

FFMPEG 读取 rtsp 源视频

蒋亮

2019-9-20

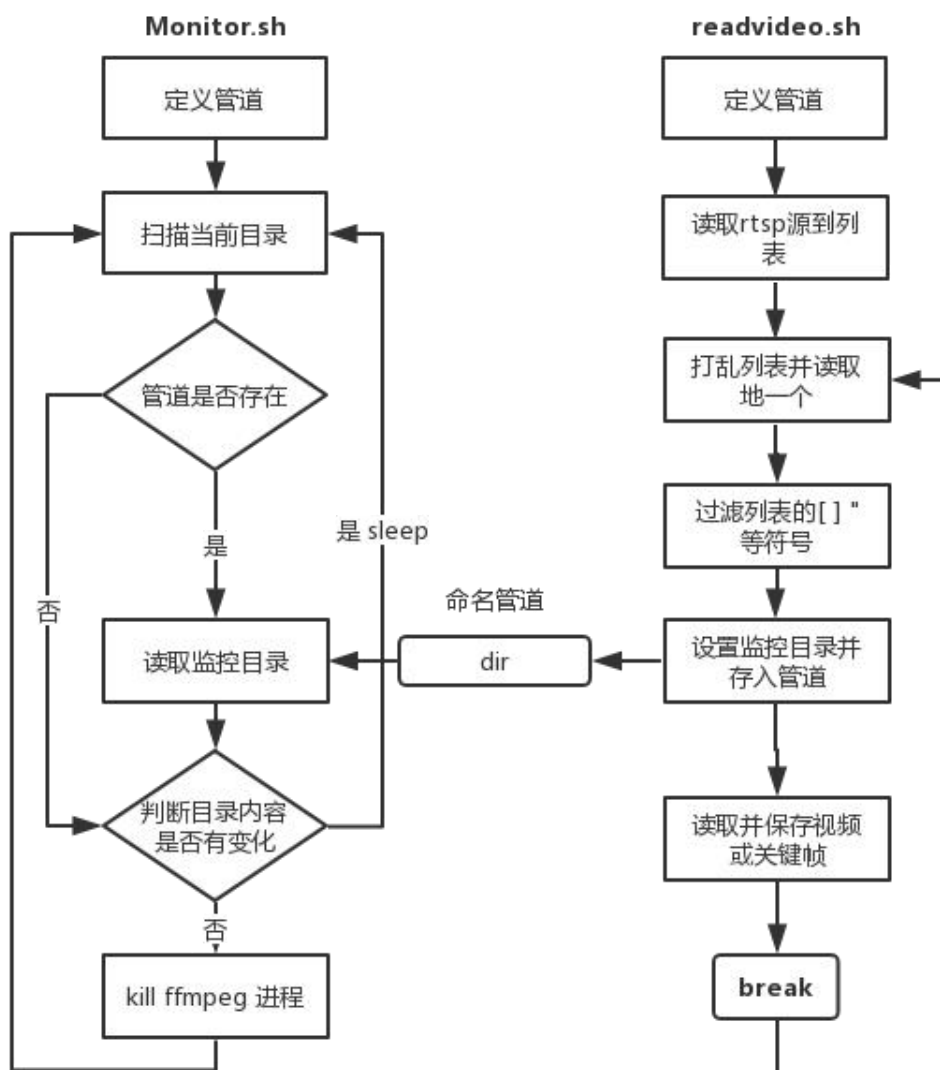
背景

从一个 json 文件中读取 rtsp 视频源，以关键帧或视频的形式保存下来

大纲流程图

Monitor.sh: 作为一个监控进程，防止 ffmpeg 读取视频过程卡在中间过程，导致进程一直处于阻塞状态，这种状态一般会出现现在 ffmpeg 读取到一半，网络断开。

Readvideo.sh: 作为读取视频流并保存关键帧或视频的进程，反复读取 rtsp 的 json 列表，并随机打乱顺序，每次只读取第一条视频源里面的视频流。



流程详解：

Readvideo.sh:

1. **cat** 读取 **rtsp.json** 文件根据 **keys** 提取并输出重定向到 **tmp** 文件
2. **awk** 命令 对 **keys** 打乱顺序
3. 读取地一个其中一个 **key**，去掉引号，并过滤掉 '['或']'
4. 根据 **key** 获取 **rtsp**
5. 定义文件保存路径 **./年月日小时/rtsp** 的 **rm5** 字符串/分秒.**mp4(jpg)**
6. 保存视频/图片
7. **Break**
8. 重复步骤 3
9. 重复步骤 2

备注：

关于 **break** 的解释 :**break** 的作用是保证内层循环只读取打乱后的 **rtsp** 的第一个 ,如果去掉 **break** ,则是每次打乱 **rtsp** 之后都将所有的 **rtsp** 读取完

Monitor.sh:

1. 检测命名管道文件是否存在，若存在则读取里面的内容，读取之后删除管道文件(此处采用命名管道的非阻塞模式，但是每次读取管道内容之后不会删除内容，导致后面)
2. 扫描目标文件夹的大小 **size** 是否变化
3. 如果没有变化，找到对应的 **ffmpeg** 进程，并 **kill**
4. 将新的文件夹 **size** 值赋值给旧的文件夹 **size** 值、
5. **Sleep** 并重复步骤 1

参数详解：

Readvideo.sh

❖ **#define the file path**

```
Dir=$(date '+%y%m%d%H')
```

```
ms=$(date '+%M%S')
```

#use md5 map the rtsp to a unique string

```
rtspmd5=$(echo $prot | md5sum |cut -d" " -f1)
```

```
Dir="$Dir/$rtspmd5/$ms"
```

定义命名规则，根据 **shell** 获取系统时间，如当前时间为 **2019.9.22** ， 下午 **15.30.22**,此时获取到的 **rtsp** 为:"**rtsp://admin:data2010@192.168.3.201:554/h264/ch33/main/av_stream**" 为了让同一

个 rtsp 源的视频或关键帧保存至同一个文件夹，则需要唯一标识符，于是采用 md5 将 rtsp 映射为唯一的 32 位字符串

19092315/9be0c8efe7e7ed6a911e1b44dcaa6025/3022/ 递归建立这个文件夹，并通过管道通知 Monitor.sh 进程监控该文件夹状况

❖ `ffmpeg -stimeout 5000000 -i $prot -vf select='eq(pict_type\,I)' -vframes 120 "$Dir/video.mp4"`

-stimeout 5000000: 设置链接超时选项 5s,这个选项对 rtsp 协议有用,对 rtmp 需要使用 -timeout 5000000 选项

-i \$port: -i 表示 input file/video stream

-vf: 表示滤镜, select='eq(pict_type\,I)' 表示关键帧选取, I 这个参数则表示关键帧

-vframes 120 表示保存 120 帧的视频,也可以替换成 -t 100 表示保存 100 秒的视频

\$Dir/video.mp4: 19092315/9be0c8efe7e7ed6a911e1b44dcaa6025/3022/video.mp4

❖ `ffmpeg -stimeout 5000000 -i $prot -vf select='eq(pict_type\,I)*not(mod(n\,2))' -vsync 0 -vframes 120 "$Dir/img_%5d.jpg"`

-vf select='eq(pict_type\,I)*not(mod(n\,2))': 表示每隔两帧保存一帧关键帧

-vsync 0 不增加或减少 Frame, 减少粘滞帧

Img_%5d.jpg: 以 img_00001.jpg img_00002.jpg 方式快速命名保存一系列图片

Monitor.sh

❖ `filesizenew=$(ls -l "$Dirnew/" | awk '{ print $5 }')`

ls -l 获取文件夹下所有的文件的详细信息,其中包含各个文件的详细大小信息