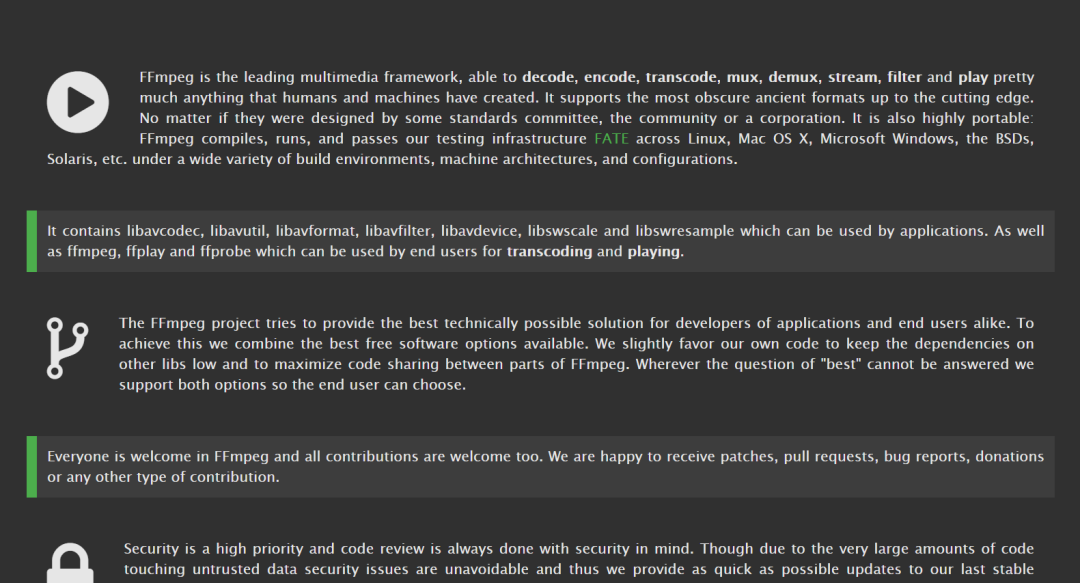
文档：

<https://mp.weixin.qq.com/s/nDXCriWe_D-m-pCqm73JPw>

从本篇文章开始，将开始写一些学习音视频方面的学习笔记，记录自己的成长过程！

我们先来了解一下ffmpeg是什么吧？最官方的解释和介绍，当然是去ffmpeg官方看，ffmpeg官方地址：

<https://www.ffmpeg.org/about.html>



这里大概的说一下：

FFmpeg是领先的多媒体框架，能够解码，编码， 转码，mux，demux，流，过滤和播放人类和机器创建的几乎所有内容。它支持最模糊的古代格式，直至最前沿。无论它们是由某些标准委员会，社区还是公司设计的。它还具有高度的可移植性：FFmpeg可以在各种构建环境，机器体系结构和配置下，跨Linux，Mac OS X，Microsoft Windows，BSD，Solaris等编译，运行并通过我们的测试基础架构 FATE。

它包含可以由应用程序使用的**libavcodec，libavutil，libavformat，libavfilter，libavdevice，libswscale和libswresample。与ffmpeg，ffplay和ffprobe一样，最终用户也可以使用它们进行转码和播放。**

当然如果这些还不够你作为知识认知上来了解一个东西的话，可以再多看几篇文章，这里我就不再造轮子了，咋们接下来开始来搭建windows和Linux开发环境，**正所谓：工欲善其事 必先利其器**。可能搭建环境的时候，会遇到一些坑。

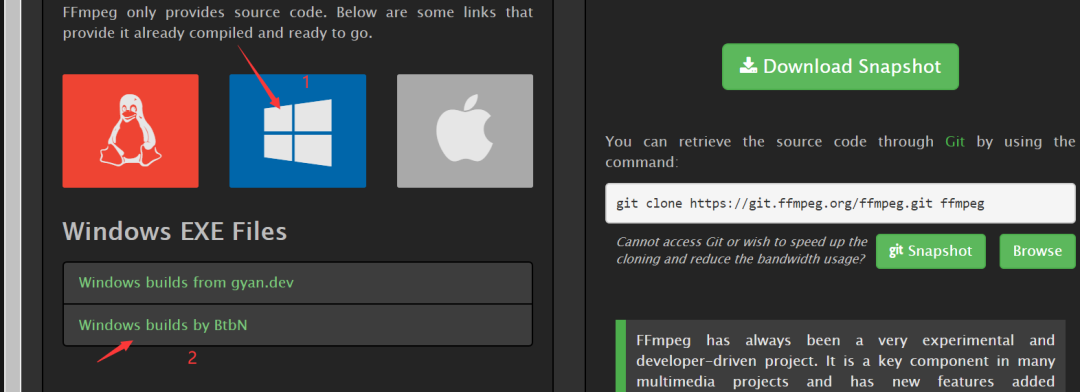
**一、windows环境搭建：**

其实这个礼拜我当时去搭建环境的时候，去网上找了一些博客资料，但是发现之前的网站打不开了，而且现在官网变化有点大，你可能看以前别的文章资料参考，按照操作来，去官网下载安装包，但是其实根本就不是那回事，如果没有仔细观察的话，现在官网下载windows包是这样的：



1、环境搭建步骤;

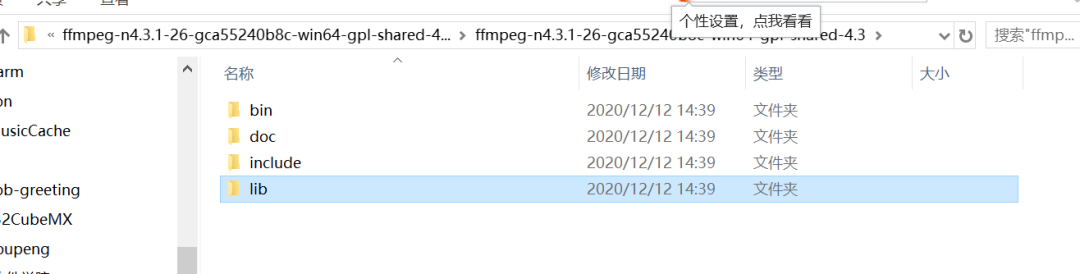
(1)从官网下载安装包：





(2)开始解压安装包和文件说明：

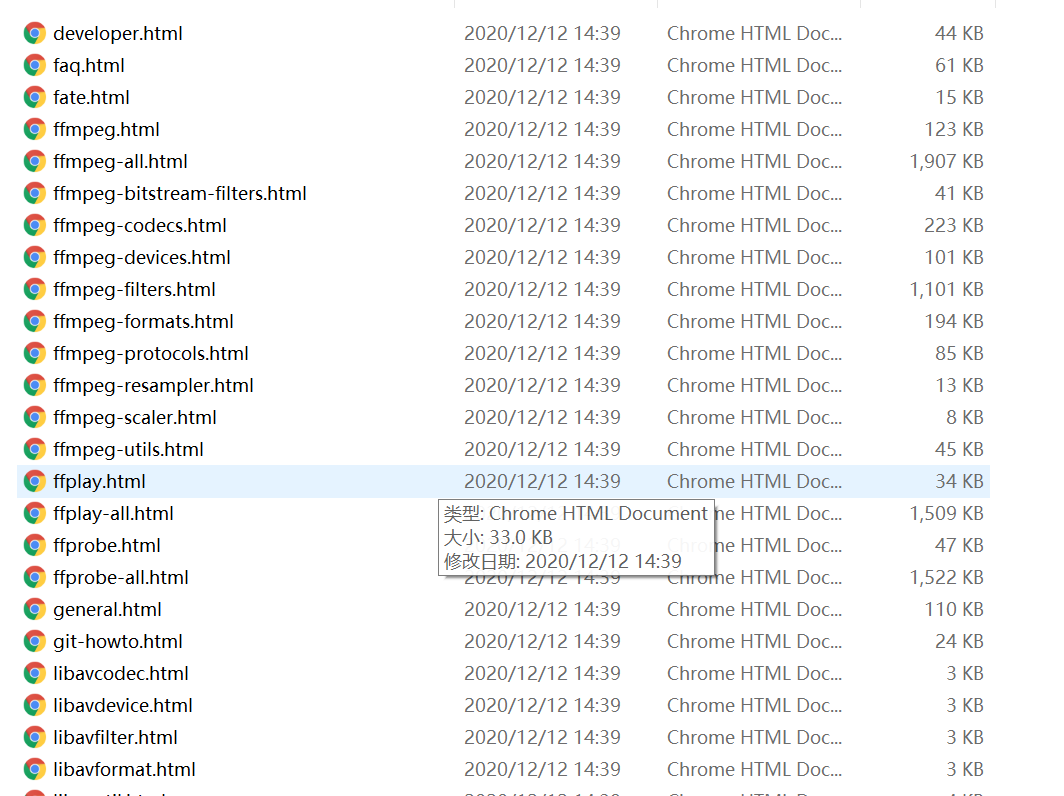
我们把刚才下载下来的包进行解压：



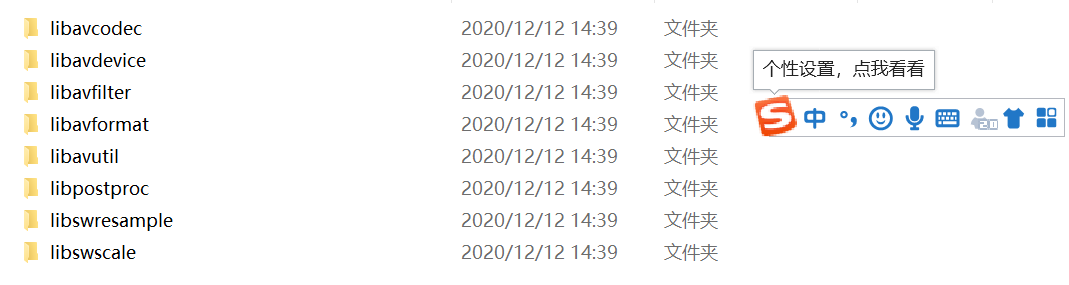
解压后有四个文件：bin里面是安装的应用程序以及一些动态链接库：



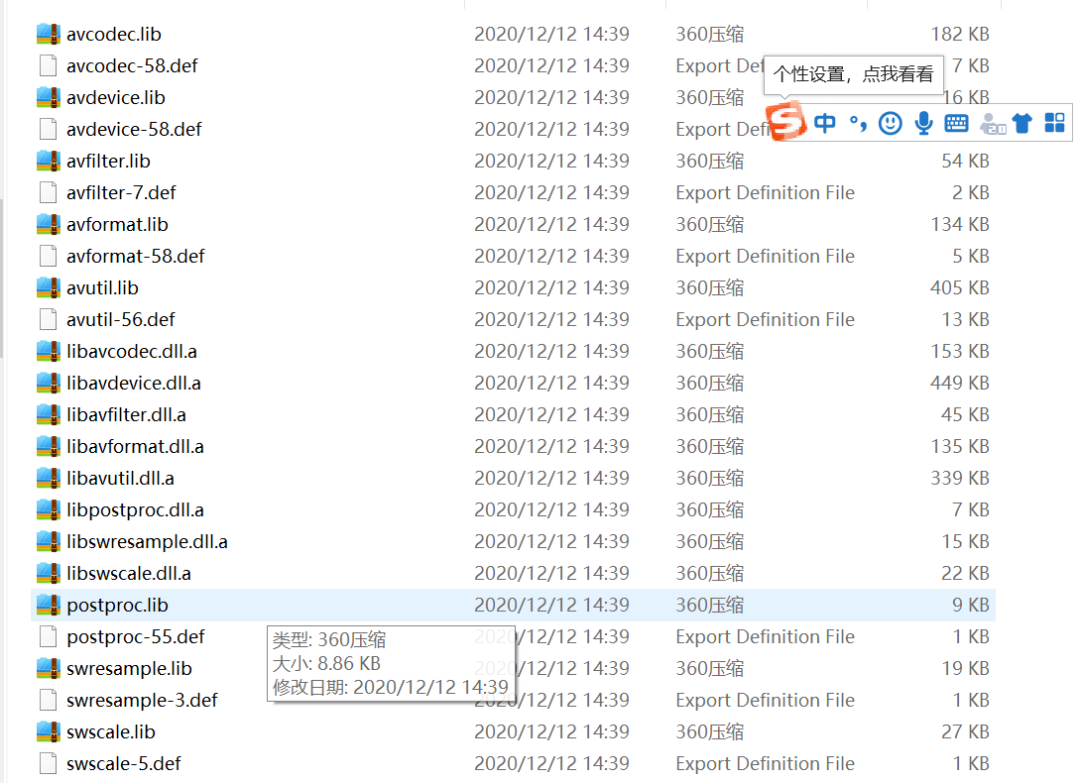
doc文件里面是一些相关关于ffmpeg的html说明文档：



include里面是一些相关库的头文件：



lib里面是相关的库：



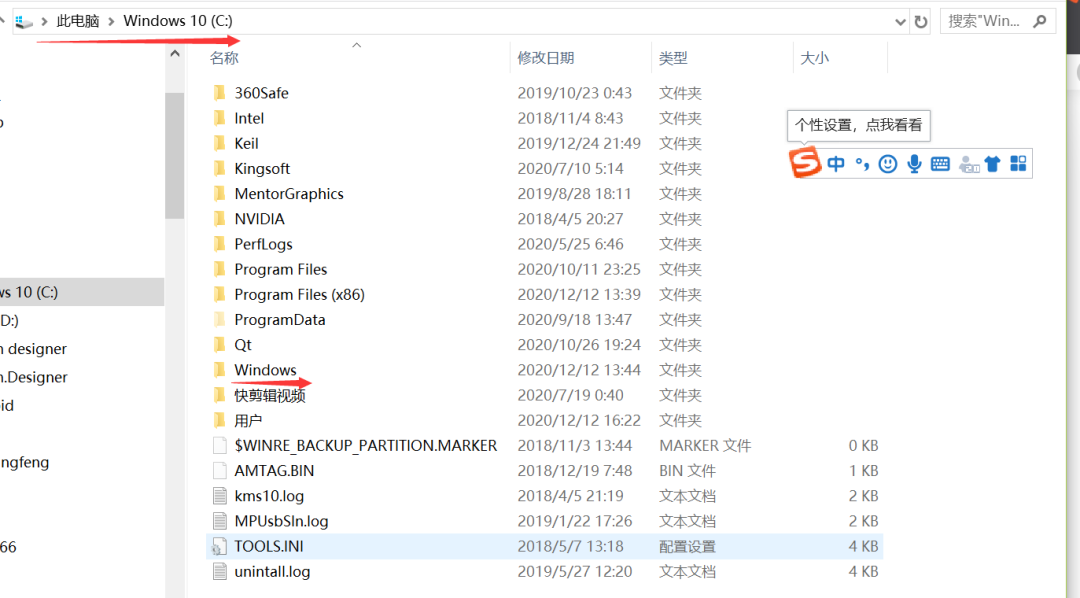
(3)开始安装：

这里我们把bin文件下的这三个应用程序拷贝到你的c盘下的Windows目录下：

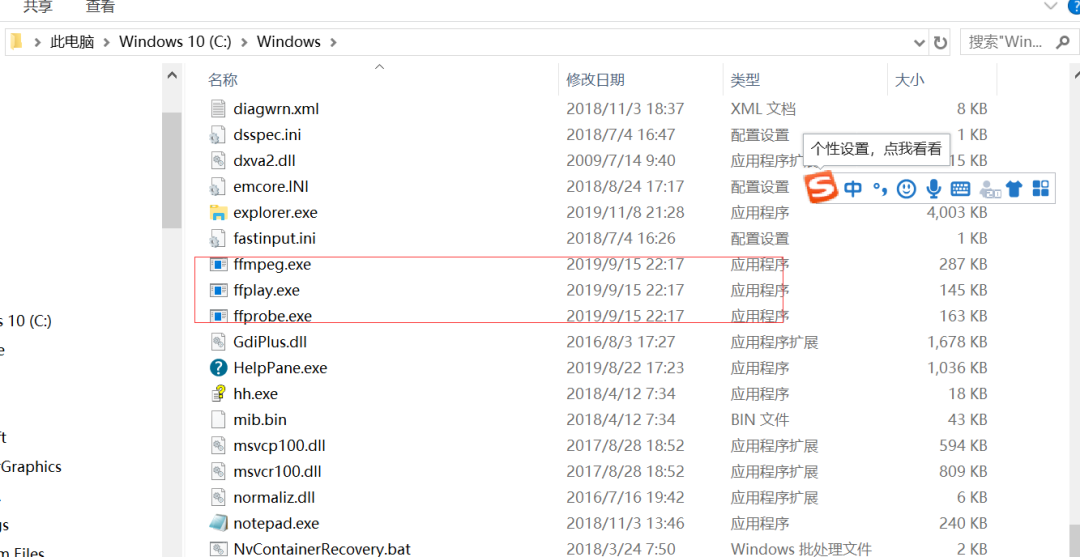
a、



b、



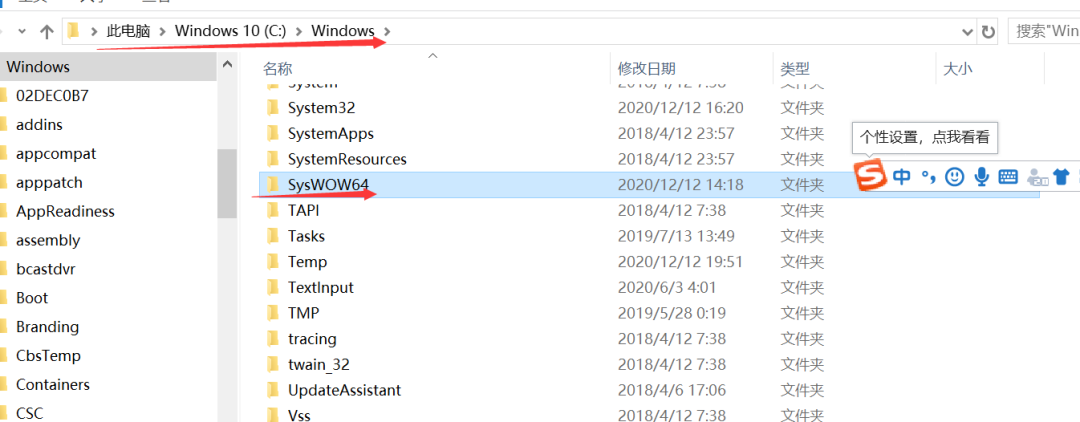
c、直接粘贴到到windows目录即可：

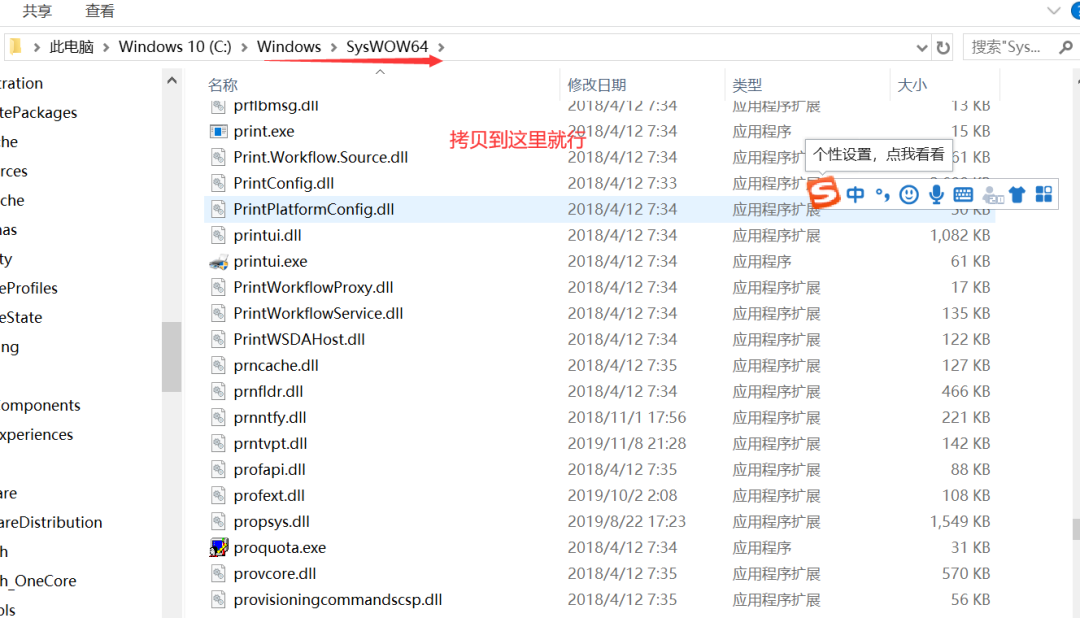


现在我们打开命令提示符，检查一下ffmpeg是否安装成功：



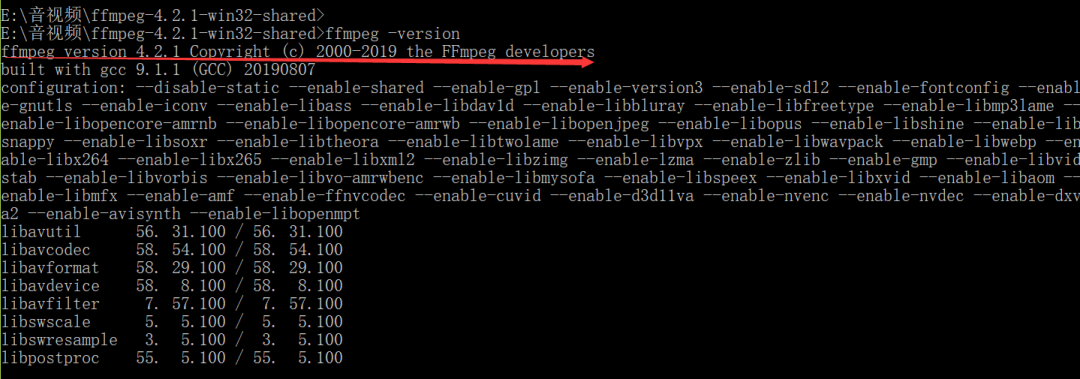
你会发现还缺乏一下动态链接，这个时候我们可以按照提示把刚才剩下的那些以“.dll”的文件拷贝到windows目录的SysWOW64目录即可：





然后我们再测试一下ffmpeg的版本是多少：

ffmpeg -version



注：估计这里可能有些人会遇到问题，还是不能查看，我这里后面还是采用了4.2.1的版本。

没错，如果你到了这里可以看到我上面一样的打印，那么恭喜，你的windows环境就搭建成功了

**二、linux环境搭建:**

这里我没有采用直接用命令的方式来下载ffmpeg，这里我参考了官网的源码安装步骤，可能会比较慢：

<https://trac.ffmpeg.org/wiki/CompilationGuide/Ubuntu>

1、首先在home目录下创建需要的目录（注这里最好和我一样的操作）：

txp@txp-virtual-machine:~$ mdkir ffmpeg\_sources ffmpeg\_build bin

2、下载所要的依赖环境配置：

在下载之前，你最好把自己的ubuntu镜像服务器换成国内的比较好，速度会快很多（这里我用的是ubuntu16.04）;同时在之前最好更新一下：

sudo apt install update

然后直接进行下面操作：

sudo apt-get update -qq && sudo apt-get -y install autoconf automake build-essential cmake git-core libass-dev libfreetype6-dev libgnutls28-dev libmp3lame-dev libsdl2-dev libtool libva-dev libvdpau-dev libvorbis-dev libxcb1-dev libxcb-shm0-dev libxcb-xfixes0-dev meson ninja-build pkg-config texinfo wget yasm3、源码安装一些所需的库：

可以把这里下载的库统一放在我们刚才建立的ffmpeg目录下，方便管理!

a、NASM

cd ~/ffmpeg\_sources && \  
wget https://www.nasm.us/pub/nasm/releasebuilds/2.14.02/nasm-2.14.02.tar.bz2 && \  
tar xjvf nasm-2.14.02.tar.bz2 && \  
cd nasm-2.14.02 && \  
./autogen.sh && \  
PATH="$HOME/bin:$PATH" ./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin" && \  
make && \  
make install

b、Yasm

cd ~/ffmpeg\_sources && \  
wget -O yasm-1.3.0.tar.gz https://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz && \  
tar xzvf yasm-1.3.0.tar.gz && \  
cd yasm-1.3.0 && \  
./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin" && \  
make && \  
make install

c、libx264

cd ~/ffmpeg\_sources && \  
git -C x264 pull 2> /dev/null || git clone --depth 1 https://code.videolan.org/videolan/x264.git && \  
cd x264 && \  
PATH="$HOME/bin:$PATH" PKG\_CONFIG\_PATH="$HOME/ffmpeg\_build/lib/pkgconfig" ./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin" --enable-static --enable-pic && \  
PATH="$HOME/bin:$PATH" make && \  
make install

d、libx265

sudo apt-get install libnuma-dev && \  
cd ~/ffmpeg\_sources && \  
git -C x265\_git pull 2> /dev/null || git clone https://bitbucket.org/multicoreware/x265\_git && \  
cd x265\_git/build/linux && \  
PATH="$HOME/bin:$PATH" cmake -G "Unix Makefiles" -DCMAKE\_INSTALL\_PREFIX="$HOME/ffmpeg\_build" -DENABLE\_SHARED=off ../../source && \  
PATH="$HOME/bin:$PATH" make && \  
make install

e、libvpx

cd ~/ffmpeg\_sources && \  
git -C libvpx pull 2> /dev/null || git clone --depth 1 https://chromium.googlesource.com/webm/libvpx.git && \  
cd libvpx && \  
PATH="$HOME/bin:$PATH" ./configure --prefix="$HOME/ffmpeg\_build" --disable-examples --disable-unit-tests --enable-vp9-highbitdepth --as=yasm && \  
PATH="$HOME/bin:$PATH" make && \  
make install

f、libfdk-aac

cd ~/ffmpeg\_sources && \  
git -C fdk-aac pull 2> /dev/null || git clone --depth 1 https://github.com/mstorsjo/fdk-aac && \  
cd fdk-aac && \  
autoreconf -fiv && \  
./configure --prefix="$HOME/ffmpeg\_build" --disable-shared && \  
make && \  
make install

g、libmp3lame

cd ~/ffmpeg\_sources && \  
wget -O lame-3.100.tar.gz https://downloads.sourceforge.net/project/lame/lame/3.100/lame-3.100.tar.gz && \  
tar xzvf lame-3.100.tar.gz && \  
cd lame-3.100 && \  
PATH="$HOME/bin:$PATH" ./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin" --disable-shared --enable-nasm && \  
PATH="$HOME/bin:$PATH" make && \  
make install

h、libopus

cd ~/ffmpeg\_sources && \  
git -C opus pull 2> /dev/null || git clone --depth 1 https://github.com/xiph/opus.git && \  
cd opus && \  
./autogen.sh && \  
./configure --prefix="$HOME/ffmpeg\_build" --disable-shared && \  
make && \  
make install

i、libaom

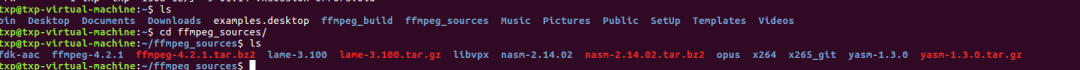
cd ~/ffmpeg\_sources && \  
git -C aom pull 2> /dev/null || git clone --depth 1 https://aomedia.googlesource.com/aom && \  
mkdir -p aom\_build && \  
cd aom\_build && \  
PATH="$HOME/bin:$PATH" cmake -G "Unix Makefiles" -DCMAKE\_INSTALL\_PREFIX="$HOME/ffmpeg\_build" -DENABLE\_SHARED=off -DENABLE\_NASM=on ../aom && \  
PATH="$HOME/bin:$PATH" make && \  
make install

注：这个暂时国内下载不了，服务器在外国。。。。。

o、libsvtav1

cd ~/ffmpeg\_sources && \  
git -C SVT-AV1 pull 2> /dev/null || git clone https://github.com/AOMediaCodec/SVT-AV1.git && \  
mkdir -p SVT-AV1/build && \  
cd SVT-AV1/build && \  
PATH="$HOME/bin:$PATH" cmake -G "Unix Makefiles" -DCMAKE\_INSTALL\_PREFIX="$HOME/ffmpeg\_build" -DCMAKE\_BUILD\_TYPE=Release -DBUILD\_DEC=OFF -DBUILD\_SHARED\_LIBS=OFF .. && \  
PATH="$HOME/bin:$PATH" make && \  
make install

最终我们可以下载ffmpeg目录看到下载的库文件;



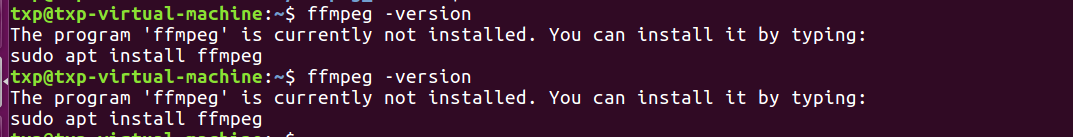
4、现在我们可以开始下载ffmpeg安装包并解压：

cd ~/ffmpeg\_sources && \  
wget -O ffmpeg-snapshot.tar.bz2 https://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2 && \  
tar xjvf ffmpeg-snapshot.tar.bz2 && \  
cd ffmpeg && \  
PATH="$HOME/bin:$PATH" PKG\_CONFIG\_PATH="$HOME/ffmpeg\_build/lib/pkgconfig" ./configure \  
  --prefix="$HOME/ffmpeg\_build" \  
  --pkg-config-flags="--static" \  
  --extra-cflags="-I$HOME/ffmpeg\_build/include" \  
  --extra-ldflags="-L$HOME/ffmpeg\_build/lib" \  
  --extra-libs="-lpthread -lm" \  
  --bindir="$HOME/bin" \  
  --enable-gpl \  
  --enable-gnutls \  
  --enable-libaom \  
  --enable-libass \  
  --enable-libfdk-aac \  
  --enable-libfreetype \  
  --enable-libmp3lame \  
  --enable-libopus \  
  --enable-libsvtav1 \  
  --enable-libvorbis \  
  --enable-libvpx \  
  --enable-libx264 \  
  --enable-libx265 \  
  --enable-nonfree && \  
PATH="$HOME/bin:$PATH" make && \  
make install && \  
hash -r



注意：这里一定是在之前的那些库安装之后再操作这个步骤，不然有些库找不到，导致你编译报错，而且这里编译的时间蛮久的，大概十几分钟左右！

这里环境就搭建完了，下面我来测试一下环境是否搭建成功。记得我们得重新打开一个终端来操作，不然会显示操作失败：



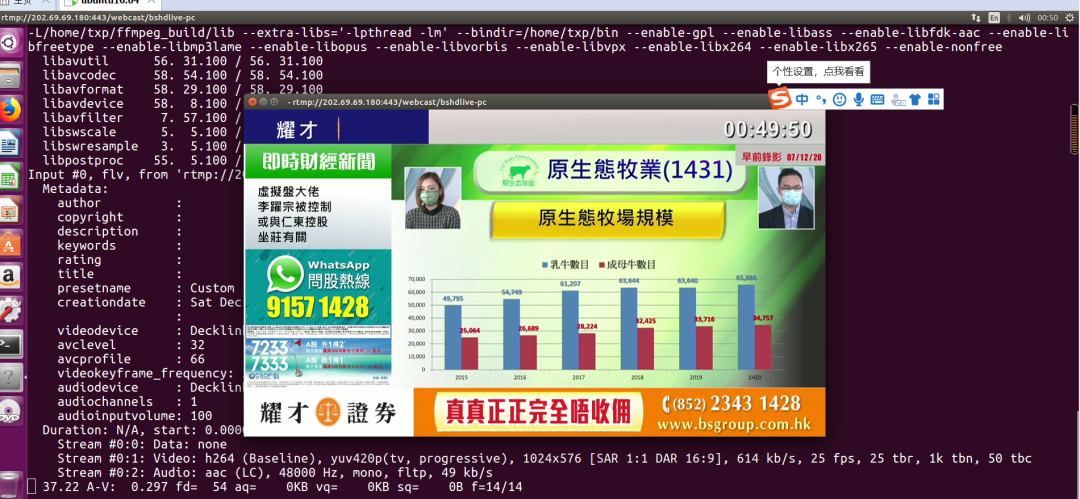
所以重新打开一个终端：



最后我们来播放一个视频看看，是否真的有效果：

ffplay rtmp://202.69.69.180:443/webcast/bshdlive-pc

效果展示：



当然你也可以播放你本地文件里面的视频，这里我就不演示了，很简单！

**三、总结：**

上面的环境搭建可能费点时间，不过还是不建议本地安装ffmpeg。

文章相关参考：

1、

<https://trac.ffmpeg.org/wiki/CompilationGuide/Ubuntu>

2、

<https://www.ffmpeg.org/about.html>

<https://ke.qq.com/webcourse/index.html#cid=468797&term_id=100561187&taid=4217103834359613&vid=5285890796072702423>