

工程师经验手记

嵌入式Linux开发技术

主 编 孙天泽

副主编 钟伟 蒙洋



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

工程师经验手记

嵌入式 Linux 开发技术

主 编 孙天泽

副主编 钟 伟 蒙 洋

北京航空航天大学出版社

内 容 简 介

本书共分 10 章,全面介绍了嵌入式 Linux 开发过程中涉及的知识点。其中,前 3 章是基础部分,分别介绍了 Linux 系统基础知识和嵌入式系统开发需要的环境。第 4~7 章介绍了嵌入式 Linux 系统开发的内容,包括引导启动代码、Linux 内核移植以及 Linux 驱动开发的内容。第 8~10 章介绍了应用层面的开发内容,包括多进程的控制和通信、多线程的控制与编程以及程序调试的相关内容。

本书可作为高等院校电子类、电气类、控制类专业高年级本科生、研究生学习嵌入式 Linux 的教材,也可供希望转入嵌入式领域的科研和工程技术人员参考使用,还可作为嵌入式培训班的教材和参考书。

图书在版编目(CIP)数据

嵌入式 Linux 开发技术 / 孙天泽主编. —北京:北京航空航天大学出版社,2011.4
ISBN 978-7-5124-0367-3

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

中国版本图书馆 CIP 数据核字(2011)第 038219 号

版权所有,侵权必究。

嵌入式 Linux 开发技术

主 编 孙天泽

副主编 钟 伟 蒙 洋

责任编辑 董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×960 1/16 印张:20.25 字数:454 千字

2011 年 4 月第 1 版 2011 年 4 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-0367-3 定价:38.00 元

绪 论

嵌入式技术是 21 世纪最有生命力的新技术之一,目前已经广泛应用于社会生活的各个方面。随着国内外嵌入式产品的进一步开发和推广,嵌入式技术越来越和人们的生活紧密相关,并已经广泛地渗透到科学研究、工程设计、军事技术、各类产业和商业文化艺术、娱乐业以及人们的日常生活等方面中。特别是移动网络互联的日益普及,使得嵌入式与互联网成为最为热门的应用领域。

近年来,嵌入式系统不断蓬勃发展,由于嵌入式 Linux 在价格上具有很大优势,其资源丰富而且内核可裁减配置,Linux 系统为众多厂商所青睐。特别是随着 Google、Intel 等国际领先的公司纷纷推出基于 Linux 系统的解决方案,更肯定了 Linux 在嵌入式系统的前景。在移动开发、消费类电子,汽车电子等行业,Linux 具有非常强劲的实力。

本书内容

本书以 ARM 开发板为例,介绍了基于 Linux 操作系统的嵌入式系统开发流程。为了便于初学者学习,本书立足于将复杂的嵌入式系统开发过程模块化,目标是在每一个章节中讲授嵌入式 Linux 开发中的一个知识点以及理解这个知识点的相关概念;由于每部分的内容都比较多,所以有些知识点列出表格,方便读者归纳和总结。本书按照嵌入式 Linux 系统开发的整个过程进行组织,开发中的每个环节都独立成章,共分 10 章,具体内容安排如下:

第 1 章 操作系统的基础知识,包括 Linux 系统目录结构和基本命令等内容。为了使初学者能够尽快掌握 Linux 系统的使用,本章用了较大篇幅详细介绍 Linux 系统的大部分命令,以便读者在 Linux 系统中查询。由于本书介绍的是嵌入式 Linux 开发,因此 Linux 系统使用是后续章节学习的基础,需要读者掌握。

第 2 章 系统任务自动化。本章介绍了实现系统任务自动化的一些技能,包括 shell 编程和流编辑器 sed 的使用。其中,shell 编程是 Linux 系统管理员以及开发人员必备的知识。本章还介绍了 Linux 系统初始化的过程,分析了其中的几个重要配置文件,这些内容与后面讲到的制作根文件系统相互呼应,也是嵌入式 Linux 开发人员必须掌握的内容。

第 3 章 工具链。本章主要介绍如何构建工具链。和传统的 PC 桌面开发相比,Linux 系统开发缺乏界面友好的 IDE 开发环境,特别是对于需要交叉编译的嵌入式系统来说,正确安装一套开发工具是项目开发的首要任务。同时,在构建交叉工具链的过程中,也有助于读者进一步熟悉 Linux 下的一些工具使用。

第4章 构建主机开发环境。本章介绍如何构建嵌入式 Linux 开发环境,介绍串口控制台工具的配置、使用以及几个 Linux 系统服务的配置。tftp 服务和 NFS 服务可以使嵌入式系统的开发效率得到很大提高。在本章最后以具体开发板为例,讲解向开发板下载程序的方法。

第5章 引导启动代码。引导启动代码(Bootloader)是嵌入式开发的一个重点,本章首先介绍了引导启动代码的概念及作用,并以 U-Boot 工程为例,分析了其代码结构。最后,详细介绍了移植 U-Boot 的整个过程。对于本章的学习建议是先熟悉 ARM 体系结构和 ARM 编程。

第6章 Linux 内核概述与移植。本章介绍 Linux 内核的相关知识。随着 Linux 2.6 内核加入对三星系列处理器的支持,移植工作变得简单很多,而本章则以介绍移植 Linux 系统的关键步骤和思路为主。同时,本章还重点介绍了 Linux 内核的启动流程,协助读者开始 Linux 内核代码的研究。

第7章 设备驱动开发。本章介绍 Linux 系统设备驱动开发的有关内容,以一个非常简单的字符设备为例,介绍 Linux 2.6 内核下驱动程序开发方法。限于篇幅且由于大部分嵌入式设备是字符设备,本章没有介绍 Linux 系统中其他两种设备类型:块设备和网络设备。学完本章的内容后,读者能对 Linux 字符设备驱动程序框架和开发流程有清晰的印象。其实,驱动开发是非常复杂的一个主题,用整本书的篇幅都不能完全阐述其中的奥秘。因此,本章只是传授了 Linux 内核编程的思想和字符设备开发的大体框架。

第8章 嵌入式 Linux 应用程序开发—多进程。本章主要讲述 Linux 系统下的多进程开发,其内容属于上层开发,相对来说要容易一些,但却是嵌入式产品开发的一个重要环节。本章首先介绍了进程的基本内容,如进程环境、进程创建等。随后介绍了 Linux 系统的几种进程间通信方式及编程方法,包括管道、消息队列、信号量,最后还介绍了信号机制。

第9章 嵌入式 Linux 应用程序开发—多线程。本章仍然讲述 Linux 上层开发内容,包括线程的概念、线程的控制等;特别是对线程同步技术做了介绍。

第10章 嵌入式 Linux 调试。本章的主题是 Linux 开发中的调试技术,重点介绍了 GDB 工具的使用方法。通过简单的示例阐述了调试的基本流程。在介绍了 GDB 的本地调试以后,本章还介绍了使用 GDB 进行远程调试的方法,包括编译和使用 arm-linux-gdb 的过程。通过本章的学习,读者应该具备嵌入式 Linux 调试的基本技能。

读者对象

本书是一本介绍嵌入式系统开发的书籍,重点以 Linux 操作系统为例,介绍了嵌入式 Linux 系统开发以及应用开发的内容。本书定位的读者群为嵌入式及 Linux 开发初学者,例如:

- ▶ 想学习或者刚刚进入嵌入式系统领域的开发人员;
- ▶ 对嵌入式系统或者 Linux 系统非常有兴趣的人员;
- ▶ 具有单片机开发经验,想从事嵌入式 Linux 开发的工程师。

致 谢

全书由孙天泽统稿并组织完成。钟伟、蒙洋、袁文菊、游成伟、易松华、唐坤勇、田贺祥、黄真、贾佳参与完成了其中部分章节的编写和代码调试工作,以及资料收集、整理工作。本书在编写过程中得到了许多朋友和专家的帮助,在此特别感谢他们的辛勤工作。

此外,还要感谢北京中芯优电科技信息有限公司的无私帮助。本书的前期组织和后期审校工作都凝聚了他们的支持,他们认真阅读了书稿,提出了大量中肯的建议,并帮助纠正了书稿中的很多错误。

由于篇幅所限、时间仓促,加之作者的编写水平有限,书中的不妥之处在所难免,恳请读者批评指正。读者可以发送电子邮件到 xdhydc5@sina.com,与本书策划编辑进一步沟通。

编 者
2011 年 2 月

目 录

第 1 章 Linux 操作系统基础	1
1.1 Linux 与嵌入式 Linux	1
1.2 Linux 系统的目录结构	5
1.3 Linux 的常用命令	11
1.3.1 Linux 系统必备命令	11
1.3.2 /bin 目录下的命令	11
1.3.3 /sbin 目录下的命令	15
第 2 章 系统任务自动化	21
2.1 理解 shell 脚本	21
2.1.1 创建第一个脚本	22
2.1.2 重定向和管道	23
2.1.3 环境变量	24
2.1.4 shell 编程基本元素	27
2.1.5 shell 脚本实例	30
2.2 流编辑器-sed	31
2.2.1 sed 选项	31
2.2.2 sed 使用实例	32
2.3 Linux 系统初始化	34
第 3 章 工具链	39
3.1 GNU Tools 简介	39
3.1.1 binutils	40
3.1.2 GCC 编译器	41
3.1.3 Glibc	42
3.2 ARM Linux 交叉编译工具链的构建	45
3.2.1 创建编译环境	46
3.2.2 准备内核头文件	48
3.2.3 编译 binutils	49
3.2.4 编译 Bootstrap GCC	50

3.2.5	编译 Glibc	51
3.2.6	编译完全版 GCC	52
3.2.7	编译 GDB	52
3.3	获得工具链的其他方式	53
3.3.1	crosstool	54
3.3.2	Buildroot	56
3.3.3	ELDK	59
第 4 章	构建主机开发环境	63
4.1	串口控制台工具	63
4.2	Linux 系统服务配置	66
4.2.1	配置网络地址	66
4.2.2	配置 TFTP 服务	67
4.2.3	配置 NFS 服务	68
4.2.4	BOOTP/DHCP 服务	70
4.3	玩转你的开发板	71
第 5 章	引导启动代码	82
5.1	什么是 Bootloader	82
5.1.1	Bootloader 的功能	83
5.1.2	GRUB 实例	84
5.1.3	链接器命令脚本	87
5.2	U-Boot 介绍	87
5.2.1	U-Boot 的目录结构	88
5.2.2	编译 U-Boot	88
5.2.3	U-Boot 中 .lds 连接脚本文件	89
5.3	U-Boot 移植	90
5.4	为 U-Boot 添加新命令	106
第 6 章	Linux 内核概述与移植	109
6.1	Linux 内核目录结构	109
6.2	Linux 内核的体系结构	110
6.3	内核启动步骤及代码分析	112
6.3.1	引导过程概述	112
6.3.2	压缩内核的启动	113
6.3.3	Linux 在 ARM 中的启动流程	125
6.4	从“零”开始移植内核	135
6.4.1	驱动程序的配置与移植	149



6.4.2 保存内核配置选项	158
第7章 设备驱动开发	160
7.1 理解 Linux 模块编程	160
7.1.1 创建第一个模块程序	160
7.1.2 内核模块的编译与使用	163
7.1.3 模块参数	166
7.1.4 模块符号导出	168
7.2 理解 Linux 的设备驱动程序	170
7.2.1 字符设备	171
7.2.2 块设备	172
7.2.3 简单的字符设备驱动程序实例	173
7.2.4 深入学习设备驱动	184
7.3 Linux 驱动开发中的并发控制	195
7.3.1 信号量	195
7.3.2 自旋锁	199
第8章 嵌入式 Linux 应用程序开发——多进程	201
8.1 进程环境	201
8.1.1 从 main 函数说起	201
8.1.2 清理函数 atexit	203
8.2 进程控制	205
8.2.1 进程创建	205
8.2.2 exec 函数族	207
8.2.3 进程终止	209
8.2.4 进程退出的同步	211
8.3 进程间通信	217
8.3.1 概 述	217
8.3.2 管道 PIPE	219
8.3.3 有名管道 FIFO	223
8.3.4 IPC 综述	228
8.3.5 消息队列	230
8.3.6 共享内存	236
8.3.7 信号量	238
8.4 信号机制	245
8.4.1 概 述	245
8.4.2 信号的发送与捕捉	248

8.4.3 信号的处理	252
8.5 小 结	260
第 9 章 嵌入式 Linux 应用程序开发——多线程	261
9.1 线程概述	261
9.2 线程控制	262
9.2.1 线程创建	262
9.2.2 线程的 Linux 实现	264
9.2.3 有关线程退出	265
9.2.4 辅助函数	269
9.3 线程同步	271
9.3.1 概 述	271
9.3.2 互斥锁	272
9.3.3 条件变量	278
9.3.4 线程与信号量	283
9.3.5 线程取消	286
9.3.6 线程的私有数据	290
9.4 小 结	294
第 10 章 嵌入式 Linux 调试	295
10.1 GDB 的基本使用	295
10.1.1 GDB 的功能	295
10.1.2 调试基本流程	296
10.2 GDB 常用命令	302
10.2.1 工作环境命令	304
10.2.2 设置断点与恢复命令	304
10.2.3 源码查看命令	305
10.2.4 查看运行数据命令	306
10.2.5 修改运行参数命令	306
10.2.6 堆栈管理	307
10.3 GDB 远程调试	307
10.3.1 制作交叉 GDB	308
10.3.2 使用交叉 GDB 调试	311
10.4 小 结	313
参考文献	314

第 1 章

Linux 操作系统基础

知识点：

Linux 与嵌入式 Linux；

Linux 系统的目录结构；

Linux 的常用命令。

1.1 Linux 与嵌入式 Linux

1. Linux

Linux 与 UNIX 操作系统密切相关。所以在开始学习 Linux 以前,有必要知道 UNIX 和 Linux 的一些历史。

UNIX 的第一个版本是用汇编程序编写的。为了使 UNIX 在将来的发展中独立于机器,1971 年又使用由 Dennis Ritchie 开发的编程 C 语言重新编写了 UNIX。由于 Bell 实验室以很低的价格向大学提供 UNIX 的文档和源码,所以该系统的普及相对较快。加之该系统操作简单、并且相对可移植,所以许多用户和公司开始积极踊跃地加入到开发行列中,因此 UNIX 在短时间内就新增了许多功能且日趋成熟。与此同时,还产生了一系列与 UNIX 有关的商业产品,包括 IBM、DEC、BSD UNIX、HP 的版本。

1983 年,AT&T 开始通过其姊妹公司 USL 将 UNIX System V 投放商业市场,并宣布将 System V 用作 UNIX 标准。这也导致了与 BSD 的长期法律纠纷,并产生了其他一些影响。

目前,可以将现代的 UNIX 操作系统划分 System V 或 BSD 类型,如图 1.1 所示。

Linux 也是一个具有完整功能的 UNIX 操作系统,并试图把 UNIX 各版本的优势集于一身,包括一整套 UNIX shell,如 BASH、TCSH 和 Z-shell。熟悉这些 UNIX 接口的用户可以很

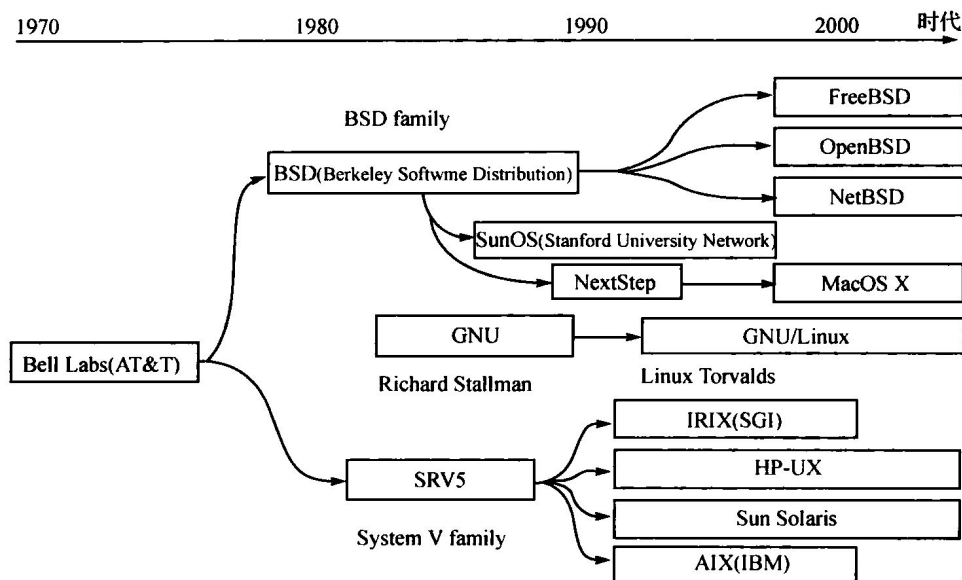


图 1.1 UNIX 操作系统家族

容易地使用任何一种 shell,因为它和 UNIX 的命令、过滤器和配置方法是一样的。因为 Linux 是用 C 语言编写的,所以它适用于许多不同的硬件平台,包括:

- Intel/AMD: 32 位/64 位;
- PowerPC(Macintosh,RS/6000);
- SPARC(Sun);
- IBM pSeries;
- IBM zSeries(S/390);
- 嵌入式系统。

Linux 最初由芬兰大学生 Linus Torvalds 在 1991 年编写,并发布于 FTP 服务器上,用户可以免费下载。正是有了众多开发人员的加入,使得 Linux 在短短几年的时间内发展为世界闻名的操作系统,并广泛应用于各重要领域。在嵌入式系统领域,据 VDC 调查显示 Linux 继续领跑嵌入式操作系统市场。有 18% 的工程师在使用 Linux 进行相关的嵌入式开发,占有率第一。Linux 广受欢迎的原因包括以下几方面:

- 在操作系统许可证费用方面的优势;
- 优良的代码可扩展性;
- 用户对此系统的熟悉程度;
- 操作系统的成熟和丰富的应用程序;
- 开发者对嵌入式 Linux 开发经验的日益丰富;
- 日益统一的 Linux 标准。

Linux 的发行版本(distribution)可以分为两类,一类是商业公司维护的发行版本,一类是社区组织维护的发行版本,前者以著名的 Red Hat Linux 为代表,后者以 Debian 为代表。现在比较著名的发行版本包括: Redhat, Fedora, Ubuntu, SuSE, Mandriva Linux, KNOPPIX, Slackware Linux 等。图 1.2 是 Linux 发行版的家族谱,读者可以从中知道各发行版的衍生。如果你对世界上有哪些发行版本感兴趣,可以访问下面的网站以获得详细信息: <http://distrowatch.com/>。

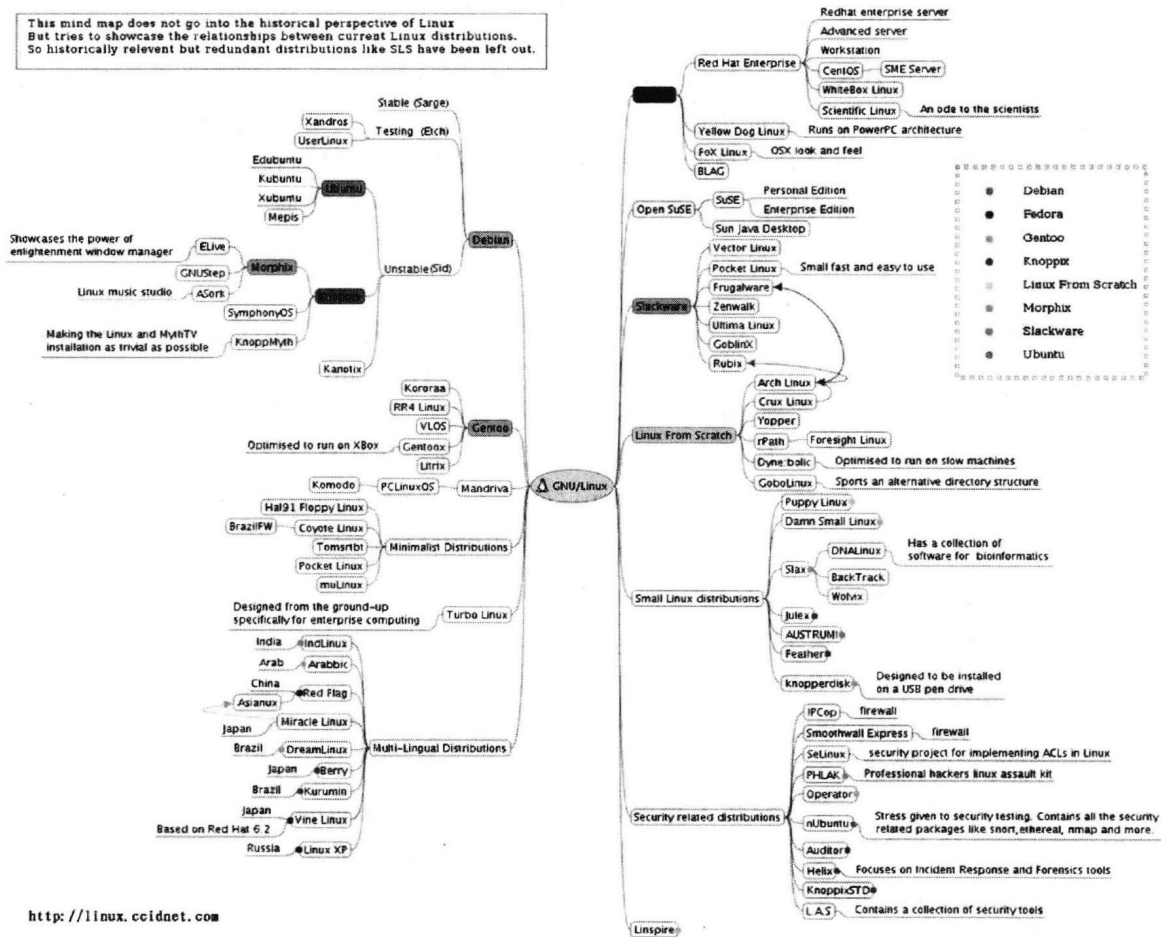


图 1.2 Linux 操作系统家族

面对众多发行版本,如何进行选择是刚刚接触 Linux 用户所面临的第一个问题。尽管每一种发行版本都有各自的特点,但是 Linux 发行版本之间并没有本质区别,特别是其安装越来越容易。我们在选择发行版本时,最主要的因素包括内核版本、软件包管理、工具、社区活跃

度、参考资料等。

发行版本最基本的是 Linux 内核,内核是构成操作系统的核心软件。在内核之上,通常是一些 GNU 库(例如 Glibc)和工具,它们对内核功能提供了高级的访问能力,这意味着可以进行更高级的编程。应用程序都是在这之上进行绑定的。现在常用的内核是 v2.6,并且以极高的频率进行更新,截止本书截稿时,Linux 内核的最新版本是 2.6.35.4。因此在选择发行版本时,尽量选择提供 v2.6 内核的发行版本。经典 Red Hat Linux 9.0 曾是无数程序员的首选,但这个版本只提供了 2.4 的内核(当然可以为其升级到 2.6 的内核),所以不建议初学者使用。

Ubuntu 是一个由社区开发的基于 Linux 的操作系统,由 Canonical 公司以及全球数百家公司来提供商业支持。Ubuntu 近几年发展非常快,其中的原因之一是采用 debian 的 ATP 软件包管理器,另外一个原因是社区异常活跃。在 Ubuntu 系统下,只须执行 apt-get install 就可以下载并安装软件,非常方便。但根据笔者使用各种发行版本的经验来看,Red Hat Linux 系列更容易上手一些,这其中包括 Fedora、CentOS 和 RHEL;而 Ubuntu 更适合有些 Linux 基础的人使用。

本书使用的开发环境就是基于 Windows 的 VMWare 所构建的 CentOS(Community Enterprise Operating System)系统。CentOS 是 RHEL(Red Hat Enterprise Linux)源代码再编译的产物,而且在 RHEL 的基础上修正了不少已知的 BUG。相对于其他 Linux 发行版,CentOS 更具稳定性。最重要的是,CentOS 提供和 RHEL 的命令几乎百分之百兼容,这非常适合初学者学习使用。CentOS 的下载地址为 <http://mirrors.kernel.org/centos/5.4/isos/i386/> 及 <http://mirrors.kernel.org/centos/5.4/isos/i386/CentOS-5.4-i386-bin-DVD.iso>。

2. 嵌入式 Linux

所谓嵌入式 Linux,是指 Linux 在嵌入式系统中的应用,而不是什么嵌入式功能。实际上,嵌入式 Linux 和 Linux 是同一件事。换句话说,嵌入式 Linux 和 Linux 开发应用程序没有太大区别。嵌入式 Linux 开发就是构建一个 Linux 系统,这需要熟悉 Linux 系统组成部分、熟悉 Linux 开发工具、熟悉 Linux 系统编程。

嵌入式 Linux 系统包含引导装入程序(Bootloader)、内核(Kernel)和文件系统(Filesystem)这 3 部分。对于嵌入式 Linux 系统来说,这 3 个部分是必不可少的。另外,我们要明确,开发一个嵌入式 Linux 项目之前,应包括下面几个要点:

(1) 构建开发环境

交叉开发环境是嵌入式 Linux 开发的基本模型。Linux 环境配置、GNU 工具链、测试工具甚至集成开发环境都是开发嵌入式 Linux 开发的利器。当然,可以购买商业的 Linux 发行版本,如 MontaVista Linux,它会为开发者提供可靠的软件和完整的开发工具包。

(2) 构建 Linux 内核

因为嵌入式 Linux 开发一般需要重新定制 Linux 内核,所以熟悉内核配置编译和移植也

很重要。

(3) 构建目标板的引导方式

开发板的 Bootloader 负责硬件平台的最基本初始化,并且具备引导 Linux 内核启动的功能。由于硬件平台是专门定制的,一般需要修改(移植)并编译 Bootloader。

(4) 构建 Linux 根文件系统

Linux 内核启动后期需要挂载根文件系统,程序和文件都存放在于文件系统中。系统启动必需的程序和文件也必须放在根文件系统中。

(5) 理解 Linux 调度机制和进程线程编程

Linux 调度机制影响到任务的实时性,理解调度机制可以更好地运用任务优先级。进程和线程编程则是应用程序开发所必须的。

1.2 Linux 系统的目录结构

Linux 继承了 UNIX 操作系统结构清晰的特点,在 Linux 下的文件结构非常有条理。先看我们熟悉的 Windows 系统,Windows 也有自己的目录结构,如 Program Files、Document and Settings、Windows、Windows\system。学习 Linux 的第一步就是要了解它的目录结构,要知道命令的存放位置、配置文件的存放路径。

图 1.3 列出了 Linux 系统的目录结构的树状图。图中只给出了系统中的一些典型目录,不代表全部。为了使读者掌握 Linux 系统的目录结构,请参考表 1.1 中的解释说明。

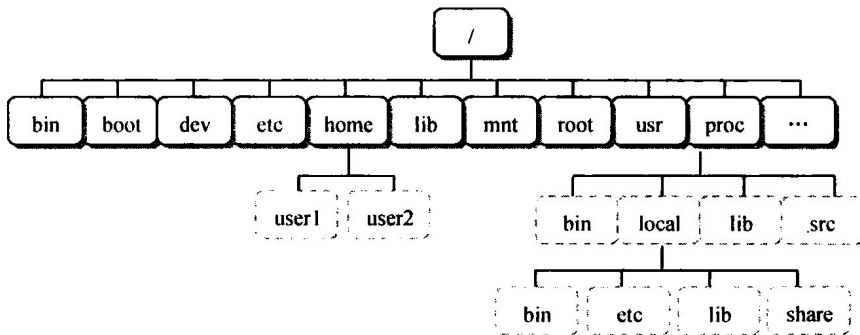


图 1.3 Linux 系统目录结构树状图

表 1.1 Linux 根目录

挂载点	说 明
/bin	存放 Linux 系统下经常使用的命令,如 ls、cd、cp、rm、mv 等
/boot	存放启动 Linux 时使用的一些核心文件,包括系统映像和链接文件,如 vmlinuz-2.6.18
/dev	存放 Linux 的设备文件。在 Linux 中访问设备的方式和访问文件的方式是相同。例如,/dev/fd0 表示第一个软盘驱动器,/dev/lp0 表示第一个并行端口
/etc	存放系统管理的配置文件。例如,该目录下的 exports 文件是 NFS 系统服务的配置文件,passwd 文件是系统用户密码文件(当然密码无法看到,真正的密码文件是 shadow)
/home	是用户的主目录。每当创建一个用户,就会产生一个和该用户名相同的文件夹,当这个用户登录时,首先登录这个目录中,也就是这个用户的家目录。例如,/home/user1 是 user1 用户的家目录,/home/user2 是 user2 用户的家目录
/lib	存放系统动态连接共享库。人们平时使用的 ls、cd 等系统命令,都是要到这个目录下找依赖的库。例如,libc.so.6 库是 ls 命令执行时需要使用的
/mnt	是系统提供安装额外文件系统时候的安装目录,如挂载软驱、光驱等
/root	root 用户的家目录
/usr	这个目录是 Linux 系统中最大的目录,存放许多应用程序。重要的子目录包括: /usr/bin 存放用户的应用程序 /usr/sbin 存放 root 用户使用的管理程序 /usr/include 存放 Linux 下开发和编译应用程序所需的头文件 /usr/lib 存放一些常用的动态链接共享库和静态档案库 /usr/local 是一般用户的目录,用户安装的应用软件常存放于此 /usr/man 帮助文件所在目录
/proc	一种文件系统,该目录下的文件都存在于内存中,其细节在后面“文件系统”中讲到
/var	存放经常发生变化的文件,如日志文件
/tmp	存放临时文件

表 1.1 中的目录是 Linux 系统根目录下的挂载点,也是一些最基本的内容,读者需要牢牢记住。掌握这些根目录后,还需要了解 Linux 下更多的内容。表 1.2 给出了 Linux 系统中一些比较重要的配置文件的介绍。

表 1.2 Linux 的重要文件

文 件	解释说明
/boot/grub/grub.conf	GRUB 的配置文件
/boot/System.map-*	内核的内核符号表,常用于调试过程中
/boot/vmlinuz-*	可引导的、压缩的内核。vm 代表 Virtual Memory
/boot/initrd-*	是在实际根文件系统可用之前挂载到系统中的一个初始根文件系统。initrd 与内核绑定在一起,并作为内核引导过程的一部分进行加载
/etc/aliases	sendmail 服务会使用该文件用做使用者名称转换的动作。当 sendmail 收到一个要送给某人的邮件时,它会依据 aliases 文件中的内容送给另一个使用者
/etc/adjtime	系统根据该文件中的参数调整硬件时间,并更新系统时间为硬件时间
/etc/anacrontab	anacron 服务的配置文件
/etc/auto.misc	定义/misc 目录中的挂载点,这种关系在/etc/auto.master 文件中定义
/etc/bashrc	bash 配置文件
/etc/cron.daily/*	参见计划任务工具 cron 服务的配置
/etc/cron.hourly/*	参见计划任务工具 cron 服务的配置
/etc/cron.monthly/*	参见计划任务工具 cron 服务的配置
/etc/cron.weekly/*	参见计划任务工具 cron 服务的配置
/etc/crontab	参见计划任务工具 cron 服务的配置
/etc/csh.cshrc	启动时 Cshells 执行的文件
/etc/csh.login	登录时 Cshells 执行的文件
/etc/cups/*	cups 服务配置文件
/etc/default/useradd	创建用户时用到的模板,建议读者打开看一下
/etc/dev.d	udev 从内核中过滤出磁盘的插拔事件,随后会调用这个目录下的脚本
/etc/DIR_COLORS	目录列表颜色
/etc/exports	NFS 服务的配置文件,详见“文件系统”章节的内容
/etc/filesystems	支持的文件系统类型
/etc/fstab	系统启动时,根据该文件的配置,挂载文件系统
/etc/group	系统中组的信息
/etc/host.conf	解析主机名
/etc/hosts	该文件记录每个主机名以及相对的 IP 地址,进行本地解析
/etc/hosts.allow	控制可以访问本机的 IP 地址
/etc/hosts.deny	控制禁止访问本机的 IP 地址