Contents

Introduction **xxix**

Chapter 1 It's All in the Plan **1**
Another Pleasant Valley Saturday 1
Steps and Tests 3
More Than Two Ways? 3
Computers Think Like Us 4
Had This Been the Real Thing . . . 5
Assembly Language Programming As a Square Dance 5
Assembly Language Programming As a Board Game 6
Code and Data 8
Addresses 8
Metaphor Check! 9

Chapter 2 Alien Bases **11**
The Return of the New Math Monster 11
Counting in Martian 12
Dissecting a Martian Number 14
The Essence of a Number Base 16
Octal: How the Grinch Stole Eight and Nine 16
Who Stole Eight and Nine? 17
Hexadecimal: Solving the Digit Shortage 20
From Hex to Decimal and from Decimal to Hex 24
From Hex to Decimal 24
From Decimal to Hex 25
Practice. Practice! PRACTICE! 27
Arithmetic in Hex 28

Columns and Carries 30
Subtraction and Borrows 31
Borrows Across Multiple Columns 33
What's the Point? 33
Binary 34
Values in Binary 36
Why Binary? 38
Hexadecimal as Shorthand for Binary 38
Prepare to Compute 40

Chapter 3 Lifting the Hood 41
RAXie, We Hardly Knew Ye 41
Gus to the Rescue 42
Switches, Transistors, and Memory 43
One If by Land... 43
Transistor Switches 44
The Incredible Shrinking Bit 46
Random Access 47
Memory Access Time 49
Bytes, Words, Double Words, and Quad Words 50
Pretty Chips All in a Row 51
The Shop Supervisor and the Assembly Line 54
Talking to Memory 55
Riding the Data Bus 56
The Shop Supervisor's Pockets 57
The Assembly Line 58
The Box That Follows a Plan 58
Fetch and Execute 60
The Supervisor's Innards 61
Changing Course 62
What vs. How: Architecture and Microarchitecture 63
Evolving Architectures 64
The Secret Machinery in the Basement 65
Enter the Plant Manager 67
Operating Systems: The Corner Office 67
BIOS: Software, Just Not as Soft 68
Multitasking Magic 68
Promotion to Kernel 70
The Core Explosion 70
The Plan 72

Chapter 4 Location, Location, Location 73
The Joy of Memory Models 73
16 Bits'll Buy You 64 KB 75

The Nature of a Megabyte 78
Backward Compatibility and Virtual 86 Mode 79
16-Bit Blinders 79
The Nature of Segments 80
A Horizon, Not a Place 84
Making 20-Bit Addresses Out of 16-Bit Registers 84
Segment Registers 87
Segment Registers and x64 88
General-Purpose Registers 88
Register Halves 91
The Instruction Pointer 92
The Flags Register 94
Math Coprocessors and Their Registers 94
The Four Major Assembly Programming Models 95
Real-Mode Flat Model 95
Real-Mode Segmented Model 97
32-Bit Protected Mode Flat Model 99
64-Bit Long Mode 101

Chapter 5 The Right to Assemble 103
The Nine and Sixty Ways to Code 103
Files and What's Inside Them 104
Binary vs. Text Files 105
Looking at Binary File Internals with the GHex Hex Editor 106
Interpreting Raw Data 110
"Endianness" 111
Text In, Code Out 115
Assembly Language 116
Comments 118
Beware "Write-Only" Source Code! 119
Object Code, Linkers, and Libraries 120
Relocatability 123
The Assembly Language Development Process 123
The Discipline of Working Directories 125
Editing the Source Code File 126
Assembling the Source Code File 126
Assembler Errors 127
Back to the Editor 128
Assembler Warnings 129
Linking the Object Code File 130
Linker Errors 131
Testing the EXE File 131
Errors vs. Bugs 132

Are We There Yet? 133
Debuggers and Debugging 133
Taking a Trip Down Assembly Lane 134
Installing the Software 134
Step 1: Edit the Program in an Editor 137
Step 2: Assemble the Program with NASM 138
Step 3: Link the Program with ld 140
Step 4: Test the Executable File 141
Step 5: Watch It Run in the Debugger 141

Chapter 6 A Place to Stand, with Access to Tools 143
Integrated Development Environments 143
Introducing SASM 146
Configuring SASM 146
SASM's Fonts 147
Using a Compiler to Link 148
A Quick Tour of SASM 149
SASM's Editor 152
What SASM Demands of Your Code 152
Linux and Terminals 153
The Linux Console 153
Character Encoding in Konsole 154
The Three Standard Unix Files 156
I/O Redirection 158
Simple Text Filters 159
Using Standard Input and Standard Output from Inside SASM 161
Terminal Control with Escape Sequences 161
So Why Not GUI Apps? 163
Using Linux Make 164
Dependencies 165
When a File Is Up-to-Date 167
Chains of Dependencies 167
Invoking Make 169
Creating a Custom Key Binding for Make 170
Using Touch to Force a Build 172
Debugging with SASM 172
Pick up Your Tools. . . 174

Chapter 7 Following Your Instructions 175
Build Yourself a Sandbox 176
A Minimal NASM Program for SASM 176
Instructions and Their Operands 178
Source and Destination Operands 178

Immediate Data 179
Register Data 181
Memory Data and Effective Addresses 184
Confusing Data and Its Address 185
The Size of Memory Data 185
The Bad Old Days 186
Rally Round the Flags, Boys! 186
Flag Etiquette 190
Watching Flags from SASM 190
Adding and Subtracting One with INC and DEC 191
How Flags Change Program Execution 192
How to Inspect Variables in SASM 194
Signed and Unsigned Values 195
Two's Complement and NEG 196
Sign Extension and MOVSX 198
Implicit Operands and MUL 200
MUL and the Carry Flag 202
Unsigned Division with DIV 203
MUL and DIV Are Slowpokes 204
Reading and Using an Assembly Language Reference 205
Memory Joggers for Complex Memories 205
An Assembly Language Reference for Beginners 206
Flags 207
NEG Negate (Two's Complement; i.e., Multiply by −1) 208
Flags Affected 208
Legal Forms 208
Examples 208
Notes 208
Legal Forms 209
Operand Symbols 209
Examples 210
Notes 210
What's Not Here. . . 210

Chapter 8 Our Object All Sublime 213
The Bones of an Assembly Language Program 213
The Initial Comment Block 215
The .data Section 216
The .bss Section 216
The .text Section 217
Labels 217
Variables for Initialized Data 218
String Variables 219

Deriving String Length with EQU and $ 221
Last In, First Out via the Stack 223
Five Hundred Plates an Hour 223
Stacking Things Upside Down 225
Push-y Instructions 226
POP Goes the Opcode 227
PUSHA and POPA Are Gone 228
Pushing and Popping in Detail 229
Storage for the Short Term 231
Using Linux Kernel Services Through Syscall 231
X64 Kernel Services via the SYSCALL Instruction 232
ABI vs. API? 232
The ABI's Register Parameter Scheme 233
Exiting a Program via SYSCALL 234
Which Registers Are Trashed by SysCall? 235
Designing a Nontrivial Program 235
Defining the Problem 235
Starting with Pseudocode 236
Successive Refinement 237
Those Inevitable "Whoops!" Moments 241
Scanning a Buffer 242
"Off by One" Errors 244
From Pseudocode to Assembly Code 246
The SASM Output Window Gotcha 248
Going Further 248

Chapter 9 Bits, Flags, Branches, and Tables 251
Bits Is Bits (and Bytes Is Bits) 251
Bit Numbering 252
"It's the Logical Thing to Do, Jim. . ." 252
The AND Instruction 253
Masking Out Bits 254
The OR Instruction 255
The XOR Instruction 256
The NOT Instruction 257
Segment Registers Don't Respond to Logic! 258
Shifting Bits 258
Shift by What? 258
How Bit Shifting Works 259
Bumping Bits into the Carry Flag 260
The Rotate Instructions 260
Rotating Bits Through the Carry Flag 261
Setting a Known Value into the Carry Flag 262

Bit-Bashing in Action 262
Splitting a Byte into Two Nybbles 264
Shifting the High Nybble into the Low Nybble 265
Using a Lookup Table 266
Multiplying by Shifting and Adding 267
Flags, Tests, and Branches 270
Unconditional Jumps 271
Conditional Jumps 271
Jumping on the Absence of a Condition 272
Flags 273
Comparisons with CMP 274
A Jungle of Jump Instructions 275
“Greater Than” Versus “Above” 275
Looking for 1-Bits with TEST 277
Looking for 0-Bits with BT 279
X64 Long Mode Memory Addressing in Detail 279
Effective Address Calculations 281
Displacements 282
The x64 Displacement Size Problem 283
Base Addressing 283
Base + Displacement Addressing 283
Base + Index Addressing 284
Index X Scale + Displacement Addressing 285
Other Addressing Schemes 287
LEA: The Top-Secret Math Machine 289
Character Table Translation 290
Translation Tables 291
Translating with MOV or with XLAT 293
Tables Instead of Calculations 298

Chapter 10 Dividing and Conquering 299
Boxes within Boxes 300
Procedures as Boxes for Code 301
Calling and Returning 309
Calls Within Calls 311
The Dangers of Accidental Recursion 313
A Flag Etiquette Bug to Beware Of 314
Procedures and the Data They Need 315
Saving the Caller’s Registers 316
Preserving Registers Across Linux System Calls 317
PUSHAD and POPAD Are Gone 319
Local Data 321
Placing Constant Data in Procedure Definitions 322

More Table Tricks 323
Local Labels and the Lengths of Jumps 325
"Forcing" Local Label Access 328
Short, Near, and Far Jumps 329
Building External Procedure Libraries 330
When Tools Reach Their Limits 330
Using Include Files in SASM 331
Where SASM's Include Files Must Be Stored 337
The Best Way to Create an Include File Library 338
Separate Assembly and Modules 339
Global and External Declarations 339
The Mechanics of Globals and Externals 342
Linking Libraries into Your Programs 351
The Dangers of Too Many Procedures and Too Many Libraries 352
The Art of Crafting Procedures 352
Maintainability and Reuse 353
Deciding What Should Be a Procedure 354
Use Comment Headers! 355
Simple Cursor Control in the Linux Console 356
Console Control Cautions 363
Creating and Using Macros 364
The Mechanics of Macro Definition 366
Defining Macros with Parameters 371
The Mechanics of Invoking Macros 372
Local labels Within Macros 373
Macro Libraries as Include Files 374
Macros vs. Procedures: Pros and Cons 375

Chapter 11 Strings and Things 377
The Notion of an Assembly Language String 378
Turning Your "String Sense" Inside-Out 378
Source Strings and Destination Strings 379
A Text Display Virtual Screen 379
REP STOSB, the Software Machine Gun 387
Machine-Gunning the Virtual Display 388
Executing the STOSB Instruction 389
STOSB and the Direction Flag DF 390
Defining Lines in the Display Buffer 391
Sending the Buffer to the Linux Console 391
The Semiautomatic Weapon: STOSB Without REP 392
Who Decrements RCX? 392
The LOOP Instructions 393
Displaying a Ruler on the Screen 394
MUL Is Not IMUL 395

Ruler's Lessons 396
The Four Sizes of STOS 396
Goodbye, BCD Math 397
MOVSB: Fast Block Copies 397
DF and Overlapping Block Moves 398
Single-Stepping REP String Instructions 401
Storing Data to Discontinuous Strings 402
Displaying an ASCII Table 402
Nested Instruction Loops 404
Jumping When RCX Goes to 0 405
Closing the Inner Loop 406
Closing the Outer Loop 407
Showchar Recap 408
Command-Line Arguments, String Searches, and the Linux Stack 408
Displaying Command-Line Arguments from SASM 408
String Searches with SCASB 411
REPNE vs. REPE 413
You Can't Pass Command-Line Arguments to Programs Within SASM 413
The Stack, Its Structure, and How to Use It 414
Accessing the Stack Directly 417
Program Prologs and Epilogs 419
Addressing Data on the Stack 420
Don't Pop! 422

Chapter 12 Heading Out to C 423
What's GNU? 424
The Swiss Army Compiler 425
Building Code the GNU Way 425
SASM Uses GCC 427
How to Use gcc in Assembly Work 427
Why Not gas? 428
Linking to the Standard C Library 429
C Calling Conventions 431
Callers, Callees, and Clobbers 431
Setting Up a Stack Frame 433
Destroying a Stack Frame in the Epilog 434
Stack Alignment 435
Characters Out Via puts() 437
Formatted Text Output with printf() 438
Passing Parameters to printf() 440
Printf() Needs a Preceding 0 in RAX 442
You Shall Have –No-Pie 442
Data In with fgets() and scanf() 442

Using scanf() for Entry of Numeric Values 445
Be a Linux Time Lord 448
The C Library's Time Machine 449
Fetching time_t Values from the System Clock 451
Converting a time_t Value to a Formatted String 451
Generating Separate Local Time Values 452
Making a Copy of glibc's tm Struct with MOVSD 453
Understanding AT&T Instruction Mnemonics 456
AT&T Mnemonic Conventions 457
AT&T Memory Reference Syntax 459
Generating Random Numbers 460
Seeding the Generator with srand() 461
Generating Pseudorandom Numbers 461
Some Bits Are More Random Than Others 467
Calls to Addresses in Registers 469
Using puts() to Send a Naked Linefeed to the Console 470
How to Pass a libc Function More Than Six Parameters 470
How C Sees Command-Line Arguments 472
Simple File I/O 474
Converting Strings into Numbers with sscanf() 475
Creating and Opening Files 477
Reading Text from Files with fgets() 478
Writing Text to Files with fprintf() 481
Notes on Gathering Your Procedures into Libraries 482

Conclusion: Not the End, But Only the Beginning 489

Appendix A The Return of the Insight Debugger 493
Insight's Shortcomings 494
Opening a Program Under Insight 495
Setting Command-Line Arguments with Insight 496
Running and Stepping a Program 496
The Memory Window 497
Showing the Stack in Insight's Memory View 498
Examining the Stack with Insight's Memory View 498
Learn gdb! 500

Appendix B Partial x64 Instruction Reference 501
What's Been Removed from x64 502
Flag Results 502
Size Specifiers 503
Instruction Index 505
ADC: Arithmetic Addition with Carry 507
Flags Affected 507

Legal Forms	507
Examples	507
Notes	507
ADD: Arithmetic Addition	509
Flags Affected	509
Legal Forms	509
Examples	509
Notes	509
AND: Logical AND	511
Flags Affected	511
Legal Forms	511
Examples	511
Notes	511
BT: Bit Test	513
Flags Affected	513
Legal Forms	513
Examples	513
Notes	513
CALL: Call Procedure	515
Flags Affected	515
Legal Forms	515
Examples	515
Notes	515
CLC: Clear Carry Flag (CF)	517
Flags Affected	517
Legal Forms	517
Examples	517
Notes	517
CLD: Clear Direction Flag (DF)	518
Flags Affected	518
Legal Forms	518
Examples	518
Notes	518
CMP: Arithmetic Comparison	519
Flags Affected	519
Legal Forms	519
Examples	519
Notes	519
DEC: Decrement Operand	521
Flags Affected	521
Legal Forms	521
Examples	521
Notes	521

DIV: Unsigned Integer Division 522
Flags Affected 522
Legal Forms 522
Examples 522
Notes 522
INC: Increment Operand 524
Flags Affected 524
Legal Forms 524
Examples 524
Notes 524
J??: Jump If Condition Is Met 525
Flags Affected 525
Examples 525
Notes 525
JECXZ: Jump if ECX=0 527
Flags Affected 527
Legal Forms 527
Examples 527
Notes 527
JRCXZ: Jump If RCX=0 528
Flags Affected 528
Legal Forms 528
Examples 528
Notes 528
JMP: Unconditional Jump 529
Flags Affected 529
Legal Forms 529
Examples 529
Notes 529
LEA: Load Effective Address 531
Flags Affected 531
Legal Forms 531
Examples 531
Notes 531
LOOP: Loop Until CX/ECX/RCX=0 532
Flags Affected 532
Legal Forms 532
Examples 532
Notes 532
LOOPNZ/LOOPNE: Loop Until CX/ECX/RCX=0 and ZF=0 534
Flags Affected 534
Legal Forms 534

Examples 534
Notes 534
LOOPZ/LOOPE: Loop Until CX/ECX/RCX=0 and ZF=1 535
Flags Affected 535
Legal Forms 535
Examples 535
Notes 535
MOV: Copy Right Operand into Left Operand 536
Flags Affected 536
Legal Forms 536
Examples 536
Notes 536
MOVS: Move String 538
Flags Affected 538
Legal Forms 538
Examples 538
Notes 538
MOVSX: Copy with Sign Extension 540
Flags Affected 540
Legal Forms 540
Examples 540
Notes 540
MUL: Unsigned Integer Multiplication 542
Flags Affected 542
Legal Forms 542
Examples 542
Notes 542
NEG: Negate (Two's Complement; i.e., Multiply by –1) 544
Flags Affected 544
Legal Forms 544
Examples 544
Notes 544
NOP: No Operation 546
Flags Affected 546
Legal Forms 546
Examples 546
Notes 546
NOT: Logical NOT (One's Complement) 547
Flags Affected 547
Legal Forms 547
Examples 547
Notes 547

OR: Logical OR 548
Flags Affected 548
Legal Forms 548
Examples 548
Notes 548
POP: Copy Top of Stack into Operand 550
Flags Affected 550
Legal Forms 550
Examples 550
Notes 550
POPF/D/Q: Copy Top of Stack into Flags Register 552
Flags Affected 552
Legal Forms 552
Examples 552
Notes 552
PUSH: Push Operand onto Top of Stack 553
Flags Affected 553
Legal Forms 553
Examples 553
Notes 553
PUSHF/D/Q: Push Flags Onto the Stack 555
Flags Affected 555
Legal Forms 555
Examples 555
Notes 555
RET: Return from Procedure 556
Flags Affected 556
Legal Forms 556
Examples 556
Notes 556
ROL/ROR: Rotate Left/Rotate Right 558
Flags Affected 558
Legal Forms 558
Examples 558
Notes 558
SBB: Arithmetic Subtraction with Borrow 560
Flags Affected 560
Legal Forms 560
Examples 560
Notes 560
SHL/SHR: Shift Left/Shift Right 562
Flags Affected 562
Legal Forms 562
Examples 562

Notes 562

STC: Set Carry Flag (CF) 564

Flags Affected 564

Legal Forms 564

Examples 564

Notes 564

STD: Set Direction Flag (DF) 565

Flags Affected 565

Legal Forms 565

Examples 565

Notes 565

STOS/B/W/D/Q: Store String 566

Flags Affected 566

Legal Forms 566

Examples 566

Notes 566

SUB: Arithmetic Subtraction 568

Flags Affected 568

Legal Forms 568

Examples 568

Notes 569

SYSCALL: Fast System Call into Linux 570

Flags Affected 570

Legal Forms 570

Examples 570

Notes 570

XCHG: Exchange Operands 571

Flags Affected 571

Legal Forms 571

Examples 571

Notes 571

XLAT: Translate Byte Via Table 572

Flags Affected 572

Legal Forms 572

Examples 572

Notes 572

XOR: Exclusive OR 573

Flags Affected 573

Legal Forms 573

Examples 573

Notes 573

Appendix C Character Set Charts 575

Index 579

Duntemann155248_ftoc.indd 28 8/28/2023 3:30:01 PM