

Matthew Helmke



Ubuntu 13.10
on DVD

Ubuntu

UNLEASHED



SAMS

2014 Edition
Covering 13.10 and 14.04

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Matthew Helmke
with Andrew Hudson
and Paul Hudson

Ubuntu

UNLEASHED

2014 Edition



800 East 96th Street, Indianapolis, Indiana 46240 USA

Ubuntu Unleashed 2014 Edition

Copyright © 2014 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: 978-0-672-33693-5

ISBN-10: 0-672-33693-6

Library of Congress Control Number: 2013947841

Printed in the United States of America

First Printing November 2013

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.

Bulk Sales

Sams Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact

International Sales
international@pearsoned.com

Editor-in-Chief

Mark Taub

Acquisitions Editor

Debra Williams
Cauley

Managing Editor

Kristy Hart

Project Editor

Andy Beaster

Copy Editor

Charlotte Kughen

Indexer

Cheryl Lenser

Proofreader

Williams Woods
Publishing Services

Technical Editor

Micah Brown

Editorial Assistant

Kim Boedigheimer

Media Producer

Dan Scherf

Cover Designer

Anne Jones

Compositor

Gloria Schurick

Contents at a Glance

Introduction	1
Part I Installation and Configuration	
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	233
13 Automating Tasks and Shell Scripting	261
14 The Boot Process	307
15 System-Monitoring Tools	317
16 Backing Up	335
17 Networking	363
18 Remote Access with SSH and Telnet	409
19 Securing Your Machines	419
20 Performance Tuning	435
21 Kernel and Module Management	447
Part IV Ubuntu as a Server	
22 Sharing Files and Printers	469
23 Apache Web Server Management	491
24 Nginx Web Server Management	521
25 Other Http Servers	531

26	Remote File Serving with FTP	535
27	Handling Email	545
28	Proxying, Reverse Proxying, and Virtual Private Networks (VPN)	565
29	Administering Relational Database Services	581
30	NoSQL Databases	607
31	Lightweight Directory Access Protocol (LDAP)	619
32	Linux Terminal Server Project (LTSP)	627
33	Virtualization on Ubuntu	635
34	Ubuntu in the Cloud	645
35	Managing Sets of Servers	681
36	Name Serving with the Domain Name System (DNS)	685
 Part V Programming Linux		
37	Opportunistic Development	695
38	Helping with Ubuntu Development	715
39	Helping with Ubuntu Testing and QA	725
40	Using Perl	731
41	Using Python	753
42	Using PHP	773
43	C/C++ Programming Tools for Ubuntu	805
44	Using Other Popular Programming Languages	817
45	Beginning Mobile Development for Android	829
46	Developing for Ubuntu Mobile/Touch	837
 Index		841

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.....	12
The Boot Loader	13
Installing from DVD or USB Drive	14
Step-by-Step Installation.....	14
Installing.....	15
First Update	19
Shutting Down	19
Finding Programs and Files	20
Software Updater	21
The <code>sudo</code> Command	24
Configuring Software Repositories.....	25
System Settings	27
Detecting and Configuring a Printer	28
Configuring Power Management in Ubuntu.....	29
Setting the Time and Date	29
Configuring Wireless Networks.....	31
Troubleshooting Post-Installation Configuration Problems.....	33
References	34

2 Background Information and Resources	35
What Is Linux?	35
Why Use Linux?	36
What Is Ubuntu?	38
Ubuntu for Business	39
Ubuntu in Your Home	40
Getting the Most from Ubuntu and Linux Documentation	40
Ubuntu Developers and Documentation.....	42
Websites and Search Engines	42
Web Search Tips	42
Google Is Your Friend	43
Ubuntu Package Listings.....	43
Commercial Support	44
Documentation	44
Linux Guides	44
Ubuntu	45
Mailing Lists	46
Ubuntu Project Mailing Lists	46
Internet Relay Chat	47
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.....	65
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
Ubuntu One Cloud Storage.....	82
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.....	130
Scorched 3D.....	130
Frozen Bubble.....	131
SuperTux.....	132
Battle for Wesnoth	132
Frets on Fire	134
FlightGear	134
Speed Dreams	134
Games for Kids	134
Commercial Games	135
Playing Windows Games.....	136
References	136

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>	190
Changing Directories with <code>cd</code>	191
Changing File Access Permissions with <code>chmod</code>	193
Copying Files with <code>cp</code>	193
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
Listing Processes with <code>ps</code>	206
Deleting Files and Directories with <code>rm</code>	206

Printing the Last Lines of a File with <code>tail</code>	207
Printing Resource Usage with <code>top</code>	207
Using <code>echo</code>	209
Printing the Location of a Command with <code>which</code>	210
Redirecting Output and Input.....	210
<code>stdin</code> , <code>stdout</code> , <code>sdterr</code> , and Redirection.....	211
Comparing Files.....	212
Finding Differences in Files with <code>diff</code>	213
Finding Similarities in Files with <code>comm</code>	213
Combining Commands.....	214
Pipes.....	214
Running One or More Tasks in the Background.....	216
Combining Commands with Boolean Operators.....	216
Running Separate Commands in Sequence.....	217
Using Environment Variables	217
Using Common Text Editors.....	220
Working with <code>nano</code>	222
Working with <code>vi</code>	222
Working with <code>emacs</code>	223
Working with <code>sed</code> and <code>awk</code>	225
Working with Compressed Files	227
Using Multiple Terminals with <code>byobu</code>	227
Polite System Reset using REISUB	229
Limiting Resource Use.....	230
References	231
12 Managing Users 233	
User Accounts	233
The Super User/Root User	234
User IDs and Group IDs	236
File Permissions	236
Managing Groups	237
Group Listing	237
Group Management Tools	238
Managing Users	240
User Management Tools	240
Adding New Users	242
Monitoring User Activity on the System.....	246
Managing Passwords.....	246
System Password Policy.....	246
The Password File	247
Shadow Passwords.....	248

Managing Password Security for Users	250
Changing Passwords in a Batch	251
Granting System Administrator Privileges to	
Regular Users	251
Temporarily Changing User Identity with the <code>su</code> Command	251
Granting Root Privileges on Occasion: The <code>sudo</code> Command.....	253
Disk Quotas.....	256
Implementing Quotas.....	257
Manually Configuring Quotas	257
Related Ubuntu Commands.....	258
References	259
13 Automating Tasks and Shell Scripting	261
Scheduling Tasks.....	261
Using <code>at</code> and <code>batch</code> to Schedule Tasks for Later	261
Using <code>cron</code> to Run Jobs Repeatedly	264
Using <code>rtcwake</code> to Wake Your Computer from	
Sleep Automatically	266
Basic Shell Control	268
The Shell Command Line	269
Shell Pattern-Matching Support.....	270
Redirecting Input and Output	271
Piping Data	272
Background Processing.....	272
Writing and Executing a Shell Script	273
Running the New Shell Program	274
Storing Shell Scripts for System-wide Access	275
Interpreting Shell Scripts Through Specific Shells.....	276
Using Variables in Shell Scripts.....	277
Assigning a Value to a Variable	278
Accessing Variable Values	278
Positional Parameters	278
A Simple Example of a Positional Parameter.....	279
Using Positional Parameters to Access and Retrieve Variables	
from the Command Line.....	280
Using a Simple Script to Automate Tasks	280
Built-In Variables.....	282
Special Characters	283
Using Double Quotes to Resolve Variables in Strings	
with Embedded Spaces	284
Using Single Quotes to Maintain Unexpanded Variables.....	284
Using the Backslash as an Escape Character	285

Using the Backtick to Replace a String with Output.....	286
Comparison of Expressions in <code>pdksh</code> and <code>bash</code>	286
Comparing Expressions with <code>tcsh</code>	291
The <code>for</code> Statement.....	295
The <code>while</code> Statement.....	297
The <code>until</code> Statement.....	298
The <code>repeat</code> Statement (<code>tcsh</code>)	299
The <code>select</code> Statement (<code>pdksh</code>)	299
The <code>shift</code> Statement.....	300
The <code>if</code> Statement.....	300
The <code>case</code> Statement.....	302
The <code>break</code> and <code>exit</code> Statements.....	304
Using Functions in Shell Scripts	304
References	305
14 The Boot Process	307
Running Services at Boot.....	307
Beginning the Boot Loading Process.....	308
Loading the Linux Kernel	309
System Services and Runlevels	310
Runlevel Definitions	310
Booting into the Default Runlevel.....	311
Understanding <code>init</code> Scripts and the Final Stage of Initialization....	311
Controlling Services at Boot with Administrative Tools.....	312
Changing Runlevels	312
Troubleshooting Runlevel Problems.....	313
Starting and Stopping Services Manually	314
Using Upstart.....	315
References	316
15 System-Monitoring Tools	317
Console-Based Monitoring	317
Using the <code>kill</code> Command to Control Processes.....	319
Using Priority Scheduling and Control	320
Displaying Free and Used Memory with <code>free</code>	323
Disk Space.....	323
Disk Quotas	324
Checking Log Files	324
Graphical Process and System Management Tools.....	326
System Monitor	326
Conky	328
Other.....	333

KDE Process- and System-Monitoring Tools.....	333
Enterprise Server Monitoring	333
Landscape	334
Other.....	334
References	334
16 Backing Up	335
Choosing a Backup Strategy	335
Why Data Loss Occurs	336
Assessing Your Backup Needs and Resources	337
Evaluating Backup Strategies.....	339
Making the Choice.....	342
Choosing Backup Hardware and Media.....	342
Removable Storage Media	342
CD-RW and DVD+RW/-RW Drives.....	343
Network Storage	343
Tape Drive Backup	343
Cloud Storage	344
Using Backup Software	344
tar: The Most Basic Backup Tool	345
The GNOME File Roller.....	347
The KDE ark Archiving Tool.....	348
Déjà Dup.....	348
Back In Time.....	350
Unison	352
Using the Amanda Backup Application.....	352
Alternative Backup Software	353
Copying Files	354
Copying Files Using tar.....	354
Compressing, Encrypting, and Sending tar Streams	355
Copying Files Using cp.....	355
Copying Files Using mc.....	356
Using rsync	356
Version Control for Configuration Files.....	358
System Rescue	360
The Ubuntu Rescue Disc	361
Restoring the GRUB2 Boot Loader.....	361
Saving Files from a Nonbooting Hard Drive	362
References	362

17 Networking	363
Laying the Foundation: The <code>localhost</code> Interface.....	364
Checking for the Availability of the Loopback Interface	364
Configuring the Loopback Interface Manually	364
Checking Connections with <code>ping</code> , <code>traceroute</code> , and <code>mtr</code>	366
Networking with TCP/IP	368
TCP/IP Addressing.....	369
Using IP Masquerading in Ubuntu	371
Ports.....	372
IPv6 Basics.....	372
Network Organization	375
Subnetting	375
Subnet Masks.....	376
Broadcast, Unicast, and Multicast Addressing.....	376
Hardware Devices for Networking	377
Network Interface Cards	377
Network Cable.....	379
Hubs and Switches	380
Routers and Bridges.....	381
Initializing New Network Hardware	381
Using Network Configuration Tools	383
Command-Line Network Interface Configuration.....	384
Network Configuration Files.....	389
Using Graphical Configuration Tools.....	391
Dynamic Host Configuration Protocol	393
How DHCP Works.....	393
Activating DHCP at Installation and Boot Time.....	394
DHCP Software Installation and Configuration.....	395
Using DHCP to Configure Network Hosts.....	397
Other Uses for DHCP	399
Wireless Networking.....	399
Support for Wireless Networking in Ubuntu.....	399
Advantages of Wireless Networking	401
Choosing from Among Available Wireless Protocols.....	401
Beyond the Network and onto the Internet.....	402
Common Configuration Information.....	402
Configuring Digital Subscriber Line Access	404
Understanding PPP over Ethernet	404
Configuring a PPPoE Connection Manually	405
Configuring Dial-Up Internet Access	406
Troubleshooting Connection Problems	407
References	408

18 Remote Access with SSH and Telnet	409
Setting Up a Telnet Server	409
Telnet Versus SSH	411
Setting Up an SSH Server.....	411
SSH Tools	411
Using <code>scp</code> to Copy Individual Files Between Machines	412
Using <code>sftp</code> to Copy Many Files Between Machines.....	413
Using <code>ssh-keygen</code> to Enable Key-Based Logins	413
Virtual Network Computing	415
References	417
19 Securing Your Machines	419
Understanding Computer Attacks.....	419
Assessing Your Vulnerability	421
Protecting Your Machine	422
Securing a Wireless Network.....	423
Passwords and Physical Security	423
Configuring and Using Tripwire.....	424
Devices.....	425
Viruses.....	425
Configuring Your Firewall.....	426
AppArmor	430
Forming a Disaster Recovery Plan.....	432
References	433
20 Performance Tuning	435
Hard Disk	435
Using the BIOS and Kernel to Tune the Disk Drives	436
The <code>hdparm</code> Command	437
File System Tuning.....	438
The <code>tune2fs</code> Command	438
The <code>e2fsck</code> Command	439
The <code>badblocks</code> Command.....	439
Disabling File Access Time	439
Kernel.....	440
Apache	441
MySQL	442
Measuring Key Buffer Usage	442
Using the Query Cache	444
Miscellaneous Tweaks	445
Query Optimization	446
References	446

21 Kernel and Module Management	447
The Linux Kernel	448
The Linux Source Tree	449
Types of Kernels	451
Managing Modules	452
When to Recompile.....	454
Kernel Versions.....	455
Obtaining the Kernel Sources.....	456
Patching the Kernel	457
Compiling the Kernel.....	458
Using <code>xconfig</code> to Configure the Kernel.....	461
Creating an Initial RAM Disk Image.....	464
When Something Goes Wrong	465
Errors During Compile	465
Runtime Errors, Boot Loader Problems, and Kernel Oops	466
References	466
Part IV Ubuntu as a Server	
22 Sharing Files and Printers	469
Using the Network File System	470
Installing and Starting or Stopping NFS	470
NFS Server Configuration.....	470
NFS Client Configuration	472
Putting Samba to Work	473
Manually Configuring Samba with <code>/etc/samba/smb.conf</code>	474
Testing Samba with the <code>testparm</code> Command	477
Starting, Stopping, and Restarting the <code>smbd</code> Daemon	478
Mounting Samba Shares.....	479
Configuring Samba Using SWAT	480
Network and Remote Printing with Ubuntu	483
Creating Network Printers	484
Using the Common UNIX Printing System GUI	486
Avoiding Printer Support Problems	488
References	489
23 Apache Web Server Management	491
About the Apache Web Server	491
Installing the Apache Server.....	492
Installing from the Ubuntu Repositories	493
Building the Source Yourself.....	494

Starting and Stopping Apache.....	497
Starting the Apache Server Manually	497
Using /etc/init.d/apache2	498
Runtime Server Configuration Settings	499
Runtime Configuration Directives.....	500
Editing apache2.conf	500
Apache Multiprocessing Modules.....	503
Using .htaccess Configuration Files	503
File System Authentication and Access Control.....	505
Restricting Access with allow and deny.....	506
Authentication	507
Final Words on Access Control.....	509
Apache Modules	510
mod_access	510
mod_alias	511
mod_asis	511
mod_auth	511
mod_auth_anon.....	512
mod_auth_dbm.....	512
mod_auth_digest	512
mod_autoindex.....	512
mod_cgi	512
mod_dir and mod_env.....	512
mod_expires.....	513
mod_headers.....	513
mod_include.....	513
mod_info and mod_log_config.....	513
mod_mime and mod_mime_magic.....	513
mod_negotiation.....	513
mod_proxy	514
mod_rewrite.....	514
mod_setenvif.....	514
mod_speling.....	514
mod_status	514
mod_ssl.....	514
mod_unique_id.....	515
mod_userdir.....	515
mod_usertrack.....	515
mod_vhost_alias	515
Virtual Hosting	515
Address-Based Virtual Hosts.....	515
Name-Based Virtual Hosts.....	516

Logging	517
References	519
24 Nginx Web Server Management	521
About the Nginx Web Server	521
Installing the Nginx Server	523
Installing from the Ubuntu Repositories	523
Building the Source Yourself	524
Configuring the Nginx Server	524
Virtual Hosting	527
Setting Up PHP	528
Adding and Configuring Modules	530
References	530
25 Other HTTP Servers	531
lighttpd	531
Yaws	532
Cherokee	533
Jetty	533
thttpd	534
Apache Tomcat	534
References	534
26 Remote File Serving with FTP	535
Choosing an FTP Server	535
Choosing an Authenticated or Anonymous Server	536
Ubuntu FTP Server Packages	536
Other FTP Servers	536
Installing FTP Software	537
The FTP User	538
Configuring the Very Secure FTP Server	540
Controlling Anonymous Access	541
Other <code>vsftpd</code> Server Configuration Files	542
Using the <code>ftphosts</code> File to Allow or Deny FTP Server Connection	543
References	544
27 Handling Email	545
How Email Is Sent and Received	545
The Mail Transport Agent	546
Choosing an MTA	548
The Mail Delivery Agent	548
The Mail User Agent	549

Basic Postfix Configuration and Operation.....	550
Configuring Masquerading	552
Using Smart Hosts	553
Setting Message Delivery Intervals	553
Mail Relaying.....	554
Forwarding Email with Aliases.....	554
Using Fetchmail to Retrieve Mail.....	555
Installing Fetchmail	555
Configuring Fetchmail	555
Choosing a Mail Delivery Agent	559
Procmail.....	559
Spamassassin	559
Squirrelmail	560
Virus Scanners	560
Autoresponders.....	560
Alternatives to Microsoft Exchange Server	560
Microsoft Exchange Server/Outlook Client	561
CommuniGate Pro	561
Oracle Beehive.....	562
Bynari	562
Open-Xchange.....	562
phpgroupware	562
PHPProjekt.....	562
Horde	562
References	563
28 Proxying, Reverse Proxying, and Virtual Private Networks (VPN)	565
What Is a Proxy Server?.....	565
Installing Squid.....	566
Configuring Clients.....	566
Access Control Lists.....	567
Specifying Client IP Addresses	571
Sample Configurations.....	572
Virtual Private Networks (VPN).....	574
Setting Up a VPN Client	575
Setting Up a VPN Server.....	577
References	579
29 Administering Relational Database Services	581
A Brief Review of Database Basics	582
How Relational Databases Work.....	584
Understanding SQL Basics	586

Creating Tables.....	586
Inserting Data into Tables.....	587
Retrieving Data from a Database	588
Choosing a Database: MySQL Versus PostgreSQL.....	590
Speed.....	590
Data Locking	591
ACID Compliance in Transaction Processing to Protect Data Integrity	591
SQL Subqueries.....	592
Procedural Languages and Triggers.....	592
Configuring MySQL.....	593
Setting a Password for the MySQL Root User.....	594
Creating a Database in MySQL	594
Configuring PostgreSQL	596
Initializing the Data Directory in PostgreSQL.....	596
Creating a Database in PostgreSQL.....	597
Creating Database Users in PostgreSQL.....	598
Deleting Database Users in PostgreSQL.....	598
Granting and Revoking Privileges in PostgreSQL.....	599
Database Clients	599
SSH Access to a Database	600
Local GUI Client Access to a Database	601
Web Access to a Database	602
The MySQL Command-Line Client.....	603
The PostgreSQL Command-Line Client.....	604
Graphical Clients.....	605
References	605
30 NoSQL Databases	607
Key/Value Stores	610
Berkeley DB	610
Cassandra.....	611
Memcached and MemcacheDB.....	611
Redis	612
Riak	612
Document Stores	612
CouchDB	613
MongoDB	614
BaseX	614
Wide Column Stores	615
BigTable	615
HBase	615

Graph Stores	616
Neo4j	616
OrientDB.....	616
HyperGraphDB.....	616
FlockDB.....	617
References	617
31 Lightweight Directory Access Protocol (LDAP)	619
Configuring the Server	620
Creating Your Schema.....	620
Populating Your Directory	622
Configuring Clients.....	624
Evolution	624
Thunderbird	625
Administration.....	625
References	626
32 Linux Terminal Server Project (LTSP)	627
Requirements	628
Installation.....	631
Using LTSP	632
References	633
33 Virtualization on Ubuntu	635
KVM	637
VirtualBox.....	641
VMware.....	643
Xen.....	643
References	643
34 Ubuntu in the Cloud	645
Why a Cloud?.....	646
Software as a Service (SaaS)	647
Platform as a Service (PaaS).....	647
Infrastructure as a Service (IaaS)	647
Metal as a Service (MaaS)	647
Before You Do Anything.....	648
Ubuntu Cloud and Eucalyptus.....	648
Deploy/Install Basics: Public or Private?.....	650
Public	650
Private.....	651
A euca2ools Primer.....	654

Ubuntu Cloud and OpenStack.....	656
Compute Infrastructure (Nova)	656
Storage Infrastructure (Swift)	657
Imaging Service (Glance)	657
Installation	657
Creating an Image.....	667
Instance Management.....	670
Storage Management.....	671
Network Management.....	671
An OpenStack Commands Primer.....	672
Learning More	672
Juju.....	672
Getting Started	673
Charms	676
The Juju GUI	677
Juju on Mac OS X.....	678
Ubuntu Metal as a Service (MaaS).....	678
Landscape	679
References	679
35 Managing Sets of Servers	681
Juju.....	681
Puppet.....	682
Chef.....	682
CFEngine.....	683
Ansible	683
Landscape	683
References	683
36 Name Serving with the Domain Name System (DNS)	685
Understanding Domain Names.....	687
DNS Servers	687
DNS Records	688
Setting Up a DNS Server with BIND	691
References	693
Part V Programming Linux	
37 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
Ground Control.....	707
Bikeshed and Other Tools	711
References	713
38 Helping with Ubuntu Development	715
Introduction to Ubuntu Development	716
Setting Up Your Development System.....	717
Install Basic Packages and Configure.....	717
Create a Launchpad Account.....	718
Set Up Your Environment to Work with Launchpad	718
Fixing Bugs and Packaging.....	720
Finding Bugs to Fix with Harvest.....	723
Masters of the Universe.....	723
References	723
39 Helping with Ubuntu Testing and QA	725
Community Teams.....	725
Ubuntu Testing Team	726
QA Team.....	726
Bug Squad	727
Test Drive.....	727
References	730
40 Using Perl	731
Using Perl with Linux.....	731
Perl Versions.....	732
A Simple Perl Program	732
Perl Variables and Data Structures	734
Perl Variable Types	735
Special Variables.....	735
Operators	736
Comparison Operators	736
Compound Operators	737
Arithmetic Operators.....	737
Other Operators.....	738
Special String Constants.....	738

Conditional Statements: if/else and unless	739
if	739
unless.....	740
Looping.....	740
for	740
foreach.....	741
while.....	741
until.....	742
last and next.....	742
do ... while and do ... until	742
Regular Expressions	743
Access to the Shell	744
Modules and CPAN	745
Code Examples	745
Sending Mail.....	745
Purging Logs.....	747
Posting to Usenet	748
One-Liners.....	749
Command-Line Processing	750
References	750
41 Using Python	753
Python on Linux	754
The Basics of Python	755
Numbers	755
More on Strings.....	757
Lists.....	760
Dictionaries	762
Conditionals and Looping	763
Functions	765
Object Orientation	766
Class and Object Variables.....	767
Constructors and Destructors	768
Class Inheritance.....	769
The Standard Library and the Python Package Index	771
References	771
42 Using PHP	773
Introduction to PHP	774
Entering and Exiting PHP Mode.....	774
Variables	774
Arrays.....	776

Constants.....	778
References.....	778
Comments	779
Escape Sequences.....	779
Variable Substitution.....	780
Operators	781
Conditional Statements	783
Special Operators.....	784
Switching.....	785
Loops	787
Including Other Files	789
Basic Functions.....	790
Strings.....	790
Arrays.....	793
Files.....	795
Miscellaneous	797
Handling HTML Forms.....	801
Databases	801
References	804
43 C/C++ Programming Tools for Ubuntu	805
Programming in C with Linux.....	806
Using the C Programming Project Management	
Tools Provided with Ubuntu.....	807
Building Programs with <code>make</code>	807
Using Makefiles	807
Using the <code>autoconf</code> Utility to Configure Code	809
Debugging Tools.....	810
Using the GNU C Compiler	811
Graphical Development Tools	812
Using the KDevelop Client	812
The Glade Client for Developing in GNOME	813
References	814
Part VI Appendices	
44 Using Other Popular Programming Languages	817
Ada	818
Clojure	819
COBOL.....	819
D.....	820
Erlang.....	820
Forth.....	821

Go	821
Fortran	822
Groovy	822
Haskell.....	822
Java.....	823
JavaScript	823
Lisp.....	824
Lua	824
Mono.....	825
Ruby	825
Rust	826
Scala	826
Scratch	826
Vala	827
References	827
45 Beginning Mobile Development for Android	829
Introduction to Android.....	830
Hardware	830
Linux Kernel.....	830
Libraries	830
Android Runtime.....	830
Application Framework.....	830
Applications.....	831
Installing the Android SDK	831
Install Java	831
Install Eclipse.....	831
Install the SDK.....	831
Install the ADT Eclipse Plug-In	832
Install Other Components	832
Install Virtual Devices	833
Create Your First Application.....	834
References	835
46 Developing for Ubuntu Mobile/Touch	837
Install the SDK.....	838
Create Your First Application.....	838
Learn About Ubuntu Design	839
Study the User Interface Toolkit	839
References	840
Index	841

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 and Information 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 Heather, Saralyn, Sedona, and Philip—the most amazing family
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 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.

IN THIS INTRODUCTION

Introduction

We are pleased to present the 2014 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

- ▶ 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. The note about a free upgrade to 14.04 is true in that we will send an upgrade kit to anyone who has purchased and registered the book. The kit includes install media for the 14.04 release and a supplemental guide to the changes from 13.10 to 14.04.

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 nearly 20 programming languages are covered in about a dozen pages. These are useful topics to some, but not topics I would consider vital.

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

Those Wanting to Become Intermediate or Advanced Users

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

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

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

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

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

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

Sysadmins, Programmers, and DevOps

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

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

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

What This Book Contains

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

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

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

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

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

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

In addition to what has already been mentioned, after the spring release of Ubuntu, a bonus chapter will be available online at www.informit.com/title/9780672336935.

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 a DVD of Ubuntu 14.04 and an upgrade kit with information about changes from 13.10.

The ebook edition does not provide a DVD of Ubuntu or the upgrade kit.

If you purchased the ebook edition, or if you don't want to wait for a DVD, you can download the most current release of Ubuntu from www.ubuntu.com/download.

Conventions Used in This Book

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

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

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

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

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

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

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

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

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

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; it 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 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.

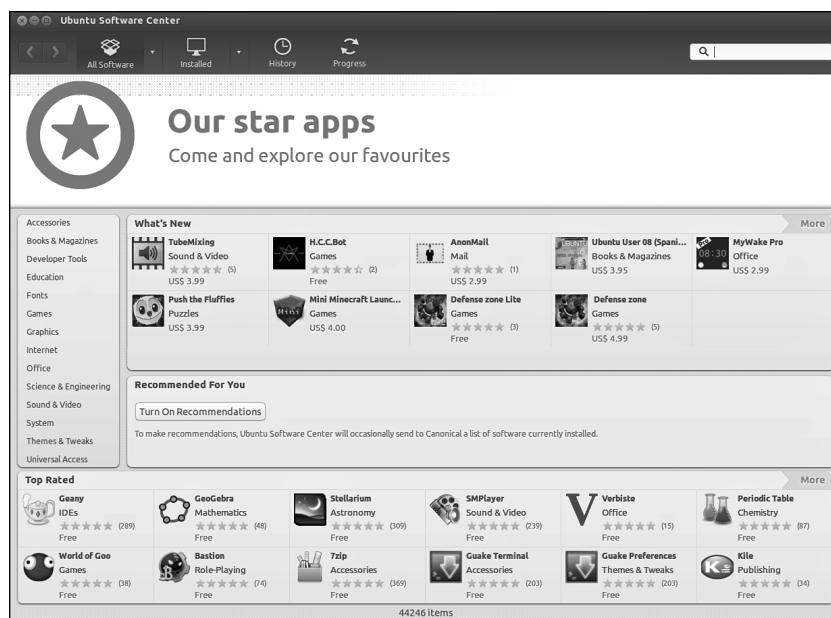


FIGURE 9.1 The initial Ubuntu Software Center screen enables you to browse through packages sorted by groups.

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 (Figure 9.2). You can install Synaptic using the Ubuntu Software Center described earlier; it is not installed by default.

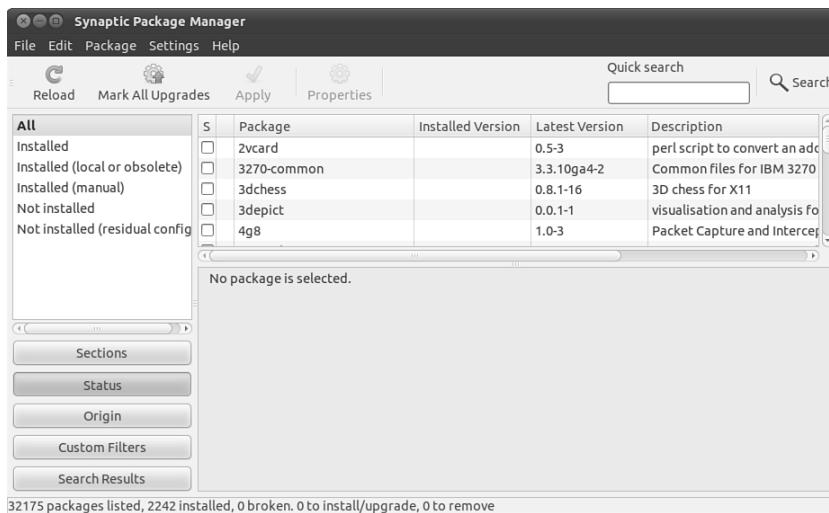


FIGURE 9.2 For more advanced software management in a GUI, Synaptic is the preferred tool.

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 more than 15,000 software packages available for installation. Ubuntu uses only a small subset of that number but makes it easy for you to enable the others. When you use Synaptic, you see small orange Ubuntu logos next to every package; this identifies them as being officially supported by the Canonical-supported Ubuntu developers. The other packages do not have this logo, but they 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 need 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.

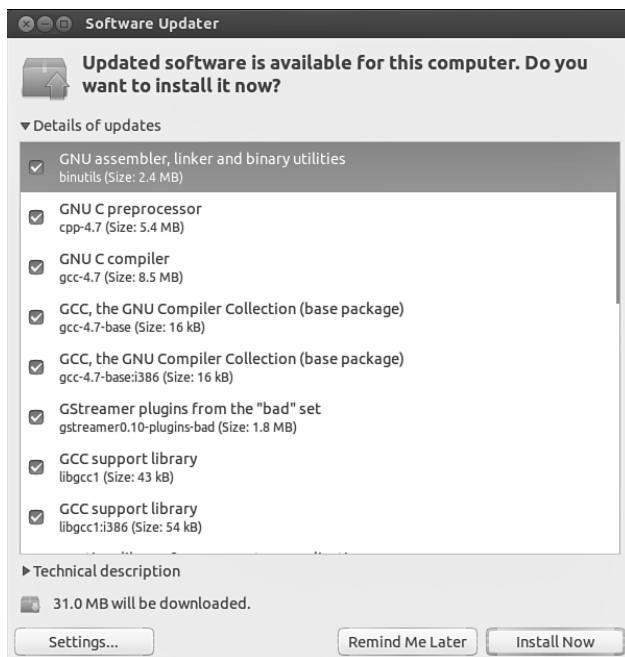


FIGURE 9.3 If you need to update your software to apply bug fixes and security upgrades, use Software Updater.

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:

```
matthew@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 libsmclient 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 5.0, 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 smaller things.

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

Symbols

/bin directory, 161-163
/etc directory, 163
/etc/host.conf file, 391
/etc/hosts file, 389
/etc/init.d/apache2, 498-499
/etc/modprobe.conf file
 editing, 382
 manually loading kernel modules, 382-383
/etc/nsswitch.conf file, 390
/etc/resolv.conf file, 390-391
/etc/samba/smb.conf file, 474-477
/etc/services file, 389-390
/home directory, 163-164
/proc directory, 164-165, 318
/sbin directory, 161-163
/tmp directory, 166
/usr directory
 shared data, 165-166
 subdirectories, 51-52
 /usr/bin, 51
 /usr/include, 51
 /usr/lib, 51
 /usr/lib/modules, 52
/usr/src/linux-3.2 directory, 449
/var directory, 166

Numbers

10BASE-T, 377
32-bit Ubuntu, 64-bit Ubuntu versus, 12
64-bit Ubuntu, 32-bit Ubuntu versus, 12
100BASE-T, 377-378
1000BASE-T, 378

A

AAAA record (DNS), 688
AbiWord (GNOME Office component), 88
ac command, 246
accept command, 489
access control, Apache web server, 504-510
access control lists (ACLs), 567-571
accessing
 command line, 155-158
 databases
 local GUI clients, 601-602
 SSH, 600-601
 Perl shell, 744
 accounts, Launchpad, 718
ACID compliance, MySQL versus PostgreSQL, 591-592
ACLs (access control lists), 567-571
activation, DHCP, 394-395
Ada, 818
Adblock Plus plug-in, 71
adding, users, 242-245
Additional Drivers manager, 129
address-based virtual hosts (Apache), 515-516
addressing
 broadcast, 376
 IPv4, 371
 IPv6, 372-375
 multicasts, 376
 TCP/IP, 369-370
 Unicast, 376
adduser command, 183
adjusting volume, music and sound, 95
admin group, sudo group versus, 181, 256
administration, LDAP, 625-626
administrative tools (BIOS), 312-313
Adobe Flash, 114
Adobe Photoshop, 101
ADT (Android Development Tools) plug-in, 831
ADT Eclipse plug-in, 832
Advanced Linux Sound Architecture (ALSA), 94

- Advanced Package Tool (APT), 143-148, 641
 - day-to-day usage, 144-147
 - finding software, 147-148
- afio backup tool, 353
- aliases, forwarding email, 554-555
- Alien Arena, 130
- all-in-one devices (print/fax/scan), 488
- allow directive, Apache web server access control, 506-507
- AllowOverrides directive, Apache web server configuration, 504
- Alpine mail client, 76
- ALSA (Advanced Linux Sound Architecture), 94
- Amanda backup application, 352-353
- AMD, proprietary drivers, 128
- American Registry for Internet Numbers, 369
- Android Development Tools (ADT) plug-in, 831
- Android mobile development, 829-835
 - Android Runtime, 830
 - Application Framework, 830-831
 - core applications, 831
 - creating application, 834-835
 - installation of Android SDK, 831-833
 - libraries, 830
 - Linux kernel, 830
- Android Runtime, mobile development for Android, 830
- Android SDK, installation, 831-833
- Android Virtual Device (AVD), 833
- anonymous access (FTP), 541-542
- anonymous servers (FTP), 536
- Ansible, managing sets of servers, 683
- Apache Module Registry, 510
- Apache Software Foundation, 492
- Apache Tomcat, 534
- Apache web server, 491-519, 537
 - access control, 504-510
 - Apache package directories, 493-494
 - directives, CustomLog, 519
 - file system authentication, 504-510
 - installation, 492-496
 - building the source, 494-496
 - Ubuntu repositories, 493-494
 - logging, 517-519
 - modules, 510-515
 - disabling, 510
 - enabling, 510
 - performance tuning, 441-442
- quick guide setup, 496
- runtime server configuration settings, 499-504
 - directives, 500
 - editing apache2.conf, 500-503
 - .htaccess configuration file, 503-504
 - multiprocessing modules, 503
- starting/stopping, 497-499
- user files, 508-509
- virtual hosting, 515-517
- wide use, 491
- apache2.conf, editing, 500-503
- AppArmor, 430-432
- Application Framework, mobile development for Android, 830-831
- applications
 - creating for Ubuntu Mobile/Touch, 838-839
 - design guidelines, 839
 - Internet, 69-82
 - Chromium, 71-72
 - email clients, 73-76
 - Firefox, 70-71
 - Google Chrome, 71-72
 - mobile development for Android, 831
 - multimedia, 93-116
 - burning CDs/DVDs, 107-112
 - digital cameras, 106-107
 - graphics manipulation, 100-105
 - sound and music, 93-100
 - viewing video, 112-116
 - productivity
 - GNOME Office, 87-88
 - KOffice, 87-89
 - LaTeX, 91
 - LibreOffice, 84-86
 - Microsoft Windows, 91-92
 - PDF, 89-90
 - XML and DocBook, 89-90
 - User Interface Toolkit, 839-840
- apply-patch tool (Bikeshed), 711
- apropros command, 160
- APT (Advanced Package Tool), 143-148, 641
 - 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 remove command, 147
- apt-get update command, 144
- A record (DNS), 688
- arithmetic operators (Perl), 736
- ark archiving tool (KDE), 348-350
- ARM processors, 830
- ARPANET, 69
- array_keys() function (PHP), 794
- array_unique() function (PHP), 793
- array_values() function (PHP), 794
- arrays
 - Perl, 735
 - PHP functions, 793-795
 - PHP programming, 775-777
- Artistic License3
- asort() function (PHP), 794
- assessment
 - backup needs, 337-338
 - resources for backup, 337-338
 - vulnerability, 421-422
- assigning permissions, 170-171
- at command, scheduling tasks, 261-264
- audio formats, 97
- authenticated servers (FTP), 536
- authentication, Apache web server, 507-509
- autoconf utility, 809-810
- autocracking scripts, 420
- automating tasks, 261-266
 - running jobs repeatedly, 264-266
 - scheduling tasks, 261-266
 - waking computer from sleep, 266-268
- automation of tasks, writing shell scripts, 280-282
- Autoresponders, 560
- AVD (Android Virtual Device), 833
- AVI (video format), 114
- awk text editor, 225

- B**
- Back in Time backup tool, 350-351
- background processing
 - command line, 215
 - shell control, 272-273
- backing up data, 335-362
- choosing a strategy, 335-342
- copying files, 354-358
- hardware and media, 342-344
- software, 344-353
- system rescue, 360-362
- version control, 358-360
- backslash, writing shell scripts, 285
- backticks
 - accessing the shell in Perl, 744
 - writing shell scripts, 286
- badblocks command, 439
- Bandwidth Meter, 71
- Banshee, 99-100
- Base (LibreOffice component), 85
- BaseX, 614-615
- Bash shell, configuration, 719-720
- Basic Authentication, Apache web server, 507
- basic input/output system. See BIOS (basic input/output system)
- batch command, 261-264
- Battle for Wesnoth*, 132-133
- Bazaar, 697-698, 719
- bch tool (Bikeshed), 711
- Berkeley DB, 610-611
- Berkeley Internet Name Domain (BIND), 691-693
- Bernes-Lee, Tim, 69
- BigTable, 615
- Bikeshed, 711-713
- /bin directory, commands, 161-163
- BIND (Berkeley Internet Name Domain), 691-693
- BIOS (basic input/output system), 308-314
 - booting into default runlevel, 311
 - controlling services, 312-313
 - final stage of initialization, 311-312
 - init scripts, 311-312
 - loading Linux Kernel, 309
 - runlevel definitions, 310
 - system services and runlevels, 310
 - troubleshooting runlevel problems, 313-314
 - tuning disk drives, 436-437
- Bitbucket, 699
- Blender, 105
- BMP (graphics format), 104

- body of message (newsgroup articles), 81
 - Boolean operators, combining commands, 215
 - Boot Loader
 - problems when compiling kernel, 466
 - Ubuntu installation, 13-14
 - boot process, 307-315
 - BIOS, 308-314
 - manually starting/stopping services, 314-315
 - running services, 307-308
 - Upstart, 315-316
 - booting into default runlevel (BIOS), 311
 - bootmail tool, 713
 - Brasero, 108
 - break statement, 304
 - bridged networking, 639
 - bridges, 381
 - broadcast addressing, 376
 - BSD License, 3
 - buffer usage, MySQL, 442-443
 - Bug Squad, 727
 - bugs
 - fixing, 720-722
 - Harvest, 723
 - built-in security, kernel, 419
 - built-in variables, executing shell scripts, 282-283
 - BulletProofX, 52
 - bunzip2 command, 227
 - burning CDs/DVDs, 107-112
 - Brasero, 108
 - command line, 109-112
 - business usage of Ubuntu, 39-40
 - buying music, 99
 - Bynari, 562
 - Byobu, 78
 - byobu command, 227-229
 - bzip2 command, 227
 - bzrp tool (Bikeshed), 711
-
- ## C
- C programming tools, 805-813
 - autoconf utility, 809-810
 - debugging tools, 810-811
 - GNU C compiler, 811-812
- graphical development, 812-813
 - Linux, 805-807
 - macros, 808-809
 - make command, 807
 - makefile targets, 809
 - makefiles, 807-809
 - C++ programming tools, 805-813
 - autoconf utility, 809-810
 - debugging tools, 810-811
 - GNU C compiler, 811-812
 - graphical development, 812-813
 - Linux, 805-807
 - macros, 808-809
 - make command, 807
 - makefile targets, 809
 - makefiles, 807-809
 - c10k problem, 522
 - cable (network), 379-380
 - Calc (LibreOffice component), 85
 - cameras, digital, 106-107
 - cancel command, 489
 - captured screen images, 105
 - case statement, 302-303
 - Cassandra, 611
 - cat command, 164-165, 179, 189-191
 - cd command, 168-169, 189, 191-193
 - Cd-RW drives, 343
 - CDs, burning, 107-112
 - Brasero, 108
 - command line, 109-110
 - Cedega, 136
 - CFEngine, 683
 - change command, 258
 - changing
 - passwords in a batch, 251
 - runlevels (BIOS), 312-313
 - characters
 - shell pattern matching, 270
 - writing shell scripts, 283
 - charms, Juju, 676-678
 - chat. See IRC (Internet Relay Chat)
 - Checkbox, 727
 - checking connections, networking, 366-368
 - Chef, 682
 - Cherokee, 533
 - chfn command, 258
 - chgrp command, 173, 258
 - children's games, 134-135

- Childsplay, 134
- chmod command, 172-173, 189, 193, 258
- chown command, 258
- chpasswd command, 251, 258
- Chromium, 71-72
- chsh command, 242, 258
- CinePaint, 105
- class inheritance, Python object orientation, 769-770
- class variables, Python object orientation, 767-768
- classes of networks, 369-370
- Claws mail client, 75
- CLI (command-line interface). See command line
- clients
 - configuration, proxy servers, 566-567
 - graphical, relational database services, 605
 - IP addresses, 571-572
 - LDAP, configuration, 624-625
 - NFS, configuration, 472
- client/server system, relational database services, 599-605
- Clojure, 819
- Cloud, 645-672
 - benefits, 645-648
 - Eucalyptus, 648-656
 - deployment/installation, 650
 - euca2ools primer, 654-656
 - IaaS (Infrastructure as a Service), 647
 - Juju, 672-678
 - charms, 676-678
 - GUI, 678
 - installation, 673-675
 - on Mac OS X, 678
 - Landscape, 679
 - MaaS (Metal as a Service), 648
 - OpenStack, 656-672
 - commands, 672
 - Compute Infrastructure (Nova), 656-657
 - creating an image, 667-670
 - Imaging Service (Glance), 657
 - installation, 657-667
 - instance management, 670
 - network management, 671-672
 - Storage Infrastructure (Swift), 657
 - storage management, 671
- PaaS (Platform as a Service), 647
- SaaS (Software as a Service), 647
- cloud storage, 82, 344
- cloud-sandbox tool (Bikeshed), 711
- CNAME record (DNS), 689
- COBOL (Common Business Oriented Language), 819-820
- code
 - configuring, autoconf utility, 809-810
 - packaging, 720-722
- code examples, Perl, 745-750
- code names, Ubuntu, 39
- CodeWeavers, 92
- coll tool (Bikeshed), 711
- combining commands, 214-215
 - Boolean operators, 215
 - piping data, 214-215
 - running commands in order, 215
- comm command, 213
- command line, 153-155
 - accessing, 155-158
 - APT
 - day-to-day usage, 144-147
 - finding software, 147-148
 - burning CDs/DVDs, 109-112
 - commands, 185-186, 189-210
 - background processing, 215
 - cat, 190-191
 - cd, 168-169, 191-193
 - chmod, 193
 - combining, 214-215
 - common commands and programs, 185-186
 - cp, 193-195
 - du, 194-195
 - echo, 209
 - find, 195-198
 - grep, 197-198
 - less, 198-200
 - ln, 200-201
 - locate, 202
 - ls, 166-168, 202-204
 - man, 204-205
 - mkdir, 205
 - mv, 205

ps, 206
 pwd, 169
 rm, 206-207
 running in order, 215
 system reset, 229
 tail, 207
 top, 207-209
 which, 210
 comparing files, 212-213
 compressed files, 227
 defined, 154-155
 directories
 changing, 168-169
 listing contents, 166-168
 pwd command, 169
 environment variables, 217-220
 errors in Perl, 733
 files
 copying, 178
 creating directories, 175-176
 creating with touch command, 175
 deleting, 177
 deleting directories, 176
 displaying contents, 179
 moving/rename, 177-178
 wildcards/regular expressions, 179
 Linux file system hierarchy, 161-166
 logging in/out from a remote computer,
 157-158
 logging out, 157
 multiple terminals, 227-229
 MySQL client, 603-604
 navigating Linux file system, 166-169
 network interface configuration, 384-388
 permissions, 169-175
 assigning permissions, 170-171
 chmod command, 172-173
 directory permissions, 171-172
 file permissions, 173
 set UIDs/GID permissions, 173-175
 PostgreSQL client, 604
 reading documentation, 160
 reasons for use, 188-189
 redirecting input and output, 210-211
 redirection of streams, 211-212
 root users, 180-185
 scripting. See Python
 setting priorities, 229
 shell control, 269-270
 system reset, 229
 text editors, 220-224
 emacs, 223-224
 nano, 222
 sed and awk, 225
 vi, 222-223
 user accounts, 158-159
 working with files, 175-179
 command prompt. See command line
 command-line interface (CLI). See command
 line
 commands
 /bin directory, 161-163
 /sbin directory, 161-163
 ac, 246
 adduser, 183
 AppArmor, 432
 apropros, 160
 apt-get autoclean, 146
 apt-get clean, 146
 apt-get dist-upgrade, 145
 apt-get install, 145
 apt-get remove, 147
 apt-get update, 144
 at, scheduling tasks, 261-264
 badblocks, 439
 batch, scheduling tasks, 261-264
 byobu, 227-229
 cat, 164-165, 179
 cd, 168-169
 chgrp, 173, 258
 chmod, 172-173, 258
 chown, 258
 chpasswd, 251
 chsh, 258
 comm, 213
 command line, 185-186, 189-210
 background processing, 215
 cat, 190-191
 cd, 168-169, 191-193
 chmod, 193
 combining, 214-215
 common commands and programs,
 185-186
 cp, 193-195
 du, 194-195
 echo, 209

find, 195-198
 grep, 197-198
 less, 198-200
 ln, 200-201
 locate, 202
 ls, 166-168, 202-204
 man, 204-205
 mkdir, 205
 mv, 205
 ps, 206
 pwd, 169
 rm, 206-207
 running in order, 215
 system reset, 229
 tail, 207
 top, 207-209
 which, 210
 compression of files, 227
 cp, 178, 355-356
 cron, 264-266
 date, 30
 deluser, 184
 diff, 213
 dmesg, 33
 dropuser (PostgreSQL), 598
 e2fsck, 439
 edquota, 257
 emacs text editor, 224
 env, 218
 Eucalyptus, 654-656
 exit, 157
 faillog, 326
 find, 174
 gpasswd, 239
 gprof, 811
 groupadd, 239
 groupdel, 239
 groupmod, 239
 grpck, 239
 hdparm, 437-438
 hdx=ide-scsi, 436
 hwclock, 31
 idebus=xx, 436
 idex=autotune, 436
 idex=dma, 436
 ifconfig, 369, 384-388, 425
 init, 310
 Internet connectivity, 408
 ionice, 229
 iptables, 429
 kernel module management, 452-453
 kill, 319-320
 lastlog, 326
 less, 160, 179, 41
 logout, 157
 ls, 166-170
 make, 807
 man, 41, 160
 mc, 356-358
 mkdir, 171, 175-176
 mv, 177-178
 mysql -u root, 594
 nice, 229, 320-321
 OpenStack, 672
 passwd, 113
 patch, 458
 pci=biosirq, 437
 Perl, 750
 ping, 366-368
 postconf, 551
 printenv, 218, 253
 printing, 489
 pwd, 157, 169
 Quickly, 704
 quotacheck, 257
 quotaoff, 257
 quotaon, 257
 rcp, 412
 relational database services, 605
 renice, 229, 320-321
 repquota, 257
 rm, 177
 rmdir, 176
 route, 384-388
 rsync, 356-358
 rtcwake, 266-268
 scp, 412
 sed and awk text editors, 224
 setting priorities, 229
 sftp, 411, 413
 shutdown, 20, 184-185
 smbclient, 479

smbstatus, 478
 splint, 810-811
 ssh-keygen, 413-415
 su, 251-253
 sudo, 24-25, 180-183, 234
 tar, 354-355
 testparm, 477-478
 time, 321
 top, 321
 touch, 169, 175, 414
 tune2fs, 438-439
 UFW (Uncomplicated Firewall), 427
 umask, 169
 uname, 36
 uptime, 323
 user accounts, 258-259
 useradd, 240, 259
 usermod, 242, 259
 vi text editor, 223
 whereis, 161
 zless, 41
 comments, PHP programming, 779
 commercial games, 135-136
 commercial support websites, 44
 Common Business Oriented Language (COBOL), 819-820
 Common UNIX Printing System (CUPS), 484
 CommuniGate Pro, 561
 community teams, testing, 725-727
 comparison, files, 212-213
 comparison operators (Perl), 736-737
 compiling applications from source, software management, 148-150
 compiling the kernel, 457-465
 troubleshooting, 465-466
 Compiz, 120
 compound operators (Perl), 736
 Comprehensive Perl Archive Network (CPAN), 745
 compressed files, command line, 227
 Compute Infrastructure (Nova) service, OpenStack, 656-657
 computer attacks, hacker versus cracker, 420
 computer engineering, Cloud. See Cloud
 conditional statements
 Perl, 739-740
 PHP programming, 783-784
 Python, 763-765
 configuration
 Apache web server, 499-504
 directives, 500
 editing apache2.conf, 500-503
 .htaccess configuration files, 503-504
 multiprocessing modules, 503
 Bazaar, 719
 clients, proxy servers, 566-567
 DHCP, 395-396
 DHCP network hosts, 397-399
 Dial-Up access, 406-407
 Digital Subscriber Line (DSL) access, 404-406
 Fetchmail, 555-559
 FireFox, 566
 graphical tools, 391-392
 kernel, 457-465
 local Bash shell, 719-720
 loopback interface, 364-365
 management, 150-151
 MySQL, 593-596
 networking tools, 383-392
 NFS client, 472
 NFS server, 470-472
 Nginx, 524-527
 Nginx modules, 530
 Postfix, 550-555
 Postfix masquerading, 552
 PostgreSQL, 596-599
 PPPoE, 405-406
 quotas, 257-258
 Samba, 474-477, 480-483
 software repositories, 25-27
 system settings, 27-31
 printers, 28
 time and date, 29-31
 Very Secure FTP server, 540-543
 wireless networks, 31-32
 configuration files
 /etc directory, 163
 .htaccess, 503-504
 networking, 389
 version control, 358-360
 ./configure, building Apache, 495
 configuring
 code, autoconf utility, 809-810
 firewalls, 426-429
 LDAP clients, 624-625

- LDAP server, 620-623
- Tripwire, 424-425
- Unity, 65-66
- Conky, 328-333
- console. See command line
- console-based monitoring tools, 317-324
 - disk quotas, 324
 - disk space, 323-324
 - free and used memory, 323
 - kill command, 319-320
 - log files, checking, 324-326
 - priority scheduling and control, 320-322
- constants
 - Perl string constants, 736
 - PHP programming, 778
- constructors, Python object orientation, 768-769
- control structures, Perl, 740-743
- controlling services, BIOS (basic input/output system), 312-313
- convert utility (ImageMagick), 104
- copying files, 354-358
 - cp command, 355-356
 - Midnight Commander, 356-358
 - rsync command, 356-358
 - tar command, 354-355
- core applications, mobile development for
 - Android, 831
- CouchDB, 613-614
- cp command, 178, 189, 193-195, 355-356
- CPAN (Comprehensive Perl Archive Network), 745
- CREATE DATABASE statement
 - MySQL, 594
 - PostgreSQL, 597
- creating
 - MySQL database, 594-596
 - OpenStack images, 667-670
 - PostgreSQL database, 597
 - tables, relational database services, 586-587
- cron command, 264-266
- Crossover Games, 136
- CrossOver Office, 92
- Cube 2: Sauerbraten, 130
- CUPS (Common UNIX Printing System), 484, 486-488
 - custom tools, Eucalyptus, 654-656
 - customizing Unity, 65-66
 - CustomLog directive (Apache), 519
- D
 - D programming language, 820
 - Dalvik, 830
 - darktable, 105
 - Dash (Unity desktop), 20, 61-63, 66
 - data
 - mirroring, 342
 - retrieval from databases, 588-590
 - data backup, 335-362
 - choosing a strategy, 335-342
 - copying files, 354-358
 - hardware and media, 342-344
 - software, 344-353
 - system rescue, 360-362
 - version control, 358-360
 - data directory (PostgreSQL), initializing, 596-597
 - data files, /var directory, 166
 - data integrity, MySQL versus PostgreSQL, 591-592
 - data locking, MySQL versus PostgreSQL, 591
 - data loss, 336-337
 - data structures, Perl, 734-736
 - database administrators (DBAs), 582
 - databases
 - NoSQL, 583, 607-617
 - document stores, 612-615
 - graph stores, 616-617
 - key/value stores, 610-612
 - wide column stores, 615-616
 - PHP programming, 801-804
 - relational database services, 581-605
 - client/server system, 599-605
 - commands, 605
 - comparison of MySQL and PostgreSQL, 590-593
 - creating tables, 586-587
 - future of MySQL, 581
 - how they work, 584-586

- inserting data in tables, 587-588
- MySQL, 593-596
- PostgreSQL, 596-599
- retrieving data, 588-590
- SQL basics, 586
- date, configuring system settings, 29-31
- date command, 30
- day job crackers, 420
- day-to-day usage, APT, 144-147
- DBAs (database administrators), 582
- Debian, 25, 36, 38
- debugging tools, 810-811
- default runlevel, booting into, 311
- definitions
 - Python functions, 765
 - runlevels, 310
- Deja Dup backup tool, 348-350
- deleting users, PostgreSQL, 598-599
- deluser command, 184
- deny directive, Apache web server access control, 506-507
- deployment, Eucalyptus, 650
- design guidelines for applications, 839
- Desktop Couch, 703
- Desktop DVD, 10
- desktop environment, interfaces, 120
- destructors, Python object orientation, 768-769
- development, 715-723
 - finding bugs with Harvest, 723
 - fixing bugs and packaging, 720-722
 - helping with Ubuntu development, 42
 - installation packages, 717-720
 - MOTU (Masters of the Universe), 723
 - opportunistic. See opportunistic development
 - six-month cycle, 716-717
- Device section (`xorg.conf` file), 56
- devices, security, 425
- DevOps, SysAdmin versus, 646
- DHCP (Dynamic Host Configuration Protocol), 393-399, 632
 - configuring network hosts, 397-399
 - how it works, 393-394
 - installation and activation, 394-395
 - server, 396
 - software installation and configuration, 395-396
 - uses, 399
- Dia (LibreOffice component), 86
- Diagnostics, 71
- Dial-Up access, Internet connectivity, 406-407
- ictionaries, Python, 762-763
- diff command, 213
- digiKam, 105
- digital cameras, 106-107
- Digital Subscriber Line (DSL) access, 404-406
- directives (Apache web server), 500
 - access control, 506-507
 - CustomLog, 519
 - satisfy, 509
- directories
 - Apache package, 493-494
 - command line
 - changing with `cd` command, 168-169
 - listing contents with `ls` command, 166-168
 - `pwd` command, 169
 - Linux, 162
 - Linux source tree, 449-451
- Directory Information Tree (DIT), 620
- directory permissions, 171-172
- DirectoryIndex directive, Apache web server configuration, 503
- disable command, 426
- disabling
 - Apache modules, 510
 - file access time, 439
- disaster recovery plan, 432-433
- disk drives, tuning, 436-437
- disk quotas, 256-258, 324
- disk space, console-based monitoring tools, 323-324
- display manager, X Server, 58
- dist-upgrade option, 24
- DIT (Directory Information Tree), 620
- dman tool (Bikeshed), 712
- dmesg command, 33
- DNS (Domain Name System)
 - DNS records, 688-691
 - A, 688
 - AAAA, 688
 - CNAME, 689
 - MX, 689
 - NS, 689-690
 - SOA, 690-691
 - TXT, 691

- DNS servers, 687-688, 691-693
 - domain names, explained, 687
 - reasons for using, 685-686
 - DocBook, 89-90
 - document stores, NoSQL databases, 612-615
 - documentation
 - command line, 153-155
 - helping with development, 42
 - Ubuntu and Linux, 40-42, 44
 - DocumentRoot directive, Apache web server configuration, 502
 - documents, kernel programmers, 449-450
 - Domain Name System (DNS). See DNS (Domain Name System)
 - domain names
 - explained, 687
 - mapping IP addresses to, 685-686
 - DNS records, 688-691
 - DNS servers, 687-688
 - server setup with BIND, 691-693
 - dotdee, 150-151
 - do.until loop (Perl), 742
 - do.while loop
 - Perl, 742
 - PHP, 789
 - Draw (LibreOffice component), 86
 - Drizzle, 581
 - dropuser command (PostgreSQL), 598
 - DSL (Digital Subscriber Line) access, 404-406
 - du command, 194-195
 - DVD installation jump start, 10
 - DVD+RW/-RW drives, 343
 - 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, 439
 - echo command, 209
 - Eclipse, 823
 - Eclipse Foundation project, Jetty, 533
 - editing
 - apache2.conf, 500-503
 - /etc/modprobe.conf file, 382
 - video, 116
 - editing commands
 - emacs text editor, 224
 - sed and awk text editors, 224
 - edquota command, 257
 - Edubuntu, 628
 - elements (xorg.conf file), 52-57
 - Device section, 53, 56
 - Files section, 53-54
 - InputDevice section, 54-55
 - Module section, 54
 - Monitor section, 53, 55-56
 - Screen section, 53, 56-57
 - ServerLayout section, 53
 - emacs command, 189
 - emacs text editor, 221, 223-224
 - email, 545-562
 - alternatives to Microsoft Exchange Server, 560-562
 - Fetchmail, 555-559
 - how email is sent and received, 545-550
 - mail delivery agents, 559-560
 - Postfix configuration and operation, 550-555
 - email clients, 73-76
 - Evolution, 74-75
 - Mozilla Thunderbird, 73-74
 - Empathy, 77
 - emulators, 127
 - enable command, 489
 - enabling Apache modules, 510
 - endless loops, shell programs, 297
 - Enlightenment, 120
 - entering PHP mode, 774
 - enterprise servers, monitoring tools, 333-334
 - env command, 218
 - environment, Launchpad, 718-720
 - environment variables, command line, 217-220
 - Erlang, 820-821
 - errors, compiling kernel, 465
 - escape sequences, PHP programming, 779-780
 - /etc directory, configuration files, 163

/etc/host.conf file, 391
 /etc/hosts file, adding hosts, 389
 /etc/init.d/apache2, starting/stopping Apache web server, 498-499
 /etc/modprobe.conf file
 editing, 382
 manually loading kernel modules, 382-383
 /etc/nsswitch.conf file, 390
 /etc/resolv.conf file, 390-391
 /etc/samba/smb.conf file, 474-477
 /etc/services file, service settings, 389-390
 ethereal, 333
 etiquette, IRC (Internet Relay Chat), 80
 euca2ools primer, Eucalyptus, 654-656
 Eucalyptus, 645, 648-656
 deployment/installation, 650
 euca2ools primer, 654-656
 event-driven architecture, 522-523
 Evolution (GNOME Office component), 74-75, 88, 624
 executing
 Python scripts, 754-755
 shell scripts, 273-305
 accessing variable values, 278
 assigning value to variables, 278
 automation of tasks, 280-282
 backslash, 285
 backtick, 286
 break statement, 304
 built-in variables, 282-283
 case statement, 302-303
 comparison of expressions, 286-291
 comparison of expressions with tcsh, 291-295
 exit statement, 304-305
 if statement, 300-301
 interpreting shell scripts, 276-277
 positional parameters, 278-279
 repeat statement, 299
 running shell program, 274-275
 select statement, 299-300
 shift statement, 300
 sorting scripts for access, 275-276
 special characters, 283
 for statement, 295-296
 strings with embedded spaces, 284
 un3til statement, 298-266
 unexpanded variables, 284-285

until statement, 298-299
 variables, 277
 while statement, 297-298
 execution operator, 785
 Exim, 547
 exit command, 157, 304
 exiting PHP mode, 774
 expressions, Perl regular expressions, 743
 extensions, Firefox, 71
 external attacks, 420
 extract() function (PHP), 795
 Extraversion level (kernel), 455

F

faillog command, 326
 fclose() function (PHP), 797
 features, LTSP, 632-633
 Fetchmail, 555-559
 configuration, 555-559
 installation, 555
 user accounts, 557-559
 fiber optic cable, 380
 fiber optics, 378
 file access time, disabling, 439
 file operators, writing shell scripts, 289-290, 293-294
 file permissions, 173, 236-237
 file system, Linux hierarchy, 161-166
 file system authentication, Apache web server, 504-510
 file system settings, tuning, 438
 File Systems tab (System Monitor), 326
 File Transfer Protocol. See FTP (File Transfer Protocol)
 file_get_contents() function (PHP), 795
 file_put_contents() function (PHP), 795
 files
 .htaccess configuration, Apache web server configuration, 503-504
 /etc/host.conf, 391
 /etc/hosts, adding hosts, 389
 /etc/nsswitch.conf, 390
 /etc/resolv.conf, 390-391
 /etc/services, service settings, 389-390
 Apache file locations after install, 495-496

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/rename files, 177-178
 wildcards/regular expressions, 179
 comparison, 212-213
 compressed, 227
 configuration, version control, 358-360
 copying, 354-358
 cp command, 355-356
 Midnight Commander, 356-358
 rsync command, 356-358
 tar command, 354-355
 ftphosts, 543-544
 PHP functions, 795-797
 restoring from an archive, 346-347
 saving from nonbooting hard drive, 362
 sharing, 469-483
 NFS (Network File System), 470-472
 Samba, 480-483
 Ubuntu installation, 20
 Files section (xorg.conf file), 53-54
 filesize() function (PHP), 797
 find command, 174, 189, 195-198
 Firefox, 70-71. See also Apache web server
 configuration, 566
 RSS feeds, 76
 firewalls, configuring, 426-429
 first-person shooter (FPS) games, 130
 fixing bugs, 720-722
 FLAC (sound format), 96
 Flash, 71, 114
 flavors (Ubuntu), 10
 flexbackup tool, 353
 FlightGear, 134-135
 FlockDB, 617
 FLV (video format), 114
 fopen() function (PHP), 796
 foreach loop
 Perl, 741
 PHP, 788
 ForecastFox, 71
 for loop
 Perl, 740
 PHP, 787
 Python, 764
 formats
 DVDs, 110
 graphics manipulation, 103-105
 sound, 96-97
 video, 114
 for statement, writing shell scripts, 295-296
 Forth, 821
 Fortran, 822
 forwarding email, 554-555
 FPS (first-person shooter) games, 130
 FQDN (fully qualified domain name), 620
 fread() function (PHP), 797
 free and used memory, console-based monitoring tools, 323
 Frets on Fire, 134
 Frozen Bubble, 131
 FTP (File Transfer Protocol), 535-544
 configuring Very Secure FTP server, 540-543
 ftphosts file, 543-544
 server selection, 535-537
 servers, 536-537
 software installation, 537-538
 users, 538-540
 ftphosts file, 543-544
 full backups
 incremental backups, 341
 periodic basis, 341
 tar backup tool, 345-346
 fully qualified domain name (FQDN), 620
 functions
 PHP programming, 790-801
 arrays, 793-795
 files, 795-797
 miscellaneous functions, 797-800
 strings, 790-793
 Python, 765-766
 shell scripts, 304-305
 future of MySQL, 581
 fwrite() function (PHP), 797

G

- games, 127-136
 - Battle for Wesnoth, 132
 - children, 134-135
 - commercial, 135-136
 - FlightGear, 134-135
 - Frets on Fire, 134
 - Frozen Bubble, 131
 - installation, 129-136
 - proprietary video drivers, 128-129
 - Scorched 3D, 130-131
 - Speed Dreams, 134-135
 - SuperTux, 132-133
 - Warsow, 130
 - Windows, 136
- gcc (GNU Compiler Collection), 189, 805
- gCompris, 134
- gdb tool, 811
- gecos field, 247
- gedit text editor, 221, 703
- Genprof, 431
- GHC (Glorious Glasgow Haskell Compilation system), 823
- GIDs (group IDs), 173-175, 236
- GIF (graphics format), 104
- Gigabit Ethernet, 378
- Gilt, 699-700
- GIMP (GNU Image Manipulation Program), 88, 101
- GitHub, 700
- Glade, 703, 813
- Glance (Imaging Service), OpenStack, 657
- global behavior, Samba, 476
- global options, configuring Fetchmail, 556
- Glorious Glasgow Haskell Compilation system (GHC), 823
- Gnat Programming System, 818
- GNOME, Glade client, 813
- GNOME file roller, 347
- GNOME 3, 124-125
- gnome-nettool, 333
- gnome-screenshot, 105
- gnome-shell, 125
- GNU C compiler, 811-812
- GNU Compiler Collection (gcc), 805
- GNU Fortran 95 compiler, 822
- GNU GPL, 35-36
- GNU Image Manipulation Program (GIMP), 101
- GNU Privacy Guard, 717
- GNU Project, 153
- GNU/Linux, 448
- Gnumeric (GNOME Office component), 88
- Go, 821-822
- Google, search tips, 43
- Google Chrome, 71-72
- gpasswd command, 239
- GPG key, 717-718
- gprof command, 811
- Grand Unified Boot Loader (GRUB2), 13-14
- GRANT statement (PostgreSQL), 599
- granting privileges, PostgreSQL databases, 599
- graph stores, NoSQL databases, 616-617
- graphical clients, relational database services, 605
- graphical configuration tools, 391-392
- graphical development tools, 812-813
- graphical process monitoring tools, 326-333
 - Conky, 328-333
 - System Monitor, 326
- graphical user interfaces (GUIs). See GUIs (graphical user interfaces)
- graphics manipulation, 100-105
 - editor options, 105
 - formats, 103-105
 - GNU Image Manipulation Program (GIMP), 101
 - Photoshop, 101
 - scanners, 103
 - screen images, 105
- grep command, 189, 197-198
- Groklaw, 77
- Groovy, 822
- Ground Control, 707-711
- Group directive, Apache web server configuration, 501
- group IDs (GIDs), 173-175, 236
- group listing, 237-238
- group management, 237-240
- group permissions, 236
- groupadd command, 239
- groupdel command, 239
- groupmod command, 239
- groups command, 258
- grpck command, 239
- GRUB2 (Grand Unified Boot Loader), 13-14, 361

GStreamer, 703
 GTK, 87-88, 703
 GUFW, 428
 GUIs (graphical user interfaces), 119-121
 desktop environment, 120
 GNOME 3 and Ubuntu GNOME, 124-125
 Juju, 678
 KDE and Kubuntu, 121-122
 LXDE and Lubuntu, 123-124
 touchscreen interface, 837-838
 application design guidelines, 839
 creating applications, 838-839
 SDK installation, 838
 User Interface Toolkit, 839-840
 Ubuntu Kylin, 125
 Unity desktop, 49-67
 Mir, 50
 X Server, 49-58
 Xfce and Xubuntu, 122-123
 gunzip command, 227
 gzip command, 227

H

hackers, crackers versus, 420
 handheld digital cameras, 106
 hard disk performance tuning, 435-439
 badblocks command, 439
 disabling file access time, 439
 e2fsck command, 439
 file system settings, 438
 hdparm command, 437-438
 tune2fs command, 438-439
 tuning disk drives, 436-437
 hardware
 data backup, 342-344
 networking, 377-383
 hubs and switches, 380-381
 initializing new hardware, 381-383
 network cable, 379-380
 NIC (Network Interface Cards), 377-379
 routers and bridges, 381
 specifications, 10
 TV and video, 112-114

Harvest, finding bugs, 723
 hashes (Perl), 735
 Haskell, 822-823
 HBase, 615-616
 HDLC (high-level data link control), 406
 hdparm command, 437-438
 hdx=ide-scsi command, 436
 header lines (newsgroup articles), 81
 help
 IRC (Internet Relay Chat), 47
 mailing lists, 46-47
 software management, 43
 with Ubuntu development, 42
 websites, 42-46
 commercial support, 44
 Linux guides, 44-45
 LUGs (Linux User Groups), 44
 search tips, 42-43
 Ubuntu-specific sites, 45-46
 hibernate, 17
 hierarchy, Linux file system, 161-166
 high-level data link control (HDLC), 406
 history
 Internet, 69
 LibreOffice, 86
 Linux, 35-36
 Ubuntu, 38-39
 home directories, sharing, 476-477
 /home directory, user directories, 163-164
 home users
 backup programs, 342-353
 of Ubuntu, 40
 Horde, 562
 host names, 687
 hosts, adding to /etc/hosts file, 389
 HOWTO documents, 41-42
 .htaccess configuration files, Apache web server configuration, 503-504
 HTML (Hypertext Markup Language), 69
 HTML forms, PHP programming, 801
 HTTP (Hypertext Transfer Protocol), 69
 HTTP servers, 531-534
 Apache Tomcat, 534
 Cherokee, 533
 Jetty, 533
 lighttpd, 531-532

- Nginx, 521-523
 - configuration, 524-527
 - installation, 523-524
 - modules, 530
 - PHP setup, 528-529
 - virtual hosting, 527-528
- thttpd, 534
- YAWS, 532
- hubs, 380-381
- Hugin, 105
- Humble Indie Bundle, 136
- hwclock command, 31
- HyperGraphDB, 616-617
- Hypertext Markup Language (HTML), 69
- Hypertext Transfer Protocol (HTTP), 69

- IaaS (Infrastructure as a Service), 647
- idebus=xx command, 436
- idex=autotune command, 436
- idex=dma command, 436
- if statement, executing shell scripts, 300-301
- ifconfig command, 369, 384-388, 425
- if/else conditional statements, Perl, 739-740
- ImageMagick, convert utility, 104
- images, OpenStack, 667-670
- Imaging Service (Glance), OpenStack, 657
- IMAP (Internet Message Access Protocol), 549
- implementation, quotas, 257
- Impress (LibreOffice component), 85
- in_array() function (PHP), 794
- incremental backups, tar backup tool, 345-346
- Infrastructure as a Service (IaaS), 647
- init command, 310
- init scripts, BIOS (basic input/output system), 311-312
- initialization
 - BIOS (basic input/output system), 311-312
 - data directory (PostgreSQL), 596-597
 - network hardware, 381-383
- Inkscape, 105
- input, redirecting, 210-211, 271-272
- InputDevice section (*xorg.conf* file), 53-55
- inserting data in tables, relational database services, 587-588

- installation
 - ADT Eclipse plug-in, 832
 - Android SDK, 831-833
 - Apache web server, 492-496
 - building the source, 494-496
 - Ubuntu repositories, 493-494
 - CPAN module (Perl), 747
 - development packages, 717-720
 - DHCP, 394-396
 - Eucalyptus, 650
 - Fetchmail, 555
 - FTP software, 537-538
 - games, 129-136
 - Ground Control, 707
 - Java, 831
 - Juju, 673-675
 - LTSP, 631-632
 - NFS, 470
 - Nginx, 523-524
 - OpenStack, 657-667
 - proprietary video drivers, 128-129
 - SDK, 831, 838
 - Squid, 566
 - Ubuntu, 9-34
 - 32-bit versus 64-bit Ubuntu, 12
 - on Mac hardware, 11
 - post-installation configuration problems, 33-34
 - preparation, 9-14
 - programs and files, 20
 - shutting down, 19-20
 - software repositories, 25-27
 - Software Updater, 21-24
 - step-by-step installation, 14-19
 - sudo command, 24-25
 - system settings, 27-31
 - wireless network configuration, 31-32
 - virtual devices, 833
 - instance management, OpenStack, 670
 - instant messaging, Empathy, 77
 - interfaces. See GUIs (graphical user interfaces)
 - internal attacks, 420
 - Internet, history, 69
 - Internet applications, 69-82
 - Chromium, 71-72
 - email clients, 73-76
 - Evolution, 74-75
 - Mozilla Thunderbird, 73-74

- Empathy, 77
 - Firefox, 70-71
 - Google Chrome, 71-72
 - Internet Relay Chat (IRC), 78-79
 - RSS feeds, 76-77
 - Firefox, 76
 - Liferea, 76-77
 - Ubuntu One cloud storage, 82
 - Usenet newsgroups, 79-82
 - Internet connectivity, 402-408
 - commands, 408
 - common configuration information, 402-403
 - Dial-Up access, 406-407
 - DSL access, 404-406
 - troubleshooting connection problems, 407-408
 - Internet Message Access Protocol (IMAP), 549
 - Internet Relay Chat (IRC), 47, 78-79
 - interprettting shell scripts, 276-277
 - ionice command, 229
 - IP addresses
 - clients, 571-572
 - mapping to domain names, 685-686
 - DNS records, 688-691
 - DNS servers, 687-688
 - server setup with BIND, 691-693
 - IP masquerading, 370
 - iptables command, 429
 - IPv4, 371
 - IPv6, 371-375
 - IRC (Internet Relay Chat), 47, 78-79
 - IRCd server, 79
 - isset() function (PHP), 798
 - iwconfig tool, 399
 - iwlist tool, 399
 - iwpriv tool, 399
 - iwspy tool, 399
-
- J**
 - Java, 71, 823, 831
 - Java Runtime Environment (JRE), 831
 - Java Virtual Machine (JVM), 831
 - JavaScript, 823-824
-
- K**
 - kate text editor, 221
 - KDE, 121-122
 - KDE ark archiving tool, 348-350
 - KDE process, monitoring tools, 333
 - KDevelop client, 812-813
 - KDevelop Setup Wizard, 812
 - kdf tool, 333
 - keep-one-running tool, 713
 - kernel
 - built-in security, 419
 - interacting via /proc directory, 164-165
 - management, 447-466
 - compiling the kernel, 457-465
 - Linux kernel, 448-452
 - modular kernels, 452-454
 - obtaining sources, 456
 - patching the kernel, 457-458
 - recompiling the kernel, 454-455
 - troubleshooting during compile, 465-466
 - versions, 455-456
 - performance tuning, 440-441
 - tuning disk drives, 436-437
 - version numbers, 36
 - kernel source tree, 449-451
 - Kernel-based Virtual Machine (KVM), 637-641
 - key-based logins, ssh-keygen command, 413-415
 - key/value stores, NoSQL databases, 610-612
 - Kile, 91

kill command, 319-320
 Kirkland, Dustin, 727
 Kmail mail client, 76
 Knoppix, 124
 KOffice, 87-89
 ksort() function (PHP), 794
 KSpread, 89
 ksysguard tool, 333
 Kubuntu, 121-122
 KVM (Kernel-based Virtual Machine), 637-641
 KWord, 88

L

labels in DNS (Domain Name System), 687
 LAN, enabling network printing, 484
 Landscape, 334, 679, 683
 LANG environment variable, 217
 lapd-utils package, 620
 laptops, security, 424
 large enterprise users, backup programs, 339
 last looping construct (Perl), 742
 lastlog command, 326
 LaTeX, 91
 Launcher (Unity desktop), 60
 Launchpad, 698, 701-702
 accounts, 718
 environment, 718-720
 LDAP (Lightweight Directory Access Protocol), 619-626
 administration, 625-626
 client configuration, 624-625
 server configuration, 620-623
 ldapadd utility, 625
 ldapdelete utility, 626
 ldapmodify utility, 625
 ldapsearch utility, 625
 LDM (LTSP Display Manager), 632
 less command, 41, 160, 179, 189, 198-200
 levels of backup, 340
 libraries, mobile development for Android, 830
 LibreOffice, 84-86
 licensing, 2-3, 35-36
 Liferea, RSS feeds, 76-77
 lighttpd, 531-532

Lightweight Directory Access Protocol. See
 LDAP (Lightweight Directory Access Protocol)
 limits, IPv4 addressing, 371
 Linux
 C programming tools, 805-807
 directories, 162
 documentation, 40-42, 44
 file system hierarchy, 161-166
 history, 35-36
 IRC (Internet Relay Chat), 47
 kernel management, 448-452
 kernel source tree, 449-451
 types of kernels, 451-452
 mailing lists, 46-47
 Perl, 731-734
 reasons for use, 36-38
 websites, 42-46
 commercial support, 44
 Linux guides, 44-45
 LUGs (Linux User Groups), 44
 search tips, 42-43
 Ubuntu-specific sites, 45-46
 Linux Documentation Project, 407
 Linux file system, navigating, 166-169
 Linux kernel
 loading, 309
 mobile development for Android, 830
 Linux Terminal Server Project, 627-633
 features, 632-633
 installation, 631-632
 requirements, 628-631
 Linux User Groups (LUGs), 44
 Lisp, 824
 Listen directive, Apache web server configuration, 501
 listening to music, 97-100
 lists, Python, 760-762
 Live Bookmarks, 76
 In command, 189, 200-201
 loading Linux Kernel, 309
 local GUI clients, database access, 601-602
 localhost interface, 364-365
 loopback interface availability, 364
 loopback interface configuration, 364-365
 locate command, 189, 202
 log files, console-based monitoring tools, 324-326
 log files (Apache), 517

- LogFormat statements (Apache), variables, 518-519
- logging in (command line), 157-158
- logging out (command line), 157-158
- logical operators, writing shell scripts, 290-293, 294-295
- logname command, 258
- logout command, 157
- logs
 - Apache web server, 517-519
 - purging in Perl, 747-748
- Long Term Support (LTS), 27
- loopback interface
 - availability, 364
 - configuration, 364-365
- loops
 - Perl, 740-743
 - PHP, 787-789
 - Python, 763-765
- lp command, 489
- ipc command, 489
- lpq command, 489
- lprm command, 489
- lpstat command, 489
- ls command, 166-170, 202-204
- LTS (Long Term Support), 27
- LTSP (Linux Terminal Server Project), 627-633
 - features, 632-633
 - installation, 631-632
 - requirements, 628-631
- LTSP Display Manager (LDM), 632
- ltsp-server-standalone package, 631
- Lua, 824
- Lubuntu, 123-124
- LUGs (Linux User Groups), 44
- LXDE, 123-124
- machine protection, 422-425
- devices, 425
- passwords and physical security, 423-424
- Tripwire, 424-425
- wireless networks, 423
- macros, 808-809
- Mago, 727
- mail, sending in Perl, 745-747
- mail delivery agents (MDAs), 548-549, 559-560
- mail relaying, 554
- mail server options, configuring Fetchmail, 556-557
- mail transfer agents (MTAs), 545-548
- mail user agent (MUA), 549-550
- mailing lists, 46-47
- Major version (kernel), 455
- make command, 189, 807
- make utility (kernel), 450
- make xconfig tool, 461-462
- makefile targets, 809
- makefiles, 807-809
- man command, 41, 160, 190, 204-205
- man pages, 41, 160
- management
 - configuration, 150-151
 - kernel, 447-466
 - compiling the kernel, 457-465
 - Linux kernel, 448-452
 - modular kernels, 452-454
 - obtaining sources, 456
 - patching the kernel, 457-458
 - recompiling the kernel, 454-455
 - troubleshooting during compile, 465-466
 - versions, 455-456
 - modular kernels, 452-454
 - OpenStack instances, 670
 - OpenStack network, 671-672
 - OpenStack storage, 671
 - passwords, 246-251
 - changing in a batch, 251
 - password file, 247-248
 - policy, 246-247
 - security, 249-251
 - shadow passwords, 248-249
 - photos, Shotwell Photo Manager, 107

M

- MaaS (Metal as a Service), 648, 678
- MAC (Mandatory Access Control) system, AppArmor, 430-432
- Mac hardware, Ubuntu installation, 11
- Mac OS X, Juju on, 678

sets of servers
 Ansible, 683
 CfEngine, 683
 Chef, 682
 Juju, 681-682
 Landscape, 683
 Puppet, 682

software. See software management

users. See user management

Mandatory Access Control (MAC) system, AppArmor, 430-432

manual configuration
 loopback interface, 364-365
 PPPoE, 405-406
 quotas, 257-258
 Samba, 474-477

manual start, Apache web server, 497-498

mapping IP addresses to domain names, 685-686
 DNS records, 688-691
 DNS servers, 687-688
 server setup with BIND, 691-693

MariaDB, 581

masquerading, Postfix, 552

Master Boot Record (MBR), 13, 309

Masters of the Universe (MOTU), 25, 723

Math (LibreOffice component), 85

MBR (Master Boot Record), 13, 309

mc command, 356-358

MDAs (mail delivery agents), 548-549, 559-560

measuring MySQL buffer usage, 442-443

media, data backup, 342-344

Memcached, 611-612

MemcacheDB, 611-612

Mercurial, 698-699

message body (newsgroup articles), 81

message delivery interval (email), 553-554

Metacity, 120

Metal as a Service (MaaS), 648, 678

methods
 Python lists, 762
 Python strings, 759

Microsoft Exchange Server, 560-562

Microsoft Windows, productivity applications, 91-92

Midnight Commander, copying files, 356-358

MIME (Multipurpose Internet Mail Extensions), 81

Minor version (kernel), 455

Mir, 50

mirroring data, 342

mkdir command, 171, 175-176, 190, 205

mnemonic characters, 170

mobile development for Android, 829-835
 Android Runtime, 830
 Application Framework, 830-831
 core applications, 831
 creating application, 834-835
 installation of Android SDK, 831-833
 libraries, 830
 Linux kernel, 830

mobile devices, security, 424

mobile network sniffing, 423

mod_access (Apache), 510

mod_alias (Apache), 511

mod_asis (Apache), 511

mod_auth (Apache), 511

mod_auth_anon (Apache), 512

mod_auth_dbm (Apache), 512

mod_auth_digest (Apache), 512

mod_cgi (Apache), 512

mod_dir (Apache), 512

mod_env (Apache), 512

mod_expires (Apache), 513

mod_headers (Apache), 513

mod_include (Apache), 513

mod_info (Apache), 513

mod_log_config (Apache), 513

mod_mime (Apache), 513

mod_mime_magic (Apache), 513

mod_negotiation (Apache), 513

mod_proxy (Apache), 514

mod_rewrite (Apache), 514

mod_setenvif (Apache), 514

mod_speling (Apache), 514

mod_ssl (Apache), 514

mod_status (Apache), 514

mod_unique_id (Apache), 515

mod_userdir (Apache), 515

mod_usertrack (Apache), 515

mod_vhost_alias (Apache), 515

modular kernels, management, 452-454

Module section (`xorg.conf` file), 53-54

- modules
 - Apache web server, 510-515
 - disabling, 510
 - enabling, 510
 - Nginx, 530
 - Perl, 745
- MongoDB, 614
- Monitor section (`xorg.conf` file), 53, 55-56
- monitoring tools, 317-334
 - console-based monitoring, 317-324
 - disk quotas, 324
 - disk space, 323-324
 - free and used memory, 323
 - kill command, 319-320
 - log files, checking, 324-326
 - priority scheduling and control, 320-322
 - enterprise servers, 333-334
 - graphical process, 326-333
 - Conky, 328-333
 - System Monitor, 326
 - KDE process, 333
 - user activity, 246
- Mono, 825
- MOTU (Masters of the Universe), 25, 723
- mounting shares, Samba, 479-480
- 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, 503
- MTAs (mail transfer agents), 545-548
- mtr tool, 366-368
- MUA (mail user agent), 549-550
- multicast addressing, 376
- multiline loops, Python, 764
- multimedia applications, 93-116
 - burning CDs/DVDs, 107-112
 - digital cameras, 106-107
 - graphics manipulation, 100-105
 - sound and music, 93-100
 - viewing video, 112-116
- multiple terminals, command line, 227-229
- multiprocessing modules (MPMs), Apache web server configuration, 503
- Multipurpose Internet Mail Extensions (MIME), 81
- Multiverse repository, 142
- music, 93-100
 - adjusting volume, 95
 - buying, 99
 - listening to, 97-100
 - sound cards, 94-95
 - sound formats, 96-97
- Mutt mail client, 75
- mv command, 177-178, 190, 205
- MX record (DNS), 689
- MySQL
 - client/server system, 599-605
 - command-line client, 603-604
 - configuration, 593-596
 - future of, 581
 - performance tuning, 442-446
 - measuring buffer usage, 442-443
 - query cache, 444-445
 - query optimization, 446
 - versus PostgreSQL, 590-593
- mysql -u root command, 594

N

- Nagios, 334
- name-based virtual hosts (Apache), 516-517
- nano text editor, 221-222
- NAT (Network Address Translation), 371
- nautilus-image-converter package, 105
- navigating Linux file system, 166-169
- NcFTPd, 537
- negative indexes, Python strings, 759
- Neo4j, 616
- Nessus, assessing security vulnerabilities, 422
- NetBeans, 823
- netpbm tools, 105
- Network Address Translation (NAT), 371
- network cable, 379-380
- Network File System. See NFS (Network File System)
- Network Interface Cards. See NICs (Network Interface Cards)

network interface configuration, command line, 384-388
 network management, OpenStack, 671-672
 Network Manager, configuring wireless networks, 31-32
 Network News Transfer Protocol (NNTP), 81
 network storage, 343
 networking
 checking connections, 366-368
 configuration tools, 383-392
 DHCP, 393-399
 hardware, 377-383
 hubs and switches, 380-381
 initializing new hardware, 381-383
 network cable, 379-380
 NIC (Network Interface Cards), 377-379
 routers and bridges, 381
 Internet connectivity, 402-408
 commands, 408
 common configuration information, 402-403
 Dial-Up access, 406-407
 DSL access, 404-406
 troubleshooting connection problems, 407-408
 IPv6, 372-375
 localhost interface, 364-365
 loopback interface availability, 364
 loopback interface configuration, 364-365
 organization, 375-376
 broadcast addressing, 376
 multicast addressing, 376
 subnet masks, 376
 subnetting, 375-376
 Unicast addressing, 376
 printers, 483-489
 avoiding support problems, 488-489
 creating network printers, 484-485
 CUPS, 486-488
 TCP/IP, 368-372
 addressing, 369-370
 IP masquerading, 370
 ports, 372
 wireless, 399-402

newsgroups, Usenet, 79-82
 NewSQL, 609
 newusers command, 258
 next looping construct (Perl), 742
 Nexus, 130
 NFS (Network File System), 470-472
 client configuration, 472
 installation, 470
 server configuration, 470-472
 Nginx, 521-523
 configuration, 524-527
 installation, 523-524
 modules, 530
 PHP setup, 528-529
 virtual hosting, 527-528
 nice command, 229, 320-321
 NICs (Network Interface Cards), 377-379
 10BASE-T, 377
 100BASE-T, 377-378
 1000BASE-T, 378
 Token Ring, 377
 Nmap, assessing security vulnerabilities, 422
 NNTP (Network News Transfer Protocol), 81
 node controller, Eucalyptus, 651
 nodes, Eucalyptus, 651
 nonbooting hard drive, saving files, 362
 NoSQL, 583
 NoSQL databases, 607-617
 document stores, 612-615
 graph stores, 616-617
 key/value stores, 610-612
 wide column stores, 615-616
 Nova (Compute Infrastructure) service, OpenStack, 656-657
 NS record (DNS), 689-690
 number comparison, writing shell scripts, 288-289, 292-293
 numbers, Python, 755-757
 Nvidia, proprietary drivers, 128

O

object orientation (Python), 766-770
 class and object variables, 767-768
 class inheritance, 769-770
 constructors and destructors, 768-769

- object variables, Python object orientation, 767-768
- obtaining sources (kernel), 456
- octal characters, 171
- Ogg-Vorbis (sound format), 96
- OGV/OGG (video format), 114
- OneConf, 151
- one-liners, Perl, 749-750
- oops (kernel), compiling kernel, 466
- Open Sound System (OSS), 94
- open source, proprietary drivers versus, 26
- OpenLDAP, 619
- OpenSSH server, 411
- openssh-server package, 631
- OpenStack, 645, 656-672
 - commands, 672
 - Compute Infrastructure (Nova), 656-657
 - creating an image, 667-670
 - Imaging Service (Glance), 657
 - installation, 657-667
 - instance management, 670
 - network management, 671-672
 - Storage Infrastructure (Swift), 657
 - storage management, 671
- Open-Xchange, 562
- operation, Postfix, 550-555
- operators
 - Perl, 736
 - additional operators, 736
 - arithmetic operators, 736
 - comparison operators, 736-737
 - compound operators, 736
 - PHP programming, 781-783
- opportunistic development, 695-713
 - Bikeshed, 711-713
 - Ground Control, 707-711
 - Launchpad, 701-702
 - Quickly, 703-706
 - version control systems, 696-700
 - Bazaar, 697-698
 - Git, 699-700
 - Mercurial, 698-699
 - Subversion, 696-697
- optimization, 435-446
 - Apache, 441-442
 - hard disk, 435-439
- badblocks command, 439
- disabling file access time, 439
- e2fsck command, 439
- file system settings, 438
- hdparm command, 437-438
- tune2fs command, 438-439
- tuning disk drives, 436-437
- kernel, 440-441
- MySQL, 442-446
 - measuring buffer usage, 442-443
 - query cache, 444-445
 - query optimization, 446
- Options directive, Apache web server configuration, 504
- Oracle Beehive, 562
- order statement, Apache web server access control, 506
- organization, networking, 375-376
 - broadcast addressing, 376
 - multicast addressing, 376
 - subnet masks, 376
 - subnetting, 375-376
 - Unicast addressing, 376
- OrientDB, 616
- OSS (Open Sound System), 94
- Outlook, Microsoft Exchange Server, 561
- output, redirecting, 210-211, 271-272

P

- PaaS (Platform as a Service), 647
- Package Browsing screen (Software Center), 139
- packaging code, 720-722
- packaging-dev package, installation, 717-718
- packet writing, creating DVDs from command line, 112
- PAM (Pluggable Authentication Modules), 250
- panel (Unity desktop), 64
- parameters, positional, executing shell scripts, 278-279
- partition strategies, Ubuntu installation, 12-13
- passwd command, 163, 186
- password file, 247-248
- passwords, 18

management, 246-251
 changing in a batch, 251
 password file, 247-248
 policy, 246-247
 security, 249-251
 shadow passwords, 248-249

MySQL root users, 594
 security, 423-424

pastebinit tool, 712

patch command, 458
 patching the kernel, 457-458

PATH environment variable, 217

pattern-matching support (shells), 270-271

pb get tool (*Bikeshed*), 712

pbsput tool (*Bikeshed*), 712

pbsputs tool (*Bikeshed*), 712

pbuilder, 717

pci=biosirq command, 437

pconfiguration, system settings, power management, 29

PCRE (Perl-Compatible Regular Expressions), 799

PCX (graphics format), 104

PDF, 89-90

PDP-11 minicomputers, 818

PEAR project, 802

Percona Server, 581

performance tuning, 435-446
 Apache, 441-442
 hard disk, 435-439
 badblocks command, 439
 disabling file access time, 439
 e2fsck command, 439
 file system settings, 438
 hdparm command, 437-438
 tune2fs command, 438-439
 tuning disk drives, 436-437

kernel, 440-441

MySQL, 442-446
 measuring buffer usage, 442-443
 query cache, 444-445
 query optimization, 446

Perl (Practical Extraction and Report Language), 731-750
 accessing the shell, 744
 code examples, 745-750
 command-line errors, 733
 command-line processing, 750

commands, 750
 conditional statements, 739-740
 CPAN (Comprehensive Perl Archive Network), 745
 data structures, 734-736
 installation of CPAN module, 747

Linux, 731-734

looping constructs, 740-743

modules, 745

one-liners, 749-750

operators, 736
 additional operators, 736
 arithmetic operators, 736
 comparison operators, 736-737
 compound operators, 736

purging logs, 747-748

regular expressions, 743

sending mail, 745-747

simple Perl program example, 732-734

string constants, 736

Usenet posts, 748-749

variables, 734-736

versions, 732

Perl-Compatible Regular Expressions (PCRE), 799

permissions
 command line, 169-175
 assigning permissions, 170-171
 chmod command, 172-173
 directory permissions, 171-172
 file permissions, 173
 set UIDs/GID permissions, 173-175
 file, 236-237

personal home page programming. See PHP programming

personal package archive (PPA), Launchpad, 702

personal video recorders, 116

photo management, Shotwell Photo Manager, 107

Photoshop, 101

PHP mode, entering/exiting, 774

PHP programming, 773-804
 arrays, 775-777
 comments, 779
 conditional statements, 783-784
 constants, 778
 databases, 801-804

- entering/exiting PHP mode, 774
- escape sequences, 779-780
- functions, 790-801
 - arrays, 793-795
 - files, 795-797
 - miscellaneous functions, 797-800
 - strings, 790-793
- HTML forms, 801
- including other files, 789-790
- loops, 787-789
- operators, 781-783
- references, 778-779
- special operators, 784-785
- switching, 785-787
- variable substitution, 780-781
- variables, 774-776
- PHP setup, Nginx, 528-529
- phpgroupware, 562
- PHPorjekt, 562
- physical security, 423-424
- PID (process ID), 309
- Pidgin, 78
- ping command, 366-368
- piping data
 - command line, 214-215
 - shell control, 272
- Planet Debian, 77
- Planner (LibreOffice component), 86
- Platform as a Service (PaaS), 647
- Pluggable Authentication Modules (PAM), 250
- plug-ins, Firefox, 71
- PNG (graphics format), 104
- Point-to-Point Protocol over Ethernet, 404-406
 - policy, password management, 246-247
- POP3 (Post Office Protocol version 3), 549
- populaing directory, configuring LDAP server, 622-623
- portable anymap file format, 105
- portable bitmap file format, 105
- portable graymap file format, 105
- portable pixmap file format, 105
- ports, TCP/IP networking, 372
- positional parameters, executing shell scripts, 278-279
- POSIX Extended regular expressions, 799
- Post Office Protocol version 3 (POP3), 549
- postconf command, 551
- Postfix, 547
 - configuration and operation, 550-555
 - masquerading, 552
- PostgreSQL
 - client/server system, 599-605
 - command-line client, 604
 - configuration, 596-599
 - creating database, 597
 - deleting users, 598-599
 - versus MySQL, 590-593
 - users, 598
- posting to Usenet, Perl, 748-749
- post-installation configuration problems, 33-34
- POV-Ray, 105
- power management, configuring system settings, 29
- power shortcuts, Unity desktop, 66-67
- PPA (personal package archive), Launchpad, 702
- pppd daemon, 406
- PPPoE (Point-to-Point Protocol over Ethernet), 404-406
- Practical Extraction and Report Language.
 - See Perl (Practical Extraction and Report Language)
- preg_match() function (PCRE), 799
- preg_match_all() function (PCRE), 799
- preg_replace() function (PCRE), 800
- Prey, 424
- printenv command, 218, 253
- printers
 - configuring system settings, 28
 - sharing, 483-489
 - avoiding support problems, 488-489
 - creating network printers, 484-485
 - CUPS, 486-488
 - printing, commands, 489
- priorities for processes, setting, 229
- priority scheduling, console-based monitoring tools, 320-322
- private cloud, Eucalyptus, 650-654
- privileges
 - MySQL databases, 594
 - PostgreSQL databases, 599
- /proc directory, 164-165, 318
- procedural languages, MySQL versus PostgreSQL, 592-593

process ID (PID), 309
processes
 priorities, setting, 229
 threads versus, 522
Processes tab (System Monitor), 326
Procmail, 559
productivity applications, 83-92
 GNOME Office, 87-88
 KOffice, 87-89
 LaTeX, 91
 LibreOffice, 84-86
 Microsoft Windows, 91-92
 PDF, 89-90
 XML and DocBook, 89-90
ProFTPD, 537
programming languages, 817-827
 Ada, 818
 Clojure, 819
 COBOL, 819-820
 D, 820
 Erlang, 820-821
 Forth, 821
 Fortran, 822
 Go, 821-822
 Groovy, 822
 Haskell, 822-823
 Java, 823
 JavaScript, 823-824
 Lisp, 824
 Lua, 824
 Mono, 825
 Perl. See Perl (Practical Extraction and Report Language)
 reasons for learning, 818
 Ruby, 825-826
 Rust, 826
 Scala, 826
 Scratch, 826-827
 Vala, 827
programming tools, 805-813
 autoconf utility, 809-810
 C with Linux, 805-807
 debugging tools, 810-811
 GNU C compiler, 811-812
 graphical development, 812-813
 macros, 808-809
make command, 807
makefile targets, 809
makefiles, 807-809
programs
 command line, 185-186
 Ubuntu installation, 20
promiscuous mode, 425
proprietary drivers, open source versus, 26
proprietary video drivers, installation, 128-129
protection (security). See security
protection of data, MySQL versus PostgreSQL, 591-592
proxy servers, 565-573
 access control lists, 567-571
 client configuration, 566-567
 client IP addresses, 571-572
 defined, 565-566
 sample configuration, 572-573
 Squid installation, 566
ps command, 190, 206
public cloud, Eucalyptus, 650-651
PulseAudio, 94-95
Puppet, managing sets of servers, 682
Puppet Forge, 682
purging logs, Perl, 747-748
pwd command, 157, 169
PWD environment variable, 217
PyPI (Python Package Index), 771
Python, 703, 753-771
 conditional statements, 763-765
 dictionaries, 762-763
 executing scripts, 754-755
 functions, 765-766
 lists, 760-762
 loops, 763-765
 numbers, 755-757
 object orientation, 766-770
 class and object variables, 767-768
 class inheritance, 769-770
 constructors and destructors, 768-769
 PyPI (Python Package Index), 771
 Standard Library, 771
 strings, 757-760
Python 2.x, 753
Python 3.x, 753
Python Package Index (PyPI), 771

Q

Q Public License3
 QA, 725-730
 Bug Squad, 727
 QA Team, 726-727
 Test Drive, 727-730
 QA Team, 726-727
 Qmail, 547
 QML, 838-839
 QT (video format), 114
 Quassel, 78
 query cache, tuning MySQL, 444-445
 query optimization, MySQL, 446
 Quickly, 703-706, 704
 quotacheck command, 257
 quotaoff command, 257
 quotaon command, 257
 quotas, disk, 256-258

R

Radiance, 105
 RAID arrays, 342
 RAM disk image, configuring kernel, 462-465
 RAW (sound format), 96
 rcp command, 412
 RDBMSs (relational database management systems). See relational database services
 RDP (Remote Display Protocol), 642
 reading documentation, command line, 160-161
 recompiling the kernel, 454-455
 recorders, personal video, 116
 records (DNS), 688-691
 A, 688
 AAAA, 688
 CNAME, 689
 MX, 689
 NS, 689-690
 SOA, 690-691
 TXT, 691
 redirecting input and output
 command line, 210-211
 shell control, 271-272

Redis, 612
 reduced instruction set computer (RISC) processors, 830
 redundant array of independent disks. See RAID arrays
 references, PHP programming, 778-779
 regular expressions, 179, 743
 regular users, 235
 granting system administrator privileges to, 251-256
 changing UID, 251-253
 root privileges, 253-256
 REISUB (system reset), 229
 relational database services, 581-605
 client/server system, 599-605
 commands, 605
 creating tables, 586-587
 how they work, 584-586
 inserting data in tables, 587-588
 MySQL
 configuration, 593-596
 future of, 581
 versus PostgreSQL, 590-593
 PostgreSQL, 590-593, 596-599
 retrieving data, 588-590
 SQL basics, 586
 release tool (Bikeshed), 712
 release-build tool (Bikeshed), 712
 reload command (AppArmor), 432
 remote access, 409-417
 SSH (Secure Shell), 411-415
 Telnet, 409-411
 VNC (virtual network computing), 415-417
 remote computer, logging in/out of command line, 157-158
 Remote Desktop Viewer, 416
 Remote Display Protocol (RDP), 642
 remote file serving (FTP), 535-544
 configuring Very Secure FTP server, 540-543
 FTP, server selection, 535-537
 FTP users, 538-540
 ftphosts file, 543-544
 servers, 536-537
 software installation, 537-538
 remote printing, 483-489
 removable storage media, 342-343
 renice command, 229, 320-321

- repeat statement, executing shell scripts, 299
- repositories (software), 25-27
 - Apache web server installation, 493-494
 - compiling from source, 149-150
- repquota command, 257
- Requests for Comments (RFC), 81
- requirements
 - hardware specifications, 10
 - LTSP, 628-631
- rescue disk, 361
- researching hardware specifications, 10
- Resources tab (System Monitor), 326
- responsibilities, DBAs (database administrators), 582
- restoring
 - files from an archive, 346-347
 - GRUB2 boot loader, 361
- retrieving
 - data, 588-590
 - email, 555-559
- REVOKE statement
 - MySQL, 596
 - PostgreSQL, 599
- revoking privileges, PostgreSQL databases, 599
- RFC (Requests for Comments), 81
- Rhythmbox, 97-98
- Riak, 612
- rights
 - MySQL databases, 594
 - PostgreSQL databases, 599
- RISC (reduced instruction set computer) processors, 830
- rm command, 177, 190, 206-207
- rmdir command, 176
- root users, 234-236
 - command line, 159, 180-185
 - MySQL, passwords, 594
- root zone, 687
- Rootkit Hunter, 433
- rootsign tool, 713
- route command, 384-388
- routers, 381
- RSS feeds, 76-77
 - Firefox, 76
 - Liferea, 76-77
- rsync command, 356-358
- rtcwake command, 266-268
- Ruby, 825-826
- runlevels (BIOS), 310
 - changing, 312-313
 - definitions, 310
 - troubleshooting problems, 313-314
- running jobs repeatedly, cron command, 264-266
- running services, boot process, 307-308
- run-one tool, 713
- run-this-one tool, 713
- runtime errors, compiling kernel, 466
- runtime server configuration settings (Apache web server), 499-504
 - directives, 500
 - editing apache2.conf, 500-503
 - .htaccess configuration file, 503-504
 - multiprocessing modules, 503
- Rust, 826

S

- SaaS (Software as a Service), 647
- Samba, 480-483
 - configuration, SWAT, 480-483
 - global behavior, 476
 - manual configuration, 474-477
 - mounting shares, 479-480
 - smbd daemon, 478
 - testing, 477-478
- Samba Web Administration Tool (SWAT), 473
- satisfy directive, Apache web server, 509
- saving files, nonbooting hard drive, 362
- /sbin directory, commands, 161-163
- Scala, 826
- scalar variables (Perl), 735
- scanners, 103
- schemas, configuring LDAP server, 620-621
- Scorched 3D, 130-131
- scp command, 412
- Scratch, 826-827
- Screen, 78
- screen images, graphics manipulation, 105
- Screen section (xorg.conf file), 53, 56-57
- script kiddies, 420

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), 830
 Android, installation, 831-833
 installation, 831, 838

sdterr stream, redirection, 211-212

searches
 with Smart Scopes (Unity desktop), 66
 website search tips, 42-43

Secure Shell. See SSH (Secure Shell)

security, 419-433
 AppArmor, 430-432
 computer attacks, hacker versus cracker, 420
 disaster recovery plan, 432-433
 firewalls, 426-429
 machine protection, 422-425
 devices, 425
 passwords and physical security, 423-424
 Tripwire, 424-425
 wireless networks, 423
 passwords, 249-251
 viruses, 425-426
 vulnerability assessment, 421-422

sed text editor, 225

select statement, executing shell scripts, 299-300

sending mail, Perl, 745-747

Sendmail, 546-547

sequences (Python)
 lists, 760-762
 strings, 757-760

Server install DVD, 10

server packages, FTP, 536

ServerAdmin directive, Apache web server configuration, 502

ServerAGroup directive, Apache web server configuration, 501

ServerLayout section (`xorg.conf` file), 53

ServerName directive, Apache web server configuration, 502

ServerRoot directive, Apache web server configuration, 501

servers
 Apache web server. See Apache web server configuration for LDAP, 620-623
 DHCP, 396
 DNS, 687-688, 691-693
 FTP, 535-537
 HTTP, 531-534
 Apache Tomcat, 534
 Cherokee, 533
 Jetty, 533
 lighttpd, 531-532
 Nginx, 521-530
 thttpd, 534
 YAWS, 532

managing sets
 Ansible, 683
 CFEngine, 683
 Chef, 682
 Juju, 681-682
 Landscape, 683
 Puppet, 682

NFS, configuration, 470-472

SSH (Secure Shell), 411

Telnet, 409-410

service models (Cloud), 647-648

service set identifier (SSID), 32

service settings, `/etc/services` file, 389-390

services
 BIOS (basic input/output system), controlling, 312-313
 boot process, 307-308
 session message block printing, 485
 Session Message Block (SMB), 473
 session writing, creating DVDs from command line, 111

sets of servers, management
 Ansible, 683
 CFEngine, 683
 Chef, 682
 Juju, 681-682
 Landscape, 683
 Puppet, 682

sftp command, 411, 413

shadow passwords, 248-249, 539

shared data, /usr directory, 165-166

shares (Samba), mounting, 479-480
sharing
 files, 469-483
 NFS (Network File System), 470-472
 Samba, 480-483
 home directories, 476-477
 printers, 483-489
 avoiding support problems, 488-489
 creating network printers, 484-485
 CUPS, 486-488
SHELL environment variable, 217
Shell module, accessing the shell in Perl, 744
shells, 268-273. *See also* command line
 accessing in Perl, 744
 basic shell control, 268-273
 background processing, 272-273
 command line, 269-270
 pattern-matching support, 270-271
 piping data, 272
 redirecting input/output, 271-272
 endless loops, 297
 scripts, 273-305
 functions, 304-305
 writing and executing, 273-305
shift statement, executing shell scripts, 300
Shockwave, 71
shortcuts, Unity desktop, 66-67
Shotwell Photo Manager, 107
show status command (AppArmor), 432
shuffle() function (PHP), 794
shutdown command, 20, 184-185
shutting down, Ubuntu, 19-20
Simple Mail Transfer Protocol (SMTP), 545
simple strategy, data backup, 340-341
six-month cycle, development, 716-717
slapd package, 620
Slashdot, 77
Slashdot Effect, 441
sleep mode, waking computer from, 266-268
small enterprise users, backup programs, 342-343
small office users, backup programs, 342-343
smart hosts, 553
Smart Scopes, 66
SMB (Session Message Block), 473
smbclient command, 479
smbd daemon, 478
smbstatus command, 478
Smokin' Guns, 130
SMTP (Simple Mail Transfer Protocol), 545
SOA record (DNS), 690-691
socks-prox tool (Bikeshed), 712
software
 data backup, 344-353
 DHCP, 395-396
 installing FTP software, 537-538
 licensing, 2-3
 Software as a Service (SaaS), 647
 Software Center, 139-140
 software development kit (SDK). *See* SDK (software development kit)
 software management, 43, 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, 25-27
 Software Sources GUI tool, 26
 Software Updater, 21-24, 142-143
 sorting shell scripts, 275-276
sound, 93-100
 adjusting volume, 95
 formats, 96-97
 listening to music, 97-100
 sound cards, 94-95
sound cards, 94-95
Sound Juicer, 99
sound practices, backup programs, 338
SourceForge, 697
sources (kernel), 456
Spamassassin, 559-560
special characters, writing shell scripts, 283
special operators, PHP programming, 784-785
special variables (Perl), 735-736
speed, MySQL versus PostgreSQL, 590
Speed Dreams, 134-135
splint command, 810-811
SQL, basics, 586
SQL subqueries, MySQL versus PostgreSQL, 592
Squid installation, 566
Squirrelmail, 560
SSH (Secure Shell), 78, 411-415
 access to databases, 600-601

setting up server, 411
 Telnet versus, 411
 tools, 411-415
 ssh command, 408
 SSH key, 413
 sshd daemon, 411
 ssh-import-id tool, 713
 ssh-keygen command, 413-415
 SSID (service set identifier), 32
 Stallman, Richard M., 35
 standard input, 210
 Standard Library, Python, 771
 standard output, 210
 start command (AppArmor), 432
 starting
 Apache web server, 497-499
 NFS, 470
 X Server sessions, 57
 Startup Applications, 312-313
 stdin stream, redirection, 211-212
 stdout stream, redirection, 211-212
 step-by-step installation (Ubuntu), 14-19
 first update, 19
 installing, 15-19
 stereotypes (users), 237
 stop command (AppArmor), 432
 stopping
 Apache web server, 497-499
 NFS, 470
 Storage Infrastructure (Swift) service, OpenStack, 657
 storage management, OpenStack, 671
 str_replace() function (PHP), 791
 strategies, backup programs, 339-342
 stri_replace() function (PHP), 791
 string comparison, writing shell scripts, 286-288
 string constants, Perl, 736
 strings
 PHP functions, 790-793
 Python, 757-760
 strlen() function (PHP), 790
 strpos() function (PHP), 792
 StumbleUpon plug-in, 71
 su command, 251-253, 259
 subdirectories, /usr directory, 51-52
 Sublevel number version (kernel), 455
 subnet masks, 376
 subnetting, 375-376
 subsections (kernel), configuring kernel, 463
 substr() function (PHP), 792
 Subversion, 696-697
 sudo command, 24-25, 180-183, 234, 259
 sudo group, admin group versus, 181, 239
 Super User/Root User, 234-236
 super users, 158-159, 180-185, 235
 SuperTux, 132-133
 support
 network printers, 488-489
 wireless networking, 399-400
 suspend, 17
 SVG (graphics format), 104
 SWAT (Samba Web Administration Tool), 473, 480-483
 Swift (Storage Infrastructure) service, OpenStack, 657
 switch/case block, 786
 switches, 380-381
 switching, PHP programming, 785-787
 symbolic debugging, 811
 Synaptic, software management, 140-142
 SysAdmin, DevOps versus, 646
 system administrators, granting privileges to regular users, 251-256
 changing UID, 251-256
 root privileges, 253-256
 system function call, accessing the shell in Perl, 744
 System Monitor, 326
 system rescue, 360-362
 system reset, command line, 229
 system services, BIOS (basic input/output system), 310
 system settings
 configuration, 27-31
 power management, 29
 printers, 28
 time and date, 29-31
 Unity desktop, 65-66
 System tab (System Monitor), 326
 system users, 236
 system-monitoring tools. See monitoring tools
 system-search tool (Bikeshed), 712

T

tables, relational database services, 586-588
tail command, 190, 207
tape drives, 343-344
tar command, 227, 345-347, 354-355
Tarball, 148-149
task automation, 261-266
 running jobs repeatedly, 264-266
 scheduling tasks, 261-266
 waking computer from sleep, 266-268
 writing shell scripts, 280-282
TCP/IP (Transport Control Protocol/Internet Protocol), 368-372
 addressing, 369-370
 IP masquerading, 370
 ports, 372
Telnet
 setting up server, 409-410
 SSH versus, 411
temporary file storage, /tmp directory, 166
TERM environment variable, 217
terminal. See command line
ternary operator, 784-785
Test Drive tool, 727-730
testing, 725-730
 Bug Squad, 727
 community teams, 725-727
 Samba, 477-478
 Test Drive, 727-730
testparm command, testing Samba, 477-478
text editors (command line), 220-225
 emacs, 223-224
 nano, 222
 sed and awk, 225
 vi, 222-223
text-based console login, 156-157
threads, processes versus, 522
httpd, 534
Thunderbird, configuring for LDAP, 625
TIF (graphics format), 104
time, configuring system settings, 29-31
time command, 321
TLD (top-level domain), 687
/tmp directory, temporary file storage, 166
Token Ring, 377

tools

Eucalyptus, 654-656
group management, 238-240
SSH (Secure Shell), 411-415
system-monitoring. See monitoring tools
top command, 190, 207-209, 321
top-level domain (TLD), 687
Torvalds, Linus, 35
Totem Movie Player, 114
touch command, 169, 175, 414
touchscreen interface, 837-838
 application design guidelines, 839
 creating applications, 838-839
 SDK installation, 838
 User Interface Toolkit, 839-840
traceroute tool, checking network connections, 366-368
Transport Control Protocol/Internet Protocol.
 See TCP/IP
triggers, MySQL versus PostgreSQL, 592-593
trim() function (PHP), 790
triple quoting, Python strings, 759
Tripwire, 424-425
troubleshooting
 BIOS runlevel problems, 313-314
 Internet connection problems, 407-408
 kernel compile, 465-466
 post-installation configuration problems, 33-34
tune2fs command, 438-439
TuxPaint, 134
TV hardware, 112-114
TXT record (DNS), 691
types of kernels, 451-452

U

Ubuntu
 business usage, 39-40
 documentation, 40-42, 44
 helping with development, 42
 history, 38-39
 home usage, 40
 IRC (Internet Relay Chat), 47
 mailing lists, 46-47
 version numbers, 38-39

- websites, 42-46
 - commercial support, 44
 - Linux guides, 44-45
 - LUGs (Linux User Groups), 44
 - search tips, 42-43
 - Ubuntu-specific sites, 45-46
- Ubuntu Cloud. See Cloud
- Ubuntu GNOME, 124-125
- Ubuntu Kylin, 125
- Ubuntu Mobile/Touch, 837-838
 - application design guidelines, 839
 - creating applications, 838-839
 - SDK installation, 838
 - User Interface Toolkit, 839-840
- Ubuntu One cloud storage, 82
- Ubuntu One Music Store, 99
- Ubuntu Server Edition, 645
- Ubuntu Software Center, 139-140
- Ubuntu Testing Team, 726
- ubuntu-qa-tools, 727
- ubuntu-restricted-extras package, 114
- UDP (Universal Datagram Protocol), 368
- UEFI (Unified Extensible Firmware Interface), 14
- UFW (Uncomplicated Firewall), 426-427
- UIDs (user IDs), 173-175, 236
- umask command, 169
- uname command, 36
- Uncomplicated Firewall (UFW), 426-427
- unexpanded variables, writing shell scripts, 284-285
- Unicast addressing, 376
- Unified Extensible Firmware Interface (UEFI), 14
- Unison backup tool, 352
- Unity desktop, 49-67
 - customizing and configuring, 65-66
 - Dash, 61-63, 66
 - Launcher, 60
 - Mir, 50
 - panel, 64
 - power shortcuts, 66-67
- Ubuntu Mobile/Touch development, 837-838
 - application design guidelines, 839
 - creating applications, 838-839
 - SDK installation, 838
 - User Interface Toolkit, 839-840
- workspaces, 60
- 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, 66
- Universal Datagram Protocol (UDP), 368
- Universe repository, 25, 142
- unless conditional statements, Perl, 740
- UnQL (Unstructured Query Language), 610
- unset() function (PHP), 798
- Unshielded Twisted Pair (UTP) network cable, 379-380
- until looping construct (Perl), 742
- until statement, 298-299
- UPG (user private group), 237
- upgrade option, 24
- Upstart, 315-316
- uptime command, 323
- uquick tool (Bikeshed), 712
- Urban Terror, 130
- USB drive, Ubuntu installation, 14
- Usenet newsgroups, 79-82
- Usenet posts, Perl, 748-749
- user accounts, 233-237
 - adding, 242-245
 - command line, 158-159
 - commands, 258-259
 - configuring Fetchmail, 557-559
 - file permissions, 236-237
 - stereotypes, 237
 - Super User/Root User, 234-236
 - user IDs/group IDs, 236
- User directive, Apache web server configuration, 501
- user directories, /home directory, 163-164
- USER environment variable, 217
- user files, Apache web server, 508-509
- user IDs (UIDs), 236
- User Interface Toolkit, 839-840
- user management, 233-259
 - disk quotas, 256-258
 - groups, 237-240

- group listing, 237-238
- management tools, 238-240
- passwords, 246-251
 - changing in a batch, 251
 - password file, 247-248
 - policy, 246-247
 - security, 249-251
 - shadow passwords, 248-249
- system administrator privileges, 251-256
 - root privileges, 253-256
 - temporarily changing user identity, 251-253
- user accounts, 233-237
 - commands, 258-259
 - file permissions, 236-237
 - stereotypes, 237
 - Super User/Root User, 234-236
 - user IDs/group IDs, 236
- user private group (UPG), 237
- useradd command, 239-240, 259
- UserDir directive, Apache web server configuration, 502
- usermod command, 239, 242, 259
- users
 - FTP, 538-540
 - PostgreSQL, 598-599
- /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/src/linux-3.2 directory, 449
- UTP (Unshielded Twisted Pair) network cable, 379-380
- executing shell scripts, 282-283
- LogFormat statements (Apache), 518-519
- Perl, 734-736
- PHP programming, 774-775
- shell scripts, 277
- variants (Ubuntu), 10
- VBA (Visual Basic for Applications), 83
- version control systems
 - data backup, 358-360
 - opportunistic development, 696-700
 - Bazaar, 697-698
 - Gilt, 699-700
 - Mercurial, 698-699
 - Subversion, 696-697
- versions
 - kernel, 455-456
 - Linux kernel, 36
 - Perl, 732
 - PHP programming, 773
 - Ubuntu, 38-39
- Very Secure FTP server, configuration, 540-543
- vi text editor, 222-223
- video
 - editing, 116
 - formats, 114
 - hardware, 112-114
 - personal video recorders, 116
 - viewing, 112-116
- video conferencing, Empathy, 77
- viewing video, 112-116
- vim command, 190
- vim text editor, 221
- virtual devices, installation, 833
- virtual hosting
 - Apache web server, 515-517
 - Nginx, 527-528
- virtual machines (VMs), 636, 645
- virtual network computing (VNC), 415-417
- virtual private networks. See VPNs (virtual private networks)
- VirtualBox, 641-642
- virtualization, 635-643
 - KVM (Kernel-based Virtual Machine), 637-641
 - VirtualBox, 641-642
 - VMware, 643
 - Xen, 643
- virus scanners, 426, 560

V

- Vala, 827
- /var directory, data files, 166
- var_dump() function (PHP), 798
- variable substitution, PHP programming, 780-781
- variables
 - class and object, Python object orientation, 767-768

viruses, 425-426
 Visual Basic for Applications (VBA), 83
 VMs (virtual machines), 636, 645
 vmstat tool, 323
 VMware, 643
 VNC (virtual network computing), 415-417
 vncviewer, 333
 volume adjustment, music and sound, 95
 VPNs (virtual private networks), 574-579
 setting up clients, 575-577
 setting up servers, 577-579
 vsftpd server configuration files, 542-543
 VT-x extension, 637
 vulnerability assessment, 421-422

W

waking computer from sleep, 266-268
 war driving, 423
 Warsow, 130
 WAV (sound format), 96
 web access, databases, 602
 web scripting. See PHP programming
 web servers, 531-534
 Apache. See Apache web server
 Apache Tomcat, 534
 Cherokee, 533
 Jetty, 533
 lighttpd, 531-532
 Nginx, 521-523
 configuration, 524-527
 installation, 523-524
 modules, 530
 PHP setup, 528-529
 virtual hosting, 527-528
 thttpd, 534
 YAWS, 532
 WEBM (video format), 114
 websites, 42-46
 commercial support, 44
 Linux guides, 44-45
 LUGs (Linux User Groups), 44
 search tips, 42-43
 Ubuntu-specific sites, 45-46

WEP encryption, 32
 what-provides tool (Bikeshed), 712
 whereis command, 161
 which command, 190, 210
 while loop
 Perl, 741-742
 PHP, 787
 Python, 764
 while statement, 297-298
 wide column stores, NoSQL databases, 615-616
 wifi-status tool (Bikeshed), 712
 wildcards, 179
 window managers, X Server, 58
 Windows games, 136
 Wine application, 92, 127
 wireless network interfaces, 378-379
 wireless networks, 399-402
 advantages, 401
 choosing protocol, 401-402
 configuration, 31-32
 security, 423
 Wireshark, 423, 539
Wolfenstein: Enemy Territory, 130
 workspaces (Unity desktop), 60
World of Padman, 130
 worms, 420
 WPA encryption, 32
 Writer (LibreOffice component), 85
 writing shell scripts, 273-305
 accessing variable values, 278
 assigning value to variables, 278
 automation of tasks, 280-282
 backslash, 285
 backtick, 286
 break statement, 304
 built-in variables, 282-283
 case statement, 302-303
 comparison of expressions, 286-291
 comparison of expressions with tcsh, 291-295
 exit statement, 304
 for statement, 295-296
 if statement, 300-301
 interpreting shell scripts, 276-277
 positional parameters, 278-279

repeat statement, 299
running shell program, 274-275
select statement, 299-300
shift statement, 300
sorting scripts for access, 275-276
special characters, 283
strings with embedded spaces, 284
unexpanded variables, 284-285
until statement, 298-299
variables, 277
while statement, 297-298

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

YAWS (Yet Another Web Server), 532

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, 54-54
 Monitor section, 53, 55-56
 Screen section, 53, 56-57
 ServerLayout section, 53
starting X sessions, 57
X.org software, 51-52
Xamarin, 825
Xara Xtreme, 105
xargs command, 190
XChat, 78-79
XChat Channel List window, 79
Xen, 643
Xfce, 122-123
Xmarks Sync plug-in, 71
XML and DocBook, 89-90
XML Copy Editor, 90
X.org software, 51-52

Z

Zenoss, 334
zless command, 41