

## 原 FFMpeg示例程序合集-Git批量获取脚本

2015年02月23日 13:51:58 阅读数：6291

此前做了一系列有关FFmpeg的示例程序，组成了《FFmpeg示例程序合集》，其中包含了如下项目：

simplest ffmpeg player:	最简单的基于FFmpeg的视频播放器
simplest ffmpeg audio player:	最简单的基于FFmpeg的音频播放器
simplest ffmpeg pic encoder:	最简单的基于FFmpeg的图像编码器
simplest ffmpeg video encoder:	最简单的基于FFmpeg的视频编码器
simplest ffmpeg audio encoder:	最简单的基于FFmpeg的音频编码器
simplest ffmpeg streamer:	最简单的基于FFmpeg的推流器（推送RTMP）
simplest ffmpeg mem handler:	最简单的基于FFmpeg的内存读写例子
simplest ffmpeg device:	最简单的基于FFmpeg的AVDevice例子
simplest ffmpeg format:	最简单的基于FFmpeg的封装格式方面的例子
simplest ffmpeg video filter:	最简单的基于FFmpeg的AVfilter例子（水印叠加）
simplest ffmpeg swscale:	最简单的基于FFmpeg的libswscale的示例

为了方便今后代码的升级，将上述工程的代码统一上传到了Git代码托管平台（SourceForge，Github，开源中国），并且做了几个简单的脚本用于获取代码，如下所列：

- 克隆（simplest\_ffmpeg\_demos\_git\_clone\_all.sh）：把代码从远端复制到本机（类似于下载）。
- 更新（simplest\_ffmpeg\_demos\_git\_pull\_all.sh）：把代码从远端更新到本机（类似于升级）。
- 重置（simplest\_ffmpeg\_demos\_git\_reset\_all.sh）：回退本机的代码到上一个版本（类似于还原）。
- 归档（simplest\_ffmpeg\_demos\_git\_archive\_all.sh）：将本机的项目代码归档（例如打包为zip）。

这些脚本的下载地址：

【Github】 <https://github.com/leixiaohua1020/leixiaohua1020.github.io/tree/master/batch>

## 克隆

克隆脚本可以“一键下载”（clone命令）源代码至本机，并且设置本地仓库的3个远程仓库（remote）：SourceForge，Github和开源中国。该脚本有一个originstr参数需要设置。该参数决定了从哪里克隆源代码，经过测试发现开源中国的速度明显高于SourceForge和Github。simplest\_ffmpeg\_demos\_git\_clone\_all.sh内容如下所示。



[plain]  

```
1.  echo "======"
2.  echo "simplest ffmpeg demos list ---- git clone all"
3.  echo "Lei Xiaohua"
4.  echo "Communication University of China / Digital TV Technology"
5.  echo "leixiaohua1020@126.com"
6.  echo "http://blog.csdn.net/leixiaohua1020"
7.  echo "======"
8.
9.  originstr="github"
10. #3 kinds of origins:
11. githubstr="github"
12. sourceforgestr="sourceforge"
13. oscstr="osc"
14.
15. #Git Clone
16. #Github
17. #http://leixiaohua1020.github.com/
18. if [ "$originstr" = "$githubstr" ]
19. then
20. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_player.git simplest_ffmpeg_player
21. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_audio_player.git simplest_ffmpeg_audio_player
22. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_picture_encoder.git simplest_ffmpeg_picture_encoder
23. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_video_encoder.git simplest_ffmpeg_video_encoder
24. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_audio_encoder.git simplest_ffmpeg_audio_encoder
25. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_streamer.git simplest_ffmpeg_streamer
26. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_mem_handler.git simplest_ffmpeg_mem_handler
27. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_device.git simplest_ffmpeg_device
28. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_format.git simplest_ffmpeg_format
29. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_video_filter.git simplest_ffmpeg_video_filter
30. git clone https://github.com/leixiaohua1020/simplest_ffmpeg_swscale.git simplest_ffmpeg_swscale
31. #SourceForge
32. #https://sourceforge.net/u/leixiaohua1020/wiki/
33. elif [ "$originstr" = "$sourceforgestr" ]
34. then
35. git clone git://git.code.sf.net/p/simplestffmpegplayer/code simplest_ffmpeg_player
36. git clone git://git.code.sf.net/p/simplestffmpegaudioplayer/code simplest_ffmpeg_audio_player
37. git clone git://git.code.sf.net/p/simplestffmpegpictureencoder/code simplest_ffmpeg_picture_encoder
38. git clone git://git.code.sf.net/p/simplestffmpegvideoencoder/code simplest_ffmpeg_video_encoder
39. git clone git://git.code.sf.net/p/simplestffmpegaudioencoder/code simplest_ffmpeg_audio_encoder
40. git clone git://git.code.sf.net/p/simplestffmpegstreamer/code simplest_ffmpeg_streamer
41. git clone git://git.code.sf.net/p/simplestffmpegmemhandler/code simplest_ffmpeg_mem_handler
42. git clone git://git.code.sf.net/p/simplestffmpegdevice/code simplest_ffmpeg_device
43. git clone git://git.code.sf.net/p/simplestffmpegformat/code simplest_ffmpeg_format
44. git clone git://git.code.sf.net/p/simplestffmpegvideofilter/code simplest_ffmpeg_video_filter
45. git clone git://git.code.sf.net/p/simplestffmegswscale/code simplest_ffmpeg_swscale
46. #OSC
47. #http://git.oschina.net/leixiaohua1020
48. else
49. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_player.git simplest_ffmpeg_player
50. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_audio_player.git simplest_ffmpeg_audio_player
51. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_picture_encoder.git simplest_ffmpeg_picture_encoder
52. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_video_encoder.git simplest_ffmpeg_video_encoder
53. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_audio_encoder.git simplest_ffmpeg_audio_encoder
54. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_streamer.git simplest_ffmpeg_streamer
55. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_mem_handler.git simplest_ffmpeg_mem_handler
56. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_device.git simplest_ffmpeg_device
57. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_format.git simplest_ffmpeg_format
58. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_video_filter.git simplest_ffmpeg_video_filter
59. git clone https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_swscale.git simplest_ffmpeg_swscale
60. fi
```

## 更新

更新脚本可以把本地库的源代码更新到最新版本。随着FFmpeg的API的变更以及一些程序中Bug的修复，项目的源代码需要与时俱进地做一些微调。可以使用pull命令将本地库的源代码“升级”到最新版本。

simplest\_ffmpeg\_demos\_git\_pull\_all.sh内容如下所示。

```
[plain]  
1. echo "=====
2. echo "simplest ffmpeg demos list ---- git pull all"
3. echo "Lei Xiaohua"
4. echo "Communication University of China / Digital TV Technology"
5. echo "leixiaohua1020@126.com"
6. echo "http://blog.csdn.net/leixiaohua1020"
7. echo "=====
8.
9. #remotestr="github"
10. remotestr="osc master"
11.
12. #Git Pull
13. cd simplest_ffmpeg_player
14. git pull $remotestr
15. cd ..
16.
17. cd simplest_ffmpeg_audio_player
18. git pull $remotestr
19. cd ..
20.
21. cd simplest_ffmpeg_picture_encoder
22. git pull $remotestr
23. cd ..
24.
25. cd simplest_ffmpeg_video_encoder
26. git pull $remotestr
27. cd ..
28.
29. cd simplest_ffmpeg_audio_encoder
30. git pull $remotestr
31. cd ..
32.
33. cd simplest_ffmpeg_streamer
34. git pull $remotestr
35. cd ..
36.
37. cd simplest_ffmpeg_mem_handler
38. git pull $remotestr
39. cd ..
40.
41. cd simplest_ffmpeg_device
42. git pull $remotestr
43. cd ..
44.
45. cd simplest_ffmpeg_format
46. git pull $remotestr
47. cd ..
48.
49. cd simplest_ffmpeg_video_filter
50. git pull $remotestr
51. cd ..
52.
53. cd simplest_ffmpeg_swscale
54. git pull $remotestr
55. cd ..
```



## 重置

重置脚本用于将源代码回退到上一个版本。重置命令（reset）在两种情况会使用：

一种情况是改代码的时候改着改着忽然发现代码编译不过了，可是又难以恢复到原来可以编译通过的代码，这时候可以使用重置命令；

另一种情况是改完代码之后，使用更新命令（pull）却提示错误，这时候可以先使用重置命令，然后再使用更新命令。

simplest\_ffmpeg\_demos\_git\_reset\_all.sh内容如下所示。

```
[plain]  

1.  echo "=====
2.  echo "simplest ffmpeg demos list ---- git reset all"
3.  echo "Lei Xiaohua"
4.  echo "Communication University of China / Digital TV Technology"
5.  echo "leixiaohua1020@126.com"
6.  echo "http://blog.csdn.net/leixiaohua1020"
7.  echo "=====
8.
9.
10. #Git Pull
11. cd simplest_ffmpeg_player
12. git reset --hard
13. cd ..
14.
15. cd simplest_ffmpeg_audio_player
16. git reset --hard
17. cd ..
18.
19. cd simplest_ffmpeg_picture_encoder
20. git reset --hard
21. cd ..
22.
23. cd simplest_ffmpeg_video_encoder
24. git reset --hard
25. cd ..
26.
27. cd simplest_ffmpeg_audio_encoder
28. git reset --hard
29. cd ..
30.
31. cd simplest_ffmpeg_streamer
32. git reset --hard
33. cd ..
34.
35. cd simplest_ffmpeg_mem_handler
36. git reset --hard
37. cd ..
38.
39. cd simplest_ffmpeg_device
40. git reset --hard
41. cd ..
42.
43. cd simplest_ffmpeg_format
44. git reset --hard
45. cd ..
46.
47. cd simplest_ffmpeg_video_filter
48. git reset --hard
49. cd ..
50.
51. cd simplest_ffmpeg_swscale
52. git reset --hard
53. cd ..
```

## 归档

归档脚本用于将本地的代码库打包为压缩包 (\*.zip, \*.tar.gz)。我个人感觉Git归档和直接压缩源代码文件夹最大的不同在于Git归档使用了".gitignore"文件，从而忽略掉一些无关紧要的文件。

该脚本运行之后会生成一个名称为"simplerest\_ffmpeg\_demos\_list\_{日期}"的文件夹，并且将归档后的文件保存在其中，归档后文件的命名方式为"simplerest\_ffmpeg\_player\_{日期}.zip"。

simplerest\_ffmpeg\_demos\_git\_archive\_all.sh内容如下所示。

[plain]  

```
1. echo "=====
2. echo "simplest ffmpeg demos list ---- git archive all"
3. echo "Lei Xiaohua"
4. echo "Communication University of China / Digital TV Technology"
5. echo "leixiaohua1020@126.com"
6. echo "http://blog.csdn.net/leixiaohua1020"
7. echo "=====
8.
9.
10.
11. #save in a folder named with date
12. #file suffix
13. #fsuffix="latest"
14. fsuffix=$(date +%y%m%d)
15. #folder to save files
16. archivefolder="simplest_ffmpeg_demos_list_${fsuffix}"
17. echo "Archive Folder : "${archivefolder}
18.
19. #if folder not exist, create it
20. if [ ! -d ${archivefolder} ]; then
21.     mkdir ${archivefolder}
22. fi
23.
24. #Git Archive
25. echo Archive simplest_ffmpeg_player...
26. cd simplest_ffmpeg_player
27. git archive -o "../${archivefolder}/simplest_ffmpeg_player_${fsuffix}.zip" HEAD
28. cd ..
29.
30.
31. echo Archive simplest_ffmpeg_audio_player...
32. cd simplest_ffmpeg_audio_player
33. git archive -o "../${archivefolder}/simplest_ffmpeg_audio_player_${fsuffix}.zip" HEAD
34. cd ..
35.
36. echo Archive simplest_ffmpeg_picture_encoder...
37. cd simplest_ffmpeg_picture_encoder
38. git archive -o "../${archivefolder}/simplest_ffmpeg_picture_encoder_${fsuffix}.zip" HEAD
39. cd ..
40.
41. echo Archive simplest_ffmpeg_video_encoder...
42. cd simplest_ffmpeg_video_encoder
43. git archive -o "../${archivefolder}/simplest_ffmpeg_video_encoder_${fsuffix}.zip" HEAD
44. cd ..
45.
46. echo Archive simplest_ffmpeg_audio_encoder...
47. cd simplest_ffmpeg_audio_encoder
48. git archive -o "../${archivefolder}/simplest_ffmpeg_audio_encoder_${fsuffix}.zip" HEAD
49. cd ..
50.
51. echo Archive simplest_ffmpeg_streamer...
52. cd simplest_ffmpeg_streamer
53. git archive -o "../${archivefolder}/simplest_ffmpeg_streamer_${fsuffix}.zip" HEAD
54. cd ..
55.
56. echo Archive simplest_ffmpeg_mem_handler...
57. cd simplest_ffmpeg_mem_handler
58. git archive -o "../${archivefolder}/simplest_ffmpeg_mem_handler_${fsuffix}.zip" HEAD
59. cd ..
60.
61. echo Archive simplest_ffmpeg_device...
62. cd simplest_ffmpeg_device
63. git archive -o "../${archivefolder}/simplest_ffmpeg_device_${fsuffix}.zip" HEAD
64. cd ..
65.
66. echo Archive simplest_ffmpeg_format...
67. cd simplest_ffmpeg_format
68. git archive -o "../${archivefolder}/simplest_ffmpeg_format_${fsuffix}.zip" HEAD
69. cd ..
70.
71. echo Archive simplest_ffmpeg_video_filter...
72. cd simplest_ffmpeg_video_filter
73. git archive -o "../${archivefolder}/simplest_ffmpeg_video_filter_${fsuffix}.zip" HEAD
74. cd ..
75.
76. echo Archive simplest_ffmpeg_swscale...
77. cd simplest_ffmpeg_swscale
78. git archive -o "../${archivefolder}/simplest_ffmpeg_swscale_${fsuffix}.zip" HEAD
79. cd ..
```

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:liushide@163.com