

Seventh Edition

- Set up, maintain, and secure the latest distributions of Linux
- Configure and manage file systems, as well as e-mail, FTP, DNS, web, VoIP, LDAP, and print servers
- Build and deploy virtual machines, containers, and servers

Linux Administration *A Beginner's Guide*

Wale Soyinka

**Mc
Graw
Hill**
Education



Linux Administration

A Beginner's Guide Seventh Edition

WALE SOYINKA



New York Chicago San Francisco
Athens London Madrid Mexico City
Milan New Delhi Singapore Sydney Toronto

Copyright © 2016 by McGraw-Hill Education. All rights reserved. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

ISBN: 978-0-07-184537-3

MHID: 0-07-184537-2

The material in this eBook also appears in the print version of this title: ISBN: 978-0-07-184536-6,
MHID: 0-07-184536-4.

eBook conversion by codeMantra
Version 1.0

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial caps.

McGraw-Hill Education eBooks are available at special quantity discounts to use as premiums and sales promotions or for use in corporate training programs. To contact a representative, please visit the Contact Us page at www.mhprofessional.com.

Information has been obtained by McGraw-Hill Education from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, McGraw-Hill Education, or others, McGraw-Hill Education does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from the use of such information.

TERMS OF USE

This is a copyrighted work and McGraw-Hill Education and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill Education's prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED "AS IS." MCGRAW-HILL EDUCATION AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill Education and its licensors do not warrant or guarantee that the functions contained in the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill Education nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill Education has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill Education and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

Dedicated to everyone who has contributed to open source technologies and ideals in one form or another. Without you, we would have nothing to write about in this book.

About the Author

Wale Soyinka is a systems/network engineering consultant and has written a decent library of Linux administration training materials. In addition to several editions of *Linux Administration: A Beginner's Guide*, he is the author of *Wireless Network Administration: A Beginner's Guide* and a projects lab manual, *Microsoft Windows 2000 Managing Network Environments* (Prentice Hall). Wale participates in several open source discussions, projects, and ventures—usually all centered around promoting and showcasing open source technologies and culture.

About the Contributor

Buki Adeniji is a seasoned telephony and telecomms professional who is extremely passionate about Linux, VoIP, and Asterisk, and has many global deployments under his belt. His core skill set was developed by working on key telephone systems (KTS), after which he mastered TDM/IP PBXs, including the Ericsson MD110, the most advanced TDM PBX at that time. Buki also has experience building and managing high-availability carrier-grade VoIP platforms. He designed interactive voice response (IVR) and other voice-based solutions while heading global operations teams that managed mission-critical cloud-based services, achieving 99.99 percent (“four nines”) service availability. Being a lifetime learner with advanced degrees in Telecomms, Business, and Project Management, Buki’s goal is to continue to excel at the boundaries of technology and innovation by using principles of ethnography and design thinking to develop technical solutions that address current and next-generation business needs. Above all, Buki is passionate about converting complex concepts and procedures into simple, concise, and executable steps. Buki currently resides in the tech hub of the Bay Area, where he is a passionate family man, as well as a maker who loves to explore the gadget world and the outdoors.

About the Technical Editor

David Lane is an infrastructure architect and IT manager working and living in the Washington, DC, area. He has been working with open source software since the early 1990s and was introduced to Linux via the Slackware distribution early in its development. David soon discovered Red Hat 3 and has never looked back. Unlike most Linux people, David does not have a programming background and fell into IT as a career after discovering he was not cut out for sleeping on the street. He has implemented Linux solutions for a variety of government and private companies with solutions ranging from the simple to the complex. In his spare time, David writes about open source issues, especially those related to business, for the *Linux Journal*, as well as championing Linux to the next generation. David is an amateur radio operator and Emergency Coordinator for amateur radio responders for his local county. David speaks regularly to both Linux and amateur radio user groups about the synergies between open source and amateur radio.



At a Glance

PART I Introduction, Installation, and Software Management

1	Technical Summary of Linux Distributions	3
2	Installing Linux in a Server Configuration	17
3	The Command Line	47
4	Managing Software	87

PART II Single-Host Administration

5	Managing Users and Groups	121
6	Booting and Shutting Down	153
7	File Systems	181
8	Core System Services	213
9	The Linux Kernel	245
10	Knobs and Dials: API (Virtual) File Systems	269

PART III Networking and Security		
11	TCP/IP for System Administrators	287
12	Network Configuration	329
13	Linux Firewall (Netfilter)	355
14	Local Security	387
15	Network Security	405
PART IV Internet Services		
16	Domain Name System (DNS)	421
17	File Transfer Protocol (FTP)	459
18	Apache Web Server	479
19	Simple Mail Transfer Protocol (SMTP) ...	499
20	Post Office Protocol and Internet Mail Access Protocol (POP and IMAP)	513
21	Voice over Internet Protocol (VoIP)	535
22	Secure Shell (SSH)	581
PART V Intranet Services		
23	Network File System (NFS)	603
24	Samba	623
25	Distributed File Systems (DFS)	643
26	Network Information Service (NIS)	655
27	Lightweight Directory Access Protocol (LDAP)	681
28	Printing	703
29	Dynamic Host Configuration Protocol (DHCP)	721
30	Virtualization	735
31	Backups	755
PART VI Appendixes		
A	Creating a Linux Installer on Flash/USB Devices	773
B	Demo Virtual Machine	785
	Index	803



Contents

Acknowledgments	xxiii
Introduction	xxv

PART I

Introduction, Installation, and Software Management

1 Technical Summary of Linux Distributions	3
Linux: The Operating System	4
What Is Open Source Software and GNU All About?	5
What Is the GNU Public License?	7
Upstream and Downstream	8
The Advantages of Open Source Software	9
Understanding the Differences Between Windows and Linux ...	11
Single Users vs. Multiple Users vs. Network Users	11
The Monolithic Kernel and the Micro-Kernel	12
Separation of the GUI and the Kernel	12
My Network Places	14
The Registry vs. Text Files	15
Domains and Active Directory	15
Summary	16

2 Installing Linux in a Server Configuration	17
Hardware and Environmental Considerations	18
Server Design	19
Uptime	20
Methods of Installation	20
Installing Fedora	21
Project Prerequisites	21
The Installation	23
Installation Summary	24
Localization Section	24
Software Section	25
System Section	26
Start the Installation, Set the Root Password, and Create a User Account	39
Complete the Installation	40
Log In	41
Installing Ubuntu Server	41
Start the Installation	42
Configure the Network	43
Set Up Users and Passwords	43
Configure the Time Zone	43
Set Up the Disk Partition	44
Other Miscellaneous Tasks	45
Summary	46
3 The Command Line	47
An Introduction to Bash	48
Job Control	49
Environment Variables	50
Pipes	52
Redirection	53
Command-Line Shortcuts	53
Filename Expansion	53
Environment Variables as Parameters	54
Multiple Commands	54
Backticks	55
Documentation Tools	56
The man Command	56
The texinfo System	58
Files, File Types, File Ownership, and File Permissions	58
Normal Files	58
Directories	58
Hard Links	59
Symbolic Links	59
Block Devices	59
Character Devices	59
Named Pipes	60
Listing Files: ls	60

Change Ownership: chown	61
Change Group: chgrp	61
Change Mode: chmod	62
File Management and Manipulation	64
Copy Files: cp	65
Move Files: mv	65
Link Files: ln	66
Find a File: find	66
File Compression: gzip	67
File Compression: bzip2	68
File Compression: xz	68
Create a Directory: mkdir	68
Remove a Directory: rmdir	69
Show Present Working Directory: pwd	69
Tape Archive: tar	69
Concatenate Files: cat	71
Display a File One Screen at a Time: more	72
Show the Directory Location of a File: which	72
Locate a Command: whereis	72
Editors	73
vi	73
emacs	74
joe	74
pico	74
Miscellaneous Tools	74
Disk Utilization: du	75
Disk Free: df	75
Synchronize Disks: sync	76
List Processes: ps	76
Show an Interactive List of Processes: top	77
Send a Signal to a Process: kill	79
Show System Name: uname	81
Who Is Logged In: who	82
A Variation on who: w	82
Switch User: su	82
Putting It All Together (Moving a User and Its Home Directory)	83
Summary	86
4 Managing Software	87
The Red Hat Package Manager	88
Managing Software Using RPM	91
Querying for Information the RPM Way (Getting to Know One Another)	91
Installing Software with RPM (Moving in Together)	94
Uninstalling Software with RPM (Ending the Relationship)	98
Other Things RPM Can Do	99
Yum	102

DNF	103
GUI RPM Package Managers	104
Fedora	104
openSUSE and SLE	104
The Debian Package Management System	105
APT	106
Software Management in Ubuntu	106
Querying for Information	107
Installing Software in Ubuntu	107
Removing Software in Ubuntu	108
Compile and Install GNU Software	110
Getting and Unpacking the Source Package	111
Looking for Documentation	113
Configuring the Package	113
Compiling the Package	114
Installing the Package	115
Testing the Software	116
Cleanup	116
Common Problems When Building from Source Code	117
Problems with Libraries	117
Missing Configure Script	118
Broken Source Code	118
Summary	118

PART II

Single-Host Administration

5 Managing Users and Groups	121
What Exactly Constitutes a User?	122
Where User Information Is Kept	123
The /etc/passwd File	123
The /etc/shadow File	128
The /etc/group File	129
User Management Tools	130
Command-Line User Management	130
GUI User Managers	136
Users and Access Permissions	138
Understanding SetUID and SetGID Programs	138
Sticky Bit	139
Pluggable Authentication Modules	140
How PAM Works	140
PAM's Files and Their Locations	141
Configuring PAM	142
An Example PAM Configuration File	144
The "Other" File	146
D'oh! I Can't Log In!	146
Debugging PAM	147

A Grand Tour	147
Creating Users with useradd	147
Creating Groups with groupadd	149
Modifying User Attributes with usermod	150
Modifying Group Attributes with groupmod	150
Deleting Users and Groups with userdel and groupdel	151
Summary	151
6 Booting and Shutting Down	153
Boot Loaders	154
GRUB Legacy	154
GRUB 2	159
LILO	165
Bootstrapping	166
The init Process	167
rc Scripts	168
Writing Your Own rc Script	169
Enabling and Disabling Services	174
Enabling a Service	174
Disabling a Service	175
Graphical Service Managers	175
Odds and Ends of Booting and Shutting Down	178
fsck!	178
Booting into Single-User (“Recovery”) Mode	179
Summary	180
7 File Systems	181
The Makeup of File Systems	182
i-Nodes	182
Blocks	182
Superblocks	183
ext3	184
ext4	185
Btrfs	186
XFS	186
Which File System Should You Use?	187
Managing File Systems	187
Mounting and Unmounting Local Disks	188
Using fsck	193
Adding a New Disk	195
Overview of Partitions	196
Traditional Disk and Partition Naming Conventions	196
Volume Management	197
Creating Partitions and Logical Volumes	198
Creating File Systems	208
Summary	210

8 Core System Services	213
The init Daemon	214
upstart: Die init. Die Now!	215
xinetd and inetd	222
The /etc/xinetd.conf File	224
Examples: A Simple Service Entry and Enabling/Disabling a Service	228
The Logging Daemon	230
rsyslogd	230
systemd-journald (journald)	238
The cron Program	240
The crontab File	241
Editing the crontab File	242
Summary	243
9 The Linux Kernel	245
What Exactly Is a Kernel?	246
Finding the Kernel Source Code	248
Getting the Correct Kernel Version	248
Unpacking the Kernel Source Code	249
Building the Kernel	250
Preparing to Configure the Kernel	251
Kernel Configuration	253
Compiling the Kernel	257
Installing the Kernel	259
Booting the Kernel	261
The Author Lied—It Didn't Work!	262
Patching the Kernel	263
Downloading and Applying Patches	263
If the Patch Worked	266
If the Patch Didn't Work	266
Summary	267
10 Knobs and Dials: API (Virtual) File Systems	269
What's Inside the /proc Directory?	270
Tweaking Files Inside of /proc	271
Some Useful /proc Entries	272
Enumerated /proc Entries	274
Common proc Settings and Reports	274
SYN Flood Protection	276
Issues on High-Volume Servers	277
Debugging Hardware Conflicts	277
SysFS	277
cgroupfs	280
tmpfs	281
tmpfs Example	282
Summary	283

PART III**Networking and Security**

11 TCP/IP for System Administrators	287
The Layers	288
Packets	288
TCP/IP Model and the OSI Model	291
Headers	295
Ethernet	295
IP (IPv4)	297
TCP	300
UDP	303
A Complete TCP Connection	304
Opening a Connection	304
Transferring Data	306
Closing the Connection	307
How ARP Works	307
The ARP Header: ARP Works with Other Protocols, Too!	309
Bringing IP Networks Together	310
Hosts and Networks	310
Subnetting	311
Netmasks	312
Static Routing	313
Dynamic Routing with RIP	315
tcpdump Bits and Bobs	320
Reading and Writing Dumpfiles	321
Capturing More or Less per Packet	322
Performance Impact	322
Don't Capture Your Own Network Traffic	322
Troubleshooting Slow Name Resolution (DNS) Issues	323
IPv6	324
IPv6 Address Format	324
IPv6 Address Types	325
IPv6 Backward-Compatibility	326
Summary	327
12 Network Configuration	329
Modules and Network Interfaces	330
Network Device Configuration Utilities (ip, ifconfig, and nmcli)	332
Sample Usage—ifconfig, ip, and nmcli	334
Setting Up NICs at Boot Time	339
Managing Routes	343
Sample Usage: Route Configuration	343
Displaying Routes	345
A Simple Linux Router	347
Routing with Static Routes	348

How Linux Chooses an IP Address	351
Hostname Configuration	351
Summary	353
13 Linux Firewall (Netfilter)	355
How Netfilter Works	356
A NAT Primer	358
NAT-Friendly Protocols	361
Chains	361
Installing Netfilter	364
Enabling Netfilter in the Kernel	365
Configuring Netfilter	368
Saving Your Netfilter Configuration	368
The iptables Command	370
firewalld	378
Cookbook Solutions	380
Simple NAT: iptables	381
Simple NAT: nftables	382
Simple Firewall: iptables	382
Summary	385
14 Local Security	387
Common Sources of Risk	389
SetUID Programs	389
Unnecessary Processes	391
Picking the Right Runlevel	393
Nonhuman User Accounts	393
Limited Resources	394
Mitigating Risk	396
chroot	396
SELinux	400
AppArmor	401
Monitoring Your System	401
Logging	402
Using ps and netstat	402
Using df	402
Automated Monitoring	403
Mailing Lists	403
Summary	403
15 Network Security	405
TCP/IP and Network Security	406
The Importance of Port Numbers	406
Tracking Services	407
Using the netstat Command	407
Security Implications of netstat's Output	408

Binding to an Interface	409
Shutting Down Services	410
Shutting Down xinetd and inetd Services	411
Shutting Down Non-xinetd Services	412
Monitoring Your System	412
Making the Best Use of syslog	413
Monitoring Bandwidth with MRTG	414
Handling Attacks	414
Trust Nothing (and No One)	414
Change Your Passwords	415
Pull the Plug	415
Network Security Tools	415
nmap	416
Snort	416
Nessus	416
Wireshark/tcpdump	417
Summary	417

PART IV

Internet Services

16 Domain Name System (DNS)	421
The Hosts File	422
How DNS Works	423
Domain and Host Naming Conventions	423
The Root Domain	424
Subdomains	426
The in-addr.arpa Domain	427
Types of Servers	428
Installing a DNS Server	430
Understanding the BIND Configuration File	432
The Specifics	432
Configuring a DNS Server	435
Defining a Primary Zone in the named.conf File	435
Defining a Secondary Zone in the named.conf File	437
Defining a Caching Zone in the named.conf File	437
DNS Records Types	438
SOA: Start of Authority	438
NS: Name Server	440
A: Address Record	440
PTR: Pointer Record	440
MX: Mail Exchanger	441
CNAME: Canonical Name	442
RP and TXT: The Documentation Entries	442
Setting Up BIND Database Files	443
DNS Server Setup Walk-Through	444

The DNS Toolbox	449
host	449
dig	450
nslookup	451
whois	452
nsupdate	453
The rndc Tool	453
Configuring DNS Clients	454
The Resolver	454
Configuring the Client	457
Summary	457
17 File Transfer Protocol (FTP)	459
The Mechanics of FTP	460
Client/Server Interactions	460
Obtaining and Installing vsftpd	462
Configuring vsftpd	462
Starting and Testing the FTP Server	467
Customizing the FTP Server	470
Setting Up an Anonymous-Only FTP Server	471
Setting Up an FTP Server with Virtual Users	472
Summary	477
18 Apache Web Server	479
Understanding HTTP	480
Headers	480
Ports	481
Process Ownership and Security	482
Installing the Apache HTTP Server	483
Apache Modules	485
Starting Up and Shutting Down Apache	486
Starting Apache at Boot Time	487
Testing Your Installation	488
Configuring Apache	489
Creating a Simple Root-Level Page	489
Apache Configuration Files	489
Common Configuration Options	490
Troubleshooting Apache	497
Summary	498
19 Simple Mail Transfer Protocol (SMTP)	499
Understanding SMTP	500
Rudimentary SMTP Details	500
Security Implications	502
Email Components	502
Installing the Postfix Server	503
Installing Postfix via RPM in Fedora	503

Installing Postfix via APT in Ubuntu	504
Installing Postfix from Source Code	505
Configuring the Postfix Server	507
The main.cf File	507
Checking Your Configuration	509
Running the Server	510
Checking the Mail Queue	510
Flushing the Mail Queue	511
The newaliases Command	511
Making Sure Everything Works	511
Summary	512
20 Post Office Protocol and Internet Mail Access	
Protocol (POP and IMAP)	513
POP3 and IMAP Protocol Basics	516
Dovecot (IMAP and POP3 Server)	517
Installing Dovecot	517
Dovecot Configuration Files and Options	519
Configuring Dovecot	520
Running Dovecot	525
Checking Basic POP3 Functionality	527
Checking Basic IMAP Functionality	527
Other Issues with Mail Services	529
SSL Security	529
Availability	532
Log Files	533
Summary	533
21 Voice over Internet Protocol (VoIP)	535
VoIP Overview	536
VoIP Server	537
Analog Telephone Adapter (ATA)	537
IP Phones	537
VoIP Protocols	538
VoIP Implementations	541
Asterisk	542
How Asterisk Works	542
Asterisk Installation	542
Starting and Stopping Asterisk	544
Understanding Asterisk Configuration Files and Structure	545
SIP Channel Config: sip.conf	545
The Dialplan: extensions.conf	550
Modules: modules.conf	552
Asterisk Network, Port, and Firewall Requirements	555
Configuring the Local Firewall for Asterisk	556
Configuring the PBX	557
Local Extensions	558
Outside Connection—(VoIP Trunking)	565
Trunking Using Google Voice	566

Asterisk Maintenance and Troubleshooting	576
Asterisk CLI Commands	576
Helpful CLI Commands	576
Common Issues with VoIP	577
Summary	579
22 Secure Shell (SSH)	581
Understanding Public Key Cryptography	582
Key Characteristics	584
Cryptography References	585
Understanding SSH Versions	585
OpenSSH and OpenBSD	586
Alternative Vendors for SSH Clients	586
Installing OpenSSH via RPM in Fedora	588
Installing OpenSSH via APT in Ubuntu	588
Server Start-up and Shutdown	591
SSHD Configuration File	592
Using OpenSSH	593
Secure Shell (ssh) Client Program	593
Secure Copy (scp) Program	598
Secure FTP (sftp) Program	598
Files Used by the OpenSSH Client	599
Summary	599

PART V

Intranet Services

23 Network File System (NFS)	603
The Mechanics of NFS	604
Versions of NFS	605
Security Considerations for NFS	606
Mount and Access a Partition	606
Enabling NFS in Fedora, RHEL, and Centos	607
Enabling NFS in Ubuntu	608
The Components of NFS	609
Kernel Support for NFS	610
Configuring an NFS Server	610
The /etc/exports Configuration File	610
Configuring NFS Clients	614
The mount Command	614
Soft vs. Hard Mounts	616
Cross-Mounting Disks	617
The Importance of the intr Option	617
Performance Tuning	617
Troubleshooting Client-Side NFS Issues	618
Stale File Handles	618
Permission Denied	618

Sample NFS Client and NFS Server Configuration	619
Common Uses for NFS	621
Summary	622
24 Samba	623
The Mechanics of SMB	624
Usernames and Passwords	624
Encrypted Passwords	625
Samba Daemons	626
Installing Samba via RPM	627
Installing Samba via APT	627
Samba Administration	629
Starting and Stopping Samba	630
Creating a Share	630
Using smbclient	632
Mounting Remote Samba Shares	635
Samba Users	636
Creating Samba Users	637
Allowing Null Passwords	637
Changing Passwords with smbpasswd	638
Using Samba to Authenticate Against a Windows Server	638
winbindd Daemon	639
Troubleshooting Samba	641
Summary	642
25 Distributed File Systems (DFS)	643
DFS Overview	644
DFS Implementations	647
GlusterFS	649
Summary	654
26 Network Information Service (NIS)	655
Inside NIS	656
The NIS Servers	657
Domains	658
Configuring the Master NIS Server	658
Establishing the Domain Name	659
Starting NIS	660
Editing the Makefile	661
Using ypinit	664
Configuring an NIS Client	667
Install NIS Client-Side Package	667
Editing the /etc/yp.conf File	668
Enabling and Starting ypbind	669
Editing the /etc/nsswitch.conf File	669
NIS at Work	672
Testing Your NIS Client Configuration	674
Configuring a Secondary NIS Server	674
Setting the Domain Name	674

Setting Up the NIS Master to Push to Slaves	675
Running ypinit	675
NIS Tools	676
Using NIS in Configuration Files	677
Implementing NIS in a Real Network	678
A Small Network	678
A Segmented Network	678
Networks Bigger than Buildings	679
Summary	679
27 Lightweight Directory Access Protocol (LDAP)	681
LDAP Basics	682
LDAP Directory	683
Client/Server Model	684
Uses of LDAP	684
LDAP Terminology	685
OpenLDAP	685
Server-Side Daemons	686
OpenLDAP Utilities	686
Installing OpenLDAP	687
Configuring OpenLDAP	688
Configuring slapd	690
Starting and Stopping slapd	693
Configuring OpenLDAP Clients	694
Creating Directory Entries	695
Searching, Querying, and Modifying the Directory	697
Using OpenLDAP for User Authentication	698
Configuring the Server	699
Configuring the Client	700
Summary	702
28 Printing	703
Printing Terminologies	704
The CUPS System	705
Running CUPS	705
Installing CUPS	705
Configuring CUPS	707
Adding Printers	708
Local Printers and Remote Printers	708
Using the Web Interface to Add a Printer	710
Using Command-Line Tools to Add a Printer	713
Routine CUPS Administration	714
Setting the Default Printer	714
Enabling, Disabling, and Deleting Printers	714
Accepting and Rejecting Print Jobs	715
Managing Printing Privileges	715
Managing Printers via the Web Interface	716
Using Client-Side Printing Tools	717
lpr	717

lpq	718
lprm	718
Summary	719
29 Dynamic Host Configuration Protocol (DHCP)	721
The Mechanics of DHCP	722
The DHCP Server	723
Installing DHCP Software via RPM	723
Installing DHCP Software via APT in Ubuntu	724
Configuring the DHCP Server	725
A Sample dhcpd.conf File	731
The DHCP Client Daemon	733
Configuring the DHCP Client	733
Summary	734
30 Virtualization	735
Why Virtualize?	736
Virtualization Concepts	737
Virtualization Implementations	738
Hyper-V	738
Kernel-Based Virtual Machine (KVM)	738
QEMU	739
User-Mode Linux (UML)	739
VirtualBox	739
VMware	739
Xen	739
KVM	740
KVM Example	741
Managing KVM Virtual Machines	744
Setting Up KVM in Ubuntu/Debian	745
Containers	748
Containers vs. Virtual Machines	749
Docker	749
Summary	753
31 Backups	755
Evaluating Your Backup Needs	756
Amount of Data	756
Backup Hardware and Backup Medium	757
Network Throughput	758
Speed and Ease of Data Recovery	759
Data Deduplication	759
Tape Management	760
Command-Line Backup Tools	763
dump and restore	764
tar	768
rsync	769
Miscellaneous Backup Solutions	769
Summary	770

PART VI**Appendixes**

A Creating a Linux Installer on Flash/USB Devices	773
Creating a Linux Installer on Flash/USB Devices (via Linux OS)	774
Creating a Linux Installer on Flash/ USB Devices (via Microsoft Windows)	776
Fedora Installer Using Live USB Creator on Windows	777
Ubuntu Installer Using UNetbootin on Windows	779
OpenSUSE Installer Using Pendrivelinux.com's Universal USB Installer on Windows	781
B Demo Virtual Machine	785
Basic Host System Requirements	787
Installing the Virtualization Applications and Utilities	788
Download and Prep the Demo VM Image File	789
Import the Demo VM Image and Create a New VM Instance	791
Managing the Demo Virtual Machine	791
Connecting to the Demo VM	794
Virtual Network Computing (VNC)	794
Connecting via SSH	796
Virtual Serial TTY Console	797
Cockpit Application	798
Just Use It!	801
Feedback	802
Index	803

Acknowledgments

My acknowledgment list is a very long and philosophical one. It includes everybody who has ever believed in me and provided me with one opportunity or another to experience various aspects of my life up to this point. It includes everybody I have ever had any kind of direct or indirect contact with. It includes everyone I have ever had a conversation with. It includes everybody I have ever looked at. It includes everyone who has ever given to or taken away from me. You have all contributed to and enriched my life. I am me because of you. You know who you are, and I thank you.

This page intentionally left blank

Introduction

On October 5, 1991, Linus Torvalds posted this message to the news-group comp.os.minix:

Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on an OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-)

Linus went on to introduce the first cut of Linux to the world. Unbeknown to him, he had unleashed what was to become one of the world's most popular and disruptive operating systems. More than 24 years later, an entire industry has grown up around Linux. And, chances are, you've probably already used it (or benefitted from it) in one form or another!

Who Should Read This Book

A part of the title of this book reads "A Beginner's Guide"; this is mostly apt. But what the title should say is "A Beginner to Linux Administration Guide," because we do make a few assumptions about you, the reader. (And we jolly well couldn't use that title because it was such a mouthful and not sexy enough.) We assume that you are already familiar with Microsoft Windows servers (or other operating systems, but we'll assume Windows for the sake of discussion) at a "power user" level or better. We assume that you are familiar with the terms (and some concepts) necessary to run a small- to medium-sized network of computers. Any experience with bigger networks or advanced Windows technologies, such as Active Directory, will allow you to get more from the book but is not required.

We make these assumptions because we did not want to write a guide for dummies. There are already enough books on the market that tell you what to click without telling you why; this book is not meant to be among those ranks.

In addition to your Windows background, we assume that you're interested in having more information about the topics here than the material we have written alone. After all, we've spent only 30 to 35 pages on topics that have entire books devoted to them! For this reason, we have scattered references to other resources throughout the chapters. We urge you to take advantage of these recommendations.

We believe that seasoned Linux system administrators can also benefit from this book because it can serve as a quick how-to cookbook on various topics that might not be the seasoned reader's strong points. We understand that system administrators generally have aspects of system administration that they like or loath a lot. For example, making backups is not one of our favorite aspects of system administration, and this is reflected in the half a page we've dedicated to backups. (Just kidding, there's an entire chapter on the topic.)

What's in This Book?

Linux Administration: A Beginner's Guide, Seventh Edition comprises six parts.

Part I: Introduction, Installation, and Software Management

Part I includes four chapters (Chapter 1, "Technical Summary of Linux Distributions"; Chapter 2, "Installing Linux in a Server Configuration"; Chapter 3, "The Command Line"; and Chapter 4, "Managing Software"). The first two chapters provide you with a nice overview of what Linux is, how it compares to Windows in several key areas, and how to install server-grade Fedora and Ubuntu Linux distributions.

Chapter 3, "The Command Line," begins covering the basics of working with the Linux command-line interface (CLI) so that you can become comfortable working without a GUI. Although it is possible to administer a system from within the graphical desktop, your greatest power comes from being comfortable with both the CLI and the GUI. (This is true for Windows, too. Don't believe that? Open a command prompt, run **netsh**, and try to do what **netsh** does in the GUI.)

Part I ends with a chapter on how to install software from prepackaged binaries and source code, as well as how to perform standard software management tasks.

Ideally, the information in Part I should be enough information to get you started and help you draw parallels to how Linux works based on your existing knowledge of other operating systems. Some of the server installation and software installation tasks performed in Part I help serve as a reference point for some other parts of the book.

Part II: Single-Host Administration

Part II covers the material necessary to manage what we call a “stand-alone” system, a system that does not require or provide any services to other systems on the network. Although the notion of a server *not* serving anything might seem counterintuitive at first, it is the foundation on which many other concepts are built, and it will come in handy for your understanding of network-based services later on.

Part II comprises six chapters. Chapter 5, “Managing Users and Groups,” covers the underlying basics of user and group concepts on Linux platforms, as well as day-to-day management tasks of adding and removing users and groups. The chapter also introduces the basic concepts of multiuser operation and the Linux permissions model.

Chapter 6, “Booting and Shutting Down,” documents the entire booting and shutting down processes. This chapter includes details on how to start up services properly and shut them down properly. You’ll learn how to add new services manually, which will also come in handy later on in the book.

Chapter 7, “File Systems,” continues with the basics of file systems—their organization, creation, and, most important, their management.

The basics of operation continue in Chapter 8, “Core System Services,” with coverage of basic tools such as `xinetd`, `upstart`, `rsyslog`, `cron`, `systemd`, `journald`, and so on. `xinetd` is the Linux equivalent of Windows’ `svchost`, and `rsyslog` manages logging for all applications in a unified framework. You might think of `rsyslog` and `journald` as more flexible versions of the Windows Event Viewer.

Kernel coverage in Chapter 9, “The Linux Kernel,” documents the process of configuring, compiling, and installing your own custom kernel in Linux. This capability is one of the points that gives Linux administrators an extraordinary amount of fine-grained control over how their systems operate.

Chapter 10, “Knobs and Dials: Virtual File Systems,” covers some kernel-level tweaking through the `/proc` and `/sys` file systems. The ability to view and modify certain kernel-level configuration and runtime variables through `/proc` and `/sys`, as shown in this chapter, gives administrators almost infinite kernel fine-tuning possibilities. When applied properly, this ability amounts to an arguably better and easier way to tweak the kernel than is afforded by Windows platforms.

Part III: Networking and Security

Part III begins our journey into the world of networking and security. We are not quite sure why we combined these two things together under the same part, but if forced to justify this, we would say that “the network is the root of most evils and therefore it needs to be secured.” Moving right along; with the ongoing importance of security on the Internet, as well as compliance issues with Sarbanes-Oxley (SOX) and the Health Insurance Portability and Accountability Act (HIPAA), the use of Linux in scenarios that require high security is unprecedented.

We deliberately decided to cover security before introducing network-based services (Parts IV and V) so that we could touch on some essential security best practices that can help in protecting your network-based services from attacks.

This section kicks off with Chapter 11, “TCP/IP for System Administrators,” which provides a detailed overview of TCP/IP in the context of what system (and/or network) administrators need to know. The chapter provides a lot of detail on how to use troubleshooting tools such as `tcpdump` to capture packets and read them back, as well as a step-by-step analysis of how TCP connections work. These tools should enable you to troubleshoot network peculiarities effectively.

Chapter 12, “Network Configuration,” returns to administration issues by focusing on basic network configuration (for both IPv4 and IPv6). This includes setting up IP addresses, routing entries, and so on. We extend past the basics in Chapter 13, “Linux Firewall (Netfilter),” by delving into advanced networking concepts and showing you how to build a Linux-based firewall and router. We touch on the new **nftables** project, which is slated to replace the existing and popular **iptables** framework.

Chapter 14, “Local Security,” and Chapter 15, “Network Security,” discuss aspects of system and network security in detail. They include Linux-specific issues as well as general security tips and tricks and commonsensical stuff so that you can better configure your system and protect it against attacks.

Part IV: Internet Services

The remainder of the book is divided into two distinct parts: “Internet Services” and “Intranet Services.” Although they sound similar, they are different—InTER(net) and InTRA(net). We define Internet services as those running on a Linux system exposed directly to the Internet. Examples include web and Domain Name System (DNS) services. We define intranet services as those that are typically run behind a firewall for internal users and mostly for internal consumption only.

This section starts off with Chapter 16, “Domain Name System (DNS),” which covers the information you need to know to install, configure, and manage a DNS server. In addition to the actual details of running a DNS server, we provide a detailed background on how DNS works and several troubleshooting tips, tricks, and tools.

From DNS, we move on to Chapter 17, “File Transfer Protocol (FTP),” which covers the installation and care of FTP servers. Like the DNS chapter, this chapter also includes a background on FTP itself and some notes on its evolution.

Chapter 18, “Apache Web Server,” moves on to what may be considered one of the most popular uses of Linux today: running a web server with the popular Apache software. This chapter covers the information necessary to install, configure, and manage the Apache web server.

Chapter 19, “Simple Mail Transfer Protocol (SMTP),” and Chapter 20, “Post Office Protocol and Internet Mail Access Protocol (POP and IMAP),” dive into e-mail through the setup and configuration of SMTP, POP, and IMAP servers.

We cover the information needed to configure all three, as well as show how they interact with one another. We have chosen to cover the Postfix SMTP server instead of the classic Sendmail server, because Postfix provides a more flexible server with a better security record.

Chapter 21, “Voice over Internet Protocol (VoIP),” is an all new contribution and addition to this edition. A lot of work and effort went into brewing and distilling the very extensive topic of VoIP into just over 50 pages of this chapter. We think it was well worth it, and at the end of this chapter we build a simple VoIP-based PBX based on Asterisk software that can easily be extended to replace commercial-grade PBX solutions, make phone calls among local extensions, or interface with third-party VoIP providers to interface with the rest of the world.

Part IV ends with Chapter 22, “Secure Shell (SSH).” Knowing how to set up and manage the SSH service is useful in almost any server environment—regardless of the server’s primary function.

Part V: Intranet Services

Again, we define intranet services as those that are typically run behind a firewall for internal users and mostly for internal consumption only. Even in this environment, Linux has a lot to offer. Part V starts off with Chapter 23, “Network File System (NFS).” NFS has been around for over 26 years now and has evolved and grown to fit the needs of its users quite well. This chapter covers Linux’s NFS server capabilities, including how to set up both clients and servers, as well as troubleshooting.

Chapter 24, “Samba,” continues the idea of sharing disks and resources with coverage of the Samba service. Using Samba, administrators can share disks and printing facilities and provide authentication for Windows (and Linux) users without having to install any special client software. Thus, Linux can become an effective server, able to support and share resources between UNIX/Linux systems as well as Windows systems. If you are into that *sort* of thing, Samba can even be configured to serve as a drop-in replacement for full-fledged Active Directory Microsoft Windows servers! Including Chapter 25, “Distributed File Systems (DFS),” in Part V instead of Part IV came down to a coin toss, because DFS can be used/deployed in both Internet- and intranet-facing scenarios. DFS solutions are especially important and relevant in today’s cloud-centric world. Among the many DFS implementations available, we have selected to cover GlusterFS because of its ease of configuration and cross-distribution support.

In Chapter 26, “Network Information Service (NIS),” we talk about NIS, which is typically deployed alongside NFS servers to provide a central naming service for all users within a network. The chapter pays special attention to scaling issues and how you can make NIS work in an environment with a large user base.

We revisit directory services in Chapter 27, “Lightweight Directory Access Protocol (LDAP),” with coverage of LDAP and how administrators can use this standard service for providing a centralized user database (directory) for use among heterogeneous operating systems and also for managing tons of users.

Chapter 28, “Printing,” takes a tour of the Linux printing subsystem. The printing subsystem, when combined with Samba, allows administrators to support seamless printing from Windows-based clients. The result is a powerful way of centralizing printing options for Linux, Windows, and even Mac OS X users on a single server.

Chapter 29, “Dynamic Host Configuration Protocol (DHCP),” covers another common use of Linux systems: DHCP servers. This chapter discusses how to deploy the Internet Systems Consortium (ISC) DHCP server, which offers a powerful array of features and access control options.

Then comes Chapter 30, “Virtualization.” Over several months, we agonized and struggled over what title to use for this chapter, because we cover both virtualization and containers (containerization). We ended up naming the chapter simply “Virtualization” because we are not even sure if containerization is a word. (Our copy editor Bill McManus says it is a word in the *shipping* industry!) Virtualization is everywhere. It allows companies to consolidate services and hardware that previously required several dedicated bare-metal machines into much fewer bare-metal machines. We discuss the basic virtualization concepts and briefly cover some of the popular virtualization technologies in Linux. The chapter also covers the Kernel-based Virtual Machine (KVM) implementation in detail, with examples. The KVM concepts that we discuss will help prepare you for the brand-new goodies that we have in store for you in Appendix B. The tail end of Chapter 30 features containers. The concepts behind containers are *old* but *new* again. Like virtualization, containers are everywhere ... and probably here to stay. We use the popular Docker platform for our container implementation and walk you through the process of how to deploy a web server container-style!

They say that all good things must come to an end, and that even applies to this book! The final chapter is Chapter 31, “Backups.” Backups are arguably one of the most critical pieces of administration. Linux-based systems support several methods of providing backups that are easy to use and readily usable by tape drives and other media. The chapter discusses some of the methods and explains how they can be used as part of a backup schedule. In addition to the mechanics of backups, we discuss general backup design and how you can optimize your backup system.

Part VI: Appendices

At the end of the book, we include some useful reference materials and real-world resources that you can use today and every day at work, at home, in the classroom, or in the lab.

Appendix A, “Creating a Linux Installer on Flash/USB Devices,” details alternate and generic methods for creating an installation media on non-optical media, such as a USB flash drive, SD card, and so on.

Appendix B is another brand-new addition to this book. It covers one of the new features that we’ve added to this seventh edition: how to obtain and use a

purpose-built virtual machine (VM) image file that we created especially for you as an accompaniment to this book. Once you power up the VM, you'll see most of the commands, scripts, software packages, hacks, and servers/daemons that we discuss throughout this book. If you look carefully, you might even find some of our very own sys admin *mistakes* enshrined in this VM.

Typographic Conventions

We were very ambitious in aiming for this book to be the one-stop Linux administration resource for many of the mainstream Linux distros. But we didn't stop there! We also wanted to reflect the subtle nuances and idiosyncrasies that sometimes exist between mainstream distros. To help with this, we had to adopt some unique conventions throughout this book. Some of these conventions appear in code listings, commands, command output, and command prompts.

Commands that are meant to be typed out by the reader are in bold Courier font. For example:

```
command --options foobar
```

The output of commands is in plain Courier font:

```
This is a sample output
```

When we intend to reference or explain the output of commands that span multiple lines, we may sometimes number the output of such commands (also in Courier font). For example:

1. This is line 1 of a sample command output
2. This is line 2 of a sample command output

Some Linux commands and their options and output can be rather long. For these situations we place the backslash symbol (\) at the end of each line (except the final line) to indicate that the command spans multiple lines. For example, for a command with multiple options that spans multiple lines, we would show this as

```
command --option1 foobar1 --option2 foobar2 \  
--option3 foobar3 --option4 foobar4 --option4 foobar4 \  
--option5 foobar5
```

For distribution-independent commands, the command prompt will not have any particular distribution name. For example:

```
[root@server ~]#
```


For commands that target a specific Linux distribution, the command prompt will incorporate the distribution name. For example, for a Fedora-based server, the command prompt will resemble

```
[root@fedora-server ~]#
```

And for an Ubuntu-based server, the command prompt will resemble

```
root@ubuntu-server: ~#
```

Debian-based distros such as Ubuntu often do not directly use the superuser (*root*) or administrator account for administering the system. Regular user accounts with elevated privileges are used for this purpose. To achieve this, most commands are often prefixed with the **sudo** command on such platforms. Consider a regular user named “*master*” with **sudo** privileges who wants to run a restricted command. The command prompt and command prefixed with **sudo** will resemble

```
master@ubuntu-server:~$ sudo command --options
```

For projects or exercises that are meant to be run from a client system, the command prompt might appear as

```
user@client ~$
```

For projects or exercises that are best suited for multi-server scenarios, when you absolutely need more than one server, the command prompt will change to reflect the steps or commands to be executed on the particular server. For example, the prompt for the first server (Server-A) might resemble

```
[root@server-A ~]#
```

and the prompt for the second server (Server-B) might resemble

```
[root@server-B ~]#
```

Updates and Feedback

Although we hope that we've published a book with no errors, we have set up an errata list for this book at www.linuxserverexperts.com. If you find any errors, we welcome your submissions for errata updates. We also welcome your feedback and comments. Unfortunately, our day jobs prevent us from answering detailed questions, so if you're looking for help on a specific issue, you may find one of the many online communities a useful resource. However, if you have two cents to share about the book, we welcome your thoughts. You can send us an e-mail to feedback@linuxserverexperts.com.



PART I | **Introduction, Installation,
and Software Management**

This page intentionally left blank



CHAPTER 1

Technical Summary of Linux Distributions



Linux has hit the mainstream. Hardly a day goes by without a mention of Linux (or open source software) in widely read and viewed print or digital media. What was only a hacker's toy many eons ago has grown up tremendously and is well known for its stability, performance, and extensibility.

If you need more proof concerning Linux's penetration, just pay attention to the frequency with which "Linux" is listed as a *desirable* and *must have* skill for technology-related job postings of Fortune 500 companies, small to medium-sized businesses, tech start-ups, and government, research, and entertainment industry jobs—to mention a few. The skills of good Linux system administrators and engineers are highly desirable!

With the innovations that are taking place in different open source projects (such as K Desktop Environment, GNOME, Unity, LibreOffice, Android, Apache, Samba, Mozilla, and so on), Linux has made serious inroads into consumer desktop, laptop, tablet, and mobile markets. This chapter looks at some of the core server-side technologies as they are implemented in the Linux (open source) world and in the Microsoft Windows Server world (possibly the platform you are more familiar with). But before delving into any technicalities, this chapter briefly discusses some important underlying concepts and ideas that form the genetic makeup of Linux and Free and Open Source Software (FOSS).

Linux: The Operating System

Some people (mis)understand Linux to be an entire software suite of developer tools, editors, graphical user interfaces (GUIs), networking tools, and so forth. More formally and correctly, such software *collectively* is called a *distribution*, or *distro*. The distro is the entire software suite that makes Linux useful.

So if we consider a distribution everything you need for Linux, what then *is* Linux exactly? Linux itself is the core of the operating system: the *kernel*. The kernel is the program acting as chief of operations. It is responsible for starting and stopping other programs (such as text editors, web browsers, services, and so on), handling requests for memory, accessing disks, and managing network connections. The complete list of kernel activities could easily fill a chapter in itself, and, in fact, several books documenting the kernel's internal functions have been written.

The kernel is a nontrivial program. It is also what puts the Linux badge on all the numerous Linux distributions. All distributions use essentially the same kernel, so the fundamental behavior of all Linux distributions is the same.

You've most likely heard of the Linux distributions named Red Hat Enterprise Linux (RHEL), Fedora, Debian, Mageia, Ubuntu, Mint, openSUSE, CentOS, Arch, Chrome OS, Slackware, and so on, which have received a great deal of press.

Linux distributions can be broadly categorized into two groups. The first category includes the purely commercial distros, and the second includes the noncommercial distros. The commercial distros generally offer support for their distribution—at a cost. The commercial distros also tend to have a longer release life cycle. Examples of commercial flavors of Linux-based distros are RHEL and SUSE Linux Enterprise (SLE).

The noncommercial distros, on the other hand, are free. These distros try to adhere to the original spirit of the open source software movement. They are mostly community supported and maintained—the community consists of the users and developers. The community support and enthusiasm can sometimes supersede that provided by the commercial offerings.

Several of the so-called noncommercial distros also have the backing and support of their commercial counterparts. The companies that offer the purely commercial flavors have vested interests in making sure that free distros exist. Some of the companies use the free distros as the proofing and testing ground for software that ends up in the commercial spins. Examples of noncommercial flavors of Linux-based distros are Fedora, openSUSE, Ubuntu, Linux Mint, Gentoo, and Debian. Linux distros such as Gentoo might be less well known and have not reached the same scale of popularity as Fedora, openSUSE, and others, but they are out there and in active use by their respective (and dedicated) communities.

What's interesting about the commercial Linux distributions is that most of the programs with which they ship were not written by the companies themselves. Rather, other people have released their programs with licenses, allowing their redistribution with source code. By and large, these programs are also available on other variants of UNIX, and some of them are becoming available under Windows as well. The makers of the distribution simply bundle them into one convenient and cohesive package that's easy to install. In addition to bundling existing software, several of the distribution makers also develop value-added tools that make their distribution easier to administer or compatible with more hardware, but the software that they ship is generally written by others. To meet certain regulatory requirements, some commercial distros try to incorporate/implement more specific security requirements that the FOSS community might not care about but that some institutions/corporations do care about.

Open Source Software and GNU: Overview

In the early 1980s, Richard Matthew Stallman began a movement within the software industry. He preached (and still does) that software should be free. Note that by *free*, he doesn't mean in terms of price, but rather free in the same sense as *freedom* or *libre*. This means shipping not just a product, but the entire source code as well. To clarify the meaning of free software, Stallman was once famously quoted as saying:

“Free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer.”

Stallman's position was, somewhat ironically, a return to classic computing, when software was freely shared among hobbyists on small computers and provided as part of the hardware by mainframe and minicomputer vendors. It was not until the late

1960s that IBM considered selling application software. Through the 1950s and most of the 1960s, IBM considered software as merely a tool for enabling the sale of hardware.

This return to *openness* was a wild departure from the early 1980s convention of selling prepackaged software, but Stallman's concept of open source software was in line with the initial distributions of UNIX from Bell Labs. Early UNIX systems did contain full source code. Yet by the late 1970s, source code was typically removed from UNIX distributions and could be acquired only by paying large sums of money to AT&T (now SBC). The Berkeley Software Distribution (BSD) maintained a free version, but its commercial counterpart, BSDi, had to deal with many lawsuits from AT&T until it could be proved that nothing in the BSD kernel came from AT&T.

Kernel Differences

Each company that sells a Linux distribution of its own will be quick to tell you that its kernel is better than the others. How can a company make this claim? The answer comes from the fact that each company maintains its own patch set. To make sure that the kernels largely stay in sync, most companies do adopt patches that are posted on www.kernel.org, the "Linux Kernel Archives." Vendors, however, typically do not track the release of every single kernel version that is released onto www.kernel.org. Instead, they take a foundation, apply their custom patches to it, run the kernel through their quality assurance (QA) process, and then take it to production. This helps organizations have confidence that their kernels have been sufficiently baked, thus mitigating any perceived risk of running open source-based operating systems.

The only exception to this rule revolves around security issues. If a security issue is found with a version of the Linux kernel, vendors are quick to adopt the necessary patches to fix the problem immediately. A new release of the kernel with the fixes is often made within a short time (commonly less than 24 hours) so that administrators who install it can be sure their installations are secure. Thankfully, exploits against the kernel itself are rare.

So if each vendor maintains its own patch set, what exactly is it patching? This answer varies from vendor to vendor, depending on each vendor's target market. Red Hat, for instance, is largely focused on providing enterprise-grade reliability and solid efficiency for application servers. This might be different from the mission of the Fedora team, which is more interested in trying new technologies quickly, and even more different from the approach of a vendor that is trying to put together a desktop-oriented or multimedia-focused Linux system.

What separates one distribution from the next are the value-added tools that come with each one. Asking, "Which distribution is better?" is much like asking, "Which is better, Coke or Pepsi?" Almost all colas have the same basic

ingredients—carbonated water, caffeine, and high-fructose corn syrup—thereby giving the similar effect of quenching thirst and bringing on a small caffeine-and-sugar buzz. In the end, it's a question of requirements: Do you need commercial support? Did your application vendor recommend one distribution over another? Does the software (package) updating infrastructure suit your site's administrative style better than another distribution? When you review your requirements, you'll find that there is likely a distribution that is geared toward your exact needs.

The idea of giving away source code is a simple one: A user of the software should never be forced to deal with a developer who might or might not support that user's intentions for the software. The user should never have to wait for bug fixes to be published. More important, code developed under the scrutiny of other programmers is typically of higher quality than code written behind locked doors. One of the great benefits of open source software comes from the users themselves: Should they need a new feature, they can add it to the original program and then contribute it back to the source so that everyone else can benefit from it.

This line of thinking sprung a desire to release a complete UNIX-like system (Linux) to the public, free of license restrictions. Of course, before you can build any operating system, you need to build tools. And this is how the GNU project and its namesake license were born.



NOTE GNU stands for GNU's Not UNIX—recursive acronyms are part of hacker humor. If you don't understand why it's funny, don't worry. You're still in the majority.

The GNU Public License

An important thing to emerge from the GNU project is the GNU Public License (GPL). This license explicitly states that the software being released is free and that no one can ever take away these freedoms. It is acceptable to take the software and resell it, even for a profit; however, in this resale, the seller must release the full source code, including any changes. Because the resold package remains under the GPL, the package can be distributed for free and resold yet again by anyone else for a profit. Of primary importance is the liability clause: The programmers are not liable for any damages caused by their software.

It should be noted that the GPL is not the only license used by open source software developers (although it is arguably the most popular). Other licenses, such as BSD and Apache, have similar liability clauses but differ in terms of their redistribution. For instance, the BSD license allows people to make changes to the code and ship those changes without having to disclose the added code. (Whereas the GPL requires that the added code is shipped.) For more information about other open source licenses, check out www.opensource.org.

Historical Footnote

Many, many moons ago, Red Hat started a commercial offering of its erstwhile free product (Red Hat Linux). The commercial release gained steam with the Red Hat Enterprise Linux (RHEL) series. Because the foundation for RHEL is GPL, individuals interested in maintaining a free version of Red Hat's distribution have been able to do so. Furthermore, as an outreach to the community, Red Hat created the Fedora Project, which is considered the testing grounds for new software before it is adopted by the RHEL team. The Fedora Project is freely distributed and can be downloaded from <http://fedoraproject.org> or <https://getfedora.org>.

Upstream and Downstream

To help you understand the concept of upstream and downstream components, let's start with an analogy. Picture, if you will, a pizza with all your favorite toppings.

The pizza is put together and baked by a local *pizza shop*. Several things go into making a great pizza—cheeses, vegetables, flour (dough), herbs, meats, to mention a few. The pizza shop will often make some of these ingredients in-house and rely on other businesses to supply other ingredients. The pizza shop will also be tasked with assembling the ingredients into a complete finished pizza.

Let's consider one of the most common pizza ingredients—cheese. The cheese is made by a *cheesemaker* who makes her cheese for many other industries or applications, including the pizza shop. The cheesemaker is pretty set in her ways and has very strong opinions about how her product should be paired with other food stuffs (wine, crackers, bread, vegetables, and so on). The pizza shop owners, on the other hand, do not care about other food stuffs—they care only about making a great pizza. Sometimes the cheesemaker and the pizza shop owners will bump heads because of differences in opinion and objectives. And at other times they will be in agreement and cooperate beautifully. Ultimately (and sometimes unbeknown to them), the pizza shop owners and cheesemaker care about the same thing: producing the best product that they can.

The pizza shop in our analogy here represents the Linux distributions vendors/projects (Fedora, Debian, RHEL, openSUSE, and so on). The cheesemaker represents the different software project maintainers that provide the important programs and tools (such as the Bourne Again Shell [Bash], GNU Image Manipulation Program [GIMP], GNOME, KDE, Nmap, and GNU Compiler Collection [GCC]) that are packaged together to make a complete distribution (pizza). The Linux distribution vendors are referred to as the *downstream* component of the open source food chain; the maintainers of the accompanying different software projects are referred to as the *upstream* component.

Standards

One argument you hear regularly against Linux is that too many different distributions exist, and that by having multiple distributions, fragmentation occurs. The argument opines that this fragmentation will eventually lead to different versions of incompatible distros.

This is, without a doubt, complete nonsense that plays on “FUD” (fear, uncertainty, and doubt). These types of arguments usually stem from a misunderstanding of the kernel and distributions.

Ever since becoming so mainstream, the Linux community understood that it needed a formal method and standardization process for how certain things should be done among the numerous Linux spins. As a result, two major standards are actively being worked on.

The *Filesystem Hierarchy Standard (FHS)* is an attempt by many of the Linux distributions to standardize on a directory layout so that developers have an easy time making sure their applications work across multiple distributions without difficulty. As of this writing, several major Linux distributions have become completely compliant with this standard.

The *Linux Standard Base (LSB)* specification is a standards group that specifies what a Linux distribution should have in terms of libraries and tools.

A developer who assumes that a Linux machine complies only with LSB and FHS is almost guaranteed to have an application that will work with all compliant Linux installations. All of the major distributors have joined these standards groups. This should ensure that all desktop distributions will have a certain amount of commonality on which a developer can rely.

From a system administrator’s point of view, these standards are interesting but not crucial to administering a Linux environment. However, it never hurts to learn more about both. For more information on the FHS, go to their web site at www.pathname.com/fhs. To find out more about LSB, check out www.linuxfoundation.org/collaborate/workgroups/lsb.

The Advantages of Open Source Software

If the GPL seems like a bad idea from the standpoint of commercialism, consider the surge of successful open source software projects—they are indicative of a system that does indeed work. This success has evolved for two reasons. First, as mentioned earlier, errors in the code itself are far more likely to be caught and quickly fixed under the watchful eyes of peers. Second, under the GPL system, programmers can release code without the fear of being sued. Without that protection, people might not feel as comfortable to release their code for public consumption.