

嵌入式实时操作系统 测试理论和方法

张明杰 雷迅 杨云 等著

航空工业出版社

嵌入式实时操作系统 测试理论和方法

张明杰 雷 迅 杨 云 王 林
宋 恒 李 强 张 军 著



航空工业出版社

内 容 提 要

嵌入式实时操作系统在通信、医疗、核电站控制等领域的广泛应用，使人们越来越关注嵌入式实时操作系统的安全性。目前，系统介绍高安全嵌入式实时操作系统测试理论和方法的文献还比较少，本书介绍了关于高安全嵌入式实时操作系统的测试理论和方法。

本书以理论为指导，提出了操作系统测试需要解决的问题，证明了基于有限状态机测试策略的正确性，使用动态代码走查方法提高代码走查的效率和效果，使用地址监控方法解决临界保护正确性测试问题。每种测试方法都以理论为指导，测试充分性得到有效保证。

本书适合于从事嵌入式操作系统设计和测试的科研人员参考和使用。

图书在版编目 (C I P) 数据

嵌入式实时操作系统测试理论和方法 / 张明杰等著

. -- 北京 : 航空工业出版社, 2014. 7

ISBN 978 - 7 - 5165 - 0499 - 4

I. ①嵌… II. ①张… III. ①实时操作系统—系统测试—研究 IV. ①TP316. 2

中国版本图书馆 CIP 数据核字 (2014) 第 143378 号

嵌入式实时操作系统测试理论和方法
Qianrushishi Shishi Caozuo Xitong Ceshi Lilun he Fangfa

航空工业出版社出版发行

(北京市朝阳区北苑 2 号院 100012)

发行部电话: 010 - 84934379 010 - 84936343

北京地质印刷厂印刷

全国各地新华书店经售

2014 年 7 月第 1 版

2014 年 7 月第 1 次印刷

开本: 787 × 1092 1/16

印张: 15. 5

字数: 410 千字

印数: 1—2000

定价: 48. 00 元

前言

嵌入式操作系统应用越来越广泛，在国民经济、国防军事、医疗设备等各方面发挥着重要作用，是信息化社会的支撑。在某些特定领域，对嵌入式操作系统的可靠性要求很高。在医疗设备中（如 CT 扫描），如果出现问题可能会对患者造成严重的伤害；核电站控制系统出现问题带来更严重的后果；手机中的嵌入式操作系统越来越复杂，虽然系统崩溃不会造成严重后果，重启往往可以解决问题，但是必须承认，可靠性不高也会影响手机的市场地位。也就是说，在很多领域对嵌入式实时操作系统的可靠性要求越来越高，在很多情况下可靠性要求甚至性能要求。

操作系统的可靠性由许多方面构成，但最重要的是其内核的可靠性，内核主要包括进程管理和存储管理等。有的嵌入式实时操作系统也包括简单的文件系统，但不是标准配置，属于可配置组件，因此本书主要关注内核的可靠性。

测试是提高嵌入式实时操作系统内核可靠性的重要手段，但是无论是学术界还是工程界都鲜有关于嵌入式实时操作系统测试理论方面的资料。笔者曾在一些知名的计算机学术期刊上检索以期得到一些启发，结果没有检索到相关学术文献。虽然 LTP（Linux Test Project）是对 Linux 操作系统进行测试的国际化项目，但是对高可靠嵌入式实时操作系统测试而言帮助甚微。笔者认为高可靠性嵌入式实时操作系统测试必须在理论指导下进行，以保证测试的充分性和可信性。

有关测试资料的缺乏在笔者看来可能是属于商业机密，或者是更多的学者更倾向于构建新的操作系统，而没有太多兴趣测试操作系统的可靠性。毕竟构建操作系统是更具创造性的活动。于是我们决定坐下来静静地思考一下嵌入式实时操作系统的测试方法，经过几年的努力有了这本书中描述的测试方法。必须说明其中的测试方法，如有限状态机方法不是笔者创造而是拿来用于指导操作系统测试。本书旨在构建高可靠嵌入式实时操作系统内核测试的框架，以供感兴趣的读者进一步探讨。

测试理论告诉我们，基于需求的测试是最有效的软件测试方法。然而基于需求的测试方法往往在现有国内软件工程情况下效果不佳。操作系统的需求往往以系统提供的系统调用为纲目进行描述，单独对系统调用进行测试，没有考虑系统调用之间相互影响可能导致的系统缺陷，往往不能发现系统深层次问题。操作系统提供大量系统调用，而应用程序使用系统调用的方式千变万化，所以必须采用

分而治之的策略才能实现对操作系统的高效测试。

本书提出了操作系统测试问题；动态的代码走查方法，提高代码走查效率；基于地址监控的临界资源保护正确性测试方法；基于有限状态机的操作系统应用编程接口测试方法。这些方面基本涵盖了嵌入式实时操作系统内核的关键机制。虽然这些方法不一定是最有效的方法，但是从一定意义上讲，测试的充分性得到了保证。

限于笔者能力水平和经验有限，同时由于成书时间仓促，书中难免存在不足之处，希望读者提出中肯的批评和建议。如果能够吸引更多读者开展嵌入式实时操作系统的测试研究将是最令人欣慰的事。

目 录

第 1 部分 嵌入式实时操作系统的基本原理

第 1 章 嵌入式实时操作系统概述	(3)
1.1 操作系统概述	(3)
1.1.1 操作系统作用	(3)
1.1.2 操作系统发展简史	(5)
1.2 嵌入式实时操作系统	(6)
1.2.1 嵌入式实时操作系统的特点	(6)
1.2.2 嵌入式实时操作系统实例	(7)
1.3 本章小结	(8)
第 2 章 RTEMS 嵌入式实时操作系统	(9)
2.1 RTEMS 概述	(9)
2.2 RTEMS 超级内核	(11)
2.2.1 对象	(11)
2.2.2 任务队列	(12)
2.3 RTEMS 功能组件	(13)
2.3.1 任务组件	(13)
2.3.2 中断组件	(14)
2.3.3 时钟组件	(15)
2.3.4 定时器组件	(16)
2.3.5 进程间通信组件	(17)
2.3.6 存储管理组件	(18)
2.4 调度策略	(19)
2.4.1 调度策略控制	(21)
2.4.2 任务状态转换	(21)
2.5 本章小结	(22)

第 2 部分 嵌入式实时操作系统测试的理论和方法

第 3 章 问题和策略	(25)
3.1 软件测试概述	(25)
3.1.1 软件测试过程	(25)

3.1.2	软件测试类型	(26)
3.2	操作系统测试特殊性	(26)
3.2.1	三角形判定问题描述	(27)
3.2.2	三角形判定程序实现	(27)
3.2.3	三角形判定程序测试	(29)
3.2.4	操作系统测试的特殊性	(30)
3.3	操作系统测试问题描述	(30)
3.3.1	Ψ_{prg} 的特点	(30)
3.3.2	Ψ_{prg} 的分类	(31)
3.3.3	问题描述	(31)
3.3.4	基于独立系统调用的测试方法的不足	(31)
3.4	基于资源管理视点的测试策略	(32)
3.4.1	资源的表示方法	(32)
3.4.2	资源表示的本质	(33)
3.4.3	问题重新描述	(33)
3.4.4	基路径测试	(36)
3.5	信号量测试案例	(37)
3.5.1	常规测试法	(38)
3.5.2	基于有限状态机的测试法	(39)
3.5.3	正确性证明	(41)
3.6	其他考虑	(43)
3.7	本章小结	(43)
第 4 章	动态代码走查方法	(44)
4.1	代码走查定义	(44)
4.2	基于软件动态执行的代码走查方法 DCW	(44)
4.2.1	软件移植	(45)
4.2.2	代码走查	(46)
4.3	RTEMS 超级内核代码走查	(47)
4.3.1	Score 移植	(47)
4.3.2	Score 代码走查	(57)
4.4	本章小结	(64)
第 5 章	临界保护正确性测试	(65)
5.1	问题描述	(65)
5.2	基于地址监控的临界保护正确性测试	(66)
5.2.1	RTEMS 临界保护特点	(66)
5.2.2	地址监控的临界区保护正确性测试	(66)
5.2.3	正确性证明	(69)
5.2.4	SVAM 方法优化	(69)
5.2.5	Bochs 实现 SVAM	(70)

5.2.6 实际操作	(73)
5.3 本章小结	(75)
第6章 基于有限状态机的 API 测试	(76)
6.1 方法概述	(76)
6.2 RTEMS 有限状态机	(76)
6.2.1 固定大小存储管理有限状态机	(77)
6.2.2 可变大小存储管理有限状态机	(79)
6.2.3 事件通信有限状态机	(86)
6.2.4 消息队列有限状态机	(88)
6.2.5 信号量有限状态机	(90)
6.2.6 异步信号有限状态机	(95)
6.2.7 BARRIER 通信有限状态机	(97)
6.2.8 任务管理有限状态机	(99)
6.3 状态树	(102)
6.3.1 分区状态树	(102)
6.3.2 堆状态树	(103)
6.3.3 事件状态树	(106)
6.3.4 消息队列状态树	(106)
6.3.5 信号量状态树	(108)
6.3.6 异步信号状态树	(110)
6.3.7 BARRIER 状态树	(111)
6.3.8 任务管理状态树	(112)
6.4 测试用例设计	(113)
6.5 本章小结	(116)
第 3 部分 嵌入式实时操作系统测试方法的具体实现	
第7章 构建测试环境	(119)
7.1 安装 VMware 虚拟机	(119)
7.2 安装 Linux 操作系统	(124)
7.3 安装 RTEMS 交叉编译环境	(130)
7.4 编译 RTEMS 操作系统	(131)
7.5 本章小结	(132)
第8章 测试程序设计	(133)
8.1 测试程序架构	(133)
8.2 RMMT 详细设计	(134)
8.2.1 分区创建测试	(134)
8.2.2 分区删除测试	(136)
8.2.3 获取分区 ID 测试	(137)

8.2.4	分区申请缓冲测试	(139)
8.2.5	分区释放缓冲测试	(140)
8.2.6	状态转换测试	(141)
8.2.7	程序文件结构和编译指令	(143)
8.3	分区测试运行效果	(144)
8.4	本章小结	(150)
第9章	测试结果分析	(151)
9.1	测试结果统计	(151)
9.2	典型问题分析	(156)
9.2.1	自动释放 BARRIER 任务问题	(156)
9.2.2	任务在休眠状态下挂起问题	(156)
9.2.3	双口地址映射错误问题	(157)
9.2.4	堆扩展后及时分配问题	(159)
9.2.5	优先级变化后堆及时分配问题	(160)
9.3	复杂度的抢占式资源队列调度	(161)
9.3.1	问题提出	(161)
9.3.2	抢占式资源调度	(162)
9.3.3	具体实现	(163)
9.4	本章小结	(172)
第10章	有限状态机测试程序源代码	(173)
10.1	堆存储有限状态机测试	(173)
10.2	信号量有限状态机测试	(221)
参考文献	(240)

第 1 部分 嵌入式实时操作系统的基本原理

第 1 部分主要介绍嵌入式实时操作系统设计原理，内容包括嵌入式实时操作系统概述以及作为本书研究对象的 RTEMS 操作系统的实现介绍。如果读者对嵌入式实时操作系统比较熟悉可以跳过这一部分，如果读者对操作系统不甚了解，本部分给出的介绍可以让读者了解操作系统设计的基本原理，对阅读第 2 部分是必需的。

第 1 章 嵌入式实时操作系统概述

操作系统伴随着计算机硬件技术的发展应运而生并不断发展，作为信息社会最重要的基础支撑，操作系统存在于几乎所有的电子设备中，从大型通信系统、科技含量极高的航天设备、医疗器械、核电站控制系统、家用电器和无所不在的移动终端，操作系统在这些系统和设备中发挥着不可替代的作用，没有操作系统的默默支撑，信息技术的飞速发展是不可能实现的。作为操作系统家族重要一员的嵌入式实时操作系统更是无处不在，不知不觉中改变着我们的生活。本章主要介绍操作系统的作用、历史回顾，最后重点介绍本书的研究对象——嵌入式实时操作系统。

1.1 操作系统概述

在现代计算机系统中，无论是大型分布式系统还是功能单一的家电控制系统，操作系统都起着十分重要的作用，在信息化设备中扮演着关键角色。

1.1.1 操作系统作用

操作系统的作用主要体现在如下两个方面：一是屏蔽计算机硬件资源细节，为用户提供方便使用的硬件资源接口；另一方面为特定领域的程序开发人员提供抽象的逻辑开发环境。操作系统在整个计算机系统的位置如图 1-1 所示。

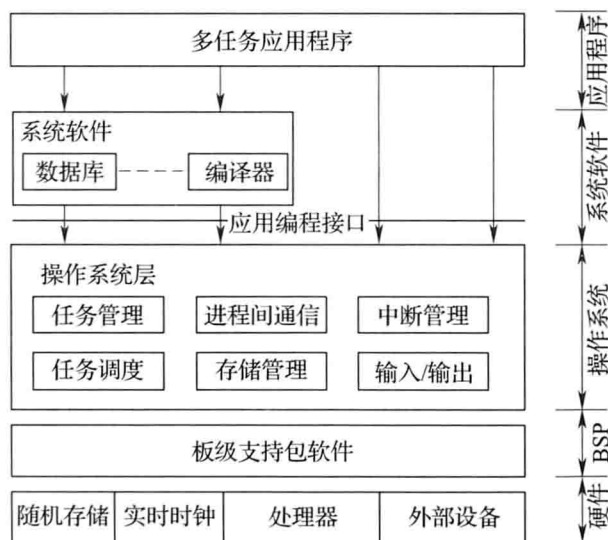


图 1-1 计算机系统构成

计算机系统采用分层结构，最底层是硬件，硬件之上是板级支持包（BSP），实现对不同 CPU 体系结构的访问以及驱动各种外部设备，板级支持包屏蔽不同硬件实现细节，使操作系统层

代码具有很好的可移植性。操作系统层主要为用户提供逻辑开发环境，操作系统层之上是系统软件层，比如数据库系统或编译器系统，应用程序处于最上层，面向特定应用领域，应用程序既可以直接使用操作系统提供的编程接口，也可以通过系统软件使用计算机资源。随着硬件技术的飞速发展，越来越多的电子设备集成到计算机系统中，用户不可能熟悉每种设备的操作，因为这些设备工作原理不尽相同，甚至差别很大，所以需要操作系统屏蔽硬件实现细节，使用户能够更加方便、快捷地使用这些不断出现的硬件设备。对用户来说，不论 CPU 体系结构采用单核架构还是多核架构，对计算机资源的使用方式不应该有什么不同。用户不需要关心自己的计算任务是运行在 CPU 的哪个核上，不需要关心自己的信息存储在硬盘的哪个扇区。当用户使用打印机时，只需要点击字处理软件上的打印菜单就可以完成打印作业，而不需要了解具体打印机的工作原理和控制细节。用户之所以感觉到计算机是如此方便使用，正是操作系统在其中起了关键性作用。

并发现象存在于日常生活中，计算机系统中也存在很多并发现象。我们都习惯于一边上网浏览新闻一边等待编译系统编译、链接新开发的软件，同时在播放器中播放 mp3 格式的音乐。多任务并发处理是现代操作系统必须具备的特征。操作系统负责维护任务状态、调度多个硬件设备并发工作。

随着软、硬件技术的发展，特别是面向服务体系结构的出现，现在应用程序开发人员可以在短时间内开发出几十万行，甚至上百万行的应用系统。在几十年前这还是不可思议的事情。那时程序开发人员使用机器语言开发软件，编写逻辑控制代码直接操作外部设备，这样做既低效又容易出错。后来操作系统出现了，操作系统屏蔽了硬件细节，为应用程序开发人员提供了一个抽象的逻辑计算环境。应用程序开发人员不需要再使用机器码编写程序，不需要直接操作硬件寄存器。在操作系统提供的计算环境下，开发人员把更多的注意力集中到特定领域的问题建模、软件体系的结构设计。使用操作系统提供的编程接口，软件编码实现变得更加方便、快捷，开发的软件运行更加稳定、更易于移植到其他系统。一般来说，任何操作系统都应该为应用程序提供以下接口。

任务管理。任务是操作系统维护的程序执行实例，是并发执行的基础，是调度的基本单位，是操作系统中最核心的概念之一。不同的操作系统往往定义不同的任务状态，但是对一般而言任务应该具有休眠状态、执行状态、就绪状态和阻塞状态。操作系统需要管理、维护任务的不同状态。为应用程序提供任务创建、任务删除、任务挂起等应用编程接口。操作系统的任务调度策略也不尽相同，典型的调度策略包括：基于优先级的抢占调度、优先级轮转调度等。操作系统的调度策略一般可以通过任务调度模式进行修改。

任务间通信。多个任务相互配合完成某项工作时，任务之间需要同步和互斥操作，操作系统为任务间通信提供基本支持。任务间通信机制包括：信号量机制、异步信号机制以及消息队列等。信号量机制又可以分为计数信号量、二值信号量和互斥信号量等。在嵌入式实时系统中还会出现优先级反转问题，一般采用优先级继承和天花板方法来解决。

存储管理。任务在运行过程中，往往需要不断地申请和释放存储资源。有两种典型的情况需要考虑：一种情况是应用程序需要固定大小的存储区，比如 TCP/IP 协议处理程序需要的协议头具有固定的大小和格式；另一种情况是可变大小的存储需求。操作系统提供分区和堆管理两种方法满足上述两种不同需求。堆管理容易造成存储碎片，所以堆管理需要在适当时机融合堆中出现的存储碎片。

原书缺页

网络操作系统和分布式操作系统方向发展。

网络操作系统。计算机网络几乎浸润到人们工作、生活的方方面面。浏览网页、发送电子邮件、撰写微博几乎成为每天必不可少的活动。网络上的协同工作成为一种工作方式。无处不在的网络缩短了人与人之间地理上的距离。为了更有效地利用网络资源解决人与人之间的信息交互、协同工作问题，网络操作系统应运而生。在个人计算机操作系统的基础上，网络操作系统根据标准化的协议构建网络管理模块，其中主要包括：通信、资源共享、系统安全和各种网络应用服务。

分布式操作系统。与网络操作系统不同，分布式操作系统主要实现目标是分布式环境下系统操作的统一性，为用户提供统一的操作界面，用户通过统一的界面实现对分布式资源的使用，而不必关心分布式操作系统把用户的操作调度到哪台计算机上执行。分布式操作系统能够充分利用网络中的各种计算资源，提高系统可靠性和容错能力。用户的计算需求可能分布到多台计算机上并行执行，从而使系统具有更短的响应时间。

1.2 嵌入式实时操作系统

嵌入式系统没有统一的定义。电气和电子工程师协会（IEEE—Institute of Electrical and Electronics Engineers）定义的嵌入式系统是指“用于控制、监视、辅助设备或机器运行的装置”（Devices Used to Control, Monitor, or Assist the Operation of Equipment, Machinery or Plants）。从上述定义可以看出嵌入式系统是软硬件结合体，软件负责逻辑控制，硬件负责按照软件发出的指令实施具体操作。嵌入式系统一般由环境感知、信息处理、控制设备操作三部分组成。实时系统属于一类非常复杂的计算机应用，与普通的应用不同，实时系统最重要的特点是对系统的实时响应有严苛的要求。系统的正确性不仅仅依赖于结果的正确，而且还依赖于结果计算的时间。

1.2.1 嵌入式实时操作系统的特点

嵌入式实时操作系统不同于桌面操作系统，由于特定的应用领域和实时性要求使其具有一些桌面系统不具有的特点，主要表现在以下几个方面。

微内核。操作系统内核架构主要分两种：宏内核（Monolithic Kernel）与微内核（Micro Kernel）。由于嵌入式实时应用的存储空间受限，不能容纳规模大的操作系统驻留，因此嵌入式实时操作系统主要采用微内核架构，能够把内核代码控制在几千字节范围内。

强实时。强实时要求操作系统不论在任何情况下都要对最紧急的事件做出快速响应，否则可能带来灾难性后果，比如在医疗设备控制系统中如果不能及时控制发生的辐射剂量异常将对患者造成不可挽回的严重后果。

高可靠。嵌入式实时系统控制设备操作，因此需要系统运行高度可靠。嵌入式实时操作系统崩溃将导致上层应用不能完成预定目标。航天、医疗以及核设备都需要高可靠的嵌入式实时系统，因此操作系统提供分区保护是必然的选择。在分区操作系统中，一个分区的崩溃不会导致其他分区任务崩溃，提高了系统可靠性。

交叉开发。嵌入式系统开发不同于桌面系统开发，需要在宿主机上运行交叉开发环境，

也就是说我们可以在 Windows 操作系统下使用交叉编译器 GCC 开发基于 ARM 或 PPC 体系架构的嵌入式应用系统。编译、链接后的嵌入式系统还需要使用通信工具把目标码下载到目标机运行。目标机上一般安装调试代理工具，方便对软件进行排错。

1.2.2 嵌入式实时操作系统实例

嵌入式实时操作系统应用十分广泛，面向不同领域的应用需求，已有大量的嵌入式实时操作系统投入使用，下面介绍几种典型的嵌入式实时操作系统。

1.2.2.1 μ C/OS - II

μ C/OS - II 由美国嵌入式系统专家 Jean J. Labrosse 采用 ANSI C 语言开发，该系统具有可移植、可裁剪、可抢占、实时多任务特点。良好的可移植性使 μ C/OS - II 广泛应用于几乎所有已知的微处理器。 μ C/OS - II 使用汇编语言开发与 CPU 硬件部分相关的代码，目的是便于移植到任何一种 CPU 上。基于 μ C/OS - II 开发应用软件，用户只需要标准的 ANSI C 交叉编译器、汇编器、连接器等工具软件。 μ C/OS - II 执行效率高、占用空间小、实时性能优良，最小内核可编译至 2KB。作为实时操作系统内核， μ C/OS - II 包含基本的操作系统功能，如任务调度、时钟管理、内存管理以及任务间通信，但由于 μ C/OS - II 良好的可扩展性和源码开放，用户可以很容易开发或移植已有的输入输出管理、文件系统以及网络相关协议栈等软件。

1.2.2.2 μ CLinux

与 UNIX 相兼容的 Linux 操作系统广泛应用于桌面系统和服务器领域。由于 Linux 性能优异并且开放源代码，使其在嵌入式领域也得到广泛应用。 μ CLinux 是基于 Linux 的嵌入式操作系统，其中 μ 表示 Micro，C 表示 Control，所以 μ CLinux 就是 Micro - Control - Linux，即微型控制 Linux。 μ CLinux 继承了 Linux 操作系统的优良特性，具有运行稳定、可移植性好的特点，同时支持各种常见的文件系统、网络协议和标准应用编程接口 (API)。 μ CLinux 可运行于各种已知的 CPU 体系结构，其编译后目标文件可控制在几百千字节，因此特别适用于内存容量受限的嵌入式系统。

1.2.2.3 VxWorks

VxWorks 操作系统是美国风河公司 (已被 Intel 收购) 于 1983 年设计开发的一款嵌入式实时操作系统，该系统广泛应用于航天、通信等嵌入式设备中。VxWorks 内核性能优越，具有良好的可靠性和卓越的实时性，其开发环境 Tornado 使用方便、具有用户友好的界面。VxWorks 操作系统支持几乎所有的嵌入式 CPU，包括 80X86 系列、MIPS、PowerPC、SPARC 以及 ARM 等。在美国装备的 F-16、F/A-18 战斗机、B-2 隐身轰炸机、爱国者导弹上都使用了 VxWorks。VxWorks 内核采用基于优先级的抢占式调度方式，同时支持同优先级任务间的时间片调度。

VxWorks 支持与 ANSI C 相兼容的输入/输出系统以及符合 POSIX 标准的异步输入/输出。VxWorks 板级支持包提供各种硬件操作的统一软件接口，保证 VxWorks 操作系统的可移植性。板级支持包功能包括硬件初始化、中断管理、时钟管理、内存地址映射、内存分配等。VxWorks 网络组件提供标准 TCP/IP 协议栈实现，包括套接字编程接口、远程过程调用 RPC、SNMP、远程文件访问以及 BOOTP 等。

1.2.2.4 Symbian

塞班 (Symbian) 是一款专为智能手机设计的嵌入式操作系统, 其前身是英国 Psion 公司的 EPOC 操作系统, 塞班的设计理念是简单实用, 支持多任务和存储器保护。塞班的运行采用事件驱动模式, 当应用程序没有需要处理的事件时, 系统会关闭 CPU, 因此使得塞班系统非常节能。2008 年 12 月 2 日, 诺基亚收购塞班公司, 2011 年 6 月 22 日, 诺基亚将塞班承包给埃森哲。由于对新兴的社交网络和 web 2.0 标准支持欠佳, 塞班占智能手机的市场份额日益萎缩。2010 年年末, 其市场占有率已被 Android 超过。诺基亚宣布, 2014 年将彻底终止对塞班系统的所有支持。

1.2.2.5 Android

Android 是一款基于 Linux 的操作系统, 主要应用于智能手机领域。Android 操作系统最初由 Andy Rubin 开发, 2005 年由谷歌注资收购, 经过谷歌开发团队改良, Android 逐渐扩展到平板电脑及其他领域。Android 操作系统采用分层架构, 包括四层, 从高层到低层分别是应用程序层、应用程序框架层、系统运行库层以及核心层。作为一款性能优异的源代码开放的操作系统, 各大知名智能手机厂商纷纷转向 Android 操作系统。基于 Android 开发应用程序, 开发人员既可以使用 Java 语言, 也可以使用 C/C++ 语言作为编程语言, 还可以使用脚本语言进行开发。

1.3 本章小结

本章简要介绍了操作系统的作用和发展简史, 重点介绍了嵌入式实时操作系统特点以及典型的嵌入式实时操作系统实例。如果读者想要了解更多的关于操作系统的基础知识可以参考本书的参考文献 [1] ~ [5]。