
华中科技大学

毕 业 论 文 翻 译

论文题目：

Google's Android as an application environment for
DTV decoder system

院 系：计算机学院

专 业：计算机科学与技术

姓 名：苑雪冉

学 号：U200915132

指导老师：赵贻竹

日 期：2013 年 1 月 10 日

谷歌 Android 作为 DTV 解码器系统的软件环境

Nikola Kuzmanovic
Faculty of Technical Sciences
Novi Sad, Serbia
nikola.kuzmanovic@rt-rk.com

Tomislav Maruna, Milan Savic, Goran Miljkovic,
Djordje Isailovic
RT-RK d.o.o.
Novi Sad, Serbia

摘要一文提出了一种使用谷歌 Android 软件栈作为 DTV(电视集)和机顶盒应用环境的方法。通过扩展 Android 软件栈来支持 DTV 解码器设备,同时为 Android 环境开发万能电视和机顶盒应用,由此可加速软件的发展并且缩短下一代产品的发布周期。本文介绍了将完整的 Android 软件栈(包括 Android linux 内核,系统驱动程序,用户空间库,Android 特定的库和程序)移植到数字电视解码器系统上(主要是利用 Linux 内核和稳定的设备驱动程序)的过程,实现的扩展将支持数字电视频道搜索,记忆,播放和录音。最终的系统性能和特点由本文展现给我们。

关键字—数字电视机, 机顶盒, Linux 内核, 嵌入式系统, Android 软件栈

I. 简介

今天,DTV(数字电视)和机顶盒(进一步说是文本多媒体设备)已经由自定义的专用软件(操作系统和应用程序)来控制,一般说来,是一个特定的模式或者是一个家庭的同类模式。不同的多媒体设备模型之间的代码可重用性是很小的,因此一个多媒体设备模型的开发周期,需要大量的时间和资源。

手持设备和现代化的家庭多媒体设备之间的差异不断减少,同时,使用的硬件相似性正在变大(如 CPU 的处理能力, RAM 大小, 闪存大小以及其他集成设备的可用性如: LAN 和 Wi-Fi 无线局域网, 蓝牙, 硬件加速图形等)。虽然更新、更先进的功能不断地加入到它们中(如视频录制, 网络电视, 网上冲浪, 等等),

但在这一段时间内，手持设备和多媒体设备的主要目的并没有改变。

Android 是第一个为移动设备开发而建立的软件平台，它是完全开放和免费的，由开放手机联盟（以谷歌为首的超过 30 个公司的集团）发展起来。Android 在嵌入式市场变得越来越流行，主要原因有两个。首先，源代码是完全免费的；而且使用 Java 虚拟机没有特权使用费。第二，源于第一，Android 非常适合扩展和增强。

本文提出了一种使用谷歌 Android 软件栈作为嵌入式多媒体设备（主要是数字电视解码器系统和机顶盒）的应用环境的方法。进行 Android 的移植和修改的动力在于它的可用性（开放源码）以及它使用的是 Linux 内核的硬件抽象层（开放源码）。Android 软件栈和 Linux 内核，使开发人员能够访问多媒体，网络连接，图形，电话等，这使得在开发不同的 Android 设备上普遍使用的应用程序的过程中，不需要考虑特定的硬件系统上应用程序是否可以运行的问题。

II. 安卓软件栈

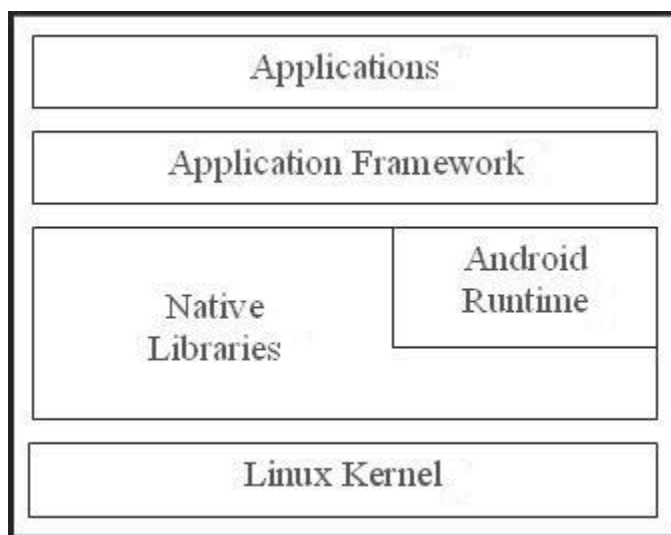


图 1 安卓系统架构

Android 的系统核心服务依赖于 Linux 内核，如安全性，内存管理，进程管理，网络协议栈和驱动模型。同时 Linux 内核也在硬件和剩余的软件栈之间扮演抽象层的角色。

Android 的原生库是用 C / C++ 开发的，并且被 Android 系统的各个组成部分所使用，通过 Android 应用程序框架将它的特性暴露给开发者。一些核心库如：系统 C 库（一个针对基于嵌入式 Linux 设备，用 BSD 派生实现的标准 C 系统库

(libc)), 媒体库 (基于 PureVideo OPENCORE, 主要针对流行音频和视频的播放和录制, 以及图像文件的处理), WebKit 库 (一个众所周知的现代 Web 浏览器引擎, 现在作为 Android 浏览器引擎), OpenGL ES 3D 的库 (这个库是使用硬件 3D 加速或者高度优化的 3D 软件光栅来实现的), FreeType 字体 (位图和矢量字体渲染), SQLite (一个强大的和轻量级的关系数据库引擎, 用于所有应用程序) 等。

Android 运行时包括了一套核心库, 这套核心库提供了大量 Java 核心库所具有的功能, Android 应用程序利用 Dalvik 虚拟机 (registerbased VM, 运行用 Java 编译器编译的类, 这些类被转化为 Dalvik 可执行 (.dex) 的格式, 并针对嵌入式设备进行了优化 (占用很小的内存空间)) 运行在单独的进程上。Dalvik 虚拟机基本功能的实现利用的是 Linux 内核, 如线程和低级别的内存管理。

Android 使开发者能够完全访问核心应用程序所使用的框架 API, 开发者可以自由免费地使用硬件设备、地理位置信息、运行后台服务、设置闹钟、添加通知、重用和替换现有的组件。简单的说, Android 提供了一个完全开放的免费开发平台, 使开发人员能够开发出先进的、功能强大的应用程序。

Android 的最高层是应用层, 这一层可容纳许多内置的应用程序如: SMS 程序, 电子邮件客户端, 日历, 地图, 浏览器, 电话等, 另外还包括从网上下载或通过 SD 卡安装的特定的应用程序。

III. 系统概述

这个项目的目的就是 will 将 Android 系统移植到一个真正的数字电视解码器 (Micronas 公司的 IDTV 开发平台) 设备上, 这个设备包括一个集成的系统芯片 (SoC) 和外围设备, 其中 SoC 包括了 MIPS24kc (240MHz 的 Linux 专用处理器), 解码处理器 (多路 A/V 专用), 图像处理单元 (GPU), 图像加速器 (GA), 通用串行总线 (USB) 控制器等。可用的软件有: 2.6.21.7 版本的 Linux 内核 (具有稳定的驱动程序, 适用于在 SoC 上的所有设备), GNU 的 gcc/ glibc 交叉编译工具, 支持 DirectFB 的图形库等等。

IV. Android 移植

移植 Android 系统到大端 MIPS 平台上需要三个步骤，利用的是 Micronas 平台和 2.6.21.7 版本的（这是一个在今天看起来很老的版本）Linux 内核。尽管如此，平台驱动程序所使用的框架，使得将它们移植到 Android 2.6.27 或者更高的内核版本上（相对于它的逆过程来说）变得更难（工作量超过了这个项目的预期），因此，在第一个阶段，Android 软件栈所使用的 Linux 内核驱动程序需要移植到该平台所提供的内核上，第二个阶段，则需要扩展 Android 编译环境以支持大端 MIPS 平台，在编译过程中，从环境得到的配置必须正确地编译。最后阶段，则包括音频驱动的集成和 DirectFB 视频帧缓冲的支持。

A. Linux 内核移植

为了支持 Android 软件栈，Linux 内核树需要集成一些新的驱动程序，这些驱动程序有：

- ASHMEM（匿名共享内存）——共享内存管理器，类似于 POSIX SHM，但有不同的行为，它支持简单的基于文件映射的 API。
- BINDER ——类似 CORBA 的 IPC 通信机制。
- Android event logger——Linux 内核驱动程序，直接在内核中记录调试信息。

为支持 Android 软件栈，2.6.21.7 版本的 Linux 内核树配置参数以及所使用的平台，需要在许多方面进行修改，并且，上述提及的新的驱动程序也需要集成。

B. Android 环境修改

Android 系统已经成功移植到了小端 MIPS 平台上，但是由于所使用的平台（大端平台）与小端平台的不同，对 Android 系统的修改还有很多。

- 第一种修改涉及平台本地的一些二进制文件，主要是开源库和用 C/C++ 编写的程序（endianes 支持通常是通过内部编译系统（主要是

“autoconf ”) 实现的), 同时修改 Android 编译系统的 makefiles 文件 (Android.mk) 来支持大端变量和定义。

- 第二种修改涉及编译工具的代码尾数不能识别的问题, 这造成的错误是罕见的, 并且很难发现, 它们通常会导致 Android 编译失败或者平台重启失败, 受影响的程序有 Android pre-linker Apriori、stripping application Soslim、resource manager aapt (Android 资源打包工具) 等。
- 第三种也是最后一种修改, 包括添加一些在 Android 上运行的程序和支持代码尾数识别的库, 例如 Dalvik VM (即明确支持大端的平台) 使用的一些数据类型的指针分配在小端体系结构中工作正常, 但是很难移植到大端平台上, 罪魁祸首是使用贯穿整个 VM 代码的 JValue 结构作为许多值类型容器的结构, 这是典型的误用小端系统中数据类型的指针分配, 在日志文件中找也到了相同类型的错误。其他的一些修改包括颜色问题的矫正, 尾数排序, SKIA 引擎, 像素和 Surface Flinger。

C. Android DirectFB 支持

DirectFB (直接帧缓存) 框架是一个基于 GNU / Linux/ UNIX、在 Linux 的帧缓冲设备之上的图形软件库 (占用很小的内存空间), 它提供了图形加速和输入设备处理等功能。由于它是开源的, 所以很多 DTV SoC 厂商用它来实现自己的帧缓冲 IP-S 驱动程序。

Android 支持用标准的 Linux 帧缓冲设备驱动程序作为主要的绘图表面。随着 DTV SoC 供应商实现硬件加速的 DirectFB 视频驱动程序, Android 系统对 DirectFB 的支持已经遍及系统不同层次。首先, 支持 UI 的绘制, 其次, 通过图形硬件加速来绘制图形。

- DirectFB 图形分配设备实现了支持主视频帧缓冲内存的分配, 以及所有其他种类的多颜色空间的表面分配。
- DirectFB 的 copybit 库还实现了支持硬件加速的 2D 位块传送操作。如果提供的 DirectFB 驱动程序支持 2D 位块传输操作 (不同颜色空间之间的 blit

操作和 stretch-blit 操作), 那么中央 CPU 单元将会从耗时的块传输任务过程中解放出来。

使用 DFB 作为 UI 图形缓冲的唯一缺点是它本身不支持 3D 加速 (Android 是可以使用的)。但是过去几年稳步发展的 DirectFB 图形库 (有尝试过添加像 3D 加速一样的 OpenGL) 使我们相信如果硬件 3D 加速的 IP 出现的话, 基于 DirectFB 图形库的 3D 硬件加速的实现是可能的。

V. 多媒体和 DTV 回放支持

这部分介绍了 Android 的运行环境和应用框架的修改和扩展, 以支持数字电视和多媒体设备控制。第一步将支持硬件音频和视频解码器, 并允许本地多媒体文件回放 (这一步是非常重要的, 因为 CPU 发现 DTV 及多媒体解码器设备没有那么强大来做软件多媒体解码, 而是使用集成多媒体硬件加速设备)。第二步将支持 DTV (数字电视) 频道搜索和记忆。第三步也是最后一步将增加对 DTV (数字电视) 频道回放和录制的支持。

A. 多媒体回放支持

Android 的 Java 运行库, 拥有一个媒体对象 “android.media”, 是一个用于控制播放的音频/视频文件和流的接口。Android 多媒体的实现在很大程度上依赖于一个开源软件 “OpenCore” 媒体管理库 (最初 Android 使用的是 PacketVideo), 它执行多媒体数据所有的收集、加工、解码和编码任务。而这种做法深深的依赖于 CPU 的处理能力 (这种处理能力是 DTV 和机顶盒平台很可能没有的)。

为了支持硬件加速多媒体回放, 媒体框架将扩展选项, 支持硬件处理、解码和呈现多媒体数据, 但是只能通过事件通知 Android (状态变化)。这将通过增加额外的硬件插件机制 (进一步称为 MMlib) 来实现。

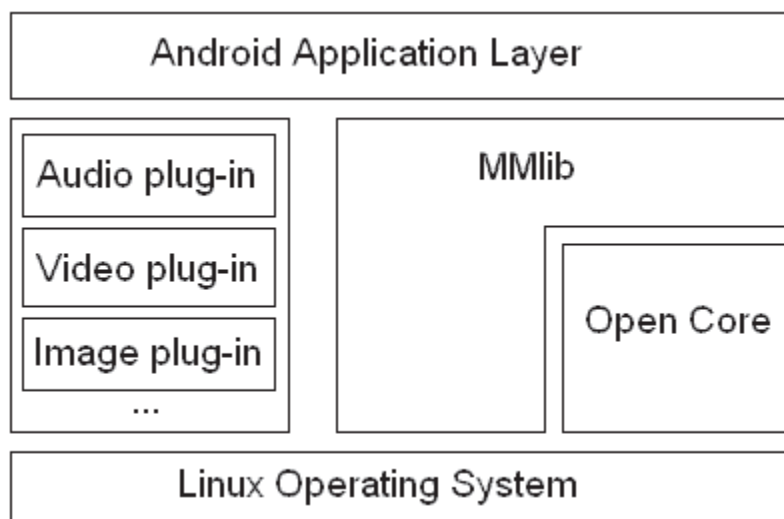


图 2 MMLib 库架构

MMLib 库作为硬件加速平台解码器和 Android 多媒体对象之间的粘合层，位于“android_media_MediaPlayer”JNI 对象中。MMLIB 查询接口的初始化通过（为硬件支持信息）安装硬件平台插件以及存储相应的播放接口（插件）来实现。使用所提供的支持信息，当媒体播放对象发送多媒体指令时，MMLib 解释处理此命令并选择最佳的播放接口来完成此次请求，如果没有硬件加速的支持，MMLIB 直接将命令传到 OPENCORE 的库（保留在 Android 中定义的默认程序）。

Micronas 平台支持多种视频和音频的格式，因此，基于这个项目的目的，对这些音频、视频和图像格式的支持将通过一些独立的插件来实现。

B. DTV 频道记忆及搜索

DTV（数字电视）和机顶盒通常包括各种前端装置，来提供模拟和数字通道再现。为了使信道能够搜索和记忆，在 Android 的编程方式中，我们引入了两个 Java 类（一个利用对象的过程服务），一个用户接口和一个 JNI 接口对象，以支持本地信道服务。

引进的对象包括：

➤ 信道服务线程对象

这个信道服务线程运行在后台进程中，使用了 JNI 接口。这项服务的主要功

能是收集和修改有关支持平台的前端设备信息以及这些设备的能力。这项服务的辅助功能是异步提供对手动和自动搜台的支持，以及支持收集记忆频道中广播数据所提供的各种频道服务，另外还支持网络上提供这方面信息的插件。通过它所维护的 ChannelServiceDatabase 对象来存储和使用信息。

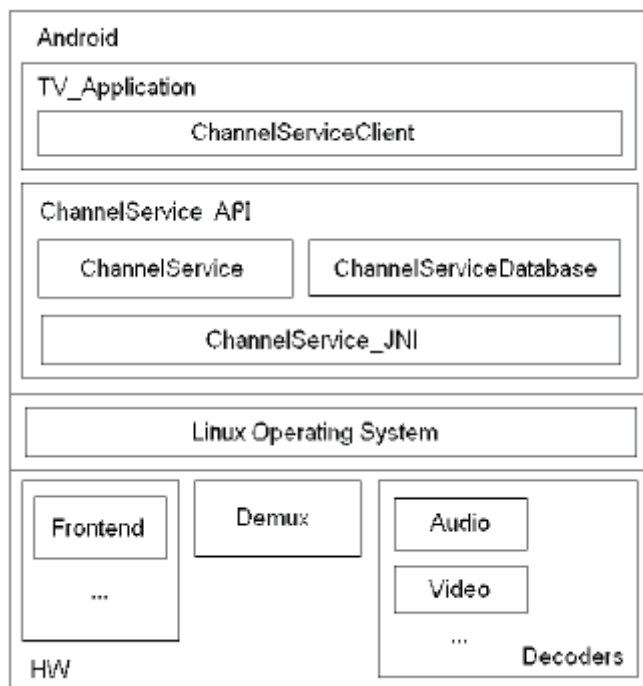


图 3 信道服务架构

➤ ChannelServiceDatabase 对象

信道服务数据库提供了一个高度灵活的方式来存储关于所有不同信道类型的信息（DVBT，DVB-S，DVB-C，，Analogue，Radio 等，以及他们所有的子类型和将来的类型）。结合使用 SQL 数据库，Android 提供了一个很好的框架来灵活地存储数据。ChannelService -Database 对象对最终用户并不是直接可见的，它的内容需要通过 ChannelServiceClient 接口来浏览和修改。

通道服务数据库的实现只提供了频道搜索和记忆的支持，但不是所有的 Android 设备都能够识别。识别的支持是通过通道数据库与本地的 android 文件系统同步，以及利用信道所提供的文件来实现的，这些文件有一个已知的 MIME 类型，这些文件解析为一个新的多媒体对象，且他们代表了 Android 多媒体框架

的说明。

➤ **ChannelServiceClient 接口**

通道服务客户端接口为通道服务子系统提供了一个入口，它实现了一些函数和对象，这些函数和对象是前端、通道和通道服务管理在控制用户使用它们的方式的时候常用的。因为 Android 同时可以有多个 TV 程序在运行，所以 ChannelServiceClient 接口必须实现异步方式来访问 ChannelService 子系统。

为了使用通道服务子系统，Android 应用程序只需要实现 ChannelServiceClient 这个接口。

➤ **信道服务 JNI 接口**

该接口提供了本地平台的前端框架和 Java 信道服务之间的联系。它提供了查询平台的能力和获取前端、信道、信道服务信息的能力。这个接口编译连接成一个共享对象。它从 Java 程序中动态打开，以检查平台是否已支持信道服务。

C. DTV 频道的回放及录制

通过扩大媒体对象框架支持 MMlib 库的能力，并通过定义通道服务的多媒体文件，DTV 设备的控制和回放的实现，仅仅是一个 MMlib 的插件问题。

DTV 回放和录制的 MMlib 插件的实现依赖于平台，在 Micronas 平台上 DTV 接口是直接通过内核驱动程序来实现的，所以实现 DTV 插件是很简单的，因为频道设置的指令，已经被频道服务的多媒体文件所接收。

VI. Android TV 应用

Android 平台没有 DTV 播放应用，然而，从之前的段落中可以看出，由于对 Android 软件栈的修改和补充，Android 对 DTV 播放是支持的。

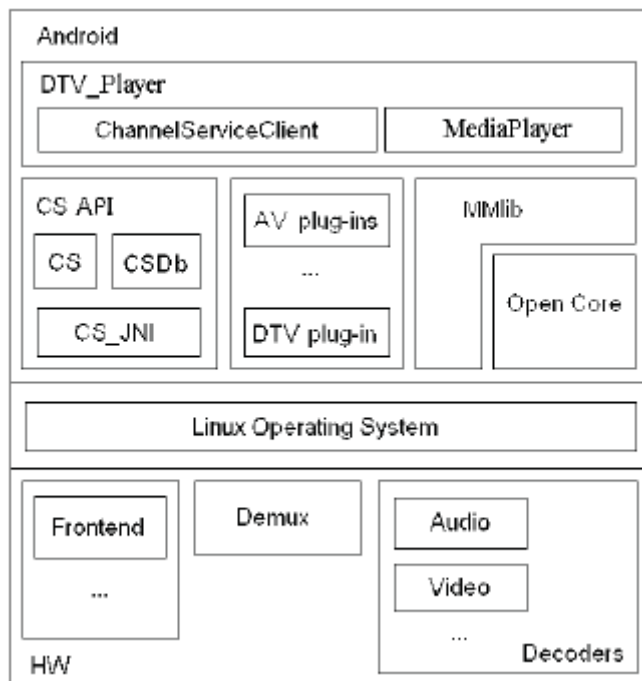


图 4 DTV 播放器应用程序概述

为了达到测试所有 DTV 平台能力的目的，实现了 DTV 播放器。

七、测试结果

Android 平台是专为移动设备而设计的，而且它还是一个正在广泛发展的软件平台。然而，当把它正确的移植在新系统上，它将自动成为能够执行的内置应用程序。尽管是为小屏幕的手持移动设备设计的，但是移植 Android 操作在测试型的 DTV(数字电视)解码器设备上是可接受的。假设提供的测试平台有一个相当缓慢的 CPU，GUI 速度也是很少缓慢。这对于最新的机顶盒是没有问题的，因为它使用的 Micronas 平台有多达两倍的功率。

为了支持数字电视频道的搜索和回放，修改和扩展的 Android 软件栈已经用数字电视播放器应用程序进行了测试，测试包括频道的搜索、记忆、播放和录制。

记录在[6][7][8]的平台和目前的商业解决方案，多媒体只有有限数量的选项提供给用户，相对于他们来说，Android 驱动平台能够让用户添加选项，修改的选项，换句话说，Android 驱动平台将完全改变用户使用嵌入式多媒体设备的方式。

八、结论

今天, Android 并不打算只用在 DTV 平台上。然而, 随着它的修改和扩展, 它将为新的嵌入式多媒体设备提供非常好的软件环境。作为一个开源平台, Android 可通过扩展、内核的修改、新应用的开发, 为我们带来几乎无可限量的可能。本文为多媒体设备的开发提供了一种新的方式。在 Micronas IDTV 平台上, 通过为 Android 使数字电视服务, 再加上内置的 Android 应用程序 (Web 浏览器, JAVA 游戏, 图片, 音频和视频管理等), 电视机的使用方式将彻底改变。在嵌入式多媒体系统上使用 Android 软件栈 (或一些未来的 Android 衍生物) 在不久的将来将成为现实, 这将整体提升现代多媒体系统的体验。

参考文献

- 【1】谷歌 Android, <http://www.android.com>
- 【2】Dalvik 虚拟机, <http://www.dalvikvm.com>
- 【3】分析 Dalvik 虚拟机和类路径库,
<http://imsciences.edu.pk/serg/wp-content/uploads/2009/07/Analysis-of-Dalvik-VM.pdf>
- 【4】Micronas IDTV 开发平台,
http://www.micronas.com/pressroom/press_releases/articles/0812/index.html?newslang=1
- 【5】MIPS Android, <http://www.mips.com/android>
- 【6】Lalit Kumar, Rajesh Kushwaha, Rishi Prakash, “设计与开发基于数字机顶盒浏览器的小型 Linux 操作系统”, 2009 年第一次关于计算智能, 通信系统和网络的国际会议, 277-281 页
- 【7】李琼, 郭民强, “基于嵌入式 Linux 操作系统的数字可记录综合电视“, 2009 年世界大会关于计算机科学与信息工程, 81-84 页
- 【8】Ganesh Sivaraman, Pablo Cesar, Petri Vuorimaa “系统软件数字电视应用“, 2001 年 IEEE 国际会议多媒体与博览会 (ICME'01), 第 154 页
- 【9】MIPS Android 源代码,
<http://www.linuxfordevices.com/c/a/News/MIPS-to-release-Android-source-code-by-August/>