原 FFmpeg源代码简单分析：makefile

2015年03月23日 11:22:29　阅读数：18482

===

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

===

本文记录FFmpeg的Makefile的源代码。Makefile用于编译FFmpeg的源代码。通过分析Makefile文件，可以了解FFmpeg的源代码生成的流程。有关Makefile这部分我本身基础不算很牢，很多地方还在慢慢摸索，所以分析的内容不能保证完全准确。以后有时间和其他朋友多交流再慢慢完善这篇文章。

PS：make有一个很有用的命令："make -n"。该选项会显示命令，但不会执行命令，十分有助于分析Makefile

FFmpeg中与Makefile相关的文件

FFmpeg中与Makefile相关的文件主要有以下几个：

根目录Makefile：最基本的Makefile；

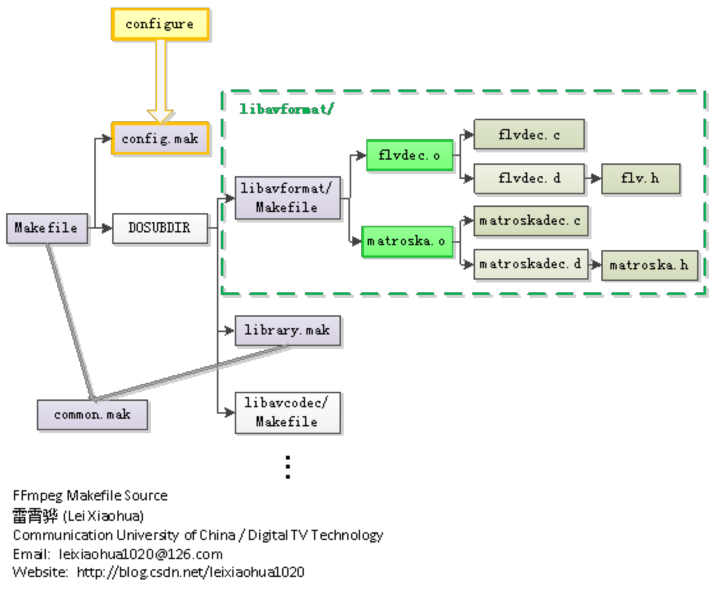
config.mak：由configure生成的Makefile，保存了Configure的设置信息；

libavXXXX/Makefile：每个类库的Makefile（仅仅设置了几个变量）；

library.mak：编译类库的Makefile（和libavXXXX/Makefile配合使用）；

common.mak：包含一些通用代码的Makefile；

它们之间的关系如下图所示。



图中除了画出了Makefile之外，还画出了和Makefile有关的一些文件：

XXX.c：C语言文件；

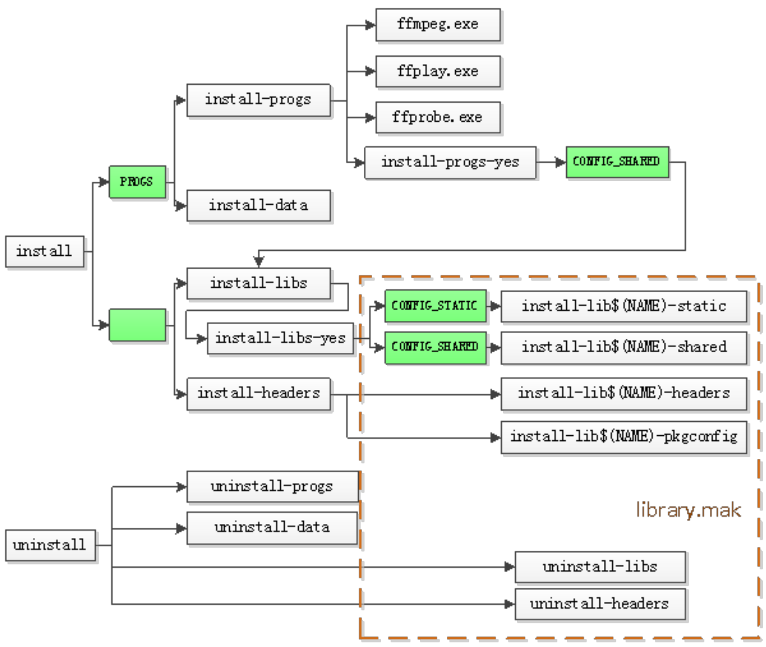
XXX.h：C语言文件用到的头文件；

XXX.o：C语言文件对应的目标文件；

XXX.d：C语言文件对应的依赖关系文件；

Make Install 之间的关系

简单分析了一下Makefile中的make install之间的关系，如下图所示（使用的是MinGW编译器）。



FFmpeg Makefile Source: install
雷霄骅 (Lei Xiaohua)
Communication University of China / Digital TV Technology
Email: leixiaohua1020@126.com
Website: http://blog.csdn.net/leixiaohua1020

从图中可以看出，install伪目标依赖于4个伪目标：

install-progs：安装应用程序ffmpeg.exe，ffplay.exe，ffprobe.exe；
install-data：安装数据（*.ffpreset之类的文件，没研究过）
install-libs：安装类库（libavcodec.dll等文件）
install-headers：安装头文件（libavcodec/avcodec.h等文件）

install-progs依赖于ffmpeg.exe，ffplay.exe，ffprobe.exe以及install-progs-yes伪目标。在CONFIG_SHARED取值为yes的情况下，install-progs-yes依赖于install-libs。

install-libs依赖于伪目标install-libs-yes。install-libs-yes的依赖关系位于library.mak文件中。如果CONFIG_STATIC取值为yes，install-libs-yes依赖于install-lib$(NAME)-static（其中${NAME}为类库文件名，例如avformat）；如果CONFIG_SHARED取值为yes，install-libs-yes依赖于install-lib$(NAME)-shared。
install-headers依赖于伪目标install-lib$(NAME)-headers和install-lib$(NAME)-pkgconfig。

和install相对应，uninstall伪目标依赖于4个伪目标：

uninstall-progs：卸载应用程序；
uninstall-data：卸载数据；
uninstall-libs：卸载类库；
uninstall-headers：卸载头文件；

其中uninstall-libs和uninstall-headers的依赖关系位于library.mak中。

根目录Makefile

根目录Makefile是最重要的。简单注释过的Makefile如下所示。

[python]

```python
# FFmpeg Main Makefile
#
# 注释：雷霄骅
# leixiaohua1020@126.com
# http://blog.csdn.net/leixiaohua1020
#
# FFmpeg的 Main Makefile。最重要。

MAIN_MAKEFILE=1
#重要：包含了configure信息
include config.mak

#config.mak中：
#SRC_PATH=.
#在SRC_PATH搜索各种类型的文件
vpath %.c    $(SRC_PATH)
vpath %.cpp  $(SRC_PATH)
vpath %.h    $(SRC_PATH)
vpath %.S    $(SRC_PATH)
vpath %.asm  $(SRC_PATH)
vpath %.v    $(SRC_PATH)
vpath %.texi $(SRC_PATH)
vpath %/fate_config.sh.template $(SRC_PATH)

#CONFIG_XXX取值为yes
#PROGS-yes= ffmpeg ffplay ffprobe
PROGS-$(CONFIG_FFMPEG)   += ffmpeg
PROGS-$(CONFIG_FFPLAY)   += ffplay
PROGS-$(CONFIG_FFPROBE)  += ffprobe
PROGS-$(CONFIG_FFSERVER) += ffserver

#config.mak中：
#EXESUF=.exe
#PROGSSUF=
#$(var:%.c=%.o)意思是把.c 为结尾的变量替换成.o。
#没有“.c”的时候，代表匹配所有
PROGS      := $(PROGS-yes:%=%$(EXESUF))
INSTPROGS   = $(PROGS-yes:%=%$(PROGSSUF)$(EXESUF))
OBJS        = $(PROGS-yes:%=%.o) cmdutils.o
TESTTOOLS   = audiogen videogen rotozoom tiny_psnr base64
HOSTPROGS  := $(TESTTOOLS:%=tests/%)
TOOLS       = qt-faststart trasher
TOOLS-$(CONFIG_ZLIB) += cws2fws

#PROGS= ffmpeg.exe ffplay.exe ffprobe.exe
#INSTPROGS= ffmpeg.exe ffplay.exe ffprobe.exe
#OBJS= ffmpeg.o ffplay.o ffprobe.o

BASENAMES   = ffmpeg ffplay ffprobe ffserver
ALLPROGS    = $(BASENAMES:%=%$(PROGSSUF)$(EXESUF))
ALLPROGS_G  = $(BASENAMES:%=%$(PROGSSUF)_g$(EXESUF))
ALLMANPAGES = $(BASENAMES:%=%.1)

#ALLPROGS= ffmpeg.exe ffplay.exe ffprobe.exe ffserver.exe
#ALLPROGS_G= ffmpeg_g.exe ffplay_g.exe ffprobe_g.exe ffserver_g.exe
#ALLMANPAGES=ffmpeg.1 ffplay.1 ffprobe.1 ffserver.1
FFLIBS-$(CONFIG_AVDEVICE) += avdevice
FFLIBS-$(CONFIG_AVFILTER) += avfilter
FFLIBS-$(CONFIG_AVFORMAT) += avformat
FFLIBS-$(CONFIG_AVCODEC)  += avcodec
FFLIBS-$(CONFIG_POSTPROC) += postproc
FFLIBS-$(CONFIG_SWRESAMPLE)+= swresample
FFLIBS-$(CONFIG_SWSCALE)  += swscale

#FFLIBS-yes= avdevice avfilter avformat avcodec postproc swresample swscale
#一定需要libavutil
FFLIBS := avutil
#让通配符在变量中展开，需要使用wildcard关键字
DATA_FILES := $(wildcard $(SRC_PATH)/presets/*.ffpreset) $(SRC_PATH)/doc/ffprobe.xsd

SKIPHEADERS = cmdutils_common_opts.h
#重要
include $(SRC_PATH)/common.mak
#依赖的类库
FF_EXTRALIBS := $(FFEXTRALIBS)
FF_DEP_LIBS  := $(DEP_LIBS)

#伪目标
#all是最关键的，生成最后的程序
#all: ffmpeg.exe ffplay.exe ffprobe.exe
#
all: $(PROGS)
#config.mak中：
#EXESUF=.exe
#PROGSSUF=
#$@是一个自动化变量。可以简单理解为目标的集合。
#$<是一个自动化变量。可以简单理解为依赖目标的集合。
#%是通配符
#两个冒号，“静态模式规则”。
#ffmpeg_g.exe生成ffmpeg.exe；fflplay_g.exe生成ffplay.exe；ffprobe_g.exe生成ffprobe.exe
#strip经常用来去除目标文件中的一些符号表、调试符号表信息,以减小程序的大小
$(PROGS): %$(EXESUF): %$(PROGSSUF)_g$(EXESUF)
    $(CP) $< $@$(PROGSSUF)
    $(STRIP) $@$(PROGSSUF)

$(TOOLS): %$(EXESUF): %.o
    $(LD) $(LDFLAGS) -o $@ $< $(ELIBS)

tools/cws2fws$(EXESUF): ELIBS = -lz

config.h: .config
.config: $(wildcard $(FFLIBS:%=$(SRC_PATH)/lib%/all*.c))
    @-tput bold 2>/dev/null
    @-printf '\nWARNING: $(?F) newer than config.h, rerun configure\n\n'
    @-tput sgr0 2>/dev/null
#给子目录中的Makefile使用的变量
SUBDIR_VARS := OBJS FFLIBS CLEANFILES DIRS TESTPROGS EXAMPLES SKIPHEADERS \
               ALTIVEC-OBJS MMX-OBJS NEON-OBJS X86-OBJS YASM-OBJS-FFT YASM-OBJS \
               HOSTPROGS BUILT_HEADERS TESTOBJS ARCH_HEADERS ARMV6-OBJS TOOLS

define RESET
```

```
$(1) :=
$(1)-yes :=
endef

#$(call <expression>,<parm1>,<parm2>,<parm3>...)
#当make执行这个函数时，<expression>参数中的变量，如$(1)，$(2)，$(3)等，会被参数
#<parm1>，<parm2>，<parm3>依次取代。而<expression>的返回值就是call函数的返回值。

#命令包============================
#用于编译每个库
#$(1)取值为libavcodec，libavcodec等等
define DOSUBDIR
$(foreach V,$(SUBDIR_VARS),$(eval $(call RESET,$(V))))
SUBDIR := $(1)/
#每个库目录下的Makefile
include $(SRC_PATH)/$(1)/Makefile
#注：make一般情况下如果在中途检测到有执行出错的情况(返回非 0 状态)，那么就会放弃对当前规则后续命令的执行。
#在命令前面加上“-”号之后，就算执行错误了，也会继续执行下去
-include $(SRC_PATH)/$(1)/$(ARCH)/Makefile
#编译类库
include $(SRC_PATH)/library.mak
endef
#=================================

#$(foreach <var>,<list>,<text>)
#把参数<list>中的单词逐一取出放到参数<var>所指定的变量中，
#然后再执行<text>所包含的表达式。
#$(eval text)
#text的内容将作为makefile的一部分而被make解析和执行
#
#循环调用DOSUBDIR命令包
#这一步会将libavcodec，libavformat等文件夹下的Makefile包含进来。
$(foreach D,$(FFLIBS),$(eval $(call DOSUBDIR,lib$(D))))
#ffplay需要SDL
ffplay.o: CFLAGS += $(SDL_CFLAGS)
ffplay_g$(EXESUF): FF_EXTRALIBS += $(SDL_LIBS)
ffserver_g$(EXESUF): LDFLAGS += $(FFSERVERLDFLAGS)
#链接生成ffmpeg_g.exe等等
#FF_DEP_LIBS= libavcodec/libavcodec.a libavutil/libavutil.a ....
%$(PROGSSUF)_g$(EXESUF): %.o cmdutils.o $(FF_DEP_LIBS)
	$(LD) $(LDFLAGS) -o $@ $< cmdutils.o $(FF_EXTRALIBS)

OBJDIRS += tools

-include $(wildcard tools/*.d)

VERSION_SH  = $(SRC_PATH)/version.sh
GIT_LOG     = $(SRC_PATH)/.git/logs/HEAD

.version: $(wildcard $(GIT_LOG)) $(VERSION_SH) config.mak
.version: M=@

version.h .version:
	$(M)$(VERSION_SH) $(SRC_PATH) version.h $(EXTRA_VERSION)
	$(Q)touch .version

# force version.sh to run whenever version might have changed
-include .version
#安装install
#安装程序
ifdef PROGS
install: install-progs install-data
endif
#安装类库和头文件
install: install-libs install-headers
#install-libs-yes位于library.mak
install-libs: install-libs-yes

install-progs-yes:
install-progs-$(CONFIG_SHARED): install-libs

#config.mak中：
#BINDIR=$(DESTDIR)${prefix}/bin
#INSTALL=install
#cp与install区别：
#cp会先清空文件后往里写入新文件，而install则会先删除掉原先的文件然后写入新文件。
install-progs: install-progs-yes $(PROGS)
	$(Q)mkdir -p "$(BINDIR)"
	$(INSTALL) -c -m 755 $(INSTPROGS) "$(BINDIR)"

install-data: $(DATA_FILES)
	$(Q)mkdir -p "$(DATADIR)"
	$(INSTALL) -m 644 $(DATA_FILES) "$(DATADIR)"
#卸载
uninstall: uninstall-libs uninstall-headers uninstall-progs uninstall-data
#addprefix()用于加前缀
#在这里获取ffmpeg.exe等的完整路径（用于删除）
uninstall-progs:
	$(RM) $(addprefix "$(BINDIR)/", $(ALLPROGS))

uninstall-data:
	$(RM) -r "$(DATADIR)"
#清空
clean::
	$(RM) $(ALLPROGS) $(ALLPROGS_G)
	$(RM) $(CLEANSUFFIXES)
	$(RM) $(TOOLS)
	$(RM) $(CLEANSUFFIXES:%=tools/%)
	$(RM) coverage.info
	$(RM) -r coverage-html

distclean::
	$(RM) $(DISTCLEANSUFFIXES)
	$(RM) config.* .version version.h libavutil/avconfig.h

config:
	$(SRC_PATH)/configure $(value FFMPEG_CONFIGURATION)

# Without the sed genthml thinks "libavutil" and "./libavutil" are two different things
coverage.info: $(wildcard *.gcda *.gcno */*.gcda */*.gcno */*/*.gcda */*/*.gcno)
	$(Q)lcov -c -d . -b . | sed -e 's#/./#/#g' > $@

coverage-html: coverage.info
	$(Q)mkdir -p $@
	$(Q)genhtml -o $@ $<
	$(Q)touch $@

include $(SRC_PATH)/doc/Makefile
include $(SRC_PATH)/tests/Makefile

$(sort $(OBJDIRS)):
	$(Q)mkdir -p $@

# Dummy rule to stop make trying to rebuild removed or renamed headers
%.h:
	@:

# Disable suffix rules.  Most of the builtin rules are suffix rules,
# so this saves some time on slow systems.
.SUFFIXES:
#显示地指明一个目标是“伪目标”
.PHONY: all all-yes alltools *clean config examples install*
.PHONY: testprogs uninstall*
```

根目录Makefile代码一开始的时候包含了config.mak文件。这个文件是运行./configure的后生成的配置文件，包含了所有的配置信息。

随后代码定义了ffplay.exe，ffmpeg.exe，ffprobe.exe与ffplay_g.exe，ffmpeg_g.exe，ffprobe_g.exe的依赖关系。然后定义了ffplay_g.exe，ffmpeg_g.exe，ffprobe_g.exe与libavformat，libavcodec等这些类库的依赖关系。

根目录Makefile中也定义了all，install，uninstall，clean等等一系列的伪目标，这样可以通过给Makefile指定不同的目标来完成不同的事。

此外根目录的Makefile中有一个很重要的命令包DOSUBDIR。在该命令包通过包含libavXXX/Makefile和library.mak等文件，定义了FFmpeg类库（例如libavformat，libavcodec，libavutil等）的依赖关系。

config.mak

config.mak文件是运行./configure的后生成的配置文件，包含了所有的配置信息。简单注释过的config.mak的代码如下所示。

[python]

```
# FFmpeg config.mak
#
# 注释：雷霄骅
# leixiaohua1020@126.com
# http://blog.csdn.net/leixiaohua1020
#
# Configure脚本生成的Makefile，包含了各种配置信息。
#
# Automatically generated by configure - do not modify!
#基本信息
ifndef FFMPEG_CONFIG_MAK
FFMPEG_CONFIG_MAK=1
FFMPEG_CONFIGURATION=
#各种路径=======================================
prefix=/usr/local
LIBDIR=$(DESTDIR)${prefix}/lib
SHLIBDIR=$(DESTDIR)${prefix}/bin
INCDIR=$(DESTDIR)${prefix}/include
BINDIR=$(DESTDIR)${prefix}/bin
DATADIR=$(DESTDIR)${prefix}/share/ffmpeg
MANDIR=$(DESTDIR)${prefix}/share/man
#是个相对路径
SRC_PATH=.
ifndef MAIN_MAKEFILE
SRC_PATH:=$(SRC_PATH:.%=..%)
endif
#工具集=========================================
CC_IDENT=gcc 4.6.2 (GCC)
#架构
ARCH=x86
#编译器
CC=gcc
CXX=g++
AS=gcc
#链接器
LD=gcc
DEPCC=gcc
#汇编器
YASM=yasm
YASMDEP=yasm
#生成静态库.a工具
AR=ar
RANLIB=ranlib
CP=cp -p
LN_S=ln -sf
STRIP=strip
#参数集=========================================
#编译器的参数
CPPFLAGS= -D_ISOC99_SOURCE -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -U__STRICT_ANSI__
CFLAGS=   -std=c99 -fno-common -fomit-frame-pointer -I/include/SDL -D_GNU_SOURCE=1 -Dmain=SDL_main -g -Wdeclaration-after-statement -Wall -Wno-parentheses -Wno-switch -Wno-format-zero-length -Wdisabled-optimization -Wpointer-arith -Wredundant-decls -Wno-pointer-sign -Wcast-qual -Wwrite-strings -Wtype-limits -Wundef -Wmissing-prototypes -Wno-pointer-to-int-cast -Wstrict-prototypes -O3 -fno-math-errno -fno-signed-zeros -fno-tree-vectorize -Werror=implicit-function-declaration -Werror=missing-prototypes
CXXFLAGS=  -D__STDC_CONSTANT_MACROS
ASFLAGS=   -g
#目标文件有关的参数
AS_O=-o $@
CC_O=-o $@
CXX_O=-o $@
#链接器有关的参数
LDFLAGS= -Wl,--as-needed -Wl,--warn-common -Wl,-rpath-link=libpostproc:libswresample:libswscale:libavfilter:libavdevice:libavformat:libavcodec:libavutil
FFSERVERLDFLAGS=-Wl,-E
SHFLAGS=-shared -Wl,--output-def,$$(@:$(SLIBSUF)=.def) -Wl,--out-implib,$(SUBDIR)lib$(SLIBNAME:$(SLIBSUF)=.dll.a) -Wl,--enable-runtime-pseudo-reloc -Wl,--enable-auto-image-base -Wl,-Bsymbolic -Wl,--version-script,$(SUBDIR)lib$(NAME).ver
YASMFLAGS=-f win32  -DPREFIX
#前缀后缀========================================
BUILDSUF=
PROGSSUF=
#${NAME}位于每个liavXXX/Makefile中，例如avformat
FULLNAME=$(NAME)$(BUILDSUF)
LIBPREF=lib
LIBSUF=.a
#例如libavformat.a
LIBNAME=$(LIBPREF)$(FULLNAME)$(LIBSUF)
SLIBPREF=
SLIBSUF=.dll
EXESUF=.exe
EXTRA_VERSION=
DEPFLAGS=$(CPPFLAGS) $(CFLAGS) -MM
CCDEP=
CXXDEP=$(DEPCC) $(DEPFLAGS) $< | sed -e "/^\#.*/d" -e "s,^[[:space:]]*$(*F)\\.o,$(@D)/$(*F).o," > $(@:.o=.d)
ASDEP=
CC_DEPFLAGS=-MMD -MF $(@:.o=.d) -MT $@
AS_DEPFLAGS=-MMD -MF $(@:.o=.d) -MT $@
HOSTCC=gcc
HOSTCFLAGS=-D_ISOC99_SOURCE -O3 -g -std=c99 -Wall
HOSTEXESUF=.exe
HOSTLDFLAGS=
HOSTLIBS=-lm
TARGET_EXEC=
TARGET_PATH=$(CURDIR)
#SDL
SDL_LIBS=-L/lib -lmingw32 -lSDLmain -lSDL -mwindows
SDL_CFLAGS=-I/include/SDL -D_GNU_SOURCE=1 -Dmain=SDL_main
LIB_INSTALL_EXTRA_CMD=$$(RANLIB) "$(LIBDIR)/$(LIBNAME)"
#链接
EXTRALIBS=-lavicap32 -lws2_32 -L/lib -lmingw32 -lSDLmain -lSDL -mwindows -lm -lz -lpsapi
INSTALL=install
LIBTARGET=i386
#例如libavformat.dll
SLIBNAME=$(SLIBPREF)$(FULLNAME)$(SLIBSUF)
#LIBVERSION变量位于library.mak
#例如libavformat-53.dll
#生成的Dll似乎就是这个版本的
SLIBNAME_WITH_VERSION=$(SLIBPREF)$(FULLNAME)-$(LIBVERSION)$(SLIBSUF)
#例如libavformat-53.31.100.dll
SLIBNAME_WITH_MAJOR=$(SLIBPREF)$(FULLNAME)-$(LIBMAJOR)$(SLIBSUF)
SLIB_CREATE_DEF_CMD=
#生成导出库lib，会调用lib.exe
SLIB_EXTRA_CMD=-lib.exe /machine:$(LIBTARGET) /def:$$(@:$(SLIBSUF)=.def) /out:$(SUBDIR)$(SLIBNAME:$(SLIBSUF)=.lib)
SLIB_INSTALL_NAME=$(SLIBNAME_WITH_MAJOR)
SLIB_INSTALL_LINKS=
SLIB_INSTALL_EXTRA_LIB=lib$(SLIBNAME:$(SLIBSUF)=.dll.a) $(SLIBNAME_WITH_MAJOR:$(SLIBSUF)=.def)
SLIB_INSTALL_EXTRA_SHLIB=$(SLIBNAME:$(SLIBSUF)=.lib)
SAMPLES:=$(FATE_SAMPLES)
NOREDZONE_FLAGS=-mno-red-zone
#版本信息========================================
libavcodec_VERSION=53.60.100
libavcodec_VERSION_MAJOR=53
libavdevice_VERSION=53.4.100
libavdevice_VERSION_MAJOR=53
libavfilter_VERSION=2.60.100
libavfilter_VERSION_MAJOR=2
libavformat_VERSION=53.31.100
libavformat_VERSION_MAJOR=53
libavutil_VERSION=51.34.101
libavutil_VERSION_MAJOR=51
libpostproc_VERSION=52.0.100
libpostproc_VERSION_MAJOR=52
libswresample_VERSION=0.6.100
libswresample_VERSION_MAJOR=0
libswscale_VERSION=2.1.100
libswscale_VERSION_MAJOR=2
#组件配置========================================
#ARCH_
!ARCH_ALPHA=yes
!ARCH_ARM=yes
!ARCH_AVR32=yes
!ARCH_AVR32_AP=yes
!ARCH_AVR32_UC=yes
!ARCH_BFIN=yes
!ARCH_IA64=yes
!ARCH_M68K=yes
!ARCH_MIPS=yes
!ARCH_MIPS64=yes
!ARCH_PARISC=yes
!ARCH_PPC=yes
!ARCH_PPC64=yes
!ARCH_S390=yes
!ARCH_SH4=yes
!ARCH_SPARC=yes
!ARCH_SPARC64=yes
!ARCH_TOMI=yes
ARCH_X86=yes
ARCH_X86_32=yes
!ARCH_X86_64=yes
#HAVE_
!HAVE_ALTIVEC=yes
HAVE_AMD3DNOW=yes
HAVE_AMD3DNOWEXT=yes
!HAVE_ARMV5TE=yes
!HAVE_ARMV6=yes
!HAVE_ARMV6T2=yes
!HAVE_ARMVFP=yes
HAVE_AVX=yes
!HAVE_IWMMXT=yes
!HAVE_MMI=yes
HAVE_MMX=yes
HAVE_MMX2=yes
!HAVE_NEON=yes
!HAVE_PPC4XX=yes
HAVE_SSE=yes
HAVE_SSSE3=yes
!HAVE_VFPV3=yes
!HAVE_VIS=yes
!HAVE_BIGENDIAN=yes
HAVE_FAST_UNALIGNED=yes
!HAVE_PTHREADS=yes
HAVE_W32THREADS=yes
!HAVE_OS2THREADS=yes
HAVE_ALIGNED_STACK=yes
!HAVE_ALSA_ASOUNDLIB_H=yes
!HAVE_ALTIVEC_H=yes
!HAVE_ARPA_INET_H=yes
!HAVE_ASM_MOD_Y=yes
!HAVE_ASM_TYPES_H=yes
HAVE_ATTRIBUTE_MAY_ALIAS=yes
HAVE_ATTRIBUTE_PACKED=yes
HAVE_CBRTF=yes
HAVE_CLOSESOCKET=yes
!HAVE_CMOV=yes
!HAVE_DCBZL=yes
!HAVE_DEV_BKTR_IOCTL_BT848_H=yes
!HAVE_DEV_BKTR_IOCTL_METEOR_H=yes
!HAVE_DEV_IC_BT8XX_H=yes
!HAVE_DEV_VIDEO_BKTR_IOCTL_BT848_H=yes
!HAVE_DEV_VIDEO_METEOR_IOCTL_METEOR_H=yes
!HAVE_DLFCN_H=yes
!HAVE_DLOPEN=yes
HAVE_DOS_PATHS=yes
HAVE_EBP_AVAILABLE=yes
HAVE_EBX_AVAILABLE=yes
HAVE_EXP2=yes
HAVE_EXP2F=yes
!HAVE_FAST_64BIT=yes
HAVE_FAST_CLZ=yes
!HAVE_FAST_CMOV=yes
!HAVE_FCNTL=yes
```

```
!HAVE_FORK=yes
!HAVE_GETADDRINFO=yes
!HAVE_GETHRTIME=yes
HAVE_GETPROCESSAFFINITYMASK=yes
HAVE_GETPROCESSMEMORYINFO=yes
HAVE_GETPROCESSTIMES=yes
!HAVE_GETRUSAGE=yes
HAVE_GNU_AS=yes
!HAVE_IBM_ASM=yes
!HAVE_INET_ATON=yes
HAVE_INLINE_ASM=yes
HAVE_ISATTY=yes
HAVE_KBHIT=yes
!HAVE_LDBRX=yes
HAVE_LLRINT=yes
HAVE_LLRINTF=yes
HAVE_LOCAL_ALIGNED_16=yes
HAVE_LOCAL_ALIGNED_8=yes
!HAVE_LOCALTIME_R=yes
HAVE_LOG2=yes
HAVE_LOG2F=yes
!HAVE_LOONGSON=yes
HAVE_LRINT=yes
HAVE_LRINTF=yes
!HAVE_LZO1X_999_COMPRESS=yes
!HAVE_MACHINE_IOCTL_BT848_H=yes
!HAVE_MACHINE_IOCTL_METEOR_H=yes
HAVE_MAKEINFO=yes
HAVE_MALLOC_H=yes
HAVE_MAPVIEWOFFILE=yes
!HAVE_MEMALIGN=yes
!HAVE_MKSTEMP=yes
!HAVE_MMAP=yes
HAVE_PEEKNAMEDPIPE=yes
!HAVE_POLL_H=yes
!HAVE_POSIX_MEMALIGN=yes
HAVE_ROUND=yes
HAVE_ROUNDF=yes
!HAVE_SCHED_GETAFFINITY=yes
HAVE_SDL=yes
HAVE_SDL_VIDEO_SIZE=yes
HAVE_SETMODE=yes
!HAVE_SETRLIMIT=yes
!HAVE_SNDIO_H=yes
HAVE_SOCKLEN_T=yes
!HAVE_SOUNDCARD_H=yes
!HAVE_STRERROR_R=yes
!HAVE_STRPTIME=yes
HAVE_STRUCT_ADDRINFO=yes
HAVE_STRUCT_IPV6_MREQ=yes
!HAVE_STRUCT_RUSAGE_RU_MAXRSS=yes
HAVE_STRUCT_SOCKADDR_IN6=yes
!HAVE_STRUCT_SOCKADDR_SA_LEN=yes
HAVE_STRUCT_SOCKADDR_STORAGE=yes
!HAVE_STRUCT_V4L2_FRMIVALENUM_DISCRETE=yes
HAVE_SYMVER=yes
HAVE_SYMVER_ASM_LABEL=yes
!HAVE_SYMVER_GNU_ASM=yes
!HAVE_SYSCONF=yes
!HAVE_SYSCTL=yes
!HAVE_SYS_MMAN_H=yes
HAVE_SYS_PARAM_H=yes
!HAVE_SYS_RESOURCE_H=yes
!HAVE_SYS_SELECT_H=yes
!HAVE_SYS_SOUNDCARD_H=yes
!HAVE_SYS_VIDEOIO_H=yes
!HAVE_TERMIOS_H=yes
HAVE_THREADS=yes
HAVE_TRUNC=yes
HAVE_TRUNCF=yes
!HAVE_VFP_ARGS=yes
HAVE_VIRTUALALLOC=yes
HAVE_WINSOCK2_H=yes
!HAVE_XFORM_ASM=yes
!HAVE_XMM_CLOBBERS=yes
HAVE_YASM=yes
#CONFIG_
CONFIG_BSFS=yes
CONFIG_DECODERS=yes
CONFIG_DEMUXERS=yes
CONFIG_ENCODERS=yes
CONFIG_FILTERS=yes
!CONFIG_HWACCELS=yes
CONFIG_INDEVS=yes
CONFIG_MUXERS=yes
CONFIG_OUTDEVS=yes
CONFIG_PARSERS=yes
CONFIG_PROTOCOLS=yes
CONFIG_FFPLAY=yes
CONFIG_FFPROBE=yes
!CONFIG_FFSERVER=yes
```

```
CONFIG_FFMPEG=yes
!CONFIG_AVPLAY=yes
!CONFIG_AVPROBE=yes
!CONFIG_AVSERVER=yes
CONFIG_AANDCT=yes
CONFIG_AC3DSP=yes
CONFIG_AVCODEC=yes
CONFIG_AVDEVICE=yes
CONFIG_AVFILTER=yes
CONFIG_AVFORMAT=yes
!CONFIG_AVISYNTH=yes
!CONFIG_BZLIB=yes
!CONFIG_CRYSTALHD=yes
CONFIG_DCT=yes
!CONFIG_DOC=yes
CONFIG_DWT=yes
!CONFIG_DXVA2=yes
CONFIG_FASTDIV=yes
CONFIG_FFT=yes
!CONFIG_FREI0R=yes
!CONFIG_GNUTLS=yes
CONFIG_GOLOMB=yes
!CONFIG_GPL=yes
!CONFIG_GRAY=yes
CONFIG_H264CHROMA=yes
CONFIG_H264DSP=yes
CONFIG_H264PRED=yes
!CONFIG_HARDCODED_TABLES=yes
CONFIG_HUFFMAN=yes
!CONFIG_LIBAACPLUS=yes
!CONFIG_LIBASS=yes
!CONFIG_LIBCDIO=yes
!CONFIG_LIBCELT=yes
!CONFIG_LIBDC1394=yes
!CONFIG_LIBDIRAC=yes
!CONFIG_LIBFAAC=yes
!CONFIG_LIBFREETYPE=yes
!CONFIG_LIBGSM=yes
!CONFIG_LIBMODPLUG=yes
!CONFIG_LIBMP3LAME=yes
!CONFIG_LIBNUT=yes
!CONFIG_LIBOPENCORE_AMRNB=yes
!CONFIG_LIBOPENCORE_AMRWB=yes
!CONFIG_LIBOPENCV=yes
!CONFIG_LIBOPENJPEG=yes
!CONFIG_LIBPULSE=yes
!CONFIG_LIBRTMP=yes
!CONFIG_LIBSCHROEDINGER=yes
!CONFIG_LIBSPEEX=yes
!CONFIG_LIBSTAGEFRIGHT_H264=yes
!CONFIG_LIBTHEORA=yes
!CONFIG_LIBUTVIDEO=yes
!CONFIG_LIBV4L2=yes
!CONFIG_LIBVO_AACENC=yes
!CONFIG_LIBVO_AMRWBENC=yes
!CONFIG_LIBVORBIS=yes
!CONFIG_LIBVPX=yes
!CONFIG_LIBX264=yes
!CONFIG_LIBXAVS=yes
!CONFIG_LIBXVID=yes
CONFIG_LPC=yes
CONFIG_LSP=yes
CONFIG_MDCT=yes
CONFIG_MEMALIGN_HACK=yes
!CONFIG_MLIB=yes
CONFIG_MPEGAUDIODSP=yes
CONFIG_NETWORK=yes
!CONFIG_NONFREE=yes
!CONFIG_OPENAL=yes
!CONFIG_OPENSSL=yes
!CONFIG_PIC=yes
!CONFIG_POSTPROC=yes
CONFIG_RDFT=yes
CONFIG_RTPDEC=yes
!CONFIG_RUNTIME_CPUDETECT=yes
CONFIG_SAFE_BITSTREAM_READER=yes
!CONFIG_SHARED=yes
CONFIG_SINEWIN=yes
!CONFIG_SMALL=yes
!CONFIG_SRAM=yes
CONFIG_STATIC=yes
CONFIG_SWRESAMPLE=yes
CONFIG_SWSCALE=yes
CONFIG_SWSCALE_ALPHA=yes
!CONFIG_THUMB=yes
!CONFIG_VAAPI=yes
!CONFIG_VDA=yes
!CONFIG_VDPAU=yes
!CONFIG_VERSION3=yes
!CONFIG_X11GRAB=yes
CONFIG_ZLIB=yes
CONFIG_AVUTIL=yes
```

```
!CONFIG_GPLV3=yes
!CONFIG_LGPLV3=yes
CONFIG_AAC_ADTSTOASC_BSF=yes
CONFIG_CHOMP_BSF=yes
CONFIG_DUMP_EXTRADATA_BSF=yes
CONFIG_H264_MP4TOANNEXB_BSF=yes
CONFIG_IMX_DUMP_HEADER_BSF=yes
CONFIG_MJPEG2JPEG_BSF=yes
CONFIG_MJPEGA_DUMP_HEADER_BSF=yes
CONFIG_MP3_HEADER_COMPRESS_BSF=yes
CONFIG_MP3_HEADER_DECOMPRESS_BSF=yes
CONFIG_MOV2TEXTSUB_BSF=yes
CONFIG_NOISE_BSF=yes
CONFIG_REMOVE_EXTRADATA_BSF=yes
CONFIG_TEXT2MOVSUB_BSF=yes
CONFIG_AASC_DECODER=yes
CONFIG_AMV_DECODER=yes
CONFIG_ANM_DECODER=yes
CONFIG_ANSI_DECODER=yes
CONFIG_ASV1_DECODER=yes
CONFIG_ASV2_DECODER=yes
CONFIG_AURA_DECODER=yes
CONFIG_AURA2_DECODER=yes
CONFIG_AVRP_DECODER=yes
CONFIG_AVS_DECODER=yes
CONFIG_BETHSOFTVID_DECODER=yes
CONFIG_BFI_DECODER=yes
CONFIG_BINK_DECODER=yes
CONFIG_BMP_DECODER=yes
CONFIG_BMV_VIDEO_DECODER=yes
CONFIG_C93_DECODER=yes
CONFIG_CAVS_DECODER=yes
CONFIG_CDGRAPHICS_DECODER=yes
CONFIG_CINEPAK_DECODER=yes
CONFIG_CLJR_DECODER=yes
CONFIG_CSCD_DECODER=yes
CONFIG_CYUV_DECODER=yes
CONFIG_DFA_DECODER=yes
CONFIG_DIRAC_DECODER=yes
CONFIG_DNXHD_DECODER=yes
CONFIG_DPX_DECODER=yes
CONFIG_DSICINVIDEO_DECODER=yes
CONFIG_DVVIDEO_DECODER=yes
CONFIG_DXA_DECODER=yes
CONFIG_DXTORY_DECODER=yes
CONFIG_EACMV_DECODER=yes
CONFIG_EAMAD_DECODER=yes
CONFIG_EATGQ_DECODER=yes
CONFIG_EATGV_DECODER=yes
CONFIG_EATQI_DECODER=yes
CONFIG_EIGHTBPS_DECODER=yes
CONFIG_EIGHTSVX_EXP_DECODER=yes
CONFIG_EIGHTSVX_FIB_DECODER=yes
CONFIG_ESCAPE124_DECODER=yes
CONFIG_ESCAPE130_DECODER=yes
CONFIG_FFV1_DECODER=yes
CONFIG_FFVHUFF_DECODER=yes
CONFIG_FLASHSV_DECODER=yes
CONFIG_FLASHSV2_DECODER=yes
CONFIG_FLIC_DECODER=yes
CONFIG_FLV_DECODER=yes
CONFIG_FOURXM_DECODER=yes
CONFIG_FRAPS_DECODER=yes
CONFIG_FRWU_DECODER=yes
CONFIG_GIF_DECODER=yes
CONFIG_H261_DECODER=yes
CONFIG_H263_DECODER=yes
CONFIG_H263I_DECODER=yes
CONFIG_H264_DECODER=yes
#此处省略若干条…
CONFIG_RTMP_PROTOCOL=yes
CONFIG_RTMPT_PROTOCOL=yes
CONFIG_RTMPE_PROTOCOL=yes
CONFIG_RTMPTE_PROTOCOL=yes
CONFIG_RTMPS_PROTOCOL=yes
CONFIG_RTP_PROTOCOL=yes
CONFIG_TCP_PROTOCOL=yes
!CONFIG_TLS_PROTOCOL=yes
CONFIG_UDP_PROTOCOL=yes
#Test
ACODEC_TESTS=ac3_fixed adpcm_adx adpcm_ima_qt adpcm_ima_wav adpcm_ms adpcm_swf adpcm_yam alac aref flac g722 g723_1 g726 mp2 pcm_alaw
m_f32be pcm_f32le pcm_f64be pcm_f64le pcm_mulaw pcm_s16be pcm_s16le pcm_s24be pcm_s24daud pcm_s24le pcm_s32be pcm_s32le pcm_s8 pcm_u8
av1 wmav2
VCODEC_TESTS=amv asv1 asv2 cljr dnxhd_1080i dnxhd_720p dnxhd_720p_10bit dnxhd_720p_rd dv dv50 dv_411 error ffv1 flashsv flashsv2 flv
1 h263 h263p huffyuv jpeg2000 jpegls ljpeg mjpeg mpeg mpeg1b mpeg2 mpeg2_422 mpeg2_idct_int mpeg2_ilace mpeg2_ivlc_qprd mpeg2thread m
2thread_ilace mpeg4 mpeg4_adap mpeg4_qpel mpeg4_qprd mpeg4adv mpeg4nr mpeg4thread mpng msmpeg4 msmpeg4v2 msvideo1 prores qtrle qtrleg
 rc rgb roq rv10 rv20 snow snowll svq1 v210 vref wmv1 wmv2 yuv zlib zmbv
LAVF_TESTS=aiff alaw asf au avi bmp caf dpx dv_fmt ffm flv_fmt gif gxf jpg mkv mmf mov mpg mulaw mxf mxf_d10 nut ogg pbmpipe pcx pgm
pipe pixfmt png ppm ppmpipe rm rso sgi sox swf tga tiff ts voc voc_s16 wav wtv yuv4mpeg
LAVFI_TESTS=crop crop_scale crop_scale_vflip crop_vflip null pixdesc pixfmts_copy pixfmts_crop pixfmts_hflip pixfmts_null pixfmts_pad
xfmts scale pixfmts vflip scale200 scale500 vflip vflip crop vflip vflip
```

```
SEEK_TESTS=seek_ac3_rm seek_adpcm_ima_wav seek_adpcm_ms_wav seek_adpcm_qt_aiff seek_adpcm_swf_flv seek_adpcm_yam_wav seek_alac_m4a se
asv1_avi seek_asv2_avi seek_dnxhd_1080i_mov seek_dnxhd_720p_dnxhd seek_dnxhd_720p_rd_dnxhd seek_dv411_dv seek_dv50_dv seek_dv_dv seek
ror_mpeg4_adv_avi seek_ffv1_avi seek_flac_flac seek_flashsv_flv seek_flv_flv seek_g726_wav seek_h261_avi seek_h263_avi seek_h263p_avi
ek_huffyuv_avi seek_image_bmp seek_image_jpg seek_image_pcx seek_image_pgm seek_image_ppm seek_image_sgi seek_image_tga seek_image_ti
seek_jpegls_avi seek_lavf_aif seek_lavf_al seek_lavf_asf seek_lavf_au seek_lavf_avi seek_lavf_dv seek_lavf_ffm seek_lavf_flv seek_lav
if seek_lavf_gxf seek_lavf_mkv seek_lavf_mmf seek_lavf_mov seek_lavf_mpg seek_lavf_mxf seek_lavf_mxf_d10 seek_lavf_nut seek_lavf_ogg
k_lavf_rm seek_lavf_swf seek_lavf_ts seek_lavf_ul seek_lavf_voc seek_lavf_wav seek_lavf_wtv seek_lavf_y4m seek_ljpeg_avi seek_mjpeg_a
seek_mp2_mp2 seek_mpeg1_mpg seek_mpeg1b_mpg seek_mpeg2_422_mpg seek_mpeg2_idct_int_mpg seek_mpeg2i_mpg seek_mpeg2ivlc_qprd_mpg seek_m
2reuse_mpg seek_mpeg2thread_mpg seek_mpeg2threadivlc_mpg seek_mpeg4_adap_avi seek_mpeg4_adv_avi seek_mpeg4_nr_avi seek_mpeg4_qprd_avi
ek_mpeg4_rc_avi seek_mpeg4_thread_avi seek_msmpeg4_avi seek_msmpeg4v2_avi seek_odivx_mp4 seek_pbmpipe_pbm seek_pcm_alaw_wav seek_pcm_
be_au seek_pcm_f32le_wav seek_pcm_f64be_au seek_pcm_f64le_wav seek_pcm_mulaw_wav seek_pcm_s16be_mov seek_pcm_s16le_wav seek_pcm_s24be
v seek_pcm_s24daud_302 seek_pcm_s24le_wav seek_pcm_s32be_mov seek_pcm_s32le_wav seek_pcm_s8_mov seek_pcm_u8_wav seek_pgmpipe_pgm seek
mpipe_ppm seek_rgb_avi seek_roqav_roq seek_rv10_rm seek_rv20_rm seek_snow53_avi seek_snow_avi seek_svq1_mov seek_wmav1_asf seek_wmav2
f seek_wmv1_avi seek_wmv2_avi seek_yuv_avi
endif # FFMPEG_CONFIG_MAK
```

config.mak代码大致可以分为以下几类信息：

（1）各种路径（prefix等）

（2）工具集（arch、cc、ld、yasm等）

（3）参数集（cppflag、cflag、ldflag等）

（4）前缀后缀（.a、.dll、.exe等）

（5）类库版本（libavXXX_version信息）

（6）组件配置。这一部分信息使用{组件名}=yes的方式进行书写。对于不支持的组件，则在该组件所在行的前面标记上“！”号（感叹号似乎在Makefile语法中并没有什么特殊的用意，此处可能仅仅是作为一种标记？）。这一部分可以分为3类信息：

a)

ARCH_信息

b)

HAVE_信息

c)

CONFIG_信息。这一部分内容最多，将近有1000行。

（7）Test信息（测试组件的结果？还没研究）

libavXXXX/Makefile

libavXXXX/Makefile指的是FFmpeg类库（libavformat、libavcodec、libavutil等）所在的文件夹下的Makefile。例如libavformat文件夹下的Makefile代码如下所示。

[python]

```
# FFmpeg Libavformat Makefile
#
# 注释：雷霄骅
# leixiaohua1020@126.com
# http://blog.csdn.net/leixiaohua1020
#
# FFmpeg中libavformat的Makefile。
# 注意该Makefile并没有定义类库的编译规则（这一部分统一在library.mak中完成）。
# 该Makefile中只是赋值了几个重要的字符串：
#       NAME，FFLIBS，HEADERS，OBJS，OBJS-yes

#重要：包含了configure信息，位于上一级目录
include $(SUBDIR)../config.mak
#名称
NAME = avformat
#用到的库？
FFLIBS = avcodec avutil
#SDK中的头文件
HEADERS = avformat.h avio.h version.h

#OBJS存储的是必须的目标文件
OBJS = allformats.o         \
       cutils.o             \
       id3v1.o              \
       id3v2.o              \
       metadata.o           \
       options.o            \
       os_support.o         \
       sdp.o                \
       seek.o               \
       utils.o              \

#OBJS-yes存储的是可选的目标文件
OBJS-$(CONFIG_NETWORK)                   += network.o

# muxers/demuxers
OBJS-$(CONFIG_A64_MUXER)                 += a64.o
OBJS-$(CONFIG_AAC_DEMUXER)               += aacdec.o rawdec.o
OBJS-$(CONFIG_AC3_DEMUXER)               += ac3dec.o rawdec.o
OBJS-$(CONFIG_AC3_MUXER)                 += rawenc.o
OBJS-$(CONFIG_ACT_DEMUXER)               += act.o
OBJS-$(CONFIG_ADF_DEMUXER)               += bintext.o sauce.o
OBJS-$(CONFIG_ADX_DEMUXER)               += adxdec.o
```

```
OBJS-$(CONFIG_ADX_MUXER)                 += rawenc.o
OBJS-$(CONFIG_ADTS_MUXER)                += adtsenc.o
OBJS-$(CONFIG_AEA_DEMUXER)               += aea.o pcm.o
OBJS-$(CONFIG_AIFF_DEMUXER)              += aiffdec.o riff.o pcm.o isom.o
OBJS-$(CONFIG_AIFF_MUXER)                += aiffenc.o riff.o isom.o
OBJS-$(CONFIG_AMR_DEMUXER)               += amr.o
OBJS-$(CONFIG_AMR_MUXER)                 += amr.o
OBJS-$(CONFIG_ANM_DEMUXER)               += anm.o
OBJS-$(CONFIG_APC_DEMUXER)               += apc.o
OBJS-$(CONFIG_APE_DEMUXER)               += ape.o apetag.o
OBJS-$(CONFIG_APPLEHTTP_DEMUXER)         += applehttp.o
OBJS-$(CONFIG_ASF_DEMUXER)               += asfdec.o asf.o asfcrypt.o \
                                            riff.o avlanguage.o
OBJS-$(CONFIG_ASF_MUXER)                 += asfenc.o asf.o riff.o
OBJS-$(CONFIG_ASS_DEMUXER)               += assdec.o
OBJS-$(CONFIG_ASS_MUXER)                 += assenc.o
OBJS-$(CONFIG_AU_DEMUXER)                += au.o pcm.o
OBJS-$(CONFIG_AU_MUXER)                  += au.o
OBJS-$(CONFIG_AVI_DEMUXER)               += avidec.o riff.o
OBJS-$(CONFIG_AVI_MUXER)                 += avienc.o riff.o
OBJS-$(CONFIG_AVISYNTH)                  += avisynth.o
OBJS-$(CONFIG_AVM2_MUXER)                += swfenc.o
OBJS-$(CONFIG_AVS_DEMUXER)               += avs.o vocdec.o voc.o
OBJS-$(CONFIG_BETHSOFTVID_DEMUXER)       += bethsoftvid.o
OBJS-$(CONFIG_BFI_DEMUXER)               += bfi.o
OBJS-$(CONFIG_BINK_DEMUXER)              += bink.o
OBJS-$(CONFIG_BINTEXT_DEMUXER)           += bintext.o sauce.o
OBJS-$(CONFIG_BIT_DEMUXER)               += bit.o
OBJS-$(CONFIG_BIT_MUXER)                 += bit.o
OBJS-$(CONFIG_BMV_DEMUXER)               += bmv.o
OBJS-$(CONFIG_C93_DEMUXER)               += c93.o vocdec.o voc.o
OBJS-$(CONFIG_CAF_DEMUXER)               += cafdec.o caf.o mov.o mov_chan.o \
                                            riff.o isom.o
OBJS-$(CONFIG_CAF_MUXER)                 += cafenc.o caf.o riff.o isom.o
OBJS-$(CONFIG_CAVSVIDEO_DEMUXER)         += cavsvideodec.o rawdec.o
OBJS-$(CONFIG_CAVSVIDEO_MUXER)           += rawenc.o
OBJS-$(CONFIG_CDG_DEMUXER)               += cdg.o
OBJS-$(CONFIG_CRC_MUXER)                 += crcenc.o
OBJS-$(CONFIG_DAUD_DEMUXER)              += daud.o
OBJS-$(CONFIG_DAUD_MUXER)                += daud.o
OBJS-$(CONFIG_DFA_DEMUXER)               += dfa.o
OBJS-$(CONFIG_DIRAC_DEMUXER)             += diracdec.o rawdec.o
OBJS-$(CONFIG_DIRAC_MUXER)               += rawenc.o
OBJS-$(CONFIG_DNXHD_DEMUXER)             += dnxhddec.o rawdec.o
OBJS-$(CONFIG_DNXHD_MUXER)               += rawenc.o
OBJS-$(CONFIG_DSICIN_DEMUXER)            += dsicin.o
OBJS-$(CONFIG_DTS_DEMUXER)               += dtsdec.o rawdec.o
OBJS-$(CONFIG_DTS_MUXER)                 += rawenc.o
OBJS-$(CONFIG_DV_DEMUXER)                += dv.o
OBJS-$(CONFIG_DV_MUXER)                  += dvenc.o
OBJS-$(CONFIG_DXA_DEMUXER)               += dxa.o riff.o
OBJS-$(CONFIG_EA_CDATA_DEMUXER)          += eacdata.o
OBJS-$(CONFIG_EA_DEMUXER)                += electronicarts.o
OBJS-$(CONFIG_EAC3_DEMUXER)              += ac3dec.o rawdec.o
OBJS-$(CONFIG_EAC3_MUXER)                += rawenc.o
OBJS-$(CONFIG_FFM_DEMUXER)               += ffmdec.o
OBJS-$(CONFIG_FFM_MUXER)                 += ffmenc.o
OBJS-$(CONFIG_FFMETADATA_DEMUXER)        += ffmetadec.o
OBJS-$(CONFIG_FFMETADATA_MUXER)          += ffmetaenc.o
OBJS-$(CONFIG_FILMSTRIP_DEMUXER)         += filmstripdec.o
OBJS-$(CONFIG_FILMSTRIP_MUXER)           += filmstripenc.o
OBJS-$(CONFIG_FLAC_DEMUXER)              += flacdec.o rawdec.o \
                                            oggparsevorbis.o \
                                            vorbiscomment.o
OBJS-$(CONFIG_FLAC_MUXER)                += flacenc.o flacenc_header.o \
                                            vorbiscomment.o
OBJS-$(CONFIG_FLIC_DEMUXER)              += flic.o
OBJS-$(CONFIG_FLV_DEMUXER)               += flvdec.o
OBJS-$(CONFIG_FLV_MUXER)                 += flvenc.o avc.o
OBJS-$(CONFIG_FOURXM_DEMUXER)            += 4xm.o
OBJS-$(CONFIG_FRAMECRC_MUXER)            += framecrcenc.o
OBJS-$(CONFIG_FRAMEMD5_MUXER)            += md5enc.o
OBJS-$(CONFIG_GIF_MUXER)                 += gif.o
OBJS-$(CONFIG_GSM_DEMUXER)               += gsmdec.o
OBJS-$(CONFIG_GXF_DEMUXER)               += gxf.o
OBJS-$(CONFIG_GXF_MUXER)                 += gxfenc.o audiointerleave.o
OBJS-$(CONFIG_G722_DEMUXER)              += rawdec.o
OBJS-$(CONFIG_G722_MUXER)                += rawenc.o
OBJS-$(CONFIG_G723_1_DEMUXER)            += g723_1.o
OBJS-$(CONFIG_G723_1_MUXER)              += rawenc.o
OBJS-$(CONFIG_G729_DEMUXER)              += g729dec.o
OBJS-$(CONFIG_H261_DEMUXER)              += h261dec.o rawdec.o
OBJS-$(CONFIG_H261_MUXER)                += rawenc.o
OBJS-$(CONFIG_H263_DEMUXER)              += h263dec.o rawdec.o
OBJS-$(CONFIG_H263_MUXER)                += rawenc.o
OBJS-$(CONFIG_H264_DEMUXER)              += h264dec.o rawdec.o
OBJS-$(CONFIG_H264_MUXER)                += rawenc.o
OBJS-$(CONFIG_ICO_DEMUXER)               += icodec.o
OBJS-$(CONFIG_IDCIN_DEMUXER)             += idcin.o
OBJS-$(CONFIG_IDF_DEMUXER)               += bintext.o
OBJS-$(CONFIG_IFF_DEMUXER)               += iff.o
```

```
OBJS-$(CONFIG_IMAGE2_DEMUXER)            += img2.o
OBJS-$(CONFIG_IMAGE2_MUXER)              += img2.o
OBJS-$(CONFIG_IMAGE2PIPE_DEMUXER)        += img2.o
OBJS-$(CONFIG_IMAGE2PIPE_MUXER)          += img2.o
OBJS-$(CONFIG_INGENIENT_DEMUXER)         += ingenientdec.o rawdec.o
OBJS-$(CONFIG_IPMOVIE_DEMUXER)           += ipmovie.o
OBJS-$(CONFIG_ISS_DEMUXER)               += iss.o
OBJS-$(CONFIG_IV8_DEMUXER)               += iv8.o
OBJS-$(CONFIG_IVF_DEMUXER)               += ivfdec.o riff.o
OBJS-$(CONFIG_IVF_MUXER)                 += ivfenc.o
OBJS-$(CONFIG_JV_DEMUXER)                += jvdec.o
OBJS-$(CONFIG_LATM_DEMUXER)              += rawdec.o
OBJS-$(CONFIG_LATM_MUXER)                += latmenc.o
OBJS-$(CONFIG_LMLM4_DEMUXER)             += lmlm4.o
OBJS-$(CONFIG_LOAS_DEMUXER)              += loasdec.o
OBJS-$(CONFIG_LXF_DEMUXER)               += lxfdec.o
OBJS-$(CONFIG_M4V_DEMUXER)               += m4vdec.o rawdec.o
OBJS-$(CONFIG_M4V_MUXER)                 += rawenc.o
OBJS-$(CONFIG_MATROSKA_DEMUXER)          += matroskadec.o matroska.o \
                                            riff.o isom.o rmdec.o rm.o
OBJS-$(CONFIG_MATROSKA_MUXER)            += matroskaenc.o matroska.o \
                                            riff.o isom.o avc.o \
                                            flacenc_header.o avlanguage.o
OBJS-$(CONFIG_MD5_MUXER)                 += md5enc.o
OBJS-$(CONFIG_MICRODVD_DEMUXER)          += microdvddec.o
OBJS-$(CONFIG_MICRODVD_MUXER)            += microdvdenc.o rawenc.o
OBJS-$(CONFIG_MJPEG_DEMUXER)             += rawdec.o
OBJS-$(CONFIG_MJPEG_MUXER)               += rawenc.o
OBJS-$(CONFIG_MLP_DEMUXER)               += rawdec.o
OBJS-$(CONFIG_MLP_MUXER)                 += rawenc.o
OBJS-$(CONFIG_MM_DEMUXER)                += mm.o
OBJS-$(CONFIG_MMF_DEMUXER)               += mmf.o pcm.o
OBJS-$(CONFIG_MMF_MUXER)                 += mmf.o riff.o
OBJS-$(CONFIG_MOV_DEMUXER)               += mov.o riff.o isom.o mov_chan.o
OBJS-$(CONFIG_MOV_MUXER)                 += movenc.o riff.o isom.o avc.o \
                                            movenchint.o rtpenc_chain.o \
                                            mov_chan.o
OBJS-$(CONFIG_MP2_MUXER)                 += mp3enc.o rawenc.o
OBJS-$(CONFIG_MP3_DEMUXER)               += mp3dec.o
OBJS-$(CONFIG_MP3_MUXER)                 += mp3enc.o rawenc.o id3v2enc.o
OBJS-$(CONFIG_MPC_DEMUXER)               += mpc.o apetag.o
OBJS-$(CONFIG_MPC8_DEMUXER)              += mpc8.o
OBJS-$(CONFIG_MPEG1SYSTEM_MUXER)         += mpegenc.o
OBJS-$(CONFIG_MPEG1VCD_MUXER)            += mpegenc.o
OBJS-$(CONFIG_MPEG2DVD_MUXER)            += mpegenc.o
OBJS-$(CONFIG_MPEG2VOB_MUXER)            += mpegenc.o
OBJS-$(CONFIG_MPEG2SVCD_MUXER)           += mpegenc.o
OBJS-$(CONFIG_MPEG1VIDEO_MUXER)          += rawenc.o
OBJS-$(CONFIG_MPEG2VIDEO_MUXER)          += rawenc.o
OBJS-$(CONFIG_MPEGPS_DEMUXER)            += mpeg.o
OBJS-$(CONFIG_MPEGTS_DEMUXER)            += mpegts.o isom.o
OBJS-$(CONFIG_MPEGTS_MUXER)              += mpegtsenc.o adtsenc.o
OBJS-$(CONFIG_MPEGVIDEO_DEMUXER)         += mpegvideodec.o rawdec.o
OBJS-$(CONFIG_MPJPEG_MUXER)              += mpjpeg.o
OBJS-$(CONFIG_MSNWC_TCP_DEMUXER)         += msnwc_tcp.o
OBJS-$(CONFIG_MTV_DEMUXER)               += mtv.o
OBJS-$(CONFIG_MVI_DEMUXER)               += mvi.o
OBJS-$(CONFIG_MXF_DEMUXER)               += mxfdec.o mxf.o
OBJS-$(CONFIG_MXF_MUXER)                 += mxfenc.o mxf.o audiointerleave.o
OBJS-$(CONFIG_MXG_DEMUXER)               += mxg.o
OBJS-$(CONFIG_NC_DEMUXER)                += ncdec.o
OBJS-$(CONFIG_NSV_DEMUXER)               += nsvdec.o
OBJS-$(CONFIG_NULL_MUXER)                += nullenc.o
OBJS-$(CONFIG_NUT_DEMUXER)               += nutdec.o nut.o riff.o
OBJS-$(CONFIG_NUT_MUXER)                 += nutenc.o nut.o riff.o
OBJS-$(CONFIG_NUV_DEMUXER)               += nuv.o riff.o
OBJS-$(CONFIG_OGG_DEMUXER)               += oggdec.o         \
                                            oggparsecelt.o   \
                                            oggparsedirac.o  \
                                            oggparseflac.o   \
                                            oggparseogm.o    \
                                            oggparseskeleton.o \
                                            oggparsespeex.o  \
                                            oggparsetheora.o \
                                            oggparsevorbis.o \
                                            riff.o \
                                            vorbiscomment.o
OBJS-$(CONFIG_OGG_MUXER)                 += oggenc.o \
                                            vorbiscomment.o
OBJS-$(CONFIG_OMA_DEMUXER)               += omadec.o pcm.o oma.o
OBJS-$(CONFIG_OMA_MUXER)                 += omaenc.o rawenc.o oma.o id3v2enc.o
OBJS-$(CONFIG_PCM_ALAW_DEMUXER)          += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_ALAW_MUXER)            += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_F32BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_F32BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_F32LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_F32LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_F64BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_F64BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_F64LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_F64LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_MULAW_DEMUXER)         += pcmdec.o pcm.o rawdec.o
```

```
OBJS-$(CONFIG_PCM_MULAW_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_MULAW_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S16BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S16BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S16LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S16LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S24BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S24BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S24LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S24LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S32BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S32BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S32LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S32LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_S8_DEMUXER)            += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_S8_MUXER)              += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U16BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U16BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U16LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U16LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U24BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U24BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U24LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U24LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U32BE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U32BE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U32LE_DEMUXER)         += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U32LE_MUXER)           += pcmenc.o rawenc.o
OBJS-$(CONFIG_PCM_U8_DEMUXER)            += pcmdec.o pcm.o rawdec.o
OBJS-$(CONFIG_PCM_U8_MUXER)              += pcmenc.o rawenc.o
OBJS-$(CONFIG_PMP_DEMUXER)               += pmpdec.o
OBJS-$(CONFIG_PVA_DEMUXER)               += pva.o
OBJS-$(CONFIG_QCP_DEMUXER)               += qcp.o
OBJS-$(CONFIG_R3D_DEMUXER)               += r3d.o
OBJS-$(CONFIG_RAWVIDEO_DEMUXER)          += rawvideodec.o rawdec.o
OBJS-$(CONFIG_RAWVIDEO_MUXER)            += rawenc.o
OBJS-$(CONFIG_RL2_DEMUXER)               += rl2.o
OBJS-$(CONFIG_RM_DEMUXER)                += rmdec.o rm.o
OBJS-$(CONFIG_RM_MUXER)                  += rmenc.o rm.o
OBJS-$(CONFIG_ROQ_DEMUXER)               += idroqdec.o
OBJS-$(CONFIG_ROQ_MUXER)                 += idroqenc.o rawenc.o
OBJS-$(CONFIG_RSO_DEMUXER)               += rsodec.o rso.o pcm.o
OBJS-$(CONFIG_RSO_MUXER)                 += rsoenc.o rso.o
OBJS-$(CONFIG_RPL_DEMUXER)               += rpl.o
OBJS-$(CONFIG_RTP_MUXER)                 += rtp.o         \
                                            rtpenc_aac.o     \
                                            rtpenc_latm.o    \
                                            rtpenc_amr.o     \
                                            rtpenc_h263.o    \
                                            rtpenc_mpv.o     \
                                            rtpenc.o      \
                                            rtpenc_h264.o \
                                            rtpenc_vp8.o  \
                                            rtpenc_xiph.o \
                                            avc.o
OBJS-$(CONFIG_RTPDEC)                    += rdt.o         \
                                            rtp.o         \
                                            rtpdec.o      \
                                            rtpdec_amr.o  \
                                            rtpdec_asf.o  \
                                            rtpdec_g726.o \
                                            rtpdec_h263.o \
                                            rtpdec_h264.o \
                                            rtpdec_latm.o \
                                            rtpdec_mpeg4.o \
                                            rtpdec_qcelp.o \
                                            rtpdec_qdm2.o \
                                            rtpdec_qt.o   \
                                            rtpdec_svq3.o \
                                            rtpdec_vp8.o  \
                                            rtpdec_xiph.o
OBJS-$(CONFIG_RTSP_DEMUXER)              += rtsp.o rtspdec.o httpauth.o
OBJS-$(CONFIG_RTSP_MUXER)                += rtsp.o rtspenc.o httpauth.o \
                                            rtpenc_chain.o
OBJS-$(CONFIG_SAP_DEMUXER)               += sapdec.o
OBJS-$(CONFIG_SAP_MUXER)                 += sapenc.o rtpenc_chain.o
OBJS-$(CONFIG_SBG_DEMUXER)               += sbgdec.o
OBJS-$(CONFIG_SDP_DEMUXER)               += rtsp.o
OBJS-$(CONFIG_SEGAFILM_DEMUXER)          += segafilm.o
OBJS-$(CONFIG_SEGMENT_MUXER)             += segment.o
OBJS-$(CONFIG_SHORTEN_DEMUXER)           += rawdec.o
OBJS-$(CONFIG_SIFF_DEMUXER)              += siff.o
OBJS-$(CONFIG_SMACKER_DEMUXER)           += smacker.o
OBJS-$(CONFIG_SMJPEG_DEMUXER)            += smjpegdec.o smjpeg.o
OBJS-$(CONFIG_SMJPEG_MUXER)              += smjpegenc.o smjpeg.o
OBJS-$(CONFIG_SOL_DEMUXER)               += sol.o pcm.o
OBJS-$(CONFIG_SOX_DEMUXER)               += soxdec.o pcm.o
OBJS-$(CONFIG_SOX_MUXER)                 += soxenc.o
OBJS-$(CONFIG_SPDIF_DEMUXER)             += spdif.o spdifdec.o
OBJS-$(CONFIG_SPDIF_MUXER)               += spdif.o spdifenc.o
OBJS-$(CONFIG_SRT_DEMUXER)               += srtdec.o
OBJS-$(CONFIG_SRT_MUXER)                 += rawenc.o
```

```
OBJS-$(CONFIG_STR_DEMUXER)               += psxstr.o
OBJS-$(CONFIG_SWF_DEMUXER)               += swfdec.o
OBJS-$(CONFIG_SWF_MUXER)                 += swfenc.o
OBJS-$(CONFIG_THP_DEMUXER)               += thp.o
OBJS-$(CONFIG_TIERTEXSEQ_DEMUXER)        += tiertexseq.o
OBJS-$(CONFIG_MKVTIMESTAMP_V2_MUXER)     += mkvtimestamp_v2.o
OBJS-$(CONFIG_TMV_DEMUXER)               += tmv.o
OBJS-$(CONFIG_TRUEHD_DEMUXER)            += rawdec.o
OBJS-$(CONFIG_TRUEHD_MUXER)              += rawenc.o
OBJS-$(CONFIG_TTA_DEMUXER)               += tta.o
OBJS-$(CONFIG_TTY_DEMUXER)               += tty.o sauce.o
OBJS-$(CONFIG_TXD_DEMUXER)               += txd.o
OBJS-$(CONFIG_VC1_DEMUXER)               += rawdec.o
OBJS-$(CONFIG_VC1T_DEMUXER)              += vc1test.o
OBJS-$(CONFIG_VC1T_MUXER)                += vc1testenc.o
OBJS-$(CONFIG_VMD_DEMUXER)               += sierravmd.o
OBJS-$(CONFIG_VOC_DEMUXER)               += vocdec.o voc.o
OBJS-$(CONFIG_VOC_MUXER)                 += vocenc.o voc.o
OBJS-$(CONFIG_VQF_DEMUXER)               += vqf.o
OBJS-$(CONFIG_W64_DEMUXER)               += wav.o riff.o pcm.o
OBJS-$(CONFIG_WAV_DEMUXER)               += wav.o riff.o pcm.o
OBJS-$(CONFIG_WAV_MUXER)                 += wav.o riff.o
OBJS-$(CONFIG_WC3_DEMUXER)               += wc3movie.o
OBJS-$(CONFIG_WEBM_MUXER)                += matroskaenc.o matroska.o \
                                            riff.o isom.o avc.o \
                                            flacenc_header.o avlanguage.o
OBJS-$(CONFIG_WSAUD_DEMUXER)             += westwood_aud.o
OBJS-$(CONFIG_WSVQA_DEMUXER)             += westwood_vqa.o
OBJS-$(CONFIG_WTV_DEMUXER)               += wtvdec.o wtv.o asfdec.o asf.o asfcrypt.o \
                                            avlanguage.o mpegts.o isom.o riff.o
OBJS-$(CONFIG_WTV_MUXER)                 += wtvenc.o wtv.o asf.o asfenc.o riff.o
OBJS-$(CONFIG_WV_DEMUXER)                += wv.o apetag.o
OBJS-$(CONFIG_XA_DEMUXER)                += xa.o
OBJS-$(CONFIG_XBIN_DEMUXER)              += bintext.o sauce.o
OBJS-$(CONFIG_XMV_DEMUXER)               += xmv.o riff.o
OBJS-$(CONFIG_XWMA_DEMUXER)              += xwma.o riff.o
OBJS-$(CONFIG_YOP_DEMUXER)               += yop.o
OBJS-$(CONFIG_YUV4MPEGPIPE_MUXER)        += yuv4mpeg.o
OBJS-$(CONFIG_YUV4MPEGPIPE_DEMUXER)      += yuv4mpeg.o

# external libraries
OBJS-$(CONFIG_LIBMODPLUG_DEMUXER)        += libmodplug.o
OBJS-$(CONFIG_LIBNUT_DEMUXER)            += libnut.o riff.o
OBJS-$(CONFIG_LIBNUT_MUXER)              += libnut.o riff.o

# protocols I/O
OBJS+= avio.o aviobuf.o

OBJS-$(CONFIG_APPLEHTTP_PROTOCOL)        += applehttpproto.o
OBJS-$(CONFIG_CACHE_PROTOCOL)            += cache.o
OBJS-$(CONFIG_CONCAT_PROTOCOL)           += concat.o
OBJS-$(CONFIG_CRYPTO_PROTOCOL)           += crypto.o
OBJS-$(CONFIG_FILE_PROTOCOL)             += file.o
OBJS-$(CONFIG_GOPHER_PROTOCOL)           += gopher.o
OBJS-$(CONFIG_HTTP_PROTOCOL)             += http.o httpauth.o
OBJS-$(CONFIG_HTTPPROXY_PROTOCOL)        += http.o httpauth.o
OBJS-$(CONFIG_HTTPS_PROTOCOL)            += http.o httpauth.o
OBJS-$(CONFIG_MMSH_PROTOCOL)             += mmsh.o mms.o asf.o
OBJS-$(CONFIG_MMST_PROTOCOL)             += mmst.o mms.o asf.o
OBJS-$(CONFIG_MD5_PROTOCOL)              += md5proto.o
OBJS-$(CONFIG_PIPE_PROTOCOL)             += file.o

# external or internal rtmp
RTMP-OBJS-$(CONFIG_LIBRTMP)               = librtmp.o
RTMP-OBJS-$(!CONFIG_LIBRTMP)              = rtmpproto.o rtmppkt.o
OBJS-$(CONFIG_RTMP_PROTOCOL)             += $(RTMP-OBJS-yes)

OBJS-$(CONFIG_RTP_PROTOCOL)              += rtpproto.o
OBJS-$(CONFIG_TCP_PROTOCOL)              += tcp.o
OBJS-$(CONFIG_TLS_PROTOCOL)              += tls.o
OBJS-$(CONFIG_UDP_PROTOCOL)              += udp.o

SKIPHEADERS-$(CONFIG_NETWORK)            += network.h rtsp.h
TESTPROGS = seek
TOOLS     = aviocat ismindex pktdumper probetest
```

从代码可以看出，libavformat文件夹下的Makefile的规则十分简单，并不包含文件之间的依赖关系（依赖关系位于library.mak中），仅仅是设置了几个变量的值：

NAME：类库名称。注意不包含类库前面的“lib”以及类库的后缀。在这里是“avformat”。

FFLIBS：该类库依赖的类库名称。在这里用到了“avcodec”和“avutil”。

HEADERS：该类库导出的头文件。在这里是“avformat.h”，“avio.h”，“version.h”。

OBJS：该类库依赖的目标文件（必须的）。在这里是“utils.o”等等。

OBJS-yes：该类库依赖的目标文件（可选的）。在这里是“flvdec.o”、“flvenc.o”等等。

library.mak

library.mak专门用于存储编译类库的规则，是和libavXXX/Makefile配合使用的。它的源代码如下所示。

[python]

```
# FFmpeg library.mak
#
# 注释：雷霄骅
# leixiaohua1020@126.com
# http://blog.csdn.net/leixiaohua1020
#
# 编译类库(libavformat等)专用的Makefile，其中包含了编译类库的规则。

#【NAME位于每个类库的Makefile】，可以取avcodec，avformat等等
SRC_DIR := $(SRC_PATH)/lib$(NAME)

include $(SRC_PATH)/common.mak

#这些信息都位于config.mak中
#例如：
# libavformat_VERSION=53.31.100
# libavformat_VERSION_MAJOR=53

LIBVERSION := $(lib$(NAME)_VERSION)
LIBMAJOR   := $(lib$(NAME)_VERSION_MAJOR)
INCINSTDIR := $(INCDIR)/lib$(NAME)
THIS_LIB   := $(SUBDIR)$($(CONFIG_SHARED:yes=S)LIBNAME)

all-$(CONFIG_STATIC): $(SUBDIR)$(LIBNAME)
all-$(CONFIG_SHARED): $(SUBDIR)$(SLIBNAME)


$(SUBDIR)%-test.o: $(SUBDIR)%-test.c
    $(COMPILE_C)

$(SUBDIR)%-test.o: $(SUBDIR)%.c
    $(COMPILE_C)
#汇编？
$(SUBDIR)x86/%.o: $(SUBDIR)x86/%.asm
    $(YASMDEP) $(YASMFLAGS) -I $(<D)/ -M -o $@ $< > $(@:.o=.d)
    $(YASM) $(YASMFLAGS) -I $(<D)/ -o $@ $<

$(OBJS) $(OBJS:.o=.s) $(SUBDIR)%.ho $(TESTOBJS): CPPFLAGS += -DHAVE_AV_CONFIG_H
$(TESTOBJS): CPPFLAGS += -DTEST

#【OBJS来自于每个类库的Makefile】
#$@  表示规则中的目标文件集
#$^  所有的依赖目标的集合。
#生成静态库？
$(SUBDIR)$(LIBNAME): $(OBJS)
    $(RM) $@
    $(AR) rc $@ $^ $(EXTRAOBJS)
    $(RANLIB) $@
#安转头文件，根目录的Makefile调用
install-headers: install-lib$(NAME)-headers install-lib$(NAME)-pkgconfig
#install-libs-yes被install-libs（位于根目录Makefile）调用
install-libs-$(CONFIG_STATIC): install-lib$(NAME)-static
install-libs-$(CONFIG_SHARED): install-lib$(NAME)-shared

define RULES
$(EXAMPLES) $(TESTPROGS) $(TOOLS): %$(EXESUF): %.o
    $$(LD) $(LDFLAGS) -o $$@ $$^ -l$(FULLNAME) $(FFEXTRALIBS) $$(ELIBS)

$(SUBDIR)$(SLIBNAME): $(SUBDIR)$(SLIBNAME_WITH_MAJOR)
    $(Q)cd ./$(SUBDIR) && $(LN_S) $(SLIBNAME_WITH_MAJOR) $(SLIBNAME)

$(SUBDIR)$(SLIBNAME_WITH_MAJOR): $(OBJS) $(SUBDIR)lib$(NAME).ver
    $(SLIB_CREATE_DEF_CMD)
    $$(LD) $(SHFLAGS) $(LDFLAGS) -o $$@ $$(filter %.o,$$^) $(FFEXTRALIBS) $(EXTRAOBJS)
    $(SLIB_EXTRA_CMD)

#SLIBNAME_WITH_MAJOR包含了Major版本号。例如：libavformat-53.dll
ifdef SUBDIR
$(SUBDIR)$(SLIBNAME_WITH_MAJOR): $(DEP_LIBS)
endif
#清空
clean::
    $(RM) $(addprefix $(SUBDIR),*-example$(EXESUF) *-test$(EXESUF) $(CLEANFILES) $(CLEANSUFFIXES) $(LIBSUFFIXES)) \
        $(foreach dir,$(DIRS),$(CLEANSUFFIXES:%=$(SUBDIR)$(dir)/%)) \
        $(HOSTOBJS) $(HOSTPROGS)

distclean:: clean
    $(RM) $(DISTCLEANSUFFIXES:%=$(SUBDIR)%) \
        $(foreach dir,$(DIRS),$(DISTCLEANSUFFIXES:%=$(SUBDIR)$(dir)/%))
#安装库文件=====================
install-lib$(NAME)-shared: $(SUBDIR)$(SLIBNAME)
    $(Q)mkdir -p "$(SHLIBDIR)"
    $$(INSTALL) -m 755 $$< "$(SHLIBDIR)/$(SLIB_INSTALL_NAME)"
    $$(STRIP) "$(SHLIBDIR)/$(SLIB_INSTALL_NAME)"
    $(Q)$(foreach F,$(SLIB_INSTALL_LINKS),cd "$(SHLIBDIR)" && $(LN_S) $(SLIB_INSTALL_NAME) $(F);)
    $(if $(SLIB_INSTALL_EXTRA_SHLIB),$$(INSTALL) -m 644 $(SLIB_INSTALL_EXTRA_SHLIB:%=$(SUBDIR)%) "$(SHLIBDIR)")
    $(if $(SLIB_INSTALL_EXTRA_LIB),$(Q)mkdir -p "$(LIBDIR)")
    $(if $(SLIB_INSTALL_EXTRA_LIB),$$(INSTALL) -m 644 $(SLIB_INSTALL_EXTRA_LIB:%=$(SUBDIR)%) "$(LIBDIR)")

install-lib$(NAME)-static: $(SUBDIR)$(LIBNAME)
    $(Q)mkdir -p "$(LIBDIR)"
    $$(INSTALL) -m 644 $$< "$(LIBDIR)"
    $(LIB_INSTALL_EXTRA_CMD)
#安装头文件=====================
#-m
#权限：644,755,777
#644 rw-r--r--
#755 rwxr-xr-x
#777 rwxrwxrwx
#从左至右，1-3位数字代表文件所有者的权限，4-6位数字代表同组用户的权限，7-9数字代表其他用户的权限。
#通过4、2、1的组合，得到以下几种权限：0（没有权限）；4（读取权限）；5（4+1 | 读取+执行）；6（4+2 | 读取+写入）；7（4+2+1 | 读取+写入+执行）
#addprefix()
#$(addprefix src/,foo bar)返回值是“src/foo src/bar”。

#【HEADERS来自于每个类库的Makefile】
#例如libavformat中HEADERS = avformat.h avio.h version.h
install-lib$(NAME)-headers: $(addprefix $(SUBDIR),$(HEADERS) $(BUILT_HEADERS))
    $(Q)mkdir -p "$(INCINSTDIR)"
    $$(INSTALL) -m 644 $$^ "$(INCINSTDIR)"

install-lib$(NAME)-pkgconfig: $(SUBDIR)lib$(NAME).pc
    $(Q)mkdir -p "$(LIBDIR)/pkgconfig"
    $$(INSTALL) -m 644 $$^ "$(LIBDIR)/pkgconfig"

#卸载
uninstall-libs::
    -$(RM) "$(SHLIBDIR)/$(SLIBNAME_WITH_MAJOR)" \
           "$(SHLIBDIR)/$(SLIBNAME)"            \
           "$(SHLIBDIR)/$(SLIBNAME_WITH_VERSION)"
    -$(RM) $(SLIB_INSTALL_EXTRA_SHLIB:%="$(SHLIBDIR)"%)
    -$(RM) $(SLIB_INSTALL_EXTRA_LIB:%="$(LIBDIR)"%)
    -$(RM) "$(LIBDIR)/$(LIBNAME)"

uninstall-headers::
    $(RM) $(addprefix "$(INCINSTDIR)/",$(HEADERS)) $(addprefix "$(INCINSTDIR)/",$(BUILT_HEADERS))
    $(RM) "$(LIBDIR)/pkgconfig/lib$(NAME).pc"
    -rmdir "$(INCINSTDIR)"
endef

$(eval $(RULES))

$(EXAMPLES) $(TESTPROGS) $(TOOLS): $(THIS_LIB) $(DEP_LIBS)
$(TESTPROGS): $(SUBDIR)$(LIBNAME)

examples: $(EXAMPLES)
testprogs: $(TESTPROGS)
```

library.mak代码中首先包含了common.mak文件。这个文件定义了通用的一些编译规则。然后定义了类库的依赖关系。
此外library.mak中也定义了install-headers，install-lib$(NAME)-shared，install-lib$(NAME)-static，install-lib$(NAME)-headers，clean等等一系列的伪目标（NAME取值avformat、avcodec等）。这些目标主要配合根目录的Makefile使用。

common.mak

common.mak文件定义了通用的一些编译规则。代码如下所示。

[python]

```
# FFmpeg common.mak
#
# 注释：雷霄骅
# leixiaohua1020@126.com
# http://blog.csdn.net/leixiaohua1020
#
# 通用的Makefile，其中包含了通用的编译规则。
#
# common bits used by all libraries
#

# first so "all" becomes default target
all: all-yes

ifndef SUBDIR
#在控制台打印信息
ifndef V
Q      = @
#输出
ECHO   = printf "$(1)\t%s\n" $(2)
BRIEF  = CC CXX AS YASM AR LD HOSTCC STRIP CP
SILENT = DEPCC YASMDEP RM RANLIB
MSG    = $@
M      = @$(call ECHO,$(TAG),$@);
$(foreach VAR,$(BRIEF), \
    $(eval override $(VAR) = @$$(call ECHO,$(VAR),$$(MSG)); $($(VAR))))
$(foreach VAR,$(SILENT),$(eval override $(VAR) = @$($(VAR))))
$(eval INSTALL = @$(call ECHO,INSTALL,$$(^:$(SRC_DIR)/%=%)); $(INSTALL))
```

```makefile
endif
#所有的lib
ALLFFLIBS = avcodec avdevice avfilter avformat avutil postproc swscale swresample

# NASM requires -I path terminated with /
#各种Flag
#SRC_PATH=.
IFLAGS     := -I. -I$(SRC_PATH)/
CPPFLAGS   := $(IFLAGS) $(CPPFLAGS)
CFLAGS     += $(ECFLAGS)
CCFLAGS     = $(CFLAGS)
CXXFLAGS   := $(CFLAGS) $(CXXFLAGS)
YASMFLAGS  += $(IFLAGS) -I$(SRC_PATH)/libavutil/x86/ -Pconfig.asm
HOSTCFLAGS += $(IFLAGS)
#avcodec处理后成为-Llibavcodec
#config.mak文件中：
#LDFLAGS= -Wl,--as-needed -Wl,--warn-common -Wl,
#-rpath-link=libpostproc:libswresample:libswscale:libavfilter:libavdevice:libavformat:libavcodec:libavutil
LDFLAGS    := $(ALLFFLIBS:%=-Llib%) $(LDFLAGS)

#命令包
#具体编译命令
#
#$(1)可以取CC、CXX等
#例如取$(1)取CC
#config.mak文件中：
#SRC_PATH=.
#CC=gcc
#
#CCFLAGS=$(CFLAGS)
#CFLAGS=   -std=c99 -fno-common -fomit-frame-pointer -I/include/SDL -D_GNU_SOURCE=1 -Dmain=SDL_main
# -g -Wdeclaration-after-statement -Wall -Wno-parentheses -Wno-switch -Wno-format-zero-length
# -Wdisabled-optimization -Wpointer-arith -Wredundant-decls -Wno-pointer-sign -Wcast-qual -Wwrite-strings
# -Wtype-limits -Wundef -Wmissing-prototypes -Wno-pointer-to-int-cast -Wstrict-prototypes
# -O3 -fno-math-errno -fno-signed-zeros -fno-tree-vectorize -Werror=implicit-function-declaration -Werror=missing-prototypes
#
#CPPFLAGS= -D_ISOC99_SOURCE -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -U__STRICT_ANSI__
#CC_O=-o $@
#CC_DEPFLAGS=-MMD -MF $(@:.o=.d) -MT $@
#举例：
#gcc -I. -Itest/ -c -o $@ $<
#再例如$(1)取CXX
#CXXFLAGS=  -D__STDC_CONSTANT_MACROS

define COMPILE
       $($(1)DEP)
       $($(1)) $(CPPFLAGS) $($(1)FLAGS) $($(1)_DEPFLAGS) -c $($(1)_O) $<
endef

#编译命令
#$(call <expression>,<parm1>,<parm2>,<parm3>...)
#当make执行这个函数时，<expression>参数中的变量，如$(1)，$(2)，$(3)等，会被参数
#<parm1>，<parm2>，<parm3>依次取代。而<expression>的返回值就是call函数的返回值。
COMPILE_C = $(call COMPILE,CC)
COMPILE_CXX = $(call COMPILE,CXX)
COMPILE_S = $(call COMPILE,AS)

#COMPILE_C为：
#$(CC DEP)
#$($(CC) $(CPPFLAGS) $($(1)FLAGS) $($(1)_DEPFLAGS) -c $($(1)_O) $<

#依赖关系
#C语言
%.o: %.c
#编译
    $(COMPILE_C)

#C++
%.o: %.cpp
    $(COMPILE_CXX)

%.s: %.c
    $(CC) $(CPPFLAGS) $(CFLAGS) -S -o $@ $<

%.o: %.S
    $(COMPILE_S)

%.ho: %.h
    $(CC) $(CPPFLAGS) $(CFLAGS) -Wno-unused -c -o $@ -x c $<

%.ver: %.v
    $(Q)sed 's/$$MAJOR/$($(basename $(@F))_VERSION_MAJOR)/' $^ > $@

%.c %.h: TAG = GEN

# Dummy rule to stop make trying to rebuild removed or renamed headers
%.h:
    @:

# Disable suffix rules.  Most of the builtin rules are suffix rules,
# so this saves some time on slow systems.
```

```
.SUFFIXES:

# Do not delete intermediate files from chains of implicit rules
$(OBJS):
endif

OBJS-$(HAVE_MMX) +=  $(MMX-OBJS-yes)

#源自Makefile
#OBJS：该类库必须的目标文件
#OBJS-yes：该类库可配置的目标文件
OBJS      += $(OBJS-yes)
#FFLIBS：必须的类库
#FFLIBS-yes：可选的类库
#FFLIBS = avcodec avutil ....
FFLIBS    := $(FFLIBS-yes) $(FFLIBS)
TESTPROGS += $(TESTPROGS-yes)

FFEXTRALIBS := $(FFLIBS:%=-l%$(BUILDSUF)) $(EXTRALIBS)

EXAMPLES  := $(EXAMPLES:%=$(SUBDIR)%-example$(EXESUF))
#排序？
OBJS      := $(sort $(OBJS:%=$(SUBDIR)%))
TESTOBJS  := $(TESTOBJS:%=$(SUBDIR)%) $(TESTPROGS:%=$(SUBDIR)%-test.o)
TESTPROGS := $(TESTPROGS:%=$(SUBDIR)%-test$(EXESUF))
HOSTOBJS  := $(HOSTPROGS:%=$(SUBDIR)%.o)
HOSTPROGS := $(HOSTPROGS:%=$(SUBDIR)%$(HOSTEXESUF))
TOOLS     += $(TOOLS-yes)
TOOLOBJS  := $(TOOLS:%=tools/%.o)
TOOLS     := $(TOOLS:%=tools/%$(EXESUF))

#DEP_LIBS= libavcodec/libavcodec.a libavutil/libavutil.a ....
DEP_LIBS := $(foreach NAME,$(FFLIBS),lib$(NAME)/$($(CONFIG_SHARED:yes=S)LIBNAME))

ALLHEADERS := $(subst $(SRC_DIR)/,$(SUBDIR),$(wildcard $(SRC_DIR)/*.h $(SRC_DIR)/$(ARCH)/*.h))
SKIPHEADERS += $(ARCH_HEADERS:%=$(ARCH)/%) $(SKIPHEADERS-)
SKIPHEADERS := $(SKIPHEADERS:%=$(SUBDIR)%)
checkheaders: $(filter-out $(SKIPHEADERS:.h=.ho),$(ALLHEADERS:.h=.ho))

alltools: $(TOOLS)

$(HOSTOBJS): %.o: %.c
    $(HOSTCC) $(HOSTCFLAGS) -c -o $@ $<

$(HOSTPROGS): %$(HOSTEXESUF): %.o
    $(HOSTCC) $(HOSTLDFLAGS) -o $@ $< $(HOSTLIBS)

$(OBJS):     | $(sort $(dir $(OBJS)))
$(HOSTOBJS): | $(sort $(dir $(HOSTOBJS)))
$(TESTOBJS): | $(sort $(dir $(TESTOBJS)))
$(TOOLOBJS): | tools

OBJDIRS := $(OBJDIRS) $(dir $(OBJS) $(HOSTOBJS) $(TESTOBJS))

CLEANSUFFIXES     = *.d *.o *~ *.ho *.map *.ver *.gcno *.gcda
DISTCLEANSUFFIXES = *.pc
LIBSUFFIXES       = *.a *.lib *.so *.so.* *.dylib *.dll *.def *.dll.a *.exp

#依赖文件.d（dependence）
-include $(wildcard $(OBJS:.o=.d) $(TESTOBJS:.o=.d))
```

从代码中可以看出，common.mak定义了一些通用的编译规则，例如编译时候的控制台输出格式，编译命令COMPILE_C、COMPILE_CXX、COMPILE_S，以及.c、.o等文件之间的依赖关系等。

雷霄骅
leixiaohua1020@126.com
http://blog.csdn.net/leixiaohua1020

文章标签： ffmpeg makefile make configure 编译
个人分类： FFMPEG
所属专栏： FFmpeg