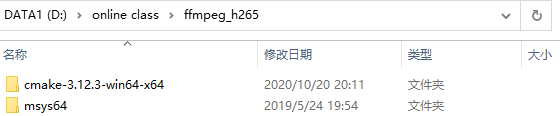
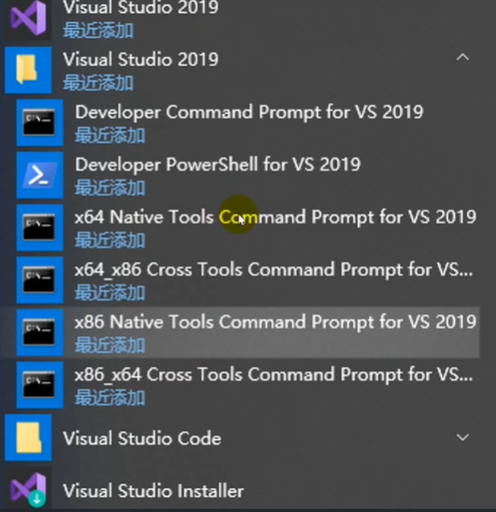
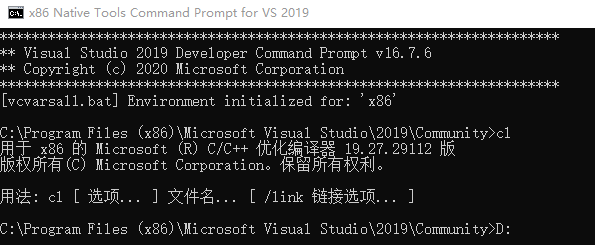
### msys2安装和配置

1. 将下载的压缩包“1环境准备步骤说明VS2019控制台和msys2运行环境准备”解压，解压后的目录如下（注意解压后可能存在二级目录，可适当调整下，另外用英文目录，避免不必要的麻烦）

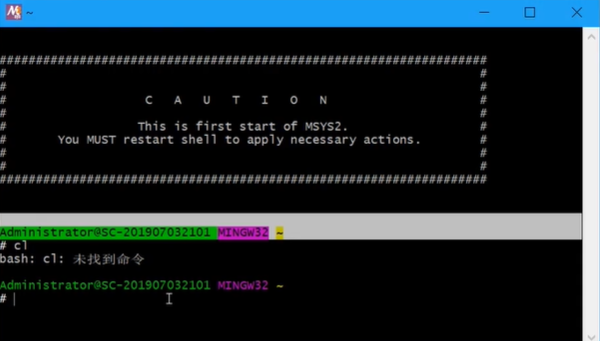


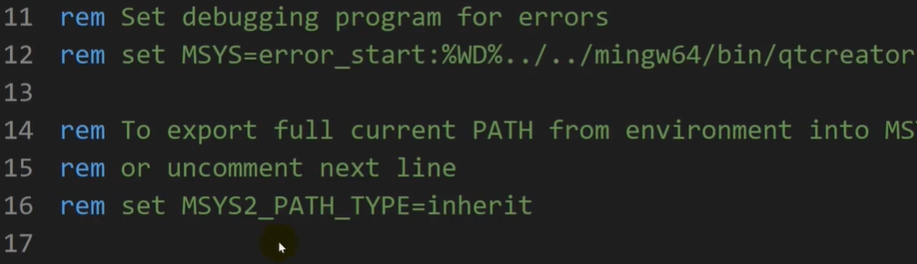
1. 打开32位的vs2019原生编译环境，如下。当然，如果要编64位的，选择x64 Native即可，这里以32位作为例子。



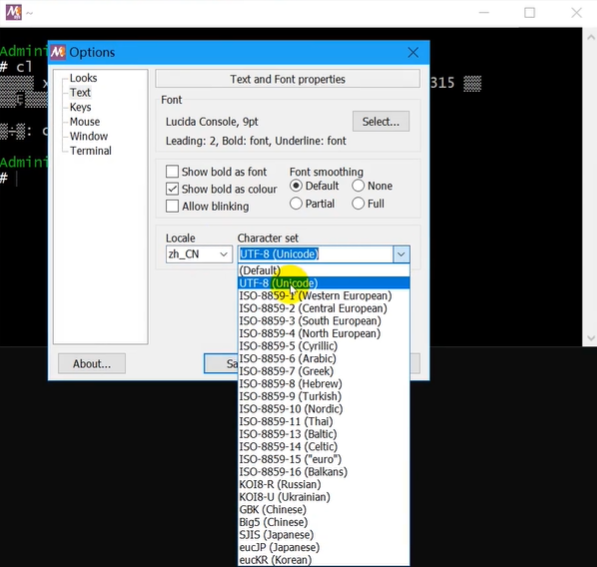


1. 进入到msys2目录，输入命令msys2\_shell.cmd -mingw32。这时，msys2会做一些初始化命令。初始化完之后，输入cl，会提示找不到该命令，主要是环境变量的问题，此时要修改命令msys2\_shell.cmd文件，去掉16行前面的“rem”即可。

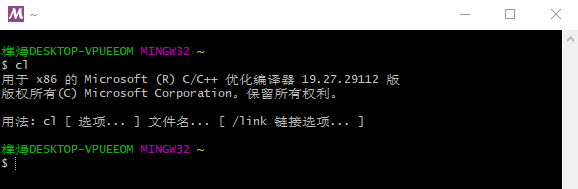




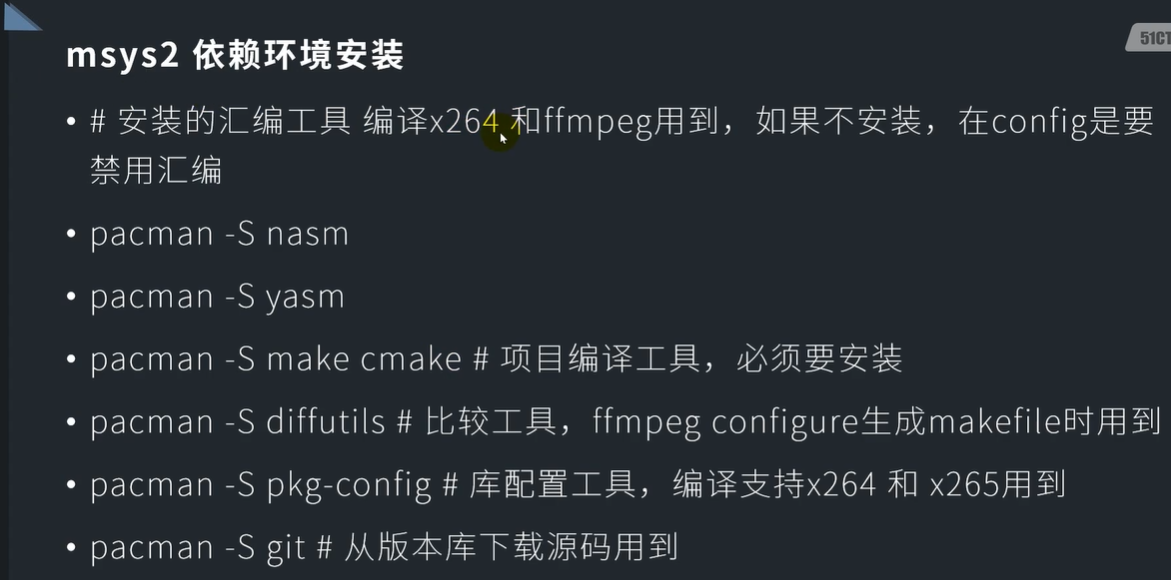
1. 如果msys2出现乱码，选择下GBK的中文编码方式，如下。保存后，关闭窗口，重新输入命令msys2\_shell.cmd -mingw32，再次进入到msys2的窗口，输入cl，此时显示正常。也即msys2环境配置成功。



有些时候也不见得中文一定是GBK编码，如果发现设置成了GBK还是显示乱码，可以尝试其他的字符集格式，比如utf-8。



### 2、msys2依赖环境安装

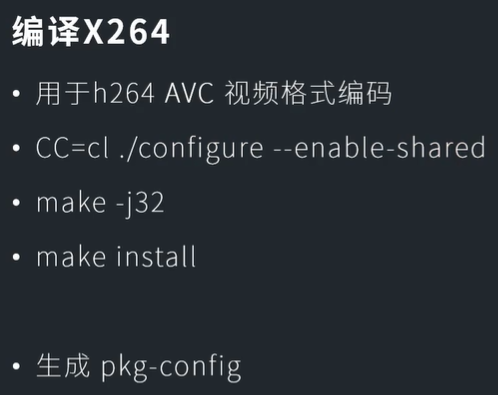


nasm和yasm是编译libx264和ffmpeg时做汇编优化用的。另外，如果编译好了x264库，然后ffmpeg要用到这个库，那么ffmpeg到哪里去找到这个库的头文件呢？此时可以用pkg-config来进行指定。

实际安装msys2的依赖环境时，这一步会有各种各样的问题，最主要的还是网络的问题，网络差无法下载相关的安装包。另外，还会有一些验证问题，如下。解决这些问题不仅仅需要课程视频中介绍的方法，也可参考网络上的方法，例如这个网址的介绍：<https://blog.csdn.net/qq_26973089/article/details/82669348>。

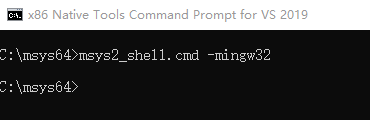


### 3、编译libx264

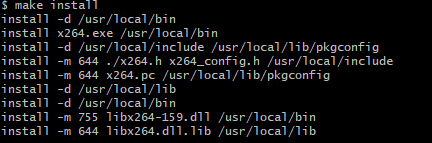


一般来说，不管是在linux中还是在win中的msys2中，默认情况下是用gcc来编译libx264的。这里，在win的msys2中，我们采用vs的编译工具cl来编译，所以上面利用CC=cl来指定编译工具，使得编译出来的库可以在vs中使用。

注：

* 课程上面下载下来的资料中，msys2是免安装的（估计是夏曹俊安装好后直接将生成的安装文件打包给我们的），用这种方式操作的话，后面在安装依赖环境时，会存在密钥错误的问题。所以建议还是到官网<https://www.msys2.org/>下载msys2安装包一步步安装好，然后再安装依赖环境。这样安装的话，没有上面提到的安装问题。但是还是要从x86 Native Tool进入msys2，否则在msys2界面还是会不认得cl命令（当然别忘了修改命令msys2\_shell.cmd文件，去掉16行前面的“rem”）。
* 

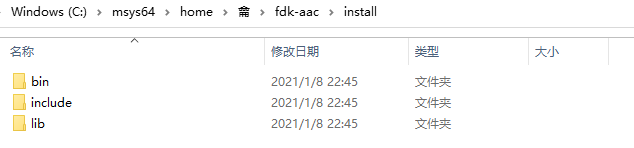
后面编译libx264库时，把源代码拷贝到msys2安装目录里的home下（因为是在msys2环境下编译libx264库），然后进入源代码目录，执行前面截图中的编译步骤。最终make install，如下所示。make install的过程其实是把前面编译产生的头文件、库文件等拷贝到相关路径下。



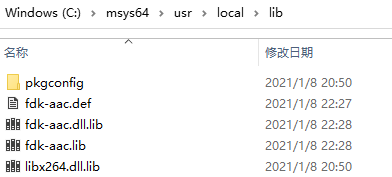
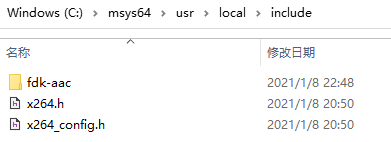
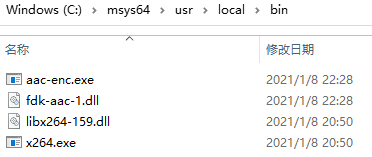
### 4、编译fdk-aac

fdk-aac已经提供了vc的makefile编译环境，所以不需要用到msys2（该环境主要是为了生成windows下的makefile）。直接在vs的原生编译环境窗口中进入到fdk-aac的源码目录，执行“nmake -f Makefile.vc”命令即可。

上一步执行完，fdk-aac已经编译完毕，下面是安装。安装其实就是将编译的结果输出到指定目录，这里将其安装到install目录，然后再将install中的文件拷贝到msys2中usr下面的local中，和前面的libx264编译文件放到一起，以便后面编译ffmpeg时用到。安装的命令是“nmake -f Makefile.vc prefix=.\install install”。



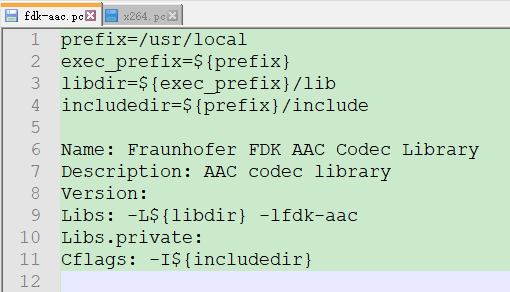
然后选中上面的三个文件夹，将它们全部拷贝到msys2中usr下面的local中，如下。



接下来需要手动生成用于pkgconfig的文件（不知为何编译完后没有自动生成该文件，可能是编译时要在编译命令中添加额外的选项吧。Anyway，这里就手动生成吧！）。将fdk-aac源码目录下的fdk-aac.pc.in文件拷贝到msys64\usr\local\lib\pkgconfig中，并将文件名改成fdk-aac.pc。



下面是修改fdk-aac.pc的文件内容。参照x264.pc修改，修改后的内容如下，像版本号之类用不到的内容先空着。后续编译ffmpeg时有可能还会遇到有关这个pc文件的问题，到时再根据问题的具体log信息来修改该文件内容。



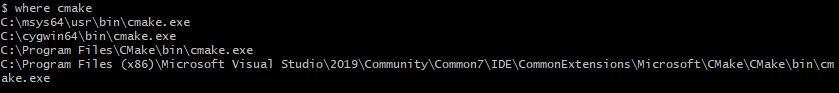
### 5、编译x265

总的来说，编译x265包含下面截图中的几个步骤。

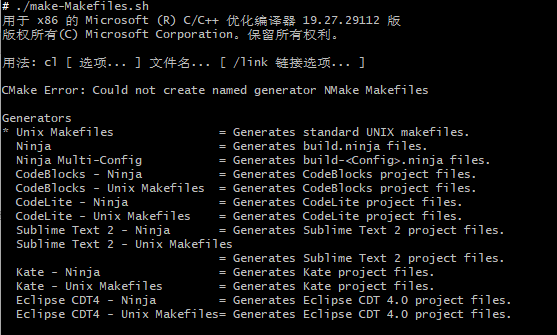


解压下载的压缩包“x265\_20200330源码.tar.gz”，将x265文件夹所包含的源码拷贝到msys2的home目录下，进入到源码中，可以看到相关文件夹。其中，source文件夹中的内容就是源码，build文件夹中的内容是不同平台下面的编译环境。这里进入到build\msys-cl目录中进行编译。

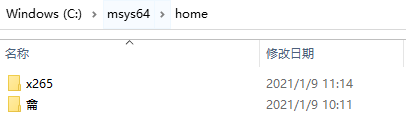
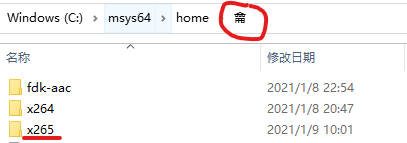
首先输入命令“where cmake”看下在当前环境下能找到的cmake命令，如下所示。

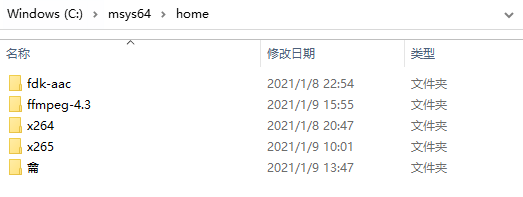


不知为何，在msys2环境下利用cl命令编译x265时，cmake不能使用msys2的usr/bin下的cmake，也不能使用cygwin下的，所以需要把前面两个cmake命令暂时先改革名字（编译完后记得改回来），否则会出现下面的错误。

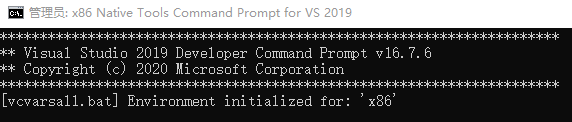


编译x265时，源码路径不能有中文。这个比较隐晦，因为如果有中文，最后报错的提示会是其他信息，不大容易联想到中文路径引起的问题！下面左图x265就在中文“龠”目录下面，编译会有问题（但是为什么之前编译x264和aac没问题不得而知），右图目录往上提了一个层级则OK。以后编译时建议将264和aac的目录都往上提一个层级。

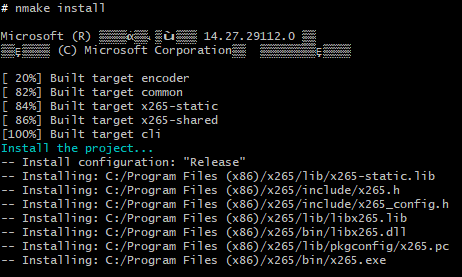




再者，最后install时，默认会把编译的结果安装到C:/Program Files (x86)下面，此时可能出现权限不够无法安装的问题，因此，一开始打开x86 Native编译环境时需要用管理员权限打开，如下。



在管理员权限下一步步进到msys2的编译环境下进行编译。最后编译及安装成功后，会有下面的打印输出。



为了统一管理，把C:/Program Files (x86)/x265下面的内容拷贝到msys64\usr\local下面，跟前的aac和x264一样。最后，修改msys64\usr\local\lib\pkgconfig\下面的x265.pc文件中的第一行prefix字段值。因为已经将x265的编译结果从C:/Program Files (x86)/x265拷贝到了msys64\usr\local下面，所以prefix要相应的改成/usr/local。

### 6、FFmpeg编译

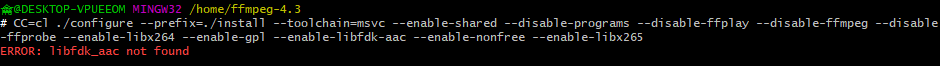
总的来说，编译ffmpeg的命令如下：

CC=cl ./configure --prefix=./install --toolchain=msvc --enable-shared --disable-programs --disable-ffplay --disable-ffmpeg --disable-ffprobe --enable-libx264 --enable-gpl --enable-libfdk-aac --enable-nonfree --enable-libx265

下图是对该命令相关配置项的说明。



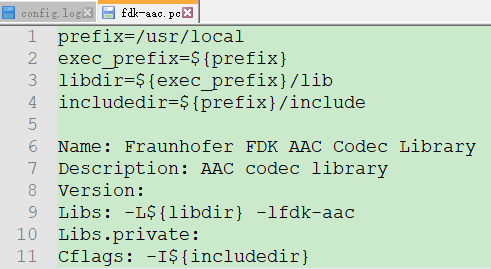
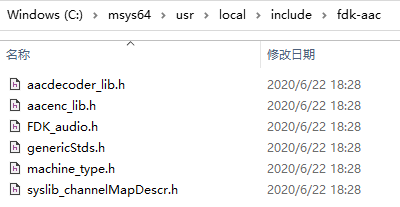
编译ffmpeg的过程中会遇到各种各样的问题，比如下面的libfdk\_aac找不到。



ffmpeg为我们提供了查看详细错误信息的文件，就是目录\home\ffmpeg-4.3\ffbuild下面的config.log文件。以这里的libfdk-aac找不到为例，查看该文件，发现有如下详细信息。

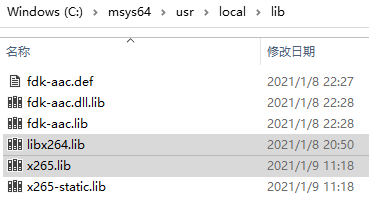
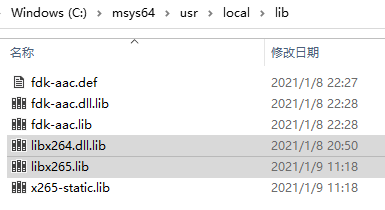


查看头文件目录下的内容以及fdk-aac.pc文件的内容，发现都没有问题，那会是什么原因呢？



原来是没有找到fdk-aac.pc文件。默认情况下，在msys2中配置时，它是到\msys64\mingw32\lib下面去寻找lib相关信息的（当然，如果是64位版本就到\msys64\mingw64\lib中），所以只要将\usr\local\lib下面的pkgconfig 文件夹内容拷贝到\msys64\mingw32\lib\pkgconfig中即可。

继续编译，这时又报出了“ERROR: libx264 not found”的错误，原来是之前编译出来的x264库的名字和现在配置时的要求不一致导致。同样的，x265库也存在着命名导致的配置错误（ERROR: x265 not found using pkg-config）。

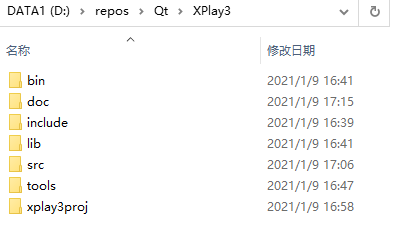


在解决所有的错误后，配置成功。接下来就是make，静静地等待编译结束。最后是make install。该命令会将编译结果输出安装到前面配置的install目录下（--prefix=./install）。

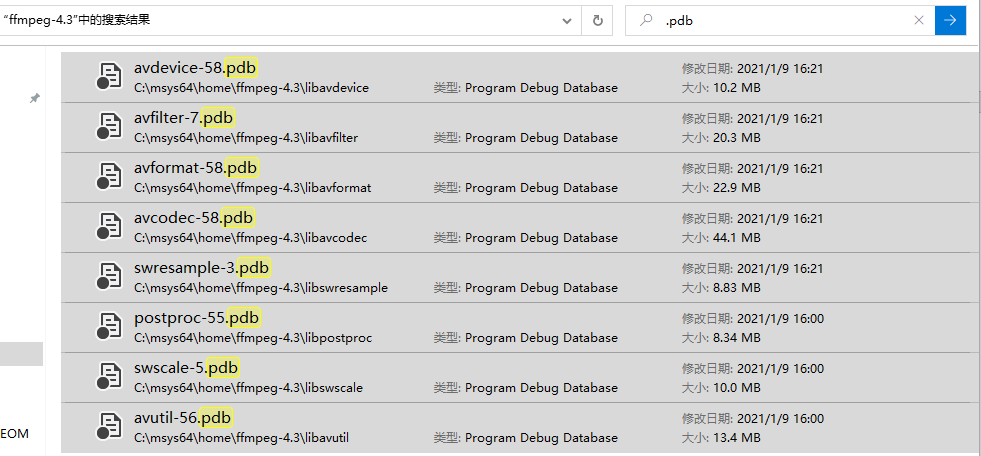
至此，带x264、x265和aac的可用于VS开发环境的ffmpeg就编译完毕了。

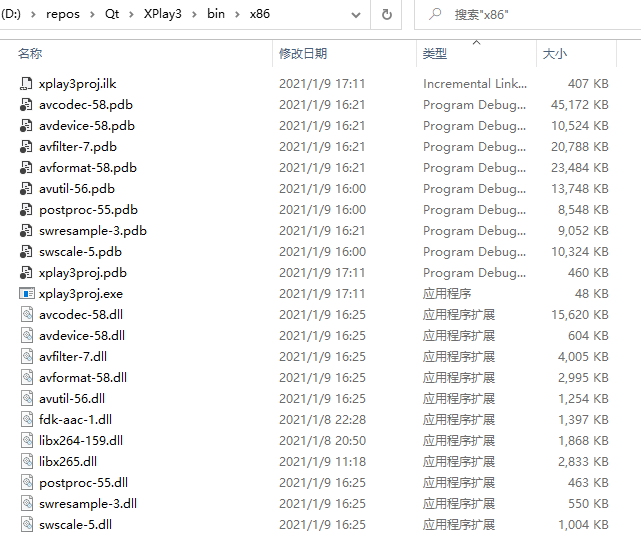
### 7、FFmpeg使用

一般来说，实际工程中的项目都会有一个目录结构，这里采用下面的目录结构。右侧的xplay3proj放VS工程配置文件。具体怎么配置这个VS工程或者说是解决方案（VS里一个解决方案可以包含多个项目）这里不细说了，网上有很多教程，也比较简单，下面讲下怎么在该工程里支持对ffmpeg源码的追踪功能，该功能在调试时会用得到。



首先在前面编译后的ffmpeg源码路径下搜索pdb文件，将这些pdb文件拷贝到Qt\XPlay3\bin\x86下面（这个路径就是实际程序输出及运行的路径，如果当前的程序是64位程序，则在Qt\XPlay3\bin\x64下面）。





然后将ffmpeg的源码路径（已将源码从之前的编译路径复制到了项目路径Qt\XPlay3\tools\ffmpeg-4.3中）配置到解决方案属性中的调试源文件中，如下面所示。**保存后，重新生成下解决方案。**此时，如果在ffmpeg函数上打断点，按F11就能跟踪到ffmpeg的源码中去了！

