

王晓薇 主编
孙静 刘天华 副主编

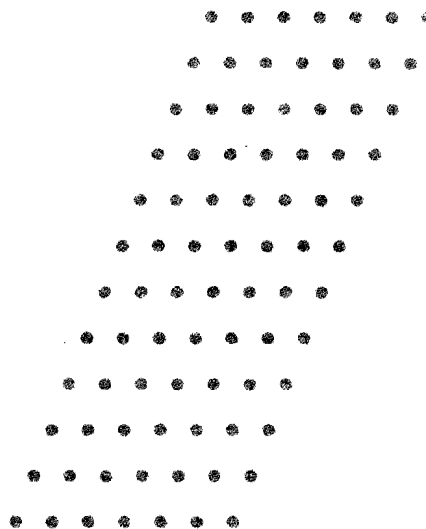
嵌入式操作系统

$\mu\text{C}/\text{OS-II}$ 及应用开发

21世纪高等学校嵌入式系统专业规划教材

王晓薇 主编
孙静 刘天华 副主编
姜岩 侯锟 编著

嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 及应用开发



清华大学出版社
北京

内 容 简 介

本书以理论为核心,以实用为导向,最大的特色就是将 $\mu\text{C}/\text{OS-II}$ 操作系统的原理和应用有机地融合到这一本书中,使学生既能掌握理论,又能懂得应用。

本书先介绍了 PC 上 $\mu\text{C}/\text{OS-II}$ 的开发环境 BC45,在这个环境下演示一个 $\mu\text{C}/\text{OS-II}$ 的实例,使读者从整体上理解 $\mu\text{C}/\text{OS-II}$ 。然后从操作系统的原理上详细讲述 $\mu\text{C}/\text{OS-II}$ 这个实时操作系统的实现原理,对其中的内核、任务的管理、任务的同步和通信、时钟和中断进行了详细的论述,理论讲述后附有该理论的实验例程及实现方法。最后介绍了 $\mu\text{C}/\text{OS-II}$ 的移植方法,并从应用的角度描述了一个基于 $\mu\text{C}/\text{OS-II}$ 的综合开发案例,使读者在应用开发中真正会用 $\mu\text{C}/\text{OS-II}$,掌握 $\mu\text{C}/\text{OS-II}$ 的应用技巧。

本书适合作为嵌入式系统原理及应用的学习教材,同时适合作为高等院校计算机相关专业嵌入式操作系统教材或参考书,适合相关学科的本、专科学生、高职高专及成教类学生阅读,也可供嵌入式操作系统应用开发人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 及应用开发 / 王晓薇主编. —北京:清华大学出版社,2012.8

21 世纪高等学校嵌入式系统专业规划教材

ISBN 978-7-302-28472-7

I. ①嵌… II. ①王… III. ①实时操作系统—程序设计—高等学校—教材 IV. ①TP316.2

中国版本图书馆CIP数据核字(2012)第065081号

责任编辑:梁颖 高买花 薛阳

封面设计:傅瑞学

责任校对:李建庄

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.75 字 数:437千字

版 次:2012年8月第1版 印 次:2012年8月第1次印刷

印 数:1~3000

定 价:29.00元

产品编号:046271 01

出版说明

嵌入式计算机技术是 21 世纪计算机技术两个重要发展方向之一，其应用领域相当广泛，包括工业控制、消费电子、网络通信、科学研究、军事国防、医疗卫生、航空航天等方方面面。我们今天所熟悉的电子产品几乎都可以找到嵌入式系统的影子，它从各个方面影响着我们的生活。

技术的发展和生产力的提高，离不开人才的培养。目前国内外各高等院校、职业学校和培训机构都涉足了嵌入式技术人才的培养工作，高校及其软件学院和专业的培训机构更是嵌入式领域高端人才培养的前沿阵地。国家有关部门针对专业人才需求大增的现状，也着手开发“国家级”嵌入式技术培训项目。2006 年 6 月底，国家信息技术紧缺人才培养工程（NITE）在北京正式启动，首批设定的 10 个紧缺专业中，嵌入式系统设计与软件开发、软件测试等 IT 课程一同名列其中。嵌入式开发因其广泛的应用领域和巨大的人才缺口，其培训也被列入国家商务部门实施服务外包人才培训“千百十工程”，并对符合条件的人才培训项目予以支持。

为了进一步提高国内嵌入式系统课程的教学水平和质量，培养适应社会经济发展需要的、兼具研究能力和工程能力的高质量专业技术人才。在教育部相关教学指导委员会专家的指导和建议下，清华大学出版社与国内多所重点大学共同对我国嵌入式系统软硬件开发人才培养的课程框架和知识体系，以及实践教学内容进行了深入的研究，并在该基础上形成了“嵌入式系统教学现状分析及核心课程体系研究”、“微型计算机原理与应用技术课程群的研究”、“嵌入式 Linux 课程群建设报告”等多项课程体系的研究报告。

本系列教材是在课程体系的研究基础上总结、完善而成，力求充分体现科学性、先进性、工程性，突出专业核心课程的教材，兼顾具有专业教学特点的相关基础课程教材，探索具有发展潜力的选修课程教材，满足高校多层次教学的需要。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

（1）反映嵌入式系统学科的发展和专业教育的改革，适应社会对嵌入式人才的培养需求，教材内容坚持基本理论的扎实和清晰，反映基本理论和原理的综合应用，在其基础上强调工程实践环节，并及时反映教学体系的调整 and 教学内容的更新。

（2）反映教学需要，促进教学发展。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

（3）实施精品战略，突出重点。规划教材建设把重点放在专业核心（基础）课程的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现工程型和应用型的专业教学内容和课程体系改革成果的教材。

（4）支持一纲多本，合理配套。专业核心课和相关基础课的教材要配套，同一门课程可以有多本具有各自内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、

教学参考书，文字教材与软件教材的关系，实现教材系列资源的配套。

(5) 依靠专家，择优落实。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的、以老带新的教材编写队伍才能保证教材的编写质量，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校嵌入式系统专业规划教材
联系人：魏江江 weijj@tup.tsinghua.edu.cn

前 言

$\mu\text{C}/\text{OS-II}$ 是一种可移植的、可裁剪的、抢占式的、典型的实时多任务操作系统内核。它被广泛应用于微处理器、微控制器和数字信号处理器。

嵌入式操作系统是与应用紧密结合的，脱离实际应用去讲述嵌入式操作系统，学生不容易明白如何去使用这样的操作系统，如何在这样的操作系统上进行应用程序的开发，那么就失去了学习这些理论的意义。

本书将嵌入式操作系统的原理与典型的嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 结合起来，并给出应用的实例，使得学生在学习理论的同时掌握了应用，既提高了学生的实际动手能力，又满足了应用型计算机人才培养的需要。

本书具有以下特色和价值。

(1) 具有结构优化、内容精炼、重点突出的优点，强调原理与典型的嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 结合，并给出应用的实例。

(2) 教材中介绍了目前广泛使用的、成熟的新技术，较好地体现了课程内容的先进性。

(3) 教材每章配有恰当的应用实例，着眼于提高学生分析问题和解决问题的能力，较好地体现了课程教学的实用性，着眼于提高学生对嵌入式操作系统的开发与设计能力。

(4) 本教材所有的程序均由 C 语言给出，体现了软件的可移植性。

(5) 教材配有电子教案，学生可从教学网站 (http://210.30.208.205/homepage/common/index_jpk.jsp) 和从清华大学出版社的网站 (www.tup.com.cn) 上下载，以方便学生课后的学习和复习。

本书第 1~3 章由王晓薇和姜岩编写，第 5~7 章由孙静编写，第 4、第 8、第 9 章由刘天华和侯锟编写，第 10 章和附录由张勇编写，全书由王晓薇和孙静统稿。

在本书的编写和使用过程中，得到了许多教师和同行的帮助，在此表示感谢。还要感谢清华大学出版社，有了出版社的大力支持才使本书能够很快与读者见面。本书还参考和引用了有关方面的书籍，其来源都在参考文献中列出，在此对有关作者表示感谢。

限于编者的学识水平，本书中难免有疏漏和不当之处，敬请广大同行及读者指正。同时也欢迎读者，尤其是采用本书进行教学的教师和学生，共同探讨相关教学内容、教学方法等问题。敬请广大师生和读者通过电子信箱与编者联系 (wangxwvv@gmail.com)。

本书所用的免费软件开发工具和教材中的开发实例可到清华大学出版社网站下载，或通过邮箱联系。

编者

2012 年 6 月

目 录

第 1 章 嵌入式操作系统概述	1
1.1 操作系统概述	1
1.1.1 操作系统的作用	1
1.1.2 操作系统的特征	2
1.1.3 操作系统的发展	2
1.2 操作系统的体系结构	3
1.2.1 层次结构	3
1.2.2 微内核结构	4
1.3 嵌入式操作系统概述	5
1.3.1 嵌入式操作系统的演变	5
1.3.2 嵌入式操作系统的特点	6
1.3.3 嵌入式操作系统与通用操作系统的区别	7
1.4 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 概述	8
1.4.1 $\mu\text{C}/\text{OS-II}$ 的特点	8
1.4.2 $\mu\text{C}/\text{OS-II}$ 的文件结构	9
小结	10
习题	11
第 2 章 $\mu\text{C}/\text{OS-II}$ 的入门知识	12
2.1 开发工具	12
2.1.1 Hello World 程序	12
2.1.2 BCC 编译器	14
2.1.3 TLINK 链接器	18
2.1.4 TASM 汇编语言编译器	19
2.2 make 和 makefile	20
2.2.1 简单 makefile 的书写规则	20
2.2.2 make 命令	22
2.3 初识 $\mu\text{C}/\text{OS-II}$	24
小结	33
习题	33
第 3 章 $\mu\text{C}/\text{OS-II}$ 的内核结构	34
3.1 $\mu\text{C}/\text{OS-II}$ 任务的描述	34

3.1.1	任务的定义	34
3.1.2	任务的基本状态	36
3.1.3	任务控制块	38
3.1.4	任务堆栈	43
3.2	任务调度	45
3.2.1	基本概念	45
3.2.2	调度的时机	48
3.2.3	操作系统常用的调度算法	49
3.2.4	实时系统中的调度	50
3.2.5	$\mu\text{C}/\text{OS-II}$ 的任务调度	51
3.3	$\mu\text{C}/\text{OS-II}$ 的系统任务	60
3.3.1	空闲任务	60
3.3.2	统计任务	61
3.4	$\mu\text{C}/\text{OS-II}$ 的初始化和任务的启动	64
3.4.1	$\mu\text{C}/\text{OS-II}$ 的初始化	64
3.4.2	$\mu\text{C}/\text{OS-II}$ 的启动	67
小结		68
习题		69
第 4 章	任务管理	70
4.1	任务的基本操作	70
4.1.1	任务创建	70
4.1.2	任务删除	74
4.1.3	任务挂起与恢复	76
4.1.4	其他任务管理函数	78
4.2	应用举例	81
4.2.1	任务的状态转变举例	81
4.2.2	堆栈功能检测应用举例	87
小结		93
习题		94
第 5 章	中断管理	95
5.1	中断概述	95
5.1.1	中断的基本概念	95
5.1.2	中断处理过程	96
5.1.3	中断评价指标	97
5.2	$\mu\text{C}/\text{OS-II}$ 的中断	98
5.2.1	$\mu\text{C}/\text{OS-II}$ 的中断服务子程序 ISR	98
5.2.2	$\mu\text{C}/\text{OS-II}$ 的中断级的任务切换	100

小结	101
习题	101
第 6 章 时间管理	102
6.1 时钟	102
6.1.1 时钟硬件	102
6.1.2 时钟软件	103
6.1.3 系统时钟	104
6.2 时钟中断服务程序	104
6.3 $\mu\text{C}/\text{OS-II}$ 的时钟	105
6.3.1 $\mu\text{C}/\text{OS-II}$ 的时钟中断服务子程序 ISR	105
6.3.2 时钟节拍函数 OSTimeTick()	106
6.3.3 时钟节拍任务	107
6.3.4 时钟节拍中断服务子程序举例	107
6.4 $\mu\text{C}/\text{OS-II}$ 时间管理	114
6.4.1 $\mu\text{C}/\text{OS-II}$ 时间管理的基本操作	114
6.4.2 $\mu\text{C}/\text{OS-II}$ 时间管理应用举例	116
小结	122
习题	123
第 7 章 任务的同步与通信	124
7.1 同步和通信的基本概念	124
7.1.1 竞争条件	124
7.1.2 同步与通信	125
7.1.3 共享资源	126
7.1.4 可重入函数	126
7.1.5 临界区	127
7.1.6 $\mu\text{C}/\text{OS-II}$ 的临界区	128
7.2 任务通信的数据结构——事件控制块	129
7.2.1 事件	129
7.2.2 事件控制块 ECB	130
7.2.3 空事件控制块链表	133
7.2.4 事件控制块的操作	134
7.3 信号量	135
7.3.1 信号量概述	135
7.3.2 $\mu\text{C}/\text{OS-II}$ 信号量的数据结构	135
7.3.3 信号量的操作	136
7.3.4 信号量应用举例	142
7.4 互斥型信号量	145

7.4.1	互斥型信号量概述	145
7.4.2	互斥型信号量的数据结构	146
7.4.3	互斥型信号量的操作	146
7.4.4	优先级反转	152
7.4.5	优先级反转应用举例	154
7.5	事件标志组	162
7.5.1	事件标志组概述	162
7.5.2	事件标志组的数据结构	164
7.5.3	事件标志组的操作	165
7.5.4	事件标志组应用举例	171
7.6	消息邮箱	176
7.6.1	消息邮箱概述	176
7.6.2	消息邮箱的数据结构	176
7.6.3	消息邮箱的操作	177
7.6.4	消息邮箱应用	182
7.7	消息队列	187
7.7.1	消息队列概述	187
7.7.2	消息队列的数据结构	188
7.7.3	消息队列的操作	189
7.7.4	消息队列应用举例	196
小结		202
习题		202
第 8 章	内存管理	203
8.1	分区内存管理技术	203
8.1.1	单一分区内存管理	203
8.1.2	固定大小的多分区管理	204
8.1.3	动态分区管理	204
8.2	实时系统的内存管理	206
8.2.1	存储管理原理	206
8.2.2	动态内存管理的方法	207
8.3	$\mu\text{C}/\text{OS-II}$ 内存管理	209
8.3.1	$\mu\text{C}/\text{OS-II}$ 内存管理概述	209
8.3.2	$\mu\text{C}/\text{OS-II}$ 内存管理的数据结构	210
8.4	$\mu\text{C}/\text{OS-II}$ 内存管理的基本操作	211
8.5	$\mu\text{C}/\text{OS-II}$ 内存管理应用举例	215
小结		222
习题		222

第 9 章 $\mu\text{C}/\text{OS-II}$ 在 ARM7 上移植	223
9.1 $\mu\text{C}/\text{OS-II}$ 移植的条件	223
9.2 编译器的选择	225
9.3 移植	226
9.3.1 OS_CPU.H 文件	226
9.3.2 OS_CPU_C.C 文件	227
9.3.3 OS_CPU_A.ASM 文件	232
9.4 移植测试	240
9.4.1 确保 C 编译器、汇编编译器及链接器正常工作	240
9.4.2 $\mu\text{C}/\text{OS-II}$ 操作系统基础上实现多任务机制	243
小结	245
习题	245
第 10 章 室内智能节电综合监控系统设计	246
10.1 室内智能节电综合监控系统简介	246
10.2 系统功能需求与性能指标	246
10.3 系统方案设计	247
10.4 系统硬件结构设计	248
10.5 系统软件设计	249
10.5.1 $\mu\text{C}/\text{OS-II}$ 在 LPC1768 微处理器上的移植	249
10.5.2 基于 $\mu\text{C}/\text{OS-II}$ 的任务的划分和设计	254
小结	260
附录 配置手册	261
参考文献	269

第 1 章 嵌入式操作系统概述

1.1 操作系统概述

操作系统（OS）是一种为应用程序提供服务的系统软件，是一个完整计算机系统的有机组成部分。从计算机系统层次结构来看，操作系统位于计算机硬件之上、应用软件之下，所以也把它称为应用软件的运行平台。

本章主要内容：

- 操作系统的作用和特征。
- 操作系统的体系结构。
- 嵌入式操作系统特点概述。
- 嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 的特点。

1.1.1 操作系统的作用

我们可以从不同的角度来分析 OS 的作用，从一般用户的角度，可把 OS 看做是用户与计算机硬件系统之间的接口；从资源管理角度，可把 OS 看做计算机系统资源的管理者。

1. OS 作为用户与计算机硬件系统之间的接口

OS 作为用户与计算机硬件系统之间的接口的含义是：OS 处于用户与计算机硬件系统之间，用户通过 OS 来使用计算机系统。或者说，用户在 OS 的帮助下能够方便、快捷、安全可靠地操纵计算机硬件和运行自己的程序。应当注意，OS 是一个系统软件，因而这种接口是软件接口，如图 1-1 所示。

OS 在计算机应用软件与计算机硬件系统之间，它屏蔽了计算机硬件工作的一些细节，并对系统中的资源进行有效的管理。通过提供应用程序接口（API）函数，从而使应用软件的设计人员得以在一个友好的平台上进行应用软件的设计和开发，大大地提高了应用软件的开发效率。

2. OS 作为计算机系统资源的管理者

一个计算机系统就是一组资源，这些资源用于对数据的移动、存储、处理，以及对这些功能的控制，而 OS 负责管理这些资源。OS 对计算机资源的管理有以下几个方面。

- （1）处理机管理——用于分配和控制处理机。
- （2）存储器管理——主要负责内存的分配与回收。
- （3）I/O 设备管理——负责 I/O 设备的分配与操纵。
- （4）文件管理——负责文件的存取、共享和保护。

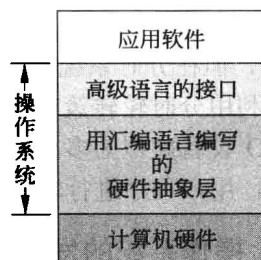


图 1-1 OS 作为接口的示意图

1.1.2 操作系统的特征

操作系统的种类很多,不同的操作系统分别具有各自的特征,一般来说,采用了多道程序设计技术的操作系统具有如下4个基本特征。

1. 并发

在处理机系统中,并发是指宏观上有多道程序同时运行,但在微观上是交替执行的。多道程序并发执行能提高资源利用率和系统吞吐量。

多个进程的并发执行由操作系统统一控制,为保证并发进程的顺利运行,操作系统提供了一系列管理机制。

2. 共享

共享是指计算机系统资源可被多个并发执行的用户程序或系统程序共同使用,而不是被其中某一个程序所独占。共享的原因如下。

(1) 用户或任务独占系统资源将导致资源浪费。

(2) 多个任务共享一个程序的同一副本,而不是分别向每个用户提供一个副本,可以避免重复开发。

并发和共享是紧密相关的。一方面,资源共享是以进程的并发执行为条件的,若不允许进程的并发执行,就不会有资源的共享;另一方面,进程的并发执行以资源共享为条件,若系统不运行共享资源,程序就无法并发执行。

3. 异步

在多道程序系统中,多进程并发执行,但在微观上,进程是交替执行的,因此进程以“走走停停”的不连续方式运行。由于并发运行环境的复杂性,每个进程在何时开始执行,何时暂停,以怎样的速度向前推进,多长时间完成,何时发生中断,都是不可预知的,此种特征称为异步。

4. 虚拟

虚拟指的是通过某种技术把一个物理实体映射为多个逻辑实体,用户程序使用逻辑实体。逻辑实体是让用户感觉上有但实际上不存在的事物,例如在分时系统中,虽然只有一个CPU,但在分时系统的管理下,每个终端用户都认为自己独占一台主机。此时,分时操作系统利用分时轮转策略把一台物理上的CPU虚拟为多台逻辑上的CPU,也可以把一台物理I/O设备虚拟为多台逻辑上的I/O设备,方法是用内存中的输入输出缓冲区来虚拟物理设备,用户程序进行输入输出时,其实是在和缓冲区进行输入输出。

1.1.3 操作系统的发展

操作系统最早产生于1955年,至今已发展了50多年,其发展历程可粗略地划分为4代。

第一代操作系统是单任务自动批处理操作系统,通过作业控制语言使多个程序可自动在计算机上连续运行,在上一个程序结束与下一个程序开始之间不需人工装卸和干预,第一代操作系统通过避免手工装卸而大大提高了机器利用率,但程序执行过程中输入输出数据时,主机空闲降低了处理机利用率。

第二代操作系统是多任务和多用户操作系统,最大特征是采用并发技术,使得当一个

程序在进行 I/O 操作时, CPU 可转去执行其他程序, 从而使多个程序并发执行, CPU 和 I/O 并行工作。第二代操作系统通过并发技术大大提高了机器利用率, 但并发技术的实现代价是使操作系统的复杂程度和功能规模大大增加, 从而增加了操作系统的开发周期和开发成本, 并影响了操作系统的正确性和可靠性。

第三代操作系统是结构化与小型化, 其典型特征是重视操作系统的结构和功能精简。第三代操作系统还具有网络特征。

第四代操作系统是网络和开放系统、并行与分布操作系统。

总之, 操作系统经过几十年的发展, 就单机环境下的系统而言, 其基本原理和设计方法已趋成熟。出现了许多流行的嵌入式系统, 如 UNIX, Windows NT 等。20 世纪 80 年代后, 随着通用微处理器芯片的高速发展, 个人计算机和 workstation 系统得到了迅猛的发展, 强烈冲击着传统小型计算机、中大型计算机的市场。相应地, 微型计算机及 workstation 的操作系统获得了快速的发展和应用, 如 MS-DOS、Windows、Solaris 等。从操作系统的发展历史看, 推动其发展的动力主要是计算机系统的不断完善和计算机应用的不断深入。

随着计算机应用技术的发展, 适应不同应用系统的操作系统也相继出现, 并在应用中得以不断发展。

- 嵌入式操作系统: 主要伴随着个人数字助理 PDA、掌上电脑、电视机顶盒、智能家电等设备的发展, 对操作系统在功能和所占存储空间大小的权衡上提出了新的要求, 对实时响应也有较高的要求。
- 实时操作系统: 对操作系统的实时响应要求从来就没有停止过, 要求计算机的最大响应时间也越来越短, 任务调度时机、算法要求越来越高。特别是针对操作系统的实时性研究还在不断发展中。
- 并行操作系统: 随着高性能通用微处理器的发展, 人们已经成功地提出了用它们构造“多处理机并行”的体系结构。
- 网络操作系统和分布式操作系统: 就目前情形而言, 计算机网络系统也还在不断完善中, 基于 Client-Server 模型的分布式系统也已不断走向应用, 完全分布式的系统还未成形, 仍将是研究的热点问题。

1.2 操作系统的体系结构

操作系统的体系结构设计是指选择合适的结构, 按照这一结构可以对操作系统进行分层、分模块或资源等方式的功能划分; 通过逐步的分解、抽象和综合, 使操作系统功能完备、结构清晰。常用的操作系统体系结构有: 层次结构的操作系统和微内核结构的操作系统。

1.2.1 层次结构

层次结构的操作系统的设计思想是, 按照操作系统各模块的功能和相互依存关系, 把系统中的模块分为若干层次, 每一层(除底层模块)都建立在它下面一层的基础上, 每一层仅使用其下层模块提供的服务, 将系统问题分解成多个子问题, 然后分别解决。操作系统的层次结构如图 1-2 所示。

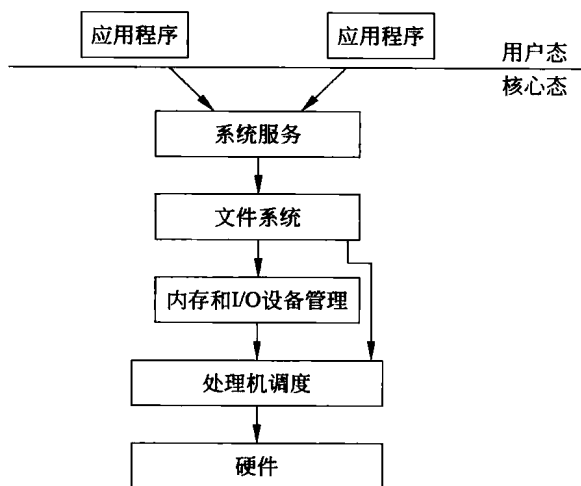


图 1-2 操作系统层次结构的模型

一个操作系统应划分为多少层、各层处于什么位置是层次结构操作系统设计的关键问题，没有固定的模式。一般原则是，接近用户应用的模块在上层，接近硬件的驱动模块在下层。处于下层的程序模块往往称为操作系统的内核。这部分程序模块包括中断处理模块、各种驱动模块和运行频率较高的模块（如时钟管理程序、进程调度程序、低级通信模块、内存管理模块等）。为提高操作系统的执行效率，操作系统内核一般常驻内存。

层次型结构技术在操作系统设计中的应用比较成功，许多典型的商用操作系统都是基于这种结构实现的，如 MS-DOS、早期的 Windows 及传统的 UNIX 等。

1.2.2 微内核结构

近年来备受关注的概念是微内核。微内核是一个小型的操作系统核心，它为模块化扩展提供了基础。

微内核的设计思想是，为了有效地提高系统的可靠性，并使系统具备良好的可适应性，需要将核心模块设计精练化。具体做法是，将原来内核模块中的内容精简，把一些原来属于核心模块的功能提出到外部去完成，只将操作系统最主要的功能包含在内核中，使内核尽量简单，形成微内核的结构。

微内核结构用一个水平分层的结构代替了传统的纵向分层的结构，在微内核外部的操作系统部件被当做服务器进程实现，它们可以通过微内核传递消息来实现相互之间的交互。因此，微内核起着信息交互、验证信息、在部件间传递信息并授权访问硬件的作用。微内核还执行保护功能，除非允许交换，否则它阻止信息传递。微内核结构的模型如图 1-3 所示。

- 运行在核心态的微内核，提供所有操作系统都具有的基本操作，如线程调度、虚拟存储、消息传递、设备驱动、原语操作、中断处理等，这些部分通常采用层次结构并构成了基本操作系统。
- 运行在用户态，并以客户-服务器方式运行的进程层，除内核部分外，操作系统的其他部分都被分成若干个相对独立的进程，每个进程实现一组服务，称为服务进程。这些服务进程可以提供各种系统功能，如文件系统服务以及网络服务等。服务进程的任务是检查是否有客户提出要求服务的请求，并在满足客户进程的请求后将结果返回。

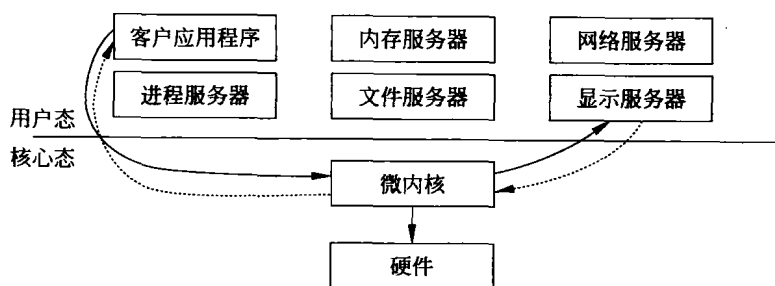


图 1-3 微内核结构的模型

- 客户进程与服务进程之间通信时采用消息通信，客户进程发出消息，内核将消息传给服务进程，服务进程执行相应的操作，其结果又通过内核用发消息的方式返回给客户进程。

基于微内核的操作系统具有如下特征。

微内核提供一组“最基本”的服务，如进程调度、进程间通信、存储管理、处理 I/O 设备以及其他服务，如文件管理、网络支持等，通过接口连到微内核。与此相反，内核是集成的，比微内核更大。

微内核具有很好的扩展性，并可简化应用程序开发。用户只运行他们需要的服务，这有利于减少磁盘空间和存储器需求。

厂商可以很容易地将微内核移植到其他处理器平台，并在上面增加适合其他平台需要的模块化部件，例如文件服务器、工程应用。

微内核和硬件部件有接口，并向可安装模块提供一个接口。在微内核中，进程通过传递消息或运行“线程”来发生相互作用。线程为将一个任务分解为多个子任务提供了途径，在多处理器环境下，线程可以在不同的处理器上独立运行。

1.3 嵌入式操作系统概述

嵌入式操作系统又称实时操作系统，是一种支持嵌入式系统应用的操作系统软件，它是嵌入式系统（包括硬件、软件系统）极为重要的组成部分，通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器 Browser 等。嵌入式操作系统具有通用操作系统的特点，如能够有效管理越来越复杂的系统资源；能够把底层虚拟化，使得开发人员从繁忙的驱动程序移植和维护中解脱出来；能够提供库函数、驱动程序、工具集以及应用程序。嵌入式操作系统能够负责嵌入式系统的全部硬件、软件资源的分配、调度、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达到系统所要求的功能。

1.3.1 嵌入式操作系统的演变

近 10 年来，嵌入式操作系统得到飞速的发展，从支持 8 位微处理器到 16 位、32 位甚至 64 位微处理器；从支持单一品种的微处理器芯片到支持多品种微处理器芯片；从只有内

核到除了内核外还提供其他功能模块，如文件系统，TCP/IP 网络系统，窗口图形系统等。

随着嵌入式系统应用领域的扩展，目前嵌入式操作系统的市场在不断细分，出现了针对不同领域的产品，这些产品按领域的要求和标准提供特定的功能。

嵌入式系统经历了 30 多年的发展，尤其是近几年来，计算机、通信、消费电子的一体化趋势日益明显，嵌入式技术已成为一个研究热点，嵌入式系统也给众多商家带来了良好商机。

目前，嵌入式技术与 Internet 技术的结合正在推动着嵌入式技术的飞速发展，嵌入式系统的研究和应用产生了如下新的显著变化。

- 新的微处理器层出不穷，嵌入式操作系统自身结构的设计更加便于移植，能够在短时间内支持更多的微处理器。
- 嵌入式系统的开发成了一项系统工程，开发厂商不仅要提供嵌入式操作系统本身，同时还要提供强大的软件开发支持包。
- 通用计算机上使用的新技术、新观念开始逐步移植到嵌入式系统中，如嵌入式数据库、移动代理、实时 CORBA、Java 等，嵌入式软件平台得到进一步完善。
- 各类嵌入式 Linux 操作系统迅速发展，由于具有源代码开放、系统内核小、执行效率高、网络结构完整等特点，很符合信息家电等嵌入式系统的需要，目前已经形成了能与 Windows CE、Symbian 等嵌入式操作系统进行有力竞争的局面。
- 网络化、信息化的要求随着 Internet 技术的成熟和带宽的提高而日益突出，以往功能单一的设备如电话、手机、冰箱、微波炉等功能不再单一，结构变得更加复杂，网络互联成为必然趋势。
- 精简系统内核，优化关键算法，降低功耗和软硬件成本。
- 提供更加友好的多媒体人机交互界面。

1.3.2 嵌入式操作系统的特点

一个典型的嵌入式操作系统应该具备下列特点。

1. 可裁剪性

可裁剪性是嵌入式操作系统最大的特点，因为嵌入式操作系统的目标硬件配置差别很大，有的硬件配置非常高，有的却因为成本原因，硬件配置十分紧凑，所以，嵌入式操作系统必须能够适应不同的硬件配置环境，具备较好的可裁剪性。在一些配置高、功能要求多的情况下，嵌入式操作系统可以通过加载更多的模块来满足这种需求；而在一些配置相对较低、功能单一的情况下，嵌入式操作系统必须能够通过裁剪的方式，把一些不相关的模块裁剪掉，只保留相关的功能模块。为了实现可裁剪，在编写嵌入式操作系统的时候，就需要充分考虑、仔细规划，对整个操作系统的功能进行细致的划分，每个功能模块尽量以独立模块的形式来实现。

可通过两种方式来具体实现可裁剪。一种方式是把整个操作系统功能分割成不同的功能模块，进行独立编译，形成独立的二进制可加载映像，这样就可以根据应用系统的需要，通过加载或卸载不同的模块来实现裁剪。另外一种方式，是通过宏定义开关的方式来实现裁剪，针对每个功能模块，定义一个编译开关（`#define`）来进行标志。若应用系统需要该模块，则在编译的时候，定义该标志，否则取消该标志，这样就可以选择需要的操作系统