



Python网络编程 (Linux)

◎ 赵宏 包广斌 马栋林 编著



精彩实例推荐

- 网络嗅探实例
- GUI聊天室实例
- 网络爬虫实例
- 收发邮件实例
- Web应用程序实例

清华大学出版社

大数据与人工智能技术丛书



Python网络编程 (Linux)

◎ 赵宏 包广斌 马栋林 编著

清华大学出版社
北京

内 容 简 介

本书选用各项性能指标优良的 Ubuntu 作为 Linux 系统实例,以 Python 为编程语言,理论结合实践,系统地讲解网络各层次的功能、所包含的常用协议、数据报文格式,并给出实际操作的程序实例。本书包括 Linux 系统介绍、Python 语言基础、TCP/IP 协议簇、Socket 原理、进程和线程、网络应用程序实例、Web 应用程序开发等内容。本书内容也适用于其他版本的 Linux。

本书可作为普通高等院校信息类专业本科生和研究生教材,也可作为广大 Linux 用户、网络管理员、程序员的自学用书和参考手册。

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

Python 网络编程:Linux/赵宏,包广斌,马栋林编著. —北京:清华大学出版社,2018(2018.10重印)
(大数据与人工智能技术丛书)

ISBN 978-7-302-50483-2

I. ①P… II. ①赵… ②包… ③马… III. ①Linux 操作系统—程序设计—高等学校—教材
②软件工具—程序设计—高等学校—教材 IV. ①TP316.85 ②TP311.561

中国版本图书馆 CIP 数据核字(2018)第 131741 号

责任编辑:郑寅堃 李 晔

封面设计:刘 键

责任校对:李建庄

责任印制:董 瑾

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:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:14

字 数:342千字

版 次:2018年10月第1版

印 次:2018年10月第2次印刷

印 数:1001~2000

定 价:39.00元

产品编号:073742-01

前言

Linux 系统与互联网相伴而生,共同成长,成为现代信息技术高速发展的重要支撑和驱动力。Python 作为一种开源、跨平台、面向对象的新型计算机程序设计语言,语法简洁,语义清晰,有丰富、强大的库的支持,广泛应用在网络编程、科学计算、人工智能等各个领域。两者的结合能够使读者快速理解基础理论,掌握实践技能,提高学习和工作的效率。

本书选用 Ubuntu Desktop 为实践平台,以 Python 语言为编程工具,针对互联网所使用的 TCP/IP 协议簇进行分层介绍和解析,并给出实际操作的程序实例。

全书共分为 7 章。第 1 章介绍 Linux 的历史、特点、组成、常见发行版本以及 Linux 常用的各种安装方式,由马栋林编写。第 2 章讲解 Python 语言的特点、开发环境安装、数据类型、语法规则、语句、函数、模块、类、对象、异常、文件等内容,由赵宏编写。第 3 章分层讲解 TCP/IP 各层主要协议、数据报文格式、层间数据交换规则、常见网络应用与各层协议的对应、程序实例等内容,由包广斌编写。第 4 章讲解 Socket 原理、SOCK_STREAM、SOCK_DGRAM、SOCK_RAW 等内容,并通过程序实例演示 Socket 在 C/S 结构编程和网络嗅探中的实际应用,由赵宏和马栋林编写。第 5 章讲解多进程和多线程技术在网络编程中的应用,通过实例对比多进程与多线程实现方案的异同,并介绍了利用 socketserver 编写多进程和多线程程序的方法,最后通过 GUI 聊天室程序实例说明了多进程和多线程编程技术的实际应用,由包广斌和赵宏编写。第 6 章讲解网页内容获取、访问 FTP 服务器、访问 DNS、收发 E-mail、获取 DHCP 信息等实用程序的编写方法与过程,这些实例能够让读者进一步理解网络理论和工具软件的运行机制,由赵宏和包广斌编写。第 7 章介绍 Python 开发 Web 应用程序的方法,讲解 WSGI 工作原理,以流行的 Web 开发框架 Django 为例,演示 Web 应用程序开发工程,由赵宏和马栋林编写。

本书在编写过程中,得到兰州理工大学计算机与通信学院和信息中心各位老师的支持。硕士研究生韩泽宇、蒋家俊、张浩和王孝通帮助调试本书部分代码。

本书的编写受到兰州理工大学教学研究项目和 CERNET 下一代互联网技术项目(NGII20160311,NGII20160112)的资助。

因时间仓促,不足在所难免,请大家批评指正,读者可通过 zhaoh@lut.edu.cn 或 594286500@qq.com 与作者联系,也可以加入本书的 QQ 群 643116956 进行讨论。

为方便教学,作者开发了与本书相配套的多媒体课件、课后习题答案和程序代码,读者可扫描封底课件二维码下载或者向编者索取。

作 者

2018 年 6 月于兰州理工大学

目 录

第 1 章 Linux 系统介绍	1
1.1 Linux 的诞生	1
1.2 Linux 的特点	2
1.3 Linux 的组成	4
1.4 Linux 的应用	6
1.5 常见 Linux 发行版本	7
1.6 Linux 的安装	9
1.6.1 常用的安装方式	9
1.6.2 安装前的准备	10
1.6.3 虚拟机安装 Linux	10
1.6.4 多操作系统的安装	14
1.7 本章小结	14
习题	14
第 2 章 Python 语言基础	15
2.1 Python 语言简介	15
2.2 Python 语言解释器安装	15
2.2.1 在线安装	15
2.2.2 下载安装	16
2.2.3 管理多个 Python 版本	16
2.2.4 安装 Python 的 IDE 环境	17
2.2.5 测试 Python IDE	18
2.3 数据类型	18
2.3.1 布尔型	19
2.3.2 整型	20
2.3.3 浮点型	20
2.3.4 复数型	20
2.3.5 字符串型	22
2.3.6 列表型	27
2.3.7 元组型	29
2.3.8 字典型	29
2.3.9 日期型	31

2.3.10	数组型	38
2.4	语法规则与语句	40
2.4.1	输出与输入	40
2.4.2	条件判断	41
2.4.3	循环	42
2.5	函数与模块	45
2.5.1	自定义函数	45
2.5.2	默认参数	45
2.5.3	可变参数	46
2.5.4	关键字参数	47
2.5.5	命名关键字参数	47
2.5.6	参数组合规则	48
2.5.7	实参与形参	48
2.5.8	递归	49
2.5.9	模块	50
2.6	类与对象	52
2.6.1	类的定义与实例化对象	52
2.6.2	类属性与实例属性	52
2.6.3	属性封装	55
2.6.4	类的继承	57
2.6.5	多态	58
2.7	异常和异常处理	60
2.7.1	异常捕获与处理	62
2.7.2	抛出异常	64
2.8	文件	65
2.8.1	读写文本文件	67
2.8.2	读写二进制文件	69
2.8.3	读写JSON	71
2.8.4	读写StringIO	73
2.8.5	读写BytesIO	74
2.9	本章小结	74
	习题	75
第3章	TCP/IP 协议簇	76
3.1	TCP/IP 协议簇介绍	76
3.2	链路层	77
3.3	网络层	79
3.3.1	IPv4	79
3.3.2	IPv6	82

3.3.3	网络层协议	84
3.3.4	获取计算机 IP 地址实例	85
3.3.5	获取局域网网关地址实例	86
3.4	传输层	86
3.4.1	UDP	87
3.4.2	TCP	88
3.4.3	主机收发数据统计信息程序实例	89
3.5	应用层	90
3.5.1	HTTP	90
3.5.2	HTTPS	93
3.5.3	FTP	94
3.5.4	DNS	96
3.5.5	SMTP	99
3.5.6	POP3	100
3.5.7	DHCP	101
3.6	本章小结	104
	习题	104
第 4 章	Socket	105
4.1	Socket 介绍	105
4.2	SOCK_STREAM	108
4.2.1	字符串转换实例	108
4.2.2	文件下载实例	110
4.2.3	扫描主机端口实例	113
4.3	SOCK_DGRAM	113
4.3.1	获取服务器 CPU 使用情况实例	114
4.3.2	获取服务器内存使用情况实例	117
4.4	SOCK_RAW	119
4.4.1	ICMP 报文	119
4.4.2	ICMP 报文校验和计算	121
4.4.3	数据转换为 bytes 格式	121
4.4.4	探测主机是否在线实例	123
4.4.5	网络嗅探实例	125
4.5	本章小结	127
	习题	127
第 5 章	进程与线程	128
5.1	进程与线程介绍	128
5.2	多进程编程	129

5.2.1	多进程文件下载服务实例	129
5.2.2	进程池扫描主机端口实例	130
5.2.3	多进程返回服务器负载情况实例	132
5.3	多线程编程	134
5.3.1	多线程文件下载服务实例	134
5.3.2	线程池扫描主机端口实例	135
5.4	socketserver	137
5.4.1	多进程 TCP 实例	137
5.4.2	多进程 UDP 实例	139
5.4.3	多线程 TCP 与多线程 UDP	141
5.5	GUI 聊天室实例	142
5.5.1	Tkinter	142
5.5.2	服务器端程序	147
5.5.3	客户端程序	148
5.5.4	程序运行结果	151
5.6	本章小结	152
	习题	152
第 6 章	网络应用程序实例	153
6.1	网页内容获取	153
6.1.1	通过 API 获取天气数据实例	153
6.1.2	正则表达式	155
6.1.3	通过爬虫获取天气数据实例	156
6.1.4	通过爬虫下载网页中的图片实例	158
6.1.5	爬虫获取需要验证用户身份的网站信息实例	159
6.1.6	爬虫获取使用 HTTPS 网站信息实例	163
6.2	访问 FTP 服务器	166
6.2.1	搭建 FTP 服务器	166
6.2.2	访问 FTP 服务器的常用函数	166
6.2.3	访问 FTP 服务器程序实例	168
6.3	访问 DNS	169
6.3.1	DNS 记录类型	169
6.3.2	访问 DNS 程序实例	170
6.4	收发 E-mail	171
6.4.1	设置 QQ 邮箱授权码	171
6.4.2	简单邮件发送实例	172
6.4.3	HTML 格式邮件发送实例	175
6.4.4	带附件的邮件发送实例	176
6.4.5	带图片的邮件发送实例	177

6.4.6 邮件接收实例	178
6.5 获取 DHCP 信息	183
6.5.1 Scapy 简介及安装	183
6.5.2 获取 DHCP 信息程序实例	183
6.6 本章小结	185
习题	186
第 7 章 Web 应用程序开发	187
7.1 WSGI	187
7.2 Django	189
7.2.1 Django 安装与配置	190
7.2.2 SQLite3 数据库	192
7.2.3 向客户端回应简单信息	195
7.2.4 向客户端回应 HTML 文件	196
7.2.5 模板标签	197
7.2.6 框架实例	201
7.3 本章小结	210
习题	211
参考文献	212

Linux 系统介绍

Linux 是一款开源的操作系统软件,是开源软件的代表,以高效性和灵活性著称。在桌面操作系统领域,Windows 仍占据主导地位,但是,在其他的大多数领域,Linux 的主导地位不可撼动。在服务器领域,Linux 以其安全性和稳定性成为众多管理员的首选;在嵌入式领域,Linux 以其可裁剪、可定制和高效性,备受嵌入式系统开发者的青睐,目前最为流行的手机操作系统 Android 就是基于 Linux 内核开发而成。当前,越来越多的组织和个人选择 Linux 系统作为工作和开发平台。本章介绍 Linux 的诞生、特点、组成、应用、常见发行版本和安装等。

1.1 Linux 的诞生

Linux 是一个完整的多用户、多任务的类 UNIX 操作系统,可以运行在如 Intel、Alpha、Power PC、Sun Sparc、ARM 等多种硬件平台上。

Linux 操作系统的诞生、发展和成长过程始终依赖着五个重要因素:UNIX 操作系统、Minix 操作系统、GNU (GNU is Not Unix) 计划、POSIX (Portable Operating System Interface of UNIX) 标准和 Internet 网络。

UNIX 操作系统是美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 于 1969 年夏在 PDP7 小型计算机上开发的一个分时操作系统。当时使用的是 BCPL (Basic Combined Programming Language) 语言,后经 Dennis Ritchie 于 1973 年用移植性很强的 C 语言进行了改写,使得 UNIX 系统在大专院校得到了推广。但从版本 7 后,起源于贝尔实验室的 AT&T 公司为了商业利益,禁止在课程中研究 UNIX 源代码,使得 UNIX 的应用范围和用户群体大为缩减。

1987 年,荷兰 Amsterdam 的 Vrije 大学教授 Andrew Tanenbaum 为了方便教学,自己设计编写了一个在用户看来与 UNIX 完全兼容,但有全新内核的操作系统 Minix。Minix 主要是为教师进行教学研究和学生学习操作系统原理的目的而设计。为了能让学生在一个学期内就能学完并易于理解,Andrew Tanenbaum 教授没有接纳全世界许多人士对 Minix 进行扩展的要求,而坚定保持了 Minix 小型化的特点。

GNU 和 FSF (Free Software Foundation) 由 Richard Stallman 于 1984 年创办,旨在开发一个类似 UNIX,并且是开放源代码,完全免费的完整操作系统,其中,GNU 是 GNU is Not Unix 的递归缩写。到 20 世纪 90 年代初,GNU 已经开发出许多高质量的自由软件,其

中包括著名的 Emacs 编辑系统、BASH Shell 程序、GCC 系列编译程序、GDB 调试程序等。这些软件为 Linux 操作系统的开发创造了一个合适的环境,是 Linux 能够诞生的基础之一。以至于目前许多人将 Linux 操作系统称为 GNU/Linux 操作系统。

1991 年初,芬兰 University of Helsinki 的学生 Linus Torvalds 开始在一台 386SX 兼容微机上学习 Minix 操作系统。通过学习,他不再满意 Minix 系统的现有性能,并开始酝酿开发一个新的免费操作系统。从 1991 年的 4 月开始,Linus 几乎花了全部时间研究 386-Minix 系统,并且尝试着移植 GNU 的软件(GCC、BASH、GDB 等)到该系统上。到了 1991 年的 10 月 5 日,Linus 在 comp. os. minix 新闻组上发布消息,正式向外宣布 Linux 内核系统的诞生(Free minix-like kernel sources for 386-AT)。这段消息可以称为 Linux 的诞生宣言,并且一直广为流传。因此,10 月 5 日对 Linux 社区来说是一个特殊的日子,许多后来的 Linux 新版本的发布时间都选择了这个日子。

POSIX(Portable Operating System Interface for Computing Systems)是由 IEEE(Institute of Electrical and Electronics Engineers)和 ISO/IEC(International Organization for Standardization/International Electrotechnical Commission)开发的一簇标准。该标准基于现有的 UNIX 实践和经验,描述了操作系统的调用服务接口,用于保证编写的应用程序源代码可以在多种操作系统上移植运行。20 世纪 90 年代初,POSIX 标准的制定处在最后投票阶段的时间也是 Linux 刚刚起步的时间,使得 POSIX 这个为 UNIX 制定的标准,成了指导 Linux 开发的规范,导致 Linux 系统与 UNIX 系统的高度兼容。

伴随着 Internet 的发展,在 Linus 本人和许多自由软件开发者努力下,Linux 不断完善,使越来越多的人认识 Linux,越来越多的人使用 Linux。直到现在,Linus 仍然在从事 Linux 内核开发与维护工作。2014 年,Linus 获得 IEEE-CS(Computer Society)计算机先驱奖(For pioneering development of the Linux kernel using the open-source approach)。

1.2 Linux 的特点

Linux 之所以受到广大计算机从业人员的青睐,主要是因为 Linux 具有如下特点。

1. 开放性

由于 Linux 遵循 GPL(GNU General Public License)约定,使得其可以通过 Internet,由全球众多的自由软件爱好者维护。在 Linux 中,几乎所有的源代码都是开放的,包括核心程序、设备驱动程序等,用户可以根据自己的实际需要来定制模块、修改源码,使系统满足自己的个性化需求。这个特点吸引了大量的专业用户。

2. 多用户

在 Linux 系统中,可以创建多个用户账号,这些用户账号对相同资源(例如文件、服务等)具有不同的访问和操作权限,保证了用户对资源操作的个性。另外,多个用户可以同时登录到同一个 Linux 系统中同时工作,每个用户都能够按照自己的意愿定制工作环境,安排自己的桌面图标,访问操作权限许可下的文件,好像自己正在独占 Linux 系统。

3. 多任务

多任务是现代操作系统最主要的一个特点,一般将一个进程看作一个任务。在 Linux 系统中,单个用户就可以启动多个程序同时执行,使得系统内存在多个用户启动的众多程序同时执行。一个正在执行的程序可以形成一个或多个进程,使得同时执行的众多程序形成系统内的多个进程,而一个进程又可包括多个线程,因此,多任务的操作系统中同时存在多个进程和线程。Linux 系统按一定的策略调度这些进程和线程,使用户觉得这些进程和线程在同时执行,但实际上,这些进程和线程可能在相同的 CPU 上交替轮流执行,由于 CPU 的处理速度非常快,一般情况下用户感觉不到多个进程和线程的轮流执行。

4. 良好的图形用户界面

Linux 向用户提供了两种界面:字符界面和图形界面。Linux 的字符界面通过 Shell 实现,以高效、强大著称,其灵活多变的 Shell 脚本非常有利于对 Linux 服务器的管理,是 Linux 高级用户常用的界面。该界面除了支持命令行方式外,还具有很强的程序设计功能,通过编程,用户可通过程序调用系统提供的函数来实现相应的功能。

与 Windows 的图形化界面一样,Linux 也有自己的图形化界面,它主要由两部分组成:X-Window 系统以及 KDE、GNOME 或其他桌面环境(如 XFCE 等)。用户利用鼠标对其操作,给用户呈现一个直观、易操作、交互的图形化界面。与 Windows 系统不同,Linux 的图形界面仅仅是应用程序而不是系统的内核,因此,在启动 Linux 系统时,可选择不启动图形界面。

5. 设备的独立性

设备的独立性指系统屏蔽掉物理设备的具体细节,给用户提供统一的标准操作接口来使用设备,即系统给用户展现的是逻辑设备。用户通过标准操作接口使用设备,不需要了解设备的具体特性,由操作系统来完成逻辑设备到物理设备的映射。Linux 的所有设备都是以文件的方式命名,每一个设备是一个特殊类型的文件,用户访问设备就像访问文件一样方便。当增加新设备时,在系统内核中添加必要的驱动程序,以确保操作系统内核以合理的方式来操作这些设备。

Linux 的内核具有高度适应能力,已经包含了常用硬件的驱动程序。Linux 系统会自动识别、加载并管理硬件设备,供用户直接使用。对于驱动程序未包含在 Linux 系统中的设备,用户可以下载这些设备的驱动程序,并进行安装后即可使用。另外,由于 Linux 的内核源代码可以免费下载,高级用户可以通过修改内核源代码给系统添加新的设备,然后重新编译内核,使 Linux 系统能够自动识别和加载这些设备。

6. 丰富的网络功能

丰富并且完善的网络功能是 Linux 的一大特点。由于 Linux 与互联网相伴而生,因此, Linux 具有全套的网络服务,如 DNS(Domain Name Server)、FTP(File Transfer Protocol)、DHCP(Dynamic Host Configuration Protocol)等,与此同时,还提供了大量免费的 Internet 软件,例如,网络浏览器、FTP 工具、远程管理工具等,使用户可以方便地通过这些软件访问

Internet。此外, Linux 还向用户提供了远程访问工具软件, 例如, Telnet、SSH (Secure Shell)、VNC (Virtual Network Computer) 等, 用户可以通过这些工具软件, 远程登录到 Linux 系统中, 对 Linux 系统进行操作和维护。

7. 可靠的系统安全

Linux 采取了许多安全技术措施, 如文件读/写权限控制、用户授权、带保护的子系统、审计跟踪、核心授权等; 还有开放源代码, 大大减少了操作系统存在未知“后门”的可能性; 这些都为整个系统提供了必要的安全保障。

8. 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, 仍然能按其自身的方式运行的能力。Linux 符合 POSIX 标准, 具有良好的可移植性, 不仅可以运行在 Intel 系列 CPU 的计算机上, 还可以运行在 APPLE、AMD、ARM 等系列 CPU 的计算机上。

Linux 遵循标准的通信协议, 为符合标准通信协议的计算机之间的通信提供了丰富的实现手段, 且不需要额外增加特殊和昂贵的通信设备。

9. 丰富的应用软件支持

Linux 与 POSIX 标准及其他应用程序接口兼容, 因此, 包括 GNU 在内的大量免费或共享软件都能够在 Linux 上运行, 这些软件包括 Shell 类、编辑器类、编程工具类、数据库类、Internet 应用类、办公软件类、游戏类等。

10. 内核完全免费

Linux 的内核完全免费, 用户可以通过网络或其他途径获得, 并可以任意修改其源代码, 这是其他的操作系统不具备的特点。正是由于这一点, 来自全世界的无数程序员参与 Linux 的修改、编写工作, 根据自己的兴趣和灵感对其进行改变, 这让 Linux 吸纳了无数程序员的工作成果, 不断壮大。

1.3 Linux 的组成

Linux 一般由四个主要部分组成, 即内核、Shell、文件系统和实用工具。

1. Linux 内核

内核是 Linux 操作系统的核心, 是运行程序和管理像磁盘和打印机等硬件设备的核心程序。它负责管理系统中的进程、内存、设备驱动程序、文件和网络子系统, 决定整个系统的性能和稳定性。内核执行最底层任务, 协调多个并发进程的运行, 管理进程使用的内存, 满足进程访问磁盘的请求等。用户的各种操作请求和命令最终都要传递给内核执行。

内核不是一套完整的操作系统, 仅仅是 Linux 系统的核心模块。内核之上附加其他系统模块便形成了一套完整的 Linux 系统。

2. Linux Shell

Shell 是 Linux 系统的字符型用户操作界面,提供了用户与内核进行交互的命令接口。它接收用户输入的命令并将命令送入内核去执行,最后把内核执行的结果返回给用户。

Shell 既是一种命令解释器,它解释由用户输入的命令并把它们送到内核去执行;同时,它又是一种程序设计语言,可以定义各种变量和函数,并提供许多在高级语言中才具有的控制结构,包括循环和分支。Shell 虽然不是 Linux 系统核心的一部分,但它可以调用系统核心的功能来执行程序、创建文件并协调各个程序的运行。因此,对于用户来说,Shell 是最重要的实用程序,深入了解和熟练掌握 Shell 的特性和使用方法,是用好 Linux 系统的关键。可以说,Shell 使用的熟练程度反映了用户对 Linux 系统使用的熟练程度。

3. Linux 文件系统

文件系统是 Linux 系统的一个子系统,是文件存放在磁盘等存储设备上的组织方法,主要体现在对文件和目录的组织上。Linux 使用标准的多级树形目录结构,用户可以浏览整个目录树,进入任何一个已授权的目录,并访问其中的文件。Linux 文件系统提供用户设置目录和文件权限的功能,也能够按照事先设定的权限,允许或拒绝用户对文件或目录的访问,同时,还可以提供文件共享功能,实现多个用户对同一个文件进行操作。

在安装 Linux 时,安装程序就已经为用户创建了文件系统和完整而固定的目录,并指定了每个目录的作用和其中存放的文件,例如“/dev”目录存放设备文件,“/etc”目录存放配置文件等。

内核、Shell 和文件系统一起形成了基本的操作系统结构,它们使得用户可以运行程序,管理文件和使用系统。此外,Linux 还有许多实用工具,辅助用户完成一些特定的任务。

4. Linux 实用工具

每个版本的 Linux 系统都有一套自己的实用工具集,一般包括编辑器、过滤器、交互程序、网络工具等。

- 编辑器。用于编辑文件,Linux 的编辑器主要有 vi、ed、ex 和 Emacs。
- 过滤器。用于接收并过滤数据,Linux 的过滤器读取从用户文件或其他地方(如来自键盘)的输入,检查和处理数据,然后输出结果。过滤器可以相互连接,一个过滤器的输出可能是另一个过滤器的输入。用户可以根据需要编写自己的过滤器。
- 交互程序。允许用户发送信息或接收来自其他用户的信息,交互程序是用户与计算机的信息接口。Linux 是一个多用户系统,它必须和所有用户保持联系,实现信息的发送或接收。信息的发送有两种方式:一种方式是用户一对一地建立连接进行对话,另一种方式是一个用户对多个用户建立连接进行通信,即所谓分组或广播式通信。
- 网络工具。如网络浏览器、远程连接、桌面共享、数据下载等工具软件。

1.4 Linux 的应用

自 Linux 诞生以来,得到了世界上数以万计的编程高手和计算机爱好者的共同开发和维护,新功能不断增加,应用范围不断扩展,大大推动了 Linux 的发展。如今, Linux 已经成为一个稳定可靠、功能完善、性能卓越的操作系统。在目前的市场上, Linux 的占有率越来越高,已经对 Windows 系列操作系统造成了很大的冲击。各国政府和企事业单位越来越多地采用 Linux 系统,在一些特定的领域,如服务器端、集群计算机、嵌入式系统等, Linux 都占据主导地位。

1. 桌面系统应用

目前, Linux 桌面操作系统的性能有了很大提高。 Linux 的文字处理、图片编辑、办公软件、网络通信、多媒体工具等有了长足的发展,并且具有一个能够与 Microsoft Office 相媲美的 Open Office 办公应用软件。

从 Linux 桌面系统所涉及的行业来看,其使用范围已经逐渐扩展到各行各业,如政府、教育、金融、制造业等,尤其在国内电子政务的发展上,国家已经明确提出要以 Linux 为核心平台,采用以 Linux 为主的解决方案。

2. 服务器端应用

Linux 由于具有运行稳定、安全性好、性能卓越、易于维护等特点,被广泛应用在服务器端,现在的云计算平台绝大多数采用 Linux 系统。 Linux 在服务器端领域的市场份额已经超过三分之二,伴随着计算机技术的发展, Linux 一定会继续占据绝对重要的地位。

3. 嵌入式系统

随着应用领域的不断扩大,为了适应不同的应用场合,考虑到系统的灵活性、可伸缩性以及可裁剪性,一种以应用为中心,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、功耗等严格要求的专用计算机系统——嵌入式系统便应运而生。

嵌入式系统的涵盖面非常广泛,其中家电市场包括机顶盒、智能电视、可视电话、家庭网络等信息家电;工业市场包括工业控制设备、智能仪器、智能仪表;商用市场包括智能手机、平板电脑、POS 终端、可穿戴设备等。

由于 Linux 具有开放源代码、内核可按需裁剪、支持多种硬件平台、应用软件丰富等优点,使得 Linux 成为嵌入式系统的首选操作系统。目前广泛运行在智能手机上的 Android 系统就是基于 Linux 系统开发而来。

4. 集群计算机

集群就是利用商品化的工业标准互连网络,将各种服务器连接起来,通过特定的方法,向用户提供更高的系统计算性能、存储性能和 I/O 性能,并具备单一系统映像(Single System Image, SSI)特征的分布式/并行计算机系统。与对称多处理系统(Symmetrical Multi-Processing, SMP)、大规模并行处理(Massively Parallel Processing, MPP)及 Beowulf

集群相比,采用 Linux 的集群在性价比、可靠性、可扩展性、可管理性和应用支持性等方面有着更为明显的优势。著名的搜索引擎 Google 就是在 Linux 集群平台上实现的。

1.5 常见 Linux 发行版本

Linux 的版本号分为两部分:内核(Kernel)与发行套件(Distribution)版本。内核版本是指在 Linus 领导下的开发小组开发出的某个版本的系统内核;而发行版本指的是一些组织或厂家将 Linux 的内核与应用软件和文档包装起来,并提供安装界面、系统设定与管理工具、应用软件等而形成的 Linux 发行套件版本,例如,Ubuntu、Red Hat 等,Linux 的内核是开源并且免费的,但 Linux 的发行套件不一定开源和免费。

Linux 的发行版本实际上是一个大的软件包,核心软件就是 Linux 内核。Linux 发行套件的版本号与 Linux 内核版本号是相对独立的,不同的 Linux 发行套件所包含的 Linux 内核一般是不同的,例如,Ubuntu 14.04 所包含的 Linux 内核为 4.4,Ubuntu 17.04 所包含的 Linux 内核为 4.10。用户可通过命令“uname -a”查看 Linux 的内核版本号。

Linux 的各种发行套件版本大约有 300 多种,下面是常见的 Linux 发行套件版本。

1. Mandriva

原名 Mandrake,最早由 Gaël Duval 创建并在 1998 年 7 月发布。Mandriva Linux 率先采用 KDE 桌面,并简化 Linux 的安装过程,具有友好的操作界面、图形配置工具、庞大的社区技术支持。

官方主页: <http://www.mandrivalinux.com/>

2. Red Hat

全世界的 Linux 用户所最熟悉的发行版,由 Bob Young 和 Marc Ewing 在 1995 年创建。从 Red Hat 9.0 发行版后,Red Hat 分为两个系列:由 Red Hat 公司提供收费技术支持和更新的 Red Hat Enterprise Linux,以及由社区开发的免费的 Fedora Core。Red Hat 拥有数量庞大的用户,优秀的社区技术支持。

官方主页: <http://www.redhat.com/>

3. SUSE

SUSE 是德国最著名的 Linux 发行版,在全世界范围内享有较高的声誉,SUSE 于 2003 年年末被 Novell 收购。SUSE Linux 适合于专业用户,具有易用的 YaST 软件包管理系统。

官方主页: <http://www.suse.com/>

4. Debian GNU/Linux

Debian 最早由 Ian Murdock 于 1993 年创建,是完全遵循 GNU 规范的 Linux 系统。Debian 有三个版本分支: stable、testing 和 unstable,这三个版本分别对应的具体版本为: Woody、Sarge 和 Sid。其中,unstable 为最新的测试版本,其中包括最新的软件包,但是也有相对较多的 bug,适合桌面用户;testing 的版本已经通过 unstable 版本的测试,相对较为稳

定; Woody 一般只用于服务器, 上面的软件包大部分都比较成熟, 因此稳定和安全性都比较高。Debian 遵循 GNU 规范, 拥有优秀的网络和社区资源。

官方主页: <http://www.debian.org/>

5. Ubuntu

Ubuntu 基于 Debian Sid 开发, 拥有 Debian 的所有优点, 安装简便, 被誉为对硬件支持最好、最全面的 Linux 发行版。Ubuntu 采用自行加强的内核, 具有优秀的安全性能, 版本更新速度快, 且采用软件包的在线安装与更新, 深得用户喜爱, 是使用最为广泛的 Linux 发行版。

官方主页: <https://www.ubuntu.com/>

6. Gentoo

Gentoo 最初由 Daniel Robbins(Stampede Linux 和 FreeBSD 的开发者之一)创建, 具有高度的可定制性和完整的使用手册。

官方主页: <http://www.gentoo.org/>

7. Slackware

Slackware 由 Patrick Volkerding 创建于 1992 年, 非常稳定、安全, 高度坚持 UNIX 的规范。

官方主页: <http://www.slackware.com/>

8. Knoppix

由德国的 Klaus Knopper 开发, 是一个基于 Debian 的发行版, 无须安装, 可直接运行于 CD 上, 具有优秀的硬件检测能力, 可作为系统急救盘使用。

官方主页: <http://www.knoppix.com/>

9. MEPIS

由 Warren Woodford 在 2003 年建立, 集合了 Debian Sid 和 Knoppix 的特点, 用户既能将它当作 LiveCD 使用, 也能使用常规的图形界面进行安装, 具有优秀的硬件检测能力, 预装了许多实用的软件。

官方主页: <http://www.mepis.org/>

10. Xandros

Xandros 建立在 Corel Linux 之上, 当初 Corel Linux 公司由于财政上的困难, 被迫终止了 Corel Linux 的开发, 此时, Xandros 适时地将 Corel Linux 部门买下, 于 2002 年 10 月推出全新的 Xandros Desktop。Xandros 的特点在于它极其简单的安装和使用, 所以它的市场定位是那些没有任何 Linux 使用经验的新手, 或是习惯使用 Windows 的用户。

官方主页: <http://www.xandros.com/>