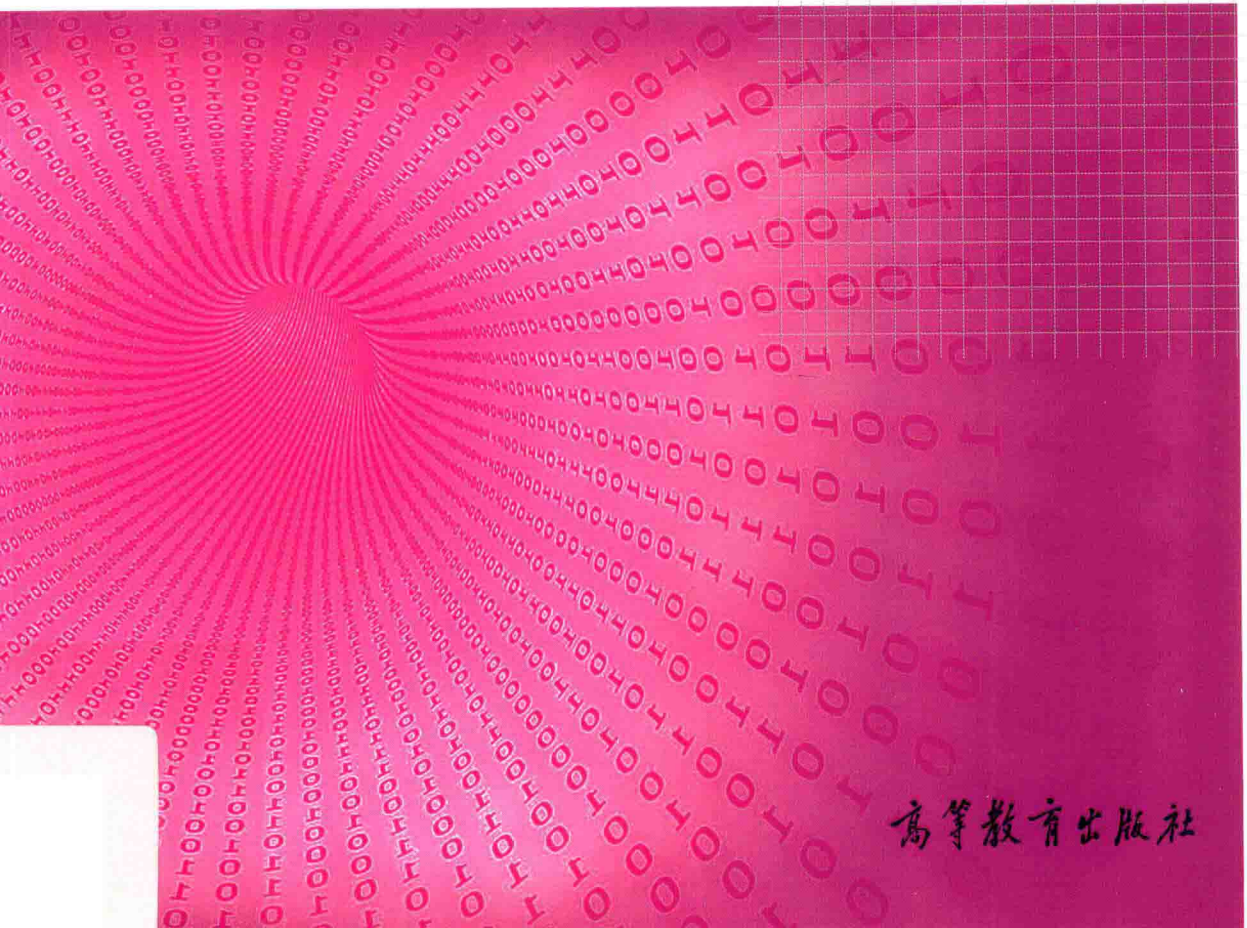
嵌入式系统课程教学实施方案项目规划教材
高等学校计算机专业特色教材

嵌入式操作系统

Embedded Operating System

廖勇　杨霞　主编



高等教育出版社

嵌入式系统课程教学实施方案项目规划教材
高等学校计算机专业特色教材

嵌入式操作系统

Qianrushi Caozuo Xitong

廖 勇 杨 霞 主编

高等教育出版社·北京

内容提要

本书主要介绍嵌入式操作系统内核的工作原理和设计思路。以开源嵌入式操作系统 μC/OS II 为例，深入剖析其在嵌入式平台 ARM9 Mini2440（处理器为三星 S3C2440A）上的实现，重点描述内核基本调度机制、调度策略、任务协调机制（通信、同步、互斥等）、事务处理机制（中断、时钟等）、内存管理机制的设计和实现，并对嵌入式操作系统在不同嵌入式硬件平台上的移植等进行介绍。在此基础上，介绍实时调度理论、可信保障理论以及基于嵌入式操作系统的应用程序开发方法等内容。

本书可作为计算机及相关专业嵌入式系统方向高年级本科生的必修课教材，也可作为相关专业的选修课教材，同时可作为对嵌入式操作系统有浓厚兴趣的读者的参考资料。

图书在版编目（CIP）数据

嵌入式操作系统 / 廖勇，杨霞主编. —北京：高等教育出版社，2017.1

ISBN 978-7-04-046607-2

Ⅰ. ①嵌… Ⅱ. ①廖… ②杨… Ⅲ. ①实时操作系统－高等学校－教材 Ⅳ. ①TP316.2

中国版本图书馆 CIP 数据核字（2016）第 251616 号

策划编辑 时 阳	责任编辑 时 阳	封面设计 杨立新	版式设计 马敬茹
插图绘制 杜晓丹	责任校对 胡美萍	责任印制 朱学忠	

出版发行	高等教育出版社	咨询电话	400-810-0598
社 址	北京市西城区德外大街 4 号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	高教社（天津）印务有限公司	网上订购	http://www.hepmall.com.cn
			http://www.hepmall.com
开 本	787mm × 1092mm 1/16		http://www.hepmall.cn
印 张	20.5	版 次	2017 年 1 月第 1 版
字 数	460 千字	印 次	2017 年 1 月第 1 次印刷
购书热线	010-58581118	定 价	38.20 元

物 料 号 46607-00

出 版 说 明

为推动嵌入式系统专业方向建设和课堂教学，建设嵌入式系统优质教学资源，解决嵌入式系统课程建设和教学中存在的问题，教育部高等学校计算机类专业教学指导委员会、中国计算机学会教育专业委员会和高等教育出版社组织来自电子科技大学、浙江大学、国防科学技术大学、南京航空航天大学、上海交通大学、杭州电子科技大学、深圳大学、电子科技大学中山学院等在嵌入式系统专业方向建设和相关课程建设方面具有一定基础的高校的专家，开展了“嵌入式系统课程教学实施方案”项目研究。

项目研究工作自 2010—2012 年，历时两年，正式出版了《高等学校嵌入式专业方向核心课程教学实施方案》。通过将嵌入式系统人才培养目标，特别是能力培养目标分解、细化到不同课程，同时考虑到课程之间的前后衔接、课程与培养目标实现的支撑，《高等学校嵌入式专业方向核心课程教学实施方案》确定了该专业方向的 5 门核心课程：嵌入式系统概论、嵌入式微控制器及其应用、嵌入式操作系统、嵌入式硬件设计和嵌入式软件开发，并编写相应教材。这套教材旨在反映项目最新的研究成果，体现教学实施方案对于相应课程的定位和要求，以循序渐进的方式安排教学内容，讲解嵌入式系统平台设计的基本原理和基本方法，并给出典型性的示例。为保证教材质量，项目组确定由具有相应课程丰富教学和开发经验，并参与了课程教学实施方案研究的专家担任主编，同时邀请相关领域的权威专家对书稿进行了认真审读，数次修改，才最终交付出版。

这套教材具有如下特点：

- 内容全面，结构新颖。在内容上，理论与实践并重，涵盖课程教学实施方案中涉及的重点内容，并提供大量实例；在结构安排上，从基础知识入手，循序渐进；在叙述上，力求简明扼要，由浅入深。
- 原理部分通用性好。将教材的理论部分与实验部分分开，尽量减少理论内容对实验平台的依赖，以方便实际教学。
- 实验内容丰富、翔实。针对理论部分设计相应的实验，这些实验均系作者亲自设计、验证，力求展现细节，让读者易于上手，可以边干边学，具有较大的参考价值。
- 在拓展性上有突破。教材每章后都有相应的实验思考题和进一步探索的题目，供读者进一步深入研究。
- 面向实践，适用面广。教材兼顾教学、科研和工程开发的需要，对于广大高校本科生和研究生而言，是学习嵌入式系统的教科书；对于从事嵌入式系统开发的工程人员来说，则是实用的参考书。

我们希望这套教材的出版能够为国内高校嵌入式系统相关课程教学的开展提供有益的参考和帮助，提升高校嵌入式系统的教学水平和开发水平，培养更多适应社会需求的嵌入式技术人才。我们将为此不懈努力，也希望得到各位读者的热情帮助，使这套教材能够不断完善和提高。

教育部高等学校计算机类专业教学指导委员会
中国计算机学会教育专业委员会
高等教育出版社

前　言

一、编写意图

在后 PC 时代的今天，嵌入式操作系统琳琅满目，在各行各业已有广泛应用，例如 VxWorks、QNX、RTEMS、Cisco-IOS、ERISON-EPOC、μC/OS、aCoral、INTEGRITY、pSOS+、VRTX、TinyOS、DeltaOS、T-kernel、Windows CE、Windows Mobile、Symbian、Android、iOS、嵌入式 Linux 等。并且，嵌入式操作系统的相关技术日趋成熟，一些大学的本科生编写一个简易的操作系统内核也不是什么难事。此外，市面上也已出版了关于嵌入式操作系统编写和代码实现的书籍，如 1999 年出版的《MicroC/OS-II: The Real-Time Kernel》、2005 年出版的《自己动手写操作系统完全版》、2011 年出版的《一步步写嵌入式操作系统: ARM 编程的方法与实践》等。为什么还要编写一本关于嵌入式操作系统设计的书呢？其目的是从本科教学特点出发，根据计算机及相关专业嵌入式方向系列教材建设的需要，将课程知识体系与操作系统设计实现过程有机结合，让学生更好地理解课程知识点；通过嵌入式操作系统在某一嵌入式平台上的设计与实现，融会贯通相关核心知识点，进而激发学生自己动手设计操作系统的热情；最终目的是以此锻炼本科学生的系统设计能力、工程实践能力、分析与解决问题能力，充分体现教育家约翰·杜威先生“To learn by doing”的教学理念。

二、本书特点

本书以嵌入式操作系统在一款单核 ARM9 Mini2440 嵌入式平台(处理器为三星 S3C2440A)上的设计过程为主线，逐步介绍开源操作系统 μC/OS II 的实现，并以此为基础，阐述嵌入式操作系统的相关技术和理论，如实时调度机制、实时调度策略、可信保障技术等，使学生对嵌入式操作系统有更全面的认识。此外，全书综合应用了嵌入式系统概论、嵌入式控制器及其应用、计算机组成原理、汇编语言、C 语言程序设计、计算机操作系统、数据结构等课程的知识点，力求理论与实践紧密结合，帮助学生融会贯通上述课程的相关知识。

三、编写方法

本书的编写遵循教育部高等学校计算机类专业教学指导委员会制定的《高等学校嵌入式系统专业方向核心课程教学实施方案》，是嵌入式专业方向核心课程系列教材之一。教材以 μC/OS II 的设计过程为线索，一步一步介绍其实现细节。例如，从任务定义开始，逐步启发读者如何用 C 语言描述一个任务，如何创建任务，如何将任务挂载到就绪队列中，如何从就绪队列中找到最高优先级任务，如何在 ARM9 Mini2440 上实现任务切换，如何触发任务调度，如何协调任务执行，如何启动内核，如何移植内核，如何基于嵌入式操作系统开发应用程序，等等。全书将代码实现与基本原理紧密结合，从工程实践中发现问题，在理论上对其进行详细分析，并最

终指导操作系统设计与实现。此外，本书对编写嵌入式操作系统过程中涉及的相关知识进行了整合。因此，为确保阅读效果，读者最好提前掌握操作系统基本原理、ARM9 处理器的基础知识，并且对 ARM 汇编、ADS 编译器交叉开发环境的使用有足够了解。这些内容不属于本书介绍的范围，但却是深入理解本书的重要基础。

四、编者分工

廖勇编写了前言、第 1 章概述、第 2 章任务调度机制、第 3 章任务调度策略、第 5 章中断和时间管理；杨霞编写了第 4 章任务的同步与通信，第 6 章内存管理、I/O 管理和文件系统，第 7 章嵌入式操作系统移植，第 8 章嵌入式软件开发，第 9 章高可信的嵌入式操作系统。全书由廖勇统稿。

五、资料来源

贯穿本书的嵌入式操作系统源代码来自于 Jean J. Labrosse 先生设计的 μC/OS II。本书在编写过程中，参考了三星公司 ARM9 S3C2440 芯片手册、开源嵌入式多核实时操作系统 aCoral、Linux 开源社区、高可信航空电子应用软件标准接口 ARINC 653、实时系统相关的学术论文等资料。此外，还借鉴了邵贝贝教授翻译的 Jean J. Labrosse 先生的《MicroC/OS-II: The Real-Time Kernel》、C. M. Krishna 和 KANG G. Shin 的《Real-Time Computing》、罗蕾教授的《嵌入式实时操作系统及应用开发》（第 4 版）、桑楠教授的《嵌入式系统原理及应用开发技术》（第 2 版）、李无言老师的《一步步写嵌入式操作系统：ARM 编程的方法与实践》、于渊老师的《自己动手写操作系统完全版》等书籍。

六、致谢

本书在编写过程中得到了“嵌入式系统课程教学实施方案”研究项目和电子科技大学教务处特色教材建设项目的支持，作者对此深表感谢。

本书还得到了电子科技大学副校长马争教授，电子科技大学信息与软件学院雷航教授、蔡竟业教授、桑楠教授，电子科技大学计算机学院熊光泽教授、罗克露教授、罗蕾教授、周世杰教授，浙江大学陈文智教授、施青松教授，南京航空航天大学马维华教授，上海交通大学方向忠教授，国防科学技术大学陆洪毅老师，深圳大学薛丽萍老师，电子科技大学中山学院李文生老师，高等教育出版社张龙、武林晓和时阳编辑的帮助，他们为本书的编写提供了宝贵的意见和建议。

感谢电子科技大学实时计算实验室的博士研究生杨茂林、葛旭阳、李佳阳、李恒瑞、刘黎民、张晋川、陈泽玮、罗超、Furkan Hassan Saleh Rabee 以及美国宾夕法尼亚大学研究生廖聪、南加州大学研究生李龙杰、卡内基·梅隆大学在读研究生席臣、南加州大学在读研究生高枫、法国图卢兹大学在读博士生苏铅坤、国防科学技术大学研究生龚俊如、上海交通大学在读研究生罗赟、复旦大学研究生胡伟松、中国人民大学研究生范旭、清华大学研究生陆文详和沈游人、中国工程物理研究院研究生张朝，在电子科技大学实时计算实验室学习的本科生兰宇航、陈维伟、胡斌，去哪儿网软件研发工程师蒋世勇、华为成都研究所软件工程师徐新、华为深圳研究所软件工程师陈朱旭、百度软件研发工程师李天华、微软苏州研究院林添、美团网研发工程师

许斌、美国云控公司成都研发中心谢鑫、美国豪威公司上海研发中心周强、中国民航成都研究院王小溪等。他们在作者的引导下，从本科二年级开始学习嵌入式实时系统，并在本科期间围绕 ARM9 Mini2440 + μC/OS II + LWIP + μGUI + aCoral + Linux 等平台进行系列实验、课程设计、毕业设计，还在此平台上实现了智能家居及安防系统、智慧交通流量信息采集系统、学习型遥控器、网络收音机、智能小车防撞系统、四轴飞行器等本科创新/实践项目，他们为 μC/OS II 的代码分析及验证做了许多细节工作。

感谢在上海创业的申建晶、在成都创业的刘坚、朱葛、郑亚斌，感谢完美世界的高攀、中国建设银行四川分行闫志强、美国美满电子（Marvell）上海研究所的陈勇明及成都研究所的魏守峰、程潜、任艳伟，感谢淘宝杭州研发中心的张林江等曾在电子科技大学实时计算实验室学习过的同志们，他们在校期间陆续加入开源项目组，与作者共同探讨嵌入式操作系统的相关技术、教学改革、本科生学习及就业等问题，这为本书结构的拟定、撰写方法等提供了思路和素材。

特别感谢西南交通大学土木工程学院刘萍老师，她在本书编写过程中给予了极大鼓励和支持；同时，特别感谢摩托罗拉成都研究所肖炜先生，他对本书的编写给予了诸多建议。

由于作品水平有限，书中难免存在疏漏之处，恳请广大读者批评指正。

作　者

2016 年 9 月

目　录

第 1 章　嵌入式操作系统概述

1.1　什么是嵌入式操作系统

20 世纪末，随着信息技术与网络技术的迅猛发展，计算机技术已经进入后 PC（Post-PC）时代。该时代的计算机更加多样化，它们遍布在人们周围，功能强大，向人们提供各种便捷的服务，而人们似乎又感觉不到它们的存在，这就是无处不在的计算（pervasive computing 或 ubiquitous computing）[4,5]。无处不在的计算模式依赖于通用计算机和嵌入式计算机，通用计算机只占大约 5%的比例，而嵌入式计算机占大约 95%的比例。

嵌入式计算机通常被称为嵌入式系统（embedded system）。嵌入式系统是以应用为中心，以计算机技术为基础，软硬件可配置，对功能、可靠性、成本、体积、功耗有严格约束的专用计算机系统。纵观嵌入式系统的发展过程，其出现至今已经有 50 多年的历史，大致经历了以下五个阶段。

第一阶段大致在 20 世纪 70 年代之前，可看作嵌入式系统的萌芽阶段。这一阶段的嵌入式系统是以单芯片为核心的可编程控制器形式的系统，具有与监测、伺服、指示设备相配合的功能。这类系统大部分应用于一些专业性较强的工业控制系统中，一般没有操作系统的支持，通过汇编语言编程对系统进行直接控制。这一阶段系统的主要特点是：系统结构和功能相对单一，处理效率较低，存储容量较小，只有很少的用户接口。由于这种嵌入式系统使用简单、价格低廉，即使现在依然在简单、低成本的嵌入式应用领域大量使用，但已经远不能适应高效、需要大容量存储的现代工业控制和新兴信息家电等领域的需求。

第二阶段为第一阶段之后的十多年。这一阶段的嵌入式系统以嵌入式处理器为基础、以简单操作系统为核心。在此阶段，大多数嵌入式系统使用 8 位处理器，不需要嵌入式操作系统的支持。其主要特点是：处理器种类繁多，通用性较弱；系统开销小，效率高；高端应用所需操作系统已具备一定的实时性、兼容性和扩展性；应用软件较专业化，用户界面不够友好。

第三阶段大致是 20 世纪 80 年代末到 20 世纪 90 年代后期，是嵌入式应用开始普及的阶段。这一阶段的嵌入式系统以嵌入式操作系统为标志。其主要特点是：嵌入式操作系统内核小、效率高，具有高度的模块化和扩展性；能运行于各种不同类型的微处理器上，兼容性好；具备文件和目录管理、多任务、网络支持、图形窗口以及用户界面等功能；提供大量的应用程序接口（Application Programming Interface，API）和集成开发环境，简化了应用程序开发；嵌入式应用

软件丰富。在此阶段，嵌入式系统的软硬件技术加速发展，应用领域不断扩大。例如，日常生活中使用的手机、数码相机，网络设备中的路由器、交换机等，都是嵌入式系统；一辆豪华汽车中有数十个嵌入式处理器，分别控制发动机、传动装置、安全装置等；一个飞行器上可以有数百个甚至上千个嵌入式微处理器。

第四阶段从20世纪90年代末开始，这一阶段的嵌入式系统以网络化和因特网（Internet）为标志。随着Internet的发展以及Internet技术与信息家电、工业控制、航空航天等技术的结合日益密切，嵌入式设备与Internet的结合代表了嵌入式系统的未来。1998年11月在美国加州圣·何塞举行的嵌入式系统大会上，基于嵌入式实时操作系统的Internet成为一个新的技术热点。

最后一个阶段是从21世纪初到现在，这一阶段的嵌入式系统以物联网、云计算和智能化为标志，也是多核芯片技术、无线技术、互联网发展与信息家电、工业控制、航空航天等技术相结合的必然结果。从应用角度而言，移动互联网设备是嵌入式产品的热点。目前，具备网络互联功能的智能终端出货量已达到4亿部，比同时期笔记本式计算机和台式计算机出货量的总和还要多。无处不在的嵌入式系统，如智能手机、无线传感器网络（Wireless Sensor Network，WSN）、RFID电子标签等遍布在人们周围，为人们提供方便快捷的服务。

由此可见，嵌入式操作系统是随着嵌入式计算机的发展而发展的。嵌入式系统软件的日益复杂，在客观上使得软件的编写需要多人分工合作完成。从嵌入式软件体系结构的角度考虑，就是将一个软件功能划分成多个任务，采用实时多任务体系实现。因此，功能强大的嵌入式操作系统成为支撑其运行的基础。由于嵌入式操作系统及其应用软件往往被嵌入特定的控制设备或者仪器中，用于实时响应并处理外部事件，所以嵌入式操作系统有时也称为实时操作系统（Real-Time Operating System，RTOS）；另一方面，由于RTOS也往往存在于嵌入式系统中，因此本书约定：为描述方便，下文提到的RTOS就代表实时操作系统或者嵌入式操作系统。

RTOS可以简单认为是功能强大的主控程序，它嵌入在目标代码中，系统复位后首先执行；它负责在硬件基础上为应用软件建立一个功能强大的运行环境，用户的应用程序都建立在RTOS之上。在这个意义上，RTOS的作用是为用户提供一台等价的扩展计算机，它比底层硬件更容易编程操作。

RTOS内含一个实时内核，完成最基本却又必不可少的功能，如CPU、中断、时钟、I/O等资源的管理，为用户提供一套标准编程接口；并可根据各个任务的优先级，合理地在不同任务间分配CPU资源。在此意义上，RTOS的作用相当于系统资源管理器。

对嵌入式系统而言，RTOS的引入会带来很多好处。首先，一个RTOS就是一套标准化的任务管理机制，可以提升开发单位的管理水平和开发人员的业务素质；其次，每个RTOS都提供一套较完整的应用编程接口，可以大大简化应用编程，提高系统的可靠性；第三，RTOS的引入客观上导致应用软件与下层硬件环境无关，便于嵌入式软件系统的移植；最后，基于RTOS可以直接使用许多的应用编程中间件，既可增强嵌入式软件的复用能力，又可降低开发成本，缩短开发周期。

据统计，到目前为止，世界各国的数十家公司已成功推出 200 多种可供嵌入式应用的 RTOS，其中包括 WindRiver System 公司的 VxWorks、pSOS+，Mentor Graphics 公司的 VRTX，微软公司的 Windows CE、Windows Mobile，Symbian 公司的 Symbian OS，Enea 公司的 OSE，Microware 公司的 OS-9，苹果公司的 iOS，3Com 公司的 Palm OS，国产的 DeltaOS、Hopen，以及多种多样的嵌入式 Linux 等。

1.2 嵌入式操作系统的特点

相对于通用操作系统(如 Windows、PC 版 Linux 等)而言，RTOS 往往具有以下共同特点[1-5]。

1. 实时性

实时性(timeliness)是嵌入式实时系统最基本的特点，也是 RTOS 必须保证的特性。RTOS 的主要任务是对外部事件做出实时响应。虽然事件可能在无法预知的时刻到达，但是软件必须在事件发生时能够在严格的时限（称为“系统响应时间”，response time）内做出响应，即使在峰值负载下也应如此。系统响应时间超时可能就意味着致命的失败。

由于不同的实时系统对实时性的要求有所不同，实时性可以分为以下两类。

- 硬实时（hard real-time）：系统对外部事件的响应略有延迟就会造成灾难性的后果，也就是说，系统响应时间必须严格小于规定的截止时间（deadline）。
- 软实时（soft real-time）：系统对外部事件响应超时可能会导致系统产生一些错误，但不会造成灾难性后果，且大多数情况下不会影响系统的正常工作。

对于 RTOS 而言，实时性主要由实时多任务内核的任务调度机制和调度策略共同确保。不同的 RTOS 所提供的策略有所不同，有些支持硬实时性，有些只支持软实时性，但主流 RTOS 需要支持多种实时性。

2. 可确定性

RTOS 的一个重要特点是具有可确定性（deterministic），即系统在运行过程中，系统调用的时间可以预测。虽然系统调用的执行时间不是一个固定值，但是其最大执行时间可以确定，从而能对系统运行的最好情况和最坏情况做出精确的估计。

衡量操作系统确定性的一个重要指标是截止时间，它规定系统对外部事件的响应必须在给定时刻内完成。截止时间的长短随应用的不同而不同，可以从纳秒（ns）级、微秒（μs）级直到分钟（min）级、小时级、天级。

在实时系统中，外部事件随机到达。但在规定的时序范围内，有多少外部事件可以到达却必须是可预测（可控）的。这是 RTOS 可确定性的第二种体现。

可确定性的第三种体现是对系统资源占用的确定化。对大多数嵌入式系统，特别是硬实时系统而言，在系统开始运行前，每个任务需要哪些资源、哪种情况下（何时）占用资源都应是可预测的。在极端情况下，资源占用必须用静态资源分配表一一列出。

3. 并发性

并发性（concurrence）有时也称为同时性（simultaneousness）。在复杂的实时系统中，外部事件的到达是随机的，因此某一时刻可能有多个外部事件到达，RTOS 需要同时激活多个任务（task）处理对应的外部请求。通常，实时系统采用多任务机制或者多处理机结构来解决并发性问题，而 RTOS 则用于相应的管理。

4. 高可信性

不管外部条件如何恶劣，实时系统都必须能够在任意时刻、任意地方、任意环境下对外部事件做出准确响应。这就要求 RTOS 比通用操作系统更具可靠性（reliability）、稳健性（robustness）和防危性（safety）。这些特性统称为高可信性（high dependability）。

可靠性是指在一组特定条件下，系统在一定时期内不发生故障的概率。它强调的是系统连续工作的能力，是一个“好”系统的必要指标。

稳健性特别强调容错处理和出错自动恢复，确保系统不会因为软件错误而崩溃甚至出现灾难性后果。即使在最坏情况下，RTOS 也应能够让系统性能平稳降级，最好能自动恢复正常运行状态。

防危性研究系统是否会导致灾难发生，关心的是引起危险的软件故障。在实际应用中，它主要确保系统对外部设备的操作不出现异常，这一点在安全关键系统（如核电控制系统、航空航天系统）中尤为突出。

5. 安全性

信息安全（security）是目前 Internet 上最热门的话题之一，其中很大一部分原因归结于基础网络设备（路由器、交换机等）的安全管理机制，其核心是保密。

RTOS 自然需要从系统软件级就为嵌入式设备提供安全保障措施，关注外部环境对系统的恶意攻击，减少应用开发者的重复劳动。

6. 可嵌入性

RTOS 及其应用软件基本上都需要嵌入具体设备或者仪器中，因此，RTOS 必须具有足够小的体积及很好的可裁剪性和灵活性。这就是可嵌入性（embeddability）的含义。

由于大多数嵌入式设备的资源有限，不大可能像个人计算机一样预装操作系统、设备驱动程序等。因此，最常见的 RTOS 应用原则是：将 RTOS 与上层应用软件捆绑成一个完整的可执行程序，下载到目标系统中；当目标系统启动时，首先引导 RTOS 执行，再控制管理其他应用软件模块。

7. 可剪裁性

嵌入式系统对资源有严格限制，RTOS 就不可能如桌面操作系统（Windows 等）一样装载大量的功能模块，而必须对应用有极强的针对性。因此，RTOS 必须具有可剪裁性（tailorability），即组成 RTOS 的各模块（组件）能根据不同应用的要求合理剪裁，做到够用即可。

8. 可扩展性

当前，嵌入式应用的发展异常迅猛，新型嵌入式设备的功能多种多样，这对 RTOS 提出了

可扩展性（extensibility）的要求。即除提供基本的内核支持外，还须提供越来越多的可扩展功能模块（含用户扩展），如功耗控制、动态加载、嵌入式文件系统、嵌入式图形用户界面（Graphic User Interface，GUI）系统、嵌入式数据库系统等。

1.3 嵌入式操作系统的主要功能

RTOS 的基本功能由内核完成，主要负责任务管理、中断管理、时钟管理、任务协调（通信、同步、资源互斥访问等）、内存管理等，这些管理功能是通过内核系统调用的形式交给用户调用的；其他功能以 RTOS 扩展组件形式实现，包括嵌入式网络、嵌入式文件系统、功耗管理、嵌入式数据库、流媒体支持、用户编程接口、嵌入式 GUI 等。

1. 任务管理

多任务机制是现代操作系统的基础。一个多任务的环境允许将实时应用构造成一套独立的任务集合，每个任务拥有各自的执行进程和系统资源，这些任务共同合作以实现整个系统的功能。

多任务并发执行造成了一种多个任务同时执行的假象。事实上，内核是将某种调度算法加入这些任务的执行中，使每个任务拥有自己的上下文，包括 CPU 执行环境和系统资源。这种内核调度机制是任务运行时所必需的，类似于通用多任务操作系统（如 UNIX、Windows）中的处理，只是增加了实时性的要求。

2. 中断管理

中断（interrupt）是外部事件通知 RTOS 的主要机制。外部事件产生的中断属于硬件机制，它向 CPU 发出中断信号，表示外部异步事件发生。异步事件是指无一定时序关系的随机事件，如外部设备完成数据传输，实时控制设备出现异常情况等。

与应用任务相比，应用任务是由 RTOS 调度的，而中断处理程序一定是异步执行的，不需要 RTOS 调度。当中断被触发时，中断处理程序就开始运行。实时系统必须能够快速响应外部产生的中断，以成功地与外部环境进行交互。实时多任务系统有如下三种方式来处理外部的中断请求。

（1）中断作为任务切换。

（2）中断作为系统调用。

（3）中断作为前台事务。

3. 时钟管理

在实时系统中，实时时钟（clock）是实时软件运行必不可少的硬件设施。实时时钟单纯地提供一个规则的脉冲序列，脉冲之间的间隔可以作为系统的时间基准，称为时基（tick）。时基的大小代表了实时时钟的精度，这个精度取决于系统的要求。

为了计准时间间隔，最重要的问题是确保 CPU 能与时钟同步工作。同步可以用硬件方法