Matthew Helmke

Ubuntu

UNLEASHED

SAMS

2012 Edition

Covering 11.10 and 12.04

Matthew Helmke
with Andrew Hudson
and Paul Hudson

Ubuntu

UNLEASHED

2012 Edition
Covering 11.10 and 12.04

SAMS | 800 East 96th Street, Indianapolis, Indiana 46240 USA

Ubuntu Unleashed 2012 Edition: Covering Ubuntu 11.10 and 12.04
Copyright © 2012 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: 978-0-672-33578-5
ISBN-10: 0-672-33578-6

Library of Congress Cataloging-in-Publication Data:

Helmke, Matthew.
Ubuntu unleashed / Matthew Helmke. — 2012 ed.
p. cm.
"Covering 11.10 and 12.04."
ISBN-13: 978-0-672-33578-5 (pbk. : alk. paper)
ISBN-10: 0-672-33578-6 (pbk. : alk. paper)
1. Ubuntu (Electronic resource) 2. Linux. 3. Operating systems (Computers) I. Title.
QA76.76.O63U36 2012
005.4'32—dc23

2011041953

Printed in the United States of America

First Printing: January 2012

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson Education, Inc. 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 authors 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 from the use of the CD or programs accompanying it.

Bulk Sales

Pearson 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
+1-317-581-3793
international@pearsontechgroup.com

Editor-in Chief
Mark Taub

Executive Editor
Debra Williams Cauley

Senior Development Editor
Chris Zahn

Managing Editor
Kristy Hart

Project Editor
Andrew Beaster

Copy Editor
Keith Cline

Indexer
Christine Karpeles

Proofreader
Water Crest Publishing

Technical Editors
Kris Healy
Dustin Kirkland
John Wregglesworth

Publishing Coordinator
Kim Boedigheimer

Multimedia Developer
Dan Scherf

Interior Designer
Gary Adair

Cover Designer
Gary Adair

Compositor
Nonie Ratcliff

Contents at a Glance

About the Authors xxvii
Dedication/Acknowledgements xxviii
Introduction 1

Part I Installation and Configuration

1 Installing Ubuntu 7
2 Post-Installation Configuration 19

Part II Desktop Ubuntu

3 Working with Unity 35
4 On the Internet 53
5 Productivity Applications 67
6 Multimedia Applications 77
7 Other Ubuntu Desktops 101
8 Games 109

Part III System Administration

9 Managing Software 119
10 Command-Line Quickstart 133
11 Command-Line Master Class 165
12 Managing Users 203
13 Automating Tasks and Shell Scripting 231
14 The Boot Process 275
15 System-Monitoring Tools 285
16 Backing Up 301
17 Networking 329
18 Remote Access with SSH and Telnet 373
19 Securing Your Machines 383
20 Performance Tuning 399
21 Kernel and Module Management 411

Part IV Ubuntu as a Server

22 File and Print ... 433
23 Apache Web Server Management ... 455
24 Other HTTP Servers ... 485
25 Remote File Serving with FTP ... 491
26 Handling Email ... 501
27 Proxying and Reverse Proxying ... 521
28 Administering Relational Database Services ... 531
29 NoSQL Databases ... 557
30 Lightweight Directory Access Protocol (LDAP) ... 567
31 Linux Terminal Server Project (LTSP) ... 575
32 Virtualization on Ubuntu ... 583
33 Ubuntu in the Cloud ... 593

Part V Programming Linux

34 Opportunistic Development ... 623
35 Helping with Ubuntu Testing and QA ... 643
36 Using Perl ... 649
37 Using PHP ... 671
38 Using Python ... 705
39 C/C++ Programming Tools for Ubuntu ... 725
40 Using Mono ... 737
41 Using Other Popular Programming Languages ... 749
42 Beginning Mobile Development for Android ... 759

Part VI Appendices

A Ubuntu Under the Hood ... 767
B Ubuntu and Linux Internet Resources ... 777

Index ... 787

Table of Contents

Part I Installation and Configuration

1 Installing Ubuntu 7

Before You Begin the Installation 7
Researching Your Hardware Specifications 8
Installation Options 8
Planning Partition Strategies 10
The Boot Loader 10
Installing from CD or DVD or USB Drive 11
Step-by-Step Installation 11
Installing 12
First Update 16
Wubi: The Easy Installer for Windows 16
Shutting Down 17
References 18

2 Post-Installation Configuration 19

Troubleshooting Post-Installation Configuration Problems 19
The **sudo** Command 21
Finding Programs and Files 21
Software Update 22
Configuring Software Repositories 25
System Settings 28
Installing Additional Drivers 28
Detecting and Configuring a Printer 30
Configuring Power Management in Ubuntu 30
Setting the Date and Time 31
Configuring Wireless Networks 33
References 34

Part II Desktop Ubuntu

3 Working with Unity 35

Foundations and the X Server 35
Basic X Concepts 36
Using X 37

Elements of the **xorg.conf** File ... 38
Starting X ... 43
Using a Display Manager ... 43
Changing Window Managers ... 44
Using Unity, a Primer ... 44
The Desktop ... 45
Customizing and Configuring Unity ... 50
Power Shortcuts ... 50
References ... 51

4 On the Internet **53**

Getting Started with Firefox ... 54
Checking Out Google Chrome and Chromium ... 55
Choosing an Email Client ... 56
Evolution ... 57
Mozilla Thunderbird ... 59
Other Mail Clients ... 59
RSS Readers ... 60
Firefox ... 60
Liferea ... 60
Instant Messaging and Video Conferencing with Empathy ... 61
Internet Relay Chat ... 61
Usenet Newsgroups ... 64
Ubuntu One Cloud Storage ... 66
References ... 66

5 Productivity Applications **67**

Introducing LibreOffice ... 69
Other Office Suites for Ubuntu ... 71
Working with GNOME Office ... 71
Working with KOffice ... 72
Other Useful Productivity Software ... 73
Working with PDF ... 73
Working with XML and DocBook ... 74
Working with LaTeX ... 75
Productivity Applications Written for Microsoft Windows ... 76
References ... 76

6 Multimedia Applications **77**

Sound and Music ... 77
Sound Cards ... 78
Adjusting Volume ... 79

Sound Formats 79
Listening to Music 81
Buying Music in the Ubuntu One Music Store 83
Graphics Manipulation 85
The GNU Image Manipulation Program 85
Using Scanners in Ubuntu 87
Working with Graphics Formats 87
Capturing Screen Images 89
Using Digital Cameras with Ubuntu 90
Handheld Digital Cameras 90
Using Shotwell Photo Manager 90
Burning CDs and DVDs in Ubuntu 90
Creating CDs and DVDs with Brasero 91
Creating CDs from the Command Line 92
Creating DVDs from the Command Line 94
Viewing Video 96
TV and Video Hardware 96
Video Formats 97
Viewing Video in Linux 98
Personal Video Recorders 99
Video Editing 99
References 100

7 Other Ubuntu Desktops 101

Desktop Environment 101
KDE and Kubuntu 102
Xfce and Xubuntu 104
LXDE and Lubuntu 105
GNOME 106
References 107

8 Games 109

Ubuntu Gaming 109
Emulators 109
Installing Proprietary Video Drivers 110
Installing Games in Ubuntu 112
Warsow 112
Scorched 3D 112
Frozen Bubble 113
SuperTux 114
Battle for Wesnoth 114
Frets on Fire 116

Games for Kids ... 116
Commercial Games ... 116
Playing Windows Games ... 117
References ... 117

Part III System Administration

9 Managing Software 119

Ubuntu Software Center ... 119
Using Synaptic for Software Management ... 120
Staying Up-to-Date ... 122
Working on the Command Line ... 123
Day-to-Day Usage ... 124
Finding Software ... 127
Compiling Software from Source ... 128
Compiling from a Tarball ... 128
Compiling from Source from the Ubuntu Repositories ... 129
Server/Configuration Management ... 130
Puppet ... 131
Chef ... 131
Juju ... 131
Landscape ... 131
dotdee ... 131
References ... 132

10 Command-Line Quickstart 133

What Is the Command Line? ... 134
Accessing the Command Line ... 135
Text-Based Console Login ... 136
Logging Out ... 137
Logging In and Out from a Remote Computer ... 137
User Accounts ... 138
Understanding the Linux File System Hierarchy ... 140
Essential Commands in **/bin** and **/sbin** ... 141
Configuration Files in **/etc** ... 141
User Directories: **/home** ... 142
Using the Contents of the **/proc** Directory to Interact with the Kernel ... 142
Working with Shared Data in the **/usr** Directory ... 144
Temporary File Storage in the **/tmp** Directory ... 144
Accessing Variable Data Files in the **/var** Directory ... 144

Navigating the Linux File System 145
Listing the Contents of a Directory with **ls** 145
Changing Directories with **cd** 147
Finding Your Current Directory with **pwd** 147
Working with Permissions 148
Assigning Permissions 148
Directory Permissions 150
Altering File Permissions with **chmod** 150
File Permissions with **chgrp** 152
Changing File Permissions with **chown** 152
Understanding Set User ID and Set Group ID Permissions 152
Working with Files 153
Creating a File with **touch** 153
Creating a Directory with **mkdir** 154
Deleting a Directory with **rmdir** 154
Deleting a File or Directory with **rm** 155
Moving or Renaming a File with **mv** 156
Copying a File with **cp** 157
Displaying the Contents of a File with **cat** 157
Displaying the Contents of a File with **less** 157
Using Wildcards and Regular Expressions 158
Working as Root 158
Creating Users 159
Deleting Users 160
Shutting Down the System 161
Rebooting the System 162
Reading Documentation 162
Using **apropos** 162
Using Man Pages 163
References 164

11 Command-Line Master Class 165

Why Use the Command Line? 166
Using Basic Commands 167
Printing the Contents of a File with **cat** 168
Changing Directories with **cd** 169
Changing File Access Permissions with **chmod** 171
Copying Files with **cp** 171
Printing Disk Usage with **du** 172
Finding Files by Searching with **find** 173
Searches for a String in Input with **grep** 175
Paging Through Output with **less** 176

Creating Links Between Files with **ln** 178
Finding Files from an Index with **locate** 180
Listing Files in the Current Directory with **ls** 180
Reading Manual Pages with **man** 182
Making Directories with **mkdir** 183
Moving Files with **mv** 183
Listing Processes with **ps** 184
Deleting Files and Directories with **rm** 184
Printing the Last Lines of a File with **tail** 185
Printing Resource Usage with **top** 186
Printing the Location of a Command with **which** 187
Redirecting Output and Input 187
Combining Commands 189
Using Environment Variables 191
Using Common Text Editors 194
Working with **nano** 195
Working with **vi** 196
Working with **emacs** 197
Working with Compressed Files 199
Using Multiple Terminals with **byobu** 200
References 201

12 Managing Users 203

User Accounts 203
The Super User/Root User 204
User IDs and Group IDs 206
File Permissions 206
Managing Groups 207
Group Listing 207
Group Management Tools 208
Managing Users 210
User Management Tools 210
Adding New Users 212
Monitoring User Activity on the System 216
Managing Passwords 217
System Password Policy 217
The Password File 217
Shadow Passwords 218
Managing Password Security for Users 221
Changing Passwords in a Batch 221

Granting System Administrator Privileges to Regular Users 221
Temporarily Changing User Identity with the **su** Command 222
Granting Root Privileges on Occasion: The **sudo** Command 224
Disk Quotas 226
Implementing Quotas 227
Manually Configuring Quotas 228
Related Ubuntu Commands 228
References 229

13 Automating Tasks and Shell Scripting 231

Scheduling Tasks 231
Using **at** and **batch** to Schedule Tasks for Later 231
Using **cron** to Run Jobs Repeatedly 234
Basic Shell Control 236
The Shell Command Line 237
Shell Pattern-Matching Support 239
Redirecting Input and Output 240
Piping Data 241
Background Processing 241
Writing and Executing a Shell Script 242
Running the New Shell Program 243
Storing Shell Scripts for Systemwide Access 244
Interpreting Shell Scripts Through Specific Shells 245
Using Variables in Shell Scripts 246
Assigning a Value to a Variable 246
Accessing Variable Values 247
Positional Parameters 247
A Simple Example of a Positional Parameter 248
Using Positional Parameters to Access and Retrieve Variables from the Command Line 248
Using a Simple Script to Automate Tasks 249
Built-In Variables 251
Special Characters 252
Using Double Quotes to Resolve Variables in Strings with Embedded Spaces 253
Using Single Quotes to Maintain Unexpanded Variables 253
Using the Backslash as an Escape Character 254
Using the Backtick to Replace a String with Output 255
Comparison of Expressions in **pdksh** and **bash** 255
Comparing Expressions with **tcsh** 260
The **for** Statement 264

The **while** Statement 266
The **until** Statement 268
The **repeat** Statement (**tcsh**) 268
The **select** Statement (**pdksh**) 269
The **shift** Statement 269
The **if** Statement 270
The **case** Statement 271
The **break** and **exit** Statements 273
Using Functions in Shell Scripts 273
References 274

14 The Boot Process 275

Running Services at Boot 275
Beginning the Boot Loading Process 276
Loading the Linux Kernel 277
System Services and Runlevels 278
Runlevel Definitions 278
Booting into the Default Runlevel 279
Understanding **init** Scripts and the Final Stage of Initialization 279
Controlling Services at Boot with Administrative Tools 280
Changing Runlevels 281
Troubleshooting Runlevel Problems 282
Starting and Stopping Services Manually 283
Using Upstart 283
References 284

15 System-Monitoring Tools 285

Console-Based Monitoring 285
Using the **kill** Command to Control Processes 287
Using Priority Scheduling and Control 288
Displaying Free and Used Memory with **free** 290
Disk Space 291
Disk Quotas 291
Graphical Process and System Management Tools 292
System Monitor 292
Conky 294
Other 298
KDE Process- and System-Monitoring Tools 299
Enterprise Server Monitoring 299
Landscape 299
Other 299
References 300

16 Backing Up 301

Choosing a Backup Strategy 301
Why Data Loss Occurs 302
Assessing Your Backup Needs and Resources 303
Evaluating Backup Strategies 305
Making the Choice 307
Choosing Backup Hardware and Media 308
Removable Storage Media 308
CD-RW and DVD+RW/-RW Drives 308
Network Storage 309
Tape Drive Backup 309
Cloud Storage 310
Using Backup Software 310
tar: The Most Basic Backup Tool 310
The GNOME File Roller 312
The KDE **ark** Archiving Tool 313
Déjà Dup 313
Back In Time 315
Unison 317
Using the Amanda Backup Application 318
Alternative Backup Software 319
Copying Files 319
Copying Files Using **tar** 319
Compressing, Encrypting, and Sending **tar** Streams 320
Copying Files Using **cp** 321
Copying Files Using **mc** 321
Using **rsync** 322
Version Control for Configuration Files 323
System Rescue 326
The Ubuntu Rescue Disc 326
Restoring the GRUB2 Boot Loader 326
Saving Files from a Nonbooting Hard Drive 327
References 328

17 Networking 329

Laying the Foundation: The **localhost** Interface 330
Checking for the Availability of the Loopback Interface 330
Configuring the Loopback Interface Manually 330
Networking with TCP/IP 333
TCP/IP Addressing 334
Using IP Masquerading in Ubuntu 336
Ports 336

Network Organization ... 337
Subnetting ... 337
Subnet Masks ... 338
Broadcast, Unicast, and Multicast Addressing ... 338
Hardware Devices for Networking ... 339
Network Interface Cards ... 339
Network Cable ... 341
Hubs and Switches ... 342
Routers and Bridges ... 343
Initializing New Network Hardware ... 344
Using Network Configuration Tools ... 346
Command-Line Network Interface Configuration ... 346
/bin/netstat ... 350
Network Configuration Files ... 351
Using Graphical Configuration Tools ... 353
Dynamic Host Configuration Protocol ... 355
How DHCP Works ... 355
Activating DHCP at Installation and Boot Time ... 356
DHCP Software Installation and Configuration ... 357
Using DHCP to Configure Network Hosts ... 359
Other Uses for DHCP ... 361
Wireless Networking ... 361
Support for Wireless Networking in Ubuntu ... 361
Advantages of Wireless Networking ... 363
Choosing from Among Available Wireless Protocols ... 363
Beyond the Network and onto the Internet ... 364
Common Configuration Information ... 364
Configuring Digital Subscriber Line Access ... 366
Understanding PPP over Ethernet ... 366
Configuring a PPPoE Connection Manually ... 367
Configuring Dial-Up Internet Access ... 368
Troubleshooting Connection Problems ... 369
References ... 370

18 Remote Access with SSH and Telnet 373

Setting Up a Telnet Server ... 373
Telnet Versus SSH ... 375
Setting Up an SSH Server ... 375
SSH Tools ... 375
Using **scp** to Copy Individual Files Between Machines ... 376
Using **sftp** to Copy Many Files Between Machines ... 377
Using **ssh-keygen** to Enable Key-Based Logins ... 377

Virtual Network Computing ... 379
References ... 382

19 Securing Your Machines 383

Understanding Computer Attacks ... 383
Assessing Your Vulnerability ... 385
Protecting Your Machine ... 386
Securing a Wireless Network ... 387
Passwords and Physical Security ... 387
Configuring and Using Tripwire ... 388
Devices ... 389
Viruses ... 389
Configuring Uncomplicated Firewall ... 390
AppArmor ... 392
Forming a Disaster Recovery Plan ... 395
References ... 396

20 Performance Tuning 399

Hard Disk ... 399
Using the BIOS and Kernel to Tune the Disk Drives ... 400
The **hdparm** Command ... 401
File System Tuning ... 402
The **tune2fs** Command ... 402
The **e2fsck** Command ... 403
The **badblocks** Command ... 403
Disabling File Access Time ... 403
Kernel ... 404
Apache ... 405
MySQL ... 406
Measuring Key Buffer Usage ... 406
Using the Query Cache ... 408
Miscellaneous Tweaks ... 409
Query Optimization ... 410
References ... 410

21 Kernel and Module Management 411

The Linux Kernel ... 412
The Linux Source Tree ... 413
Types of Kernels ... 415
Managing Modules ... 416
When to Recompile ... 418
Kernel Versions ... 419

Obtaining the Kernel Sources 420
Patching the Kernel 421
Compiling the Kernel 422
Using **xconfig** to Configure the Kernel 425
Creating an Initial RAM Disk Image 429
When Something Goes Wrong 429
Errors During Compile 429
Runtime Errors, Boot Loader Problems, and Kernel Oops 430
References 431

Part IV Ubuntu as a Server

22 File and Print 433

Using the Network File System 434
Installing and Starting or Stopping NFS 434
NFS Server Configuration 434
NFS Client Configuration 436
Putting Samba to Work 437
Manually Configuring Samba with **/etc/samba/smb.conf** 438
Testing Samba with the **testparm** Command 442
Starting, Stopping, and Restarting the **smbd** Daemon 442
Mounting Samba Shares 443
Configuring Samba Using SWAT 444
Network and Remote Printing with Ubuntu 448
Creating Network Printers 448
Using the Common UNIX Printing System GUI 450
Avoiding Printer Support Problems 453
References 454

23 Apache Web Server Management 455

About the Apache Web Server 455
Installing the Apache Server 457
Installing from the Ubuntu Repositories 457
Building the Source Yourself 458
Starting and Stopping Apache 460
Starting the Apache Server Manually 461
Using **/etc/init.d/apache2** 462
Runtime Server Configuration Settings 463
Runtime Configuration Directives 464
Editing **apache2.conf** 464
Apache Multiprocessing Modules 467
Using **.htaccess** Configuration Files 467

File System Authentication and Access Control 469
Restricting Access with **allow** and **deny** 470
Authentication 471
Final Words on Access Control 473
Apache Modules 474
mod_access 475
mod_alias 475
mod_asis 475
mod_auth 476
mod_auth_anon 476
mod_auth_dbm 476
mod_auth_digest 476
mod_autoindex 477
mod_cgi 477
mod_dir and **mod_env** 477
mod_expires 477
mod_headers 477
mod_include 478
mod_info and **mod_log_config** 478
mod_mime and **mod_mime_magic** 478
mod_negotiation 478
mod_proxy 478
mod_rewrite 478
mod_setenvif 479
mod_speling 479
mod_status 479
mod_ssl 479
mod_unique_id 479
mod_userdir 479
mod_usertrack 479
mod_vhost_alias 479
Virtual Hosting 480
Address-Based Virtual Hosts 480
Name-Based Virtual Hosts 481
Logging 482
References 484

24 Other HTTP Servers 485

Nginx 485
lighttpd 487
Yaws 488
Cherokee 488

Jetty ... 489
thttpd ... 489
Apache Tomcat ... 490
References ... 490

25 Remote File Serving with FTP 491

Choosing an FTP Server ... 491
Choosing an Authenticated or Anonymous Server ... 492
Ubuntu FTP Server Packages ... 492
Other FTP Servers ... 492
Installing FTP Software ... 493
The FTP User ... 494
Configuring the Very Secure FTP Server ... 496
Controlling Anonymous Access ... 497
Other **vsftpd** Server Configuration Files ... 498
Using the **ftphosts** File to Allow or Deny FTP Server Connection ... 499
References ... 500

26 Handling Email 501

How Email Is Sent and Received ... 501
The Mail Transport Agent ... 502
Choosing an MTA ... 504
The Mail Delivery Agent ... 504
The Mail User Agent ... 505
Basic Postfix Configuration and Operation ... 506
Configuring Masquerading ... 508
Using Smart Hosts ... 509
Setting Message Delivery Intervals ... 509
Mail Relaying ... 510
Forwarding Email with Aliases ... 510
Using Fetchmail to Retrieve Mail ... 511
Installing Fetchmail ... 511
Configuring Fetchmail ... 511
Choosing a Mail Delivery Agent ... 515
Procmail ... 515
Spamassassin ... 515
Squirrelmail ... 516
Virus Scanners ... 516
Autoresponders ... 516
Alternatives to Microsoft Exchange Server ... 516
Microsoft Exchange Server/Outlook Client ... 517
CommuniGate Pro ... 517

Oracle Beehive 517
Bynari 518
Open-Xchange 518
phpgroupware 518
PHProjekt 518
Horde 518
References 519

27 Proxying and Reverse Proxying 521

What Is a Proxy Server? 521
Installing Squid 522
Configuring Clients 522
Access Control Lists 523
Specifying Client IP Addresses 527
Sample Configurations 528
References 530

28 Administering Relational Database Services 531

A Brief Review of Database Basics 532
How Relational Databases Work 533
Understanding SQL Basics 536
Creating Tables 536
Inserting Data into Tables 537
Retrieving Data from a Database 538
Choosing a Database: MySQL Versus PostgreSQL 540
Speed 540
Data Locking 541
ACID Compliance in Transaction Processing to Protect Data Integrity 542
SQL Subqueries 542
Procedural Languages and Triggers 542
Configuring MySQL 543
Setting a Password for the MySQL Root User 544
Creating a Database in MySQL 544
Configuring PostgreSQL 546
Initializing the Data Directory in PostgreSQL 547
Creating a Database in PostgreSQL 547
Creating Database Users in PostgreSQL 548
Deleting Database Users in PostgreSQL 548
Granting and Revoking Privileges in PostgreSQL 549
Database Clients 550
SSH Access to a Database 550
Local GUI Client Access to a Database 551

Web Access to a Database 552
The MySQL Command-Line Client 553
The PostgreSQL Command-Line Client 555
Graphical Clients 555
References 556

29 NoSQL Databases 557

Key/Value Stores 559
Berkeley DB 560
Cassandra 560
Memcached and MemcacheDB 561
Redis 561
Document Stores 562
CouchDB 562
MongoDB 563
BaseX 564
Wide Column Stores 564
BigTable 565
HBase 565
References 565

30 Lightweight Directory Access Protocol (LDAP) 567

Configuring the Server 568
Creating Your Schema 568
Populating Your Directory 570
Configuring Clients 572
Evolution 572
Thunderbird 572
Administration 572
References 574

31 Linux Terminal Server Project (LTSP) 575

Requirements 576
Installation 579
Using LTSP 580
References 581

32 Virtualization on Ubuntu 583

KVM 585
VirtualBox 589
VMware 591
Xen 591
References 591

33 Ubuntu in the Cloud 593

Why a Cloud? 594
Ubuntu Cloud and Eucalyptus 595
Deploy/Install Basics: Public or Private? 596
Public 597
Private 598
A `euca2ools` Primer 601
Ubuntu Cloud and OpenStack 603
Compute Infrastructure (Nova) 603
Storage Infrastructure (Swift) 604
Imaging Service (Glance) 604
Installation 604
Creating an Image 614
Instance Management 617
Storage Management 617
Network Management 618
An OpenStack Commands Primer 618
Learning More 618
Landscape 619
Juju 619
Orchestra 620
References 620

Part V Programming Linux

34 Opportunistic Development 623

Version Control Systems 624
Managing Software Projects with Subversion 624
Managing Software Projects with Bazaar 625
Managing Software Projects with Mercurial 626
Managing Software Projects with Git 627
Introduction to Opportunistic Development 628
Launchpad 629
Quickly 631
Ground Control 635
Bikeshed and Other Tools 638
References 641

35 Helping with Ubuntu Testing and QA 643

Community Teams 643
Ubuntu Testing Team 644
QA Team 645

Bug Squad 645
Test Drive 645
References 648

36 Using Perl 649

Using Perl with Linux 649
Perl Versions 650
A Simple Perl Program 650
Perl Variables and Data Structures 652
Perl Variable Types 653
Special Variables 653
Operators 654
Comparison Operators 654
Compound Operators 655
Arithmetic Operators 655
Other Operators 656
Special String Constants 657
Conditional Statements: **if/else** and **unless** 657
if 657
unless 658
Looping 658
for 659
foreach 659
while 660
until 660
last and **next** 660
do ... while and **do ... until** 661
Regular Expressions 661
Access to the Shell 662
Modules and CPAN 663
Code Examples 664
Sending Mail 664
Purging Logs 666
Posting to Usenet 667
One-Liners 668
Command-Line Processing 668
References 669

37 Using PHP 671

Introduction to PHP 672
Entering and Exiting PHP Mode 672
Variables 673

Arrays 674
Constants 676
References 676
Comments 677
Escape Sequences 677
Variable Substitution 679
Operators 679
Conditional Statements 681
Special Operators 683
Switching 683
Loops 685
Including Other Files 687
Basic Functions 688
Strings 688
Arrays 692
Files 693
Miscellaneous 696
Handling HTML Forms 699
Databases 700
References 702

38 Using Python 705

Python on Linux 706
The Basics of Python 707
Numbers 707
More on Strings 709
Lists 712
Dictionaries 714
Conditionals and Looping 715
Functions 717
Object Orientation 718
Class and Object Variables 719
Constructors and Destructors 720
Class Inheritance 721
The Standard Library and the Python Package Index 722
References 723

39 C/C++ Programming Tools for Ubuntu 725

Programming in C with Linux 726
Using the C Programming Project Management Tools Provided with Ubuntu 727
Building Programs with **make** 727
Using Makefiles 727

Using the **autoconf** Utility to Configure Code 729
Debugging Tools 730
Using the GNU C Compiler 731
Graphical Development Tools 732
Using the KDevelop Client 732
The Glade Client for Developing in GNOME 733
References 734

40 Using Mono 737

Why Use Mono? 738
MonoDevelop 739
The Structure of a C# Program 741
Printing Out the Parameters 743
Creating Your Own Variables 743
Adding Some Error Checking 744
Building on Mono's Libraries 745
Creating a GUI with Gtk# 745
References 746

41 Using Other Popular Programming Languages 749

Ada 750
Clojure 750
COBOL 751
Erlang 752
Forth 752
Fortran 753
Groovy 753
Haskell 754
Java 754
JavaScript 755
Lisp 755
Lua 756
Ruby 756
Scala 756
Vala 757
References 757

42 Beginning Mobile Development for Android 759

Introduction to Android 760
Hardware 760
Linux Kernel 760

Libraries 760
Android Runtime 760
Application Framework 761
Applications 761
Installing the Android SDK 761
Install Java 761
Install Eclipse 761
Install the SDK 762
Install the ADT Eclipse Plug-In 762
Install Other Components 762
Install Virtual Devices 763
Create Your First Application 764
References 765

Part VI Appendices

A Ubuntu Under the Hood 767

What Is Linux? 767
Why Use Linux? 769
What Is Ubuntu? 770
Ubuntu for Business 771
Ubuntu in Your Home 773
64-Bit Ubuntu 773
Getting the Most from Ubuntu and Linux Documentation 773
Ubuntu Developers and Documentation 775
References 775

B Ubuntu and Linux Internet Resources 777

Websites and Search Engines 778
Web Search Tips 778
Google Is Your Friend 779
Ubuntu Package Listings 779
Certification 779
Commercial Support 780
Documentation 780
Linux Guides 781
Ubuntu 781
Mini-CD Linux Distributions 781
Various Intel-Based Linux Distributions 782
PowerPC-Based Linux Distributions 782

Linux on Laptops and PDAs ... 783
The X Window System ... 783
Usenet Newsgroups ... 783
Mailing Lists ... 784
Ubuntu Project Mailing Lists ... 785
Internet Relay Chat ... 785

Index **787**

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 for The iPlant Collaborative (www.iplantcollaborative.org), which is funded by the National Science Foundation and is building the world's first cyberinfrastructure for the biological sciences. 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 recently completed 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, and DevOps communities.

—Matthew Helmke

Acknowledgments

I am solely responsible for this edition of *Ubuntu Unleashed* but 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*) 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, especially Andrew Lenards, John Wregglesworth, Kris Healy, Dustin Kirkland, and several developers who wanted to remain anonymous. A huge debt of gratitude is owed 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. 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.

—Matthew Helmke

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.

I welcome your comments. You can email or write me directly to let me 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 I cannot help you with technical problems related to the topic of this book. We do have a User Services group, however, where I will forward specific technical questions related to the book.

Email: opensource@samspublishing.com

Mail: Debra Williams Cauley
Sams Publishing
800 East 96th Street
Indianapolis, IN 46240 USA

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

Reader Services

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

This page intentionally left blank

Introduction

We are pleased to present the 2012 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 allows 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 joining their ranks.

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

A QUICK WORD ABOUT MARKETING

Almost all of the content in this book will apply 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 12.04 is true in that we will send an upgrade kit to anyone who has purchased and registered the book. The kit will include install media for the 12.04 release and a supplemental guide to the changes from 11.10 to 12.04.

Do not let the 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

There are several groups who this book is for, but in this section, two broad audiences are discussed.

Those Wanting to Become Intermediate or Advanced Users

Ubuntu Unleashed is intended for intermediate and advanced users, or those who wish 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 will help you overcome your fear of the system by learning 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 levels 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.

A myriad of 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 much of the knowledge needed to get started in this role can be found in this book.

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 above. 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 causes 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 below. 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 may 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.
- Part VI, "Appendices," are references that give you scope to explore in even more depth some of the topics covered in this book, as well as historical context to Ubuntu and installation resources.

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 since most of these details are only a quick Google search away. Instead, we focus on teaching you how to find information you need while giving a quality overview worthy of the intermediate or advanced user. Sometimes this book offers tables of various options, commands, and keystrokes to help condense, organize, and present information about a variety of subjects.

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

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

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

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

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

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

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

NOTE

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

TIP

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

CAUTION

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

SIDEBARS CAN BE GOLDMINES

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

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

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

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

CHAPTER 1

Installing Ubuntu

IN THIS CHAPTER

- Before You Begin the Installation
- Step-by-Step Installation
- References

Not that long ago, the mere mention of installing Linux struck fear into the hearts of mortal men. Thanks to a campaign of fear, uncertainty, and doubt (commonly referred to as FUD), Linux garnered a reputation as something of an elitist operating system that could be installed, configured, and used only by extreme computer geeks and professionals. Nowadays, it is a different story entirely, and Ubuntu is one of the easiest Linux distributions (distros) to install. In this chapter, we cover how to get started with the install disc, including booting from an Ubuntu Live CD, DVD, or USB drive to test your system. Then we cover the actual installation of Ubuntu, looking at the various options available. The whole process is pain free with Ubuntu, as you are about to learn.

Before You Begin the Installation

Installing a new operating system is a major event, and you should make sure that you have properly thought through what is going to take place. The first thing to consider is how the hardware will be affected by the software that you propose to install. Although Ubuntu runs well on a wide variety of hardware, it is worthwhile to check your hardware components because some bits of hardware do not work well with Ubuntu. This section provides some areas for you to investigate and think about, and may even save you hours of frustration when something goes wrong. Problems are becoming much less frequent, but they still crop up occasionally.

You start by researching and documenting your hardware. This information will prove helpful later on during the installation.

Researching Your Hardware Specifications

At the absolute minimum, you should know the basics of your system, such as how much RAM you have installed and what type of mouse, keyboard, and monitor you have. Knowing the storage capacity and type of hard drive you have is important because it helps you plan how you will divide it up for Ubuntu and troubleshoot if problems occur. A small detail, such as whether your mouse uses the USB or PS/2 interface, will ensure proper pointer configuration—something that should happen without any problem, but you will be glad you knew in case something does go wrong. The more information you have, the better prepared you will be for any problems.

You can make an inventory or at least a quick list of some basic features of your system. Again, the items you most want to know include the amount of installed memory, the size of your hard drive, the type of mouse, the capabilities of the display monitor (such as maximum resolution), and the number of installed network interfaces (if any).

DVD Installation Jump Start

To install Ubuntu Desktop from the disc included with this book, you must have at least a 1GHz Pentium 4 CPU, 5GB of hard drive space, and 512MB RAM. A monitor with a display resolution of at least 1024 x 768 is strongly recommended. Internet access is not required, but is very helpful and also recommended.

Installation Options

Ubuntu is made available in three main forms: the Desktop CD, the Server install CD, and the Alternate install CD. (We are not including derivative distributions like Kubuntu or Lubuntu in this list.) For most people, the Desktop CD is what you want; the alternate is mainly used for upgrading existing Ubuntu users to the latest version, as well as allowing installation on either low-powered systems or in places where fine-tuned, custom requirements exist. The Server install CD can get a LAMP server up and running in about 20 minutes (Linux, Apache, MySQL, and PHP), but as you learn in this book, all these components are available to the Ubuntu default distribution. You can find a list of the currently available ISO images in a couple of places. An ISO image contains the entire contents of a CD or DVD in a single file that can be used as if it were a CD or DVD and which can be burned to a physical CD or DVD if desired. The easy place that will be best for most people is www.ubuntu.com/download, which includes a nice graphical menu system and links to easy-to-read information and detailed instructions. Those with more specific requirements, like a desire to use one of the official alternate Ubuntu versions like Kubuntu or Lubuntu, can find what they need by navigating the menus at cdimage.ubuntu.com.

Official Ubuntu Flavors

Ubuntu has 12 official variants, called *flavors*, as follows:

- Ubuntu
- Ubuntu Server
- Ubuntu Cloud
- Ubuntu Netboot
- Ubuntu Core
- Kubuntu Desktop
- Kubuntu Mobile
- Xubuntu Desktop
- EdUbuntu
- Mythbuntu
- Ubuntu Studio
- Lubuntu

Almost everything in this book applies to any of these flavors. The exceptions include GUI-specific content, such as Unity-specific descriptions, and content referring to programs not installed by default, such as many of the server and programming options.

To install using the DVD included with this book, ignore the next couple of paragraphs and then keep reading this chapter.

To install using a CD or DVD that you create, download the ISO image you need from either source above. You need to download a program to enable you to burn this image to physical media. For Windows, try either InfraRecorder (http://infrarecorder.org) or ISO Recorder (http://isorecorder.alexfeinman.com/isorecorder.htm) and follow the instructions given by the authors. For Mac OS X, you can use Apple's Disk Utility, which is installed by default. For Ubuntu, right-click on the icon for an ISO image and select Write to Disc. Once the CD or DVD is created, use it as you follow the installation instructions in this section.

You can also install using a USB thumb drive (use one that holds at least 2GB). Download the ISO image you need from either source above. You need to download a program that enables you to use this image to create a bootable USB drive. For Windows, try Universal USB Installer (www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/) or ISO Recorder (http://isorecorder.alexfeinman.com/isorecorder.htm) and follow the instructions given by the authors. For Ubuntu, use the installed Startup Disk Creator program available from the System, Administration menu (for Ubuntu versions 10.10 or older) or by searching in the Dash (Ubuntu versions 11.04 or newer). It is not recommended to use a bootable USB drive to install on Apple Mac hardware. Once the ISO is written to the USB drive, use it as you follow the installation instructions in this section.

Planning Partition Strategies

Partitioning is a topic that can make novice Linux users nervous. Coming from a Microsoft world, where you might be used to having just one hard drive, it can seem a bit strange to use an operating system that makes partitioning a hard drive possible or even preferable and common.

Depending on your needs, you can opt to have a single large partition to contain everything, which is the official recommendation of the Ubuntu community and developers. You may prefer to segment your installation across several partitions if you have advanced knowledge and specific needs.

If you are installing Ubuntu in a corporate or business environment, the needs of the business should be a primary concern. Be careful to ensure that you build in an adequate upgrade path that allows you to extend the life of the system and add any additional storage or memory.

Knowing how software is placed on your hard drive for Linux involves knowing how Ubuntu organizes its file system. This knowledge will help you make the most out of hard drive space. In some instances, such as when you're planning to have user directories mounted via NFS or other means, this information can help head off data loss, increase security, and accommodate future needs. Create a great system, and you'll be the hero of information services. The Linux file system is covered along with commands to manipulate files and directories in Chapter 10, "Command-Line Quickstart."

To plan the best partitioning scheme, research and know the answers to these questions:

- How much disk space does your system require?
- Do you expect your disk space needs to grow significantly in the future?
- Will the system boot only Ubuntu, or do you need a dual-boot system?
- How much data requires backup, and what backup system will work best? (See Chapter 16, "Backing Up," for more information about backing up your system.)

The Boot Loader

During installation, Ubuntu automatically installs *GRUB2 (Grand Unified Boot Loader)* to the *Master Boot Record (MBR)* of your hard drive. Handily enough, it also detects any other operating systems, such as Windows, and adds entries in GRUB2 as appropriate. If you have a specific requirement not to install GRUB2 to the MBR, you need to install using the Alternate disc, which will allow you to specify the install location for GRUB2.

NOTE

Dual Boot Not Recommended, But You Can Try It If You Want

If you are attempting to create a dual-boot system using both Windows and Ubuntu, where multiple operating systems exist on the hard drive and the user selects which one to use at boot time, you should install Windows first because it will overwrite the

MBR and ignore any other operating systems on the disk. Ubuntu also overwrites the MBR, but does so in a way that creates a boot menu that includes all operating systems it detects on the disk. Dual booting works, but in the past few years, some options have arisen that are better for most people.

If you decide you must dual boot, make sure you have your Windows recovery media available and that you either already have enough free space on your hard drive or know how to shrink the existing Windows partition and create a new partition on the hard drive for Ubuntu. No support or instructions for doing this are given in this book.

You should consider using Wubi instead of dual booting. See the "Wubi: The Easy Installer for Windows" section later in this chapter. This book recommends this option or the next one.

You can also use virtualization. See Chapter 32, "Virtualization on Ubuntu," for more information.

Installing from CD or DVD or USB Drive

The BIOS of most PCs support booting directly from a CD, DVD, or USB drive and enable you to set a specific order of devices (such as floppy, hard drive, CD-ROM, or USB) to search for bootable software. Turn on your PC and set its BIOS if required (usually accessed by pressing a Function or the Del key after powering on); then insert your Ubuntu install media and boot to install Ubuntu.

Step-by-Step Installation

This section provides a basic step-by-step installation of Ubuntu from the install DVD included with this book, but these instructions also work with media you create yourself using an ISO image you downloaded and wrote to a disk or USB drive using the instructions provided earlier; just replace mentions of DVD with your install medium. The install process itself is fairly straightforward, and you should not encounter any real problems.

WARNING

Back Up First

If you have anything at all on your computer that you want to save, back it up first. Installing an operating system has become easier to do, but it is still a major change. Your entire hard drive will be erased and new information written to it. This is expected when installing Ubuntu to the entire hard drive, but can even happen due to user error or gremlins (unexplained problems) when attempting a dual boot installation. Back up anything you want to preserve. Save data and files to an external hard drive or other medium. You can even back up your entire operating system and current installation using something like Clonezilla (http://clonezilla.org). Whatever you do, go in with the perspective that everything currently on the computer will disappear. If this is okay, continue.

It is useful and recommended to have your computer connected to the Internet as you proceed so that you can download updates while installing.

Installing

To get started, insert the DVD into your drive and reboot your computer. The initial screen offers a variety of languages for you to use during installation (see Figure 1.1) and two options. Try Ubuntu boots and runs Ubuntu from the DVD without making any changes to your system so that when you remove the DVD and reboot, everything will be as it was before. Install Ubuntu installs Ubuntu instead of your current operating system, or alongside it (for dual booting). Select Install Ubuntu to begin.

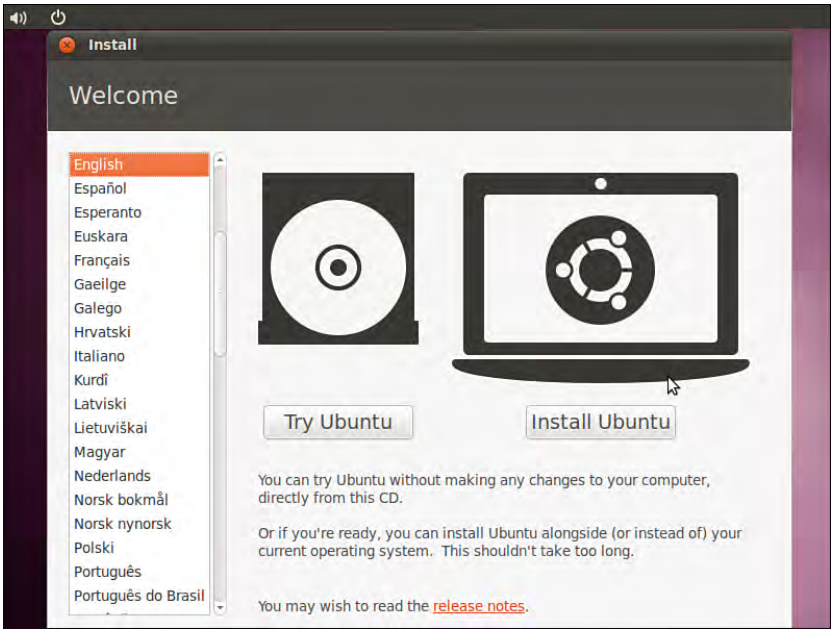

FIGURE 1.1 Choose a language for the installation in this opening screen.

Figure 1.2 shows the Preparing to Install Ubuntu screen. If you select the two check boxes at the bottom, Ubuntu downloads extra software, such as multimedia codecs and any software updates that have been released since the disk was created, and includes them in the installation. Doing so is recommended.

If other operating systems are found on your system, you are given the option to install Ubuntu alongside them or to erase them and use the whole disk for Ubuntu. See the dual-boot note and backup warning earlier in this chapter before continuing.

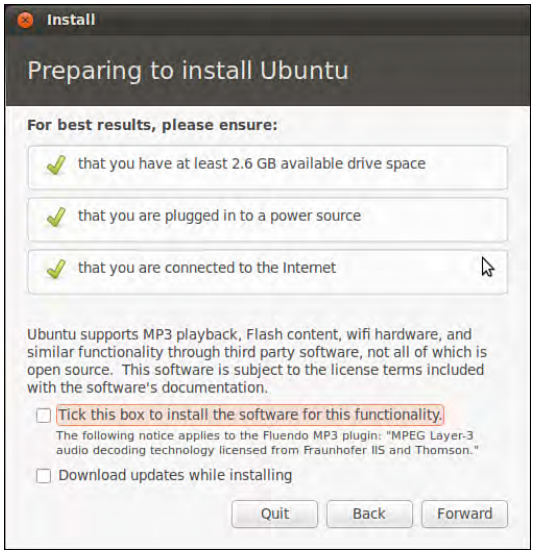

FIGURE 1.2 Before proceeding, double-check that your system meets the requirements shown.

Next, you have the option to either erase and use the entire hard disk for your installation (recommended for people who are new to Linux and are nervous about partitioning a hard drive) or specifying partitions manually, which is not as difficult as it sounds, as shown in Figure 1.3.

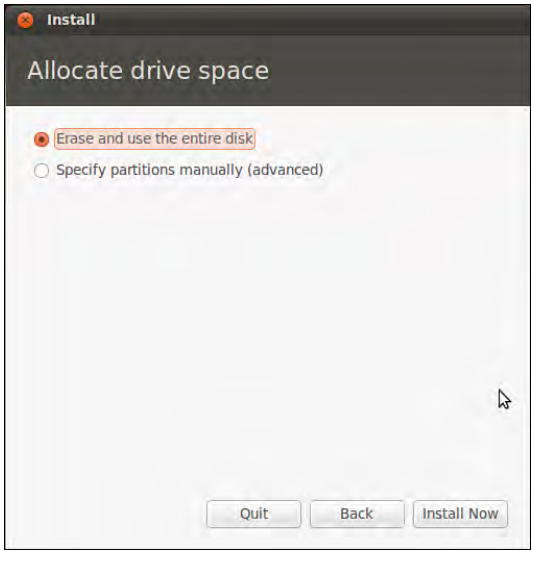

FIGURE 1.3 What do you want to do with your hard drive?

Because this is a book for people who want a deeper understanding of their system, we show a simple partitioning scheme here and give some reasons for our choices.

Your needs may vary, so we give some principles to consider so that you may adjust our recommendations to your situation.

Many users will benefit from three simple partitions, shown in Figure 1.4. Here we have a swap partition, a root partition (denoted by /), and a home partition where data for each user on the system will be stored. The swap partition was previously used for those times when there was not enough memory for running processes, so the Linux kernel could move stored information from RAM to the hard drive and use that swap space as extra RAM. This is slow, but sometimes it is quite useful. These days, desktop systems come with more memory, and the swap space is used mainly for storing information during suspend and hibernate, which can be used instead of turning off your computer. Having /home on a separate partition can be a good idea because it enables you to install a different Linux distribution or a different release of Ubuntu on your / partition while retaining your user data (neither of which are covered in this book).

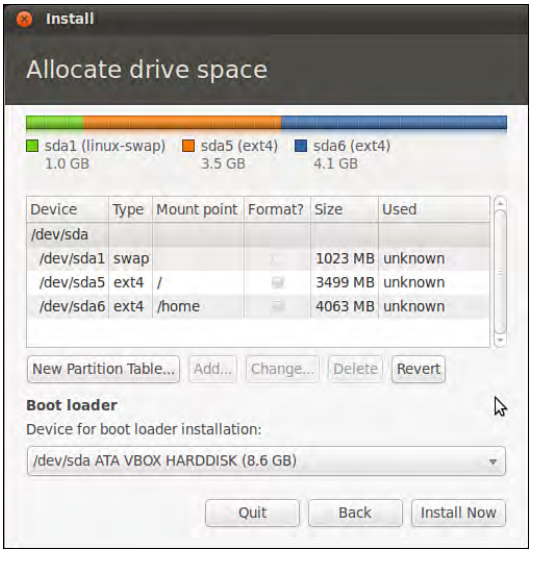

FIGURE 1.4 A simple, but useful, partitioning scheme.

NOTE

Suspend saves your current system state to RAM and then turns off power to most of the system, but maintains just enough power use to keep that state saved in RAM. When you restart your system, it comes back quickly and to the same state it was in. Hibernate saves your current system state to the hard disk. These days, it is just as fast to boot from scratch as it is to hibernate and restore, so there isn't much point in using it.

If you choose to manually partition instead of using the entire disk, we recommend a / partition of at least 3GB, a swap partition that is about 1.5 to 2 times the size of your installed RAM, and then using the rest of the disk for /home. More complex systems, such as servers, may find more complex partitioning schemes useful, especially when multiple hard drives are available. Read Chapter 10 for a quick discussion of the parts of the Linux file system to get an idea of parts of the file system that could benefit from residing on their own (or even multiple!) partitions or disks.

After you have made your partitioning selections, installation continues by asking about your current location; it helps you select the most appropriate keyboard layout for your system based on that location and the most common language used there, and asks you to enter your name, a username that you will use to login to the system, and a password. You can even choose to encrypt your /home folder during the process.

A Quick Aside on Passwords

When you create your password, be sure to remember what you entered. If you forget it, you cannot use your new system because you will not be able to log on to it.

When setting a password, the common advice is to make sure that it has a mixture of letters and numbers to make it more secure. For instance, a good example of a historically recommended style of password is **`T1a5c0p`**. Although this might seem like garbage at first glance, the easy way to remember it is by remembering the phrase This Is A Good Choice Of Password, shortened to Tiagcop, and finally substituting some of the letters with similar-looking numbers.

There are some reasons why this might not be the best recommendation anymore, because computer systems are much faster than they used to be. It is a true statement that the longer a password is, the harder it is to break. For this reason, the newest recommendation is to use a passphrase consisting of at least four words, perhaps something like *green monkeys chortle often.* There is no doubt that a password the length of four common words combined together would be harder to break than the **`T1a5c0p`** example. From that perspective, it seems like a no-brainer. On the other hand, a longer password that does not use any words found in a dictionary would be even better, but the problem here is that these passwords, and even the **`T1a5c0p`** example, can be hard to remember and may end up being written down on a sticky note next to the computer, perhaps even stuck to the monitor. That is worse, especially if you use good security and create a different password for every website and computer system that requires one.

One solution is to choose a really good password for your system, one that you will remember, like the four-words example or a long passphrase like PeanutButterandJelly$andwiches, and then create one more good password to use with a password manager program like KeyPassX (available in the Ubuntu software repositories; see Chapter 9, “Managing Software,” for details on how to find and install it), which can generate long, completely random passwords for you and keep them in a list that can only be viewed by accessing your system and then accessing the program, both of which will use good passwords. Let’s get back to the installation.

While you are answering the questions asked by the installer, the Ubuntu install has already begun to copy files to your hard drive. Performing these tasks in parallel makes the process even faster than it used to be. Now that all necessary information has been input, you are shown a series of information screens while the installation is completed. These are filled with interesting content about Ubuntu and are worth reading while you wait.

When the process is complete, you are prompted to restart the computer. Do so, and remove the install media when it is ejected. Then log in when the reboot is complete. That's it.

In previous editions of this book, this chapter was longer and described a process that was sometimes confusing and fraught with peril. The Ubuntu developers deserve high praise for making this so incredibly easy and fast. The Linux kernel supports more hardware than ever, and the Ubuntu kernel gurus (who make the decisions about what hardware modules to enable, among other things) do a great job so that most hardware works out of the box.

If you do run into trouble, go to ubuntuforums.org as a great first resource, give a list of your hardware (you did make that list we suggested, right?) and any error messages you have received, and you should either find someone who can help you or get some direction to another source of assistance.

First Update

It used to be that the first thing that you needed to do with your new system was update it to the latest package versions. You do this mainly to ensure that you have the latest security updates available. Remember that first installation step where we recommended checking the box to have software updates downloaded during the installation process? That means these updates were acquired during the installation and that your system should be up-to-the-minute current.

If you want to double-check that you have all the current versions of software and security updates installed, see Chapter 9 for assistance.

Wubi: The Easy Installer for Windows

Also included on the Ubuntu CD is Wubi, which allows you to install an Ubuntu system within Windows. In effect, Wubi creates a large file to hold all the Ubuntu information and provides a way, using the Windows bootloader, to boot into Ubuntu without having to do any partitioning at all. What's more, it's almost as fast as the real thing and much faster than a Live CD. To get started with Wubi, you need to insert the Ubuntu disc into your CD drive while Windows is booted. You'll see a screen similar to Figure 1.5.

Click the option to Install inside Windows to be taken to the next screen; there you can choose some basic configuration options using the drop-down menus, as shown in Figure 1.6.

Wubi creates an image file on your hard drive and modifies the Windows bootloader to allow you to boot into Ubuntu or Windows. Don't worry, though; your hard disk hasn't actually been modified, but a rather large file has been created to hold Ubuntu. When you restart your system, you'll see the Ubuntu option, so press the down-arrow key and press Enter to begin booting into your new Ubuntu system.

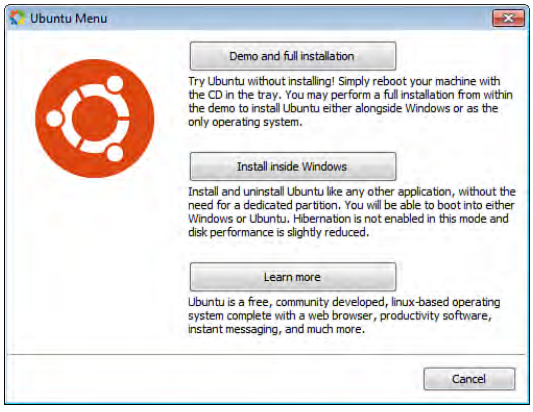

FIGURE 1.5 The auto-start menu gives you a gateway to Ubuntu. Click Install inside Windows to start the Wubi installation.

FIGURE 1.6 You can select some configuration options before you start the installation.

The first time you start up a Wubi Ubuntu installation, it does some post-installation configuration before loading the Unity desktop. You now have a fully functional Ubuntu system.

If you decide that you want to do a regular installation, you'll need to remove Ubuntu and Wubi. Fortunately, as part of the Wubi installation, an entry is added to the Add/Remove Programs tool in Windows, allowing you to quickly and easily remove Wubi and the associated Ubuntu installation. Just remember to first back up any files that you want to keep.

Shutting Down

At some point, you are going to want to shut down your computer. As with most things in Linux, there are different ways to do it. You can use the power icon located in the upper-right corner of your screen and its drop-down menu, as shown in Figure 1.7. From

there, you can also choose to hibernate (saves a copy of the running state to disk and shuts down) or suspend (saves the current running state to memory).

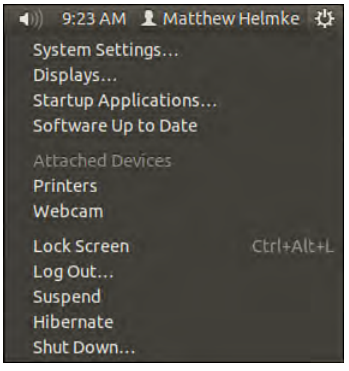

FIGURE 1.7 You have several options in the power icon menu.

If you are working at the command line, you can immediately shut down your system by using the **shutdown** command like this:

```
$ sudo shutdown -h now
```

You can also use the shutdown command to restart your computer, as follows:

```
$ sudo shutdown -r now
```

For new users, installing Ubuntu is the beginning of a new and highly rewarding journey on the path to learning Linux. For Ubuntu system administrators, the task ahead is to fine-tune the installation and to customize the server or user environment.

References

- **www.ubuntu.com**—The place to start when looking for news, information, and documentation about installing, configuring, and using Ubuntu.
- **www.ubuntu.com/desktop/get-ubuntu/download**—This page has specific information and links for downloading, burning, and installing the current release of the Ubuntu Desktop edition. There are comparable links available from the main page for Ubuntu Netbook and Server editions and other projects.
- **www.nwc.com/columnists/1101colron.html**—How to use Intel's Pre-execution Environment (PXE) protocol to remote boot workstations.
- **www.gnu.org/software/grub/**—Home page for the GRUB boot loader.

CHAPTER 2

Post-Installation Configuration

IN THIS CHAPTER

- Troubleshooting Post-Installation Configuration Problems
- The **sudo** Command
- Finding Programs and Files
- Software Update
- Configuring Software Repositories
- System Settings
- Configuring Wireless Networks
- References

Now that the primary task of installing Ubuntu has been completed, you can begin to customize your new operating system.

In this chapter, we look at getting up and running with Ubuntu. A tour around the desktop is the goal of Chapter 3, "Working with Unity." These two chapters should be read together. In this chapter, you learn how to do some basic administration tasks. By the end of this chapter, you should feel comfortable enough to move on through the rest of the book.

Troubleshooting Post-Installation Configuration Problems

A lot of work has gone into Ubuntu to make it as versatile as possible, but sometimes you may come across a piece of hardware that Ubuntu is not sure about. Knowing what to do in these situations is important, especially when you are working with Ubuntu for the first time.

Because Ubuntu (and Linux in general) is built on a resilient UNIX foundation, it is much more stable than some operating systems. However, even though things might seem to be working fine, Ubuntu could have a problem that might not affect the appearance of the system. In this section, you learn how to examine some of Ubuntu's built-in error logs to help you discover or diagnose unseen problems.

Ubuntu has a command that responds with detailed messages that are output directly by the operating system:

the **dmesg** command. This command is commonly used with the **grep** command to filter output. The **dmesg** command takes its output directly from the **/var/log/messages** file, so you can choose to either run **dmesg** or to read the file directly by typing **less /var/log/messages**. The output is fairly detailed, so be prepared for an initial shock when you see how much information is generated. You might find it easier to generate a file with the **dmesg** output by using the following command:

```
matthew@seymour:~$ dmesg > dmesg.txt
```

This takes the output from the **dmesg** command and stores it in a new text file called **dmesg.txt.** You can then browse it at your leisure using your choice of text editor, such as **vi** or **emacs**. You can even use the **less** command, like so:

```
matthew@seymour:~$ less dmesg.txt
```

The messages are generated by the kernel, by other software run by **/etc/init.d**, and by Ubuntu's runlevel scripts. You might find what appear to be errors at first glance, but some errors are not really problems (for example, if a piece of hardware is configured but not present on your system).

Thanks to Google, troubleshooting is no longer the slow process it used to be. You can copy and paste error messages into Google's search bar to bring up a whole selection of results similar to the problem you face. Remember, Google is your friend, especially www.google.com/linux, which provides a specialized search engine for Linux. You can also try http://marc.info, which browses newsgroup and mailing list archives. Either way, you are likely to come across people who have had the same problem as you.

It is important to work on finding and testing only one solution to one problem at a time; otherwise, you might end up getting no work done whatsoever. You should also get into the habit of making backup copies of all files that you modify, just in case you make a bad situation worse. Use the copy (**cp**) command like this:

```
matthew@seymour:~$ cp file file.backup20101010
```

You should not use a **.bak** extension on your backup file because this could get overwritten by another automatic process and leave you frustrated when you try to restore the original file. I like to use backupYYYYMMDD, as in the preceding code where I used the release date for Ubuntu 10.10: 2010(year)10(month)10(day).

If something breaks as a result of your changing the original file, you can always copy the original back into place using the command like this:

```
matthew@seymour:~$ cp file.backup20101010 file
```

(Something as simple as this can really save you, especially when you are under pressure because you've changed something you shouldn't have changed on a production system. It is better not to make sweeping changes on a production system.)

The sudo Command

You will find as you work through this book that Ubuntu relies on the **sudo** command while working at the command line. This command is used in front of other commands to tell Ubuntu that you want to run the specified command with super user powers. This sounds really special, and it actually is. When you work using the **sudo** command, you can make wide-ranging changes to your system that impact the way it runs. Be extra careful when running any command prefixed with **sudo**, however; a wrong option or incorrect command can have devastating consequences.

The use of **sudo** is straightforward. All you have to do is enter it like this:

```
matthew@seymour:~$ sudo command commandoptions
```

Just replace the word *command* with the command that you want to run, along with any options. For example, the following command opens your **xorg.conf** file in **vi** and enables you to make any changes as the super user before being able to save it:

```
matthew@seymour:~$ sudo vi /etc/X11/xorg.conf
```

Whenever you execute a command using **sudo**, you are prompted for your password. This is the same password that you use to log in to Ubuntu, so it is important that you remember it.

Sometimes, however, you may want to work with a classic root prompt instead of having to type **sudo** in front of every command (perhaps if you have to work with lots of commands at the command line that require super user access, for example). **sudo** enables you to do this by using the **sudo -i** command. Again, you are prompted for your password, which you should enter, after which Ubuntu gives you the standard root prompt, as follows:

```
matthew@seymour:~#
```

From here, you can execute any command without having to keep entering **sudo**.

Finding Programs and Files

In the past, Ubuntu used a system of menus to guide users searching for programs. Use the Dash to find these programs. Access the Dash by clicking the Ubuntu logo at the upper left of the screen. This brings up the menu shown in Figure 2.1. Start typing in the Search box to find specific programs or documents on your system. The icons in the row along the bottom of the Dash provide categories of things that will be listed or to narrow searches. The house icon brings you to the main Dash screen, as shown here. The second icon shows installed applications and lists others available for installation. Click the arrow next to See more results to view all installed software. The third icon shows documents. The fourth icon shows media. The Dash is covered in more detail in Chapter 3.

FIGURE 2.1 The Dash is the place to find programs and files.

Software Update

This topic is covered in greater detail in Chapter 9, “Managing Software,” but it is worthy of a quick mention now so that from the start you can benefit from any available security and bug fixes. The easiest way to check for updates is Update Manager, which is really a GUI wrapper for the command **apt-get**. Open Update Manager from the Dash by typing **update manager** to search for it. When the window opens, click Check and enter your password. Update Manager then checks the Ubuntu software repositories to determine whether any updates are available. When it detects that new versions of installed software are available, they appear in the window. Each one has a check box that you can uncheck if you don’t want to install a particular software update. Then click Install Updates to complete the process (see Figure 2.2). Software repositories are discussed later in this chapter.

Another way of updating your system is to use the command line. This is vital on servers that do not have a GUI installed, and it is sometimes quicker than using Update Manager on a desktop computer. I like to use the command line to manage all the computers on my home network because I can use Secure Shell (SSH) to connect to each from a terminal and perform updates from another room in the house, and anyone using that computer is left undisturbed. You learn how to connect like this later in the book; I just want to whet your appetite for now.

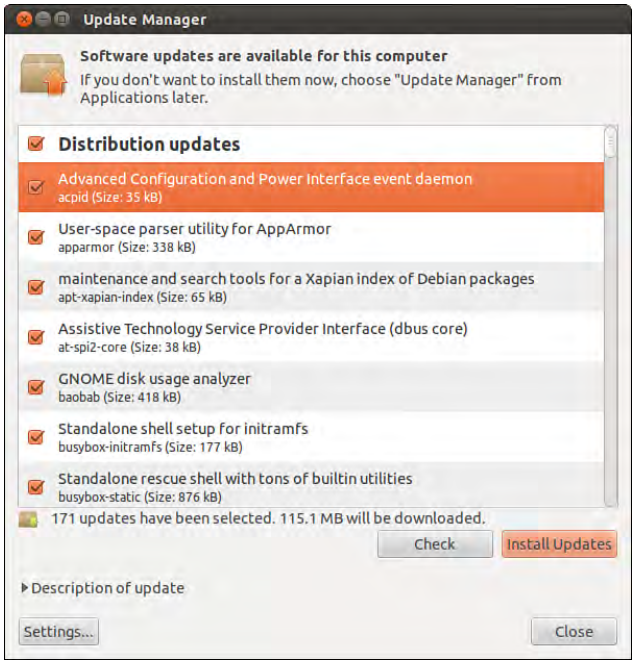

FIGURE 2.2 Update Manager showing available software updates.

To use the command line, go to the Applications, Accessories menu and select Terminal. This is the command line (commonly referred to as the *terminal*) and is one of the most powerful features of Linux. It is covered in more detail in Chapter 10, "Command-Line Quickstart," and Chapter 11, "Command-Line Master Class," so we do not delve too deeply here.

You are greeted with a prompt similar to the one here:

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

A blinking cursor also displays. Ubuntu is awaiting your first command. Here we want to issue the following command:

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

This command tells the package management utility **apt-get** to check the Ubuntu repositories and look for any updates to your installed software. In a matter of seconds, Ubuntu completes all of this, and your screen should look something like this:

```
matthew@seymour:~$ sudo apt-get update
[sudo] password for matt:
Hit http://security.ubuntu.com oneiriconeiric-security Release.gpg
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/main Translation-en
Hit http://us.archive.ubuntu.com oneiriconeiric Release.gpg
```

```
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/main Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/main Translation-en_US
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/main Translation-en_US
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/multiverse
➥Translation-en
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/multiverse
➥Translation-en_US
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/restricted
➥Translation-en
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/restricted
➥Translation-en_US
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/universe
➥Translation-en
Ign http://security.ubuntu.com/ubuntu/ oneiriconeiric-security/universe
➥Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/multiverse Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/restricted Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/restricted Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/universe Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric/universe Translation-en_US
Hit http://us.archive.ubuntu.com oneiriconeiric-updates Release.gpg
Hit http://security.ubuntu.com oneiriconeiric-security Release
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric-updates/main Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric-updates/main
➥Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric-updates/multiverse
➥Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric-updates/multiverse
➥Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ oneiriconeiric-updates/restricted
➥Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiric-updates/restricted Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ oneiric-updates/universe Translation-en
Ign http://us.archive.ubuntu.com/ubuntu/ oneiric-updates/universe Translation-en_US
Hit http://security.ubuntu.com oneiric-security/main Sources
Hit http://us.archive.ubuntu.com oneiric Release
Hit http://us.archive.ubuntu.com oneiric-updates Release
Hit http://security.ubuntu.com oneiric-security/restricted Sources
Hit http://security.ubuntu.com oneiric-security/universe Sources
Hit http://security.ubuntu.com oneiric-security/multiverse Sources
Hit http://security.ubuntu.com oneiric-security/main i386 Packages
Hit http://security.ubuntu.com oneiric-security/restricted i386 Packages
Hit http://us.archive.ubuntu.com oneiric/main Sources
Hit http://security.ubuntu.com oneiric-security/universe i386 Packages
Hit http://security.ubuntu.com oneiric-security/multiverse i386 Packages
```

```
Hit http://us.archive.ubuntu.com oneiric/restricted Sources
Hit http://us.archive.ubuntu.com oneiric/universe Sources
Hit http://us.archive.ubuntu.com oneiric/multiverse Sources
Hit http://us.archive.ubuntu.com oneiric/main i386 Packages
Hit http://us.archive.ubuntu.com oneiric/restricted i386 Packages
Hit http://us.archive.ubuntu.com oneiric/universe i386 Packages
Hit http://us.archive.ubuntu.com oneiric/multiverse i386 Packages
Hit http://us.archive.ubuntu.com oneiric-updates/main Sources
Hit http://us.archive.ubuntu.com oneiric-updates/restricted Sources
Hit http://us.archive.ubuntu.com oneiric-updates/universe Sources
Hit http://us.archive.ubuntu.com oneiric-updates/multiverse Sources
Hit http://us.archive.ubuntu.com oneiric-updates/main i386 Packages
Hit http://us.archive.ubuntu.com oneiric-updates/restricted i386 Packages
Hit http://us.archive.ubuntu.com oneiric-updates/universe i386 Packages
Hit http://us.archive.ubuntu.com oneiric-updates/multiverse i386 Packages
Reading package lists... Done
matthew@seymour:~$
```

Upgrade your software by entering the following:

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

Because you have already checked for updates, Ubuntu automatically knows to download and install only the packages it needs. The **dist-upgrade** option works intelligently to ensure that any dependencies that are needed can be satisfied and will be installed, even if major changes are needed. You can also use the option **upgrade**. It isn't as smart as **dist-upgrade**, but may be a better choice on a production server because **upgrade** will not make major changes to software installations, but only those necessary for security and simple package updates. This allows the systems administrator more flexibility to keep up-to-date with security but keep running setups otherwise unchanged.

Command Line apt-get Alternatives

Ubuntu has an alternative to **apt-get** called **aptitude**. It works in pretty much the same way as **apt-get** with the exception that **aptitude** also includes any recommended packages on top of the requested packages when installing or upgrading. You use **aptitude** mostly the same as you use **apt-get**; for instance, you could issue the command **aptitude update** and it would update the repository information. To upgrade, the options are **aptitude safe-upgrade** and **aptitude full-upgrade**. You can also issue the command **aptitude** without any options, which brings up a text-based package manager that you can use to select packages.

Configuring Software Repositories

Ubuntu uses software repositories to get information about available software that can be installed on your system. Ubuntu is based on a much older Linux distribution called

Debian. Debian has access to tens of thousands of different packages, which means that Ubuntu has access to these packages, too. The Debian packages are made available in Ubuntu's Universe repository. A set of volunteers called *Masters of the Universe (MOTUs)* are well trained and follow strict guidelines to package software and make even more packages available to Ubuntu users in Universe. The Universe repository is filled with optional and often useful or fun software and is enabled by default, along with other official repositories containing software necessary for Ubuntu to be installed and run in all its various official forms, security updates, and software updates.

You can adjust which repositories are enabled using the Software Sources GUI tool, available in Update Manager by clicking Settings. On the first tab (Ubuntu Software), you have five options to choose from. The default settings are shown in Figure 2.3. It is entirely up to you which options you check, but make sure that as a minimum, the first check box is checked to allow you access to Canonical-supported Open Source software, which includes all the packages necessary for a basic Ubuntu installation and a few more that are commonly used. The more boxes you check, the wider your selection of software. It's also a good idea to make sure that the Proprietary Drivers for Devices box is checked so that you can benefit from drivers that may enhance your system's performance.

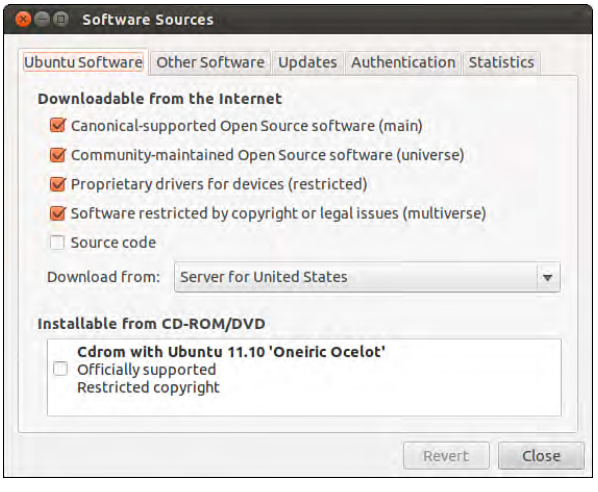

FIGURE 2.3 You can find or add other options under the Updates and Other Software tabs.

Open Source Versus Proprietary

You may hear some arguments about using proprietary drivers or other software in Ubuntu. Some people feel that it goes against what *open source* stands for because the program code that is used for the drivers or software cannot be viewed and modified by the wider community but only by the original developers or company that owns it. There is also a strong argument that says users should have to undergo the least amount of work for a fully functional system.

Ubuntu takes a middle-of-the-road stance on this and leaves it up to the user to decide. Open-source software is installed by default, but options are given to allow proprietary software to be installed easily.

After you are happy with your selections, switch to the Updates tab to configure Ubuntu's behavior when updates are available (see Figure 2.4). By default, both the important security updates and recommended updates are checked to ensure that you have the latest bug fixes and patches. You can also choose to receive proposed updates and *backports* (software that is released for a newer version of Ubuntu but reprogrammed to be compatible with the current release), but we recommend this only if you are happy to carry out testing for the community, because any updated software from these repositories can adversely affect your system.

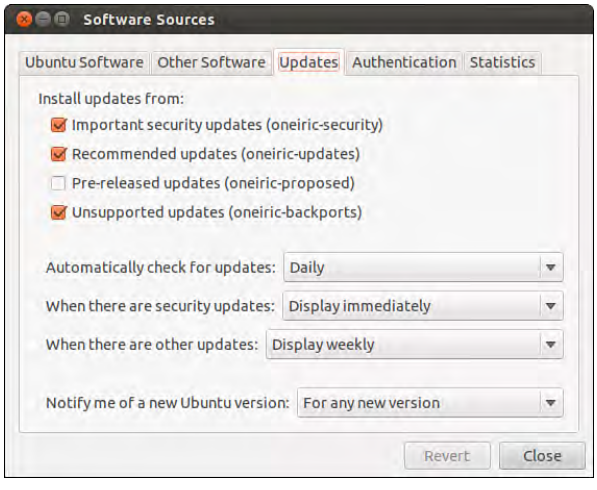

FIGURE 2.4 In the Updates tab of Software Sources, configure which updates you want and how you want them to be handled.

Ubuntu also allows you to configure how often it checks for updates and how they are installed. By default, Ubuntu checks daily for updates and, if any are available, notifies you. However, you can change the frequency and the actions Ubuntu carries out when it finds available updates. We recommend keeping the notification-only option because this allows you to see what updates are available prior to them being installed. If you want to save time, choose Download All Updates in the Background to configure Ubuntu to silently download the updates before giving you the option to install them.

CAUTION

We don't recommend selecting the option that automatically installs security updates. It's important that you have the option of choosing to install updates because there is always the chance that an update may cause problems. The last thing you want is for your system to suddenly stop working because an update was installed that has broken something without your knowledge.

Part of the magic of Ubuntu is the ease in which you can upgrade from major version to major version, such as moving from 11.04 to 11.10. Some Ubuntu releases are called LTS, for *Long Term Support*, and are intended for those who do not want to upgrade their operating system as often. It is possible to do this. By ensuring that the release upgrade option is set to Long Term Supported releases only, you'll be prompted to upgrade your version of Ubuntu only to what will be the next LTS version, which is scheduled to be 12.04, to be released in April 2012.

The Other Software tab lets you add other repositories. It comes by default with everything you need to connect to and use Canonical's partner repository, with nonfree (usually in the licensing sense, but occasionally for payment) software from companies that have an agreement with Canonical to make it easily available to interested users. This repository is disabled by default and you must enable it by checking a box next to its entry in the Other Software tab if you want to use it.

System Settings

To configure system settings, search the Dash for System Settings. This brings up the menu in Figure 2.5 from which you may make adjustments as desired. A couple of the options are described here.

Installing Additional Drivers

Ubuntu is extremely good at detecting and configuring graphics cards. You can check whether you have a video card or some other piece of hardware that could benefit from using a proprietary driver with the Additional Drivers manager. When you open it, it scans your system, and then you can choose to enable or disable any proprietary drivers found, depending on your personal preference. (See the "Open-Source Versus Proprietary" note earlier in this chapter.)

You can find the Additional Drivers manager in the System Settings menu. The window is shown in Figure 2.6.

If you elect to use a listed proprietary driver, Ubuntu first confirms that you are happy to proceed and then automatically downloads and installs the driver. This may require you to log out and back in again for the driver to take effect.

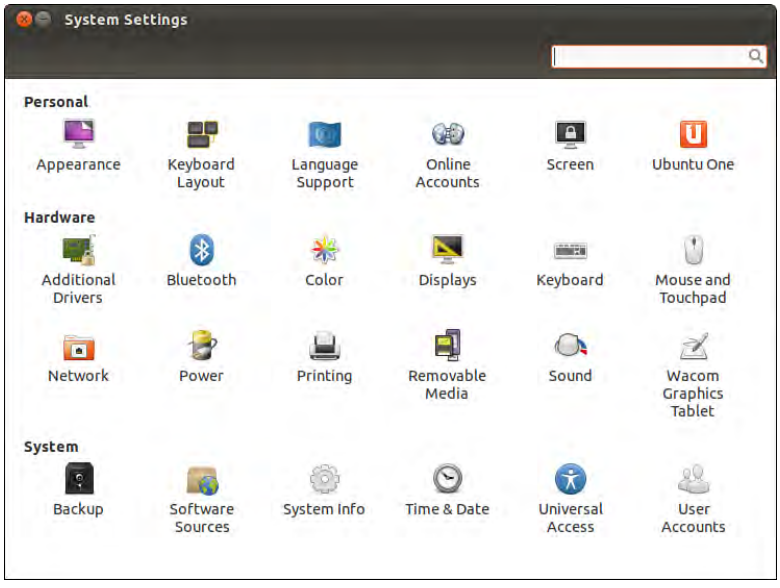

FIGURE 2.5 Adjust your system settings.

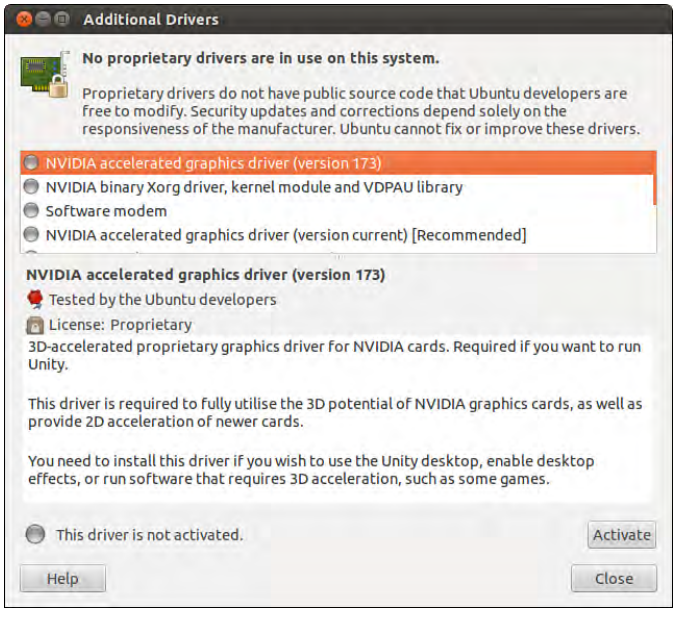

FIGURE 2.6 Toggle the usage of proprietary and other drivers with a simple point-and-click interface.

For the most part, Ubuntu detects and configures the majority of graphics cards from the start, and even if it has problems, it attempts to give you a display of some sort. This feature is known as *BulletProofX*, and it gives you access to tools that may help you fix your problem by trying different combinations of graphics cards and monitors.

Detecting and Configuring a Printer

Setting up a printer in Linux used to be so difficult that previous editions of this book included an entire chapter filled with command-line magic and scary-looking configuration files. This is no longer the case most of the time. Ubuntu includes drivers for many printers, and it is usually easier to install and use a printer in Ubuntu than in other operating systems. This is not an absolute rule, though. Some printer manufacturers do not write and release drivers for Linux, and for some printers, no open source driver exists. It is still a good idea to spend some time on the Internet searching for printers that are well known to work with Linux before buying. One great resource is the Open Printing database from The Linux Foundation, at www.openprinting.org/printers.

If you choose wisely, all you need to do is plug your printer into the computer and turn it on. In many cases, Ubuntu finds the printer and adds the driver automatically. Within a couple of minutes, you should be able to use it. You can find Printing in the System Settings menu and use it to see all installed and configured printers and change their settings. From here, you can choose to enable printer sharing on a network, set options for default print quality, print a test page, and more.

Configuring Power Management in Ubuntu

Using the Power option in the System Settings menu, you can control how Ubuntu will handle power-saving features in specific situations (see Figure 2.7). You can have the display dim the backlight to use less power when on battery, put the display to sleep after a certain amount of inactivity, spin down your hard disks (kind of like turning them off, but not completely; this just makes them stop moving until a request is made from them, saving power in the interim), and set your computer to suspend, hibernate, or turn off when the battery reaches a certain level.

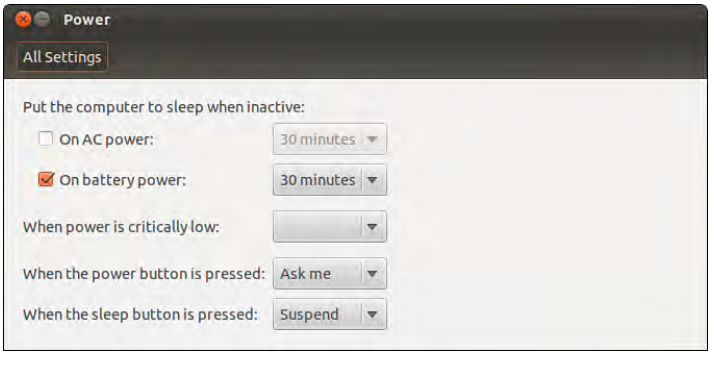

FIGURE 2.7 Configure specific power-related actions with Power Management Preferences.

Ubuntu provides good support for suspend and hibernate. *Suspend* means that your computer writes its current state to memory and goes into a low-power mode, whereas *Hibernate* writes the current state of the system to disk and powers off the computer. Either way, your computer will start much faster the next time you use it.

Setting the Date and Time

Linux provides a system date and time; your computer hardware provides a hardware clock-based time. In many cases, it is possible for the two times to drift apart. Linux system time is based on the number of seconds elapsed since January 1, 1970. Your computer's hardware time depends on the type of clock chips installed on your PC's motherboard, and many motherboard chipsets are notoriously subject to drift.

Keeping accurate time is important on a single workstation, but it is critically important in a network environment. Backups, scheduled downtimes, and other networkwide actions need to be accurately coordinated.

The Ubuntu installer sets the date and time during the installation process when it asks for your location. If you move or just want to change the settings (for example, to have your computer automatically synchronize its clock with official time servers on the Internet), you can do so.

Changing the Time and Date

Ubuntu's graphical tool is the simplest way to set your system date and time and the most obvious for a desktop user. The client is called Time & Date in the System Settings menu.

To make any changes to this tool, you need to unlock it by clicking the Unlock button. Only a user who has administrator privileges may use this tool, and you must enter your password to authenticate. After you've done this, you can change any of the options.

Manually set the date and time in the GUI (see Figure 2.8), or have your computer obtain updated date and time information via the Internet by selecting Automatically from the Internet. The Clock tab includes settings to adjust what is displayed at the upper right of your desktop screen.

Using the `date` Command

Use the **`date`** command to display or set your Linux system time. This command requires you to use a specific sequence of numbers to represent the desired date and time. To see your Linux system's idea of the current date and time, use the **`date`** command like this:

```
matthew@seymour:~$ date
Sat Sep  4 10:20:51 MST 2010
```

To adjust your system's time (for example, to September 4, 2010 at 10:33 a.m.), use a command line with the month, day, hour, minute, and year, like so:

```
matthew@seymour:~$ sudo date 090410332010
Sat Sep  4 10:33:00 MST 2010
```

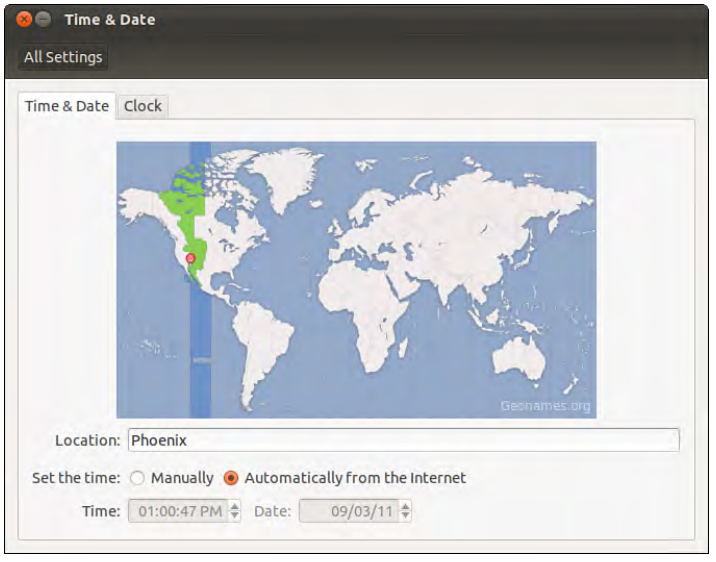

FIGURE 2.8 Set your system date and time with this GUI.

Using the hwclock Command

Use the **hwclock** command to display or set your Linux system time, display or set your PC's hardware clock, or to synchronize the system and hardware times. To see your hardware date and time, use **hwclock** with its **--show** option, like so:

```
matthew@seymour:~$ sudo hwclock --show
Sat 04 Sep 2010 05:25:17 PM MST  -0.806770 seconds
```

Use **hwclock** with its —set and —**date** options to manually set the hardware clock, as follows:

```
matthew@seymour:~$ sudo hwclock --set --date "09/04/10 10:33:00"
matthew@seymour:~$ hwclock --show
Sat 04 Sep 2010 10:33:09 AM MST  -0.904668 seconds
```

In these examples, the hardware clock has been set using **hwclock**, which is then used again to verify the new hardware date and time. You can also use **hwclock** to set the Linux system date and time using your hardware clock's values with the Linux system date and time.

For example, to set the system time from your PC's hardware clock, use the **--hctosys** option, like so:

```
matthew@seymour:~$ sudo hwclock --hctosys
```

To set your hardware clock using the system time, use the **--sytohc** option, like so:

```
matthew@seymour:~$ sudo hwclock --systohc
```

Configuring Wireless Networks

Wireless networking used to be a bear to configure for Linux, requiring a lot of complicated steps to connect to a wireless network. That has become a much easier task, and Ubuntu includes a great utility called Network Manager that makes connecting to and managing wireless networks extremely easy.

When you log in to Ubuntu, the Network Manager applet appears in the top panel (see Figure 2.9). This applet handles and monitors network connections.

FIGURE 2.9 The Network Manager notification applet, shown here already connected to a wireless network.

Click the applet icon in the toolbar to connect to a wireless network. If your wireless access point broadcasts its *service set identifier (SSID)*, it should appear in the list under wireless networks. Click the desired network, and Network Manager detects what encryption (if any) is in use and asks you for the passkey. Enter this and Network Manager starts the wireless connection. The passkey is then stored in the default keyring, a secure area that is unique to your login. From now on, whenever you log in to Ubuntu and are in range of this network, Network Manager will start the connection automatically.

If for some reason your wireless network does not appear (you might have your SSID hidden), you must use the Connect to Other Wireless Network option, in which you enter the network name, wireless security type, and when needed, the password for the connection.

Network Manager can handle WEP and WPA Personal encryption. You are advised to use WPA encryption because it is the stronger of the two. WEP encryption is easily broken.

Network Manager can also connect to Cisco VPN connections, using the **vpnc** software. Install this from the Ubuntu repositories (see Chapter 9), and you will be able to specify connection settings as appropriate, or if you have access to a predefined configuration (PCF file), you can import it directly into Network Manager.

References

In lieu of reference material for this chapter, you are instead directed to read Chapter 3.

CHAPTER 3

Working with Unity

IN THIS CHAPTER

- Foundations and the X Server
- Using Unity, a Primer
- Power Shortcuts
- References

Imagine a world of black screens with white text, or for those of you who remember, dark green screens with light green text. That used to be the primary interface for users to access computers. Computing has moved on significantly and has adopted the *graphical user interface (GUI),* as standard on most desktop and workstation platforms. Not only that, but GUIs have gradually changed and evolved over time. This chapter starts with low-level information about what lies underneath the GUI and builds up to a description of the Unity desktop and how to use it.

Foundations and the X Server

Ubuntu uses the X Window System, the graphical networking interface found on many Linux distributions that provides the foundation for a wide range of graphical tools and window managers. More commonly known as just X, it can also be referred to as X11R7 and X11 (as found on Mac OS X). Coming from the world-renowned Massachusetts Institute of Technology, X has gone through several versions, each of which has extended and enhanced the technology. The open-source implementation is managed by the X.Org foundation, whose board includes several key figures from the open-source world.

The best way to think about how X works is to see it as a client/server system. The X server provides services to programs that have been developed to make the most of the graphical and networking capabilities that are available under the server and in the supported libraries. X.Org provides versions for many platforms, including Linux and Mac OS X. Originally implemented as XFree86, X.Org was

forked when a row broke out over certain restrictions that were going to be included in the XFree86 license. Taking a snapshot of code that was licensed under the previous version of the license, X.Org drove forward with its own implementation based on the code. Almost in unison, most Linux distributions turned away from XFree86 and switched their development and efforts to X.Org.

A desktop environment for X provides one or more window managers and a suite of clients that conform to a standard graphical interface based on a common set of software libraries. When they are used to develop associated clients, these libraries provide graphical consistency for the client windows, menus, buttons, and other onscreen components, along with some common keyboard controls and client dialogs. In this chapter, you learn how to work with GNOME and learn something about the version of X that is included with Ubuntu. If you want to learn more about some of the other desktop environments that are available to use with Ubuntu, including KDE and Xfce, take a look at Chapter 7, "Other Ubuntu Desktops."

The Ubuntu Family

When people talk about Ubuntu, they generally mean the standard distribution, which uses the GNOME window manager. However, a number of derivatives of Ubuntu use other window managers, and they are freely available for you to download and use. For instance, Kubuntu uses the KDE window manager, whereas Xubuntu uses Xfce. Despite their visual differences, all three rely on the same core system. It is easy for you to add that window manager to your existing system by using one of the **`*-desktop`** metapackages. These packages are designed to install all the base applications that are linked to that version of Ubuntu, so installing the **`kubuntu-desktop`** package automatically installs all the software needed for Kubuntu while retaining your existing desktop environment.

Basic X Concepts

The underlying engine of X11 is the X protocol, which provides a system of managing displays on local and remote desktops. The protocol uses a client/server model that allows an abstraction of the drawing of client windows and other decorations locally and over a network. An X server draws client windows, dialog boxes, and buttons that are specific to the local hardware and in response to client requests. The client, however, does not have to be specific to the local hardware. This means that system administrators can set up a network with a large server and clients and enable users to view and use those clients on workstations with totally different CPUs and graphics displays.

Because X offers users a form of distributed processing, Ubuntu can be used as a very cheap desktop platform for clients that connect to a powerful X server. The more powerful the X server, the larger the number of X-based clients that can be accommodated. This functionality can breathe new life into older hardware, pushing most of the graphical processing on to the server. A fast network is a must if you intend to run many X clients because X can become bandwidth hungry.

NOTE

A great way to demonstrate the capability of X to handle remote clients is Edubuntu's use of LTSP, covered in Chapter 31, "Linux Terminal Server Project (LTSP)." It is designed to set up a main server and a set of dumb terminals that display programs that are being run on the server.

X is hugely popular in the UNIX and Linux world for a variety of reasons. That it supports nearly every hardware graphics system is a strong point. This and strong multiplatform programming standards give it a solid foundation of developers committed to X. Another key benefit of X is its networking capability, which plays a central point in administration of many desktops and can also assist in the deployment of a thin-client computing environment. Being able to launch applications on remote desktops and also standardize installations highlight the versatility of this powerful application.

More recent versions of X have also included support for shaped windows (that is, nonrectangular), graphical login managers (also known as *display managers*), and compressed fonts. Each release of X brings more features designed to enhance the user experience, including being able to customize how X client applications appear, right down to buttons and windows. Most office and home environments run Linux and X on their local machines. The more-enlightened companies and users harness the power of the networking features of X, enabling thin-client environments and allowing the use of customized desktops designed specifically for that company. Having applications launch from a single location makes the lives of system administrators a lot easier because they have to work on only one machine, rather than several.

Using X

X.Org (www.x.org) is the X server that is used with Ubuntu. The base distribution consists of many packages, including the server, support and development libraries, fonts, various clients, and documentation. An additional 1,000 or more X clients, fonts, and documentation are also available in the Ubuntu repositories.

The **/usr** directory and its subdirectories contain the majority of the **Xorg** software. Some important subdirectories are as follows:

- **/usr/bin**—This is the location of the X server and various X clients. (Note that not all X clients require active X sessions.)
- **/usr/include**—This is the path to the files necessary for developing X clients and graphics such as icons.
- **/usr/lib**—This directory contains required software libraries to support the X server and clients.
- **/usr/lib/X11**—This directory contains fonts, default client resources, system resources, documentation, and other files that are used during X sessions and for various X clients. You will also find a symbolic link to this directory, named **X11**, under the **/usr/lib** directory.

. **/usr/lib/modules**—This path to drivers and the X server modules used by the X server enables use of various graphics cards.

The main components required for an active local X session are installed on your system if you choose to use a graphical desktop. These components are the X server, miscellaneous fonts, a *terminal client* (that is, a program that provides access to a shell prompt), and a client known as a *window manager*. Window managers administer onscreen displays, including overlapping and tiling windows, command buttons, title bars, and other onscreen decorations and features.

Elements of the `xorg.conf` File

Traditionally, the most important file for **Xorg** has been the **xorg.conf** configuration file. This file contained configuration information that was vital for X to function correctly and was usually created during the installation of Ubuntu.

BulletProofX

Ubuntu now comes with a useful feature called BulletProofX. In short, it is designed to work no matter what may happen. So, in the event of some cataclysmic event that destroys your main X system, you will still have some graphical way of getting yourself back into a fully functional X-based system.

Modern versions of **Xorg** do not create an **xorg.conf** file by default. Instead, various files ending in ***.conf** reside in the **/usr/share/X11/xorg.conf.d** directory and are automatically loaded by X at boot, prior to reading any **xorg.conf**. These files can each contain one or more sections in the same format used by **xorg.conf**. Users can create the file and continue making custom configurations in **/etc/xorg.conf** as has been traditionally done, but the file is not created by default. What is included in the previously mentioned individual files should not be changed, but you may override those settings by creating your own **xorg.conf** file.

NOTE

We refer to using an **xorg.conf** file from here on, but keep the preceding information in mind to prevent confusion.

Let's take a look at the potential contents of **xorg.conf** so that you can get an idea of what X is looking for. The components, or sections, of the **xorg.conf** file specify the X session or *server layout*, along with pathnames for files that are used by the server, any options relating directly to the server, any optional support modules needed, information relating to the mouse and keyboard attached to the system, the graphics card installed,

the monitor in use, and the resolution and color depth that Ubuntu uses. These are the essential components:

- **ServerLayout**—Defines the display, defines one or more screen layouts, and names input devices.
- **Files**—Defines the location of colors, fonts, or port number of the font server.
- **Module**—Tells the X server what graphics display support code modules to load.
- **InputDevice**—Defines the input devices, such as the keyboard and mouse; multiple devices can be used.
- **Monitor**—Defines the capabilities of any attached display; multiple monitors can be used.
- **Device**—Defines one or more graphics cards and specifies what optional features (if any) to enable or disable.
- **Screen**—Defines one or more resolutions, color depths, perhaps a default color depth, and other settings.

The following sections provide short descriptions of these elements; the **xorg.conf** man page contains full documentation of all the options and other keywords you can use to customize your desktop settings.

The ServerLayout Section

As noted previously, the **ServerLayout** section of the **xorg.conf** file defines the display and screen layouts, and it names the input devices. A typical **ServerLayout** section from an automatically configured **xorg.conf** file might look like this:

```
Section “ServerLayout”
        Identifier     “single head configuration”
        Screen      0  “Screen0” 0 0
        InputDevice    “Mouse0” “CorePointer”
        InputDevice    “Keyboard0” “CoreKeyboard”
        InputDevice    “DevInputMice” “AlwaysCore”
EndSection
```

In this example, a single display is used (the numbers designate the position of a screen), and two default input devices, Mouse0 and Keyboard0, are used for the session.

The Files Section

The **Files** section of the **xorg.conf** file might look like this:

```
Section “Files”
    RgbPath     “/usr/lib/X11/rgb”
    FontPath   “unix/:7100”
EndSection
```

This section lists available session colors (by name, in the text file **rgb.txt**) and the port number to the X font server. The font server, xfs, is started at boot and does not require an active X session. If a font server is not used, the **FontPath** entry could instead list each font directory under the **/usr/lib/X11/fonts** directory, as in this example:

```
FontPath "/usr/lib/X11/fonts/100dpi"
FontPath "/usr/lib/X11/fonts/misc"
FontPath "/usr/lib/X11/fonts/75dpi"
FontPath "/usr/lib/X11/fonts/type1"
FontPath "/usr/lib/X11/fonts/Speedo"
...
```

These directories contain the default compressed fonts that are available for use during the X session. The font server is configured by using the file named **config** under the **/etc/X11/fs** directory. This file contains a listing, or catalog, of fonts for use by the font server. By adding an alternate-server entry in this file and restarting the font server, you can specify remote font servers for use during X sessions. This can help centralize font support and reduce local storage requirements (even though only 25MB is required for the almost 5,000 fonts installed with Ubuntu and X).

The Module Section

The **Module** section of the **xorg.conf** file specifies loadable modules or drivers to load for the X session. This section might look like this:

```
Section "Module"
        Load  "dbe"
        Load  "extmod"
        Load  "fbdevhw"
        Load "glx"
        Load "record"
        Load  "freetype"
        Load  "type1"
        Load  "dri"

EndSection
```

These modules can range from special video card support to font rasterizers. The modules are located in subdirectories under the **/usr/lib/modules** directory.

The InputDevice Section

The **InputDevice** section configures a specific device, such as a keyboard or mouse, as in this example:

```
Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "kbd"
```

```
        Option  "XkbModel"      "pc105"
        Option  "XkbLayout"     "us"
EndSection
Section "InputDevice"
        Identifier  "Mouse0"
        Driver      "mouse"
        Option      "Protocol" "IMPS/2"
        Option      "Device" "/dev/input/mice"
        Option      "ZAxisMapping" "4 5"
        Option      "Emulate3Buttons" "yes"
EndSection
```


You can configure multiple devices, and multiple **InputDevice** sections might exist. The preceding example specifies a basic keyboard and a two-button PS/2 mouse (actually, a Dell touchpad pointer). An **InputDevice** section that specifies use of a USB device could be used at the same time (to enable mousing with PS/2 and USB pointers) and might look like this:

```
Section "InputDevice"
        Identifier  "Mouse0"
        Driver      "mouse"
        Option      "Device" "/dev/input/mice"
        Option      "Protocol" "IMPS/2"
        Option      "Emulate3Buttons" "off"
        Option      "ZAxisMapping" "4 5"
EndSection
```

The Monitor Section

The **Monitor** section configures the designated display device as declared in the **ServerLayout** section, as shown in this example:

```
Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        DisplaySize  300     220
        HorizSync   31.5-48.5
        VertRefresh 50-70
        Option "dpms"
EndSection
```

Note that the X server automatically determines the best video timings according to the horizontal and vertical sync and refresh values in this section. If required, old-style mode-line entries (used by distributions and servers prior to XFree86 4.0) might still be used. If the monitor is automatically detected when you configure X (see the "Configuring X" section, later in this chapter), its definition and capabilities are inserted in your **xorg.conf** file from the MonitorsDB database. This database contains more than 600 monitors and is located in the **/usr/share/hwdata** directory.

The `Device` Section

The **`Device`** section provides details about the video graphics chipset used by the computer, as in this example:

```
Section "Device"
        Identifier      "Intel Corporation Mobile 945GM/GMS,\943/940GML Express
Integrated Graphics Controller"
        Driver          "intel"
        BusID           "PCI:0:2:0"
EndSection
```

This example identifies an installed video card as using an integrated Intel 945 graphics chipset. The **`Driver`** entry tells the **`Xorg`** server to load the intel kernel module. Different chipsets have different options. For example, here's the entry for a NeoMagic video chipset:

```
Section "Device"
        Identifier   "NeoMagic (laptop/notebook)"
        Driver       "neomagic"
        VendorName   "NeoMagic (laptop/notebook)"
        BoardName     "NeoMagic (laptop/notebook)"
    Option     "externDisp"
    Option     "internDisp"
EndSection
```

In this example, the **`Device`** section specifies the driver for the graphics card (**`neomagic_drv.o`**) and enables two chipset options (**`externDisp`** and **`internDisp`**) to allow display on the laptop's LCD screen and an attached monitor.

The **`Xorg`** server supports hundreds of different video chipsets. If you configure X11 but subsequently change the installed video card, you need to edit the existing **`Device`** section or generate a new **`xorg.conf`** file, using one of the X configuration tools discussed in this chapter, to reflect the new card's capabilities. You can find details about options for some chipsets in a companion man page. You should look at these sources for hints about optimizations and troubleshooting.

The `Screen` Section

The **`Screen`** section ties together the information from the previous sections (using the **`Screen0`**, **`Device`**, and **`Monitor Identifier`** entries). It can also specify one or more color depths and resolutions for the session. Here's an example:

```
Section "Screen"
        Identifier "Screen0"
        Device     "Videocard0"
        Monitor    "Monitor0"
        DefaultDepth     24
        SubSection "Display"
                Viewport   0 0
```

```
            Depth     16
            Modes    “1024x768” “800x600” “640x480”
        EndSubSection

EndSection
```

In this example, a color depth of thousands of colors and a resolution of 1024768 is the default, with optional resolutions of 800 x 600 and 64 x 480. Multiple **Display** subsection entries with different color depths and resolutions (with settings such as Depth 24 for millions of colors) can be used if supported by the graphics card and monitor combination. You can also use a **DefaultDepth** entry (which is 24, or thousands of colors, in the example), along with a specific color depth to standardize display depths in installations.

You can also specify a desktop resolution larger than that supported by the hardware in your monitor or notebook display. This setting is known as a *virtual* resolution in the **Display** subsection. This allows, for example, an 800 x 600 display to pan (that is, slide around inside) a virtual window of 1024 x 768.

NOTE

If your monitor and graphics card support multiple resolutions and the settings are properly configured, you can use the key combination of Ctrl+Alt+(Keypad)+ or Ctrl+Alt+(Keypad)- to change resolutions on-the-fly during your X session.

Starting X

You can start X sessions in a variety of ways. The Ubuntu installer sets up the system to have Linux boot directly to an X session using a display manager called *LightDM*, for Light(weight) Display Manager. This is an X client that provides a graphical login. After you log in, you use a local session (running on your computer) or, if the system is configured to do so, an X session running on a remote computer on the network.

Logging in via a display manager requires you to enter a username and password. You can also start X sessions from the command line. The following sections describe these two methods.

NOTE

If you have used the Server install, your system boots to a text login. See Chapter 10, “Command-Line Quickstart,” for more information about what to do here.

Using a Display Manager

An X display manager presents a graphical login that requires a username and password to be entered before access is granted to the X desktop. It also allows you to choose a different desktop for your X session. Whether an X display manager is presented after you boot

Linux is controlled by a *runlevel*—a system state entry in **/etc/event.d/**. The following runlevels as handled by Ubuntu are as follows:

- **0**—Halt (Do *not* set **initdefault** to this.)
- **1**—Multiuser text mode
- **2**—X graphical multiuser mode
- **6**—Reboot (Do *not* set **initdefault** to this.)

You may see mention of runlevels 3 through 5; these can be ignored because they are treated the same as runlevel 2 in Ubuntu. Historically, Ubuntu used the **/etc/inittab** file to handle runlevels, but with Upstart, this file no longer exists. (See Chapter 14, "The Boot Process," for information about Upstart.) Instead, there are several files under the **/etc/events.d/** directory, including an individual file for each of the virtual consoles (accessible by pressing Ctrl+Alt+F1 to F7). However, you can still create an **inittab** file, if you want to override any defaults held by Ubuntu. Make sure you create the **inittab** file as root and include at least one line similar to the following:

```
id:1:initdefault:
```

This forces your system to start up in text mode.

Changing Window Managers

Ubuntu makes it fairly painless to switch to another window manager or desktop environment. By *desktop environment*, we refer to not only the window manager but also the suite of related applications, such as productivity or configuration tools.

First, you need to ensure that you have the relevant desktop environment installed on your system; the easiest way to do this is by installing the relevant ***-desktop** package (as described earlier in the sidebar "The Ubuntu Family") You can do this by installing the package **kubuntu-desktop**, for example (in the case of a KDE desktop); just search for "desktop" and look for Xubuntu or Kubuntu, and so on. After the download and installation is complete (you might want to grab a coffee while you wait because these packages include a ton of dependencies and take some time to download, install, and configure), you are all set to change environments.

Next, you need to log out of Ubuntu. When you return to the login page, select your name as usual and then select the session named for the desktop you want to use. Chapter 7 provides a brief introduction to some of the desktop environments other than Unity that are available from the Ubuntu repositories.

Using Unity, a Primer

Unity is a fresh take on the GUI. It has been created with the assistance of user interface professionals and graphic designers and aims toward elegance, beauty, and efficiency while providing obvious ways to perform tasks. Unity is a long-term project being created

in steps that coincide with Ubuntu's six-month release schedule. Each new Ubuntu release benefits from new features and extended functionality. Unity is new, but it is already growing in popularity and is considered to be at the forefront of GUI development, not just for Linux, but for human-computer interaction in general. This section gives a tour of the basic features and use and then moves into more specific details that are likely to thrill power users.

Mark Shuttleworth on Unity

"Our goal with Unity is unprecedented ease of use, visual style, and performance on the Linux desktop."

—from a blog post on August 16, 2011; www.markshuttleworth.com/archives/717

The Desktop

When you boot your computer and log in to a Unity session, you are greeted by the desktop you see in Figure 3.1. This screen is made up of many parts, each of which is described in this chapter.

FIGURE 3.1 The default look of Ubuntu Unity.

The Launcher

To the left of the screen is a vertical bar. This is the Launcher (see Figure 3.2). Icons in the Launcher represent programs that are either running or which have been set to be listed in the Launcher at all times. Click an icon to open a program. Right-click an icon for additional options, including an option to keep an item in the Launcher at all times.

FIGURE 3.2 The Launcher as it looks by default.

A special icon in the Launcher is the Workspace Switcher. It is near the bottom of the main set of icons and looks like a rectangle divided into four. Click this icon to show all workspaces. Double-click one of the workspaces when they appear on the screen to switch to it.

Workspaces

Workspaces are kind of like extra screens. Unity has four workspaces by default. You can open an application on one workspace, and then keep it open there while switching to another workspace where you can open something else. Then, switching back and forth is as easy as pressing a couple of keys. The four workspaces are arranged in a 2 x 2 square. To switch between them, use Ctrl+Alt and an arrow key.

The Dash

Click the Ubuntu logo icon at the top of the Launcher to open the Dash (see Figure 3.3). The Dash is the primary way to find programs and files in Unity. At the bottom of the Dash is a row of icons. These represent *lenses*, which are different ways to focus your searching with the Dash.

FIGURE 3.3 The Dash helps you find programs and files.

The default Dash lens is represented by a house icon. This lens gives a set of shortcuts to often-used features. At the top is a Search box that searches your computer for programs, files, and directories that match any search terms input and then categorizes the results according to type (see Figure 3.4).

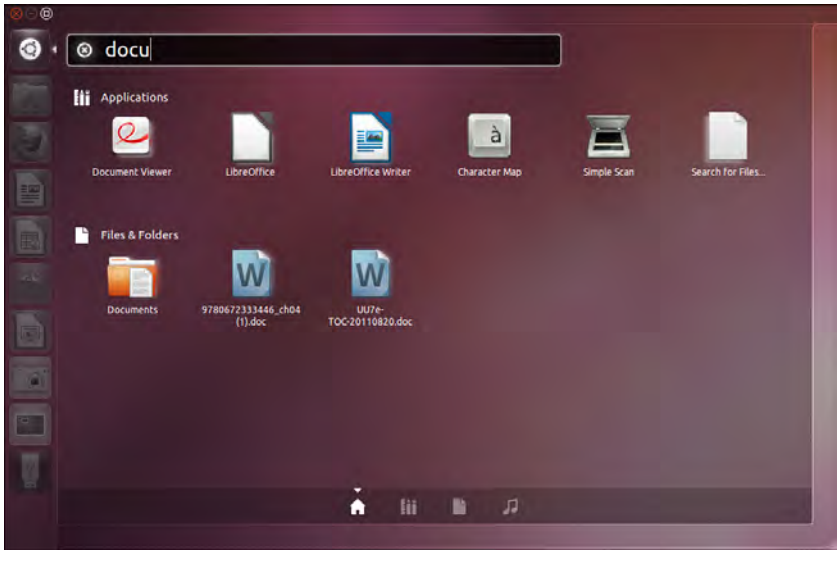

FIGURE 3.4 Sorted results in the main Dash lens.

The applications Dash lens is represented by a ruler-pencil-pen icon. This lens will list and search installed applications and also applications that are available from the Ubuntu Software Center (see Chapter 9, “Managing Software”), as in Figure 3.5. You can click to view more results and perform a visual search if you do not know the name of a program.

FIGURE 3.5 Exploring applications in the Dash.

The files and folders Dash lens is represented by a sheet of paper icon. You can search by entering a term in the Search box or visually by clicking through what is shown here, as in Figure 3.6.

Other Dash lenses may appear depending on your configuration. Possibilities include, but are not limited to, a Music lens for audio files and a Gwibber lens for interactive media such as Twitter and Facebook updates. More are likely to arrive as Unity matures.

In any Dash lens in which it appears, you can click Filter results to view predefined ways to filter what is shown. Figure 3.7 shows a list of available filters for the applications lens.

The Panel

The panel is the bar across the top of the screen. When a program is being used, it will house various menus and buttons. (Some of these appear only when you hover your mouse over the panel; so if you can’t find the File menu or close button, point at the panel with your mouse, and it is likely to appear.) The right side of the panel has a series of useful indicators. The Networking indicator was discussed in Chapter 2, “Post-Installation Configuration,” because it is often needed right away to connect a computer to the Internet. Other indicators:

- Help manage your mail and social applications like chat (envelope icon).
- Show your batter status and provide quick access to power settings.

- Provide volume and playback controls for the Banshee media player.
- Give quick access to the clock and calendar.
- Show active users and sessions.
- Give convenient access to commonly used system administration options and the logout menu.

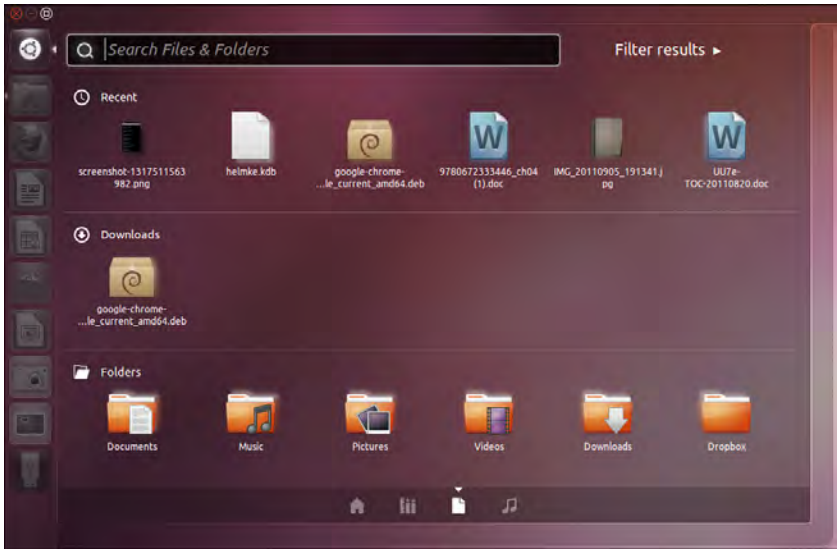

FIGURE 3.6 Exploring applications in the Dash.

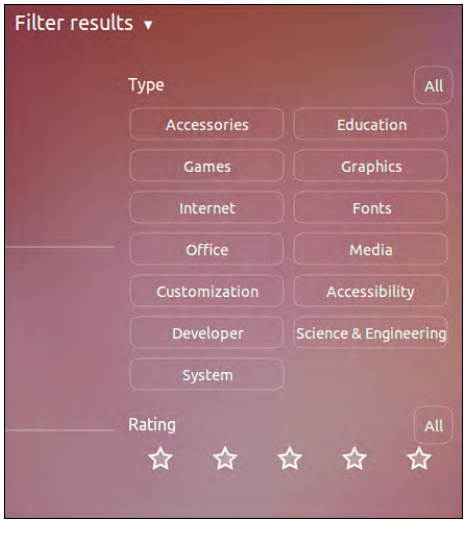

FIGURE 3.7 Filtering applications in the Dash.

Customizing and Configuring Unity

Everyone likes to customize. Unity is quite new, and as a result, not everything is customizable (yet). All existing options for customization and system settings are available in the System Settings menu.

System Settings

Search for "system settings" in the Dash. Click the System Settings icon to open the menu. The menu includes entries that allow you to change your desktop appearance, set up online accounts, configure additional languages for the system to use for colors, keyboards, displays, and much more (see Figure 3.8).

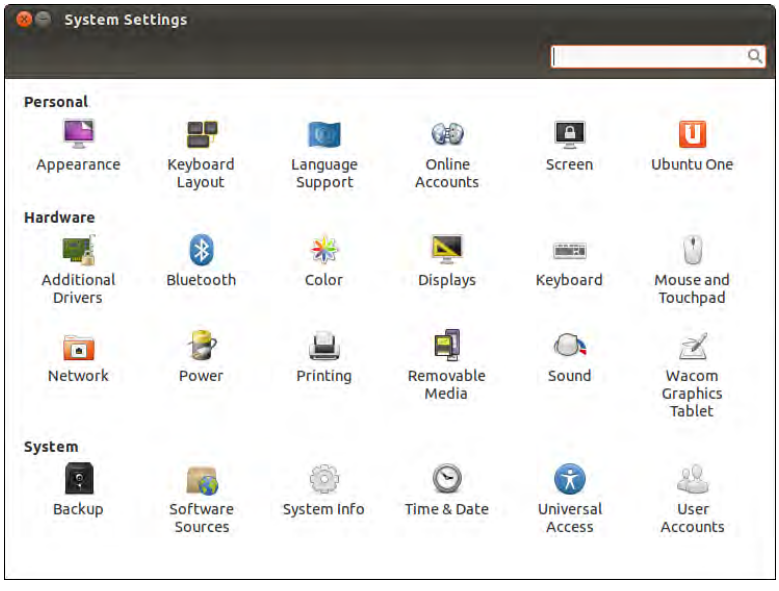

FIGURE 3.8 The System Settings menu.

Other Settings

Many other GUI-related settings can be configured or adjusted by installing a package from the Ubuntu repositories called the CompizConfig Settings Manager (see Chapter 9). The documentation for these settings is not always clear, so go slowly.

Power Shortcuts

Often, power users want to avoid using the mouse. They don't want to take their hands off of the keyboard unless absolutely necessary because doing so slows them down. Here are some of the more useful keyboard shortcuts that readers of this book are likely to

appreciate, many of which can be configured or adjusted by installing a package from the Ubuntu repositories called the CompizConfig Settings Manager (see Chapter 9):

- Use the Special key (that is, the Windows key) to open the Dash. Click Esc to close the Dash without selecting anything in it.
- Use the middle mouse button to click on an icon in the Launcher to open more than one instance of the same program. On a laptop, you can do this by clicking both right and left buttons simultaneously.
- Holding Alt+Tab brings up a menu of icons showing open programs so that you may switch between them quickly. Hold Alt and press Tab repeatedly to move from one program to the next until you reach the one you desire. Shift+Alt+Tab scrolls through them in reverse order. This and the next shortcut work together as a powerful combination when the application you want has multiple windows or instances.
- Switching from Alt+Tab to Alt+` (on an English keyboard; for other keyboards, read on) modifies the Alt+Tab menu so that it shows miniature images of open program windows when there are multiple instances of the same program. Hold Alt and press ` repeatedly to scroll through and select a specific instance. If you begin this while the desktop focus is on an instance of a program that has multiple instances open, you automatically begin with this set of open instances to select from among them. (On an English keyboard, ` is the key just above the Tab key. For other language keyboards, it is the whatever key in this same location, regardless of the character represented on the key itself.)
- Icons in the Launcher indicate when multiple instances of a program are open or when a program has multiple windows. Click twice on an icon in the Launcher that shows multiple arrows next to it, and all of its windows will be displayed for you to select the one you desire.
- Click PrtSc (Print Screen) to take a screenshot of the current workspace.
- Use Special + T to open the trash can.
- Click Ctrl+Alt+*arrow-key* to move up, down, right, and left from workspace to workspace.

References

- **www.gnome.org/**—The launch point for more information about GNOME, links to new clients, and GNOME development projects.
- **www.x.org/**—Curators of the X Window System.
- **www.x.org/Downloads_mirror.html**—Want to download the source to the latest revision of X? Start at this list of mirror sites.
- **www.xfree86.org/**—Home of The XFree86 Project, Inc., which provided a graphical interface for Linux for nearly 10 years.

- **https://wiki.ubuntu.com/X**—The place to get started when learning about X and Ubuntu.
- **https://wiki.ubuntu.com/X/Config**—Great information about configuring X on Ubuntu.
- **http://www.freedesktop.org/wiki/Software/LightDM**—LightDM, a new window manager.
- **http://castrojo.tumblr.com/post/4795149014/the-power-users-guide-to-unity**—From Jorge Castro, a leader in the Ubuntu community, this is a great guide to moving beyond the basics. Because Unity is still maturing, some info here may be out-of-date, but at the time of this writing, it is the best resource outside of this book.

CHAPTER 4

On the Internet

IN THIS CHAPTER

- Getting Started with Firefox
- Checking Out Google Chrome and Chromium
- Choosing an Email Client
- RSS Readers
- Instant Messaging and Video Conferencing with Empathy
- Internet Relay Chat
- Usenet Newsgroups
- Ubuntu One Cloud Storage
- References

The Internet is everywhere. From cell phones to offices, from game consoles to iPads, we are surrounded by multiple access routes to online information and communication. Ubuntu is no outsider when it comes to accessing information through the Internet; it comes equipped with all the tools you need to connect to other people across the globe.

In this chapter, we look at some of the popular Internet applications that are available with Ubuntu. You'll find out about Firefox and the new option, Google Chrome. We also investigate some of the email clients available. Other topics include RSS feed readers, instant messaging (through IRC and other networks), and reading newsgroups.

A Brief Introduction to the Internet

The Internet itself was first brought to life by the U.S. Department of Defense in 1969. It was called ARPANET after the Department of Defense's Advanced Research Projects Agency. Designed to build a network that would withstand major catastrophe (this was the peak of the Cold War), it soon grew to encompass more and more networks to build the Internet. Then, in 1991, Tim Berners-Lee of CERN developed the idea of the World Wide Web, including *Hypertext Transfer Protocol (HTTP)* and *Hypertext Markup Language (HTML)*. This gave us what we now know to be the Internet.

Getting Started with Firefox

One of the most popular web browsers, and in fact the default web browser in Ubuntu, is Mozilla Firefox (see Figure 4.1). Built on a solid code base that is derived from the Mozilla suite, Firefox offers an open source and more secure alternative to surfing the Internet using Internet Explorer, regardless of your operating system.

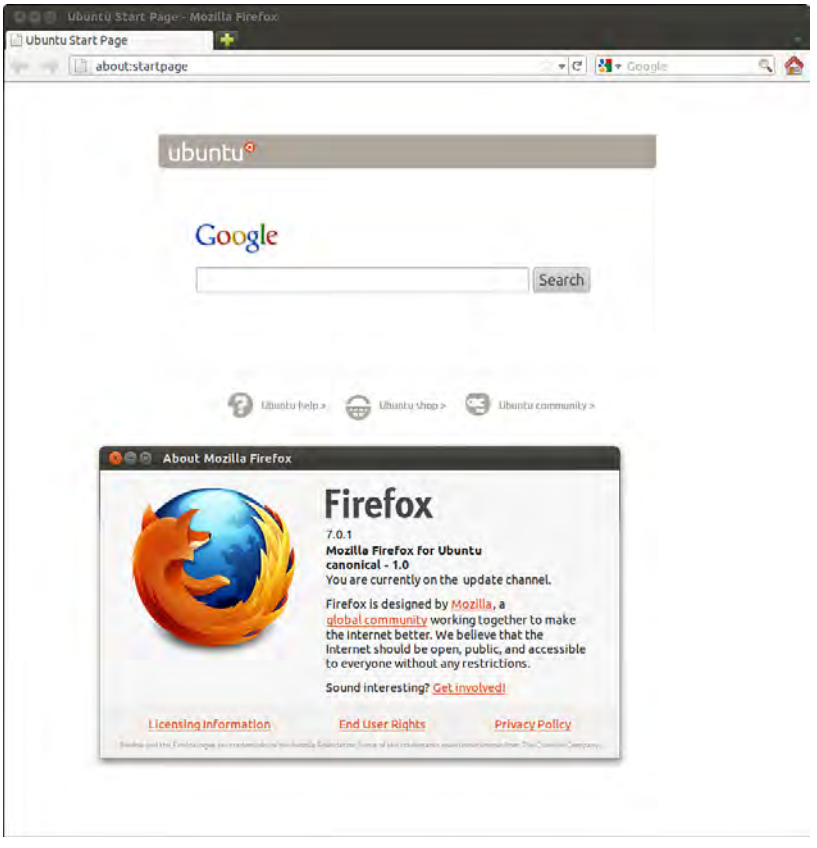

FIGURE 4.1 Mozilla Firefox—rediscover the Web. Firefox enables you to add on numerous extensions, further enhancing your experience.

In Ubuntu, you can find Firefox in the menu at Applications, Internet. An even simpler way to start Firefox is to click the small Firefox icon in the top panel next to the System menu. Either way, Firefox opens.

Beyond the basic program are a wealth of plug-ins and extensions that can increase the capabilities of Firefox beyond simple web browsing. Plug-ins such as Shockwave, Flash, and Java are available for installation instantly, offered to you when first needed, as are multimedia codecs for viewing video content. Extensions provide useful additions to the

browsing experience. For example, ForecastFox is an extension that gives you your local weather conditions, and Bandwidth Tester is a tool that calculates your current bandwidth. As Firefox grows, there will be more and more extensions and plug-ins that you can use to enhance your browsing pleasure.

You can find and obtain these plug-ins and extensions easily because Mozilla developers have created a site dedicated to helping you get more from Firefox. You don't have to search to find the site as there is a link in the Firefox menu at Tools, Add-ons, Get Add-ons that takes you directly to it. Particular favorites are the Adblock Plus and the StumbleUpon plug-ins. Adblock Plus enables you to nuke all those annoying banners and animations that take up so much bandwidth while you are browsing. StumbleUpon is a neat plug-in that takes you to web pages based on your preferences. Be warned, though, that StumbleUpon can be quite addictive, and you will end up wasting away many hours clicking the stumble button.

Flash

You can easily enable Flash in your browser by installing the **`flashplugin-installer`** package or better yet, you can deal with the whole issue of codecs and functionality for your entire operating system right off by installing **`ubuntu-restricted-extras`**. Either way, Ubuntu downloads the official package from Adobe and installs it on your system. A browser restart is required before you try to access any Flash-enabled content.

To make things even easier, if you have not yet installed this package and visit a web page that requires flash, a pop-up window appears with information and a link to help you install it easily.

Another plug-in that we make a lot of use of is Xmarks. If, like us, you work across multiple computers and operating systems, you will no doubt have had to re-create bookmarks at every different computer and try to keep them the same. Xmarks makes this whole process much easier by allowing you to synchronize your bookmarks to multiple browsers (for example, Firefox and Google Chrome) running on different operating systems (Windows, Mac OS X, or Linux).

Checking Out Google Chrome and Chromium

Anyone who has been paying attention to tech news over the past year has heard about the explosive appearance of Google Chrome (see Figure 4.2). This new browser is fast, secure, expandable with extensions, and the hottest new toy on geek desktops. It is based on WebKit and other technologies, and it downloads and renders quickly while being standards compliant. You can learn more about the easy installation of Google Chrome at www.google.com/chrome?platform=linux.

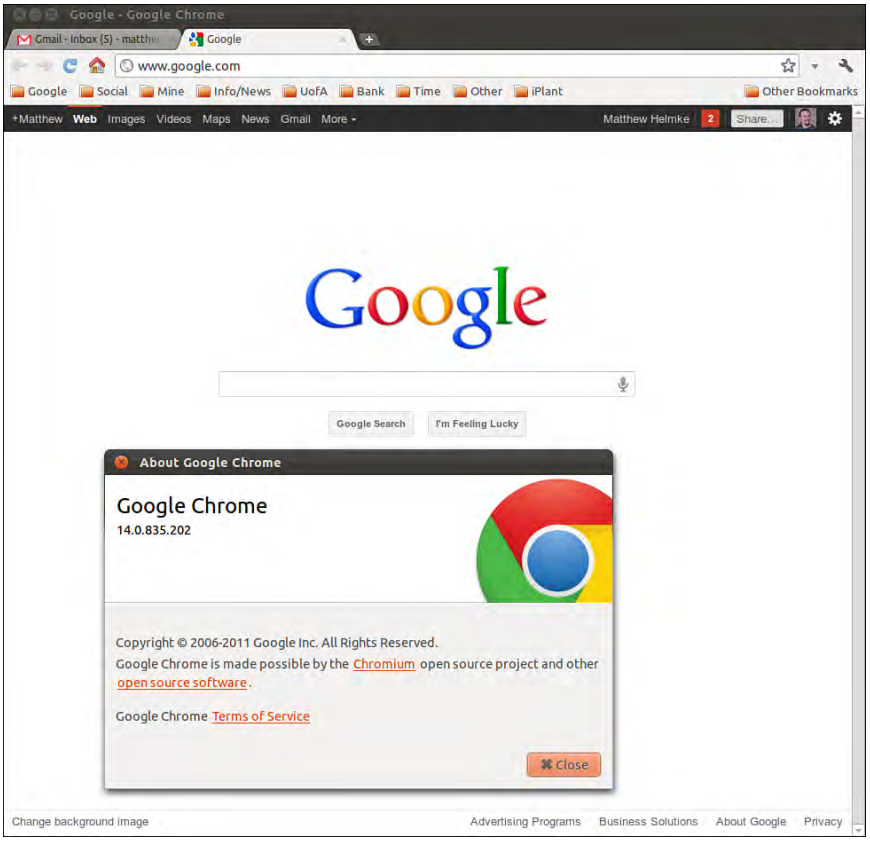

FIGURE 4.2 Google Chrome running in Ubuntu.

Chrome is not a completely open source, but the main foundational parts are made available as Google Chromium, shown in Figure 4.3, which has most of the functionality of Chrome, but it is missing the Google branding, automatic updates, and is used as the test and development foundation for Chrome releases. As a point of interest, chromium is the metal from which chrome is made, hence the names.

You can learn more about Chromium at http://code.google.com/chromium/. It used to be necessary to install Chromium from the Google website or from the Personal Package Archive of the Google Chromium developers. Now, you may simply install **`chromium-browser`** from the Ubuntu repositories.

Choosing an Email Client

Back in the earlier days of UNIX, there were various text-based email clients such as elm and pine (Pine Is Not Elm). Although they looked basic, they allowed the average user to interact with email, both for composing and reading correspondence, and had some sophisticated and useful features that might not have been expected. Still, when

computing became mainstream, there was a realization that people wanted friendly *graphical user interfaces (GUIs)*. Soon there came a flood of email clients, some of them even cross-platform and compatible among Linux, Windows, and Mac OS X, and even traditional UNIX.

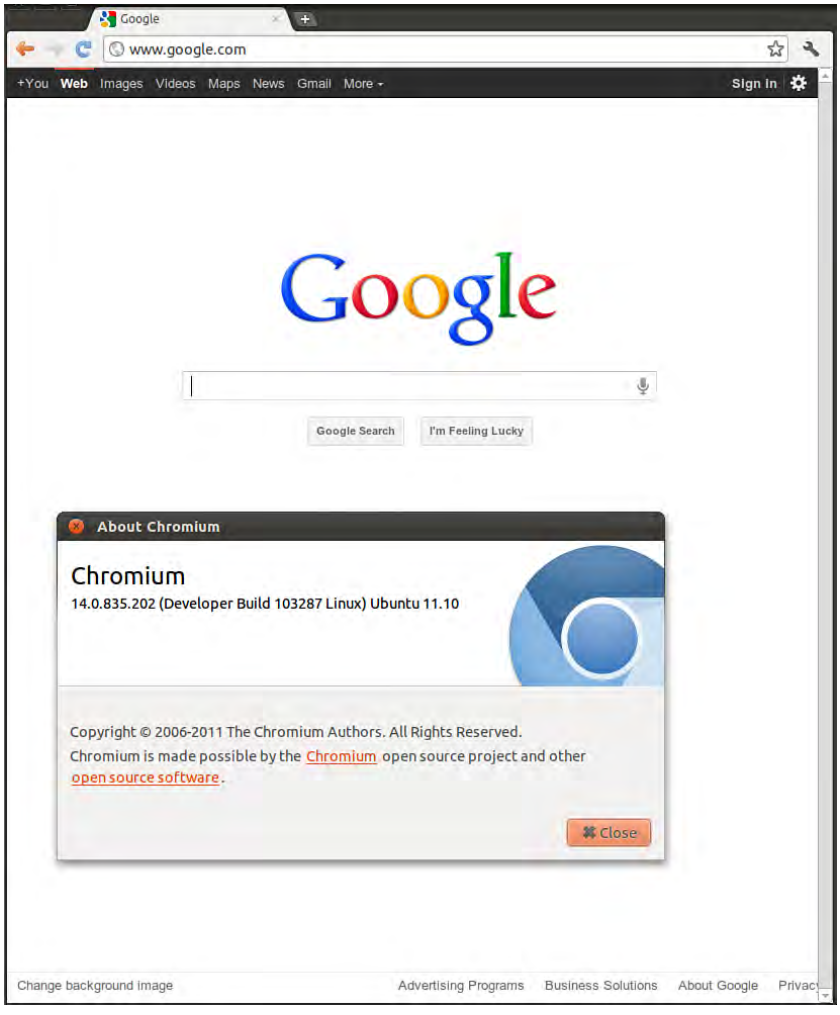

FIGURE 4.3 Chromium is the source from which Chrome is made, and it works very well.

Evolution

Up until Ubuntu 11.10, Evolution was the standard email client in Ubuntu, although to call it simply an email client is to sincerely underrate its usefulness as an application. Not only does it handle email, but it can also look after contacts and calendars and help you manage your tasks (see Figure 4.4).

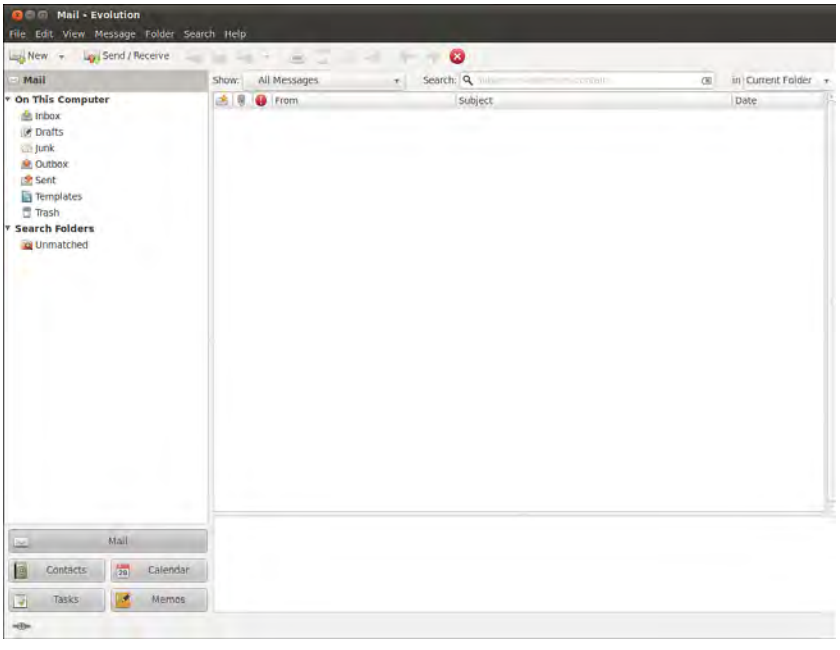

FIGURE 4.4 With Evolution, you can handle all your email and contacts, as well as make appointments and track tasks.

You need to have the following information to successfully configure Evolution (or any email client):

- Your email address
- Your incoming email server name and type (that is, pop.email.com, POP, and IMAP)
- Your username and password for the incoming mail server
- Your outgoing mail server name (that is, smtp.email.com)

As many people are moving to web-based mail options like Google's Gmail or Yahoo! Mail, email programs have started to decline in popularity. One thing that sets Evolution apart is that you can still use it for the other features and disable the email portion of the program by selecting "none" as your mail transfer agent instead of POP, IMAP, and so on. In addition, Evolution can interact with an existing web calendar program if it uses one of the common formats, such as CalDAV. This allows you to use your calendar remotely as you are used to, but also permits the Ubuntu desktop to have access to your events and embed them into your desktop panel (which it does by default using Evolution).

Mozilla Thunderbird

Mozilla Thunderbird (see Figure 4.5) is the sister program to Firefox and, starting with Ubuntu 11.10, is the default email application in Ubuntu. Whereas Firefox is designed to browse the Web, Thunderbird's specialty is communication. It can handle email, network news (see later in this chapter), and RSS feeds.

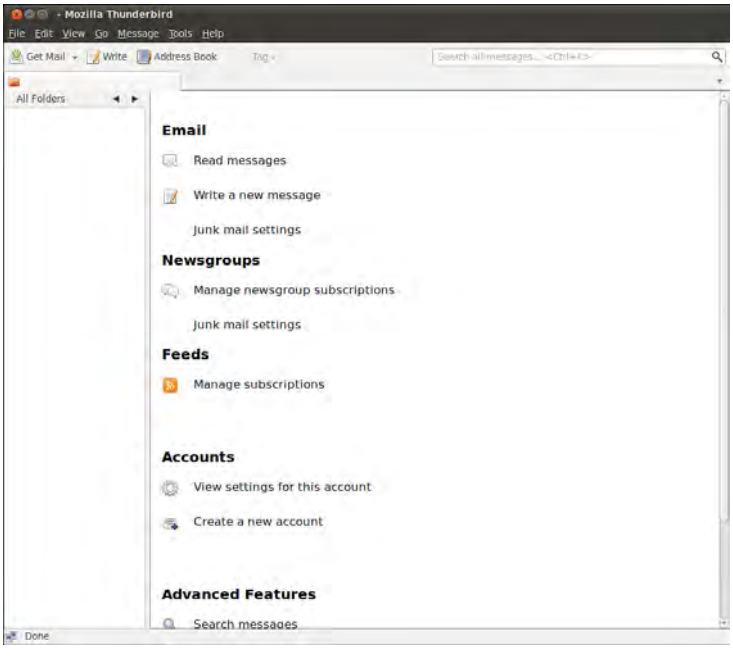

FIGURE 4.5 The natural companion to Firefox, Mozilla's lightweight email client, Thunderbird, can be found in use all over the world.

As with Firefox, there are many plug-ins and extensions to enhance your email. You can also use Thunderbird for Usenet news reading.

Other Mail Clients

The mail clients that we've covered are only a few of those available. Claws is very popular because it offers spell check while typing and is well suited for use in large network environments in which network overhead and RAM usage are important considerations. Mutt is an older text-based email program that is still beloved by sysadmins for its configurability and feature set, including the easy use of GPG keys for signing email. Alpine is another text-based program based on the classic Pine, with a few upgrades. Kmail is commonly

used by lovers of the KDE desktop that is commonly run as a part of many Linux-based operating systems. All of these and more are available for installation from the Ubuntu software repositories.

RSS Readers

RSS is one of the protocols of Web 2.0, the current generation of Internet content. RSS has really taken off, thanks to adoption by a large number of websites and portals.

The key advantage of RSS is that you can quickly read news from your choice of websites at a time that suits you and from one location, meaning you don't have to go to each site to view their content. Some services offer just the articles' headlines, whereas others offer full articles for you to view. RSS feeds can be accessed in various ways, even through your web browser!

Firefox

Firefox implements RSS feeds as what it calls Live Bookmarks (shown in Figure 4.6), which are essentially bookmarks with subbookmarks, each linking to a new page from your chosen website. Some people like to have several news sites grouped together in a folder called News in the Firefox toolbar, because this allows them to quickly browse through a collection of sites and pick out articles of interest.

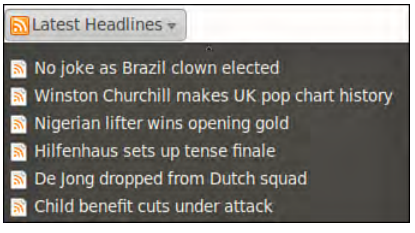

FIGURE 4.6 Live Bookmarks for Firefox, making all your news fixes just a mouse click away.

Liferea

Of course, not everyone wants to read RSS feeds with the browser. The main problem with reading RSS feeds with Firefox is that you get to see only the headline, rather than any actual text. This is where a dedicated RSS reader comes in handy, and Liferea is one of the best (see Figure 4.7).

It is not installed by default, but Liferea is available from the repositories under the name liferea. After it is installed, you can find it in the menu at Applications, Internet, Liferea.

By default, Liferea offers a number of RSS feeds, including Planet Debian, Groklaw, and Slashdot. Adding a new feed is straightforward. You select New Subscription under the Feeds menu and paste the URL of the RSS feed into the box. Liferea then retrieves all the current items available through that field and displays the feed name on the left side for you to select and start reading.

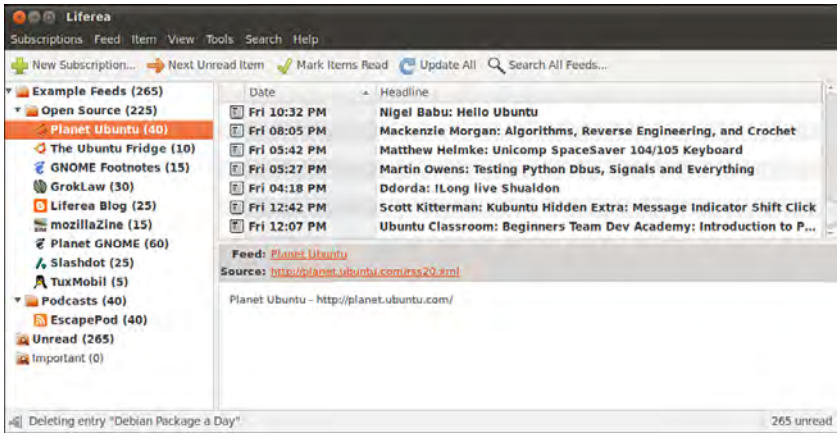

FIGURE 4.7 Read your daily news feeds with Liferea, a fantastic and easy-to-use RSS feed reader.

Instant Messaging and Video Conferencing with Empathy

Instant messaging is one of the biggest ways for people to interact over the Web. AOL was the primary force behind this, especially in America, but other networks and systems soon entered the market, providing users with a wealth of choice.

No longer just a consumer tool, instant messaging is now a part of the corporate world, with many companies deploying internal instant messaging software for collaboration.

Empathy is the default messaging program in Ubuntu; it can be found in the menu at Applications, Internet, Empathy and is shown in Figure 4.8. It supports text, voice, and video chat as well as file transfers using common protocols like Google Talk, AIM, and more. All you have to do is know your account information with a specific service and enter it in to Empathy—and the rest is easy.

You can learn more about Empathy at https://help.ubuntu.com/community/Empathy.

Internet Relay Chat

As documented in RFC 2812 and RFC 2813, the *Internet Relay Chat (IRC)* protocol is used for text conferencing. Like mail and news, IRC uses a client/server model. Although it is rare for an individual to set up and run an IRC server, it can be done. Most people use public IRC servers and access them with IRC clients. IRC is also the favorite and most common form of quick communication among developers in the Ubuntu community, so if that interests you, you will certainly want to learn to use this tool.

FIGURE 4.8 Empathy is easy to set up and use for great collaboration.

Ubuntu provides a number of graphical IRC clients in the repositories, including XChat, Pidgin, and Quassel. You can even use Empathy, although other products are better suited for this specific task. Ubuntu also makes the console clients epic and irssi available in the repositories for those who eschew X and a graphical interface. Don't laugh—there are actually many who do this, and not just server admins, especially because they can be run on a remote server and accessed over *Secure Shell (SSH)* using Byobu or Screen, allowing them to be left running 24/7 on that remote server. For more about Byobu and Screen, see Chapter 11, "Command-Line Master Class." If you don't already have a favorite IRC client, you should try them all.

XChat is a very popular IRC client, and the most common recommendation for IRC newcomers using Ubuntu, so we use it as an example. After you install the program, the documentation for XChat is available in **/usr/share/docs/xchat**. It is a good idea to read that before you begin because they include an introduction and cover some of the basics of IRC.

The XChat application enables you to assign yourself up to three nicknames. You can also specify your real name and your username. Because many people choose not to use their real names in IRC chat, you are free to enter any names you desire in any of the spaces provided. You can select multiple nicknames; you might be banned from an IRC channel under one name, and you could then rejoin using another. If this seems slightly juvenile to you, you are beginning to get an idea of the type of behavior on many IRC channels (but not the ones where serious developer and Ubuntu community interactions take place).

When you open the main XChat screen, a list of IRC servers appears, as shown in Figure 4.9. After you choose a server by double-clicking it, you can view a list of channels