

转 windows 下使用 MinGW + msys 编译 ffmpeg

2013年10月16日 19:24:35 阅读数：10274

本文参考了网络上的不少文章，但由于版本环境的问题参考文章并不能直接指导编译，本文吸收多方经验，并在自己多次编译实验的基础上写成，欢迎转载，请注明出处。

FFmpeg是在Linux平台下开发的，但它同样可以在其它操作系统环境中编译运行，本文介绍的就是在windows下编译FFmpeg。

一 编译准备

您最好新建一个目录专门用来保存以下下载的文件。

1 下载 MinGW 和 MSYS

MinGW(Minimalist GNU on Windows)：一个可自由使用和自由发布的Windows特定头文件和使用GNU工具集导入库的集合，允许你生成本地的Windows程序而不需要第三方C运行时库。

到：<http://sourceforge.net/projects/mingw/> 去下载即可

A 点击“DownLoad mingw-get-inst-...exe”

B 切换到页面：download? Check your browser's security bar, or try a direct link, or try another mirror.

C 点击“direct link”下载类似：mingw-get-inst-20111118.exe 的文件

提示：msys此处就不用再下载了，mingw-get-inst-20111118.exe 中已经包含了msys1.0，后面安装的时候就可以看到该选项。

2 下载 yasm(汇编编译器)

ffmpeg编译的时候里面的汇编代码需要yasm.exe

去官网：<http://yasm.tortall.net/Download.html> 下载

在Latest Release栏，直接下载：Win32 .exe (for general use on 32-bit Windows)

文件名类似为：yasm-1.2.0-win32.exe

3 下载 ffmpeg

ffmpeg可以到 <http://www.ffmpeg.org/releases/> 下载

选择的一个最新版本下载即可，如：ffmpeg-0.8.6.tar.gz

文件名类似为：ffmpeg-0.8.6.tar.gz

当然你也使用网上介绍的使用SVN直接下载源代码的方法。

<svn://svn.mplayerhq.hu/ffmpeg/trunk>

4 下载 mp3lame(支持mp3，可选项)

为了让编译出来的ffmpeg支持对mp3格式的编解码，您需要先下载lame。

到：<http://sourceforge.net/projects/lame/files/> 下载

文件名类似为：lame-3.99.2.tar.gz 的包。

5 下载 x264(支持x264，可选项)

为了让编译出来的ffmpeg支持对x264格式的编解码，您需要先下载x264。

到：<http://www.videolan.org/developers/x264.html> 下载

文件名类似为：last_x264.tar.bz2 的包。

6 下载 Xvid(支持Xvid，可选项)

为了让编译出来的ffmpeg支持对Xvid格式的编解码，您需要先下载Xvid。

到：<http://www.xvid.org/> 下载

文件名类似为：xvidcore-1.3.2.tar.gz 的包。

7 下载 faad2(支持faad，可选项)

到：<http://www.audiocoding.com/downloads.html> 下载：Version 2.7 bootstrapped TAR.GZ Package

文件名类似为：faad2-2.7.tar.gz 的包。

8 下载 faac(支持faac，可选项)

到：<http://www.audiocoding.com/downloads.html> 下载：Version 1.28 bootstrapped TAR.GZ Package

文件名类似为：faac-1.28.tar.gz 的包。

9 下载 aacenc(支持aac，可选项)

到：<http://sourceforge.net/projects/opencore-amr/files/> 下载：vo-aacenc

文件名类似为：vo-aacenc-0.1.1.tar.gz 的包。

10 下载 amr-nb与amr-wb (早期ffmpeg支持3gp，可选项)

到：<http://www.penguin.cz/%7Eutx/amr> 下载：amrnb-7.0.0.2.tar.bz2与amrwb-7.0.0.4.tar.bz2

文件名类似为：amrnb-7.0.0.2.tar.tar 与amrwb-7.0.0.4.tar.tar 的包。

11 下载 opencore-amr与vo-amrwbenc (支持3gp，可选项)

到：<http://sourceforge.net/projects/opencore-amr/files/> 下载：opencore-amr 与 vo-amrwbenc

文件名类似为：opencore-amr-0.1.2.tar.gz 与vo-amrwbenc-0.1.1.tar.gz 的包。

12 下载 SDL(支持SDL，可选项)

SDL下载可以到：<http://www.libsdl.org/download-1.2.php>

在Source Code栏选择类似：SDL-1.2.14.tar.gz - GPG signed 进行下载，

文件名类似为：SDL-1.2.14.tar.tar，此文件需要编译才能生成lib库。

也可以直接下载 SDL-devel-1.2.14-mingw32.tar.gz (Mingw32)，文件名类似为：SDL-devel-1.2.14-mingw32.tar.tar，此文件带有编译好的lib，在编译ffmpeg时可以直接使用。

13 下载 pthreadGC2.dll

pthreadgc2.dll是功能强大的处理数字化影视作品编辑软件premiere6运行所需的一个DLL文件。这是一个无威胁文件。属于Open Source Software community project的文件。

直接去baidu.com三搜索下载就可以了。

二 编译步骤

1 安装 mingw 和 msys

执行下载文件：mingw-get-inst-20111118.exe，

在第3步选择“Download latest repository catalogues”

选择安装路径：C:\MinGW（我直接采用默认路径，未做修改）

在“Select Components”步骤中选择：

√ C Compiler (默认且一定选择)

√ C++ Compiler

√ MSYS Basic System (选择此项将会安装msys, msys则不需要另外安装)

√ MinGW Developer ToolKit

等待安装完成! (需要等待一定的时间, 因为安装过程需要下载文件, 可想而知, 网速越快, 安装越快)

安装完成后, 在C:\MinGW目录下可以看到 msys 文件夹, msys也已经安装。

2 配置 msys.bat 批处理文件

为了方便VS200X以及VS2010调用ffmpeg的动态库, 可以通过配置让ffmpeg编译时产生windows下调用dll对应的lib, 当然, 如果你没有这个需要, 那么就可以省略以下处理, 直接进入步骤3。

进入: C:\MinGW\msys\1.0\文件夹, 使用UltraEdit打开msys.bat文件, 在文件的最前面加入如下一行:

```
call "D:\Program Files\Microsoft Visual Studio 10.0\VC\bin\vcvars32.bat"
```

添加后效果如下:

```
call "D:\Program Files\Microsoft Visual Studio 10.0\VC\bin\vcvars32.bat"
```

```
@echo off
```

```
rem Copyright (C): 2001, 2002, 2003, 2004, 2005 Earnie Boyd
```

```
rem mailto:earnie@users.sf.net
```

```
rem This file is part of Minimal SYStem
```

```
rem http://www.mingw.org/msys.shtml
```

```
rem
```

```
rem File: msys.bat
```

```
rem Revision: 2.4
```

```
rem Revision Date: December 8th, 2005
```

```
rem ember to set the "Start in:" field of the shortcut.
```

```
rem A value similar to C:\msys\1.0\bin is what the "Start in:" field needs
```

```
rem to represent.
```

其中"D:\Program Files\Microsoft Visual Studio 10.0"为你机器上安装VS200X或VS2010的目录, 我的电脑安装的是VS2010, 并且安装在了D盘。文件修改后保存即可。

3 系统整合

使用UltraEdit修改C:\MinGW\msys\1.0\etc目录下的fstab文件, 如果不存在fstab, 请复制一份fstab.sample, 然后改名为: fstab, 再修改其内容为: C:/MinGW /mingw, 也即为MinGW的安装路径。

4 配置 yasm

为了减少配置环境变量的麻烦, 本处直接把 yasm-1.1.0-win32.exe 文件名修改成: yasm.exe, 并放到系统目录下: C:\WINDOWS\system32。

5 编译 mp3lame(可选项)

lame-3.99.2.tar.gz 解压到: D:\ffmpeg\ 目录下, 双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 切换到lame目录: D:\ffmpeg\lame-3.99.2

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd lame-3.99.2
```

再执行以下命令:

```
./configure --disable-shared (首选)
```

```
或: ./configure --disable-static --enable-shared
```

```
$ make
```

```
$ make install
```

编译结果在: C:\MinGW\msys\1.0\local 目录下

6 编译 x264(可选项)

last_x264.tar.bz2 解压到: D:\ffmpeg\ 目录下, 双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 切换到x264目录: D:\ffmpeg\x264-20111124-2245

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd x264-20111124-2245
```

再执行以下命令:

```
./configure --enable-shared --disable-asm
```

```
$ make
```

```
$ make install
```

编译结果在: C:\MinGW\msys\1.0\local 目录下

7 编译 Xvid(可选项)

xvidcore-1.3.2.tar.gz 解压到: D:\ffmpeg\ 目录下, 双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 切换到xvidcore目录: D:\ffmpeg\xvidcore\build\generic

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd xvidcore/build/generic
```

再执行以下命令:

```
./configure --prefix=C:/MinGW/msys/1.0/local
```

```
$ make
```

```
$ make install
```

编译结果在: C:\MinGW\msys\1.0\local 目录下。

需要注意的就是: 此处编译加上了路径

如果make时出现: cc1.exe: error: unrecognized command line option '-mno-cygwin' 错误, 请在./configure后make之前修改xvidcore\build\generic里的 platform.inc文件, 直接删除里面的"-mno-cygwin"。

具体可以参考: <http://ffmpeg.zeranoe.com/forum/viewtopic.php?f=5&t=111>里2楼的回答, 如果按他说的: I recommend re running ./bootstrap.sh after you do this., 我发现不太行, 一运行bootstrap.sh后, platform.inc文件会被还原, 并且./configure时也会还原该文件, 所以我选择在./configure后make前修改, 可以编译通过, 目前没检验Xvid是否可用。

8 编译 faad2(可选项)

faad2-2.7.tar.gz 解压到: D:\ffmpeg\ 目录下, 双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 已经启动的话就不用再重新启动了。切换到faad2目录: D:\ffmpeg\

```
faad2-2.7
```

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd faad2-2.7
```

再执行以下命令:

```
$.bootstrap
$./configure --disable-static --enable-shared
$ make
$ make install
编译结果在：C:\MinGW\msys\1.0\local 目录下
9 编译faac(可选项)
faac-1.28.tar.gz 解压到：D:\ffmpeg\ 目录下,双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 已经启动的话就不用再重新启动了。切换到faac2目录：D:\ffmpeg\
faac-1.28
$ cd /d
$ cd ffmpeg
$ cd faac-1.28
再执行以下命令：
$.bootstrap
$./configure --disable-static --enable-shared
$ make
$ make install
编译结果在：C:\MinGW\msys\1.0\local 目录下
10 编译vo-aacenc (可选项)
vo-aacenc-0.1.1.tar.gz 解压到：D:\ffmpeg\ 目录下,双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 已经启动的话就不用再重新启动了。切换到vo-aacenc目录
：D:\ffmpeg\vo-aacenc-0.1.1
$ cd /d
$ cd ffmpeg
$ cd vo-aacenc-0.1.1
再执行以下命令：
$./configure --disable-shared
$ make
$ make install
编译结果在：C:\MinGW\msys\1.0\local 目录下
11 编译amr-nb和amr-wb (早期ffmpeg版本, 可选项)
在编译amrnb和amrwb之前还需要做点其它的准备工作：
A 先到网上搜索下载wget工具, 解压后放到C:\MinGW\bin目录下, 此工具是个网络下载工具, 用来在编译时自动下载3gpp源码。
B 再到http://downloads.sourceforge.net/gnuwin32/unzip-5.51-1-bin.zip 下载unzip-5.51-1-bin.zip后进行解压, 并把unzip-5.51-1-bin\bin中的unzip.exe拷贝到C:\MinG
W\bin目录下, 此工具用来在编译时自动解压3gpp源码。
amrnb-7.0.0.2.tar.tar 解压到：D:\ffmpeg\ 目录下,双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 已经启动的话就不用再重新启动了。切换到amrnb目录：D:\ff
mpeg\amrnb-7.0.0.2
$ cd /d
$ cd ffmpeg
$ cd amrnb-7.0.0.2
再执行以下命令：
$./configure --disable-static --enable-shared
$ make
$ make install
编译结果在：C:\MinGW\msys\1.0\local 目录下
amrwb-7.0.0.4.tar.tar 解压到：D:\ffmpeg\ 目录下, 双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 已经启动的话就不用再重新启动了。切换到amrwb目录：D:
\ffmpeg\amrwb-7.0.0.4
$ cd /d
$ cd ffmpeg
$ cd amrwb-7.0.0.4
再执行以下命令：
$./configure --disable-static --enable-shared
$ make
$ make install
编译结果在：C:\MinGW\msys\1.0\local 目录下
12 编译opencore-amr 和vo-amrwbenc (可选项)
opencore-amr-0.1.2.tar.gz 解压到：D:\ffmpeg\ 目录下,双击"C:\MinGW\msys\1.0\msys.bat", 启动"MinGW32", 已经启动的话就不用再重新启动了。
如果是opencore-amr-0.1.1.tar.gz请注释掉以下Makefile中的几行
文件: ./opencore-amr/amrnb/Makefile
install: libopencore-amrnb.a $(SHLIB)
install -d $(DESTDIR)$(PREFIX)/lib
install -m 644 libopencore-amrnb.a $(DESTDIR)$(PREFIX)/lib
# install $(SHLIB) $(DESTDIR)$(PREFIX)/lib
ifneq ($(shell uname), Darwin)
# ln -sf $(SHLIB) $(DESTDIR)$(PREFIX)/lib/$(SONAME)
# ln -sf $(SONAME) $(DESTDIR)$(PREFIX)/lib/libopencore-amrnb.so
endif
文件: ./opencore-amr/amrwb/Makefile
install: libopencore-amrwb.a $(SHLIB)
install -d $(DESTDIR)$(PREFIX)/lib
install -m 644 libopencore-amrwb.a $(DESTDIR)$(PREFIX)/lib
# install $(SHLIB) $(DESTDIR)$(PREFIX)/lib
ifneq ($(shell uname), Darwin)
# ln -sf $(SHLIB) $(DESTDIR)$(PREFIX)/lib/$(SONAME)
# ln -sf $(SONAME) $(DESTDIR)$(PREFIX)/lib/libopencore-amrwb.so
endif
```

切换到opencore-amr目录：D:\ffmpeg\ opencore-amr-0.1.2

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd opencore-amr-0.1.2
```

再执行以下命令：

```
$/configure --disable-shared
```

```
$ make CC=gcc
```

```
$ make install
```

编译结果在：C:\MinGW\msys\1.0\local 目录下

vo-amrwbenc-0.1.1.tar.gz 解压到：D:\ffmpeg\ 目录下,双击"C:\MinGW\msys\1.0\msys.bat"，启动"MinGW32"，已经启动的话就不用再重新启动了。切换到vo-amrwbenc

c目录：D:\ffmpeg\ vo-amrwbenc-0.1.1

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd vo-amrwbenc-0.1.1
```

再执行以下命令：

```
$/configure --disable-shared
```

```
$ make CC=gcc
```

```
$ make install
```

编译结果在：C:\MinGW\msys\1.0\local 目录下

13 编译SDL(可选项)

SDL-1.2.14.tar.tar 解压到：D:\ffmpeg\ 目录下, 双击"C:\MinGW\msys\1.0\msys.bat"，启动"MinGW32"，切换到SDL目录：D:\ffmpeg\SDL-1.2.14，切换方法如下：

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd SDL-1.2.14
```

再执行以下命令：

```
./configure
```

```
make
```

```
make install
```

编译结果在：C:\MinGW\msys\1.0\local 目录下

注意：如果是使用编译好的SDL-devel-1.2.14-mingw32.tar.tar，那么就可以省掉上面的编译工作，直接解压SDL-devel-1.2.14-mingw32.tar.tar文件即可。自己编译或使用使用编译好的都需要做下面的修改和拷贝工作。

使用UltraEdit打开C:\MinGW\msys\1.0\local\bin下的 sdl-config文件

把 prefix=/usr 该成：prefix=c:/mingw

其中：c:/mingw 为 mingw的安装路径，请根据你的安装进行修改。为了编译时msys能识别sdl并开启 SDL support yes 进行编译，请把C:\MinGW\msys\1.0\local\编译结果bin、include和lib中有关sdl的拷贝一份到C:\MinGW\的对应目录中。

14 拷贝编译结果

C:\MinGW\msys\1.0\local 到 D:\ffmpeg\ffmpeg-0.8.6

bin拷贝到 bin中

include拷贝到 include中

lib拷贝到 lib中

当然该拷贝过程你可以不做，只要在编译的时候把路径指到：C:\MinGW\msys\1.0\local里面。

15 编译 ffmpeg

ffmpeg-0.8.6.tar.bz2 解压到：D:\ffmpeg\ 目录下,双击"C:\MinGW\msys\1.0\msys.bat"，启动"MinGW32"，切换到ffmpeg目录：D:\ffmpeg\ffmpeg-0.8.6，切换方法如下：

```
$ cd /d
```

```
$ cd ffmpeg
```

```
$ cd ffmpeg-0.8.6
```

如果你的"MinGW32"没有关闭，可以直接切换路径到：D:\ffmpeg\ffmpeg-0.8.6。

再执行以下命令：

```
$ ./configure --enable-shared --disable-static --enable-ffplay
```

```
--enable-nonfree --enable-memalign-hack
```

```
--enable-libmp3lame
```

```
--enable-gpl--enable-libx264
```

```
--enable-libxvid
```

```
--enable-faad --enable-libfaac
```

```
--enable-libvo-aacenc
```

```
--enable-version3 --enable-libopencore-amrnb
```

```
--enable-libopencore-amrwb --enable-libvo-amrwbenc
```

```
--extra-cflags=-Id:/ffmpeg/ffmpeg-0.8.6/include
```

```
--extra-ldflags=-Ld:/ffmpeg/ffmpeg-0.8.6/lib
```

请耐心等待一段时间，以上命令不能直接换行哦并且根据ffmpeg的版本不同有所变化,具体看./configure --help，你知道的。。。

```
$ make
```

需要等待很长一段时间，等待编译结束。

为了汇总编译结果，请继续执行以下命令：

```
$ make install
```

等待完成，在C:\MinGW\msys\1.0\local中将会生成 bin、include、lib等文件夹，其中包含了ffmpeg.exe、ffplay.exe、ffprobe.exe、dll、lib和头文件等。

16 运行测试

直接双击运行"ffplay.exe"，如果提示没有"pthreadGC2.dll"文件，你可以包"pthreadGC2.dll"直接拷贝到当前目录，也可放到系统目录C:\WINDOWS\system32中。

ffplay f:/xxx.mp4

参考：

<http://blog.csdn.net/yangshu168/archive/2010/01/17/5204269.aspx>

<http://blog.csdn.net/himulakensin/article/details/6693457>

<http://blog.csdn.net/huangxiansheng1980/article/details/5712003>

原文地址：<http://blog.chinaunix.net/uid-20718335-id-2980793.html>

另一篇资料：<http://download.csdn.net/detail/leixiaohua1020/6389413>

文章标签：[windows](#) [MinGW](#) [msys](#) [编译](#) [ffmpeg](#)

个人分类：[FFMPEG](#)

所属专栏：[FFmpeg](#)

此PDF由[spygg](#)生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com