

刘彦文 编著

嵌入式系统 实践教程

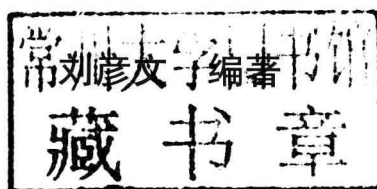
计算机科学与技术专业实践系列教材

清华大学出版社



计算机科学与技术专业实践系列教材

嵌入式系统 实践教程



清华大学出版社
北京

内 容 简 介

本书针对嵌入式系统教学过程中的实验及编程实践环节,详细描述了 Linux 环境下嵌入式开发系统的组成、主机及目标板的运行、主机开发环境配置和目标板出厂软件烧写;给出了实验例程并对驱动程序进行分析,其中包括时间和日期、多线程、串行端口及终端、ADC、DAC、PWM 直流电机、LED 实验例程以及 ADC、DAC、PWM、LED 驱动程序源代码分析。

本书在给出实验例程的同时,还详细描述了相关设备的工作原理或硬件特性、应用程序编程方法及驱动程序含义。书中全部应用程序和驱动程序源代码都加了注释或说明。另外,书中还给出大量的操作实例、编程环境网络通信测试实例,并且全部实验例程均经过上机调试、验证。

本书内容新颖,实用性和实践性强,书中配有大量的图、表、例和程序源代码,每章都附有习题、操作、编程与实验,便于读者学习。

本书可作为高等院校计算机、软件、电子、自动化、通信等专业的本科生《嵌入式系统原理及接口技术》课程的实验或编程实践的配套教材使用,也可单独作为嵌入式系统实践教材使用。本书还可以作为研究生的参考教材;同时可供相关技术人员参考或作为培训教材使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式系统实践教程/刘彦文编著. —北京:清华大学出版社,2013.3

(计算机科学与技术专业实践系列教材)

ISBN 978-7-302-31225-3

I. ①嵌… II. ①刘… III. ①微型计算机—系统设计—高等学校—教材 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2013)第 001712 号

责任编辑:袁勤勇 徐跃进

封面设计:傅瑞学

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:11

字 数:273 千字

版 次:2013 年 3 月第 1 版

印 次:2013 年 3 月第 1 次印刷

印 数:1~3000

定 价:23.00 元

产品编号:050225-01

前 言

本书是作者近年来为本科生、研究生讲授《嵌入式系统原理及接口技术》课程期间,针对本校所使用的教学实验设备,对编写的实验大纲、实验例程及代码注释、与实验例程相关的硬件原理或设备性能资料、与实验例程相关的驱动程序分析等文档和讲稿,经过汇集、整理,重新编写而成的教材。作者使用的嵌入式目标板是国内博创科技公司的产品,微处理器是 S3C2410,开发环境为 Linux。目前该教学实验设备在国内大学使用较为广泛。

本书可作为《嵌入式系统原理及接口技术》课程的实验或编程实践配套教材使用,也可以单独作为嵌入式系统实践教材使用。

本书内容大体上可以分为以下 3 部分:

(1) 嵌入式开发系统的组成、主机及目标板的运行、主机开发环境配置。

(2) 实验例程及驱动程序分析,包括时间和日期、多线程、串行端口及终端、ADC、DAC、PWM 直流电机、LED 例程,以及 ADC、DAC、PWM、LED 驱动程序源代码分析。

(3) 目标板出厂软件烧写实验,包括装载引导程序、内核、根文件系统、应用程序的传送和烧写实验。

本书讲述的实验及例程都是较为基础的部分,适合初次学习嵌入式系统的人员使用。书中在讲述实验例程时,将相关的硬件原理及特性、应用程序编程方法、相关设备驱动程序的含义等内容结合在一起讲述,希望有助于加深读者对所学内容的理解。

本书的实验例程,均给出完整的源程序代码,并且上机调试通过。

书中实验例程及驱动程序源代码,均加了注释或文字说明。

感谢我所在的学院,在教材编写期间提供的编程、调试环境和设备。

特别感谢李惠林女士,在稿件交付出版社前,对全部内容进行了录入、排版和核对;在统稿过程中提出了许多建议和修改意见。

感谢清华大学出版社袁勤勇编辑对本书编写工作提出的建议,以及在编辑过程中付出的辛勤劳动。

由于编者水平有限,书中的错误和不当之处在所难免,敬请专家和读者批评指正。

刘彦文
2012 年 9 月

目 录

第 1 章 Linux 及嵌入式开发系统组成简介	1
1.1 Linux 起源及发展	1
1.1.1 UNIX 和 GNU 软件	1
1.1.2 Linux 的发展	2
1.2 内核、shell、文件系统及实用程序	3
1.2.1 Linux 内核	3
1.2.2 shell 简介	3
1.2.3 文件系统	7
1.2.4 实用程序	7
1.3 选择 Linux 的理由	8
1.3.1 Linux 是成熟的、广泛使用的操作系统	8
1.3.2 开源软件带来的优点	8
1.3.3 免费软件带来的优点	8
1.3.4 Linux 得到了众多软、硬件厂商的支持	9
1.3.5 使用 Linux 的不足之处	9
1.4 Linux 环境嵌入式开发系统组成简介	9
1.4.1 Linux 环境嵌入式开发系统组成	9
1.4.2 主机(Host)	9
1.4.3 目标板(Target)	10
1.4.4 主机与目标板的连接	10
1.4.5 主机中运行的程序	10
1.4.6 目标板中运行的程序	11
1.4.7 开发系统附带的光盘文档	12
1.5 主机安装的 Linux 发行版	12
1.5.1 常用的 Linux 发行版	13
1.5.2 Linux 发行版在主机的安装方式	13
1.6 本章小结	15
1.7 习题、操作与实验	15
第 2 章 主机及目标板的运行	16
2.1 终端基础知识	16
2.1.1 tty	16
2.1.2 早期的终端及终端与主机的连接	16
2.1.3 X-Window、GNOME 及 KDE 简介	17
2.2 主机系统引导、登录及退出 Linux	19

2.2.1	系统引导	19
2.2.2	登录 Linux	19
2.2.3	Linux 的终端	20
2.2.4	查看及修改用户使用的 shell	23
2.2.5	通过退出与 shell 的会话退出 Linux	25
2.2.6	关闭运行的 Linux 系统	25
2.2.7	使用帮助命令 help 和 man	26
2.3	目标板初次运行	28
2.3.1	目标板连接的电源、端口及安装的操作系统	28
2.3.2	目标板引导过程简述	29
2.3.3	主机 Windows 下的超级终端	31
2.3.4	主机 Linux 下的 minicom 仿真终端	33
2.3.5	装载引导程序支持的命令简介	36
2.3.6	目标板 Linux 登录、退出讨论	38
2.3.7	目标板 Linux 基础	39
2.4	本章小结	42
2.5	习题、操作与实验	42
第 3 章	主机开发环境配置	44
3.1	交叉开发平台基础	44
3.1.1	本地开发及交叉开发	44
3.1.2	GNU 跨平台工具链组成	45
3.1.3	如何获取工具链	45
3.1.4	工具链的组件及 GDB 简介	46
3.2	交叉工具链安装举例	47
3.2.1	交叉工具链及实验例程安装位置	47
3.2.2	对查找工具链的路径变量进行修改	48
3.3	主机 Linux 环境网络配置举例	48
3.3.1	主机 Linux 环境 IP 地址设置	48
3.3.2	主机 Linux 环境网络防火墙设置	52
3.3.3	主机与目标板网络通信测试	52
3.3.4	主机 Linux 环境 NFS 设置	54
3.4	输入、编译、运行一个程序全过程举例	57
3.5	本章小结	59
3.6	习题、操作与实验	59
第 4 章	实验例程及驱动程序分析(一)	61
4.1	读取、转换时间函数编程举例	61
4.1.1	Linux 查看、设置时间的命令	61
4.1.2	常用的读取、转换时间的函数	62

4.1.3	读取、转换时间的函数编程举例	63
4.1.4	例 4.1 对应的 Makefile 文件	65
4.2	多线程编程举例	65
4.2.1	Linux 线程概述	65
4.2.2	线程、互斥量、条件变量函数	66
4.2.3	生产者、消费者编程举例	68
4.3	串行端口及标准输入/输出/错误编程举例	73
4.3.1	串行通信基础	73
4.3.2	Linux 标准输入/输出/错误系统调用编程举例	77
4.3.3	Linux 串行端口编程举例	80
4.3.4	C 语言标准输入/输出/错误函数编程举例	83
4.4	ADC 应用程序编程举例	85
4.4.1	ADC 编程基础	85
4.4.2	ADC 应用程序编程举例	87
4.5	ADC 驱动程序源代码分析	91
4.5.1	概述	91
4.5.2	Linux 设备驱动程序概述	92
4.5.3	s3c2410-adc.h 头文件源代码分析	96
4.5.4	s3c2410.h-chip.H 头文件源代码分析	96
4.5.5	s3c2410-adc.c 设备驱动程序源代码分析	98
4.6	本章小结	104
4.7	习题、操作、编程与实验	105
第 5 章	Linux 环境下目标板出厂软件烧写实验	107
5.1	目标板出厂软件烧写概述	107
5.1.1	目标板出厂软件	107
5.1.2	Linux 环境要烧写的文件	107
5.1.3	什么情况下烧写目标板出厂软件	108
5.1.4	烧写环境	108
5.2	目标板出厂软件烧写实验(一)	108
5.2.1	主机 Windows 下安装 JTAG 驱动程序	109
5.2.2	通过 JTAG 端口传送并烧写装载引导程序 vivi	109
5.2.3	通过串口传送并烧写内核 zImage	110
5.2.4	通过串口传送并烧写根文件系统 root.cramfs	112
5.2.5	通过网口传送并烧写应用程序 yaffs.tar.bz2	112
5.2.6	格式化闪存以及通过串口传送并烧写 vivi	116
5.3	目标板出厂软件烧写实验(二)	117
5.3.1	通过网口传送并烧写内核 zImage	117
5.3.2	通过网口传送并烧写根文件系统 root.cramfs	120
5.4	目标板出厂软件烧写实验(三)	121

5.4.1	通过 JTAG 端口传送并烧写装载引导程序 u-boot	121
5.4.2	通过网口传送并烧写内核 uImage	123
5.4.3	通过网口传送并烧写根文件系统 root, cramfs	125
5.4.4	通过网口传送并烧写应用程序 yaffs, tar, bz2	126
5.5	u-boot 装载引导程序支持的命令	126
5.5.1	u-boot 支持的命令集	126
5.5.2	查询 u-boot 命令的参数含义	128
5.6	本章小结	128
5.7	习题、操作与实验	129
第 6 章	实验例程及驱动程序分析(二)	131
6.1	DAC 应用程序编程举例及驱动程序分析	131
6.1.1	DAC 硬件基础	131
6.1.2	DAC 应用程序编程举例	133
6.1.3	DAC 驱动程序主要代码分析	135
6.2	PWM 直流电机应用程序编程举例及驱动程序分析	138
6.2.1	PWM 定时器基础	138
6.2.2	PWM 直流电机	143
6.2.3	编译、加载驱动程序模块	145
6.2.4	PWM 直流电机应用程序编程举例	146
6.2.5	PWM 直流电机驱动程序主要代码分析	148
6.3	LED 应用程序编程举例及驱动程序分析	149
6.3.1	七段 LED 概述	149
6.3.2	8×8 点阵 LED 显示概述	152
6.3.3	CPLD 与七段 LED、8×8 点阵 LED 连接举例	154
6.3.4	七段 LED、8×8 点阵 LED 应用程序编程举例	156
6.3.5	LED 驱动程序主要代码分析	158
6.4	本章小结	160
6.5	习题、操作、实验及编程	161
附录 A	英汉名词术语对照汇总表	162
参考文献		166

第 1 章 Linux 及嵌入式开发系统组成简介

本章主要内容如下:

- (1) Linux 起源及发展, Linux 与 UNIX 和 GNU 的关系;
- (2) Linux 组成, 如内核、shell、文件系统及实用程序等内容;
- (3) 嵌入式系统开发选择 Linux 的理由;
- (4) Linux 环境嵌入式开发系统组成, 包括在主机、目标板中运行的程序;
- (5) 主机中安装的 Linux 发行版及安装方式。

近年来, Linux 操作系统的使用面越来越广, 特别是在嵌入式产品和各种各样的服务器中, 大量地使用了 Linux 操作系统。据国外报道, 每年新开工的众多嵌入式开发项目, 有一半以上使用了 Linux 系统。在服务器市场, Linux 也逐渐成为主流操作系统。

1.1 Linux 起源及发展

Linux 的发展和成功, 主要归功于它之前的 UNIX 操作系统和 GNU 软件。

1.1.1 UNIX 和 GNU 软件

UNIX 最初是由美国电报电话(AT&T)公司下属的贝尔实验室开发的。1969 年, 贝尔实验室的研究人员创建了 UNIX 的第一个版本, 当时称为 UNICS(Uniplexed Operating and Computing System, 单操作和计算机系统), 由这个系统演化为之后通用的 UNIX 操作系统。

UNIX 最初是为 DEC(数字设备公司)PDP-7 小型计算机设计的, 现在已经成为一种广泛使用的多用户、多任务操作系统。其应用范围主要集中于 PC 工作站、服务器和超级计算机领域。

UNIX 主要有两个可用的基础版本, AT&T System V 和 Berkley Software Distribution(BSD), 其他版本都是建立在这两个版本中的一个版本上。两个版本的不同在于某些实现技术的不同。

UNIX 是由开放组织(Open Group)管理的一个商标, 它指的是一种遵守专门规范的操作系统, 这个规范也称为单一 UNIX 规范(Single UNIX Specification, SUS)。这个规范在很大程度上是早期电气和电子工程师协会(Institute of Electrical and Electronics Engineers, IEEE)开发的 IEEE 1003 规范的一个超集。IEEE 1003 规范也称为 POSIX(Portable Operating System Interface for UNIX, UNIX 便携式操作系统接口)。

早期的 UNIX 版本都是具有商业性质的, 如 IBM 公司的 AIX、HP 的 HP-UX 和 Sun 公司的 Solaris, 通常都要支付一定的费用才能获得运行该操作系统的许可。1984 年, 工程师 Richard Stallman 开始着手 GNU 计划, 该计划致力于创建一个类 UNIX(UNIX-like)的、任何人都可以免费发布和使用的、不受 UNIX 名字和源代码私有权限限制的操作系统和开发

环境,开发环境主要包括高质量的编译器、命令行 make 环境、实用程序等。Richard Stallman 现在管理着一个免费的软件基金会(Free Software Foundation,FSF),他和他的支持者创建的许多程序,广泛地应用于商业的和开放源码的 UNIX 版本中。

GNU 代表 GNU's Not Unix,是一个首字母递归的缩略词。GNU 项目已经提供了许多 UNIX 系统上应用程序的仿制品,所有这些程序,统称为 GNU 软件。

1.1.2 Linux 的发展

Linux 是一个可以自由使用、学习、修改、发布的,开源(open source)的,类 UNIX 内核的一种操作系统的内核。Linux 是由芬兰赫尔辛基大学的研究生 Linus Torvalds 在 1991 年开发的。其开发的目的是,保证 Linux 除了包含可以自由发布的代码外,不会集成任何专有代码,也就是人们通常说的源码全部公开,并可以免费使用。Linux 内核遵循 GNU GPL 条款。

Linux 只是一个操作系统的内核,而操作系统的 shell、文件系统和工具软件等程序的编写,主要由因特网上 GNU 支持者、UNIX 程序员完成,并使其逐渐发展成为一个完整的、强大的系统。Linux 的出现,使台式机具备了以前需要花费数千美元购买 UNIX 才能获得的功能。几乎所有为 UNIX 编写的程序,都可以在 Linux 上编译运行;一些专用于 UNIX 商业版本的应用软件,可以不加改变地以二进制可执行文件形式在 Linux 上运行。

Linux 能够发展、强大,在于它的开放使用许可证制度,即允许软件自由发布而不需要任何授权。发布 Linux 的许可证称为 GNU 公共许可证(GNU Public License,GPL),可以在 <http://gnu.org/licenses/licenses.html> 获得。GPL 对终端用户的唯一要求是,对软件所做的任何修改都要与其他人共享。

GPL 最重要的核心条款是:

- 终生有效;
- 允许用户免费运行程序;
- 允许用户学习和修改源代码;
- 允许用户发布源代码或其修改版;
- GPL 软件的传播对象拥有同等权力。

Linux 的发展是许多人共同努力的结果。操作系统内核本身仅仅是一个可用的操作系统的一小部分。Linux 内核以外的大量程序,是由许许多多的编程人员,特别是以前为 UNIX 编程的、GNU 项目的程序员编写,并免费、自由地以源代码形式发布的。

GNU 项目为 Linux 提供了许多 UNIX 系统上应用程序的仿制品,所有这些 GNU 软件,都是在遵守 GPL 条款下发布的,常用的软件有:

- GCC GNU 编译器集,包含 GNU C 编译器。
- G++ C++ 编译器,是 GCC 的一部分。
- GDB 源代码级调试器。
- GNU make UNIX make 免费版。
- bash 命令行解释器,或称外壳,shell 的一种。
- GNU Emacs 文本编译器及环境。

另外还有许多软件包,如电子表格、源代码控制工具、不同语言的编译器和解释器、因特网工具、图形图像处理工具等,都是遵守 GPL 条款的情况下发行的。

1.2 内核、shell、文件系统及实用程序

内核、shell、文件系统及实用程序是构成 Linux 操作系统的主要组件。

1.2.1 Linux 内核

Linux 内核最初是在 Intel 80386 上完成开发的,之后为了移植到摩托罗拉 68000 系列处理器上,Linus Torvalds 重新设计了内核,改进了内核的可移植性。此后 10 年,Linux 内核已经被移植到几乎每一种主要的 32/64 位微处理器上。

可以从多种途径得到不同版本的 Linux 内核,例如从所购买的图书附带的光盘,或者从某些网站下载。Linux 内核的官方主页是 www.kernel.org。通常说的主流源代码(mainline source)或主流内核(mainline kernel),就是指在 kernel.org 网站上能够找到的 Linux 内核源码树。

Linux 内核版本可以分为两大类:一类是专门用于开发的试验版本,另一类是稳定的产品版本。Linux 内核版本号由主版本号、次版本号及序列号组成。次版本号是奇数的为试验版,偶数的为产品版。例如,Linux 2.5.x,次版本号是奇数,是试验版内核;而 Linux 2.6.x,次版本号是偶数,是产品版。

目前顶级 Linux 源码树,由 Linus Torvalds 和 Andrew Morton 共同维护。

内核是 Linux 系统的最底层,内核提供了系统的核心功能,并允许进程以一种有序的、高效的方式访问硬件。内核控制进程、I/O 设备、文件系统的操作,内核还管理着内存。内核支持系统以多用户、多任务模式运行,并支持这些运行模式对资源的有序访问。

能够在一种体系结构处理器(如 Intel x86)上运行的内核,不能在另一种体系结构处理器(如 ARM)上直接运行,必须经过移植,才能在另一种体系结构处理器上运行。内核代码的绝大部分是用标准的 C 语言编写的,仅有很少一部分与处理器相关的代码,是用汇编语言编写的,这给代码移植带来了很大的方便。

在 Linux 操作系统命令提示符下,使用 cat 命令能够查看并显示当前系统的内核版本信息:

```
[root@ vm-dev~]#cat /proc/version
Linux version 2.6.9-42.ELsmp (bhcompile@hs20-bc1-1.build.redhat.com) (gcc version
3.4.6 20060404 (Red Hat 3.4.6-2)) #1 SMP Wed Jul 12 23:27:17 EDT 2006
```

1.2.2 shell 简介

1. shell 是什么

shell 作为用户与 Linux 内核的接口程序,称为外壳。根(root)用户使用的 shell 在内核启动后启动,普通用户使用的 shell,在用户登录进入系统时启动。内核能够接收用户在命

令行提示符后输入的命令和参数。shell 在 Linux 系统中的位置如图 1.1 所示。

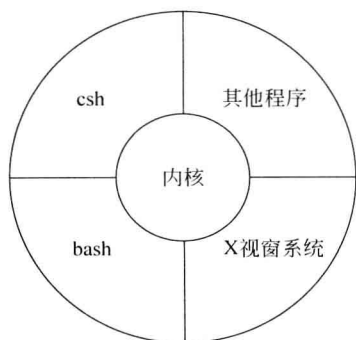


图 1.1 shell 在 Linux 中的位置

在图 1.1 中, bash、csh 是不同种类 shell 中较为常用的两种 shell。

shell 是一个命令行解释器, 也称为命令语言解释器 (command-language interpreter), 它拥有自己内建的 shell 命令集。使用 shell 能够让用户与操作系统交互, 当执行用户命令或者在提示符后进行输入时, 就会与 shell 进行交互。shell 将用户的命令和键盘输入的参数转化成内核能够理解的内容, 内核响应并处理, 处理结果送 shell, 再输出给用户。shell 为系统提供了内核之上的一层功能, 可以直接用它来管理和运行系统, shell 也能被系统中其他实用程序 (utilities) 和应用程序 (application

programs) 所调用。

类似于 MS-DOS 的 command.com 程序, shell 等待并接收提示符后用户输入的命令和参数, 然后解释并传送给内核。有些常用命令, 比如打印当前工作目录命令 pwd, shell 内部包含这些命令对应的程序, 就像 MS-DOS 的内部命令那样, shell 将直接转去执行对应的程序; 另外更多的命令, 比如拷贝 cp 和移动 mv 命令, 类似于 MS-DOS 的外部命令, 命令对应的程序不在 shell 内部, 而是文件系统中某个目录下的单独的可执行文件。当用户在命令行提示符后输入命令和参数, 按下 Enter 键, shell 将寻找命令对应的可执行文件, 然后装入内存并执行; 对于用户编写的应用程序, 以及用 shell 语言编写的脚本程序, 同样可以通过指定文件名, 由 shell 读入并执行。

shell 总是通过调用, 将需要执行的内容传送给内核, 由内核去执行。执行结果由内核传送给 shell。

Linux 根目录下的 bin 子目录, 通常存放着许多系统自带的、用户命令对应的二进制可执行文件。

例如, cp 和 rm 命令对应的文件是:

```
[root@vm-dev bin]#ls /bin/cp -l
-rwxr-xr-x 1 root root 59100 2006-04-11 /bin/cp
[root@vm-dev bin]#ls /bin/rm -l
-rwxr-xr-x 1 root root 34952 2006-04-11 /bin/rm
```

在 /bin 目录下, 还可以看到系统自带的各种版本的 shell 程序文件、用于编辑文件的 vi 程序文件, 以及许多用户命令对应的可执行文件。

Linux 根目录下的/sbin 子目录, 通常存放着许多系统自带的、系统管理员命令对应的二进制可执行文件, 例如, 用于划分物理磁盘的 fdisk 和用于配置网络接口的 ifconfig 命令对应的文件:

```
[root@vm-dev/sbin]#ls /sbin/fdisk -l
-rwxr-xr-x 1 root root 83024 2006-05-25 /sbin/fdisk
[root@vm-dev/sbin]#ls /sbin/ifconfig -l
-rwxr-xr-x 1 root root 66024 2006-06-07 /sbin/ifconfig
```

在 shell 内部,有 shell 自带的命令所对应的程序,可以通过输入 help 命令,查看用户使用的 shell 种类及该 shell 内部自带的命令有哪些:

```
[root@vm-dev sbin]#help
GNU bash, version 3.00.15(1)-release (i386-redhat-linux-gnu)
These shell commands are defined internally.  Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.
```

A star (*) next to a name means that the command is disabled.

```
% [DIGITS|WORD] [&]                (( expression ))
. filename [arguments]              :
[ arg... ]                          [[ expression ]]
alias [-p] [name[=value]...]        bg [job_spec]
bind [-lpvsPVS] [-m keymap] [-f fi break [n]
builtin [shell-builtin [arg...]]    caller [EXPR]
case WORD in [PATTERN [| PATTERN]) cd [-L|-P] [dir]
command [-pVv] command [arg...]     compgen [-abcdefgjkusv] [-o option
complete [-abcdefgjkusv] [-pr] [-o continue [n]
declare [-afFirtx] [-p] [name[=val dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [jobspec...]      echo [-neE] [arg...]
enable [-pnds] [-a] [-f filename]   eval [arg...]
exec [-cl] [-a name] file [redirec exit [n]
export [-nf] [name[=value]...] or false
fc [-e ename] [-nlr] [first] [last fg [job_spec]
for NAME [in WORDS...;] do COMMA for (( exp1; exp2; exp3 )); do COM
function NAME { COMMANDS ; } or NA getopts optstring name [arg]
hash [-lr] [-p pathname] [-dt] [na help [-s] [pattern...]]
history [-c] [-d offset] [n] or hi if COMMANDS; then COMMANDS; [ elif
jobs [-lnprs] [jobspec...] or job kill [-s sigspec|-n signal|-si
let arg [arg...]                    local name[=value]...
logout                               popd [+N|-N] [-n]
printf format [arguments]           pushd [dir |+N|-N] [-n]
pwd [-PL]                           read [-ers] [-u fd] [-t timeout] [
readonly [-af] [name[=value]...]    return [n]
select NAME [in WORDS...;] do CO set [--abefhkmnptuvxBCHP] [-o opti
shift [n]                           shopt [-pqsu] [-o long-option] opt
source filename [arguments]         suspend [-f]
test [expr]                         time [-p] PIPELINE
times                               trap [-lp] [arg signal_spec...]
true                                type [-afptP] name [name...]
typeset [-afFirtx] [-p] name[=valu ulimit [-SHacdflnmpstuv] [limit]
```



```
umask [-p] [-S] [mode]                unalias [-a] name [name...]
unset [-f] [-v] [name...]              until COMMANDS; do COMMANDS; done
variables -Some variable names an wait [n]
while COMMANDS; do COMMANDS; done { COMMANDS ; }
```

以上列出的所有命令,都是 GNU bash shell 自带的命令。

shell 还有一个重要的特性是,它本身就是一种解释型的程序设计语言,它支持在高级语言中使用的大多数程序控制结构及函数、变量和数组等。这些控制结构及函数、变量和数组等,和那些能够在提示符下输入的命令和参数,允许存放在一个称为脚本的可执行的 shell 程序文件中。

2. shell 启动

根用户使用的 shell,在系统启动时,使用系统指定的 shell,自动启动。普通用户在登录进入系统时,系统会启动指定种类的 shell。所有用户在系统运行过程中,可以改变当前使用的 shell,成为另一种 shell,例如登录时使用的 shell 种类为 bash,用户可以改为 tcsh。Linux 是一个多用户系统,系统中每个用户可以使用一个与其他用户不同的 shell。只有当用户退出 Linux 系统时,shell 运行才结束。

3. 常用的 shell

Linux 和 UNIX 系统里有许多种类不同的 shell 可以使用,例如 bash、sh、tcsh、csh、ash、bsh、ksh、zsh 等,它们中的大多数都是从 sh shell 演变而来的。最常用的 3 种 shell 是 bash、csh 和 ksh。

Bourne shell 也直接称为 shell,用 sh 调用,是 UNIX 的第一个 shell,也是 Linux 上使用较为广泛的一个 shell。bash shell 最初是为 GNU 操作系统编写的 shell 环境,开发人员构建了一个模拟 sh 的脚本编程功能的 shell,即 Bourne Again shell,用 bash 调用。bash 经过多年的发展,目前已经增加了许多特性,如用户界面工具和更先进的脚本编程语言,这是 Bourne shell 中所不存在的。目前 bash 是 Linux 中应用最为广泛的一种 shell。

有些系统可能运行的是 ash shell,而不是 Bourne shell。ash 最初是由 NetBSD 开发小组创建的,它是一个小尺寸的 Bourne 克隆版本,占用的外存(磁盘或闪存)以及内存空间,比 Bourne shell 少很多。这对于外存和内存空间比较小的嵌入式系统开发板,更适合使用。它虽然缺少一些标准的 Bourne 性能,例如最近所执行命令的历史记录文件,但它是一个功能完整的 shell 环境。

C shell 通常指 csh 和 tcsh,zsh 也是它的一个变体,C shell 中使用的编程语法派生于 C 语言。使用 C 语言编程的用户,学习、理解和使用 csh 和 tcsh 会觉得更习惯和容易一些。tcsh 比 csh 增加了一些细微的功能,如 tcsh 可以使用键盘上的光标上下移动键,在命令历史记录中前后移动;允许命令行编辑及自动补齐(command completion)等操作。

Korn shell 是由 David Korn 创建的,它集合了 C shell 和 Bourne shell 的优点,并且与 Bourne shell 完全兼容。Korn shell 用 ksh 调用。

Linux 系统中,默认安装的总是 GNU 工具集中的 bash(GNU Bourne Again shell),它是开源的并且可以被移植到几乎所有的类 UNIX 系统上。在本书后续章节,主机 Linux 默认使用的 shell 均指 bash。可以使用如下命令查看用户主机使用的 Linux 中有哪些 shell,

shell 的链接及当前使用的 shell 的版本号。

```
[root@vm-dev bin]# ls * sh
ash  bash  bsh  csh  ksh  sh  tcsh  zsh

[root@vm-dev bin]# ls -l * sh
-rwxr-xr-x 1 root root 98356 2004-06-16 ash
-rwxr-xr-x 1 root root 616248 2006-04-13 bash
lrwxrwxrwx 1 root root 3 2008-03-30 bsh ->ash
lrwxrwxrwx 1 root root 4 2008-03-30 csh ->tcsh
-rwxr-xr-x 1 root root 183492 2005-08-08 ksh
lrwxrwxrwx 1 root root 4 2008-03-30 sh ->bash
-rwxr-xr-x 1 root root 307488 2004-09-22 tcsh
-rwxr-xr-x 1 root root 453836 2005-06-28 zsh

[root@vm-dev bin]# /bin/bash --version
GNU bash, version 3.00.15(1)-release (i386-redhat-linux-gnu)
Copyright (C) 2004 Free Software Foundation, Inc.
```

以上输入的命令对应的显示内容,使用的环境是 Red Hat Enterprise Linux 4.0 版。

Linux 中,不同种类的 shell 对应的文件,都安装在/bin 目录;上述代码中 sh->bash 表示,sh shell 对应的文件,链接到 bash 文件。如果在命令提示符后输入 sh 并回车,装入并运行的是 bash 对应的文件。

1.2.3 文件系统

文件系统是 Linux 系统的一个组件,它能够让用户以统一的方式查看、组织以及保护存储设备上的文件和目录,并与其进行交互。

Linux 环境中的文件,具有特别重要的意义,因为它们为操作系统服务并为设备操作提供了一个简单而一致的接口。在 Linux 中,一切都可以看作文件,包括硬件设备在内。

1.2.4 实用程序

实用程序(utilities)也是 Linux 系统的一个组件。实用程序是一些可执行的程序,用来帮助用户在系统上进行工作。例如,用于 Internet 定位的 Web 浏览器、文字处理程序、编辑程序、E-mail 程序,以及除了 shell 内部命令外,其他 Linux 命令对应的程序,都可以称为实用程序。

实用程序是一些应用程序(application programs)。实用程序这一术语,更多地用于描述运行在 PC 或笔记本电脑上的 Linux 的组件。在描述目标板上的 Linux 的组件时,更经常称其为应用程序。例如在 5.2.5 节中讲述的“通过网口传送并烧写应用程序 yaffs.tar.bz2”,yaffs.tar.bz2 文件是一个应用程序压缩包,从 PC 传送到目标板后,解压的子目录和文件保存在目标板的闪存。子目录包括\ad、\bluetooth、\can、\da、\dc-motor 等,子目录下的文件有模块(驱动程序)和测试程序等。

1.3 选择 Linux 的理由

开发嵌入式产品选择 Linux 作为操作系统有以下理由。

1.3.1 Linux 是成熟的、广泛使用的操作系统

Linux 已经在以下领域或方面广泛使用：

- 全球范围的公共电话交换网络设备；
- 全球数据网络设备；
- 无线移动手持设备；
- 手机及无线通信网络设备；
- 路由器产品；
- 数字电视机顶盒及 DVD 播放器；
- 打印机、游戏机、PDA、数码相机、摄像机。

还有许许多多的嵌入式产品，使用了 Linux 操作系统。大量的使用实例证明，Linux 是一个成熟、高性能且稳定的产品。

Linux 适应性比较强，不仅可以在小型嵌入式产品中使用，还可以在大型、重型、电信级交换网络和路由器中使用。

1.3.2 开源软件带来的优点

Linux 操作系统及运行在其上的各种工具软件，源代码都是开放的。

开源软件带来如下优点：

(1) 一些对安全敏感的产品，比如用于航天、军事、银行等方面的产品，产品开发者更希望使用操作系统本身的源代码是可见的，而不希望使用由二进制可执行文件组成的操作系统。

(2) 使用开源的操作系统，如果需要，开发者可以查阅操作系统的某一段代码，理解它的含义及产生的行为。这样做的好处是，对于开发过程中产生的错误，容易区分是系统软件引起的，还是应用软件引起的。

(3) 使用开源软件，便于将 Linux 操作系统从一种处理器体系结构，移植到另一种处理器体系结构。

1.3.3 免费软件带来的优点

Linux 操作系统及运行在其上的各种工具软件，通常都是免费的。

嵌入式系统使用免费的操作系统和工具软件，会使最终产品成本下降。

免费 Linux 及工具软件与传统商业性质操作系统及工具软件相比，可以节约以下费用：

- 节省开发期间购买操作系统的费用；
- 节省开发期间购买工具软件的费用；
- 节省在最终产品上安装或烧写操作系统时，商业操作系统厂商收取的每个设备的版税。

免费的 Linux 不仅是嵌入式系统开发商首选的操作系统,而且是大学、培训机构和研究单位优先考虑选择的操作系统。

1.3.4 Linux 得到了众多软、硬件厂商的支持

目前常见的 32 位以上、带 MMU 的嵌入式微处理器,几乎都得到 Linux 的支持。

通常处理器厂商开发出一种微处理器时,同时要设计对应的开发板,也称评估板、目标板;然后设计出装载引导程序;将 Linux 移植到该体系结构的微处理器和开发板上,并编写出相应的驱动程序(这个过程也可能有其他公司或人员参与编程)。此外,通常还要在 Linux 环境下,编写多个应用程序,告诉使用这款微处理器和开发板的人员,如何编程使用硬件和软件资源。Linux 事实上已经成为测试微处理器(包括微处理器芯片内部的外设和接口),以及开发板上外设的一种操作系统。

目前 Linux 已经移植到几十种不同体系结构的处理器上;支持 200 多种不同的评估板;支持 200 多种网络适配器;支持 5 家厂商的闪存和 10 种以上的 USB 存储设备及各种各样的众多外设。

1.3.5 使用 Linux 的不足之处

使用 Linux 的不足之处,主要有以下几点:

(1) 在 Linux 环境下编程的人员,比在 Windows 环境下编程的人员要少,因此遇到问题讨论、交流的对象比较少。另外 Linux 的书籍和资料,比 Windows 的也少一些。

(2) Linux 是一个免费软件,因此不像商业软件那样对操作系统本身存在的问题,能够快速处理和解决。

(3) Linux 在图形界面支持方面,比 Windows 还是有一定差距的。

虽然 Linux 有以上不足之处,但是嵌入式产品开发所使用的操作系统,Linux 还是占到半数以上。

1.4 Linux 环境嵌入式开发系统组成简介

用户购买的嵌入式开发系统,通常由目标板、三条电缆、与目标板配套使用的电源适配器和随机附带的光盘软件组成。买来的嵌入式开发系统,可能会支持多种开发环境,比如 Windows CE、 μ C/OS 及 Linux。用户可以选择使用其中的一种开发环境。

以下描述的是 Linux 环境嵌入式开发系统的组成。

1.4.1 Linux 环境嵌入式开发系统组成

如图 1.2 所示,Linux 环境嵌入式开发系统由主机、目标板和相关软件组成。

1.4.2 主机(Host)

图 1.2 中主机在不同的资料中也称为宿主机、Linux 宿主机、Linux 工作站、桌面机,本书统一称为主机。主机可以是基于 80x86 的普通台式 PC 或笔记本电脑,但是普通笔记本电脑通常不带串口,可以选用与串口电缆连接在一起的 USB 口转串口的部件。当然主机也