INDEX

A

ABI (application binary interface), 68
addend, 493
address space layout randomization (ASLR), 151, 486
    disabling, 152
`add_test_cpp_target` function, 154, 335
alignment, 204, 262, 571, 582, 664
all-stop mode, 514, 687
`an_innocent_function` function, 198
anonymous namespaces, 28
anti-debugging, 197
application binary interface (ABI), 68
ASLR. *See* address space layout randomization
assembly language
    6502, 20–22
    AT&T syntax, 98
    directives, 99
    immediates, 21
    Intel syntax, 98
    labels, 98
    mnemonic, 98
    opcode, 20, 98
    operand, 20, 98
    RIP-relative addressing, 104
    x64, 98–99
        `int3` instruction, 128, 144, 197
        `leaq` instruction, 104
        `movq` instruction, 98
async-signal-safe, 214
`attach` function, 28, 46, 276
augmentation data, 431, 436
auxiliary vector, 273, 486

B

backtrace, 427, 475–480
    commands, 479–480
    example, 428–429
    testing, 480–482
bitfields, 594, 607, 620, 656
breakpoints
    address, 401
    commands, 146–150, 407–412
    creating, 405
    deleting, 149
    disabling, 146, 149, 399–400
    enabling, 144–146, 149, 399–400
    function, 400
    hit handlers, 496
    line, 401, 404
    listing, 147
    resuming after hit, 156
    setting, 148–149
    source level, 393–412
    subtypes, 400–405
    testing, 153, 159–164, 418–425
breakpoint sites
    creating, 130
    linking to breakpoints, 395
    managing, 132
    testing, 140–143
buffered communication, 54
byte conversion, 83

C

callee-saved registers, 68
caller-saved registers, 68
call frame information (CFI), 427, 429–446, 449, 473
    executing, 449–463
    register restoration, 463
    rule representation, 451–453
calling conventions, 631, 644–647
canonical frame address (CFA), 429, 462, 554, 563, 566, 576
Catch2, 8, 53
catchpoints, 225, 687
    commands, 233–238
    testing, 239

cats, 16, 18, 22, 74, 187, 255, 285, 364, 365, 393, 414, 421, 428, 490, 594, 605, 629, 675
CFA. *See* canonical frame address
`cfa_expr_rule` type, 576
`cfa_register_rule` type, 451
CFI. *See* call frame information
CIEs. *See* common information entries
`classify_class_field` function, 655
`classify_class_type` function, 653
CMake
  configuration file, 6
  library namespaces, 4
  target exporting and installation, 5
  visibility, 4
`collect_arguments` function, 638
common information entries (CIEs), 430–432, 458
  parsing, 433–439
compilation, 12–14
  lossy, 12
compile unit, 285, 298
complement operator, 145
computer architecture, 19–23
converting syscall header information, 233
core dumps, 248
`create_loaded_elf` function, 275
CTest, 3
`cursor` type, 293
  `fixed_int` function, 293
  fixed-size integer functions, 294
  `skip_form` function, 306–308
  `sleb128` function, 295
  `string` function, 294
  `uleb128` function, 295

D

DAP (Debug Adapter Protocol), 683, 684
data alignment. *See* alignment
data spans, 166
Debug Adapter Protocol (DAP), 683, 684
dependency loading, 487
development tool communication, 682–685
/dev/null, 240
disassembly, 165, 176, 178, 388
discriminated union, 81
`dlopen` function, 483
Docker, xxvii
`dwarfdump`, 285, 287
DWARF expressions, 553
  address location descriptions, 556
  arithmetic operations, 566–568
  bitwise operations, 566–568
  composite location descriptions, 555, 556, 570–571
  control flow operations, 569
  dereferencing operations, 565–566
  empty location descriptions, 556
  implicit location descriptions, 555, 569–570
  individual opcodes, 562–564
  location lists, 555, 571–574
  opcode ranges, 561
  reading results, 580–584
  register location descriptions, 555, 569–570
  relational operations, 566–568
  simple location descriptions, 555, 556, 560
  single location descriptions, 555
  stack operations, 559–560, 564–565
  taxonomy, 554–556
  testing, 587–590
dwarf.h file, 289
DWARF information, 14, 75, 283, 473
  32-bit vs. 64-bit, 284
  abbreviation tables, 288
    extraction, 296
    layout, 292
    parsing, 290–298
  compile unit headers, 298–301
  cursor, 292
  DWARF information entries (DIEs), 285, 287
    address ranges, 321
    attributes, 313–329
    encoding, 302
    forms, 287, 306–308
    indexing, 333–334, 667–670
    member functions, 668
    names, 332–333

parsing, 301
siblings, 320
tag, 286
testing, 335–338
traversing the tree, 309
types, 592–597
exposing attributes, 574–576
function inlining, 375
line table, 342
lookups, 364–366
testing, 368
line table program, 342
abstract machine, 350–354
extended opcodes, 359–360
header, 342–350
instructions, 355–364
special opcodes, 361
standard opcodes, 356–359, 364
lookups, 329–334, 403, 613
member functions, 633
range lists, 321–327
sections, 284
types, 591
array, 593–596
base, 593
bitfields, 594, 607
built-in, 635
class, 594
computing size, 598, 636
equality, 642
field alignment, 660–661
identifying constructors, 658–661
identifying destructors, 658–661
member functions, 667–670
member pointers, 596–597
union, 594
visualization, 602–612
version compatibility, 284
dyadic rationals, 112
dynamic executables, 486
dynamic linker, 101, 249, 486, 495
link map, 489
rendezvous structure, 487–489, 495
resolution, 497
dynamic section, 487, 499

E

EH frame header, 442, 457
`eh_frame_pointer_encoding_size` function, 444
EH frame pointers, 436, 441
eightbytes, 644
`Elf64_Ehdr` type, 251
`Elf64_Shdr` type, 253
`Elf64_Sym` type, 266
ELF file, 13, 247
collection, 500
header, 161, 250
layout, 250
lookups, 404
magic number, 251
program headers, 249
section headers, 249, 253
section map, 256
section name table, 256
sections, 248
layout in memory, 262
segments, 248, 484
string tables, 255
symbol table
parsing, 265–272
testing, 280
`encode_hardware_stoppoint_mode` function, 195
`encode_hardware_stoppoint_size` function, 195
endianness, 21
entry point, 161, 273, 430, 486
`execute_cfi_instruction` function, 458–463, 577–579
`execute_unwind_rules` function, 463, 579–580
`exit_with_perror` function, 58
expression evaluation, 602, 615, 671–673
algorithm, 630
argument allocation, 649–650
argument setup, 661–665
commands, 673
inferior calls
high-level, 670
low-level, 647–649
parsing arguments, 634–640

expression evaluation *(continued)*
return values, 665–667
scoping, 630–631
testing, 674–678
`expr_rule` type, 576

F

FDEs. *See* frame description entries
file address, 151, 259, 486
conversion to virtual address, 261–265
file offset, 151, 259, 440
`find_free_stoppoint_register` function, 196
floating-point representation, 93, 112
exponent, 93
mantissa, 93
`fmtlib` library, 114, 148
fold expressions, 600
follow-fork mode, 688
FPU (floating-point unit) stack, 107
frame base. *See* canonical frame address
frame description entries (FDEs), 430, 433–455
lookups, 441–446
parsing, 439–441
frame pointer, 390
function epilogue, 101, 373
function inlining, 372–381
problems for debuggers, 373
function prologue, 101, 373, 384

G

GDB Machine Interface (GDB/MI), 683
GDB remote serial protocol (GDB RSP), 682
general string table, 255
parsing, 258
`get_entry_point_offset` function, 161
`get_initial_variable_data` function, 618, 633
`get_load_address` function, 162
`get_next_id` function, 130, 203, 399
`get_process_status` function, 61
`get_section_load_bias` function, 160
`get_signal_stop_reason` function, 278, 417
`get_sigtrap_info` function, 223–224, 238, 548
get some sleep, 27
global offset table (GOT), 458, 492, 495
`g_register_infos` variable, 72, 79
`g_sdb_process` variable, 214
`g_syscall_name_map` variable, 235

H

`handle_breakpoint_command` function, 147–149, 191
`handle_breakpoint_list_command` function, 408
`handle_breakpoint_set_command` function, 409
`handle_breakpoint_toggle` function, 411
`handle_catchpoint_command` function, 236
`handle_command` function, 33, 46, 112, 113, 146, 159, 170, 206, 236, 278, 392, 407, 476, 477, 479, 546, 586, 673
`handle_disassemble_command` function, 180–181
`handle_memory_command` function, 170
`handle_memory_read_command` function, 170–172
`handle_memory_write_command` function, 172–173
`handle_register_command` function, 114, 477
`handle_register_read` function, 115–116, 478
`handle_register_write` function, 117
`handle_sigint` function, 214
`handle_stop` function, 180, 278, 412, 475
`handle_syscall_catchpoint_command` function, 237
`handle_thread_command` function, 546
`handle_variable_command` function, 586, 622
`handle_variable_locals_command` function, 622
`handle_variable_location_command` function, 623
`handle_variable_read_command` function, 623, 634
`handle_watchpoint_command` function, 207
`handle_watchpoint_list` function, 208
`handle_watchpoint_set` function, 208

hardware breakpoints, 128, 185
clearing, 196–197
commands, 206–210
setting, 192–196
testing, 197
tracking, 187–192
hardware interrupt, 19, 245
hardware registers, 20, 67, 453
AVX, 69, 78
debug, 70, 79, 88, 186
condition bits, 187
DR7 layout, 186
size bits, 187
tracking assignments, 219–221
describing, 72–79
general purpose, 68, 75, 104
MMX, 69, 78, 106
`orig_rax`, 230
reading, 83, 109, 113
from other stack frames, 477–479
SSE, 69, 78, 106, 664, 667
testing, 104–112
writing, 85, 117
x64 register naming scheme, 68
x87, 69, 77, 107–109
header guards, 3

I

indirect names, 615–621, 631–634
inferior, 26
`inferior_call_from_dwarf` function, 670
`inferior_malloc` function, 637, 649
initial directory structure, 1
inline function stacks, 376–381, 388, 473
inline namespaces, 286
instruction encoding, 20, 178
instruction pointer, 20, 123, 321
inter-process communication, 54
interrupt handler, 245
interrupt vector, 19
interrupt vector table, 128, 245
`is_copy_or_move_constructor` function, 659
`is_destructor` function, 658
`is_prefix` function, 34
Itanium ABI, 268, 599, 630, 687
iterators, 309
tags, 311

K

kernel, 16, 486

L

lambdas, 86
init-captures, 143
lazy binding, 494
`LEB128`, 291
parsing, 295–296
lexical scoping, 614
`libedit` library, 7, 32
linker. *See* dynamic linker
linking, 248
link map, 505
Linux kernel, 217, 241, 486, 513
Linux Standard Base (LSB), 429, 431, 442, 487
load bias, 262, 273
Lospinoso, Josh, xxv

M

machine code, 12, 98
machine interfaces, 682–685
macro stringification, 235
`main` function, 27, 31, 32, 48, 214, 277, 545
`main_loop` function, 48
malware, 197
member pointer types, 596–597
memory
commands, 170
pages, 262
reading, 166
testing, 174–176
writing, 166
memory alignment. *See* alignment
memory protection, 15
`merge_parameter_classes` function, 656
Meyers, Scott, 38, 87, 136
move semantics, 135
multithreading, 513
cleanup, 535
commands, 545–549
resuming threads, 532–535
stopping threads, 532–535
testing, 549–551
thread lifecycle events, 536, 544
tracing thread creation, 517–518
mutual exclusion, 516

N

named constructors, 227
name mangling, 268–269
native code, 12, 98
nondeterminism, 515
non-stop mode, 514, 687
non-trivial for the purposes of calls (NTFPOC), 645, 654, 657–658

O

objdump utility, 151
object file, 13, 247
`offset_rule` type, 451
opcode, 20, 98
operand, 20, 98
operating systems, 14–19
overload resolution, 640–644

P

`parse_abbrev_table` function, 297–298
`parse_argument` function, 637–638
`parse_call_frame_information` function, 447
`parse_cie` function, 435–437
`parse_compile_unit` function, 300
`parse_compile_units` function, 300–301
`parse_die` function, 304–305
`parse_eh_frame_pointer` function, 438
`parse_eh_frame_pointer_with_base` function, 437
`parse_eh_hdr` function, 443–444
`parse_fde` function, 439–441
`parse_line_table_file` function, 349
`parse_line_table` function, 346–349
parsing
  floats, 119
  integers, 118
  vectors, 120
`path_ends_in` function, 365
PIE (position-independent executable), 151, 486, 491
pineapple on pizza, 197, 239
pipes, 54
`pkgconfig`, 8
PLT (procedure linkage table), 104, 458, 488, 494
position-independent executable (PIE), 151, 486, 491
pragma `pack` directive, 660
preemption, 515
`print_backtrace` function, 479
`print_code_location` function, 475
`print_disassembly` function, 179
`print_help` function, 113, 149, 159, 173, 181, 191, 209, 238, 392, 476, 547, 587
`print_source` function, 413–416
`print_stop_reason` function, 47, 223, 279, 548
procedure linkage table (PLT), 104, 458, 488, 494
processes
  allow launching without debugging, 62–64
  attaching, 27, 40, 62
  identifier, 26
  inferior, 26
  launching, 27, 40
  multithreaded, 521–540
  resuming, 44, 65
    from breakpoint, 156
  terminating, 44
  testing, 52–54, 62, 65
  waiting on signals, 45–49
`process_exists` function, 53
process group
  setting, 216
procfs, 60–62, 152, 155, 162, 166, 199, 273, 516, 519
  execution state, 60
program counter, 20, 123, 321
program headers, 484
program loading, 15, 248, 484, 486, 487
program stack, 15, 22, 101, 390, 465
  with base pointers, 390
  tracking, 465–467
`pthreads` library, 514–515
`ptrace`, 18, 26, 70, 241
  `PTRACE_ATTACH`, 29
  `PTRACE_CONT`, 35
  `PTRACE_GETFPREGS`, 70, 88
  `PTRACE_GETREGS`, 70, 88
  `PTRACE_GETSIGINFO`, 217, 229
  `PTRACE_GET_SYSCALL_INFO`, 229

PTRACE_O_TRACECLONE, 516
PTRACE_O_TRACEEXEC, 688
PTRACE_O_TRACEFORK, 688
PTRACE_PEEKDATA, 144
PTRACE_PEEKUSER, 70, 88, 241
PTRACE_POKEDATA, 145, 168
PTRACE_POKEUSER, 70, 90
PTRACE_SETFPREGS, 70, 91
PTRACE_SETREGS, 70, 91
PTRACE_SINGLESTEP, 157
PTRACE_SYSCALL, 226
PTRACE_TRACEME, 30
pure virtual functions, 395, 658

R

race conditions, 515–516
RAII (resource acquisition is initialization), 56
raw string literals, 113
readelf utility, 154, 248, 493
read_return_value function, 666
reentrant, 214
refactoring into a library, 36–49
register. *See* hardware registers
register_id type, 79
register_rule type, 451
relocatable files, 248
relocation, 487, 488, 490
addend, 493
remote debugging, 681
resolve_overload function, 641
resource acquisition is initialization (RAII), 56
resume function, 35
rr tool, 687

S

same_rule type, 451
scopes, 614
scopes_at_address_in_die function, 614
sdb::abbrev type, 296
sdb::address_breakpoint type, 401
resolve function, 402
sdb::as_bytes function, 83
sdb::attr_spec type, 296
sdb::attr type, 315
as_address function, 316
as_block function, 317
as_evaluated_location function, 575
as_expression function, 574
as_int function, 316
as_location_list function, 574
as_range_list function, 326
as_reference function, 318–319
as_section_offset function, 316
as_string function, 319
as_type function, 598
sdb::breakpoint_site type, 129
constructor, 130, 188, 395
disable function, 146, 189
enable function, 144, 189
is_hardware function, 188
is_internal function, 188
sdb::breakpoint type, 393
at_address function, 398
breakpoint_sites function, 398
constructor, 399
disable function, 399
enable function, 399
in_range function, 398
sdb::builtin_type type, 635
sdb::byte64 type, 82
sdb::byte128 type, 82
sdb::call_frame_information::common_information_entry type, 433
sdb::call_frame_information::eh_hdr type, 445
sdb::call_frame_information::frame_description_entry type, 439
sdb::call_frame_information type, 433
constructor, 447
get_cie function, 434
unwind function, 456
sdb::children_range::iterator type, 310
sdb::compile_unit type, 298
abbrev_table function, 299
constructor, 346
lines function, 345
root function, 304
sdb::die::bitfield_information type, 607
sdb::die::children_range::iterator type
sdb::die::children_range type, 309
constructor, 312
equality operator, 312

sdb::die::children_range type *(continued)*
increment operator, 312, 320
post-fix increment operator, 313
sdb::die type, 303
children function, 313
contains_address function, 328
contains function, 314
file function, 366
get_bitfield_information function, 607
high_pc function, 321, 329
index operator, 314
line function, 366
location function, 366
low_pc function, 321, 328
parameter_types function, 640
sdb::disassembler type, 177
disassemble function, 178–179
sdb::dwarf_expression::address_result type, 556
sdb::dwarf_expression::data_result type, 556
sdb::dwarf_expression::empty_result type, 556
sdb::dwarf_expression::literal_result type, 556
sdb::dwarf_expression::pieces_result type, 556
sdb::dwarf_expression::register_result type, 556
sdb::dwarf_expression::result type, 556
sdb::dwarf_expression::simple_location type, 556
sdb::dwarf_expression type, 556
constructor, 558
eval function, 558
sdb::dwarf type, 289
cfi function, 447
compile_unit_containing_address function, 331
constructor, 300, 448
find_functions function, 331
find_global_variable function, 585
find_local_variable function, 614
function_containing_address function, 331
get_abbrev_table function, 290
index_die function, 333, 584, 669
index function, 332
line_entry_at_address function, 367
scopes_at_address function, 614
sdb::elf type, 251
build_section_map function, 257
build_symbol_maps function, 268
constructor, 252, 254, 257, 266, 268, 334
data_pointer_as_file_offset function, 440
destructor, 253
file_offset_as_data_pointer function, 440
get_dwarf function, 334
get_section_containing_address function, 263
get_section_contents function, 257
get_section function, 257
get_section_name function, 256
get_section_start_address function, 265
get_string function, 258
get_symbol_at_address function, 270
get_symbol_containing_address function, 271
get_symbols_by_name function, 270
load_bias function, 263
notify_loaded function, 263
parse_section_headers function, 254, 255
parse_symbol_table function, 266
symbol address comparator, 267
sdb::error type, 42
sdb::file_addr type, 259
operator overloads, 260
to_virt_addr function, 264
sdb::file_offset type, 261
sdb::from_bytes function, 83
sdb::function_breakpoint type, 400
resolve function, 403–404
sdb::line_breakpoint type, 401
resolve function, 404
sdb::line_table::entry type, 351
equality operator, 352
sdb::line_table::file type, 344
sdb::line_table::iterator type, 353
begin function, 354
constructor, 354
end function, 354

execute_instruction function, 357–364
increment operator, 355
post-increment operator, 356
sdb::line_table type, 344
constructor, 345
cu function, 345
file_names function, 345
get_entries_by_line function, 365
get_entry_by_address function, 364
sdb::location_list type, 572
eval function, 572
sdb::memcpy_bits function, 583
sdb::parameter_class type, 650
sdb::parse_vector function, 172
sdb::pipe type, 55
close_read function, 57
close_write function, 57
constructor, 56
destructor, 57
read function, 57
release_read function, 57
release_write function, 57
write function, 57
sdb::process_state type, 39
sdb::process type, 37, 39, 522
attach function, 40, 42, 63, 225
augment_stop_reason function, 218, 229–231, 526
breakpoint_sites function, 134
cleanup_exited_threads function, 535
clear_hardware_stoppoint function, 196, 527
constructor, 39, 519
create_breakpoint_site function, 134, 190, 396
create_watchpoint function, 206
current_thread function, 518
destructor, 44, 63
get_auxv function, 273
get_current_hardware_stoppoint function, 220, 526
get_pc function, 523
get_registers function, 87, 523
handle_signal function, 537
inferior_call function, 648
install_thread_lifecycle_callback function, 536
launch function, 40, 42, 54, 58, 62, 216, 225
maybe_resume_from_syscall function, 231–526
populate_existing_threads function, 519
read_all_registers function, 88, 524
read_memory_as function, 169
read_memory function, 167
read_memory_without_traps function, 182
read_string function, 605
report_thread_lifecycle_event function, 536
resume_all_threads function, 533
resume function, 44, 157, 227, 525, 533, 539
send_continue function, 533
set_current_thread function, 518
set_hardware_breakpoint function, 193
set_hardware_stoppoint function, 193–195, 527
set_pc function, 156, 523
set_syscall_catch_policy function, 227
set_target function, 376
set_watchpoint function, 205
should_resume_from_syscall function, 526
step_instruction function, 158, 525, 539
step_over_breakpoint function, 533
stop_running_threads function, 534
swallow_pending_sigstop function, 539
thread_states function, 518
wait_on_signal function, 45, 46, 89, 156, 219, 222, 232–532, 539
watchpoints function, 205
write_fprs function, 91, 524
write_gprs function, 91, 524
write_memory function, 168
write_user_area function, 88, 524
sdb::range_list::iterator type, 323
constructor, 324
increment operator, 325
post-increment operator, 326

sdb::range_list type, 322
    begin function, 327
    contains function, 327
    end function, 327
sdb::register_format type, 72
sdb::register_id type, 72
sdb::register_info_by_dwarf function, 79
sdb::register_info_by function, 79
sdb::register_info_by_id function, 79
sdb::register_info_by_name function, 79
sdb::register_info type, 72
sdb::registers type, 80, 82, 453
    constructor function, 519
    flush function, 454, 528
    is_undefined function, 454
    read function, 84, 454
    undefine function, 454
    write function, 85, 87, 90, 527
sdb::register_type type, 72
sdb::source_location type, 366
sdb::span type, 166
sdb::stack_frame type, 465
sdb::stack type, 379, 465
    constructor, 521
    create_base_frame function, 473
    create_inline_stack_frames function, 474
    frames function, 466
    get_pc function, 467
    inline_stack_at_pc function, 380
    regs function, 466
    reset_inline_height function, 380
    simulate_inlined_step_in function, 382
    tid function, 521
    unwind function, 471–473, 544
sdb::stoppoint_collection type, 132, 397
    contains_address function, 138
    contains_id function, 138
    find_by_address function, 137, 139
    find_by_id function, 137
    for_each function, 140
    get_by_address function, 139
    get_by_id function, 138
    get_in_region function, 183
    push function, 135
    remove_by_id function, 139
sdb::stoppoint_mode type, 192
sdb::stop_reason type, 45, 218, 228, 517
    constructor, 45, 385, 518
    is_breakpoint function, 385
    is_step function, 385
sdb::syscall_catch_policy type, 226
sdb::syscall_id_to_name function, 234
sdb::syscall_information type, 228
sdb::syscall_name_to_id function, 235
sdb::target::evaluate_expression_result type, 671
sdb::target::find_functions_result type, 402
sdb::target::resolve_indirect_name_result type, 631
sdb::target type, 274
    attach function, 276, 377
    breakpoints function, 406
    constructor, 521
    create_address_breakpoint function, 406
    create_function_breakpoint function, 406
    create_line_breakpoint function, 406
    evaluate_expression function, 672
    find_functions function, 403
    find_variable function, 615
    function_name_at_address function, 416
    get_pc_file_address function, 380, 540
    get_stack function, 381, 540
    inferior_malloc function, 649
    inline_stack_at_address function, 378
    inline_stack_at_pc function, 544
    launch function, 276, 377
    line_entry_at_pc function, 386, 540
    notify_stop function, 376, 381, 467, 544
    notify_thread_lifecycle_event function, 544
    read_location_data function, 581
    resolve_indirect_name function, 616, 632
    run_until_address function, 386, 541
    step_in function, 383–385, 542

step_out function, 391, 480, 541
step_over function, 388–543
threads function, 520
sdb::thread_state type, 518, 532
sdb::thread type, 520
sdb::to_byte64 function, 84
sdb::to_byte128 function, 84
sdb::to_byte_span function, 636
sdb::to_byte_vec function, 636
sdb::to_string_view function, 105
sdb::trap_type type, 218, 228, 517
sdb::typed_data type, 602
deref_pointer function, 619
fixup_bitfield function, 608
index function, 620
read_member function, 620
visualize function, 602
sdb::type type, 597, 635
alignment function, 660
byte_size function, 598
compute_byte_size function, 598, 636
equality operator, 642
get_builtin_type function, 635
get_die function, 635
get_expression_result function, 673
get_member_function_definition function, 669
get_parameter_classes function, 651
has_unaligned_fields function, 660
inequality operator, 642
is_char_type function, 601
is_class_type function, 654
is_from_dwarf function, 635
is_non_trivial_for_calls function, 657
is_reference_type function, 654
strip_all function, 601
strip_cvref_typedef function, 601
strip_cv_typedef function, 601
strip function, 600
sdb::virt_addr type, 259
to_file_addr function, 264
sdb::watchpoint type, 202
constructor, 203, 204, 221
data function, 221
disable function, 203
enable function, 203
previous_data function, 221
update_data function, 221
section hashing, 197
section load bias, 155
computing, 160
sed, 233
segment registers, 69
setpgid function, 216
set_ptrace_options function, 225, 517
setup_arguments function, 661–665
shared libraries, 483
tracing loads, 495
siginfo_t type, 217
signals, 18, 45, 48
handling, 19, 213–217, 277, 537–540
multithreading, 528–532
installing handler, 214
internals, 241
SIGILL, 156
SIGINT, 19, 213
SIGKILL, 44
SIGSEGV, 19
SIGSTOP, 44, 532, 538
SIGTERM, 19
sign extension, 92
smart pointers, 37
software breakpoints, 128
split function, 34
stack alignment, 664
stack frame. *See* program stack
stack pointer, 23
stack unwinding, 390, 427, 449, 455–457, 464–475
abstract machine, 452
commands, 475–480
DWARF expressions, 576–580
example, 467–471
implementation, 471–475
_start function, 161, 430
static executables, 486
std::array type, 81
std::byte type, 55
std::filesystem::path type, 276
std::map type
lower_bound function, 271
std::multimap type, 270

`std::optional` type, 100
`std::perror` function, 29
`std::runtime_error` type, 42
`std::strerror` function, 42
`std::string_view` type, 29
`std::unique_ptr` type, 38, 132
`std::unordered_multimap` type, 332
`std::variant` type, 81, 86, 221
stepping, 382–391, 529
  commands, 392–393
  step in, 382–387
  step out, 389–391
  step over, 387–389
  testing, 421
stop information, 278
  printing, 217, 223–225, 412
stop point, 132, 396–399
string tables, 255
  parsing, 258
structured bindings, 270
symbol, 492
symbol lookup, manual, 150
symbols, 101, 266
  weak, 267
synchronization, 516
system calls (syscalls), 16, 26, 53, 76, 486
  catch policy, 226
  `clock_gettime`, 486
  `clone`, 514
  `dlmopen`, 489
  `exec`, 26, 30, 688
  `execlp`, 30
  `fork`, 26, 30, 688
  `fstat`, 253
  `getcpu`, 486
  `gettime`, 486
  `gettimeofday`, 486
  internals, 241, 242
  `kill`, 18, 44, 53, 102
  `mmap`, 252, 253
  `open`, 253
  `personality`, 152
  `pipe`, 54
  `process_vm_readv`, 166, 168
  `process_vm_writev`, 166
  resuming after, 231
  testing, 236
  `tgkill`, 534
  tracing, 225–233
  translating names and IDs, 234
  `waitpid`, 31, 45
  `wrmsrl`, 242
System V ABI (SYSV ABI), 68, 102, 104, 107, 230, 247, 273, 429, 431, 485, 487, 630
  argument allocation, 646
  class merging, 645, 654, 656
  parameter classification, 644–646, 650–657
    array types, 653
    class fields, 654
    class types, 653
  post-merger cleanup, 646, 654
  return values, 647
  stack alignment, 664
  unaligned fields, 660–661

T

target, 274
  attaching, 276
  launching, 276
  line entry at program counter, 386
  multithreading, 540–545
  run until address, 386
target platform, xxvii
tasks, 513
template specialization, 121
test targets
  anti_debugger.cpp, 197, 200
  blocks.cpp, 612
  end_immediately.cpp, 65
  expr.cpp, 675
  global_variable.cpp, 587, 625
  hello_sdb.cpp, 150
  libmeow.cpp, 490
  marshmallow.cpp, 490
  member_pointer.cpp, 596
  memory.cpp, 174, 175
  multi_cu_main.cpp, 336
  multi_cu_other.cpp, 336
  multi_threaded.cpp, 514
  overloaded.cpp, 418
  reg_read.s, 109
  reg_write.s, 100
  run_endlessly.cpp, 64
  step.cpp, 421
text encoding, 13
thread group ID (TGID), 514

thread ID (TID), 514
thread_lifecycle_callback function, 545
thread-local storage, 69, 566, 685, 686
thread safety, xxv
thread states, representing, 518–521
TID (thread ID), 514
time-travel debugging, 686
TLS (thread local storage), 69, 566, 685, 686
top-down programming, 27
top-level *CMakeLists.txt*, 2
trailing return types, 137
trivial copyability, 645
Turing, Alan, 154
Turing-complete, 554
two's complement, 92, 291
type traits, 93
type visualization
    arrays, 609–611
    base types, 611–612
    classes, 605–609
    member pointers, 603
    pointers, 604–605

U

undefined_rule type, 451
UndoDB, 687
Unix vs. Linux vs. POSIX, 18
Unix pipes, 54–60, 99
unwind_context type, 452, 577
user area, 70, 87
user_fpregs_struct type, 71
user input, 33
user_regs_struct type, 70
user space, 16, 486
user type, 71
UTM, xxvii

V

val_expr_rule type, 576
val_offset_rule type, 451
varargs functions, 31, 104, 646, 665
variables, 591
    commands, 586–587, 621–625
    indexing globals, 584–586
    reading globals, 580–587
    reading locals, 612
    testing, 587–590, 625–628
variadic templates, 600
vcpkg, 7–8
    toolchain file, 7
vcpkg.json, 7
virtual address, converting to file address, 261–265
virtual dynamic shared object (vDSO), 486, 490, 507
virtual memory, 17, 121, 151, 259
visualize_array_type function, 609
visualize_base_type function, 611
visualize_class_type function, 606
visualize_member_pointer_type function, 603
visualize_pointer_type function, 604
visualize_subrange function, 610

W

wait_on_signal function, 35
waitpid function, 45
watchpoints, 185, 202
    testing, 210–211
    tracking data, 221–223
widen function, 93
WinDbg, 687
Windows Subsystem for Linux (WSL), xxvii

X

X-Macros, 74, 233

Y

Yama Linux Security Module, 36

Z

zero extension, 92
Zydis library, 176