

基于 ARM 的双操作系统

苏建志^{1,2}

- (1. 泉州师范学院物理与信息工程学院, 福建 泉州, 362000;
2. 信息功能材料福建省高校重点实验室, 福建 泉州, 362000)

[摘要] 针对现阶段单操作系统的智能手机和平板电脑等手持设备, 提出一种基于 ARM 的双操作系统. 其主操作系统为手持设备本身的嵌入式操作系统, 以二进制文件保存, 不对开发者开放, 通过添加从操作系统和虚拟机, 把从操作系统完全开放给开发者. 实现了开发者可以通过从操作系统利用手持设备本身的硬件资源, 不需要添加任何硬件电路, 直接实现对手持设备的二次开发.

[关键词] 虚拟机; 操作系统; 单核

[中图分类号] TP 311.1

[文献标志码] A

Two Operation Systems Based on ARM

SU Jian-zhi^{1,2}

- (1. College of Physical and Information Engineering, Quanzhou Normal University, Quanzhou 362000, China;
2. Key Laboratory of Information Functional Material for Fujian Higher Education, Quanzhou 362000, China)

Abstract: Now generally smart phone or Pad is embedded RTOS (real time operation system) which is based on single core or multi code ARM. They are all based on single OS. This paper brings up one method which the two OS base on ARM core. The main OS is the embedded RTOS which is provided by hand device. And it does not open to developer. By adding the secondary OS and virtual machine, at the same time the secondary OS will fully open to developer. It realizes that the developer can control the hardware source of hand device directly by secondary OS and do not need to add any other hardware. Thus will realize the secondly development for the device without adding any new hardware circuit.

Key words: virtual machine; operation system; single core

0 引言

一般来说, 智能手机和平板电脑等手持设备是基于 ARM 单核或者多核的嵌入式实时操作系统, 它们都是单操作系统^[1-5]. 其与硬件资源密切相关的底层软件都是以二进制存储于 Flash 中, 没有对开发者开放系统的外部接口等硬件资源, 同时也没有开放相关的底层软件^[6-10]. 这种基于 ARM 单核或者多核的单嵌入式实时操作系统的手持设备不利于开发者进行二次开发^[11-15], 尤其是无法充分利用手持设备本身丰富的硬件资源进行二次开发. 当需要利用手持设备本身的外部硬件接口资源时, 只能另外添加相应外部硬件接口, 同时需要添加一个 MCU (单片机) 来对这些外部硬件接口进行控制,

[收稿日期] 2014-12-12

[修回日期] 2015-01-21

[基金项目] 信息功能材料福建省高校重点实验室项目 (1009-YA0206); 泉州师范学院校自选项目 (2014KJ17)

[作者简介] 苏建志 (1980—), 男, 硕士, 从事芯片底层及其对应软件的架构、通信和计算机底层软件、无线系统射频设计的研究.

并且还要建立 MCU 与 ARM 核之间的通信与信息交互。

因此，有必要开发基于 ARM 的双操作系统，使开发者不需要增加任何硬件即可直接对手持设备进行二次开发。

1 双操作系统和虚拟机

双操作系统框架如图 1 所示，在 Flash 的一个分区中存放主操作系统应用层及其内核，其中内核要把本身的硬件驱动虚拟化，包括 IO 口的虚拟化、外部中断的虚拟化、SPI 和 UART 的虚拟化、I2C 的虚拟化、定时器的虚拟化、模数转换的虚拟化等等；在 Flash 的另外一个分区中存放从操作系统应用层及其内核。从操作系统应用层及其内核把控制和操作命令传递给虚拟机，接着虚拟机再传递给主操作系统。主操作系统接收到从操作系统命令后，先解析命令，然后去驱动已经虚拟化的硬件设备。这就实现了从操作系统控制手持设备系统硬件的过程。因此，基于 ARM 的双操作系统首先要把 Flash 重新分区，使之能够同时装载主从操作系统和虚拟机；接着需要把主操作系统硬件驱动虚拟化，使之能够与从操作系统共享硬件资源；最后需要建立主从操作系统之间的通信，使得开发者通过从操作系统下命令给主操作系统，主操作系统才能对手持设备的硬件资源进行操作和控制，从而实现开发者直接控制手持设备的硬件，对手持设备进行二次开发的目的。

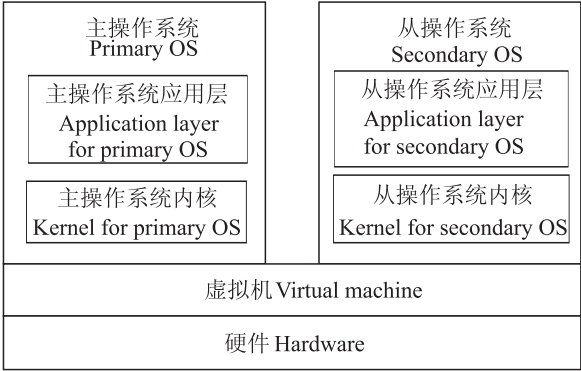


图 1 双操作系统框架
Fig. 1 System structure of two OS

2 主操作系统

为了能够在 ARM 单核中运行双操作系统，需要对主操作系统的 Flash 重新分区，使之能够同时容纳主从操作系统以及虚拟机，同时需要把主操作系统的内核硬件虚拟化，使得从操作系统可以通过虚拟机对主操作系统下命令，从而实现从操作系统利用和控制系统硬件资源的目的。

2.1 主操作系统的 Flash 重新分区

为了能够同时运行主从操作系统和虚拟机，不仅需要加大 Flash 存储空间，使得在 Flash 中能够同时存储主从操作系统和虚拟机，还要对 Flash 重新进行分区，使得 Flash 的不同分区存储不同的系统。一般来说 Flash 包含有主信息区、失败保护区、客户标志区、虚拟机区、主操作系统区、从操作系统区以及嵌入式文件系统区，可以根据所加的从操作系统以及虚拟机的大小来调整 Flash 的大小以及每个区的容量。假设 Flash 的每块大小为 128 KB，典型的每个分区占有的块大小和空间如表 1 所示。

表 1 Flash 重新分区
Tab. 1 Re-partition of Flash

标识符 ID	描述 Description	块开始 Start block	块结束 End block	空间/KB Space	备用空间/KB Reserve space	总空间/KB Total space
MIB	主信息区 Main information block	000	00A	768	512	1280
FPB	失败保护区 Fail protection block	00A	00C	128	128	256
CUS	客户标志区 Customer mark block	00C	026	2944	384	3328
VLX	虚拟机 Virtual machine	02C	03E	2048	256	2304
PS	主操作系统 Primary OS	03E	136	30720	1024	31744
SS	从操作系统 Secondary OS	136	1D0	3072	1024	19712
EFS	嵌入式文件系统 Embedded file system	1D0	278	20480	1024	21504

2.2 主操作系统内核硬件驱动虚拟化

主操作系统内核硬件驱动虚拟化包含 VGPIO（外部 IO 口虚拟化）、VPIC（外部中断虚拟化）、VSPI（SPI 总线虚拟化）、VI2C（I2C 总线虚拟化）、VUART（UART 串行总线虚拟化）、VADC（ADC 模拟数字转换接口虚拟化）和 VTIME（定时器虚拟化）。整个内核硬件驱动虚拟化框图如图 2 所示。虚拟化后，从操作系统通过应用层下命令，经过系统内核解析以及虚拟机传送命令，最终到达主操作系统；主操作系统根据主从操作系统定义的协议来解析命令，下达命令到虚拟化的外设接口，最终调用硬件抽象层驱动手持设备的硬件。这就达到了本研究的目的：从操作系统顺利操作和控制主操作系统硬件，同时虚拟化的外部接口驱动返回操作硬件的状态结果给从操作系统。

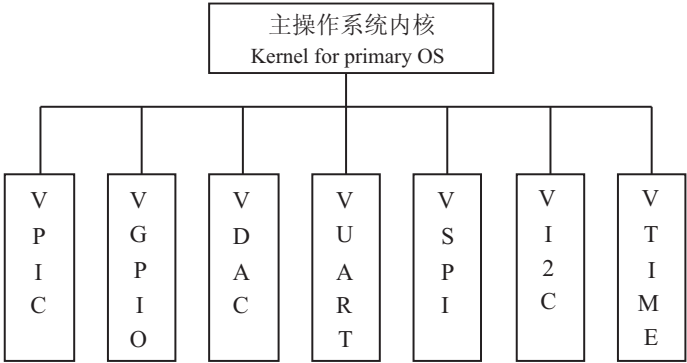


图 2 主操作系统内核硬件驱动虚拟化

Fig.2 Hardware driver virtualization for the kernel of primary OS

3 从操作系统

为了使开发者能够对手持设备进行二次开发，需要添加从操作系统，同时添加从操作系统的命令处理流程以及对手持设备硬件资源进行操作的程序，主要是中断响应处理、IO 口处理和 I2C 总线处理等。这样开发者可以通过从操作系统发送相应的命令给主操作系统，从而直接实现手持设备硬件资源的二次开发。

3.1 从操作系统中断响应流程

从操作系统中断响应流程如图 3 所示。如果手持设备的硬件中断触发，主操作系统的 VPIC 就会先关中断，同时把这个中断通过虚拟机传递给从操作系统，从操作系统接收到中断信号去处理中断响应程序，处理完毕过后重新使能中断，把开中断状态通知主操作系统，此时主操作系统通过 VPIC 来重新打开中断触发器，如果有新的中断到来，又重新开始一次新的中断响应。因此整个流程如图 3 所示。

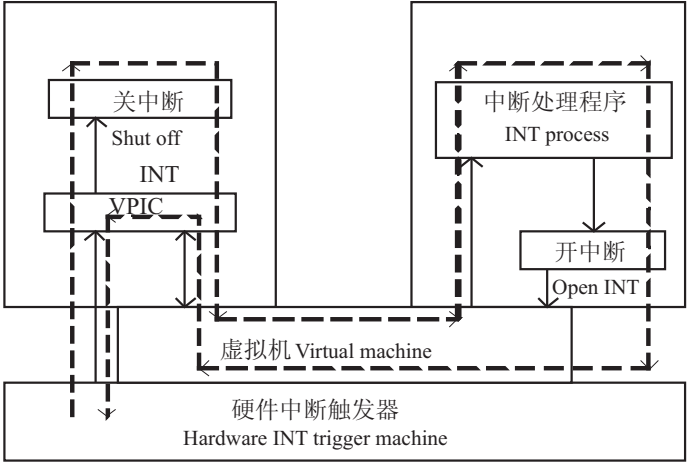


图 3 从操作系统中断响应流程

Fig.3 Flow of interrupt response for secondary OS

3.2 从操作系统 I2C 总线

从操作系统操作控制 I2C 总线流程如图 4 所示。

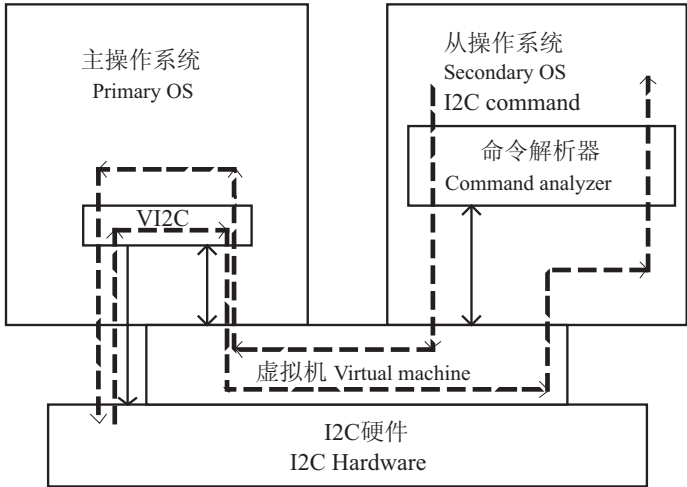


图 4 从操作系统操作控制 I2C 总线

Fig.4 Secondary OS control I2C bus

当从操作系统需要对 I2C 总线接口进行读写的时候，首先可以通过从操作系统应用发送开 I2C 总线命令，该命令经过虚拟机转发给 VI2C 驱动器，然后 VI2C 驱动会去打开 I2C 总线和切换相应的硬件资源给从操作系统，同时发送开 I2C 总线成功的状态标志给虚拟机，虚拟机把状态转发给从操作系统，此时从操作系统就成功打开 I2C 总线，然后再根据需要发送读写 I2C 总线命令给主操作系统，就可以顺利实现从操作系统的 I2C 总线读写。SPI 总线和 UART 串口总线的控制与操作流程和 I2C 总线一样。

3.3 从操作系统 IO 口读写

从操作系统 IO 口读写流程如图 5 所示。当从操作系统需要对 IO 口进行读写操作的时候，首先通过从操作系统应用层发送 IO 口查询命令，该命令经过虚拟机转发给主操作系统 VGPIO 驱动器，然后 VGPIO 会自动判断该 IO 口是否要被占用，如果被占用就将 IO 口处于不可用状态的信息返回给从操作系统，此时从操作系统会告诉用户该 IO 口被占用，不能对 IO 进行读写操作；如果返回的是该 IO 口处于可用状态，则从操作系统接下来即可发送 IO 读写命令，然后主操作系统通过虚拟机得到该 IO 口命令，再根据命令要求对 IO 口进行读写操作，同时把操作结果返回给从操作系统，从操作系统就完成了对 IO 口的读写。ADC 和 GPIO 的控制和操作流程一样。

3.4 从操作系统的发送命令处理流程

从操作系统的发送命令处理流程如图 6 所示。从操作系统发送命令处理流程主要是从操作系统发送命令，然后虚拟机转发命令，最后主操作系统处理命令，以及将命令结果返回给从操作系统。

4 结束语

综上所述，通过添加从操作系统和虚拟机，使得手持设备的嵌入式主操作系统以二进制文件的形式保存在 Flash 中，实现了保密功能；同时给开发者提供从操作系统，使开发者不需要添加任何硬件，只通过从操作系统和虚拟机对手持设备硬件资源直接操作和控制，就

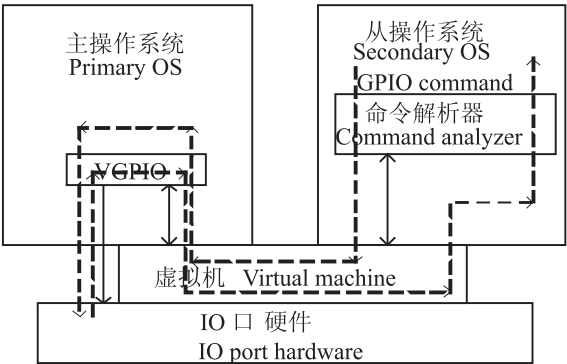


图 5 从操作系统读写 IO 口
Fig.5 Secondary OS read/write IO port

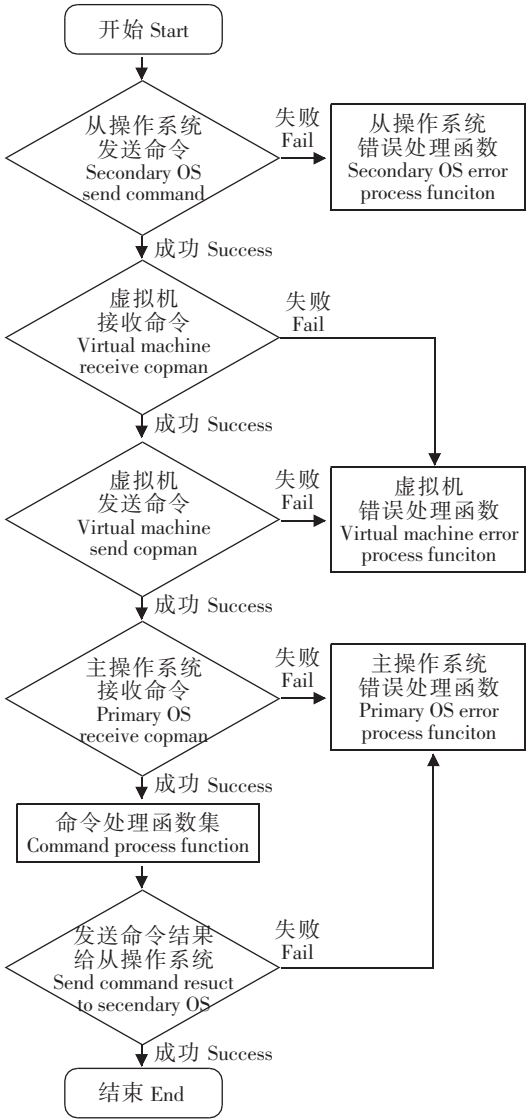


图 6 从操作系统发送命令处理流程

Fig.6 Flow of command processing sent by secondary OS

实现了对手持设备的二次开发. 该双操作系统测试结果运行良好, 可靠性强.

实践证明该方案能够大量缩短开发时间, 节省硬件开发成本. 深圳亚高乐无线科技有限公司曾经用此方法给客户开发过 3G 无线数据卡, 客户在该数据卡上面没有添加硬件, 直接在无线数据卡上进行二次开发, 有效节省开发时间和硬件成本, 产生了良好的经济效益.

[参 考 文 献]

- [1] DINH - DUC, ANH - VU, HO N. A run - time detector for violated memory access in embedded systems [C] //Mechatronics and Embedded Systems and Applications (MESA), 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. Qingdao, Shandong, China: IEEE, 2010: 217-223.
- [2] YAMADA S, NAKAMOTO Y, AZUMI T, et al. Generic memory protection mechanism for embedded system and its application to embedded component systems [C] //Proceedings of the 8th International Conference on Computer and Information Technology Workshops. Los Alamitos, CA, USA: IEEE, 2008: 557-562.
- [3] 常德成, 徐高潮. 虚拟机动态迁移方法 [J]. 计算机应用研究, 2013, 30(4): 971-976.
- [4] 李彦冬, 雷航. 多核操作系统发展综述 [J]. 计算机应用研究, 2011, 28(9): 3215-3219.
- [5] 陈丽蓉, 燕立明, 罗蕾. 汽车电子嵌入式操作系统的隔离保护机制 [J]. 电子科技大学学报, 2014, 43(3): 450-456.
- [6] 张磊. 基于双核 ARM 平台的双操作系统移植的研究与实现 [D]. 西安: 西安电子科技大学, 2013.
- [7] 何翔, 任晓瑞, 刘帅. 嵌入式多核操作系统确定性研究 [J]. 航空计算技术, 2014, 44(3): 96-100.
- [8] 苏建志. 双操作系统任务调度策略及其改进 [J]. 内江师范学院学报: 自然科学版, 2014(2): 18-23.
- [9] 李刚. 多核协同计算平台的研究与实现 [D]. 成都: 电子科技大学, 2014.
- [10] 张博. 嵌入式实时操作系统中的数据缓冲技术 [J]. 电子世界, 2014(4): 24-25.
- [11] 黄鹏. 嵌入式软件开放式开发技术研究 [D]. 武汉: 华中科技大学, 2012.
- [12] 周伟明. 多核计算与程序设计 [M]. 武汉: 华中科技大学出版社, 2009: 315-379.
- [13] 杨铸, 唐攀. 深入浅出: 嵌入式底层软件开发 [M]. 北京: 北京航空航天大学出版社, 2011: 190-207.
- [14] 郑杰. ARM 嵌入式系统开发与应用完全手册 [M]. 北京: 中国铁道出版社, 2013: 88-126.
- [15] 桑楠, 赵丽, 郭文生. 多核平台嵌入式浏览器并行机制的研究与设计 [J]. 电子科技大学学报, 2014, 43(3): 400-404.

(责任编辑 朱雪莲 英文审校 黄振坤)