

文章编号: 1001- 5167(2002)04-0293-05

嵌入式操作系统中硬件抽象层的描述^{*}

张益农¹, 黄文玲²

(1.清华大学自动化系, 北京 100084; 2.内蒙古工业大学信息工程学院, 内蒙古 呼和浩特 010062)

摘要: 本文结合一个具体的嵌入式系统的实现, 探讨了硬件抽象层 HAL(Hardware Abstraction Layer) 的作用、主要内容以及具体实现方式等, 并提出了实现硬件抽象层的基本原则和方法。

关键词: 嵌入式; 操作系统; 可移植性; 硬件抽象层; HAL

中图分类号: TP3 **文献标识码:** B

0 引言

近年来, 随着微电子技术的飞速发展, 计算机硬件的集成度越来越高, 体积越来越小, 而功能却不断提高, 象常用于嵌入式系统的微控制器 MIPS ARM, StrongARM, SH4等的主频也达到或超过了 200M HZ, 这些微处理器在体积上不仅没有增大且有一定幅度的下降, 在单位功耗上均有大幅度的下降。在价格方面, 高档嵌入式微控制器近年来随着使用量的大幅度增加和技术的进步, 也在快速地下降。这些都为嵌入式系统的发展提供了必要的条件, 大量的个人电子消费品, 如各种 HPC(Handheld PC)、PDA(Personal Digital Assistant) 数码相机等就是上述技术发展的产物。从本质上看许多嵌入式系统是计算机的微型化, 但在很多地方又有所不同, 如嵌入式系统一般体积较小, 系统程序、应用程序一般装在 FLASH或 ROM中, 无存储量大的硬盘等设备。这些特点决定了嵌入式设备的操作系统和台式机的不同。特别是嵌入式系统不象台式机那样有相对统一的硬件标准, 即使是采用同一种 CPU设计制造的不同硬件平台有时也相差很大(如各种外围设备用的控制芯片可能差别很大)。这些原因导致了在嵌入式应用领域难以出现象台式机那样由少数操作系统垄断的局面, 因为任何一个嵌入式操作系统也难以做到能适应各种 CPU以及多种多样的硬件平台, 这为我们发展我国自己的嵌入式系统提供了机遇。但也应看到, 国外嵌入式系统的开发和研究早在二十年多年前就已开始, 商品化的嵌入式操作系统产品就有数十种之多。这些嵌入式产品从开发角度看大致可分两大类:

- 1) 从台式机操作系统演化而来的嵌入式操作系统: 如微软公司的 Windows CE, SUN公司的 Java OS, 朗讯科技公司的 Inferno;
- 2) 专门从事嵌入式系统软件的开发商开发的产品, 如: 3COM公司的 Palm OS, Microware公司的 OS-9, Accelerated Technology公司的 Nucleus, WindRiver公司的 VxWorks, ISI的 pSOS, QNX系统软件公司的 QNX, INTEL公司的 iRMX等。

由于我国在嵌入式操作系统的研究开发方面起步比较晚, 商品化的嵌入式操作系统还比较少, 在这方面还缺少经验, 但跨越尽可能多的硬件平台是各种嵌入式操作系统的基本追求之一。如何能将嵌入式操作系统快速地移植到新的硬件平台, 是开发嵌入式操作系统时必须加以注意的问题, 这个问题应该在开发的初始阶段, 也即在整体设计阶段就应该通盘考虑。如果在整个操作系统成型后, 在移植时才考虑

^{*} 收稿日期: 2002-09-28

作者简介: 张益农, 女, 信息工程学院讲师, 现为清华大学在读硕士。

的话,将会付出很大的代价.所以从可移植性考虑,合理设计整个嵌入式系统的体系结构是十分重要的一环,有时,为了可移植性,可以适当牺牲一些其它方面的要求,如内核可能因此会加大等.但我个人认为,在大多数情况下,特别是用于个人消费类电子产品中时,适当地以程序空间换取移植时间是非常值得的.

通过多年的开发研究,人们对操作系统的认识已很深入,操作系统的开发方法已逐渐完善,模块化的分层设计已成为目前绝大多数操作系统的实现方式.嵌入式操作系统的开发和其它操作系统的开发方式有很大的相似性,但嵌入式操作系统有其自身的特点,其主要特点是:

- (1) 规模较小 整个系统所占的空间要小,这主要是由其应用领域所决定的.一般由微内核及一些可以根据需要进行定制的系统模块组成,整体占用空间比较小,功能一般较微机上的操作系统弱.
- (2) 实时性强 由于嵌入式系统大多用于和各种控制有关的系统,因而对其实时性的要求一般比台式机要强.
- (3) 专用性强 一般和硬件的联系比较紧密,都是针对各种特定用途而设计,根据不同的应用而呈现多样化的形式.
- (4) 应用广泛 广泛应用于各种航天飞行器控制系统,数字电视机、VCD 学习机、寻呼机、手持电话, PDA 掌上 PC 电话系统、数码相机、各种智能设备、仪器仪表等众多的军用和民用领域.
- (5) 可移植性要好 由于其应用的广泛性和专用性,决定了嵌入式操作系统的可移植性要好,这样才有生命力.

对于可移植性的保证,现代操作系统主要通过引进硬件抽象层 HAL(Hardware Abstract Layer)来实现,我们在分析研究后认为在嵌入式操作系统的开发中引进硬件抽象层是十分必要的,它会给日后平台间的移植工作带来很大的便利,否则,日后在平台间的移植将成为耗时很大的工作.同时硬件抽象层结构的好坏对于开发、完善系统的工作量等均有很大的影响.

我们在研制开发嵌入式操作系统时,为了能够实现跨越多种硬件平台(整个系统的应用目标是个人消费类电子产品 PDA HPC和嵌入式控制器)对硬件抽象层作了一定的研究,反复分析了当今流行的多种操作系统(包括台式机和嵌入式操作系统)后,提出了下列结构模型(见图一):其中的图形用户界面主要用于个人消费类电子产品中,在用于各种控制器时,不需要该层.

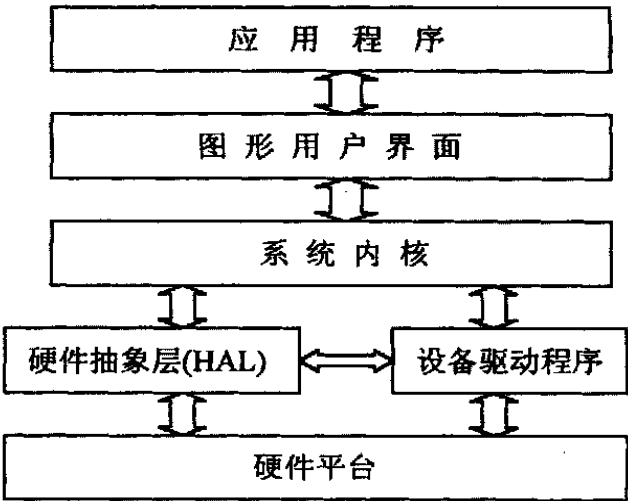


图 1 清华嵌入式操作系统的层次结构

其中的硬件抽象层是嵌入式操作系统和硬件直接接触的基本层,它将系统内核和具体的硬件平台隔离开来,从而实现系统内核和硬件无关.从这种意义上来说,它和台式机中的 ROM BIOS有一定的相似之处.下面结合我们实际的工作对硬件抽象层作一些阐述,希望和大家探讨.

1 硬件抽象层的主要作用

硬件抽象层是从软件角度从操作系统内核提炼出来的一个层次,严格来说,它本身就是操作系统内核的一个不可缺少的部分,只是从可移植性等方面考虑才将其作为一个单独的模块来对待,它处于操作系统的最低层,直接控制和访问硬件,是介于硬件平台和操作系统内核模块的一个基本层,其主要作用是向其它与机器无关的代码提供服务,从而使系统内核和具体的硬件平台隔离开来,实现和具体硬件平台的无关性。

有些操作系统没有硬件抽象层,这说明该操作系统没有将硬件抽象层单独作为一个相对独立的模块进行对待,而是将其作为系统内核的一部分看待而已。

由于嵌入式操作系统面临着大量的不同硬件平台,因此嵌入式操作系统可移植性的好坏不仅关系到开发的嵌入式操作系统的发展前途,而且对于开发、完善、移植的工作量等均有很大的影响。通过硬件抽象层对硬件平台合理的描述,可使操作系统内核基本和具体的硬件无关,从而容易地实现不同平台间的移植,这样就简化了嵌入式 OS 内核的移植工作,除了设备驱动程序,只需要修改 HAL 的代码就可以移植到其它平台上。

2 硬件抽象层实现的基本原则及目的

硬件抽象层究竟该包括那些内容,很难有一个明确的回答,这主要看系统的侧重点,不同的侧重点将导致不同的结果,这从当今各种商品化的 OS 中可以看出,但考虑到目前的技术发展现状(产品的更新速度越来越快)及 HAL 的主要作用,我们将其从满足最大易移植性出发,定出如下的基本原则:

1) 将硬件抽象层进一步划分为:

A. 系统体系层

该层用于描述 CPU 的基本结构,并实现传递中断,上下文切换和启动 CPU 等一些和具体 CPU 体系有关的部分。

B. 硬件实现层

该层则用于描述和具体的硬件实现平台相关的部分,主要用于描述和具体实现中相关的一些特性,同时提供了对定时器, I/O 寄存器访问,中断控制器和具体实现的启动处理等。

C. 其它实现层

该层用于抽象介于系统体系层和硬件实现层之间的一些属性,比如 CPU 结构变量和集成在 CPU 内部的各种外设控制器等设备。

将硬件抽象层进一步细化的目的主要是考虑到嵌入式产品不象台式机那样有相对统一的标准,各种硬件实现间的差异较大,即使使用同一种 CPU,由于在不同的硬件实现中,可能要用到不同的其它部件,如使用不同的中断控制器、不同的显示设备等,将造成即使嵌入式 OS 支持某种 CPU 也难以在某些具体的平台上运行的困境。为更快地实现相同 CPU 不同实现平台间的移植,因而将硬件抽象层进一步细分为三层,这样做的目的是为了进一步提高嵌入式 OS 的可移植性。

2) 用高级语言 C 或 C++ 和少量的汇编语言实现,这样可使硬件抽象层具有更大的适应性。

3) 用 C 或 C++ 宏实现 HAL 的接口。这样可以在内核内把它们实现为内联的 C 代码,内联的汇编码或外部的 C 或汇编函数调用。这将在不影响 HAL 和其它模块接口的前提下,选择最有效的实现方式。同时也容许在硬件实现层替换或改进已定义在系统体系层中的宏,这使系统的实现具有更大的灵活性。

4) 利用条件编译技术在硬件抽象层中提供对诊断和调试等的支持,因系统内核的调试比较困难,因此在硬件抽象层中提供对诊断、调试的支持就显得十分必要。和监控程序配合,可使系统内核的调试工作变得相对容易,易于发现和改正系统内核中的存在的缺陷。

3 硬件抽象层的主要内容及注意事项

硬件抽象层的内容虽然因不同的着眼点而不同,但按照上述硬件抽象层实现的基本原则,我们将其分为

1) 初始化相关硬件和提供诊断调试等部分

该部分主要用于为系统的启动提供基本的支持,包括初始化中断、时钟、内存管理单元(MMU (Memory Manage Unit)、数据 Cache、指令 Cache以及和调试相关的部分(一般为串行口)等。

2) 寄存器堆的描述

该部分主要描述和 CPU 有关的部分,如 CPU 中各个寄存器的数量、寄存器的位宽,以及线程的上下文切换等。

3) 中断控制器的描述

为了提高嵌入式系统的实时性,将系统的某些中断分成两级进行处理,响应中断部分和处理中断部分,其中响应中断部分直接和硬件相关,属于硬件抽象层部分,这部分主要负责响应中断,然后向系统内核发出有关消息,代码量一般比较少,而处理中断部分则放在系统内核中按线程进行处理,这样不仅提高了系统的响应速度,而且对系统的移植也大有好处。

4) 内存控制器的描述

该部分并不是必需的部分,有些 CPU 根本就没有 MMU,对于有 MMU 的处理器,如果要使用该功能的话,则应对其进行相应的描述。

5) 数据、指令缓冲器的描述

为了提高 CPU 的性能,现在的 CPU 一般均有数据 CACHE 和指令 CACHE,为了充分发挥 CPU 的性能,对各种 Cache 要进行相应的描述。

6) 时钟描述

该部分用于描述系统时钟的各种行为。

7) 基本 I/O 的描述

这部分主要包含嵌入式系统硬件平台中各种 I/O 的数据结构、端口地址等硬件参数,以及系统启动时用到而在无需用设备驱动程序描述的一些硬件设备,如系统运行状态的指示灯等

8) 对编程的支持

由于 C/C++ 的基本数据类型有限,难以满足嵌入式操作系统的设计要求,同时考虑具体的 CPU 结构和从可移植性方面考虑,需要定义一些基本的数据类型,如 8 位、16 位、32 位有(无)符号类型等。同时定义一些原子操作语句,以便用来实现系统线程的同步和互斥等操作,这些原子操作语句一般用关中断和开中断的方式来实现。

4 结束语

我们研制开发的嵌入系统已基本完成了硬件抽象层、内核层和大多数的设备驱动程序以及部分图形用户界面。通过上述提出的对硬件抽象层的进一步分层描述,不仅使整个系统的可移植性大幅度提高,而且整个系统的模块化程度特别高,模块间的接口简单清晰,各个模块的开发可并行进行,极大地提高了开发效率,从而使整个开发工作进展的非常顺利和快速,在可移植方面,我们只花很少的时间就已将系统的内核实现了在 X86、SH3、StrongArm 等 CPU 上的运行。

参考文献:

- [1] William Stallings Operating Systems Internals and Design Principles Prentice Hall 1998.
- [2] Bennett S. Real-Time Computer Control. Prentice Hall 1988
- [3] RTKernel Real-Time Multitasking Kernel User Hand Book. On Time Informatic GM BH, 1991.
- [4] ARM Architecture Reference Manual. Prentice Hall 1996
- [5] Hardware-Software Co-Design of Embedded systems by Felice Balarin etc., 1997 by Kluwer Academic publishers.

[6] 张云生. 实时控制系统软件设计原理及应用 [M]. 国防工业出版社, 1998. 12.
[7] 沈昌祥. 实时系统软件设计初步 [M]. 人民邮电出版社.
[8] 杨芙清文集. 北京大学出版社, 1998. 5

THE DESCRIPTION OF HAL IN EMBEDDED OS

ZHAN G Yi-nong

(*Department of Automation, Tsinghua University, Beijing 100084, PRC*)

Abstract In this paper, HAL (Hardware abstraction layer), especially its functions, essentialities and realization, is discussed in detail, based on implementation of a specific embedded OS (operating system). The principle and fundamental method for the realization of HAL are put forward as well.

Keywords embedded; OS (operating system); portability; HAL (hardware abstraction layer)