侧边栏

咸鱼Jay

FFmpeg介绍与编译

目录

- FFmpeg
  - FFmpeg核心模块
- FFmpeg编译

FFmpeg

FFmpeg是一套可以用来记录、转换数字音频、视频，并能将其转化为流的开源计算机程序。采用LGPL或GPL许可证。它提供了录制、转换以及流化音视频的完整解决方案。它包含了非常先进的音频/视频编解码库libavcodec，为了保证高可移植性和编解码质量，libavcodec里很多code都是从头开发的FFmpeg在Linux平台下开发，但它同样也可以在其它操作系统环境中编译运行，包括Windows、Mac OS X等。这个项目最早由Fabrice Bellard发起，2004年至2015年间由Michael Niedermayer主要负责维护。许多FFmpeg的开发人员都来自MPlayer项目，而且当前FFmpeg也是放在MPlayer项目组的服务器上。项目的名称来自MPEG视频编码标准，前面的"FF"代表"Fast Forward"。多媒体视频处理工具FFmpeg有非常强大的功能包括视频采集功能、视频格式转换、视频抓图、给视频加水印等。

FFmpeg核心模块

`libavformat`

用于各种音视频封装格式的生成和解析，包括获取解码所需信息以生成解码上下文结构和读取音视频帧等功能；音视频的格式解析协议，为 libavcodec 分析码流提供独立的音频或视频码流源。

`libavcodec`

用于各种类型声音/图像编解码；该库是音视频编解码核心，实现了市面上可见的绝大部分解码器的功能，libavcodec 库被其他各大解码器 ffdshow，Mplayer 等所包含或应用。

`libavfilter`

filter（FileIO、FPS、DrawText）音视频滤波器的开发，如水印、倍速播放等。

`libavutil`

包含一些公共的工具函数的使用库，包括算数运算 字符操作；

公告

侧边栏代码

昵称： 咸鱼Jay
园龄： 8年3个月
粉丝： 10
关注： 5
+加关注

< 2024年3月 >

日	一	二	三	四	五	六
25	26	27	28	29	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

- 1 FFmpeg核心模块

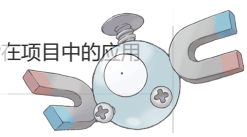
搜索

找随笔

0

最新随笔

1 流媒体通信中RTP/RTCP在项目中的应用



`libswresample`

原始音频格式转码。

`libswscale`

（原始视频格式转换）用于视频场景比例缩放、色彩映射转换；图像颜色空间或格式转换，如 rgb565,rgb888 等与 yuv420 等之间转换。

libpostproc+libavcodec

FFmpeg编译

> ndk最新的发布版是r17，如果只能找到r16的下载地址，那只要把这个地址中的16改成17就可以了

https://baike.baidu.com/item/ffmpeg/2665727?fr=aladdin

FFmpeg是一个开源的音视频处理程序，也可以把它看成一个用于处理音视频的库。如果需要在Android中使用这个库，那就需要进行交叉编译。

从官网下载FFmpeg的源码：

```
wget https://ffmpeg.org/releases/ffmpeg-4.0.2.tar.bz2
```

下载完成后使用 `tar` 工具解压：

```
tar xvf ffmpeg-4.0.2.tar.bz2
```

解压出来后进入ffmpeg解压目录，可以看到里面有各种文件、文档。需要关注的是 `configure` 文件。这个文件本身就是一个shell脚本，作用为生成 `makfile` 文件，然后使用 `make` 执行。

使用 `configure --help` 命令可以查看 `configure` 命令的一些参数说明。

使用 `./configure --help > configrue.txt` 可以把 `configure` 命令的参数说明导出到 `configrue.txt` 文件中

下面是交叉编译shell脚本内容：

```
#!/bin/bash
#这里定义变量，后续会使用
#TOOLCHAIN 变量指向ndk中的交叉编译gcc所在的目录
TOOLCHAIN=$NDK_ROOT/toolchains/arm-linux-androideabi-4.9/prebuilt/darwin-x86_64
#FLAGS与INCLUDES变量 可以从AS ndk工程的.externativeBuild/cmake/debug/armeabi-v7a/build.ninja中拷贝，需要注意的是**地址**
FLAGS="-isystem $NDK_ROOT/sysroot/usr/include/arm-linux-androideabi -D__ANDROID_API__=21 -g -DANDROID -ffunction-sections -funwind-tables -fstack-protector-strong -no-canonical-prefixes -march=armv7-a -mfloat-abi=softfp -mfpu=vfpv3-d16 -mthumb -Wa,--noexecstack -Wformat -Werror=format-security -std=c++11  -O0 -fPIC"
INCLUDES="-isystem $NDK_ROOT/sources/cxx-stl/llvm-libc++/include -isystem $NDK_ROOT/sources/android/support/include -isystem $NDK_ROOT/sources/cxx-stl/llvm-libc++abi/include"

#此变量用于编译完成之后的库与头文件存放在哪个目录
PREFIX=./android/armeabi-v7a2

#执行configure脚本，用于生成makefile
#--prefix : 安装目录
#--enable-small : 优化大小
#--disable-programs : 不编译ffmpeg程序(命令行工具)，我们是需要获得静态(动态)库。
#--disable-avdevice : 关闭avdevice模块，此模块在android中无用
#--disable-encoders : 关闭所有编码器 (播放不需要编码)
#--disable-muxers :  关闭所有复用器(封装器)，不需要生成mp4这样的文件，所以关闭
#--disable-filters :关闭视频滤镜
#--enable-cross-compile : 开启交叉编译 (ffmpeg比较**跨平台**,并不是所有库都有这么happy的选项 )
#--cross-prefix: 看右边的值应该就知道是干嘛的，gcc的前缀 xxx/xxx/xxx-gcc 则给xxx/xxx/xxx-
#disable-shared enable-static 不写也可以，默认就是这样的。
#--sysroot:
```


2.投屏项目中Sink端CPU占用过高问题

3.键鼠共享工具使用(软KVM)-barrier(symless开源版)

4.Android热点SoftAP使用方式

5.使用ConnectivityManager.bindProcessToNetwork绑定特殊网络

6.Android USB开发—USB通信

7.Android USB开发1—开发环境搭建

8.AOSP中获取系统签名

9.Android13源码下载环境搭建

10.AOSP12内置三方apk

积分与排名

积分 - 44152

排名 - 36626

随笔分类

Android(4)

C/C++(8)

FFmpeg(2)

- 1 FFmpeg核心模块

Framework(10)

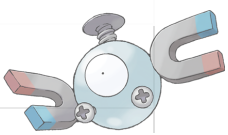
Handler(2)

ijkplayer(2)

Linux(6)

live555(2)

Miracast技术详解(8)



```
#--extra-cflags: 会传给gcc的参数
#--arch --target-os : 不给不行，为什么给这些值?
./configure \
--prefix=$PREFIX \
--enable-small \
--disable-programs \
--disable-avdevice \
--disable-encoders \
--disable-muxers \
--disable-filters \
--enable-cross-compile \
--cross-prefix=$TOOLCHAIN/bin/arm-linux-androideabi- \
--disable-shared \
--enable-static \
--sysroot=$NDK_ROOT/platforms/android-21/arch-arm \
--extra-cflags="$FLAGS $INCLUDES" \
--extra-cflags="-isysroot $NDK_ROOT/sysroot" \
--arch=arm \
--target-os=android

#上面运行脚本生成makefile之后，使用make执行脚本
make clean
make install
```

我的脚本示例

```
#!/bin/bash

#执行生成makefile的shell脚本
PREFIX=./android/armeabi-v7a2

NDK_ROOT=/home/zuojie/android-ndk-r17c

CPU=arm-linux-androideabi
TOOLCHAIN=$NDK_ROOT/toolchains/$CPU-4.9/prebuilt/linux-x86_64

FLAGS="-isystem $NDK_ROOT/sysroot/usr/include/arm-linux-androideabi -D__ANDROID_API__=21 -g -DANDROID -ffunction-sections -funwind-tables -fstack-protector-strong -no-canonical-prefixes -march=armv7-a -mfloat-abi=softfp -mfpu=vfpv3-d16 -mthumb -Wa,--noexecstack -Wformat -Werror=format-security  -O0 -fPIC"

INCLUDES=" -isystem $NDK_ROOT/sources/android/support/include"

# \ 换行连接符
./configure --prefix=$PREFIX \
--enable-small \
--disable-programs \
--disable-avdevice \
--disable-postproc \
--disable-encoders \
--disable-muxers \
--disable-filters \
--enable-cross-compile \
--cross-prefix=$TOOLCHAIN/bin/$CPU- \
--disable-shared \
--enable-static \
--sysroot=$NDK_ROOT/platforms/android-21/arch-arm \
--extra-cflags="$FLAGS $INCLUDES" \
--extra-cflags="-isysroot $NDK_ROOT/sysroot/" \
--arch=arm \
--target-os=android

# 清理一下
make clean
#执行makefile
make install
```

当在执行这个脚本时，如果报下面错，则说明没有安装make，则使用 `sudo apt install make` 命令安装make

```
......
Enabled indevs:
build.sh: 35: build.sh: make: not found
build.sh: 37: build.sh: make: not found
```

最后执行成功后会在 `./android/armeabi-v7a2/` 目录下生成三个文件 `include` -头文件，`lib` -静态库，`share` -示例

NDK(6)

其他(2)

问题解决(11)

音视频(12)

音视频基础学习(35)

侧边栏

随笔档案

2024年1月(2)

2023年12月(1)

2023年9月(4)

2023年7月(1)

2023年6月(1)

2023年5月(10)

2023年4月(2)

2023年3月(7)

2023年2月(5)

2023年1月(2)

2022年12月(2)

2022年11月(12)

2022年10月(29)

2022年9月(2)

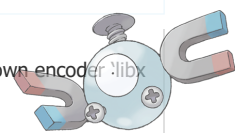
2022年8月(6)

更多

阅读排行榜

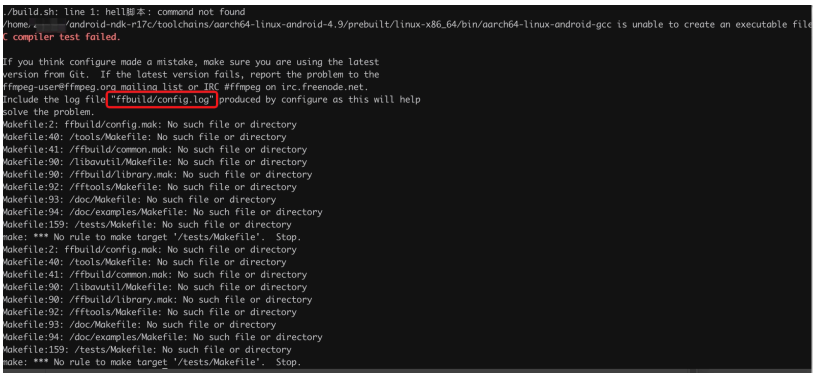
1 FFmpeg命令行之 Unknown encoder 'libx

1 FFmpeg核心模块

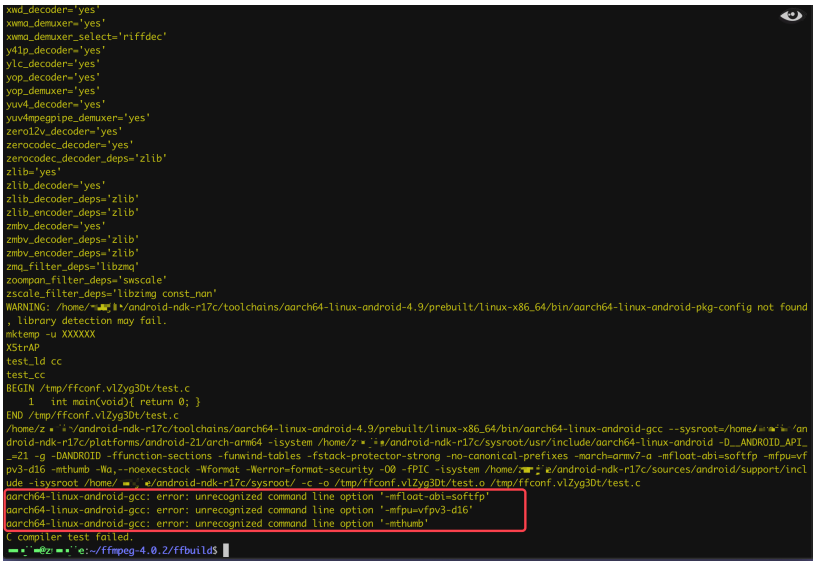


使用 `tar cvf ffmpeg.tar *` 命令将这三个文件夹压缩一下

当执行这个脚本后报错了，可以去ffmpeg的ffbuild/config.log文件，config.log是在编译时遇到问题时的记录日志，直接查看最后一行



我这使用 `aarch64-linux-android` 编译脚本报错后查看 `config.log` 文件，可以看出最后那三行说的很清楚是无法识别三个命令，需要将脚本中这三个命令删除，重新编译。



configure脚本文件参数说明：

```
1   Usage: configure [options]
2   Options: [defaults in brackets after descriptions]
3
4   帮助选项
5   Help options:
6     --help                   print this message [当前所看到的所有内容，就是用--help输出的帮助信息]
7     --quiet                  Suppress showing informative output [禁止显示信息输出，这个从来没有用过]
8     --list-decoders          show all available decoders [显示所有可用的解码器](非常常用)
9     --list-encoders          show all available encoders [显示所有可用的编码器](非常常用)
10    --list-hwaccels          show all available hardware accelerators [显示所有可用的硬件加速器]
11    --list-demuxers          show all available demuxers [显示所有可用的解复用器]
12    --list-muxers            show all available muxers [列出所有支持的封装]
13    --list-parsers           show all available parsers [显示所有可用的解析器]
14    --list-protocols         show all available protocols [列出支持的通信协议]
15    --list-bsfs              show all available bitstream filters [显示所有可用的位流筛选器]
16    --list-indevs            show all available input devices [显示所有可用的输入设备]
17    --list-outdevs           show all available output devices [显示所有可用的输出设备]
18    --list-filters           show all available filters [显示所有可用过滤器，当要做复杂特效时，就有用]
19
```


264)(1984)

2. Miracast技术详解（一）：Wi-Fi Display (1797)

侧边栏

3. 图像基础概念与YUV/RGB深入理解(1752)

4. Miracast技术详解（二）：RTSP协议(1728)

5. FFmpeg命令行之ffmpeg调整音视频播放速度 (1484)

评论排行榜

1. 17_详解YUV(1)

推荐排行榜

1. 28_FFmpeg音视频解封装格式(2)

2. Android USB开发—USB通信(1)

3. 29_SDL多线程与锁机制(1)

4. Android WifiDisplay分析一：相关Service的启动(1)

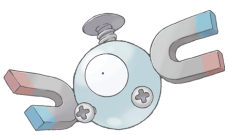
5. Miracast技术详解（四）：Sink源码解析(1)

最新评论

1 FFmpeg核心模块

大师的博客是学习音视频者的宝地。

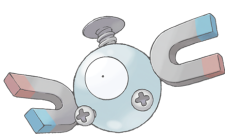
--hychyc_2008



```
标准选项
Standard options:
  --logfile=FILE           log tests and output to FILE [ffbuild/config.log] [编译时日志信息输出路径，默认： ffbuild/config. log]
  --disable-logging        do not log configure debug information [关闭:不记录配置调试信息]
  --fatal-warnings         fail if any configure warning is generated [如果生成任何配置警告则失败，注意:不能加，因为编译过程中,警告是会有的]
  --prefix=PREFIX          install in PREFIX [/usr/local]   [我们最终产出的目录，动态库，静态库] ( 非常常用)
  --bindir=DIR             install binaries in DIR [PREFIX/bin] [在目录中安装二进制文件] PREFIX
  --datadir=DIR            install data files in DIR [PREFIX/share/ffmpeg] [ 在目录中安装数据文件] PREFIX
  --docdir=DIR             install documentation in DIR [PREFIX/share/doc/ffmpeg] [在目录中安装文档] PREFIX
  --libdir=DIR             install libs in DIR [PREFIX/lib]  [安装库到指定目录]
  --shlibdir=DIR           install shared libs in DIR [LIBDIR]  [指定共享库路径]
  --incdir=DIR             install includes in DIR [PREFIX/include] [指定includes路径]
  --mandir=DIR             install man page in DIR [PREFIX/share/man] [指定man page路径]
  --pkgconfigdir=DIR       install pkg-config files in DIR [LIBDIR/pkgconfig] [在目录中安装pka配置文件]
  --enable-rpath           use rpath to allow installing libraries in paths
                           not part of the dynamic linker search path
                           use rpath when linking programs (USE WITH CARE) [使用rpath允许在路径中安装库，没有用过]
  --install-name-dir=DIR   Darwin directory name for installed targets [已安装目标的目录名]

许可选项
Licensing options:
  --enable-gpl             allow use of GPL code, the resulting libs [允许使用GPL]
                           and binaries will be under GPL [no]
  --enable-version3        upgrade (L)GPL to version 3 [no] 更新GPL版本
  --enable-nonfree         allow use of nonfree code, the resulting libs
                           and binaries will be unredistributable [no] [允许使用非免费程序]

配置选项
Configuration options:
  --disable-static         do not build static libraries [no] [关闭静态库(默认开启)]
  --enable-shared          build shared libraries [no]  [打开动态库(默认关闭)]
  --enable-small           optimize for size instead of speed   [优化大小]
  --disable-runtime-cpudetect disable detecting CPU capabilities at runtime (smaller binary) [禁用实时的CPU效率检测]
  --enable-gray            enable full grayscale support (slower color) [启用灰度(颜色、空间转换)]
  --disable-swscale-alpha  disable alpha channel support in swscale [禁用swscale中的 透明度]
  --disable-all            disable building components, libraries and programs [禁用所有，包括组 件、库、程序]
  --disable-autodetect     disable automatically detected external libraries [no]

项目选项
Program options:
  --disable-programs       do not build command line programs [禁止生成所有exe]
  --disable-ffmpeg         disable ffmpeg build [禁止生成ffmpeg.exe]
  --disable-ffplay         disable ffplay build [禁止生成ffplay.exe]
  --disable-ffprobe        disable ffprobe build [禁止生成 ffprobe.exe]

文档选项
Documentation options:
  --disable-doc            do not build documentation [禁止生成doc文件]
```

侧边栏

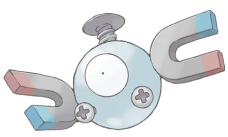
- 1 FFmpeg核心模块



```
  --disable-htmlpages      do not build HTML documentation pages [禁止生成HTML文档页]
  --disable-manpages       do not build man documentation pages [禁止生成帮助文档页]
  --disable-podpages       do not build POD documentation pages [禁止生成POD文档页]
  --disable-txtpages       do not build text documentation pages [禁止生成txt文档页]

组件选项(FFmpeg是由多个模块组成的)
Component options:
  --disable-avdevice       disable libavdevice build [模块之一(可以操控我们的摄像头)(Android中不支持)]
  --disable-avcodec        disable libavcodec build   [audio video codec(编码和解码)]
  --disable-avformat       disable libavformat build  [音视频格式生成和解析相关]
  --disable-swresample     disable libswresample build [音频重采样(如果想把单声道，变成双声道)]
  --disable-swscale        disable libswscale build [对视频显示相关(对视频的缩放，放大缩小)]
  --disable-postproc       disable libpostproc build [后期处理，很少用，可以关闭掉]
  --disable-avfilter       disable libavfilter build [给视频加水印，加字幕，特殊效果]
  --enable-avresample      enable libavresample build (deprecated) [no]
  --disable-pthreads       disable pthreads [autodetect]
  --disable-w32threads     disable Win32 threads [autodetect]
  --disable-os2threads     disable OS/2 threads [autodetect]
  --disable-network        disable network support [no]
  --disable-dct            disable DCT code
  --disable-dwt            disable DWT code
  --disable-error-resilience disable error resilience code
  --disable-lsp            disable LSP code
  --disable-lzo            disable LZO decoder code
  --disable-mdct           disable MDCT code
  --disable-rdft           disable RDFT code
  --disable-fft            disable FFT code
  --disable-faan           disable floating point AAN (I)DCT code
  --disable-pixelutils     disable pixel utils in libavutil

单个组件选项
Individual component options:
  --disable-everything     disable all components listed below [禁用下面已列出的全部组件]
  --disable-encoder=NAME   disable encoder NAME
  --enable-encoder=NAME    enable encoder NAME
  --disable-encoders       disable all encoders 如果我们去播放视频，不需要编码，那么此功能就可以关闭掉
  --disable-decoder=NAME   disable decoder NAME
  --enable-decoder=NAME    enable decoder NAME
  --disable-decoders       disable all decoders
  --disable-hwaccel=NAME   disable hwaccel NAME
  --enable-hwaccel=NAME    enable hwaccel NAME
  --disable-hwaccels       disable all hwaccels
  --disable-muxer=NAME     disable muxer NAME
  --enable-muxer=NAME      enable muxer NAME
  --disable-muxers         disable all muxers
  --disable-demuxer=NAME   disable demuxer NAME
  --enable-demuxer=NAME    enable demuxer NAME
  --disable-demuxers       disable all demuxers
  --enable-parser=NAME     enable parser NAME
  --disable-parser=NAME    disable parser NAME
  --disable-parsers        disable all parsers
  --enable-bsf=NAME        enable bitstream filter NAME
  --disable-bsf=NAME       disable bitstream filter NAME
  --disable-bsfs           disable all bitstream filters
  --enable-protocol=NAME   enable protocol NAME
  --disable-protocol=NAME  disable protocol NAME
  --disable-protocols      disable all protocols
  --enable-indev=NAME      enable input device NAME
  --disable-indev=NAME     disable input device NAME
  --disable-indevs         disable input devices
  --enable-outdev=NAME     enable output device NAME
  --disable-outdev=NAME    disable output device NAME
  --disable-outdevs        disable output devices
  --disable-devices        disable all devices
  --enable-filter=NAME     enable filter NAME
  --disable-filter=NAME    disable filter NAME
  --disable-filters        disable all filters
```

侧边栏

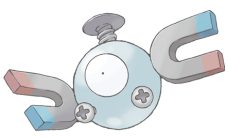
- 1 FFmpeg核心模块



```
添加依赖库支持
External library support:

  Using any of the following switches will allow FFmpeg to link to the
  corresponding external library. All the components depending on that library
  will become enabled, if all their other dependencies are met and they are not
  explicitly disabled. E.g. --enable-libwavpack will enable linking to
  libwavpack and allow the libwavpack encoder to be built, unless it is
  specifically disabled with --disable-encoder=libwavpack.

  Note that only the system libraries are auto-detected. All the other external
  libraries must be explicitly enabled.

  Also note that the following help text describes the purpose of the libraries
  themselves, not all their features will necessarily be usable by FFmpeg.

  --disable-alsa           disable ALSA support [autodetect]
  --disable-appkit         disable Apple AppKit framework [autodetect]
  --disable-avfoundation   disable Apple AVFoundation framework [autodetect]
  --enable-avisynth        enable reading of AviSynth script files [no]
  --disable-bzlib          disable bzlib [autodetect]
  --disable-coreimage      disable Apple CoreImage framework [autodetect]
  --enable-chromaprint     enable audio fingerprinting with chromaprint [no]
  --enable-frei0r          enable frei0r video filtering [no]
  --enable-gcrypt          enable gcrypt, needed for rtmp(t)e support
                           if openssl, librtmp or gmp is not used [no]
  --enable-gmp             enable gmp, needed for rtmp(t)e support
                           if openssl or librtmp is not used [no]
  --enable-gnutls          enable gnutls, needed for https support
                           if openssl or libtls is not used [no]
  --disable-iconv          disable iconv [autodetect]
  --enable-jni             enable JNI support [no]
  --enable-ladspa          enable LADSPA audio filtering [no]
  --enable-libaom          enable AV1 video encoding/decoding via libaom [no]
  --enable-libass          enable libass subtitles rendering,
                           needed for subtitles and ass filter [no]
  --enable-libbluray       enable BluRay reading using libbluray [no]
  --enable-libbs2b         enable bs2b DSP library [no]
  --enable-libcaca         enable textual display using libcaca [no]
  --enable-libcelt         enable CELT decoding via libcelt [no]
  --enable-libcdio         enable audio CD grabbing with libcdio [no]
  --enable-libcodec2       enable codec2 en/decoding using libcodec2 [no]
  --enable-libdc1394       enable IIDC-1394 grabbing using libdc1394
                           and libraw1394 [no]
  --enable-libfdk-aac      enable AAC de/encoding via libfdk-aac [no]
  --enable-libflite        enable flite (voice synthesis) support via libflite [no]
  --enable-libfontconfig   enable libfontconfig, useful for drawtext filter [no]
  --enable-libfreetype     enable libfreetype, needed for drawtext filter [no]
  --enable-libfribidi      enable libfribidi, improves drawtext filter [no]
  --enable-libgme          enable Game Music Emu via libgme [no]
  --enable-libgsm          enable GSM de/encoding via libgsm [no]
  --enable-libiec61883     enable iec61883 via libiec61883 [no]
  --enable-libilbc         enable iLBC de/encoding via libilbc [no]
  --enable-libjack         enable JACK audio sound server [no]
  --enable-libkvazaar      enable HEVC encoding via libkvazaar [no]
  --enable-libmodplug      enable ModPlug via libmodplug [no]
```

侧边栏

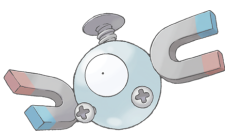
- 1 FFmpeg核心模块



```
  --enable-libmp3lame      enable MP3 encoding via libmp3lame [no]
  --enable-libopencore-amrnb enable AMR-NB de/encoding via
libopencore-amrnb [no]
  --enable-libopencore-amrwb enable AMR-WB decoding via libopencore-
amrwb [no]
  --enable-libopencv       enable video filtering via libopencv [no]
  --enable-libopenh264     enable H.264 encoding via OpenH264 [no]
  --enable-libopenjpeg     enable JPEG 2000 de/encoding via OpenJPEG
[no]
  --enable-libopenmpt      enable decoding tracked files via
libopenmpt [no]
  --enable-libopus         enable Opus de/encoding via libopus [no]
  --enable-libpulse        enable Pulseaudio input via libpulse [no]
  --enable-librsvg         enable SVG rasterization via librsvg [no]
  --enable-librubberband   enable rubberband needed for rubberband
filter [no]
  --enable-librtmp         enable RTMP[E] support via librtmp [no]
  --enable-libshine        enable fixed-point MP3 encoding via
libshine [no]
  --enable-libsmbclient    enable Samba protocol via libsmbclient
[no]
  --enable-libsnappy       enable Snappy compression, needed for hap
encoding [no]
  --enable-libsoxr         enable Include libsoxr resampling [no]
  --enable-libspeex        enable Speex de/encoding via libspeex
[no]
  --enable-libsrt          enable Haivision SRT protocol via libsrt
[no]
  --enable-libssh          enable SFTP protocol via libssh [no]
  --enable-libtesseract    enable Tesseract, needed for ocr filter
[no]
  --enable-libtheora       enable Theora encoding via libtheora [no]
  --enable-libtls          enable LibreSSL (via libtls), needed for
https support
                           if openssl or gnutls is not used [no]
  --enable-libtwolame      enable MP2 encoding via libtwolame [no]
  --enable-libv4l2         enable libv4l2/v4l-utils [no]
  --enable-libvidstab      enable video stabilization using vid.stab
[no]
  --enable-libvmaf         enable vmaf filter via libvmaf [no]
  --enable-libvo-amrwbenc  enable AMR-WB encoding via libvo-amrwbenc
[no]
  --enable-libvorbis       enable Vorbis en/decoding via libvorbis,
                           native implementation exists [no]
  --enable-libvpx          enable VP8 and VP9 de/encoding via libvpx
[no]
  --enable-libwavpack      enable wavpack encoding via libwavpack
[no]
  --enable-libwebp         enable WebP encoding via libwebp [no]
  --enable-libx264         enable H.264 encoding via x264 [no]
  --enable-libx265         enable HEVC encoding via x265 [no]
  --enable-libxavs         enable AVS encoding via xavs [no]
  --enable-libxcb          enable X11 grabbing using XCB
[autodetect]
  --enable-libxcb-shm      enable X11 grabbing shm communication
[autodetect]
  --enable-libxcb-xfixes   enable X11 grabbing mouse rendering
[autodetect]
  --enable-libxcb-shape    enable X11 grabbing shape rendering
[autodetect]
  --enable-libxvid         enable Xvid encoding via xvidcore,
                           native MPEG-4/Xvid encoder exists [no]
  --enable-libxml2         enable XML parsing using the C library
libxml2 [no]
  --enable-libzimg         enable z.lib, needed for zscale filter
[no]
  --enable-libzmq          enable message passing via libzmq [no]
  --enable-libzvbi         enable teletext support via libzvbi [no]
  --enable-lv2             enable LV2 audio filtering [no]
  --disable-lzma           disable lzma [autodetect]
  --enable-decklink        enable Blackmagic DeckLink I/O support
[no]
  --enable-libndi_newtek   enable Newteck NDI I/O support [no]
  --enable-mediacodec      enable Android MediaCodec support [no]
  --enable-libmysofa       enable libmysofa, needed for sofalizer
filter [no]
  --enable-openal          enable OpenAL 1.1 capture support [no]
  --enable-opencl          enable OpenCL processing [no]
  --enable-opengl          enable OpenGL rendering [no]
  --enable-openssl         enable openssl, needed for https support
                           if gnutls or libtls is not used [no]
```

侧边栏

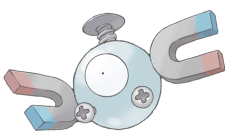
- 1 FFmpeg核心模块



```
  --disable-sndio          disable sndio support [autodetect]
  --disable-schannel       disable SChannel SSP, needed for TLS support on
                           Windows if openssl and gnutls are not used [autodetect]
  --disable-sdl2           disable sdl2 [autodetect]
  --disable-securetransport disable Secure Transport, needed for TLS support
                           on OSX if openssl and gnutls are not used [autodetect]
  --disable-xlib           disable xlib [autodetect]
  --disable-zlib           disable zlib [autodetect]

  The following libraries provide various hardware acceleration features:
  --disable-amf            disable AMF video encoding code [autodetect]
  --disable-audiotoolbox   disable Apple AudioToolbox code [autodetect]
  --enable-cuda-sdk        enable CUDA features that require the CUDA SDK [no]
  --disable-cuvid          disable Nvidia CUVID support [autodetect]
  --disable-d3d11va        disable Microsoft Direct3D 11 video acceleration code [autodetect]
  --disable-dxva2          disable Microsoft DirectX 9 video acceleration code [autodetect]
  --disable-ffnvcodec      disable dynamically linked Nvidia code [autodetect]
  --enable-libdrm          enable DRM code (Linux) [no]
  --enable-libmfx          enable Intel MediaSDK (AKA Quick Sync Video) code via libmfx [no]
  --enable-libnpp          enable Nvidia Performance Primitives-based code [no]
  --enable-mmal            enable Broadcom Multi-Media Abstraction Layer (Raspberry Pi) via MMAL [no]
  --disable-nvdec          disable Nvidia video decoding acceleration (via hwaccel) [autodetect]
  --disable-nvenc          disable Nvidia video encoding code [autodetect]
  --enable-omx             enable OpenMAX IL code [no]
  --enable-omx-rpi         enable OpenMAX IL code for Raspberry Pi [no]
  --enable-rkmpp           enable Rockchip Media Process Platform code [no]
  --disable-v4l2-m2m       disable V4L2 mem2mem code [autodetect]
  --disable-vaapi          disable Video Acceleration API (mainly Unix/Intel) code [autodetect]
  --disable-vdpau          disable Nvidia Video Decode and Presentation API for Unix code [autodetect]
  --disable-videotoolbox   disable VideoToolbox code [autodetect]

Toolchain options:
  --arch=ARCH              select architecture []
  --cpu=CPU                select the minimum required CPU (affects
                           instruction selection, may crash on older CPUs)
  --cross-prefix=PREFIX    use PREFIX for compilation tools []
  --progs-suffix=SUFFIX    program name suffix []
  --enable-cross-compile   assume a cross-compiler is used
  --sysroot=PATH           root of cross-build tree
  --sysinclude=PATH        location of cross-build system headers
  --target-os=OS           compiler targets OS []
  --target-exec=CMD        command to run executables on target
  --target-path=DIR        path to view of build directory on target
  --target-samples=DIR     path to samples directory on target
  --tempprefix=PATH        force fixed dir/prefix instead of mktemp for checks
  --toolchain=NAME         set tool defaults according to NAME
                           (gcc-asan, clang-asan, gcc-msan, clang-msan,
                           gcc-tsan, clang-tsan, gcc-usan, clang-usan,
                           valgrind-massif, valgrind-memcheck,
                           msvc, icl, gcov, llvm-cov, hardened)
  --nm=NM                  use nm tool NM [nm -g]
  --ar=AR                  use archive tool AR [ar]
  --as=AS                  use assembler AS []
  --ln_s=LN_S              use symbolic link tool LN_S [ln -s -f]
  --strip=STRIP            use strip tool STRIP [strip]
  --windres=WINDRES        use windows resource compiler WINDRES [windres]
```

侧边栏

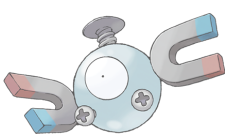
- 1 FFmpeg核心模块



```
  --x86asmexe=EXE          use nasm-compatible assembler EXE [nasm]
  --cc=CC                  use C compiler CC [gcc]
  --cxx=CXX                use C compiler CXX [g++]
  --objcc=OCC              use ObjC compiler OCC [gcc]
  --dep-cc=DEPCC           use dependency generator DEPCC [gcc]
  --nvcc=NVCC              use Nvidia CUDA compiler NVCC [nvcc]
  --ld=LD                  use linker LD []
  --pkg-config=PKGCONFIG   use pkg-config tool PKGCONFIG [pkg-
config]
  --pkg-config-flags=FLAGS pass additional flags to pkgconf []
  --ranlib=RANLIB          use ranlib RANLIB [ranlib]
  --doxygen=DOXYGEN        use DOXYGEN to generate API doc [doxygen]
  --host-cc=HOSTCC         use host C compiler HOSTCC
  --host-cflags=HCFLAGS    use HCFLAGS when compiling for host
  --host-cppflags=HCPPFLAGS use HCPPFLAGS when compiling for host
  --host-ld=HOSTLD         use host linker HOSTLD
  --host-ldflags=HLDFLAGS  use HLDFLAGS when linking for host
  --host-libs=HLIBS        use libs HLIBS when linking for host
  --host-os=OS             compiler host OS []
  --extra-cflags=ECFLAGS   add ECFLAGS to CFLAGS []
  --extra-cxxflags=ECFLAGS add ECFLAGS to CXXFLAGS []
  --extra-objcflags=FLAGS  add FLAGS to OBJCFLAGS []
  --extra-ldflags=ELDFLAGS add ELDFLAGS to LDFLAGS []
  --extra-ldexeflags=ELDFLAGS add ELDFLAGS to LDEXEFLAGS []
  --extra-ldsoflags=ELDFLAGS add ELDFLAGS to LDSOFLAGS []
  --extra-libs=ELIBS       add ELIBS []
  --extra-version=STRING   version string suffix []
  --optflags=OPTFLAGS      override optimization-related compiler
flags
  --nvccflags=NVCCFLAGS    override nvcc flags [-gencode
arch=compute_30,code=sm_30 -O2]
  --build-suffix=SUFFIX    library name suffix []
  --enable-pic             build position-independent code
  --enable-thumb           compile for Thumb instruction set
  --enable-lto             use link-time optimization
  --env="ENV=override"     override the environment variables

Advanced options (experts only):
  --malloc-prefix=PREFIX   prefix malloc and related names with
PREFIX
  --custom-allocator=NAME  use a supported custom allocator
  --disable-symver         disable symbol versioning
  --enable-hardcoded-tables use hardcoded tables instead of runtime
generation
  --disable-safe-bitstream-reader
                           disable buffer boundary checking in
bitreaders
                           (faster, but may crash)
  --sws-max-filter-size=N  the max filter size swscale uses [256]

Optimization options (experts only):
  --disable-asm            disable all assembly optimizations
  --disable-altivec        disable AltiVec optimizations
  --disable-vsx            disable VSX optimizations
  --disable-power8         disable POWER8 optimizations
  --disable-amd3dnow       disable 3DNow! optimizations
  --disable-amd3dnowext    disable 3DNow! extended optimizations
  --disable-mmx            disable MMX optimizations
  --disable-mmxext         disable MMXEXT optimizations
  --disable-sse            disable SSE optimizations
  --disable-sse2           disable SSE2 optimizations
  --disable-sse3           disable SSE3 optimizations
  --disable-ssse3          disable SSSE3 optimizations
  --disable-sse4           disable SSE4 optimizations
  --disable-sse42          disable SSE4.2 optimizations
  --disable-avx            disable AVX optimizations
  --disable-xop            disable XOP optimizations
  --disable-fma3           disable FMA3 optimizations
  --disable-fma4           disable FMA4 optimizations
  --disable-avx2           disable AVX2 optimizations
  --disable-avx512         disable AVX-512 optimizations
  --disable-aesni          disable AESNI optimizations
  --disable-armv5te        disable armv5te optimizations
  --disable-armv6          disable armv6 optimizations
  --disable-armv6t2        disable armv6t2 optimizations
  --disable-vfp            disable VFP optimizations
  --disable-neon           disable NEON optimizations
  --disable-inline-asm     disable use of inline assembly
  --disable-x86asm         disable use of standalone x86 assembly
  --disable-mipsdsp        disable MIPS DSP ASE R1 optimizations
  --disable-mipsdspr2      disable MIPS DSP ASE R2 optimizations
```

侧边栏

- 1 FFmpeg核心模块



```
  --disable-msa             disable MSA optimizations
  --disable-mipsfpu         disable floating point MIPS optimizations
  --disable-mmi             disable Loongson SIMD optimizations
  --disable-fast-unaligned consider unaligned accesses slow

Developer options (useful when working on FFmpeg itself):
  --disable-debug           disable debugging symbols
  --enable-debug=LEVEL      set the debug level []
  --disable-optimizations   disable compiler optimizations
  --enable-extra-warnings   enable more compiler warnings
  --disable-stripping       disable stripping of executables and
shared libraries
  --assert-level=level      0(default), 1 or 2, amount of assertion
testing,
                            2 causes a slowdown at runtime.
  --enable-memory-poisoning fill heap uninitialized allocated space
with arbitrary data
  --valgrind=VALGRIND       run "make fate" tests through valgrind to
detect memory
                            leaks and errors, using the specified
valgrind binary.
                            Cannot be combined with --target-exec
  --enable-ftrapv           Trap arithmetic overflows
  --samples=PATH            location of test samples for FATE, if not
set use
                            $FATE_SAMPLES at make invocation time.
  --enable-neon-clobber-test check NEON registers for clobbering
(should be
                            used only for debugging purposes)
  --enable-xmm-clobber-test check XMM registers for clobbering
(Win64-only;
                            should be used only for debugging
purposes)
  --enable-random           randomly enable/disable components
  --disable-random
  --enable-random=LIST      randomly enable/disable specific
components or
  --disable-random=LIST     component groups. LIST is a comma-
separated list
                            of NAME[:PROB] entries where NAME is a
component
                            (group) and PROB the probability
associated with
                            NAME (default 0.5).
  --random-seed=VALUE       seed value for --enable/disable-random
  --disable-valgrind-backtrace do not print a backtrace under
Valgrind
                            (only applies to --disable-optimizations
builds)
  --enable-osfuzz           Enable building fuzzer tool
  --libfuzzer=PATH          path to libfuzzer
  --ignore-tests=TESTS      comma-separated list (without "fate-"
prefix
                            in the name) of tests whose result is
ignored
  --enable-linux-perf       enable Linux Performance Monitor API

NOTE: Object files are built at the place where configure is
launched.

```

分类： FFmpeg

好文要顶 关注我 收藏该文 微信分享

咸鱼Jay
粉丝 - 10 关注 - 5

+加关注

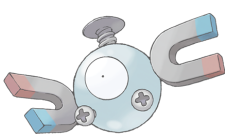
« 上一篇： ijkplayer编译-RTSP

» 下一篇： ffmpeg播放器-音视频解码流程

posted @ 2022-07-09 15:47 咸鱼Jay 阅读(719) 评论(0) 编辑 收藏 举报

侧边栏

- 1 FFmpeg核心模块



会员力量，点亮园子希望 刷新页面 返回顶部

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

【推荐】阿里云年度大降价：百款产品直降，平均降幅达20%
【推荐】园子的第一款简陋鼠标垫，是否是您值得拥有的周边
【推荐】会员力量，点亮园子希望，期待您升级成为园子会员

编辑推荐：
· 万字长文学会对接 AI 模型：Semantic Kernel 和 Kernel Memory
· .NET 高级调试之 sos 命令输出看不懂怎么办
· ASP.NET Core MVC 应用模型的构建[3]: Controller 的收集
· 现代 CSS 解决方案：accent-color 强调色
· 细聊 ASP.NET Core WebAPI 格式化程序

阅读排行：
· 万字长文学会对接 AI 模型：Semantic Kernel 和 Kernel Memory，工良出
· 10个技巧，3分钟教会你高效寻找开源项目
· 2024年，提升Windows开发和使用体验实践 - 小工具篇
· 面试官：说说SSO单点登录的实现原理?
· 万字带你走过数据库的这激荡的三年

侧边栏

页脚HTML代码

1 FFmpeg核心模块