

基于 Android 内核的流媒体播放器分析与研究

邓昌明,周渊平,张兴龙

(四川大学 电子信息学院,四川 成都 610065)

摘要:Android 是一个移动平台上的开源操作系统,被广泛应用于各种移动终端设备。但其对流媒体的支持功能有限,比如仅支持 3gp 等格式,对标准流媒体协议支持兼容性不好,不能播放 VLC 作为流媒体服务器的视频流,且不能用作服务器端。文中在深入分析 Android 内核的流媒体实现方案之后提出将开源流媒体解决方案 Live555 移植到 Android 内核中,扩充其对流媒体的功能,且使其可以作为流媒体服务器来使用。实验结果表明 Live555 在 Android 内核中可以有效地完成对流媒体数据的接收和处理。

关键词:Android;流媒体;Live555

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2014)02-0179-03

doi:10.3969/j.issn.1673-629X.2014.02.044

Analysis and Research of Streaming Media Player Based on Android Kernel

DENG Chang-ming, ZHOU Yuan-ping, ZHANG Xing-long

(School of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China)

Abstract:Android is an open source operating system on the mobile platform, and it is widely used in a variety of mobile devices. But its streaming media function is limited, for example, it only supports 3gp format and so on, and standard streaming protocol is not supported well, it can not play a streaming video when VLC is considered as a streaming media server, and can not be used as a server-side. In this paper, analyze the streaming media implementation of Android kernel in-depth, and Live555 is ported to the Android kernel to expand its streaming media function, so Android can be used as a streaming media server. The experiment result shows that Live555 can effectively accomplish the receiving and processing of streaming media data in Android kernel.

Key words:Android; streaming media; Live555

0 引言

Android 是 Google 公司在开源操作系统 Linux 基础上专为移动终端设备而开发的操作系统。Android 系统自推出以来,受到了手机厂商、软件厂商、运营商以及广大使用者的青睐。Android 系统已经在越来越多的终端设备上运行,其发展必将在未来的智能社会中占据一席之地。

Android 系统的多媒体播放功能主要分为本地播放和远程播放,本地播放即播放存储卡中的多媒体文件,远程播放即播放存储在远端服务器上的多媒体文件,多以 RTSP、HTTP 等实现视频流传输,但 Android 系统本身自带的播放器功能对流媒体协议的解析兼容

性并不好,比如不能够播放通过 VLC 播放器串流的音视频文件。文中深入分析了 Android 内核流媒体播放器和开源流媒体解决方案 Live555 的源码实现,并提出将 Live555 移植入 Android 内核来扩充系统的流媒体功能,使其具有更好的兼容性及更多的应用功能。

1 Android OS

Android 系统分为 4 层,从高到低分别是应用层、应用框架层、系统运行层和 Linux 内核层^[1-2]。

(1)应用层。该层由运行在 Dalvik 虚拟机(Google 公司专为 Android 设计的基于寄存器的 Java 虚拟机)上的应用程序组成,例如联系人、浏览器、日历等。

收稿日期:2013-05-06

修回日期:2013-08-10

网络出版时间:2013-11-29

基金项目:国家自然科学基金资助项目(61071160)

作者简介:邓昌明(1989-),男,硕士研究生,研究方向为信号与信息处理;周渊平,教授,博士生导师,研究方向为通信与信息系统、信号与信息处理。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20131129.0826.005.html>

(2)应用框架层。该层包含基本的控件、通知管理器、活动管理等由开发人员直接调用的 API 组成(这些 API 主要由 Java 语言编写)。

(3)系统运行层。也就是 Native 层,该层主要包括由 C/C++ 编写的一些底层的库,比如多媒体库、OpenGL ES、SQLite、Dalvik 虚拟机等,该层是对应用框架层提供支持的层。文中所提出的 Live555 移植也即移植到该层上。

(4)Linux 内核层。该层主要包括驱动、内存管理、进程管理、网络协议栈等内容,主要是对硬件的抽象及底层系统的管理。

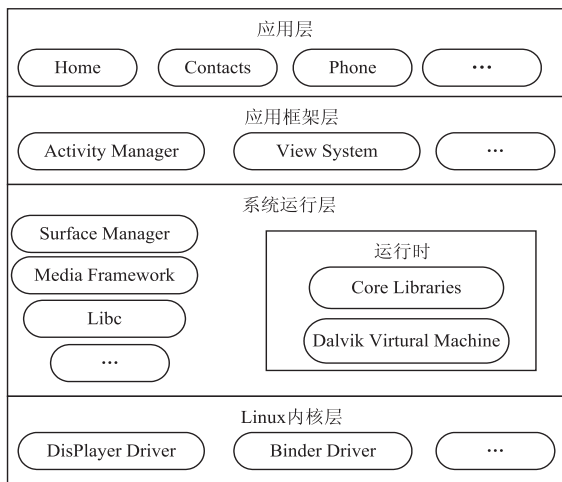


图 1 Android 系统架构图

2 Android 内核流媒体播放器分析

文中以 Android 2.3 内核版本为研究对象,在该版本及以后的版本中 Google 引进了架构稍微简洁的 Stagefright 多媒体框架,并逐步取代了 Android 2.2 及其以前版本的 OpenCore 多媒体框架。

Stagefright 位于 Android 系统的 Native 层,它在 Android 系统中以共享库 libstagefright.so 的形式存在,其核心是能够独立播放音视频的 AwesomePlayer 类,该类提供播放器相关的 API 通过 JNI 接口供上层 Java 层(MediaPlayer)调用。

AwesomePlayer 的整体架构是基于事件调度机制的,它通过 event 事件调度机制来实现包括解析 URI、音视频解码及输出显示等功能之间的驱动。在 AwesomePlayer 内部有一个 TimedEventQueue mQueue 变量,这个变量就是 AwesomePlayer 的事件队列,通过不断检查队列是否为空来驱动事件的执行,如果为空则等待 event 的插入,否则,获取队列中的第一个 event,然后计算 event 所要求的延时时间后进行延时操作,最后将 event 从队列中删除并执行该 event。整个事件调度器就是通过不断循环上述过程来进行调度。Awe-

somePlayer 播放器的各部分具体流程如图 2 所示^[3-4]。

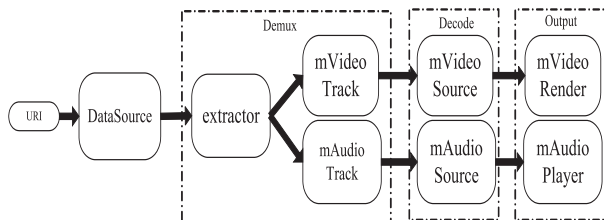


图 2 AwesomePlayer 播放流程

首先,通过 setDataSource() 设置播放器的数据源,URI 可以是 http://或 rtsp://等。

然后,通过 prepare() 函数指定事件 onPrepareAsyncEvent() 回调函数,在该函数中调用 finishSetDataSource_l(), 根据对应的 URI 连接服务器,通过从服务器端获取的媒体信息来创建 extractor,得到 extractor 之后,通过 setVideoSource() 和 setAudioSource() 产生独立的 mVideoTrack(视频)、mAudioTrack(音频)数据流,分别为音视频解码器提供有待解码的音视频帧数据。接下来由 initVideoDecoder() 和 initAudioDecoder() 根据上面产生的 mVideoTrack(视频)、mAudioTrack(音频)数据流类型来构造各自的 mVideoSource 和 mAudioSource 音视频解码器。

最后,AwesomePlayer 分别调用 mVideoRenderer 和 mAudioPlayer 将音视频数据输出。

3 Android 内核 RTSP 实现分析

Android 内核对 RTSP 的处理位于 frameworks/base/media/libstagefright/rtsp 目录下,由于流媒体常常涉及比如连网、解码等费时的的工作,为了不阻塞整个程序的执行,因此通常将费时的的工作交由后台线程去执行。Android 通过将 Linux 底层的线程 API 封装到 ALooper 类中来执行线程的处理,并结合 AHandler 和 AMessage 从而构成 Android 的整个消息处理机制。RTSP 及 RTP 媒体数据包的解析是基于 Android 的 handler 消息处理机制来驱动执行的。

3.1 Android 内核流媒体消息处理机制

Android 流媒体的消息处理机制由 ALooper, AHandler 和 AMessage 构成^[5]。ALooper 在调用 start() 后启动一个线程不断去检查 mEventQueue 队列上是否有 event 事件以及是否达到执行时间;若有,则调用 gLooperRoster.deliverMessage(event, mMessage) 分发事件消息,根据消息的 handler_id 最终调用对应的 handler->onMessageReceived(msg) 去处理相应的消息;若无,则等待事件到达。ALooper 的 post() 用于向队列 mEventQueue 上添加 event 事件并唤醒等待的线程去执行上述检查。event 中包含消息,该消息由 AMessage 表示。而具体的每一个消息对应的处理由 AHandler

子类去执行,子类实现其纯虚函数 `onMessageReceived()` 用于处理队列上的某个时刻对应的 `AMessage`。

3.2 RTSP 处理过程

RTSP 的处理分为两部分:一部分为开始建立与流媒体服务器的连接和处理解析 RTSP,包括 `DESCRIBE`、`SETUP`、`PLAY` 等协议命令;另一部分用来处理接收到的 RTP 媒体数据包以及解析 RTP 数据头,并将媒体数据包存储到帧缓冲区中,以供解码部分使用。其中每一部分都有一个 `ALooper` 和一个 `AHandler` 对应,RTSP 是通过 `ARTSPConnection` (继承自 `AHandler`) 来处理,而 RTP 数据包由 `ARTPConnection` (继承自 `AHandler`) 来处理,它们通过发送不同 `AMessage` 来驱动程序的执行。

4 Live555 流媒体客户端分析

Live555 是一个为流媒体提供解决方案的跨平台的 C++ 开源项目,它实现了对标准流媒体传输协议如 RTSP、RTP/RTCP、SIP 等的支持。支持多种视频编解码格式,被大多数流媒体服务器应用为开发平台。Live555 有四个基本的库,即 `BasicUsageEnvironment`、`UsageEnvironment`、`GroupSock` 和 `LiveMedia`^[6-7]。对于 Live555 的客户端编程而言,最重要的即是 `MediaSession` 类,它用于表示一个 RTP 会话,一个 `MediaSession` 可能包含多个子会话 (`MediaSubSession`),子会话可以是音频子会话、视频子会话等。通过 `MediaSubSession` 对象可以获取媒体的属性,比如 `fSubsession` 为一个子会话的对象,则通过 `fSubsession.mediumName()` 可以获知该会话是音频还是视频会话,`fSubsession.codecName()` 可以获知该多媒体的编码格式;对于媒体数据则可以通过 `fSubsession->readSource()->getNextFrame()` 函数来获取 RTP 数据包解析过后的一帧媒体数据,这个数据经过简单处理即可送往解码端;还可以通过 `fSubsession.rtpSource()->hasBeenSynchronizedUsingRTCP()` 来获知收到的数据是否已经经过 RTCP 进行同步处理等。

5 Live555 的移植及结果显示

文中以 Android2.3 内核为移植对象,系统测试采用九鼎创展科技有限公司的 x210ii 开发板,该开发板是基于 ARM Cortex-A8,采用三星公司的 S5PV210 为 CPU,能够流畅运行 Android 系统。在文中,Live555 被移植到系统内核的 stagefright 模块,该模块位于 Android 内核源码 `frameworks/base/media/libstagefright` 目录下,这样 stagefright 的其他模块就可以通过直接调用 Live555 库来实现流媒体的功能,也即将流媒体解决方案封装在 Native 层,这样相比于在 Java 层通过 JNI 来

调用提高了效率。移植步骤如下^[8-12]:

第一步:编写 Android.mk。

首先在 `libstagefright` 目录下创建 `Live555` 目录文件,将 `Live555` 源码目录下的文件复制到该目录下,在除了 `testProgs` 文件夹外的其他五个文件夹下创建 `Android.mk` 编译文件用于生成静态库;然后在 `Live555` 目录下创建 `Android.mk`,用于生成一个测试应用程序 `openRTSP`,以验证移植是否成功;最后再修改 `libstagefright` 目录下的 `Android.mk` 文件,主要是添加 `Live555` 生成的静态库以及 `Live555` 源文件所在目录,这样其他程序就可以使用 `Live555` 了。`Android.mk` 文件形式如下:

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := openRTSP
LOCAL_CFLAGS := .....
LOCAL_SRC_FILES := testProgs/openRTSP.cpp .....
LOCAL_C_INCLUDES := $(LOCAL_PATH)/liveMedia .....
LOCAL_STATIC_LIBRARIES := libliveMedia .....
LOCAL_SHARED_LIBRARIES := libcutils libstdc++ libc
LOCAL_MODULE_TAGS := debug
```

第二步:编译。

由于编译整个系统需要花费较长时间,因此为了缩短编译时间,这里使用模块式编译方法。首先在 Android 源码的 `build` 目录下执行 `envsetup.sh`,然后在 `libstagefright` 目录下执行模块化编译命令 `mm PRODUCT-full_smdkv210-eng`,这样 Live555 即被编译到 `libstagefright` 模块中。

第三步:测试应用程序。

为了验证第一步中生成的应用程序 `openRTSP` 是否能在 Android 系统中运行,将该程序放入 Android 内核中运行。首先,在 PC 端运行 Live555 的流媒体服务器程序 `live555MediaServer.exe`,然后在终端 (ubuntu 系统) 输入如下命令:

```
adb push openRTSP /data/openRTSP
adb shell
cd data
./openRTSP rtsp://192.168.0.72/test.mp4
```

测试结果如图 3 所示。

```
# ./openRTSP rtsp://192.168.0.72/test.mp4
Opening connection to 192.168.0.72, port 554...
...remote connection opened
Sending request: OPTIONS rtsp://192.168.0.72/test.mp4 RTSP/1.0
CSeq: 2
User-Agent: ./openRTSP (LIVE555 Streaming Media v2012.11.30)
```

```
.....
Started playing session
Receiving streamed data (signal with "kill -HUP 2773" or "kill -USR1 2773" to terminate)...
```

图 3 实验结果图

图 3 为截取应用程序 openRTSP 输出的一部分显示输出,通过输出结果可知,openRTSP 可在 Android 系统中运行连接到流媒体服务器端,并成功接收流媒体数据,实现了预期的结果。

6 结束语

文中在深入分析 Android 内核流媒体的源码实现的基础上,提出通过将开源流媒体解决方案 Live555 移植进入 Android 内核使系统支持更多流媒体功能,比如可以编写调用 Live555 的 JNI 接口,上层应用即可直接实现流媒体服务器的功能等。文中的研究对于流媒体引擎实现及基于 Android 系统的智能机顶盒及 IPTV 等流媒体应用解决方案有一定借鉴意义。

参考文献:

- [1] 李 宁. Android 开发权威指南[M]. 北京:人民邮电出版社,2011.
- [2] Google Inc. Google Android developer[EB/OL]. 2013-03. <http://developer.android.com/about/versions/index.html>.
- [3] Zhao Xueliang, Tian Dan. The architecture design of streaming media applications for Android OS[C]//Proc of 2012 IEEE 3rd international conference on software engineering and serv-

ice science. Beijing:[s. n.], 2012;280-283.

- [4] Song Maoqiang, Xiong Wenkuo, Fu Xiangling. Research on architecture of multimedia and its design based on Android [C]//Proc of 2010 international conference on internet technology and applications. Wuhan:[s. n.], 2010.
- [5] 邓凡平. 深入理解 Android 卷 I[M]. 北京:机械工业出版社,2011.
- [6] 刘畅棣,包 杰,王宁国. 基于 Live555 的网络视频监控系统设计与实现[J]. 现代电信科技,2012(12):3842-3842.
- [7] 茅炎菲,黄忠东. 基于 RTSP 协议网络监控系统的研究与实现[J]. 计算机工程与设计,2011,32(7):2523-2526.
- [8] 胡 成,任平安,李文莉. 基于 Android 系统的 FFmpeg 多媒体同步传输算法研究[J]. 计算机技术与发展,2011,21(10):85-87.
- [9] Song Maoqiang, Sun Jie, Fu Xiangling. Design and implementation of media player based on Android[C]//Proc of 2010 6th international conference on wireless communications networking and mobile computing. Chengdu:[s. n.], 2010.
- [10] 韩 超,梁 泉. Android 系统级深入开发移植与调试[M]. 北京:电子工业出版社,2011.
- [11] 柯元旦. Android 内核剖析[M]. 北京:电子工业出版社,2011.
- [12] 赵学良. 基于 Android 的流媒体引擎设计与实现[D]. 成都:电子科技大学,2012.

(上接第 178 页)

参考文献:

- [1] 郭亚军. 综合评判理论、方法及拓展[M]. 北京:科学出版社,2012.
- [2] 杨 楠,曹 雷,陈希亮,等. 模糊综合评价在战术训练模拟系统中的运用[J]. 计算机技术与发展,2012,22(6):203-206.
- [3] 李云峰,王 琛. 装备制造企业供应商综合评价系统[J]. 计算机技术与发展,2012,22(4):149-152.
- [4] Xu Gang, Yang Yongping, Lu Shiyuan, et al. Comprehensive evaluation of coal-fired power plants based on grey relational analysis and analytic hierarchy process [J]. Energy policy, 2011,39(5):2343-2351.
- [5] Agmon-Levin N, Shapira Y, Selmi C, et al. A comprehensive evaluation of serum autoantibodies in primary biliary cirrhosis [J]. Journal of autoimmunity, 2010,34(1):55-58.
- [6] 李 鹏,俞国燕. 多指标综合评价方法研究综述[J]. 机电产品开发与创新,2009,22(4):24-26.
- [7] 朱建军. 层次分析法的若干问题研究及应用[D]. 沈阳:东北大学,2005.
- [8] Krohling R A, Campanharo V C. Fuzzy TOPSIS for group deci-

sion making: A case study for accidents with oil spill in the sea [J]. Expert systems with applications, 2011, 38(4):4190-4197.

- [9] 张润楚. 多元统计分析[M]. 北京:科学出版社,2006.
- [10] 刘秀梅,赵克勤. 基于集对分析联系数的信息不完全直觉模糊多属性决策[J]. 数学的实践与认识,2010,40(1):67-76.
- [11] Wang Wensheng, Jin Juliang, Ding Jing, et al. A new approach to water resources system assessment-set pair analysis method [J]. Science in China (Series E: Technological Sciences), 2009,52(10):3017-3023.
- [12] 赵克勤,宣爱理. 集对论——一种新的不确定性理论方法与应用[J]. 系统工程,1996,14(1):18-23.
- [13] Kan Yuan, Jiang Wei, Ji Xiao-juan. Intrusion detection model based on set pair analysis theory [J]. Future communication, computing, control and management, 2012,141:35-42.
- [14] 张 鹏,王光远. 新集对论[J]. 哈尔滨建筑大学学报,2000,33(3):1-5.
- [15] 马守明. 基于 WSN 的普适计算情景感知关键技术研究[D]. 南京:南京邮电大学,2011.

基于Android内核的流媒体播放器分析与研究

作者：[邓昌明](#)，[周渊平](#)，[张兴龙](#)，[DENG Chang-ming](#)，[ZHOU Yuan-ping](#)，[ZHANG Xing-long](#)

作者单位：[四川大学 电子信息学院, 四川 成都, 610065](#)

刊名：[计算机技术与发展](#)

英文刊名：

Computer Technology and Development

ISTIC

年，卷(期)：

2014(2)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201402045.aspx