Windows Debugging

Practical Foundations

Dmitry Vostokov

OpenTask

Published by OpenTask, Republic of Ireland

Copyright © 2009 by Dmitry Vostokov

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

You must not circulate this book in any other binding or cover and you must impose the same condition on any acquirer.

OpenTask books are available through booksellers and distributors worldwide. For further information or comments send requests to:

press@opentask.com

Microsoft, MSDN, Visual C++, Visual Studio, Win32, Windows, Windows Server and Windows Vista are registered trademarks of Microsoft Corporation. Other product and company names mentioned in this book may be trademarks of their owners.

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

ISBN-13: 978-1-906717-10-0 (Paperback)

First printing, 2009

Summary of Contents

Preface 13

Acknowledgements 15

About the Author 17

Chapter 1: Memory, Registers and Simple Arithmetic 19

Chapter 2: Debug and Release Binaries 33

Chapter 3: Number Representations 45

Chapter 4: Pointers 53

Chapter 5: Bytes, Words and Double Words 71

Chapter 6: Pointers to Memory 77

Chapter 7: Logical Instructions and EIP 99

Chapter 8: Reconstructing a Program with Pointers 107

Chapter 9: Memory and Stacks 117

Chapter 10: Frame Pointer and Local Variables 137

Chapter 11: Function Parameters 151

Chapter 12: More Instructions 165

Chapter 13: Function Pointer Parameters 177

Chapter 14: Summary of Code Disassembly Patterns 183

Index 187

Contents

Preface 13

Acknowledgements 15

About the Author 17

Chapter 1: Memory, Registers and Simple Arithmetic 19

Memory and Registers inside an Idealized Computer 19

Memory and Registers inside Intel 32-bit PC 20

“Arithmetic” Project: Memory Layout and Registers 21

“Arithmetic” Project: A Computer Program 22

“Arithmetic” Project: Assigning Numbers to Memory Locations 23

Assigning Numbers to Registers 25

“Arithmetic” Project: Adding Numbers to Memory Cells 26

Incrementing/Decrementing Numbers in Memory and Registers 28

Multiplying Numbers 30

Multiplication and Registers 32

Chapter 2: Debug and Release Binaries 33

“Arithmetic” Project: C/C++ Program 33

Downloading and Configuring WinDbg Debugger 34

WinDbg Disassembly Output – Debug Executable 36

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) 47

Binary Representation (Base Two) 48

Hexadecimal Representation (Base Sixteen) 49

Why Hexadecimals are used? 50

Chapter 4: Pointers 53

A Definition 53

“Pointers” Project: Memory Layout and Registers 54

“Pointers” Project: Calculations 55

Using Pointers to Assign Numbers to Memory Cells 56

Adding Numbers Using Pointers 63

Multiplying Numbers Using Pointers 66

Chapter 5: Bytes, Words and Double Words 71

Using Hexadecimal Numbers 71

Byte Granularity 72

Bit Granularity 73

Memory Layout 74

Chapter 6: Pointers to Memory 77

Pointers Revisited 77

Addressing Types 78

Registers Revisited 84

NULL Pointers 85

Invalid Pointers 86

Variables as Pointers 87

Pointer Initialization 88

Note: Program Sections 89

More Pseudo Notation 90

"MemoryPointers" Project: Memory Layout 91

Chapter 7: Logical Instructions and EIP 99

Instruction Format 99

Logical Shift Instructions 100

Logical Operations 101

Zeroing Memory or Registers 102

Instruction Pointer 103

Note: Code Section 104

Chapter 8: Reconstructing a Program with Pointers 107

Example of Disassembly Output – No Optimization 107

Reconstructing C/C++ Code – Part 1 110

Reconstructing C/C++ Code – Part 2 112

Reconstructing C/C++ Code – Part 3 113

Reconstructing C/C++ Code – C/C++ program 114

Example of Disassembly Output – Optimized Program 115

Chapter 9: Memory and Stacks 117

Stack: A Definition 117

Stack Implementation in Memory 118

Things to Remember 120

PUSH Instruction 121

POP instruction 122

Register Review 123

Application Memory Simplified 124

Stack Overflow 125

Jumps 127

Calls 129

Call Stack 131

Exploring Stack in WinDbg 133

Chapter 10: Frame Pointer and Local Variables 137

Stack Usage 137

Register Review 138

Addressing Array Elements 139

Stack Structure (No Function Parameters) 140

Raw Stack (No Local Variables and Function Parameters) 141

Function Prolog 142

Function Epilog 143

“Local Variables” Project 144

Disassembly of Optimized Executable (Release Configuration) 148

Advanced Topic: FPO 149

Chapter 11: Function Parameters 151

“FunctionParameters” Project 151

Stack Structure 152

Stack Structure with FPO 154

Function Prolog and Epilog 156

Project Disassembled Code with Comments 157

Release Build with FPO Enabled ... 162

Cdecl Calling Convention ... 163

Parameter Mismatch Problem ... 164

Chapter 12: More Instructions ... 165

CPU Flags Register ... 165

The Fastest Way to Fill Memory ... 166

Testing for 0 ... 168

TEST - Logical Compare ... 169

CMP – Compare Two Operands ... 170

TEST or CMP? ... 171

Conditional Jumps ... 172

The Structure of Registers ... 173

Function Return Value ... 174

Using Byte Registers ... 175

Chapter 13: Function Pointer Parameters ... 177

"FunctionPointerParameters" Project ... 177

Commented Disassembly ... 178

Dynamic Addressing of Local Variables ... 181

Chapter 14: Summary of Code Disassembly Patterns ... 183

Function Prolog / Epilog ... 183

Passing Parameters ... 184

LEA (Load Effective Address) ... 185

Accessing Parameters and Local Variables ... 186

Index ... 187