

高等院校信息技术规划教材

嵌入式 Linux程序设计

田卫新 张莉莉 编著



清华大学出版社

高等院校信息技术规划教材

嵌入式 Linux程序设计

田卫新 张莉莉 编著



清华大学出版社
北京

内 容 简 介

本书详细介绍了 Linux 操作系统的配置与使用,在嵌入式系统上移植 Linux 系统,以及 Linux 系统上应用程序的开发技术。主要内容包括:嵌入式系统简介、Linux 系统安装与配置、Linux 系统的使用基础、Linux 系统 C 语言程序设计、开发环境搭建、嵌入式 Linux 系统移植、Linux 并发程序设计、Linux 网络程序设计、Linux 文件应用程序设计、Linux 系统设备驱动模型、驱动程序开发以及图形用户界面程序设计。

阅读本书,要求读者具有 C 语言程序设计与嵌入式系统基础知识。本书可以作为高等院校计算机类、电子信息科学类及物联网工程等专业的本科生或研究生教材,也可以作为嵌入式工程技术人员学习 Linux 程序设计的参考书。

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 Linux 程序设计/田卫新,张莉莉编著. —北京:清华大学出版社,2017
(高等院校信息技术规划教材)
ISBN 978-7-302-46057-2

I. ①嵌… II. ①田… ②张… III. ①Linux 操作系统—程序设计—高等学校—教材
IV. ①TP316.85

中国版本图书馆 CIP 数据核字(2017)第 004885 号

责任编辑:袁勤勇 薛 阳

封面设计:常雪影

责任校对:时翠兰

责任印制:杨 艳

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 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.5

字 数:569 千字

版 次:2017 年 5 月第 1 版

印 次:2017 年 5 月第 1 次印刷

印 数:1~2000

定 价:45.00 元

产品编号:071268-01

前言

foreword

嵌入式系统在功耗、移动性等方面具有传统计算机所不具备的优点,在当前工业控制、物联网以及消费电子等多个领域具有广阔的应用前景,发展潜力巨大。随着嵌入式 CPU 的性能不断提高,嵌入式系统具有逐渐取代桌面 PC 的趋势,成为计算机技术发展的新方向。

Linux 内核支持 x86、PowerPC、ARM 等主流的 CPU 架构,移植性能好,内核可以根据需要进行裁减,同时 Linux 有庞大的用户群,Linux 平台应用程序具有统一的开发接口,在系统之间迁移比较容易。基于 Linux 操作系统平台已成为嵌入式系统开发的主流,越来越多的嵌入式系统开发商在产品中使用 Linux 系统作为应用平台。随着 Linux 内核结构以及实时性功能的不断完善,Linux 系统在嵌入式领域的应用范围将越来越广。

当前,嵌入式 Linux 程序设计方面的人才缺口巨大。一方面,企业研发嵌入式系统产品需要大量嵌入式软件开发人才;另一方面,由于专业划分及课程本身对知识结构的要求,只有少数高校在相关专业开设了嵌入式 Linux 程序设计课程,适合高校相关专业课程体系及教学特点的教材也比较缺乏。基于上述背景,本书的编写旨在以 C 语言程序设计、嵌入式系统程序设计课程为先修课,满足计算机应用、电子信息及物联网工程等专业学时安排的嵌入式 Linux 程序设计教学的需要。全书共分为 11 章。

第 1 章绪论介绍 Linux 的发展过程及版权特点,讲述了 Linux 操作系统的特点和内核的主要组成部分,嵌入式 Linux 交叉编译过程以及 Linux 内核在嵌入式系统应用中的优势、不足及改进途径。

第 2 章嵌入式系统概述介绍了计算机系统功能的演变过程,嵌入式系统在计算机系统分类中所处的位置,嵌入式系统的概念与应用领域、分类及特点,嵌入式系统的组成,ARM 体系结构特点及常见的 ARM 微处理器,ARM 的编程模型,常见的嵌入式操作系统以及嵌入式系统的开发过程。

第 3 章 Linux 系统安装与配置介绍了 Linux 的内核版本和发行版本、安装过程,Linux 系统的启动配置及程序安装方法。

第4章 Linux 系统操作基础介绍了 Linux 系统的使用和操作,包括文件管理、用户管理以及网络管理等部分。

第5章 Linux 系统 C 语言程序设计介绍了 C 语言特点,常用头文件及编译环境变量的配置,C 语言程序设计过程,编译器 GCC、调试器 GDB 的用法,工程管理器 make 的用法,模块化程序设计方法,ARM 平台交叉编译环境的搭建以及 Linux 系统常见的 C 语言集成开发环境。

第6章 Linux 系统移植介绍了在嵌入式硬件上安装 Linux 操作系统的过程,分为内核引导程序 Bootloader、Linux 内核、文件系统等三个部分的移植。

第7章 Linux 并发程序设计介绍了 Linux 多进程程序设计、进程之间的通信以及多线程程序设计。

第8章 Linux 网络编程介绍了 Linux 套接字,Linux 基于 TCP 的程序以及 Linux 基于 UDP 的程序设计。

第9章 Linux 文件编程介绍了 Linux 文件的概念,文件的读写操作、文件的加锁访问以及文件的并行访问复用模型等。

第10章 Linux 设备驱动程序设计介绍了设备驱动模型,总线型的设备驱动程序开发、中断设备驱动程序以及混杂型设备驱动程序开发。

第11章用户界面程序设计介绍了 Qt 的安装,Qt 应用程序基本结构以及 Qt 的常用类。

本书第4章、第7章、第8章和第9章由张莉莉编写,其余各章由田卫新编写并负责全书统稿。在本书编写过程中,袁军、张富生、鲍灵利、徐昊等提供了帮助,在此表示感谢。

由于编者水平有限,书中疏漏之处在所难免,恳请读者批评指正!

作 者

2016 年 10 月

目录

Contents

第 1 章 绪论	1
1.1 Linux 系统简介	1
1.1.1 早期操作系统的发展	1
1.1.2 Linux 系统的发展过程	2
1.1.3 Linux 的版权	2
1.2 Linux 操作系统特点	3
1.2.1 Linux 系统特性	3
1.2.2 Linux 与其他操作系统的比较	5
1.3 Linux 内核组成	7
1.3.1 进程管理	8
1.3.2 内存管理	9
1.3.3 虚拟文件系统	9
1.3.4 网络子系统	11
1.4 Linux 在嵌入式系统中的应用	12
1.4.1 嵌入式 Linux 系统优势	12
1.4.2 嵌入式 Linux 交叉编译过程	13
1.4.3 嵌入式 Linux 的不足及改进	14
习题	16
第 2 章 嵌入式系统概述	17
2.1 计算机系统发展及分类	17
2.1.1 计算机的主要应用领域	17
2.1.2 计算机系统分类	20
2.2 嵌入式系统简介	21
2.2.1 嵌入式系统定义	21
2.2.2 嵌入式系统的分类与特点	22
2.2.3 嵌入式系统发展历史	23

2.2.4 嵌入式系统组成	24
2.3 ARM 简介	27
2.3.1 ARM 体系结构特点	27
2.3.2 ARM 系列微处理器	28
2.3.3 ARM 编程模型	33
2.4 嵌入式操作系统	36
2.4.1 嵌入式 Linux	36
2.4.2 VxWorks	37
2.4.3 QNX	37
2.4.4 Windows CE	37
2.4.5 Palm OS	37
2.5 嵌入式系统开发过程	38
2.5.1 嵌入式系统整体开发过程	38
2.5.2 嵌入式硬件开发流程	41
2.5.3 嵌入式软件开发流程	43
习题	44
第 3 章 Linux 系统安装与配置	45
3.1 Linux 系统版本	45
3.1.1 Linux 内核版本	45
3.1.2 Linux 系统发行版本	49
3.2 Linux 的安装过程	51
3.2.1 系统最低配置与推荐配置	51
3.2.2 安装前准备工作	51
3.2.3 硬盘分区	52
3.2.4 安装系统	52
3.3 Linux 系统目录结构及基本配置	60
3.3.1 Linux 目录结构	60
3.3.2 系统启动过程	61
3.3.3 登录切换	64
3.3.4 安装程序	65
习题	67
第 4 章 Linux 系统操作基础	68
4.1 Shell 与命令行	68
4.1.1 Shell 概述	68
4.1.2 Shell 命令的分类	69

4.1.3	Shell 命令使用技巧	69
4.2	文件管理	70
4.2.1	什么是文件	70
4.2.2	Linux 文件系统	70
4.2.3	Linux 文件系统相关命令	71
4.3	用户管理	82
4.3.1	用户类型	82
4.3.2	用户账号管理	83
4.3.3	组管理	86
4.4	网络管理	87
4.4.1	网络的基本配置	87
4.4.2	网络状态命令	89
4.5	进程管理	93
4.5.1	什么是进程	93
4.5.2	进程管理命令	93
4.5.3	进程查看命令	95
	习题	97
第 5 章	Linux 系统 C 语言程序设计	98
5.1	C 语言概述	98
5.1.1	C 语言发展过程	98
5.1.2	C 语言的特点	99
5.1.3	C 语言与 Linux 系统	100
5.2	Linux 系统 C 语言程序设计过程	102
5.2.1	Linux 系统 C 语言程序流程	102
5.2.2	gcc 的编译选项	103
5.3	GDB 调试器用法	105
5.3.1	gdb 使用流程	105
5.3.2	gdb 基本命令	109
5.4	make 工程管理器	113
5.4.1	makefile 基本结构	114
5.4.2	makefile 变量	115
5.4.3	makefile 规则	118
5.5	模块化程序设计	119
5.5.1	静态库的实现	119
5.5.2	动态库的实现	121
5.6	ARM 平台 Linux 交叉编译环境	121
5.6.1	源码编译方式	122



5.6.2	二进制解压安装	123
5.7	集成开发环境	124
5.7.1	Qt Creator	125
5.7.2	KDevelop	125
5.7.3	Code::Blocks	125
5.7.4	NetBeans	126
5.7.5	Eclipse	127
	习题	128
第 6 章	Linux 系统移植	129
6.1	U-Boot 移植	129
6.1.1	BootLoader 简介	129
6.1.2	U-Boot 介绍	133
6.1.3	U-Boot 装载过程	133
6.1.4	OK6410 开发板 U-Boot 的移植	149
6.2	Linux 内核移植	150
6.2.1	计算机处理器架构	150
6.2.2	Linux 内核及源码结构	153
6.2.3	Linux 内核移植	155
6.3	根文件系统移植	163
6.3.1	虚拟文件系统	164
6.3.2	文件系统分类	165
6.3.3	根文件系统的挂载过程	167
6.3.4	根文件系统制作	170
	习题	173
第 7 章	Linux 并发程序设计	174
7.1	Linux 进程	174
7.2	进程控制	175
7.2.1	获取进程 ID	176
7.2.2	创建进程	176
7.2.3	进程调度	179
7.2.4	进程终止	180
7.2.5	等待进程结束	181
7.2.6	守护进程	183
7.3	进程之间的通信	186
7.3.1	管道	186

7.3.2	信号	193
7.3.3	消息队列	203
7.3.4	信号量	208
7.3.5	共享内存	213
7.4	线程	216
7.4.1	创建轻量级进程	217
7.4.2	POSIX 线程	218
7.4.3	线程属性	222
7.4.4	线程之间的同步与互斥	226
习题		237
第 8 章	Linux 网络编程	238
8.1	网络编程的基本知识	238
8.2	套接字	239
8.2.1	套接字函数	239
8.2.2	本地套接字	243
8.2.3	网络套接字	246
8.3	TCP 程序设计	253
8.4	UDP 程序设计	257
8.5	套接字属性设置	261
习题		265
第 9 章	Linux 文件编程	266
9.1	底层文件 I/O	266
9.1.1	文件描述符	267
9.1.2	底层文件 I/O 函数	267
9.2	高级文件操作	270
9.2.1	文件加锁	270
9.2.2	多路复用	276
9.3	标准文件 I/O	285
9.3.1	打开和关闭文件	286
9.3.2	文件读写	288
习题		291
第 10 章	Linux 设备驱动程序设计	292
10.1	设备驱动模型简介	292
10.1.1	底层数据模型	292



10.1.2	上层设备模型	297
10.1.3	字符设备驱动模型应用接口	302
10.2	platform 设备驱动程序设计	306
10.2.1	platform_device 注册	306
10.2.2	platform_driver 注册	307
10.2.3	platform 设备驱动框架源码	310
10.3	设备驱动程序编译及加载	313
10.3.1	编译 Linux 设备驱动程序	313
10.3.2	驱动程序的加载	314
10.4	设备驱动程序硬件访问	317
10.4.1	内存与 I/O 设备	317
10.4.2	中断	320
10.5	驱动程序设计实例分析	321
10.5.1	蜂鸣器驱动程序设计	321
10.5.2	中断驱动程序	333
10.5.3	混杂设备驱动程序	336
习题		339
第 11 章 用户界面程序设计		340
11.1	Qt 简介及安装	340
11.1.1	Qt 简介	340
11.1.2	Qt 安装	341
11.2	Qt 程序设计入门	343
11.2.1	命令行方式编程	343
11.2.2	Qt Creator 方式	345
11.2.3	Qt 程序的运行机制	353
11.3	Qt 常用类介绍	357
11.3.1	Qt 分组类库	357
11.3.2	Qt 类库介绍	359
习题		382

绪 论

嵌入式系统产业伴随着国家产业发展从通信、消费电子领域转战到汽车电子、智能安防、工业控制和北斗导航等领域,今天嵌入式技术已经无处不在,在应用数量上已远超通用计算机。嵌入式技术与人工智能、模式识别技术的结合,开发出了各种更具人性化、智能化的嵌入式应用系统,嵌入式系统已经成为物联网中的关键设备。随着芯片计算能力的提升,体积小、功耗低的嵌入式产品正在逐渐替代传统 PC 的应用领域,随着智能手机大量使用,“计算机”将会消失,以嵌入式移动设备为中心的“无处不在的计算”将成为现实。

在嵌入式软件开发方面,开源、可裁减、具有统一应用接口的 Linux 系统,成为主流的开发平台。

1.1 Linux 系统简介

1.1.1 早期操作系统的发展

自第一台计算机问世以来,操作系统作为最重要的系统软件,不断更新发展,推动了计算机功能的提高和向各领域的应用渗透。计算机从最初主要作为科学计算的工具,逐步发展到自动控制、信息管理、多媒体服务、电子商务等各项工作的主要工具。在此期间,计算机在体系结构在保持不变的情况下,硬件性能基本按照摩尔规律以每 18 个月提高一倍的速度增长,为操作系统快速稳定发展提供了条件。

Windows 操作系统正是这一时期的产物。20 世纪 80 年代初,美国微软公司开发出第一代 PC 操作系统 DOS,主要功能是对磁盘的管理,其关于文件格式、磁盘划分的理念一直延续下来。

UNIX 是 20 世纪 60 年代末在小型计算机上开发出来的一款操作系统。UNIX 始于由贝尔实验室、通用电气和麻省理工学院在 1965 年合作开展的 MULTICS 系统项目。MULTICS 项目停止后,后来被称为 UNIX 之父的 Ken Thompson 于 1971 年与合作者在 PDP-11/24 的机器上开发出了 UNIX 第一个版本,这台计算机只有 24KB 的物理内存和 500KB 磁盘空间,UNIX 占用了 12KB 的内存。到了 1973 年的时候,Ken Thompson 与 Dennis Ritchie 感到用汇编语言编写的操作系统移植太困难,他们想用高级语言来完成第三版。对于当时完全以汇编语言来开发程序的年代,他们的想法相当地疯狂。一开



始他们想尝试用 FORTRAN,可是失败了。后来他们用一个叫 BCPL(Basic Combined Programming Language)的语言开发。他们整合了 BCPL 形成 B 语言,后来 Dennis Ritchie 觉得 B 语言还是不能满足要求,于是就改良了 B 语言,这就是今天大名鼎鼎的 C 语言。Ken Thompson 与 Dennis Ritchie 成功地用 C 语言重写了 UNIX 的第三版内核。至此,UNIX 这个操作系统修改、移植相当便利,为 UNIX 日后的普及打下了坚实的基础。而 UNIX 和 C 完美地结合成为一个统一体,C 与 UNIX 很快成为各自领域的主导。

20 世纪 70 年代,AT&T 公司开始注意到 UNIX 所带来的商业价值。公司的律师开始寻找一些手段来保护 UNIX,并让其成为一种商业机密。从 1979 年 UNIX 的版本 V7 开始,UNIX 的许可证开始禁止大学使用 UNIX 的源码,包括在授课中学习。

到了 1980 年,UNIX 有两个最主要的版本线,一个是 Berkeley 的 BSD UNIX,另一个是 AT&T 的 UNIX,同时出于可观的商业前景,UNIX 开始出现各种各样的变种。1983 年,AT&T 公司发布的 UNIX System V 是一个非常成功的版本,成为一些 UNIX 共同特性的源头,引入了初始化脚本(/etc/init.d)、vi 编辑器和 curses 库等。UNIX System V 最终造就了 IBM 公司的 AIX、HP 公司的 HP-UX 以及 Sun 公司的 Solaris。后者是服务器市场的主要操作系统。

UNIX 是计算机系统发展史上第一款成熟的操作系统,许多特性成为操作系统的工业标准。但是其商业运作模式被指不符合软件协作的原则,触发了自由软件运动兴起。

1.1.2 Linux 系统的发展过程

Linux 操作系统内核原型是芬兰赫尔辛基大学 Linus Torvalds 在一个类 UNIX 操作系统源码基础上编写的,于 1991 年 10 月 5 日正式对外发布了第一个版本。Linux 的第一个版本完全按照自由软件的扩散版权进行分发,内核所有源代码必须公开,任何人都不得从 Linux 交易中获利。这种方式限制 Linux 以磁盘复制或者 CD-ROM 等媒体形式发行,也限制了商业公司参与 Linux 开发并提供技术支持,妨碍了 Linux 的健康发展。这种情况下,Linus 决定转向 GPL 版权,这一版权除了规定自由软件的各项许可权之外,还允许用户出售自己的程序备份。从此以后,很多技术力量雄厚又善于市场运作的商业软件公司加入,开发出了多种 Linux 的发行版本,增加了更易于用户使用的图形用户界面和众多的软件开发工具,极大地拓展了 Linux 的用户基础,使 Linux 操作系统得以很快形成并成熟。

1.1.3 Linux 的版权

Linux 的研发与普及,是和由斯托曼发起的自由软件运动兴起密不可分的。自由软件运动的目的是让软件使用者不受限制地使用软件,对软件具有使用、复制、研究、修改和分发的自由。GNU 计划是这一运动的支柱,旨在为自由软件运动开发相关软件,Linux 内核的加入有力地支持了这一计划。加入 GNU 计划的软件,要以自由软件许可证发布,GNU 通用公共许可证(GNU GPL)是其中最主要的一种,Linux 内核即采用 GPL 许可证。它要求:

(1) 软件以源代码的形式发布,并规定任何用户能够以源代码的形式将软件复制或发布给别的用户。

(2) 允许对软件进行修改及部分删除后发布,只要软件使用了受 GPL 保护的任一部分,那么该软件就继承了 GPL 许可证,并因此而成为 GPL 软件,必须发布源代码。

(3) 允许在分发 GPL 软件时收费,但在收费前必须向客户提供该软件的 GNU GPL 许可协议。

由于 GPL 很难被商业软件所应用,所以后来 GNU 推出了 LGPL 许可证,该许可证对产品所保留的权利比 GPL 少,总的来说,LGPL 适合那些用于非 GPL 或非开源产品的开源类库或框架。因为 GPL 要求,使用了 GPL 代码的产品必须也使用 GPL 协议,开发者不允许将 GPL 代码用于商业产品。LGPL 绕过了这一限制。

Linux 是 GNU 软件,不同于开源软件和免费软件。GNU 软件是一个比开源软件更严格的观念,所有 GNU 软件都是开放源代码的,但不是所有的开源软件都能被称为 GNU 软件。免费软件是一种不需付费就可取得的软件,但是通常有其他的限制,使用者并没有使用、复制、研究、修改和分发的自由。该软件的源代码不一定会公开,也有可能限制再发行的自由,所以免费软件的重点是不需要付费。另一方面,GNU 软件也不一定是免费的。

1.2 Linux 操作系统特点

1.2.1 Linux 系统特性

Linux 操作系统在短短的几年之内便得到了非常迅猛的发展,这与 Linux 具有的良好特性是分不开的。Linux 包含 UNIX 的全部功能和优点。简单地说,Linux 具有以下主要特性。

1. 开放性

Linux 是一个遵循 POSIX 的多用户、多任务、支持多线程和多 CPU 的操作系统,它能运行主要的 UNIX 工具软件、应用程序和网络协议。其内核源码遵循内核编码规范、各发行版应用程序之间遵循 Linux 标准规范、在网络互联方面遵循 OSI 规范、TCP/IP 协议等。

POSIX 表示可移植操作系统接口 (Portable Operating System Interface)。POSIX 标准定义了操作系统为应用程序提供的接口标准,是 IEEE 为了要在各种 UNIX 操作系统上运行的软件而定义的一系列 API 标准的总称,其正式称呼为 IEEE 1003,而国际标准名称为 ISO/IEC 9945。POSIX 标准确保了应用程序在源码级别的可移植性。



2. 多用户

多用户是指计算机系统资源可以被不同用户各自拥有和使用,即每个用户对自己的资源(如文件、设备)有特定的权限,互不影响。Linux 和 UNIX 都具有多用户的特性。

3. 抢占多任务

多任务是现代计算机操作系统的一个主要特点,它是指计算机同时执行多个程序,而且各个程序的运行互相独立。Linux 系统采用抢占式多任务调度方式。抢占是指操作系统允许满足某些重要条件(例如:优先级,公平性)的任务打断当前正在 CPU 上运行的任务而得到调度执行。并且这种打断不需要当前正在运行的任务的配合,同时被打断的程序在后来可以再次被调度恢复执行。

多任务操作系统可以按照协作多任务和抢占式多任务来划分。本质上,抢占就是允许高优先级的任务可以立即打断低优先级的任务而得到运行,使用抢占式内核可以保证系统响应时间。最高优先级的任务一旦就绪,总能得到 CPU 的使用权。当一个运行着的任务使一个比它优先级高的任务进入了就绪态,当前任务的 CPU 使用权就会被剥夺,或者说被挂起,高优先级的任务立刻得到了 CPU 的控制权。如果是中断服务子程序使一个高优先级的任务进入就绪态,中断完成时,中断了的任务被挂起,优先级高的那个任务开始运行。对低调度延迟或者实时操作系统的需求来说,支持完全抢占的特性是必需的。Linux 系统有内核抢占和用户抢占两种方式,实现了 SCHED_NORMAL、SCHED_BATCH、SCHED_FIFO、SCHED_RR、SCHED_DEADLINE 以及 SCHED_IDLE 等 6 种调度算法。

4. 良好的用户界面

Linux 向用户提供了两种接口:用户界面和系统调用。Linux 的传统用户界面是基于文本的命令行界面,即 Shell,它既可以联机使用,又可存储在文件上脱机使用。Shell 有很强的程序设计能力,用户可方便地用它编制程序,从而为用户扩充系统功能提供了更高级的手段。可编程 Shell 是指将多条命令组合在一起,形成一个 Shell 程序,这个程序可以单独运行,也可以与其他程序同时运行。Linux 还为用户提供了图形用户界面,利用鼠标、菜单、窗口、滚动条等图形操作对象,给用户呈现一个直观、易操作、交互性强、友好的图形化界面。系统调用给用户编程时使用的界面。用户可以在编程时直接使用系统提供的系统调用命令。系统通过这个界面为用户程序提供底层的、高效率的服务。

5. 设备独立性

设备独立性是指操作系统把所有外部设备统一视为文件,只要安装它们的驱动程序,任何用户都可以像使用文件一样操纵、使用这些设备,而不必知道它们的具体存在形式。

具有设备独立性的操作系统通过把每一个外围设备看作一个独立文件来简化增加

新设备的工作。当需要增加新设备时,系统管理员在内核中增加必要的连接。这种连接,也称作设备驱动程序,保证每次调用设备提供服务时,内核以相同的方式来处理它们。当新的或更好的外设被开发并交付给用户时,只要这些设备连接到内核,就能不受限制地立即访问它们。设备独立性的关键在于内核的适应能力。其他操作系统只允许一定数量或一定种类的外部设备连接,而具有设备独立性的操作系统能够容纳任意种类及任意数量的设备,因为每一个设备都是通过其与内核的专用连接独立进行访问的。

Linux 是具有设备独立性的操作系统,它的内核具有高度适应能力,随着更多的程序员利用 Linux 编程,会有更多的硬件设备加入到各种 Linux 内核和发行版本中。另外,由于用户可以免费得到 Linux 的内核源代码,因此,用户也可以修改内核源代码,以便适应新增加的外部设备。

6. 丰富的网络功能

完善的内置网络是 Linux 的一大特点,Linux 在通信和网络方面的功能优于其他操作系统。它的联网能力与内核紧密地结合在一起,为用户提供了完善的、强大的网络功能。Linux 内置了大量支持 Internet 的软件,实现了网络文件传输、路由功能等。

7. 良好的可移植性

可移植性是指将操作系统从一个平台转移到另一个平台时它仍然能按其自身的方式运行的能力。Linux 是一种可移植的操作系统,能够在从微型计算机到大型计算机的任何环境和任何平台上运行。可移植性为运行 Linux 的不同计算机平台与其他计算机进行准确而有效的通信提供了手段,不需要另外增加特殊和昂贵的通信接口。

Linux 最初在 x86 平台上开发,但 Linux 内核可以运行在多种处理器平台上,如 ARM、680x0、SPARC、Alpha、Intel 64 位芯片架构等。同时也支持多处理器技术,多个处理器同时工作,使系统性能大大提高。Linux 可以适应各种主要的计算机硬件设备类型,比如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型计算机和超级计算机。

8. 可靠的系统安全

Linux 系统遵循标准规范,开放源代码,系统安全性建立在公开、透明的基础上,Linux 系统处在不断开发及完善的过程中,有一套严格有效的代码审计与纠错机制,核心代码及关键算法经过广泛测试。采取了许多安全技术措施,包括对读、写控制、带保护的子系统、审计跟踪、核心授权等,为网络多用户环境提供了安全保障,从而确保了 Linux 是当前最安全的操作系统之一。

1.2.2 Linux 与其他操作系统的比较

Linux 可以与 MS-DOS、OS/2、Windows 等操作系统共存于同一台机器上。它们均为操作系统,具有一些共性,但是又各有特色,有所区别,可以根据个人需要或者使用习惯选择安装其中的一种或几种。

目前,运行在 PC 上的操作系统主要有 Microsoft 的 Windows、IBM 的 OS/2 和



Apple 的 Mac 等。早期的 PC 用户普遍使用 MS-DOS, 因为这种操作系统对机器的硬件配置要求不高, 而随着计算机硬件技术的飞速发展, 硬件设备的价格越来越低, 人们可以相对容易地提高计算机的硬件配置, 于是开始使用 Windows 等具有图形界面的操作系统。Linux 是新近被人们所关注的操作系统, 它正在逐渐为 PC 用户所接受。那么, Linux 与其他操作系统的主要区别是什么呢? 下面从三个方面加以比较。

1. Linux 与 MS-DOS 之间的区别

不运行 X Window 时的 Linux 与 MS-DOS 的操作界面和使用方式非常相似, 但二者的功能和性能有很大区别。就发挥处理器功能来说, MS-DOS 没有完全发挥 x86 处理器的功能, 而 Linux 完全在处理器保护模式下运行, 充分利用了处理器的所有特性。Linux 可以直接访问计算机内的所有可用内存, 提供完整的 UNIX 接口。就操作系统的功能来说, MS-DOS 是单用户、单任务的操作系统, 一旦用户运行一个 MS-DOS 的应用程序, 它就独占了系统的资源, 用户不可能再同时运行其他应用程序。而 Linux 是多用户、多任务的操作系统, 可以有多个用户同时登录, 而且可以同时运行多个应用程序。就使用费用而言, MS-DOS 是商业软件, 需要付费购买使用, 而 Linux 是免费的, 用户可以从 Internet 上或者其他途径获得它的版本, 而且可以任意使用, 不用考虑付费购买问题。

2. Linux 与 Windows 之间的区别

从发展的背景看, Linux 是从一个比较成熟的操作系统发展而来的, 而其他操作系统 (如 Windows 等) 都是自成体系, 没有相依托的操作系统。这一区别使得 Linux 的用户能大大地从 UNIX 团体的贡献中获益。Linux 给个人计算机带来了能够与 UNIX 系统匹敌的速度、效率和灵活性, 使个人计算机具有的潜力得到了充分发挥。Linux 不仅在性能上能够与 UNIX 系统相匹敌, 而且具有强大的网络功能, 能够支持 Internet、Intranet、Windows、AppleTalk 等网络。在 Linux 中, 几乎可以找到需要的所有内容。

Linux 拥有与 Windows 和 Mac 一样功能完备的图形用户界面——X Window 系统。X Window 系统是用于 UNIX 机器的一个图形系统, 它支持许多应用程序, 并且是业界的标准界面。Linux 不仅提供了强大的操作系统功能, 而且提供了丰富的应用软件, 在 Internet 上, 大量免费软件都是针对 UNIX 系统编写的, 这些程序包罗万象, 任何人都可以下载适合自己需要的软件及其源码, 以便修改和扩充操作系统或应用程序的功能。

Linux 稳定性好, 运行 Linux 的机器启动一次可以运行数月。Linux 提供了完全的内存保护, 每个进程都运行在自己的虚拟地址空间中, 不会损坏其他进程或内核使用的地址空间。任务与内核间相互隔离, 行为不良或编写不良的程序只能毁坏自己, 被破坏的进程几乎不可能使系统崩溃。

Windows 对硬件配置要求高, 而 Linux 在低端 PC 系统上仍然可以流畅运行, Linux 的最小安装仅需要 4MB 内存, Linux 内核允许在运行时装载和卸载硬件的驱动程序, 这样就不必装载全部的驱动程序, 可以最大化地节约内存资源。

Linux 的组网能力非常强大, 它提供了对 TCP/IP 的完善支持, 并且包括对下一代 Internet 协议 IPv6 的支持。Linux 内核还包括 IP 防火墙代码、IP 伪装、IP 服务质量控制