

Android 流媒体应用架构设计

Xueliang Zhao

Research institute electronic science and technology

University of Electronic Science and Technology of China

Chengdu, Sichuan Province, P.R China

zhaoxueliang 1986@163.com

Dan Tian

Research institute electronic science and technology

University of Electronic Science and Technology of China

Chengdu, Sichuan Province, P. R China

tiandan@188.com

摘要

Android 操作系统中多媒体功能不完善、流媒体系统复杂，开发周期长，并且效率低。为了解决这些问题，本论文设计了 Linux 内核层之间的流引擎层和 Android 平台的应用程序框架层构建快速、方便的流媒体应用开发框架。首先，分析了 Android 操作系统，然后将流媒体引擎加入 Android 操作系统架构。基于流媒体系统的特点，将流媒体引擎分为五层，包括用户界面、数据捕获、数据输出、编解码器和网络传输层。最后，将流媒体架构应用于智慧医疗终端项目。通过视频数据传输客户端的例子证实了框架的可行性。

关键词：Android；软件架构；流媒体引擎；智慧医疗。

1. 绪论

随着半导体技术不断向前发展,移动终端的处理能力不断增强,移动互联网带宽逐渐扩大。移动终端的流媒体应用程序已经成为消费的主要方式。虽然市场上有很多种移动终端操作系统,但 Android 操作系统由于其开源的特点,一直很受欢迎。在 2011 年下半年全球份额排名第一,它已经成为其中一个最流行的手机操作系统。

Android 流媒体应用已经成为一个热点了,但 Android 系统的出现较晚,流媒体应用的开发还不成熟。同时,由于流媒体系统的复杂性,Android 的多媒体功能开发不容易理解,其扩展开发非常困难,不利于流媒体应用的发展。

针对 Android 多媒体架构中的问题,本文设计了一种基于 Linux 内核层与 Android 应用程序层之间的流媒体引擎,为流媒体应用开发提供了用户友好的 API。是一个为 Android 流媒体应用程序设计的开放开发平台。最后用智慧医疗项目验证可行性并显示其用途。

II. Android 系统架构

Android 操作系统是一种由 Google 发布的开源操作系统。在 Android 中，层次结构就类似其他操作系统；它分为五层，包括 Linux 内核层、硬件抽象层、类库和运行时层、应用程序框架和应用程序层。系统架构如图 1 所示。各层的描述如下

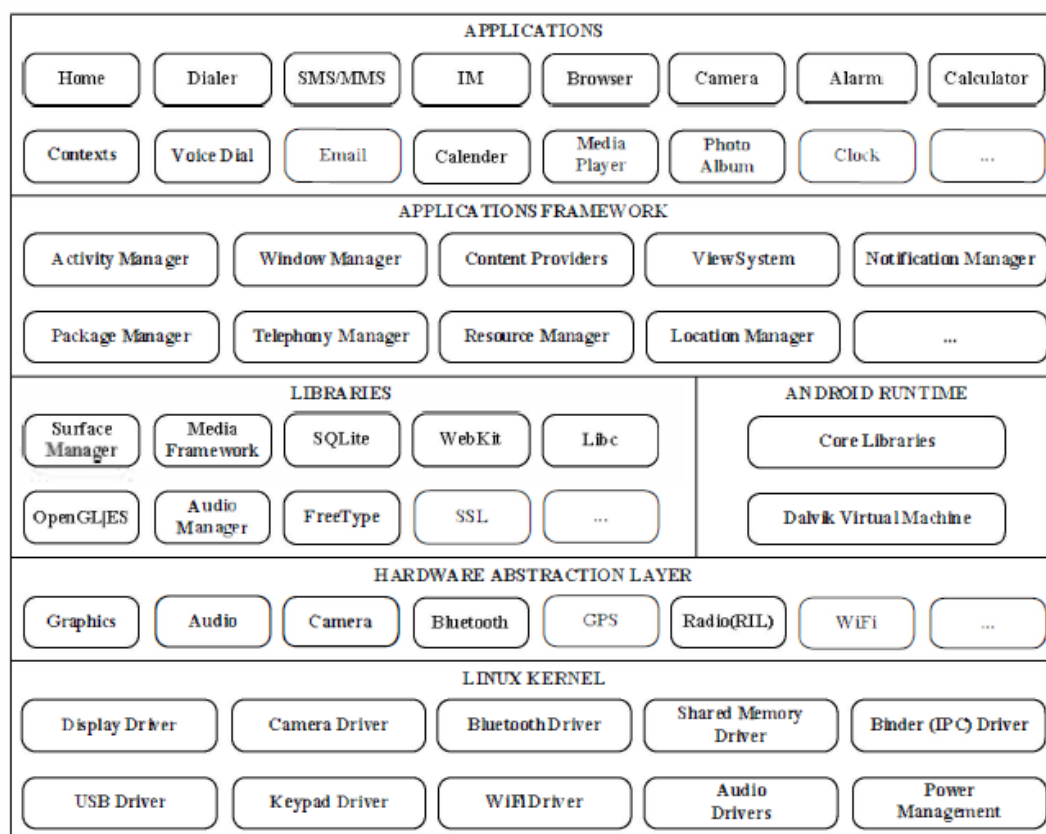


图 1 Android 系统架构

A. Linux 内核层

Android 操作系统基于 Linux 2.6 内核，Linux 内核为底层的硬件抽象提供服务基本内存管理，线程管理，网络协议栈和设备驱动程序等。

B. 硬件抽象层

硬件抽象层用于抽象服务，由底层 Linux 内核层提供；它可以屏蔽底层实现细节。

C. 类库和 Android 运行时

Android 库提供了许多基本的 C / C++ 库，它们由应用程序调用。他们是最终的实

现在上层应用。 Android 运行时是一组提供 Java 编程语言的大部分功能的库。 包括 Dalvik 虚拟机。

每个进程都有自己的 Dalvik 虚拟机实例。它用于运行 dex 格式的可执行文件。

D. 应用框架层

应用程序框架层提供了一个开发平台为开发人员，这有助于重用和更换部件和建造各种应用程序。

E. 应用程序层

Android 应用层提供了一系列的包，其中包含大多数 Google 的应用程序，例如，时钟，日历等。他们是由 Java 编程语言写的。

III. 流媒体应用框架设计

一般来说，本地 native 代码比 Java 代码运行快。所以，鉴于流媒体应用的效率要求高的特点，以及 Android 系统分层的特点，流媒体引擎设计在 Linux 内核层和应用层之间，并通过 C / C ++编程语言实现。在流媒体引擎中，调用 Linux 内核和库的功能来实现流媒体编解码和传输功能。Android 应用层可以调用流提供的服务媒体引擎使用 JNI 接口。因为软件的可靠性与软件的分层数量成反比，因此整个结构设计为四层。流媒体应用的架构是如图 2 所示。

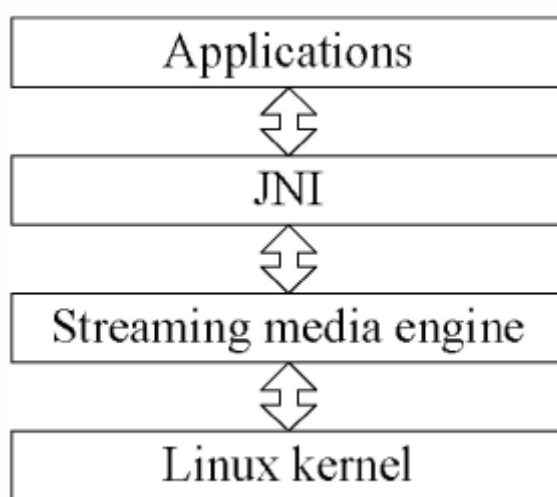


图 2 Android 流媒体应用开发框架

A. 流媒体引擎的总体设计

Android 平台的特性是层次化的设计，而模块化设计通常用于流媒体系统。组合层次和模块化设计，流媒体引擎分为五层，包括用户界面，数据采集，数据输出，编解码，和网络传输。整体框架如图 3. 这种设计策略可以简化媒体信息处理。它有助于开发人员开发并使用流传输来维护流媒体应用媒体引擎。这也是容易的 streaming 媒体应用程序扩展新的编解码器或传输协议。

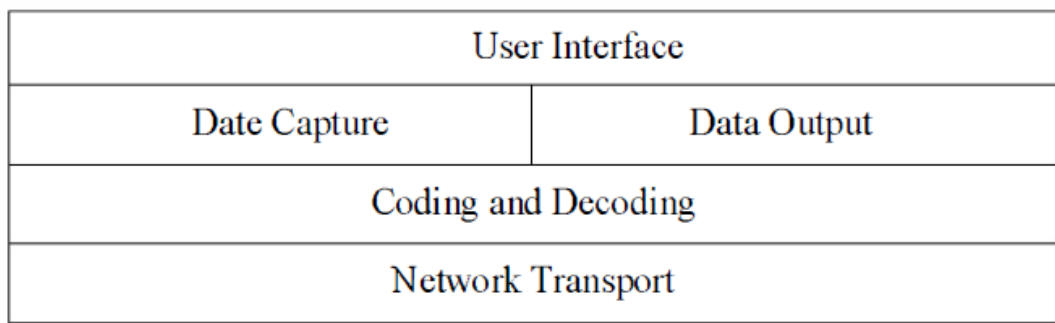


图 3 流媒体引擎框架

B. 数据捕获

数据捕获的主要任务是捕获音频或

视频数据。它定义了一个派生类，它提供了一些接口包括消息输入和输出接口，设备打开和关闭界面。开发人员只需要继承相应的类，然后根据我们想要的数据类型捕获，实现类的功能。事实上，调用设备相应的驱动程序函数接口实现初始化和打开音频或视频设备。open 函数结束时，线程开始捕获音频或视频数据。最后，输入功能可用于输入数据采集命令，然后通过线程捕获数据复制到消息数据区。当输出 0 功能为称为，我们可以得到包括音频或视频数据。

C. 数据输出

数据输出层的设计类似于设计数据捕获层。它还创建一个名为 DataOutput 的派生类，并提供设备打开和关闭接口，数据输出接口。在这一层中没有 output 函数，因为没有什么数据可以直接获得。首先，开发者需要继承相应的类，然后实现不同的输出设备和业务需求。当需要时使用，首先实例化。第二步，使用 open 函数，这是通过使用设备驱动程序来实现初始化设备，并创建音频或视频的输出生线程。在线程中，输出数据被输出输出设备用于播放音频或视频。将要输出的数据用 input 函数输入。

D. 代码层

代码层主要实现流媒体数据的压缩和解压缩，定义了一个编解码类，它具有初始化，编解码选择，数据输入输出接口。继承类后，开发人员可以调用 select 函数选择不同的编解码算法，并使用 avcodecinit 函数初始化代码。该算法通过不同的方式在 input 函数中实现。输出函数用于获取已处理的数据。解码处理是相反的。

E. 网络传输

网络传输层主要完成流媒体系统的传递功能；它创建了一个网络传输类，包括数据

输入和会话的初始化接口。根据要求，开发人员继承相关类并在 mIt 函数中初始化传输会话，然后使用 input0 函数，开发人员可以将数据传输到目标。多线程机制可以用来实现数据的传输。该信号用于与线程通信的过程。

F. 用户界面

用户界面层是流媒体的接口引擎，它是流媒体的 JNI 接口包发动机。通过 JNI 接口，Java 应用程序可以调用相应的流引擎功能。 API 实现参数控制，参数设置和查询接口。控制从一层到另一层的消息流。在本机函数和 Android 之间有一个 Java 函数映射表。它将注册到 Dalvik 虚拟机，当调用 systemloadLibrary 时。这样，时间消耗将减少。流媒体引擎用户界面层包含 start 和 stop 接口。当调用 start 函数时，消息从数据捕获通过编解码器发送到网络传输或从经由编解码器的网络传输到数据输出。可以通过配置一个 IS 配置文件。这些类在流媒体引擎中的每一层都是编译成使用 Android NDK 工具的动态库，其他人可以调用。这个的目的很容易升级和重用每个层。通过改变动态库，开发人员可以开发新的应用程序。

这些类的关系如图 4 所示。

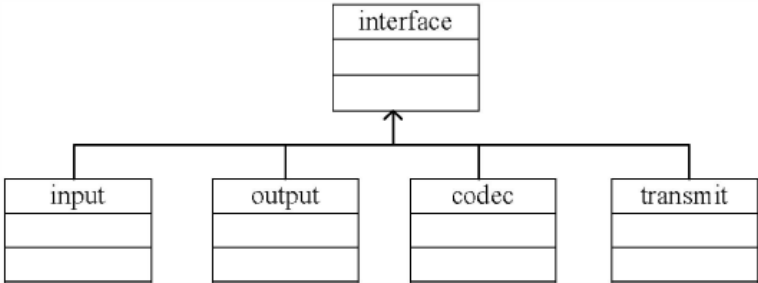


图 4 流媒体引擎中相关类之间的关系

IV. 构建流媒体应用的实例

智慧医疗器械是一种用来帮助医生救助病人的终端设备。救护车在到达医院之前，它传输从多通道生理监测数据以及图像或音频数据到控制中心进行保存和远程帮助。使用本文设计的流媒体应用程序开发框架，我们可以快速实现视频数据采集和传输功能的智慧医疗终端设备。

A. 开发方法

在智慧医疗终端的设计中，我们使用 H.263 编解码器的压缩算法，以及传输的 RTP 协议视频流。应用程序的开发步骤描述如下：

1) 要继承每一层中的相关类，使用 libav 库来实现编解码器类，并使用 jrtplib 库实现 RTP 传输协议，使用 Android Camera 类来实现视频采集，使用 Android AudioRecord 类实现音频采集。用户界面层的实现通过调用其他层实现类来完成；将这些实现的代码编译成动态的库通过使用 Android NDK 工具，生成流媒体引擎库命名为 libstreamingenginejni.so；

3) 在 Android 应用程序中，流引擎类是通过使用本地 native 函数语句创建。它调用流媒体引擎的接口。用户通过配置流媒体引擎，然后调用 start 函数启动处理取数据和输数据。这时候，我们实现了视频采集和传输功能。

B. 软件实现

该软件分为 C / C++ 层和 Java 层。在 C / C++ 层，它主要是流媒体引擎。在 Java 层，网络传输设置界面如图 5 所示。

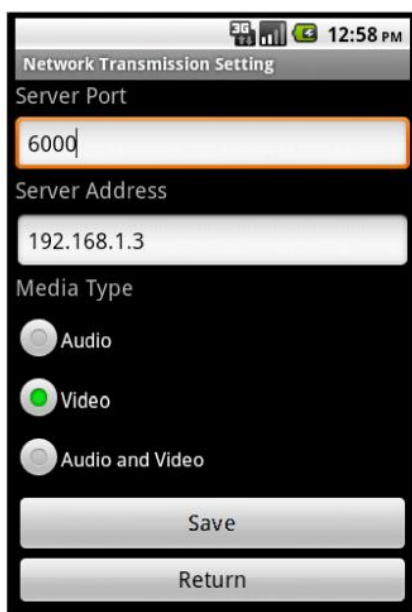


图 5 网络传输设置界面

远程控制中心服务器接收视频显示如图 6, 7 所示。图 6 是静态画面显示，图 7 是动态画面显示。



图 6 传输的静态画面



图 7 传输的动态画面

通过实验，本文设计的框架可以快速实现流媒体应用开发。它对于 Android 开发人员非常有用。

V. 结论

论文提出了一种 Android 中的流媒体应用程序开发的软件框架。该框架基于 Android 平台流媒体开发的不足之处, 通过实现流媒体引擎, 提供开发人员方便高效的开发框架, 并减少流媒体应用开发周期和复杂性, 提高软件可维护性。最后, 智慧医疗设备项目已经验证了框架的可行性。在未来, 该框架将会用于更具体的应用以增强其稳定性。

致谢

我们要感谢电子科技大学 Zhong Tian 对本论文的鼓励和支持。此外，我们还需要感谢嵌入式系统实验室的教师，特别是我的导师 Dan Tian。最后，我们需要感谢实验室指导我们以及帮助我们修改论文的学长。

参考文献

- [1] Li Zichen, "Streaming Media: Status, Prospect and Obstacle," information science, vol. 20, no. 9, pp.1005-1008, September 2002.
- [2] Song Bi-lian, Wu Hua-ping, Chen Jia-xun and Sun Li, "Research on Media Stream and Comparison between Its System Design Schemes," Application Research of Computers, vol. 21, no. 1, pp. 204-207, 2004.
- [3] [http://baike.baidu.com/view11241829 .htm](http://baike.baidu.com/view11241829.htm).
- [4] "DevGuideIAndroidDevelopers," <http://developer.android.com/guide/basics/what-is-android.html>.
- [5] Shanker, A. Lal, S, "Android porting concepts," 2011 3rd International Conference on Electronics Computer Technology, vol. 5, pp. 129-133, April 2011.
- [6] Cheng-Min Lin, Jyh-Horng Lin, Chyi-Ren Dow, Chang-Ming Wen, "Benchmark Dalvik and Native Code for Android System," Innovations in Bio-inspired Computing and Applications, 2011 Second International Conference on, pp. 320-323, December 2011.
- [7] Sangchul Lee, Jae Wook Jeon, "Evaluating Performance of Android Platform Using Native C for Embedded Systems," 2010 International Conference on Control, Automation and Systems, pp. 1160-1163, October 2010.
- [8] Huang Jin-guo, Luo Zhen, "Research on the Architecture of Mobile Application Development," Computer engineering and science, vol. 32, no. 11, pp. 141-144, November 2010.
- [9] Mo Dong-song, Peng Xiao-dong and Xia Ke-jian, 'Three Architecture Patterns for Multi-tiers Object-Oriented Applications," Application Research of Computers, vol. 20, no. 2, pp. 34-38, 2003.
- [10] He Dan-dan, Shi Zhan, "Design and Realization of VoIP Based on Android OS," Modern Electronics Technique, vol. 34, no. 6, pp. 28-31, March 2011.