H264_DXVA2, h264_dxva2);
    REGISTER_HWACCEL (H264_VAAPI, h264_vaapi);
    REGISTER_HWACCEL (H264_VDA, h264_vda);
    REGISTER_HWACCEL (MPEG1_VDPAU, mpeg1_vdpau);
    REGISTER_HWACCEL (MPEG2_DXVA2, mpeg2_dxva2);
    REGISTER_HWACCEL (MPEG2_VAAPI, mpeg2_vaapi);
    REGISTER_HWACCEL (MPEG2_VDPAU, mpeg2_vdpau);
    REGISTER_HWACCEL (MPEG4_VAAPI, mpeg4_vaapi);
    REGISTER_HWACCEL (VC1_DXVA2, vc1_dxva2);
    REGISTER_HWACCEL (VC1_VAAPI, vc1_vaapi);
    REGISTER_HWACCEL (WMV3_DXVA2, wmv3_dxva2);
    REGISTER_HWACCEL (WMV3_VAAPI, wmv3_vaapi);

    /* video codecs */
    REGISTER_ENCODER (A64MULTI, a64multi);
    REGISTER_ENCODER (A64MULTI5, a64multi5);
    REGISTER_DECODER (AASC, aasc);
    REGISTER_ENCDEC  (AMV, amv);
    REGISTER_DECODER (ANM, anm);
    REGISTER_DECODER (ANSI, ansi);
    REGISTER_ENCDEC  (ASV1, asv1);
    REGISTER_ENCDEC  (ASV2, asv2);
    REGISTER_DECODER (AURA, aura);
    REGISTER_DECODER (AURA2, aura2);
    REGISTER_ENCDEC  (AVRP, avrp);
    REGISTER_DECODER (AVS, avs);
    REGISTER_DECODER (BETHSOFTVID, bethsoftvid);
    REGISTER_DECODER (BFI, bfi);
    REGISTER_DECODER (BINK, bink);
    REGISTER_ENCDEC  (BMP, bmp);
    REGISTER_DECODER (BMV_VIDEO, bmv_video);
    REGISTER_DECODER (C93, c93);
    REGISTER_DECODER (CAVS, cavs);
    REGISTER_DECODER (CDGRAPHICS, cdgraphics);
    REGISTER_DECODER (CINEPAK, cinepak);
    REGISTER_ENCDEC  (CLJR, cljr);
    REGISTER_DECODER (CSCD, cscd);
    REGISTER_DECODER (CYUV, cyuv);
    REGISTER_DECODER (DFA, dfa);
    REGISTER_DECODER (DIRAC, dirac);
    REGISTER_ENCDEC  (DNXHD, dnxhd);
    REGISTER_ENCDEC  (DPX, dpx);
    REGISTER_DECODER (DSICINVIDEO, dsicinvideo);
    REGISTER_ENCDEC  (DVVIDEO, dvvideo);
    REGISTER_DECODER (DXA, dxa);
    REGISTER_DECODER (DXTORY, dxtory);
    REGISTER_DECODER (EACMV, eacmv);
    REGISTER_DECODER (EAMAD, eamad);
    REGISTER_DECODER (EATGQ, eatgq);
    REGISTER_DECODER (EATGV, eatgv);
    REGISTER_DECODER (EATQI, eatqi);
    REGISTER_DECODER (EIGHTBPS, eightbps);
```

```
    REGISTER_DECODER (EIGHTSVX_EXP, eightsvx_exp);
    REGISTER_DECODER (EIGHTSVX_FIB, eightsvx_fib);
    REGISTER_DECODER (ESCAPE124, escape124);
    REGISTER_DECODER (ESCAPE130, escape130);
    REGISTER_ENCDEC  (FFV1, ffv1);
    REGISTER_ENCDEC  (FFVHUFF, ffvhuff);
    REGISTER_ENCDEC  (FLASHSV, flashsv);
    REGISTER_ENCDEC  (FLASHSV2, flashsv2);
    REGISTER_DECODER (FLIC, flic);
    REGISTER_ENCDEC  (FLV, flv);
    REGISTER_DECODER (FOURXM, fourxm);
    REGISTER_DECODER (FRAPS, fraps);
    REGISTER_DECODER (FRWU, frwu);
    REGISTER_ENCDEC  (GIF, gif);
    REGISTER_ENCDEC  (H261, h261);
    REGISTER_ENCDEC  (H263, h263);
    REGISTER_DECODER (H263I, h263i);
    REGISTER_ENCODER (H263P, h263p);
    REGISTER_DECODER (H264, h264);
    REGISTER_DECODER (H264_CRYSTALHD, h264_crystalhd);
    REGISTER_DECODER (H264_VDPAU, h264_vdpau);
    REGISTER_ENCDEC  (HUFFYUV, huffyuv);
    REGISTER_DECODER (IDCIN, idcin);
    REGISTER_DECODER (IFF_BYTERUN1, iff_byterun1);
    REGISTER_DECODER (IFF_ILBM, iff_ilbm);
    REGISTER_DECODER (INDEO2, indeo2);
    REGISTER_DECODER (INDEO3, indeo3);
    REGISTER_DECODER (INDEO4, indeo4);
    REGISTER_DECODER (INDEO5, indeo5);
    REGISTER_DECODER (INTERPLAY_VIDEO, interplay_video);
    REGISTER_ENCDEC  (JPEG2000, jpeg2000);
    REGISTER_ENCDEC  (JPEGLS, jpegls);
    REGISTER_DECODER (JV, jv);
    REGISTER_DECODER (KGV1, kgv1);
    REGISTER_DECODER (KMVC, kmvc);
    REGISTER_DECODER (LAGARITH, lagarith);
    REGISTER_ENCODER (LJPEG, ljpeg);
    REGISTER_DECODER (LOCO, loco);
    REGISTER_DECODER (MDEC, mdec);
    REGISTER_DECODER (MIMIC, mimic);
    REGISTER_ENCDEC  (MJPEG, mjpeg);
    REGISTER_DECODER (MJPEGB, mjpegb);
    REGISTER_DECODER (MMVIDEO, mmvideo);
    REGISTER_DECODER (MOTIONPIXELS, motionpixels);
    REGISTER_DECODER (MPEG_XVMC, mpeg_xvmc);
    REGISTER_ENCDEC  (MPEG1VIDEO, mpeg1video);
    REGISTER_ENCDEC  (MPEG2VIDEO, mpeg2video);
    REGISTER_ENCDEC  (MPEG4, mpeg4);
    REGISTER_DECODER (MPEG4_CRYSTALHD, mpeg4_crystalhd);
    REGISTER_DECODER (MPEG4_VDPAU, mpeg4_vdpau);
    REGISTER_DECODER (MPEGVIDEO, mpegvideo);
    REGISTER_DECODER (MPEG_VDPAU, mpeg_vdpau);
    REGISTER_DECODER (MPEG1_VDPAU, mpeg1_vdpau);
    REGISTER_DECODER (MPEG2_CRYSTALHD, mpeg2_crystalhd);
    REGISTER_DECODER (MSMPEG4_CRYSTALHD, msmpeg4_crystalhd);
    REGISTER_DECODER (MSMPEG4V1, msmpeg4v1);
    REGISTER_ENCDEC  (MSMPEG4V2, msmpeg4v2);
    REGISTER_ENCDEC  (MSMPEG4V3, msmpeg4v3);
    REGISTER_DECODER (MSRLE, msrle);
    REGISTER_ENCDEC  (MSVIDEO1, msvideo1);
    REGISTER_DECODER (MSZH, mszh);
    REGISTER_DECODER (MXPEG, mxpeg);
    REGISTER_DECODER (NUV, nuv);
    REGISTER_ENCDEC  (PAM, pam);
    REGISTER_ENCDEC  (PBM, pbm);
    REGISTER_ENCDEC  (PCX, pcx);
    REGISTER_ENCDEC  (PGM, pgm);
    REGISTER_ENCDEC  (PGMYUV, pgmyuv);
    REGISTER_DECODER (PICTOR, pictor);
    REGISTER_ENCDEC  (PNG, png);
    REGISTER_ENCDEC  (PPM, ppm);
    REGISTER_ENCDEC  (PRORES, prores);
    REGISTER_DECODER (PRORES_LGPL, prores_lgpl);
    REGISTER_DECODER (PTX, ptx);
    REGISTER_DECODER (QDRAW, qdraw);
    REGISTER_DECODER (QPEG, qpeg);
    REGISTER_ENCDEC  (QTRLE, qtrle);
    REGISTER_ENCDEC  (R10K,  r10k);
    REGISTER_ENCDEC  (R210,  r210);
    REGISTER_ENCDEC  (RAWVIDEO, rawvideo);
    REGISTER_DECODER (RL2, rl2);
    REGISTER_ENCDEC  (ROQ, roq);
    REGISTER_DECODER (RPZA, rpza);
    REGISTER_ENCDEC  (RV10, rv10);
    REGISTER_ENCDEC  (RV20, rv20);
    REGISTER_DECODER (RV30, rv30);
    REGISTER_DECODER (RV40, rv40);
    REGISTER_DECODER (S302M, s302m);
    REGISTER_ENCDEC  (SGI, sgi);
    REGISTER_DECODER (SMACKER, smacker);
    REGISTER_DECODER (SMC, smc);
    REGISTER_ENCDEC  (SNOW, snow);
```

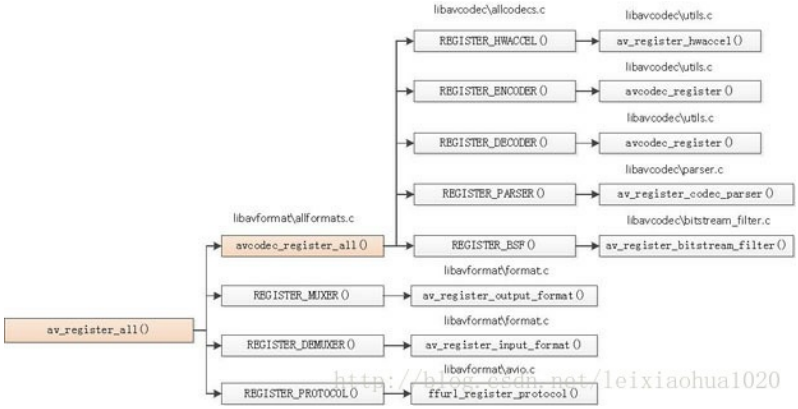
```
    REGISTER_ENCDEC  (SNOW, snow);
    REGISTER_DECODER (SP5X, sp5x);
    REGISTER_DECODER (SUNRAST, sunrast);
    REGISTER_ENCDEC  (SVQ1, svq1);
    REGISTER_DECODER (SVQ3, svq3);
    REGISTER_ENCDEC  (TARGA, targa);
    REGISTER_DECODER (THEORA, theora);
    REGISTER_DECODER (THP, thp);
    REGISTER_DECODER (TIERTEXSEQVIDEO, tiertexseqvideo);
    REGISTER_ENCDEC  (TIFF, tiff);
    REGISTER_DECODER (TMV, tmv);
    REGISTER_DECODER (TRUEMOTION1, truemotion1);
    REGISTER_DECODER (TRUEMOTION2, truemotion2);
    REGISTER_DECODER (TSCC, tscc);
    REGISTER_DECODER (TXD, txd);
    REGISTER_DECODER (ULTI, ulti);
    REGISTER_DECODER (UTVIDEO, utvideo);
    REGISTER_ENCDEC  (V210,  v210);
    REGISTER_DECODER (V210X, v210x);
    REGISTER_ENCDEC  (V308, v308);
    REGISTER_ENCDEC  (V410, v410);
    REGISTER_DECODER (VB, vb);
    REGISTER_DECODER (VBLE, vble);
    REGISTER_DECODER (VC1, vc1);
    REGISTER_DECODER (VC1_CRYSTALHD, vc1_crystalhd);
    REGISTER_DECODER (VC1_VDPAU, vc1_vdpau);
    REGISTER_DECODER (VC1IMAGE, vc1image);
    REGISTER_DECODER (VCR1, vcr1);
    REGISTER_DECODER (VMDVIDEO, vmdvideo);
    REGISTER_DECODER (VMNC, vmnc);
    REGISTER_DECODER (VP3, vp3);
    REGISTER_DECODER (VP5, vp5);
    REGISTER_DECODER (VP6, vp6);
    REGISTER_DECODER (VP6A, vp6a);
    REGISTER_DECODER (VP6F, vp6f);
    REGISTER_DECODER (VP8, vp8);
    REGISTER_DECODER (VQA, vqa);
    REGISTER_ENCDEC  (WMV1, wmv1);
    REGISTER_ENCDEC  (WMV2, wmv2);
    REGISTER_DECODER (WMV3, wmv3);
    REGISTER_DECODER (WMV3_CRYSTALHD, wmv3_crystalhd);
    REGISTER_DECODER (WMV3_VDPAU, wmv3_vdpau);
    REGISTER_DECODER (WMV3IMAGE, wmv3image);
    REGISTER_DECODER (WNV1, wnv1);
    REGISTER_DECODER (XAN_WC3, xan_wc3);
    REGISTER_DECODER (XAN_WC4, xan_wc4);
    REGISTER_DECODER (XL, xl);
    REGISTER_ENCDEC  (XWD, xwd);
    REGISTER_ENCDEC  (Y41P, y41p);
    REGISTER_DECODER (YOP, yop);
    REGISTER_ENCDEC  (YUV4, yuv4);
    REGISTER_ENCDEC  (ZLIB, zlib);
    REGISTER_ENCDEC  (ZMBV, zmbv);

    /* audio codecs */
    REGISTER_ENCDEC  (AAC, aac);
    REGISTER_DECODER (AAC_LATM, aac_latm);
    REGISTER_ENCDEC  (AC3, ac3);
    REGISTER_ENCODER (AC3_FIXED, ac3_fixed);
    REGISTER_ENCDEC  (ALAC, alac);
    REGISTER_DECODER (ALS, als);
    REGISTER_DECODER (AMRNB, amrnb);
    REGISTER_DECODER (AMRWB, amrwb);
    REGISTER_DECODER (APE, ape);
    REGISTER_DECODER (ATRAC1, atrac1);
    REGISTER_DECODER (ATRAC3, atrac3);
    REGISTER_DECODER (BINKAUDIO_DCT, binkaudio_dct);
    REGISTER_DECODER (BINKAUDIO_RDFT, binkaudio_rdft);
    REGISTER_DECODER (BMV_AUDIO, bmv_audio);
    REGISTER_DECODER (COOK, cook);
    REGISTER_ENCDEC  (DCA, dca);
    REGISTER_DECODER (DSICINAUDIO, dsicinaudio);
    REGISTER_ENCDEC  (EAC3, eac3);
    REGISTER_DECODER (FFWAVESYNTH, ffwavesynth);
    REGISTER_ENCDEC  (FLAC, flac);
    REGISTER_ENCDEC  (G723_1, g723_1);
    REGISTER_DECODER (G729, g729);
    REGISTER_DECODER (GSM, gsm);
    REGISTER_DECODER (GSM_MS, gsm_ms);
    REGISTER_DECODER (IMC, imc);
    REGISTER_DECODER (MACE3, mace3);
    REGISTER_DECODER (MACE6, mace6);
    REGISTER_DECODER (MLP, mlp);
    REGISTER_DECODER (MP1, mp1);
    REGISTER_DECODER (MP1FLOAT, mp1float);
    REGISTER_ENCDEC  (MP2, mp2);
    REGISTER_DECODER (MP2FLOAT, mp2float);
    REGISTER_DECODER (MP3, mp3);
    REGISTER_DECODER (MP3FLOAT, mp3float);
    REGISTER_DECODER (MP3ADU, mp3adu);
    REGISTER_DECODER (MP3ADUFLOAT, mp3adufloat);
    REGISTER_DECODER (MP3ON4, mp3on4);
```

```
    REGISTER_DECODER (MP3ON4FLOAT, mp3on4float);
    REGISTER_DECODER (MPC7, mpc7);
    REGISTER_DECODER (MPC8, mpc8);
    REGISTER_ENCDEC  (NELLYMOSER, nellymoser);
    REGISTER_DECODER (QCELP, qcelp);
    REGISTER_DECODER (QDM2, qdm2);
    REGISTER_ENCDEC  (RA_144, ra_144);
    REGISTER_DECODER (RA_288, ra_288);
    REGISTER_DECODER (SHORTEN, shorten);
    REGISTER_DECODER (SIPR, sipr);
    REGISTER_DECODER (SMACKAUD, smackaud);
    REGISTER_ENCDEC  (SONIC, sonic);
    REGISTER_ENCODER (SONIC_LS, sonic_ls);
    REGISTER_DECODER (TRUEHD, truehd);
    REGISTER_DECODER (TRUESPEECH, truespeech);
    REGISTER_DECODER (TTA, tta);
    REGISTER_DECODER (TWINVQ, twinvq);
    REGISTER_DECODER (VMDAUDIO, vmdaudio);
    REGISTER_ENCDEC  (VORBIS, vorbis);
    REGISTER_DECODER (WAVPACK, wavpack);
    REGISTER_DECODER (WMALOSSLESS, wmalossless);
    REGISTER_DECODER (WMAPRO, wmapro);
    REGISTER_ENCDEC  (WMAV1, wmav1);
    REGISTER_ENCDEC  (WMAV2, wmav2);
    REGISTER_DECODER (WMAVOICE, wmavoice);
    REGISTER_DECODER (WS_SND1, ws_snd1);

    /* PCM codecs */
    REGISTER_ENCDEC  (PCM_ALAW, pcm_alaw);
    REGISTER_DECODER (PCM_BLURAY, pcm_bluray);
    REGISTER_DECODER (PCM_DVD, pcm_dvd);
    REGISTER_ENCDEC  (PCM_F32BE, pcm_f32be);
    REGISTER_ENCDEC  (PCM_F32LE, pcm_f32le);
    REGISTER_ENCDEC  (PCM_F64BE, pcm_f64be);
    REGISTER_ENCDEC  (PCM_F64LE, pcm_f64le);
    REGISTER_DECODER (PCM_LXF, pcm_lxf);
    REGISTER_ENCDEC  (PCM_MULAW, pcm_mulaw);
    REGISTER_ENCDEC  (PCM_S8, pcm_s8);
    REGISTER_DECODER (PCM_S8_PLANAR, pcm_s8_planar);
    REGISTER_ENCDEC  (PCM_S16BE, pcm_s16be);
    REGISTER_ENCDEC  (PCM_S16LE, pcm_s16le);
    REGISTER_DECODER (PCM_S16LE_PLANAR, pcm_s16le_planar);
    REGISTER_ENCDEC  (PCM_S24BE, pcm_s24be);
    REGISTER_ENCDEC  (PCM_S24DAUD, pcm_s24daud);
    REGISTER_ENCDEC  (PCM_S24LE, pcm_s24le);
    REGISTER_ENCDEC  (PCM_S32BE, pcm_s32be);
    REGISTER_ENCDEC  (PCM_S32LE, pcm_s32le);
    REGISTER_ENCDEC  (PCM_U8, pcm_u8);
    REGISTER_ENCDEC  (PCM_U16BE, pcm_u16be);
    REGISTER_ENCDEC  (PCM_U16LE, pcm_u16le);
    REGISTER_ENCDEC  (PCM_U24BE, pcm_u24be);
    REGISTER_ENCDEC  (PCM_U24LE, pcm_u24le);
    REGISTER_ENCDEC  (PCM_U32BE, pcm_u32be);
    REGISTER_ENCDEC  (PCM_U32LE, pcm_u32le);
    REGISTER_DECODER (PCM_ZORK , pcm_zork);

    /* DPCM codecs */
    REGISTER_DECODER (INTERPLAY_DPCM, interplay_dpcm);
    REGISTER_ENCDEC  (ROQ_DPCM, roq_dpcm);
    REGISTER_DECODER (SOL_DPCM, sol_dpcm);
    REGISTER_DECODER (XAN_DPCM, xan_dpcm);

    /* ADPCM codecs */
    REGISTER_DECODER (ADPCM_4XM, adpcm_4xm);
    REGISTER_ENCDEC  (ADPCM_ADX, adpcm_adx);
    REGISTER_DECODER (ADPCM_CT, adpcm_ct);
    REGISTER_DECODER (ADPCM_EA, adpcm_ea);
    REGISTER_DECODER (ADPCM_EA_MAXIS_XA, adpcm_ea_maxis_xa);
    REGISTER_DECODER (ADPCM_EA_R1, adpcm_ea_r1);
    REGISTER_DECODER (ADPCM_EA_R2, adpcm_ea_r2);
    REGISTER_DECODER (ADPCM_EA_R3, adpcm_ea_r3);
    REGISTER_DECODER (ADPCM_EA_XAS, adpcm_ea_xas);
    REGISTER_ENCDEC  (ADPCM_G722, adpcm_g722);
    REGISTER_ENCDEC  (ADPCM_G726, adpcm_g726);
    REGISTER_DECODER (ADPCM_IMA_AMV, adpcm_ima_amv);
    REGISTER_DECODER (ADPCM_IMA_APC, adpcm_ima_apc);
    REGISTER_DECODER (ADPCM_IMA_DK3, adpcm_ima_dk3);
    REGISTER_DECODER (ADPCM_IMA_DK4, adpcm_ima_dk4);
    REGISTER_DECODER (ADPCM_IMA_EA_EACS, adpcm_ima_ea_eacs);
    REGISTER_DECODER (ADPCM_IMA_EA_SEAD, adpcm_ima_ea_sead);
    REGISTER_DECODER (ADPCM_IMA_ISS, adpcm_ima_iss);
    REGISTER_ENCDEC  (ADPCM_IMA_QT, adpcm_ima_qt);
    REGISTER_DECODER (ADPCM_IMA_SMJPEG, adpcm_ima_smjpeg);
    REGISTER_ENCDEC  (ADPCM_IMA_WAV, adpcm_ima_wav);
    REGISTER_DECODER (ADPCM_IMA_WS, adpcm_ima_ws);
    REGISTER_ENCDEC  (ADPCM_MS, adpcm_ms);
    REGISTER_DECODER (ADPCM_SBPRO_2, adpcm_sbpro_2);
    REGISTER_DECODER (ADPCM_SBPRO_3, adpcm_sbpro_3);
    REGISTER_DECODER (ADPCM_SBPRO_4, adpcm_sbpro_4);
    REGISTER_ENCDEC  (ADPCM_SWF, adpcm_swf);
    REGISTER_DECODER (ADPCM THP, adpcm thp);
```

```
397.     REGISTER_DECODER (ADPCM_XA, adpcm_xa);
398.     REGISTER_ENCDEC  (ADPCM_YAMAHA, adpcm_yamaha);
399.
400.     /* subtitles */
401.     REGISTER_ENCDEC  (ASS, ass);
402.     REGISTER_ENCDEC  (DVBSUB, dvbsub);
403.     REGISTER_ENCDEC  (DVDSUB, dvdsub);
404.     REGISTER_DECODER (PGSSUB, pgssub);
405.     REGISTER_ENCDEC  (SRT, srt);
406.     REGISTER_ENCDEC  (XSUB, xsub);
407.
408.     /* external libraries */
409.     REGISTER_ENCODER (LIBAACPLUS, libaacplus);
410.     REGISTER_DECODER (LIBCELT, libcelt);
411.     REGISTER_ENCDEC  (LIBDIRAC, libdirac);
412.     REGISTER_ENCODER (LIBFAAC, libfaac);
413.     REGISTER_ENCDEC  (LIBGSM, libgsm);
414.     REGISTER_ENCDEC  (LIBGSM_MS, libgsm_ms);
415.     REGISTER_ENCODER (LIBMP3LAME, libmp3lame);
416.     REGISTER_ENCDEC  (LIBOPENCORE_AMRNB, libopencore_amrnb);
417.     REGISTER_DECODER (LIBOPENCORE_AMRWB, libopencore_amrwb);
418.     REGISTER_ENCDEC (LIBOPENJPEG, libopenjpeg);
419.     REGISTER_ENCDEC  (LIBSCHROEDINGER, libschroedinger);
420.     REGISTER_ENCDEC  (LIBSPEEX, libspeex);
421.     REGISTER_DECODER (LIBSTAGEFRIGHT_H264, libstagefright_h264);
422.     REGISTER_ENCODER (LIBTHEORA, libtheora);
423.     REGISTER_DECODER (LIBUTVIDEO, libutvideo);
424.     REGISTER_ENCODER (LIBVO_AACENC, libvo_aacenc);
425.     REGISTER_ENCODER (LIBVO_AMRWBENC, libvo_amrwbenc);
426.     REGISTER_ENCODER (LIBVORBIS, libvorbis);
427.     REGISTER_ENCDEC  (LIBVPX, libvpx);
428.     REGISTER_ENCODER (LIBX264, libx264);
429.     REGISTER_ENCODER (LIBX264RGB, libx264rgb);
430.     REGISTER_ENCODER (LIBXAVS, libxavs);
431.     REGISTER_ENCODER (LIBXVID, libxvid);
432.
433.     /* text */
434.     REGISTER_DECODER (BINTEXT, bintext);
435.     REGISTER_DECODER  (XBIN, xbin);
436.     REGISTER_DECODER  (IDF, idf);
437.
438.     /* parsers */
439.     REGISTER_PARSER  (AAC, aac);
440.     REGISTER_PARSER  (AAC_LATM, aac_latm);
441.     REGISTER_PARSER  (AC3, ac3);
442.     REGISTER_PARSER  (ADX, adx);
443.     REGISTER_PARSER  (CAVSVIDEO, cavsvideo);
444.     REGISTER_PARSER  (DCA, dca);
445.     REGISTER_PARSER  (DIRAC, dirac);
446.     REGISTER_PARSER  (DNXHD, dnxhd);
447.     REGISTER_PARSER  (DVBSUB, dvbsub);
448.     REGISTER_PARSER  (DVDSUB, dvdsub);
449.     REGISTER_PARSER  (FLAC, flac);
450.     REGISTER_PARSER  (GSM, gsm);
451.     REGISTER_PARSER  (H261, h261);
452.     REGISTER_PARSER  (H263, h263);
453.     REGISTER_PARSER  (H264, h264);
454.     REGISTER_PARSER  (MJPEG, mjpeg);
455.     REGISTER_PARSER  (MLP, mlp);
456.     REGISTER_PARSER  (MPEG4VIDEO, mpeg4video);
457.     REGISTER_PARSER  (MPEGAUDIO, mpegaudio);
458.     REGISTER_PARSER  (MPEGVIDEO, mpegvideo);
459.     REGISTER_PARSER  (PNM, pnm);
460.     REGISTER_PARSER  (RV30, rv30);
461.     REGISTER_PARSER  (RV40, rv40);
462.     REGISTER_PARSER  (VC1, vc1);
463.     REGISTER_PARSER  (VP3, vp3);
464.     REGISTER_PARSER  (VP8, vp8);
465.
466.     /* bitstream filters */
467.     REGISTER_BSF     (AAC_ADTSTOASC, aac_adtstoasc);
468.     REGISTER_BSF     (CHOMP, chomp);
469.     REGISTER_BSF     (DUMP_EXTRADATA, dump_extradata);
470.     REGISTER_BSF     (H264_MP4TOANNEXB, h264_mp4toannexb);
471.     REGISTER_BSF     (IMX_DUMP_HEADER, imx_dump_header);
472.     REGISTER_BSF     (MJPEG2JPEG, mjpeg2jpeg);
473.     REGISTER_BSF     (MJPEGA_DUMP_HEADER, mjpega_dump_header);
474.     REGISTER_BSF     (MP3_HEADER_COMPRESS, mp3_header_compress);
475.     REGISTER_BSF     (MP3_HEADER_DECOMPRESS, mp3_header_decompress);
476.     REGISTER_BSF     (MOV2TEXTSUB, mov2textsub);
477.     REGISTER_BSF     (NOISE, noise);
478.     REGISTER_BSF     (REMOVE_EXTRADATA, remove_extradata);
479.     REGISTER_BSF     (TEXT2MOVSUB, text2movsub);
480. }
```

整个代码的过程就是首先确定是不是已经初始化过了（initialized），如果没有，就注册，注册，注册...直到完成所有注册。

函数的调用关系图如下图所示。 av_register_all()调用了avcodec_register_all()。因此如果调用过 av_register_all()的话就不需要再调用 avcodec_register_all()了。



下面附上硬件加速器，编码器/解码器，parser，Bitstream Filter的注册代码。

硬件加速器注册函数是av_register_hwaccel()。

```cpp
void av_register_hwaccel(AVHWAccel *hwaccel)
{
    AVHWAccel **p = last_hwaccel;
    hwaccel->next = NULL;
    while(*p || avpriv_atomic_ptr_cas((void * volatile *)p, NULL, hwaccel))
        p = &(*p)->next;
    last_hwaccel = &hwaccel->next;
}
```

编解码器注册函数是avcodec_register()。

```cpp
av_cold void avcodec_register(AVCodec *codec)
{
    AVCodec **p;
    avcodec_init();
    p = last_avcodec;
    codec->next = NULL;

    while(*p || avpriv_atomic_ptr_cas((void * volatile *)p, NULL, codec))
        p = &(*p)->next;
    last_avcodec = &codec->next;

    if (codec->init_static_data)
        codec->init_static_data(codec);
}
```

parser注册函数是av_register_codec_parser()。

```cpp
void av_register_codec_parser(AVCodecParser *parser)
{
    do {
        parser->next = av_first_parser;
    } while (parser->next != avpriv_atomic_ptr_cas((void * volatile *)&av_first_parser, parser->next, parser));
}
```

Bitstream Filter注册函数是av_register_bitstream_filter()。

```cpp
void av_register_bitstream_filter(AVBitStreamFilter *bsf)
{
    do {
        bsf->next = first_bitstream_filter;
    } while(bsf->next != avpriv_atomic_ptr_cas((void * volatile *)&first_bitstream_filter, bsf->next, bsf));
}
```

后两个函数中的avpriv_atomic_ptr_cas()定义如下。

```cpp
void *avpriv_atomic_ptr_cas(void * volatile *ptr, void *oldval, void *newval)
{
    if (*ptr == oldval) {
        *ptr = newval;
        return oldval;
    }
    return *ptr;
}
```

文章标签： ffmpeg 源代码 avcodec_register_all 编码器 解码器

个人分类： FFMPEG

所属专栏： 开源多媒体项目源代码分析 FFmpeg