EMBEDDED SYSTEM

嵌入式技术与应用丛书

嵌入式可配置实时操作系统eCos技术及实现机制

王京起　黄健　沈中杰　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

附光盘

嵌入式技术与应用丛书

嵌入式可配置实时操作系统 eCos 技术及实现机制

王京起　黄　健　沈中杰　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

可配置操作系统 eCos 是源码公开的嵌入式实时操作系统，本书从源码的层次上阐述了其主要技术和实现机制。

全书共四部分。第一部分（第 1～2 章）阐述了 eCos 的发展概况和开发环境的建立；第二部分（第 3～7 章）详细论述了内核机制、可配置机制、HAL 和虚拟向量机制及其实现；第三部分（第 8～10 章）介绍了 RedBoot, GCC, GDB 和 Insight 等工具的应用，并提供了两个 eCos 的移植实例；第四部分（附录 A～C）简要介绍了 eCos 许可协议，以及 GCC 和 GDB 的常见命令的使用。

本书适合从事嵌入式系统研发的技术人员及高校相关专业的师生阅读。

图书在版编目（CIP）数据

嵌入式可配置实时操作系统 eCos 技术及实现机制 / 王京起，黄健，沈中杰编著. —北京：电子工业出版社，2005.4
（嵌入式技术与应用丛书）
ISBN 7-121-00998-6

Ⅰ.嵌… Ⅱ.①王… ②黄… ③沈… Ⅲ.实时操作系统，eCos Ⅳ.TP316.2

中国版本图书馆 CIP 数据核字（2005）第 016345 号

责任编辑：刘志红
印　　刷：北京京科印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：16.25　字数：426 千字
印　　次：2005 年 4 月第 1 次印刷
印　　数：5000 册　定价：28.00 元（含光盘 1 张）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前　言

嵌入式实时操作系统（RTOS）的实现机制对于众多爱好者而言一直是一个非常模糊的概念，出于商业产品保密的需要，RTOS 的源码对于公众而言都是遥不可及的。随着 ucos、eCos 的出现，这种状况得到了改变。

嵌入式可配置操作系统 eCos（embedded Configurable operating system）是开放源码的 RTOS，只要遵循其兼容于 GNU GPL（通用许可协议）的 eCos 许可协议，就可以对 eCos 的源码进行修改和应用。

eCos 的另外一个特点是高度的可配置性，它实现了源码层次上的配置。通过其配置工具和配置模板，用户可以快速、方便地建立适合不同平台的应用。

本书结合 eCos 的源代码，对于其各种实现机制，如调度器、线程、定时和同步机制、硬件抽象层、虚拟向量等，都做了详细的介绍。另外，对于开放过程中使用的各种工具，如 RedBoot、GCC、GDB、Insight 等也做了一定的描述。最后，通过两个 eCos 的移植实例对于移植过程提供了指导。

本书适合嵌入式操作系统的爱好者、嵌入式系统开发人员阅读，也可以作为计算机专业嵌入式实时操作系统（RTOS）课程或相关培训的参考书。

本书由王京起、黄健、沈中杰编写。其中，王京起编写了第 1 章～第 7 章、9.1 节、9.2 节、附录 A～B；黄健编写了第 8 章、9.3 节、9.4 节、第 10 章，以及附录 C；沈中杰对于本书的完成也作出了一定的贡献。另外，对侯福国为本书的审校工作提供的支持，在此表示衷心的感谢。

由于作者时间和水平所限，错误和不当之处在所难免，欢迎广大读者批评指正，作者的联系方式（E-mail）：lunarkylin@yahoo.com.cn，highbrow@eyou.com。

作　者

2005 年 1 月

目　录

第 1 章　eCos 操作系统概述

在本章中，将会讲述嵌入式实时操作系统（RTOS）的历史、现状及发展趋势，在简单介绍几种较有影响力的 RTOS 实例之后，着重介绍了 eCos 操作系统的发展概况及功能特性。

1.1　嵌入式实时操作系统简介

嵌入式实时操作系统是能在确定的时间内执行其功能，并对外部的异步事件做出响应的计算机系统。

它的关键特性之一是允许一个实时应用作为一系列独立任务来运行，各任务有各自的线程和系统资源。另外，它的关键特性还包括对于硬件中断的处理必须限制在一定的时间内。

1981 年，Ready System 发展了世界上第一个商业嵌入式实时内核（VRTX32）。在这之后，由于成本等方面的原因，嵌入式实时操作系统主要应用于军事和电信等行业。

进入 20 世纪 90 年代之后，现代操作系统的设计思想，如微内核设计技术和模块化设计思想，开始渗入嵌入式实时操作系统领域。Ready System（在 1995 年与 Microtec Research 合并），也推出新一代的 VRTXsa 实时内核，而 Windriver 推出了 VxWork。

随着嵌入式实时操作系统技术能力的成熟，以及消费类电子产品对于操作系统的需求不断迫切，嵌入式实时操作系统逐渐进入了消费类电子行业的应用之中。在目前的许多消费类电子，如手机、PDA 中，都可以看到嵌入式实时操作系统的身影。

1.1.1　嵌入式实时操作系统发展现状与未来

在进入 20 世纪 90 年代以后，嵌入式实时操作系统的发展表现出了以下明显的特点。

- 由于所需应用产品使用的处理器的多样性越来越高，要求 RTOS 自身结构的设计更易于移植，以便在短时间内支持更多种微处理器。
- 开放源码之风已波及 RTOS 厂家。许多 RTOS 厂家出售 RTOS 时，就附加了源程序代码并含生产版税。更有甚者，已经有若干 RTOS 厂家宣布自己的操作系统内核不需要版税，而转向通过配套的工具链和驱动程序来获得利益。
- 嵌入式 Linux 已经在消费电子设备中得到应用。嵌入式 Linux 得到了许多半导体厂商的支持和投资，如 Intel 和 Motorola。

未来 RTOS 的应用可能划分为 3 个不同的领域。

- 系统级。运行在一个小型的计算机系统中完成实时的控制作用。这个领域将主要是微软与 Sun 竞争之地，传统上 UNIX 在这里占有绝对优势。Sun 通过收购，让他的 Solaris 与 Chrous os（原欧洲的 1 种 RTOS）结合，微软力推 NT 的嵌入式版本“Embedded NT”。此外，嵌入式 Linux 将依托源程序码开放和软件资源丰富的优势，进入系统级 RTOS 的市场。
- 板级。RTOS 传统上的主要市场。如 Vxwork, PSOS, QNX, Lynx 和 VRTX 的应用将主要集中在航空航天、电话电信等设备上。
- SOC 级（System on Chip，片上系统）。新一代 RTOS 的领域，主要的应用场合是消费电子、互联网络和手持设备。代表的产品有 Symbian 的 Epoc，ATI 的 Nucleus、Express logic 的 Threadx。老牌的 RTOS 厂家的产品 VRTX 和 VxWorks 也很注意这个市场。

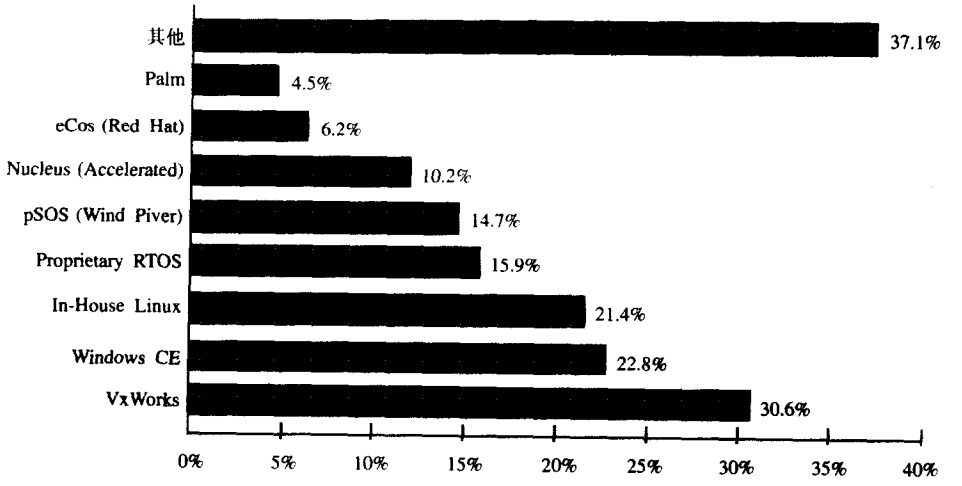

图 1.1　2003 年亚洲 RTOS 使用状况

1.1.2　VxWorks

VxWorks 是美国 Wind River System 公司（以下简称风河公司，即 WRS 公司）推出的一个实时操作系统。

VxWorks 是一个运行在目标机上的高性能、可裁剪的嵌入式实时操作系统。它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高、精、尖技术及实时性要求极高的领域中，如卫星通信、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18 战斗机、B-2 隐形轰炸机和爱国者导弹上，甚至连 1997 年 4 月在火星表面登陆的火星探测器上也使用到了 VxWorks。

1984 年 WRS 公司推出它的第一个版本——VxWorks 1.0.1，之后版本不断升级，功能也不断扩展和完善，至 2004 年又推出了 VxWorks 6.0 Beta 版，正在与合作厂商进行测试。

VxWorks 系统提供多处理器间和任务间高效的信号灯、消息队列、管道、网络透明的套接字。为了获得最快速、可靠的中断响应，VxWorks 系统的中断服务程序 ISR 有自己的

上下文。

VxWorks实时操作系统由400多个相对独立的、短小精炼的目标模块组成，用户可根据需要选择适当模块来裁剪和配置系统，这有效地保证了系统的安全性和可靠性。系统的链接器可按应用的需要自动链接一些目标模块。这样，通过目标模块之间的按需组合，可得到许多满足功能需求的应用。

VxWorks操作系统的基本构成模块包括以下部分。

- 高效的实时内核Wind

VxWorks实时内核（Wind）主要包括基于优先级的任务调度、任务同步和通信、中断处理、定时器和内存管理。

- 兼容实时系统标准POSIX

VxWorks提供接口支持实时系统标准P.1003.1b。

VxWorks提供快速灵活的与ANSI-C相兼容的I/O系统，包括UNIX的缓冲I/O和实时系统标准POSIX的异步I/O。VxWorks包括以下驱动。

网络 —— 网络设备（以太网、共享内存）。

管道 —— 任务间通信。

RAM —— 驻留内存文件。

SCSI —— SCSI硬盘，磁碟，磁带。

键盘 —— PC x86键盘（BSP仅支持x86）。

显示器 —— PC x86显示器（BSP仅支持x86）。

磁碟 —— IDE和软盘（BSP仅支持x86）。

并口 —— PC格式的目标硬件。

- 本机文件系统

VxWorks的文件系统与MS-DOS、RT-11、RAM、SCSI等兼容。

- 网络特性

VxWorks网络能与许多运行其他协议的网络进行通信，如TCP/IP、4.3BSD、NFS、UDP、SNMP、FTP等。VxWorks可通过网络允许任务存取文件到其他系统中，并对任务进行远程调用。

- 虚拟内存（可选单元VxVMI)

VxVMI主要用于对指定内存区的保护，如内存块只读等，加强了系统的健壮性。

- 共享内存（可选单元VxMP)

VxMP主要用于多处理器上运行的任务之间的共享信号量、消息队列、内存块的管理。

- 驻留目标工具

Tornado集成环境中，开发工具工作于主机侧。驻留目标外壳、模块加载和卸载、符号表都可进行配置。

- Wind基类

VxWorks系统提供对C++的支持。

- 工具库

VxWorks系统向用户提供丰富的系统调用，包括中断处理、定时器、消息注册、内存

分配、字符串转换、线性和环形缓冲区管理，以及标准 ANSI-C 程序库。

- 性能优化

VxWorks 系统通过运行定时器来记录任务对 CPU 的利用率，从而进行有效的调整，合理安排任务的运行，给定适宜的任务属性。

- 目标代理

目标代理可使用户远程调试应用程序。

- 板级支持包

板级支持包提供硬件的初始化、中断建立、定时器、内存映像等。

- VxWorks 仿真器（VxSim)

可选产品 VxWorks 仿真器，能模拟 VxWorks 目标机的运行，用于应用系统的分析。

由于 VxWorks 进入市场的时间较早，并且本身在产品性能和市场拓展方面的优秀表现，目前在 RTOS 的市场中占据极其显著的强势地位，从图 1.1 可以看出其在市场份额方面的优势。

1.1.3 OSEK/VDX

OSEK/VDX 是用于分布式实时结构的一组标准，是由欧洲汽车工业联合开发的，它的最初目标是为了解决汽车软件方面对操作系统和对软件可重用性的日益迫切的需求而制定的，目前已经在汽车产品和相关产品方面得到了非常广泛的应用。

OSEK 是德文“Offene Systeme und deren Schnittstellen fur die Elektronik im Kraftfahrzeug”的各字的首字母的缩写，其含义是“用于汽车控制器的开放式系统及其相应的接口”。该系统是由汽车制造商 BMW、Daimler Benz（现在是 Daimler-Chryler）、Opel 和 Volkswagen，汽车零部件供应商 Bosch 和 Seimens，以及德国 Karlsruhe 大学的工业信息技术学院于 1993 年 5 月在德国开始合作开发的。与此同时，法国汽车公司 PSA 和 Renault 也着手开发一个类似的系统，该系统名为 VDX。

1994 年两个组织合并为 OSEK/VDX 协会，并创立了 OSEK/VDX 指导委员会，其他公司作为技术委员会的成员加盟，协助制定技术标准。

虽然 OSEK/VDX 是由欧洲汽车工业开发的，但它不只是一个用于汽车的实时操作系统。基于这个标准的系统能够并且将要用于其他应用中，这些应用是被静态地定义，并且需要一个紧凑的分布式实时系统，这些系统必须和最小资源相适应。理想的应用将是在控制领域（制造、加工、汽车、太空）内，使用带有 8～512 KB ROM 和 1～32 KB RAM 的 8 位、16 位和 32 位的微控制器，但像小型购物电子设备和电子玩具等应用也能从这个标准中受益。在广告中，甚至曾有过基本核小到 800 字节的应用。

OSEK/VDX 又分为 3 个标准，分别是操作系统 OS、通信 COM 和网络管理 NM。

- 操作系统

本部分定义了 OSEK/VDX 中的一个可以缩放的操作系统，并详细讨论了标准中的对象，如任务、报警、资源、事件和内部消息等机制。

- 通信

本部分定义了 OSEK/VDX 通信方面的标准，规定了进程间和处理机之间的通信机制。

该标准定义了 3 个层面，数据链路层、网络层和交互层，参见图 1.2。这三个层面分别和 OSI 的某些层面对应。

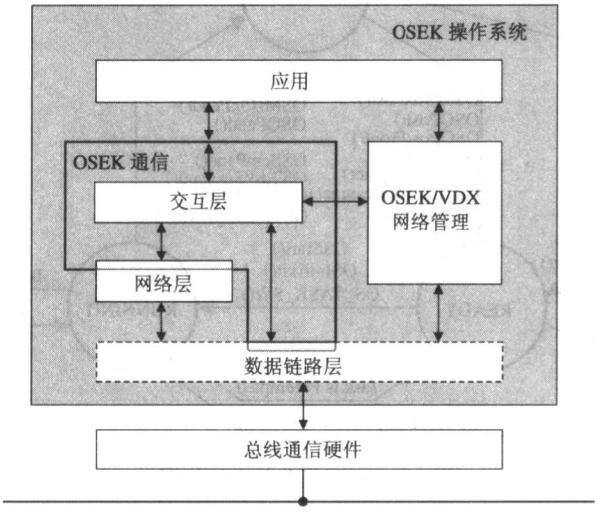

图 1.2　OSEK/VDX 分层结构

- 网络管理

本部分描述了 OSEK/VDX 网络管理标准，说明了管理网络的两种方法：直接方法和间接方法。每种方法都描述了启动节点时网络瞬态操作、稳态操作的网络管理及关闭。

需要说明的是，网络管理标准虽然定义了识别网络上节点状态的方法，却没有定义在应用程序中使用这些信息的方式。

随着 OSEK/VDX 标准的扩展和实现，它已经在包括汽车产品、小型消费电子产品方面得到了非常广泛的应用。

1.1.4　ucos II

ucos 和 ucosII 对于许多读者来说一定不会陌生，自从系统的作者 Jean Labrosse 推出该系统之后，已经在许多商业化产品或者个人案例中得到了成功的移植和应用。

ucos 之所以能够引起如此大的影响一个非常重要的原因在于它是第一个公开实现机制的实时操作系统内核。正如该系统的作者所说的那样，该系统及相关文档的公布对于许多 RTOS 厂商来说是一件噩梦般的事，从此，实时操作系统的内核不再是雾里看花。

与其他实时操作系统相比，ucos 的实现较简单，其调度机制只能使用 Bitmap 策略，每个优先级上最多只能有一个任务，其任务控制块也只是保持了任务控制所需的基本的信息。另外，在任务的同步机制方面，对于信号量的实现也较为单一。

上面介绍了几种目前比较有影响力的嵌入式实时操作系统，这些操作系统之所以能够拥有较大的市场份额或者影响力，有许多影响因素。包括成本、性能、移植方便性、源码开放程度等。

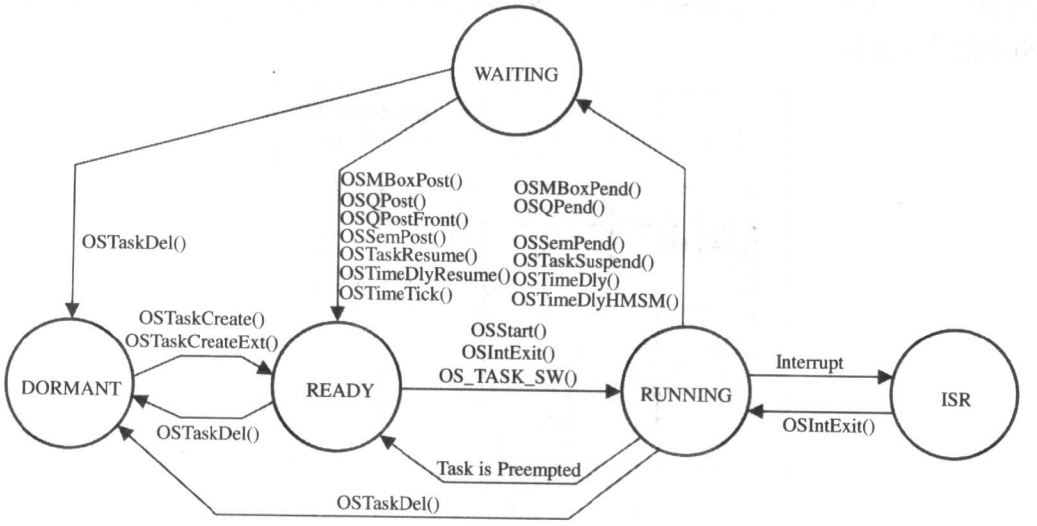

图 1.3 ucos 任务状态转换图

图 1.4 是制约 RTOS 选择结果的各个因素的影响力比较。从图 1.4 可以看出，易用性和源码开放程度在重要性方面居于各因素之首。而本文的主要目标，eCos 操作系统在易用性和源码开放程度方面具有非常明显的优势，这也决定了 eCos 系统的光明前景。

1.2 eCos 操作系统历史、现状及前景

1997 年，市场上有 100 多种商业化的嵌入式实时操作系统。在这些操作系统中，没有哪一种能够占据明显的优势。除此之外，还有数不清的非商业化的嵌入式实时操作系统应用在各个特定项目中，而在这些特定的项目之外，不可能再有其他项目使用这些系统。

eCos 就是在这种情况下横空出世，并脱颖而出的。

1.2.1 eCos 操作系统的历史

为了解决嵌入式实时操作系统方面群雄割据的窘境，cygnus 公司希望能够设计一种具有高度适应性的操作系统，这种操作系统能够实现尽量多的组合变种，从而可以满足各种复杂程度的应用状况，而其源代码库却只有一个。

通过市场调查发现，许多用户自己动手编写 RTOS 的主要原因有两个。一个是 RTOS 每个拷贝的版税；另外一个是用户希望自己手动的改动最小，从而不希望使用自己无法理解或控制的系统。

考虑到上述各种因素，1998 年，cygnus 推出了自己的 RTOS 解决方案——eCos（Embedded Configurable Operating System）。这种操作系统开放源码不需要用户支付版税，另外，通过该系统提供的配置工具，可以实现源代码级别的配置性（裁剪性），用户不需要做什么改动，或者只需要做极小的改动就可以建立其自己特定的系统。

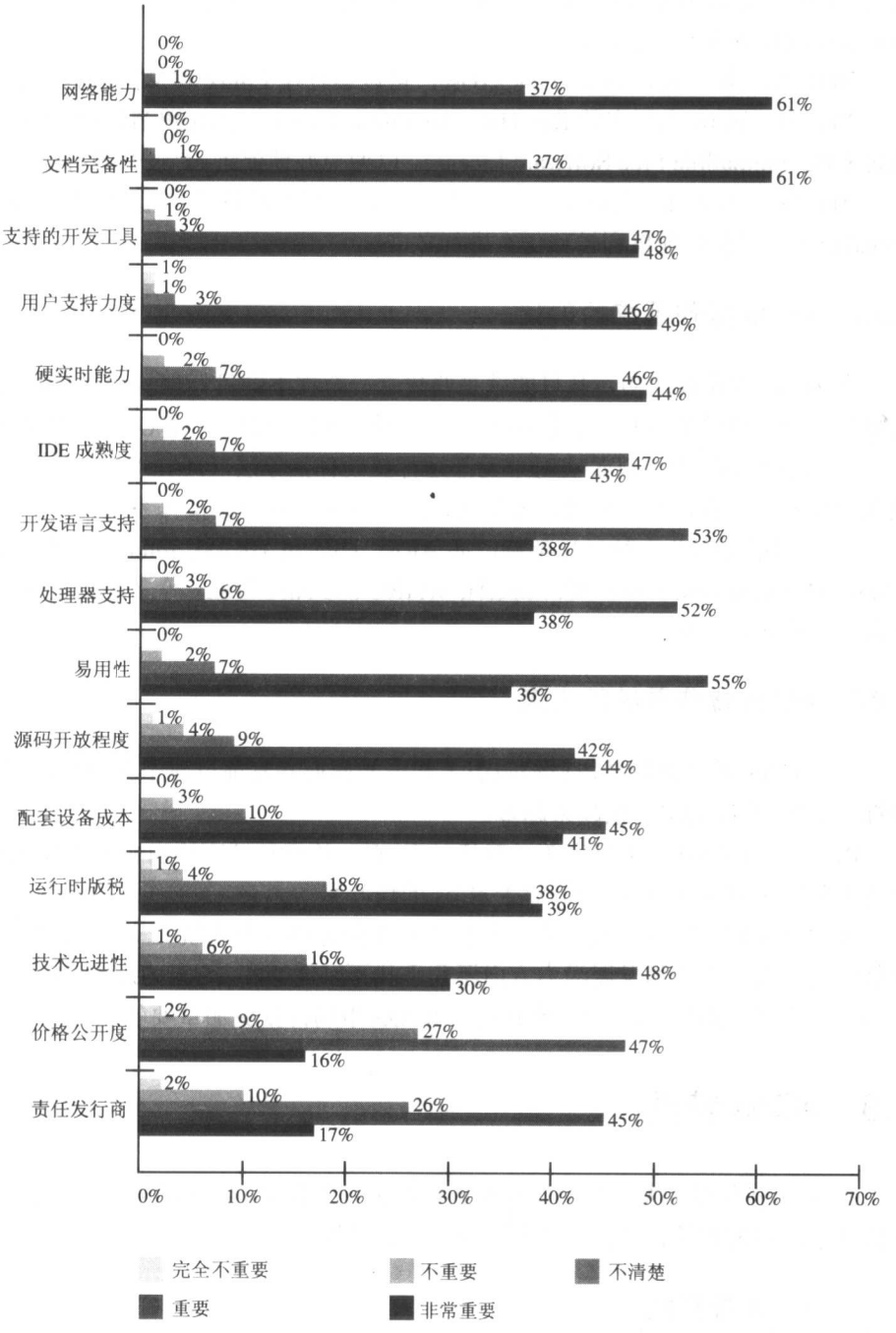

图 1.4　影响 RTOS 选择的因素

1999 年，cygnus 提出了 EL/IX API 接口的概念，该接口使得开发人员可以在一般的 PC 上开发 Linux 或嵌入式软件，并且很方便地使之运行在 linux 或 eCos 的嵌入式设备。

2000 年，推出了 eCos 1.3 版本。该版本支持网络开发功能，提供了自己的 TCP/IP 栈

支持。同年，cygnus 被 Red Hat 公司收购，eCos 被继续支持，并加入了嵌入式调试和 boot rom 监控软件 redboot 的支持。

2001 年，推出 eCos 配置工具 2.0 版，可以运行在 Linux 和 Windows 平台上。

2002 年，eCos 由原来的 Red Hat eCos Public License (RHEPL)认证转为与 GPL 兼容的认证 GPL-compatible Free Software License，以便更好地促进 eCos 开源特性的维护和发展。

2004 年，Red Hat 宣布了要将其对于 eCos 的权利转移给自由软件联盟 Free Software Foundation，目前此项权利转移仍在进行之中。

1.2.2 eCos 操作系统的现状

在 eCos 的设计之初，其目的就是要建立一个在不同目标架构和平台间可以轻松移植的操作系统。这些架构和平台包括 8 位、16 位、32 位架构，MPU、MCU 或者 DSP。

eCos 的内核、库，以及实时单元都运行在硬件抽象层（HAL）之上，只要将 HAL 和相关的设备驱动移植到目标处理器架构之后，就可以运行该操作系统。

eCos 目前支持几十种目标架构，如 ARM, Hitachi H8300, Intel x86, MIPS, Matsushita AM3x, Motorola 68k, PowerPC, SuperH, SPARC and NEC V8xx 等。目前还有许多的移植和开发工作在进行之中。

1.2.3 eCos 操作系统的未来

由于 eCos 的开源特性，已经吸引了若干的商业开发项目或个人项目应用 eCos，学习和探讨 eCos 系统的用户也越来越多。

Red Hat 在 2004 年 1 月宣布了其将对于 eCos 的权利转交给自由软件联盟的想法之后，也更加坚定了 eCos 的拥护者们对于 eCos 的信心和动力。

环顾目前的 RTOS 市场，在市场份额和性能方面确有系统比 eCos 更有优势，但从自由软件的巨大号召力和发展潜力方面还几乎找不到能够和 eCos 相比的系统。相信，随着自由软件号召力的显现，eCos 也必将在 RTOS 市场占据更加重要的地位。

1.3 eCos 特性

eCos 能够获得广大用户的认可和无数拥护者的肯定，都来源于其各项突出的特性，这些特性包括可配置性、开源特性和功能方面的特性。

1.3.1 可配置性

eCos 的配置特性是它的一个关键特性。eCos 可以实现源代码行级别的配置，也就是可以对源代码中的某一行或某几行进行选择，已决定最后实现的功能。

传统方式的操作系统的可配置特性一般停留在函数级别或库级别，即通过在运行时或链接时决定是否链接某些函数库或库中的函数来进行功能上的配置控制，很明显，这些配

置实现形式对于最终的代码尺寸并没有明显的帮助。

而与之形成鲜明对比的 eCos 配置特性不仅可以实现功能上的控制，还能对最后得到的代码尺寸有明显的控制效果。

eCos 可配置机制的实现是通过其配置工具和组件库来实现的。

所谓配置工具包括图形化或命令行的 configure tool。该工具可以用来选择需要添加到最终系统中的配置包、组件、开关选项、编译选项等，可以用来有效地控制系统的功能模块、功能特性及生成代码的特征。图 1.5 是图形化配置工具的界面，其中各个窗口及菜单的功能将会在后面做详细的介绍。

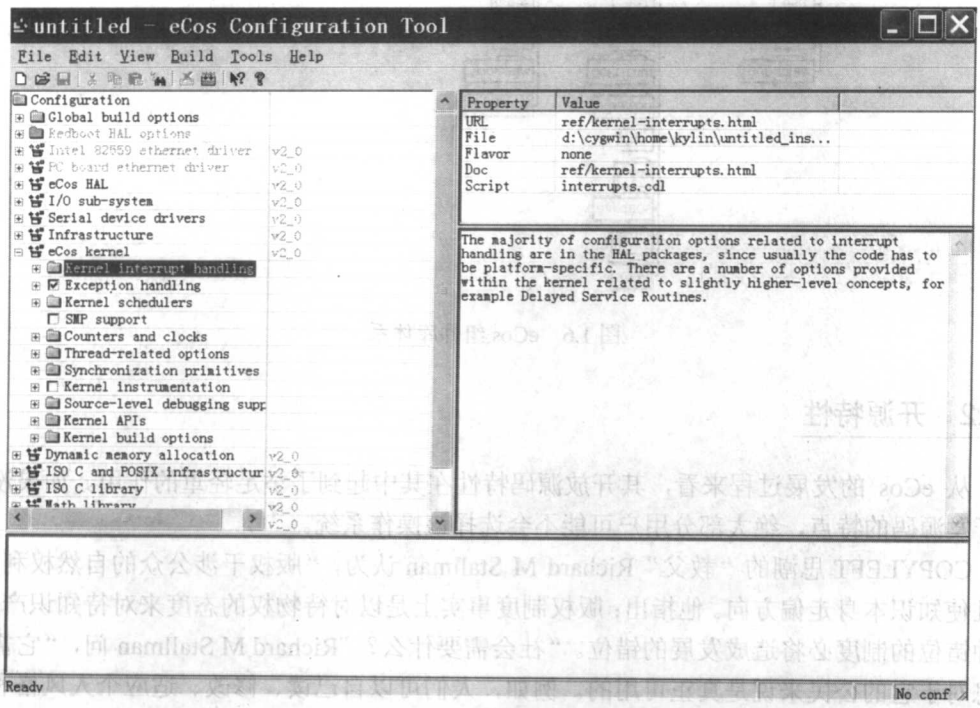

图 1.5　配置工具图形化界面

组件库是 eCos 可重用的代码库，库里面的目录体系在很大程度上代表了 eCos 系统的模块体系，如不同的网络协议栈、不同的硬件抽象层等。图 1.6 是一个组建库的体系结构实例，各目录的功能也将会在后面的章节做详细介绍。

通过配置工具除了可以方便快捷地建立用户自定义的系统之外，还可以用来管理组件库，比如往组件库中添加新的组件包、从组件库中卸载某些组件包等。按照 eCos 制作配置包的流程，用户还可以建立用于发行的配置包，这些配置包建立起来之后，其他用户在得到配置包后，通过添加配置包就可以将其方便地加入到用户自己的 eCos 组件库中，从而提高了系统的可重用性，并避免了不必要的重复和时间、人力方面的浪费。

实际经验表明，通过 eCos 的配置机制，可以有效地建立用户自定义的应用，而不会带入不必要的功能或实现。