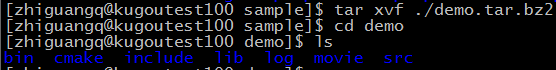
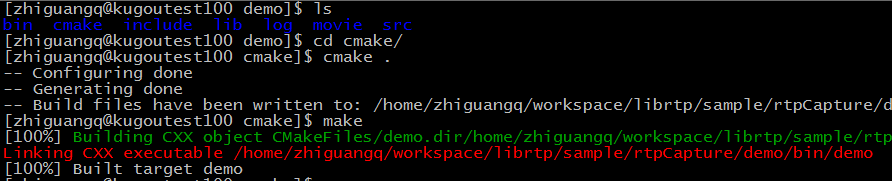
SDK开发手册

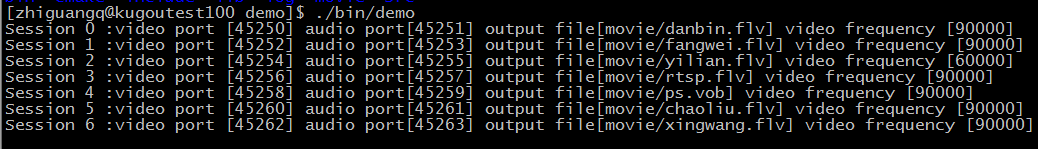
* 解压demo，并进入demo，SDK头文件为include/RTPCaptureSDK.h，库文件为lib/libRTPCaptureSDK.a。其它目录主要是demo程序所有：bin是生成的执行程序存放目录，cmake是编译文件存放目录，log是log文件存放目录，movie是生成的视频文件存放目录，src是demo的main.cpp文件存放目录。



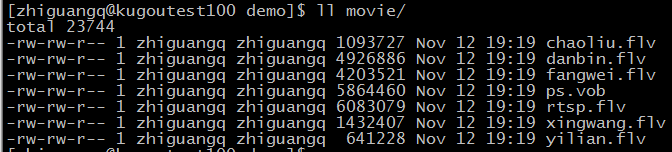
* 怎样编译这个demo呢？在Linux 64环境中，进入cmake目录，用cmake进行编译。分别输入两个命令：cmake . 和make



* 怎么运行demo呢？就是上一步编译成功后，在bin目录里生成demo可执行程序。出来demo目录，运行 ./bin/demo就可以启动了。



* 这个时候就可以向启动的服务端口发rtp视频流了。完了可以在movie里查看生成的视频文件。



* 接口说明，SDK用到两个接口，addCaptureRtpSession和delCaptureRtpSession。

int addCaptureRtpSession(std::string fileName, int listenVideoPort, int listenAudioPort, int videoFrequency = 90000);

//1.生成视频文件的文件名，

//2接受视频的rtp端口，

//3接受音频的rtp端口，

//4视频时间戳使用频率，（默认都使用90000，只有亿联需要使用60000）

//return 返回的是session的ID

void delCaptureRtpSession(int session);

//1.session的ID

* Cmake说明，这个demo使用cmake进行编译，使用者也可以自己写Makefile。

cmake\_minimum\_required(VERSION 2.6)

PROJECT(demo) // 生成执行文件的名称

INCLUDE\_DIRECTORIES($ENV{PWD}/../include) // 程序头文件的目录

LINK\_DIRECTORIES($ENV{PWD}/../lib) // 程序依赖库的目录

SET(EXECUTABLE\_OUTPUT\_PATH $ENV{PWD}/../bin) // 生成文件存放位置

ADD\_EXECUTABLE(demo $ENV{PWD}/../src/main.cpp) // 编译哪些源文件

TARGET\_LINK\_LIBRARIES(demo pthread)

TARGET\_LINK\_LIBRARIES(demo libRTPCaptureSDK.a) // 这个就是我们的SDK库

TARGET\_LINK\_LIBRARIES(demo libPocoUtild.a)

TARGET\_LINK\_LIBRARIES(demo libPocoNetd.a)

TARGET\_LINK\_LIBRARIES(demo libPocoFoundationd.a)

TARGET\_LINK\_LIBRARIES(demo srs\_librtmp.a)

// 其它依赖的库使用者直接引用进来就行了