Technology Tarena 达内科技

流媒体高级编程

STREAMING MEDIA DAY04

内容

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

屏幕流捕获


屏幕流捕获
需求分析
概要设计
编码实现
需求分析
概要设计
avdevice_register_all
av_find_input_format

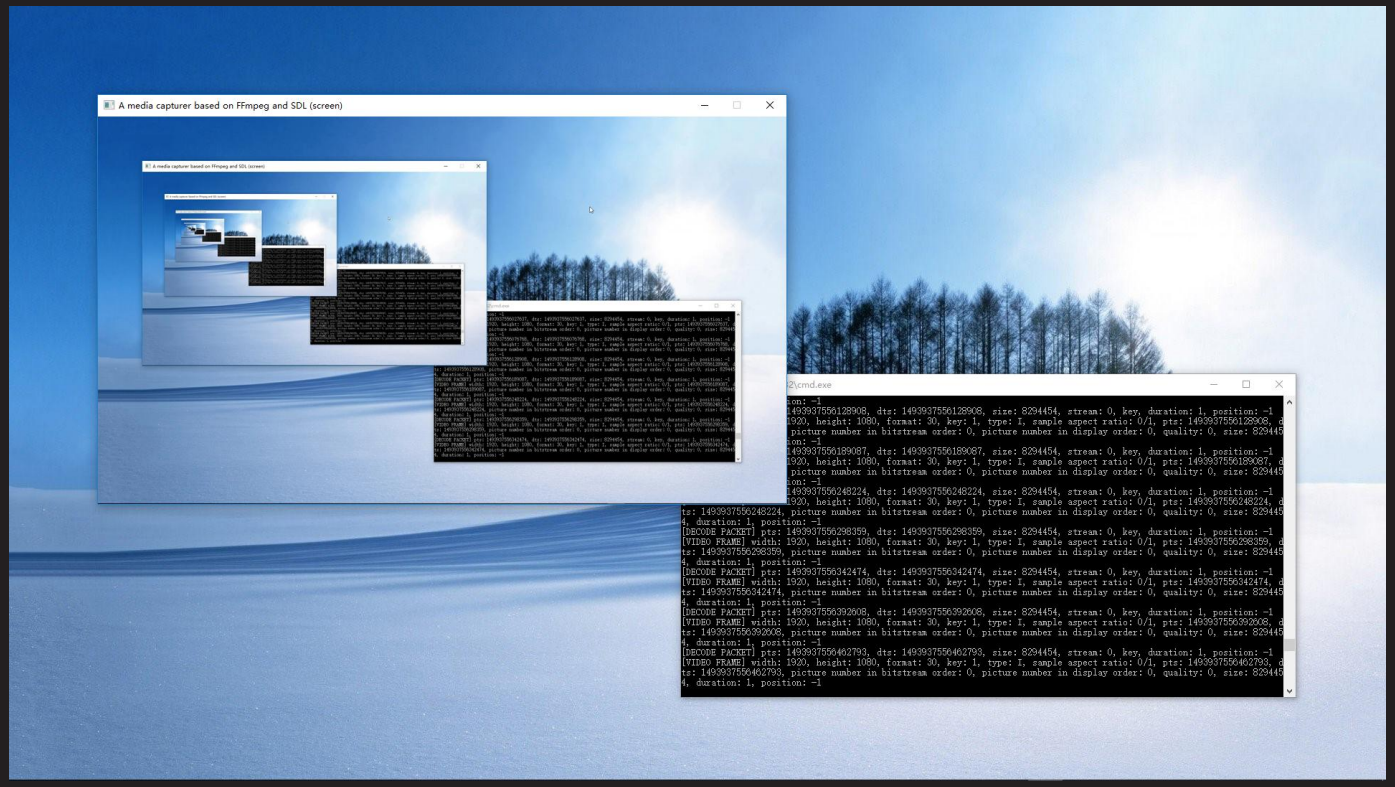
Technology Tarena 达内科技

需求分析

需求分析

- 捕获屏幕流
  - 抓取视频显示器上的动态影像，实时本地回放

知识讲解



Technology Tarena 达内科技

概要设计

概要设计

知识讲解

进程入口
main

捕获
capture

捕获屏幕
captureScreen

打印源格式信息
printSrcFmt

打印解码包信息
printDecPkt

打印视频帧信息
printVidFrm

打印FFmpeg错误
ffmpegError

打印SDL错误
sdlError

事件处理
doEvent

Technology Tarena 达内科技

编码实现

avdevice_register_all

Technology Tarena 达内科技

- 初始化libavdevice库并注册所有输入和输出设备
  - #include <libavdevice/avdevice.h>
    void avdevice_register_all (void);
  - 但凡涉及到输入输出设备的流媒体访问，无论该设备是物理的，比如摄像头、麦克风等，还是逻辑的，比如桌面窗口等，都需要libavdevice库的支持，必须先调用此函数
  - 用户代码通常无需直接访问libavdevice库，但用户代码所使用的某些FFmpeg对象，比如AVFormatContext类型的源格式上下文，却可能会借助该库从特定设备中采集数据
  - 此函数不是线程安全的，在多线程中使用需要互斥保护

知识讲解

av_find_input_format

- 根据输入格式名称获取输入格式对象
  - `#include <libavformat/avformat.h>`

    `AVInputFormat* av_find_input_format (`

    `const char* short_name); //` 输入格式名称
  - 成功返回指向输入格式对象的指针，失败返回NULL
  - short_name参数可取如下值：
    - ✓ `gdigrab` - Windows屏幕
    - ✓ `vfwcap` - Windows音视频采集
    - ✓ `dshow` - Windows音视频采集
    - ✓ `x11grab` - Linux屏幕
    - ✓ `video4linux2` - Linux视频采集
    - ✓ `alsa` - Linux音频采集
    - ✓ …

知识讲解

av_find_input_format(续1)

- gdigrab/x11grab抓屏格式的特殊选项
  - 可以在调用avformat_open_input函数时，通过其AVDictionary类型的第四个参数，为其AVInputFormat类型的第三个参数，即输入格式，指定如下特殊选项：
    - ✓ `framerate` - 抓屏帧率
    - ✓ `video_size` - 抓屏矩形宽高
    - ✓ `offset_x` - 抓屏矩形左上角水平坐标
    - ✓ `offset_y` - 抓屏矩形左上角垂直坐标

知识讲解

av_find_input_format(续2)

Tarena Technology 达内科技

- 流媒体源统一资源定位符
  - 调用avformat_open_input函数时，通过其const char*类型的第二个参数，即源统一资源定位符，显式指明流媒体源的位置，例如：
    - ✓ `mayitbe.vob` // 本地文件
    - ✓ `rtmp://192.168.1.166/vod/1.mp4` // 远程文件
    - ✓ `rtmp://192.168.1.166/live/1` // 远程应用
    - ✓ `desktop` // 整个屏幕
    - ✓ `title=Media Player` // 特定窗口
    - ✓ `video=BisonCam, NB Pro` // 视频采集(Windows)
    - ✓ `audio=麦克风 (Realtek ...)` // 音频采集(Windows)
    - ✓ `/dev/video0` // 视频采集(Linux)
    - ✓ `hw:0` // 音频采集(Linux)

知识讲解

ScreenCapturer

【参见：FFmpeg/Primer/ScreenCapturer】

课堂练习

- 捕获屏幕流
  - 抓取视频显示器上的动态影像，实时本地回放

混合流捕获


混合流捕获
需求分析
概要设计
编码实现
需求分析
概要设计
MultiByteToWideChar
WideCharToMultiByte
GetConsoleOutputCP
SetConsoleOutputCP

Technology Tarena 达内科技

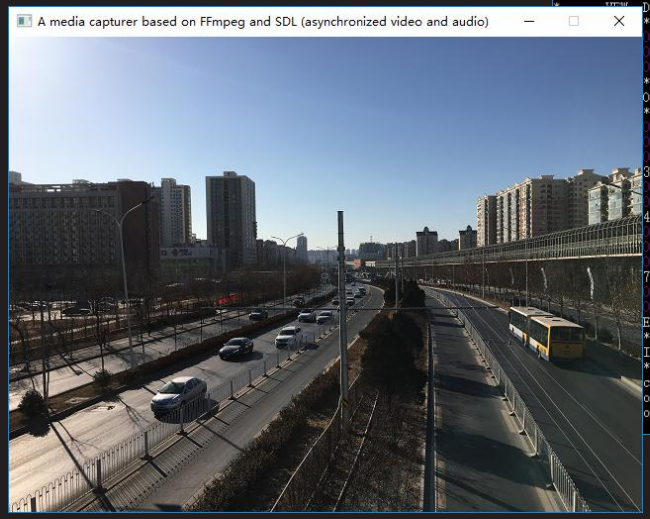
需求分析

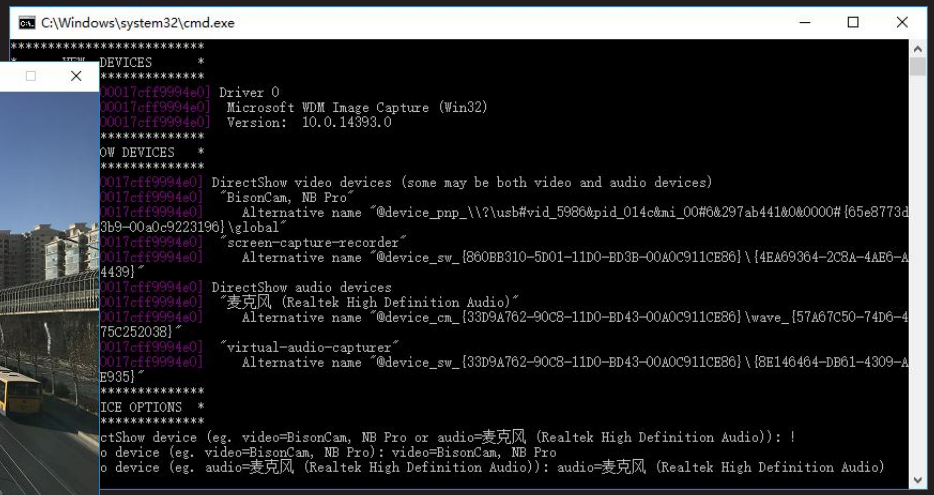
需求分析

Tarena Technology 达内科技

- 捕获音视频混合流
  - 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，在本地回放
  - 分别在独立的线程中，以并发的方式，采集、播放音频和视频流，避免因捕获过程的相互等待，而导致帧数据丢失

知识讲解





Technology Tarena 达内科技

概要设计

概要设计

知识讲解

Technology Tarena 达内科技

编码实现

MultiByteToWideChar

- 将多字节字符串转换为宽字符字符串
  -
```
#include <winnls.h>
int MultiByteToWideChar (
  UINT    CodePage,          // 多字节字符串代码页标识
  DWORD   dwFlags,           // 标志位，0表示缺省
  LPCSTR lpMultiByteStr,     // 多字节字符串
  int     cbMultiByte,       // 多字节字符串的字节数
                             // (不含终止空字符),
                             // -1表示由函数自动计算
  LPWSTR lpWideCharStr,      // 宽字符字符串缓冲区
  int     cchWideChar);      // 宽字符字符串缓冲区的最
                             // 大字符数(含终止空字符)
```
  - 成功返回转换后宽字符字符串的字符数(不含终止空字符)，若cchWideChar参数取0，则返回需要为宽字符字符串缓冲区分配的字符数(含终止空字符)，失败返回0

知识讲解

WideCharToMultiByte

Tarena Technology 达内科技

知识讲解

- 将宽字符字符串转换为多字节字符串
  -
```
#include <winnls.h>
int WideCharToMultiByte (
  UINT     CodePage,            // 多字节字符串代码页标识
  DWORD    dwFlags,             // 标志位，0表示缺省
  LPCWSTR  lpWideCharStr,       // 宽字符字符串
  int      cchWideChar,         // 宽字符字符串的字符数
                                // (不含终止空字符)
                                // -1表示由函数自动计算
  LPSTR    lpMultiByteStr,      // 多字节字符串缓冲区
  int      cbMultiByte,         // 多字节字符串缓冲区的最
                                // 大字节数(含终止空字符)
  LPCSTR   lpDefaultChar,       // 缺省多字节字符
  LPBOOL   lpUsedDefaultChar); // 是否使用(了)缺省字符
```
  - 成功返回转换后多字节字符串的字节数(不含终止空字符)，若cbMultiByte参数取0，则返回需要为多字节字符串缓冲区分配的字节数(含终止空字符)，失败返回0。lpDefaultChar和lpUsedDefaultChar参数取空指针，函数执行速度更快

WideCharToMultiByte(续1)

- 字符编码转换
  - 在Windows操作系统上缺省使用多字节字符串，比如中文版Windows使用936代码页，即GBK编码的多字节字符串
  - FFmpeg使用的字符串同样是多字节字符串，但并非GBK编码，而是UTF-8编码，因此需要在两者之间进行转换，特别是在字符串中包含非ASCII字符时，比如汉字和特殊符号，因为GBK和UTF-8两种编码的ASCII部分是完全一致的
  - 转换的方法是先将一种多字节字符串(GBK或UTF-8)转换为宽字符字符串(UCS-2)，再将宽字符字符串(UCS-2)转换为另一种多字节字符串(UTF-8或GBK)
  - 这两次转换可以借助于前述函数MultiByteToWideChar和WideCharToMultiByte来完成，注意两次所用代码页不同

知识讲解

WideCharToMultiByte(续2)

- 字符编码转换
  - 例如：将GBK编码的字符串转换为UTF-8编码

```
string gbk2utf8 (string const& gbk) {
  int len = MultiByteToWideChar (936, 0, gbk.c_str (), -1,
    NULL, 0);
  wchar_t* wcs = new wchar_t[len];
  MultiByteToWideChar (936, 0, gbk.c_str (), -1,
    wcs, len);
  len = WideCharToMultiByte (CP_UTF8, 0, wcs, -1,
    NULL, 0, NULL, NULL);
  char* mbs = new char[len];
  WideCharToMultiByte (CP_UTF8, 0, wcs, -1,
    mbs, len, NULL, NULL);
  string utf8 (mbs);
  delete[] mbs;
  delete[] wcs;
  return utf8;
}
```

知识讲解

GetConsoleOutputCP

- 获取控制台输出代码页
  - `#include <wincon.h>`

    `UINT GetConsoleOutputCP (void);`
  - 返回当前控制台输出字符时所使用的代码页标识
  - Windows操作系统所定义的代码页标识：
    - ✓ `437` – 美国英语，ASCII，1字节/字符
    - ✓ `936` – 简体中文，GBK/GB2312，2字节/字符
    - ✓ `950` – 繁体中文，BIG5，2字节/字符
    - ✓ `54936` – 繁简中文，GB18030，4字节/字符
    - ✓ `65001` – 8位统一字符编码转换格式，UTF-8，1-4字节/字符
    - ✓ `...`

知识讲解

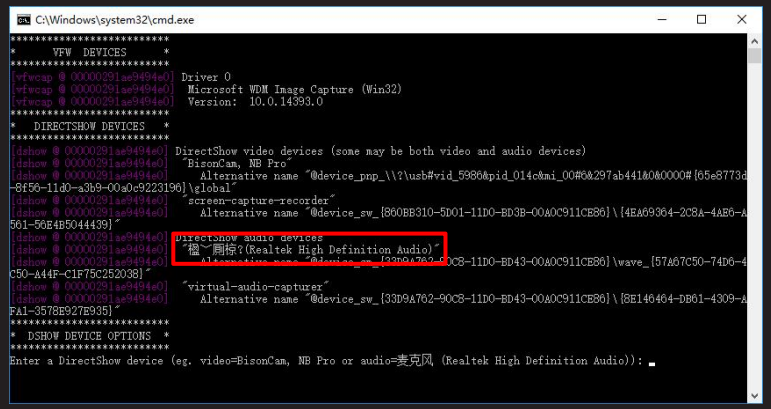
SetConsoleOutputCP

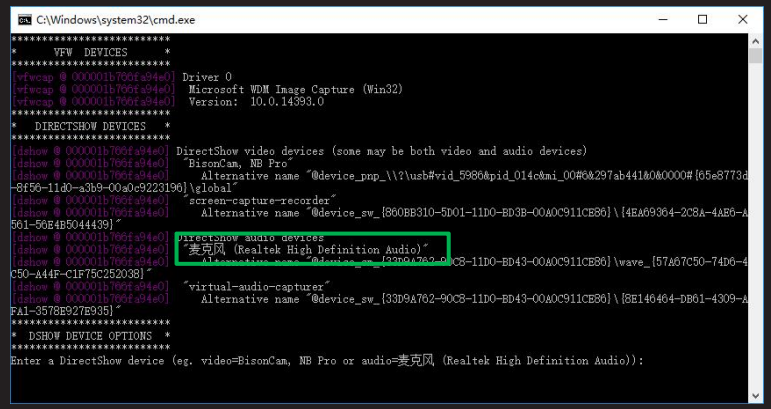
知识讲解

- 设置控制台输出代码页
  - `#include <wincon.h>`

    `BOOL SetConsoleOutputCP (`

    `UINT wCodePageID); // 代码页标识`
  - 成功返回TRUE，失败返回FALSE
  - 通过控制台输出字符所使用的字符编码，必须与控制台所使用的输出代码页保持一致，否则某些字符将显示为乱码





AsyncCapturer

【参见：FFmpeg/Primer/AsyncCapturer】

课堂练习

- 捕获音视频混合流
  - 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，在本地回放
  - 分别在独立的线程中，以并发的方式，采集、播放音频和视频流，避免因捕获过程的相互等待，而导致部分帧丢失

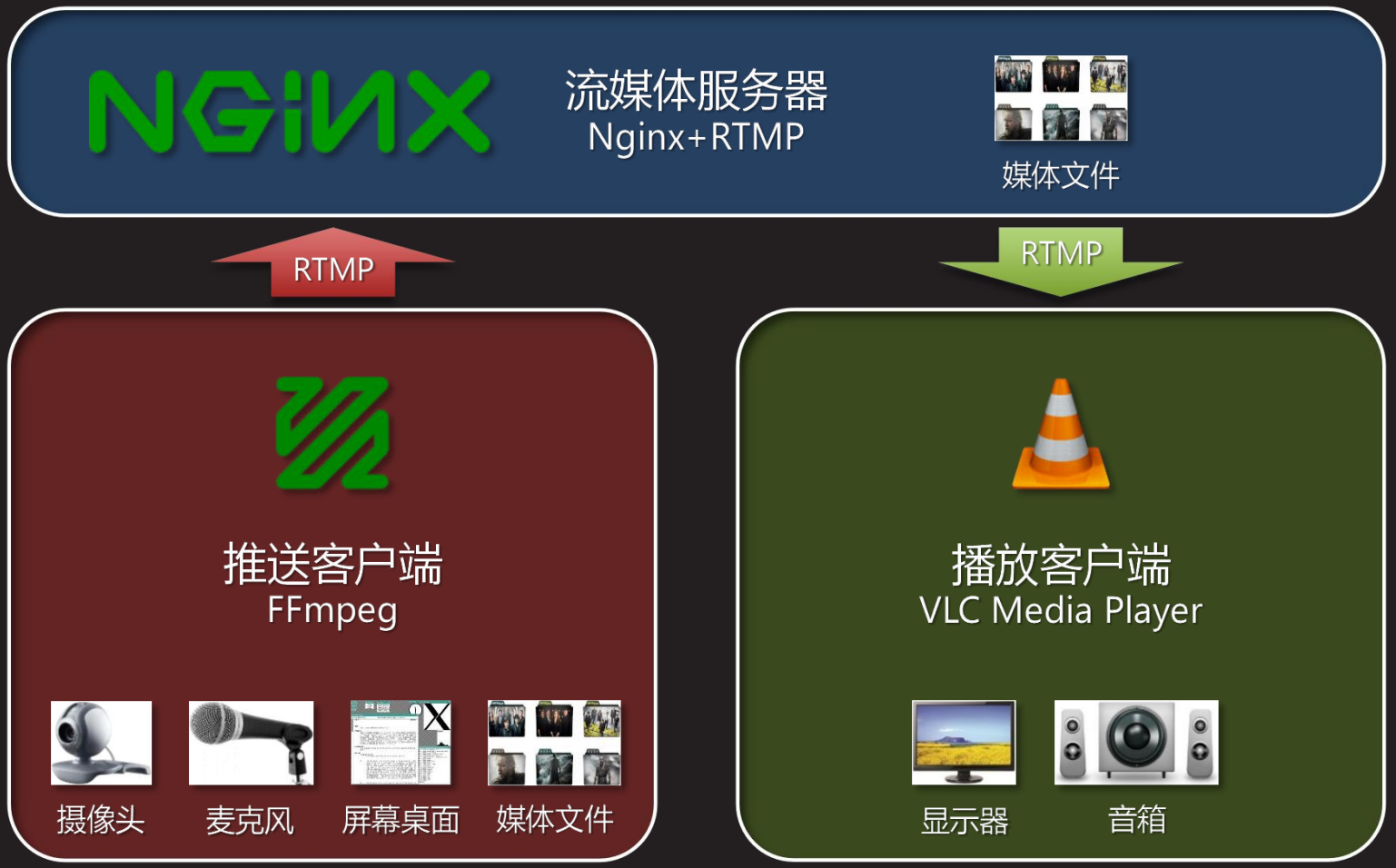
Technology Tarena 达内科技

附录

基于Nginx+RTMP的流媒体系统(上)

Tarena Technology 达内科技

1. 系统架构



知识讲解

基于Nginx+RTMP的流媒体系统(上)

Technology Tarena 达内科技

2. 在服务器上下载并安装编译依赖
   - $ sudo apt-get install build-essential libpcre3 libpcre3-dev libssl-dev
3. 在服务器上创建并切换工作目录
   - $ cd ~
   - $ mkdir Nginx-RTMP
   - $ cd Nginx-RTMP
4. 在服务器上下载nginx和nginx-rtmp源代码
   - $ wget http://nginx.org/download/nginx-1.7.5.tar.gz
   - $ wget https://github.com/arut/nginx-rtmp-module/archive/master.zip

知识讲解

基于Nginx+RTMP的流媒体系统(上)

Technology Tarena 达内科技

5. 在服务器上安装unzip工具
   - $ sudo apt-get install unzip
6. 在服务器上解压nginx和nginx-rtmp源代码包
   - $ tar -zxvf nginx-1.7.5.tar.gz
   - $ unzip master.zip
7. 在服务器上切换到nginx源代码目录
   - $ cd nginx-1.7.5
8. 在服务器上配置nginx编译选项以添加nginx-rtmp模块
   - $ ./configure --with-http_ssl_module --add-module=../nginx-rtmp-module-master

知识讲解

基于Nginx+RTMP的流媒体系统(上)

9. 在服务器上编译安装nginx
   - $ make
   - $ sudo make install
10. 在服务器上安装nginx启动脚本
   - $ sudo wget https://raw.github.com/JasonGiedymin/nginx-init-ubuntu/master/nginx -O /etc/init.d/nginx
   - $ sudo chmod +x /etc/init.d/nginx
   - $ sudo update-rc.d nginx defaults

知识讲解

基于Nginx+RTMP的流媒体系统(上)

11. 在服务器上启动并停止nginx服务，生成配置文件
    - $ sudo service nginx start
    - $ sudo service nginx stop
12. 在服务器上编辑nginx配置文件，在http部分和文件末尾添加配置信息并保存
    - $ sudo vi /usr/local/nginx/conf/nginx.conf

知识讲解

基于Nginx+RTMP的流媒体系统(上)

知识讲解

```
–    http {
       ...
       server {
         listen 8080;
         ...
         location /stat {
           rtmp_stat all;
           rtmp_stat_stylesheet stat.xsl;
         }
         location /stat.xsl {
           root /home/minwei/Nginx-RTMP/nginx-rtmp-module-master/;
         }
         location /hls {
           types {
             application/vnd.apple.mpegurl m3u8;
             video/mp2t ts;
           }
           root /tmp;
           add_header Cache-Control no-cache;
         }
       }
       ...
     }
```

基于Nginx+RTMP的流媒体系统(上)

知识讲解

```
rtmp {
  server {
    listen 1935;
    chunk_size 4096;
    application vod {
      play /home/minwei/Videos;
    }
    application live {
      live on;
      record off;
    }
    application hls {
      live on;
      hls on;
      hls_path /tmp/hls;
      wait_key on;
      hls_fragment 10s;
      hls_playlist_length 60s;
      hls_continuous on;
      hls_cleanup on;
      hls_nested on;
    }
  }
}
```

Technology Tarena 达内科技

总结和答疑