

后学历IT技能教育丛书

总主编 李世收

LINUX

实战教程

陈 健 陈俊良 编 著

南京
大学
出版社

后学历IT技能教育丛书

TP316/421

总主编 李世收

2007

LINUX 实战教程

陈 健 陈俊良 编 著

南京大學出版社

图书在版编目(CIP)数据

Linux 实战教程 / 陈健, 陈俊良编著. — 南京: 南京大学出版社, 2007. 9

(后学历 IT 技能教育丛书 / 李世收总主编)

ISBN 978 - 7 - 305 - 04664 - 3

I. L... II. ①陈... ②陈... III. Linux 操作系统—教材 IV. TP316. 89

中国版本图书馆 CIP 数据核字(2007)第 138309 号

出 版 者 南京大学出版社
社 址 南京市汉口路 22 号 邮 编 210093
网 址 <http://press.nju.edu.cn>
出 版 人 左 健
丛 书 名 后学历 IT 技能教育丛书
总 主 编 李世收
书 名 **LINUX 实战教程**
编 著 陈 健 陈俊良
责 任 编 辑 阎居梅 编辑热线 025 - 83597087
照 排 南京紫藤制版印务中心
印 刷 盐城市华光印刷厂
开 本 787×1092 1/16 26 印张 630 千字数
版 次 2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷
ISBN 978 - 7 - 305 - 04664 - 3
定 价 40.00 元
发 行 热 线 025 - 83594756
电 子 邮 箱 sales@press.nju.edu.cn(销售部)
nupress1@public1.ppt.js.cn

* 版权所有, 侵权必究

* 凡购买南大版图书, 如有印装质量问题, 请与所购
图书销售部门联系调换

序

FOREWORD

我国高等教育从精英教育向大众教育发展后,高校毕业生的就业问题一直是社会关注的热点,一方面每年有大量的毕业生走入社会,找不到合适的工作;另一方面大量的工作岗位找不到合适的员工。本系列丛书从后学历教育的角度出发,在传统学历教育之后更多地倾注于对学生实际应用能力的培养,其目的是使教育与社会需求接轨,更好地适应社会对人才的要求,其着眼点在“技能”。

这一新的尝试,不仅肯定了学历教育的必要,更是强调了学历之后教育的必需。在知识更新迅速的 21 世纪,从业人员知识与技能的学习更是终身性的。南京工业大学信息中心主任李世收教授所带领的研究团队正是基于这种共识,率先在国内 IT 领域推出了此套以网络技能后学历教育为主题的系列丛书。

本丛书在编写方法与内容上,最大的特色是做到了理论与实践的紧密结合。该丛书的编写者既是大型网络的管理员,具有丰富的实践经验,同时也是富有教学经验的高校教师。本丛书体例独到、内容丰富、理论基础扎实,更难能可贵的是,实验全部经过验证。真实的案例、详尽的讲解,即使是对网络了解不多的读者也不会觉得生涩难懂;专业的技术、浅显的表述,为初次接触网络的爱好者打开了一扇通往网络世界的大门。

本丛书既可作为网络教育的教材,也可作为网络从业人员的自学参考书或工具手册。

本丛书以“后学历教育”为切入点,提出了网络教育的新的内容、新的视角、新的方法,是很有意义的,我对本书的出版感到高兴,并对作者取得的成果表示衷心祝贺。我相信,该丛书的出版,对于推进网络教育的发展必将起到积极的作用。

中国工程院院士 徐南平

2007.5. 南京.

前

言

PREFACE

Linux 最早由一位名叫 Linus Torvalds 的计算机业余爱好者编写,当时他是芬兰赫尔辛基大学的学生。最初他想设计一个代替 Minix(Andrew Tannebaum 教授编写的一个操作系统教学程序)的操作系统,他把自己开发的代码放到网络上,供爱好者使用和修改,从而形成了 Linux 的雏形。许多黑客成了这个程序的发烧友,努力修改 Linux,多年来,成千上万的编程爱好者给 Linux 找漏洞,修改程序,编写各种应用和驱动程序,使得 Linux 变得越来越稳定、安全、可靠,以至优美(关于 Linux 诞生过程在本书第 1 章“Linux 与 Unix 的异同”中有详细介绍)。

现在,使用 Linux 的人越来越多,越来越多的机构使用 Linux 来做基本平台。据不完全统计,在全世界已经有近 800 万 Linux 用户,其中,中国用户大约有近 10 万人。据 IDC 统计:2003 年,Linux 服务器的销售已经超过 Unix 服务器,市场占有率达到 16%,在 2004 年第一季度达到 17%,成为服务器领域第二大操作系统。IDC 还预测,Linux 在未来几年内将以每年 25% 的速度增长,中国的 Linux 市场更将保持 40% 的增长。

Linux 系统和其他系统相比,究竟有哪些优势?首先来看看商品化的操作系统对我们有哪些限制:

- 只能有偿使用,而且必须满足供应商所要求的一切条件;
- 不能擅自修改操作系统的内核;
- 应用程序都为特定的操作系统和机器编写,使用者无权也无法修改这些应用程序;

我们来看看 Linux 的好处:

Linux 是一套具有 Unix 全部功能的免费操作系统,为广大的计算机爱好者提供了学习、探索以及修改计算机操作系统内核的机会。

因为代码开放,我们可以创造和修改 Linux,把操作系统变成一种操作环境。

Linux 是一套自由软件,用户可以无偿地得到它及其源代码,无偿地获得大量的应用程序,而且可以任意地修改和补充它们。这对用户学习、了解 Unix 操作系统的内核非常有益。Linux 提供了比其他任何操作系统都要强大的功能,可以使用户远离各种商品化软件提供者促销广告的诱惑,也不用承受商业软件的升级之苦,人们将



节省大量用于购买或升级程序的资金。

Linux 本身包含的应用程序以及移植到 Linux 上的应用程序包罗万象,任何一位用户都能在网上找到适合自己需要的应用程序及其源代码,根据自己的需要下载源代码,修改和扩充操作系统或应用程序。对于任何商品化操作系统来说这都是不可想象、无法做到的。

Linux 具有其他商业操作系统不可能有的“自由”与“效能”。在 Linux 中,任何人都可以随心所欲地组合出所想要的功能:如果要 will email 和传真进行组合,三行不到的 script 就可以完成;而在 Windows 平台上,需要等待两者功能都有的软件出现;性能方面,只要一台 486+32 MB RAM 的 PC 机,一个中小型的公司就可以把 mail server、web server 以及 ftp server 的任务承担起来,而 Windows 的结果,大家都知道。

Linux 所支持的平台种类是所有操作系统中最多的。它支持 Sun Sparc、SGI Mips、AppleMac、PowerPC、Alpha、HP-PA、X86 和 PDA 等设备。Linux 核心的高度可移植性可见一斑。关于 Linux 的文章一篇又一篇,但总会有很多人在问:什么是 Linux? 为什么要用 Linux? 怎样去使用 Linux? 在这本书里,相信您一定可以找到答案。

录

CONTENTS

第 1 部分 安装 Linux

第 1 章 Linux 与 Unix 的异同

- 1.1 什么是 Linux 3
- 1.2 从 Unix 到 Linux 的历史 4
- 1.3 什么是 Linux 的套件(distribution) 7
- 1.4 Linux 的优缺点 8

第 2 章 如何学习 Linux

- 2.1 Linux 的应用方向 10
- 2.2 从头学起 11
- 2.3 X-Window 还是 command line 12
- 2.4 网络资料查询与提问 12
- 2.5 其他建议 13

第 3 章 为什么要选择 Slackware Linux

- 3.1 Slackware Linux 的历史 14
- 3.2 Slackware Linux 的特点 15
- 3.3 关于 Slackware Linux 的抱怨 16
- 3.4 Slackware Linux 10 的新变化及相关网站 16

第 4 章 安装前的准备

- 4.1 认识电脑硬件配备 18
- 4.2 选择硬件配备 22
- 4.3 硬件在 Linux 中的设备号 24
- 4.4 主机的服务规划 25
- 4.5 主机硬盘的规划 26
- 4.6 两个范例 27



第 5 章 Slackware Linux 10 的安装

5.1 安装过程简介	30
5.2 Slackware Linux 10 光盘内容介绍	30
5.3 安装实例	32

第 6 章 Linux 的初步特点

6.1 第一次登录系统与离开系统	51
6.2 图形模式与文字模式的切换方式	53
6.3 命令下达方式与基础命令的练习	55
6.4 在线求助: man info	60
6.5 正确的关机方式	62
6.6 开机过程中遇到的问题及解决方案	64

第 2 部分 Linux 基础

第 7 章 Linux 文件属性、目录与管理

7.1 用户与组	69
7.2 Linux 文件属性	70
7.3 如何修改文件的权限	73
7.4 Linux 支持的文件系统格式与文件类型	76
7.5 Linux 目录配置	76
7.6 目录与路径	78
7.7 文件与目录管理	82
7.8 查看文件内容	85
7.9 显示、修改文件或目录的权限	89
7.10 搜寻文件或目录	92
7.11 SetUID SetGID Sticky bit 与 file 命令	95

第 8 章 Linux 磁盘和硬件管理

8.1 认识硬盘	99
8.2 查看硬盘或目录的容量	101
8.3 连接文件的介绍	103
8.4 划分与格式化硬盘	106
8.5 检查硬盘与数据同步写入	109
8.6 制作启动软盘	111
8.7 分区的装载	111
8.8 设定开机自动装载	114
8.9 虚拟内存 (swap)	116

第 9 章 文件的压缩与打包

- 9.1 Linux 压缩文件的后缀名 118
- 9.2 Linux 常用的压缩命令 118

第 3 部分 了解 Shell**第 10 章 vi 文件编辑器**

- 10.1 vi 简易说明 127
- 10.2 一个范例 127
- 10.3 非编辑模式命令说明 129
- 10.4 案例练习 131
- 10.5 vi 的额外功能 132

第 11 章 Bash shell 基本概念和命令

- 11.1 Bash shell 135
- 11.2 变量与变量的设定 136
- 11.3 命令别名与历史命令 143
- 11.4 Bash shell 的配置文件 144
- 11.5 通配符与特殊符号 146
- 11.6 命令重定向 148
- 11.7 管道命令(pipe) 150
- 11.8 正则表达式与相关命令 153

第 12 章 学习 Shell 脚本

- 12.1 Shell 脚本与脚本(Scripts)的执行 157
- 12.2 变量定义 Declare 159
- 12.3 交互式脚本 160
- 12.4 脚本参数的定义 160
- 12.5 脚本中的逻辑判断符与运算符 161
- 12.6 循环语句 169
- 12.7 调试脚本 172

第 4 部分 用户与进程管理**第 13 章 账号管理**

- 13.1 账号管理 177
- 13.2 密码管理与设定 185
- 13.3 用户身份的切换 185
- 13.4 用户查询 188

13.5	手工增加用户	189
第 14 章 例行性命令		
14.1	例行性命令与突发性工作安排	192
14.2	周期性工作安排	193
14.3	系统的 crontab 设定	195
第 15 章 进程管理		
15.1	后台工作管理	197
15.2	进程与资源管理	199
15.3	调整进程的优先级; nice renice	204
15.4	系统基本信息管理	205
15.5	网络校时	208

第 5 部分 Linux 高级技术

第 16 章 Linux 开机流程		
16.1	开机流程分析	213
16.2	lilo 的配置	229
第 17 章 源代码的编译、安装和管理		
17.1	开放源代码 编译器 可执行文件	231
17.2	基本概念	232
17.3	一个简单的范例	233
17.4	副程序的编译	234
17.5	加入连接的函数库	236
17.6	gcc 的用法	237
17.7	make 的简易用法	237
17.8	Tarball 套件的管理	241
17.9	使用 patch 更新源代码	243
17.10	函数库管理	244
17.11	ldconfig 与/etc/ld. so. conf	245
第 18 章 Linux 内核编译		
18.1	为什么要编译内核	247
18.2	内核版本与下载内核	248
18.3	编译内核	248
第 19 章 认识系统服务		
19.1	什么是 Daemon	273
19.2	分析/etc/inetd. conf	274
19.3	TCP_Wrappers	276

19.4	查看系统服务	276
第 20 章	系统日志	
20.1	什么是系统日志	279
20.2	配置日志文件: Syslogd Logrotate	280
第 21 章	系统备份	
21.1	哪些 Linux 资料具有备份的意义	285
21.2	备份介质和方式	286
21.3	备份工具选择	287
21.4	备份实例	287
<hr/> 第 6 部分 Linux 建站实战 <hr/>		
第 22 章	Linux 常用网络命令介绍	
22.1	网络参数设定命令	293
22.2	网络侦错命令	296
22.3	网络观察与查询命令	298
22.4	Ssh 命令	302
第 23 章	Telnet 和 SSH 服务器	
23.1	Telnet 服务器的安装、启动和关闭	304
23.2	Telnet 服务器的安全性	305
23.3	SSH 服务器	305
第 24 章	WEB 服务器	
24.1	Apache 服务器的配置和使用	309
24.2	安装 Apache/MySQL/PHP/SSL	313
第 25 章	DNS BIND 服务器	
25.1	DNS 原理	318
25.2	BIND 软件及安装	322
25.3	BIND 的配置	323
25.4	Client 端的设定	335
第 26 章	电子邮件服务器	
26.1	邮件服务器的运行原理	340
26.2	Sendmail 的安装与配置	342
26.3	POP3 服务器的安装与配置	350
26.4	邮件防毒软件的安装与配置	351
26.5	防垃圾邮件软件的安装与配置	355
第 27 章	代理服务器	
27.1	HTTP 代理服务器	359

27.2	Socks 代理服务器	366
第 28 章 FTP 服务器		
28.1	Proftpd 软件的安装	373
28.2	Proftpd 的主要文件	374
28.3	Proftpd 的配置	374
第 29 章 Iptables 防火墙及 NAT 服务器		
29.1	Iptables 数据报处理流程	384
29.2	Iptables 的语法	387
29.3	一个简单的 NAT 防火墙设定范例	392
第 30 章 基于 WEB 的 Linux 集成管理工具 Webmin		
30.1	Webmin 是什么	396
30.2	谁应该使用 Webmin	397
30.3	安装 Webmin	397
30.4	Webmin 的结构	399
30.5	保护 Webmin	401

第 1 部分

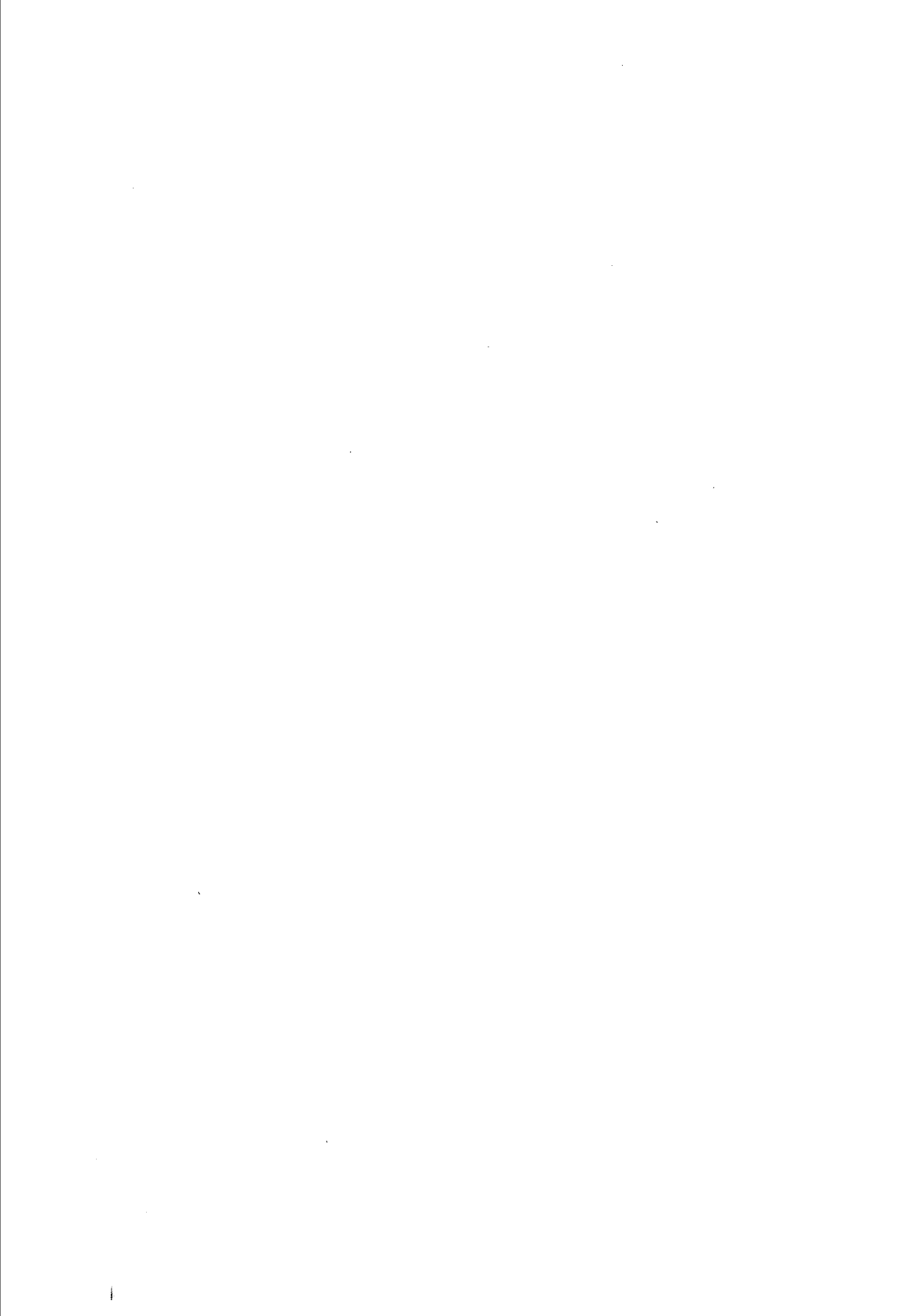
安装 LINUX

安装 LINUX

安装 LINUX

安装 LINUX

第 1 部分



第1章

Linux 与 Unix 的异同

1.1 什么是 Linux

在了解 Linux 之前,先谈谈什么是操作系统(Operation System, OS)。当我们使用电脑时,屏幕上显示的东西是通过显卡显像的;当我们看 DVD 影碟时,需要光驱、光碟、声卡等的支持;当我们使用互联网时,需由 Internet、网卡、网线以及其他相关的电子器材和网络器材来共同完成。由此可见,电脑工作的完成都是通过硬件来实现的。硬件是通过操作系统来进行控制的,操作系统的任务就是在使用者和硬件之间进行信息的传递。也就是说,没有操作系统,电脑硬件什么工作都不能完成。

操作系统里包含着什么东西?简单地讲,操作系统主要分为两个部分:核心和核心提供的工具。我们以用户常使用的 Windows 操作系统为例来说明。当我们打开 Windows 资源管理器时,硬盘中的文件被显示出来,这个显示硬盘当中文件的功能就是由“核心”来完成的。但是,要核心显示硬盘中具体哪一个目录下的文件,则是由“资源管理器”这个“工具”来完成的。再举一个例子,如果使用过 Windows 2000 或者 Windows XP 的话,常听过 FAT32 和 NTFS 文件系统格式,早期的 Windows 98 操作系统是无法读取或找到 NTFS 文件系统格式的硬盘的,为什么呢?因为它的“核心”不认识 NTFS 文件系统。所以,即使有资源管理器,由于核心不认识 NTFS,资源管理器还是没有办法完成我们想要做的事情。由此可见,整个硬件是由核心来管理的,如果核心不认识我们的硬件,我们就无法使用该硬件设备。核心的英文是“Kernel”,它是一个操作系统的最底层的東西,管理整个硬件资源的工作状态。Linux 有自己的核心,Windows 也有自己的核心。当有新的硬件加入到系统的时候,如果没有“Kernel”的支持,这个新的硬件是无法工作的(这一点在第 18 章“Linux 内核编译”中会有更详细的解释)。

一般情况下,操作系统核心(Kernel)管理的事务有:

- 系统调用接口(System call interface) 系统服务和 Kernel 沟通的接口。
- 进程控制(Process control) 系统程序控制中心。
- 内存管理(Memory management) 管理整个系统的内存。
- 文件系统管理(File system management) 管理 I/O、不同文件系统格式的支持等。
- 设备驱动(Device drivers) 硬件驱动程序的支持(注:Linux 提供“模块载入”功能,可以将驱动程序编译成模块,不需要重新编译核心,相关内容见第 18 章)。

由上可见,所有硬件资源都是由核心来管理的,当我们进行某些工作时,除了使用核心本身提供的功能(如上面提到的资源管理器)外,还要安装其他的应用软件。例如,我们看 DVD,除了 Windows 提供的媒体播放程序之外,还可以安装其他的 DVD 播放程序。这个播放程序就是应用软件,而这个应用软件可以帮我们控制核心来工作(即播放影片)。因此,可以这样说,核心是控制整个硬件的东西,也是一个操作系统的最底层。但是,要让整个操作系统更完备的话,还需要具备丰富的核心提供的工具,以及其他的应用软件的支持。

什么是 Linux? Linux 就是一个操作系统,这个操作系统里包含最主要的 Kernel 以及 Kernel 提供的工具。它提供了一个完整的操作系统中最底层的硬件控制与资源管理的架构,这个架构沿袭了它的老前辈 Unix 良好的传统,所以相当稳定而且功能强大。此外,由于这个优良的架构可以在目前的个人电脑(X86 系统)上运行,所以很多人都是在这个架构之下进行软件开发的,也就是说,Linux 拥有众多的应用软件。虽然 Linux 仅包含核心和核心提供的基本工具,但我们可以通过将核心、核心工具和众多的应用软件整合在一起,使得 Linux 成为一个完整的、功能强大的操作系统。我们将 Linux 系统与用户的相关性表示为下面的图示:

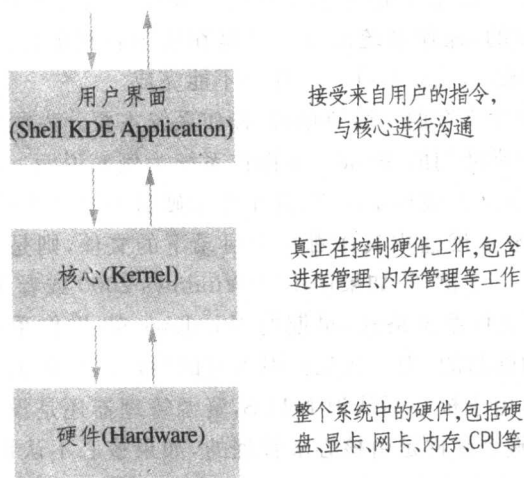


图 1-1 用户、用户界面、核心与硬件的关系

1.2 从 Unix 到 Linux 的历史

Linux 是很稳定的操作系统,它的诞生离不开 Unix 家族。有了 Unix 的帮助,Linux 才能站在巨人的肩膀上,成为一个品质优良的操作系统。

1. 一个没有完成的梦想: Bell, MIT 与 GE 的 Multics 系统

早期的个人电脑不像现在这样普及。在那个年代,电脑属于贵重物品,一个学校拥有一部大型主机,众多的教师和学生无法使用它。20 世纪 60 年代初期,麻省理工学院(MIT)发展了“相容分时系统”(Compatible Time-Sharing System, CTSS),就字面上的



意义来看,是一种可以允许多个终端机同时连线进入主机使用主机资源的大型主机,同时可以有大约 30 位用户连线使用主机。为了让这个分时系统功能更加强大,1965 年前后,由贝尔实验室、麻省理工学院和 GE 电器公司共同开发了一个名为 Multics 的大型操作系统,目标是让大型主机可以连线 300 位以上的用户。但是,在开发了 4 年之后,该计划宣布失败。

2. 一个小型文件系统的诞生:1969 年 Ken Thompson 的小型文件服务器系统

在复杂的 Multics 系统的计划失败后,贝尔实验室退出了该计划,曾经参与 Multics 的贝尔实验室的成员 Ken Thompson 由于自己的工作需要一套小型的文件存取操作系统,1969 年他以 DEC 公司的电脑 PDP-7 为硬件基准,设计了一个适合自己工作环境的小型文件系统,其中也包含他自行开发的一些小工具。这个系统就是最早期的 Unix 的源头。Ken 设计这个文件系统,主要是为了自己的存取方便,因此他只是对原来的 Multics 系统进行了大幅度的简化,当时还没有出现 Unix 这个名词。

3. Unix 的正式诞生:1973 年 Ritchie 等人以 C 语言写出第一个正式的 Unix 核心

由于 Thompson 设计的操作系统非常好用,所以在 Bell 实验室内部广为流传。经过数次修改,到了 1973 年,Bell 实验室的 Dennis Ritchie 等人用编译性能更好的 C 语言将原有系统的核心进行了大幅度的改写,并将这个操作系统命名为 Unix,这就是最早的 Unix 操作系统。

4. 重要的 Unix 分支:1977 年 BSD Unix 的诞生

在 Unix 早期的发展中,有一件相当重要的事情,就是 BSD Unix 的诞生。在 Unix 系统发表之后,伯克利大学(Berkeley)的教授对这个操作系统很感兴趣,在取得 Unix 的源代码并经过数次修改之后,1977 年发布了第一代的 Berkeley Software Distribution Unix (简称 BSD Unix)。BSD 是 Unix 的一个分支,它的发展对于 Unix 有相当大的影响,后来的 Sun 公司就是使用 BSD Unix 的核心进行了自己的商业 Unix 版本的开发。

5. 百家争鸣的 Unix 版本

由于 C 语言是一个高级程序设计语言,与硬件没有直接的相关性。而 Unix 操作系统本身就是由 C 语言写成的,因此,只要对 Unix 源代码进行简单的修改,就可以在不同的硬件架构上面运行 Unix 操作系统了。同样,由于 Unix 的便利与容易改写,一时间市场上出现了很多不同版本的 Unix,如 AT&T 的 System V,加州大学的 BSD、IBM 的 AIX 等等。

操作系统的核心必须要和硬件配合,才能有效地控制硬件资源,而早期生产电脑硬件时,公司之间没有约定统一的硬件标准,所以每一个电脑公司生产的硬件互不兼容,每个公司还要为自己的电脑硬件开发适合的 Unix 系统,而且自行开发的 Unix 只能配置本身的硬件设备,有名的 Sun、HP 等都是如此,他们开发的 Unix 没有办法在其他的硬件架构下工作。当时没有厂商针对个人电脑开发 Unix 系统,也没有出现支持个人电脑的 Unix 操作系统,所以,早先的 Unix 只能与服务器或者大型工作站划上等号。

