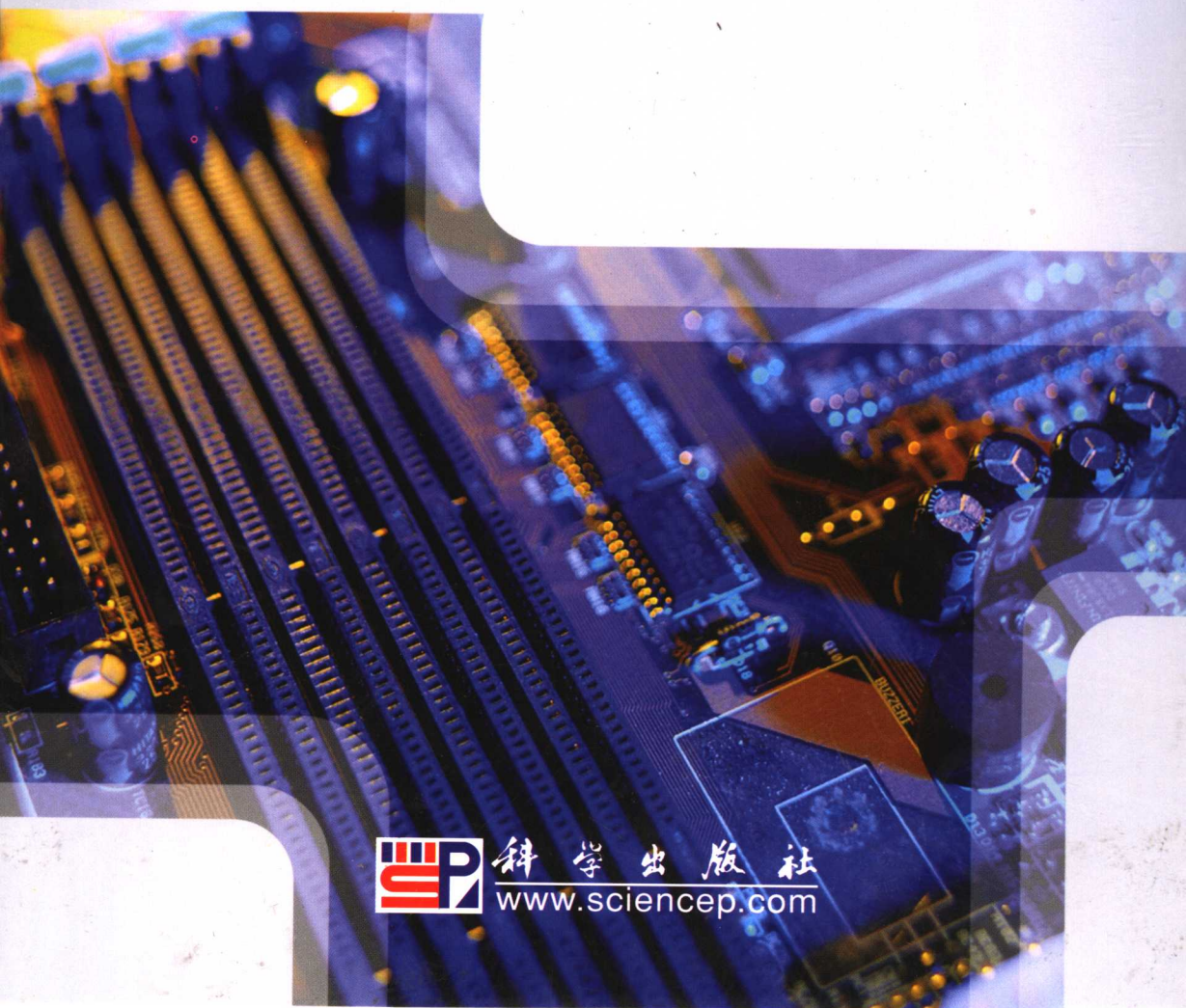


嵌入式系统的设计及应用

张海涛 著



科学出版社

www.sciencep.com

嵌入式系统的设计及应用

张海涛 著

本书由河南科技大学学术著作出版基金资助出版

科学出版社

北京

内 容 简 介

本书重点介绍由嵌入式处理器组成的复杂分布式系统设计的一些最新研究成果。在阐述了嵌入式系统的定义、发展史、组成、特定化以及基本原理后,详细讨论了基于裕度的调度算法,该算法是一种启发式的静态调度算法,结合了优先级抢先和不可抢先的优点,适合于复杂的多速率任务图。随后介绍了基于 SoPC 的嵌入式系统设计方法,并重点讨论了基于 Petri 网的嵌入式系统设计方法的最新研究成果。最后将 agent 技术和各种网络技术结合,给出了由信息家电组成智能家居系统的理论框架。

本书可作为从事嵌入式系统研究的科研人员的参考书。

图书在版编目(CIP)数据

嵌入式系统的设计及应用/张海涛著. —北京:科学出版社,2007

ISBN 978-7-03-019538-8

I. 嵌… II. 张… III. 微型计算机—系统设计 IV. TP360.21

中国版本图书馆 CIP 数据核字(2007)第 119690 号

责任编辑:耿建业 于宏丽/责任校对:钟 洋

责任印制:刘士平/封面设计:耕者工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

源海印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2007年8月第一版 开本:BS(720×1000)

2007年8月第一次印刷 印张:10 1/2

印数:1—2 500 字数:204 000

定价:28.00元

(如有印装质量问题,我社负责调换<明辉>)

前 言

嵌入式技术发展迅猛,日新月异,从事嵌入式系统的人员迫切需要相关的书籍作为参考。然而,现有的介绍嵌入式系统的书籍存在以下不足:①主要是针对单处理器组成的嵌入式系统,很少考虑各种网络系统;②主要是针对传统的嵌入式系统设计方法,这种设计方法主要是由设计者根据经验,将系统划分为软件和硬件,然后进行软硬件的单独设计,最后集成软硬件而得到整个系统;③主要是针对某一处理器或某一操作系统介绍具体的设计方法。

嵌入式系统是融合了计算机、通信、电子、控制、网络等知识的一个交叉领域,基于各种总线和协议的分布式嵌入式系统在控制系统中的应用已经相当普及。本书在介绍嵌入式操作系统、嵌入式处理器时,重点强调理论和最新研究成果,以便于读者设计特定化的嵌入式系统。在分布式嵌入式系统中,调度算法的设计是一个难点。针对目前算法中的不足,本书详细介绍了作者提出的基于裕度的调度算法,该算法适合于复杂的多速率任务图。

随着可编程逻辑器件和 VHDL 语言的出现,基于 SoPC 的嵌入式系统设计方法成为目前嵌入式系统的主要研究方向。基于形式语言的嵌入式系统设计方法也是其中的一个分支,只是在设计过程中,强调形式语言的建模和形式的校验。合理的系统模型是该方法的关键,针对现有各种时间 Petri 网在建模调度时的不足,本书详细介绍了作者提出的扩展时间 Petri 网模型 RBTRN,分析了其语义和可以建立的基本模型后;针对常规调度分析算法在分布式嵌入式系统中的不足,详细介绍了两种有效的调度分析算法,这使问题得到了解决,而且更为直观便捷,并进一步给出了基于可达图的调度分析方法及相关的简化分析算法。

智能家居系统是由各种嵌入式设备组成的网络系统,结合 agent 技术和 UPnP 技术,给出了一个控制构架,从中可以看到嵌入系统的网络化发展趋势,以及相应的设计思路。

在中国科学院自动化研究所攻读博士研究生期间,作者就嵌入式系统的实时调度和基于形式语言的嵌入式系统设计方法论进行了研究,这些研究成果构成了本书的核心内容。在此,对中国科学院自动化研究所给予的帮助表示诚挚的谢意!

本书的出版得到了河南科技大学学术著作出版基金和河南科技大学人才科学研究基金(06-8)的资助,在此表示感谢。

由于作者水平有限,书中难免存在不当之处,恳请读者批评指正。

作 者

2007 年 5 月

目 录

前言

第 1 章 嵌入式系统基本理论	1
1.1 嵌入式系统的定义	1
1.2 嵌入式系统的发展史	2
1.3 嵌入式系统的基本组成	5
1.4 嵌入式实时操作系统	7
1.5 嵌入式处理器	14
1.6 嵌入式系统的通信结构	22
1.7 传统的嵌入式系统的设计方法论	25
1.8 本书结构安排	27
第 2 章 嵌入式系统的实时调度算法	29
2.1 实时系统简介	29
2.2 实时调度算法	29
2.3 扩展的两种调度算法	35
2.4 基于裕度的列表调度算法	38
2.5 本章小结	47
第 3 章 基于 SoPC 的嵌入式系统的设计	48
3.1 SoC 和 SoPC 技术	48
3.2 设计流程	54
3.3 本章小结	57
第 4 章 一种扩展的时间 Petri 网模型	58
4.1 基于形式语言的嵌入式系统设计方法论	58
4.2 形式语言	59
4.3 Petri 网建模	64
4.4 RBTPN 模型	72
4.5 本章小结	81
第 5 章 基于 RBTPN 模型的调度分析	82
5.1 可达图分析	82
5.2 调度分析	98
5.3 基于可达图的调度分析	108

5.4	调度序列的分解和合并	118
5.5	调度序列的时间限制	124
5.6	实时系统的静态校验	126
5.7	同现有调度模型的比较	128
5.8	本章小结	134
第 6 章	智能家居系统	135
6.1	智能家居概述	135
6.2	相关技术	137
6.3	智能家居系统	148
6.4	本章小结	157
参考文献	158

第 1 章 嵌入式系统基本理论

1.1 嵌入式系统的定义

近些年,嵌入式系统已经成为最热门的研究领域之一。但是关于嵌入式系统定义的问题,还在不断讨论之中,到目前还没有统一的标准。

嵌入式系统内部含有微处理器,具有计算功能,但嵌入式系统又不是普通的 PC(personal computer)。IEEE 协会给出了如下定义:嵌入式系统是用于控制、监视或者辅助操作机器和设备的装置。而国内从事嵌入式技术的工作者普遍认同的定义为:以应用为中心,以计算机技术为基础,软硬件可裁剪,适应对功能、安全性、体积、实时性、可靠性、重量、成本、功耗、环境等方面严格要求的专用计算机系统。

因此,从应用的角度看,可以将计算机分为嵌入式计算机和非嵌入式的通用型计算机。从嵌入式的定义可以看出,嵌入式系统实质上就是实现某些特定要求的计算机应用系统。嵌入式系统的应用十分广泛,小到 mp3、手机、机顶盒、数码照相机,大到网络家电、智能家电、车载电子设备、工业机器人、航空航天设备等。

可以从广义和狭义的角度来看待嵌入式系统的定义。从广义上讲,可以认为凡是带有微处理器的专用软硬件系统都可以称为嵌入式系统。从这点来讲,嵌入式系统虽然是近几年的时髦名词,但如果从 20 世纪 70 年代的 MCS-51 单片机算起,也已经存在三四十年了,并不是什么新鲜的技术。然而从狭义上讲,仅仅是指那些使用 32 位以上的嵌入式微处理器、具有嵌入式操作系统并且具有某些特定功能的系统。按照这种定义,典型的嵌入式系统有使用 x86 的小型嵌入式工控主板,其在各种自动化设备中有着广泛的应用;另外一大类是使用 Intel、Motorola、Samsung 等专用芯片构成的小系统,广泛应用在各种消费电子和通信领域。值得一提的是,国内习惯把 4 位、8 位、16 位微控制器称为单片机。实际上,“嵌入式系统”这一名词也正是由于 32 位微处理器崛起而出现的,近十年才传入我国,并渐渐应用于各个领域。

1.2 嵌入式系统的发展史

1.2.1 嵌入式系统的诞生历程

1. 微型计算机的发展史

1946年,美国宾夕法尼亚大学研制出了世界上第一台电子管计算机——ENIAC,其核心器件是电子管。通常学术界根据电子计算机的逻辑单元所采用的物理器件的发展来划分计算机的发展史。到目前为止,电子计算机的发展已经经历了四代。第一代是电子管计算机,第二代是晶体管计算机,第三代是集成电路计算机,第四代是大规模集成电路计算机。

微型计算机是第四代计算机的典型代表,构成微型计算机的核心是微处理器,又称CPU(中央处理单元),基本上每两三年就有新产品问世。所谓微型计算机是把微处理器配上一定容量的半导体存储器(RAM和ROM)、接口电路(并行接口、串行接口、USB等)和必要的外设组成的。

微型计算机的升级换代也取决于它的核心器件——微处理器的性能。以“芯片之王”Intel公司为例,从20世纪70年代诞生第一片微处理器以来,已经推出了五代微处理器产品,经历了8086、80286、80386、80486、Pentium等五代。目前,Pentium处理器已经发展到了第四代Pentium IV,以及多核处理器。从总体上看,微处理器的速度越来越快、容量越来越大、功能越来越强。

2. 嵌入式系统的诞生

以微处理器为核心的微型计算机具有小型、价廉、高可靠性和智能性等特点。为了解决复杂的控制问题,一些设计人员将微型计算机引入到控制系统中,利用微型计算机强大的处理能力,完成各种复杂的控制算法。从本质上来看,就是将微型计算机嵌入到一个被控对象中,实现被控对象的智能化控制。为了便于安装,就需要减小微型计算机的体积;而为了提高系统的整体性能,可能无法减小体积。因此,为了满足不同的需要,微型计算机开始朝两个方向发展,一方面向微巨型化机发展,以便获得基于微型机的巨型机体系结构;另一方面,向更小型化方向发展,从而出现了单板机、单片机等。

微巨型机是将超级微型计算机与巨型机的技术结合起来形成的一种完全新型的体系结构。微巨型机具有明显的优点,即微型机的代价,巨型机的性能。

单板机是指把微处理器集成在一个或两个硅片上,再加上一个或多个存储器芯片、输入/输出(I/O)接口芯片和必要的I/O设备,并把它们装配在一块PCB板上。国内曾广泛使用以Z80CPU为微处理器的TP801单板机,但由于其设计复

杂、成本高,已经被单片机所取代。

单片机是指把微处理器、一定容量的存储器和必要的 I/O 接口电路集成在一个芯片上,就构成了单片微型计算机。有的单片机还可以集成更多的功能,如 A/D、D/A 等。单片机使用广泛,被用在自动控制、智能仪表、家用电器等各种设备中。其中,Intel 公司 8051 系列应用最为广泛。

不论是单板机还是单片机,都已不再具有通用计算机的外观,并且体系结构和功能也发生了很大改变。通常,为了与通用计算机系统相区别,从应用的角度,把嵌入到被控对象中实现被控对象智能化控制的计算机,称作嵌入式计算机。因此,嵌入式计算机是由于应用的需要,由微型计算机发展而来,并且还在迅速发展。由嵌入式计算机构成的系统,就称为嵌入式系统。

3. 计算机技术两大分支的形成

嵌入式计算机与通用计算机具有不同的工作环境,服务于不同的目的,因此嵌入式计算机具有与通用计算机系统完全不同的技术要求,从而形成了不同的技术发展方向。对于通用计算机,虽然速度越来越快、存储容量越来越大、性价比越来越高,但是人们总是期待拥有速度更快、存储容量更大、性价比更高的计算机。通用计算机总是不断地更新换代,在性能越来越高的同时价格却不断降低。比如 Pentium IV 当前的价格甚至比十年前的 80386 计算机还便宜,而性能却提高了许多倍。当然,某个型号的通用计算机总是会被淘汰的,现在就已经没有谁会再使用 80386 计算机了。而对于嵌入式计算机,人们则希望其在满足应用要求的情况下,尽可能地降低成本。如嵌入式处理器虽然也从 4 位发展到 8 位、16 位、32 位、64 位,但是没有因为 64 位嵌入式处理器的出现就淘汰了 32 位。在低端市场和高端市场,32 位和 8 位嵌入式处理器分别占据着统治地位。并且,8 位单片机仍然占据最大的市场,而且 4 位单片机也没有被淘汰,仍然在被使用。因此,选择嵌入式处理器时,性能最高的不一定是最好的、最适合的选择。

既然面向应用的嵌入式计算机与通用计算机具有不同的特点,那么它们就需要不同的体系结构。因此,嵌入式计算机的发展思路就与通用计算机有所不同,使计算机技术的发展形成了两大分支:通用计算机和嵌入式计算机。嵌入式计算机的诞生,标志着计算机进入了通用计算机与嵌入式计算机两大分支并行发展的时代,使计算机在更广泛的领域获得了应用。通用计算机系统与嵌入式计算机系统的专业化分工发展,也使计算机技术得到了迅猛的发展。由于计算机技术是一门基础学科,在它的带动下,其他技术也得到了迅猛发展。

对于通用计算机领域,由于不需要考虑具体的应用,根据摩尔定律“微芯片上集成的晶体管数目每三年翻两番”,从而使同样大小的芯片具有更强大的功能。通用微处理器发展到奔腾系列后,出现了双核、四核 CPU,1GB 的内存,300GB 的硬

盘也越来越普通。伴随着硬件性能越来越高,操作系统功能也越来越强大,从DOS发展到今天的 Windows Vista,使通用计算机的整体性能越来越高。

嵌入式计算机的发展则有所不同。一方面,同通用计算机类似,微处理器性能不断提高,还产生了操作系统,使 32 位的嵌入式计算机的性能甚至超过了早期的 80386、80486 计算机。另一方面,面向特定应用的特点产生了适合于不同场合的嵌入式计算机。比如,在 Intel51 单片机上集成不同功能模块,如 A/D、D/A、PWM、定时器、串口、存储器等,派生出了很多种不同的单片机,从而使 51 系列单片机成为应用最为广泛的嵌入式计算机。当然,目前,嵌入式计算机总体的发展方向是片上系统(SoC)。

SoC 是一个微小型系统。SoC 是将微处理器、模拟 IP 核、数字 IP 核和存储器(或片外存储控制接口)集成在单一芯片上。它通常是客户定制的,或是面向特定用途的标准产品。SoC 是面向特定用户的、能最大满足嵌入式系统要求的芯片,因而具有很多优势。由于 SoC 缩短了片内连线,因而能极大改善功耗开销,减少印制板上的部件数和管脚数,从而减少板卡失效的可能性。另外,还可降低风冷要求,减少系统开发成本,尤其适合数字化产品开发,如手持设备、信息家电等。

1.2.2 嵌入式系统的发展历程

计算机技术形成两大分支以后,嵌入式系统走上了一条与通用计算机系统不同的发展道路。除了性能不断提高之外,最重要的就是片上系统的产生。实际上,不论是低端还是高端,嵌入式系统都在朝着这一方向发展。常用的单片机就是在一个芯片上集成尽量多的功能模块。为了进一步提高嵌入式系统的性能,嵌入式计算机结构采用了不同于通用计算机的设计模式。

在单片机的发展道路上,有过两种模式,即“裁剪模式”与“创新模式”。“裁剪模式”就是将通用计算机系统中的基本单元进行裁剪后,集成在一个芯片上,构成单片微型计算机。“创新模式”则完全按嵌入式应用要求设计全新的,满足嵌入式应用要求的体系结构、微处理器、指令系统、总线方式等。比如,获得广泛应用的 Intel 公司 MCS-51 系列就是按照创新模式发展起来的单片微型计算机。MCS-51 系列放弃了通用计算机所采用的微程序结构,采用了 RISC 的硬件结构,并且内嵌了常用的功能模块,使其得到了大量应用,成为了单片嵌入式系统的典型结构体系。

从 MCS-51 系列的成功可以看出,嵌入式计算机虽然起源于微型计算机时代,但是微型计算机的体系结构明显无法满足嵌入式系统的要求。因此,嵌入式计算机走上了一条与通用计算机不同的发展道路,其最重要的一点就是芯片化。也就是将嵌入式处理器及其外围接口、存储器等模块集成在一个芯片上,从而进入了嵌入式系统独立发展的片上系统时代。

嵌入式系统的核心是嵌入式处理器。从嵌入式处理器的角度来看,嵌入式系统的发展主要经历了嵌入式微处理器(EMPU)、微控制器(MCU)、嵌入式 DSP 处理器、片上系统(SoC)等四个阶段。

1.3 嵌入式系统的基本组成

同任何计算机系统一样,嵌入式系统由嵌入式硬件和嵌入式软件两部分组成。硬件是支撑,软件是灵魂,几乎所有的嵌入式产品中都需要嵌入式软件来提供灵活多样、且应用特制的功能。由于嵌入式系统应用广泛,嵌入式软件在整个软件产业中占据了重要地位,并受到世界各国的广泛关注,如今已成为信息产业中最受人瞩目的行业。

1.3.1 嵌入式系统的结构

1. 系统结构

在整体上,嵌入式系统的结构同通用计算机还是很类似的。如图 1.1 所示,嵌入式系统由四层结构组成。

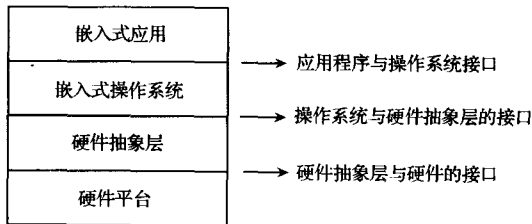


图 1.1 嵌入式系统基本结构

硬件平台是整个嵌入式操作系统和应用程序运行的平台。为了满足不同的应用,通常应该选择不同的硬件环境。硬件平台的多样性是嵌入式系统的一个主要特点。

嵌入式操作系统完成嵌入式应用的任务调度和控制等核心功能。具有内核精简、可配置、与高层应用紧密关联等特点。嵌入式操作系统具有相对不变性。

嵌入式应用程序运行于操作系统之上,利用操作系统提供的机制完成特定功能的嵌入式应用。不同的系统需要设计不同的嵌入式应用程序。

硬件抽象层通过硬件抽象层接口向操作系统以及应用程序提供对硬件进行抽象后的服务。当操作系统或应用程序使用硬件抽象层 API 进行设计时,只要硬件抽象层 API 能够在下层硬件平台上实现,那么操作系统和应用程序的代码就可以

移植。

2. BSP

板级支持包 BSP(board support package)作为硬件抽象层的一种实现,是目前大多数嵌入式操作系统实现可移植性所采用的一种方案。BSP 隔离了所支持的嵌入式操作系统与硬件平台之间的相关性,使嵌入式操作系统能够运行在 BSP 所支持的硬件平台之上,从而实现了嵌入式操作系统的可移植性、跨平台性、通用性等。也就是说, BSP 是相对于操作系统而言的,不同的操作系统对应于不同定义形式的 BSP,例如, WinCE 的 BSP 和 Linux 的 BSP 相对于同一个 CPU 来说,尽管实现的功能一样,可是写法和接口定义是完全不同的。BSP 的引入极大地推动了嵌入式操作系统的通用化。

然而 BSP 形式的硬件抽象层,完全是为了使现有的嵌入式操作系统可以移植到不同的硬件平台上,因此 BSP 形式的硬件抽象层与其所支持的嵌入式操作系统是密不可分的。在同一种嵌入式微处理器的硬件平台上,为了支持不同嵌入式操作系统所编写的 BSP 也具有很大的差别。所以即使针对某个特定的嵌入式微处理器,一种嵌入式操作系统的 BSP 也无法用于其他的嵌入式操作系统。可见, BSP 是一种封闭的专用硬件抽象层。

由上可知, BSP 具有以下主要特点:

(1) 作为操作系统与硬件之间的接口, BSP 需要为操作系统提供操作和控制具体硬件的方法。

(2) 不同的操作系统具有不同的软件层次结构,因此,不同的操作系统具有特定的硬件接口形式。

在实现上, BSP 是一个介于操作系统和底层硬件之间的软件层次,包括了系统中大部分与硬件相关的软件模块。在功能上包含两部分:系统初始化,与硬件相关的设备驱动。需要指出的是,通常 BSP 中所包含的与硬件相关的设备驱动程序并不直接由 BSP 使用,而是在系统初始化过程中由 BSP 把它们与操作系统中通用的设备驱动程序关联起来,使得在以后的应用中,通用的设备驱动程序可以调用 BSP 中与硬件相关的设备驱动程序,从而实现对硬件设备的操作。设计与硬件相关的驱动程序是 BSP 设计中另一个关键环节。

1.3.2 嵌入式系统的硬件组成

嵌入式系统硬件的核心是嵌入式处理器。以微处理器为核心的嵌入式系统硬件组成如图 1.2 所示。

嵌入式处理器与通用处理器有很大不同,其一般具有以下特点:

(1) 具有支持实时多任务系统的能力,这主要得益于其具有较短的中断响应

时间。

- (2) 具有存储区域保护功能,从而可以避免错误传播,易于纠正错误。
- (3) 具有可扩展的处理器结构。
- (4) 具有较低的功耗,从而便于使用电池供电。

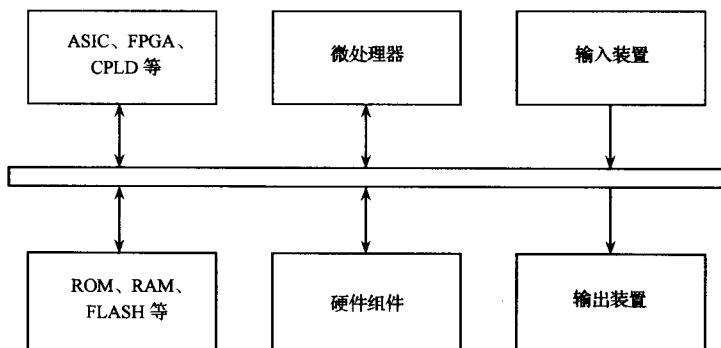


图 1.2 嵌入式系统硬件组成

1.4 嵌入式实时操作系统

面对复杂的任务,需要采用 32 位的嵌入式处理器,这样嵌入式操作系统就成为嵌入式系统设计中必不可少的一个环节。桌面操作系统(如 Microsoft Windows 系列的操作系统)并不适合直接应用在嵌入式操作系统上,为了适应嵌入式系统的需要,必须在整个系统的软件架构中引入嵌入式操作系统。

伴随着通用型嵌入式实时操作系统的发展,一个以面向 Internet 网络的、特定应用的嵌入式操作系统正日益引起人们的重视,成为极为重要的发展方向。嵌入式系统与 Internet 的真正结合、嵌入式操作系统与应用设备的无缝结合代表着嵌入式操作系统发展的真正未来。

1.4.1 实时操作系统

在嵌入式系统中,任务可以分为实时性任务和非实时性任务。实时任务又根据其对于实时性要求的不同,可以分为软实时和硬实时两种类型。硬实时是指任务要有确保在最坏情况下的服务时间,即对于事件的响应时间的截止期限是必须得到满足,否则系统可能产生严重的错误。其他所有具有实时特性的任务都可以称之为软实时任务。软实时任务可以这样理解:一个任务能够得到有确保的处理时间,到达系统的事件也能够在截止期限到来之前得到处理,但违反截止期限并不会带来致命的错误。

1. 嵌入式操作系统的分类

针对不同的应用,嵌入式操作系统也分为两类:实时操作系统和非实时操作系统。大多数嵌入式操作系统都是实时操作系统。

非实时操作系统的典型代表是分时操作系统,分时操作系统注重系统的平均表现性能,总是希望在尽量少的时间内为尽可能多的用户提供服务。由此可知,由于分时操作系统不关注每个具体任务的性能,因此不适合用于实时系统。

而对于实时操作系统,首要目标是利用所有资源完成每个实时任务,然后才考虑提高计算机系统的使用效率。因此,实时操作系统的设计原则是,在系统运行的任何时刻,任何情况下,其调度算法都能为每个实时任务合理地分配系统资源,使每个实时任务的性能要求得到满足。与分时操作系统不同,实时操作系统注重的不是系统的平均工作效率,而是要求每个实时任务在最坏情况下都要满足每个任务的实时性要求,也就是要求在实时操作系统中,每个任务的最坏执行时间必须满足时限要求。

实时操作系统具有如下特点:中断响应快、中断处理时间短、任务切换快、占有较小的空间。实时操作系统具有如下功能:任务管理、内存管理、中断管理和设备管理等服务。

实时操作系统可分为可抢占型和不可抢占型两类。而硬实时操作系统必须采用可抢占型。对于基于优先级的系统而言,可抢占型实时操作系统是指内核可以中断正在运行的任务,而执行已处于就绪态的优先级更高的任务。在不可抢占型实时操作系统中,一旦使用某种算法决定让某个任务运行后,即使有更高优先级的任务到来,也必须等待这个正在运行的任务执行完毕,才能够获得处理器使用权。可抢占型实时操作系统的实时性好,优先级高的任务只要进入就绪状态,就可以立即运行。也就是在这种系统中,除了优先级最高的任务,其他任务在运行过程中都可能随时被比具有更高优先级的任务中断,从而让出处理器。通过这种方式的任务调度保证了系统的实时性,但是,如果没有处理好任务之间抢占处理器的控制权,系统会崩溃。

2. 使用实时操作系统的必要性

目前,大多数嵌入式操作系统都是实时系统,嵌入式实时操作系统目前越来越广泛地用于嵌入式应用中。嵌入式实时操作系统的诞生、直到占有主导地位,主要有以下几方面的原因。

(1) 非实时操作系统尽管可以提高系统整体的执行效率,却可能导致某个任务的完成时间无法满足它的时限要求,如果是硬实时系统,还可能造成严重的错误。实时操作系统将满足每个任务的实时性要求放在首要位置,尽管完成某个应

用也许花费了较多的时间,但却可以满足该应用的所有任务的时限要求,保证系统的正确性。

(2) 早期的嵌入式系统中采用前后台系统软件设计,在遇到强干扰时,就会使正在运行的程序产生各种异常事件,甚至使系统崩溃。而在实时操作系统中,干扰可能仅仅破坏一个进程,而系统可以将其修复或者清除。

(3) 在嵌入式实时操作系统环境下,可以通过将应用程序分成许多独立的任务模块,使应用程序的设计过程大为简化。

(4) 嵌入式实时操作系统充分发挥了 32 位处理器的多任务潜力。32 位处理器采用利于提高系统可靠性和稳定性的设计,使其更不容易崩溃。

3. 嵌入式实时操作系统的组成

同通用操作系统一样,嵌入式操作系统位于底层硬件与用户之间,是两者沟通的桥梁。嵌入式操作系统的主要功能主要在于消除不同硬件的差异,让应用程序可以在其上进行操作。通过操作系统统一提供的应用程序接口,程序员就无须考虑系统所选用的硬件,只需要专心地开发各种应用程序。

用户可以通过操作系统提供的用户界面输入命令。操作系统则对命令进行解释,驱动硬件设备,实现用户要求。

大部分嵌入式系统的功能由以下几部分完成:

(1) 驱动程序是底层的、控制和监视硬件的部分,它们的职责是隐藏硬件的具体细节,并向操作系统的其余部分提供一个抽象的、通用的接口。

(2) 内核是操作系统的核心部分,一般都运行在最高特权级,负责提供基础性、结构性的功能。内核还包含了接口库,接口库是由许多程序库组成,负责把系统所提供的基本服务封装成应用程序所能够使用的应用程序接口(API)。这样应用程序使用接口库所提供的各种函数就能够获得操作系统提供的各种功能,使应用程序的编写变得很简单。

(3) 外围系统是指用于提供特定高级服务的部件。例如,在微内核结构中,大部分系统服务,以及 Linux 中各种守护进程都可以划归为外围系统。

当然,由于嵌入式操作系统是一个从无到有的发展过程,并且还在不断发展中,因此并不是每一种嵌入式操作系统都包括这三部分。例如,在使用外核结构的嵌入式操作系统中,就没有驱动程序的概念。

1.4.2 内核的体系机构

内核的结构往往对操作系统的外部特性以及应用领域有着一定程度的影响。随着操作系统的不断发展,其高层特性与内核结构之间的耦合不断缩小,但是内核结构仍然是操作系统分类的常用标准。

嵌入式操作系统的体系结构主要有以下几种:单内核、微内核、混合内核、外内核。其中,单内核为潜在的硬件提供了大量完善的硬件抽象操作;微内核只为硬件提供了很小一部分的抽象,大部分功能由一种特殊的用户态程序,即服务器来完成;混合内核很像微内核结构,只不过它的组件更多地运行在核心态中,以获得更快的执行速度;外内核不提供任何硬件抽象操作,但是通过为内核增加运行库,使应用程序可以直接地或者间接地对硬件进行操作。

下面详细介绍这些内核的结构。

1. 单内核

单内核是一个很大的进程,作为一个独立的二进制映象文件运行。其内部可以被分为若干模块,模块间的通信是通过调用其他模块中的函数实现的。图 1.3 显示了单内核的结构。

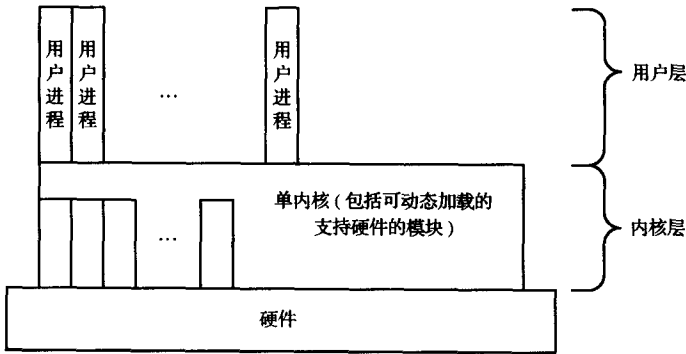


图 1.3 单内核体结构图

Linux 操作系统就是单内核体系结构的典型代表。对于单内核体系机构,用户层依靠内核层提供的接口实现各种用户需要的功能。用户可以直接通过函数调用请求操作系统内核服务,实现操作系统的功能,例如,进程管理、文件管理和存储管理等,这些功能由多个运行在内核的模块来完成。另外,内核层封装了与硬件相关的操作接口,为可能安装的硬件提供了大量完善的硬件抽象操作,可以提供与硬件交互的接口。

用户模式和内核模式拥有独立的地址空间。在单内核中,当用户模式的程序调用系统服务时,CPU 会切换到内核模式。同样,在内核模式下,执行系统服务后,CPU 就返回到用户模式,继续响应用户的服务请求。

由于只有系统调用时,才能进入内核模式,因此提高了系统的可靠性和安全性。另外,由于单内核作为一个独立的二进制文件运行,减少了层与层之间通信,进程切换少,负载小,提高了应用程序的执行速度。

然而,单内核结构有以下明显的不足:

(1) 用户为了获得内核提供的系统服务,必须通过内核层封装的接口,缺乏灵活性。

(2) 在单内核体内部,由于各个模块之间关系密切,不仅代码难以编写,而且可维护性较差,一个很小的错误就会使整个系统崩溃。

很多现代的单内核结构内核,如 Linux 和 FreeBSD 内核,可以在运行时将模块调入执行,这就可以使扩充内核的功能变得更简单,也可以使内核层部分变得更简洁。

2. 微内核

在微内核结构中,微内核作为操作系统的核心,集成了操作系统运行的基本功能。这些功能包括:进程通信 IPC(inter process communication)、虚拟存储 VM (virtual memory)、任务和线程管理、中断处理及与硬件相关部分。微内核只是为各种操作系统提供了一个公共基础,因而规模较小。如 QNX 微内核仅仅只有 8KB 内存和 14 条系统调用。

在微内核体系结构中,微内核仅仅提供了有限的系统调用。如图 1.4 所示,显示了微内核的结构。通过各个服务器,实现了传统的操作系统服务。服务器可以独立运行在用户地址空间,易于扩展操作系统的功能。由于各个服务模块在独立的地址空间运行,错误不会相互传递,因此解决了单内核的不足。由于微内核具有核心小巧、可靠、易于 ROM 固化、可模块化扩展等特点,成为了多数实时操作系统的选择。

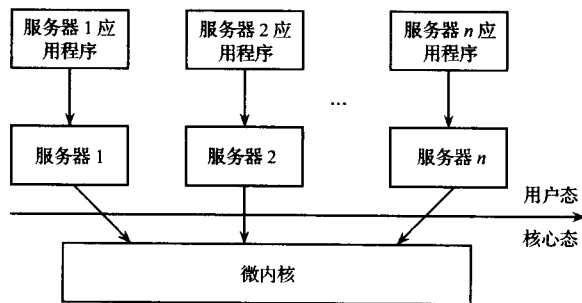


图 1.4 微内核结构图

微内核主要有以下的特点:

(1) 内核精简,仅仅具有操作系统的基本功能。

(2) 借助于各种服务器,同一个微内核可以支持一个或者多个不同操作系统的的应用程序,从而易于实现应用软件的继承。