

Matthew Helmke

Ubuntu
14.10
on DVD

Ubuntu

UNLEASHED

SAMS

2015 Edition
Covering 14.10 and 15.04

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Matthew Helmke
with Andrew Hudson
and Paul Hudson

Ubuntu

UNLEASHED

2015 Edition

SAMS

800 East 96th Street, Indianapolis, Indiana 46240 USA

Ubuntu Unleashed 2015 Edition

Copyright © 2015 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-672-33837-3

ISBN-10: 0-672-33837-8

Library of Congress Control Number: 2014946899

Printed in the United States of America

First Printing November 2014

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or programs accompanying it.

Special Sales

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 international@pearsoned.com.

Editor-in-Chief

Mark Taub

Acquisitions Editor

Debra Williams
Cauley

Managing Editor

Kristy Hart

Project Editor

Andy Beaster

Indexer

Cheryl Lenser

Proofreader

Sarah Kearns

Technical Editors

Micah Brown
Elizabeth Krumbach

Editorial Assistant

Kim Boedigheimer

Media Producer

Dan Scherf

Cover Designer

Mark Shirar

Compositor

Nonie Ratcliff

Contents at a Glance

Introduction	1
Part I Getting Started	
1 Installing Ubuntu and Post-Installation Configuration	9
2 Background Information and Resources	35
Part II Desktop Ubuntu	
3 Working with Unity	49
4 On the Internet	69
5 Productivity Applications	83
6 Multimedia Applications	93
7 Other Ubuntu Interfaces	119
8 Games	127
Part III System Administration	
9 Managing Software	139
10 Command-Line Quickstart	153
11 Command-Line Master Class	187
12 Managing Users	239
13 Automating Tasks and Shell Scripting	267
14 The Boot Process	313
15 System-Monitoring Tools	323
16 Backing Up	345
17 Networking	373
18 Remote Access with SSH, Telnet, and VNC	419
19 Securing Your Machines	429
20 Performance Tuning	445
21 Kernel and Module Management	457
Part IV Ubuntu as a Server	
22 Sharing Files and Printers	479
23 Apache Web Server Management	495
24 Nginx Web Server Management	527
25 Other Http Servers	539

26	Remote File Serving with FTP	543
27	Handling Email	553
28	Proxying, Reverse Proxying, and Virtual Private Networks (VPN).....	573
29	Administering Relational Database Services	589
30	NoSQL Databases	615
31	Lightweight Directory Access Protocol (LDAP)	627
32	Linux Terminal Server Project (LTSP)	635
33	Virtualization on Ubuntu	643
34	Ubuntu in the Cloud	653
35	Managing Sets of Servers	669
36	Name Serving with the Domain Name System (DNS)	673
Part V	Programming Linux	
37	Using Programming Tools for Ubuntu	683
38	Opportunistic Development	695
39	Helping with Ubuntu Development	711
40	Helping with Ubuntu Testing and QA	721
41	Using Perl	727
42	Using Python	749
43	Using PHP	769
44	Using Other Popular Programming Languages	801
45	Beginning Mobile Development for Android	815
46	Developing for Ubuntu Mobile/Touch	823
	Index	829

Table of Contents

Introduction	1
Licensing	2
Who This Book Is For	3
Those Wanting to Become Intermediate or Advanced Users	3
Sysadmins, Programmers, and DevOps	4
What This Book Contains	5
Conventions Used in This Book	6
Part I Getting Started	
1 Installing Ubuntu and Post-Installation Configuration	9
Before You Begin the Installation	9
Researching Your Hardware Specifications	10
Installation Options	10
32-Bit vs. 64-Bit Ubuntu	12
Planning Partition Strategies	13
The Boot Loader	13
Installing from DVD or USB Drive	14
Step-by-Step Installation	14
Installing	15
First Update	20
Shutting Down	20
Finding Programs and Files	21
Software Updater	22
The <code>sudo</code> Command	25
Configuring Software Repositories	26
System Settings	28
Detecting and Configuring a Printer	29
Configuring Power Management in Ubuntu	29
Setting the Time and Date	30
Configuring Wireless Networks	32
Troubleshooting Post-Installation Configuration Problems	33
References	34

2	Background Information and Resources	35
	What Is Linux?	35
	Why Use Linux?	37
	What Is Ubuntu?	39
	Ubuntu for Business	40
	Ubuntu in Your Home	41
	Getting the Most from Ubuntu and Linux Documentation	41
	Ubuntu Developers and Documentation	43
	Websites and Search Engines	43
	Web Search Tips	43
	Google Is Your Friend	44
	Ubuntu Package Listings	44
	Commercial Support	44
	Documentation	45
	Linux Guides	45
	Ubuntu	46
	Mailing Lists	46
	Ubuntu Project Mailing Lists	47
	Internet Relay Chat	48
Part II	Desktop Ubuntu	
3	Working with Unity	49
	Foundations and the X Server	49
	Basic X Concepts	50
	Using X	51
	Elements of the <code>xorg.conf</code> File	52
	Starting X	57
	Using a Display Manager	58
	Changing Window Managers	58
	Using Unity, a Primer	59
	The Desktop	59
	Customizing and Configuring Unity	64
	Power Shortcuts	66
	References	67
4	On the Internet	69
	Getting Started with Firefox	70
	Checking Out Google Chrome and Chromium	71

Choosing an Email Client	73
Mozilla Thunderbird	73
Evolution	74
Other Mail Clients	75
RSS Readers	76
Firefox	76
Liferea	76
Instant Messaging and Video Conferencing with Empathy	77
Internet Relay Chat	78
Usenet Newsgroups	80
References	82
5 Productivity Applications	83
Introducing LibreOffice	85
Other Office Suites for Ubuntu	87
Working with GNOME Office	87
Working with KOffice	88
Other Useful Productivity Software	89
Working with PDF	89
Working with XML and DocBook	89
Working with LaTeX	91
Productivity Applications Written for Microsoft Windows	91
References	92
6 Multimedia Applications	93
Sound and Music	93
Sound Cards	94
Adjusting Volume	95
Sound Formats	96
Listening to Music	97
Buying Music in the Ubuntu One Music Store	99
Graphics Manipulation	100
The GNU Image Manipulation Program	101
Using Scanners in Ubuntu	103
Working with Graphics Formats	103
Capturing Screen Images	105
Other Graphics Manipulation Options	106
Using Digital Cameras with Ubuntu	106
Handheld Digital Cameras	106
Using Shotwell Photo Manager	107

Burning CDs and DVDs in Ubuntu	107
Creating CDs and DVDs with Brasero	108
Creating CDs from the Command Line	109
Creating DVDs from the Command Line	110
Viewing Video	112
TV and Video Hardware	112
Video Formats	114
Viewing Video in Linux	114
Personal Video Recorders	116
Video Editing	116
References	117
7 Other Ubuntu Interfaces	119
Desktop Environment	120
KDE and Kubuntu	121
Xfce and Xubuntu	122
LXDE and Lubuntu	123
GNOME 3 and Ubuntu GNOME	124
Ubuntu Kylin	125
References	126
8 Games	127
Ubuntu Gaming	127
Installing Proprietary Video Drivers	128
Installing Games in Ubuntu	129
Warsow	129
Scorched 3D	130
Frozen Bubble	131
SuperTux	131
Battle for Wesnoth	133
Frets on Fire	133
FlightGear	134
Speed Dreams	134
Games for Kids	134
Commercial Games	135
Steam	136
Playing Windows Games	136
References	137

Part III System Administration

9	Managing Software	139
	Ubuntu Software Center	139
	Using Synaptic for Software Management	140
	Staying Up-to-Date	142
	Working on the Command Line	143
	Day-to-Day Usage	144
	Finding Software	147
	Compiling Software from Source	148
	Compiling from a Tarball	148
	Compiling from Source from the Ubuntu Repositories	149
	Configuration Management	150
	dotdee	150
	OneConf	151
	References	151
10	Command-Line Quickstart	153
	What Is the Command Line?	154
	Accessing the Command Line	155
	Text-Based Console Login	156
	Logging Out	157
	Logging In and Out from a Remote Computer	157
	User Accounts	158
	Reading Documentation	160
	Using Man Pages	160
	Using apropos	160
	Using whereis	161
	Understanding the Linux File System Hierarchy	161
	Essential Commands in /bin and /sbin	162
	Configuration Files in /etc	163
	User Directories: /home	163
	Using the Contents of the /proc Directory to Interact with the Kernel	164
	Working with Shared Data in the /usr Directory	165
	Temporary File Storage in the /tmp Directory	166
	Accessing Variable Data Files in the /var Directory	166
	Navigating the Linux File System	166
	Listing the Contents of a Directory with ls	166
	Changing Directories with cd	168
	Finding Your Current Directory with pwd	169

Working with Permissions	169
Assigning Permissions	170
Directory Permissions	171
Altering File Permissions with <code>chmod</code>	172
File Permissions with <code>chgrp</code>	173
Changing File Permissions with <code>chown</code>	173
Understanding Set User ID and Set Group ID Permissions	173
Working with Files	175
Creating a File with <code>touch</code>	175
Creating a Directory with <code>mkdir</code>	175
Deleting a Directory with <code>rmdir</code>	176
Deleting a File or Directory with <code>rm</code>	177
Moving or Renaming a File with <code>mv</code>	177
Copying a File with <code>cp</code>	178
Displaying the Contents of a File with <code>cat</code>	179
Displaying the Contents of a File with <code>less</code>	179
Using Wildcards and Regular Expressions	179
Working as Root	180
Understanding and Fixing <code>sudo</code>	180
Creating Users	183
Deleting Users	184
Shutting Down the System	184
Rebooting the System	185
Commonly Used Commands and Programs	185
References	186
11 Command-Line Master Class	187
Why Use the Command Line?	188
Using Basic Commands	189
Printing the Contents of a File with <code>cat</code>	191
Changing Directories with <code>cd</code>	191
Changing File Access Permissions with <code>chmod</code>	193
Copying Files with <code>cp</code>	194
Printing Disk Usage with <code>du</code>	194
Finding Files by Searching with <code>find</code>	195
Searches for a String in Input with <code>grep</code>	197
Paging Through Output with <code>less</code>	198
Creating Links Between Files with <code>ln</code>	200
Finding Files from an Index with <code>locate</code>	202
Listing Files in the Current Directory with <code>ls</code>	202
Reading Manual Pages with <code>man</code>	204
Making Directories with <code>mkdir</code>	205

Moving Files with <code>mv</code>	205
Deleting Files and Directories with <code>rm</code>	206
Sorting the Contents of a File with <code>sort</code>	206
Printing the Last Lines of a File with <code>tail</code>	208
Using <code>echo</code>	209
Printing the Location of a Command with <code>which</code>	209
Redirecting Output and Input	209
<code>stdin</code> , <code>stdout</code> , <code>stderr</code> , and Redirection	211
Comparing Files	212
Finding Differences in Files with <code>diff</code>	212
Finding Similarities in Files with <code>comm</code>	213
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>	216
Printing Resource Usage with <code>top</code>	216
Setting Processes Priority with <code>nice</code>	219
Combining Commands	220
Pipes	220
Combining Commands with Boolean Operators	222
Running Separate Commands in Sequence	222
Using Environment Variables	222
Using Common Text Editors	226
Working with <code>nano</code>	227
Working with <code>vi</code>	228
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 <code>REISUB</code>	234
Tips and Tricks	235
Running the Previous Command	235
Running Any Previous Command	236
Running a Previous Command That Started with Specific Letters	236
Running the Same Thing You Just Ran with a Different First Word	236
Viewing Your History and More	236
Do Two or More Things	236
Shortcuts	237
<code>Coreutils</code>	237
References	237

12	Managing Users	239
	User Accounts	239
	The Super User/Root User	240
	User IDs and Group IDs	242
	File Permissions	242
	Managing Groups	243
	Group Listing	243
	Group Management Tools	245
	Managing Users	246
	User Management Tools	246
	Adding New Users	248
	Monitoring User Activity on the System	252
	Managing Passwords	253
	System Password Policy	253
	The Password File	253
	Shadow Passwords	254
	Managing Password Security for Users	257
	Changing Passwords in a Batch	257
	Granting System Administrator Privileges to	
	Regular Users	258
	Temporarily Changing User Identity with the <code>su</code> Command	258
	Granting Root Privileges on Occasion: The <code>sudo</code> Command	260
	Disk Quotas	263
	Implementing Quotas	263
	Manually Configuring Quotas	264
	Related Ubuntu Commands	265
	References	265
13	Automating Tasks and Shell Scripting	267
	Scheduling Tasks	267
	Using <code>at</code> and <code>batch</code> to Schedule Tasks for Later	267
	Using <code>cron</code> to Run Jobs Repeatedly	270
	Using <code>rtcwake</code> to Wake Your Computer from Sleep	
	Automatically	272
	Basic Shell Control	274
	The Shell Command Line	275
	Shell Pattern-Matching Support	276
	Redirecting Input and Output	277
	Piping Data	278
	Background Processing	278

Writing and Executing a Shell Script	279
Running the New Shell Program	280
Storing Shell Scripts for System-Wide Access	281
Interpreting Shell Scripts Through Specific Shells	282
Using Variables in Shell Scripts	283
Assigning a Value to a Variable	284
Accessing Variable Values	284
Positional Parameters	284
A Simple Example of a Positional Parameter	285
Using Positional Parameters to Access and Retrieve Variables from the Command Line	286
Using a Simple Script to Automate Tasks	286
Built-In Variables	288
Special Characters	289
Using Double Quotes to Resolve Variables in Strings with Embedded Spaces	290
Using Single Quotes to Maintain Unexpanded Variables	290
Using the Backslash as an Escape Character	291
Using the Backtick to Replace a String with Output	292
Comparison of Expressions in <code>pdksh</code> and <code>bash</code>	292
Comparing Expressions with <code>tcsh</code>	297
The <code>for</code> Statement	301
The <code>while</code> Statement	303
The <code>until</code> Statement	304
The <code>repeat</code> Statement (<code>tcsh</code>)	305
The <code>select</code> Statement (<code>pdksh</code>)	305
The <code>shift</code> Statement	306
The <code>if</code> Statement	306
The <code>case</code> Statement	308
The <code>break</code> and <code>exit</code> Statements	310
Using Functions in Shell Scripts	310
References	311
14 The Boot Process	313
Running Services at Boot	313
Beginning the Boot Loading Process	314
Loading the Linux Kernel	315
System Services and Runlevels	316
Runlevel Definitions	316
Booting into the Default Runlevel	317
Understanding <code>init</code> Scripts and the Final Stage of Initialization	317

Controlling Services at Boot with Administrative Tools	318
Changing Runlevels	318
Troubleshooting Runlevel Problems	319
Starting and Stopping Services Manually	320
Using Upstart	321
systemd	322
Boot Repair	322
References	322
15 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	336
System Monitor	336
Conky	338
Other	342
KDE Process- and System-Monitoring Tools	343
Enterprise Server Monitoring	343
Landscape	343
Other	343
References	344
16 Backing Up	345
Choosing a Backup Strategy	345
Why Data Loss Occurs	346
Assessing Your Backup Needs and Resources	347
Evaluating Backup Strategies	349
Making the Choice	352
Choosing Backup Hardware and Media	352
Removable Storage Media	352
CD-RW and DVD+RW/-RW Drives	353
Network Storage	353
Tape Drive Backup	353
Cloud Storage	354

Using Backup Software	354
tar: The Most Basic Backup Tool	355
The GNOME File Roller	357
The KDE ark Archiving Tool	358
Déjà Dup	358
Back In Time	360
Unison	362
Using the Amanda Backup Application	362
Alternative Backup Software	363
Copying Files	364
Copying Files Using tar	364
Compressing, Encrypting, and Sending tar Streams	365
Copying Files Using cp	365
Copying Files Using mc	366
Using rsync	366
Version Control for Configuration Files	368
System Rescue	370
The Ubuntu Rescue Disc	371
Restoring the GRUB2 Boot Loader	371
Saving Files from a Nonbooting Hard Drive	372
References	372
17 Networking	373
Laying the Foundation: The localhost Interface	374
Checking for the Availability of the Loopback Interface	374
Configuring the Loopback Interface Manually	374
Checking Connections with ping, traceroute, and mtr	376
Networking with TCP/IP	378
TCP/IP Addressing	379
Using IP Masquerading in Ubuntu	381
Ports	382
IPv6 Basics	382
Network Organization	385
Subnetting	385
Subnet Masks	386
Broadcast, Unicast, and Multicast Addressing	386
Hardware Devices for Networking	387
Network Interface Cards	387
Network Cable	389
Hubs and Switches	390
Routers and Bridges	391
Initializing New Network Hardware	391

Using Network Configuration Tools	393
Command-Line Network Interface Configuration	394
Network Configuration Files	399
Using Graphical Configuration Tools	401
Dynamic Host Configuration Protocol	403
How DHCP Works	403
Activating DHCP at Installation and Boot Time	404
DHCP Software Installation and Configuration	405
Using DHCP to Configure Network Hosts	407
Other Uses for DHCP	409
Wireless Networking	409
Support for Wireless Networking in Ubuntu	409
Advantages of Wireless Networking	411
Choosing from Among Available Wireless Protocols	411
Beyond the Network and onto the Internet	412
Common Configuration Information	412
Configuring Digital Subscriber Line Access	414
Understanding PPP over Ethernet	414
Configuring a PPPoE Connection Manually	415
Configuring Dial-Up Internet Access	416
Troubleshooting Connection Problems	417
References	418
18 Remote Access with SSH, Telnet, and VNC	419
Setting Up a Telnet Server	419
Telnet Versus SSH	421
Setting Up an SSH Server	421
SSH Tools	421
Using <code>scp</code> to Copy Individual Files Between Machines	422
Using <code>sftp</code> to Copy Many Files Between Machines	423
Using <code>ssh-keygen</code> to Enable Key-Based Logins	423
Virtual Network Computing	425
References	427
19 Securing Your Machines	429
Understanding Computer Attacks	429
Assessing Your Vulnerability	431
Protecting Your Machine	432
Securing a Wireless Network	433
Passwords and Physical Security	433
Configuring and Using Tripwire	434
Devices	435

Viruses	435
Configuring Your Firewall	436
AppArmor	440
Forming a Disaster Recovery Plan	442
References	443
20 Performance Tuning	445
Hard Disk	445
Using the BIOS and Kernel to Tune the Disk Drives	446
The <code>hdparm</code> Command	447
File System Tuning	448
The <code>tune2fs</code> Command	448
The <code>e2fsck</code> Command	449
The <code>badblocks</code> Command	449
Disabling File Access Time	449
Kernel	450
Apache	451
MySQL	452
Measuring Key Buffer Usage	452
Using the Query Cache	454
Miscellaneous Tweaks	455
Query Optimization	456
References	456
21 Kernel and Module Management	457
The Linux Kernel	458
The Linux Source Tree	459
Types of Kernels	461
Managing Modules	462
When to Recompile	464
Kernel Versions	465
Obtaining the Kernel Sources	466
Patching the Kernel	467
Compiling the Kernel	468
Using <code>xconfig</code> to Configure the Kernel	471
Creating an Initial RAM Disk Image	474
When Something Goes Wrong	475
Errors During Compile	475
Runtime Errors, Boot Loader Problems, and Kernel Oops	476
References	476

Part IV Ubuntu as a Server

22	Sharing Files and Printers	479
	Using the Network File System	480
	Installing and Starting or Stopping NFS	480
	NFS Server Configuration	480
	NFS Client Configuration	482
	Putting Samba to Work	482
	Manually Configuring Samba with <code>/etc/samba/smb.conf</code>	484
	Testing Samba with the <code>testparm</code> Command	487
	Starting, Stopping, and Restarting the <code>smbd</code> Daemon	488
	Mounting Samba Shares	489
	Network and Remote Printing with Ubuntu	489
	Creating Network Printers	490
	Using the Common UNIX Printing System GUI	491
	Avoiding Printer Support Problems	493
	References	494
23	Apache Web Server Management	495
	About the Apache Web Server	495
	Installing the Apache Server	496
	Installing from the Ubuntu Repositories	496
	Building the Source Yourself	498
	Starting and Stopping Apache	500
	Starting the Apache Server Manually	500
	Using <code>/etc/init.d/apache2</code>	502
	Runtime Server Configuration Settings	503
	Runtime Configuration Directives	503
	Editing <code>apache2.conf</code>	504
	Apache Multiprocessing Modules	507
	Using <code>.htaccess</code> Configuration Files	507
	File System Authentication and Access Control	509
	Restricting Access with <code>allow</code> and <code>deny</code>	510
	Authentication	511
	Final Words on Access Control	513
	Apache Modules	514
	<code>mod_access</code>	514
	<code>mod_alias</code>	514
	<code>mod_asis</code>	515
	<code>mod_auth</code>	515
	<code>mod_auth_anon</code>	515
	<code>mod_auth_dbm</code>	516

mod_auth_digest	516
mod_autoindex	516
mod_cgi	516
mod_dir and mod_env	516
mod_expires	516
mod_headers	516
mod_include	517
mod_info and mod_log_config	517
mod_mime and mod_mime_magic	517
mod_negotiation	517
mod_proxy	517
mod_rewrite	518
mod_setenvif	518
mod_speling	518
mod_status	518
mod_ssl	518
mod_unique_id	518
mod_userdir	519
mod_usertrack	519
mod_vhost_alias	519
Virtual Hosting	519
Address-Based Virtual Hosts	519
Name-Based Virtual Hosts	520
Logging	521
HTTPS	523
References	525
24 Nginx Web Server Management	527
About the Nginx Web Server	527
Installing the Nginx Server	529
Installing from the Ubuntu Repositories	529
Building the Source Yourself	530
Configuring the Nginx Server	530
Virtual Hosting	533
Setting Up PHP	534
Adding and Configuring Modules	536
HTTPS	536
References	538

25	Other HTTP Servers	539
	lighttpd	539
	Yaws	540
	Cherokee	541
	Jetty	541
	thttpd	542
	Apache Tomcat	542
	References	542
26	Remote File Serving with FTP	543
	Choosing an FTP Server	543
	Choosing an Authenticated or Anonymous Server	544
	Ubuntu FTP Server Packages	544
	Other FTP Servers	544
	Installing FTP Software	545
	The FTP User	546
	Configuring the Very Secure FTP Server	548
	Controlling Anonymous Access	549
	Other vsftpd Server Configuration Files	550
	Using the ftpchosts File to Allow or Deny FTP Server Connection	551
	References	552
27	Handling Email	553
	How Email Is Sent and Received	553
	The Mail Transport Agent	554
	Choosing an MTA	556
	The Mail Delivery Agent	556
	The Mail User Agent	557
	Basic Postfix Configuration and Operation	558
	Configuring Masquerading	560
	Using Smart Hosts	561
	Setting Message Delivery Intervals	561
	Mail Relaying	562
	Forwarding Email with Aliases	562
	Using Fetchmail to Retrieve Mail	563
	Installing Fetchmail	563
	Configuring Fetchmail	563
	Choosing a Mail Delivery Agent	567
	Procmail	567
	Spamassassin	567

Squirrelmail	568
Virus Scanners	568
Autoresponders	568
Alternatives to Microsoft Exchange Server	568
Microsoft Exchange Server/Outlook Client	569
CommuniGate Pro	569
Oracle Beehive	570
Bynari	570
Open-Xchange	570
phpgroupware	570
PHProjekt	570
Horde	570
References	571
28 Proxying, Reverse Proxying, and Virtual Private Networks (VPN)	573
What Is a Proxy Server?	573
Installing Squid	574
Configuring Clients	574
Access Control Lists	575
Specifying Client IP Addresses	579
Sample Configurations	580
Virtual Private Networks (VPN)	582
Setting Up a VPN Client	583
Setting Up a VPN Server	585
References	587
29 Administering Relational Database Services	589
A Brief Review of Database Basics	590
How Relational Databases Work	592
Understanding SQL Basics	594
Creating Tables	594
Inserting Data into Tables	595
Retrieving Data from a Database	596
Choosing a Database: MySQL Versus PostgreSQL	598
Speed	598
Data Locking	599
ACID Compliance in Transaction Processing to Protect Data Integrity	599
SQL Subqueries	600
Procedural Languages and Triggers	600

Configuring MySQL.....	601
Setting a Password for the MySQL Root User.....	602
Creating a Database in MySQL.....	602
Configuring PostgreSQL.....	604
Initializing the Data Directory in PostgreSQL.....	604
Creating a Database in PostgreSQL.....	605
Creating Database Users in PostgreSQL.....	606
Deleting Database Users in PostgreSQL.....	606
Granting and Revoking Privileges in PostgreSQL.....	607
Database Clients.....	607
SSH Access to a Database.....	608
Local GUI Client Access to a Database.....	609
Web Access to a Database.....	610
The MySQL Command-Line Client.....	611
The PostgreSQL Command-Line Client.....	612
Graphical Clients.....	613
References.....	613
30 NoSQL Databases.....	615
Key/Value Stores.....	618
Berkeley DB.....	618
Cassandra.....	619
Memcached and MemcacheDB.....	619
Redis.....	620
Riak.....	620
Document Stores.....	620
CouchDB.....	621
MongoDB.....	622
BaseX.....	622
Wide Column Stores.....	623
BigTable.....	623
HBase.....	623
Graph Stores.....	624
Neo4j.....	624
OrientDB.....	624
HyperGraphDB.....	624
FlockDB.....	625
References.....	625

31	Lightweight Directory Access Protocol (LDAP)	627
	Configuring the Server	628
	Creating Your Schema	628
	Populating Your Directory	630
	Configuring Clients	632
	Evolution	632
	Thunderbird	633
	Administration	633
	References	634
32	Linux Terminal Server Project (LTSP)	635
	Requirements	636
	Installation	639
	Using LTSP	640
	References	641
33	Virtualization on Ubuntu	643
	KVM	645
	VirtualBox	649
	VMware	651
	Xen	651
	References	651
34	Ubuntu in the Cloud	653
	Why a Cloud?	654
	Software as a Service (SaaS)	655
	Platform as a Service (PaaS)	655
	Infrastructure as a Service (IaaS)	655
	Metal as a Service (MaaS)	656
	Before You Do Anything	656
	Ubuntu Cloud and Eucalyptus	656
	Deploy/Install Basics: Public, Private, or Hybrid?	656
	Ubuntu Cloud and OpenStack	657
	Compute Infrastructure (Nova)	658
	Storage Infrastructure (Swift)	659
	Networking Service (Neutron)	659
	Identity Service (Keystone)	659
	Imaging Service (Glance)	659
	Dashboard (Horizon)	659
	Learning More	659

Juju	660
Getting Started	660
Charms	663
The Juju GUI	665
Juju Quickstart	665
Juju on Mac OS X and Windows	665
Ubuntu Metal as a Service (MaaS)	666
Landscape	666
References	666
35 Managing Sets of Servers	669
Juju	669
Puppet	670
Chef	670
CFEngine	671
Ansible	671
Landscape	671
References	671
36 Name Serving with the Domain Name System (DNS)	673
Understanding Domain Names	675
DNS Servers	675
DNS Records	676
Setting Up a DNS Server with BIND	679
References	681
Part V Programming Linux	
37 Using Programming Tools for Ubuntu	683
Programming with Linux	684
Using the C Programming Project Management Tools	
Provided with Ubuntu	685
Building Programs with <code>make</code>	685
Using Makefiles	685
Using the <code>autoconf</code> Utility to Configure Code	687
Debugging Tools	688
Using the GNU C Compiler	689
Graphical Development Tools	690
Using the KDevelop Client	690
The Glade Client for Developing in GNOME	691
Use an IDE or SDK	692
References	694

38	Opportunistic Development	695
	Version Control Systems	696
	Managing Software Projects with Subversion	696
	Managing Software Projects with Bazaar	697
	Managing Software Projects with Mercurial	698
	Managing Software Projects with Git	699
	Introduction to Opportunistic Development	700
	Launchpad	701
	Quickly	703
	Bikeshed and Other Tools	707
	References	709
39	Helping with Ubuntu Development	711
	Introduction to Ubuntu Development	712
	Setting Up Your Development System	713
	Install Basic Packages and Configure	713
	Create a Launchpad Account	714
	Set Up Your Environment to Work with Launchpad	714
	Fixing Bugs and Packaging	716
	Finding Bugs to Fix with Harvest	719
	Masters of the Universe	719
	References	719
40	Helping with Ubuntu Testing and QA	721
	Community Teams	721
	Ubuntu Testing Team	722
	QA Team	722
	Bug Squad	723
	Test Drive	723
	References	726
41	Using Perl	727
	Using Perl with Linux	727
	Perl Versions	728
	A Simple Perl Program	728
	Perl Variables and Data Structures	730
	Perl Variable Types	731
	Special Variables	731
	Operators	732
	Comparison Operators	732
	Compound Operators	733

Arithmetic Operators	733
Other Operators	734
Special String Constants	734
Conditional Statements: <code>if/else</code> and <code>unless</code>	735
<code>if</code>	735
<code>unless</code>	736
Looping	736
<code>for</code>	736
<code>foreach</code>	737
<code>while</code>	737
<code>until</code>	738
<code>last and next</code>	738
<code>do ... while</code> and <code>do ... until</code>	738
Regular Expressions	739
Access to the Shell	740
Modules and CPAN	741
Code Examples	741
Sending Mail	741
Purging Logs	743
Posting to Usenet	744
One-Liners	745
Command-Line Processing	746
References	746
42 Using Python	749
Python on Linux	750
The Basics of Python	751
Numbers	751
More on Strings	753
Lists	756
Dictionaries	758
Conditionals and Looping	759
Functions	761
Object Orientation	762
Class and Object Variables	763
Constructors and Destructors	764
Class Inheritance	765
The Standard Library and the Python Package Index	767
References	767

43	Using PHP	769
	Introduction to PHP	770
	Entering and Exiting PHP Mode	770
	Variables	770
	Arrays	772
	Constants	774
	References	774
	Comments	775
	Escape Sequences	775
	Variable Substitution	776
	Operators	777
	Conditional Statements	779
	Special Operators	780
	Switching	781
	Loops	783
	Including Other Files	785
	Basic Functions	786
	Strings	786
	Arrays	789
	Files	791
	Miscellaneous	793
	Handling HTML Forms	797
	Databases	797
	References	800
44	Using Other Popular Programming Languages	801
	Ada	802
	Clojure	803
	COBOL	803
	D	804
	Erlang	804
	Forth	805
	Go	805
	Fortran	806
	Groovy	806
	Dart	806
	Haskell	807
	Java	807
	JavaScript	808
	Lisp	808
	Lua	809

Mono	809
OCaml	810
Ruby	810
Rust	811
Scala	811
Scratch	811
Vala	811
References	812
45 Beginning Mobile Development for Android	815
Introduction to Android	816
Hardware	816
Linux Kernel	816
Libraries	816
Android Runtime	816
Application Framework	816
Applications	817
Installing the Android SDK	817
Install Java	817
Install Eclipse	817
Install the SDK	817
Install the ADT Eclipse Plug-In	818
Install Other Components	818
Install Virtual Devices	819
Create Your First Application	820
References	821
46 Developing for Ubuntu Mobile/Touch	823
Install the SDK	824
Create Your First Application	824
Learn About Ubuntu Design	825
Study the User Interface Toolkit	825
References	826
Index	829

About the Authors

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 senior technical writer for Pearson North America's Assessment Technology Engineering division, documenting assessment 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 also 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 my grandfather for always
believing in me and teaching me to believe in myself;
and to my friends in the Ubuntu, developer, sysadmin,
cloud computing, and DevOps communities.*

Acknowledgments

I am solely responsible for this edition of *Ubuntu Unleashed*, but I freely acknowledge that I am standing on the shoulders of giants. I want to express my gratitude to Andrew and Paul Hudson for the solid foundation that past editions of the book (up to *Ubuntu Unleashed, 2008 Edition*) provided to this update. Thanks to Ryan Troy for helping with the 2010 edition. Thank you to the many people who helped with technical edits and 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 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: errata@informit.com

Mail: Addison-Wesley/Prentice Hall Publishing
ATTN: Reader Feedback
330 Hudson Street
7th Floor
New York, New York, 10013

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

We are pleased to present the 2015 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 user base 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 technologically 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

IN THIS INTRODUCTION

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

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, “Getting Started,” 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, a bonus chapter will be available online at www.informit.com/title/9780672338373.

If you have the print copy of this book, follow the instructions on the inside back cover page to register your product, and you will receive an email notification when the bonus chapter is available.

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 boldface formatting. In addition, command lines that require root or super-user access are prefaced with the `sudo` command, as follows:

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

The following elements provide you with useful tidbits of information that relate to the discussion of the text:

NOTE

A note provides additional information you might find useful as you are working. Notes augment a discussion with ancillary details or point you to an article, a whitepaper, or another online reference for more information about a specific topic.

TIP

A tip contains a special insight or a timesaving technique, as well as information about items of particular interest to you that you might not find elsewhere.

CAUTION

A caution warns you about pitfalls or problems before you run a command, edit a configuration file, or choose a setting when administering your system.

SIDEBARS CAN BE GOLDMINES

Just because it is in a sidebar does not mean that you will not find something new here. Be sure to watch for these elements that bring in outside content that is an aside to the discussion in the text. You will read about other technologies, Linux-based hardware, and special procedures to make your system more robust and efficient.

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.

This page intentionally left blank

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 Center

The Ubuntu Software Center is a graphical utility for package management in Ubuntu. You can find it in the Applications menu as Ubuntu Software Center; the package and executable program is named `software-center`. The Ubuntu Software Center 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 Package Browsing screen, as shown in Figure 9.1.

Along the left side of the screen, you have three menu options: Get Software, Installed Software, and History. At the top is a search bar that you can use to search for packages. When you click the Get Software link, you are

IN THIS CHAPTER

- ▶ Ubuntu Software Center
- ▶ Using Synaptic for Software Management
- ▶ Staying Up-to-Date
- ▶ Working on the Command Line
- ▶ Compiling Software from SourceConfiguration Management
- ▶ References

presented with options to explore software Provided by Ubuntu or software For Purchase. Clicking the Installed Software link presents you with a list of all the installed applications on your Ubuntu desktop.

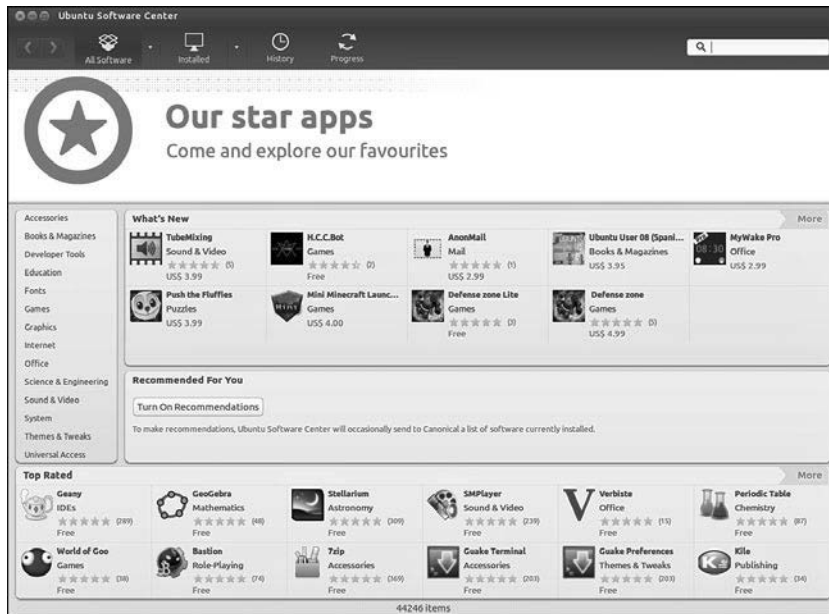


FIGURE 9.1 The initial Ubuntu Software Center screen enables you to browse through packages sorted by groups.

Installing new software via Ubuntu Software Center 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 Center and clicking the Remove button.

Use the Search box at the top to search for a specific application in the list. Note that this searches within the current category; so if you are in the Games category and search for “office,” you will get no results. The best place to search is within the Get Free Software category, to make sure you search all areas.

Using Synaptic for Software Management

The Add/Remove Applications dialog works just fine for adding applications, but if you need to install something specific—such as a library—or if you want to reconfigure your installation system, you need to use Synaptic (see Figure 9.2). You can install Synaptic using the Ubuntu Software Center described earlier; it is not installed by default.

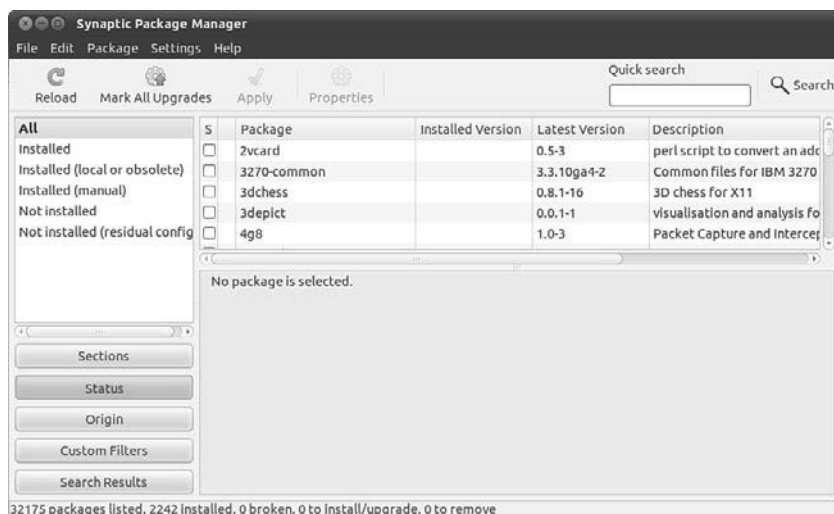


FIGURE 9.2 For more advanced software management in a GUI, Synaptic is the preferred tool.

At first glance, Synaptic looks a little like the Add/Remove Applications window. 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.
- ▶ **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 four buttons: Sections, Status, Search, and Custom, with Sections selected. These customize the left list: Sections is the Categories view; Status enables you to view packages that are installed or upgradable; Search stores results of your searches; and Custom has some esoteric groupings that are useful only to advanced users.

You can press Ctrl+F at any time to search for a particular package. By default, it is set to search by package name. You may change the Look In box setting to Description and 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 of 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 still 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.

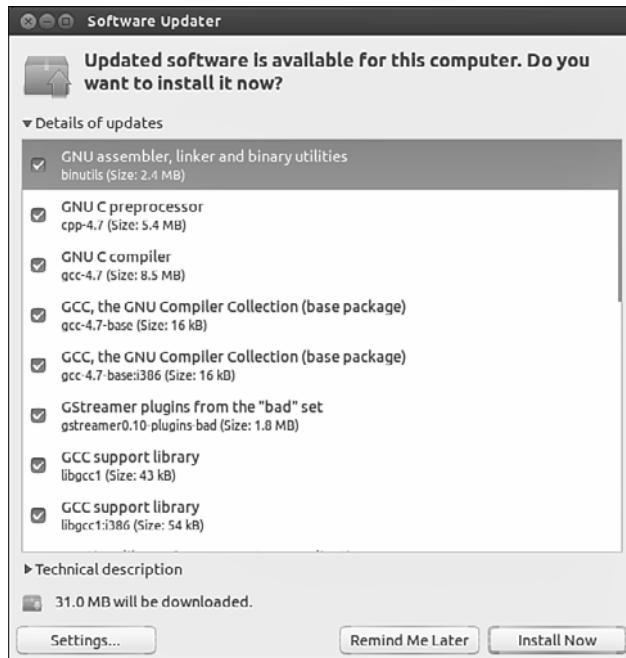


FIGURE 9.3 If you need to update your software to apply bug fixes and security upgrades, use Software Updater.

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 icedtea6-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
```

That performs 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.

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 package `build-essential` 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-4.5.2-1ubuntu1custom.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-4.5.2-1ubuntu1custom.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 35, “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.

OneConf

OneConf is a mechanism for recording software information in Ubuntu One and synchronizing with other computers as needed. Open the Ubuntu Software Center and select File, Sync Between Computers from the menu. You're prompted to create an Ubuntu Software Center account, if you have not already done so. Then, on any other Ubuntu computer you use, you can log in to the same account, and all the same applications will be installed, along with your copied and saved application data, to the new computer. No one else can see what you have installed or how it is configured.

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

Numbers

- a2p command, 746
- 10BASE-T, 387
- 32-bit Ubuntu, 64-bit Ubuntu versus, 12-13
- 64-bit Ubuntu, 32-bit Ubuntu versus, 12-13
- 100BASE-T, 387-388
- 1000BASE-T, 388

A

- A record (DNS), 676
- AAAA record (DNS), 676
- AbiWord (GNOME Office component), 88
- ac command, 252, 265
- accept command, 494
- access control, Apache web server, 509
- access control lists (ACLs), 575-579
- accessing
 - command line, 155-158
 - databases
 - local GUI clients, 609
 - SSH, 608-609
 - Perl shell, 740
- accounts, Launchpad, 714
- ACID compliance, MySQL versus PostgreSQL, 599-601
- ACLs (access control lists), 575-579
- activation, DHCP, 404-405
- Ada, 802
- Adblock Plus plug-in, 71
- adding, users, 248-252
- Additional Drivers manager, 129
- address-based virtual hosts (Apache), 519-520

addressing

- broadcast, 386
 - IPv4, 380-381
 - IPv6, 382-385
 - multicasts, 386
 - TCP/IP, 379-381
 - Unicast, 386
- adduser command, 183
- adjusting volume, music and sound, 95
- admin group, sudo group versus, 181, 262-263
- administration, LDAP, 633-634
- administrative tools (BIOS), 318
- Adobe Flash, 115
- Adobe Photoshop, 102
- ADT (Android Development Tools) plug-in, 817
- ADT Eclipse plug-in, 817
- Advanced Linux Sound Architecture (ALSA), 94
- Advanced Package Tool (APT), 143-148, 649
- day-to-day usage, 144-147
 - finding software, 147-148
- afio backup tool, 363
- aliases, forwarding email, 562-563
- Alien Arena, 129
- all-in-one devices (print/fax/scan), 493
- allow directive, Apache web server access control, 510-511
- AllowOverrides directive, Apache web server configuration, 509
- Alpine mail client, 75
- ALSA (Advanced Linux Sound Architecture), 94
- Amanda backup application, 362-363
- AMD, proprietary drivers, 128
- American Registry for Internet Numbers, 379
- Android Development Tools (ADT) plug-in, 817
- Android mobile development, 815-817
- Android Runtime, 816
 - Application Framework, 816-817
 - core applications, 817
 - creating application, 820-821
 - installation of Android SDK, 817-820
 - libraries, 816
 - Linux kernel, 816
- Android Runtime, mobile development for Android, 816
- Android SDK, installation, 817
- Android Virtual Device (AVD), 819
- anonymous access (FTP), 549
- anonymous servers (FTP), 544
- Ansible, managing sets of servers, 671
- Apache Module Registry, 514
- Apache Software Foundation, 496
- Apache Tomcat, 542
- Apache web server, 495-496, 545
- access control, 509
 - Apache package directories, 497
 - directives, CustomLog, 523
 - file system authentication, 509
 - installation, 496-499
 - building the source, 498
 - Ubuntu repositories, 496-498
 - logging, 521-523
 - modules, 514-519
 - disabling, 514
 - enabling, 514
 - performance tuning, 451-452
 - quick guide setup, 497
 - runtime server configuration settings, 498-499
 - .htaccess configuration file, 507-509
 - directives, 503
 - editing apache2.conf, 504-506
 - multiprocessing modules, 507
 - starting/stopping, 497
 - user files, 513
 - virtual hosting, 519-521
 - wide use, 495-496

- apache2.conf, editing, 504-506
- AppArmor, 440-442
- Application Framework, mobile development for Android, 816-817
- applications
 - creating for Ubuntu Mobile/Touch, 824-825
 - design guidelines, 825
 - Internet, 69-82
 - Chromium, 71-73
 - email clients, 73-76
 - Firefox, 70-71
 - Google Chrome, 71-73
 - mobile development for Android, 817
 - multimedia, 93-117
 - burning CDs/DVDs, 107-112
 - digital cameras, 106-107
 - graphics manipulation, 100-106
 - sound and music, 93-100
 - viewing video, 112-117
 - productivity
 - GNOME Office, 87-88
 - KOffice, 87-89
 - LaTeX, 91
 - LibreOffice, 85-86
 - Microsoft Windows, 91-92
 - PDF, 89
 - XML and DocBook, 89-90
 - User Interface Toolkit, 825-826
- apply-patch tool (Bikeshed), 707
- apropros command, 160-161
- APT (Advanced Package Tool), 143-148, 649
 - day-to-day usage, 144-147
 - finding software, 147-148
- apt-get autoclean command, 146
- apt-get clean command, 146
- apt-get dist-upgrade command, 145
- apt-get install command, 145
- apt-get moo command, 147
- apt-get remove command, 146
- apt-get upgrade command, 144
- arithmetic operators (Perl), 733-734
- ark archiving tool (KDE), 358
- ARM processors, 816
- ARPANET, 69
- array_keys() function (PHP), 790
- array_unique() function (PHP), 789
- array_values() function (PHP), 790
- arrays
 - Perl, 731
 - PHP functions, 789-791
 - PHP programming, 772-773
- arsort() function (PHP), 790
- The Art of Unix Programming* (Raymond), 187
- Artistic License, 3
- asort() function (PHP), 790
- assessment
 - backup needs, 347-348
 - resources for backup, 347-348
 - vulnerability, 431-432
- assigning permissions, 170-171
- at command, scheduling tasks, 267-270
- audio formats, 97
- authenticated servers (FTP), 544
- authentication, Apache web server, 511-513
- autoconf utility, 687-688
- autocracking scripts, 430
- automating tasks, 267-274
 - running jobs repeatedly, 270-272
 - scheduling tasks, 267-274
 - waking computer from sleep, 272-274
- automation of tasks, writing shell scripts, 286-288
- Autoresponders, 568

AVD (Android Virtual Device), 819

AVI (video format), 114

Avidemux, 117

awk text editor, 230-232

B

Back in Time backup tool, 360-361

background processing

command line, 215-216

shell control, 278-279

backing up data, 345-372

choosing a strategy, 345-352

copying files, 364-368

hardware and media, 352-354

software, 354-363

system rescue, 370-372

version control, 368-370

backslash, writing shell scripts, 291

backticks

accessing the shell in Perl, 740

writing shell scripts, 292

badblocks command, 449

Bandwidth Meter and Diagnostics, 71

Banshee, 99

Base (LibreOffice component), 85

BaseX, 622-623

Bash shell, configuration, 715-716

bash shell, comparison of expressions,
292-301

Basic Authentication, Apache web server,
511-513

basic input/output system. See BIOS (basic
input/output system)

batch command, 267-270

Battle for Wesnoth, 133

Bazaar, 697-698, 715

bch tool (Bikeshed), 707

beep codes, 314

Berkeley DB, 618-619

Berkeley Internet Name Domain (BIND),
679-681

Bernes-Lee, Tim, 69

bg command, 216

BigTable, 623

Bikeshed, 707-709

/bin directory, 162-163

BIND (Berkeley Internet Name Domain),
679-681

BIOS (basic input/output system), 313-320

booting into default runlevel, 317

controlling services, 318

final stage of initialization, 317-318

init scripts, 317-318

loading Linux Kernel, 315

runlevel definitions, 316

system services and runlevels, 316

troubleshooting runlevel problems, 319-320

tuning disk drives, 446-447

Bitbucket, 699

Blender, 106, 117

BMP (graphics format), 104

body of message (newsgroup articles), 81

Boolean operators, combining commands, 222

boot loader

defined, 313

problems when compiling kernel, 476

system rescue, 370-372

Ubuntu installation, 13

boot process, 313-322

BIOS, 313-320

dual boot, 14

manually starting/stopping services,
320-321

- running services, 313-314
- systemd, 322
- troubleshooting, 322
- Upstart, 321-322
- Boot Repair, 322
- booting, to recovery mode, 182
- booting into default runlevel (BIOS), 317
- bootmail tool, 709
- Brasero, 108
- break statement, 310
- bridged networking, 646-647
- bridges, 391
- broadcast addressing, 386
- brute-forcing, 423
- BSD License, 3
- buffer usage, MySQL, 452-454
- Bug Squad, 723
- bugs
 - fixing, 716-718
 - Harvest, 719
- built-in security, kernel, 429
- built-in variables
 - executing shell scripts, 288-289
 - shell scripts, 283
- BulletProofX, 52
- bunzip2 command, 232
- burning CDs/DVDs, 107-112
 - Brasero, 108
 - command line, 109-112
- business usage of Ubuntu, 40-41
- Bynari, 570
- Byobu, 78
- byobu command, 233-234
- bzip2 command, 232
- bzrp tool (Bikeshed), 707

C

- C programming tools, 683-694
 - autoconf utility, 687-688
 - debugging tools, 688-689
 - GNU C compiler, 689-690
 - graphical development, 690-693
 - Linux, 684-685
 - macros, 686-687
 - make command, 685
 - makefile targets, 686-687
 - makefiles, 685-687
- C++ programming tools, 683-694
 - autoconf utility, 687-688
 - debugging tools, 688-689
 - GNU C compiler, 689-690
 - graphical development, 690-693
 - Linux, 684-685
 - macros, 686-687
 - make command, 685
 - makefile targets, 686-687
 - makefiles, 685-687
- c10k problem, 528
- cable (network), 389-390
- Calc (LibreOffice component), 85
- cameras, digital, 106-107
- cancel command, 494
- captured screen images, 105
- case statement, 308-309
- Cassandra, 619
- cat command, 164, 179, 189, 191
- cd command, 168-169, 189, 191-193
- CD-RW drives, 353
- CDs, burning, 107-112
 - Brasero, 108
 - command line, 109-110
- Cedega, 136

- CFEngine, 671
- change command, 265
- changing
 - passwords in a batch, 257
 - runlevels (BIOS), 318-319
- characters
 - shell pattern matching, 276-277
 - writing shell scripts, 289
- charms, Juju, 663-665
- chat. See IRC (Internet Relay Chat)
- Checkbox, 723
- checking connections, networking, 376-378
- Chef, 670
- Cherokee, 541
- chfn command, 174, 265
- chgrp command, 173, 242, 265
- children's games, 134
- Childsplay, 134
- chmod command, 172-173, 189, 193, 242, 265
- chown command, 242, 265
- chpasswd command, 257, 265
- Chromium, 71-73
- chsh command, 248, 265
- Cinelerra, 117
- CinePaint, 106, 117
- class inheritance, Python object orientation, 765-766
- class variables, Python object orientation, 763-764
- classes of networks, 379-380
- Claws mail client, 75
- CLI (command-line interface). See command line
- client/server system, relational database services, 607-613
- clients
 - configuration, proxy servers, 574-575
 - DHCP, 405-406
 - graphical, relational database services, 613
 - IP addresses, 579-580
 - LDAP, configuration, 632-633
 - NFS, configuration, 482
- Clojure, 803
- Cloud, 653-667
 - benefits, 654-656
 - Eucalyptus, 656
 - deployment/installation, 656-657
 - laaS (Infrastructure as a Service), 655-656
 - Juju, 660-665
 - charms, 663-665
 - GUI, 665
 - installation, 660-663
 - on Mac OS X, 665
 - Landscape, 666
 - MaaS (Metal as a Service), 656, 666
 - OpenStack, 657-660
 - Compute Infrastructure (Nova), 658-659
 - Dashboard (Horizon), 659
 - Identity Service (Keystone), 659
 - Imaging Service (Glance), 659
 - Networking Service (Neutron), 659
 - Storage Infrastructure (Swift), 659
 - PaaS (Platform as a Service), 655
 - SaaS (Software as a Service), 655
- cloud storage, 354
- cloud-sandbox tool (Bikeshed), 707
- CNAME record (DNS), 677
- COBOL (Common Business Oriented Language), 803-804
- code
 - configuring autoconf utility, 687-688
 - packaging, 716-718
- code examples, Perl, 741-746
- code names, Ubuntu, 39
- codecs, 114

- CodeWeavers, 92
- col1 tool (Bikeshed), 707
- combining commands, 220-222
 - Boolean operators, 222
 - piping data, 220-222
 - running commands in order, 222
- comm command, 213
- command line, 153-238
 - accessing, 155-158
 - APT
 - day-to-day usage, 144-147
 - finding software, 147-148
 - burning CDs/DVDs, 109-112
 - commands, 185-186, 189-209
 - background processing, 215-216
 - cat, 191
 - cd, 168-169, 191-193
 - chmod, 193
 - combining, 220-222
 - common commands and programs, 185-186
 - Coreutils, 237
 - cp, 194
 - du, 194-195
 - echo, 209
 - find, 195-197
 - grep, 197-198
 - less, 198-200
 - ln, 200-202
 - locate, 202
 - ls, 166-168, 202-204
 - man, 204-205
 - mkdir, 205
 - mv, 205
 - pwd, 169
 - rm, 206
 - running in order, 222
 - running previous command, 235-236
 - separating multiple command, 236-237
 - shortcuts, 237
 - sort, 206-208
 - system reset, 234-235
 - tail, 208-209
 - viewing history, 236
 - which, 209
 - compressed files, 232-233
 - defined, 154-155
 - directories
 - changing, 168-169
 - listing contents, 166-168
 - pwd command, 169
 - environment variables, 222-226
 - errors in Perl, 729
 - files
 - comparing, 212-213
 - copying, 178
 - creating directories, 175-176
 - creating with touch command, 175
 - deleting, 177
 - deleting directories, 176
 - displaying contents, 179
 - moving/renaming, 177-178
 - wildcards/regular expressions, 179
 - job control and limiting resource use, 213-220
 - Linux file system hierarchy, 161-166
 - logging in/out from a remote computer, 157-158
 - logging out, 157
 - multiple terminals, 233-234
 - MySQLclient, 611-612
 - navigating Linux file system, 166-169
 - network interface configuration, 394-398

- permissions, 169-175
 - assigning permissions, 170-171
 - chmod command, 172-173
 - directory permissions, 171-172
 - file permissions, 172-173
 - set UIDs/GID permissions, 173-175
- PostgreSQL client, 612
- reading documentation, 160-161
- reasons for use, 188-189
- redirecting input and output, 209-211
- redirection of streams, 211-212
- root users, 180-185
- scripting. See Python
- shell control, 275-276
- system reset, 234-235
- text editors, 226-232
 - emacs, 229-230
 - nano, 227-228
 - sed and awk, 230-232
 - vi, 228-229
- user accounts, 158-159
- working with files, 175-179
- command prompt. See command line
- command-line interface (CLI). See command line
- commands
 - ac, 252
 - adduser, 183
 - AppArmor, 442
 - apropros, 160-161
 - apt-get autoclean, 146
 - apt-get clean, 146
 - apt-get dist-upgrade, 145
 - apt-get install, 145
 - apt-get moo, 147
 - apt-get remove, 146
 - apt-get upgrade, 144
 - at, scheduling tasks, 267-270
 - badblocks, 449
 - batch, scheduling tasks, 267-270
 - bg, 216
 - /bin directory, 162-163
 - byobu, 233-234
 - cat, 164, 179
 - cd, 168-169
 - chfn, 174
 - chgrp, 173, 242
 - chmod, 172-173, 242
 - chown, 242
 - chpasswd, 257
 - chsh, 248
 - comm, 213
 - command line, 185-209
 - background processing, 215-216
 - cat, 191
 - cd, 168-169, 191-193
 - chmod, 193
 - combining, 220-222
 - common commands and programs, 185-186
 - Coreutils, 237
 - cp, 194
 - du, 194-195
 - echo, 209
 - find, 195-197
 - grep, 197-198
 - less, 198-200
 - ln, 200-202
 - locate, 202
 - ls, 166-168, 202-204
 - man, 204-205
 - mkdir, 205
 - mv, 205
 - pwd, 169

- rm, 206
- running in order, 222
- running previous command, 235-236
- separating multiple command, 236-237
- shortcuts, 237
- sort, 206-208
- system reset, 234-235
- tail, 208-209
- viewing history, 236
- which, 209
- compression of files, 232-233
- cp, 178, 346, 365-366
- cron, 270-272
- date, 30
- deluser, 184
- diff, 212-213
- dmesg, 33
- dropuser (PostgreSQL), 607
- e2fsck, 449
- echo, 164
- edquota, 264
- emacs text editor, 229-230
- env, 223
- exit, 157
- faillog, 330
- fg, 216
- find, 174
- gpasswd, 245
- gprof, 689
- group, 244
- groupadd, 245
- groupdel, 245
- groupmod, 245
- grpck, 245
- hdparm, 447-448
- hdx=ide-scsi, 446
- htop, 327
- hwclock, 31
- idebus=xx, 446
- idex=autotune, 446
- idex=dma, 446
- ifconfig, 345, 379, 394-396
- info, 160
- init, 316
- Internet connectivity, 418
- iptables, 439
- jobs, 214-215
- kernel module management, 462-463
- kill, 325-326
- lastlog, 330
- less, 42, 160, 179
- logout, 157
- logrotate, 331-336
- ls, 166-170
- make, 685
- man, 41, 160
- mc, 366
- mkdir, 171, 175-176
- mv, 177-178
- mysql -u root, 602
- nano text editor, 227-228
- netstat, 398
- nice, 219-220, 326
- passwd, 174, 247
- patch, 467-468
- pci=biosirq, 447
- Perl, 746
- perldoc, 730
- ping, 376-378
- postconf, 559
- printenv, 223, 259
- printing, 494
- ps, 213-214, 323-325
- pwd, 157, 169

- Quickly, 704-706
- quotacheck, 264
- quotaoff, 264
- quotaon, 264
- rcp, 422
- relational database services, 613
- renice, 326
- repquota, 264
- rm, 177
- rmdir, 176
- route, 396-398
- rsync, 366-368
- rtcwake, 272-274
- /sbin directory, 162-163
- scp, 422
- sed and awk text editors, 230-232
- sftp, 421-423
- shutdown, 20, 184-185
- smbclient, 488-489
- smbstatus, 488
- splint, 688-689
- ssh-keygen, 423-425
- su, 258-260
- sudo, 25, 180-183, 240-241
- sync, 448
- tail, 329
- tar, 355-357, 364-365
- testparm, 487
- time, 326-327
- top, 216-218, 327
- touch, 169, 175, 424
- tune2fs, 448-449
- UFW (Uncomplicated Firewall), 436-438
- umask, 169
- uname, 36
- uptime, 328
- user accounts, 265
- useradd, 245, 247
- usermod, 245, 248
- vi text editor, 228-229
- watch, 328
- whereis, 161
- zless, 42
- comments, PHP programming, 775
- commercial games, 135-136
- commercial support websites, 44-45
- Common Business Oriented Language (COBOL), 803-804
- Common UNIX Printing System (CUPS), 490
- CommuniGate Pro, 569
- community teams, testing, 721-723
- comparison
 - files, 212-213
 - numbers
 - in pdksh and bash, 294-295
 - in tcsh, 298-299
 - strings
 - in pdksh and bash, 292-294
 - in tcsh, 297-298
- comparison operators (Perl), 732-733
- compiling applications from source, software management, 148-150
- compiling the kernel, 468-475
 - troubleshooting, 475-476
- Compiz, 120
- compound operators (Perl), 733
- Comprehensive Perl Archive Network (CPAN), 741, 743
- compressed files, command line, 232-233
- Compute Infrastructure (Nova) service, OpenStack, 658-659
- computer attacks, hacker versus cracker, 430
- computer engineering, Cloud. See Cloud
- Concurrent Versioning System (CVS), 696

- conditional statements
 - Perl, 735-736
 - PHP programming, 779-780
 - Python, 759-761
- configuration
 - Apache web server, 498-499
 - directives, 503-504
 - editing `apache2.conf`, 504-506
 - `.htaccess` configuration files, 507-509
 - multiprocessing modules, 507
 - Bazaar, 715
 - clients, proxy servers, 574-575
 - DHCP, 405-406
 - DHCP network hosts, 407-409
 - Dial-Up access, 416-417
 - Digital Subscriber Line (DSL) access, 414-416
 - Fetchmail, 563-564
 - FireFox, 574
 - graphical tools, 401-403
 - kernel, 468-475
 - local Bash shell, 715-716
 - loopback interface, 374-375
 - management, 150-151
 - MySQL, 601-604
 - networking tools, 393-403
 - NFS client, 482
 - NFS server, 480-481
 - Nginx, 530-533
 - Nginx modules, 536
 - Postfix, 558-560
 - Postfix masquerading, 560-561
 - PostgreSQL, 604-607
 - PPPoE, 415-416
 - quotas, 264
 - Samba, 484-487
 - software repositories, 26-28
 - system settings, 28-31
 - printers, 29
 - time and date, 30-31
 - Very Secure FTP server, 548-551
 - wireless networks, 32-33
- configuration files
 - `/etc` directory, 163
 - `.htaccess`, 507-509
 - networking, 399-401
 - version control, 368-370
- `./configure`, building Apache, 498
- configuring
 - code, `autoconf` utility, 687-688
 - firewalls, 436-439
 - LDAP clients, 632-633
 - LDAP server, 628-631
 - Tripwire, 434-435
 - Unity, 64-66
- Conky, 338-342
- console. See command line
- console-based monitoring tools, 323-336
 - disk quotas, 329
 - disk space, 328-329
 - free and used memory, 327-328
 - kill command, 325-326
 - log files, checking, 329-331
 - log files, rotating, 331-336
 - priority scheduling and control, 326-327
- constants
 - Perl string constants, 734-735
 - PHP programming, 774
- constructors, Python object orientation, 764-765
- control structures, Perl, 736-739
- controlling services, BIOS (basic input/output system), 318
- convert utility (ImageMagick), 104

- copying files, 364-368
 - cp command, 365-366
 - Midnight Commander, 366
 - rsync command, 366-368
 - tar command, 364-365
- core applications, mobile development
 - for Android, 817
- Coreutils, 237
- CouchDB, 621-622
- cp command, 178, 189, 194, 346, 365-366
- CPAN (Comprehensive Perl Archive Network), 741, 743
- crackers, hackers versus, 430
- CREATE DATABASE statement
 - MySQL, 602
 - PostgreSQL, 605
- creating
 - MySQL database, 602-604
 - PostgreSQL database, 605
 - tables, relational database services, 594-595
- cron command, 270-272
- Crossover Games, 137
- CrossOver Office, 92
- Cube 2: Sauerbraten, 129
- CUPS (Common UNIX Printing System), 490, 490-492
- customizing Unity, 64-66
- CustomLog directive (Apache), 523
- CVS (Concurrent Versioning System), 696

D

- D programming language, 804
- Dalvik, 816
- darktable, 106
- Dart, 806-807

- Dash (Unity desktop), 21, 60-63, 65
- Dashboard (Horizon), OpenStack, 659
- data
 - mirroring, 352
 - retrieval from databases, 596-598
- data backup, 345-372
 - choosing a strategy, 345-352
 - copying files, 364-368
 - hardware and media, 352-354
 - software, 354-363
 - system rescue, 370-372
 - version control, 368-370
- data directory (PostgreSQL), initializing, 604-605
- data files, /var directory, 166
- data integrity, MySQL versus PostgreSQL, 599-601
- data locking, MySQL versus PostgreSQL, 599
- data loss, 346-347
- data structures, Perl, 730-732
- database administrators (DBAs), 590
- databases
 - NoSQL, 591, 615-618
 - document stores, 620-623
 - graph stores, 624-625
 - key/value stores, 618-620
 - wide column stores, 623-624
 - PHP programming, 797-800
 - relational database services, 589-613
 - client/server system, 607-613
 - commands, 613
 - comparison of MySQL and PostgreSQL, 598-601
 - creating tables, 594-595
 - future of MySQL, 589
 - how they work, 592-594
 - inserting data in tables, 595-596
 - MySQL, 601-604

- PostgreSQL, 604-607
 - retrieving data, 596-598
 - SQL basics, 594
- date, configuring system settings, 30-31
- date command, 30
- day job crackers, 430
- day-to-day usage, APT, 144-147
- DBAs (database administrators), 590
- Debian, 26, 36, 39
- debugging tools, 688-689
- default runlevel, booting into, 317
- definitions
 - Python functions, 761-762
 - runlevels, 316
- Deja Dup backup tool, 358-360
- deleting users, PostgreSQL, 606
- deluser command, 184
- deny directive, Apache web server access control, 510-511
- deployment, Eucalyptus, 656-657
- depmod command, 463
- design guidelines for applications, 825
- Desktop Couch, 703
- Desktop DVD, 10
- desktop environment, interfaces, 120-121
- destructors, Python object orientation, 764-765
- development, 711-720
 - finding bugs with Harvest, 719
 - fixing bugs and packaging, 716-718
 - helping with Ubuntu development, 43
 - installation packages, 713-716
 - MOTU (Masters of the Universe), 719
 - opportunistic. *See* opportunistic development
 - six-month cycle, 712-713
- device drivers, 461
- Device section (xorg.conf file), 53, 56
- devices, security, 345
- DevOps, SysAdmin versus, 653-654
- dhclient, 405-406
- DHCP (Dynamic Host Configuration Protocol), 403-409, 640
 - client, 405-406
 - configuring network hosts, 407-409
 - how it works, 403-404
 - installation and activation, 404-405
 - server, 406
 - software installation and configuration, 405-406
 - uses, 409
- Dia (LibreOffice component), 86
- Dial-Up access, Internet connectivity, 416-417
- dictionaries, Python, 758-759
- diff command, 212-213
- digiKam, 106
- digital cameras, 106-107
- Digital Subscriber Line (DSL) access, 414-416
- directives (Apache web server), 503-504
 - access control, 510-511
 - CustomLog, 522-523
 - satisfy, 513
- directories
 - Apache package, 497
 - command line
 - changing with cd command, 168-169
 - listing contents with ls command, 166-168
 - pwd command, 169
 - Linux, 161-162
 - Linux source tree, 459-461
- Directory Information Tree (DIT), 628
- directory permissions, 171-172
- DirectoryIndex directive, Apache web server configuration, 506
- disable command, 494

disabling

- Apache modules, 514
- file access time, 449
- disaster recovery plan, 442-443
- disk drives, tuning, 446-447
- disk quotas, 263-264, 329
- disk space, console-based monitoring tools, 328-329
- display manager, X Server, 58
- dist-upgrade option, 25
- DIT (Directory Information Tree), 628
- dman tool (Bikeshed), 707
- dmesg command, 33
- DNS (Domain Name System)
 - DNS records, 676-679
 - A, 676
 - AAAA, 676
 - CNAME, 677
 - MX, 677
 - NS, 677-678
 - SOA, 678-679
 - TXT, 679
 - DNS servers, 675-676, 679-681
 - domain names, explained, 675
 - reasons for using, 673-674
- DocBook, 89-90
- document stores, NoSQL databases, 620-623
- documentation
 - command line, 160-161
 - helping with development, 43
 - Ubuntu and Linux, 41-45
- DocumentRoot directive, Apache web server configuration, 506
- documents, kernel programmers, 459-460
- Domain Name System (DNS). See DNS (Domain Name System)

domain names

- explained, 675
- mapping IP addresses to, 673-674
 - DNS records, 676-679
 - DNS servers, 675-676
 - server setup with BIND, 679-681
- dotdee, 150-151
- do.until loop (Perl), 738-739
- do.while loop
 - Perl, 738-739
 - PHP, 785
- Draw (LibreOffice component), 85
- drivers
 - device drivers, 461
 - inline drivers, 461-462
- Drizzle, 589
- dropuser command (PostgreSQL), 607
- DSL (Digital Subscriber Line) access, 414-416
- du command, 189, 194-195
- dual boot, 14
- DVD installation jump start, 10
- DVD+RW/-RW drives, 353
- DVDs
 - burning, 107-112
 - Brasero, 108
 - command line, 110-112
 - formats, 110
- Dynamic Host Configuration Protocol. See DHCP (Dynamic Host Configuration Protocol)

E

- e2fsck command, 449
- echo command, 164, 209
- Eclipse, 693
- Eclipse Foundation project, Jetty, 541

- editing
 - apache2.conf, 504-506
 - /etc/modprobe.conf file, 392
 - video, 116-117
- editing commands
 - emacs text editor, 229-230
 - sed and awk text editors, 230-232
- edquota command, 264
- Edubuntu, 636
- elements (xorg.conf file), 52-57
 - Device section, 53, 56
 - Files section, 53-54
 - InputDevice section, 53, 55
 - Module section, 53-54
 - Monitor section, 53, 55-56
 - Screen section, 53, 56-57
 - ServerLayout section, 53
- emacs command, 189
- emacs text editor, 226, 229-230
- email, 553-571
 - alternatives to Microsoft Exchange Server, 568-570
 - Fetchmail, 563-566
 - how email is sent and received, 553-558
 - mail delivery agents, 567
 - Postfix configuration and operation, 558-560
- email clients, 73-76
 - Evolution, 74-75
 - Mozilla Thunderbird, 73-74
- embedded spaces, strings with, 290
- Empathy, 77-78
- emulators, 127
- enable command, 494
- enabling Apache modules, 514
- endless loops, shell programs, 303
- Enlightenment, 120-121
- entering PHP mode, 770
- enterprise servers, monitoring tools, 343
- env command, 223
- environment, Launchpad, 714-716
- environment variables
 - command line, 222-226
 - shell scripts, 283
- Erlang, 804-805
- errors, compiling kernel, 475-476
- escape sequences, PHP programming, 775-776
- /etc directory, 163
- /etc/host.conf file, 401
- /etc/hosts file, 399
- /etc/init.d/apache2, starting/stopping Apache web server, 501
- /etc/modprobe.conf file, 463
 - editing, 392
 - manually loading kernel modules, 392-393
- /etc/nsswitch.conf file, 400
- /etc/resolv.conf file, 400-401
- /etc/samba/smb.conf file, 484-487
- /etc/services file, 399-400
- etiquette, IRC (Internet Relay Chat), 79-80
- Eucalyptus, 654, 656
 - deployment/installation, 656-657
- event-driven architecture, 528
- Evolution (GNOME Office component), 74-75, 88, 632
- executing
 - Python scripts, 750-751
 - shell scripts, 279-311
 - accessing variable values, 284
 - assigning value to variables, 284
 - automation of tasks, 286-288
 - backslash, 291
 - backtick, 292
 - break statement, 310
 - built-in variables, 288-289

- case statement, 308-309
- comparison of expressions in `pdksh` and `bash`, 292-301
- comparison of expressions with `tcsh`, 297-301
- exit statement, 310
- for statement, 301-302
- if statement, 306-307
- interpreting shell scripts, 282-283
- positional parameters, 284-286
- repeat statement, 305
- running shell program, 280-281
- select statement, 305-306
- shift statement, 306
- special characters, 289
- storing scripts for access, 281-282
- strings with embedded spaces, 290
- unexpanded variables, 290-291
- until statement, 304-305
- variables, 283-284
- while statement, 303-304

execution operator, 780-781

Exim, 555

exit command, 157, 310

exit function (PHP), 794

exiting PHP mode, 770

expressions, Perl regular expressions, 739

extensions, Firefox, 70-71

external attacks, 430

extract() function (PHP), 790-791

Extraversion level (kernel), 465

F

faillog command, 330

fclose() function (PHP), 793

FDDI (fiber distributed data interface), 388

features, LTSP, 640-641

Fetchmail, 563-566

- configuration, 563-564
- installation, 563
- user accounts, 565-566

fg command, 216

fiber optic cable, 390

fiber optics, 388

file access time, disabling, 449

file operators

- in `pdksh` and `bash`, 295-296
- in `tcsh`, 299-300

file permissions, 172-173, 242-243

file system, Linux hierarchy, 161-166

file system authentication, Apache web server, 509

file system settings, tuning, 448

File Systems tab (System Monitor), 338

File Transfer Protocol. See FTP (File Transfer Protocol)

file_get_contents() function (PHP), 791

file_put_contents() function (PHP), 791

files

- Apache file locations after install, 497
- command line, 175-179
 - copying files, 178
 - creating directories, 175-176
 - creating files with `touch` command, 175
 - deleting directories, 176
 - deleting files, 177
 - displaying file contents, 179
 - moving/renaming files, 177-178
 - wildcards/regular expressions, 179
- comparison, 212-213
- compressed, 232-233
- configuration, version control, 368-370

- copying, 364-368
 - cp command, 365-366
 - Midnight Commander, 366
 - rsync command, 366-368
 - tar command, 364-365
- /etc/host.conf, 401
- /etc/hosts, 399
- /etc/nsswitch.conf, 400
- /etc/resolv.conf, 400-401
- /etc/services, 399-400
- ftphosts, 551-552
- .htaccess configuration, Apache web server configuration, 507-509
- PHP functions, 791-793
- restoring from an archive, 356-357
- saving from nonbooting hard drive, 372
- sharing, 479
 - NFS (Network File System), 480-482
 - Ubuntu installation, 21
- Files section (xorg.conf file), 53-54
- filesize() function (PHP), 793
- find command, 174, 189, 195-197
- find2perl command, 746
- Firefox, 70-71. *See also* Apache web server
 - configuration, 574
 - RSS feeds, 76
- firewalls, configuring, 436-439
- first-person shooter (FPS) games, 129
- fixing bugs, 716-718
- FLAC (sound format), 96
- Flash, 70-71, 115
- flavors (Ubuntu), 11
- flexbackup tool, 363
- FlightGear, 134-135
- FlockDB, 625
- FLV (video format), 114
- fopen() function (PHP), 792
- for loop
 - Perl, 736-737
 - PHP, 783
 - Python, 759-760
- foreach loop
 - Perl, 737
 - PHP, 784
- ForecastFox, 71
- for statement, writing shell scripts, 301-302
- formats
 - DVDs, 110
 - graphics manipulation, 103-105
 - sound, 96-97
 - video, 114
- Forth, 805
- Fortran, 806
- forwarding email, 562-563
- FPS (first-person shooter) games, 129
- FQDN (fully qualified domain name), 628
- fread() function (PHP), 792-793
- free and used memory, console-based monitoring tools, 327-328
- Frets on Fire, 133-134
- Frozen Bubble, 131
- FTP (File Transfer Protocol), 543
 - configuring Very Secure FTP server, 548-551
 - ftphosts file, 551-552
 - server selection, 543-545
 - servers, 544-545
 - software installation, 545-546
 - users, 546-548
- ftphosts file, 551-552
- full backups
 - incremental backups, 351
 - periodic basis, 351
 - tar backup tool, 355-356
- fully qualified domain name (FQDN), 628

functions

- PHP programming, 786-797
 - arrays, 789-791
 - files, 791-793
 - miscellaneous functions, 793-797
 - strings, 786-789
- Python, 761-762
- shell scripts, 310-311
- future of MySQL, 589
- fwrite() function (PHP), 793

G

games, 127-138

- Battle for Wesnoth, 133
- children, 134
- commercial, 135-136
- FlightGear, 134-135
- Frets on Fire, 133-134
- Frozen Bubble, 131
- installation, 129-136
- proprietary video drivers, 128-129
- Scorched 3D, 130
- Speed Dreams, 134
- Steam, 136
- SuperTux, 131-132
- Warsow, 129-130
- Windows, 136-137
- gcc (GNU Compiler Collection), 683
- gCompris, 134
- gdb tool, 689
- gecos field, 253
- gedit text editor, 226, 703
- Genprof, 442
- GIDs (group IDs), 173-175, 242
- GIF (graphics format), 104

- Gigabit Ethernet, 388
- GIMP (GNU Image Manipulation Program), 88, 101-102
- Git, 699-700
- GitHub, 700
- Glade, 691-692, 703
- Glance (Imaging Service), OpenStack, 659
- global behavior, Samba, 486-487
- global options, configuring Fetchmail, 564
- Gmail, 76
- Gnat Programming System, 802
- GNOME, Glade client, 691-692
- GNOME file roller, 357
- GNOME Office, 87-88
- GNOME3, 124-125
- gnome-nettool, 343
- gnome-screenshot, 105
- gnome-shell, 125
- GNU C compiler, 689-690
- GNU Compiler Collection (gcc), 683
- GNU Coreutils, 237
- GNU Fortran 95 compiler, 806
- GNU GPL, 35-36
- GNU Image Manipulation Program (GIMP), 101-102
- GNU Privacy Guard, 713
- GNU Project, 153
- GNU/Linux, 458
- Gnumeric (GNOME Office component), 88
- Go, 805-806
- Google, search tips, 44
- Google Chrome, 71-73
- gpasswd command, 245
- GPG key, 713-714
- gprof command, 689
- Grand Unified Boot Loader (GRUB2), 13
- GRANT statement (PostgreSQL), 607
- granting privileges, PostgreSQL databases, 607

- graph stores, NoSQL databases, 624-625
- graphical clients, relational database services, 613
- graphical configuration tools, 401-403
- graphical development tools, 690-693
- graphical process monitoring tools, 336-343
 - Conky, 338-342
 - System Monitor, 336-338
- graphical user interfaces (GUIs). See GUIs (graphical user interfaces)
- graphics manipulation, 100-106
 - editor options, 106
 - formats, 103-105
 - GNU Image Manipulation Program (GIMP), 101-102
 - Photoshop, 102
 - scanners, 103
 - screen images, 105
- grep command, 189, 197-198
- Groklaw, 77
- Groovy, 806
- group command, 244
- Group directive, Apache web server configuration, 505
- group IDs (GIDs), 173-175, 242
- group listing, 243-244
- group management, 243-246
- group permissions, 242
- groupadd command, 245
- groupdel command, 245
- groupmod command, 245
- groups command, 265
- grpck command, 245
- GRUB2 (Grand Unified Boot Loader), 13, 371
- GStreamer, 96, 703
- GTK, 87, 703
- GUPFW, 438-439

- GUIs (graphical user interfaces), 119-126
 - desktop environment, 120-121
 - GNOME3 and Ubuntu GNOME, 124-125
 - Juju, 665
 - KDE and Kubuntu, 121-122
 - LXDE and Lubuntu, 123-124
 - touchscreen interface, 823-824
 - application design guidelines, 825
 - creating applications, 824-825
 - SDK installation, 824
 - User Interface Toolkit, 825-826
 - Ubuntu Kylin, 125
 - Unity desktop, 49-67
 - Mir, 50
 - X Server, 49-58
 - Xfce and Xubuntu, 122-123
- gunzip command, 232
- gzip command, 232

H

- hackers, crackers versus, 430
- handheld digital cameras, 106-107
- hard disk performance tuning, 445-449
 - badblocks command, 449
 - disabling file access time, 449
 - e2fsck command, 449
 - file system settings, 448
 - hdparm command, 447-448
 - tune2fs command, 448-449
 - tuning disk drives, 446-447
- hardware
 - data backup, 352-354
 - networking, 387-393
 - hubs and switches, 390-391
 - initializing new hardware, 391-393

- network cable, 389-390
- NIC (Network Interface Cards), 387-389
- routers and bridges, 391
- specifications, 10
- TV and video, 112-114
- Harvest, finding bugs, 719
- hashes (Perl), 731
- Haskell, 807
- HBase, 623
- HDLC (high-level data link control), 416
- hdparm command, 447-448
- hdx=ide-scsi command, 446
- header lines (newsgroup articles), 81
- help
 - IRC (Internet Relay Chat), 48
 - mailing lists, 46-48
 - software management, 44
 - with Ubuntu development, 43
 - websites, 43-46
 - commercial support, 44-45
 - Linux guides, 45-46
 - LUGs (Linux User Groups), 45
 - search tips, 43-44
 - Ubuntu-specific sites, 46
- hibernate, 18
- hierarchy, Linux file system, 161-166
- high-level data link control (HDLC), 416
- history
 - command history, viewing, 236
 - Internet, 69
 - LibreOffice, 86
 - Linux, 35-36
 - Ubuntu, 39
- home directories, sharing, 486-487
- /home directory, 163-164
- home users
 - backup programs, 349
 - of Ubuntu, 41
- hooks, 664
- Horde, 570
- Horizon, OpenStack, 659
- host names, 675
- hosts, adding to /etc/hosts file, 399
- Hotmail, 76
- HOWTO documents, 42
- .htaccess configuration files, Apache web server configuration, 507-509
- HTML (Hypertext Markup Language), 69
- HTML forms, PHP programming, 797
- htop command, 327
- HTTP (Hypertext Transfer Protocol), 69
- HTTP servers, 523-525
 - Apache Tomcat, 542
 - CA authority, 525
 - Cherokee, 541
 - Jetty, 541
 - lighttpd, 523-525, 539-540
 - Nginx, 527-529
 - configuration, 530-533
 - installation, 529-530
 - modules, 536
 - PHP setup, 534-535
 - virtual hosting, 533-534
 - thttpd, 542
 - Yaws (Yet Another Web Server), 540
- HTTPS, Nginx, 536-538
- hubs, 390-391
- Hugin, 106
- Humble Indie Bundle, 136
- hwclock command, 31
- hybrid cloud, Eucalyptus, 657

- HyperGraphDB, 624-625
 - Hypertext Markup Language (HTML), 69
 - Hypertext Transfer Protocol (HTTP), 69
- I**
- IaaS (Infrastructure as a Service), 655-656
 - idebus=xx command, 446
 - Identity Service (Keystone), OpenStack, 659
 - IDEs (integrated development environments), 692-693
 - idex=autotune command, 446
 - idex=dma command, 446
 - if statement, executing shell scripts, 306-307
 - ifconfig command, 345, 379, 394-396
 - if/else conditional statements, Perl, 735-736
 - ImageMagick, convert utility, 104
 - Imaging Service (Glance), OpenStack, 659
 - IMAP (Internet Message Access Protocol), 557
 - implementation, quotas, 263-264
 - Impress (LibreOffice component), 85
 - in_array() function (PHP), 790
 - include keyword (PHP), 785-786
 - incremental backups, tar backup tool, 355-356
 - info command, 160
 - Infrastructure as a Service (IaaS), 655-656
 - init command, 316
 - init scripts, BIOS (basic input/output system), 317-318
 - initialization
 - BIOS (basic input/output system), 317-318
 - data directory (PostgreSQL), 604-605
 - network hardware, 391-393
 - Inkscape, 106
 - inline drivers, 461-462
 - input, redirecting, 209-211, 277-278
 - InputDevice section (xorg.conf file), 53, 55
 - inserting data in tables, relational database services, 595-596
 - insmod command, 462
 - installation
 - ADT Eclipse plug-in, 818
 - Android SDK, 817
 - Apache web server, 496-499
 - building the source, 498
 - Ubuntu repositories, 496-498
 - CPAN module (Perl), 743
 - development packages, 713-716
 - DHCP, 404-406
 - Eucalyptus, 656-657
 - Fetchmail, 563
 - FTP software, 545-546
 - games, 129-136
 - Java, 817
 - Juju, 660-663
 - LTSP, 639-640
 - NFS, 480
 - Nginx, 529-530
 - proprietary video drivers, 128-129
 - SDK, 817, 824
 - Squid, 574
 - Ubuntu, 9-34
 - 32-bit versus 64-bit Ubuntu, 12-13
 - on Mac hardware, 11
 - post-installation configuration problems, 33-34
 - preparation, 9-14
 - programs and files, 21
 - shutting down, 20-21
 - software repositories, 26-28
 - Software Updater, 22-25
 - step-by-step, 14-20
 - sudo command, 25

- system settings, 28-31
 - wireless network configuration, 32-33
- virtual devices, 819
- instant messaging, Empathy, 77-78
- integrated development environments (IDEs), 692-693
- interfaces. See GUIs (graphical user interfaces)
- internal attacks, 429
- Internet, history, 69
- Internet applications, 69-82
 - Chromium, 71-73
 - email clients, 73-76
 - Evolution, 74-75
 - Mozilla Thunderbird, 73-74
 - Empathy, 77-78
 - Firefox, 70-71
 - Google Chrome, 71-73
 - Internet Relay Chat (IRC), 78-80
 - RSS feeds, 76-77
 - Firefox, 76
 - Liferea, 76-77
 - Usenet newsgroups, 80-82
- Internet connectivity, 412-418
 - commands, 418
 - common configuration information, 412-413
 - Dial-Up access, 416-417
 - DSL access, 414-416
 - troubleshooting connection problems, 417-418
- Internet Message Access Protocol (IMAP), 557
- Internet Relay Chat (IRC), 48, 78-80
- interpreting shell scripts, 282-283
- IP addresses
 - clients, 579-580
 - mapping to domain names, 673-674
 - DNS records, 676-679
 - DNS servers, 675-676
 - server setup with BIND, 679-681

- IP masquerading, 381
- iptables command, 439
- IPv4, 380-381
- IPv6, 381-385
- IRC (Internet Relay Chat), 48, 78-80
- IRCd server, 80
- isset() function (PHP), 793
- iwconfig tool, 409
- iwlist tool, 409
- iwpriv tool, 409
- iwspy tool, 409

J

- Java, 70, 807-808, 817
- Java Runtime Environment (JRE), 817
- Java Virtual Machine (JVM), 817
- JavaScript, 808
- Javascript Object Notation (JSON), 618
- Jetty, 541
- job control commands, 213-220
- jobs command, 214-215
- JPG (graphics format), 104
- JRE (Java Runtime Environment), 817
- JSON (Javascript Object Notation), 618
- Juju, 660-665
 - charms, 663-665
 - GUI, 665
 - installation, 660-663
 - on Mac OS X, 665
 - managing sets of servers, 669-670
- JVM (Java Virtual Machine), 817

K

kate text editor, 226
 KDE, 121-122
 KDE ark archiving tool, 358
 KDE process, monitoring tools, 343
 Kdenlive, 117
 KDevelop client, 690-691
 KDevelop Setup Wizard, 690
 kdf tool, 343
 kedit text editor, 226
 keep-one-running tool, 709
 kernel

- built-in security, 429
- interacting via /proc directory, 164-165
- management, 457-477
 - compiling the kernel, 468-475
 - Linux kernel, 458-462
 - modular kernels, 462-464
 - obtaining sources, 466
 - patching the kernel, 467-468
 - recompiling the kernel, 464-465
 - troubleshooting during compile, 475-476
 - versions, 465-466
- performance tuning, 450-451
- tuning disk drives, 446-447
- version numbers, 36

 kernel source tree, 459-461
 Kernel-based Virtual Machine (KVM), 645-649
 key-based logins, ssh-keygen command, 423-425
 Keystone, OpenStack, 659
 key/value stores, NoSQL databases, 618-620
 Kile, 91
 kill command, 325-326
 Kirkland, Dustin, 723
 Kmail mail client, 75
 Knoppix, 124

KOffice, 87-89
 krsort() function (PHP), 790
 ksort() function (PHP), 790
 KSpread, 89
 ksysguard tool, 343
 Kubuntu, 121-122
 KVM (Kernel-based Virtual Machine), 645-649
 KWord, 88

L

labels in DNS (Domain Name System), 675
 LAN, enabling network printing, 490
 Landscape, 343, 666, 671
 LANG environment variable, 223
 lapd-utils package, 628
 laptops, security, 434
 large enterprise users, backup programs, 349
 last looping construct (Perl), 738
 lastlog command, 330
 LaTeX, 91
 Launcher (Unity desktop), 59-60
 Launchpad, 698, 701-702

- accounts, 714
- environment, 714-716

 LDAP (Lightweight Directory Access Protocol), 627-634

- administration, 633-634
- client configuration, 632-633
- server configuration, 627

 ldapadd utility, 633
 ldapdelete utility, 634
 ldapmodify utility, 633
 ldapsearch utility, 633
 LDM (LTSP Display Manager), 640
 LDTP (Linux Desktop Testing Project), 723

- less command, 42, 160, 179, 189, 198-200
- levels of backup, 350
- libraries, mobile development for Android, 816
- LibreOffice, 85-86
- licensing, 2-3, 35
- Liferea, RSS feeds, 76-77
- lighttpd, 539-540
- Lightweight Directory Access Protocol. *See* LDAP (Lightweight Directory Access Protocol)
- Lightworks, 117
- limits, IPv4 addressing, 380-381
- Linux
 - C programming tools, 684-685
 - components of, 36-37
 - directories, 161-162
 - documentation, 41-45
 - file system hierarchy, 161-166
 - history, 35-36
 - IRC (Internet Relay Chat), 48
 - kernel management, 458-462
 - kernel source tree, 459-461
 - types of kernels, 461-462
 - mailing lists, 46-48
 - Perl, 727-730
 - reasons for use, 37-39
 - websites, 43-46
 - commercial support, 44-45
 - Linux guides, 45-46
 - LUGs (Linux User Groups), 45
 - search tips, 43-44
 - Ubuntu-specific sites, 46
- Linux Desktop Testing Project (LDTP), 723
- Linux Documentation Project, 417
- Linux file system, navigating, 166-169
- Linux kernel
 - loading, 315
 - mobile development for Android, 816
- Linux Terminal Server Project, 635-641
 - features, 640-641
 - installation, 639-640
 - requirements, 636-639
- Linux User Groups (LUGs), 45
- Lisp, 808-809
- Listen directive, Apache web server configuration, 505
- listening to music, 97-99
- lists, Python, 756-758
- Live Bookmarks, 76
- In command, 189, 200-202
- loading Linux Kernel, 315
- local GUI clients, database access, 609
- localhost interface, 374-375
 - loopback interface
 - availability, 374
 - configuration, 374-375
- locate command, 189, 202
- log files, console-based monitoring tools
 - checking, 329-331
 - rotating, 331-336
- log files (Apache), 521-523
- LogFormat statements (Apache), variables, 522-523
- logging in (command line), 157-158
- logging out (command line), 157-158
- logical operators
 - in pdksh and bash, 296-297
 - in tcsh, 300-301
- logname command, 265
- logout command, 157
- logrotate command, 331-336
- logs
 - Apache web server, 521-523
 - purging in Perl, 743-744
- Long Term Support (LTS), 28

- loopback interface
 - availability, 374
 - configuration, 374-375
- loops
 - endless, 303
 - Perl, 736-739
 - PHP, 783-785
 - Python, 759-761
- lp command, 494
- lpc command, 494
- lpq command, 494
- lprm command, 494
- lpstat command, 494
- ls command, 166-170, 190, 202-204
- lsmod command, 462
- LTS (Long Term Support), 28
- LTSP (Linux Terminal Server Project), 635-641
 - features, 640-641
 - installation, 639-640
 - requirements, 636-639
- LTSP Display Manager (LDM), 640
- ltsp-server-standalone package, 639
- Lua, 809
- Lubuntu, 123-124
- LUGs (Linux User Groups), 45
- LXDE, 123-124
- LyX, 91

M

- MaaS (Metal as a Service), 656, 666
- MAC (Mandatory Access Control) system, AppArmor, 440-442
- Mac hardware, Ubuntu installation, 11
- Mac OS X, Juju on, 665

- machine protection, 432-435
 - devices, 345
 - passwords and physical security, 433-434
 - Tripwire, 434-435
 - wireless networks, 433
- macros, 686-687
- Mago, 723
- mail, sending in Perl, 741-743
- mail delivery agents (MDAs), 556-557, 567
- mail relaying, 562
- mail server options, configuring Fetchmail, 564-565
- mail transfer agents (MTAs), 553-556
- mail user agent (MUA), 557-558
- mailing lists, 46-48
- Major version (kernel), 465
- make command, 190, 685
- make utility (kernel), 460
- make xconfig tool, 471-474
- makefile targets, 686-687
- makefiles, 685-687
- man command, 41, 160, 190, 204-205
- man pages, 41, 160
- management
 - configuration, 150-151
 - kernel, 457-477
 - compiling the kernel, 468-475
 - Linux kernel, 458-462
 - modular kernels, 462-464
 - obtaining sources, 466
 - patching the kernel, 467-468
 - recompiling the kernel, 464-465
 - troubleshooting during compile, 475-476
 - versions, 465-466
 - modular kernels, 462-464

- passwords, 253-257
 - changing in a batch, 257
 - password file, 253-254
 - policy, 253
 - security, 257
 - shadow passwords, 254-257
- photos, Shotwell Photo Manager, 107
- sets of servers
 - Ansible, 671
 - CFEngine, 671
 - Chef, 670
 - Juju, 669-670
 - Landscape, 671
 - Puppet, 670
- software. See software management
- users. See user management
- Mandatory Access Control (MAC) system, AppArmor, 440-442
- manual configuration
 - loopback interface, 374-375
 - PPPoE, 415-416
 - quotas, 264
 - Samba, 484-485
- manual start, Apache web server, 500-501
- mapping IP addresses to domain names, 673-674
 - DNS records, 676-679
 - DNS servers, 675-676
 - server setup with BIND, 679-681
- MariaDB, 589
- masquerading, Postfix, 560-561
- Massachusetts, standardization regulations, 87
- Master Boot Record (MBR), 13, 315
- Masters of the Universe (MOTU), 26, 719
- Math (LibreOffice component), 85
- MBR (Master Boot Record), 13, 315
- mc command, 366
- MDAs (mail delivery agents), 556-557, 567
- measuring MySQL buffer usage, 452-454
- media, data backup, 352-354
- Memcached, 619-620
- MemcacheDB, 619-620
- Mercurial, 698-699
- message body (newsgroup articles), 81
- message delivery interval (email), 561
- Metacity, 120
- Metal as a Service (MaaS), 656, 666
- methods
 - Python lists, 757-758
 - Python strings, 754-755
- Microsoft Exchange Server, 569
- Microsoft Office compatibility, 83-85
- Microsoft Windows, productivity applications, 91-92
- Midnight Commander, copying files, 366
- MIME (Multipurpose Internet Mail Extensions), 81
- Minor version (kernel), 465
- Mir, 50
- mirroring data, 352
- mkdir command, 171, 175-176, 190, 205
- mnemonic characters, 170
- mobile development for Android, 815-821
 - Android Runtime, 816
 - Application Framework, 816-817
 - core applications, 817
 - creating application, 820-821
 - installation of Android SDK, 817-820
 - libraries, 816
 - Linux kernel, 816
- mobile devices, security, 434
- mobile network sniffing, 433
- mod_access (Apache), 514
- mod_alias (Apache), 514
- mod_asis (Apache), 515
- mod_auth (Apache), 515

- mod_auth_anon (Apache), 515
- mod_auth_dbm (Apache), 516
- mod_auth_digest (Apache), 516
- mod_cgi (Apache), 516
- mod_dir (Apache), 516
- mod_env (Apache), 516
- mod_expires (Apache), 516
- mod_headers (Apache), 516-517
- mod_include (Apache), 517
- mod_info (Apache), 517
- mod_log_config (Apache), 517
- mod_mime (Apache), 517
- mod_mime_magic (Apache), 517
- mod_negotiation (Apache), 517
- mod_proxy (Apache), 517
- mod_rewrite (Apache), 518
- mod_setenvif (Apache), 518
- mod_speling (Apache), 518
- mod_ssl (Apache), 518
- mod_status (Apache), 518
- mod_unique_id (Apache), 518
- mod_userdir (Apache), 519
- mod_usertrack (Apache), 519
- mod_vhost_alias (Apache), 519
- moderated newsgroups, 81
- modinfo command, 463
- modprobe command, 462
- modular kernels
 - defined, 461-462
 - management, 462-464
- Module section (xorg.conf file), 53-54
- modules
 - Apache web server, 514-519
 - disabling, 514
 - enabling, 514
 - Nginx, 536
 - Perl, 741
 - MongoDB, 622
 - Monitor section (xorg.conf file), 53-56
 - monitoring tools, 323-344
 - console-based monitoring, 323-336
 - disk quotas, 329
 - disk space, 328-329
 - free and used memory, 327-328
 - kill command, 325-326
 - log files, checking, 329-331
 - log files, rotating, 331-336
 - priority scheduling and control, 326-327
 - enterprise servers, 343
 - graphical process, 336-343
 - Conky, 338-342
 - System Monitor, 336-338
 - KDE process, 343
 - user activity, 252-253
 - Mono, 809
 - monolithic kernels, 461
 - MOTU (Masters of the Universe), 26, 719
 - mounting shares, Samba, 489
 - MOV (video format), 114
 - Mozilla Public License, 3
 - Mozilla Thunderbird, 73-74
 - MP3 (sound format), 96
 - MPEG (video format), 114
 - MPMs (multiprocessing modules), Apache web server configuration, 507
 - MTAs (mail transfer agents), 553-556
 - mtr tool, 376-378
 - MUA (mail user agent), 557-558
 - multicast addressing, 386
 - multiline loops, Python, 760
 - multimedia applications, 93-117
 - burning CDs/DVDs, 107-112
 - digital cameras, 106-107

- graphics manipulation, 100-106
- sound and music, 93-100
- viewing video, 112-117
- multiple commands, separating, 236-237
- multiple terminals, command line, 233-234
- multiprocessing modules (MPMs), Apache web server configuration, 507
- Multipurpose Internet Mail Extensions (MIME), 81
- Multiverse repository, 142
- music, 93-100
 - adjusting volume, 95
 - listening to, 97-99
 - sound cards, 94-95
 - sound formats, 96-97
- Mutt mail client, 75
- mv command, 177-178, 190, 205
- MX record (DNS), 677
- MySQL
 - client/server system, 607-613
 - command-line client, 611-612
 - configuration, 601-604
 - future of, 589
 - performance tuning, 452-456
 - measuring buffer usage, 452-454
 - query cache, 454-455
 - query optimization, 456
 - versus PostgreSQL, 598-601
- mysql -u root command, 602

N

- Nagios, 343
- name-based virtual hosts (Apache), 520-521
- nano command, 190
- nano text editor, 226-228
- NAT (Network Address Translation), 380

- nautilus-image-converter package, 105
- navigating Linux file system, 166-169
- NcFTPd, 545
- negative indexes, Python strings, 755
- Neo4j, 624
- Nessus, assessing security vulnerabilities, 432
- NetBeans, 693
- netpbm tools, 104-105
- netstat command, 398
- Network Address Translation (NAT), 380
- network cable, 389-390
- Network File System. See NFS (Network File System)
- Network Interface Cards. See NICs (Network Interface Cards)
- network interface configuration, command line, 394-398
- Network Manager, configuring wireless networks, 32-33
- Network News Transfer Protocol (NNTP), 81
- network storage, 353
- networking, 373-418
 - checking connections, 376-378
 - configuration tools, 393-403
 - DHCP, 403-409
 - hardware, 387-393
 - hubs and switches, 390-391
 - initializing new hardware, 391-393
 - network cable, 389-390
 - NIC (Network Interface Cards), 387-389
 - routers and bridges, 391
 - Internet connectivity, 412-418
 - commands, 418
 - common configuration information, 412-413
 - Dial-Up access, 416-417
 - DSL access, 414-416
 - troubleshooting connection problems, 417-418

- IPv6, 382-385
 - localhost interface, 374-375
 - loopback interface availability, 374
 - loopback interface configuration, 374-375
 - organization, 385-386
 - broadcast addressing, 386
 - multicast addressing, 386
 - subnet masks, 386
 - subnetting, 385-386
 - Unicast addressing, 386
 - printers, 490
 - avoiding support problems, 493
 - creating network printers, 490-491
 - CUPS, 490-492
 - TCP/IP, 378-382
 - addressing, 379-381
 - IP masquerading, 381
 - ports, 382
 - troubleshooting, 391
 - wireless, 409-412
 - Networking Service (Neutron), OpenStack, 659
 - Neutron, OpenStack, 659
 - Newell, Gabe, 136
 - newsgroups, Usenet, 80-82
 - NewSQL, 617
 - newusers command, 265
 - next looping construct (Perl), 738
 - Nexuiz, 129
 - NFS (Network File System), 480
 - client configuration, 482
 - installation, 480
 - server configuration, 480-481
 - Nginx, 527-529
 - configuration, 530-533
 - HTTPS, 536-538
 - installation, 529-530
 - modules, 536
 - PHP setup, 534-535
 - virtual hosting, 533-534
 - nice command, 219-220, 326
 - NICs (Network Interface Cards), 387-389
 - 10BASE-T, 387
 - 100BASE-T, 387-388
 - 1000BASE-T, 388
 - Token Ring, 387
 - wireless network interfaces, 388-389
 - Nmap, assessing security vulnerabilities, 432
 - NNTP (Network News Transfer Protocol), 81
 - nonbooting hard drive, saving files, 372
 - NoSQL, 591
 - NoSQL databases, 615-618
 - document stores, 620-623
 - graph stores, 624-625
 - key/value stores, 618-620
 - wide column stores, 623-624
 - Nova (Compute Infrastructure) service, OpenStack, 658-659
 - NS record (DNS), 677-678
 - number comparison
 - in pdksh and bash, 294-295
 - in tcsh, 298-299
 - numbers, Python, 751-753
 - Nvidia, proprietary drivers, 128
- ## O
- object orientation (Python), 762-766
 - class and object variables, 763-764
 - class inheritance, 765-766
 - constructors and destructors, 764-765
 - object variables, Python object orientation, 763-764

- obtaining sources (kernel), 466
- OCaml, 810
- octal characters, 171
- Ogg-Vorbis (sound format), 96
- OGV/OGG (video format), 114
- OneConf, 151
- one-liners, Perl, 745-746
- oops (kernel), compiling kernel, 476
- Open Sound System (OSS), 94
- open source, proprietary drivers versus, 27
- OpenDocument, 87
- OpenLDAP, 627
- OpenShot Video Editor, 117
- OpenSSH server, 421
- openssh-server package, 639
- OpenStack, 654, 657-660
 - Compute Infrastructure (Nova), 658-659
 - Dashboard (Horizon), 659
 - Identity Service (Keystone), 659
 - Imaging Service (Glance), 659
 - Networking Service (Neutron), 659
 - Storage Infrastructure (Swift), 659
- Open-Xchange, 570
- operation, Postfix, 558-560
- operators
 - Perl, 732-735
 - additional operators, 734
 - arithmetic operators, 733-734
 - comparison operators, 732-733
 - compound operators, 733
 - PHP programming, 777-779
- opportunistic development, 695-710
 - Bikeshed, 707-709
 - Launchpad, 701-702
 - Quickly, 703-706
 - version control systems, 696-700
 - Bazaar, 697-698
 - Git, 699-700
 - Mercurial, 698-699
 - Subversion, 696-697
- optimization, 445-456
 - Apache, 451-452
 - hard disk, 445-449
 - badblocks command, 449
 - disabling file access time, 449
 - e2fsck command, 449
 - file system settings, 448
 - hdparm command, 447-448
 - tune2fs command, 448-449
 - tuning disk drives, 446-447
 - kernel, 450-451
 - MySQL, 452-456
 - measuring buffer usage, 452-454
 - query cache, 454-455
 - query optimization, 456
- Options directive, Apache web server configuration, 508
- Oracle Beehive, 570
- Oracle JDeveloper, 693
- order statement, Apache web server access control, 510
- organization, networking, 385-386
 - broadcast addressing, 386
 - multicast addressing, 386
 - subnet masks, 386
 - subnetting, 385-386
 - Unicast addressing, 386
- OrientDB, 624
- OSS (Open Sound System), 94
- Outlook, Microsoft Exchange Server, 569
- output, redirecting, 209-211, 277-278

P

- PaaS (Platform as a Service), 655
- Package Browsing screen (Software Center), 139
- package numbering, 150
- packaging code, 716-718
- packaging-dev package, installation, 713-714
- packet writing, creating DVDs from command line, 112
- PAM (Pluggable Authentication Modules), 256-257
- panel (Unity desktop), 63-64
- parameters, positional, executing shell scripts, 284-286
- partition strategies, Ubuntu installation, 13
- passwd command, 174, 247, 265
- password file, 253-254
- passwords, 19
 - management, 253-257
 - changing in a batch, 257
 - password file, 253-254
 - policy, 253
 - security, 257
 - shadow passwords, 254-257
 - MySQL root users, 602
 - security, 433-434
- pastebinit tool, 708
- patch command, 467-468
- patching the kernel, 467-468
- PATH environment variable, 223
- pattern-matching support (shells), 276-277
- pbget tool (Bikeshed), 707
- pbput tool (Bikeshed), 708
- pbputs tool (Bikeshed), 708
- pbuilder, 714
- pci=biosirq command, 447
- pconfiguration, system settings, power management, 29-30
- PCRE (Perl-Compatible Regular Expressions), 795
- PCX (graphics format), 104
- PDF, 87, 89
- pdksh shell, comparison of expressions, 292-301
- PDP-11 minicomputers, 802
- PEAR project, 798
- Percona Server, 589
- performance tuning, 445-456
 - Apache, 451-452
 - hard disk, 445-449
 - badblocks command, 449
 - disabling file access time, 449
 - e2fsck command, 449
 - file system settings, 448
 - hdparm command, 447-448
 - tune2fs command, 448-449
 - tuning disk drives, 446-447
- kernel, 450-451
- MySQL, 452-456
 - measuring buffer usage, 452-454
 - query cache, 454-455
 - query optimization, 456
- Perl (Practical Extraction and Report Language), 727-747
 - accessing the shell, 740
 - code examples, 741-746
 - command-line errors, 729
 - command-line processing, 746
 - commands, 746
 - conditional statements, 735-736
 - CPAN (Comprehensive Perl Archive Network), 741
 - data structures, 730-732
 - installation of CPAN module, 743
 - Linux, 727-730
 - looping constructs, 736-739

- modules, 741
- one-liners, 745-746
- operators, 732-735
 - additional operators, 734
 - arithmetic operators, 733-734
 - comparison operators, 732-733
 - compound operators, 733
- purging logs, 743-744
- regular expressions, 739
- sending mail, 741-743
- simple Perl program example, 728-730
- string constants, 734-735
- Usenet posts, 744-745
- variables, 730-732
- versions, 728
- Perl-Compatible Regular Expressions (PCRE), 795
- perldoc command, 730, 746
- permissions
 - command line, 169-175
 - assigning permissions, 170-171
 - chmod command, 172-173
 - directory permissions, 171-172
 - file permissions, 172-173
 - set UIDs/GID permissions, 173-175
 - file, 242-243
- personal home page programming. *See* PHP programming
- personal package archive (PPA), Launchpad, 702
- personal video recorders, 116
- photo management, Shotwell Photo Manager, 107
- Photoshop, 102
- PHP mode, entering/exiting, 770
- PHP programming, 769-800
 - arrays, 772-773
 - comments, 775
 - conditional statements, 779-780
 - constants, 774
 - databases, 797-800
 - entering/exiting PHP mode, 770
 - escape sequences, 775-776
 - functions, 786-797
 - arrays, 789-791
 - files, 791-793
 - miscellaneous functions, 793-797
 - strings, 786-789
 - HTML forms, 797
 - including other files, 785-786
 - loops, 783-785
 - operators, 777-779
 - references, 774-775
 - special operators, 780-781
 - switching, 781-783
 - variable substitution, 776-777
 - variables, 770-772
- PHP setup, Nginx, 534-535
- phpgroupware, 570
- PHPProjekt, 570
- physical security, 433-434
- PID (process ID), 315
- Pidgin, 78
- ping command, 376-378
- piping data
 - command line, 220-222
 - shell control, 278
- PiTiVi, 116
- Planet Debian, 77
- Planner (LibreOffice component), 86
- Platform as a Service (PaaS), 655
- Pluggable Authentication Modules (PAM), 256-257
- plug-ins, Firefox, 70-71
- PNG (graphics format), 104

- Point-to-Point Protocol over Ethernet, 414-416
- policy, password management, 253
- POP3 (Post Office Protocol version 3), 557
- populating directory, configuring LDAP server, 630-631
- portable anymap file format, 105
- portable bitmap file format, 105
- portable graymap file format, 105
- portable pixmap file format, 105
- ports, TCP/IP networking, 382
- positional parameters, executing shell scripts, 284-286
- POSIX Extended regular expressions, 795
- Post Office Protocol version 3 (POP3), 557
- postconf command, 559
- Postfix, 555
 - configuration and operation, 558-560
 - masquerading, 560-561
- PostgreSQL
 - client/server system, 607-613
 - command-line client, 612
 - configuration, 604-607
 - creating database, 605
 - deleting users, 606
 - versus MySQL, 598-601
 - users, 606
- posting to Usenet, Perl, 744-745
- post-installation configuration problems, 33-34
- POV-Ray, 106
- power management, configuring system settings, 29-30
- power shortcuts, Unity desktop, 66
- PPA (personal package archive), Launchpad, 702
- pppd daemon, 416
- PPPoE (Point-to-Point Protocol over Ethernet), 414-416
- Practical Extraction and Report Language. See Perl (Practical Extraction and Report Language)
- preg_match() function (PCRE), 795
- preg_match_all() function (PCRE), 795
- preg_replace() function (PCRE), 796
- previous commands, running, 235-236
- Prey, 434
- printenv command, 223, 259
- printers
 - configuring system settings, 29
 - sharing, 487, 490-491
 - avoiding support problems, 493
 - creating network printers, 490-491
 - CUPS, 490-492
- printing, commands, 494
- priority scheduling, console-based monitoring tools, 326-327
- private cloud, Eucalyptus, 657
- privileges
 - MySQL databases, 602
 - PostgreSQL databases, 607
- /proc directory, 164-165, 323
- procedural languages, MySQL versus PostgreSQL, 600-601
- process ID (PID), 315
- processes
 - limiting resource usage, 213-220
 - threads versus, 528
- Processes tab (System Monitor), 337
- Procmail, 567
- productivity applications, 83-92
 - GNOME Office, 87-88
 - KOffice, 87-89
 - LaTeX, 91
 - LibreOffice, 85-86
 - Microsoft Windows, 91-92
 - PDF, 89
 - XML and DocBook, 89-90

productivity suites, defined, 84

ProFTPD, 545

programming languages, 801-812

Ada, 802

Clojure, 803

COBOL, 803-804

D, 804

Dart, 806-807

Erlang, 804-805

Forth, 805

Fortran, 806

Go, 805-806

Groovy, 806

Haskell, 807

Java, 807-808

JavaScript, 808

Lisp, 808-809

Lua, 809

Mono, 809

OCaml, 810

Perl. See Perl (Practical Extraction and Report Language)

reasons for learning, 802

Ruby, 810

Rust, 811

Scala, 811

Scratch, 811

Vala, 811-812

programming tools, 683-694

autoconf utility, 687-688

C with Linux, 684-685

debugging tools, 688-689

GNU C compiler, 689-690

graphical development, 690-693

macros, 686-687

make command, 685

makefile targets, 686-687

makefiles, 685-687

programs

command line, 185-186

Ubuntu installation, 21

promiscuous mode, 345

proprietary drivers, open source versus, 27

proprietary video drivers, installation, 128-129

protection (security). See security

protection of data, MySQL versus PostgreSQL, 599-601

proxy servers, 573-587

access control lists, 575-579

client configuration, 574-575

client IP addresses, 579-580

defined, 573-574

sample configuration, 580-581

Squid installation, 574

ps command, 213-214, 323-325

public cloud, Eucalyptus, 657

PulseAudio, 94-95

Puppet, managing sets of servers, 670

Puppet Forge, 670

purging logs, Perl, 743-744

pwd command, 157, 169

PWD environment variable, 223

PyPI (Python Package Index), 767

Python, 703, 749-767

conditional statements, 759-761

dictionaries, 758-759

executing scripts, 750-751

functions, 761-762

lists, 756-758

loops, 759-761

numbers, 751-753

object orientation, 762-766

class and object variables, 763-764

class inheritance, 765-766

constructors and destructors, 764-765

PyPI (Python Package Index), 767
 Standard Library, 767
 strings, 753-756
 Python 2.x, 749-750
 Python 3.x, 749-750
 Python Package Index (PyPI), 767

Q

Q Public License, 3
 QA, 721-726
 Bug Squad, 723
 QA Team, 722-723
 Test Drive, 723-726
 QA Team, 722-723
 Qmail, 555
 QML, 824-825
 QT (video format), 114
 Quassel, 78
 query cache, tuning MySQL, 454-455
 query optimization, MySQL, 456
 Quickly, 703-706
 quickstart command, Juju, 665
 quotacheck command, 264
 quotaoff command, 264
 quotaon command, 264
 quotas, disk, 263-264

R

Radiance, 106
 RAID arrays, 352
 RAM disk image, configuring kernel, 474-475
 RAW (sound format), 96
 Raymond, Eric, 187

rcp command, 422
 RDBMSs (relational database management systems). See relational database services
 RDP (Remote Display Protocol), 650
 reading documentation, command line, 160-161
 recompiling the kernel, 464-465
 recorders, personal video, 116
 records (DNS), 676-679
 A, 676
 AAAA, 676
 CNAME, 677
 MX, 677
 NS, 677-678
 SOA, 678-679
 TXT, 679
 recovery mode, booting to, 182
 redirecting input and output
 command line, 209-211
 shell control, 277-278
 Redis, 620
 reduced instruction set computer (RISC) processors, 816
 redundant array of independent disks. See RAID arrays
 references, PHP programming, 774-775
 regular expressions, 179, 739
 regular users, 241
 granting system administrator privileges to, 258-263
 changing UID, 258-260
 root privileges, 260-263
 REISUB (system reset), 234-235
 relational database services, 589-613
 client/server system, 607-613
 commands, 613
 creating tables, 594-595
 how they work, 592-594
 inserting data in tables, 595-596

- MySQL
 - configuration, 601-604
 - future of, 589
 - versus PostgreSQL, 598-601
- PostgreSQL, 598-601, 604-607
- retrieving data, 596-598
- SQL basics, 594
- release tool (Bikeshed), 708
- release-build tool (Bikeshed), 708
- reload command (AppArmor), 442
- remote access, 419-428
 - SSH (Secure Shell), 421-425
 - Telnet, 419-420
 - VNC (virtual network computing), 425-427
- remote computer, logging in/out of command line, 157-158
- Remote Desktop Viewer, 426
- Remote Display Protocol (RDP), 650
- remote file serving (FTP), 543
 - configuring Very Secure FTP server, 548-551
 - FTP, server selection, 543-545
 - FTP users, 546-548
 - ftphosts file, 551-552
 - servers, 544-545
 - software installation, 545-546
- remote printing, 490
- removable storage media, 352-353
- renice command, 326
- repeat statement, executing shell scripts, 305
- repositories (software), 26-28
 - Apache web server installation, 496-498
 - compiling from source, 149-150
- reqquota command, 264
- Requests for Comments (RFC), 81
- requirements
 - hardware specifications, 10
 - LTSP, 636-639
 - rescue disc, 371
 - researching hardware specifications, 10
 - resource usage, limiting, 213-220
 - Resources tab (System Monitor), 337
 - responsibilities, DBAs (database administrators), 590
 - restoring
 - files from an archive, 356-357
 - GRUB2 boot loader, 371
 - retrieving
 - data, 596-598
 - email, 563-566
 - REVOKE statement
 - MySQL, 604
 - PostgreSQL, 607
 - revoking privileges, PostgreSQL databases, 607
 - RFC (Requests for Comments), 81
 - Rhythmbox, 97-98
 - Riak, 620
 - rights
 - MySQL databases, 602
 - PostgreSQL databases, 607
 - RISC (reduced instruction set computer) processors, 816
 - rm command, 177, 190, 206
 - rmdir command, 176
 - rmmmod command, 462
 - root users, 240-242
 - command line, 159, 180-185
 - MySQL, passwords, 602
 - root zone, 675
 - Rootkit Hunter, 443
 - rootsign tool, 709
 - rotating, log files, 331-336
 - route command, 396-398
 - routers, 391

RSS feeds, 76-77
 Firefox, 76
 Liferea, 76-77
 rsync command, 366-368
 rtcwake command, 272-274
 Ruby, 810
 runlevels (BIOS), 313, 316
 changing, 318-319
 definitions, 316
 troubleshooting problems, 319-320
 running jobs repeatedly, cron command, 270-272
 running services, boot process, 313-314
 run-one tool, 708
 run-this-one tool, 708
 runtime errors, compiling kernel, 476
 runtime server configuration settings (Apache web server), 498-499
 directives, 503-504
 editing apache2.conf, 504-506
 .htaccess configuration file, 507-509
 multiprocessing modules, 507
 Rust, 811

S

s2p command, 746
 SaaS (Software as a Service), 655
 Samba
 global behavior, 486-487
 manual configuration, 484-485
 mounting shares, 489
 smbd daemon, 488
 testing, 487
 satisfy directive, Apache web server, 513
 saving files, nonbooting hard drive, 372
 /sbin directory, 162-163
 Scala, 811
 scalar variables (Perl), 731
 scanners, 103
 scheduling tasks, 267-274
 running jobs repeatedly, 270-272
 waking computer from sleep, 272-274
 schemas, configuring LDAP server, 629
 Scorched 3D, 130
 scp command, 422
 Scratch, 811
 Screen, 78
 screen images, graphics manipulation, 105
 Screen section (xorg.conf file), 53, 56-57
 script kiddies, 430
 scripts
 command-line scripting. See Python
 Perl. See Perl (Practical Extraction and Report Language)
 shells. See shells
 web scripting. See PHP programming
 SDK (software development kit), 692-693, 816
 Android, installation, 817
 installation, 817, 824
 stderr stream, redirection, 211-212
 searches
 with Smart Scopes (Unity desktop), 65
 website search tips, 43-44
 Secure Shell. See SSH (Secure Shell)
 security, 429-444
 AppArmor, 440-442
 computer attacks, hacker versus cracker, 430
 disaster recovery plan, 442-443
 firewalls, 436-439

- machine protection, 432-435
 - devices, 345
 - passwords and physical security, 433-434
 - Tripwire, 434-435
 - wireless networks, 433
- passwords, 257
- viruses, 435-436
- vulnerability assessment, 431-432
- sed text editor, 230-232
- select statement, executing shell scripts, 305-306
- sending mail, Perl, 741-743
- Sendmail, 554-555
- separating, multiple commands, 236-237
- sequences (Python)
 - lists, 756-758
 - strings, 753-756
- Server install DVD, 10
- server packages, FTP, 544
- ServerAdmin directive, Apache web server configuration, 505
- ServerAGroup directive, Apache web server configuration, 505
- ServerLayout section (xorg.conf file), 53
- ServerName directive, Apache web server configuration, 506
- ServerRoot directive, Apache web server configuration, 504
- servers
 - Apache web server. See Apache web server configuration for LDAP, 628-631
 - DHCP, 406
 - DNS, 675-676, 679-681
 - FTP, 543-545
 - HTTP, 523-525
 - Apache Tomcat, 542
 - CA authority, 525
 - Cherokee, 541
 - Jetty, 541
 - lighttpd, 539-540
 - Nginx, 529-533
 - thttpd, 542
 - Yaws (Yet Another Web Server), 540
 - managing sets
 - Ansible, 671
 - CFEngine, 671
 - Chef, 670
 - Juju, 669-670
 - Landscape, 671
 - Puppet, 670
 - NFS, configuration, 480-481
 - SSH (Secure Shell), 421
 - Telnet, 419-420
- service models (Cloud), 655
- service set identifier (SSID), 32
- service settings, /etc/services file, 399-400
- services
 - BIOS (basic input/output system), controlling, 318
 - boot process, 313-314
 - session message block printing, 490-491
 - session writing, creating DVDs from command line, 111
 - sets of servers, management
 - Ansible, 671
 - CFEngine, 671
 - Chef, 670
 - Juju, 669-670
 - Landscape, 671
 - Puppet, 670
- sftp command, 421-423
- shadow passwords, 254-257, 547
- shared data, /usr directory, 165-166
- shares (Samba), mounting, 489

- sharing
 - files, 479
 - NFS (Network File System), 480-482
 - home directories, 486-487
 - printers, 487, 490-491
 - avoiding support problems, 493
 - creating network printers, 490-491
 - CUPS, 490-492
- shebang, 729
- shebang line, 282
- SHELL environment variable, 223
- Shell module, accessing the shell in Perl, 740
- shells, 274-311. *See also* command line
 - accessing in Perl, 740
 - basic shell control, 274-279
 - background processing, 278-279
 - command line, 275-276
 - pattern-matching support, 276-277
 - piping data, 278
 - redirecting input/output, 277-278
 - endless loops, 303
 - scripts, 279-311
 - functions, 310-311
 - writing and executing, 279-311
- shift statement, executing shell scripts, 306
- Shockwave, 70
- shortcuts
 - commands, 237
 - Unity desktop, 66
- Shotwell Photo Manager, 107
- shuffle() function (PHP), 789
- shutdown command, 20, 184-185
- shutting down Ubuntu, 20-21
- Shuttleworth, Mark, 59
- Simple Mail Transfer Protocol (SMTP), 553
- simple strategy, data backup, 350-351
- six-month cycle, development, 712-713
- slapd package, 628
- Slashdot, 77
- Slashdot Effect, 451
- sleep mode, waking computer from, 272-274
- small enterprise users, backup programs, 349
- small office users, backup programs, 349
- smart hosts, 561
- Smart Scopes, 65
- SMB (Session Message Block), 482
- smbclient command, 488-489
- smbd daemon, 488
- smbstatus command, 488
- Smokin' Guns, 129
- SMTP (Simple Mail Transfer Protocol), 553
- SOA record (DNS), 678-679
- socks-prox tool (Bikeshed), 708
- software
 - data backup, 354-363
 - DHCP, 405-406
 - installing FTP software, 545-546
 - licensing, 2-3
- Software as a Service (SaaS), 655
- Software Center, 139-140
- software development kit (SDK). *See* SDK (software development kit)
- software management, 44, 139-151
 - APT (Advanced Package Tool), 143-148
 - compiling applications from source, 148-150
 - configuration management, 150-151
 - Software Updater, 142-143
 - Synaptic, 140-142
 - Ubuntu Software Center, 139-140
- software repositories, 26-28
- Software Sources GUI tool, 26
- Software Updater, 22-25, 142-143
- sort command, 190, 206-208

sound, 93-100

- adjusting volume, 95
 - formats, 96-97
 - listening to music, 97-99
 - sound cards, 94-95
- sound cards, 94-95
- Sound Juicer, 99-100
- sound practices, backup programs, 348
- soundconverter package, 97
- SourceForge, 697
- sources (kernel), 466
- Spamassassin, 567-568
- special characters, writing shell scripts, 289
- special operators, PHP programming, 780-781
- special variables (Perl), 731-732
- speed, MySQL versus PostgreSQL, 598
- Speed Dreams, 134
- splint command, 688-689
- SQL, basics, 594
- SQL subqueries, MySQL versus PostgreSQL, 600
- Squid installation, 574
- Squirrelmail, 568
- SSH (Secure Shell), 78, 421-425
 - access to databases, 608-609
 - setting up server, 421
 - Telnet versus, 421
 - tools, 421-425
- ssh command, 190
- SSH key, 714
- sshd daemon, 421
- ssh-import-id tool, 709
- ssh-keygen command, 423-425
- SSID (service set identifier), 32
- Stallman, Richard M.35
- standard input, 210
- Standard Library, Python, 767

standard output, 210

- start command (AppArmor), 442
- starting
 - Apache web server, 500
 - NFS, 480
 - X Server sessions, 57
- Startup Applications, 318
- status command (AppArmor), 442
- stdin stream, redirection, 211-212
- stdout stream, redirection, 211-212
- Steam, 136
- step-by-step installation (Ubuntu), 14-20
 - first update, 20
 - installing, 15-19
- stereotypes (users), 243
- stop command (AppArmor), 442
- stopping
 - Apache web server, 500
 - NFS, 480
- Storage Infrastructure (Swift) service, OpenStack, 659
- storing scripts for access, 281-282
- str_replace() function (PHP), 787
- strategies, backup programs, 349-352
- stri_replace() function (PHP), 787
- string comparison
 - in pdksh and bash, 292-294
 - in tcsh, 297-298
- string constants, Perl, 734-735
- strings
 - with embedded spaces, 290
 - PHP functions, 786-789
 - Python, 753-756
- strlen() function (PHP), 786
- strpos() function (PHP), 788
- StumbleUpon plug-in, 71
- su command, 258-260, 265

- subdirectories, /usr directory, 51-52
- Sublevel number version (kernel), 465
- subnet masks, 386
- subnetting, 385-386
- subsections (kernel), configuring kernel, 472-474
- substr() function (PHP), 787-788
- Subversion, 696-697
- sudo command, 25, 180-183, 240-241, 265
- sudo group, admin group versus, 181, 262-263
- Super User/Root User, 240-242
- super users, 158-159, 180-185, 241
- SuperTux, 131-132
- support
 - network printers, 493-494
 - wireless networking, 409-410
- suspend, 18, 30
- SVG (graphics format), 104
- Swift (Storage Infrastructure) service, OpenStack, 659
- switch/case block, 782
- switches, 390-391
- switching
 - PHP programming, 781-783
 - virtual consoles, 155
- symbolic debugging, 689
- SYN flooding, 164
- Synaptic, software management, 140-142
- sync command, 448
- SysAdmin, DevOps versus, 653-654
- system administrators, granting privileges to regular users, 258-263
 - changing UID, 258-260
 - root privileges, 260-263
- system function call, accessing the shell in Perl, 740
- System Monitor, 336-338
- system rescue, 370-372

- system reset, command line, 234-235
- system services, BIOS (basic input/output system), 316
- system settings, configuration, 28-31
 - power management, 29-30
 - printers, 29
 - time and date, 30-31
 - Unity desktop, 64-66
- System tab (System Monitor), 336
- system users, 241
- systemd, 322
- system-monitoring tools. See monitoring tools
- system-search tool (Bikeshed), 708

T

- tables, relational database services, 594-596
- tail command, 190, 208-209, 329
- tape drives, 353-354
- tar command, 232, 355-357, 364-365
- tarballs, 148-149
- task automation, 267-274
 - running jobs repeatedly, 270-272
 - scheduling tasks, 267-274
 - waking computer from sleep, 272-274
 - writing shell scripts, 286-288
- TCP/IP (Transport Control Protocol/Internet Protocol), 378-382
 - addressing, 379-381
 - IP masquerading, 381
 - ports, 382
- tcsh shell, comparison of expressions, 297-301
- Telnet
 - setting up server, 419-420
 - SSH versus, 421
- temporary file storage, /tmp directory, 166
- TERM environment variable, 223

- terminal. See command line
 - ternary operator, 780-781
 - Test Drive tool, 723-726
 - testing, 721-726
 - Bug Squad, 723
 - community teams, 721-723
 - Samba, 487
 - Test Drive, 723-726
 - testparm command, testing Samba, 487
 - Texmaker, 91
 - text editors (command line), 226-232
 - emacs, 229-230
 - nano, 227-228
 - sed and awk, 230-232
 - vi, 228-229
 - text-based console login, 156-157
 - threads, processes versus, 528
 - thttpd, 542
 - Thunderbird, configuring for LDAP, 633
 - TIF (graphics format), 104
 - time, configuring system settings, 30-31
 - time command, 326-327
 - TLD (top-level domain), 675
 - /tmp directory, 166
 - Token Ring, 387
 - tools
 - group management, 245-246
 - SSH (Secure Shell), 421-425
 - system-monitoring. See monitoring tools
 - top command, 216-218, 327
 - top-level domain (TLD), 675
 - Torvalds, Linus, 35
 - Totem Movie Player, 114-115
 - touch command, 169, 175, 424
 - touchscreen interface, 823-824
 - application design guidelines, 825
 - creating applications, 824-825
 - SDK installation, 824
 - User Interface Toolkit, 825-826
 - traceroute tool, checking network connections, 376-378
 - Transport Control Protocol/Internet Protocol. See TCP/IP
 - triggers, MySQL versus PostgreSQL, 600-601
 - trim() function (PHP), 786
 - triple quoting, Python strings, 755
 - Tripwire, 434-435
 - troubleshooting
 - BIOS runlevel problems, 319-320
 - boot process, 322
 - Internet connection problems, 417-418
 - kernel compile, 475-476
 - networking, 391
 - post-installation configuration problems, 33-34
 - tty consoles, switching, 155
 - tune2fs command, 448-449
 - TuxPaint, 134
 - TV hardware, 112-114
 - TXT record (DNS), 679
 - types of kernels, 461-462
- ## U
- Ubuntu
 - business usage, 40-41
 - documentation, 41, 45
 - helping with development, 43
 - history, 39
 - home usage, 41
 - IRC (Internet Relay Chat), 48
 - mailing lists, 46-48
 - package numbering, 150
 - version numbers, 39

- websites, 43-46
 - commercial support, 44-45
 - Linux guides, 45-46
 - LUGs (Linux User Groups), 45
 - search tips, 43-44
 - Ubuntu-specific sites, 46
- Ubuntu Cloud. See Cloud
- Ubuntu GNOME, 124-125
- Ubuntu Kylin, 125
- Ubuntu Mobile/Touch, 823-824
 - application design guidelines, 825
 - creating applications, 824-825
 - SDK installation, 824
 - User Interface Toolkit, 825-826
- Ubuntu Server Edition, 654
- Ubuntu Software Center, 139-140
- Ubuntu Testing Team, 722
- ubuntu-qa-tools, 723
- ubuntu-restricted-extras package, 114
- UDP (Universal Datagram Protocol), 378
- UEFI (Unified Extensible Firmware Interface), 14
- UFW (Uncomplicated Firewall), 436-438
- UIDs (user IDs), 173-175, 242
- umask command, 169
- uname command, 36
- Uncomplicated Firewall (UFW), 436-438
- unexpanded variables, writing shell scripts, 290-291
- Unicast addressing, 386
- Unified Extensible Firmware Interface (UEFI), 14
- Unison backup tool, 362
- Unity desktop, 49-67
 - customizing and configuring, 64-66
 - Dash, 60-65
 - Launcher, 59-60
 - Mir, 50
 - panel, 63-64
 - power shortcuts, 66
- Ubuntu Mobile/Touch development, 823-824
 - application design guidelines, 825
 - creating applications, 824-825
 - SDK installation, 824
 - User Interface Toolkit, 825-826
- X Server, 49-58
 - basic concepts, 50-51
 - changing window managers, 58
 - display manager, 58
 - elements of xorg.conf file, 52-57
 - starting X sessions, 57
 - X.org software, 51-52
- Unity Tweak Tool, 65
- Universal Datagram Protocol (UDP), 378
- Universe repository, 26, 142
- unless conditional statements, Perl, 736
- UnQL (Unstructured Query Language), 617
- unset() function (PHP), 794
- Unshielded Twisted Pair (UTP) network cable, 389-390
- until looping construct (Perl), 738
- until statement, 304-305
- UPG (user private group), 243
- upgrade option, 25
- Upstart, 321-322
- uptime command, 328
- uquick tool (Bikeshed), 708
- Urban Terror, 129
- USB drive, Ubuntu installation, 11-12, 14
- Usenet newsgroups, 80-82
- Usenet posts, Perl, 744-745
- user accounts, 239-243
 - adding, 248-252
 - command line, 158-159
 - commands, 265
 - configuring Fetchmail, 565-566

- file permissions, 242-243
- stereotypes, 243
- Super User/Root User, 240-242
- user IDs/group IDs, 242
- User directive, Apache web server configuration, 505
- user directories, /home directory, 163-164
- USER environment variable, 223
- user files, Apache web server, 511-513
- user IDs (UIDs), 242
- User Interface Toolkit, 825-826
- user management, 239-266
 - disk quotas, 263-264
 - groups, 243-246
 - group listing, 243-244
 - management tools, 245-246
 - passwords, 253-257
 - changing in a batch, 257
 - password file, 253-254
 - policy, 253
 - security, 257
 - shadow passwords, 254-257
 - system administrator privileges, 258-263
 - root privileges, 260-263
 - temporarily changing user identity, 258-260
 - user accounts, 239-243
 - commands, 265
 - file permissions, 242-243
 - stereotypes, 243
 - Super User/Root User, 240-242
 - user IDs/group IDs, 242
- user private group (UPG), 243
- user variables, shell scripts, 283
- useradd command, 245-247, 265
- UserDir directive, Apache web server configuration, 506

- usermod command, 245, 248, 265
- users
 - FTP, 546-548
 - PostgreSQL, 606
- /usr directory
 - shared data, 165-166
 - subdirectories, 51-52
- /usr/bin subdirectory, 51
- /usr/include subdirectory, 51
- /usr/lib subdirectory, 51
- /usr/lib/modules subdirectory, 52
- /usr/lib/X11 subdirectory, 52
- /usr/src/linux-3.13 directory, 459
- UTP (Unshielded Twisted Pair) network cable, 389-390

V

- Vala, 811-812
- Valve Software, 136
- /var directory, 166
- var_dump() function (PHP), 794
- variable substitution, PHP programming, 776-777
- variables
 - class and object, Python object orientation, 763-764
 - executing shell scripts, 288-289
 - LogFormat statements (Apache), 522-523
 - Perl, 730-732
 - PHP programming, 770-772
 - shell scripts, 283-284
- variants (Ubuntu), 11
- VBA (Visual Basic for Applications), 83

- version control systems
 - data backup, 368-370
 - opportunistic development, 696-700
 - Bazaar, 697-698
 - Git, 699-700
 - Mercurial, 698-699
 - Subversion, 696-697
- versions
 - kernel, 465-466
 - Linux kernel, 36
 - Perl, 728
 - PHP programming, 769
 - Ubuntu, 39
 - package numbering, 150
- Very Secure FTP server, configuration, 548-551
- vi text editor, 228-229, 746
- video
 - editing, 116-117
 - formats, 114
 - hardware, 112-114
 - personal video recorders, 116
 - viewing, 112-117
- video conferencing, Empathy, 77-78
- viewing, command history, 236
- viewing video, 112-117
- vim command, 190
- vim text editor, 226
- virtual consoles, switching, 155
- virtual devices, installation, 819
- virtual file system, 164
- virtual hosting
 - Apache web server, 519-521
 - Nginx, 533-534
- virtual machines (VMs), 644, 655
- virtual network computing (VNC), 425-427
- virtual private networks. See VPNs (virtual private networks)
 - VirtualBox, 649-650
- virtualization, 643-651
 - KVM (Kernel-based Virtual Machine), 645-649
 - VirtualBox, 649-650
 - VMware, 651
 - Xen, 651
- virus scanners, 436, 568
- viruses, 435-436
- Visual Basic for Applications (VBA), 83
- VLC, 115
- VMs (virtual machines), 644, 655
- vmstat tool, 328
- VMware, 651
- VNC (virtual network computing), 425-427
- vncviewer, 342
- volume adjustment, music and sound, 95
- VPN clients, setting up, 583-584
- VPN servers, setting up, 585-587
- VPNs (virtual private networks), 582-587
 - setting up clients, 582-587
 - setting up servers, 582-587
- vsftpd server configuration files, 550
- VT-x extension, 645
- vulnerability assessment, 431-432

W

- waking computer from sleep, 272-274
- war driving, 433
- Warsow, 129-130
- watch command, 328
- WAV (sound format), 96
- web access, databases, 610
- web scripting. See PHP programming

- web servers, 523-525
 - Apache. See Apache web server
 - Apache Tomcat, 542
 - CA authority, 525
 - Cherokee, 541
 - Jetty, 541
 - lighttpd, 523-525, 539-540
 - Nginx, 527-529
 - configuration, 530-533
 - installation, 529-530
 - modules, 536
 - PHP setup, 534-535
 - virtual hosting, 533-534
 - thttpd, 542
 - Yaws (Yet Another Web Server), 540
- web-based email applications, 76
- WEBM (video format), 114
- websites, 43-46
 - commercial support, 44-45
 - Linux guides, 45-46
 - LUGs (Linux User Groups), 45
 - search tips, 43-44
 - Ubuntu-specific sites, 46
- WEP encryption, 33
- what-provides tool (Bikeshed), 708
- whereis command, 161
- which command, 190, 209
- while loop
 - Perl, 737-738
 - PHP, 783
 - Python, 760
- while statement, 303-304
- wide column stores, NoSQL databases, 623-624
- wifi-status tool (Bikeshed), 708
- wildcards, 179
- window managers, X Server, 58
- Windows games, 136-137
- Wine application, 91-92, 136
- wireless network interfaces, 388-389
- wireless networks, 409-412
 - advantages, 411
 - choosing protocol, 411-412
 - configuration, 32-33
 - security, 433
- Wireshark, 343, 433, 547
- Wolfenstein: Enemy Territory*, 129
- World of Padman*, 129
- worms, 430
- WPA encryption, 33
- Writer (LibreOffice component), 85
- writing shell scripts, 279-311
 - accessing variable values, 284
 - assigning value to variables, 284
 - automation of tasks, 286-288
 - backslash, 291
 - backtick, 292
 - break statement, 310
 - built-in variables, 288-289
 - case statement, 308-309
 - comparison of expressions in pdksh and bash, 292-301
 - comparison of expressions with tcsh, 297-301
 - exit statement, 310
 - for statement, 301-302
 - if statement, 306-307
 - interpreting shell scripts, 282-283
 - positional parameters, 284-286
 - repeat statement, 305
 - running shell program, 280-281
 - select statement, 305-306
 - shift statement, 306
 - special characters, 289

- storing scripts for access, 281-282
- strings with embedded spaces, 290
- unexpanded variables, 290-291
- until statement, 304-305
- variables, 283-284
- while statement, 303-304

WYGIWYW (What You Get Is What You Want), 91

WYSIWYG (What You See Is What You Get), 91

X

X Server, 49-58

- basic concepts, 50-51
- changing window managers, 58
- display manager, 58
- elements of xorg.conf file, 52-57
 - Device section, 53, 56
 - Files section, 53-54
 - InputDevice section, 53, 55
 - Module section, 53-54
 - Monitor section, 53, 55-56
 - Screen section, 53, 56-57
 - ServerLayout section, 53
- starting X sessions, 57
- X.org software, 51-52

Xamarin, 809

Xara Xtreme, 106

xargs command, 190

XChat, 78-79

XChat Channel List window, 79

xconfig tool, 471-474

Xen, 651

Xfce, 122-123

Xmarks Sync plug-in, 71

XML and DocBook, 89-90

XML Copy Editor, 90

X.org software, 51-52

xorg.conf file elements, 52-57

- Device section, 53, 56
- Files section, 53-54
- InputDevice section, 53-55
- Module section, 53-54
- Monitor section, 53, 55-56
- Screen section, 53, 56-57
- ServerLayout section, 53

Xubuntu, 122-123

Y

Yahoo! Mail, 76

Yaws (Yet Another Web Server), 540

Z

Zenoss, 343

zless command, 42