

流媒体高级编程

STREAMING MEDIA DAY05

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	屏幕流录制
	10:30 ~ 11:20	
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	混合流录制
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



屏幕流录制



需求分析



需求分析

- 录制屏幕流
 - 抓取视频显示器上的动态影像保存到本地或推送至远程
 - ✓ 本地录制: ScreenRecorder screen.flv
 - ✓ 推送直播: ScreenRecorder rtmp://192.168.1.166/live/1

知识讲解



概要设计



进程入口
main

录制
record

录制屏幕
recordScreen

打印源格式信息
printSrcFmt

打印目标格式信息
printDstFmt

打印解码包信息
printDecPkt

打印编码包信息
printEncPkt

打印视频帧信息
printVidFrm

打印FFmpeg错误
ffmpegError

事件处理
doEvent



编码实现



av_gettime

- 获取当前系统时间
 - #include <libavutil/time.h>
int64_t av_gettime (void);
 - 返回始自1970年1月1日0点0分0秒，直到函数被调用时的总微秒数，1微秒=10⁻⁶秒，即百万分之一秒



ScreenRecorder

【参见：FFmpeg/Primer/ScreenRecorder】

- 录制屏幕流
 - 抓取视频显示器上的动态影像保存到本地或推送至远程
 - ✓ 本地录制：ScreenRecorder screen.flv
 - ✓ 推送直播：ScreenRecorder rtmp://192.168.1.166/live/1



混合流录制



需求分析

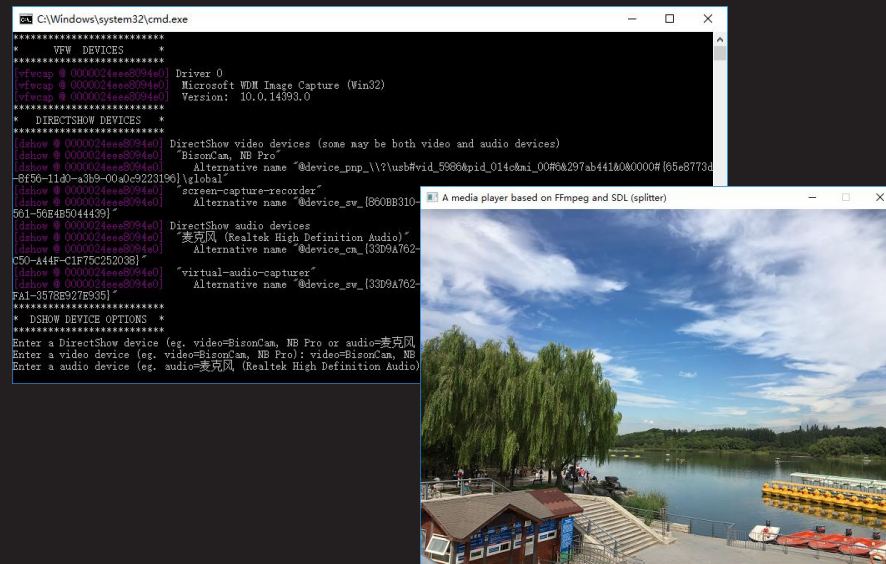


需求分析

- 录制音视频混合流

- 在Windows上列表显示所有的Video For Windows设备和DirectShow设备及其选项
- 通过用户选择的采集设备，捕获音视频数据，保存或推送
- 分别在独立的线程中，以并发的方式，采集音频和视频流，避免因捕获过程的相互等待，丢失帧数据
- 借助独立的混流线程，将捕获到的音视频帧，按解码时间戳的升序，依次写入目标格式中

知识讲解



概要设计





编码实现



av_compare_ts

- 比较时间戳

- #include <libavutil/mathematics.h>

```
int av_compare_ts (  
    int64_t      ts_a, // 时间戳a  
    AVRational  tb_a, // 时间戳a的单位  
    int64_t      ts_b, // 时间戳b  
    AVRational  tb_b); // 时间戳b的单位
```

- 若时间戳a早先于时间戳b, 则返回-1;
若时间戳a迟晚于时间戳b, 则返回1;
若时间戳a同时于时间戳b, 则返回0



AVRecorder

【参见：FFmpeg/Primer/AVRecorder】

- 录制音视频混合流
 - 在Windows上列表显示所有的Video For Windows设备和DirectShow设备及其选项
 - 通过用户选择的采集设备，捕获音视频数据，保存或推送
 - 分别在独立的线程中，以并发的方式，采集音频和视频流，避免因捕获过程的相互等待，丢失帧数据
 - 借助独立的混流线程，将捕获到的音视频帧，按解码时间戳的升序，依次写入目标格式中



附录



基于Nginx+RTMP的流媒体系统(下)

13. 在服务器上启动或重启nginx服务

- \$ sudo service nginx start
或
\$ sudo service nginx restart

14. 在服务器上更改防火墙，允许1935/tcp端口

- \$ sudo ufw allow 1935/tcp

15. 在服务器上用netstat检查端口侦听情况

- \$ netstat -ltn

```
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp          0      0 0.0.0.0:1935 0.0.0.0:* LISTEN
tcp          0      0 0.0.0.0:8080 0.0.0.0:* LISTEN
```



基于Nginx+RTMP的流媒体系统(下)

16. 在服务器上通过浏览器检查nginx服务状态

- <http://localhost:8080>
- <http://localhost:8080/stat>

The screenshot shows two browser windows. The top window, titled 'Welcome to nginx! - Mozilla Firefox', displays the 'Welcome to nginx!' page at localhost:8080. The bottom window, titled 'RTMP statistics - Mozilla Firefox', displays the 'RTMP statistics' page at localhost:8080/stat. The statistics page features a table with the following data:

RTMP	#clients	Video				Audio			In bytes	Out bytes	In bits/s	Out bits/s	State	Time
		codec	bits/s	size	fps	codec	bits/s	freq						
Accepted:	0								0 KB	0 KB	0 Kb/s	0 Kb/s		7m 51s
vod														
<i>vod streams</i>	0													
live														
<i>live streams</i>	0													
hls														
<i>live streams</i>	0													

Generated by [nginx-rtmp-module](#) 1.1.4, [nginx](#) 1.7.5, pid 1077, built Mar 17 2017 20:17:33 gcc 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.4)



基于Nginx+RTMP的流媒体系统(下)

17.在服务器上将视频点播文件(1.mp4 ...)拷贝到~/Videos目录下，保证任何用户对其可读

```
minwei@ubuntu: ~/Videos
minwei@ubuntu:~/Videos$ pwd
/home/minwei/Videos
minwei@ubuntu:~/Videos$ ls -l
总用量 3135604
-rw-r--r-- 1 minwei minwei 417170370 8月 12 2014 1.mp4
-rw-r--r-- 1 minwei minwei 297089682 8月 12 2014 2.mp4
-rw-r--r-- 1 minwei minwei 373161819 8月 12 2014 3.mp4
-rw-r--r-- 1 minwei minwei 361059118 8月 12 2014 4.mp4
-rw-r--r-- 1 minwei minwei 423277865 8月 12 2014 5.mp4
-rw-r--r-- 1 minwei minwei 960192458 8月 15 2014 6.mp4
-rw-r--r-- 1 minwei minwei 169194964 1月 24 21:20 7.mp4
-rw-r--r-- 1 minwei minwei 109132009 11月 7 2015 8.mp4
-rw-r--r-- 1 minwei minwei 100544991 1月 22 19:05 9.mp4
minwei@ubuntu:~/Videos$
```

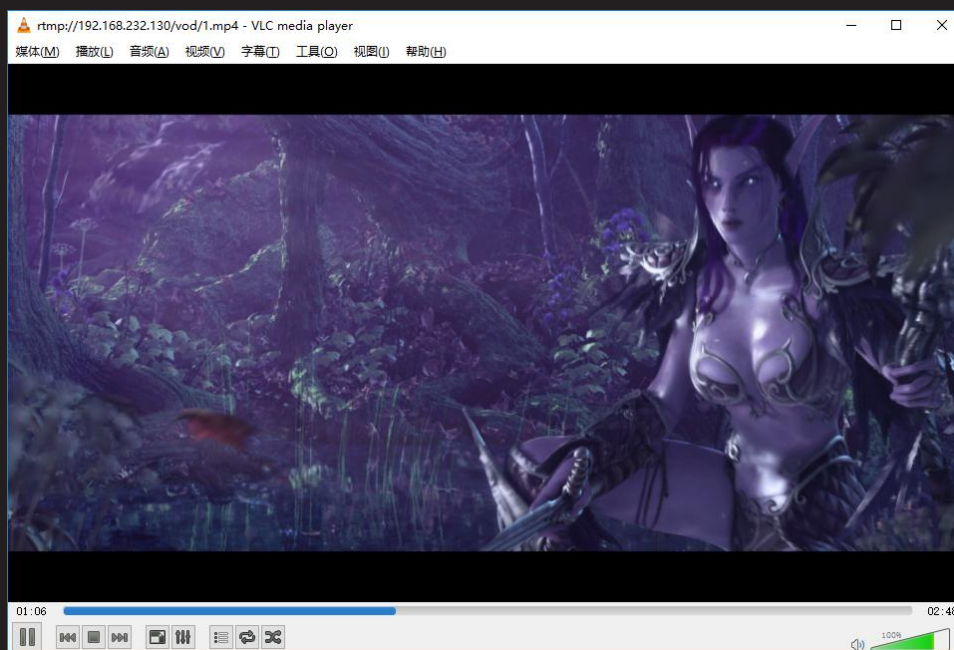
知识讲解



基于Nginx+RTMP的流媒体系统(下)

18. 测试点播媒体文件

- 播放端需要先从<http://www.videolan.org>下载并安装VLC media player
- 在播放端用VLC media player打开网络串流：
`rtmp://192.168.232.130/vod/1.mp4`



基于Nginx+RTMP的流媒体系统(下)

19. 测试RTMP直播媒体文件

- 推送端需要先从<http://ffmpeg.org>下载并安装FFmpeg
- 在推送端用FFmpeg推送媒体流：
`ffmpeg -re -i 1.mp4 -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 720x480 -q 10 rtmp://192.168.232.130/live/1`
- 在播放端用VLC media player打开网络串流：
`rtmp://192.168.232.130/live/1`



```
命令提示符 - ffmpeg -re -i 1.mp4 -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 720x480 -q 10 rtmp://192.168.232.130/live/1
[libx264 @ 000000002b40540] 264 - core 148 r2744 b97a00 - H.264/MPEG-4 AVC codec - Copyleft 2003-2010 - http://www.videolan.org/x264.html - options: cabac=0 ref=3 deblock=1:0:0 analyze=0:1:0:111 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed_ref=1 me_range=10 chroma_me=1 trellis=1 8x8dct=0 cqm=0 deadzone=21.11 fast_pskip=1 chroma_qp_offset=2 threads=2 lookahead_threads=2 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=0 weightp=0 keyint=250 keyint_min=24 scenecut=0 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=23.0 qcomp=0.00 qpmin=0 qmax=69 qstep=4 ip_ratio=1.40 aq=1:1.00
Output #0, flv, to 'rtmp://192.168.232.130/live/1':
  Metadata:
    major_brand      : mp42
    minor_version    : 0
    compatible_brands: mp42mp41
    encoder          : Lvc57.64.101
  Stream #0:0 (eng): Video: h264 (libx264) ([7][0][0][0] / 0x00007), yuv420p, 720x480 [SAR 32:27 DAR 16:9], q=-1--1, 24 fps, 1k tbn, 24 tbc (default)
  Metadata:
    creation_time   : 2013-07-11T18:21:22.000000Z
    handler_name    : MainConcept MP4 Video Media Handler
    encoder         : Lvc57.64.101 libx264
  Side data:
    cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
  Stream #0:1 (eng): Audio: aac (LC) ([10][0][0][0] / 0x0000A), 48000 Hz, stereo, fltp, 128 kb/s (default)
  Metadata:
    creation_time   : 2013-07-11T18:21:22.000000Z
    handler_name    : MainConcept MP4 Sound Media Handler
    encoder         : Lvc57.64.101 aac
  Stream mapping:
    Stream #0:0 -> #0:0 (native) -> h264 (libx264)
    Stream #0:1 -> #0:1 (aac (native)) -> aac (native)
Press [q] to stop, [?] for help
frame= 307 fps= 23 q=26.0 size= 1815KB time=00:00:13.63 bitrate=1090.5kbits/s speed= 1x
```



基于Nginx+RTMP的流媒体系统(下)

20.测试HLS直播媒体文件

- 在推送端用FFmpeg推送媒体流：
ffmpeg -re -i 1.mp4 -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 720x480 -q 10 rtmp://192.168.232.130/hls/1
- 在播放端用VLC media player打开网络串流：
rtmp://192.168.232.130/hls/1



```
命令提示符 - ffmpeg -re -i 1.mp4 -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 720x480 -q 10 rtmp://192.168.232.130/hls/1
[libx264 @ 000000002b0f0540] 264 - core 148 r2744 b97a00 - H.264/MPEG-4 AVC codec - Copyleft 2003-2010 - http://www.videolan.org/x264.html - options: cabac=0 ref=3 deblock=1:0:0 analyze=0x1:0x111 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed
_ref=1 me_range=10 chroma_me=1 trellis=1 8x8dct=0 cqm=0 deadzone=21.11 fast_pskip=1 chroma_qp_offset=2 threads=12 lookahead_threads=2 sliced_threads=0 nr=0 decimate=1 interlace=0 bluray_compat=0 constrained_intra=0 bframes=0 weightp=0 keyint=29 keyint_min=24 scenecut=0 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=23.0 qcomp=0.00 qpmin=0 qpnar=0 qpstep=4 ip_ratio=1.40 aq=1:1.00
Output #0, flv, to 'rtmp://192.168.232.130/hls/1':
  Metadata:
    major_brand      : mp42
    minor_version    : 0
    compatible_brands: mp42mp41
    encoder          : Lavc57.64.100
  Stream #0:0 (eng): Video: h264 (libx264) ([7][0][0][0] / 0x00007), yuv420p, 720x480 [SAR 32:27 DAR 16:9], q=-1--1, 24
  fps, 1k tbn, 24 tbc (default)
  Metadata:
    creation_time   : 2013-07-11T18:21:22.000000Z
    handler_name    : MainConcept MP4 Video Media Handler
    encoder        : Lavc57.64.101 libx264
  Side data:
    cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
  Stream #0:1 (eng): Audio: aac (LC) ([0][0][0][0] / 0x0000A), 48000 Hz, stereo, fltp, 128 kb/s (default)
  Metadata:
    creation_time   : 2013-07-11T18:21:22.000000Z
    handler_name    : MainConcept MP4 Sound Media Handler
    encoder        : Lavc57.64.101 aac
  Stream mapping:
    Stream #0:0 -> #0:0 (native) -> h264 (libx264)
    Stream #0:1 -> #0:1 (aac (native)) -> aac (native)
  Press [q] to stop, [?] for help
frame= 1020 fps= 24 q=26.0 size= 6599KB time=00:00:43.30 bitrate=1242.7kbits/s speed=0.99x
```



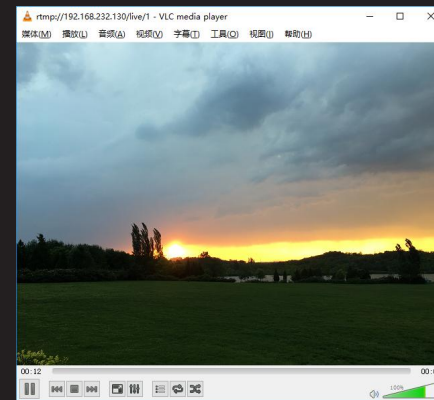
基于Nginx+RTMP的流媒体系统(下)

21.测试RTMP直播摄像头和麦克风

- 在推送端用FFmpeg推送媒体流：
`ffmpeg -f dshow -i video="BisonCam, NB Pro" -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -pix_fmt yuv420p -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 640x480 -q 10 rtmp://192.168.232.130/live/1`
- 在播放端用VLC media player打开网络串流：
`rtmp://192.168.232.130/live/1`

知识讲解

```
命令提示符 - ffmpeg -f dshow -i video="BisonCam, NB Pro" -f dshow -i audio="麦克风 (Realtek High Definition Audi...
Last message repeated 7 times
[libx264 @ 0000000080801e0] -gcache is ignored, -crf is recommended.
[libx264 @ 0000000080801e0] real-time buffer (BisonCam, NB Pro) [video input] too full or near too full (101% of size: 30416)
[libx264 @ 0000000080801e0] [real-time parameter] frame dropped
[libx264 @ 0000000080801e0] using cpu capabilities: MMX2 SSE2Fast SSE3 SSE4.2 AVX FMA3 AVX2 LZCNT BMI2
[libx264 @ 0000000080801e0] profile Constrained Baseline, level 3.0
[libx264 @ 0000000080801e0] 264 - core 146 r2744 597a00 - H.264/MP2T-1 AVC codec - Copyleft 2003-2016 - http://www.videolan.org/vlc.html - options: cabac=0 ref=3 deblock=1:0:0 analysis=0:0:0 filter=0:0:0 psy-rd=1.00:0.00 mixed_rd=1:0:0 me_range=0 chroma_smi=1 trellis=1 8080:0:0 qp=0 decoder=0:1:1 fast-pskip=1 stream-cp-strict=2:1 thread=12 look_ahead=2 sliced_thread=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=0 weightp=0 keyint=250 keyint_min=25 gop_size=0:0:0 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=26.0 qcomp=0.60 qnarm=0.90 pteps=4 qp_rate=1.40 aq=1.1.0
Output #0, flv, to 'rtmp://192.168.232.130/live/1':
  Metadata:
    encoder         : Lavf57.56.100
  Stream #0:0: Video: h264 (libx264) ([7F][0][0] / 0x0007), yuv420p, 640x480, q=1-1, 30 fps, 1k tbn, 30 tbc
  Metadata:
    encoder         : Lavf57.54.101 libx264
  Side data:
    cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
  Stream #0:1: Audio: aac (LC) ([10][0][0] / 0x000A), 48000 Hz, stereo, fltp, 128 kb/s
  Metadata:
    encoder         : real-time buffer (BisonCam, NB Pro) [video input] too full or near too full (101% of size: 30416)
    [real-time parameter] frame dropped
    Lavf57.54.101 aac
Stream mapping:
  Stream #0:0 -> #0:0 (rawvideo (native) -> h264 (libx264))
  Stream #0:1 -> #0:1 (pcm_s16le (native) -> aac (native))
Press [q] to stop, [?] for help
frame= 82 fps= 32 q=28.0 size=    247kB time=00:00:01.95 bitrate=1036.4kbits/s speed=0.77x
```

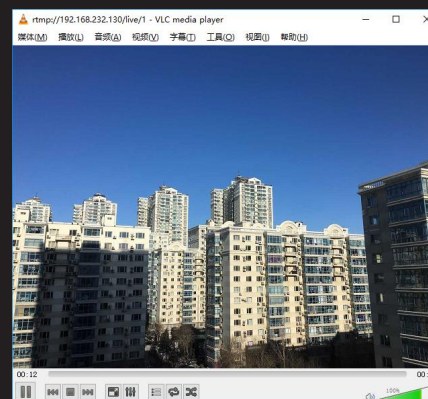


基于Nginx+RTMP的流媒体系统(下)

22.测试HLS直播摄像头和麦克风

- 在推送端用FFmpeg推送媒体流：
ffmpeg -f dshow -i video="BisonCam, NB Pro" -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -pix_fmt yuv420p -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 640x480 -q 10 rtmp://192.168.232.130/hls/1
- 在播放端用VLC media player打开网络串流：
rtmp://192.168.232.130/hls/1

```
命令提示符 - ffmpeg -f dshow -i video="BisonCam, NB Pro" -f dshow -i audio="麦克风 (Realtek High Definition Audio)...
[libx264 @ 00000000800d40] last message repeated 7 times
[libx264 @ 00000000800d40] scale is ignored, -crf is recommended.
[libx264 @ 00000000800d40] using cpu capabilities: MMX2 SSE2Fast SSE4.2 AVX FMA3 AVX2 LZCNT BMI2
[libx264 @ 00000000800d40] profile Constrained Baseline, Level 3.0
[libx264 @ 00000000800d40] 264 - core 146 2744 37480 - H.264/MPEG-4 AVC codec - Copyleft 2003-2016 - http://www.videolan.org/x264.html - options: cabac=0 refs=3 deblock=1:0:0 analyze=0:1:0:111 me=hex subme=7 psyrd=1:0:0:0:0 mixed
[libx264 @ 00000000800d40] Head threads=2 sliced threads=0 m=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=0 weightp=0 key
[libx264 @ 00000000800d40] Intra refresh=0 rc_lookahead=40 rct=0 abt=1 crf=23.0 qcomp=1.00 gain=0 qnax=0.9
[libx264 @ 00000000800d40] Output #0: flv, to 'rtmp://192.168.232.130/hls/1':
Metadata:
  encoder       : Lavf57.50.100
  Stream #0:0: Video: h264 (libx264) ([7F][0][0][0] / 0x0007), yuv420p, 640x480, q=1-1, 30 fps, 1k tbn, 30 tbc
Metadata:
  encoder       : Lavf57.64.101 libx264
Side data:
  cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
  Stream #0:1: Audio: aac (LC) ([10][0][0][0] / 0x0000), 48000 Hz, stereo, flp, 128 kb/s
Metadata:
  encoder       : Lavf57.64.101 aac
Stream mapping:
  Stream #0:0 -> #0:0 (rawvideo (native) -> h264 (libx264))
  Stream #0:1 -> #0:1 (pcm_s16le (native) -> aac (native))
Press [q] to stop, [?] for help
frame= 112 fps= 32 q=28.0 size=    368kB time=00:00:02.95 bitrate=1019.7kbits/s speed=0.83x
```



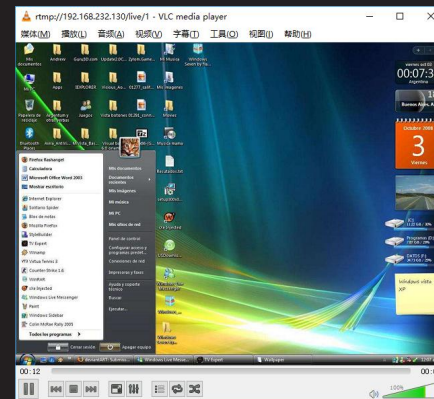
基于Nginx+RTMP的流媒体系统(下)

23.测试RTMP直播屏幕和麦克风

- 在推送端用FFmpeg推送媒体流：
`ffmpeg -f gdigrab -i desktop -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -pix_fmt yuv420p -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 640x480 -q 10 rtmp://192.168.232.130/live/1`
- 在播放端用VLC media player打开网络串流：
`rtmp://192.168.232.130/live/1`

知识讲解

```
命令提示符 - ffmpeg -f gdigrab -i desktop -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -pix_fmt yuv420p
Stream #0:0: Video: h264, 1920x1080, 1588880 kb/s, 29.97 fps, 1000k tbn, 1000k tbr, 1000k tbc
Guessed Channel Layout for Input Stream #1.0 : stereo
Input #1, dshow, from audio="麦克风 (Realtek High Definition Audio)":
Duration: 00:00:01.0000000, bitrate: 1411 kb/s
Stream #1:0: Audio: pcm_s16le, 44100 Hz, stereo, s16, 1411 kb/s
[libx264 @ 0000000000000000] qscale is ignored, use is recommended.
[libx264 @ 000000000000000000] using cpu capabilities: MMX2 SSE2Fast SSE3 SSE4.2 AVX FMA3 AVX2 LZCNT BMI2
[libx264 @ 000000000000000000] profile Constrained Baseline, level 3.0
[libx264 @ 000000000000000000] 294 - core 146 6744 bfratio = 1.204/MBSP=4 AVC codec - Copyleft 2003-2016 - http://www.videolan.org/x264.html - options: cabac=0 ref=3 deblock=1:0:0 analyze=0:0:111 me=hex subme=7 psyrd=1.00:0.00 mixed_rnd=0 me_range=0 chroma_sse0 trillis0 8bd8d0 qps=0 deadzone0 11 start_dsp1 chroma_qp_offset=2 threads=12 lookahead_threads=2 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bitrate=0 weightp=0 keyint=200 keyint_min=20 scenecut=0 intra_refresh=0 rc_lookahead=40 rc_lookahead=40 rc_qlt=20.0 qcomp=1.00 gain=0 qmax=69 qmin=0
Output #0, flv, to rtmp://192.168.232.130/live/1:
Metadata:
encoder      : Lavf57.56.100
Stream #0:0: Video: h264 (libx264) ([7F][01][01] / 0x00007) (yuv420p, 640x480, q=1-1, 29.97 fps, 1k tbn, 29.97 the
Metadata:
encoder      : Lavf57.64.101 libx264
Side data:
cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
Stream #0:1: Audio: aac (LC) ([10][01][01] / 0x0000A) (48000 Hz, stereo, flp, 128 kb/s
Metadata:
encoder      : Lavf57.64.101 aac
Stream mappings:
Stream #0:0 -> #0:0 (bmp (native) -> h264 (libx264))
Stream #1:0 -> #0:1 (pcm_s16le (native) -> aac (native))
Press [q] to stop, [?] for help
frame= 174 fps= 20 q=28.0 size= 49558 time=00:00:07.00 bitrate= 578.4kbits/s speed=0.73x
```

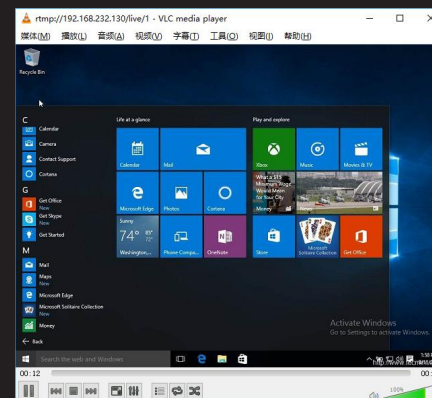


基于Nginx+RTMP的流媒体系统(下)

24.测试HLS直播屏幕和麦克风

- 在推送端用FFmpeg推送媒体流：
ffmpeg -f gdigrab -i desktop -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -pix_fmt yuv420p -vcodec libx264 -vprofile baseline -acodec aac -ar 48000 -strict -2 -ac 2 -f flv -s 640x480 -q 10 rtmp://192.168.232.130/hls/1
- 在播放端用VLC media player打开网络串流：
rtmp://192.168.232.130/hls/1

```
命令提示符 - ffmpeg -f gdigrab -i desktop -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -pix_fmt yuv420p
Stream #0:0: Video: h264, 1920x1080, 1588880 kb/s, 29.97 fps, 1000k tbn, 1000k tbc, 1000k tbc
Guessed Channel Layout for Input Stream #1.0 : stereo
Input #1, dshow, from audio="麦克风 (Realtek High Definition Audio)":
  Duration: 0:00:01.0000000, bitrate: 1411 kb/s
  Stream #1:0: Audio: pcm_s16le, 44100 Hz, stereo, s16, 1411 kb/s
[libx264 @ 0x0000000000000000] qscale is ignored, use is recommended.
[libx264 @ 0x0000000000000000] using cpu capabilities: MMX2 SSE2Fast SSE4.2 AVX FMA3 AVX2 LZCNT BMI2
[libx264 @ 0x0000000000000000] profile Constrained Baseline, level 3.0
[libx264 @ 0x0000000000000000] 294 - core 146 r174 h19aw0 - h.264/MPEG-4 AVC codec - Copyleft 2003-2016 - http://www.vivid
wolan.org/x264.html - options: cabac=0 ref=3 deblock=1:0:0 analyse=0:1:0:111 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed
psy=4 me_range=0 chroma_sse0 trillis=0 sad=0 q=0 deadzone=211 start_quant=1 chroma_qp_offset=2 threads=12 look
head_thread=2 sliced_threads=0 nr=0 decimate=1 interlace=0 bluray_compat=0 constrained_intra=0 bitrate=0 weightp=0 key
int=20 keyint_min=20 scenecut=0 intra_refresh=0 rc_lookahead=40 rc_1pass=0 rc_strategy=0 qcomp=0.00 qgain=0 qmax=09 q
min=0.00 qp_max=0 qp_min=0 vbr_buffer=0 vbr_maxrate=0 vbr_minrate=0 vbr_rnd=0 vbr_scale=1 vbr_thresh=1.00 vbr_thresh2=6.4
vbr_thresh3=7.4 vbr_thresh4=1.00
Output #0, flv, to 'rtmp://192.168.232.130/live/1':
  Metadata:
    encoder         : Lavf57.56.100
  Stream #0:0: Video: h264 (libx264) ([7F][0][0][0] / 0x00007), yuv420p, 640x480, q=1-1, 29.97 fps, 1k tbn, 29.97 the
  Metadata:
    encoder         : Lavf57.64.101 libx264
  Side data:
    cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbr_delay: -1
  Stream #0:1: Audio: aac (LC) ([10][0][0][0] / 0x0000A), 48000 Hz, stereo, fltp, 128 kb/s
  Metadata:
    encoder         : Lavf57.64.101 aac
Stream mappings:
  Stream #0:0 -> #0:0 (h264 (native) -> h264 (libx264))
  Stream #1:0 -> #0:1 (pcm_s16le (native) -> aac (native))
Press [q] to stop, [?] for help
frame= 174 fps= 20 q=28.0 size=    4955b time=00:00:07.00 bitrate= 578.4kbits/s speed=0.73x
```



知识讲解



总结和答疑

