

ARM64 Linux Debugging, Disassembling, Reversing

Practical Foundations: Training Course

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

Copyright © 2022 by Dmitry Vostokov

Copyright © 2022 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: 978-1-912636-37-2

Revision 1.00 (January 2022)

Summary of Contents

Contents.....	4
Preface.....	9
About the Author	10
Chapter A64.1: Memory, Registers, and Simple Arithmetic.....	11
Chapter A64.2: Code Optimization.....	29
Chapter A64.3: Number Representations	39
Chapter A64.4: Pointers.....	47
Chapter A64.5: Bytes, Half Words, Words, and Double Words	69
Chapter A64.6: Pointers to Memory	75
Chapter A64.7: Logical Instructions and PC	99
Chapter A64.8: Reconstructing a Program with Pointers	107
Chapter A64.9: Memory and Stacks	119
Chapter A64.10: Frame Pointer and Local Variables.....	137
Chapter A64.11: Function Parameters	149
Chapter A64.12: More Instructions	159
Chapter A64.13: Function Pointer Parameters.....	167
Chapter A64.14: Summary of Code Disassembly Patterns.....	173

Contents

Contents.....	4
Preface.....	9
About the Author	10
Chapter A64.1: Memory, Registers, and Simple Arithmetic.....	11
Memory and Registers inside an Idealized Computer	11
Memory and Registers inside ARM 64-bit Computer.....	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	18
“Arithmetic” Project: Adding Numbers to Memory Cells.....	19
Incrementing/Decrementing Numbers in Memory and Registers.....	22
Multiplying Numbers.....	25
Chapter A64.2: Code Optimization.....	29
“Arithmetic” Project: C/C++ Program.....	29
Downloading GDB	31
GDB Disassembly Output – No Optimization.....	32
GDB Disassembly Output – Optimization.....	37
Chapter A64.3: Number Representations	39
Numbers and Their Representations.....	39
Decimal Representation (Base Ten).....	40
Ternary Representation (Base Three).....	41
Binary Representation (Base Two)	42
Hexadecimal Representation (Base Sixteen).....	43

Why are Hexadecimals used?	44
Chapter A64.4: Pointers.....	47
A Definition	47
“Pointers” Project: Memory Layout and Registers	48
“Pointers” Project: Calculations.....	50
Using Pointers to Assign Numbers to Memory Cells	51
Adding Numbers Using Pointers.....	58
Incrementing Numbers Using Pointers.....	62
Multiplying Numbers Using Pointers.....	65
Chapter A64.5: Bytes, Half Words, Words, and Double Words	69
Using Hexadecimal Numbers	69
Byte Granularity.....	70
Bit Granularity.....	71
Memory Layout.....	72
Chapter A64.6: Pointers to Memory	75
Pointers Revisited	75
Addressing Types	76
Registers Revisited	81
NULL Pointers.....	82
Invalid Pointers.....	83
Variables as Pointers	84
Pointer Initialization	85
Initialized and Uninitialized Data	86
More Pseudo Notation.....	87
“MemoryPointers” Project: Memory Layout.....	88
Chapter A64.7: Logical Instructions and PC	99

Instruction Format.....	99
Logical Shift Instructions	100
Logical Operations	101
Zeroing Memory or Registers.....	102
Instruction Pointer	103
Code Section	105
Chapter A64.8: 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	114
Reconstructing C/C++ Code: C/C++ program	116
Example of Disassembly Output: Optimized Program.....	117
Chapter A64.9: Memory and Stacks.....	119
Stack: A Definition.....	119
Stack Implementation in Memory	120
Things to Remember.....	122
Stack Push Implementation	123
Stack Pop Implementation	124
Register Review	125
Application Memory Simplified.....	126
Stack Overflow.....	127
Jumps.....	128
Calls.....	130
Call Stack.....	131
Exploring Stack in GDB	133

Chapter A64.10: Frame Pointer and Local Variables.....	137
Stack Usage	137
Register Review	138
Addressing Array Elements	139
Stack Structure (No Function Parameters)	140
Function Prolog.....	141
Raw Stack (No Local Variables and Function Parameters)	142
Function Epilog	144
“Local Variables” Project.....	145
Disassembly of Optimized Executable.....	148
Chapter A64.11: Function Parameters	149
“FunctionParameters” Project.....	149
Stack Structure	150
Function Prolog and Epilog.....	152
Project Disassembled Code with Comments.....	154
Parameter Mismatch Problem	158
Chapter A64.12: More Instructions	159
PSTATE Flags	159
Testing for 0.....	160
TST - Logical Compare	161
CMP – Compare Two Operands.....	162
TST or CMP?.....	163
Conditional Jumps.....	164
Function Return Value	165
Chapter A64.13: Function Pointer Parameters.....	167
“FunctionPointerParameters” Project.....	167

Commented Disassembly.....	168
Chapter A64.14: Summary of Code Disassembly Patterns.....	173
Function Prolog / Epilog	173
ADR (Address).....	174
Passing Parameters	175
Accessing Saved Parameters and Local Variables	176