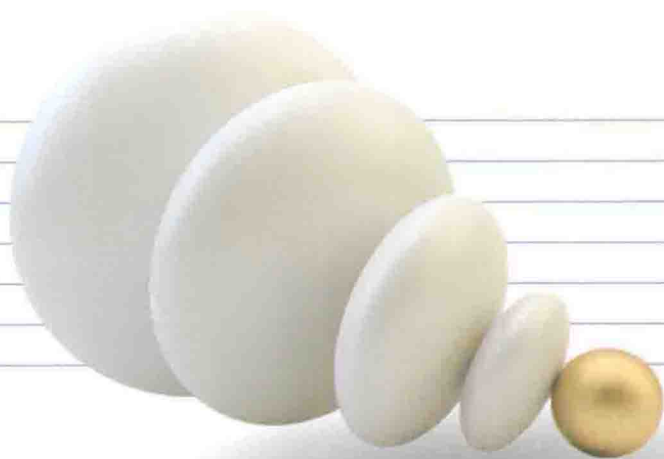


嵌入式

Linux 实战教程

河源职业技术学院
深圳信盈达电子有限公司
周永福 李令伟 邹莉莉 陈志发◎编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

信盈达技术创新系列图书

嵌入式Linux实战教程

河源职业技术学院

深圳信盈达电子有限公司

周永福 李令伟 邹莉莉 陈志发 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以信盈达 xyd2440 开发板为例,从嵌入式 Linux 系统开发必备的 Linux 知识点出发,分析并详细讲解 U-Boot、内核、根文件系统的源码与制作过程,详细介绍 Linux 系统驱动理论,尤其花了大量的精力介绍基于嵌入式 Linux 2.6 内核下的各类驱动设备,包括 LED、按键、触摸屏、LCD 和网卡等各种外设和芯片驱动程序的开发技术与实操项目详解。此外,还介绍了基于 Qt 的 GUI 系统的移植、产品固化代码及发布的方法。

本书从基本的 Linux 系统的操作命令开始,由浅及深地讲解相关内容,使读者循序渐进地掌握嵌入式开发的方法和技巧,最终能够为嵌入式开发板配置、移植、裁剪内核,编写开发驱动程序,以及移植 GUI 系统,从而掌握整个嵌入式 Linux 系统的开发方法。

本书以实操为特色,概念清晰、阐述精辟,对于所有层次的 Linux 程序员都是一本不可或缺的参考书。同时,本书也可作为高等院校相关专业师生的教学参考书,以及各类相关培训班的教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

嵌入式 Linux 实战教程 / 河源职业技术学院等编著. —北京:电子工业出版社, 2014. 6

(信盈达技术创新系列图书)

ISBN 978-7-121-22923-7

I. ①嵌… II. ①河… III. ①Linux 操作系统—程序设计—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2014)第 069226 号

策划编辑:陈晓猛

责任编辑:张 慧

印 刷:北京四季青印刷厂

装 订:北京四季青印刷厂

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

开 本:787×1092 1/16 印张:24.5 字数:627 千字

版 次:2014 年 6 月第 1 版

印 次:2014 年 6 月第 1 次印刷

印 数:3 000 册 定价:59.80 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

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。

服务热线:(010) 88258888。

随着 20 世纪 80 年代末嵌入式系统的出现,嵌入式技术得到了飞速的发展。嵌入式 Linux 是嵌入式系统的重要组成部分,是嵌入式应用与开发的核心,是以 Linux 为基础的嵌入式作业系统。通过对嵌入式 Linux 操作系统进行裁剪和修改,使之能够在嵌入式计算机系统上运行,构成了嵌入式开发的重要工作内容。

嵌入式 Linux 既继承了 Internet 上无限的开放源代码资源,又具有嵌入式操作系统的特性。基于 Linux 进行嵌入式开发,不仅无须缴纳高额的版权费,而且还能获得全世界自由软件开发者提供的支持。另外,嵌入式 Linux 具有性能优异、软件移植容易、代码开放、有许多应用软件支持、应用产品开发周期短、新产品上市迅速、有许多公开的代码可供参考和移植、实时性好、稳定性好、安全性好等特点。基于以上特点和优点,嵌入式 Linux 得到了快速的发展和广泛的应用。

嵌入式 Linux 主要的应用领域有信息家电、PDA、机顶盒、数字电话、应答机、可视电话、数据网络、Ethernet Switches、Router、Bridge、HUB、Remote Access Servers、ATM、Frame Relay、远程通信、医疗电子、交通运输、计算机外设、工业控制、航空航天等。近年来比较热门的 Android 系统,其内核也是基于 Linux 的。

从事嵌入式软件相关工作,不仅需要有一定的硬件知识基础,而且还要有良好的软件编程思想。当进行 Linux 驱动开发时,开发人员还需具有一定的 Linux 基础知识。很多想学习嵌入式 Linux 驱动的人已经具备了一定的 ARM 知识和 C 语言编程基础,却苦于对嵌入式 Linux 不熟悉而无从下手。目前市面上绝大部分嵌入式 Linux 驱动书籍的起点都比较高,实际操作性又不是很强,大多都直接进入了对 Bootloader、内核及系统编程的理论知识讲解,这更使得部分零起点的 Linux 初学者无从下手。因此,我们专门为那些已经有较全面的计算机和 ARM 基础,又希望能够快速进入嵌入式 Linux 驱动开发相关行业的开发人员编写了此书,希望能够帮助他们快速熟悉嵌入式驱动开发。

本书的主要内容

本书以信盈达 xyd2440 开发板为例,从嵌入式 Linux 驱动开发必备的 Linux 知识点出发,分析并详细讲解 U-Boot、内核、根文件系统的源码与制作过程,详细介绍 Linux 驱动理论,尤其花了大量的精力介绍基于嵌入式 Linux 2.6 内核下的各类驱动设备,包括字符设备、网络设备的开发技术与实操项目详解。

本书主要涉及以下内容。

第一部分（第1章）：简述嵌入式操作系统，讲解 Linux 系统的安装及常用命令，以及工具的使用方法（包括 ARM - Linux 交叉编译工具的安装）。

第二部分（第2～4章）：详细讲解 U - Boot 的目录结构、源码、常用环境参数，以及 U - Boot 制作与烧写的具体过程与实际操作步骤，详细分析 Linux 内核的启动过程、源码、目录结构及内核裁剪制作的具体过程与实际操作步骤，介绍根文件系统各目录作用及根文件系统制作的详细步骤。

第三部分（第5～11章）：内容涵盖 Linux 2.6 下的两类驱动设备驱动，包括字符设备驱动、网络设备驱动。

第5章介绍 Linux 驱动的基本概念及模型，以及 Linux 中断与异常的处理体系结构概述。

第6章详细讲解字符设备驱动开发的基本概念与重要数据结构，以及高级字符驱动程序具体的函数操作方法（在配套资料中还详细讲解了单模块、多模块、传参模块的驱动模型，以及模块驱动函数的正确阅读方式）。

第7章介绍 LED 驱动开发。从讲解应用程序、库、内核、驱动程序的关系出发，分析 Linux 软件系统的层次结构，概述 Linux 驱动程序的分类与开发步骤，最后详细地提供运行于 xyd2440 开发板的 LED 驱动程序、LED 驱动程序的 Makefile 和 LED 驱动测试程序，以及如何将这些程序成功运行于 xyd2440 开发板的详细操作步骤。

第8章介绍按键驱动开发。从按键的硬件原理图出发，详细讲解按键驱动中要用到的数据结构、中断处理程序、打开与释放函数、读函数，再提供按键驱动的驱动程序源码、对应的 Makefile 及测试程序源码，最后提供如何将程序运行于 xyd2440 开发板的详细操作步骤。

第9章介绍 LCD 驱动开发。首先从介绍 FrameBuffer 原理、实现机制及相关的重要数据结构出发，然后详细讲解 LCD 驱动的源码及移植方法，最后提供一个最简单的 LCD 测试程序——在 LCD 屏中画一个矩形。同样，也提供如何将程序运行于 xyd2440 开发板的详细操作步骤。

第10章首先介绍输入子系统的组成及相关 API 的使用，然后以输入子系统下的按键驱动和触摸屏驱动为例，说明输入子系统驱动的编写过程，最后讲解 Linux 下 tslib 库的移植和应用。

第11章介绍网络驱动程序的开发。分析 TCP/IP 和 UDP 族，重点讲解 TCP/IP 网络编程的原理、协议结构等，并以 DM9000 网卡作为教学载体进行驱动的编写和移植，分析 DM9000 的驱动源码。

第四部分（第12章）：介绍嵌入式 GUI 的概念，并重点介绍 Qt 的开发优势，详细介绍 Qt 的编译及测试过程。

第五部分（第13～14章）：第13章介绍嵌入式 Linux 产品的封装、发布过程，并介绍

常用文件系统镜像（包括 YAFFS、JFFS2、Cramfs 等文件系统映像）的生成和下载方法。为了给有志于继续学习 S3C6410 开发的人员提供范例，本书第 14 章介绍 S3C6410 开发环境的搭建，并用实例说明将 S3C6410 的 Bootloader 和内核烧写到 Flash 和 SD 卡的操作步骤。

本书的阅读建议

嵌入式的开发与具体的硬件环境紧密相关，作者结合实际培训课程，以比较易学、价格也比较实惠的信盈达 xyd2440 开发板为例，提供基于信盈达 xyd2440 开发板操作的实例参考，使读者更容易学习和上手。本书以实操为特色，希望读者可以动手操作书中安排的每一个实操项目，必能获得快速的提高。

编著者
2014 年 6 月

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，本社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

第 1 章 嵌入式 Linux 开发环境	1
1.1 嵌入式介绍	1
1.1.1 嵌入式系统定义	1
1.1.2 嵌入式操作系统	1
1.1.3 常见的嵌入式操作系统	3
1.2 虚拟机及 Linux 系统安装	5
1.2.1 安装 VMware Workstation 软件	5
1.2.2 在 VMware Workstation 虚拟机安装 Red Hat Linux 系统	6
1.3 Linux 常用命令	13
1.3.1 系统询问命令	13
1.3.2 文件操作命令	14
1.3.3 目录操作命令	15
1.3.4 口令、权限命令	16
1.3.5 文件压缩命令	17
1.3.6 网络相关命令	18
1.3.7 其他命令	18
1.4 Linux 下的编辑工具	19
1.4.1 编辑器分类	19
1.4.2 VIM 的使用	19
1.5 主机开发环境的配置	23
1.5.1 超级终端配置图解	23
1.5.2 Minicom 配置	26
1.5.3 TFTP 服务	28
1.5.4 NFS 服务	29
1.5.5 Telnet 服务	30
1.6 Red Hat 下通过安装镜像文件使用 yum 安装软件	30
1.6.1 修改 yum 配置文件	30
1.6.2 安装卸载软件的方法	30

1.7	Makefile 编写规则	31
1.7.1	Makefile 简介	31
1.7.2	Makefile 的基本结构	31
1.7.3	Makefile 中的变量	33
1.7.4	Makefile 的隐含规则	34
1.8	交叉编译工具基础知识	35
1.8.1	交叉编译工具链的安装	36
1.8.2	arm - linux - gcc 选项	37
1.8.3	arm - linux - ld 选项	38
1.8.4	arm - linux - objcopy	39
1.8.5	arm - linux - objdump	39
1.9	小知识	39
第2章	BootLoader 移植与开发	41
2.1	U - Boot 的目录结构	41
2.1.1	BootLoader 简介	41
2.1.2	常用 BootLoader 介绍	42
2.1.3	U - Boot 1.3.4 目录介绍	43
2.2	U - Boot 的制作过程	44
2.3	U - Boot 运行过程	57
2.3.1	程序启动过程	57
2.3.2	U - Boot 的两个阶段	58
2.4	U - Boot 源码分析	59
2.5	U - Boot 应用：主要用于启动内核进行准备工作	90
2.5.1	U - Boot 的应用	90
2.5.2	BootLoader 与内核的交互	90
2.6	U - Boot 的重要数据结构	90
2.6.1	gd 全局数据变量指针	91
2.6.2	bd 电路板数据指针	91
2.6.3	环境变量指针	92
2.6.4	设备相关设置	92
2.6.5	命令结构体类型定义	93
第3章	Linux 内核裁剪	95
3.1	嵌入式 Linux 内核启动过程	95

3.1.1	Linux 版本及特点	95
3.1.2	内核代码初始化分析	96
3.1.3	setup_arch() 函数分析	99
3.1.4	rest_init() 函数分析	102
3.1.5	挂接根文件系统	104
3.1.6	初始化设备驱动程序	105
3.2	Linux 内核源码目录介绍	108
3.3	Linux 内核配置系统分析	110
3.3.1	内核中的 Kconfig 文件介绍	112
3.3.2	Linux 内核源码 Makefile 文件分析	113
3.4	针对 S3C2440 开发板移植内核的过程	115
3.5	添加驱动程序到内核源码	120
第 4 章	根文件系统移植与制作	122
4.1	分析 Linux 文件系统层次标准	122
4.2	根文件系统各目录作用	122
4.3	移植 Busybox 开源代码、构造根文件系统	123
4.3.1	Busybox 的诞生	123
4.3.2	Busybox 许可证	124
4.3.3	POSIX 环境	124
4.4	制作 Linux 根文件系统	125
4.4.1	下载并配置安装 Busybox	125
4.4.2	完善根文件系统	128
4.4.3	通过 NFS 服务测试文件系统	130
第 5 章	Linux 驱动介绍	131
5.1	驱动原理	131
5.2	内核模块的主要相关命令	131
5.3	设备分类	131
5.4	设备驱动程序工作原理	132
5.5	应用程序、库、内核、驱动程序的软件关系	132
5.6	Linux 驱动程序开发步骤	132
5.7	驱动程序的加载和卸载	133
5.8	关键概念	133
5.8.1	不可剥夺型内核 (non-preemptive kernel) (分时操作系统内核)	133

5.8.2 可剥夺型内核 (preemptive kernel) (实时操作系统内核)	134
5.8.3 可重入函数	134
5.8.4 资源	134
5.8.5 共享资源	134
5.8.6 代码的临界段	134
5.8.7 实时系统的特点	134
5.8.8 死锁	135
5.8.9 Linux 的进程状态描述	135
5.8.10 Linux 内核的三种调度方法	135
5.8.11 所有任务都采用 Linux 分时调度策略时	136
5.8.12 所有任务都采用 FIFO 调度策略时	136
5.8.13 所有任务都采用 RR 调度策略时	137
5.8.14 进程调度依据 (系统进程调度原理)	137
5.8.15 描述进程的数据结构 task_struct (任务结构体)	139
5.8.16 Linux 使用内核函数 goodness() 对进程进行加权处理	139
5.9 中断与异常	141
5.9.1 Linux 异常处理体系结构概述	141
5.9.2 中断注册方法	144
5.10 并发和竞态	144
5.10.1 并发及其管理	145
5.10.2 信号量和互斥体	145
5.10.3 自旋锁	147
第6章 字符设备驱动开发	150
6.1 主设备号和次设备号	150
6.2 重要数据结构	150
6.2.1 file_operations 结构	150
6.2.2 file 结构	154
6.3 字符设备注册方式	154
6.3.1 早期版本的字符设备注册 (2.6 版本以前)	154
6.3.2 杂项 (misc device) 设备注册	155
6.3.3 Linux 2.6 的版本设备注册	155
6.4 打印信息	164
6.5 高级字符驱动程序操作	165

6.5.1	ioctl	165
6.5.2	定位设备 (llseek 实现)	167
6.5.3	休眠	167
6.5.4	阻塞和非阻塞操作	169
6.5.5	poll 和 select	170
6.5.6	与 read 和 write 的交互	171
6.5.7	刷新待处理输出	172
6.6	异步通知	172
6.7	内核同步机制	174
6.7.1	原子操作	174
6.7.2	信号量 (semaphore)	176
6.7.3	自旋锁 (spinlock)	177
第 7 章	LED 驱动开发与应用编程	182
7.1	LED 驱动程序 (xyd2440_leds.c)	182
7.2	LED 驱动程序的 Makefile	185
7.3	LED 驱动测试程序 (leds_app.c)	185
7.4	操作步骤	186
第 8 章	按键驱动开发与应用编程	188
8.1	按键的硬件原理	188
8.2	按键驱动中的数据结构	188
8.3	按键驱动的模块加载函数和卸载函数	189
8.4	按键设备驱动中断处理程序	190
8.5	按键设备驱动的打开函数和释放函数	190
8.6	按键设备驱动的读函数	192
8.7	按键驱动程序范例	193
8.8	操作步骤	199
第 9 章	LCD 驱动开发实例分析	201
9.1	FrameBuffer 的原理	201
9.2	FrameBuffer 在 Linux 中的实现和机制	202
9.3	Linux 内核中的 platform 机制	207
9.4	S3C2440 LCD 驱动分析	209
9.5	S3C2440 LCD 驱动移动移植	228
9.5.1	代码修改	228

9.5.2 修改 Kconfig——添加新屏支持到内核菜单	231
9.5.3 修改内核配置	232
9.6 FrameBuffer 应用例子	234
9.6.1 应用程序源码	234
9.6.2 操作步骤	235
第10章 输入子系统和触摸屏	237
10.1 认识输入子系统	237
10.1.1 为什么需要输入子系统	237
10.1.2 相关 API	237
10.1.3 一个简单的按键驱动的例子	240
10.2 触摸屏驱动分析	241
10.2.1 Linux 内核定时器	241
10.2.2 触摸屏驱动代码分析	243
10.3 触摸屏的校准	249
10.3.1 触摸屏的校准介绍	249
10.3.2 tslib 库的编译使用	250
10.3.3 将 tslib 校准引用到自己的项目中	251
第11章 网络驱动开发与应用编程	253
11.1 TCP/IP 概述	253
11.1.1 TCP/IP 族	254
11.1.2 TCP 和 UDP	254
11.2 TCP/IP 网络基础编程	255
11.2.1 socket 概述	255
11.2.2 地址及顺序处理	256
11.2.3 socket 基础编程	261
11.3 DM9000 网卡驱动程序移植	273
11.3.1 DM9000 网卡特性	273
11.3.2 DM9000 网卡与 S3C2440 硬件连接	274
11.3.3 DM9000 与开发板接口定义	275
11.3.4 DM9000 驱动移植到 S3C2440 的步骤	276
11.4 DM9000 网卡驱动源码解析	278
11.4.1 两个重要的结构体的简单介绍: sk_buff 和 net_device	278
11.4.2 驱动代码具体分析	285

11.4.3 dm9000_netdev_ops 和 dm9000_ethtod_ops	286
11.4.4 平台设备驱动的结构体定义	288
11.4.5 驱动初始化函数	288
11.4.6 probe 函数	289
11.4.7 remove 函数: 设备的删除函数	304
11.4.8 打开、关闭函数和操作函数	305
第 12 章 基于 Qt Creator 环境的 Qt 开发	321
12.1 概要说明	321
12.2 安装步骤	321
12.3 Qt Creator 测试	325
12.4 Qt Creator 中配置 ARM 版本交叉编译环境	328
12.5 交叉编译 ARM 版本 Qt4 程序	332
第 13 章 产品封装、发布	335
13.1 Linux 内核镜像格式	335
13.2 Linux 内核镜像的产生过程	336
13.3 嵌入式 Linux 常用文件系统	336
13.4 制作常用的文件系统镜像	338
13.4.1 制作 Cramfs 文件系统和部署	338
13.4.2 制作 JFFS2 文件系统镜像和部署	339
13.4.3 制作 YAFFS2 文件系统镜像和部署	340
第 14 章 S3C6410 平台 Linux 环境搭建	342
14.1 概述	342
14.2 编译器的安装	342
14.3 编译 U-Boot 和内核	342
14.3.1 编译 U-Boot	342
14.3.2 编译内核	343
14.4 烧写内核到 NAND Flash	344
14.4.1 制作用于一键烧写 Linux 的 SD 卡	344
14.4.2 烧写 Linux 到开发板的 NAND Flash 中	347
14.5 文件程序或文件下载到开发板中	350
14.5.1 通过超级终端下载	350
14.5.2 通过 SD 卡复制	352
附录 A Linux 开发平台硬件介绍	355

A.1 S3C2440 系列芯片介绍	355
A.2 S3C2440 开发板硬件设计说明	357
A.2.1 电源设计	357
A.2.2 复位电路	358
A.2.3 系统时钟电路	358
A.2.4 JTAG 接口电路	358
A.2.5 GPIO 电路、指示灯电路、按键电路	359
A.2.6 蜂鸣器电路	360
A.2.7 串口电路	360
A.2.8 A/D 和 D/A 电路	361
A.2.9 外部 Flash 芯片	362
A.2.10 SD 卡接口电路	362
A.2.11 SDARM 芯片模块	363
A.2.12 NOR Flash 芯片模块	364
A.2.13 NAND Flash 芯片模块	364
A.2.14 USB 接口	365
A.2.15 网络模块	365
A.2.16 音频输入与输出电路	365
A.2.17 LCD 接口和 CMOS 摄像接口	367
A.2.18 系统总线接口	368
A.2.19 开发板完整原理图	368
附录 B JTAG 仿真调试器下载程序的过程	369
附录 C J-Link 仿真调试器的使用	371
附录 D Linux 常用命令介绍	375
参考文献	377

1.1 嵌入式介绍

1.1.1 嵌入式系统定义

目前,对嵌入式系统的定义多种多样,但没有一种定义是最全面的。下面给出两种比较合理的定义。

从技术的角度定义:以应用为中心,以计算机技术为基础,软件硬件可裁剪,适应应用系统对功能、可靠性、成本、体积及功耗的严格要求的专用计算机系统。

从系统的角度定义:嵌入式系统是设计完成复杂功能的硬件和软件,并使其紧密耦合在一起的计算机系统。

定义反映了这些嵌入式系统通常是更大系统中的一个完整的部分。这个更大的系统称为嵌入的系统。在嵌入的系统中可以共存多个嵌入式系统。

1.1.2 嵌入式操作系统

1. 概述

计算机系统由硬件和软件组成,在发展初期没有操作系统这个概念,用户利用监控程序来使用计算机。随着计算机技术的发展,计算机系统的硬件、软件资源也越来越丰富,监控程序已不能适应计算机应用的要求。于是,在 20 世纪 60 年代中期,监控程序又进一步发展形成操作系统(Operating System, OS)。发展到现在,广泛使用的有三种操作系统,即多道批处理操作系统、分时操作系统及实时操作系统,如图 1.1 所示。全世界超过九成的 PC 使用的是微软(Microsoft)公司的 Windows 操作系统,还有一些颇具知名度的操作系统,如苹果(Apple)公司的 Mac OS;工作站级计算机常用的 Sun 公司的 Solaris,还有 Linux 或是 FreeBSD 等免费的操作系统。但是提到嵌入式系统所使用的操作系统,一般用户就了解得很少了。

由于大型嵌入式系统需要完成复杂的功能,所以需要操作系统来完成各个任务之间的调度。由于桌面型操作系统的体积及实时性等特性不能满足嵌入式系统的要求,从而促进了嵌入式操作系统的发展。

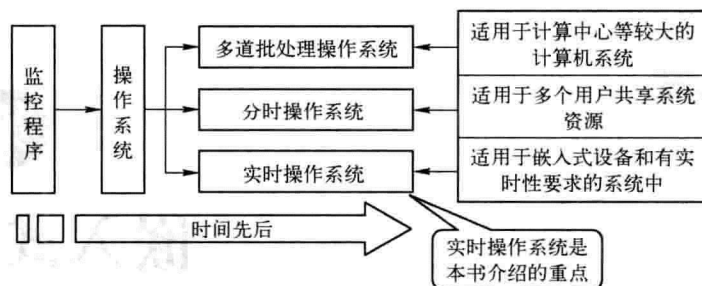


图 1.1 操作系统类别

2. 操作系统

操作系统的基本思想是隐藏底层不同硬件的差异，向在其上运行的应用程序提供一个统一的调用接口。应用程序通过这一接口实现对硬件的使用和控制，不必考虑不同硬件操作方式的差异。计算机系统组成如图 1.2 所示。

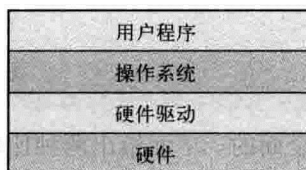


图 1.2 计算机系统组成

很多产品厂商选择购买操作系统，并在此基础上开发自己的应用程序，最终形成产品。事实上，因为嵌入式系统是将所有程序（包括操作系统、驱动程序、应用程序）的程序代码全部烧写进 ROM 里执行，所以操作系统在这里的角色更像是一套函数库（Library）。

操作系统主要完成三项任务：内存管理、多任务管理和外围设备管理。操作系统是计算机中最基本的程序。操作系统负责计算机系统中全部软硬件资源的分配与回收、控制与协调等的并发活动；提供用户接口，使用户获得良好的工作环境；为用户扩展新的系统功能提供软件平台。

嵌入式操作系统（Embedded Operating System）负责嵌入式系统的全部软、硬件资源的分配、调度、控制、协调；必须体现其所在系统的特征，能够通过加载/卸载某些模块来达到系统所要求的功能。

嵌入式系统的操作系统核心通常要求体积要很小，因为硬件 ROM 的容量有限，除了应用程序之外，不希望操作系统占用太大的存储空间。事实上，嵌入式操作系统可以很小，只提供基本的管理功能和调度功能，10KB 到 20KB 以内的嵌入式操作系统比比皆是，相信用惯微软 Windows 系统的用户，可能会觉得不可思议。

不同的应用场合会产生不同特点的嵌入式操作系统，但都会有一个核心（Kernel）和一些系统服务（System Service）。操作系统必须提供一些系统服务供应用程序调用，包括文件系统、内存分配、I/O 存取服务、中断服务、任务（Task）服务、时间（Timer）服务等，设备驱动程序（Device Driver）则是要建立在 I/O 存取和中断服务上的。有些嵌入式操作系统也会提供多种通信协议，以及用户接口函数库等。

嵌入式操作系统的性能通常取决于核心程序，其核心的工作主要是任务管理（Task Management）、任务调度（Task Scheduling）、进程间的通信（IPC）和内存管理（Memory Management）。