



Windows Debugging Disassembling Reversing

Third Edition

Practical Foundations Training Course

Dmitry Vostokov
Software Diagnostics Services

Windows Debugging, Disassembling, Reversing

Practical Foundations: Training Course

Third Edition

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

Copyright © 2025 by Dmitry Vostokov

Copyright © 2025 by Software Diagnostics Services

Copyright © 2025 by Dublin School of Security

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, transmitted in any form or by any means, or used for training artificial intelligence systems without the prior written permission of the publisher.

Product and company names mentioned in this book may be trademarks of their owners.

OpenTask books are available through booksellers and distributors worldwide. For further information or comments, send requests to press@opentask.com.

A CIP catalog record for this book is available from the British Library.

ISBN-13: 978-1-912636-47-1

Revision 3.00 (July 2025)

Summary of Contents

Contents.....	4
Preface to the Third Edition.....	9
Preface to the Second Edition.....	10
Preface to the First Edition.....	11
Combined Preface from Original Editions.....	12
About the Author	13
Chapter 1: Memory, Registers, and Simple Arithmetic.....	14
Chapter 2: Debug and Release Binaries	29
Chapter 3: Number Representations.....	45
Chapter 4: Pointers.....	49
Chapter 5: Bytes, Words, Double Words, and Quad Words	68
Chapter 6: Pointers to Memory	73
Chapter 7: Logical Instructions and RIP	91
Chapter 8: Reconstructing a Program with Pointers.....	99
Chapter 9: Memory and Stacks	109
Chapter 10: Local Variables	128
Chapter 11: Function Parameters.....	137
Chapter 12: More Instructions.....	147
Chapter 13: Function Pointer Parameters	158
Chapter 14: Arrays.....	162
Chapter 15: Floating Point.....	168
Chapter 16: Summary of Code Disassembly Patterns	172
References and Follow-up Courses.....	178

Contents

Contents.....	4
Preface to the Third Edition.....	9
Preface to the Second Edition.....	10
Preface to the First Edition.....	11
Combined Preface from Original Editions.....	12
About the Author	13
Chapter 1: Memory, Registers, and Simple Arithmetic.....	14
Memory and Registers inside an Idealized Computer	14
Memory and Registers inside an x64 Computer	15
“Arithmetic” Project: Memory Layout and Registers	16
“Arithmetic” Project: A Computer Program	17
“Arithmetic” Project: Assigning Numbers to Memory Locations	18
Assigning Numbers to Registers	20
“Arithmetic” Project: Adding Numbers to Memory Cells.....	21
Incrementing/Decrementing Numbers in Memory and Registers.....	24
Multiplying Numbers	27
Chapter 2: Debug and Release Binaries	29
“Arithmetic” Project: C & C++ Program.....	29
Downloading and Configuring WinDbg Debugger	30
WinDbg Disassembly Output – Debug Executable	32
WinDbg Disassembly Output – Release Executable.....	43
Chapter 3: Number Representations.....	45
Numbers and Their Representations.....	45
Decimal Representation (Base Ten).....	46
Ternary Representation (Base Three)	46
Binary Representation (Base Two)	47

Hexadecimal Representation (Base Sixteen).....	47
Why are Hexadecimals Used?	48
Chapter 4: Pointers.....	49
A Definition	49
Little-endian System	50
“Pointers” Project: Memory Layout and Registers.....	51
“Pointers” Project: Calculations	52
Using Pointers to Assign Numbers to Memory Cells	53
Adding Numbers Using Pointers.....	60
Multiplying Numbers Using Pointers.....	64
Chapter 5: Bytes, Words, Double Words, and Quad Words	68
Using Hexadecimal Numbers	68
Byte Granularity.....	68
Bit Granularity	69
Memory Layout	71
Chapter 6: Pointers to Memory	73
Pointers Revisited	73
Addressing Types.....	73
Registers Revisited	79
NULL Pointers.....	79
Invalid Pointers.....	79
Variables as Pointers	80
Pointer Initialization.....	80
Initialized and Uninitialized Data	81
More Pseudo Notation.....	81
“MemoryPointers” Project: Memory Layout.....	82
Chapter 7: Logical Instructions and RIP	91
Instruction Format	91

Logical Shift Instructions	92
Logical Operations	93
Zeroing Memory or Registers.....	94
Instruction Pointer	95
Code Section	97
Chapter 8: Reconstructing a Program with Pointers.....	99
Example of Disassembly Output: No Optimization	99
Reconstructing C & C++ Code: Part 1	102
Reconstructing C & C++ Code: Part 2	104
Reconstructing C & C++ Code: Part 3	105
Reconstructing C & C++ Code: Part 4	106
Reconstructing C & C++ Code: C & C++ program	107
Example of Disassembly Output: Optimized Program.....	108
Chapter 9: Memory and Stacks	109
Stack: A Definition.....	109
Stack Implementation in Memory.....	110
Things to Remember.....	111
PUSH Instruction	112
POP Instruction	113
Register Review	114
Application Memory Simplified.....	115
Stack Overflow.....	116
Jumps.....	118
Calls	120
Call Stack.....	122
Exploring Stack in WinDbg.....	124
Chapter 10: Local Variables	128
Stack Usage	128

Addressing Array Elements	129
Stack Structure (No Function Parameters)	130
Function Prolog.....	131
Function Epilog	132
“Local Variables” Project.....	133
Disassembly of Optimized Executable (Release Configuration).....	136
Chapter 11: Function Parameters.....	137
“FunctionParameters” Project	137
Stack Structure	138
Function Prolog and Epilog.....	140
Project Disassembled Code with Comments.....	142
Parameter Mismatch Problem	146
Chapter 12: More Instructions.....	147
CPU Flags Register	147
The Fastest Way to Fill Memory.....	148
Testing for 0.....	150
TEST – Logical Compare.....	151
CMP – Compare Two Operands.....	152
TEST or CMP?.....	153
Conditional Jumps.....	154
The Structure of Registers	155
Function Return Value	156
Using Byte Registers	157
Chapter 13: Function Pointer Parameters	158
“FunctionPointerParameters” Project.....	158
Commented Disassembly.....	159
Chapter 14: Arrays	162
Accessing Array Elements	162

“ArrayParameters” Project.....	163
Commented Disassembly.....	164
Chapter 15: Floating Point.....	168
SSE2 Instructions	168
“Arithmetic” FP Project.....	169
Commented Disassembly.....	170
Chapter 16: Summary of Code Disassembly Patterns	172
Function Prolog/Epilog	172
Parameters and Local Variables.....	174
LEA (Load Effective Address)	176
Accessing Parameters and Local Variables	177
References and Follow-up Courses.....	178