

第一章

专题论述

1.1 3种嵌入式操作系统的分析与比较

重庆三峡学院(404000) 熊 江

一、嵌入式系统与嵌入式操作系统

1. 嵌入式系统

嵌入式系统是以嵌入式计算机为技术核心,面向用户,面向产品,面向应用,软硬件可裁剪的,适用于对功能、可靠性、成本、体积、功耗等综合性能有严格要求的专用计算机系统。

嵌入式系统应具有的特点是:具有高可靠性;在恶劣的环境或突然断电的情况下,系统仍然能够正常工作;许多嵌入式应用要求实时性,这就要求嵌入式操作系统具有实时处理能力;嵌入式系统和具体应用有机地结合在一起,它的升级换代也是和具体产品同步进行;嵌入式系统中的软件代码要求高质量、高可靠性,一般都固化在只读存储器中或闪存中,也就是说,软件要求固态化存储,而不是存储在磁盘等载体中。

2. 嵌入式操作系统

嵌入式操作系统(EOS, Embedded Operating System)是一种用途广泛的系统软件,过去它主要应用于工业控制和国防系统领域。EOS负责嵌入系统的全部软/硬件资源的分配、调度工作,控制、协调并发活动。它必须体现其所在系统的特征,能够通过装卸某些模块来实现系统所要求的功能。目前,已推出一些应用比较成功的EOS产品系列。随着Internet技术的发展、信息家电的普及应用及EOS的微型化和专业化,EOS开始从单一的弱功能向高专业化的强功能方向发展。嵌入式操作系统在系统实时高效性、硬件的相关依赖性、软件固态化及应用的专用性等方面具有较为突出的特点。EOS是相对于一般操作系统而言的,它除具备了一般操作系统最基本的功能,如任务调度、同步机制、中断处理、文件等功能外,还有以下特点:

可装卸性。开放性、可伸缩性的体系结构。

强实时性。EOS实时性一般较强,可用于各种设备控制当中。

统一的接口。提供各种设备驱动接口。

操作方便、简单,提供友好的图形GUI,追求易学、易用。

提供强大的网络功能,支持TCP/IP协议及其他协议,提供TCP/UDP/IP/PPP协议,支持及统一的MAC访问层接口,为各种移动计算设备预留接口。

强稳定性,弱交互性。嵌入式系统一旦开始运行就不需要用户过多的干预,这就要负责系统管理的EOS具有较强的稳定性。嵌入式操作系统的用户接口一般不提供操作命令,它通过系统调用命令向用户程序提供服务。

固化代码。在嵌入系统中,嵌入式操作系统和应用软件被固化在嵌入式系统计算机的ROM中。辅助存储器在嵌入式系统中很少使用,因此,嵌入式操作系统的文件管理功能应该能够很容易地拆卸,而用各种内存文件系统。

更好的硬件适应性,也就是良好的移植性。

国际上用于信息电器的嵌入式操作系统有 40 种左右。现在市场上非常流行的 EOS 产品中,3Com 公司下属子公司的 Palm OS 在全球占有份额达 50%,Microsoft 公司的 Windows CE 约占 29%。在美国市场,Palm OS 更以 80% 的占有率远超 Windows CE。开放源代码的 Linux 很适于做信息家电的开发,例如中科红旗软件技术有限公司开发的红旗嵌入式 Linux 和美商网虎公司开发的基于 Xlinux 的嵌入式操作系统“夸克”。夸克是目前全世界最小的 Linux,它有两个很突出的特点,就是体积小和使用 GCS 编码。

二、3 种常用的嵌入式操作系统

1. Palm OS

Palm 是 3Com 公司的产品,其操作系统为 Palm OS。Palm OS 是一种 32 位的嵌入式操作系统。Palm 提供了串行通信接口和红外线传输接口,利用它可以方便地与其他外部设备通信和传输数据。Palm 拥有开放的 OS 应用程序接口,开发商可根据需要自行开发所需的应用程序。Palm OS 是一套具有极强开放性的系统,现在大约有数千种专门为 Palm OS 编写的应用程序,从程序内容上看,小到个人管理、游戏,大到行业解决方案,Palm OS 无所不包。在丰富的软件支持下,基于 Palm OS 的掌上电脑功能得以不断扩展。

Palm OS 是一套专门为掌上电脑开发的 OS。在编写程序时,Palm OS 充分考虑了掌上电脑内存相对较小的情况,因此它只占有非常小的内存。由于基于 Palm OS 编写的应用程序占用的空间也非常小(通常只有几十 KB),基于 Palm OS 的掌上电脑(虽然只有几 MB 的 RAM)可以运行众多应用程序。

Palm 产品的最大特点是使用简便,机体轻巧,因此决定了 Palm OS 应具有以下特点:

① 操作系统的节能功能。由于掌上电脑要求使用电源尽可能小,在 Palm OS 的应用程序中,若没有事件运行,则系统设备进入半休眠(Doze)的状态;若应用程序停止活动一段时间,则系统自动进入休眠(Sleep)状态。

② 合理的内存管理。Palm 的存储器全部是可读/写的快速 RAM,动态 RAM(Dynamic RAM)类似于 PC 机上的 RAM,它为全局变量和其他无需永久保存的数据提供临时的存储空间;存储 RAM(Storage RAM)类似于 PC 机上的硬盘,可以永久保存应用程序和数据。

③ Palm OS 的数据是以数据库(Database)的格式来存储的。数据库是由一组记录(Records)和一些数据库头信息组成的。为保证程序处理速度和存储器空间,在处理数据时,Palm OS 不是把数据从存储堆(Storage Heap)拷贝到动态堆(Dynamic Heap)后再进行处理,而是在存储堆中直接处理。为避免错误地调用存储器地址,Palm OS 规定这一切都必须调用其内存管理器里的 API 来实现。

Palm OS 与同步软件(HotSync)结合可以使掌上电脑与 PC 机上的信息实现同步,把台式机的功能扩展到掌上电脑。Palm 应用范围相当广泛,例如联络及工作表管理、电子邮件及互联网通信、销售人员及组别自动化等。Palm 外围硬件也十分丰富,有数码相机、GPS 接收器、调制解调器、GSM 无线电话、数码音频播放设备、便携键盘、语音记录器、条码扫描、无线寻呼接收器和探测仪。其中 Palm 与 GPS 结合的应用不但可以作导航定位,还可以结合 GPS 作气候的监测、地名调查等。

2. Windows CE

Windows CE 是微软开发的一个开放的、可升级的 32 位嵌入式操作系统,是基于掌上型

电脑类的电子设备操作系统。它是精简的 Windows 95。Windows CE 的图形用户界面相当出色。其中 CE 中的 C 代表袖珍(Compact)、消费(Consumer)、通信能力(Connectivity)和伴侣(Companion),E 代表电子产品(Electronics)。与 Windows 95/98、Windows NT 不同的是,Windows CE 是所有源代码全部由微软自行开发的嵌入式新型操作系统,其操作界面虽来源于 Windows 95/98,但 Windows CE 是基于 Win32 API 重新开发的、新型的信息设备平台。Windows CE 具有模块化、结构化和基于 Win32 应用程序接口以及处理器无关等特点。Windows CE 不仅继承了传统的 Windows 图形界面,并且在 Windows CE 平台上可以使用 Windows 95/98 上的编程工具(如 Visual Basic、Visual C++ 等),使用同样的函数,使用同样的界面风格,使绝大多数的应用软件只需简单的修改和移植就可以在 Windows CE 平台上继续使用。

Windows CE 的设计目标是:模块化及可伸缩性、实时性能好,通信能力强大,支持多种 CPU。它的设计可以满足多种设备的需要,这些设备包括了工业控制器、通信集线器以及销售终端之类的企业设备,还有像照相机、电话和家用娱乐器材之类的消费产品。一个典型的基于 Windows CE 的嵌入系统通常为某个特定用途而设计,并在不联机的情况下工作。它要求所使用的操作系统体积较小,内建有对中断的响应功能。

Windows CE 的特点如下:

- ① 具有灵活的电源管理功能,包括睡眠/唤醒模式。
- ② 使用了对象存储(Object Store)技术,包括文件系统、注册表及数据库。它还具有很多高性能、高效率的操作系统特性,包括按需换页、共享存储、交叉处理同步、支持大容量堆(Heap)等。
- ③ 拥有良好的通信能力。广泛支持各种通信硬件,亦支持直接的局域网连接以及拨号连接,并提供与 PC、内部网以及 Internet 的连接,还提供与 Windows 9x/NT 的最佳集成和通信。
- ④ 支持嵌套中断。允许更高优先级别的中断首先得到响应,而不是等待低级别的 ISR 完成。这使该操作系统具有嵌入式操作系统所要求的实时性。
- ⑤ 更好的线程响应能力。对高级别 IST(中断服务线程)的响应时间上限的要求更加严格,在线程响应能力方面的改进帮助开发人员掌握线程转换的具体时间,并通过增强的监控能力和对硬件的控制能力帮助创建新的嵌入式应用程序。
- ⑥ 256 个优先级别。可以使开发人员在控制嵌入式系统的时序安排方面有更大的灵活性。
- ⑦ Windows CE 的 API 是 Win32 API 的一个子集,支持近 1 500 个 Win32 API。有了这些 API,足可以编写任何复杂的应用程序。当然,在 Windows CE 系统所提供的 API 也可以随具体应用的需求而定。

在掌上型电脑中,Windows CE 包含如下一些重要组件:Pocket Outlook 及其组件、语音录音机、移动频道、远程拨号访问、世界时钟、计算器、多种输入法、GBK 字符集、中文 TTF 字库、英汉双向词典、袖珍浏览器、电子邮件、Pocket Office、系统设置、Windows CE Services 软件。

3. Linux

Linux 是一个类似于 Unix 的操作系统。它起源于芬兰一个名为 Linus Torvalds 的业余爱好,但是现在已经是最为流行的一款开放源代码的操作系统。Linux 从 1991 年问世到现

在,短短 10 年的时间内已发展成为一个功能强大、设计完善的操作系统,伴随网络技术进步而发展起来的 Linux OS 已成为 Microsoft 公司的 DOS 和 Windows 95/98 的强劲对手。Linux 系统不仅能够运行于 PC 平台,还在嵌入式系统方面大放光芒,在各种嵌入式 Linux OS 迅速发展的状况下, Linux OS 逐渐形成了可与 Windows CE 等 EOS 进行抗衡的局面。目前正在开发的嵌入式系统中,49% 的项目选择 Linux 作为嵌入式操作系统。Linux 现已成为嵌入式操作系统的理想选择。

中科红旗软件技术有限公司开发的红旗嵌入式 Linux 正在成为许多嵌入式设备厂商的首选。在不到一年的时间内,红旗公司先后推出了 PDA、机顶盒、瘦客户机和交换机用的嵌入式 Linux 系统,并且投入了实际应用。现以红旗嵌入式 Linux 为例来讲解嵌入式 Linux OS 的特点:

- ① 精简的内核,性能高,稳定,多任务。
- ② 适用于不同的 CPU,支持多种体系结构,如 X86、ARM、MIPS、ALPHA、SPARC 等。
- ③ 能够提供完善的嵌入式 GUI 以及嵌入式 XWindows。
- ④ 提供嵌入式浏览器、邮件程序、MP3 播放器、MPEG 播放器、记事本等应用程序。
- ⑤ 提供完整的开发工具和 SDK,同时提供 PC 上的开发版本。
- ⑥ 用户可定制,可提供图形化的定制和配置工具。
- ⑦ 常用嵌入式芯片的驱动集,支持大量的周边硬件设备,驱动丰富。
- ⑧ 针对嵌入式的存储方案,提供实时版本和完善的嵌入式解决方案。
- ⑨ 具有完善的中文支持、强大的技术支持和完整的文档。
- ⑩ 开放源码,具有丰富的软件资源,具有广泛的软件开发者的支持,价格低廉,结构灵活,适用面广。

三、3 种嵌入式操作系统的比较

1. Linux OS 与 Windows CE 的比较

嵌入式 Linux OS 与 Windows CE 相比有以下优点。第一, Linux 是开放源代码,不存在黑箱技术,遍布全球的众多 Linux 爱好者都是 Linux 开发者的强大技术支持者;而 Windows CE 是非开放性 OS,使第 3 方很难实现产品的定制。第二, Linux 的源代码随处可得,注释丰富,文档齐全,易于解决各种问题。第三, Linux 的内核小、效率高;而 Windows CE 在这方面是笨拙的,占用过多的 RAM,应用程序庞大。第四, Linux 是开放源代码的 OS,在价格上极具竞争力,适合中国国情。Windows CE 的版权费用是厂家不得不考虑的因素。第五, Linux 不仅支持 X86 芯片,还是一个跨平台的系统。到目前为止,它可以支持 20~30 种 CPU,很多 CPU(包括家电业的芯片)厂商都开始做 Linux 的平台移植工作,而且移植的速度远远超过 Java 的开发环境。如果今天采用 Linux 环境开发产品,那么将来更换 CPU 时就不会遇到更换平台的困扰。第六, Linux 内核的结构在网络方面是非常完整的,它提供了对包括十兆位、百兆位及千兆位的以太网络,还有无线网络、Token ring(令牌环)和光纤甚至卫星的支持。第七, Linux 在内核结构的设计中考虑满足系统的可裁剪性的要求, Windows CE 在内核结构的设计中并未考虑满足系统的高度可裁剪性的要求。

嵌入式 Linux OS 与 Windows CE 相比的弱点有:第一,开发难度较高,需要很高的技术实力;第二,核心调试工具不全,调试不太方便,尚没有很好的用户图形界面;第三,与某些商业

OS 一样,嵌入式 Linux 占用较大的内存,当然,人们可以去掉部分无用的功能来减小使用的内存,但是如果不仔细,那么将引起新的问题;第四,有些 Linux 的应用程序需要虚拟内存,而嵌入式系统中并没有或不需要虚拟内存,所以并非所有的 Linux 应用程序都可以在嵌入式系统中运行。

2. Palm OS 与 Windows CE 的比较

3Com 公司的 Palm OS 是掌上电脑市场中较为优秀的嵌入式操作系统,是针对这一市场专门设计的系统。它有开放的操作系统应用程序接口(API),支持开发商根据需要自行开发所需的应用程序,具有十分丰富的应用程序。在掌上电脑市场上独占其霸主地位已久。

从技术层面上讲,Palm OS 是一套专门为掌上电脑开发的操作系统,具有许多 Windows CE 无法比拟的优势;Windows CE 过于臃肿,不适合应用在廉价的掌上电脑中。

Palm OS 是一套具有极强开放性的系统。开发者向用户免费提供 Palm OS 的开发工具,允许用户利用该工具在 Palm OS 基础上方便地编写和修改相关软件。与之相比,Windows CE 的开发工具就显得复杂多了,这使一般用户很难掌握。这也是 Palm OS 与 Windows CE 的另一个主要区别。

从常用 EOS 的大小、可开发定制、互操作性、通用性、实时性及应用领域几个方面,Palm OS、Windows CE 3.0 和嵌入式 Linux 三种操作系统的比较如表 1.1-1 所列。

表 1.1-1 3 种嵌入式操作系统的比较

比较项目	Palm OS	Windows CE 3.0	嵌入式 Linux
大 小	核心几十 KB,整个嵌入环境也不大	核心占 500 KB 的 ROM 和 250 KB 的 RAM。整个 Windows CE 操作系统,包括硬件抽象层(HAL 和 Hardware Abstraction Layer)、Windows CE Kernel、User、GDI、文件系统和数据库,大约为 1.5 MB	核心从几十 KB 到 500 KB。整个嵌入式环境最小为 100 KB 左右,并且以后还将越来越小
可开发定制	可以方便地开发定制	用户开发定制不方便,受 Microsoft 公司限制较多	用户可以方便地开发定制,可以自由卸装用户模块,不受任何限制
互操作性	互操作性强	互操作性比较强,Windows C 可通过 OEM 的许可协议使用于其他设备	互操作性很强
通用性	适应于多种 CPU 和多种硬件平台	适应于多种 CPU 和多种硬件平台	不仅适应于 X86 芯片,并且可以支持 30 多种 CPU 和多种硬件平台,开发和使用都很容易
实时性	比较好	比较好	很好
适用的应用领域	应用领域较广,特别适于掌上电脑的开发	应用领域较广。Windows CE 是为新一代非传统的 PC 设备而设计的,这些设备包括掌上电脑、手持电脑以及用于车载电脑等	由于 Linux 内核结构及功能等原因,嵌入式 Linux 应用领域非常广泛,特别适于进行信息家电的开发

四、结 语

Palm OS、Windows CE、Linux 这 3 种嵌入式操作系统各有不同的特点和不同的用途,但 Linux 比 Palm OS 和 Windows CE 更小、更稳定,而且 Linux 是开放的 OS,在价格上极具竞争力。如今整个市场尚未成型,嵌入式操作系统也未形成统一的国际标准,而且 Linux 的一系列

特征又为开发国产的嵌入式操作系统提供了方便,因此有机会在这个未成熟的市场上占有一席之地。

参 考 文 献

- 1 赵海. 嵌入式 Internet——21 世纪的一场信息技术革命. 北京:清华大学出版社, 2001
- 2 Tanenbaum Andrew S. Modern Operating System. New Jersey: Prentice Hall, 1999
- 3 吕京建,肖海桥. 面向 21 世纪的嵌入式系统综述. <http://www.hol-system.com>
- 4 毛德操,胡希明. Linux 内核源代码情景分析. 杭州:浙江大学出版社, 2001

选自《单片机与嵌入式系统应用》月刊, 2003 年第 5 期

1.2 KEIL RTX51 TINY 内核的分析与应用

南京河海大学(210098) 刘玉宏

一、RTX51 简介

1. RTX51 TINY 特性

RTX51 是 KEIL 公司开发的用于 8051 系列单片机的多任务实时操作系统。它有两个版本, RTX51 FULL 和 RTX51 TINY。

RTX51 TINY 是 RTX51 FULL 的子集, 仅支持按时间片循环任务调度, 支持任务间信号传递, 最多 16 个任务, 可以并行地利用中断。具有以下等待操作: 超时、另一个任务或中断的信号。但它不能进行信息处理, 不支持存储区的分配和释放, 不支持占先式调度。RTX51 TINY 有一个很小的内核, 完全集成在 KEIL C51 编译器中。更重要的是, 它仅占用 800 字节左右的程序存储空间, 可以在没有外扩数据存储器的 8051 系统中运行, 但应用程序仍然可以访问外部存储器。RTX51 TINY 下文简称为内核。

2. RTX51 TINY 的使用

内核完全集成在 KEIL C51 编译器中, 以系统函数调用的方式运行, 因此可以很容易地使用 KEIL C51 语言编写和编译一个多任务程序, 并嵌入到实际应用系统中。内核提供以下函数供应用程序引用:

```
char os_create_task (task_id);  
char os_delete_task (task_id);  
char os_send_signal (task_id);  
char isr_send_signal (task_id);  
char os_clear_signal (task_id);  
char os_running_task_id (void);  
char os_wait (event_sel, ticks, dummy)。  
各函数的函数原型和具体意义, 可以阅读参考文献[1]。
```

二、RTX51 TINY 内核分析

1. 任务状态

RTX51 TINY 的用户任务具有以下几个状态:

RUNNING 任务处于运行中, 同一时间只有一个任务可以处于 RUNNING 状态。

READY 任务正在等待运行, 在当前运行的任务时间片完成之后, RTX51 TINY 运行下一个处于 READY 状态的任务。

WAITING 任务等待一个事件。如果所等待的事件发生, 那么任务进入 READY 状态。

DELETED 任务不处于执行队列。

TIMEOUT 任务由于时间片用完而处于 **TIMEOUT** 状态,并等待再次运行。该状态与 **READY** 状态相似,但由于是内部操作过程,使一个循环任务被切换而被冠以标记。

图 1.2-1 所示为任务状态转换图。

2. 同步机制

为了保证任务在执行次序上的协调,必须采用同步机制。内核用以下事件进行任务间的通信和同步。

SIGNAL: 用于任务之间通信的位,可以用系统函数置位或清除。若一个任务调用了 `os_wait` 函数等待 **SIGNAL** 而 **SIGNAL** 未置位,则该任务被挂起直到 **SIGNAL** 置位,才返回到 **READY** 状态,并可被再次执行。

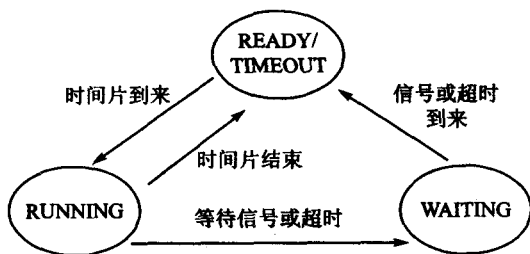


图 1.2-1 任务状态转换图

TIMEOUT: 由 `os_wait` 函数开始的时间延时,其持续时间可由定时节拍数确定。带有 **TIMEOUT** 值调用 `os_wait` 函数的任务将被挂起,直到延时结束,才返回到 **READY** 状态,并可被再次执行。

INTERVAL: 由 `os_wait` 函数开始的时间间隔,其间隔时间可由定时节拍数确定。带有 **INTERVAL** 值调用 `os_wait` 函数的任务将被挂起,直到间隔时间结束,然后返回到 **READY** 状态,并可被再次执行。与 **TIMEOUT** 不同的是,任务的节拍计数器不复位。

3. 调度规则

RTX51 TINY 使用 8051 内部定时器 T0 来产生定时节拍,各任务只在各自分配的定时节拍数(时间片)内执行。当时间片用完后,切换至下一任务运行,因此各任务是并发执行的。

调度规则如下:若任务调用了 `os_wait` 函数,且特定事件还没有发生,任务执行比循环切换所规定的时间长,则运行任务被中断;若没有其他任务正在运行,且任务处于 **READY** 或 **TIMEOUT** 状态下等待运行,则另一个任务开始。

4. 任务控制块

为了能描述和控制任务的运行,内核为每个任务定义了称作任务控制块的数据结构,主要包括 3 项内容:

`ENTRY[task_id]` task_id 任务的代码入口地址,位于 **CODE** 空间,2 字节为一个单位。

`STKP[taskid]` taskid 任务所使用堆栈栈底位置,位于 **IDATA** 空间,1 字节为一个单位。

`STATE[taskid].timer` 和 `STATE[taskid].state` 前者表示任务的定时节拍计数器,在每一次定时节拍中断后都自减一次;后者表示任务状态寄存器,用其各个位来表示任务所处的状态。位于 **IDATA** 空间,以 2 字节为一个单位。

5. 存储器管理

内核使用了 KEIL C51 编译器的对全局变量和局部变量采取静态分配存储空间的策略,因此,存储器管理简化为堆栈管理。内核为每个任务都保留一个单独的堆栈区,全部堆栈管理都在 **IDATA** 空间进行。为了给当前正在运行的任务分配尽可能大的栈区,各个任务所用的堆栈位置是动态的,并用 `STKP[taskid]` 来记录各任务的堆栈栈底位置。当堆栈自由空间小于 **FREESTACK**(默认为 20 字节)时,就会调用宏 `STACK_ERROR` 进行堆栈出错处理。

在以下情况会进行堆栈管理:

- 任务切换,将全部自由堆栈空间分配给正在运行的任务;
- 任务创建,将自由堆栈空间的 2 字节分配给新创建的任务 task_id,并将 ENTRY[task_id]放入其堆栈;
- 任务删除,回收被删除的任务 task_id 的堆栈空间并转换为自由堆栈空间。

堆栈管理如图 1.2-2 所示。

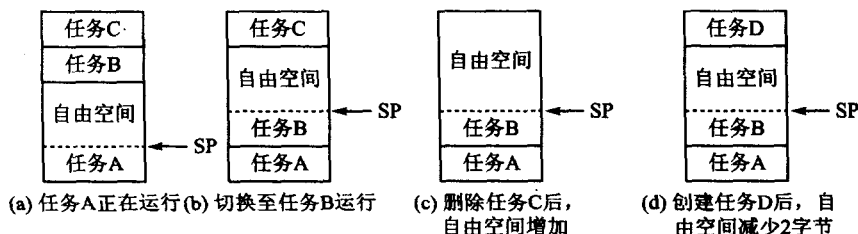


图 1.2-2 堆栈管理

三、代码分析

内核代码用汇编语言写成,可读性差,但代码效率较高,主要由 conf_tny. a51 和 rtxtny. a51 两个源程序文件组成。前者是一个配置文件,用来定义系统运行所需要的全局变量和堆栈出错的宏 STACK_ERROR。对于这些全局变量和宏,用户都可以根据自己的系统配置灵活修改。后者是系统内核,完成系统调用的所有函数。

1. 主程序 main

主程序 main 的主要任务是初始化各任务堆栈栈底指针 STKP、状态字 STATE 和定时器 T0,创建任务 0 并将其导入运行队列。这个过程加上 KEIL C51 的启动代码 CSTARTUP 正是一般嵌入式系统中 BSP 所做的工作。

2. 定时器 T0 中断服务程序

内核使用定时器 T0 作为定时节拍发生器,是任务切换、时间片轮转的依据。中断服务程序有 3 个任务。

① 更新各个任务节拍数:将 STATE[taskid]. timer 减 1,若某任务超时 (STATE[taskid]. timer=0),并且该任务正在等待超时事件,则将该任务置为 READY 状态,使其返回任务队列。

② 检查自由堆栈空间:若自由堆栈空间范围小于 FREESTACK(默认为 20 字节)时,可以调用宏 STACK_ERROR 进行堆栈出错处理。

③ 检查当前任务(处于 RUNNING 状态)的时间片是否到时。若当前任务的时间片到时,将程序转到任务切换程序段(TASKSWITCHING)切换下一任务运行。

程序流程如图 1.2-3 所示。

3. 任务切换程序段

这个程序段是整个内核中最核心的一个,主要功能是完成任务切换。它共有两个入口: TASKSWITCHING 和 SWITCHINGNOW。前者供定时器 T0 的中断服务程序调用,后者能供系统函数 qs_delete 和 os_wait 调用。相应地,也有两个不同的出口。

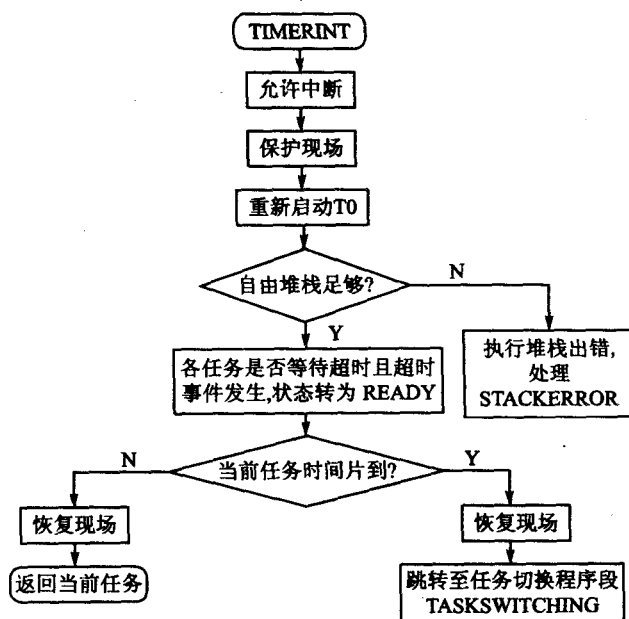


图 1.2-3 T0 中断服务程序

其基本工作流程是:首先将当前任务置为 TIMEOUT 状态,等待下一次时间片循环;其次找到下一个处于 READY 状态的任务并使其成为当前任务;然后进行堆栈管理,将自由堆栈空间分配给该任务。清除使该任务进入 READY 或 TIMEOUT 状态的相关位后,执行该任务。流程框图如图 1.2-4 所示。

4. os_wait 程序段

主要完成 os_wait 函数。任务调用 os_wait 函数,挂起当前任务,等待一个或几个间隔(K_IVL)、超时(K_TMO)、信号(K_SIG)事件。如果所等待的事件已经发生,那么继续执行当前任务;如果所等待的事件没有发生,那么置相应的等待标志后,挂起该任务,将任务切换程序段(SWITCHINGNOW)切换到下一任务。

5. 其他程序段

其他程序段主要完成 os_create_task、os_delete_task 函数和有关信号处理的 os_send_signal、isr_send_signal、os_clear_signal 函数。这些函数功能相对比较简单,若根据上述存储器管理策略进行堆栈的分配和删除,并改变任务的状态字 STATE[tasked].state,则使任务处于不同的状态。

对于以上所有程序段,若涉及到任务状态字操作,必须关中断,以防止和定时器 T0 同时操作任务状态字。

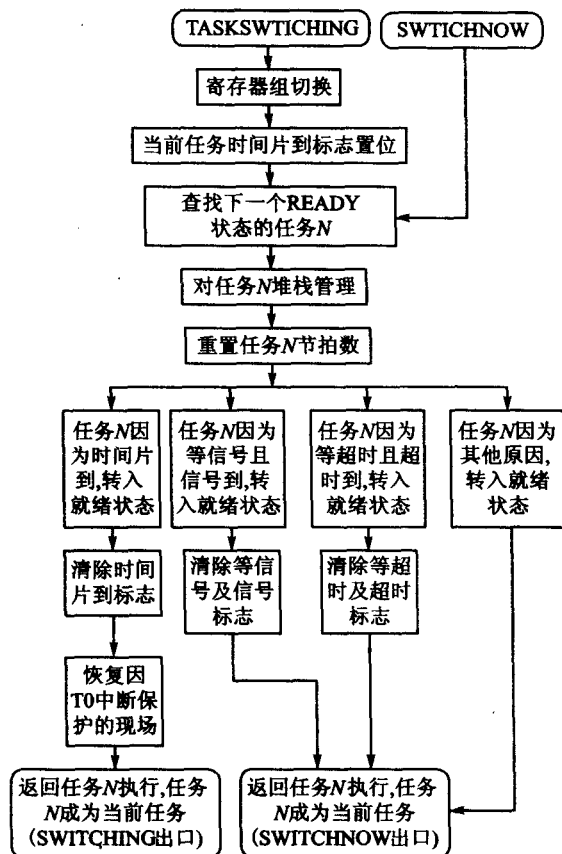


图 1.2-4 任务切换程序

四、结 语

从以上分析可以看到,这个内核简洁、高效,非常适合于运行在资源较少的单片机上。根据其设计思想,很容易把它移植到其他单片机上。但是它也有缺陷,例如:不支持外部任务切换;不支持用户使用定时器 T0 等。这些缺陷的存在限制了任务切换的灵活性。在本文的结尾要特别感谢 C51BBS 的各位网友给我提供了很大的帮助,使本文能顺利成文。

参 考 文 献

- 1 徐爱钧,彭秀华. 单片机高级语言 C51 Windows 环境编程与应用. 北京:电子工业出版社,2001

选自《单片机与嵌入式系统应用》月刊,2003 年第 10 期

1.3 中间件技术及其发展展望

广州华南师范大学计算机系(510631) 张晓刚

随着网络经济时代的来临,企业正在面临着信息化的挑战。如何利用信息技术,将分散的、不同的应用系统集成成为一个高效的整体,使企业能够更高效、灵活地运转,成为企业成败的关键。现代企业都希望通过 Internet 技术实施面向业务流程的管理、重整和自动化(BPM/BPR/BPA),即实现所谓的电子商务。为了实现这一目标,企业的应用集成(EAI)成为当前的一个技术关键和热点,而 EAI 的核心技术就是通过中间件完成跨平台的应用集成。在现代企业应用中,系统结构基本上都采用多层结构,这就更加突出了中间件的重要性。而且中间件技术的成熟使真正意义上的复杂分布式企业应用系统的实施成为可能。

一、中间件的概念、原理及特点

1. 中间件的概念

从软件系统的角度来看,中间件是处于应用软件和系统软件之间的一类软件。它主要是在分布式系统中用来帮助管理各种复杂的异构的成分,并且提供了一个贯穿于分布式系统中的通用的应用抽象。与操作系统提供的应用程序接口(API)相比,中间件提供的是一种更高层的应用抽象,而且中间件的使用很明显地减少了应用程序的负担。

中间件框架的实施可以掩盖分布式系统中必须处理的各种异构问题,例如网络和硬件的异构问题,而且绝大多数中间件也可以掩盖操作系统和程序语言的异构问题。这样,中间件所提供的应用抽象将更具有透明性。因此,与在一个没有操作系统的裸机上进行编程的困难程度相比,不使用中间件而去设计和实施一个分布式系统将更为困难,尤其是在异构操作系统的环境中。

2. 中间件的工作原理

一般情况下,中间件是这样工作的:分布式应用程序需要一定的数据或服务,而这些数据或服务在这样一个分布式系统中可能处于运行不同操作系统和特定数据库管理系统的服务器中。此应用程序中负责寻找数据的部分只须访问一个中间件系统,由中间件完成到网络中查找数据源或服务,进而传输应用程序的请求、重组答复信息,最后将结果送回应用程序的任务。中间件工作原理如图 1.3-1 所示。

3. 中间件的特点

易集成:中间件能够很好地无缝连入应用开发环境中,应用程序可以很容易地定位和共享中间件提供的逻辑和数据。

易移植:中间件使与平台有关的细节对应用程序透明,因此可以在不改变应用程序代码的情况下改换计算机低层硬件、操作系统或通信协议。

易演进:中间件所实现的功能对应用程序来说是透明的,因此可对局部进行改进而又不会影响到系统的其他部分。

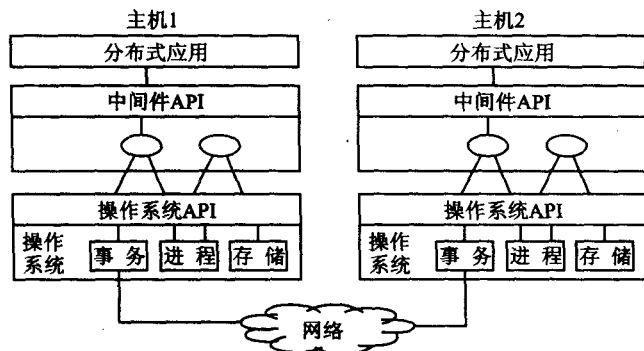


图 1.3-1 中间件工作原理图

高可靠：中间件应该是高可靠的，须提供接管和恢复功能，以保证事务及关键性业务不被丢失。

易使用：中间件能和同构或异构环境下的多种数据源通信，同时它还能够管理数据之间的公共逻辑约束，能够使用户从复杂的平台、网络、数据库选择中解放出来。

二、中间件的分类

中间件的种类很多，根据中间件在系统中所起的作用和采用的不同技术，大致划分为以下5类。

(1) 数据库中间件

数据库中间件(DM, Database Middleware)在所有的中间件中是应用最广泛、技术最成熟的一种。它允许应用程序与本地或者异地的数据库进行通信，并提供了一系列的应用程序接口API。当然在多数情况下，这些API都是隐藏在开发工具中，不被程序员直接使用。最典型的例子是ODBC和JDBC，它们都是基于数据库的中间件标准。

不过在数据库中间件处理模型中，数据库是信息存储的核心单元，中间件完成通信的功能。这种方式虽然是灵活的，但是并不适合于一些要求高性能处理的场合。因为它需要大量的数据通信，而且当网络发生故障时，系统将不能正常工作。可见，系统的灵活性提高是以处理性能的降低为代价的。

(2) 远程过程调用中间件

远程过程调用(RPC, Remote Procedure Call)中间件沿用了大多数程序员都非常熟悉的编程模式——客户/服务器模式。程序员就像调用本地过程一样在程序中调用远程过程，启动远程过程的运行，然后将运行结果返回给本地程序。不仅如此，远程过程调用还可以将程序的控制传递到远端的服务器中。

RPC的灵活性使它有比数据库中间件更广泛的应用，可以应用在更复杂的客户/服务器计算环境中。远程过程调用的灵活性还体现在它的跨平台性上，这种调用是可以跨不同操作系统平台的，而程序员在编程时并不需要考虑这些细节。

RPC也有一些缺点，主要是因为RPC一般用于应用程序之间的通信，而且采用的是同步通信方式，所以对于比较小型的简单应用还是比较适合的。但是对于一些大型且要求异步通信的应用来讲，这种方式不是很适合。因为此时程序员需要考虑网络或者系统故障，处理并发

操作、缓冲、流量控制以及进程同步等一系列复杂问题。

(3) 面向消息中间件

消息中间件(MOM, Message Oriented Middleware)提供了一个完整的处理环境,允许开发者及用户连接不同系统之间的数据和代码,或采用一致的界面进行应用处理的互连。它提供了一个高层应用接口,为不同系统提供操作核心。消息中间件的工作主要是通过将信息以消息的方式在程序间传递来完成的。

消息中间件的优点主要是能够在客户和服务器之间提供同步和异步的连接,并且在任何时刻都可以将消息进行传送或者存储、转发,这也是它比远程过程调用更具优势的原因。另外,消息中间件不会占用大量的网络带宽,可以跟踪事务,并且通过将事务存储到磁盘上,可实现网络故障时的系统恢复。当然,和远程过程调用相比,消息中间件不支持程序控制的传递,不过这种功能和它的优势比起来是无关紧要的。消息中间件适用于需要在多个进程之间进行可靠数据传送的分布式环境。

(4) 基于对象请求代理的中间件

基于对象请求代理(ORB, Object Request Broker)的中间件主要采用面向对象的技术,它可以看作是与编程语言无关的面向对象的 RPC 应用。ORB 被视为从面向对象过渡到分布式计算的强大推动力量。从管理和封装的模式看,对象请求代理和远过程调用有些类似,但对对象请求代理可以包含比远过程调用和消息中间件更复杂的信息,并且可以适用于非结构化的或者非关系型的数据。

目前,ORB 存在二个彼此竞争的标准: CORBAORB 和 DCOMORB。当使用 ORB 时,IDL(Interface Define Language)用于定义对象之间的接口。它类似于 RPC 中的 IDL 定义过程的接口。ORB 特别适用于对象接口变化不频繁,不会导致代码经常被重新编译及链接的情况。

(5) 事务处理中间件

事务处理中间件(TPM, Transaction Processing Monitor)是一种复杂的中间件产品,是针对复杂环境下分布式应用的速度和可靠性要求而实现的。它给程序员提供了一个事务处理的 API。程序员可以使用这个程序接口编写高速而且可靠的分布式应用程序,即基于事务处理的应用程序。

事务处理中间件向用户提供一系列的服务,例如应用管理、控制管理以及应用程序间的消息传递等。常见的功能包括全局事务协调、事务的分布式二段提交、资源管理器支持、故障恢复、高可靠性、网络负载均衡等。

三、当前主要的中间件技术支持平台

当前主流的中间件技术支持平台主要有 OMG 的 CORBA、Sun 的 J2EE 和 Microsoft DNA 2000。它们都是支持服务器端中间件技术开发的平台,但都有其各自的特点,下面分别阐述。

1. OMG 的 CORBA

COBRA 分布计算技术是 OMG 组织在众多开放系统平台厂商提交的分布对象互操作内容的基础上制定的公共对象请求代理体系规范。

COBRA 标准主要分为 3 个层次:对象请求代理、公共对象服务和公共设施。最底层是对

象请求代理 ORB,它规定了分布对象的定义(接口)和语言映射,实现对象间的通信和互操作,是分布对象系统中的“软总线”;在 ORB 之上定义了很多公共服务,可以提供诸如并发服务、名字服务、事务(交易)服务、安全服务等各种各样的服务;最上层的公共设施则定义了组件框架,提供可直接为业务对象使用的服务,规定业务对象有效协作所需的协定规则。

CORBA 的技术具有模型完整性和先进性。它独立于系统平台和开发语言,被支持程度高,已逐渐成为分布计算技术的标准。CORBA 的缺点是庞大而复杂,并且技术和标准的更新相对较慢,COBRA 规范从 1.0 升级到 2.0 所花的时间非常短,而再往上的版本的发布就相对十分缓慢了,在具体的应用中使用不是很多。

2. Sun 的 J2EE

为了推动基于 Java 的服务器端应用开发,Sun 在 1999 年底推出了 Java2 技术及相关的 J2EE 规范。它是完整的基于 Java 语言开发面向企业分布应用的规范。J2EE 的目标是提供平台无关性和可移植性,支持安全的并发访问。J2EE 完全基于 Java 的开发服务器端中间件的标准。

EJB 是 Sun 推出的基于 Java 的服务器端构件规范 J2EE 的一部分。自从 J2EE 推出之后,EJB 得到了广泛的发展,已经成为应用服务器端的标准技术。EJB 技术是在 Java Bean 本地构件基础上发展起来的面向服务器端分布应用构件技术。EJB 给出了系统的服务器端分布构件规范,这包括了构件、构件容器的接口规范以及构件打包、构件配置等标准规范内容。EJB 技术的推出,使得用 Java 基于构件方法开发服务器端分布式应用成为可能。从企业应用多层结构的角度,EJB 是业务逻辑层的中间件技术。与 JavaBeans 不同,它提供了事务处理的能力。自从三层结构提出以后,中间层,也就是业务逻辑层,已成为处理事务的核心,从数据存储层分离,取代了存储层的大部分地位。从分布式计算的角度,EJB 像 CORBA 一样,提供了分布式技术的基础,提供了对象之间的通信手段。

3. Microsoft DNA 2000

Microsoft DNA 2000 是 Microsoft 在推出 Windows2000 系列操作系统平台基础上,在扩展了分布计算模型以及改造 Back Office 系列服务器端分布计算产品后发布的新的分布计算体系结构和规范。它融合了当今最先进的分布计算理论和思想,例如事务处理、可伸缩性、异步消息队列、集群等内容。DNA 使开发基于 Microsoft 平台的服务器构件应用成为可能,其中,如数据库事务服务、异步通信服务和安全服务等都由底层的分布对象系统提供。

COM/DCOM 技术是 Microsoft 独家的技术,从 DDE、OLE 到 ActiveX 等,提供了中间件开发的基础。它在 DNA 2000 分布计算结构基础上,展现了一个全新的分布构件应用模型。COM 通过底层的远程支持使构件技术延伸到了分布应用领域。COM/DCOM/COM+ 更将其扩充为面向服务器端分布应用的业务逻辑中间件。通过 COM+ 的相关服务设施,如负载均衡、内存数据库、对象池、构件管理与配置等,COM/DCOM 将 COM、DCOM 和 MTS 的功能有机地统一在一起,形成了一个功能强大的构件应用体系结构;而且 DNA 2000 是单一厂家提供的分布对象构件模型,开发者使用的是同一厂家提供的系列开发工具。这比组合多家开发工具更有吸引力。

但是它的不足是依赖于 Microsoft 的操作系统平台,因而在其他开发系统平台(如 Unix、Linux)上不能发挥作用。

四、中间件技术的发展与展望

目前,中间件的发展速度是惊人的,它已与操作系统、数据库一起成为基础软件领域的三大支柱。中间件作为构筑企业信息系统和电子商务系统的基石和核心技术,今后将会如何发展,这是业界十分关注的问题。

1. 进一步规范

中间件的发展过程中,做得最好的一件事情就是规范的制订。对于不同类型的中间件,目前都有一些规范可以遵循,如消息类的 JMS,对象类的 CORBA、COM/DCOM,交易类的 XA、OTS、JTA/JTS,应用服务器类的 J2EE,数据访问类的 ODBC、JDBC,Web 服务方面的 Soap、WSDL、UDDI 等。这些规范的建立极大地促进了中间件技术的发展,同时保证了系统的扩展性、开发性和互操作。当然,中间件技术的规范还将进一步完善和扩充,以满足各种各样的中间件技术的发展需求。

2. 自身结构的开放和可配置化

传统的中间件是一个黑箱模式的体系框架结构,它不能很好地满足复杂分布式系统中要在一定程度上把分布式应用深入到中间件内部去的要求。也就是说,如果中间件内部的功能能够具有一定的开放性和可配置性,那么它能够使得用它组成的复杂分布式系统更具灵活性和透明性。因此,希望能够在一定范围内对中间件的内部进行控制,以便让中间件被更加灵活地使用。现在有一个解决办法就是,在中间件内部引入反射机制,以达到这种开放性和可配置性。

反射的概念是由 Smith 在 1982 年首次提出的,主要是指程序可以访问、检测和修改它本身的状态或行为的一种能力。在计算机科学领域,反射是指一类应用,它们能够自描述和自控制。也就是说,这类应用通过采用某种机制来实现对自己行为的描述(Self-representation)和监测(Examination),并能根据自身行为的状态和结果,调整或修改应用所描述行为的状态和相关的语义。因此,把采用了反射机制实现的开放式的中间件系统称为反射中间件。反射中间件将是中间件技术的一个具有重要意义的发展趋势。

3. 构件化和松耦合

除了已经得到较普遍应用的 CORBA、DCOM 等适应 Intranet 的构件技术外,随着企业业务流程整合电子商务应用的发展,中间件技术将趋于面向 Web 和松散耦合的方式。基于 XML 和 Web 服务的中间件技术,使不同系统之间、不同应用之间交互建立在非常灵活的基础上。XML 是一种可扩展的源标识语言,提供了一种定义新的标识语言的标准。XML 技术非常适合于异构系统间的数据交换,因此国际上已经被普遍作为电子商务的数据标准。而 Web 服务作为基于 Web 技术的构件,在流程中间件的控制和集成下可以灵活、动态地被组织成为跨企业的商务应用。二者的完美结合,将给分布式的信息集成系统带来前所未有的生命力。

4. 平台化

目前,一些大的中间件厂商在已有的中间件产品基础上提出了完整的面向互联网的软件平台战略计划和应用解决方案。Sun 公司是最早提出“网络就是计算机”的公司,并一直致力于向企业提供受到广泛欢迎的网络软件,对互联网的应用和发展发挥了重要作用。IBM 公司提出了面向网络应用的“旧金山计划”,以 WebSphere、DB2、Tivoli、Domino 等 4 大品牌组成基