**一. 推拉流路数测试。**

1）推送1路流，拉流100、200、400、500、1000测试。

2）推流2路流，拉流100、200、400、500测试。

3）推送N路流，可以拉流RTC/RTSP/RTMP流。

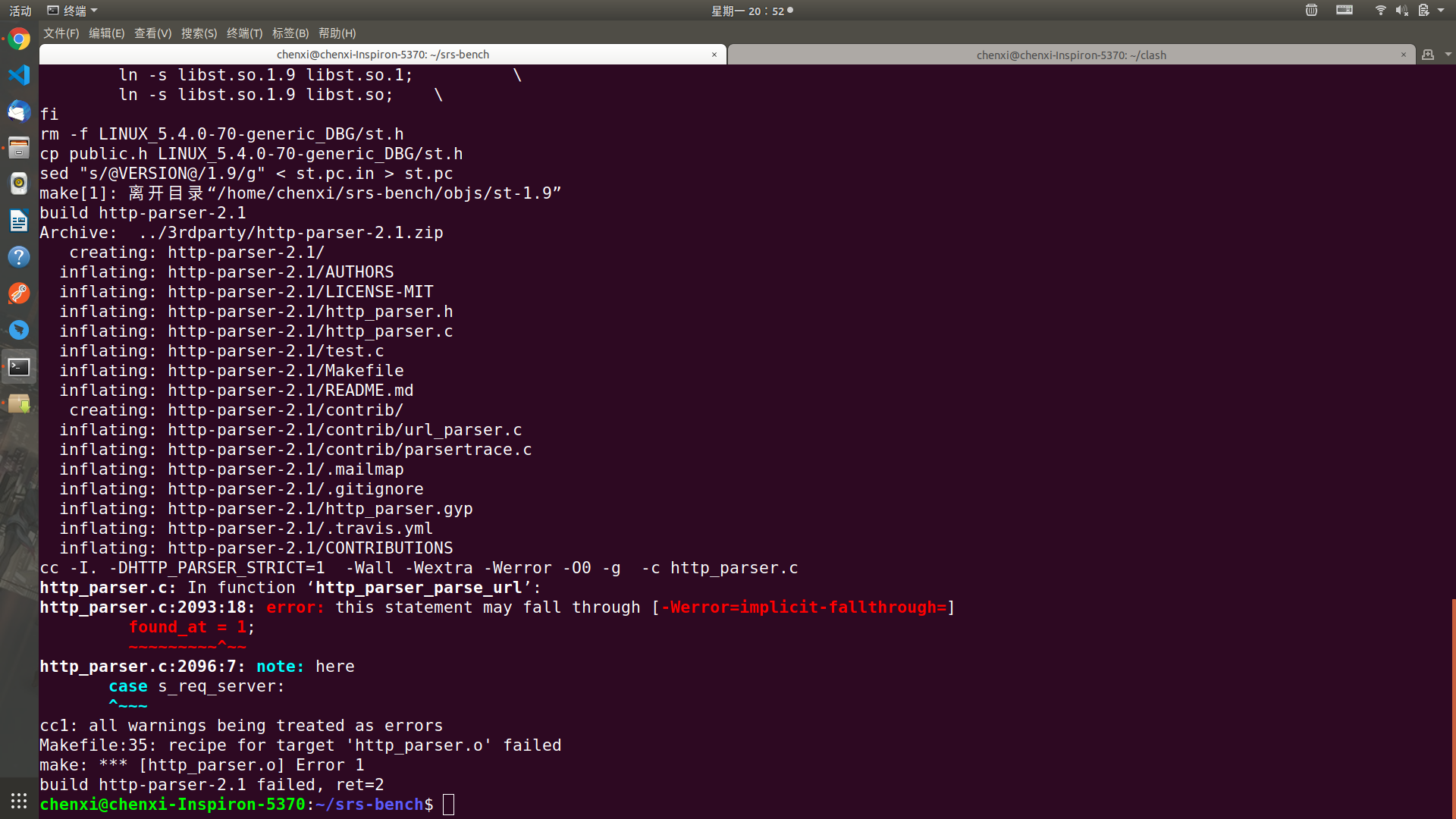
这里主要用srs-bench工具进行拉流测试。

**首先要安装srs-bench这个工具**

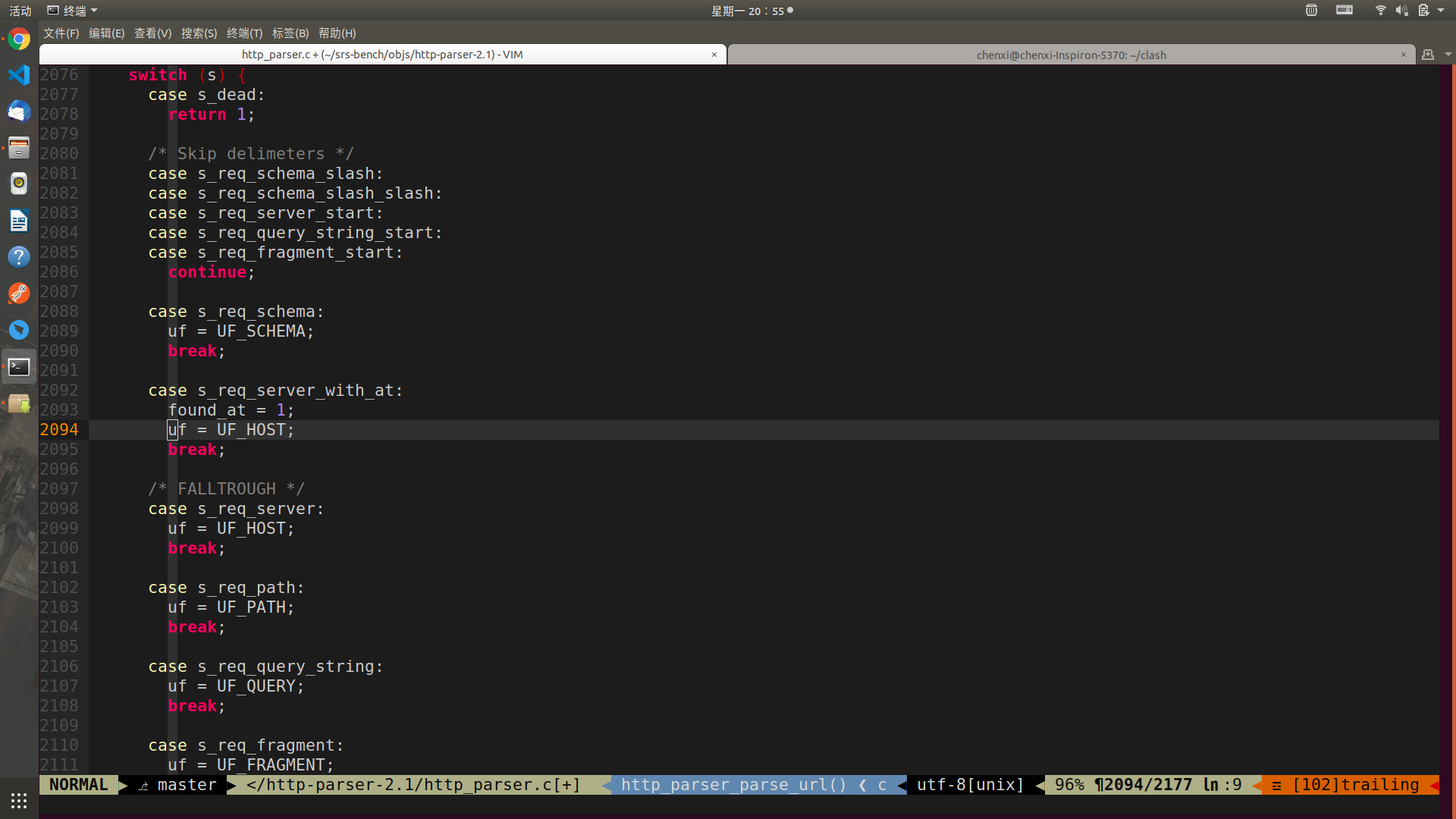
1）git clone <https://github.com/ossrs/srs-bench.git>

2）cd srs-bench && ./configure

这一步可能会出现如下错误



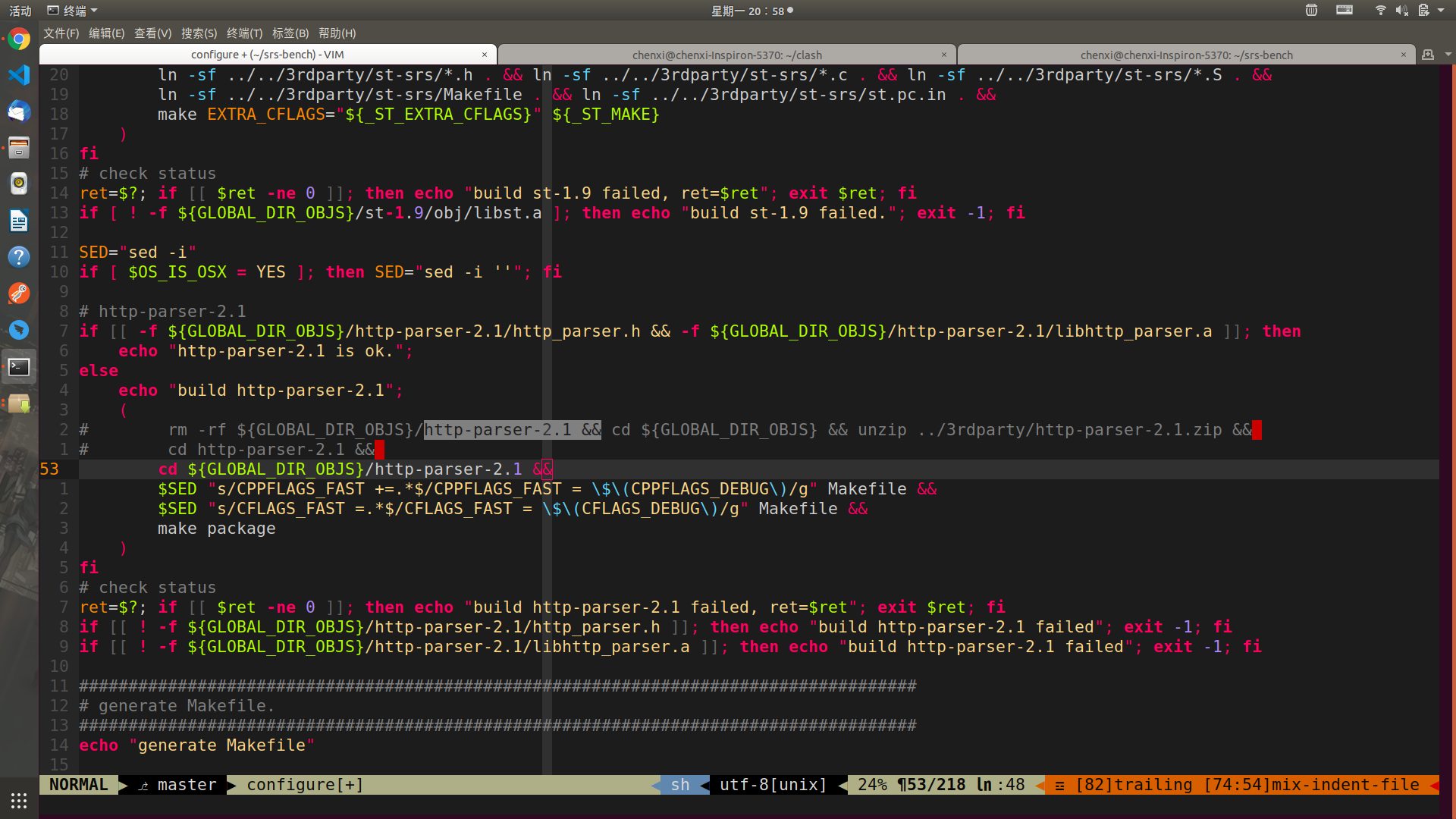
这里是因为编译器把警告当成错误输出了，解决方法：打开objs/http-parser-2.1/http\_parser.c这个文件，定位到2093行，在2093行下面加上uf = UF\_HOST; break; 如下图：



再打开configure配置文件，注释掉51，52行，在下面添加

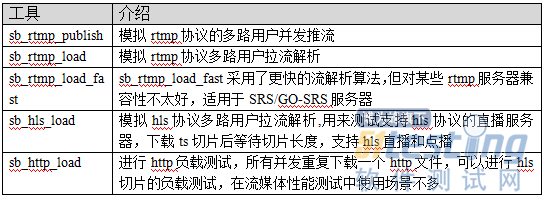
cd ${GLOBAL\_DIR\_OBJS}/http-parser-2.1 &&

如图

然后再执行./configure命令

3）执行make

编译完成后，会在obj文件夹下生成5个可执行文件



关于srs-bench工具更加详细的介绍请访问：<http://www.51testing.com/html/23/n-4472023.html?nomobile=1>

**开始测试**

**1. 用ffmpeg向boliving服务器推流**

推流可以选择转码和不转码两种方式

**1）不转码方式推流**

ffmpeg -re -stream\_loop -1 -i culture.mov -rtsp\_transport tcp -vcodec copy -acodec copy -b 1M -f rtsp rtsp://localhost:1554/live/livestream

参数解析：

-stream\_loop -1 ：无线循环推流

culture.mov :源文件

localhost：服务器的地址，1935：rtmp监听的端口

-b 1M 表示推的流的码率是1Mbps

live/livestream：存储流的路径

**2）转码方式推流**

ffmpeg -re -stream\_loop -1 -i culture.mov -rtsp\_transport tcp -vcodec libx264 -preset fast -profile:v baseline -tune zerolatency -x264-params keyint=10:min-keyint=5 -acodec copy -b 1M -f rtsp rtsp://localhost:1554/live/livestream

参数解析：

-stream\_loop -1 ：无线循环推流

culture.mov :源文件

localhost：服务器的地址，1935：rtmp监听的端口

-b 1M ：推的流的码率是1Mbps

live/livestream：存储流的路径

注：转码方式需要ffmpeg支持x264编解码，如果缺少，需要下载相应x264库，然后再重新编译安装ffmpeg，具体可百度。

如果要测试推流有没有成功，可用ffplay去拉流并播放：ffplay rtsp://localhost:1554/live/livestream

**2. 拉流**

切换到srs-bench文件下的obj文件，执行：

./sb\_rtmp\_load -c 100 -r rtmp://localhost:1935/live/livestream -m 5

参数解析：

-c 表示连接数，100表示拉100个流  
-r 连接地址 ，默认需要写固定地址，我是根据推流代码，自己改造了下，可以每个任务  
拉一个流  
-m  报告时间，单位s,5就代表5秒报告一次

然后在boliving后台程序中，可以看到服务器的状态，比如cpu，内存，带宽, 订阅流推流数等信息。

**二. 测试录制功能**

**1.录制视频路数测试。（包括混录录制数量）**

推送10路流，20路流，50路流。调用批量录制接口。

**1）首先我们要推流到boliving服务器，如果我们想要推多路流，可用下面的脚本来推**

#!/bin/bash

for j in {0..50}

do

ffmpeg -re -stream\_loop -1 -i ../下载/culture.mov -vcodec copy -acodec copy -b 1M -f rtsp rtsp://localhost:1554/live/livestream${j} &

done

**注意上面推多路流没有用转码功能推，原因是转码会消耗大量的cpu资源，推几路流就会把cpu支援耗光，所以不建议用转码方式推多路流。**

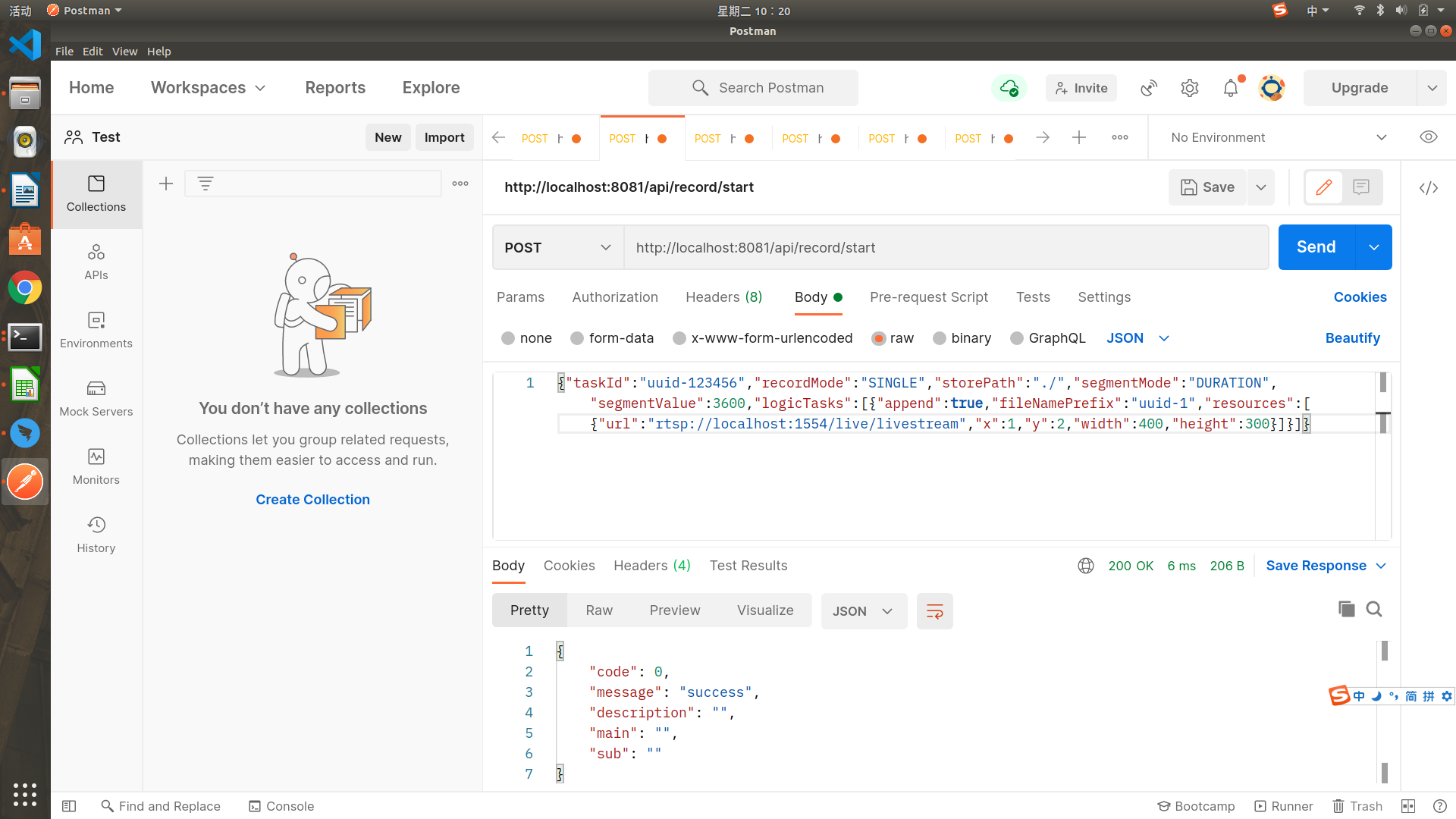
**上面推了50流，并且在后台运行（注意：因为是无限循环推流，所以如果要暂停推流，需要去杀死进程，这种方式比较麻烦，建议停止boliving服务器再重启，不去管后台这些推流的进程）**

**2. 用Postman工具录制推的流**

Postman是一个收发包工具，具体安装可百度。

可以先推1路流，然后用Postman去录制1，10，20，50，100，200，300路进行测试

**1）需要用Post去给服务器发送Json数据，如下面截图所示：**



用POST方式去发包

http://localhost:8081/api/record/start中 localhost:8081是服务器的地址和端口号

api/record/start发送消息的路径

{"taskId":"uuid-123456","recordMode":"SINGLE","storePath":"./","segmentMode":"DURATION","segmentValue":3600,"logicTasks":[{"append":**true**,"fileNamePrefix":"uuid-1","resources":[{"url":"rtsp://localhost:1554/live/livestream","x":1,"y":2,"width":400,"height":300}]}]}

上面是一段Json格式的数据，只录制了单路的流，如果要录制多路的流，需要增加 logicTasks对应数组的元素。比如要录制2路流，就要增加数组里面的元素，如下：

{"taskId":"uuid-123456","recordMode":"SINGLE","storePath":"./","segmentMode":"DURATION","segmentValue":3600,"logicTasks":[{"append":**true**,"fileNamePrefix":"uuid-1","resources":[{"url":"rtsp://localhost:1554/live/livestream","x":1,"y":2,"width":400,"height":300}]},

{"append":**true**,"fileNamePrefix":"uuid-2","resources":[{"url":"rtsp://localhost:1554/live/livestream","x":1,"y":2,"width":400,"height":300}]}]}

可以看到一个是uuid-1，一个是uuid-2,这就表示两路流在录制

如果要录制多路流，可以运行下面的我编写的bash脚本：

#!/bin/bash

#value表示你要录制多少路

value=10

i=1

#url表示服务器地址和请求的路径

url="rtsp://localhost:1554/live/livestream"

echo -e "{"\"taskId\"":"\"uuid-123456\"","\"recordMode\"":"\"SINGLE\"","\"storePath\"":"\"./\"","\"segmentMode\"":"\"DURATION\"","\"segmentValue\"":3600,"\"logicTasks\"":[\c"

while(($i<=$value))

do

if [ $i -eq $value ]

then

echo -e "{"\"append\"":true,"\"fileNamePrefix\"":"\"uuid-${i}\"","\"resources\"":[{"\"url\"":"\"${url}\"","\"x\"":1,"\"y\"":2,"\"width\"":400,"\"height\"":300}]}\c"

break;

else

echo "{"\"append\"":true,"\"fileNamePrefix\"":"\"uuid-${i}\"","\"resources\"":[{"\"url\"":"\"rtsp://localhost:1554/live/livestream\"","\"x\"":1,"\"y\"":2,"\"width\"":400,"\"height\"":300}]}",

fi

let "i++"

done

echo -e "]}"

上面脚本可以生成录制多路流的Json格式数据。

如果POST成功，服务器会response包。response回复的内容是一个Json格式的数据：

{

"code": 0,

"message": "success",

"description": "",

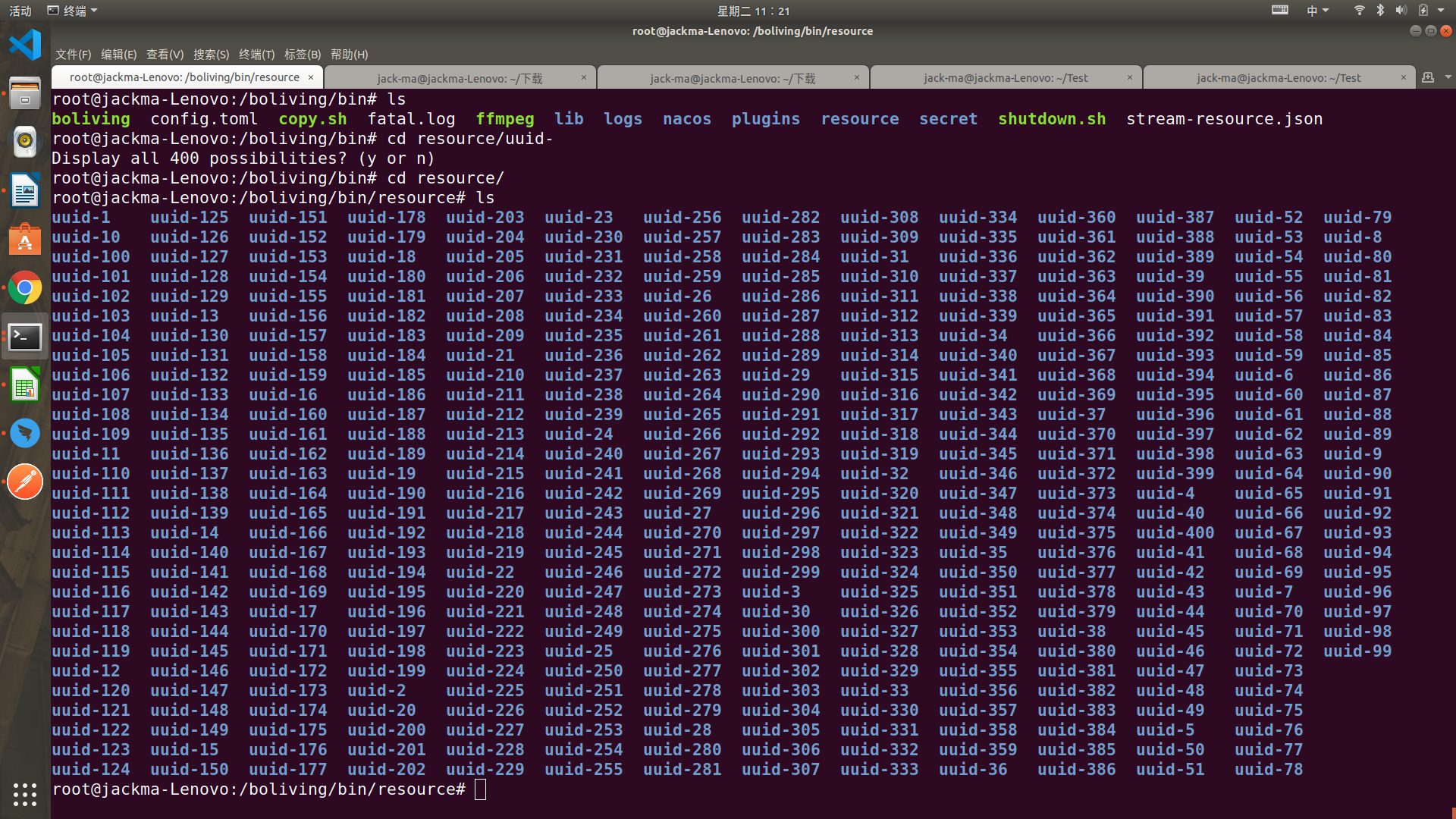
"main": "",

"sub": ""

}

这样就代表录制成功。

可以到boliving服务器的resource文件夹查看录制的内容，如图：



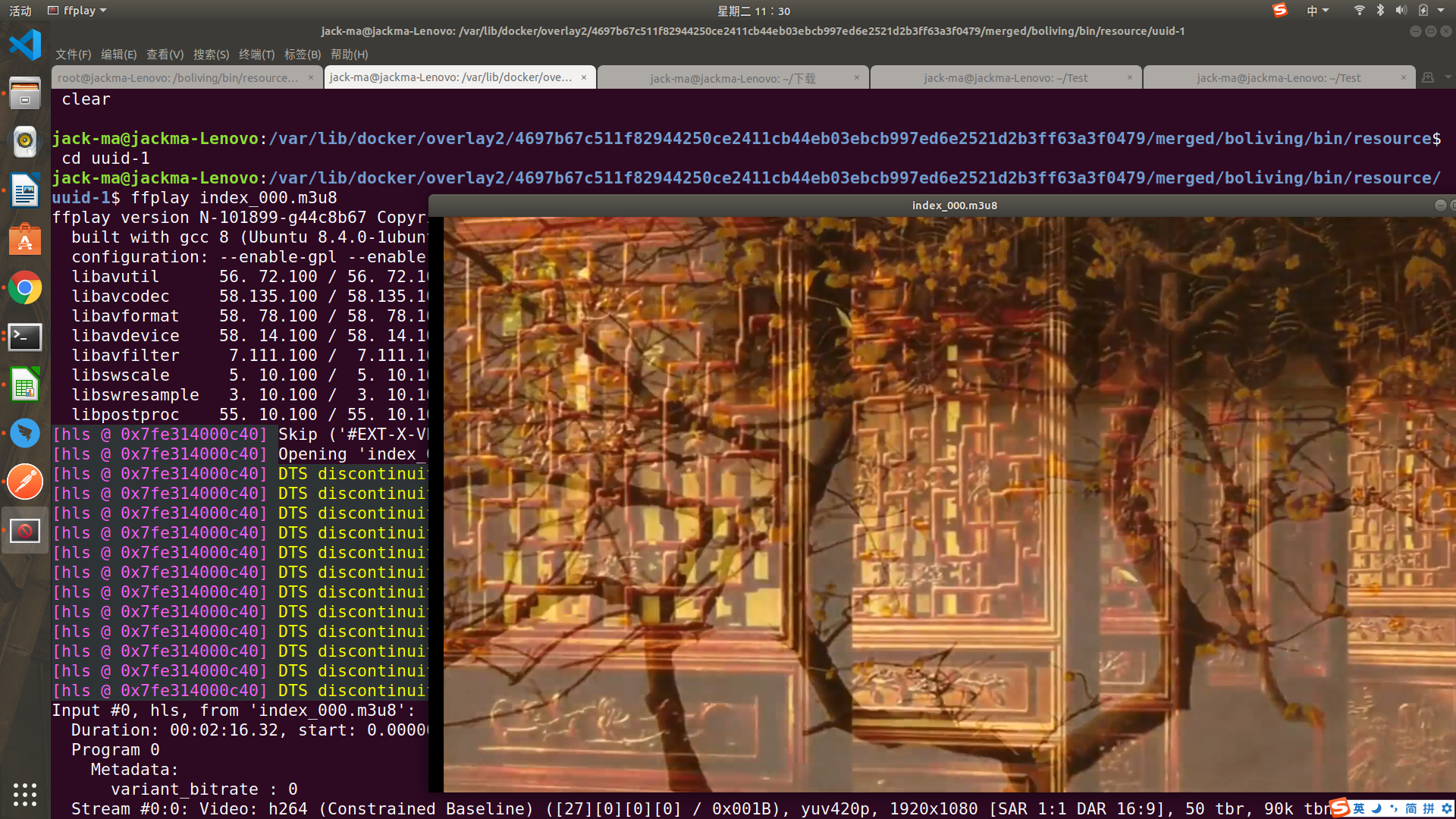
上面就代表录制了400路

**2）播放录制内容的文件**

录制完成后我们可以播放下录制的内容，检查下录制是否有问题

录制文件的格式m3u8格式，关于m3u8格式具体可以百度。

m3u8格式我们可以用ffplay播放。比如我们要播放uuid-1这个录制的文件，可以先进入uuid-1这个文件夹，然后用ffplay直接播放。如下图：

**3）暂停正在录制的流**

也是向服务器发送POST包

http://localhost:8081/api/record/stop

POST

{"taskId":"uuid-123456","logicTaskIds":["uuid-1"]}

uuid-1表示暂停这一路流的录制

构造Json格式的脚本

#!/bin/bash

value=100

i=1

url="rtsp://127.0.0.1:1554/live/livestream"

while(($i<=$value))

do

if [ $i -eq $value ]

then

printf "{\n"

printf "\t\"id\": \"test${i}\",\n"

printf "\t\"type\": \"rtsp\",\n"

printf "\t\"level\": \"main\",\n"

printf "\t\"url\": \"${url}\",\n"

printf "\t\"from\": \"config\"\n"

printf "}"

else

printf "{\n"

printf "\t\"id\": \"test${i}\",\n"

printf "\t\"type\": \"rtsp\",\n"

printf "\t\"level\": \"main\",\n"

printf "\t\"url\": \"${url}\",\n"

printf "\t\"from\": \"config\"\n"

printf "},\n"

fi

let "i++"

done

拉流的脚本

#!/bin/bash

for i in {1..100}

do

./../srs-bench/objs/sb\_rtmp\_load -c 1 -r rtmp://localhost:1935/main/test${i} &

done