

CONTENTS IN DETAIL

ACKNOWLEDGMENTS	xxi
------------------------	------------

INTRODUCTION	xxiii
---------------------	--------------

What Is a Debugger?	xxiv
What Will We Build?	xxiv
The Source Language	xxiv
Limitations	xxv
What's in This Book?	xxv
The Target Platform	xxvii
For Windows Users	xxvii
For macOS Users	xxvii

LIST OF ABBREVIATIONS	xxix
------------------------------	-------------

1	
PROJECT SETUP	1

The Directory Structure	1
Writing the Top-Level CMake File	2
Building the libstd Library	3
Building the std Executable	4
Making the Dependencies Accessible	5
Dependency Management with vcpkg	7
Testing	8
Summary	10

2	
COMPILATION AND COMPUTER ARCHITECTURE	11

Compilation	12
Encoding	13
Debug Information	13
Operating Systems and Debuggers	14
Program Loading	15
User Space vs. Kernel Space	16
Virtual Memory	17
Debug APIs	18
Signals	18

Computer Architecture	19
Registers	20
Program Counter	20
Assembly and Instruction Encodings	20
Endianness	21
Stack Frames	22
Summary	23
Check Your Knowledge	23

3 ATTACHING TO A PROCESS 25

Process Interaction	25
fork and exec	26
ptrace	26
Launching and Attaching to Processes	27
The main Function	27
The attach Function	28
Adding a User Interface	31
Handling User Input	33
Manual Testing	36
Refactoring into a Library	36
Creating a Process Type	37
Implementing launch and attach	40
Handling Errors	41
Destructing Processes	44
Resuming the Process	44
Waiting on Signals	45
Summary	49
Check Your Knowledge	49

4 PIPES, PROCFS, AND AUTOMATED TESTING 51

Test Cases	51
Testing Process Launching	52
Pipes for Inter-Process Communication	54
The Linux procfs	60
Testing Process Attaching	62
Testing Process Resuming	65
Summary	66
Check Your Knowledge	66

5		
REGISTERS		67
x64 Registers	67	
General Purpose	68	
Floating Point and Vector	69	
Debug	70	
Interactions with ptrace	70	
Describing Registers	72	
X-Macros	74	
Register Tables	75	
Register Interactions	79	
Reading	83	
Writing	85	
Troubleshooting	90	
Summary	94	
Check Your Knowledge	94	

6		
TESTING REGISTERS WITH X64 ASSEMBLY		97
Why Test with Assembly?	97	
An x64 Assembly Primer	98	
Test Setup	99	
Issuing Syscalls	102	
printf	103	
General-Purpose Registers	104	
MMX	106	
SSE	106	
x87	107	
Register Reads	109	
Exposing Registers	112	
The Help Command	112	
Register Reading	113	
Register Writing	117	
Register Interactions	124	
Summary	125	
Check Your Knowledge	125	

7		
SOFTWARE BREAKPOINTS		127
Hardware vs. Software Breakpoints	128	
Implementing Software Breakpoints	128	
Representing Locations	129	
Creating Sites	130	

Managing Sites	132
Testing Site Management	140
Enabling Breakpoints	144
Disabling Breakpoints	146
Adding Breakpoints to the Debugger	146
Listing	147
Setting	148
Enabling, Disabling, and Deleting	149
Providing Help	149
Determining Where to Set Breakpoints	150
Position-Independent Executables	151
Breakpoint Tests	153
Continuing	156
Automated Testing	159
Summary	164
Check Your Knowledge	164

8 **MEMORY AND DISASSEMBLY** **165**

Memory Operations	166
Reading and Writing	166
Exposing Memory to the User	170
Testing Memory Operations	174
Disassembly	176
Summary	184
Check Your Knowledge	184

9 **HARDWARE BREAKPOINTS AND WATCHPOINTS** **185**

Debug Registers	186
Implementing Hardware Breakpoints	187
Tracking	187
Setting	192
Clearing	196
Testing Hardware Breakpoints	197
Watchpoints	202
Exposing Watchpoints to the User	206
Testing Watchpoints	210
Summary	211
Check Your Knowledge	211

10	SIGNALS AND SYSCALLS	213
Signal Handlers		213
Printing Stop Information		217
Tracking Debug Register Assignments		219
Tracking Watchpoint Values		221
Displaying Stop Information		223
Catchpoints		225
Tracing Syscalls		225
Exposing Syscall Catchpoints		233
Testing Syscall Catchpoints		239
Signal and Interrupt Internals		241
Summary		246
Check Your Knowledge		246

11	OBJECT FILES	247
What Is an ELF?		248
Sections and Segments		248
File Structure		250
ELF Header Parsing		250
Section Header Parsing		253
String Tables		255
Getting Section Names		256
Building a Section Map		256
Parsing the General String Table		258
File Addresses, File Offsets, and Virtual Addresses		259
Parsing the Symbol Table		265
Creating a Target Type		272
Auxiliary Vectors		273
Targets		274
Testing		280
Summary		281
Check Your Knowledge		281

12	DEBUG INFORMATION	283
An Introduction to DWARF		284
DWARF Sections		284
DIE Binary Encoding		287
Fetching a Constants File		289

Parsing Abbreviation Tables	290
Integer Encoding	291
Entry Structure	292
The DWARF Cursor	292
Extraction	296
Parsing Compile Unit Headers	298
Parsing DIEs	301
Implementing Form Skipping	306
Traversing the DIE Tree	309
Reading Attributes	313
Augmenting DIEs with Attribute Support	320
Optimizing Child Iterators	320
Extracting DIE Address Ranges	321
Checking Offsets and Supporting Range Lists	327
Augmenting the dwarf Type	329
Retrieving DIE Names	332
Indexing DIEs	333
Testing the Parser	335
Summary	339
Check Your Knowledge	339

13

LINE TABLES 341

Line Table Contents	342
Interpreting the Line Table Program	342
The Program Header	342
The Abstract Machine	350
The Program Instructions	355
Retrieving Entries by Line or File Address	364
DIE Line Attribute Helpers	366
Testing the Interpreter	368
Summary	368
Check Your Knowledge	369

14

SOURCE-LEVEL BREAKPOINTS AND STEPPING 371

Function Inlining	372
Retrieving Inlined Function Stacks	376
Tracking Inlining	376
Finding the Current Stack	379
Source-Level Stepping	382
Step In	382
Step Over	387
Step Out	389
Exposing Stepping to the User	392

Source-Level Breakpoints	393
Expanding Breakpoint Sites	395
Determining the Breakpoint Site Type	396
Implementing Breakpoint Functions	399
Creating Breakpoint Subtypes	400
Creating Breakpoints	405
Exposing Breakpoints to the User	407
Printing Currently Executing Source Code	412
Testing	418
Breakpoints	418
Stepping	421
Summary	425
Check Your Knowledge	426

15

CALL FRAME INFORMATION 427

A Backtrace Example	428
DWARF Call Frame Information	429
Common Information Entries	430
Frame Description Entries	433
Parsing Common Information Entries	433
Parsing Frame Description Entries	439
Looking Up Frame Description Entries	441
Parsing the EH Frame Header	442
Searching the EH Frame Table	444
Adding the Parsers to the Debugger	447
Summary	448
Check Your Knowledge	448

16

STACK UNWINDING 449

Executing Call Frame Information	449
Representing Rules	451
Returning Registers	453
Writing a Stack Unwinder	455
Executing Instructions	457
Executing Register Rules	463
Unwinding the Stack	464
Creating a Stack Frame Type	465
Tracking and Manipulating Frames	465
Understanding the Unwinding Algorithm	467
Writing the Unwinding Function	471

Exposing Stack Unwinding to the User	475
Adding up and down Commands	475
Reading Registers from Other Frames	477
Printing the Backtrace	479
Testing	480
Summary	482
Check Your Knowledge	482

17

SHARED LIBRARIES **483**

Program Loading	484
Static Executables	486
Dynamic Executables	486
Loading Dependencies	487
The .dynamic Section	487
The Rendezvous Structure	489
Relocations	490
Global Offset Table	492
Relocation Records	493
Procedure Linkage Table	494
Tracing Shared Library Loading	495
Adding Breakpoint Hit Callbacks	496
Locating the Rendezvous Structure	497
Handling Multiple ELF Files	500
Reading the Loaded Library List	505
Testing	509
Summary	510
Check Your Knowledge	511

18

MULTITHREADING **513**

Threads on Linux	513
The pthreads Library	514
Race Conditions	515
ptrace and the procs	516
Tracing Threads	516
Trapping New Threads	517
Identifying Thread Creation	517
Representing Thread States	518
Supporting Multithreaded Processes	521
Multithreaded Signal Handling	528
Updating the Signal Handling Function	530
Stopping and Resuming Threads	532

Cleaning Up Exited Threads	535
Reporting Lifecycle Events	536
Handling the Signals	537
Tracing Threads in the Target	540
Exposing Threads to the User	545
Testing	549
Summary	551
Check Your Knowledge	551

19

DWARF EXPRESSIONS 553

A Taxonomy of DWARF Expressions	554
Single Location Descriptions	555
Simple Location Descriptions	555
Composite Location Descriptions	556
Executing Simple Location Descriptions	556
Defining Result Types	556
Evaluating Expressions	558
Performing Stack Operations	559
Finding the Current Simple Location Description	560
Handling Opcode Ranges	561
Handling Individual Opcodes	562
Performing Stack Operations	564
Executing Dereferencing Instructions	565
Pushing Specific Entities to the Stack	566
Performing Arithmetic, Bitwise, and Relational Operations	566
Executing Control Flow Instructions	569
Executing Non-Address Location Types	569
Handling Composite Locations	570
Executing Location Lists	571
Exposing Attributes to the User	574
Augmenting the Stack Unwinder	576
Reading Global Variables	580
Reading DWARF Expression Results	580
Indexing Global Variables	584
Adding a Variable Lookup Interface	586
Testing	587
Summary	590
Check Your Knowledge	590

20

VARIABLES AND TYPES 591

Type DIEs	592
Base Types	593
Array Types	593

Class and Union Types	594
Member Pointer Types	596
Storing Type Information	597
Visualizing Typed Data	602
Member Pointers	603
Pointers	604
Classes	605
Arrays	609
Base Types	611
Finding Local Variables	612
Resolving Indirect Names	615
Exposing Variables to the User	621
Testing	625
Summary	628
Check Your Knowledge	628

21

EXPRESSION EVALUATION

629

Supporting Expression Evaluation	630
Extending Name Lookup	631
Parsing Arguments	634
Representing Built-in Types	635
Parsing a Single Argument	637
Collecting All Arguments	638
Overload Resolution	640
Calling Conventions	644
Classifying Parameters	644
Passing Arguments	646
Returning Values	647
Calling Functions in the Debugger	647
Performing Low-Level Inferior Calls	647
Allocating Memory for Arguments	649
Classifying Parameters	650
Checking Call Triviality	657
Identifying Constructors and Destructors	658
Checking for Unaligned Fields	660
Implementing Argument Passing	661
Reading Return Values	665
Indexing Member Functions	667
Performing High-Level Inferior Calls	670
Evaluating Expressions	671
Exposing Expression Evaluation to the User	673
Testing the Evaluator	674
Summary	678
Check Your Knowledge	679

22		
ADVANCED TOPICS		681
Remote Debugging	681	
Development Tool Communication	682	
GDB Machine Interface	683	
Debug Adapter Protocol	684	
What You Should Use	685	
Thread-Local Storage	685	
Time-Travel Debugging	686	
Exception Catchpoints	687	
Non-Stop Mode	687	
Follow-Fork Mode	688	
Summary	688	
APPENDIX: CHECK YOUR KNOWLEDGE ANSWERS		689
GLOSSARY		695
INDEX		701