Linux Debugging, Disassembling, Reversing

Practical Foundations: Training Course

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

Copyright © 2021 by Dmitry Vostokov

Copyright © 2021 by Software Diagnostics Services

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.

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

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

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

ISBN-13: 978-1-912636-34-1

Revision 1.01 (January 2021)

Summary of Contents

Contents....4

Preface....9

About the Author....10

Chapter x64.1: Memory, Registers, and Simple Arithmetic....11

Chapter x64.2: Code Optimization....27

Chapter x64.3: Number Representations....33

Chapter x64.4: Pointers....41

Chapter x64.5: Bytes, Words, Double, and Quad Words....61

Chapter x64.6: Pointers to Memory....67

Chapter x64.7: Logical Instructions and RIP....89

Chapter x64.8: Reconstructing a Program with Pointers....97

Chapter x64.9: Memory and Stacks....107

Chapter x64.10: Frame Pointer and Local Variables....127

Chapter x64.11: Function Parameters....139

Chapter x64.12: More Instructions....149

Chapter x64.13: Function Pointer Parameters....161

Chapter x64.14: Summary of Code Disassembly Patterns....169

Contents

Contents....4

Preface....9

About the Author....10

Chapter x64.1: Memory, Registers, and Simple Arithmetic....11

Memory and Registers inside an Idealized Computer....11

Memory and Registers inside Intel 64-bit PC....12

"Arithmetic" Project: Memory Layout and Registers....13

"Arithmetic" Project: A Computer Program....14

"Arithmetic" Project: Assigning Numbers to Memory Locations....15

Assigning Numbers to Registers....17

"Arithmetic" Project: Adding Numbers to Memory Cells....18

Incrementing/Decrementing Numbers in Memory and Registers....21

Multiplying Numbers....24

Chapter x64.2: Code Optimization....27

"Arithmetic" Project: C/C++ Program....27

Downloading GDB....28

GDB Disassembly Output – No Optimization....29

GDB Disassembly Output – Optimization....32

Chapter x64.3: Number Representations....33

Numbers and Their Representations....33

Decimal Representation (Base Ten)....34

Ternary Representation (Base Three)....35

Binary Representation (Base Two)....36

Hexadecimal Representation (Base Sixteen)....37

Why are Hexadecimals used? 38

Chapter x64.4: Pointers 41

A Definition 41

"Pointers" Project: Memory Layout and Registers 42

"Pointers" Project: Calculations 43

Using Pointers to Assign Numbers to Memory Cells 44

Adding Numbers Using Pointers 50

Incrementing Numbers Using Pointers 53

Multiplying Numbers Using Pointers 56

Chapter x64.5: Bytes, Words, Double, and Quad Words 61

Using Hexadecimal Numbers 61

Byte Granularity 62

Bit Granularity 63

Memory Layout 64

Chapter x64.6: Pointers to Memory 67

Pointers Revisited 67

Addressing Types 68

Registers Revisited 73

NULL Pointers 74

Invalid Pointers 75

Variables as Pointers 76

Pointer Initialization 77

Initialized and Uninitialized Data 78

More Pseudo Notation 79

"MemoryPointers" Project: Memory Layout 80

Chapter x64.7: Logical Instructions and RIP 89

Instruction Format 89

Logical Shift Instructions 90

Logical Operations 91

Zeroing Memory or Registers 92

Instruction Pointer 93

Code Section 95

Chapter x64.8: Reconstructing a Program with Pointers 97

Example of Disassembly Output: No Optimization 97

Reconstructing C/C++ Code: Part 1 99

Reconstructing C/C++ Code: Part 2 101

Reconstructing C/C++ Code: Part 3 103

Reconstructing C/C++ Code: C/C++ program 104

Example of Disassembly Output: Optimized Program 105

Chapter x64.9: Memory and Stacks 107

Stack: A Definition 107

Stack Implementation in Memory 108

Things to Remember 110

PUSH Instruction 111

POP instruction 112

Register Review 113

Application Memory Simplified 115

Stack Overflow 116

Jumps 117

Calls 119

Call Stack 121

Exploring Stack in GDB 123

Chapter x64.10: Frame Pointer and Local Variables 127

Stack Usage 127

Register Review 128

Addressing Array Elements 129

Stack Structure (No Function Parameters) 130

Function Prolog 131

Raw Stack (No Local Variables and Function Parameters) 132

Function Epilog 134

"Local Variables" Project 135

Disassembly of Optimized Executable 138

Chapter x64.11: Function Parameters 139

"FunctionParameters" Project 139

Stack Structure 140

Function Prolog and Epilog 142

Project Disassembled Code with Comments 144

Parameter Mismatch Problem 147

Chapter x64.12: More Instructions 149

CPU Flags Register 149

The Fast Way to Fill Memory 150

Testing for 0 152

TEST - Logical Compare 153

CMP – Compare Two Operands 154

TEST or CMP? 155

Conditional Jumps 156

The Structure of Registers 157

Function Return Value 158

Using Byte Registers 159

Chapter x64.13: Function Pointer Parameters 161

“FunctionPointerParameters” Project 161

Commented Disassembly 162

Chapter x64.14: Summary of Code Disassembly Patterns 169

Function Prolog / Epilog 169

LEA (Load Effective Address) 171

Passing Parameters 172

Accessing Parameters and Local Variables 173