

一、安装ffmpeg

ex:通常情况下都没有root权限，也做不了sudo操作，所以应该下源码进行编译生成库后，才能供C/C++调用，编译器链接，安装ffmpeg的原因是，用其完成从mp3到pcm的转换。某个软件-version可以看软件是否在系统中已经有了如ffmpeg -version

1.创建总目录：mkdir -p ~/local/Fffmpeg/{src,ffmpeg} # src 存放源码，ffmpeg 存放编译后的二进制

2.ffmpeg需要依赖git make gcc yasm (nasm)，在公司机器上已经有前三者，故只需要安装yasm，以下演示安装yasm

```
1
2  mkdir -p ~/local      #创建根目录，以后的每个工程的源码编译安装都可以放在这里方便查找
3  cd ~/local && mkdir -p Yasm && cd Yasm
4  mkdir -p yasm src && cd src
5  wget http://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz && tar xzf yasm-1.3.0.tar.gz && cd yasm-1.3.0
6
7  ./configure --prefix=$HOME/local/Yasm/yasm
8  make -j$(nproc)
9  make install #此时~/local/Yasm/yasm 下能找到 bin include lib share 四个文件夹
10 export PATH=$HOME/local/Yasm/yasm/bin:$PATH # 添加到 PATH 或者写进用户环境变量 ~/.bashrc
```

4.下载ffmpeg源码，准备环境，解压，配置，编译，安装

```
1
2 mkdir -p ~/local/Ffmpeg/{src,ffmpeg} # src 存放源码, ffmpeg 存放编译后的二进制
3 cd ~/local/Ffmpeg/src
4 git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg-src && cd ffmpeg-src #git clone
  https://gitee.com/mirrors/ffmpeg.git && cd ffmpeg #下载慢可尝试
5 git checkout n5.1.2 #切换到稳定版
6 #基础配置如下
7 ./configure --prefix=$HOME/local/Ffmpeg/ffmpeg --enable-shared --disable-static --
  extra-cflags="-I$HOME/local/Ffmpeg/include" --extra-ldflags="-L$HOME/local/Ffmpeg/lib"
  #等
8
9 若启用额外编码器如下
10 #./configure --prefix=$HOME/local/ffmpeg --enable-libopus --enable-shared --disable-
  static -enable-gpl --enable-libx264 --extra-cflags="-I$HOME/local/Ffmpeg/include" --
  extra-ldflags="-L$HOME/local/Ffmpeg/lib"
11
12 make -j$(nproc)
13 make install #在~/local/Ffmpeg/ffmpeg下有bin include lib share 文件夹
14
15 检查可执行文件和动态库是否存在
16 ls ~/local/Ffmpeg/ffmpeg/bin/ffmpeg # /home/huchengyuan/local/Ffmpeg/ffmpeg/bin/ffmpeg
17 ls ~/local/Ffmpeg/ffmpeg/lib/libavcodec.so
18
19 添加环境
20 export PATH=$HOME/local/Ffmpeg/ffmpeg/bin:$PATH
21 export LD_LIBRARY_PATH=$HOME/local/Ffmpeg/ffmpeg/lib:$LD_LIBRARY_PATH # 动态库路径
22 source ~/.bashrc
23 验证环境
24 ffmpeg -version #会显示版本信息
```

二、安装opus

ex 由于运行的机器（运行环境）与服务器（编译环境）不一致，故需要进行交叉编译。opus格式音频文件需要封装成能播放的格式，则需要依赖两个库 ogg 和 opusenc，安装opusenc需要制定opus位置，故安装顺序为ogg opus opusenc

1.安装交叉编译环境

```
1 sudo apt update
2 sudo apt install gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf
3 #验证交叉编译环境 arm-linux-gnueabihf-gcc --version # 应显示ARM架构的gcc
```

1.安装ogg

```
1 mkdir -p ~/local/arm7/Ogg
2 cd ~/local/arm7/Ogg && mkdir ogg src
3 cd src
4 git clone https://github.com/xiph/ogg.git
5 cd ogg
6 ./autogen.sh # 如果缺少此脚本, 运行 autoreconf -i
7 ./configure --host=arm-linux-gnueabihf --prefix=$HOME/local/arm7/Ogg/ogg --disable-shared --enable-static
8 make -j$(nproc)
9 make install # 这里能 在 ~/local/arm7/Ogg/ogg 看到include 和 lib share
```

2.安装opus

```
1 mkdir -p ~/local/arm7/Opus
2 cd ~/local/arm7/Opus && mkdir opus src
3 cd src && wget https://archive.mozilla.org/pub/opus/opus-1.3.1.tar.gz && tar -xzf opus-1.3.1.tar.gz && cd opus-1.3.1
4 ./configure --prefix=$HOME/local/arm7/Opus/opus --host=arm-linux-gnueabihf --enable-fixed-point --enable-arm-neon-intrinsics --disable-shared
5 make -j$(nproc) && make install #编译并安装到用户目录 这里能 在 ~/local/arm7/Opus/opus 看到include 和 lib share
6
7 #file ~/local/arm7/Opus/opus/lib/libopus.a #用于验证
8 ls ~/local/arm7/Opus/opus #能看到include lib share 三个文件夹
9 file ~/local/arm7/Opus/opus/lib/libopus.a #输出如 /home/huchengyuan/opus-armv7/lib/libopus.a: current ar archive
```

3.安装opusenc

```

1  mkdir -p ~/local/arm7/Opusenc
2  cd ~/local/arm7/Opusenc && mkdir opusenc src
3  cd src
4  git clone https://github.com/xiph/opusenc.git
5  cd opusenc
6  ./autogen.sh # 如果缺少此脚本, 运行 autoreconf -i
7  export
   PKG_CONFIG_PATH=$HOME/local/arm7/Opus/opus/lib/pkgconfig:$HOME/local/arm7/Oog/oog/lib/p
   kgconfig:$PKG_CONFIG_PATH #设置ogg路径和opus路径
8  ./configure --host=arm-linux-gnueabi --prefix=$HOME/local/arm7/Opusenc/opusenc --
   disable-shared --enable-static
9  make -j$(nproc)
10 make install #在~/local/arm7/Opusenc/opusenc 下能看到include lib share

```

4.将装的所有库, 写到bash环境变量中(将以下内容, 写到~/bashrc 末尾) 保存后运行 source ~/.bashrc

```

1  export PATH=$HOME/local/Yasm/yasm/bin:$PATH
2  export PATH=$HOME/local/Ffmpeg/ffmpeg/bin:$PATH
3  export LD_LIBRARY_PATH=$HOME/local/Ffmpeg/ffmpeg/lib:$LD_LIBRARY_PATHexport
4  export
   PKG_CONFIG_PATH=$HOME/local/arm7/Opus/opus/lib/pkgconfig:$HOME/local/arm7/Oog/oog/lib/p
   kgconfig:$PKG_CONFIG_PATH

```

三、运行 demo

ex. 将一个 mp3文件用ffmpeg命令转为pcm, 再将该pcm转为可播放的opus, 注意: demo中的可执行程序, 需要在目标机器 (arm机器上面执行)

1.项目层次结构示意

```

1  demo
2      res - 资源文件
3      src - 源文件
4      Makefile - 编译规则
5      demo - 编译出来的可执行文件
6      Test - 将编译出来的可执行文件和pcm文件放在一起, 准备往目标arm机器里面拷贝, 执行

```

2. 找到一个隔壁卖菜大妈不要的 .mp3 文件，我找到的是 ljj.mp3 将其放到 demo/res 下

ex: 从 mp3 转 pcm，在编译环境下比较容易，上面的介绍中，ffmpeg 被安装在了编译机（ubuntu 服务器中），所以 ffmpeg 相关的命令，请在服务器中执行

```
1 #转换为 16-bit 单声道 PCM      ***这里已经默认 切换到了demo下***
2 cd res && ffmpeg -i input.mp3 -f s16le -ac 1 -ar 16000 ljj.pcm
3 #参数解析:
4 #-i input.mp3: 输入 MP3 文件。
5 #-f s16le: 指定输出格式为 16-bit 小端序 PCM (s16le= signed 16-bit little-endian)。
6 #-ac 1: 输出 单声道 (-ac 2为立体声)。
7 #-ar 16000: 设置采样率为 16000 (可改为 16000、8000等)。
8 #output.pcm: 输出的 PCM 文件 (纯数据流, 无文件头)。
9
10 #转换为 16-bit 立体声 PCM (16kHz) 用以下命令
11 #ffmpeg -i input.mp3 -f s16le -ac 2 -ar 16000 output_stereo.pcm
12 make #生成了可执行文件
13 cp -rf res/ljj.pcm demo Test
```

3. 将 Test 文件夹整个拷贝到 ARM 机器上面执行。

a. 切换到 Test 文件夹下

b. 执行以下指令，执行完成后，将得到 ljj.pcm.opus，该文件已经可以在 vlc 中播放。

```
1 ./demo llj.pcm
```

