

信息科学与技术丛书

电子工程系列

● 蒋句平 编著

嵌入式可配置实时操作系统 eCos 开发与应用

electrical engineering

electrical engineering

- eCos 开发环境、配置方法
- eCos 结构的全面分析与详尽的 API 函数
- eCos 的移植
- 完整的 eCos 原码和全套开发工具
- 基于 eCos 的嵌入式 Web 服务器

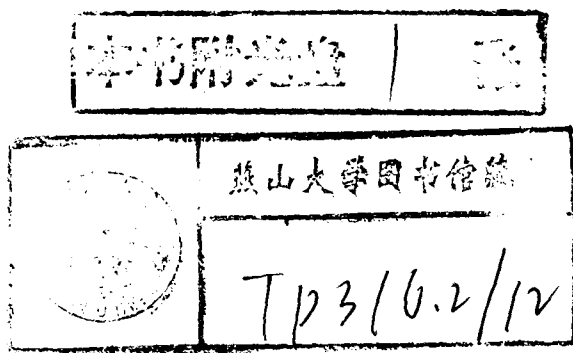
 机械工业出版社
CHINA MACHINE PRESS



信息科学与技术丛书

嵌入式可配置实时操作系统 eCos 开发与应用

蒋句平 编著



机械工业出版社



0775767

嵌入式可配置实时操作系统 eCos 是一种完全免费的开放源代码的软件, 适合于深度嵌入式应用。本书全面介绍 eCos 系统的基本结构, 详细描述 eCos 开发环境的建立及其配置方法, 从各个层次对 eCos 的各个组成部分进行阐述和分析, 并通过实例说明如何开发基于 eCos 的嵌入式应用。本书附带光盘包含了最新版本的 eCos 2.0 源代码、联机说明、开发工具源代码以及应用程序 eCos 嵌入式 Web 服务器, 可以直接使用光盘内容建立完整的 eCos 开发环境。

本书可作为广大嵌入式系统研发人员及其他相关科研人员的技术参考书, 也可作为在校学生学习嵌入式系统的参考教材。

图书在版编目 (CIP) 数据

嵌入式可配置实时操作系统 eCos 开发与应用/蒋句平编著. —北京: 机械工业出版社, 2004.1

(信息科学与技术丛书)

ISBN 7-111-13242-4

I. 嵌... II. 蒋... III. 实时操作系统, eCos IV. TP316.2

中国版本图书馆 CIP 数据核字 (2003) 第 094918 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 时 静

责任印制: 路 琳

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2004 年 1 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·23 印张·569 千字

0001—3000 册

定价: 39.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话: (010) 68993821、88379646

封面无防伪标均为盗版

出版说明

随着信息科学与技术的迅速发展,人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问,在节奏越来越快的工作和生活中,人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书,来获取新知识和新技能,从而不断提高自身素质,紧跟信息化时代发展的步伐。

众所周知,在计算机硬件方面,高性价比的解决方案和新型技术的应用一直备受青睐;在软件技术方面,随着计算机软件的规模和复杂性与日俱增,软件技术受到不断挑战,人们一直在为寻求更先进的软件技术而奋斗不止。目前,计算机在社会生活中日益普及,随着因特网延伸到人类世界的层层面面,掌握计算机网络技术和理论已成为大众的文化需求。也正是由于信息科学与技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用,所以这些专业领域中的研究人员和工程技术人员将越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状,机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容,注重理论与实践相结合,内容实用,层次分明,语言流畅,是信息科学与技术领域专业人员不可或缺的图书。

现今,信息科学与技术的发展可谓一日千里,机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作,为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

嵌入式产品是一类充满巨大商机的产业。在开发嵌入式产品的时候,开发人员将面临选择哪种嵌入式操作系统的问题。有许多因素值得考虑:软件的价格问题、版税问题、开发工具的好坏、是否提供源代码、所提供的实时操作系统具有哪些特性等等。在许多情况下,价格和版税是首先考虑的因素,低成本的解决方案是一种最好的选择,降低产品价格是提高产品竞争力的一个重要举措。另一个必须考虑的问题是前期投资问题,将现有软件代码移植到新的硬件平台上时这一问题显得尤为突出,软件的移植能力在很大程度上会直接影响产品的开发周期。另外,软件的可重用性和配置能力也是开发嵌入式产品必须考虑的因素。采用嵌入式可配置实时操作系统 eCos 便是针对所有这些问题的一个很好的解决方案。

eCos 于 1997 年起源于 Cygnus 公司,后来成为 RedHat 的一个非 Linux 嵌入式操作系统。在短短的几年时间内,eCos 的发展极为迅速,它已逐渐被人们接受,并受到越来越多的嵌入式产品开发人员的青睐,目前市场上已经有了许多成功应用 eCos 的嵌入式产品。eCos 是一种适合于深度嵌入式应用的实时操作系统,它是一种免费的、无版权限制(无版税)的开放源代码的软件。eCos 的独特之处是它的可配置能力和配置机制,这是其他嵌入式操作系统无法比拟的。此外,eCos 还具有良好的开放性、兼容性和可扩展性,可移植能力强,目前它能支持十余种市场上流行的嵌入式处理器。开发人员在不同平台上进行移植时,几乎不用修改或稍加修改就可以完成应用程序的移植工作。

eCos 不仅是开发嵌入式产品的一个很好的选择,还是学习嵌入式操作系统原理与方法的一个很好的实例教材。eCos 除了提供了嵌入式实时操作系统所必须具备的全部功能外,它的配置机制、组织结构以及软件源码都值得其他软件借鉴。与其他嵌入式操作系统相比,eCos 简单、易学、易于操作,利用普通微机就可以很容易地建立一个完整的开发环境,而且不需要特殊的设备。加之它又是一种免费软件,因此特别适合国内学生将其作为学习和研究嵌入式系统的理想平台。为此,本书大部分操作和例子都以普通微机作为目标系统。本书光盘提供了 eCos 的全部源码、开发工具和应用程序,读者可以直接使用光盘提供的内容建立完整的开发平台。

全书共分十三章,为兼顾初学者和有经验的开发人员,采用从简到繁、由浅入深的方式对 eCos 进行全面阐述,读者按照本书内容的编排可以逐步了解和掌握 eCos,并且可以着手开发自己的 eCos 应用。

本书读者对象包括广大嵌入式系统研发人员、在校学生以及其他嵌入式系统爱好者,要求具有一定的软件(C 语言)编程经验和基本的硬件知识。

本书的编写得到了我的同事及家人的帮助和支持,窦强博士对本书初稿进行了细致的审阅,在此谨向他们表示衷心的感谢!

由于编者水平有限,书中难免存在一些错误和缺点,殷切希望读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 概述	1
1.1 什么是 eCos	1
1.1.1 起源与历史	2
1.1.2 功能与特性	3
1.1.3 eCos 核心组件	4
1.1.4 对硬件的支持	5
1.1.5 eCos 资源	5
1.2 eCos 的可配置性	6
1.2.1 配置的必要性	7
1.2.2 配置方法	7
1.3 eCos 的组织结构	8
1.3.1 eCos 的层次结构	9
1.3.2 eCos 源码结构	11
1.4 eCos 的一些基本概念	12
1.4.1 组件框架	12
1.4.2 配置选项	12
1.4.3 组件	12
1.4.4 包	13
1.4.5 组件仓库	13
1.4.6 配置	13
1.4.7 目标系统	14
1.4.8 模板	14
1.4.9 属性	14
1.4.10 约束条件	15
1.4.11 冲突	15
1.4.12 组件定义语言 CDL	15
第 2 章 eCos 开发环境的建立	16
2.1 系统需求	16
2.1.1 主机系统需求	16
2.1.2 目标系统需求	16
2.2 开发工具	17
2.2.1 Cygwin	18
2.2.2 交叉编译工具	18
2.2.3 eCos 配置工具	19
2.3 Cygwin 的安装与设置	19
2.4 GNU 交叉编译工具的编译与配置	22

2.5 eCos 源码与配置工具的安装	25
2.5.1 eCos 的安装	26
2.5.2 eCos 配置工具	28
2.6 建立 eCos 开发环境	30
2.6.1 基于 x86 的 eCos 开发平台	30
2.6.2 建立 RedBoot 引导环境	31
第 3 章 eCos 配置工具与编程实例	34
3.1 eCos 图形配置工具	34
3.2 图形配置工具的使用	35
3.2.1 组件仓库位置	35
3.2.2 配置文件的管理	36
3.2.3 模板选择	36
3.2.4 选项配置	37
3.2.5 冲突的解决	39
3.2.6 配置选项的查找	40
3.2.7 编译	40
3.2.8 执行	42
3.3 命令行配置工具	44
3.3.1 ecosconfig 配置工具	44
3.3.2 使用 ecosconfig 配置 eCos	45
3.4 eCos 应用程序	46
3.4.1 使用编译工具	46
3.4.2 简单的 hello 程序	47
3.4.3 多线程编程例子	49
3.4.4 时钟和告警处理程序	51
第 4 章 RedBoot	55
4.1 功能与应用	55
4.1.1 RedBoot 的安装	55
4.1.2 RedBoot 用户界面	56
4.1.3 RedBoot 环境配置	56
4.2 RedBoot 命令	58
4.2.1 基本命令格式	58
4.2.2 RedBoot 普通命令	60
4.2.3 Flash 映像系统(FIS)	63
4.2.4 Flash 内配置信息的管理	66
4.2.5 RedBoot 程序执行控制	68
4.3 RedBoot 的配置与编译	68
4.3.1 RedBoot 软件结构	68
4.3.2 使用 eCos 图形配置工具	69
4.3.3 使用命令行配置工具 ecosconfig	72
4.4 RedBoot 的更新与运行	73

第 5 章 系统内核	76
5.1 系统内核结构	76
5.2 内核调度机制	77
5.2.1 位图调度器	78
5.2.2 多级队列调度器	78
5.2.3 调度器操作及 API 函数	79
5.3 内存分配	80
5.3.1 内存分配机制	81
5.3.2 固定长度内存分配 API	82
5.3.3 可变长度内存分配 API	84
5.4 中断处理	86
5.4.1 线程与中断处理程序	86
5.4.2 中断的处理	87
5.4.3 内核中断处理 API 函数	88
5.5 例外处理	91
5.5.1 例外处理程序	92
5.5.2 例外处理内核 API 函数	93
5.6 SMP 支持	93
5.6.1 SMP 系统的启动	94
5.6.2 SMP 系统的调度	94
5.6.3 SMP 系统的中断处理	95
5.7 计数器与时钟	95
5.7.1 计数器	96
5.7.2 时钟	98
5.7.3 告警器	100
5.8 应用程序入口	102
5.8.1 调用环境	102
5.8.2 应用程序编程要求	103
5.8.3 应用程序的启动	104
第 6 章 线程与同步	106
6.1 线程的创建	106
6.1.1 创建新线程	106
6.1.2 线程入口函数	107
6.1.3 线程优先级	108
6.1.4 堆栈和堆栈大小	108
6.1.5 线程创建例子程序	109
6.2 线程信息的获取	110
6.3 线程的控制	111
6.4 线程的终止和消除	112
6.4.1 线程终止函数	112
6.4.2 线程消除函数	113

6.5	线程优先级操作	113
6.6	per-thread 数据	114
6.7	同步原语	116
6.8	互斥体	117
6.8.1	互斥体的实现与操作	118
6.8.2	互斥体 API 函数	118
6.8.3	优先级倒置	120
6.9	条件变量	122
6.9.1	条件变量的使用	122
6.9.2	条件变量 API 函数	124
6.10	信号量	125
6.10.1	信号量的使用	126
6.10.2	信号量 API 函数	126
6.11	信箱	128
6.11.1	信箱的使用	128
6.11.2	信箱 API 函数	129
6.12	事件标志	131
6.12.1	事件标志的使用	132
6.12.2	事件标志 API 函数	133
6.13	Spinlock	135
6.13.1	Spinlock 的使用	135
6.13.2	Spinlock 内核 API 函数	136
第 7 章	标准 C 与数学库	138
7.1	标准 C 与数学库的配置	138
7.2	非 ISO 标准函数	139
7.3	数学库兼容方式	140
7.4	一些实现细节	141
7.5	线程安全性	143
7.6	C 库启动函数	144
第 8 章	设备驱动程序与 PCI 库	145
8.1	设备驱动程序用户 API	145
8.1.1	设备的查找	146
8.1.2	向设备传送数据	146
8.1.3	读取设备数据	146
8.1.4	读取设备配置信息	146
8.1.5	对设备的配置	147
8.2	驱动程序与内核及 HAL 的接口	147
8.2.1	eCos 中断模块	147
8.2.2	同步	148
8.2.3	SMP 支持	150
8.2.4	驱动程序模式	150

8.2.5	驱动程序与内核及 HAL 的接口 API 函数	151
8.3	eCos 驱动程序设计	163
8.3.1	设备驱动程序的基本结构	163
8.3.2	串口驱动程序设计	165
8.4	串口驱动程序	168
8.4.1	串口(raw serial)驱动程序	168
8.4.2	TTY 驱动程序	173
8.5	PCI 库	175
8.5.1	PCI 总线操作	175
8.5.2	PCI 库 API	181
第 9 章	文件系统	188
9.1	文件系统表格	188
9.1.1	文件系统表(File System Table)	188
9.1.2	安装表(Mount Table)	190
9.1.3	文件表	192
9.2	文件目录	194
9.3	同步	194
9.4	初始化和安装	195
9.5	文件操作	196
9.5.1	文件系统的安装 mount 与卸载 umount	197
9.5.2	open、creat 和 close 函数	198
9.5.3	read、write 和 lseek 函数	198
9.5.4	fcntl 函数	199
9.5.5	dup 和 dup2 函数	200
9.5.6	stat 和 fstat 函数	201
9.5.7	access 函数	202
9.5.8	link, unlink, remove 和 rename 函数	202
9.5.9	mkdir 和 rmdir 函数	203
9.5.10	opendir, readdir, rewinddir 和 closedir 函数	203
9.5.11	chdir 和 getcwd 函数	204
9.5.12	Socket 操作	204
9.6	创建文件系统	205
9.7	RAM 文件系统	209
9.7.1	文件和目录节点	209
9.7.2	目录	211
9.7.3	数据存储机制	211
9.8	ROM 文件系统	213
9.9	文件操作实例	214
第 10 章	网络支持与编程	222
10.1	eCos 网络配置	222
10.2	以太网驱动程序设计	223

10.2.1	底层驱动程序基本框架	224
10.2.2	驱动程序内部函数的实现	227
10.2.3	高层驱动程序函数	231
10.2.4	数据的发送和接收过程	231
10.3	TCP/IP 协议栈支持	232
10.3.1	特性支持与配置	233
10.3.2	API 函数	234
10.4	FTP 客户端	236
10.5	DNS 客户端	237
10.6	eCos 网络编程实例	238
10.6.1	网络通信测试程序	238
10.6.2	编程实例——ping 程序	239
第 11 章	硬件抽象层与 eCos 移植	245
11.1	硬件抽象层 HAL	245
11.2	硬件抽象层的结构	247
11.2.1	HAL 的类型	247
11.2.2	硬件抽象层文件描述	248
11.3	硬件抽象层接口	250
11.3.1	基本定义	250
11.3.2	体系结构描述	251
11.3.3	中断处理	255
11.3.4	I/O 操作	258
11.3.5	Cache 控制	259
11.3.6	SMP 支持	261
11.3.7	诊断支持	264
11.3.8	链接脚本	265
11.4	例外处理	266
11.4.1	HAL 的启动处理	266
11.4.2	同步例外与异步中断的处理	267
11.5	虚拟向量	269
11.6	eCos 的移植	270
11.6.1	平台抽象层的移植	271
11.6.2	变体抽象层的移植	279
11.6.3	体系结构抽象层的移植	282
第 12 章	组件结构与 CDL	290
12.1	eCos 的配置机制	290
12.2	eCos 组织结构及编译过程	291
12.2.1	软件包与组件仓库	291
12.2.2	软件包的内容与格式	292
12.2.3	编译过程	293
12.3	组件定义语言 CDL	294

12.3.1	CDL 命令	295
12.3.2	CDL 属性	298
12.4	选项命名约定	305
12.5	Tcl 简介	307
12.5.1	基本语法	307
12.5.2	变量	308
12.5.3	命令替换	308
12.5.4	引号和花括弧的使用	308
12.5.5	反斜杠和注释	309
12.6	表达式和值	310
12.6.1	选项的值	310
12.6.2	普通表达式	312
12.6.3	目标表达式	313
12.6.4	列表表达式	314
12.7	接口	314
12.8	更新 ecos.db 数据库	316
第 13 章	eCos 嵌入式 Web 服务器	318
13.1	嵌入式 Web 服务器 LibHTTPD	318
13.2	配置和建造 eCos	320
13.3	Web 服务器编程	321
13.4	运行 Web 服务器	328
附录		
附录 A	eCos 硬件支持情况	335
附录 B	eCos 实时特性	341
参考文献	359

第 1 章 概 述

随着信息技术的飞速发展和互联网的迅速普及,人们已经步入了数字化时代。形式多样的数字化产品已经开始成为继 PC 机之后信息处理的主要工具,消费电子、计算机、通信(3C)一体化已经成为数字化产品的一种趋势,并且正在逐步形成一个充满商机的巨大产业。在这样一种数字化潮流下,嵌入式系统已成为当前研究和应用的热点之一。嵌入式产品已经涉及到人们日常生活和工作的各个方面,手机、个人数字助理 PDA、MP3 播放器、机顶盒、VCD、智能外设、数码相机、数码电视、网络家电、GPS 设备、数控机床等各种各样的数字设备都是广泛使用的嵌入式产品。

嵌入式系统是一种以应用为中心、软硬件可裁减的专用计算机系统。它具有很强的灵活性,可以适应应用系统对功能、可靠性、成本、体积、功耗等方面的严格要求。嵌入式系统主要由嵌入式处理器、相关支撑硬件、嵌入式操作系统以及应用软件系统等组成,它是一种集软、硬件于一体的可独立工作的设备或组件。嵌入式操作系统是一种实时的、支持嵌入式系统应用的操作系统软件,它是嵌入式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程序、系统内核、设备驱动接口、通信协议、图形界面等。与通用操作系统相比较,嵌入式操作系统在系统实时高效性、硬件的相关依赖性、软件固态化以及应用的专用性等方面具有较为突出的特点。

目前,嵌入式操作系统的种类较多,其中比较流行的有 VxWorks、Windows CE、pSOS、Palm OS、嵌入式 Linux、QNX 和 Nuclear 等产品。这些嵌入式操作系统在开放性、实用性以及性能等方面各有千秋,而且大多数为商用产品。除了商用产品外,另外还有一些免费的嵌入式操作系统,eCos 就是这些产品的一个典型代表。嵌入式可配置操作系统 eCos(Embedded Configurable Operating System)是一种开放源代码软件,它是一种免费、无版权限制的适合于深度嵌入式应用的实时操作系统。eCos 最为显著的特点是它的可配置性、可裁剪性、可移植性和实时性,它的一个主要技术创新是其功能强大的配置系统,可以在源码级实现对系统的配置和裁剪。正是由于这些特性,eCos 已引起越来越多的关注,同时也吸引了越来越多的厂家使用 eCos 开发其新一代嵌入式产品。

1.1 什么是 eCos

eCos 是一种嵌入式可配置实时操作系统,适合于深度嵌入式应用,主要应用对象包括消费电子、电讯、车载设备、手持设备以及其他一些低成本和便携式应用。eCos 是一种开放源代码软件,无任何版权费用。eCos 具有很强的可配置能力,而且它的代码量很小,通常为几十到几百 KB。它的最小配置形式是它的硬件抽象层 HAL 所提供的引导程序 RedBoot,可以支持很大范围内许多不同的处理器和平台。它的最大配置形式是一个完整的实时操作系统,所提供的服务和支持能与其他大多数商用实时操作系统相媲美。eCos 为开发人员提供了一个能涵盖大范围内各种不同嵌入式产品的公共软件基础结构,使得嵌入式软件开发人员可以集中

精力去开发更好的嵌入式产品,而不是停留在对实时操作系统的开发、维护和配置上。

1.1.1 起源与历史

eCos 最初起源于 Cygnus 公司。Cygnus 创建于 1989 年,创始人为 Michael Tiemann、David Henkel-Wallace 和 John Gilmore,其目的是为开源软件提供高质量的开发和支持。经过几年的艰苦努力,最后推出了今天被人们广泛使用的 GNUPro 开发工具包,包括 GCC(ANSI-C 编译器)、G++(C++ 编译器)、GDB(源码级和汇编级调试工具)、GAS(GNU 汇编器)、LD(GNU 链接器)、Cygwin(Windows 下的 UNIX 环境)、Insight(GDB 图形界面 GUI)等。

Cygnus 对 eCos 的设计始于 1997 年春季,其主要目的是为市场提供一种低成本、高效率、高质量的嵌入式软件解决方案,同时要求该软件所占系统资源极少。eCos 和 GNUPro 相辅相成,扩大了 Cygnus 的产品线。eCos 从设计之初就考虑到了嵌入式系统中内存资源的限制以及嵌入式硬件平台的多样性。通过与其他许多半导体公司的协作,Cygnus 成功构造了一个可以对硬件层进行抽象的实时操作系统(RTOS),并且具有高度可配置性。这些特性使得 RTOS 可以适合于各种各样的嵌入式系统,这种 RTOS 就是 eCos。eCos 的高度可配置性可以显著缩短嵌入式产品的开发周期。

Cygnus 对 eCos 的另一个设计目标是降低嵌入式产品的成本。低成本是嵌入式系统开发中必须考虑的一个重要因素。通过使用开放源代码的形式,eCos 基本上不需要任何费用。它是一种完全免费的软件,任何公司和个人都可以直接从 Internet 上下载其源码和相应的开发工具,并且可以自由地进行修改和扩展,eCos 产品的发布也无需交纳任何版权费用。用户可以自由使用 eCos,但是要求公布对 eCos 的改动,这是为了提高或促进 eCos 发展的一种措施。当然,用户的应用程序不必公开。

Cygnus 于 1998 年 11 月发布了第一个 eCos 版本(eCos 1.1)。它只支持有限的几种处理器结构:Matsushita MN10300、Toshiba TX39 和 PowerPC。1999 年 5 月,Cygnus 发布了 eCos 的第二个版本 eCos 1.2.1,它在 eCos 1.1 的基础上增加了许多新的特性,并扩大了对处理器的支持范围,包括 ARM7、SPARClite MB8683x 系列处理器、PowerPC(MPC860、MPC850、MPC823)、VR4300 和 SH3 等。

1999 年 11 月,Red Hat 收购了 Cygnus 公司。在此后的几年里,eCos 作为 Red Hat 的一个嵌入式产品得到了迅速的发展。2000 年 3 月,Red Hat 发布了 eCos 的第三个版本 eCos 1.3.1。eCos 1.3.1 在 1.2.1 版本的基础上又增加了许多新的特性(TCP/IP 协议栈、PCI 支持等),并扩充了对处理器平台的支持,包括 ARM Thumb、ARM9、StrongARM、AM33、PowerPC、VR4300、SH3、x86 等。2000 年 8 月,eCos 增加了对目标系统引导和调试固件(Firmware)的支持,即 RedBoot,它是 Red Hat 的一个标准嵌入式系统引导和 Debug 环境。RedBoot 目前已被许多嵌入式产品所采用。

2002 年,Red Hat 由于财务方面的原因,裁减了 eCos 开发队伍,这在一定程度上影响了人们对 eCos 的信心。但 eCos 的发展并没有因此而停止,RedHat 随后宣称将继续支持 eCos 的发展,而原来的 eCos 主要开发人员组建了一个新的 eCosCentric 公司,继续进行 eCos 的开发和技术支持。2002 年 4 月,eCosCentric 发布了 eCos 2.0 alpha 版,2003 年 3 月,又发布了 eCos 2.0 beta 版,2003 年 5 月,正式发布了 eCos 2.0。它所支持的处理器包括:ARM、StrongARM、XScale、SuperH、Intel x86(IA32)、PowerPC、MIPS、AM3x、Motorola 68K/Coldfire、SPARC、

Hitachi H8/300H 和 NEC V850 等,能支持近百种当时市场上广泛使用的嵌入式系统开发平台和评估版。eCos 2.0 与它的前一个版本 eCos 1.3.1 相比,增加和改进的功能包括:

- 1) RedBoot,基于 eCos 的引导和调试 Firmware。
- 2) TCP/IP 协议栈,支持 BOOTP/DHCP、DNS、TFTP/FTP、IPv6 和 HTTPD。
- 3) RAM、ROM 和 Flash 文件系统。
- 4) 电源管理。
- 5) USB 支持。
- 6) POSIX 兼容 API。
- 7) 对称多处理器 SMP 支持。

目前,许多公司都在使用 eCos,并先后成功推出了使用 eCos 的嵌入式产品。部分产品有: Brother HL-2400 CeN 网络彩色激光打印机、Delphi Communiport 车载信息处理系统(MPU)、Omega Hip Zip 数字音频播放器、Ikendi 指纹识别系统、3G LAB 移动电话、GPS 卫星地面设备、MP3 播放器、CrosStor RAID 系统等等。

1.1.2 功能与特性

eCos 是一个适合于深度嵌入式应用的开放源代码实时操作系统。它能满足嵌入式 Linux 难以满足的对嵌入空间的需求, Linux 目前内核最小约 500KB, 占用 1.5 MB 内存, 而 eCos 只占用几十到几百 KB。eCos 使用了多任务抢占机制, 具有最小的中断延迟, 支持嵌入式系统所需的所有同步原语, 并拥有灵活的调度策略和中断处理机制。eCos 还提供了普通嵌入式应用中所需要的全部功能, 包括设备驱动程序、内存管理、例外处理、标准 C、数学库等等。除了这些对系统运行时的支持外, eCos 所提供的支持还包括开发嵌入式应用所需的所有工具, 如 eCos 配置和编译工具、基于 GNU 的编译器、汇编器、链接器、调试器和模拟器。

eCos 提供的基本功能如下:

- 1) 硬件抽象层 (HAL)。
- 2) 实时内核:
 - ① 中断处理。
 - ② 例外处理。
 - ③ 可选择的调度器。
 - ④ 多线程支持。
 - ⑤ 一组丰富的同步原语。
 - ⑥ 定时器、计数器、告警器。
 - ⑦ 内存分配算法选择。
 - ⑧ 调试和测试支持。
- 3) μ ITRON 3.0 兼容 API。
- 4) POSIX 兼容 API。
- 5) ISO C 和数学库。
- 6) 串口、以太网、墙上时钟和看门狗设备驱动程序。
- 7) USB 支持。
- 8) TCP/IP 网络栈, 包括:

- ① BOOTP/DHCP。
- ② DNS。
- ③ TFTP/FTP。
- ④ SNMP。
- ⑤ IPv6。
- ⑥ HTTPD。
- 9) 文件系统:
 - ① JFFS2 Flash 文件系统。
 - ② RAM 文件系统。
 - ③ ROM 文件系统。
- 10) 电源管理。
- 11) GDB debug 支持。

eCos 的主要特性包括:

- ① 开放源代码。
- ② 免费软件,无版权费用。
- ③ 高度可配置性。
- ④ 易于移植。
- ⑤ 实时系统。
- ⑥ 代码量小。
- ⑦ 符合标准协议。
- ⑧ 网络支持。

1.1.3 eCos 核心组件

实时嵌入式操作系统通常应该提供一些标准功能,这些功能包括线程同步机制、调度机制、中断处理、例外和错误处理、定时机制以及设备驱动程序等。eCos 通过提供以实时内核为核心的核心组件来实现这些标准功能。这些核心组件包括:

(1) 硬件抽象层(HAL):对硬件平台进行抽象,为上层软件对硬件的控制和访问提供一个标准接口。硬件抽象层的实现保证了 eCos 系统具有良好的可移植性。

(2) 内核:内核包含了中断和例外处理机制、多线程机制、同步机制、可供选择的多种调度机制、定时机制、计数器等。

(3) ISO C 和数学库:提供与标准兼容的函数和调用。

(4) 设备驱动程序:对一些典型的设备提供驱动程序支持,包括串口驱动程序、以太网驱动程序、Flash ROM 驱动程序、USB 驱动程序、PCMCIA 驱动程序等等。

(5) GNU debugger (GDB)支持:为目标平台上的软件与 GDB 主机之间提供通信机制,实现对目标平台硬件和软件的调试。

eCos 系统及其应用程序以特权方式运行,没有用户方式和内核方式之分。eCos 还提供了一些对系统基本结构进行测试的测试程序,这些测试程序也可以与系统一样进行类似的配置,使其能精确地对系统所进行的配置进行测试。

1.1.4 对硬件的支持

eCos 支持当前流行的大部分嵌入式处理器。eCos 具有很好的可移植特性,它可以在 16 位、32 位和 64 位等不同体系结构之间以及它们的各种不同平台之间进行移植。eCos 的内核、库以及运行组件位于硬件抽象层(HAL)之上,只要将硬件抽象层和相关的设备驱动程序进行移植,eCos 及其应用程序就可以在新的目标平台上运行。因此,厂家在进行产品开发时,在硬件结构的选型方面具有很大的选择余地。

eCos 目前支持十几种处理器,包括这些处理器的多种变体和多种典型开发板。eCos 源码支持当前市场上最为流行的各种不同处理器结构的标准商用评估板。下面是 eCos 目前所支持的主要处理器(对硬件的详细支持情况见“附录 A”):

- ① ARM。
- ② Fujitsu FR-V。
- ③ Hitachi H8/300。
- ④ Intel x86。
- ⑤ Matsushita AM3x。
- ⑥ MIPS。
- ⑦ NEC V8xx。
- ⑧ PowerPC。
- ⑨ Samsung CalmRISC16/32。
- ⑩ SPARC。
- ⑪ SPARClite。
- ⑫ SuperH。

读者可以在 eCos 网站上查阅 <http://sources.redhat.com/ecos/hardware.html>,了解 eCos 对硬件支持的最新情况。

1.1.5 eCos 资源

eCos 是一种开放源代码的软件,它的源代码和开发工具、配置工具都可以从 eCos 网站上免费下载。相关网站和主要网址如下:

eCos 网址:<http://ecos.sourceware.org/>或 <http://sources.redhat.com/ecos>。

RedHat 网址:<http://www.redhat.com>。

eCosCentric 网址:<http://www.ecoscentric.com>。

GNU 网址:<http://www.gnu.org>。

Cygwin 网址:<http://www.cygwin.com>。

eCos 源码下载地址:<ftp://sources.redhat.com/pub/ecos>。

GNU 开发工具包下载地址:<ftp://ftp.gnu.org/gnu/>。

图形配置工具下载地址:<http://sources.redhat.com/ecos/ct2.html>。

eCos 源码采用 CVS 系统进行版本管理,它的 CVS 服务器上具有最新的 eCos 源码。用户可以登录该 CVS 服务器及时更新 eCos 源码。eCos 的 CVS 服务器的登录方式(Linux 或 Cygwin 环境下)为: