

Matthew Helmke

**Ubuntu
16.04 LTS
on DVD**

Ubuntu

UNLEASHED

2017 Edition

Covering 16.10, 17.04, and 17.10

SAMS

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Matthew Helmke
with Andrew Hudson
and Paul Hudson

Ubuntu

UNLEASHED

2017 Edition



800 East 96th Street, Indianapolis, Indiana 46240 USA

Ubuntu Unleashed 2017 Edition

Copyright © 2017 by Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informat.com/aw

ISBN-13: 978-0-13-451118-4

ISBN-10: 0-13-451118-2

Library of Congress Cataloging-in-Publication Data: 2016951091

Printed in the United States of America

Text printed in the United States on recycled paper at RR Donnelley,
Crawfordsville, Indiana.

First Printing October 2016

Editor-in-Chief

Mark Taub

Acquisitions Editor

Debra Williams
Cauley

Managing Editor

Sandra Schroeder

Project Editor

Lori Lyons

Production Manager

Dhayanidhi

Proofreader

Sasirekha

Technical Editor

José Antonio Rey

Editorial Assistant

Kim Boedigheimer

Media Producer

Dan Scherf

Cover Designer

Chuti Prasertsith

Compositor

codeMantra

Contents at a Glance

Introduction	xxxiii
Part I Getting Started	
1 Installing Ubuntu and Post-Installation Configuration	1
2 Background Information and Resources	29
Part II Desktop Ubuntu	
3 Working with Unity	43
4 On the Internet	63
5 Productivity Applications	75
6 Multimedia Applications	85
7 Other Ubuntu Interfaces	111
8 Games	121
Part III System Administration	
9 Managing Software	133
10 Command-Line Beginner's Class	149
11 Command-Line Master Class Part 1	185
12 Command-Line Master Class Part 2	209
13 Managing Users	241
14 Automating Tasks and Shell Scripting	265
15 The Boot Process	311
16 System-Monitoring Tools	323
17 Backing Up	341
18 Networking	369
19 Remote Access with SSH, Telnet, and VNC	415
20 Securing Your Machines	425
21 Performance Tuning	441
22 Kernel and Module Management	453
Part IV Ubuntu as a Server	
23 Sharing Files and Printers	473
24 Apache Web Server Management	489
25 Nginx Web Server Management	515

26	Other HTTP Servers	527
27	Remote File Serving with FTP	531
28	Handling Email	541
29	Proxying, Reverse Proxying, and Virtual Private Networks (VPN)	561
30	Administering Relational Database Services	577
31	NoSQL Databases	603
32	Lightweight Directory Access Protocol (LDAP)	615
33	Linux Terminal Server Project (LTSP)	623
34	Virtualization on Ubuntu	631
35	Ubuntu in the Cloud	641
36	Managing Sets of Servers	655
37	Name Serving with the Domain Name System (DNS)	659
Part V Programming Linux		
38	Using Programming Tools for Ubuntu	669
39	Opportunistic Development	681
40	Helping with Ubuntu Development	693
41	Helping with Ubuntu Testing and QA	703
42	Using Popular Programming Languages	709
43	Beginning Mobile Development for Android	723
44	Developing for Ubuntu Mobile/Touch	729
	Index	733

NOTE

Chapters 45–47 can be accessed online at informit.com/title/9780134268118.

Part VI Bonus Chapters

45	Using Perl	Web:1
46	Using Python	Web:23
47	Using PHP	Web:43

Table of Contents

Introduction	xxxiii
Licensing	xxxiv
Who This Book Is For	xxxv
Those Wanting to Become Intermediate or Advanced Users.....	xxxv
Sysadmins, Programmers, and DevOps	xxxvi
What This Book Contains	xxxvii
Conventions Used in This Book	xxxviii
Part I Getting Started	
1 Installing Ubuntu and Post-Installation Configuration	1
Before You Begin the Installation	1
Researching Your Hardware Specifications	2
Installation Options	2
32-Bit vs 64-Bit Ubuntu	4
Planning Partition Strategies	5
The Boot Loader	5
Installing from DVD or USB Drive.....	6
Step-by-Step Installation	6
Installing	7
First Update	13
Shutting Down	13
Finding Programs and Files	14
Software Updater	15
The <code>sudo</code> Command	18
Configuring Software Repositories	19
System Settings	21
Detecting and Configuring a Printer	22
Configuring Power Management in Ubuntu.....	22
Setting the Time and Date	23
Configuring Wireless Networks	24
Troubleshooting Post-Installation Configuration Problems	26
References	27
2 Background Information and Resources	29
What Is Linux?	29
Why Use Linux?	31
What Is Ubuntu?	33

Ubuntu for Business	33
Ubuntu in Your Home	35
Getting the Most from Ubuntu and Linux Documentation	35
Ubuntu Developers and Documentation	36
Websites and Search Engines	37
Web Search Tips	37
Google Is Your Friend	37
Ubuntu Package Listings	38
Commercial Support	38
Documentation	39
Linux Guides	39
Ubuntu	40
Mailing Lists	40
Ubuntu Project Mailing Lists	41
Internet Relay Chat	42

Part II Desktop Ubuntu

3 Working with Unity	43
Foundations and the X Server	43
Basic X Concepts	44
Using X	45
Elements of the <code>xorg.conf</code> File	46
Starting X	51
Using a Display Manager	51
Changing Window Managers	52
Using Unity, a Primer	52
The Desktop	53
Customizing and Configuring Unity	58
Power Shortcuts	60
References	61
4 On the Internet	63
Getting Started with Firefox	63
Checking Out Google Chrome and Chromium	65
Choosing an Email Client	66
Mozilla Thunderbird	66
Evolution	67
Other Mail Clients	68
RSS Readers	69
Firefox	69
Liferea	69

Internet Relay Chat	70
Usenet Newsgroups	72
References	74
5 Productivity Applications	75
Introducing LibreOffice	76
Other Office Suites for Ubuntu	78
Working with GNOME Office	78
Working with KOffice	80
Other Useful Productivity Software	80
Working with PDF	80
Working with XML and DocBook	81
Working with LaTeX	82
Productivity Applications Written for Microsoft Windows	83
References	83
6 Multimedia Applications	85
Sound and Music	85
Sound Cards	86
Adjusting Volume	87
Sound Formats	88
Listening to Music	89
Graphics Manipulation	92
The GNU Image Manipulation Program	93
Using Scanners in Ubuntu	94
Working with Graphics Formats	95
Capturing Screen Images	97
Other Graphics Manipulation Options	97
Using Digital Cameras with Ubuntu	98
Handheld Digital Cameras	98
Using Shotwell Photo Manager	98
Burning CDs and DVDs in Ubuntu	99
Creating CDs and DVDs with Brasero	99
Creating CDs from the Command Line	100
Creating DVDs from the Command Line	102
Viewing Video	104
TV and Video Hardware	104
Video Formats	105
Viewing Video in Linux	106
Personal Video Recorders	107
Video Editing	107
References	109

7 Other Ubuntu Interfaces	111
Desktop Environment	112
KDE and Kubuntu	113
Xfce and Xubuntu	114
LXDE and Lubuntu	115
GNOME3 and Ubuntu GNOME	116
MATE and Ubuntu MATE	117
Ubuntu Kylin	118
References	118
8 Games	121
Ubuntu Gaming	121
Installing Proprietary Video Drivers	122
Steam	123
Installing Games in Ubuntu	123
Warsow	124
Scorched 3D	124
Frozen Bubble	125
SuperTux	126
Battle for Wesnoth	126
Frets on Fire	127
FlightGear	128
Speed Dreams	129
Games for Kids	129
Commercial Games	129
Playing Windows Games	130
References	130
Part III System Administration	
9 Managing Software	133
Ubuntu Software	133
Using Synaptic for Software Management	134
Staying Up-to-Date	137
Working on the Command Line	138
Day-to-Day Usage	138
Finding Software	141
Using apt Instead of apt-get	142
Compiling Software from Source	143
Compiling from a Tarball	143
Compiling from Source from the Ubuntu Repositories	144
Configuration Management	145
dotdee	145

Snappy Ubuntu Core.....	146
Using Snaps	146
References.....	147
10 Command-Line Beginner's Class	149
What Is the Command Line?	150
Accessing the Command Line.....	151
Text-Based Console Login.....	152
Logging Out.....	153
Logging In and Out from a Remote Computer	153
User Accounts	154
Reading Documentation.....	155
Using Man Pages	156
Using <code>apropos</code>	156
Using <code>whereis</code>	157
Understanding the Linux File System Hierarchy	157
Essential Commands in <code>/bin</code> and <code>/sbin</code>	158
Configuration Files in <code>/etc</code>	159
User Directories: <code>/home</code>	159
Using the Contents of the <code>/proc</code> Directory to Interact with the Kernel	160
Working with Shared Data in the <code>/usr</code> Directory.....	161
Temporary File Storage in the <code>/tmp</code> Directory.....	162
Accessing Variable Data Files in the <code>/var</code> Directory.....	162
Navigating the Linux File System	162
Listing the Contents of a Directory with <code>ls</code>	162
Changing Directories with <code>cd</code>	164
Finding Your Current Directory with <code>pwd</code>	165
Working with Permissions	165
Assigning Permissions	166
Directory Permissions	167
Altering File Permissions with <code>chmod</code>	168
File Permissions with <code>umask</code>	169
File Permissions with <code>chgrp</code>	170
Changing File Permissions with <code>chown</code>	170
Understanding Set User ID, Set Group ID, and Sticky Bit Permissions	170
Setting Permissions with Access Control Lists.....	171
Working with Files	173
Creating a File with <code>touch</code>	173
Creating a Directory with <code>mkdir</code>	173
Deleting a Directory with <code>rmdir</code>	174
Deleting a File or Directory with <code>rm</code>	175

Moving or Renaming a File with <code>mv</code>	175
Copying a File with <code>cp</code>	176
Displaying the Contents of a File with <code>cat</code>	177
Displaying the Contents of a File with <code>less</code>	177
Using Wildcards and Regular Expressions	177
Working as Root	178
Understanding and Fixing <code>sudo</code>	178
Creating Users.....	181
Deleting Users.....	182
Shutting Down the System.....	182
Rebooting the System	183
Commonly Used Commands and Programs.....	183
References.....	184
11 Command-Line Master Class Part 1	185
Why Use the Command Line?.....	186
Using Basic Commands.....	187
Printing the Contents of a File with <code>cat</code>	188
Changing Directories with <code>cd</code>	189
Changing File Access Permissions with <code>chmod</code>	191
Copying Files with <code>cp</code>	191
Printing Disk Usage with <code>du</code>	192
Using <code>echo</code>	193
Finding Files by Searching with <code>find</code>	193
Searches for a String in Input with <code>grep</code>	196
Paging Through Output with <code>less</code>	197
Creating Links Between Files with <code>ln</code>	199
Finding Files from an Index with <code>locate</code>	200
Listing Files in the Current Directory with <code>ls</code>	200
Listing System Information with <code>lblk</code> , <code>lshw</code> , <code>lsmod</code> , and <code>lspci</code>	202
Reading Manual Pages with <code>man</code>	203
Making Directories with <code>mkdir</code>	204
Moving Files with <code>mv</code>	204
Renaming Files with <code>rename</code>	204
Deleting Files and Directories with <code>rm</code>	205
Sorting the Contents of a File with <code>sort</code>	205
Printing the Last Lines of a File with <code>tail</code>	207
Printing the Location of a Command with <code>which</code>	207
Download Files with <code>wget</code>	207
References.....	208

12 Command-Line Master Class Part 2	209
Redirecting Output and Input	209
stdin, stdout, stderr, and Redirection	211
Comparing Files	212
Finding Differences in Files with <code>diff</code>	212
Finding Similarities in Files with <code>comm</code>	212
Limiting Resource Use and Job Control	213
Listing Processes with <code>ps</code>	213
Listing Jobs with <code>jobs</code>	214
Running One or More Tasks in the Background	215
Moving Jobs to the Background or Foreground with <code>bg</code> and <code>fg</code>	215
Printing Resource Usage with <code>top</code>	216
Setting Processes Priority with <code>nice</code>	218
Combining Commands	219
Pipes	219
Combining Commands with Boolean Operators	221
Running Separate Commands in Sequence	222
Process Substitution	222
Using Environment Variables	222
Using Common Text Editors	226
Working with <code>nano</code>	227
Working with <code>vi</code>	227
Working with <code>emacs</code>	229
Working with <code>sed</code> and <code>awk</code>	230
Working with Compressed Files	232
Using Multiple Terminals with <code>byobu</code>	233
Polite System Reset Using REISUB	234
Fixing an Ubuntu System That Will Not Boot	235
Checking BIOS	235
Checking GRUB	235
Reinstalling GRUB	235
Using Recovery Mode	236
Reinstalling Ubuntu	236
Tips and Tricks	236
Running the Previous Command	236
Running Any Previous Command	237
Running a Previous Command that Started with Specific Letters	237
Running the Same Thing You Just Ran with a Different First Word	237
Viewing Your History and More	237

Doing Two or More Things.....	237
Using Shortcuts.....	238
Confining a Script to a Directory.....	238
Using Coreutils.....	239
Reading the Contents of the Kernel Ring Buffer with <code>dmesg</code>	239
References.....	240
13 Managing Users	241
User Accounts	241
The Super User/Root User.....	242
User IDs and Group IDs	244
File Permissions.....	244
Managing Groups	245
Group Listing	245
Group Management Tools	246
Managing Users	248
User Management Tools	248
Adding New Users	250
Monitoring User Activity on the System	251
Managing Passwords	252
System Password Policy	252
The Password File	253
Shadow Passwords	254
Managing Password Security for Users	256
Changing Passwords in a Batch	256
Granting System Administrator Privileges to Regular Users	257
Temporarily Changing User Identity with the <code>su</code> Command	257
Granting Root Privileges on Occasion: The <code>sudo</code> Command	259
Disk Quotas	262
Implementing Quotas	262
Manually Configuring Quotas	263
Related Ubuntu Commands	264
References	264
14 Automating Tasks and Shell Scripting	265
Scheduling Tasks	265
Using <code>at</code> and <code>batch</code> to Schedule Tasks for Later	265
Using <code>cron</code> to Run Jobs Repeatedly	268
Using <code>rtcwake</code> to Wake Your Computer from Sleep Automatically	270
Basic Shell Control	272
The Shell Command Line	273
Shell Pattern-Matching Support	274

Redirecting Input and Output	275
Piping Data	276
Background Processing	277
Writing and Executing a Shell Script	277
Running the New Shell Program	279
Storing Shell Scripts for System-Wide Access	279
Interpreting Shell Scripts Through Specific Shells	280
Using Variables in Shell Scripts	281
Assigning a Value to a Variable	282
Accessing Variable Values	282
Positional Parameters	282
A Simple Example of a Positional Parameter	283
Using Positional Parameters to Access and Retrieve Variables from the Command Line	284
Using a Simple Script to Automate Tasks	284
Built-In Variables	286
Special Characters	287
Using Double Quotes to Resolve Variables in Strings with Embedded Spaces	288
Using Single Quotes to Maintain Unexpanded Variables	288
Using the Backslash as an Escape Character	289
Using the Backtick to Replace a String with Output	289
Comparison of Expressions in <code>pdksh</code> and <code>bash</code>	290
Comparing Expressions with <code>tcsh</code>	295
The <code>for</code> Statement	299
The <code>while</code> Statement	300
The <code>until</code> Statement	302
The <code>repeat</code> Statement (<code>tcsh</code>)	303
The <code>select</code> Statement (<code>pdksh</code>)	303
The <code>shift</code> Statement	304
The <code>if</code> Statement	304
The <code>case</code> Statement	305
The <code>break</code> and <code>exit</code> Statements	307
Using Functions in Shell Scripts	307
References	308
 15 The Boot Process	 311
Running Services at Boot	311
Beginning the Boot Loading Process	312
Loading the Linux Kernel	314
System Services and Runlevels	315
Runlevel Definitions	315

Booting into the Default Runlevel	316
Understanding <code>init</code> Scripts and the Final Stage of Initialization	316
Controlling Services at Boot with Administrative Tools	317
Changing Runlevels	318
Troubleshooting Runlevel Problems	319
Starting and Stopping Services Manually	319
Using Upstart	319
<code>systemd</code>	320
Boot Repair	322
References	322
16 System-Monitoring Tools	323
Console-Based Monitoring	323
Using the <code>kill</code> Command to Control Processes	325
Using Priority Scheduling and Control	326
Displaying Free and Used Memory with <code>free</code>	327
Disk Space	328
Disk Quotas	329
Checking Log Files	329
Rotating Log Files	331
Graphical Process and System Management Tools	333
System Monitor	334
Conky	334
Other	339
KDE Process- and System-Monitoring Tools	339
Enterprise Server Monitoring	340
Landscape	340
Other	340
References	340
17 Backing Up	341
Choosing a Backup Strategy	341
Why Data Loss Occurs	342
Assessing Your Backup Needs and Resources	343
Evaluating Backup Strategies	345
Making the Choice	348
Choosing Backup Hardware and Media	348
Removable Storage Media	348
CD-RW and DVD+RW/-RW Drives	349
Network Storage	349
Tape Drive Backup	349
Cloud Storage	350

Using Backup Software	350
tar: The Most Basic Backup Tool	351
The GNOME File Roller	353
The KDE ark Archiving Tool	353
Déjà Dup	354
Back In Time	356
Unison	358
Using the Amanda Backup Application	358
Alternative Backup Software	359
Copying Files	360
Copying Files Using tar	360
Compressing, Encrypting, and Sending tar Streams	361
Copying Files Using cp	361
Copying Files Using mc	362
Using rsync	362
Version Control for Configuration Files	364
System Rescue	366
The Ubuntu Rescue Disc	367
Restoring the GRUB2 Boot Loader	367
Saving Files from a Nonbooting Hard Drive	368
References	368
18 Networking	369
Laying the Foundation: The localhost Interface	370
Checking for the Availability of the Loopback Interface	370
Configuring the Loopback Interface Manually	370
Checking Connections with ping, traceroute, and mtr	371
Networking with TCP/IP	374
TCP/IP Addressing	374
Using IP Masquerading in Ubuntu	376
Ports	377
IPv6 Basics	378
Network Organization	380
Subnetting	381
Subnet Masks	381
Broadcast, Unicast, and Multicast Addressing	382
Hardware Devices for Networking	382
Network Interface Cards	382
Network Cable	384
Hubs and Switches	385
Routers and Bridges	386
Initializing New Network Hardware	387

Using Network Configuration Tools.....	389
Command-Line Network Interface Configuration.....	389
Network Configuration Files.....	394
Using Graphical Configuration Tools.....	397
Dynamic Host Configuration Protocol.....	399
How DHCP Works	399
Activating DHCP at Installation and Boot Time.....	400
DHCP Software Installation and Configuration	401
Using DHCP to Configure Network Hosts	403
Other Uses for DHCP	405
Wireless Networking	405
Support for Wireless Networking in Ubuntu.....	405
Advantages of Wireless Networking	407
Choosing from Among Available Wireless Protocols	407
Beyond the Network and onto the Internet.....	408
Common Configuration Information.....	408
Understanding PPP over Ethernet	410
Configuring a PPPoE Connection Manually	411
Configuring Dial-Up Internet Access	412
Troubleshooting Connection Problems	413
References.....	414
19 Remote Access with SSH, Telnet, and VNC	415
Setting Up a Telnet Server.....	415
Telnet Versus SSH.....	417
Setting Up an SSH Server.....	417
SSH Tools	417
Using <code>scp</code> to Copy Individual Files Between Machines	418
Using <code>sftp</code> to Copy Many Files Between Machines	418
Using <code>ssh-keygen</code> to Enable Key-Based Logins	419
Virtual Network Computing	420
References.....	423
20 Securing Your Machines	425
Understanding Computer Attacks.....	425
Assessing Your Vulnerability	427
Protecting Your Machine	428
Securing a Wireless Network.....	429
Passwords and Physical Security	429
Configuring and Using Tripwire	430
Devices	431

Viruses.....	431
Configuring Your Firewall.....	432
AppArmor.....	435
Forming a Disaster Recovery Plan.....	437
References.....	439
21 Performance Tuning	441
Hard Disk.....	441
Using the BIOS and Kernel to Tune the Disk Drives.....	442
The <code>hdparm</code> Command.....	443
File System Tuning.....	444
The <code>tune2fs</code> Command.....	444
The <code>e2fsck</code> Command.....	445
The <code>badblocks</code> Command.....	445
Disabling File Access Time.....	445
Kernel.....	445
Apache.....	446
MySQL.....	448
Measuring Key Buffer Usage.....	448
Using the Query Cache.....	449
Miscellaneous Tweaks.....	451
Query Optimization.....	451
References.....	452
22 Kernel and Module Management	453
The Linux Kernel.....	454
The Linux Source Tree.....	455
Types of Kernels.....	457
Managing Modules.....	458
When to Recompile.....	460
Kernel Versions.....	461
Obtaining the Kernel Sources.....	462
Patching the Kernel.....	463
Compiling the Kernel.....	464
Using <code>xconfig</code> to Configure the Kernel.....	467
Creating an Initial RAM Disk Image.....	470
When Something Goes Wrong.....	470
Errors During Compile.....	471
Runtime Errors, Boot Loader Problems, and Kernel Oops.....	472
References.....	472

Part IV Ubuntu as a Server

23 Sharing Files and Printers	473
Using the Network File System	474
Installing and Starting or Stopping NFS	474
NFS Server Configuration	474
NFS Client Configuration	475
Putting Samba to Work	476
Manually Configuring Samba with <code>/etc/samba/smb.conf</code>	478
Testing Samba with the <code>testparm</code> Command	481
Starting, Stopping, and Restarting the <code>smbd</code> Daemon	481
Mounting Samba Shares	482
Network and Remote Printing with Ubuntu	483
Creating Network Printers	483
Using the Common UNIX Printing System GUI	485
Avoiding Printer Support Problems	486
References	488
24 Apache Web Server Management	489
About the Apache Web Server	489
Installing the Apache Server	490
Starting and Stopping Apache	491
Runtime Server Configuration Settings	492
Runtime Configuration Directives	492
Editing <code>apache2.conf</code>	493
Apache Multiprocessing Modules	495
Using <code>.htaccess</code> Configuration Files	496
File System Authentication and Access Control	498
Restricting Access with <code>Require</code>	498
Authentication	499
Final Words on Access Control	501
Apache Modules	502
<code>mod_access</code>	502
<code>mod_alias</code>	502
<code>mod_asis</code>	503
<code>mod_auth</code>	503
<code>mod_auth_anon</code>	503
<code>mod_auth_dbm</code>	503
<code>mod_auth_digest</code>	504
<code>mod_autoindex</code>	504
<code>mod_cgi</code>	504

mod_dir and mod_env	504
mod_expires	504
mod_headers	504
mod_include	505
mod_info and mod_log_config	505
mod_mime and mod_mime_magic	505
mod_negotiation	505
mod_proxy	505
mod_rewrite	505
mod_setenvif	506
mod_speling	506
mod_status	506
mod_ssl	506
mod_unique_id	506
mod_userdir	506
mod_usertrack	507
mod_vhost_alias	507
Virtual Hosting	507
Address-Based Virtual Hosts	507
Name-Based Virtual Hosts	508
Logging	509
HTTPS	510
References	513
25 Nginx Web Server Management	515
About the Nginx Web Server	515
Installing the Nginx Server	517
Installing from the Ubuntu Repositories	517
Building the Source Yourself	517
Configuring the Nginx Server	518
Virtual Hosting	521
Setting Up PHP	522
Adding and Configuring Modules	523
HTTPS	524
References	526
26 Other HTTP Servers	527
lighttpd	527
Yaws	528
Cherokee	528
Jetty	529

thttpd	529
Apache Tomcat	530
References	530
27 Remote File Serving with FTP	531
Choosing an FTP Server	531
Choosing an Authenticated or Anonymous Server	532
Ubuntu FTP Server Packages	532
Other FTP Servers	532
Installing FTP Software	533
The FTP User	534
Configuring the Very Secure FTP Server	536
Controlling Anonymous Access	537
Other <code>vsftpd</code> Server Configuration Files	537
Using the <code>ftphosts</code> File to Allow or Deny	
FTP Server Connection	539
References	540
28 Handling Email	541
How Email Is Sent and Received	541
The Mail Transport Agent	542
Choosing an MTA	544
The Mail Delivery Agent	544
The Mail User Agent	545
Basic Postfix Configuration and Operation	546
Configuring Masquerading	548
Using Smart Hosts	549
Setting Message Delivery Intervals	549
Mail Relaying	550
Forwarding Email with Aliases	550
Using Fetchmail to Retrieve Mail	551
Installing Fetchmail	551
Configuring Fetchmail	551
Choosing a Mail Delivery Agent	555
Procmail	555
Spamassassin	555
Squirrelmail	556
Virus Scanners	556
Autoresponders	556
Alternatives to Microsoft Exchange Server	556
Microsoft Exchange Server/Outlook Client	557
CommuniGate Pro	557

Oracle Beehive	557
Bynari	558
Open-Xchange	558
Horde	558
References	558
29 Proxying, Reverse Proxying, and Virtual Private Networks (VPN)	561
What Is a Proxy Server?	561
Installing Squid	562
Configuring Clients	562
Access Control Lists	563
Specifying Client IP Addresses	567
Sample Configurations	568
Virtual Private Networks (VPN)	570
Setting Up a VPN Client	571
Setting Up a VPN Server	573
References	575
30 Administering Relational Database Services	577
A Brief Review of Database Basics	578
How Relational Databases Work	580
Understanding SQL Basics	582
Creating Tables	582
Inserting Data into Tables	583
Retrieving Data from a Database	584
Choosing a Database: MySQL Versus PostgreSQL	586
Speed	586
Data Locking	586
ACID Compliance in Transaction Processing to Protect Data Integrity	587
SQL Subqueries	588
Procedural Languages and Triggers	588
Configuring MySQL	588
Setting a Password for the MySQL Root User	589
Creating a Database in MySQL	590
Configuring PostgreSQL	592
Initializing the Data Directory in PostgreSQL	592
Creating a Database in PostgreSQL	593
Creating Database Users in PostgreSQL	593
Deleting Database Users in PostgreSQL	594
Granting and Revoking Privileges in PostgreSQL	594

Database Clients	595
SSH Access to a Database	595
Local GUI Client Access to a Database	597
Web Access to a Database	597
The MySQL Command-Line Client	598
The PostgreSQL Command-Line Client	600
Graphical Clients	600
References	601
31 NoSQL Databases	603
Key/Value Stores	606
Berkeley DB	606
Cassandra	607
Memcached and MemcacheDB	607
Redis	608
Riak	608
Document Stores	608
CouchDB	609
MongoDB	610
BaseX	610
Wide Column Stores	611
BigTable	611
HBase	611
Graph Stores	612
Neo4j	612
OrientDB	612
HyperGraphDB	612
FlockDB	613
References	613
32 Lightweight Directory Access Protocol (LDAP)	615
Configuring the Server	616
Creating Your Schema	616
Populating Your Directory	617
Configuring Clients	619
Evolution	620
Thunderbird	621
Administration	621
References	622

33 Linux Terminal Server Project (LTSP)	623
Requirements	624
Installation	627
Using LTSP	628
References	629
34 Virtualization on Ubuntu	631
KVM	633
VirtualBox	637
VMware	639
Xen	639
References	639
35 Ubuntu in the Cloud	641
Why a Cloud?	642
Software as a Service (SaaS)	643
Platform as a Service (PaaS)	643
Infrastructure as a Service (IaaS)	643
Metal as a Service (MaaS)	643
Before You Do Anything	644
Deploy/Install Basics: Public, Private, or Hybrid?	644
Ubuntu Cloud and OpenStack	645
Compute Infrastructure (Nova)	645
Storage Infrastructure (Swift)	646
Networking Service (Neutron)	646
Identity Service (Keystone)	646
Imaging Service (Glance)	647
Dashboard (Horizon)	647
Learning More	647
Juju	647
Getting Started	648
Charms	650
The Juju GUI	652
Juju Quickstart	653
Juju on Mac OS X and Windows	653
Mojo: Continuous Delivery for Juju	653
Snappy Ubuntu Core	653
Ubuntu Metal as a Service (MaaS)	653
Landscape	654
References	654

36 Managing Sets of Servers	655
Juju	655
Puppet	656
Chef	656
CFEngine	656
Ansible	657
Landscape	657
References	657
37 Name Serving with the Domain Name System (DNS)	659
Understanding Domain Names	661
DNS Servers	661
DNS Records	662
Setting Up a DNS Server with BIND	665
References	667
Part V Programming Linux	
38 Using Programming Tools for Ubuntu	669
Programming with Linux	670
Using the C Programming Project Management Tools Provided with Ubuntu	671
Building Programs with <code>make</code>	671
Using Makefiles	671
Using the <code>autoconf</code> Utility to Configure Code	673
Debugging Tools	674
Using the GNU C Compiler	675
Graphical Development Tools	676
Using the KDevelop Client	676
The Glade Client for Developing in GNOME	677
Use an IDE or SDK	678
References	680
39 Opportunistic Development	681
Version Control Systems	681
Managing Software Projects with Git	682
Managing Software Projects with Bazaar	683
Managing Software Projects with Subversion	684
Managing Software Projects with Mercurial	685
Introduction to Opportunistic Development	686
Launchpad	687

Ubuntu Make	688
Creating Snap Packages	689
Bikeshed and Other Tools	689
References	692
40 Helping with Ubuntu Development	693
Introduction to Ubuntu Development	694
Setting Up Your Development System	695
Install Basic Packages and Configure	695
Create a Launchpad Account	696
Set Up Your Environment to Work with Launchpad	696
Developing Apps and Scopes	698
Fixing Bugs and Packaging	698
Finding Bugs to Fix with Harvest	701
Masters of the Universe	701
References	702
41 Helping with Ubuntu Testing and QA	703
Community Teams	703
Ubuntu Testing Team	704
QA Team	705
Bug Squad	705
Test Drive	705
References	708
42 Using Popular Programming Languages	709
Ada	710
Clojure	710
COBOL	711
D	712
Dart	712
Elixir	713
Erlang	713
Forth	713
Go	714
Fortran	714
Groovy	715
Haskell	715
Java	715
JavaScript	716
Lisp	716

Lua	717
Mono	717
OCaml	718
Perl	718
PHP	719
Python	719
Ruby	719
Rust	720
Scala	720
Scratch	720
Vala	720
References	721
43 Beginning Mobile Development for Android	723
Introduction to Android	724
Hardware	724
Linux Kernel	724
Libraries	724
Android Runtime	724
Application Framework	725
Applications	725
Installing Android Studio	725
Install Android Studio	725
Install SDK Packages	725
Create Your First Application	727
References	728
44 Developing for Ubuntu Mobile/Touch	729
Install the SDK	730
Create Your First Application	730
References	731
Index	733

NOTE

Chapters 45–47 can be accessed online at informit.com/title/9780134268118.

Bonus Chapters

45 Using Perl	Web:1
Using Perl with Linux	Web:1
Perl Versions	Web:2
A Simple Perl Program	Web:2

Perl Variables and Data Structures	Web:4
Perl Variable Types	Web:5
Special Variables	Web:5
Operators	Web:6
Comparison Operators	Web:6
Compound Operators	Web:7
Arithmetic Operators	Web:7
Other Operators	Web:8
Special String Constants	Web:8
Conditional Statements: <code>if/else</code> and <code>unless</code>	Web:9
<code>if</code>	Web:9
<code>unless</code>	Web:10
Looping	Web:10
<code>for</code>	Web:10
<code>foreach</code>	Web:10
<code>while</code>	Web:11
<code>until</code>	Web:11
<code>last</code> and <code>next</code>	Web:12
<code>do ... while</code> and <code>do ... until</code>	Web:12
Regular Expressions	Web:12
Access to the Shell	Web:13
Modules and CPAN	Web:14
Code Examples	Web:15
Sending Mail	Web:15
Purging Logs	Web:17
Posting to Usenet	Web:18
One-Liners	Web:19
Command-Line Processing	Web:20
References	Web:20
46 Using Python	Web:23
Python on Linux	Web:24
The Basics of Python	Web:25
Numbers	Web:25
More on Strings	Web:27
Lists	Web:30
Dictionaries	Web:32
Conditionals and Looping	Web:33
Functions	Web:35
Object Orientation	Web:36

Class and Object Variables	Web:37
Constructors and Destructors	Web:38
Class Inheritance	Web:39
The Standard Library and the Python Package Index	Web:41
References	Web:41
47 Using PHP	Web:43
Introduction to PHP	Web:44
Entering and Exiting PHP Mode	Web:44
Variables	Web:44
Arrays	Web:46
Constants	Web:47
References	Web:48
Comments	Web:48
Escape Sequences	Web:49
Variable Substitution	Web:50
Operators	Web:51
Conditional Statements	Web:53
Special Operators	Web:54
Switching	Web:55
Loops	Web:56
Including Other Files	Web:59
Basic Functions	Web:60
Strings	Web:60
Arrays	Web:63
Files	Web:65
Miscellaneous	Web:67
Handling HTML Forms	Web:70
Databases	Web:71
References	Web:73

About the Author

Matthew Helmke is an active member of the Ubuntu community. He served from 2006 to 2011 on the Ubuntu Forum Council, providing leadership and oversight of the Ubuntu Forums (www.ubuntuforums.org), and spent two years on the Ubuntu regional membership approval board for Europe, the Middle East, and Africa. He has written about Ubuntu for several magazines and websites, is a lead author of *The Official Ubuntu Book*, and coauthored *The VMware Cookbook*. He works as a technical writer for Canonical, Inc., documenting cloud- and DevOps-related software. Matthew first used Unix in 1987 while studying LISP on a VAX at the university. He has run a business using only free- and open-source software, has consulted, and has a master's degree in Information Resources and Library Science from the University of Arizona. You can find out more about Matthew at matthewhelmke.com or drop him a line with errata or suggestions at matthew@matthewhelmke.com.

Andrew Hudson is a freelance journalist who specializes in writing about Linux. He has significant experience in Red Hat- and Debian-based Linux distributions and deployments and can often be found sitting at his keyboard tweaking various settings and config files just for the hell of it. He lives in Wiltshire, which is a county of England, along with his wife, Bernice, and their son, John. Andrew does not like Emacs. He can be reached at andy.hudson@gmail.com.

Paul Hudson is a recognized expert in open source technologies. He is a professional developer and full-time journalist for Future Publishing. His articles have appeared in *MacFormat*, *PC Answers*, *PC Format*, *PC Plus*, and *Linux Format*. Paul is passionate about free software in all its forms and uses a mix of Linux and BSD to power his desktops and servers. Paul likes Emacs. Paul can be contacted through <http://hudzilla.org>.

Dedication

To Saralyn, Sedona, and Philip—the most amazing kids a guy could hope for; to Sandra and Evan, who are wonderful and welcome additions to our lives; to my grandfather for always believing in me and for teaching me to believe in myself; and to my friends in the Ubuntu, developer, sysadmin, cloud computing, and DevOps communities.

Acknowledgments

Matthew wishes to thank the many people who helped with past editions, with helpful comments and ideas, with technical edits, and with both formal and informal advice. I owe a huge debt of gratitude to the Ubuntu community, Canonical, and Mark Shuttleworth for inviting me to participate in the community, including my role in the forums, a turn on the EMEA membership board, and two Ubuntu Developer Summits, back when we had to travel to be a part of them. Thanks to the Ubuntu All Stars for the chance to jam with you on guitar. Thank you to the entire Ubuntu community for your labor of love to create this wonderful operating system. Finally, thanks to my colleagues at Pearson, especially Debra Williams Cauley, for the trust placed in me and the opportunity to collaborate on projects like this one.

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: consumer@samspublishing.com

Mail: Sams Publishing
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at informit.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

This page intentionally left blank

Introduction

IN THIS INTRODUCTION

- ▶ Licensing
- ▶ Who This Book Is For
- ▶ What This Book Contains
- ▶ Conventions Used in This Book

We are pleased to present the 2017 edition of *Ubuntu Unleashed*. Ubuntu is a Linux-based computer operating system that has taken the world by storm. From its humble beginning in 2004, Ubuntu has risen to be the vanguard of desktop Linux, as well as a popular choice for servers.

Ubuntu descends from one of the oldest and most revered Linux distributions, Debian. Debian is assembled by a team of talented volunteers, is one of the most stable and customizable distributions of Linux, and is well respected for its quality and technological prowess. It is, however, an operating system for geeks; the bar for entry into the Debian realm is set high, and its userbase tends to be highly proficient and expects new users to learn the ropes before joining in. That is both appropriate and okay.

What Ubuntu has done is leverage the quality of Debian to create an operating system that ordinary people can use. That doesn't mean that Ubuntu users are not technically proficient, just that they do not have to be. In fact, many talented and respected software developers love Ubuntu because it enables them to concentrate on their specific interests instead of the details of the operating system. This book is for these people and for those who aspire to join their ranks.

If you are new to Linux, you have made a great decision by choosing this book. Sams Publishing's *Unleashed* books offer an in-depth look at their subjects, taking in both beginner and advanced users and moving them to a new level of knowledge and expertise. Ubuntu is a fast-changing distribution that has an updated release twice a year. We have tracked the development of Ubuntu from early on to make sure that the information in this book mirrors closely the

development of the distribution. A full copy of Ubuntu is included on the enclosed disc, and it is possible for you to install Ubuntu from that disc in less than an hour!

A QUICK WORD ABOUT MARKETING

Almost all of the content in this book applies regardless of what Ubuntu release version you are using, so long as it is reasonably current. The book has been written to try to focus on information that is useful for the longest amount of time possible. Some chapters, like those covering installation or the basics of the default Ubuntu graphical user interface, will have their information change frequently. Those chapters are the exception. The blurb on the cover of the book about which editions this book covers was added to account for these chapters and to denote clearly when the book was most recently revised.

Do not let the highly technical reputation of Linux discourage you, however. Many people who have heard of Linux think that it is found only on servers, looking after websites and email. Nothing could be further from the truth. Distributions like Ubuntu are making huge inroads in to the desktop market. Corporations are realizing the benefits of running a stable and powerful operating system that is easy to maintain and easy to secure. The best part is that as Linux distributions make improvements, the majority of those improvements are shared freely, allowing you to benefit from the additions and refinements made by one distribution, such as Red Hat, while continuing to use a different distribution, such as Ubuntu, which in turn shares its improvements. You can put Ubuntu to work today and be assured of a great user experience. Feel free to make as many copies of the software as you want; Ubuntu is freely and legally distributable all over the world—no copyright lawyers are going to pound on your door.

Licensing

Software licensing is an important issue for all computer users and can entail moral, legal, and financial considerations. Many consumers think that purchasing a copy of a commercial or proprietary operating system, productivity application, utility, or game conveys ownership, but this is not true. In the majority of cases, the *end user license agreement (EULA)* included with a commercial software package states that you have paid only for the right to use the software according to specific terms. This generally means you may not examine, make copies, share, resell, or transfer ownership of the software package. More onerous software licenses enforce terms that preclude you from distributing or publishing comparative performance reviews of the software. Even more insidious licensing schemes (and supporting legislation, especially in the United States) contain provisions allowing onsite auditing of the software's use!

This is not the case with the software included with this book. You are entirely free to make copies, share copies, and install the software on as many computers as you want—we encourage you to purchase additional copies of this book to give as gifts, however. Be sure to read the README file on the disc included with this book for important information regarding the included software and disk contents. After you

install Ubuntu, go to www.gnu.org/licenses/gpl.html to find a copy of the GNU GPL. You will see that the GPL provides unrestricted freedom to use, duplicate, share, study, modify, improve, and even sell the software.

You can put your copy of Ubuntu to work right away in your home or at your place of business without worrying about software licensing, per-seat workstation or client licenses, software auditing, royalty payments, or any other type of payments to third parties. However, be aware that although much of the software included with Ubuntu is licensed under the GPL, some packages on this book's disc are licensed under other terms. There is a variety of related software licenses, and many software packages fall under a broad definition known as *open source*. Some of these include the Artistic License, the BSD License, the Mozilla Public License, and the Q Public License.

For additional information about the various GNU software licenses, browse to www.gnu.org/. For a definition of open source and licensing guidelines, along with links to the terms of nearly three dozen open-source licenses, browse to www.opensource.org/.

Who This Book Is For

This book varies its coverage from deep to shallow over its wide range of topics. This is intentional. There are some topics that are Ubuntu-specific and are not covered by any other book, and so deserve deep coverage here. There are some topics that every power user really must master. There are other topics that power users should know about, so that they understand some history, know some other options, or simply have what they need to be able to listen and participate in further discussions with other technical people without being completely confused.

Some topics, like using the Linux command line, receive deep and extensive coverage because I believe that information to be vital to anyone who wants to be a power user or become a skilled DevOps guru. That topic gets two full chapters.

Other topics, like the chapter that mentions ADA and Fortran, along with more than 15 other programming languages, only get brief coverage so that people who are interested get a few guideposts to help them continue if they are interested. In this case, around 20 programming languages are covered in about a dozen pages. These are useful topics to some, but not topics I would consider vital.

Additionally, some topics are just too broad to be covered in great depth in this book, but are topics that deserve a mention because, again, an intermediate to advanced user should have at least a foundational knowledge of them. These are covered and then information is provided to help you find more resources and expand your understanding, as needed.

Those Wanting to Become Intermediate or Advanced Users

Ubuntu Unleashed is intended for intermediate and advanced users or those who want to become one. Our goal is to give you a nudge in the right direction, to help you enter the higher stages by exposing you to as many different tools and ideas as possible; we want to give you some thoughts and methods to consider and spur you on to seek out more.

Although the contents are aimed at intermediate to advanced users, new users who pay attention will benefit from the advice, tips, tricks, traps, and techniques presented in each chapter. Pointers to more detailed or related information are also provided at the end of each chapter.

If you are new to Linux, you might need to learn some new computer skills, such as how to research your computer's hardware, how to partition a hard drive, and (occasionally) how to use a command line. This book helps you learn these skills and shows you how to learn more about your computer, Linux, and the software included with Ubuntu. Most important, it helps you overcome your fear of the system by telling you more about what it is and how it works.

We would like to take a moment to introduce a concept called "The Three Levels of Listening" from Alistair Cockburn's *Agile Software Development*, published by Addison Wesley. These describe how a person learns and masters a technique. We all start at the first stage and progress from there. Few reach the last stage, but those who do are incredibly effective and efficient. People aiming for this stage are the very ones for whom we intend this book.

- ▶ **Following**—The stage where the learner looks for one very detailed process that works and sticks to it to accomplish a task.
- ▶ **Detaching**—The stage where the learner feels comfortable with one method and begins to learn other ways to accomplish the same task.
- ▶ **Fluent**—The stage where the learner has experience with or understanding of many methods and doesn't think of any of them in particular while doing a task.

Myriad books focus on the first set of users. This is not one of them. It is our goal in *Ubuntu Unleashed* to write just enough to be sufficient to get you from where you are to where you want or need to be. This is not a book for newcomers who want or need every step outlined in detail, although we do that occasionally. This is a book for people who want help learning about what can be done and a way to get started doing it. The Internet is an amazing reference tool, so this is not a comprehensive reference book. This book is a tool to help you see the landscape; to learn enough about what you seek to get you started in the right direction with a quality foundational understanding.

Sysadmins, Programmers, and DevOps

Systems administrators, or Sysadmins, are the people who keep servers and networks up and running. Their role is sometimes called *operations*. They deal with software installation and configuration, security, and do all the amazing things behind the scenes that let others use these systems for their work. They are often given less respect than they deserve, but the pay is good and it is a ton of fun to wield the ultimate power over a computer system. It is also a great responsibility, and these amazing guys and gals work hard to make sure they do their jobs well, striving for incredible system uptime and availability. Ubuntu is an excellent operating system for servers and networks, and in this book you can find much of the knowledge needed to get started in this role.

Programmers are the people who write software. They are sometimes called *developers*. Programmers work with others to create the applications that run on top of those systems. Ubuntu is a great platform for writing and testing software. This is true whether you are doing web application development or writing software for desktop or server systems. It also makes a great platform for learning new programming languages and trying out new ideas. This book can help you get started.

DevOps is a portmanteau of *developer* and *operations*. It signifies a blending of the two roles already described. The information technology (IT) world is changing, and roles are becoming less clear cut and isolated from one another. In the past, it was common to witness battles between programmers excited about new technology and sysadmins in love with stability. DevOps realizes that neither goal is healthy in isolation, but that seeking a balance between the two can yield great results by removing the barriers to communication and understanding that sometimes cause conflict within a team. Because of the rise of cloud computing and virtualization, which are also covered in this book, and more agile forms of development, DevOps is a useful perspective that enables people working in IT to do an even better job of serving their ultimate clients: end users. This book is a great foundation for those wanting to learn knowledge that will help with both roles, hopefully presented in a way that balances them nicely.

What This Book Contains

Ubuntu Unleashed is organized into six parts, described here. A disc containing the entire distribution is included so that you have everything you need to get started.

Part I, “Installation and Configuration” takes you through installing Ubuntu on your computer in the place of any other operating system you might be running, such as Windows.

Part II, “Desktop Ubuntu,” is aimed at users who want to use Ubuntu on desktop systems.

Part III, “System Administration,” covers both elementary and sophisticated details of setting up a system for specific tasks and maintaining that system.

Part IV, “Ubuntu as a Server,” gives you the information you need to start building your own file, web, and other servers for use in your home or office.

Part V, “Programming Linux,” provides a great introduction to how you can extend Ubuntu capabilities even further using the development tools supplied with it.

In addition to what has already been mentioned, after the spring release of Ubuntu, bonus chapters will be available online at www.informit.com/title/9780134511184.

In addition, this book is part of InformIT’s exciting Content Update Program, which provides content updates for major technology improvements! As significant updates are made to Ubuntu, sections of this book will be updated or new sections will be added to match the updates to the technologies. As updates become available, they will be delivered to you via a free Web Edition of this book, which can be accessed with any Internet connection. To learn more, visit informit.com/cup.

How to access the Web Edition: Follow the instructions inside to learn how to register your book to access the FREE Web Edition.

Conventions Used in This Book

It is impossible to cover every option of every command included in Ubuntu. Besides, with the rise of the Internet and high-speed connections, reference materials are far less valuable than they used to be because most of these details are only a quick Google search away. Instead, we focus on teaching you how to find information you need while giving a quality overview worthy of the intermediate or advanced user. Sometimes this book offers tables of various options, commands, and keystrokes to help condense, organize, and present information about a variety of subjects.

To help you better understand code listing examples and sample command lines, several formatting techniques are used to show input and ownership. For example, if the command or code listing example shows typed input, the input is formatted in boldface after the sample command prompt, as follows:

```
matthew@seymour:~$ ls
```

If typed input is required, as in response to a prompt, the sample typed input also is in boldface, like so:

```
Delete files? [Y/n] y
```

All statements, variables, and text that should appear on your display use the same bold-face formatting. In addition, command lines that require root or super-user access are prefaced with the sudo command, as follows:

```
matthew@seymour:~$ sudo printtool &
```

Other formatting techniques include the use of italic for placeholders in computer command syntax. Computer terms or concepts are also italicized upon first introduction in text.

Finally, you should know that all text, sample code, and screenshots in *Ubuntu Unleashed* were developed using Ubuntu and open-source tools.

Read on to start learning about and using the latest version of Ubuntu.

CHAPTER 9

Managing Software

In this chapter, we look at the options you have to manage your software in Ubuntu. If you are used to an environment where you are reliant on visiting different vendor websites to download updates, you are in for a pleasant surprise. Updating a full Ubuntu installation, including all the application software, is as simple as running the Update Manager program. You will discover just how easy it is to install and even remove various software packages.

Ubuntu provides a variety of tools for system resource management. The following sections introduce the graphical software management tools that you will use for most of your software management. This chapter also covers monitoring and managing memory and disk storage on your system.

Ubuntu Software

Ubuntu Software is a graphical utility for package management in Ubuntu. You can find it in the Applications menu as Ubuntu Software; the package and executable program is named `ubuntu-software`. Ubuntu Software enables you to easily select and install a large array of applications by using the intuitive built-in search and easy one-click installation. When you open the program, you see the main screen, as shown in Figure 9.1.

Along the top side of the screen, you have three menu options: All, Installed, and Updates. Just below that is a search bar you can use to search for packages. Scroll down to find software listed by categories.

IN THIS CHAPTER

- ▶ Ubuntu Software
- ▶ Using Synaptic for Software Management
- ▶ Staying Up-to-Date
- ▶ Working on the Command Line
- ▶ Compiling Software from Source
- ▶ Configuration Management
- ▶ Snappy Ubuntu Core
- ▶ Using Snaps
- ▶ References



FIGURE 9.1 The initial Ubuntu Software screen enables you to browse through packages sorted by groups.

Installing new software via Ubuntu Software is as simple as finding it in the package list, double-clicking, and clicking the Install button. When you do so, you may be asked for your password; then the application is downloaded and installed. You can remove an application by finding it in Ubuntu Software and clicking the Remove button.

Using Synaptic for Software Management

Ubuntu Software works just fine for adding and removing applications, but if you need to install something specific—such as a library—you need to use Synaptic (Figure 9.2). You can install Synaptic using Ubuntu Software described earlier; it is not installed by default.

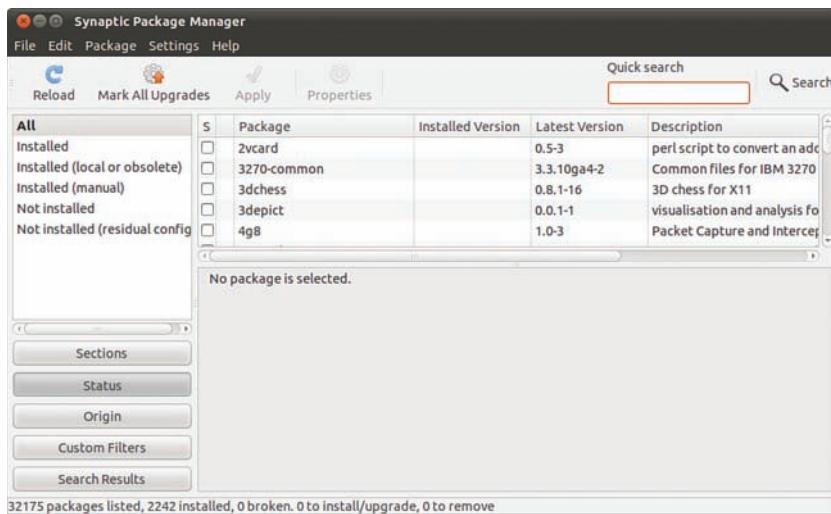


FIGURE 9.2 For more advanced software management in a GUI, Synaptic is the preferred tool.

Along the left are software categories (although this time there are more of them), along the top right are the package selections for that category, and on the bottom right is the Package Information window that shows information about the currently selected package. To install or remove software, click the check box to the left of its name, and you'll see a menu that offers the following options:

- ▶ **Unmark**—If you have marked this package for installation, upgrade, or one of the other options, this option removes that mark.
- ▶ **Mark for Installation**—Add this package to the list that will be installed.
- ▶ **Mark for Re-installation**—If you have some software already installed, but for some reason it's not working, this option reinstalls it from scratch. Existing configuration files are retained, so any edits you have made are safe.
- ▶ **Mark for Upgrade**—If the software has updates available, this option downloads and installs them.
- ▶ **Mark for Removal**—This option deletes the selected package from your system but leaves its configuration files intact so that if you ever reinstall it you do not have to reconfigure it.
- ▶ **Mark for Complete Removal**—This option deletes the selected package from your system but also removes any configuration files, purging everything from the system.

After you have made your changes, click the Apply button to have Synaptic download, install, upgrade, and uninstall as necessary. If you close the program without clicking Apply, your changes are lost.

Beneath the categories on the left side of the screen, you see six buttons: Sections, Status, Origin, Custom Filters, Search Results, and Architecture. These customize the left list: Sections is the Categories view; Status enables you to view packages that are installed or upgradable; Origin lists the different repositories available to download packages; Custom Filters has some esoteric groupings that are useful only to advanced users, Search Results stores results of your searches; and Architecture shows the packages specific to each architecture of Ubuntu.

You can press Ctrl+F at any time to search for a particular package. By default, it is set to search by package description and name. You may change the Look In box setting to only search for Name. As mentioned already, your search terms are saved under the Search view (the button on the bottom left), and you can click from that list to re-search on that term.

As well as providing the method of installing and removing software, Synaptic provides the means to configure the servers you want to use for finding packages. In fact, this is where you can make one of the most important changes to your Ubuntu system: You can open it up to the Ubuntu Universe and Multiverse.

Ubuntu is based on the Debian distribution, which has thousands software packages available for installation. Ubuntu uses only a subset of that number but makes it easy for you to install the others, along with many packages that are not available in Debian. When you use Synaptic, you see small orange Ubuntu logos next to many packages; this identifies them as being officially supported by the Canonical-supported Ubuntu developers. The packages that do not have this logo are supported by the wider Ubuntu community of developers.

To enable the Universe and Multiverse repositories, go to Settings, Repositories. This list shows all the servers you have configured for software installation and updates and includes the Universe and Multiverse repositories. When you find them, check them, and then click Close.

Synaptic shows a message box warning you that the repository listings have changed and that you need to click the Reload button (near the top left of the Synaptic window) to have it refresh the package lists. Go ahead and do that, and you should see a lot more software appear for your selection. However, notice that only a small number have the official Ubuntu “seal” attached, which means you may want to be a bit more careful when installing software.

NOTE

Much of the software discussed in this book is available only through the Universe repository. Therefore, we highly recommend enabling it to get full use out of this book and your Ubuntu installation.

Staying Up-to-Date

Although you can manage your software updates through Synaptic, Ubuntu provides a dedicated tool called Software Updater (shown in Figure 9.3). This tool is designed to be simple to use: When you run it, Software Updater automatically downloads the list of updates available and checks them all in the list it shows. If the update list was downloaded automatically not too long ago, you can force Ubuntu to refresh the list of available updates by clicking the Check button. Otherwise, all you need to do is click Install Updates to bring your system up to date. If you want a little more information about the updates, click Show Details at the bottom to see what has changed in the update.

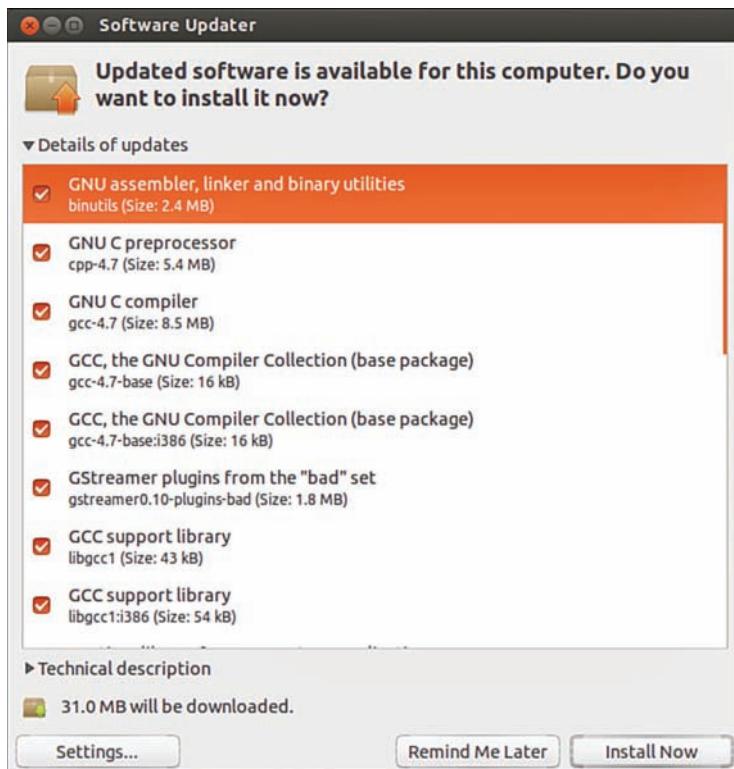


FIGURE 9.3 If you need to update your software to apply bug fixes and security upgrades, use Software Updater.

Ubuntu automatically checks for updates periodically and notifies you when critical updates are available. However, there's no harm running Software Updater yourself every so often, just to make sure; it's better to be safe than sorry.

Working on the Command Line

With so much software available for installation, it is no surprise that Debian-based distros have many ways to manage software installation. At their root, however, they all use Debian's world-renowned *Advanced Package Tool (APT)*. A person posting on Slashdot.com once said, "Welcome to Slashdot. If you can't think of anything original, just say how much APT rocks and you'll fit right in." You see, even though many other distros have tried to equal the power of APT, nothing else even comes close.

Why is APT so cool? Well, it was the first system to properly handle dependencies in software. Other distros, such as Red Hat, used RPM files that had dependencies. For example, an RPM for Gimp would have a dependency on Gtk, the graphical toolkit on which Gimp is based. As a result, if you tried to install your Gimp RPM without having the Gtk RPM, your install would fail. So, you grab the Gtk RPM and try again. Aha: Gtk has a dependency on three other things that you need to download. And those three other things have dependencies on 20 other things. And so on, and so on, usually until you can't find a working RPM for one of the dependencies, and you give up.

APT, on the other hand, was designed to automatically find and download dependencies for your packages. So, if you want to install Gimp, it downloads Gimp's package and any other software it needs to work. No more hunting around by hand, no more worrying about finding the right version, and certainly no more need to compile things by hand. APT also handles installation resuming, which means that if you lose your Internet connection part-way through an upgrade (or your battery runs out, or you have to quit, or whatever), APT picks up where it left off the next time you rerun it.

Day-to-Day Usage

To enable you to search for packages both quickly and thoroughly, APT uses a local cache of the available packages. Try running this command:

```
matthew@seymour:~$ sudo apt-get update
```

The `apt-get update` command instructs APT to contact all the servers it is configured to use and download the latest list of file updates. If your lists are outdated, it takes a minute or two for APT to download the updates. Otherwise, this command executes it in a couple of seconds.

After the latest package information has been downloaded, you are returned to the command line. You can now ask APT to automatically download any software that has been updated, using this command:

```
matthew@seymour:~$ sudo apt-get upgrade
```

If you have a lot of software installed on your machine, there is a greater chance of things being updated. APT scans your software and compares it to the latest package information from the servers and produces a report something like this:

```
mmatthew@seymour:~$ sudo apt-get upgrade
Reading package lists... Done
```

```
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  cabextract google-chrome-beta icedtead-plugin language-pack-en
  language-pack-en-base language-pack-gnome-en language-pack-gnome-en-base
  libfreetype6 libfreetype6-dev libsmbclient libwbclient0 openjdk-6-jre
  openjdk-6-jre-headless openjdk-6-jre-lib samba-common samba-common-bin
  smbclient upstart winbind xserver-common xserver-xorg-core
21 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 84.8MB of archives.
After this operation, 623kB of additional disk space will be used.
Do you want to continue [Y/n] ?
```

Each part of that report tells you something important. Starting at the top, the line “the following packages will be upgraded” gives you the exact list of packages for which updates are available. If you’re installing new software or removing software, you see lists titled “The following packages will be installed” and “The following packages will be removed.” A summary at the end shows a total of 21 packages that APT will upgrade, with 0 new packages, 0 to remove, and 0 not upgraded. Because this is an upgrade rather than an installation of new software, all those new packages take up only 623KB of additional space. Although you have an 84.8MB download, the packages are overwriting existing files.

It’s important to understand that a basic `apt-get upgrade` never removes software or adds new software. As a result, it is safe to use to keep your system fully patched because it should never break things. However, occasionally you will see the “0 not upgraded” status change, which means some things cannot be upgraded. This happens when some software must be installed or removed to satisfy the dependencies of the updated package, which, as previously mentioned, `apt-get upgrade` will never do.

In this situation, you need to use `apt-get dist-upgrade`, so named because it’s designed to allow users to upgrade from one version of Debian/Ubuntu to a newer version—an upgrade that inevitably involves changing just about everything on the system, removing obsolete software, and installing the latest features. This is one of the most-loved features of Debian because it enables you to move from version to version without having to download and install new CDs. Keeping regular upgrades and distro upgrades separate is very useful for making sure that security updates and simple bug fixes don’t change software configurations that you may be counting on, especially on a machine that needs to be consistently available and working, such as a server.

Whereas `apt-get upgrade` and `apt-get dist-upgrade` are there for upgrading packages, `apt-get install` is responsible for adding new software. For example, if you want to install the MySQL database server, you run this:

```
matthew@seymour:~$ sudo apt-get install mysql-server
```

Internally, APT queries “mysql-server” against its list of software and find that it matches the mysql-server-5.5 package. It then finds which dependencies it needs that you don’t already have installed and gives you a report like this one:

```
matthew@seymour:~$ sudo apt-get install mysql-server
[sudo] password for matt:

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
 libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
 libnet-daemon-perl libplrpc-perl libterm-readkey-perl mysql-client-5.5
 mysql-client-core-5.5 mysql-server-5.5 mysql-server-core-5.5
Suggested packages:
 libipc-sharedcache-perl tinyca mailx
The following NEW packages will be installed:
 libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
 libnet-daemon-perl libplrpc-perl libterm-readkey-perl mysql-client-5.5
 mysql-client-core-5.5 mysql-server mysql-server-5.5 mysql-server-core-5.5
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 26.8 MB of archives.
After this operation, 96.2 MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

This time, you can see that APT has picked up and selected all the dependencies required to install MySQL Server 5.5, but it has also listed one recommended package and two suggested packages that it has not selected for installation. The “recommended” package is just that: The person who made the MySQL package (or its dependencies) thinks it would be a smart idea for you to also have the mailx package. If you want to add it, press N to terminate apt-get and rerun it like this:

```
matthew@seymour:~$ sudo apt-get install mysql-server mailx
```

The “suggested” packages are merely a lower form of recommendation. They don’t add any crucial features to the software you selected for install, but it’s possible that you might need them for certain non-crucial (to the main piece of software being installed) features or tasks.

NOTE

APT maintains a package cache where it stores DEB files it has downloaded and installed. This usually lives in /var/cache/apt/archives and can sometimes take up many hundreds of megabytes on your computer. You can have APT clean out the package cache by running apt-get clean, which deletes all the cached DEB files. Alternatively, you can run apt-get autoclean, which deletes cached DEB files that are beyond a certain age, thereby keeping newer packages.

If you try running `apt-get install` with packages you already have installed, APT considers your command to be `apt-get update` and looks to see whether new versions are available for download.

The last day-to-day package operation is removing things you no longer want, which you do through the `apt-get remove` command, as follows:

```
matthew@seymour:~$ sudo apt-get remove firefox
```

Removing packages can be dangerous because APT also removes any software that relies on the packages you selected. For example, if you were to run `apt-get remove libgtk2.0-0` (the main graphical toolkit for Ubuntu), you would probably find that APT insists on removing more than a hundred other things. The moral of the story is this: When you remove software, read the APT report carefully before pressing `y` to continue with the uninstall.

A straight `apt-get remove` leaves behind the configuration files of your program so that if you ever reinstall it you do not also need to reconfigure it. If you want to remove the configuration files as well as the program files, run this command instead:

```
matthew@seymour:~$ sudo apt-get remove --purge firefox
```

Or:

```
matthew@seymour:~$ sudo apt-get purge firefox
```

Either will perform a full uninstall.

NOTE

You can see a more extensive list of `apt-get` parameters by running `apt-get` without any parameters. The cryptic line at the bottom, “This APT has Super Cow Powers,” is made even more cryptic if you run the command `apt-get moo`.

Finding Software

With so many packages available, it can be hard to find the exact thing you need using command-line APT. The general search tool is called `apt-cache` and is used like this:

```
matthew@seymour:~$ apt-cache search kde
```

Depending on which repositories you have enabled, that tool returns about a thousand packages. Many of those results will not even have KDE in the package name but will be matched because the description contains the word *KDE*.

You can filter through this information in several ways. First, you can instruct `apt-cache` to search only in the package names, not in their descriptions. You do this with the `-n` parameter, like this:

```
matthew@seymour:~$ apt-cache -n search kde
```

Now the search has gone down from more than 1,000 packages to a few hundred.

Another way to limit search results is to use some basic regular expressions, such as ^, meaning “start,” and \$, meaning “end.” For example, you might want to search for programs that are part of the main KDE suite and not libraries (usually named something like `libkde`), additional bits (such as `xmms-kde`), and things that are actually nothing to do with KDE yet still match our search (like `tkdesk`). Do this by searching for packages that have a name starting with `kde`, as follows:

```
matthew@seymour:~$ apt-cache -n search ^kde
```

Perhaps the easiest way to find packages is to combine `apt-cache` with `grep`, to search within search results. For example, if you want to find all games-related packages for KDE, you could run this search:

```
matthew@seymour:~$ apt-cache search games | grep kde
```

When you’ve found the package you want to install, run it through `apt-get install` as usual. If you first want a little more information about that package, you can use `apt-cache showpkg`, like this:

```
matthew@seymour:~$ apt-cache showpkg mysql-server-5.0
```

This shows information on “reverse depends” (which packages require, recommend, or suggest `mysql-server-5.0`), “dependencies” (which packages are required, recommended, or suggested to install `mysql-server-5.0`), and “provides” (which functions this package gives you). The “provides” list is quite powerful because it allows different packages to provide a given resource. For example, a MySQL database-based program requires MySQL to be installed, but isn’t fussy whether you install MySQL 4.1 or MySQL 5.5. In this situation, the Debian packages for MySQL 4.1 and MySQL 5.0 both have “`mysql-server-4.1`” in the provides list, meaning that they offer the functionality provided by MySQL 4.1. Therefore, you can install either version to satisfy the MySQL-based application.

Using `apt` Instead of `apt-get`

There is a new, simplified interface to APT that removes the hyphen and the second part of the command. It also includes lovely updates like a progress bar. Although this is new, in testing for this edition of the book, it was stable and pleasant to use. Table 9.1 lists some of the new commands and what they replace. Both versions work, so no relearning is necessary if you do not want to switch.

TABLE 9.1 `apt-get` versus `apt`

<code>apt-get</code> Command	<code>apt</code> Command
<code>apt-get install</code>	<code>apt install</code>
<code>apt-get remove</code>	<code>apt remove</code>
<code>apt-get update</code>	<code>apt update</code>
<code>apt-get upgrade</code>	<code>apt upgrade</code>

apt-get dist-upgrade	apt full-upgrade
apt-get remove --purge	apt purge
apt-get autoremove	apt autoremove
apt-get purge	apt purge

Compiling Software from Source

Compiling applications from source is not that difficult. There are two ways to do this: You can use the source code available in the Ubuntu repositories, or you can use source code provided by upstream developers (most useful for those projects that are not available in the Ubuntu repositories). For either method, you need to install the `build-essential` package to ensure that you have the tools you need for compilation. You may also need to install `automake` and `checkinstall`, which are build tools.

Compiling from a Tarball

Most source code that is not in the Ubuntu repositories is available from the original writer or from a company's website as compressed source *tarballs*—that is, `tar` files that have been compressed using `gzip` or `bzip`. The compressed files typically uncompress into a directory containing several files. It is always a good idea to compile source code as a regular user to limit any damage that broken or malicious code might inflict, so create a directory named `source` in your home directory.

From wherever you downloaded the source tarball, uncompress it into the `~/source` directory using the `-C` option to `tar`:

```
matthew@seymour:~$ tar zxvf packagename.tgz -C ~/source
matthew@seymour:~$ tar zxvf packagename.tar.gz -C ~/source
matthew@seymour:~$ tar jxvf packagename.bz -C ~/source
matthew@seymour:~$ tar jxvf packagename.tar.bz2 -C ~/source
```

If you are not certain what file compression method was used, use the `file` command to figure it out:

```
matthew@seymour:~$ file packagename
```

Now, change directories to `~/source/packagename` and look for a file named `README`, `INSTALL`, or a similar name. Print out the file if necessary because it contains specific instructions on how to compile and install the software. Typically, the procedure to compile source code is as follows:

```
matthew@seymour:~/source/packagename$ ./configure
```

This runs a script to check whether all dependencies are met and the build environment is correct. If you are missing dependencies, the `configure` script normally tells you exactly which ones it needs. If you have the Universe and Multiverse repositories enabled in Synaptic, chances are you will find the missing software (usually libraries) in there.

When your configure script succeeds, run the following to compile the software:

```
matthew@seymour:~/source/packagename$ make
```

And finally, run the following:

```
matthew@seymour:~/source/packagename$ sudo make install
```

If the compile fails, check the error messages for the reason and run the following before you start again:

```
matthew@seymour:~/source/packagename$ make clean
```

You can also run the following to remove the software if you do not like it:

```
matthew@seymour:~/source/packagename$ sudo make uninstall
```

Compiling from Source from the Ubuntu Repositories

You might sometimes want to recompile a package, even though a binary package is available in the Ubuntu repositories. For example, a program might have been compiled into a binary with a specific feature disabled that you would like to use. Here is how you can do this. We will call the software package we want to compile *foo*.

First, get the source from the Ubuntu repositories:

```
matthew@seymour:~$ apt-get source foo
```

Install the build dependencies for the package:

```
matthew@seymour:~$ sudo apt-get build-dep foo
```

Change to the directory for the source code (may include the version number):

```
matthew@seymour:~$ cd foo-4.5.2
```

Make whatever changes you want to make to the package or to the compilation flags. You can do this using `./configure` and `make`, or sometimes by making manual changes to a configuration file. Each package has the potential to do this differently, so you need to see that program's documentation. Try looking for a `README` file in the source code to get started.

Next, create a new `debian/changelog` entry. After you enter this command, you need to enter a message that tells why a new version was made, perhaps something like *Matthew's flight of fancy with extra sauce*.

NOTE

Ubuntu package numbering follows a specific pattern. To help yourself later, you should stick to this pattern. Using the `foo` numbers shown here, a typical Ubuntu package that was inherited from Debian with no changes would then be `4.5.2-1`. A package inherited

from Debian, but changed for Ubuntu would be 4.5.2-1ubuntu1 (and then ubuntu2 for a second version, and so on). A package that did not have a version in Debian but which was created for Ubuntu would be 4.5.2-0ubuntu1 (and ubuntu2 and so on).

```
matthew@seymour:~$ dch -i
```

Build the source package. This creates all the files necessary for uploading a package:

```
matthew@seymour:~$ debuild -s
```

Finally, you are left with a `foo-4.5.2-1ubuntulcustom.deb` package (using whatever version number or suffix you created earlier) that you can install, and later uninstall as well, using your package manager. In some instances, multiple DEB files might be created, in which case you would replace the individual package name in the example here with `*.deb`.

```
matthew@seymour:~$ sudo dpkg -Oi foo-4.5.2-1ubuntulcustom.deb
```

Configuration Management

This section provides a quick introduction to a couple tools that might be useful for those who want more control over system configuration management. For larger needs, see Chapter 36, “Managing Sets of Servers.”

dotdee

If you run Linux-based systems, you will find a series of directories that end with a `.d` and that store configuration files. These are sometimes called `.d` or “dot dee” directories. If you look in `/etc/`, you find many (such as `apparmor.d` and `pam.d`). Opening these directories reveals a large number of configuration files and perhaps other directories containing even more. In Ubuntu or other Debian-based systems, it is a violation of etiquette (and Debian policy) for any software package to be allowed to directly change the configuration files of another package. This can be problematic if you want to use system configuration management software.

dotdee solves this problem by allowing you to take any flat file in your filesystem and replace it with a symlink pointing to a file that is generated from a `.d`-style directory. It saves the original file and then updates the generated file automatically and dynamically any time any file in the original `.d` directory is added, deleted, or modified. This way, the Debian policy and general etiquette standards are met, but configurations can be modified as needed by an external program.

dotdee works its magic using `inotify` to dynamically and instantly update the master file. The master file can be built three different ways: using flat files, which are concatenated; using diff/patch files, which are applied in a quiltlike manner; and using executables, which process `stdin` and dump to `stdout`. This flexibility should make any system administrator or developer guru happy.

Snappy Ubuntu Core

Snappy Ubuntu Core takes the absolute minimum of files and code necessary for a usable Ubuntu server image and adds to it a new means of managing software packages. The idea is similar to how smart phones like Android-based phones provide software. In this method, the software packages include everything they need to run on the operating system, effectively making it so that a package is isolated from the operating system more completely. This is designed to protect from the possibility of a package breaking other packages or an entire operating system installation. It is also intended to make updates easier and cleaner. With the idea of convergence, where Ubuntu is aiming to use the same set of software for traditional desktops, laptops, tablets, and phones, all these devices will share the core operating system and Unity interface, and packages that work on any one should also work on the others. This feature first appeared in Ubuntu 16.04 LTS.

Using Snaps

Software bundles packaged this way are called *snaps*. Snaps can be installed using Ubuntu Software or from the command line. On the command line, snaps have a new command. Use the following to interact with snaps.

To show a list of snap packages that are available to be installed:

```
matthew@seymour:~$ snap find
```

Because snaps are new, few packages are available today. However, this list is growing and is likely to become unwieldy at some point. Sure, you can use grep to search through the output to try to find a specific package in the list, but you can also use:

```
matthew@seymour:~$ snap find searchterm
```

To install a snap package:

```
matthew@seymour:~$ sudo snap install packagename
```

To show a list of snap packages that are currently installed:

```
matthew@seymour:~$ snap list
```

To update a snap package:

```
matthew@seymour:~$ sudo snap refresh packagename
```

To remove a snap package:

```
matthew@seymour:~$ sudo snap remove packagename
```

To display a list of changes, such as when snaps were installed, updated, or removed:

```
matthew@seymour:~$ snap changes
```

It is possible for you to create a snap package. See Chapter 39, “Opportunistic Development,” to learn how. Learn more and keep up to date as Ubuntu Snappy Core continues develop by checking out <https://developer.ubuntu.com/en/snappy/>.

References

- ▶ www.debian.org/doc/manuals/project-history/ch-detailed.en.html—History of the Debian Linux package system.
- ▶ www.nongnu.org/synaptic/—Home of the Synaptic package manager.
- ▶ www.ubuntu.com/usn—The official list of Ubuntu security notices.

This page intentionally left blank

Index

Symbols

- & (ampersand), 215, 221–222, 277, 637
- * (asterisk), 177, 274, 539
- \ (backslash), 289
- ` (backtick), 289–290
- ! (bang), 237
- | (bar) operator, 219–222, 276
- [] (brackets), 231
- ^ (caret), 567
- \$ (dollar sign)
 - end of URL, 567
 - user prompt, 155, 243
 - variable names, 282
- " (double quotation marks), 288
- / (forward slash), 10
- < (left angle bracket), 222, 275
- () (parentheses), 222
- # (pound sign), 155, 243
- ? (question mark), 275
- > (right angle bracket), 275
- ; (semicolon), 222, 306, 403, 583
- #! (shebang), 280
- ' (single quotation marks), 288
- \$? special variable, 286
- \$# special variable, 286
- \$* special variable, 286
- \$0 special variable, 286
- ~ (tilde), 152
- 10BASE-T, 383

- 32-bit Ubuntu, 4–5
- 64-bit Ubuntu, 4–5
- 100BASE-T, 383
- 802.11b, 407–408
- 1000BASE-T, 383
- 1000BASE-X, 384
- A
 - AAAA record (DNS), 662
 - AbiWord, 79
 - ac command, 264
 - accept command, 487
 - access control
 - ACLs (access control lists), 171–173, 563–567
 - Apache
 - authentication, 498–499
 - Require directive, 498–499
 - accounts
 - Launchpad
 - creating, 696
 - uploading GPG key to, 696–697
 - uploading SSH key to, 697
 - user accounts
 - command line, 154–155
 - file permissions, 244
 - GIDs (group IDs), 244
 - overview, 241–242
 - super users/root users, 242–244
 - UIDs (user IDs), 244
 - user stereotypes, 245
 - ACID compliance (databases), 587–588
 - ACLs (access control lists), 171–173, 563–567
 - activating DHCP (Dynamic Host Control Protocol), 400–401
 - Ada, 710
 - Adblock Plus, 64
 - add-apt-repository command, 730
 - Address Resolution Protocol (ARP), 382
 - address-based virtual hosts, 507–508
 - addressing
 - broadcast addressing, 382
 - IPv4 addressing, 374–376, 567–568
 - IPv6 addressing, 378–380
 - MAC (Media Access Control) addresses, 382
 - multicasting, 382
 - unicast addressing, 382
 - adduser command, 181–182
 - administering LDAP (Lightweight Directory Access Protocol), 621–622
 - Adobe Photoshop, 94
 - “Advanced Bash-Scripting Guide” (Cooper), 39
 - Advanced Linux Sound Architecture (ALSA), 86
 - Advanced Package Tool. See APT (Advanced Package Tool)
 - afio, 359
 - aliases, 278, 550–551
 - all-in-one print/fax/scan devices, 487
 - AllowOverrides directives, 497–498
 - Alpine, 68
 - ALSA (Advanced Linux Sound Architecture), 86
 - Amanda, 358–359
 - ampersand (&), 215, 277, 637
 - Android
 - Android Runtime, 724
 - Android Studio installation, 725
 - development for
 - Android architecture, 724–725
 - Android Studio installation, 725
 - applications, creating, 727
 - online resources, 728
 - SDK (software development kit) installation, 725–727

android-studio/bin/ directory, 725
 anonymous FTP (File Transfer Protocol)
 access, 536–537
 servers, 531–532
 ANSI Common Lisp, 716–717
 Ansible, 657
 Apache Hadoop, 611
 Apache Software Foundation, 77–78, 490
 Apache Tomcat, 530
 Apache web server
 access control
 authentication, 499–501
 Require directive, 498–499
 HTTPS (HTTP Secure), 510–512
 installation, 490–491
 logging, 509–510
 modules. See modules
 online resources, 513
 optimization, 446–447
 overview, 489–490
 runtime server configuration settings
 apache2.conf, 493–495
 .htaccess configuration files, 496–498
 MPMs (multiprocessing modules),
 495–496
 overview, 492
 runtime configuration directives,
 492–493
 starting, 491–492
 stopping, 491–492
 virtual hosting
 address-based virtual hosts, 507–508
 name-based virtual hosts, 508–509
 overview, 507
 apache2 package, 490
 apache2.conf file, 493–495
 APIs (application programming interfaces), 606
 app development, 698
 AppArmor, 435–437
 apparmor-profiles package, 436
 Apple hardware, installing Ubuntu on, 3
 Application Framework (Android), 725
 application programming interfaces (APIs),
 606
 apply-patch, 690
 apport.log, 330
 apropos command, 156–157, 203
 APT (Advanced Package Tool)
 advantages of, 138
 apt simplified interface, 142–143
 apt-cache command, 141–142
 apt-get dist-upgrade command, 139
 apt-get install command, 139–141
 apt-get remove command, 141
 apt-get update command, 138
 apt-get upgrade command, 138–139
 apt-get utility, 16
 overview, 637
 archives, PPAs (personal package archives)
 creating, 688
 Ubuntu SDK, 730
 ark command, 353–354
 ARM processors, 724
 ARP (Address Resolution Protocol), 382
 ARPANET, 63
The Art of Unix Programming (Raymond), 185
 assigning file permissions, 166–167
 assigning variables, 282
 asterisk (*), 177, 274, 539
 atime, disabling, 445
 atomicity, 587
 attacks, 425–427
 authentication
 Apache, 498–499
 authenticated FTP (File Transfer Protocol),
 532

auth.log, 330
AuthType directive, 500
AuthUserFile directive, 500
AND (auto nice daemon), 340
auto nice daemon (AND), 340
autoconf command, 673–674
autocracking scripts, 426
autoresponders, 556
.avi filename extension, 106
Avidemux, 108
awk command, 230–232

B

Babbage, Charles, 710
Back in Time, 356–357
background
 background processing, 277
 moving jobs to, 215–216
 running jobs in, 215
backports, 20
backslash (\), 289
backtick (`), 289–290
backups. *See also* version control systems
 backup strategy
 evaluation, 345–346
 full incremental backups, 347
 full periodic backups, 347
 mirroring systems, 348
 needs assessment, 343–344
 RAID arrays, 348
 reasons for data loss, 341–343
 simple strategy, 346
file copying
 mc command, 362
 overview, 360

rsync command, 362–364
tar command, 360–361
hardware
 CD-RW, 349
 cloud storage, 350
 DVD+RW/-RW, 349
 network storage, 349
 tape drives, 349–350
levels, 346
online resources, 368
software
 afio, 359
 Amanda, 358–359
 Back in Time, 356–357
 Déjà Dup, 354–356
 flexbackup, 359
 GNOME File Roller, 353
 KDE ark archiving tool, 353–354
 tar, 351–353
 system rescue, 366–368
 before Ubuntu installation, 7
badblocks command, 445
Bandwidth Meter and Diagnostics, 64
bang (!), 237
Banshee, 91–93
bar operator (), 219–221, 276
Base, 77
BaseX, 610–611
bash, comparing expressions in
 file operators, 293–294
 logical operators, 294–295
 number comparison, 292–293
 string comparison, 290–292
basic input/output system. *See* BIOS
 (basic input/output system)
batch command, 265–268
Battle for Wesnoth, 126–127

Bazaar
 configuration, 697
 version control, 683–684

bch command, 690

BDB (Berkeley DB), 606–607, 615

Beekmans, Gerard, 39

beep codes, 313

Berkeley DB (BDB), 606–607, 615

Berkeley Internet Name Domain (BIND), 665–667

Berners-Lee, Tim, 63

bg command, 215–216

BigTable, 611

Bikeshed, 689–691

bikeshed package, 689

/bin directory, 158–159

BIND (Berkeley Internet Name Domain), 665–667

bind9 package, 666

BIOS (basic input/output system)
 beep codes, 313
 boot process, 311–313
 checking, 235
 hard disk optimization, 442–443

Bitbucket, 686

Blender, 97, 108

Bluetooth, 407–408

.bmp filename extension, 95

Boolean operators, 221–222

boot loader
 GRUB2 (Grand Unified Boot Loader)
 checking, 235
 reinstalling, 235–236
 installation, 5
 overview, 312–314
 restoring, 367

boot process
 beginning, 312–314

BIOS (basic input/output system), 311–313

boot loader
 GRUB2 (Grand Unified Boot Loader), 5, 235–236
 installation, 5
 overview, 312–314
 restoring, 367

dual boot, 6

init systems
 init scripts, 316–317
 overview, 312, 314
 systemd, 320–321
 Upstart, 319–320

Linux kernel, loading, 314–315

MBR (master boot record)
 definition of, 313
 installing boot loader to, 5

online resources, 322

overview, 311

runlevels
 changing, 318–319
 default runlevel, booting into, 316
 definitions, 315
 overview, 311–312
 purpose of, 315
 troubleshooting, 319

system services
 controlling at boot, 317–318
 running at boot, 311–312, 315
 starting/stopping manually, 319

troubleshooting
 Boot Repair, 320–321
 overview, 235–236
 starting/stopping services manually, 319
 with systemd, 320–321
 with Upstart, 319–320

UEFI (Unified Extensible Firmware Interface), 313–314

- Boot Repair, 320–321
- /boot/initrd.img file, 470
- boot.log, 330
- bootmail, 691
- brackets ([]), 231
- Brasero, 99–100
- break statement, 307
- Breezy Badger, 33
- bridged networking, 634–635
- bridges, 386
- bridge-utils package, 634
- broadcast addressing, 382
- Brooktree Bt*** chips, 104
- browsers. See web browsers
- brute-forcing, 419
- Bsdftpd-ssl, 533
- buffers, kernel ring, 239
- Bug Squad, 705
- bugs
 - Bug Squad, 705
 - debugging tools
 - gdb command, 675
 - gprof command, 675
 - splint command, 674–675
 - finding, 701
 - fixing, 698–701
- built-in variables, 286–287
- bunzip2 command, 232
- Burian, Michael, 39
- burning
 - CDs, 99–101
 - DVDs, 102–104
- business environments, Ubuntu in, 33–34
- Bynari, 558
- Byobu, 689
- byobu command, 233–234
- bzip2 command, 232
- bzr add command, 683
- bzr cdiff command, 684
- bzr checkout command, 683
- bzr commit command, 684
- bzr init command, 683
- bzr push command, 684
- bzr-builddeb command, 695
- bzrp command, 690

C

- C language
 - debugging
 - gdb command, 675
 - gprof command, 675
 - splint command, 674–675
 - GNU C compiler, 675–676
 - makefiles, 671–673
 - overview, 669–671
- C++ language, 669–671
- cable, 384–385
- cache (SQL)
 - query cache, 449–451
 - table cache, 451
- CAG (Carrier Advisory Group), 730
- Calc, 76
- cameras, digital, 98
- cancel command, 487
- Canonical Ltd., 33, 657
- capturing screen images, 97
- caret (^), 567
- Carrier Advisory Group (CAG), 730
- Cassandra, 607
- cat command, 160, 177, 185–189, 330
- categories of UTP (unshielded twisted-pair), 384–385

CCSM (CompizConfig Settings Manager), 59, 60
 cd command, 164–165, 189–191
 cdrecord command, 101–102
 CDs
 burning
 overview, 99–101
 shell scripts, 305–307
 CD-RW, 349
 Certificate Signing Requests (CSRs), 511, 524
 CFEngine, 656–657
 CGI (Common Gateway Interface), 718
 Chalup, Strata R., 40
 change command, 264
 changing
 directories, 164–165, 189–191
 file permissions
 chgrp command, 170
 chmod command, 168–169
 overview, 191
 umask command, 169–170
 runlevels, 318–319
 channels (IRC), 42
 charms (Juju), 650–652
 chattr command, 445
 checking
 BIOS (basic input/output system), 235
 dependency checking, 671
 GRUB2 (Grand Unified Boot Loader), 235
 log files, 329–330
 loopback interface availability, 370
 network connections, 371–373
 for updates
 APT (Advanced Package Tool), 138
 Software Updater, 15–18, 137
 Chef, 656
 Cherokee, 528–529
 cherokee-admin-launcher command, 528–529
 chfn command, 264
 chgrp command, 170, 244
 children’s games, 129
 Chinese-language Ubuntu Kylin, 118
 chmod command, 168–169, 191, 244, 264
 choosing passwords, 12
 chown command, 170, 244, 264
 chpasswd command, 264
 Chrome, 65–66
 Chromium, 65–66
 chroot jail, 238–239
 chsh command, 250, 264
 CIDR (classless interdomain routing) notation, 550, 567
 Cinelerra, 108
 CinePaint, 97, 108
 classless interdomain routing (CIDR) notation, 550, 567
 Claws, 68
 CLF (Combined Log Format), 509
 CLI (command-line interface). *See* command line
 clients
 dhclient, 401–402
 email clients
 Alpine, 68
 choosing, 66
 Claws, 68
 Evolution, 67–68
 Kmail, 68
 Mozilla Thunderbird, 66–67
 Mutt, 68
 web-based email applications, 68
 evince, 35–36
 Glade, 677–678
 gv, 36
 IRC (Internet Relay Chat) clients, 70–71
 KDevelop, 676–677

- LDAP (Lightweight Directory Access Protocol)
 - Evolution, 620
 - Thunderbird, 621
- NFS (Network File System), 475–476
- Pan, 73–74
- Squid, 562–563
- thin clients, 623
- VPNs (virtual private networks), 571–573
- clisp package, 716–717
- Clojure, 710–711
- closure package, 711
- Clonezilla, 7
- cloud computing
 - advantages of, 642
 - backups, 350
 - deployment, 644
 - IaaS (infrastructure as a service), 643
 - Juju
 - charms, 650–652
 - GUI (graphical user interface), 652
 - installation, 648–650
 - on Mac OS X and Windows, 653
 - Mojo, 653
 - overview, 647
 - Quickstart, 653
 - Landscape, 654
 - MaaS (metal as a service), 643, 653–654
 - online resources, 654
 - OpenStack
 - Glance, 647
 - Horizon, 647
 - Keystone, 646
 - Neutron, 646
 - Nova, 645–646
 - online resources, 647
 - overview, 645
 - Swift, 646
- overview, 641–642
- PaaS (platform as a service), 643
- SaaS (software as a service), 643
- Snappy Ubuntu Core, 653
- cloud-sandbox command, 690
- Cloudsmith, 656
- CNAME record (DNS), 663
- COBOL (Common Business Oriented Language), 711–712
- code names, 33
- CodeWeavers CrossOver Office, 83
- col1 command, 692, 690
- Combined Log Format (CLF), 509
- combining commands
 - Boolean operators, 221–222
 - pipes, 219–221
 - process substitution, 222
 - in sequence, 222
- comm command, 212–213
- comma separated values (CSV), 231
- at command, 265–268
- command line. *See also* commands
 - accessing, 151–152
 - advantages of, 185–187
 - burning DVDs from, 102–104
 - CDs, burning, 99–101
 - definition of, 16, 150–151
 - documentation
 - apropos command, 156–157
 - man pages, 156
 - whereis command, 157
 - DVDs, burning
 - packet writing, 103–104
 - session writing, 103
 - logging in
 - from remote computer, 153–154
 - text-based console login, 152

logging out
 from remote computer, 153–154
 text-based console login, 153
multiple terminals, 233–234
online resources, 184, 208, 240
overview, 273–274
super users/root users
 creating users, 181–182
 deleting users, 182
 overview, 178
 sudo command, 178–181
user accounts, 154–155

commands. *See also* command line
ac, 264
accept, 487
add-apt-repository, 730
adduser, 181–182
afio, 359
aliases, 278
apply-patch, 690
apropos, 156–157, 203
apt-cache, 141–142
apt-file, 695
apt-get dist-upgrade, 139
apt-get install, 139–141, 730
apt-get remove, 141
apt-get update, 138, 730
apt-get upgrade, 138–139
ark, 353–354
at, 265–268
autoconf, 673–674
awk, 230–232
badblocks, 445
batch, 265–268
bch, 690
bg, 215–216
bootmail, 691

bunzip2, 232
byobu, 233–234
bzip2, 232
bzr add, 683
bzr cdiff, 684
bzr checkout, 683
bzr commit, 684
bzr init, 683
bzr push, 684
bzr-builddeb, 695
bzrp, 690
cancel, 487
cat, 160, 177, 185–189, 330
cd, 164–165, 189–191
cdrecord, 101–102
change, 264
chattr, 445
cherokee-admin-launcher, 528–529
chfn, 264
chgrp, 170, 244
chmod, 168–169, 191, 244, 264
chown, 170, 244, 264
chpasswd, 264
chsh, 250, 264
cloud-sandbox, 690
col1, 692, 690
combining
 Boolean operators, 221–222
 pipes, 219–221
 process substitution, 222
 in sequence, 222
comm, 212–213
command history, viewing, 237
command substitution, 289–290
convert, 96
cp, 26, 176, 191–192
createuser, 593

cron, 268–270
date, 24
deluser, 182
depmod, 459
df, 328–329
diff, 212
disable, 487
dman, 690
dmesg, 26, 239
dropuser, 594
du, 192
e2fsck, 445
echo, 193
edquota, 263
emacs, 229–230
enable, 487
environment variables, 222–226
etkeeper, 365
exit, 153
faillog, 330
fg, 215–216
find, 193–195
flexbackup, 359
free, 160, 327–328
gdb, 675
git add, 683
git commit, 683
git init, 682
git pull, 683
git push, 683
git remote add, 682
git rm, 683
gnome-screenshot, 97
gnome-terminal, 151
GNU Coreutils, 239
gnupg, 695
gpasswd, 247
gprof, 675
grep, 196, 330
groovyConsole, 715
groovysh, 715
groupadd, 247
groupdel, 247
groupmod, 247
groups, 264
growisofs, 103–104
gunzip, 232
gzip, 232
hdparm, 443
hg add, 686
hg commit, 686
hg init, 686
hg push, 686
hg update, 686
htop, 327
htpasswd, 499
hwclock, 24
if config, 370
ifconfig, 389–391
info, 156
initrd, 458
insmod, 458
ip, 392
iptables, 435
iwconfig, 405–406
iwlist, 405
iwpriv, 405
iwspy, 405
jobs, 214
keep-one-running, 691
keyboard shortcuts, 238
kill, 325–326
lastlog, 330
ldapadd, 622
ldapdelete, 622
ldapmodify, 621

ldapsearch, 621
less, 36, 156, 177, 197–198
ln, 199–200
locate, 200
logname, 264
logout, 153
logrotate, 331–333
lp, 487
lpc, 487
lpq, 487
lprm, 487
lpstat, 487
ls, 162–164, 200–202
lsblk, 202–203
lshw, 202–203
lsmod, 202–203, 458
lspci, 202–203
lua, 717
make, 671–673
make config, 466
make menuconfig, 466–467
make oldconfig, 467
make xconfig, 467
man, 35, 156, 203
mc, 362
mkdir, 173–174, 204
mkinitrd, 470
mkisofs, 101
modinfo, 459
modprobe, 388–389, 458–459
mtr, 373
mv, 175–176, 204
mysql, 598–600
mysql_install_db, 589
nano, 227
netstat, 394
newusers, 264
nice, 218–219, 326–327
nm-connection-editor, 389
overview, 187–188
passwd, 249
patch, 464
pbget, 690
pbput, 690
pbputs, 690
pbuilder, 695
ping, 371–373, 667
pppconfig, 412
printing location of, 207
ps, 213–214
psql, 600
purge-old-kernels, 691
pwd, 152–153, 165
quickstart, 653
quotacheck, 263
quotaoff, 263
quotaon, 263
regular expressions, 177
release, 690
release-build, 690
rename, 204
renice, 326–327
repquota, 263
rm, 175, 205
rmdir, 173–174
rmmod, 458
route, 392–394
rsync, 362–364
rtcwake, 270–272
running
 multiple commands, 237–238
 previous command, 236–237
 in sequence, 222
run-one, 691
run-this-one, 691
scp, 418

sed, 230–232
sftp, 417, 418–419
shutdown, 13, 182–183
smbclient, 482
smbstatus, 481–482
snap, 146–147
socks-prox, 690
software updater, 15
sort, 205–207
sox, 89
splint, 674–675
ssh-import-id, 691
ssh-keygen, 419–420
stderr, 211–212
stdin, 211–212
stdout, 211–212
su, 257–259
sudo, 18, 178–181, 259–262
svn add, 685
svn checkout, 685
svn commit, 685
svn delete, 685
svn import, 684
svn update, 685
svnadmin create, 684
sync, 444
sysctl, 445–446
systemctl, 321
systemd, 318
system-search, 690
tail, 207, 329
tar, 232, 351–353, 360–361
telinit, 318
telnet, 416
terminal, 16
testdrive, 706
top, 216–218, 327
touch, 165–166, 173
traceroute, 373
tripwire, 430
tune2fs, 444
ufw, 432–433
umask, 165, 169–170
uname, 30
uptime, 328
uquick, 691
useradd, 247–248, 250–251
usermod, 249
vi, 227–228
virt-clone, 636–637
virt-install, 636
vmbuilder, 635
vmstat, 328
wget, 207–208
whatis, 203
what-provides, 691
whereis, 157, 225
which, 207
wifi-status, 692, 691
wildcards, 177
wish, 281
zless, 36
commercial games, 129
commercial support, 38–39
Common Business Oriented Language (COBOL), 711–712
Common Gateway Interface (CGI), 718
Common UNIX Printing System (CUPS), 483–486
CommuniGate Pro, 557
community teams, 703–705
comparison of expressions
 file operators
 in pdksh and bash, 293–298
 in tcsh, 297–298

logical operators
 in pdksh and bash, 294–295
 in tcsh, 298–299

number comparison
 in pdksh and bash, 292–293
 in tcsh, 296–297

string comparison
 in pdksh and bash, 290–292
 in tcsh, 290–292

comparison of files
 comm command, 212–213
 diff command, 212

compiled languages, 670

compiling
 kernel
 errors, 471
 initial RAM disk image, creating, 470
 step-by-step process, 464–467

software
 JIT (just-in-time) compilation, 724
 from tarballs, 143–144
 from Ubuntu repository source, 144–145

Compiz, 112

CompizConfig Settings Manager (CCSM), 59, 60

compressed files, 232

compute infrastructure (OpenStack), 645–646

computer attacks, 425–427

CONCAT() function, 585

configuration
 Apache
 apache2.conf, 493–495
 .htaccess configuration files, 496–498
 modules, 502
 MPMs (multiprocessing modules), 495–496
 overview, 492
 runtime configuration directives, 492–493

Bash, 697–698

Bazaar, 697

configuration-management tools, 656

desktop environments, 52

development system
 environment setup, 696–698
 Launchpad account, creating, 696
 packages, 695–696

DHCP (Dynamic Host Control Protocol)
 activation, 400–401
 dhclient, 401–402
 DHCP server, 402
 network hosts, 403–404

DNS (Domain Name System) servers, 665–667

Fetchmail
 global options, 552
 mail server options, 552–553
 overview, 551–552
 user accounts, 553–554

FTP (File Transfer Protocol)
 ftphosts file, 539
 quick-and-dirty installation, 535
 software installation, 533–534
 Very Secure FTP server, 536–538

Internet connections
 dial-up Internet access, 412–413
 PPPoE (Point-to-Point Protocol over Ethernet), 411–412

kernel, 467–470

LDAP (Lightweight Directory Access Protocol)
 clients, 620–621
 schema, 616–617
 servers, 616

loopback interface, 370–371

MySQL
 database creation, 590–591
 passwords, 589–590

networks
 /etc/hosts file, 395
 /etc/resolv.conf file, 396–397
 /etc/services file, 395–396
 graphical configuration tools, 397–399
 ifconfig command, 389–391
 ip command, 392
 netstat command, 394
 network configuration tools, 389
 route command, 392–394

NFS (Network File System)
 clients, 475–476
 servers, 474–475

Nginx
 HTTPS (HTTP Secure), 524–526
 module configuration, 523–524
 PHP configuration, 522–523
 server configuration, 518–520

online resources, 27

permissions, 171–173

Postfix, 546–548

post-installation configuration
 power management, 22
 printers, 22
 software repositories, 19–21
 system settings, 21
 time/date, 23–24
 troubleshooting, 26–27
 wireless networks, 24–25

Samba, 476–481

software management
 dotdee (.d), 145
 Snappy Ubuntu Core, 146–147

Squid
 client configuration, 562–563
 client IP addresses, 567–568
 sample configurations, 568–569

Telnet, 415–416

UFW (Uncomplicated Firewall), 432–435

Unity
 CCSM (CompizConfig Settings Manager), 59
 Smart Scopes, 59
 system settings, 58–60
 Unity Tweak Tool, 59–60
 version control systems, 364–366

VPNs (virtual private networks)
 clients, 571–573
 servers, 573–575
 window managers, 52

configure.in file, 674

confining scripts to directories, 238–239

Conky, 334–339

connections, checking, 371–373

consistency, 587

console-based monitoring
 df command, 328–329
 disk quotas, 329
 free command, 327–328
 kill command, 325–326

log files
 checking, 329–330
 rotating, 331–333
 overview, 323–325
 priority scheduling, 326–327
 vmstat command, 328

contents of files
 displaying, 177
 sorting, 205–207

control structures. See loops

convergence, 52–53

convert command, 96

Cooper, Mendel, 39

copying files

cp command, 176, 191–192
 between machines, 418–419
 mc command, 362
 overview, 360
 rsync command, 362–364
 tar command, 360–361
 core dumps, 675
 Coreutils, 239
 CouchDB, 609
 cp command, 26, 176, 191–192
 crackers, 426
 CREATE DATABASE statement, 590, 593
 CREATE statement, 582–583
 CREATE USER statement, 593
 createuser command, 593
 cron command, 268–270
 CrossOver Office, 83
 CSRs (Certificate Signing Requests), 511, 524
 CSV (comma separated values), 231
 CUPS (Common UNIX Printing System), 483–486
 cupsd.conf file, 483

D

.d (dotdee), 145
 D language, 712–713
 daemons
 AND (auto nice daemon), 340
 pppd, 549
 slapd, 616, 622
 Dapper Drake, 33
 darktable, 97
 Dart, 712–713
 Dash, 14, 54–57
 dashboard (OpenStack), 647

data
 data directory initialization (PostgreSQL), 592
 data locking, 586–587
 inserting into SQL databases, 583–584
 piping, 276
 retrieving from SQL databases, 583–584
 Data Display Debugger (ddd), 675
 database administrators (DBAs), 577–578, 605
 databases
 DBAs (database administrators), 577–578
 flat file databases, 578
 NoSQL databases. See NoSQL databases
 related Ubuntu commands, 601
 relational databases. See relational databases
 date command, 24
 date/time configuration
 date command, 24
 hwclock command, 24
 overview, 23
 Time & Date tool, 23
 DBAs (database administrators), 577–578, 605
 ddd (Data Display Debugger), 675
 Debian, 18, 30
 debugging tools
 gdb command, 675
 gprof command, 675
 splint command, 674–675
 default runlevel, booting into, 316
 default-jdk package, 716
 Déjà Dup, 354–356
 deleting
 directories
 rm command, 175, 205
 rmdir command, 173–174
 files
 from Git repositories, 683

- rm command, 205
- from Subversion, 685
- PostgreSQL users, 594
- users, 182
- Dell, 38
- deluser command, 182
- Dent, Kyle, 548
- dependencies, 673
- dependency checking, 671
- deployment of cloud computing, 644
- depmod command, 459
- Desktop DVD, 2
- desktop environments
 - configuration, 52
 - GNOME3 and Ubuntu GNOME, 116–117
 - KDE and Kubuntu, 113–114
 - LXDE and Lubuntu, 115–116
 - MATE and Ubuntu MATE, 117–118
 - overview, 112–113, 661
 - Ubuntu Kylin, 118
 - Unity
 - Dash, 54–57
 - default look, 53
 - Launcher, 53–54
 - Panel, 57–58
- virtualization
 - KVM (Kernel-based Virtual Machine), 633–637
 - online resources, 639
 - overview, 631–633
 - VirtualBox, 637–638
 - VMware, 639
 - Xen, 639
 - Xfce and Xubuntu, 114–115
- detecting printers, 22
- developers, 641–642
- development (Ubuntu)
 - for Android
 - Android architecture, 724–725
 - Android Studio installation, 725
 - applications, creating, 727
 - online resources, 728
 - overview, 723
 - SDK (software development kit) installation, 725–727
 - code names, 33
 - environment setup
 - Bash configuration, 697–698
 - Bazaar configuration, 697
 - GPG key, 695–696
 - SSH key, 697
 - helping with
 - app development, 698
 - environment setup, 696–698
 - Launchpad account, creating, 696
 - MOTUs (Masters of the Universe), 701
 - online resources, 702
 - overview, 36–37, 693–694
 - package installation/configuration, 695–696
 - patches, creating, 698–701
 - release cycles, 694–695
 - scope development, 698
 - testing and QA, 703–708
 - history of, 33
 - Launchpad account, creating, 696
 - opportunistic development
 - Bikeshed, 689–691
 - definition of, 681
 - Launchpad, 687–688
 - online resources, 692
 - overview, 686–687
 - repository tools, 691–692
 - snap packaging, 689
 - Ubuntu Make, 688–689

package installation/configuration, 695–696

for Ubuntu Mobile

- applications, creating, 730
- online resources, 731
- overview, 729–730

SDK (software development kit) installation, 730

device drivers, 457

Device section (`xorg.conf`), 49–50

device security, 431

`devices.txt` file, 455

DevOps, 641–642

`df` command, 328–329

`dhclient`, 401–402

DHCP (Dynamic Host Control Protocol)

- activating, 400–401
- `dhclient`, 401–402
- DHCP server, 402
- how it works, 399–400
- network host configuration, 403–404
- overview, 399, 567
- uses for, 405

The DHCP Handbook, 405

Dia, 77

dial-up Internet access, 412–413

`diff` command, 212

differences between files, finding, 212

digikam, 98

digital cameras, 98

digital subscriber line (DSL), 408–412

Dijkstra, Edsger, 712

directives

- `AuthType`, 500
- `AuthUserFile`, 500
- `DirectoryIndex`, 495
- `DocumentRoot`, 495
- `Group`, 494
- `Listen`, 493
- `NameVirtualHost`, 508
- `Require`, 498–499
- `satisfy`, 501
- `ServerAdmin`, 494
- `ServerName`, 494–495, 508
- `ServerRoot`, 493
- `User`, 494

directories. See *also* file system

- `AllowOverrides`, 497–498
- `android-studio/bin/`, 725
- `/bin`, 158–159
- changing, 164–165, 189–191
- confining scripts to, 238–239
- creating, 173–174
- deleting
 - `rm` command, 175, 205
 - `rmdir` command, 174
- `/etc`
 - `/etc/aliases`, 550–551
 - `/etc/bind/named.conf.local`, 666
 - `/etc/bind/named.conf.options`, 666
 - `/etc/cups/cupsd.conf`, 483
 - `/etc/events.d/`, 51–52
 - `/etc/exports`, 474–475
 - `/etc/fstab`, 475
 - `/etc/hosts`, 395
 - `/etc/init.d`, 315
 - `/etc/inittab`, 51–52, 315
 - `/etc/modprobe.conf`, 387–388, 459
 - `/etc/modules.conf`, 105
 - `/etc/nginx`, 518
 - `/etc/nginx/nginx.conf`, 518–520
 - `/etc/passwd`, 253–254
 - `/etc/postfix`, 546
 - `/etc/ppp/peers`, 549
 - `/etc/resolv.conf`, 396–397, 667

/etc/samba/smb.conf, 476–481
/etc/services, 395–396
/etc/squid3/squid.conf, 563
/etc/ssh/sshd_config, 377
/etc/sudoers, 260–261
finding current, 165
/home, 159
LDAP directories
 DIT (Directory Information Tree), 616
 populating, 617–619
listing contents of, 162–164
listing files in, 200–202
making, 204
Options, 497
permissions, 167–168
/proc, 160–161, 324
require, 500
/sbin
 overview, 158–159
 /sbin/ifconfig, 389–391
 /sbin/ip, 392
 /sbin/netstat, 394
 /sbin/route, 392–394
table of, 157–158
/tmp, 162
/usr
 overview, 161
 /usr/bin, 45
 /usr/include, 45
 /usr/lib, 45
 /usr/lib/modules, 45
 /usr/lib/X11, 45
 /usr/local/programs, 478
 /usr/share/doc, 35
 /usr/share/doc/xchat, 70
 /usr/share/man, 35
/var, 162
Directory Information Tree (DIT), 616
DirectoryIndex directive, 495
disable command, 487
disabling
 Apache modules, 502
 file access time, 445
disaster recovery plans, 437–438
disk space
 disk quotas, 262–263, 329
 disk usage, printing, 192
 monitoring, 328–329
display managers
 LDM (LTSP Display Manager), 628
 LightDM, 51
 overview, 45, 51–52
displaying
 command history, 237
 file contents, 177, 185–189
 file permissions, 165–166
 output, 197–198
distribution systems, 407
DIT (Directory Information Tree), 616
dman command, 690
dmesg command, 26, 239
DNS (Domain Name System)
 online resources, 667
 overview, 659–660
records
 A, 662
 AAAA, 662
 CNAME, 663
 MX, 663
 NS, 663–664
 SOA, 664–665
 TXT, 665
servers
 configuring with BIND, 665–667

request handling, 661–662

DocBook, 81

The Document Foundation, 77–78

document stores

- BaseX, 610–611
- CouchDB, 609
- MongoDB, 610
- overview, 608–609

documentation

- apropos command, 156–157
- IRC (Internet Relay Chat), 42
- Linux Documentation Project, 413
- Linux guides, 39–40
- mailing lists, 40–42
- man pages, 156, 203
- reading, 35–36
- Ubuntu-related websites, 40–42
- whereis command, 157

DocumentRoot directive, 495

dollar sign (\$)

- end of URL, 567
- user prompt, 155, 243
- variable names, 282

Domain Name System. See DNS
(Domain Name System)

domain names, 661. See also DNS
(Domain Name System)

dotdee (.d), 145

double quotation marks ("), 288

downloading files, 207–208

Draw, 77

drivers

- device drivers, 457
- proprietary video drivers, installing, 122–123

drives

- partition strategies
- GParted, 11
- /home partition, 10

planning, 5

root partitions, 10

swap partitions, 10

USB drives, installing Ubuntu from, 3–4

Drizzle, 577

DROP USER statement, 594

dropuser command, 594

DSL (digital subscriber line), 408–412

du command, 192

dual boot, 6

dummy interface, 371

durability, 588

DVDs

- burning, 102–104
- DVD+RW/-RW, 349
- installing Ubuntu from, 2, 6
- Ubuntu rescue disk, 367

Dynamic Host Control Protocol. See DHCP
(Dynamic Host Control Protocol)

Dynamo, 608

Dyn.com, 400

E

e2fsck command, 445

echo command, 193

Eclipse, 679

Edgy Eft, 33

editing

- PDF (Portable Document Format) files, 81
- video, 107–109

editors. See productivity; text editors

edquota command, 263

#edubuntu channel, 42

Elixir, 713

elixir package, 713

emacs command, 229–230
 email
 clients
 Alpine, 68
 choosing, 66
 Claws, 68
 Evolution, 67–68, 79
 Kmail, 68
 Mozilla Thunderbird, 66–67
 Mutt, 68
 web-based email applications, 68
 Fetchmail
 configuration, 551–552
 installation, 551
 online resources, 558–559
 Postfix
 aliases, 550–551
 configuration, 546–548
 mail relaying, 550
 masquerading, 548
 message delivery intervals, 549
 smart hosts, 549
 sending/receiving
 MDAs (mail delivery agents), 544–545, 555–556
 MTAs (mail transfer agents), 541–544
 MUAs (mail user agents), 545
 overview, 541–542
 servers, 556–558
 EMCAScript, 716
 emulation, 632
 emulators (gaming), 121
 enable command, 487
 enabling. See configuration
 encrypting/home partition, 10
 endless loops, 301
 Enlightenment, 112
 enterprise server monitoring, 340
 environment variables, 222–226, 281
 epic client, 70
 Erlang, 609, 713
 erlang package, 713
 errors, kernel, 471
 escape character, 287
 /etc directory. See directories, /etc
 etckeeper command, 365
 Ethernet
 gigabit Ethernet, 383–384
 PPPoE (Point-to-Point Protocol over Ethernet), 410–412
 etiquette for IRC (Internet Relay Chat), 71
 evaluation
 backup strategy, 345–346
 lazy evaluation, 715
 Evince, 80–81
 evince client, 35–36
 Evolution, 67–68, 79, 620
 Exchange Server, 557
 exclamation point (!), 237
 executing. See running
 Exim, 543–544
 exit statement, 153, 307
 exports file, 474–475
 expressions
 comparison of expressions
 in pdksh and bash, 290–295
 in tcsh, 295–299
 regular expressions, 177
 Extensible Markup Language (XML), 81
 external attacks, 425–426

F

faillog command, 330
 FDDI (fiber distributed data interface), 383–384

Feisty Fawn, 33

Fetchmail

- configuration
 - global options, 552
 - mail server options, 552–553
 - overview, 551–552
 - user accounts, 553–554
- installation, 551

.fetchmailrc file

- global options, 552
- mail server options, 552–553
- overview, 551–552
- user accounts, 553–554

fg command, 215–216

fiber distributed data interface (FDDI), 383–384

fiber optic, 383–384, 385

file access time, disabling, 445

file operators

- in pdksh and bash, 293–294
- in tcsh, 297–298

File Roller, 353

File section (`xorg.conf`), 47–48

file system. *See also* directories

- chroot jail, 238–239
- directories
 - changing, 189–191
 - confining scripts to, 238–239
 - creating, 173–174
 - deleting, 173–175
 - table of, 157–158
- navigating
 - `cd` command, 164–165
 - `ls` command, 162–164
 - overview, 162
 - `pwd` command, 165
- permissions. *See* permissions
- tuning, 444–445

File Transfer Protocol. *See* FTP (File Transfer Protocol)

filename extensions

- .avi, 106
- .bmp, 95
- .flac, 88
- .flv, 106
- .gif, 95
- .gz, 36
- .jpg, 96
- .ldif, 619
- .mov, 106
- .mp3, 88
- .mpeg, 106
- .ogg, 88, 106
- .ogv, 106
- .pcx, 96
- .pdf, 36
- .png, 96
- .ps, 36
- .qt, 106
- .raw, 88
- .svg, 96
- .tif, 96
- .wav, 88
- .webm, 106

files. *See also* filename extensions

- aliases, 550–551
- `apache2.conf`, 493–495
- comparing
 - `comm` command, 212–213
 - `diff` command, 212
- compressed files, 232
- `configure.in`, 674
- copying
 - `cp` command, 176, 191–192
 - between machines, 418–419
 - `mc` command, 362

overview, 360
rsync command, 362–364
tar command, 360–361
creating, 173
cupsd.conf, 483
deleting, 205
devices.txt, 455
displaying contents of, 177
downloading, 207–208
exports, 474–475
.fetchmailrc
 global options, 552
 mail server options, 552–553
 overview, 551–552
 user accounts, 553–554
finding, 14, 193–195, 200
fstab, 159, 475
FTP (File Transfer Protocol), 540
ftphosts, 539
graphics formats, 95–97
hosts, 395
.htaccess, 496–498
ide.txt, 455
include files, 670
init.d., 315
initrd.img, 470
initrd.txt, 456
inittab, 315
kernel-parameters.txt, 456
LaTeX files, 82
linking, 199–200
listing, 200–202
log files
 Apache, 509–510
 checking, 329–330
 rotating, 331–333
logrotate.conf, 331–332
makefiles, 671–673
man pages, 35, 156, 203
modprobe.conf, 387–388
modprobe.conf file, 459
modprobe.d, 159
modules.conf, 105
moving, 175–176, 204
named.conf.local, 666
named.conf.options, 666
nginx.conf, 518–520
object files, 670
OpenDocument, 78
passwd, 159, 253–254
PDF (Portable Document Format) files
 editing, 81
 reading, 36, 80–81
peers, 549
permissions. See permissions
PostScript files, reading, 36
printing, 185–189
remote file serving via FTP
 FTP users, 534–536
 ftphosts file, 539
 overview, 531
 servers, 531–533
 software installation, 533–534
renaming, 175–176, 204
resolv.conf, 396–397, 667
saving from nonbooting hard drive, 368
services, 395–396
sharing
 NFS (Network File System), 474–476
 online resources, 488
 Samba, 476–483
smb.conf, 476–481
sorting contents of, 205–207
sound formats, 88–89

squid.conf, 563
sshd_config, 377
sudoers, 159
sudoers file, 260–261
sysrq.txt, 456
tar files, 143–144
testdriverc file, 706
text files, reading, 36
version control systems
 Bazaar, 683–684
 for configuration files, 364–366
 Git, 682–683
 Mercurial, 685–686
 overview, 681–682
 Subversion, 684–685
video files
 editing, 107–109
 formats, 105–106
vsftpd.banned_emails, 537
vsftpd.chroot_list, 537
vsftpd.conf, 536–537
vsftpd.log, 537
vsftpd.user_list, 537
xorg.conf
 Device section, 49–50
 Files section, 47–48
 InputDevice section, 48–49
 Module section, 48
 Monitor section, 49
 overview, 46–47
 Screen section, 50–51
 ServerLayout section, 47
find command, 193–195
finding
 bugs, 701
 current directory, 165
 differences between files, 212
 files, 14, 193–195, 200
programs, 14
similarities between files, 212
software, 141–142
strings, 196
Firefox
 RSS feeds, 69
 web browsing, 63–65
firewalls
 iptables, 435
 UFW (Uncomplicated Firewall), 432–435
first-person shooter (FPS) games, 124
FLAC (Free Lossless Audio Format), 88
.flac filename extension, 88
flashplugin-installer package, 65
flat file databases, 578
flavors (Ubuntu), 3
flexbackup, 359
FlightGear, 128
FlockDB, 613
.flv filename extension, 106
ForecastFox, 64
foreground, moving jobs to, 215–216
Forth, 713–714
Fortran, 714–715
forward slash (/), 10
forwarding email, 550–551
FPS (first-person shooter) games, 124
FQDN (fully qualified domain name), 616
Frampton, Steve, 39
free command, 160, 327–328
Free Lossless Audio Format (FLAC), 88
free memory, displaying, 327–328
Free Software Foundation, 29–30
Freshmeat, 359
Frets on Fire, 127–128
Frozen Bubble, 124–125
fstab file, 159, 475
FTP (File Transfer Protocol)

- ftphosts file, 539
- online resources, 540
- overview, 531
- servers. See servers, FTP
- software installation, 533–534
- users, 534–536
- ftphosts file, 539
 - full incremental backups, 347
 - full periodic backups, 347
 - fully qualified domain name (FQDN), 616
 - functions. See also commands; methods
 - `CONCAT()`, 585
 - definition of, 670
 - shell scripts, 307–308
- G**
 - games
 - Battle for Wesnoth, 126–127
 - children’s games, 129
 - commercial games, 129
 - emulators, 121
 - FlightGear, 128
 - Frets on Fire, 127–128
 - Frozen Bubble, 125–126
 - online resources, 130–131
 - overview, 121
 - proprietary video drivers, installing, 122–123
 - Scorched 3D, 124–125
 - Speed Dreams, 129
 - Steam platform, 123
 - SuperTux, 126
 - Warsow, 124
 - Windows games, playing, 130
 - gcc (GNU Compiler Collection), 669, 675–676
 - gdb command, 675
 - Gedit, 81
 - generating
 - GPG keys, 695–696
 - SSH keys, 696
 - gforth package, 714
 - GHC (Glorious Glasgow Haskell Compilation system), 715
 - ghc package, 715
 - GIDs (group IDs), 244
 - .gif filename extension, 95
 - gigabit Ethernet, 383–384
 - GIMP (GNU Image Manipulation Program), 79, 93–94
 - GIMP Took Kit (GTK) widget set, 79
 - Git, 682–683
 - git add command, 683
 - git commit command, 683
 - git init command, 682
 - git pull command, 683
 - git push command, 683
 - git remote add command, 682
 - git rm command, 683
 - Glade client, 677–678
 - Glance, 647
 - [global] section (`smb.conf`), 479
 - Glorious Glasgow Haskell Compilation system (GHC), 715
 - gnat package, 710
 - gnat-gps package, 710
 - GNOME
 - File Roller, 353
 - Glade client, 677–678
 - GNOME Office, 78–79
 - GNOME3, 116–117
 - mailing lists, 41
 - website, 61
 - gnome-nettool, 339
 - gnome-screenshot commnd, 97

gnome-shell package, 116
gnome-terminal command, 151
GNU Ada Compiler, 710
GNU Compiler Collection (gcc), 669, 675–676
GNU Coreutils, 239
GNU General Public License (GPL), 29
GNU Image Manipulation Program (GIMP), 79, 93–94
GNU Project, 149
Gnumeric, 79
gnupg command, 695
Go, 714
gofortran package, 714–715
golang package, 714
golang-docs package, 714
Goobuntu, 38
Google
 BigTable, 611
 Chrome, 65–66
 Chromium, 65–66
 Dart, 712–713
 MapReduce, 611
 search tips, 37–38
GParted, 11
gpasswd command, 247
PGP keys
 generating, 695–696
 uploading to Launchpad, 696–697
GPL (GNU General Public License), 29
gprof command, 675
GPT (GUID Partition Table), 314
Grand Unified Boot Loader. See GRUB2
 (Grand Unified Boot Loader)
GRANT statement, 591, 594
granting
 PostgreSQL privileges, 594–595
 system administrator privileges
overview, 257
with su command, 257–259
with sudocommand, 259–262
graph stores
 FlockDB, 613
 HyperGraphDB, 612–613
 Neo4j, 612
 OrientDB, 612
 overview, 612
graphical database clients, 600
graphical development tools
 IDEs (integrated development environments), 678–680
 KDevelop client, 676–678
 overview, 676
 SDKs (software development kits), 678–680
graphical network configuration tools, 397–399
graphical system-management tools
 Conky, 334–339
 gnome-nettool, 339
 overview, 333–334
 System Monitor, 334
 vncviewer, 339
 wireshark, 339
graphical user interfaces. See GUIs
 (graphical user interfaces)
graphics formats, 95–97
Green, Andy, 417
grep command, 196, 330
greplog, 284–286
Groovy, 715
groovy package, 715
groovyConsole command, 715
groovysh command, 715
Group directive, 494
group IDs (GIDs), 244
groupadd command, 247
groupdel command, 247

groupmod command, 247
 groups
 group listings, 245–246
 LUGs (Linux User Groups), 39
 management tools, 246–248
 overview, 245
 groups command, 264
 growisofs command, 103–104
 GRUB2 (Grand Unified Boot Loader)
 checking, 235
 installation, 5
 reinstalling, 235–236
 restoring, 367
 Grün, Christian, 610
 GStreamer, 88
 gstreamer0.10-plugins-ugly package, 88
 GTK (GIMP Tool Kit) widget set, 79
 GUID Partition Table (GPT), 314
 guides (Linux), 39–40
 GUIs (graphical user interfaces)
 database access, 597
 desktop environments
 GNOME3 and Ubuntu GNOME, 116–117
 KDE and Kubuntu, 113–114
 LXDE and Lubuntu, 115–116
 MATE and Ubuntu MATE, 117–118
 overview, 108, 112–113
 Ubuntu Kylin, 118
 Xfce and Xubuntu, 114–115
 Juju, 652
 Mir, 44
 online resources, 118–119
 power shortcuts, 60
 Unity
 configuration, 58–60
 desktop, 53–58
 overview, 52–53

 X. See X
 gunzip command, 232
 gv client, 36
 .gz filename extension, 36
 gzip command, 232

H
 hackers, 426
 Hadoop, 611
 Hamano, Junio, 682
 hard disk optimization
 BIOS settings, 442–443
 disk usage, printing, 192
 hdparm command, 443
 overview, 441–442
 hard links, 199–200
 hardware clock, setting, 24
 hardware emulation, 632
 hardware specifications, researching, 2
 harvest, 701
 Haskell, 715
 haskell-platform package, 715
 HBase, 611
 HDLC (high-level data link control), 412
 hdparm command, 443
 header files, 670
 headerless format, 88
 Hein, Trent R., 40
 help. See online resources
 helping with Ubuntu development
 app development, 698
 environment setup, 696–698
 Launchpad account, creating, 696
 MOTUs (Masters of the Universe), 701
 online resources, 702

- overview, 36–37, 693–694
 - package installation/configuration, 695–696
 - patches, creating, 698–701
 - release cycles, 694–695
 - scope development, 698
 - testing and QA
 - community teams, 703–705
 - online resources, 708
 - Test Drive, 705–708
 - `hg add` commands, 686
 - `hg commit` command, 686
 - `hg init` command, 686
 - `hg push` command, 686
 - `hg update` command, 686
 - hibernating system, 11
 - High Performance Computing (HPC), 670
 - high-level data link control (HDLC), 412
 - Hipp, Richard, 605
 - history
 - command history, viewing, 237
 - of Internet, 63
 - of LibreOffice, 77–78
 - of Ubuntu, 33
 - of X.Org, 43–44
 - `history.log`, 330
 - Hoary Hedgehog, 33
 - Hogan, Christina J., 40
 - /home directory, 159
 - /home partition encryption, 10
 - home use of Ubuntu, 35
 - [homes] section (`smb.conf`), 479–480
 - Horde, 558
 - Horizon, 647
 - Horowitz, Eliot, 610
 - host configuration, 403–404
 - hosts file, 395
 - HOWTO documentation, 35–36
 - HP, 38
 - HPC (High Performance Computing), 670
 - .htaccess configuration files, 496–498
 - HTML (Hypertext Markup Language), 63
 - htop command, 327
 - htpasswd command, 499
 - HTTP (Hypertext Transfer Protocol)
 - HTTPS (HTTP Secure), 510–512, 524–526
 - overview, 63
 - servers. See servers
 - hubs, 385–386
 - Hugin, 98
 - Humble Indie Bundle, 129
 - hwclock command, 24
 - HyperGraphDB, 612–613
 - Hypertext Markup Language (HTML), 63
 - Hypertext Transfer Protocol (HTTP), 63
- |
- IaaS (infrastructure as a service), 643
 - ICANN (Internet Corporation for Assigned Names and Numbers), 661
 - de Icaza, Miguel, 717
 - IDE (integrated development environment), 710
 - identity service (OpenStack), 646
 - IDEs (integrated development environments), 678–680
 - `ide.txt` file, 455
 - IEEE (Institute of Electrical and Electronics Engineers), 407
 - if config command, 370
 - if statement, 304–305
 - ifconfig command, 389–391
 - IIS (Internet Information Services), 489–490
 - ImageMagick convert utility, 96
 - images

- graphics formats, 95–97
- image manipulation
 - Adobe Photoshop, 94
 - GIMP (GNU Image Manipulation Program), 93–94
 - graphics-manipulation tools, 97–98
 - scanners, 94–95
 - screen images, capturing, 97
 - Shotwell Photo Manager, 97–98
- imaging service (OpenStack), 647
- IMAP (Internet Message Access Protocol), 545
- Impress, 76
- include files, 670
- indexes, finding files in, 200
- info command, 156
- information service/information technology (IS/IT), 31
- infrastructure as a service (IaaS), 643
- init systems
 - init scripts, 316–317
 - overview, 312, 314
 - systemd, 320–321
 - Upstart, 319–320
- init.d file, 315
- initial RAM disk image, creating, 470
- initializing network hardware, 387–389
- initrd command, 458
- initrd.img file, 470
- initrd.txt file, 456
- inittab file, 315
- Inkscape, 98
- inline drivers, 457
- input
 - BIOS (basic input/output system)
 - beep codes, 313
 - boot process, 311–313
 - checking, 235
 - hard disk optimization, 442–443
 - finding strings in, 196
 - I/O (input/output redirection), 209–212, 275–276
 - InputDevice section (`xorg.conf`), 48–49
 - input/output (I/O) redirection. See I/O (input/output redirection)
 - INSERT statement, 583–584
 - inserting data into SQL databases, 583–584
 - insmod command, 458
 - installation. See also configuration
 - Android Studio, 725
 - Apache, 490–491
 - boot loader, 5
 - DHCP (Dynamic Host Control Protocol)
 - dhclient, 401–402
 - DHCP server, 402
 - dual boot, 2, 6
 - from DVD or USB drive, 2, 6
 - Fetchmail, 551
 - first updates, 13
 - FTP (File Transfer Protocol) software, 533–534
 - GRUB2 (Grand Unified Boot Loader), 235–236
 - installed packages, listing, 38
 - Juju, 648–650
 - LTSP (Linux Terminal Server Project), 627–628
 - on Mac hardware, 3
 - MySQL, 588–589
 - NFS (Network File System), 474
 - Nginx, 517–518
 - online resources, 27
 - PostgreSQL, 592
 - post-installation configuration
 - power management, 22
 - printers, 22
 - software repositories, 19–21
 - system settings, 21

- time/date, 23–24
- troubleshooting, 26–27
- wireless networks, 24–25
- preparation
 - 32-bit versus 64-bit Ubuntu, 4–5
 - backups, 7
 - hardware specifications, researching, 2
 - installation options, 2–4
 - overview, 1–2
 - partition strategies, planning, 5
 - proprietary video drivers, 122–123
 - SDKs (software development kits)
 - Android SDK, 725–727
 - Ubuntu SDK, 730
 - software
 - with APT (Advanced Package Tool), 139–141
 - with Synaptic, 134–135
 - Squid, 562
 - step-by-step installation, 6–12
 - troubleshooting, 26–27
- Institute of Electrical and Electronics Engineers (IEEE), 407
- integers. See numbers
- integrated development environments (IDEs), 678–680, 710
- internal attacks, 425–426
- Internet
 - connections
 - common configuration information, 408
 - dial-up Internet access, 412–413
 - overview, 408
 - PPPoE (Point-to-Point Protocol over Ethernet), 410–412
 - troubleshooting, 413
 - email clients
 - Alpine, 68
 - choosing, 66
 - Claws, 68
 - Evolution, 67–68, 79
 - Kmail, 68
 - Mozilla Thunderbird, 66–67
 - Mutt, 68
 - web-based email applications, 68
- history of, 63
- IRC (Internet Relay Chat), 70–72
- ISPs (Internet service providers), 374–375, 507
- online resources, 74
- overview, 63
- RSS readers
 - Firefox, 69
 - Liferea, 69–70
 - overview, 69
 - search tips, 37–38
- web browsers
 - Firefox, 63–65
 - Google Chrome, 65–66
 - Google Chromium, 65–66
- Internet Corporation for Assigned Names and Numbers (ICANN), 661
- Internet Information Services (IIS), 489–490
- Internet Message Access Protocol (IMAP), 545
- Internet newgroups, 72–74
- Internet Protocol. See IP (Internet Protocol)
- Internet Relay Chat (IRC), 42, 70–72
- Internet service providers (ISPs), 374–375, 507
- interpreting shell scripts, 280–281
- I/O (input/output redirection), 209–212, 275–276
- IP (Internet Protocol)
 - IP masquerading, 376–377
 - IPv4 addressing, 374–376
 - IPv6 addressing, 378–380
 - overview, 374
 - Squid clients, 567–568
- ip command, 392
- iptables, 435

IRC (Internet Relay Chat), 42, 70–72
 IRCD, 72
 irssi client, 70
 IS/IT (information service/information technology), 31
 ISO Recorder, 3–4
 isolation, 588
 ISPs (Internet service providers), 374–375, 507
 iwconfig command, 405–406
 iwlist command, 405
 iwpriv command, 405
 iwspy command, 405

J

Java, 715–716
 Java Virtual Machine (JVM), 710–711, 715
 JavaScript, 716
 JavaScript Object Notation (JSON), 605, 716
 JDeveloper, 679
 Jetty, 529
 JIT (just-in-time) compilation, 724
 jobs
 listing, 214
 moving to background/foreground, 215–216
 priority scheduling, 218–219
 running in background, 215
 running repeatedly, 268–270
 scheduling for later, 265–268
 jobs command, 214
 .jpg filename extension, 96
 JSON (JavaScript Object Notation), 605, 716
 Juju
 charms, 650–652
 GUI (graphical user interface), 652
 installation, 648–650
 on Mac OS X and Windows, 653

Mojo, 653
 overview, 647
 Quickstart, 653
 juju, 655
 just-in-time (JIT) compilation, 724
 JVM (Java Virtual Machine), 710–711, 715

K

Katz, Damien, 605, 609
 KDE
 ark archiving tool, 353–354
 mailing lists, 41
 monitoring tools, 339
 Kdenlive, 108
 KDevelop client, 676–677
 kdf, 339
 keep-one-running command, 691
 kernel
 for Android, 724
 compiling
 errors, 471
 initial RAM disk image, creating, 470
 step-by-step process, 464–467
 configuration, 467–470
 kernel hackers, 456
 kernel numbering system, 30
 kernel ring buffer, 239
 loading, 314–315
 module management, 458–460
 online resources, 472
 optimization, 445–446
 overview, 453–454
 patching, 463–464
 recompiling, 460–461
 source
 obtaining, 462

- source tree, 453–454
 - troubleshooting, 470–472
 - types of, 457–458
 - versions, 461–462
 - kernel hackers, 456
 - kernel ring buffer, 239
 - Kernel-based Virtual Machine (KVM), 633–637
 - kernel-parameters.txt file, 456
 - kern.log, 330
 - key buffer usage (PKI), 573
 - key-based logins, 419–420
 - keyboard shortcuts, 60, 238
 - KeyPassX, 12
 - keys
 - PGP keys
 - generating, 695–696
 - uploading to Launchpad, 696–697
 - key buffer usage, measuring, 448–449
 - key/value stores
 - Berkeley DB, 606–607
 - Cassandra, 607
 - Memcached, 607–608
 - MemcachedDB, 607–608
 - overview, 606
 - Redis, 608
 - Riak, 608
 - SSH keys
 - generating, 696
 - key-based logins, enabling, 419–420
 - uploading to Launchpad, 697
 - Keystone, 646
 - Kile, 82
 - kill command, 325–326
 - Kirkland, Dustin, 635, 705
 - Kmail, 68
 - Kmov filename extension, 106
 - KOffice, 80
 - Komarinski, Mark F., 39
 - Kontact, 80
 - Kqt filename extension, 106
 - ksysguard, 339
 - Kubuntu, KDE and, 113–114
 - #kubuntu channel, 42
 - kubuntu-desktop package, 113–114
 - KVM (Kernel-based Virtual Machine), 633–637
 - kvm-ok package, 633–634
 - Kwebm filename extension, 106
 - KWord, 80
- ## L
- Landscape, 340, 654, 657
 - languages
 - Ada, 710
 - C
 - debugging, 674–675
 - GNU C compiler, 675–676
 - makefiles, 671–673
 - overview, 669–670
 - C++, 669–671
 - choosing, 8
 - Clojure, 710–711
 - D, 712–713
 - Dart, 712–713
 - Elixir, 713
 - Erlang, 713
 - Forth, 713–714
 - Fortran, 714–715
 - Go, 714
 - Groovy, 715
 - Haskell, 715
 - Java, 715–716
 - JavaScript, 716

- Lisp, 716–717
- Lua, 717–718
- Mono, 717–718
- OCaml, 718
- online resources, 721–722
- overview, 709–710
- Perl, 718
- Python, 719
- Ruby, 719
- Rust, 720
- Scala, 720
- `lastlog` command, 330
- LaTeX, 82
- Launcher (Unity), 53–54
- Launchpad
 - account, creating
 - overview, 696
 - uploading GPG key to, 696–697
 - uploading SSH key to, 697
 - overview, 687–688
- lazy evaluation, 715
- LDAP (Lightweight Directory Access Protocol)
 - administration, 621–622
 - client configuration
 - Evolution, 620
 - Thunderbird, 621
 - directory population, 617–619
 - LDIF (LDAP Data Interchange Format), 619
 - online resources, 622
 - overview, 557, 615–616
 - schema creation, 616–617
 - server configuration, 616
- LDAP Data Interchange Format (LDIF), 619
- `ldapadd` command, 622
- `ldapdelete`, 622
- `ldapdelete` command, 622
- `ldapmodify` command, 621
- `ldapsearch` command, 621
- `ldap-utils` package, 616
- LDIF (LDAP Data Interchange Format), 619
- .ldif filename extension, 619
- LDM (LTSP Display Manager), 628
- “LDP Author Guide” (Komarinski), 39
- LDTP (Linux Desktop Testing Project), 705
- left angle bracket (<), 222
- legacy hardware, 34, 487
- Leiningen, 711
- `less` command, 36, 156, 177, 197–198
- levels, backup, 346
- libraries, Android, 724
- LibreOffice, 76–78
- `libvirt-bin` package, 634
- licensing
 - GNU General Public License (GPL), 29
- Liferea, 69–70
- LightDM, 51
- `lighttpd`, 527–528
- Lightweight Directory Access Protocol. See LDAP (Lightweight Directory Access Protocol)
- limiting resource use
 - & (ampersand), 215
 - `bg` command, 215–216
 - `fg` command, 215–216
 - `jobs` command, 214
 - `ps` command, 213–214
 - `top` command, 216–218
- Limoncelli, Thomas A., 40
- linkers, 670
- linking files, 199–200
- Linux
 - advantages of, 31
 - commercial support, 38–39
 - definition of, 29–31
 - documentation
 - Linux guides, 39–40
 - reading, 35–36
 - kernel. See kernel

LUGs (Linux User Groups), 39

“Linux Administration Made Easy” (Frampton), 39

#linux channel, 42

Linux Desktop Testing Project (LDTP), 705

Linux Documentation Project, 413

Linux Foundation, 29

“Linux from Scratch” (Beekmans), 39

“Linux Kernel Module Programming Guide” (Salzman, Burian, and Pomerantz), 39

Linux Terminal Server Project. See LTSP (Linux Terminal Server Project)

linux-backports-modules packages, 461

#linuxhelp channel, 42

linux-source package, 462

Lisp, 710–711, 716–717

Listen directive, 493

listening

- to music
 - Banshee, 91–93
 - Rhythmbox, 89–91

listing

- files, 200–202
- jobs, 214
- packages, 38
- processes, 213–214
- system information, 202–203

Live Bookmarks (Firefox), 69

In command, 199–200

Io. See loopback interface

loading Linux kernel, 314–315

localhost interface. See loopback interface

locate command, 200

location of commands, printing, 207

locking data, 586–587

log files

- Apache, 509–510
- checking, 329–330
- rotating, 331–333
- vsftpd.log, 537

LogFormat statements, 509

logging in to command line

- from remote computer, 153–154
- text-based console login, 152

logging out of command line

- from remote computer, 153–154
- text-based console login, 153

logical operators

- in pdksh and bash, 294–295
- in tcsh, 298–299

Logical Volume Manager (LVM), 646

logname command, 264

logout command, 153

logrotate command, 331–333

logrotate.conf file, 331–332

Loh, Eugene, 714

long term support (LTS), 21, 33

loopback interface, 370–371

for loops, 299–300

loops. See also statements

- endless loops, 301
- shell scripts
 - break, 307
 - case, 305–307
 - exit, 307
 - if, 304–305
 - repeat, 303
 - select, 303–304
 - shift, 304
 - until, 302–303
 - while, 300–302

Lotus Symphony, 77–78

Lovelace, Ada, 710

Ip command, 487

Ipc command, 487

lpq command, 487

- lprm command, 487
- lpstat command, 487
- ls command, 162–164, 200–202
- lsblk command, 202–203
- lshw commands, 202–203
- lsmod command, 202–203, 458
- lspci command, 105, 202–203
- LTS (long term support), 21, 33
- LTSP (Linux Terminal Server Project)
 - capabilities, 628–629
 - installation, 627–628
 - LDM (LTSP Display Manager), 628
 - online resources, 629
 - overview, 44, 623–624
 - requirements, 624–627
- LTSP Display Manager (LDM), 628
- Lua, 717–718
- lua command, 717
- lua50 package, 717
- Lubuntu, LXDE and, 115–116
- lubuntu-desktop package, 116
- LUGs (Linux User Groups), 39
- LVM (Logical Volume Manager), 646
- LXDE and Lubuntu, 115–116
- M**
- MaaS (metal as a service), 643, 653–654
- MAC (mandatory access control), 435
- MAC (Media Access Control) addresses, 382
- Mac hardware
 - installing Ubuntu on, 3
 - Juju on, 653
- macros with make command, 672–673
- Mago, 705
- mail. See email
- mail delivery agents (MDAs), 544–545
- mail relaying, 550
- mail transfer agents (MTAs), 541–544
- mail user agents (MUAs), 545
- maildir, 544
- mailing lists, 40–42
- make command, 671–673
- make config command, 466
- make menuconfig command, 466–467
- make oldconfig command, 467
- make xconfig command, 467
- makefiles, 671–673
- making directories, 204
- man command, 35, 156, 203
- man pages, 156, 203
- mandatory access control (MAC), 435
- manual pages, 35, 203
- MAPI (Messaging Application Program Interface), 557
- MapReduce, 611
- MariaDB, 577
- masquerading, 376–377, 548
- master boot record. See MBR (master boot record)
- Mastering Regular Expressions* (Freidl), 274
- Masters of the Universe (MOTUs), 18, 701
- matching patterns. See pattern matching
- MATE, 117–118
- math, 77
- Matrox Marvel, 104
- Matrox Rainbow Runner G-Series, 104
- mbox format, 544
- MBR (master boot record)
 - definition of, 313
 - installing boot loader to, 5
- mc command, 362
- MDAs (mail delivery agents), 544–545, 555–556

measuring key buffer usage, 448–449
 Media Access Control (MAC) addresses, 382
 Memcached, 607–608
 MemcachedDB, 607–608
 memory, 327–328. See also cache (SQL)
 Mercurial, 685–686
 Merriman, Dwight, 603
 message delivery intervals (Postfix), 549
 Messaging Application Program Interface (MAPI), 557
 metal as a service (MaaS), 643, 653–654
 methods. See also commands; functions
 Microsoft Exchange Server, 557
 Microsoft Office, 75
 Microsoft Sound Card, 85–86
 middleware, 597
 MIME (Multipurpose Internet Mail Extensions), 73
 Mir, 44
 mirroring systems, 348
 mkdir command, 173–174, 204
 mkinitrd command, 470
 mkisofs command, 101
 mnemonic notation (permissions), 166
 mobile development
 Android
 Android architecture, 724–725
 Android Studio installation, 725
 applications, creating, 727
 online resources, 728
 overview, 723
 SDK (software development kit) installation, 725–727
 Ubuntu Mobile
 applications, creating, 730
 online resources, 731
 overview, 729–730
 SDK (software development kit) installation, 730
 moderated newsgroups, 72
 modinfo command, 459
 modprobe command, 388–389, 458–459
 modprobe.conf file, 387–388, 459
 modprobe.d folder, 159
 modular kernels, 457
 Module section (*xorg.conf*), 48
 modules
 Apache
 mod_access, 502
 mod_alias, 502
 mod_asis, 503
 mod_auth, 503
 mod_auth_anon, 503
 mod_auth_dbm, 503
 mod_auth_digest, 504
 mod_autoindex, 504
 mod_cgi, 504
 mod_dir, 504
 mod_expires, 504
 mod_headers, 504–505
 mod_include, 505
 mod_info, 505
 mod_log_config, 505
 mod_mime, 505
 mod_mime_magic, 505
 mod_negotiation, 505
 mod_proxy, 505
 mod_rewrite, 505
 mod_setenvif, 506
 mod_speling, 506
 mod_ssl, 506
 mod_status, 506
 mod_unique_id, 506
 mod_userdir, 506
 mod_usertrack, 507
 mod_vhost_alias, 507

kernel module management, 458–460
 Nginx, 523–524
 Puppet, 656
`modules.conf` file, 105
 Mojo, 653
 MongoDB, 610
 Monitor section (`xorg.conf`), 49
 monitoring. See system-monitoring tools
 Mono, 717–718
 mono-devel package, 718
 monolithic kernels, 457
 MOTUs (Masters of the Universe), 18, 701
 mounting Samba shares, 482–483
 Mourani, Gerhard, 40
 movies. See video
 moving

- files, 175–176, 204
- jobs to background/foreground, 215–216

 Mozilla Thunderbird, 66–67
`.mp3` filename extension, 88
 MP3 format, 88
`.mpeg` filename extension, 106
 MPMs (multiprocessing modules), 495–496
 MS-DOS, 151
 MTAs (mail transfer agents), 541–544
 mtr command, 373
 MUAs (mail user agents), 545
 multicasting, 382
 multimedia applications

- CDs, burning, 99–101
- DVDs, burning, 102–104

 images

- Adobe Photoshop, 94
- digital cameras, 97–98
- graphics formats, 95–97
- overview, 93–94
- scanners, 94–95

 screen images, capturing, 97
 Shotwell Photo Manager, 98
 online resources, 109
 sound and music

- Banshee, 91–93
- overview, 85–86
- Rhythmbox, 89–91
- sound cards, 86–87
- sound formats, 88–89
- Sound Juicer, 91–92
- volume adjustment, 87

 video

- editing, 107–109
- personal video recorders, 107
- TV and video hardware, 104–105
- video formats, 105–106
- video viewers, 106

 multiple commands, running, 237–238
 multiple terminals, 233–234
 multiprocessing modules (MPMs), 495–496
 Multipurpose Internet Mail Extensions (MIME), 73
 music

- listening to
 - Banshee, 91–93
 - overview, 89
 - Rhythmbox, 89–91
- overview, 85–86
- sound cards, 86–87
- sound formats, 88–89
- volume adjustment, 87

 Mutt, 68
`mv` command, 175–176, 204
 MVRB-Tree, 612
 MX record (DNS), 663
 MySQL

- compared to PostgreSQL, 586–588
- database clients

- graphical clients, 600
- local GUI client access, 597
- MySQL command-line client, 598–600
- overview, 595
- PostgreSQL command-line client, 600
- SSH access, 595–596
- web access, 597
- database creation, 590–591
- installation, 588–589
- optimization. See optimization
- passwords, 589–590
- `mysql` command, 598–600
- `mysql_install_db` command, 589
- MySQLGUI, 600
- Myth TV, 107
- Mythbuntu, 107
- N**
- Nagios, 340
- name serving. See DNS (Domain Name System)
- name-based virtual hosts, 508–509
- names
 - domain names, 661
 - usernames, 251
- NameVirtualHost directive, 508
- nano command, 227
- NAT (network address translation), 376
- Nautilus, 114
- nautilus-actions package, 363
- nautilus-image-converter package, 97
- navigating file system
 - `cd` command, 164–165
 - `ls` command, 162–164
 - overview, 162
 - `pwd` command, 165
- NcFTPd, 532–533
- Nemeth, Evi, 40
- Neo4j, 612
- Nessus, 428
- NetBeans, 679
- NetBoot, 314
- `netpbm` utilities, 96–97
- `netstat` command, 394
- network address translation (NAT), 376
- Network File System. See NFS (Network File System)
- network host configuration, 403–404
- network interface cards (NICs), 382–384
- Network Manager, 24–25
- Network News Transfer Protocol (NNTP), 73
- networking service (OpenStack), 646
- networks
 - bridged networking, 634–635
 - command-line network interface configuration
 - `ip` command, 392
 - `netstat` command, 394
 - `route` command, 392–394
 - configuration, 389–391
 - configuration files
 - `/etc/hosts` file, 395
 - `/etc/resolv.conf` file, 396–397
 - `/etc/services` file, 395–396
 - connection, checking, 371–373
 - DHCP (Dynamic Host Control Protocol). See DHCP (Dynamic Host Control Protocol)
 - graphical configuration tools, 397–399
 - hardware devices
 - bridges, 386
 - cable, 384–385
 - hubs, 385–386
 - initializing, 387–389
 - NICs (network interface cards), 382–384

- routers, 386
- switches, 385–386
- Internet connections
 - common configuration information, 408–410
 - dial-up Internet access, 412–413
 - overview, 408
 - PPPoE (Point-to-Point Protocol over Ethernet), 410–412
 - troubleshooting, 413
- loopback interface, 370–371
- network configuration tools, 389
- network printers, creating, 483–485
- online resources, 414
- overview, 369
- related Ubuntu and Linux commands, 414
- storage, 349
- subnetting, 381
- TCP/IP (Transport Control Protocol/Internet Protocol). See TCP/IP (Transport Control Protocol/Internet Protocol)
- VPNs (virtual private networks). See VPNs (virtual private networks)
- wireless networks. See wireless networks
- Neutron, 646
- Newell, Gabe, 123
- newsgroups, 72–74
- NewSQL databases. See NoSQL databases
- newusers command, 264
- NFS (Network File System)
 - client configuration, 475–476
 - installation, 474
 - overview, 474
 - server configuration, 474–475
 - starting, 474
 - stopping, 474
- nfs-common package, 475
- Nginx
 - configuration, 518–520
- HTTPS (HTTP Secure), 524–526
- installation, 517–518
- modules, 523–524
- online resources, 526
- overview, 515–517
- PHP setup, 522–523
- virtual hosting, 521–522
- nginx package, 517
- nice command, 218–219, 326–327
- NICs (network interface cards), 382–384
- Nmap, 428
- nm-connection-editor, 389
- NNTP (Network News Transfer Protocol), 73
- nonbooting hard drive, saving files from, 368
- NoSQL databases
 - BaseX, 610–611
 - Berkeley DB, 606–607
 - BigTable, 611
 - Cassandra, 607
 - CouchDB, 609
 - FlockDB, 613
 - HBase, 611
 - HyperGraphDB, 612–613
 - Memcached, 607–608
 - MemcachedDB, 607–608
 - MongoDB, 610
 - Neo4j, 612
 - online resources, 613
 - OrientDB, 612
 - overview, 579–581, 603–606
 - Redis, 608
 - Riak, 608
 - Nova, 645–646
 - NS record (DNS), 663–664
 - numbers
 - number comparison
 - in pdksh and bash, 292–293

in tcsh, 296–297
 numbering system
 Linux kernels, 30
 Ubuntu, 33, 144–145
 octal notation, 166

O

object files, 670
 OCaml, 718
 ocaml package, 718
 ocamlc, 718
 ocamlopt, 718
 octal notation, 166
 ODSL (Open Source Development Labs), 29
 .ogg filename extension, 88, 106
 Ogg-Vorbis format, 88
 .ogv filename extension, 106
 online resources
 Android development, 728
 Apache, 513
 backups, 368
 boot process, 322
 cloud computing, 654
 command line, 184, 240
 DNS (Domain Name System), 667
 email, 558–559
 file- and printer-sharing, 488
 FTP (File Transfer Protocol), 540
 games, 130–131
 GUIs (graphical user interfaces), 118–119
 help
 commercial support, 38–39
 Linux guides, 39–40
 LUGs (Linux User Groups), 39
 mailing lists, 40–42

reading, 35–36
 Ubuntu-related websites, 40–42
 Web search tips, 37–38
 Internet, 74
 kernel, 472
 languages, 721–722
 LDAP (Lightweight Directory Access Protocol), 622
 LTSP (Linux Terminal Server Project), 629
 multimedia applications, 109
 networks, 414
 Nginx, 526
 NoSQL databases, 613
 OpenStack, 647
 opportunistic development, 692
 performance tuning, 452
 productivity suites and applications, 83
 programming tools, 680
 remote access, 423
 security, 439
 server management, 657
 shell scripts, 308–309
 software management, 147
 system-monitoring tools, 340
 testing and QA, 708
 Ubuntu development, 702
 Ubuntu installation and configuration, 27
 Ubuntu Mobile development, 731
 users, 264
 virtualization, 639
 web servers, 530
 X, 61
 Open Sound System (OSS), 86
 open source, 20
 Open Source Development Labs (ODSL), 29
 open-cobol package, 712

- OpenDocument, 78
- OpenLDAP, 615
- OpenOffice.org, 77–78
- OpenShot Video Editor, 108
- openssh-server package, 417
- OpenStack, 645–647
- OpenVPN, 573–575
- Open-Xchange, 558
- operators
 - bar operator (|), 219–221
 - Boolean operators, 221–222
 - file operators, 297–298
 - logical operators, 298–299
 - pdksh and bash
 - file operators, 293–294
 - logical operators, 294–295
- opportunistic development. See *also* version control systems
 - Bikeshed, 689–691
 - definition of, 681
 - Launchpad, 687–688
 - online resources, 692
 - overview, 686–687
 - repository tools, 691–692
 - snap packaging, 689
 - Ubuntu Make, 688–689
- optimization
 - Apache, 446–447
 - file system tuning, 444–445
 - hard disk
 - BIOS settings, 442–443
 - hdparm command, 443
 - overview, 441–442
 - kernel, 445–446
 - MySQL
 - key buffer usage, 448–449
 - overview, 448
 - query cache, 449–451
 - query optimization, 451–452
 - read buffer, 451
 - table cache, 451
 - online resources, 452
- Options directive, 497
- Oracle
 - Beehive, 557
 - JDeveloper, 679
 - OpenOffice.org and, 77–78
- orchestration engines, Ansible, 657
- Orient ODBMS, 612
- OrientDB, 612
- OSS (Open Sound System), 86
- output
 - BIOS (basic input/output system)
 - beep codes, 313
 - boot process, 311–313
 - checking, 235
 - hard disk optimization, 442–443
 - I/O (input/output redirection), 209–212, 275–276
 - paging through, 197–198

P

- PaaS (platform as a service), 643
- package management utilities, apt-get, 16
- packages
 - apache2, 490
 - apparmor-profiles, 436
 - bikeshed, 689
 - bind9, 666
 - bridge-utils, 634
 - clisp, 716–717
 - closure, 711
 - default-jdk, 716
 - elixir, 713

erlang, 713
 gforth, 714
 ghc, 715
 gnat, 710
 gnat-gps, 710
 gnome-shell, 116
 gofortran, 714–715
 golang, 714
 golang-docs, 714
 groovy, 715
 gstreamer0.10-plugins-ugly, 88
 haskell-platform, 715
 kubuntu-desktop, 113–114
 kvm-ok, 633–634
 ldap-utils, 616
 libvirt-bin, 634
 linux-backports-modules, 461
 linux-source, 462
 listing, 38
 lua50, 717
 lubuntu-desktop, 116
 mono-devel, 718
 nautilus-actions, 363
 nautilus-image-converter, 97
 nfs-common, 475
 nginx, 517
 ocaml, 718
 open-cobol, 712
 openssh-server, 417
 packaging-dev, 695–696
 patches, 698–701
 postfix, 546
 postgres, 600
 PPAs (personal package archives)
 creating, 688
 Ubuntu SDK, 730
 qemu-kvm, 634
 ruby1.8, 719
 scala, 720
 Scratch, 720
 scratch, 720
 slapd, 616
 snap packaging, 146–147, 689
 soundconverter, 89
 telnetd, 416
 ubuntu-gnome-desktop, 117
 ubuntu-make, 688
 ubuntu-qa-tools, 705
 ubuntu-restricted-extras, 88, 106
 ubuntu-software, 133
 Vala, 720–721
 valac, 721
 virtinst, 634
 virt-manager, 634
 virt-viewer, 634
 xubuntu-desktop, 115
 packaging-dev package, 695–696
 packet writing, 103–104
 paging through output, 197–198
 PAM (Pluggable Authentication Modules), 255–256
 Pan, 73–74
 panel (Unity), 57–58
 parameters, positional, 282–284
 parentheses (), 222
 partition strategies
 GParted, 11
 /home partition, 10
 planning, 5
 root partitions, 10
 swap partitions, 10
 passwd command, 249
 passwd files, 159, 253–254
 passwords
 administrator settings for, 256

changing in batches, 256–257
choosing, 12, 429–430
`/etc/passwd` file, 253–254
MySQL, 589–590
policy, 252
shadow passwords, 254–255
`pastebinit` command, 691
`patch` command, 464
patching kernel, 460, 463–464, 698–701
`patch-kernel` script, 463
pattern matching, 274–275
`pbget` command, 690
`pbm` (portable bitmap), 97
`pbsput` command, 690
`pbsputs` command, 690
`pbuilder` command, 695, 696
.pcx filename extension, 96
PDF (Portable Document Format) files
 editing, 81
 reading, 36, 80–81
PDF Editor, 81
.pdf filename extension, 36
peers file, 549
Percona Server, 577
performance tuning
 Apache, 446–447
 file system tuning, 444–445
 hard disk
 BIOS settings, 442–443
 `hdparm` command, 443
 overview, 441–442
 kernel, 445–446
MySQL
 key buffer usage, 448–449
 overview, 448
 query cache, 449–451
 query optimization, 451–452
 read buffer, 451
 table cache, 451
 online resources, 452
Perl, 718
permissions
 altering
 `chgrp` command, 170
 `chmod` command, 168–169
 `chown` command, 170
 `umask` command, 169–170
 assigning, 166–167
 changing, 191
 directory permissions, 167–168
 file permissions, 191, 244
 overview, 244
 setting, 171–173
 `sgid` (set group ID), 170–171
 sticky bit permissions, 171
 `suid` (set user ID), 170–171
 viewing, 165–166
personal package archives (PPAs)
 creating, 688
Ubuntu SDK, 730
personal video recorders, 107
`pgAdmin`, 600
`pgm` (portable graymap), 97
Photoshop, 94
PHP
 configuration, 522–523
 overview, 719
PHP and MySQL Web Development (Welling and Thompson), 591
physical security, 429–430
PID (process ID), 314, 324
Pidgin, 70
ping command, 371–373, 667
piping, 187, 219–221, 276
PiTiVi, 107–108

PKI (public key infrastructure), 573
 Plain Old Java Objects (POJOs), 529
 Planner, 77
 planning
 disaster recovery plans, 437–438
 partition strategies, 5
 platform as a service (PaaS), 643
 Pluggable Authentication Modules (PAM), 255–256
 plug-ins
 Firefox, 64–65
 GStreamer, 88
 Plymouth, 314
 .png filename extension, 96
 pnm (portable anmap), 97
 Point-to-Point Protocol over Ethernet (PPPoE), 410–412
 Point-to-Point Protocol (PPP), 408
 POJOs (Plain Old Java Objects), 528–529
 policy, password, 252
 Pomerantz, Ori, 39
 POP3 (Post Office Protocol version 3), 545
 populating LDAP directories, 617–619
 portable anmap (pnm), 97
 portable bitmap (pbm), 97
 Portable Document Format (PDF) files
 editing, 81
 reading, 36, 80–81
 portable graymap (pbm), 97
 portable pixmap (ppm), 96
 ports, 377
 positional parameters, 282–284
 Post Office Protocol version 3 (POP3), 545
 Postfix
 aliases, 550–551
 configuration, 546–548
 mail relaying, 550
 masquerading, 548
 message delivery intervals, 549
 overview, 543
 smart hosts, 549
 postfix package, 546
Postfix: The Definitive Guide (Dent), 548
 postgres package, 600
 PostgreSQL
 compared to MySQL, 586–588
 data directory initialization, 592
 database clients
 graphical clients, 600
 local GUI client access, 595–596
 MySQL command-line client, 598–600
 overview, 595
 PostgreSQL command-line client, 600
 SSH access, 595–596
 web access, 595–596
 database creation, 593
 installation, 592
 privileges, 594–595
 user creation, 593–594
 user deletion, 594
 post-installation configuration
 power management, 22
 printers, 22
 software repositories, 19–21
 system settings, 21
 time/date
 date command, 24
 hwclock command, 24
 overview, 23
 Time & Date tool, 23
 troubleshooting, 26–27
 wireless networks, 24–25
 PostScript files, reading, 36
 pound sign (#), 155, 243
 POV-Ray, 98
 power management configuration, 22

power shortcuts, 60

PPAs (personal package archives)

- creating, 688
- Ubuntu SDK, 730

ppm (portable pixmap), 96

PPP (Point-to-Point Protocol), 408

pppconfig command, 412

pppd daemon, 549

PPPoE (Point-to-Point Protocol over Ethernet), 410–412

Practical Extraction and Report Language.
See Perl

A Practical Guide to Linux Commands, Editors, and Shell Programming, Third Edition, 40

The Practice of System and Network Administration, Second Edition, 40

preparation for Ubuntu installation

- 32-bit versus 64-bit Ubuntu, 4–5
- backups, 7
- hardware specifications, researching, 2
- installation options, 2–4
- overview, 1–2
- partition strategies, planning, 5

preprocessors, 670

previous command, running, 236–237

printers

- all-in-one devices, 487
- configuration, 22
- detecting, 22
- printing-related commands, 487
- sharing
 - CUPS (Common UNIX Printing System)
GUI, 483–486
 - network printers, 483–485
 - Samba, 480–481
 - USB and legacy printers, 487
- [printers] section (smb.conf), 480–481

printing

- command location, 207
- disk usage, 192
- files, 185–189
- resource usage, 216–218

priority scheduling, 218–219, 326–327

private cloud, 644

privileges

- PostgreSQL, 594–595
- system administrator privileges, granting
 - overview, 257
 - with su command, 257–259
 - with sudocommand, 259–262
- /proc directory, 160–161, 323–324
- procedural languages, 588
- process substitution, 222

processes

- listing, 213–214
- priority scheduling, 218–219
- process control, 325–326
- process ID (PID), 314, 324
- process substitution, 222

processors, ARM, 724

Procmail, 555

productivity

- online resources, 83
- productivity applications
 - CrossOver Office, 83
 - Evince, 80–81
 - Gedit, 81
 - Kile, 82
 - PDF Editor, 81
 - Publican, 81–82
 - Texmaker, 82
 - Wine, 83
 - XML Copy Editor, 82
- productivity suites
 - definition of, 75–76
 - GNOME Office, 78–79

- KOffice, 80
- LibreOffice, 76–78
- Microsoft Office, 75
- ProFTPD, 533
- programming languages. See languages
- programming tools
 - autoconf command, 673–674
 - gdb command, 675
 - GNU Compiler Collection (gcc), 675–676
 - gprof command, 675
 - graphical development tools
 - Glade client, 677–678
 - IDEs (integrated development environments), 678–680
 - KDevelop client, 676–677
 - overview, 676
 - SDKs (software development kits), 678–680
 - make command, 671–673
 - online resources, 680
 - overview, 669
 - splint command, 674–675
- [programs] section (`smb.conf`), 479
- promiscuous mode, 431
- proprietary software, 20
- proprietary video drivers, installing, 122–123
- proxy servers
 - definition of, 561
 - online resources, 575
 - Squid
 - ACLs (access control lists), 563–567
 - client configuration, 562–563
 - client IP addresses, 567–568
 - installation, 562
 - sample configurations, 568–569
- ps command, 213–214
- .ps filename extension, 36
- psksh, comparing expressions in
 - file operators, 293–294
 - logical operators, 294–295
 - number comparison, 292–293
 - string comparison, 290–292
- pgsql command, 600
- public cloud, 644
- public key infrastructure (PKI), 573
- Publican, 81–82
- PulseAudio, 86–87
- Puppet, 656
- purge-old-kernels command, 691
- pwd command, 152–153, 165
- PXE, 314
- Python, 719

Q

- QA (quality assurance)
 - QA team, 705
 - Ubuntu testing, helping with
 - Bug Squad, 705
 - community teams, 703–704
 - online resources, 708
 - QA team, 705
 - Test Drive, 705–708
 - Ubuntu testing team, 704–705
- qemu-kvm package, 634
- Qmail, 543–544
- Quassel, 70
- query cache optimization, 449–451
- question mark (?), 275
- Quickstart (Juju), 653
- quickstart command, 653
- quotacheck command, 263

quotaoff command, 263
 quotaon command, 263
 quotas, disk, 262–263
 quotation marks, 288

R

rabbit-mq, 646
 Radiance, 98
 RAID (redundant array of independent disks), 348
 Rails, 719
 RARP (Reverse Address Resolution Protocol), 382
 .raw filename extension, 88
 RAW format, 88
 Raymond, Eric, 185
 rc script, 316
 RDBMSs (relational database management systems). See relational databases
 RDP (Remote Display Protocol), 638
 read buffer optimization, 451
 read-eval-print loop (REPL), 711
 reading
 documentation
 apropos command, 156–157
 man pages, 156, 203
 overview, 35–36
 whereis command, 157
 kernel ring buffer, 239
 PDF (Portable Document Format) files, 80–81
 rebooting system, 183
 receiving email. See email
 recompiling kernel, 460–461
 A record (DNS), 662
 records (DNS)

A, 662
 AAAA, 662
 CNAME, 663
 MX, 663
 NS, 663–664
 SOA, 664–665
 TXT, 665
 recovery mode, 236. *See also* security; troubleshooting
 redirection, I/O (input/output), 209–212, 275–276
 Redis, 608
 reduced instruction set computer (RISC), 724
 redundant array of independent disks (RAID), 348
 references. *See* online resources
 regular expressions, 177
 reinstalling
 GRUB2 (Grand Unified Boot Loader), 235–236
 Ubuntu, 236
 REISUB, 234–235
 relational database management systems (RDBMSs). *See* relational databases
 relational databases
 ACID compliance (databases), 587–588
 data insertion, 583–584
 data locking, 586–587
 data retrieval, 584–586
 database clients
 graphical clients, 600
 local GUI client access, 597
 MySQL command-line client, 598–600
 overview, 595
 PostgreSQL command-line client, 600
 SSH access, 595–596
 web access, 597
 DBAs (database administrators), 577–578
 MySQL. *See* MySQL

overview, 579–581

PostgreSQL. See PostgreSQL

speed of, 586

table creation, 582–583

release command, 690

release cycles (Ubuntu), 694–695

release-build command, 690

remote access

- command line, 153–154
- FTP (File Transfer Protocol). See FTP (File Transfer Protocol)
- online resources, 423
- SSH (Secure Shell)
 - compared to Telnet, 417
 - key-based logins, 419–420
 - scp command, 418
 - sftp command, 417, 418–419
- Telnet, 415–417
- VNC (Virtual Network Computing), 420–423

Remote Display Protocol (RDP), 638

remote file serving. See FTP (File Transfer Protocol)

removing software

- with APT (Advanced Package Tool), 141
- with Synaptic, 134–135

rename command, 204

renaming files, 175–176, 204

renice command, 326–327

repeat statement, 303

REPL (read-eval-print loop), 711

repositories (software)

- compiling software from, 144–145
- configuration, 19–21
- Git, 682–683

repquota command, 263

requests (DNS), 661–662

require directive, 498–499, 500

rescue, 366–368

researching hardware specifications, 2

resetting system, 234–235

resolv.conf file, 396–397

resource use, limiting

- & (ampersand), 215
- bg command, 215–216
- fg command, 215–216
- jobs command, 214
- ps command, 213–214
- top command, 216–218

restarting slapd, 616

restoring GRUB2 (Grand Unified Boot Loader), 367

retrieving data from SQL databases, 584–586

Reverse Address Resolution Protocol (RARP), 382

reverse-i search, 237

REVOKE statement, 591, 595

revoking PostgreSQL privileges, 594–595

RFC (Requests for Comments) 1036, 73

Rhythmbox, 89–91

Riak, 608

rights. See privileges

ripping CDs, 99

RISC (reduced instruction set computer), 724

RivaTV, 104

rm command, 175, 205

rmdir command, 173–174

rmmod command, 458

root partitions, 10

root prompt, 18

root users

- creating users, 181–182
- deleting users, 182
- overview, 178, 242–244
- sudo command, 178–181

root zones, 661
 rotating log files, 331–333
 route command, 392–394
 routers, 386
 RSS readers
 Firefox, 69
 Liferea, 69–70
 overview, 69
 rsync command, 362–364
 rtcwake command, 270–272
 Ruby, 719
 Ruby Gems, 719
 Ruby on Rails, 719
 ruby1.8 package, 719
 runlevels
 changing, 318–319
 default runlevel, booting into, 316
 definitions, 315
 overview, 51, 311–312
 purpose of, 315
 troubleshooting, 319
 running
 commands
 multiple commands, 237–238
 previous command, 236–237
 in sequence, 222
 jobs in background, 215
 services at boot, 311–312
 shell scripts, 279
 run-one command, 691
 run-this-one command, 691
 Runtime (Android), 724
 runtime server configuration settings
 .htaccess configuration files, 496–498
 MPMs (multiprocessing modules),
 495–496
 Rust, 720

S

SaaS (software as a service), 643
 Salzman, Peter J., 39
 Samba
 configuration, 476–481
 connection status, checking, 481–482
 overview, 476–478
 shares, mounting, 482–483
 smbclient command, 482
 stopping/starting, 481
 testing, 481
Sams Teach Yourself TCP/IP Network Administration in 21 Days, 381
 satisfy directive, 501
 saving files from nonbooting hard drive, 368
 /sbin directory
 overview, 158–159
 /sbin/ifconfig, 389–391
 /sbin/ip, 392
 /sbin/netstat, 394
 /sbin/route, 392–394
 Scala, 720
 scala package, 720
 scanners, 94–95
 scheduling
 priority scheduling, 218–219, 326–327
 task scheduling
 batch command, 265–268
 at command, 265–268
 cron command, 268–270
 rtcwake command, 270–272
 schema (LDAP), creating, 616–617
 Scheme, 717
 scopes
 development, 698
 Smart Scopes, 59
 Scorched 3D, 124–125

scp command, 418
Scratch package, 720
scratch package, 720
screen images, capturing, 97
Screen section (*xorg.conf*), 50–51
script kiddies, 426
scripts
 confining to directories, 238–239
 init scripts, 316–317
 patch-kernel, 463
 rc, 316
SDKs (software development kits)
 Android SDK, 725–727
 overview, 678–680
 Ubuntu SDK, 698, 730
search engines, 37–38
searching
 for files
 find command, 193–195
 locate command, 200
 for strings, 196
Secure Shell. See SSH (Secure Shell)
Secure Sockets Layer, 506
“Securing and Optimizing Linux” (Mourani), 40
security
 AppArmor, 435–437
 built-in kernel protection, 425
 computer attacks, 425–427
 devices, 431
 disaster recovery plans, 437–438
 encryption, 10
 firewalls
 iptables, 435
 UFW (Uncomplicated Firewall), 432–435
 online resources, 439
 passwords
 administrator settings for, 256
 changing in batches, 256–257
 choosing, 12, 429–430
 /etc/passwd file, 253–254
 policy, 252
 shadow passwords, 254–255
 physical security, 429–430
 Tripwire, 430–431
 viruses, 431–432
 vulnerability assessment, 427–428
 wireless networks, 429
sed command, 230–232
SELECT statement, 584–585
select statement, 303–304
semicolon (;), 222, 306, 403, 583
semistructured data, 608
Sender Policy Framework (SPF), 665
sending email. See email
Sendmail, 542–543
sequence, running commands in, 222
Server install DVD, 2
ServerAdmin directive, 494
ServerLayout section (*xorg.conf*), 47
ServerName directive, 494–495, 508
ServerRoot directive, 493
servers
 Apache
 access control, 498–501
 HTTPS (HTTP Secure), 510–512
 installation, 490–491
 logging, 509–510
 MPMs (multiprocessing modules), 495–496
 overview, 489–490
 runtime server configuration settings, 492–498
 starting, 491–492
 stopping, 491–492
 virtual hosting, 507–509

- Apache Tomcat, 530
- Cherokee, 528–529
- DHCP server, 402
- DNS (Domain Name System)
 - configuring with BIND, 665–667
 - request handling, 661–662
- email, 556–558
- FTP (File Transfer Protocol)
 - anonymous servers, 531–532
 - authenticated servers, 532
 - Bsdftpd-ssl, 533
 - NcFTPd, 532–533
 - ProFTPD, 533
 - standard servers, 531–532
 - Very Secure FTP server, 532, 536–538
 - wu-ftp, 533
- IRCd, 72
- Jetty, 529
- LDAP (Lightweight Directory Access Protocol), 616
- lighttpd, 527–528
- LTSP (Linux Terminal Server Project). See LTSP (Linux Terminal Server Project)
- management
 - Ansible, 657
 - CFEngine, 656–657
 - Chef, 656
 - juju, 655
 - Landscape, 657
 - online resources, 657
 - overview, 655
 - Puppet, 656
- NFS (Network File System), 474–475
- Nginx. See Nginx
- online resources, 530
- Squid. See Squid
- SSH (Secure Shell). See SSH (Secure Shell)
- Telnet, 415–417
- thttpd, 529–530
- virtualization. See virtualization
- VPNs (virtual private networks) servers, 573–575
- X.Org
 - directories, 45–46
 - history of, 43–44
 - xorg.conf file, 46–51
- Yaws, 528
- service set identifiers (SSIDs), 25
- services
 - booting, 315
 - controlling at boot, 317–318
 - running at boot, 311–312
 - starting/stopping manually, 319
- services file, 395–396
- Session Message Block (SMB), 476
- session writing, 103
- set group ID (sgid) permissions, 170–171
- set user ID (suid) permissions, 170–171
- settings. See configuration
- sftp command, 417, 418–419
- sgid (set group ID) permissions, 170–171
- shadow passwords, 254–255
- sharding, 610
- shares (Samba), mounting, 482–483
- sharing
 - files
 - NFS (Network File System), 474–476
 - online resources, 488
 - Samba, 476–483
 - printers
 - CUPS (Common UNIX Printing System)
 - GUI, 483–486
 - network printers, 483–485
 - Samba, 480–481
- shebang (#!), 280
- shell scripts. See also command line

available shells, 272
background processing, 277
comparison of expressions
 in pdksh and bash, 290–295
 in tcsh, 295–299
confining to directories, 238–239
data piping, 276
executing, 279
functions, 307–308
greplog, 284–286
interpreting through specific shells,
 280–281
I/O (input/output redirection), 275–276
online resources, 308–309
pattern matching, 274–275
shell command line, 273–274
special characters
 backslash (\), 289
 backtick (`), 289–290
 double quotation marks ("), 288
 single quotation marks ('), 288–289
 table of, 287–288
statements. See statements
storing, 279–280
task scheduling. See task scheduling
variables. See variables
writing, 277–278
shells. See also shell scripts
 command line, 273–274
 table of, 272
shift statement, 304
shortcuts (keyboard), 60, 238
shutdown command, 13, 182–183
Shuttleworth, Mark, 33, 52
similarities between files, finding, 212
Simple Mail Transfer Protocol (SMTP), 377, 541
Simple Scan, 94–95
single quotation marks ('), 288
slapd package, 616
Slashdot Effect, 446–447
Sleepycat Software, 606, 615
small office/home office (SOHO) users, 35
smart gateways, 386
smart hosts, 549
Smart Scopes, 59
smbclient command, 482
smb.conf file, 476–481
smbstatus command, 481–482
SMPs (symmetric multiprocessors), 33–34
SMTP (Simple Mail Transfer Protocol),
 377, 541
snap command, 146–147
snap packaging, 689
Snapcraft, 689
Snappy Ubuntu Core, 146–147, 653
snaps, 146–147
Snyder, Garth, 40
SOA record (DNS), 664–665
Sobell, Mark G., 40
socks-prox command, 690
soft links, 199–200
Software AGs Adabas D database, 77
software as a service (SaaS), 643
software compilation, JIT (just-in-time), 724
software development kits. See SDKs
 (software development kits)
software libraries (Android), 724
software licensing
 GNU General Public License
 (GPL), 29
software management
 APT (Advanced Package Tool). See APT
 (Advanced Package Tool)
 configuration
 dotdee (.d), 145
 Snappy Ubuntu Core, 146–147
 online resources, 147

- overview, 133
- software compilation
 - from tarballs, 143–144
 - from Ubuntu repository source, 144–145
- Software Updater, 137
- Synaptic, 134–136
- Ubuntu Software, 133–134
- software repositories
 - compiling software from, 144–145
 - configuration, 19–21
- Software Updater, 15–18, 137
- software updater command, 15
- SOHO (small office/home office) users, 35
- sort command, 205–207
- sorting file contents, 205–207
- sound
 - listening to
 - Banshee, 91–93
 - overview, 89
 - Rhythmbox, 89–91
 - overview, 85–86
 - sound cards, 86–87
 - sound formats, 88–89
 - volume adjustment, 87
- sound cards, 86–87
- Sound Juicer, 91–92
- soundconverter package, 89
- sox command, 89
- Spamassassin, 555–556
- Speed Dreams, 129
- speed of relational databases, 586
- SPF (Sender Policy Framework), 665
- splint command, 674–675
- SQL databases
 - ACID compliance (databases), 587–588
 - data insertion, 583–584
 - data locking, 586–587
- data retrieval, 584–586
- database clients
 - graphical clients, 600
 - local GUI client access, 597
- MySQL command-line client, 598–600
 - overview, 595
 - PostgreSQL command-line client, 600
 - SSH access, 595–596
 - web access, 597
- MySQL
 - compared to PostgreSQL, 586–588
 - database creation, 590–591
 - installation, 588–589
 - passwords, 589–590
- PostgreSQL
 - compared to MySQL, 586–588
 - data directory initialization, 592
 - database creation, 593
 - installation, 592
 - privileges, 594–595
 - user creation, 593–594
 - user deletion, 594
 - speed of, 586
 - subqueries, 588
 - table creation, 582–583
 - triggers, 588
- squid.conf file, 563
- Squirrelmail, 556
- SSH (Secure Shell)
 - accessing IRC clients over, 70
 - compared to Telnet, 417
 - database access, 595–596
 - key-based logins, 419–420
 - overview, 15
 - scp command, 418
 - sftp command, 417, 418–419
- SSH keys

generating, 696
 key-based logins, enabling, 419–420
 uploading to Launchpad, 697

`sshd_config` file, 377

`ssh-import-id` command, 691

`ssh-keygen` command, 419–420

SSIDs (service set identifiers), 25

Stallman, Richard M., 29–30, 454

standard FTP (File Transfer Protocol)servers, 531–532

Stanzone, Dan, 670

StarOffice, 77

Start a new Android Studio Project command, 727

start of authority (SOA) records, 664–665

starting

- Apache, 491–492
- NFS (Network File System), 474
- Samba, 481
- services
 - at boot, 311–312, 315
 - manually, 319
- `slapd`, 616
- X, 51

Startup Disk Creator, 3–4

statements. *See also* loops

- `DROP USER`, 594
- `LogFormat`, 509
- shell scripts
 - `break`, 307
 - `case`, 305–307
 - `exit`, 307
 - `for`, 299–300
 - `if`, 304–305
 - `repeat`, 303
 - `select`, 303–304
 - `shift`, 304
 - `until`, 302–303
- `while`, 300–302

SQL

- `CREATE`, 582–583
- `CREATE DATABASE`, 590, 593
- `CREATE USER`, 593
- `GRANT`, 591, 594
- `INSERT`, 583–584
- `REVOKE`, 591, 595
- `SELECT`, 584–585

`stderr` command, 211–212

`stdin` command, 211–212

`stdout` command, 211–212

Steam platform, 123

step-by-step Ubuntu installation, 6–12

stereotypes (user), 245

sticky bit permissions, 171

stopping

- Apache, 491–492
- NFS (Network File System), 474
- Samba, 481
- services, 319
- `slapd`, 616

storage, 279–280

storage infrastructure (OpenStack), 646

stored procedures, 588

stores. *See* databases

strategy, backup. *See* backups

Stream Editor (`sed`), 230–232

strings

- searching, 196
- string comparison
 - in pdksh and bash, 290–292
 - in tcsh, 290–292

`StumbleUpon`, 64

`su` command, 257–259

subnet masks, 381

subnetting, 381

subqueries (SQL), 588

substitution

- command substitution, 289–290
- process substitution, 222

Subversion, 684–685

sudo command

- overview, 18, 259–262
- troubleshooting, 178–181

sudoers file, 159, 260–261

suid (set user ID) permissions, 170–171

Sun Microsystems, 77, 715–716

super users/root users

- creating users, 181–182
- deleting users, 182
- overview, 154–155, 178, 242–244
- sudo command, 178–181

SuperTux, 126

suspending system, 11, 22, 270–272

.svg filename extension, 96

svn add command, 685

svn checkout command, 685

svn commit command, 685

svn delete command, 685

svn import command, 684

svnadmin create command, 684

swap partitions, 10

Swift, 646

switches, 385–386

symbolic links, 199–200

symlinks, 199–200

symmetric multiprocessors (SMPs), 33–34

Synaptic, 134–136

sync command, 444

synchronizing file system, 444

sysadmins, 641–642

sysctl command, 445–446

syslog, 330

sysrq.txt file, 456

system administrator privileges, granting

- overview, 257
- with su command, 257–259
- with sudocommand, 259–262

system administrators, 641–642

system information, listing, 202–203

System Monitor, 334

system recovery, 236

system rescue, 366–368

system reset, 234–235

system services

- booting, 315
- controlling at boot, 317–318
- starting/stopping manually, 319

system settings

- Ubuntu, 21
- Unity, 58–60

System Settings menu, Time & Date tool, 23

system-config-printer client, 483

systemctl command, 321

systemd, 312, 314, 318, 320–321, 339

system-management tools

- Conky, 334–339
- gnome-nettool, 339
- overview, 333–334
- System Monitor, 334
- vncviewer, 339
- wireshark, 339

system-monitoring tools

- console-based monitoring
- df command, 328–329
- disk quotas, 329
- free command, 327–328
- kill command, 325–326

- log files, 329–333
- overview, 323–325
- priority scheduling, 326–327
- vmstat, 328
- enterprise server monitoring, 340
- graphical and system-management tools
 - Conky, 334–339
 - gnome-nettool, 339
 - overview, 333–334
 - System Monitor, 334
 - vncviewer, 339
 - wireshark, 339
- KDE- and system-monitoring tools, 339
- online resources, 340
- user activity monitoring, 251–252
- system-search command, 690
- SyvVinit, 312, 314

T

- tables
 - cache optimization, 451
 - GPT (GUID Partition Table), 314
 - SQL tables, creating, 582–583
- tail command, 207, 329
- tape drive backups, 349–350
- tar command, 232, 351–353, 360–361
- tarballs, compiling software from, 143–144
- task scheduling
 - batch command, 265–268
 - at command, 265–268
 - cron command, 268–270
 - rtcwake command, 270–272
- tasks. See jobs
- Taylor, David, 85–86
- tcl (Tool Control Language), 281
- TCP (Transport Control Protocol), 374
- TCP/IP (Transport Control Protocol/Internet Protocol)
 - IP masquerading, 376–377
 - IPv4 addressing, 374–376
 - IPv6 addressing, 378–380
 - overview, 374
 - ports, 377
- tcsh, comparing expressions in
 - file operators, 297–298
 - logical operators, 298–299
 - number comparison, 296–297
 - string comparison, 290–292
- teams, community, 703–705
- telinit command, 318
- Telnet, 415–417
- telnet command, 416
- telnetd package, 416
- terminal. See command line
- terminal command, 16
- Test Drive, 705–708
- testdrive command, 706
- testdrive-gtk package, 708
- testdriverc file, 706
- testing
 - DNS (Domain Name System), 667
 - Samba, 481
 - Ubuntu testing, helping with
 - community teams, 703–705
 - online resources, 708
 - Test Drive, 705–708
- Texmaker, 82
- text editors
 - awk, 230–232
 - emacs, 229–230
 - nano, 227
 - overview, 226

- sed, 230–232
- vi, 227–228
- text files, reading, 36
- text-based console login, 152
- thin clients, 623
- Thompson, Laura, 591
- threads, 73, 516
- httpd, 529–530
- Thunar, 114
- Thunderbird, 66–67, 621
- .tif filename extension, 96
- tilde (~), 152
- Time & Date tool, 23
- time warp, 252
- time/date configuration
 - date command, 24
 - hwclock command, 24
 - overview, 23
 - Time & Date tool, 23
- TiVo, 107
- TLDs (top-level domains), 661
- /tmp directory, 162
- Token Ring, 383
- Tool Control Language (tcl), 281
- top command, 216–218, 327
- top-level domains (TLDs), 661
- Torvalds, Linus Benedict, 29, 454, 682
- Totem Movie Player, 106
- touch command, 165–166, 173
- touchscreen platforms, developing for
 - applications, creating, 730
 - online resources, 731
 - overview, 729–730
- SDK (software development kit)
 - installation, 730
- traceroute command, 373
- Transport Control Protocol (TCP), 374
- Transport Control Protocol/Internet Protocol.
 - See TCP/IP (Transport Control Protocol/Internet Protocol)
- Transport Layer Security, 506
- traversals, 612
- triggers, 588
- Tripwire, 430–431
- tripwire command, 430
- Trojan horses, 430
- troubleshooting. See also security boot process
 - with Boot Repair, 320–321
 - overview, 235–236
 - starting/stopping services manually, 319
 - with systemd, 320–321
 - with Upstart, 319–320
- help
 - commercial support, 38–39
 - documentation, 35–36
 - LUGs (Linux User Groups), 39
 - Web search tips, 37–38
- Internet connections, 413
- kernel, 470–472
- post-installation configuration, 26–27
- printers, 487
- runlevels, 319
- sudo command, 178–181
- UEFI (Unified Extensible Firmware Interface), 6
- tune2fs command, 444
- tuning performance. See performance tuning
- TV and video hardware, 104–105
- TXT record (DNS), 665

U

- Ubuntu Announcements mailing list, 41
- #Ubuntu channel, 42

Ubuntu Developers' Summit (UDS), 694
 Ubuntu GNOME, 116–117
 Ubuntu Kylin, 118
 Ubuntu Make, 688–689
 Ubuntu MATE, 117–118
 Ubuntu Metal as a Service (MaaS), 653–654
 Ubuntu Mobile, developing for applications, creating, 730
 online resources, 731
 overview, 729–730
 SDK (software development kit) installation, 730
 Ubuntu repository, compiling software from, 144–145
 Ubuntu SDK, 698
 Ubuntu Software, 133–134
 Ubuntu testing team, 704–705
`ubuntu-gnome-desktop` package, 117
`ubuntu-make` package, 688
`ubuntu-qa-tools` package, 705
`ubuntu-restricted-extras` package, 65, 88, 106
`ubuntu-software` package, 133
 UDP (Universal Datagram Protocol), 374
 UDS (Ubuntu Developers' Summit), 694
 UEFI (Unified Extensible Firmware Interface), 6, 313–314
 UFW (Uncomplicated Firewall), 432–435
`ufw` command, 432–433
`ufw.log`, 330
 UIDs (user IDs), 244
`umask` command, 165, 169–170
`uname` command, 30
 Uncomplicated Firewall (UFW), 432–435
 unicast addressing, 382
 Unified Extensible Firmware Interface (UEFI), 6, 313–314
 uniform resource identifiers (URIs), 485–486

Unity
 configuration
 CCSM (CompizConfig Settings Manager), 59
 Smart Scopes, 59
 system settings, 58
 Unity Tweak Tool, 59–60
 desktop
 Dash, 54–57
 default look, 53
 Launcher, 53–54
 Panel, 57–58
 overview, 52–53
 Ubuntu Mobile, developing for applications, creating, 730
 online resources, 731
 overview, 729–730
 SDK (software development kit) installation, 730
 Unity Tweak Tool, 59–60
 Universal Datagram Protocol (UDP), 374
 Universal USB Installer, 3–4
 UnQL (Unstructured Query Language), 605
 unshielded twisted-pair (UTP), 383, 384–385
 Unstructured Query Language (UnQL), 605
 until statement, 302–303
 updates
 checking for
 APT (Advanced Package Tool), 138
 Software Updater, 15–18, 137
 first updates, 13
 Ubuntu Announcements mailing list, 41
 UPG (user private group), 245
 uploading
 GPG key, 696–697
 SSH key, 697

- Upstart, 311–312, 314, 319–320
- uptime command, 328
- uquick command, 691
- URIs (uniform resource identifiers), 485–486
- USB
 - drives, installing Ubuntu from, 6
 - printers, 487
- used memory, displaying, 327–328
- Usenet newsgroups, 72–74
- User directive, 494
- user IDs (UIDs), 244
- user private group (UPG), 245
- user variables, 281
- useradd command, 247–248, 250–251
- UserDir directive, 495
- usermod command, 249
- usernames, 251
- users
 - adding, 250–251
 - creating, 181–182
 - deleting, 182
 - disk quotas, 262–263
 - FTP (File Transfer Protocol) software, 534–536
 - groups
 - group listings, 245–246
 - management tools, 246–248
 - overview, 245
 - management tools, 248–250
 - monitoring, 251–252
 - online resources, 264
 - passwords. See passwords
 - PostgreSQL
 - creating, 593–594
 - deleting, 594
 - privileges, 594–595
 - related Ubuntu commands, 264
- super users/root users
 - creating users, 181–182
 - deleting users, 182
 - overview, 178
 - sudo command, 178–181
- system administrator privileges, granting
 - overview, 257
 - with su command, 257–259
 - with sudo command, 259–262
- user accounts
 - command line, 154–155
 - file permissions, 244
 - GIDs (group IDs), 244
 - overview, 241–242
 - super users/root users, 242–244
 - UIDs (user IDs), 244
 - user stereotypes, 245
 - usernames, 251
- /usr directory. See directories, /usr
- UTP (unshielded twisted-pair), 383, 384–385
- uuencode program, 73

V

- Vala, 720–721
- valac package, 721
- Valve Software Steam platform, 123
- vanilla kernel, 454
- /var directory, 162
- variables
 - environment variables, 222–226
 - shell scripts
 - accessing, 282
 - assigning values to, 282
 - built-in variables, 286–287
 - positional parameters, 282–284

/var/log files, 330
 VBA (Visual Basic for Applications), 75
 version control systems
 Bazaar, 683–684
 for configuration files, 364–366
 Git, 682–683
 Mercurial, 685–686
 overview, 681–682
 Subversion, 684–685
 version numbers
 Linux, 30
 Ubuntu, 33
 versions (kernel), 461–462
 Very Secure FTP server, 532, 536–538
 vi command, 227–228
 video
 editing, 107–109
 personal video recorders, 107
 proprietary video drivers, installing, 122–123
 TV and video hardware, 104–105
 video formats, 105–106
 video viewers, 106
 virt-clone command, 636–637
 virtinst package, 634
 virt-install command, 636
 virt-manager package, 634
 virtual file systems, 160
 virtual hosting
 Apache
 address-based virtual hosts, 507–508
 name-based virtual hosts, 508–509
 overview, 507
 Nginx, 521–522
 virtual kernel, 461
 virtual machines (VMs), 632, 642
 virtual memory statistics (vmstat), 328
 Virtual Network Computing (VNC), 420–423
 virtual private networks. See VPNs (virtual private networks)
 virtual resolution, 51
 VirtualBox, 637–638
 virtualization
 KVM (Kernel-based Virtual Machine), 633–637
 online resources, 639
 overview, 631–633
 VirtualBox, 637–638
 VMware, 639
 Xen, 639
 virt-viewer package, 634
 virus scanners, 432, 556
 viruses, 431–432, 556
 Visual Basic for Applications (VBA), 75
 VLC, 106–107
 vmbuilder, 635
 VMs (virtual machines), 632, 642
 vmstat (virtual memory statistics), 328
 VMware, 639
 VNC (Virtual Network Computing), 420–423
 vncviewer, 339
 volume adjustment, 87
 VPNs (virtual private networks)
 client setup, 571–573
 overview, 570–571, 634
 server setup, 573–575
 vsftpd (Very Secure FTP) server, 532, 536–538
 vsftpd.banned_emails file, 537
 vsftpd.chroot_list file, 537
 vsftpd.conf, 536–537
 vsftpd.log file, 537
 vsftpd.user_list file, 537
 VT-x, 633
 vulnerability assessment, 427–428

W

waking computer, 270–272
 WAP (wireless access point), 404–405
 Warsow, 124
 Warty Warthog, 33
 watching video
 personal video recorders, 107
 TV and video hardware, 104–105
 video formats, 105–106
 video viewers, 106
 .wav filename extension, 88
 WAV format, 88
 web access to databases, 597
 web browsers
 Firefox, 63–65
 Google Chrome, 65–66
 Google Chromium, 65–66
 web search tips, 37–38
 web servers
 Apache. See Apache web server
 Apache Tomcat, 530
 Cherokee, 528–529
 Jetty, 529
 lighttpd, 527–528
 Nginx. See Nginx
 online resources, 530
 thttpd, 529–530
 Yaws, 528
 web-based email applications, 68
 websites. See online resources
 Welling, Luke, 591
 wget command, 207–208
 Whaley, Ben, 40
 what you see is what you get
 (WYSIWYG), 82
 what you see is what you want
 (WYGIWYW), 82

whatis command, 203
 what-provides command, 691
 whereis command, 157, 225
 which command, 207
 while loops, 300–302
 wide column stores
 BigTable, 611
 HBase, 611
 overview, 611
 widgets
 definition of, 112
 GTK (GIMP Tool Kit) widget set, 79
 wifi-status command, 692, 691
 wildcards, 177
 window managers, 52, 112
 Windows games, playing, 130
 Wine, 83
 wireless access point (WAP), 404–405
 wireless network interfaces, 384
 wireless networks
 advantages of, 407
 configuration, 24–25
 overview, 405
 security, 429
 support for, 405–406
 wireless protocols, 407–408
 wireless protocols, 407–408
 wireshark, 339
 wish command, 281
 worms, 426
 Writer, 76
 writing
 packet writing, 103–104
 session writing, 103
 shell scripts, 277–278
 wu-ftp, 533
 WWW Security FAQ, 498

WYGIWYW (What You Get Is What You Want), 82

WYSIWYG (what you see is what you get), 82

X

X

display managers, 45, 51–52

online resources, 61

overview, 43–44

starting, 51

window managers, 52

X protocol, 44–45

X.Org

directories, 45–46

xorg.conf file, 46–51

X Window System. See X

X11. See X

X11R7. See X

Xamarin, 717

Xara Xtreme, 98

XChat, 70–72

Xen, 639

Xfce and Xubuntu, 114–115

XFree86, 43–44

Xmarks Sync, 65

XML (Extensible Markup Language), 81

XML Copy Editor, 82

X.Org

history of, 43–44

xorg.conf file

Device section, 49–50

Files section, 47–48

InputDevice section, 48–49

Module section, 48

Monitor section, 49

overview, 46–47

Screen section, 50–51

ServerLayout section, 47

xorg.conf file

Device section, 49–50

Files section, 47–48

InputDevice section, 48–49

Module section, 48

Monitor section, 49

overview, 46–47

Screen section, 50–51

ServerLayout section, 47

Xubuntu, Xfce and, 114–115

#xubuntu channel, 42

xubuntu-desktop package, 115

Y

Yaws, 528

Yet Another Web Server (Yaws), 528

Z

Zenoss, 340

zless command, 36