

Windows System Programming

Fourth Edition

Johnson M. Hart

◆◆Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

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

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

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

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

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

International Sales
international@pearsoned.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data
Hart, Johnson M.

Windows system programming / Johnson M. Hart.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-321-65774-9 (hardback : alk. paper)

1. Application software—Development. 2. Microsoft Windows (Computer file). 3. Application program interfaces (Computer software). I. Title.

QA76.76.A65H373 2010
005.3—dc22

2009046939

Copyright © 2010 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax: (617) 671-3447

ISBN-13: 978-0-321-65774-9

ISBN-10: 0-321-65774-8

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.
First printing, February 2010

To Andrew and William

This page intentionally left blank

Contents

Figures xvii

Tables xix

Programs xxi

Program Runs xxv

Preface xxvii

About the Author xxxvii

CHAPTER 1 Getting Started with Windows 1

Operating System Essentials 1

Windows Evolution 2

Windows Versions 3

The Windows Market Role 5

Windows, Standards, and Open Systems 6

Windows Principles 7

32-bit and 64-bit Source Code Portability 10

The Standard C Library: When to Use It for
File Processing 10

What You Need to Use This Book 11

Example: A Simple Sequential File Copy 13

Summary 20

Exercises 22

CHAPTER 2 Using the Windows File System and Character I/O 25

The Windows File Systems 26

File Naming 27

Opening, Reading, Writing, and Closing Files	28
Interlude: Unicode and Generic Characters	34
Unicode Strategies	37
Example: Error Processing	38
Standard Devices	39
Example: Copying Multiple Files to Standard Output	41
Example: Simple File Encryption	43
File and Directory Management	46
Console I/O	51
Example: Printing and Prompting	53
Example: Printing the Current Directory	55
Summary	56
Exercises	57

CHAPTER 3 Advanced File and Directory Processing, and the Registry 59

The 64-Bit File System	59
File Pointers	60
Getting the File Size	64
Example: Random Record Updates	65
File Attributes and Directory Processing	70
Example: Listing File Attributes	75
Example: Setting File Times	78
File Processing Strategies	80
File Locking	81
The Registry	86
Registry Management	88
Example: Listing Registry Keys and Contents	92
Summary	96
Exercises	97

CHAPTER 4 Exception Handling 101

- Exceptions and Their Handlers 101
- Floating-Point Exceptions 108
- Errors and Exceptions 110
- Example: Treating Errors as Exceptions 112
- Termination Handlers 113
- Example: Using Termination Handlers to Improve Program Quality 117
- Example: Using a Filter Function 120
- Console Control Handlers 124
- Example: A Console Control Handler 126
- Vectored Exception Handling 128
- Summary 129
- Exercises 130

CHAPTER 5 Memory Management, Memory-Mapped Files, and DLLs 131

- Windows Memory Management Architecture 132
- Heaps 134
- Managing Heap Memory 137
- Example: Sorting Files with a Binary Search Tree 143
- Memory-Mapped Files 149
- Example: Sequential File Processing with Mapped Files 156
- Example: Sorting a Memory-Mapped File 158
- Example: Using Based Pointers 162
- Dynamic Link Libraries 167
- Example: Explicitly Linking a File Conversion Function 172
- The DLL Entry Point 174
- DLL Version Management 175
- Summary 177
- Exercises 178

CHAPTER 6 Process Management 181

Windows Processes and Threads	181
Process Creation	183
Process Identities	190
Duplicating Handles	191
Exiting and Terminating a Process	192
Waiting for a Process to Terminate	194
Environment Blocks and Strings	195
Example: Parallel Pattern Searching	197
Processes in a Multiprocessor Environment	201
Process Execution Times	202
Example: Process Execution Times	202
Generating Console Control Events	204
Example: Simple Job Management	205
Example: Using Job Objects	215
Summary	219
Exercises	220

CHAPTER 7 Threads and Scheduling 223

Thread Overview	223
Thread Basics	225
Thread Management	226
Using the C Library in Threads	231
Example: Multithreaded Pattern Searching	232
Performance Impact	235
The Boss/Worker and Other Threading Models	236
Example: Merge-Sort—Exploiting Multiple Processors	237
Introduction to Program Parallelism	244
Thread Local Storage	245
Process and Thread Priority and Scheduling	246
Thread States	249

Pitfalls and Common Mistakes 251

Timed Waits 252

Fibers 253

Summary 256

Exercises 256

CHAPTER 8 Thread Synchronization 259

The Need for Thread Synchronization 259

Thread Synchronization Objects 268

CRITICAL_SECTION Objects 269

A **CRITICAL_SECTION** for Protecting Shared Variables 271

Example: A Simple Producer/Consumer System 273

Mutexes 279

Semaphores 284

Events 287

Example: A Producer/Consumer System 289

More Mutex and **CRITICAL_SECTION** Guidelines 294

More Interlocked Functions 296

Memory Management Performance Considerations 297

Summary 298

Exercises 298

CHAPTER 9 Locking, Performance, and NT6 Enhancements 301

Synchronization Performance Impact 302

A Model Program for Performance Experimentation 307

Tuning Multiprocessor Performance with CS Spin Counts 307

NT6 Slim Reader/Writer Locks 309

Thread Pools to Reduce Thread Contention 312

I/O Completion Ports 316

NT6 Thread Pools 316

Summary: Locking Performance	324
Parallelism Revisited	325
Processor Affinity	329
Performance Guidelines and Pitfalls	331
Summary	332
Exercises	333

CHAPTER 10 Advanced Thread Synchronization 335

The Condition Variable Model and Safety Properties	336
Using <code>SignalObjectAndWait</code>	342
Example: A Threshold Barrier Object	344
A Queue Object	348
Example: Using Queues in a Multistage Pipeline	352
Windows NT6 Condition Variables	362
Asynchronous Procedure Calls	366
Queuing Asynchronous Procedure Calls	367
Alertable Wait States	368
Safe Thread Cancellation	371
Pthreads for Application Portability	372
Thread Stacks and the Number of Threads	372
Hints for Designing, Debugging, and Testing	372
Beyond the Windows API	375
Summary	375
Exercises	376

CHAPTER 11 Interprocess Communication 379

Anonymous Pipes	380
Example: I/O Redirection Using an Anonymous Pipe	380
Named Pipes	384
Named Pipe Transaction Functions	390
Example: A Client/Server Command Line Processor	393

Comments on the Client/Server
Command Line Processor 399

Mailslots 401

Pipe and Mailslot Creation, Connection,
and Naming 405

Example: A Server That Clients Can Locate 406

Summary 408

Exercises 408

CHAPTER 12 Network Programming with Windows Sockets 411

Windows Sockets 412

Socket Server Functions 414

Socket Client Functions 419

Comparing Named Pipes and Sockets 421

Example: A Socket Message Receive Function 422

Example: A Socket-Based Client 423

Example: A Socket-Based Server with New Features 426

In-Process Servers 434

Line-Oriented Messages, DLL Entry Points, and TLS 436

Example: A Thread-Safe DLL for Socket Messages 437

Example: An Alternative Thread-Safe DLL Strategy 442

Datagrams 445

Berkeley Sockets versus Windows Sockets 447

Overlapped I/O with Windows Sockets 447

Windows Sockets Additional Features 448

Summary 448

Exercises 449

CHAPTER 13 Windows Services 453

Writing Windows Services—Overview 454

The `main()` Function 454

`ServiceMain()` Functions 455

The Service Control Handler	460
Event Logging	461
Example: A Service “Wrapper”	461
Managing Windows Services	467
Summary: Service Operation and Management	471
Example: A Service Control Shell	472
Sharing Kernel Objects with a Service	476
Notes on Debugging a Service	477
Summary	478
Exercises	478

CHAPTER 14 Asynchronous Input/Output and Completion Ports 481

Overview of Windows Asynchronous I/O	482
Overlapped I/O	483
Example: Synchronizing on a File Handle	487
Example: File Conversion with Overlapped I/O and Multiple Buffers	487
Extended I/O with Completion Routines	492
Example: File Conversion with Extended I/O	496
Asynchronous I/O with Threads	500
Waitable Timers	501
Example: Using a Waitable Timer	503
I/O Completion Ports	505
Example: A Server Using I/O Completion Ports	509
Summary	516
Exercises	517

CHAPTER 15 Securing Windows Objects 519

Security Attributes	519
Security Overview: The Security Descriptor	520
Security Descriptor Control Flags	523