

Android 嵌入式系统架构及内核浅析

A Brief Analysis on the Framework and its Kernel of Android Embedded System

王 茜

(广州私立华联学院计算机科学与技术系 广州 510663)

【摘 要】对当前流行的Android 系统平台架构按层次解析,分析了各层的组成与特点,并对基于Linux 内核的系统核心分析了其新增和修改的内核驱动,新的内核支持各种基于Android 平台的外设和对强大网络功能的需求。

【关键词】Android, Linux, 嵌入式系统, 内核

中图分类号: TP391

文献标识码: A

ABSTRACT This paper introduces the current popular Android embedded system, analyzes every layer's composition and characters, and analyzes the new added and modified kernel drivers which based on Linux kernel, the new kernel supports all sorts of devices running on Android and the powerful network function.

KEYWORDS Android, Linux, embedded system, kernel

Android 是 Google 公司开发的基于Linux 平台的开源手机操作系统,它包括操作系统、中间件、用户界面和应用程序,而且不存在任何以往阻碍移动产业创新的专利权障碍,并由 Google 公司于 2007 年 11 月 5 日正式发布。同时,Google 公司组建了一个开放手机联盟,这个联盟由中国移动、摩托罗拉、高通、宏达电和 T-Mobile 等在内的全球 30 多家技术和无线应用的领军企业组成,Google 通过与运营商、设备制造商、开发商和其他有关各方结成深层次的合作伙伴关系,希望借助建立标准化、开放式的移动电话软件平台,在移动产业内形成一个开放式的生态系统;可预见地,生产和使用基于Android 系统的嵌入式手持移动设备将是未来的发展趋势,对相应软件的需求量也将日趋增长,因此对 Android 系统内部作一个完整和深入的分析,对基于 Android 平台的软件移植和开发是很有益处的。

1 Android 系统平台架构

对操作系统而言,必须做到设计合理、层次分明,同时还需考虑整个系统的结构要聚耦适当,Android 系统是基于 linux 内核的,因此还必须具备开源的特性,以符合开源人员共同工作。

从系统的组成要件来讲,Android 平台架构包括硬件设备、板级支持包、驱动程序、操作系统内核、程序运行库,运行框架,应用程序等,它们的有机结合和协同工作共同完成了整个系统的正常运行和对事务的处理。

依据 Google 开源资料可知,整个系统由Linux 内

核、程序库、Android Runtime、应用程序框架和应用程序等 5 部分组成,系统架构如图 1 所示。



图 1 Android系统平台架构

参照图 1,由上而下对组成系统各部分的主要组件作以下描述。

1.1 Linux 内核

Android 基于Linux 2.6 内核,但并非完全照搬内核,而是对内核作了部分增删和修改,在Linux 2.6 内核的基础上,Android 核心系统实现了安全性、内存管理、进程管理、网络协议栈和驱动模型等功能,Linux

内核也同时作为硬件和软件栈之间的抽象层。

硬件驱动程序: 完成与各种硬件的通信, Linux 内核提供了大部分设备的驱动程序, 如显示屏, 摄像头, 内存, 键盘, 无线网络, 音频设备, 电源等组件。

系统内存管理: 对所有可用的内存进行统一编码管理, 定义一整套内存定位, 使用与回收的策略。

系统进程管理: 内核管理进程的创建与销毁, 管理进程间的通信, 以及采取必要的措施避免死锁等内容。

网络管理系统: 无线网络设备工作原理, 内核掌控如何读取网络设备中的缓存数据。

1.2 程序库

程序库是指可供使用的各种标准程序、子程序、文件以及它们的目录等信息的有序集合, Android 包含一些 C/C++ 库, Android 系统中不同的组件通过应用程序框架可以使用这些库, 以下是一些核心库:

Surface Manager: 管理显示子系统, 并且为多个应用程序提供 2D 和 3D 图层的无缝融合;

Media Framework: 基于 OpenCORE 的多媒体框架, 支持多种常用的音频、视频格式文件的回放和录制, 同时支持静态图像文件。

SQLite: 一个对于所有应用程序可用, 功能强劲的轻型关系型数据库引擎。

OpenGL ES: 3D 图形库, 用于 3D 图形渲染, 该库可以使用 3D 硬件加速。

FreeType: 位图 (Bitmap) 和矢量 (Vector) 字体显示。

WebKit: 支持 Android 浏览器和一个可嵌入的 Web 视图。

SGL: 2D 图形库, 用于 2D 图形渲染。

libc: 一个从 BSD 继承的标准 C 系统函数库, 它是专门为基于嵌入式 Linux 设备定制的。

1.3 Android 运行库 (Android Runtime)

Android 运行库包括两部分: 一是核心库, 二是自身的虚拟机。

核心库提供 Java 编程语言核心库的大多数功能。Dalvik 虚拟机是 Google 专为 Android 开发的, 比 Sun Java 虚拟机的效率更高, 功能也更为复杂, 以更好的支撑 Android 平台, 并拥有独立的版权。每一个 Android 应用程序都在自己的进程中运行, 都拥有一个独立的 Dalvik 虚拟机实例, Dalvik 虚拟机执行 .dex 的可执行文件, 该格式文件针对小内存的使用进行了优化, 同时虚拟机是基于寄存器实现的, 所有的类由 Java 编译器编译, 然后通过 SDK 中的相应工具转化成 .dex 格式, 最后由虚拟机执行。

1.4 应用程序框架

应用程序框架是指定义了一个应用程序运行所必须的全部功能组件, 开发者也可以访问核心应用程序所使用的 API 框架。该应用程序的架构设计简化了组件的重用; 任何一个应用程序都可以发布它的功能块, 并且任何其他的应用程序都可以使用其所发布的功能块 (应该遵循框架的安全性限制)。同样, 该应用程序的重用机制也使用户可以方便地替换程序组件。

隐藏在每个应用后面的是一系列的服务和系统, 其中包括:

活动管理器 (Activity Manager): 用来管理应用程序生命周期, 并且提供常用的导航回退功能。

丰富而又可扩展的视图 (Views): 可以用来构建应用程序, 它包括列表 (Lists)、网格 (Grids)、文本框 (Text Boxes)、按钮 (Buttons), 甚至包括可嵌入的 Web 浏览器。

内容提供者 (Content Providers): 使得应用程序可以访问另一个应用程序的数据, 或者可以共享它们自己的数据。

资源管理器 (Resource Manager): 提供非代码资源的访问, 如本地字符串、图形和布局文件 (Layout Files)。

通知管理器 (Notification Manager): 使得应用程序可以在状态栏中显示自定义的提示信息。

Android 程序框架体现了“统一”与“兼容”的原则, 其优势归纳为:

任何一个应用程序都可以发布它的功能块。

所有的应用程序在 Android 平台上都是平等的。

所有的应用程序与资源都被按类别进行分别管理。

所有程序各司其职。

1.5 应用程序

Android 系统发布时, 会同一系列核心应用程序和常用程序一起发布, 如常用的手机功能程序, 包括语音电话、通讯录、短信收发、照相、话机设置等; 数据应用程序, 包括邮件工具、日程表、浏览器、地图导航等, 以及 Android Market 上的各种应用程序; 所有的应用程序都是使用 Java 语言编写。

2 Android 内核

Android 内核基于 Linux 2.6 内核作了部分修改和增删, 是一个增强内核版本, 除了修改部分 Bug 外, 它还提供了用于支持 Android 平台的设备驱动, 与标准的 Linux 内核比较, 分析可知其内核的全貌和基本结

构,下面对基于Android修改的核心驱动作初步阐述:

2.1 系统进程管理

Android平台的进程间通讯基于Binder机制实现,它提供了一种进程间通信的方法,即一个进程可以以类似远程过程调用的形式调用另一个进程所提供的功能,然后带着执行的结果返回。Android Binder是基于Service与Client的,有一个ServiceManager的守护进程管理着系统的各个服务,它负责监听是否有其他程序向其发送请求,如果有请求就响应,每个服务都要在ServiceManager中注册,而请求服务的客户端去ServiceManager请求服务。

2.2 内存管理

低内存管理器(Low Memory Killer):相对于Linux标准OOM(Out Of Memory)机制更加灵活,它可以根据需要杀死进程来释放需要的内存。

匿名共享内存(ashmem):为进程间提供大块共享内存,同时为内核提供回收和管理这个内存的机制。

Android PMEM(Physical):PMEM用于向用户空间提供连续的物理内存区域,DSP和某些设备只能工作在连续的物理内存上。

2.3 文件系统管理

Android平台采用Yaffs2作为MTD nand flash文件系统,Yaffs2是一个快速稳定的应用于NAND和NOR Flash的跨平台的嵌入式设备文件系统,同其他Flash文件系统相比,Yaffs2使用更小的内存来保存它的运行状态,因此它占用内存小;Yaffs2的垃圾回收非常简单而且快速,因此能达到更好的性能;Yaffs2在大容量的NAND Flash上性能表现尤为明显,非常适合大容量的Flash存储。

2.4 时钟管理

Android Alarm,提供了一个定时器用于把设备从睡眠状态唤醒,同时它也提供了一个即使在设备睡眠时也会运行的时钟基准,Android timed device,提供了对设备进行定时控制功能,目前支持vibrator和LED设备。

2.5 电源管理

Android电源管理(PM),一个基于标准Linux电源管理系统的轻量级的Android电源管理驱动,针对嵌入式设备做了很多优化。

2.6 USB管理

Android的USB驱动是基于gadget框架的,USB Gadget驱动是一个基于标准Linux USB gadget驱动框架的设备驱动。

2.7 系统日志管理

Android Logger,一个轻量级的日志设备,用于抓取Android系统的各种日志。

3 结束语

Android是一款开放的、基于Linux内核的嵌入式操作系统,将被广泛地应用到各种移动手持设备中,充分发挥其强大的网络功能;由于开源的特性,Android不区分核心应用程序和第三方应用程序,并可自由的吸纳新出现的尖端技术;同时,Android提供广泛而实用的函数库和工具,利用它们可以开发丰富的应用程序。由此可以展望,基于开放手机联盟(Open Handset Alliance)平台强大的技术实力和市场推广,Android系统在移动嵌入式领域的应用将会非常广阔。

参考文献

- [1] 盖索林 Android开发入门指南[M].北京:人民邮电出版社,2009
- [2] 杨文志 Google Android程序设计指南[M].北京:电子工业出版社,2009
- [3] Android平台技术架构[EB/OL].<http://book.51cto.com/art/200912/173932.htm>,2009-03-07.
- [4] 手机操作系统之Android[EB/OL].<http://new.scsdn.net/a/20090513/211233.html>,2009-08-09.

下期要目

灰色系统理论预测算法GM(1,1)模型研究及其在网络访问控制技术的分析与比较
高校网络课程建设研究
图像层次语义描述的初步研究
无线传感器网络时间同步技术
测试用例集的约简方法
基于SOA的B/S与C/S混合开发模式
两种典型图布局算法的实验性对比
基于XML的历史数据归档与重构策略
结合人工鱼群的动态克隆选择算法
ZigBee路由算法的研究与改造
基于Google云计算的Web应用与开发
蛙跳算法在WEB文本聚类技术中的应用
基于WebGIS的电力抢修调度系统的研究与应用
WEB离线应用程序的开发
地下水模拟系统的设计与实现
直驱式外转子永磁同步风力发电机计算机辅助设计
基于C#的Word2010文档自动生成系统
基于特征点的图像拼接算法研究与实现

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)

10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 CC++语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)

RT Embedded <http://www.kontronn.com>

2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)

WeChat ID: kontronn

RT Embedded <http://www.kontronn.com>

7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)