



Memory Dump Analysis Anthology

Volumes

1 - 12

Tables of Contents and Indexes

Dmitry Vostokov
Software Diagnostics Institute

About the Author



Dmitry Vostokov is an internationally recognized expert, speaker, educator, scientist, and author. He is the founder of pattern-oriented software diagnostics, forensics and prognostics discipline and Software Diagnostics Institute (DA+TA: DumpAnalysis.org + TraceAnalysis.org). Vostokov has also authored more than 50 books on software diagnostics, anomaly detection and analysis, software and memory forensics, root cause analysis and problem solving, memory dump analysis, debugging, software trace and log analysis, reverse engineering and malware analysis. He has more than 25 years of experience in software architecture, design, development and maintenance in a variety of industries including leadership, technical and people management roles. Dmitry also founded Analog.io, BriteTrace, DiaThings, Logtellect, OpenTask Iterative and Incremental Publishing (OpenTask.com), Software Diagnostics Technology and Services (former Memory Dump Analysis Services) PatternDiagnostics.com and Software Prognostics. In his spare time, he presents various topics on Debugging.TV and explores Software Narratology, its further development as Narratology of Things and Diagnostics of Things (DoT), and Software Pathology. His current areas of interest are theoretical software diagnostics and its mathematical and computer science foundations, application of artificial intelligence and machine learning to diagnostics and anomaly detection, software diagnostics engineering and diagnostics-driven development, diagnostics workflow and interaction.

Memory Dump Analysis Anthology

Volume 1

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2008 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-0-9558328-0-2 (Paperback, B/W)

ISBN-13: 978-0-9558328-1-9 (Hardcover, B/W)

ISBN-13: 978-1-906717-01-8 (Hardcover, Color)

First printing, 2008

Revision 3.1 (April, 2015)

Contents

Preface	19
Acknowledgements.....	21
About the Author.....	23
PART 1: Crash Dumps for Beginners	25
Crash Dumps Depicted	25
Right Crash Dumps	26
Crashes Explained	28
Hangs Explained	31
Symbol Files Explained	34
Crashes and Hangs Differentiated.....	36
Proactive Crash Dumps	39
PART 2: Professional Crash Dump Analysis.....	43
Minidump Analysis.....	43
Scripts and WinDbg Commands.....	43
Component Identification	46
Raw Stack Data Analysis.....	53
Symbols and Images.....	63
Interrupts and Exceptions Explained.....	68
Exceptions Ab Initio.....	68
X86 Interrupts	69
X64 Interrupts	76

Interrupt Frames and Stack Reconstruction	83
Trap Command on x86	92
Trap Command on x64	100
Exceptions in User Mode	104
How to Distinguish Between 1st and 2nd Chances	109
Who Calls the Postmortem Debugger?	113
Inside Vista Error Reporting	117
Another Look at Page Faults	132
Bugchecks Depicted	135
NMI_HARDWARE_FAILURE	135
IRQL_NOT_LESS_OR_EQUAL	136
KERNEL_MODE_EXCEPTION_NOT_HANDLED	141
KMODE_EXCEPTION_NOT_HANDLED	143
SYSTEM_THREAD_EXCEPTION_NOT_HANDLED	144
CAFF	150
CF	152
Manual Stack Trace Reconstruction	157
WinDbg Tips and Tricks	167
Looking for Strings in a Dump	167
Tracing Win32 API While Debugging a Process	168
Exported NTDLL and Kernel Structures	170
Easy List Traversing	178
Suspending Threads	181

Heap Stack Traces	182
Hypertext Commands	183
Analyzing Hangs Faster	187
Triple Dereference	188
Finding a Needle in a Hay	191
Guessing Stack Trace	193
Coping with Missing Symbolic Information.....	199
Resolving Symbol Messages.....	204
The Search for Tags	206
Old Dumps, New Extensions	212
Object Names and Waiting Threads.....	214
Memory Dumps from Virtual Images.....	219
Filtering Processes.....	220
WinDbg Scripts	221
First Encounters	221
Yet another WinDbg Script.....	222
Deadlocks and Critical Sections.....	223
Security Problem	224
Hundreds of Crash Dumps	227
Parameterized Scripts	229
Security Issues and Scripts	230
Raw Stack Dump of All Threads (Process Dump)	231
Raw Stack Dump of All Threads (Complete Dump)	236

Case Study	241
Detecting Loops in Code	244
Crash Dump Analysis Checklist.....	251
Crash Dump Analysis Poster (HTML version)	254
PART 3: Crash Dump Analysis Patterns	255
Multiple Exceptions.....	255
Dynamic Memory Corruption	257
False Positive Dump	259
Lateral Damage	264
Optimized Code.....	265
Invalid Pointer	267
Inconsistent Dump	269
Hidden Exception	271
Deadlock (Critical Sections).....	276
Changed Environment.....	283
Incorrect Stack Trace.....	288
OMAP Code Optimization	294
No Component Symbols.....	298
Insufficient Memory (Committed Memory).....	302
Spiking Thread.....	305
Module Variety	310
Stack Overflow (Kernel).....	314
Deadlock (Executive Resources).....	323

Insufficient Memory (Handle Leak).....	327
Managed Code Exception	331
Truncated Dump	340
Waiting Thread Time.....	343
Deadlock (Mixed Objects)	348
Memory Leak (Process Heap).....	356
Missing Thread	362
Unknown Component	367
Memory Leak (.NET Heap)	371
Double Free (Process Heap)	378
Double Free (Kernel Pool)	387
Coincidental Symbolic Information	390
Stack Trace	395
Virtualized Process (WOW64).....	400
Stack Trace Collection	409
Coupled Processes	419
High Contention	421
Accidental Lock	423
Passive Thread (User Space)	430
Main Thread	437
Insufficient Memory (Kernel Pool)	441
Busy System	449
Historical Information	458

IRP Distribution Anomaly	459
Local Buffer Overflow.....	461
Passive System Thread (Kernel Space).....	462
Early Crash Dump	466
Hooked Functions	469
Custom Exception Handler.....	471
Deadlock (LPC)	474
Special Stack Trace	479
Manual Dump (Kernel).....	480
Wait Chain (General).....	482
Manual Dump (Process)	487
Wait Chain (Critical Sections)	490
PART 4: Crash Dump Analysis AntiPatterns	493
Alien Component	493
Zippocracy	494
Word of Mouth	495
Wrong Dump	496
Fooled by Description	497
Need the crash dump	498
Be Language	499
Fooled by Abbreviation	500
PART 5: A Bit of Science	501
Memory Dump - A Mathematical Definition	501

Threads as Braided Strings in Abstract Space	503
What is Memory Dump Analysis?	506
Memorillion and Quadrimemorillion	507
Four Causes of Crash Dumps.....	508
Complexity and Memory Dumps	510
What is a Software Defect?.....	511
PART 6: Fun with Crash Dumps.....	513
Dump Analysis and Voice Recognition.....	513
Sending SMS Messages via Dumps	514
WinDbg as a Big Calculator	515
Dumps, Debuggers, and Virtualization.....	516
Musical Dumps.....	518
Debugging the Debugger	519
Musical Dumps: Dump2Wave	521
Dump Tomography	522
The Smallest Program	523
Voices from Process Space	526
Crash Dump Analysis Card	528
Listening to Computer Memory.....	529
Visualizing Memory Dumps.....	532
Visualizing Memory Leaks	544
Picturing Computer Memory	556
Unicode Illuminated.....	559

Teaching Binary to Decimal Conversion.....	560
Crash Dumps and Global Conspiracy	561
PART 7: WinDbg For GDB Users and Vice Versa	563
AT&T and Intel Syntax.....	563
Installation	565
Disassembler	568
Stack Trace (Backtrace)	573
Local Variables	581
PART 8: Software Troubleshooting	589
Four Pillars.....	589
Five Golden Rules.....	590
Critical Thinking.....	591
Troubleshooting as Debugging.....	592
PART 9: Citrix.....	593
Pooltags.....	593
The List of Citrix Services.....	594
Reverse Engineering Citrix ThinWire	596
PART 10: Security	599
Memory Visualization	599
WinDbg is Privacy-Aware	600
Crash Dumps and Security	604
PART 11: The Origin of Crash Dumps	605
JIT Service Debugging.....	605

Local Crash Dumps in Vista	606
COM+ Crash Dumps	607
Correcting Microsoft Article about Userdump.exe	612
Where did the Crash Dump Come from?.....	616
Custom Postmortem Debuggers in Vista	618
Resurrecting Dr. Watson in Vista	621
Process Crash - Getting the Dump Manually	624
Upgrading Dr. Watson.....	627
Savedump.exe and Pagefile	628
Dumping Vista	629
Dumping Processes without Breaking Them.....	631
Userdump.exe on x64	632
NTSD on x64 Windows	633
Need a Dump? Common Use Cases	634
PART 12: Tools	635
Memory Dump Analysis Using Excel	635
TestDefaultDebugger.NET.....	636
Cons of Symbol Server	637
StressPrinters: Stressing Printer Autocreation	638
InstantDump (JIT Process Dumper).....	639
TestDefaultDebugger	641
DumpAlerts	643
DumpDepends	644

Dump Monitor Suite	645
SystemDump	646
PART 13: Miscellaneous	649
What is KiFastSystemCallRet?	649
Understanding I/O Completion Ports.....	653
Symbol File Warnings	656
Windows Service Crash Dumps in Vista	658
The Road to Kernel Space	664
Memory Dump Analysis Interview Questions.....	666
Music for Debugging	667
PDBFinder.....	668
When a Process Dies Silently	669
ASLR: Address Space Layout Randomization	674
Process and Thread Startup in Vista	679
Race Conditions on a Uniprocessor Machine.....	681
Yet Another Look at Zw* and Nt* Functions.....	684
Programmer Universalis.....	687
Dr. Watson Logs Analysis	688
Post-Debugging Complications	691
The Elements of Crash Dump Analysis Style	692
Crash Dump Analysis in Visual Studio	693
32-bit Stack from 64-bit Dump.....	695
Asmpedia.....	696

How WINE Can Help in Crash Dump Analysis	697
Horrors of Debugging Legacy Code	698
UML and Device Drivers	700
Statistics: 100% CPU Spread over all Processes	703
Appendix A.....	705
Crash Dump Analysis Portal	705
Appendix B	707
Reference Stack Traces	707
Index	709
Notes.....	715
Cover Images.....	720

Index

!analyze, 43, 44, 46, 49, 53, 73, 81, 93, 94, 144, 187, 224, 225, 226, 227, 251, 252, 255, 271, 288, 300, 302, 318, 331, 338, 387, 395, 401, 405, 423, 490, 496, 513, 519, 625, 656
!analyze -v, 43, 44, 46, 49, 53, 73, 81, 94, 144, 187, 224, 225, 226, 227, 251, 252, 255, 260, 271, 302, 318, 331, 338, 387, 395, 401, 405, 423, 490, 496, 513, 519, 625, 656
!analyze -v -hang, 251, 252, 490
!apc, 252
!chkimg, 251, 468
!dh, 208, 209, 251, 298, 369
!dpcs, 187, 252
!exchain, 251, 470, 471
!exqueue, 224, 225, 226, 252, 462
!exqueue f, 224, 225, 226, 252, 462
!for_each_process, 222, 223, 226
!for_each_thread, 236
!flag, 251, 358, 384, 386
!heap, 182, 253, 257, 258, 274, 288, 299, 302, 356, 357, 358, 359, 360, 361, 369, 371, 372, 373, 374, 375, 378, 379, 380, 381, 384, 386, 465, 488, 548, 549, 550, 554, 556, 638, 645, 666, 675, 676, 678
!irpfind, 224, 225, 226, 252, 457, 458, 635
!list, 43, 64, 65, 153, 178, 179, 184, 189, 190, 191, 220, 222, 230, 251, 252, 264, 269, 270, 276, 300, 310, 323, 324, 328, 336, 344, 359, 409, 414, 436, 446, 448, 457, 461, 501, 507, 565, 582, 630, 638, 645, 664, 670, 676
!locks, 187, 223, 224, 225, 226, 227, 251, 252, 269, 276, 278, 323, 324, 348, 349, 421, 423, 424, 426, 429, 447, 475, 491
!ipc, 252, 457, 474, 476
!ntsdexts.locks, 223, 225, 226, 252, 278, 475
!peb, 221, 251, 616, 676, 678
!pool, 46, 132, 206, 252, 258, 302, 303, 327, 328, 387, 388, 389, 430, 440, 441, 442, 443, 444, 446, 447, 499, 556
!poolused, 206, 208, 224, 225, 226, 252, 328, 440, 443, 444, 446, 593
!process, 84, 105, 214, 216, 220, 222, 225, 226, 236, 252, 270, 328, 409, 414, 415, 422, 461, 594, 703
!ready, 252, 343, 448, 451, 452
!runaway, 251, 305, 307, 470
!running, 28, 31, 32, 33, 36, 51, 113, 153, 222, 252, 256, 283, 309, 343, 345, 376, 409, 448, 451, 452, 503, 504, 520, 594, 605, 612, 621, 629, 658, 659, 667, 670, 681, 682, 703
!session, 33, 152, 153, 154, 155, 170, 181, 193, 220, 252, 270, 400, 441, 466, 467, 470, 480, 486, 487, 495, 520, 530, 567, 594, 623, 630, 634, 644, 661, 662, 691
!sprocess, 252
!stacks, 84, 105, 109, 187, 191, 212, 220, 222, 224, 225, 226, 227, 233, 252, 255, 256, 269, 278, 309, 332, 333, 336, 347, 409, 411, 417, 457, 461, 478, 484, 498, 519, 520, 596, 632, 635, 675, 679, 687, 695
!sym, 35, 49, 204, 205
!sysinfo, 43, 44, 219
!teb, 109, 119, 165, 167, 179, 199, 232, 233, 236, 271, 289, 672, 677
!thread, 236, 307, 330, 344, 421, 427, 428, 429, 448, 449, 452, 453, 454, 455, 456, 474, 475, 476, 703
!uniqstack, 191, 251, 409, 430

!vm, 206, 219, 224, 225, 226, 252, 303,
 327, 340, 440, 441, 442, 443, 444,
 446
!vm 4, 224, 225, 226, 252, 441
\$\$><, 233
\$\$>a<, 229, 530, 557
\$arg1, 229, 529, 556
\$arg2, 229, 529, 556
\$extret, 189
\$sp, 106, 115, 123, 124, 130, 154, 155,
 167, 184, 188, 189, 222, 223, 226,
 236, 285, 361, 368, 370, 390, 414,
 415, 475, 575, 600, 605, 620, 621,
 627, 631, 633, 698, 703
\$ptrsize, 188, 189
\$t0, 78, 79, 188, 189, 221, 222, 223,
 225, 501
\$t1, 78, 79, 188, 189, 221, 222, 223,
 225, 233, 236
.catch, 144, 189, 221, 262, 267, 270,
 314, 362, 365, 465, 703
.cxr, 273, 275, 321, 614, 625, 626
.echo, 44
.exepath, 65
.formats, 133, 207, 211, 515, 560, 668
.logclose, 44, 225, 226, 227, 235
.logopen, 44, 225, 226, 227, 233, 359
.NET Heap, 371
.printf, 78, 79, 188, 189, 221, 350, 544,
 548, 550, 635
.process, 84, 154, 155, 184, 214, 221,
 222, 223, 225, 226, 328, 414, 415,
 475
.reload, 35, 64, 65, 84, 200, 204, 214,
 222, 223, 225, 328, 409, 410, 414,
 496, 498
.sympath, 35, 64, 200
.thread, 86, 106, 214, 215, 236, 319,
 321, 398, 424, 652
.trap, 28, 32, 43, 83, 92, 93, 94, 95, 97,
 98, 100, 101, 103, 132, 140, 144,
 146, 148, 153, 257, 302, 303, 314,
 317, 397, 398, 649, 651, 652
~*e, 233
~*kv, 43, 187, 191, 227, 251, 277, 305,
 409, 625
~e, 233s

A

Accidental Lock, 423
Address Space Layout Randomization
(ASLR), 674, 678
Alien Component, 493
ASCII, 43, 167, 367, 440, 460, 559

B

backtrace, 132, 574, 575, 580, 581, 587
Be Language, 499
buffer overflow, 257, 460, 508
bugcheck, 43, 46, 49, 81, 105, 132, 133,
 135, 136, 139, 140, 141, 142, 143,
 144, 148, 150, 152, 153, 252, 267,
 300, 302, 314, 316, 387, 397, 452,
 460, 479, 480, 498, 508, 509, 646
Busy System, 448

C

CAFF, 150
CDB, 111, 228, 600, 620, 639, 640, 658
CF, 152
Changed Environment, 283
code optimization, 88, 265, 294, 300
Coincidental Symbolic Information, 390
committed memory, 302, 303
complete memory dump, 28, 29, 33, 37,
 68, 83, 98, 100, 103, 105, 151, 184,
 204, 212, 214, 219, 220, 221, 223,
 224, 228, 230, 239, 269, 278, 302,
 307, 308, 323, 327, 343, 397, 409,
 414, 419, 423, 457, 458, 496, 507,
 529, 594, 604, 644, 649, 653, 656,
 666
complexity, 158, 224, 500, 510
Coupled Processes, 419
CPU spike, 305, 421, 428, 456, 470
critical section, 223, 224, 226, 230, 251,
 252, 276, 278, 279, 348, 349, 355,
 421, 475, 482, 490, 491, 492
Custom Exception Handler, 470

D

DBG, 34, 637
 deadlock, 32, 276, 278, 323, 326, 348, 423, 473, 476, 481, 666
 default postmortem debugger, 29, 111, 113, 262, 270, 523, 600, 606, 612, 621, 623, 633, 638, 641, 658, 661, 669, 670, 673, 688
 dereference, 188, 189, 235, 267, 614, 658, 659
 device driver, 440, 647, 664, 700
 double free, 378
 Double Free, 387
 dpa, 43, 44, 167, 235
 dpp, 167, 188, 189
 dps, 43, 44, 53, 82, 188, 189, 232, 233, 235, 236, 299, 650
 dpu, 43, 44, 114, 167, 235, 657
 Dr. Watson, 29, 115, 230, 605, 621, 623, 624, 627, 634, 638, 641, 643, 688, 690
 dt, 73, 76, 77, 78, 93, 94, 97, 100, 116, 170, 177, 178, 179, 180, 199, 200, 270, 315, 317, 323, 343, 402, 616, 617, 626, 654
 dv, 584, 585, 586, 587
 Dynamic Memory Corruption, 257

E

Early Crash Dump, 465
 ERESOURCE, 323
 exception, 41, 43, 68, 69, 71, 73, 74, 78, 81, 83, 84, 86, 89, 94, 104, 105, 109, 115, 117, 123, 125, 131, 141, 143, 144, 145, 146, 151, 159, 165, 187, 252, 255, 256, 257, 259, 260, 261, 262, 267, 271, 272, 273, 274, 275, 318, 319, 331, 335, 362, 364, 365, 366, 397, 400, 465, 466, 467, 470, 471, 486, 487, 488, 497, 567, 574, 585, 609, 612, 614, 615, 623, 624, 625, 636, 638, 659, 661, 662, 666, 667, 670, 672, 673, 676

exception handler, 125, 141, 144, 271, 274, 318, 362, 366, 465, 470, 471, 624, 673, 676
 Exception Monitor, 260, 400, 465, 471, 612
 execution environment, 283, 285, 678

F

False Positive Dump, 259
 first chance exception, 104, 109, 123, 366, 465, 470, 666, 670, 672
 Fooled by Abbreviation, 500
 Fooled by Description, 497
 full page heap, 27, 257, 258, 383, 666, 691

H

handle leak, 327, 330, 440, 457
 hang, 31, 32, 33, 36, 37, 38, 43, 187, 251, 252, 271, 302, 419, 423, 430, 497, 590, 638, 639, 645
 heap corruption, 257, 258, 274, 288, 378, 379, 488, 638
 Hidden Exception, 271
 High Contention, 421
 Historical Information, 457
 Hooked Functions, 468
 hooking, 251, 368, 468
 hypertext command, 183, 312

I

I/O completion port, 653
 I/O Request Packet (IRP), 173, 269, 307, 428, 429, 437, 454, 455, 456, 457, 458, 459, 482, 484, 635, 701
 import table, 169, 264, 298, 300
 Inconsistent Dump, 269, 501
 Incorrect Stack Trace, 288
 Insufficient Memory, 440
 Insufficient Memory (Committed Memory), 302
 Insufficient Memory (Handle Leak), 327

interrupt, 69, 71, 73, 74, 75, 76, 78, 81, 82, 83, 86, 89, 92, 93, 95, 100, 101, 132, 137, 146, 314, 448, 649

interrupt frame, 83, 86, 89, 93

invalid pointer, 142, 267

Invalid Pointer, 267

IRP Distribution Anomaly, 458

IRQL_NOT_LESS_OR_EQUAL, 132, 136

J

JIT debugging, 605

K

kernel memory dump, 72, 151, 187, 193, 224, 225, 278, 298, 302, 309, 342, 411, 421, 423, 436, 437, 458, 496, 498, 527, 533, 534, 535, 536, 604, 616, 629, 664, 666, 701

kernel pool, 387, 440

kernel space, 83, 86, 142, 152, 198, 267, 302, 440, 461, 527, 604, 616, 664, 684, 685

KERNEL_MODE_EXCEPTION_NOT_H
ANDLED, 94, 141, 143, 144, 268

KL, 63, 64, 66, 98, 99, 104, 106, 118, 125, 151, 256, 273, 363, 364, 366, 380, 384, 386, 389, 438, 439, 460, 471, 472, 486, 487, 519, 520, 524, 525, 577, 579, 609, 649, 652, 702

KMODE_EXCEPTION_NOT_HANDLE
D, 81, 143, 144

kv, 43, 44, 103, 159, 164, 187, 191, 215, 227, 251, 271, 277, 288, 305, 314, 340, 409, 580, 587, 614, 625, 657

L

Lateral Damage, 264

list traversing, 178

Im, 153, 199, 200, 283, 391, 392, 510, 557, 674, 677

Imt, 219, 310, 510

Imv, 43, 44, 47, 51, 52, 60, 61, 63, 224, 225, 226, 227, 298, 310, 312, 321, 341, 367, 368, 389, 398, 447, 656

In, 73, 78, 79, 252, 300, 315, 316, 689

Local Buffer Overflow, 460

Local Procedure Call (LPC), 473

local variables, 581

M

main thread, 158, 362, 363, 364, 436, 437, 483, 659, 679, 681, 682

managed code, 331, 332, 338

Manual Dump (Kernel), 479

Manual Dump (Process), 486

Manual Stack Trace, 293

memory dump, 29, 32, 33, 38, 39, 217, 220, 222, 224, 225, 226, 251, 255, 269, 276, 278, 338, 340, 342, 358, 372, 373, 419, 436, 447, 448, 451, 464, 465, 478, 479, 480, 486, 497, 501, 506, 508, 510, 514, 516, 518, 529, 532, 545, 546, 547, 554, 556, 557, 559, 561, 590, □ 604, 612, 628, 646, 647, 687

memory dump file, 252, 269, 340, 501

memory leak, 302, 327, 356, 371, 440

memory visualization, 532, 544, 556

minidump, 43, 423

Missing Thread, 362

module, 310, 367, 389

Module Variety, 310

Multiple Exceptions, 255

multiprocessor, 72, 635, 681, 703

N

Need the crash dump, 498

NMI_HARDWARE_FAILURE, 135, 480

No Component Symbols, 298

NTDLL, 170, 276, 277, 554, 624, 650, 697

NTSD, 29, 32, 262, 270, 393, 478, 523, 605, 618, 624, 627, 631, 633, 638, 639, 640, 641, 643, 644, 658, 693

NULL pointer, 249, 262, 268, 614, 658, 659, 663

O

object name, 214
 OMAP, 88, 266, 294, 296, 300, 649, 666
 Optimized Code, 265

P

page fault, 132, 133, 314
 Passive System Thread, 461
 Passive Thread, 430
 PDB, 34, 177, 185, 199, 204, 230, 391,
 498, 524, 637, 646
 PDBFinder, 637, 668
 poi, 189, 221, 222, 223, 225
 pool tag, 206, 208, 209, 328, 387, 388,
 440, 443, 444, 445, 446, 447, 499,
 593
 postmortem debugger, 28, 29, 38, 39,
 110, 113, 114, 115, 125, 131, 230,
 393, 606, 613, 619, 641, 642, 658,
 659, 666, 669
 process heap, 356, 371, 378, 387, 678
 pseudo-register, 188, 189, 236

R

r, 27, 36, 38, 43, 44, 63, 64, 76, 78, 79,
 81, 83, 95, 102, 104, 106, 110, 124,
 132, 144, 146, 154, 155, 177, 180,
 184, 188, 189, 221, 222, 223, 225,
 226, 233, 236, 242, 246, 251, 277,
 280, 288, 305, 312, 332, 340, 368,
 370, 380, 385, 405, 407, 414, 415,
 419, 420, 430, 431, 435, 460, 470,
 475, 488, 496, 559, 600, 626, 638,
 639, 640, 641, 659, 671, 673
 race condition, 181, 681, 683
 raw stack, 43, 53, 73, 82, 83, 86, 88,
 109, 110, 119, 144, 145, 147, 159,
 165, 167, 193, 236, 239, 251, 264,
 271, 274, 288, 319, 321, 332, 349,
 350, 354, 390, 498, 601, 610, 611

S

second chance exception, 109, 110,
 465, 466, 673
 security, 195, 208, 209, 217, 224, 230,
 246, 248, 267, 299, 369, 372, 469,
 570, 599, 604, 666, 671, 689
 software defect, 511
 Special Stack Trace, 478
 Spiking Thread, 305
 stack overflow, 314
 stack reconstruction, 158
 stack trace, 43, 51, 53, 63, 64, 65, 73,
 89, 91, 103, 113, 125, 147, 148, 157,
 158, 159, 163, 165, 182, 193, 230,
 236, 244, 246, 252, 257, 265, 266,
 278, 288, 292, 293, 295, 301, 318,
 319, 326, 330, 333, 335, 338, 348,
 350, 354, 358, 360, 361, 380, 384,
 385, 388, 398, 400, 409, 417, 419,
 430, 435, 448, 457, 461, 473, 478,
 479, 483, 484, 498, 565, 573, 574,
 575, 577, 579, 580, 581, 584, 587,
 600, 609, 614, 624, 653, 657, 658,
 660, 688, 689, 690, 692

Stack Trace, 395

Stack Trace Collection, 409

Structured Exception Handling (SEH),
 162, 194, 197, 267, 318, 319, 334,
 362, 465, 679

symbol file, 27, 34, 35, 63, 199, 200,
 204, 298, 301, 361, 388, 396, 414,
 596, 637, 689

symbol server, 27, 34, 35, 63, 204, 230,
 468, 498, 637, 693

system queue, 252

system thread, 144, 236, 300, 461, 463

SYSTEM_THREAD_EXCEPTION_NOT
 _HANDLED, 144, 268

SystemDump, 33, 37, 105, 269, 448,
 455, 479, 514, 629, 630, 634, 645,
 646, 647, 657

T

TestDefaultDebugger, 39, 41, 113, 117,
 118, 119, 122, 123, 125, 465, 466,

467, 470, 600, 601, 603, 605, 613, 619, 622, 623, 636, 641, 642, 674, 677, 688, 689, 690
trap, 28, 32, 43, 83, 92, 93, 94, 95, 97, 98, 100, 101, 103, 132, 140, 144, 146, 148, 153, 257, 302, 303, 314, 317, 397, 398, 649, 651, 652
Truncated Dump, 340

U

u, 34, 43, 44, 78, 88, 89, 107, 140, 164, 168, 169, 177, 210, 259, 294, 295, 296, 297, 368, 370, 393, 469, 479, 523, 524, 525, 569, 570, 572, 580, 587, 600, 605, 620, 622, 631, 633, 636, 650, 651, 684, 685, 686, 690
ub, 43, 44, 65, 66, 88, 89, 161, 162, 211, 247, 296, 388, 392, 393, 571, 572, 580, 587, 624, 649, 650, 651, 657
uf, 43, 44, 65, 92, 100, 114, 127, 130, 148, 163, 207, 241, 246, 248, 265, 266, 296, 316, 354, 449, 569, 570, 572, 575, 580, 587, 650, 670, 671, 679
UNICODE, 43, 116, 167, 171, 201, 265, 266, 273, 367, 391, 460, 524, 559, 616, 617
Unified Modeling Language (UML), 71, 76, 136, 142, 148, 255, 268, 596, 647, 655, 700
uniprocessor, 681, 683
Unknown Component, 367
user memory dump, 27, 29, 151, 227, 230, 258, 276, 356, 419, 465, 470, 478, 537, 538, 539, 540, 541, 542, 543, 606, 621, 622, 634, 669, 689
user mode, 71, 76, 83, 93, 104, 105, 108, 181, 182, 223, 230, 257, 258,

267, 305, 307, 308, 358, 387, 460, 465, 620, 632, 639, 640, 656
user space, 83, 86, 109, 141, 220, 236, 257, 267, 268, 273, 274, 278, 302, 409, 414, 430, 461, 475, 604, 664, 684, 685
userdump.exe, 32, 38, 39, 150, 151, 260, 305, 356, 366, 372, 400, 465, 470, 478, 486, 608, 612, 613, 618, 619, 620, 624, 631, 632, 638, 639, 640, 644

V

virtualization, 400, 404
Virtualized Process, 400

W

wait chain, 481, 482, 490, 492
waiting thread, 214, 252, 343, 457, 655
WinDbg command, 46, 64, 65, 100, 167, 170, 188, 191, 204, 214, 221, 222, 229, 233, 269, 283, 310, 356, 430, 443, 444, 458, 493, 560, 565, 567, 593, 635, 664
WinDbg script, 44, 76, 78, 188, 221, 227, 229, 409, 529, 556, 599, 635
Windows Error Reporting (WER), 38, 111, 113, 114, 131, 606, 624, 658, 659, 661, 662, 669
Word of Mouth, 495
WOW64, 400, 516, 632
Wrong Dump, 496

Z

Zippocracy, 494

Memory Dump Analysis Anthology

Volume 2

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2008 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-0-9558328-7-1 (Paperback)

ISBN-13: 978-1-906717-22-3 (Hardback)

First printing, 2008

Revision 3 (April, 2015)

Contents

Preface	15
Acknowledgements.....	17
PART 1: Crash Dumps for Beginners	19
The Time of the Crash	19
Stack Trace	20
EasyDbg.....	22
Citrix Symbol Server	27
PART 2: Professional Crash Dump Analysis.....	29
WinDbg Scripts	29
Introduction for C/C++ Users	29
Generating File Name for .dump Command	37
All at Once: Postmortem Logs and Dump Files	38
Common Mistakes	39
Not Looking at Full Stack Traces.....	39
Not Seeing Semantic and Pragmatic Inconsistencies.....	41
Pattern Interaction.....	43
Heuristic Stack Trace	43
Multiple Patterns	50
Exception and Deadlock	55
Heap and Spike.....	59
Hookware	63

Heap and Early Crash Dump.....	65
WinDbg Shortcuts	67
WinDbg as a Binary Editor.....	67
Command Autocompletion.....	70
!envvar	71
.quit_lock.....	72
.dumpcab	73
.f+, .f-	74
.expr	75
WinDbg as a Simple PE Viewer.....	76
.sound_notify	79
Signaled Objects.....	80
Memory Search Revisited	87
WDF and PNP BSOD: Case Study.....	95
Exploring NDIS Extension	105
The Hunt for the Debugger	109
Complete Dump: User Space Critical Sections	115
Microsoft DLL Help Database.....	116
What Does This Function Do?	118
What Was This Process Doing?	119
STL and WinDbg	122
WinDbg Cheat Sheet	125
How Old Is Your Application or System?.....	126

Demystifying First-chance Exceptions.....	129
.NET Managed Code Analysis in Complete Memory Dumps	131
Who Opened That File?.....	134
In Search of Lost CID	136
Large Heap Allocations.....	137
First-order and Second-order Memory Leaks	140
Hooked Modules	145
PART 3: Crash Dump Analysis Patterns.....	147
Wait Chain (Executive Resources).....	147
Corrupt Dump	151
Dispatch Level Spin	154
No Process Dumps	157
No System Dumps	158
Insufficient Memory (PTE).....	159
Suspended Thread	161
Special Process	164
Frame Pointer Omission.....	169
False Function Parameters.....	173
Message Box	177
Self-Dump.....	181
Blocked Thread.....	184
Zombie Processes.....	196
Wild Pointer	202

Dynamic Memory Corruption (Kernel Pool).....	204
Insufficient Virtual Memory	210
Wild Code.....	219
Hardware Error	221
Handle Limit (GDI).....	226
Missing Component	233
NULL Pointer (Code).....	237
Execution Residue	239
Optimized VM Layout.....	267
Invalid Handle	269
Overaged System	273
Thread Starvation.....	274
Stack Overflow (User Mode)	279
Missing Component (Static Linkage)	283
Duplicated Module.....	294
Not My Version	299
Data Contents Locality	300
Nested Exceptions (Unmanaged Code).....	305
Nested Exceptions (Managed Code)	310
Affine Thread.....	314
Self-Diagnosis	318
Waiting Thread Time (User Dumps).....	319
Inline Function Optimization.....	322

Critical Section Corruption	324
Lost Opportunity	332
Young System	335
Last Error Collection	337
Hidden Module	339
High Contention (CriticalSection)	341
PART 4: Crash Dump Analysis AntiPatterns	343
Debugging Architects	343
Symbolless Analysis.....	344
Myopic Troubleshooting and Debugging	345
PART 5: A Bit of Science	347
Memoretics	347
Memory Analysis.....	348
Memoidealism	349
Memiotics	350
PART 6: Fun with Crash Dumps.....	351
Music for Debugging	351
The Glory of Debugging.....	351
Memory Analysis Album	352
Biography of a Bug	354
Visual Computer Memories	355
The First Defect	356
The Songs for Remote Debugging	357

Thinking Out of the Box	358
Crash Dumps and Science Fiction	359
Colorimetric Computer Memory Dating	360
On CSI Abbreviation	362
The First Memory Dump Book	363
On SOS Abbreviation	365
Software Exceptions: a Paranormal View	366
Bug Entanglement (Bugtanglement)	367
The Standard Model of Debugging	368
Physics of Debugging	369
Can Computers Debug?	371
PART 7: Data Recovery	375
With the Help of Memory Dump Analysis.....	375
PART 8: Software Troubleshooting	377
Troubleshooter's Block	377
Causal Models	378
Object-Oriented Debugging and Troubleshooting	379
Component-Based Debugging and Troubleshooting	380
Domain-Driven Debugging and Troubleshooting	381
Myths and Facts about Software Support.....	382
Ceteris Paribus in Comparative Troubleshooting.....	383
Dancing in Software Support Environment.....	384
PARTS: Problem Solving Power of Thought	385

The Hidden Tomb in Pyramid of Software Change	386
Tracing.....	387
CDF Traces: Analyzing Process Launch Sequence	387
ETW Tracing Tools	389
Lean Tracing	390
Debugware Patterns	391
API Query	391
Tool Façade	392
Configuration Wrapper	393
Dual Interface.....	394
Tool Chain	395
Tool Box.....	396
PART 9: Security	397
Data Hiding in Crash Dumps.....	397
Hardening Dump Security: Beware of PEB Data	400
PART 10: The Origin of Crash Dumps	401
Memory Dumps from Xen-virtualized Windows.....	401
Bugchecks: SYSTEM_SERVICE_EXCEPTION	402
Bugcheck Callbacks	406
Application Verifier on x64 Platforms	413
Who Saved the Dump File?	414
ADPlus in 21 Seconds and 13 Steps.....	416
PART 11: Miscellaneous	425

Three Main Ideas of Debugging	425
Pseudo-corrupt Memory Dumps	426
Win32 Exception Frequencies.....	427
Bugcheck Frequencies.....	429
Time Travel Debugging.....	440
I/O and Memory Priority in Vista	441
Appendix A.....	443
Crash Dump File Examples	443
Appendix B	445
WinDbg.Org: WinDbg Quick Links	445
Appendix C	447
Dump2Wave Source Code	447
Appendix D	451
Dump2Picture Source Code	451
Appendix E	455
Crash Dump Analysis Checklist.....	455
CMDTREE.TXT.....	457
Index	459
Notes.....	465
About the Author	469
Cover Images.....	470

Index

- !
- !address, 57, 67, 137, 139, 141, 211, 213, 324, 325, 329, 330, 342
 - !analyze, 38, 46, 51, 70, 152, 169, 203, 204, 205, 270, 284, 310, 455, 456
 - !analyzeexception, 70
 - !apc, 456
 - !bugdump, 406, 456
 - !chkimg, 145, 302, 303, 455
 - !d*, 89
 - !devobj, 102
 - !devstack, 102
 - !dh, 76, 216, 340, 455
 - !dlls, 290, 296, 298
 - !dpcs, 456
 - !dumpil, 133
 - !dumpmt, 133
 - !EEHeap, 132
 - !envvar, 8, 71, 455
 - !error, 151, 283
 - !exchain, 403, 455
 - !exqueue, 456
 - !for_each_module, 145
 - !for_each_process, 115, 134
 - !for_each_thread, 338
 - !gchandles, 133
 - !gflag, 455
 - !gle, 65, 286, 337, 338
 - !handle, 121, 134
 - !heap, 60, 137, 139, 143, 342
 - !irp, 82, 101, 180
 - !irpfind, 456
 - !kdexts.handle, 83, 200
 - !locks, 50, 51, 55, 61, 147, 325, 326, 327, 330, 341, 371, 455, 456
 - !ipc, 456
 - !ndiskd, 105, 106, 108
 - !ntsddexts.locks, 115, 456
 - !object, 82, 187, 188
 - !peb, 138, 289, 297, 397, 400, 455
 - !pool, 90, 95, 96, 100, 208, 300, 301, 456
 - !poolused, 226, 456
 - !process, 40, 53, 80, 83, 113, 131, 132, 162, 164, 165, 166, 167, 177, 185, 186, 199, 200, 336, 397, 398, 456
 - !pte, 96, 107
 - !qlocks, 156, 456
 - !ready, 186, 456
 - !runaway, 59, 319, 321, 455
 - !running, 184, 274, 316, 456
 - !search, 88, 89, 90, 93, 94
 - !session, 335, 456
 - !sprocess, 456
 - !stacks, 114, 456
 - !stl, 122
 - !sysinfo, 401
 - !teb, 44, 110, 170, 234, 241, 280, 286, 307, 333, 338
 - !thread, 52, 80, 81, 86, 100, 150, 154, 178, 185, 187, 188, 189, 275, 276, 277, 278, 301, 314, 315, 316, 441
 - !threads, 132
 - !vad, 398, 399
 - !vm, 52, 53, 159, 196, 335, 456
- \$
- \$\$<, 31
 - \$<, 31
 - .
 - .asm, 209
 - .block, 31, 32, 33, 35, 36
 - .cmdtree, 457
 - .cxr, 45, 56, 57, 190, 222, 226, 308, 309, 402
 - .dump, 7, 37, 38, 73, 297, 319, 398, 399, 400
 - .dumpcab, 8, 73

.else, 35
 .enumtag, 406, 410, 456
 .expr, 8, 75
 .exr, 43, 222, 233, 270, 271, 279, 284, 288
 .f-, 8, 74
 .ft+, 8, 74
 .for, 36
 .foreach, 375
 .formats, 88, 202
 .frame, 74, 174
 .if, 33, 35, 338
 .ignore_missing_pages, 93
 .imgscan, 339
 .kframes, 38, 40, 455
 .lastevent, 286
 .loadby, 132
 .logclose, 38
 .logopen, 38
 .printf, 31, 32, 33, 35, 36, 37
 .process, 115, 132, 165, 167, 178, 190, 397
 .quit_lock, 8, 72
 .readmem, 68
 .sound_notify, 8, 79
 .thread, 52, 57, 185, 189, 190, 274, 276, 338
 .trap, 95, 155, 190, 300
 .while, 36
 .writemem, 69, 375

/

/3GB boot option, 160

—

_DISPATCHER_HEADER, 80, 83, 84, 85, 86, 97, 161, 162, 201, 315, 317, 404
 _EPROCESS, 91
 _OBJECT_HEADER, 80, 83, 84, 85
 _OBJECT_TYPE, 84, 85

~

~*kv, 38, 62, 332, 455

A

ADPlus, 13, 130, 269, 416, 418, 419, 420
 AeDebug, 38, 157, 414, 424
 Affinity mask, 184, 274, 314, 316, 317
 Aliases, 32, 37
 Alien component, 299
 API Query, 391
 Application verifier, 13, 269, 270, 413
 ASCII fragments, 372

B

Backward debugging, 425
 Blocked GUI thread, 195
 Blocked thread, 9, 177, 184, 195, 335
 Buffer overflow, 318, 343
 Bugcheck callbacks, 406
 Bugluon, 368
 Bugtanglement, 12, 367

C

CAD diagram, 128
 Causal models, 378
 CBDT, 380
 CDA checklist, 22
 CDA poster, 125
 CDB, 37, 38, 283, 295, 422
 CDF traces, 383, 387
 CDFControl, 389, 394
 Ceteris paribus, 12, 383
 Changed environment, 63, 157, 267
 Checklist, 455
 Classical deadlock, 58
 Classical μ -memuon, 369
 CMDTREE.TXT, 14, 457
 Code path locality, 300
 Colorimetric computer memory dating, 360
 Committed memory, 210
 Comparative memory dump analysis, 210
 Complete memory dump, 40, 87, 110, 112, 115, 119, 131, 164, 397, 402
 Component age diagram, 126, 299
 Component dependencies, 117

Component Identification, 207
 Computer memory visual images, 355
 Computer name, 71, 455, 456
 Configuration wrapper, 393
 Context record, 75
 Corrupt dump, 9, 151, 158, 426
 Corrupt structure, 324
 Coupled processes, 55, 59, 177, 335
 CPU spikes, 39, 54, 274
 Critical section corruption, 11, 324, 371
 CSI, 12, 362
 Custom exception handler, 65, 181

D

d* commands, 89
 dA, 179
 Data contents locality, 10, 300
 Data hiding, 397
 Data recovery, 375
 db, 103, 145, 215, 235, 372
 dc, 67, 68, 89, 91, 93, 103, 340
 dd, 99, 120, 123, 136, 219, 301, 397, 398
 DDDT, 381
 dds, 44, 64, 110, 171, 216, 234, 241, 251,
 280, 286, 333, 415
 Deadlock, 7, 55, 136, 335
 Debugluon, 368
 DelayLoad export missing, 233
 Delta debugging, 425
 Detoured module, 214
 Dispatch level spin, 9, 154, 314
 dl, 138, 139, 289
 Double free, 324
 dp, 237
 dpa, 124, 235
 dpp, 136
 dps, 208, 236, 271, 406
 dpu, 124, 235
 dq, 138, 220
 dqs, 308, 309
 Dr. Watson, 164, 168
 Driver verifier, 104
 dS, 456

dt, 85, 86, 91, 97, 123, 124, 138, 161, 162,
 201, 301, 315, 317, 324, 325, 329, 330,
 372, 400, 404, 405
 du, 215, 289, 330, 340, 415
 Dual interface, 13, 394
 Dump2Picture, 451
 Dump2Wave, 447
 Duplicated module, 10, 294, 299, 443
 dv, 122, 174
 Dynamic memory analysis, 348
 Dynamic memory corruption, 204
 Dynamic memory infrastructure, 140

E

e* commands, 68
 ea, 68
 Early crash dump, 8, 65, 269
 EasyDbg, 7, 22, 26
 ETW, 13, 383, 387, 389, 390
 Exception context, 45, 56, 75, 287, 302, 308,
 309
 Exception dispatcher, 305
 Exception handling residue, 239
 Exception record, 75
 Execution residue, 10, 64, 239
 Executive resources, 50, 147, 341

F

fastcall, 173, 176
 First-chance exception, 65, 129, 130, 157,
 305, 308
 First-order memory leaks, 140
 Forward debugging, 425
 fp, 397
 FPO, 154, 169
 Frame pointer omission, 169
 Full dump, 38
 Function arguments, 32, 33
 Functional memory analysis, 348

G

Garbage-collected heap, 210

GDI leaks, 226
GDI objects, 226
Gflags, 393, 394, 455

H

Handle leak, 201, 210
Handle table, 121, 134, 135, 200
HandleCount, 53, 82, 83, 84, 85, 113, 121, 131, 132, 134, 135, 164, 165, 167, 187, 188, 199, 200, 201, 336, 397, 398
Heap control structures, 61
Heap corruption, 56
Heap memory leaks, 137
Hidden exception, 43, 288
Hidden module, 11, 64, 339
High contention, 11, 341
Hooked functions, 63, 145, 222, 302
Hooking mechanism, 63, 64
Hooks, 63, 64
Hookware, 63

I

Imageless analysis, 344
Incorrect stack trace, 181
Infinite loops, 154
Inline function optimization, 322
Inner exception, 305, 310
Insufficient memory, 9, 52, 63, 140, 141, 159, 210, 226
Invalid handle exception, 269
Invalid pointer, 203, 226, 237
IRQL, 51, 95, 96, 104, 154, 204, 205, 225, 300, 314, 430, 432, 433, 434, 437
IRQL_NOT_LESS_OR_EQUAL, 95, 300, 430
IsDebuggerPresent, 109, 111, 217
Isomemotopes, 360

K

kbnL, 173
Kernel dump, 112
Kernel pool, 10, 204, 210

KiUserExceptionDispatcher, 44, 46, 47, 56, 75, 171, 264, 287, 414
kL, 43, 45, 57, 59, 65, 66, 107, 109, 119, 181, 190, 233, 240, 270, 279, 280, 284, 295, 297, 299, 315, 322, 341, 403
kn, 74
kv, 75, 120, 136, 169, 173, 237, 302, 318, 321, 372
kvL, 39, 40

L

Large heap, 9, 137, 443
Last error collection, 337
Linked list, 138, 208, 406
ListProcessStacks, 456
lm, 37, 67, 76, 210, 212, 267, 294, 296, 298, 304, 339, 372, 373, 375, 401
lmM, 37
lmt, 126, 128
lmv, 38, 103, 214, 296, 400
ln, 456
Local buffer overflow, 170, 202, 219, 325
Lost opportunity, 332

M

Main thread, 195
Managed code, 131, 310
Managed code exception, 233
Manual dump, 154
Memiotics, 350
Memoidealism, 349
Memoretics, 11, 347, 348
Memory analysis forensics, 348
Memory analysis intelligence, 348
Memory leak, 140
Memuon, 349
Message box, 9, 177, 195, 318
MFC, 39
Minidump, 38, 73, 152, 340, 344, 401
Minidump analysis, 344
Missing component, 10, 233, 283, 443
Module variety, 128, 294, 299

N

NDIS, 8, 105, 107, 108, 315, 436, 437
 Nested exceptions, 10, 305, 310, 443
 No process dumps, 9, 157, 158
 Not my version, 299
 NTSD, 109, 114, 157, 168, 414, 416
 NULL pointer, 237

O

OODT, 379, 381
 Optimized code, 175
 Overaged system, 10, 273, 335

P

PAE, 160
 Page faults, 96
 Partial stack traces, 239, 300
 Partially reconstructed stack trace, 183
 PARTS, 12, 385
 Passive thread, 39, 195
 PE Viewer, 76
 Physical address, 87, 89, 159
 Physics of debugging, 369
 PNP, 8, 95, 228, 432
 PointerCount, 82, 83, 84, 85, 121, 134, 135, 187, 188, 200, 201
 Pool allocations have failed, 53, 159
 Postmortem debugger, 37, 38, 55, 65, 129, 130, 168, 181, 250, 269, 270, 283, 295, 307, 413, 414, 416
 Private memory, 196
 Process context, 71, 190
 Process heap, 138, 139, 140, 143, 324, 341
 Pseudo-registers, 32
 PTE, 9, 91, 96, 107, 159, 160, 210, 432
 PTE allocation failures, 159, 160

Q

Quantum theory of software bugs, 367

R

Raw stack data, 43, 44, 182, 233, 235, 239, 241, 280, 286, 288, 302, 414
 Raw stack dump, 181, 305
 Raw stack dump of all threads, 181
 Ready threads, 186, 456
 Remote debugging, 357
 RPC, 45, 48, 136, 190, 337
 RtlDispatchException, 46, 47, 264, 270, 286, 403
 RtlFreeHeap, 45, 46, 59, 60, 182, 217, 252
 Runaway information, 39
 Running threads, 184, 456

S

s -d, 93
 SDT, 379, 381
 Secure dumps, 38
 Secure full dump, 38
 Secure minidump, 38
 Self-diagnosis, 10, 318
 Self-dump, 9, 181, 318
 Session pool, 53, 208, 226, 230, 231, 232
 Signaled object, 80, 187
 Signaled state, 80, 85
 Signs of exceptions, 286
 Software change, 386
 SOS, 12, 131, 365
 Special process, 9, 163, 164
 Spiking thread, 52, 59, 150, 154, 274
 Spinlocks, 156, 456
 Stack trace, 7, 20, 39, 41, 43, 45, 164, 170, 181, 283, 318
 Stack trace collection, 39
 Stack trace of the problem thread, 20
 StackBase, 44, 110, 170, 234, 280, 286, 307, 333, 338
 StackLimit, 44, 110, 161, 162, 170, 234, 280, 286, 307, 333, 338, 404
 Static memory analysis, 348
 STL, 8, 122
 StressPrinters, 392

SuspendCount, 113, 161, 162, 167, 168
Suspended thread, 9, 161, 166
Symbol server, 7, 27
Symbolless analysis, 344
System PTE allocations have failed, 159
SYSTEM_SERVICE_EXCEPTION, 13, 402, 431
SystemDump, 131, 394, 397, 398

T

TDI, 180
TestDefaultDebugger, 131, 132, 133, 416, 423
Sstandard model, 368
this pointer, 175
thiscall, 175
Thread startup residue, 241
Thread Starvation, 10, 274, 314
Time travel debugging, 425, 440
Tool box, 396
Tool chain, 13, 395, 396
Tool façade, 13, 392, 394
Troubleshooter's block, 12, 377
Truncated dump, 151, 158

U

ub, 170, 171, 174, 189, 209, 237, 271, 308
uf, 97, 155, 172, 174, 175, 404
UML diagram, 84, 405
UnhandledExceptionFilter, 46, 56, 75, 217, 262, 333, 414, 415
Unknown component, 63, 103, 215, 339
Unmanaged code, 305
User dumps, 70, 319, 335, 399, 400
User-defined variables, 32

userdump.exe, 59, 65, 111, 319, 340

V

vertarget, 19, 160, 320
VirtualPC, 164
VMWare, 164, 401

W

Wait chain, 9, 50, 57, 61, 147, 314, 335
Waiting thread time, 10, 319
Waiting threads, 191, 193, 456
WDF, 8, 95, 432
WDS, 29
Wild code, 10, 63, 202, 219, 223, 302
Wild pointer, 9, 202, 219, 237, 325
WinDbg script, 29, 30, 32, 35
Window messaging, 39
WindowHistory, 391
WOW64, 292

X

x64 Windows, 157, 323, 402, 403, 413
Xen, 401

Y

Young system, 335

Z

Zombie processes, 9, 86, 196

Memory Dump Analysis Anthology

Volume 3

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2009 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-906717-43-8 (Paperback)

ISBN-13: 978-1-906717-44-5 (Hardback)

First printing, 2009

Revision 2 (April, 2015)

Contents

Preface	17
Acknowledgements.....	19
PART 1: Professional Crash Dump Analysis.....	21
Sparse Complete x64 Memory Dumps.....	21
Common Mistakes	24
Not Looking at All Stack Traces	24
Dump Analysis on Windows 7	28
32-bit Stack Traces from x64 Complete Memory Dumps	43
Debugger Log Reading Technique.....	48
Variable Kernel Stack in Vista and W2K8	49
Advanced Local Procedure Call WinDbg Extension.....	52
!cs vs. !ntsdexts.locks	54
Copyright as Timestamp	55
NULL Data Pointer Pattern: Case Study	56
Looking for Abnormal: Case Study	60
Raw Stack Dump of All Threads	62
Comparative Memory Dump Analysis: CPU Spikes.....	63
Graphical Notation for Memory Dumps	68

Exception Addresses from Event Logs	71
The Importance of Symbols	72
Platformorphism	75
PART 2: Crash Dump Analysis Patterns	77
Data Alignment	77
Multiple Exceptions (Kernel Mode)	78
C++ Exception.....	84
Deadlock (Mixed Objects, Kernel Space)	85
Wait Chain (Thread Objects).....	92
Divide by Zero (User Mode)	96
Wait Chain (LPC/ALPC)	97
Insufficient Memory (Physical Memory).....	104
Swarm of Shared Locks	107
Process Factory	112
Paged Out Data	118
Semantic Split.....	120
Pass-Through Function.....	129
NULL Pointer (Data)	131
JIT Code	132
PART 3: Crash Dump Analysis AntiPatterns	137

No Question	137
Missing Space.....	138
Part 4: Pattern Interaction	141
Early Crash Dump, Blocked Thread, Not My Version, and Lost Opportunity.....	141
Lateral Damage, Stack Overflow, and Execution Residue.....	144
Truncated Dump, Spiking Thread, Not My Version, and Hooked Functions.....	149
Stack Trace Collection, Hidden Exception, and NULL Code Pointer.....	155
WOW64, Blocked Threads, and Coupled Processes	160
Invalid Handle, Stack Trace Collection, Multiple Exceptions, Invalid Pointer, Data Alignment on Page Boundary, Dynamic Memory Corruption, and Not My Version	163
Wait Chain and Spiking Thread	167
Blocked GUI Thread, Wait Chain, and Virtualized Process.....	170
Insufficient Memory, Handle Leak, Wait Chain, Deadlock, Inconsistent Dump, and Overaged System	175
Memory Leak, Spiking Threads, Wait Chain, High Critical Section Contention, and Module Variety	181
NULL Code Pointer, Changed Environment, Hooked Functions, and Execution Residue	196
Swarm of Shared Locks, Blocked Threads, and Waiting Time.....	201
Stack Trace Collection, Blocked Thread, and Coupled Processes	205
Insufficient Memory, Handle Leak, Process Factory, High Contention, and Busy System.....	209
Busy System, Blocked Threads, Wait Chains, and Deadlock	215

Manual Dump, Dynamic Memory Corruption, Blocked Threads, Stack Trace Collection, Multiple Exceptions, Wait Chains and Deadlock.....	224
Coupled Processes, Wait chains, Message Box, Waiting Thread Time, Paged Out Data, Incorrect Stack Trace, Hidden Exception, Unknown Component, and Execution Residue.....	228
Manual Dump, Wait Chain, Blocked Thread, Dynamic Memory Corruption, and Historical Information	236
Blocked Threads, Message Box, and Self-Diagnosis.....	240
Manual and Early Crash Dump, Stack Trace Collection, Main Thread, Blocked Threads, and Pass-Through Functions	241
Blocked Thread, Historical Information, Execution Residue, Hidden Exception, Dynamic Memory Corruption, Incorrect Stack Trace, and Not My Version	245
Null Data Pointer, Incorrect Stack Trace, Changed Environment, Hooked Functions, and Coincidental Symbolic Information.....	248
Heap Corruption, Module Variety, Execution Residue, Coincidental Symbolic Information, and Critical Section Corruption	255
Stack Trace Collection, Blocked Threads, Pass-Through Functions, and Main Thread	262
Stack Trace, Invalid Code Pointer, and Hooked Functions.....	264
Manual Dump, Virtualized Process, Stack Trace Collection, Multiple Exceptions, Optimized Code, Wild Code Pointer, Incorrect Stack Trace, and Hidden Exception	268
Main Blocked Thread, Missing Component, Execution Residue, and Data Contents Locality	275
Inconsistent Dump, Blocked Threads, Wait Chains, Incorrect Stack Trace, and Process Factory	279
Invalid Pointer, Incorrect Stack Trace, Multiple Exceptions, Insufficient Memory, and Memory Leak	288

PART 5: A Bit of Science and Philosophy.....	295
Universal Memory Dump: A Definition.....	295
The Source of Intuition about Infinite	296
Geometrical Debugging	297
Riemann Programming Language	299
Is Memory Dump Analysis a Science?	300
My Dangerous Idea: Parameterized Science	301
Unique Events and Historical Narratives.....	302
Notes on Memoidealism	303
A Copernican Revolution in Debugging.....	305
On Subjectivity of Software Defects	306
Memory Field Theories of Memuonics	307
Software Trace: A Mathematical Definition.....	308
Quantum Memory Dumps	309
Chemistry of Virtual Memory	310
PART 6: Fun with Crash Dumps.....	313
Music for Debugging	313
Bugs Never Disappear	313
Horrors of Computation.....	314
Passion, Intellect, and Expression	315

Headphones for Debugging	316
In the Memory Dump File	317
Bugteriology	318
Implausible Debugging Book Titles	319
Build Date Astrology	320
Breaking Technical Barrier	321
Occult Debugging	322
The Year of Dump Analysis!	323
Stack Traces and Poetry	324
Debugging Slang	326
Memory Dump Analysis Walks	327
E-Acheri	329
The Meaning of DATA	330
Irish Government on Dumps	331
Memory Dumps as Relics	332
The Ghost of Adelphi Training Center	333
PART 7: Software Troubleshooting	335
I'm RARE	335
To Bugcheck or Not To Bugcheck	336
T&D Labyrinth	337

Efficient vs. Effective: DATA View	339
PART 8: Software Trace Analysis.....	341
Tracing Best Practices	341
Software Narratology: A Definition.....	342
PART 9: Software Trace Analysis Patterns	343
Introduction	343
Periodic Error	344
Basic Facts	345
Circular Trace	346
Intra-Correlation	347
PART 10: The Origin of Crash Dumps	351
Hide, Seek, and Dump	351
OSMOSIS Memory Dumps	353
Tools.....	356
Crash2Hang	356
MTCrash	358
Where did the Crash Dump Come from?.....	363
FinalExceptionHandler	364
PART 11: Memory Visualization.....	367
The Art of Memory Corruption	367

Visualizing Secondary Storage	368
Pictures from Memory Space.....	369
PART 12: Miscellaneous	375
Hexadecimal / Decimal chaos	375
The Measure of Debugging and Memory Dump Analysis Complexity.....	376
How To Simulate a Process Hang?	377
A Windows Case for Delta Debugging.....	378
Sentinel Pointers	380
Collapsed Stack Trace.....	381
Appendix A.....	383
Crash Dump File Examples	383
Appendix B	385
Crash Dump Analysis Checklist.....	385
Appendix C	387
Memory Dump Analysis Pattern: A Definition	387
Wait Chain Patterns	387
DLL Link Patterns.....	387
Insufficient Memory Patterns	388
Dynamic Memory Corruption Patterns.....	388
Deadlock Patterns	388

Index	389
Notes.....	397
About the Author	402
Cover Images.....	403

Index

- !
- !address, 22, 23, 166
 - !alpc, 52
 - !analyze -v, 56, 72, 78, 85, 144, 149, 163, 224, 227, 236, 241, 269, 270, 275, 288, 376, 385
 - !apc, 386
 - !bugdump, 386
 - !chkimg, 153, 197, 248, 385
 - !cs, 54, 171, 172, 233, 237, 240, 261, 385
 - !dh, 38, 234, 253, 385
 - !dlls, 276
 - !dpcs, 386
 - !envvar, 385
 - !error, 143
 - !exchain, 385
 - !exqueue, 386
 - !fileobj, 243
 - !gflag, 166, 385
 - !handle, 94, 162
 - !heap, 181
 - !irp, 57, 59, 243
 - !irpfind, 386
 - !locks, 22, 24, 54, 85, 86, 105, 107, 120, 141, 167, 189, 212, 217, 226, 262, 313, 385, 386
 - !ipc, 52, 98, 101, 118, 171, 173, 177, 178, 179, 229, 282, 386
 - !ntsdexts.locks, 386
 - !pcr, 145
 - !peb, 23, 385
 - !pool, 386
 - !poolused, 175, 293, 386
 - !process, 28, 43, 44, 48, 52, 61, 97, 106, 113, 116, 120, 146, 176, 211, 214, 230, 242, 262, 279, 285, 386
 - !pte, 22, 23, 77, 80, 289, 290
 - !qlocks, 386
 - !ready, 111, 213, 386
 - !runaway, 63, 64, 185, 385
 - !running, 79, 111, 138, 213, 215, 262, 386
 - !session, 107, 216, 386
 - !sprocess, 210, 292, 386
 - !stacks, 24, 244, 262, 386
 - !teb, 46, 62, 73, 143, 157, 198, 231, 246, 250, 258, 276, 364
 - !thread, 79, 88, 89, 92, 93, 99, 100, 101, 102, 104, 109, 110, 111, 118, 119, 122, 123, 124, 125, 126, 127, 138, 139, 150, 169, 171, 172, 173, 174, 176, 177, 178, 179, 180, 203, 204, 215, 216, 221, 222, 228, 229, 231, 244, 282, 283, 284
 - !vm, 21, 60, 105, 112, 152, 175, 209, 286, 291, 386
 - .
 - .asm, 57, 133, 264
 - .bugcheck, 104
 - .cxr, 76, 158, 165, 233, 238, 246, 247, 272, 361
 - .dump, 195, 378
 - .effmach, 160, 269
 - .enumtag, 386
 - .expr, 133, 165, 272
 - .kframes, 224, 385
 - .load, 45, 160, 269
 - .NET Leak pattern, 60
 - .NET runtime, 132
 - .process, 23, 28, 44, 115, 152, 172, 214, 230, 262, 287
 - .reload, 46
 - .thread, 43, 45, 79, 117, 158, 165, 214, 231, 233, 238, 247, 262
 - .trap, 56, 77, 78, 80, 144, 288, 290

_FILE_OBJECT, 59
 _IO_STACK_LOCATION, 57, 58

~

~#s, 157
 ~*kb, 225
 ~*kc, 270
 ~*kv, 385
 ~~ , 142, 189

A

abstract materialism, 303
 abstract space of states, 297
 Advanced Local Procedure Calls (ALPC), 52
 AeDebug, 84, 356, 361
 alternative branches of computation, 299
 analysis region, 346
 Anaximander, 303
 Anaximenes, 303
 apeiron, 303
 Application Verifier, 166, 238, 255
 ASLR, 71
 astrology, 320, 321

B

backward disassembly of the return
 address, 264
 basic chemical formula representation, 310
 basic facts, 345, 347
 Beethoven, 315
 Bernhard Riemann, 299
 blocked thread, 54, 69, 141, 228, 245
 Blocked Thread pattern, 91, 97, 120, 123,
 201, 216, 279
 bug hauntings, 329
 bugtanglement, 306
 bugteriology, 318

bugterium, 318
 build dates, 320
 build time, 320
 Busy System pattern, 118, 212, 215

C

CAD diagram, 256
 CDB, 356
 CDF tracing, 341
 Changed Environment pattern, 196, 248
 checklist, 385
 circularity, 346
 Citrix terminal services environment, 310,
 341
 classical memuonics, 307
 code review debates, 329
 collapsed stack traces, 69
 COM/OLE, 208
 comparative analysis, 63
 complete memory dumps, 22, 24, 28, 43,
 49, 54, 56, 92, 118, 119, 120, 139, 144,
 149, 152, 170, 175, 209, 228, 262, 279,
 314, 336, 339, 368
 complexity measure, 376
 computational ghosts, 329
 computational precognition, 322
 computer name, 386
 copernican revolutions, 305
 copying a buffer, 288
 corrupt dumps, 144
 corruption signs, 260
 Coupled Process pattern, 161, 228
 CPU spikes, 63, 138, 185, 262
 Crash2Hang, 356, 357, 358, 360, 361, 362
 critical section deadlock, 220, 226
 critical section list, 195, 236, 260
 critical section owners, 54
 critical sections, 54, 85, 189, 226, 238, 240,
 261, 279, 387, 388
 CtxHideEx32, 351, 352

D

da, 214, 231, 260
 da /c, 214, 231
 Data Alignment pattern, 78
 DATA, the meaning, 293, 330, 339
 dc, 191, 278, 323
 dd, 93, 161, 162, 166, 208, 323, 376
 ddp, 208
 dds, 39, 47, 73, 80, 143, 147, 199, 232, 246,
 250, 258, 264, 265, 364
 deadlock, 24, 85, 180, 220, 223
 Debugged! MZ/PE, 54
 default hang analysis command, 227
 definition of software defects, 306
 demand zero pages, 77, 289
 device driver stack, 263
 device stack, 129
 differential memory analysis, 293, 305
 disjoint notification events, 93
 division by zero, 96
 DLL Art, 367
 DLL hooks, 196, 200
 double fault, 144, 145, 148
 double traps, 104
 dpa, 277
 dps, 62, 267
 dpv, 235, 277
 dq, 22, 159
 dqs, 157
 ds, 386
 dt, 58, 59, 87, 94, 146
 du, 22, 117, 240, 278
 Dump2Picture, 368, 401
 Duplicated Module pattern, 383

E

early crash dumps, 141
 English verse, 324
 ERESOURCE, 24, 85, 216, 386
 Ernst Mayr, 302
 ETW traces, 48, 343
 Event Tracing for Windows, 308

evolutionary causation, 302
 exception processing signs, 270
 EXCEPTION_DOUBLEFAULT, 144
 Execution Residue pattern, 47, 54, 147, 148,
 234, 238, 245, 258, 265, 276
 executive resource locks, 85
 executive resource objects, 24
 executive resources, 85, 120, 387, 388

F

fault message, 351
 faultomorphism, 75
 faultrep.dll, 238, 245
 filter functions, 129
 full page heap, 166, 225, 238, 255
 functional causation, 302

G

GDI resources, 117
 genotype, 297
 Gflags, 255, 385
 golden rules of troubleshooting, 335
 GUI thread, 161
 Guinness effect, 333

H

handle leak, 176, 292, 388
 Handle Leak pattern, 61, 117, 175, 209, 383
 -hang, 227, 236, 385, 386
 hang simulation, 205
 hard error, 238
 heap corruption, 63, 166, 181, 225, 238,
 255, 313, 388
 Heap Corruption pattern, 246
 Heap Leak pattern, 60, 63, 181
 Heraclitus, 303
 historical information, 54, 245, 343
 historical narratives, 302
 Hooked Function pattern, 196, 248
 Hooked Functions pattern, 152

hookware, 239, 310

I

ImageMagick, 368

inconsistent memory dumps, 24, 175, 279, 283, 309, 353, 385

incorrect stack trace, 54, 72, 231, 247, 248, 272, 283, 289

incorrect stack traces, 72

Infinite Memory, 296

Insufficient Memory pattern, 175

insufficient session pool memory, 209

invalid code, 264

invalid code pointer, 264

Invalid Pointer pattern, 131, 165, 288, 380

Ionian school, 303

Iridium, 335

IRQL, 77, 149, 353

K

k 100, 196, 205

kernel stack overflow, 145

KERNELBASE.dll, 28, 29, 31, 33, 34, 38, 40, 41

Kitaro, 314

kL, 49, 74, 75, 132, 141, 142, 155, 158, 160, 163, 238, 245, 247, 255, 262, 264, 273, 359, 361

knf, 148

kv, 47, 50, 57, 62, 64, 117, 133, 142, 144, 145, 160, 161, 165, 189, 207, 214, 233, 235, 248, 290, 376, 380

L

Lateral Damage pattern, 144

list of unloaded modules, 238

listing all threads, 225

ListProcessStacks, 386

live debugging, 305

LiveKd, 353

lm, 191, 245, 253

lmt, 142, 238, 247, 256

lmv, 37, 55, 110, 150, 162, 163, 190, 198, 233, 234, 274, 287, 320

ln, 71, 231, 376, 386

locked sections, 236

Lost Opportunity pattern, 142

M

main application thread, 92, 385

Main Thread pattern, 241, 263, 275, 360, 361

Managed Code Exception pattern, 84

manual dump, 83, 85, 142, 160

manual kernel dump, 241

manual memory dumps, 63, 78, 143, 149, 224, 336

manual user dump, 236, 268

mathematical definition of a memory

dump, 295

measurable values, 297

memoidealism, 303, 307

memophysical principle, 307

memory leak, 60, 181

memory religion, 304

memuonics, 307

memuons, 307

Message Box pattern, 54, 143, 229, 230, 238, 240, 336, 351, 356, 361

Milesian philosophers, 304

minidumps, 308

Missing Component pattern, 383

missing DLL, 275

mixed objects, 85, 388

modeling, 297, 298

Module Variety pattern, 191, 255

MTCrash, 358, 359, 360, 361, 362

Multiple Exceptions pattern, 77, 78, 96, 157, 164, 225, 234, 245, 272, 346, 358

multiple probabilistic explanations, 302

multi-threaded environments, 55, 255

multi-user terminal service environments,

255

multivalued functions, 299
 mutant, 86, 87, 88, 91, 99, 123, 173, 179,
 283

N

narratology, 342
 natural memory dumps, 295
 natural system, 297
 NDB (Nature Data Base) files, 295
 nested function calls, 265
NMI_HARDWARE_FAILURE, 85, 241
 no cloning theorem, 309
 Not My Version pattern, 142, 150, 166
 NULL code pointer, 131, 158, 264
 NULL Code Pointer pattern, 196
 NULL pointer access violation, 248
 NULL pointers, 56, 380

O

observables, 295, 297, 298
 occult, 322
 Optimized Code pattern, 271
 OSMOSIS, 353
 Overaged System pattern, 180
 over-aged systems, 118
 ownership semantics, 87, 92

P

page boundary, 77, 78, 80, 165
 page fault, 78, 79, 80, 83, 104, 105, 203
 page file thrashing, 104
 paranormal, 322
 partial stack traces, 47, 147, 148
 pass through functions, 244
 passive system threads, 129
 Passive Thread pattern, 129
 patched functions, 153, 197, 248, 267
 PE headers, 234
 periodic errors, 344
 permanent primary element, 303

phenotypes, 298
 political dumps, 331
 pool allocation failures, 175
 premature crash dump, 241
 printer drivers, 234, 247, 255
 problem resolution, 327, 336, 338, 341
 process crash dumps, 96
 Process Factory pattern, 112, 286
 process memory dump, 269, 275
 proximate causation, 302
 psi-computation, 322

Q

quantum computation, 309
 quantum computer simulators, 309
 quantum information, 309
 quantum mechanics, 309
 quantum memoretics, 309
 quantum memory, 307, 309

R

r, 23, 28, 44, 58, 62, 72, 75, 80, 115, 131,
 152, 172, 191, 196, 214, 230, 248, 262,
 278, 287, 380
 r8d register, 50
 raising a status, 275
 raw stack, 47, 62, 72, 80, 147, 157, 198,
 207, 231, 234, 245, 249, 260, 264, 265,
 276, 277, 385
 raw stack range, 276
 RDR_FILE_SYSTEM, 76
 ready state, 105
 ready threads, 386
 relics, 332
 residual stack trace, 247
 resource contention, 24, 212, 262
 Riemann surfaces, 299
 running threads, 386

S

science files, 301, 327
 Self-Diagnostic Message pattern, 240
 semantic inconsistency, 72
 SendReceive2, 206, 207
 shortage of paged pool, 290
 shortage of session pool, 290
 software memory dumps, 295
 software trace, 308, 341
 space of abstract memory dumps, 298
 spiking thread, 69, 70
 Spiking Thread pattern, 78, 138, 149, 167, 168, 216
 spinlocks, 386
 stack overflow, 148
 Stack Overflow pattern, 51, 78
 Stack Trace Collection pattern, 27, 129, 155, 163, 241
 stack trace pattern, 189, 264
 stack trace reconstruction, 248, 264
 STAX, 316
 structurally and semantically correct stack trace, 264
 supernatural, 322
 Supertramp, 313
 Swarm of Shared Locks pattern, 201
 synthesized dump analysis, 255
 SystemDump, 28
 systems theory, 297

T

Task Manager, 170, 181, 205, 208, 268, 336
 terminal services environment, 345
 TestDefaultDebugger64, 351
 Thales, 303
 thread waiting time, 230
 time arrow, 296
 trace file size, 346
 trace message format files, 308
 trace-based debugging, 305
 truncated complete dumps, 22
 Truncated Dump pattern, 151

truncated memory dumps, 22, 24, 85, 151, 386
 TSS, 145

U

u, 57, 67, 133, 134, 153, 154, 191, 197, 198, 199, 249, 251, 253, 259, 260, 266, 267, 289, 380
 ub, 50, 51, 67, 131, 134, 135, 158, 199, 251, 252, 259, 265, 289
 ultimate causation, 302
 UNEXPECTED_KERNEL_MODE_TRAP, 144
 unhandled exception, 84, 132, 163, 225, 237, 357, 360
 unique computational event, 302
 unity in difference, 303
 universal memory dump, 295, 298, 300, 301, 303, 305, 327, 332, 337
 universal symbol files, 295
 Universe, 296
 Unknown Module pattern, 234, 310
 unresponsive process, 268
 Urstoff, 303, 304

V

version, 28, 37, 38, 55, 94, 131, 142, 150, 166, 180, 190, 194, 195, 196, 241, 247, 253, 274, 275, 316, 317, 385
 virtual address space, 21
 virtualized process, 160, 161
 Visual Dump Objects (VDO), 68

W

W2K, 105, 386
 wait chain, 54, 89, 95, 102, 118, 128, 167, 168, 177, 229
 Wait Chain pattern, 92
 wait chains, 54, 69, 97, 120, 216, 223, 227, 236, 281
 waiting threads, 386

waiting time, 204, 262
wcscpy, 71
WDK, 50, 57
WER dialog, 268, 269, 351, 352
WerFault.exe, 352
Whitehead, 318
wild code pointer, 272
Win32DD, 353
WindowHistory, 347
WOW64, 45, 160, 174
Wrong Dump anti-pattern, 139

x64 Windows, 21, 43, 49, 269
x86 context, 45
x86 mode, 269

Y

Yanni, 317

Z

Zombie Processes pattern, 117

X

x64 calling convention, 50

Memory Dump Analysis Anthology

Volume 4

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2010 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-906717-86-5 (Paperback)

ISBN-13: 978-1-906717-87-2 (Hardback)

First printing, 2010

Revision 2 (June, 2015)

Contents

Preface	17
Acknowledgements.....	19
PART 1: Professional Crash Dump Analysis and Debugging.....	21
Common Mistakes	21
Not Using Checklists.....	21
Not Paying Attention to All Aspects of Default Analysis	23
Not Paying Attention to Context.....	26
Raw Stack Dump of WOW64 Process	31
On Space and Mode	35
Registry Corruption: A Case Study	36
Wild Code and Partial Stack Reconstruction.....	39
Manual Parameter Reconstruction on x64 Windows Systems	42
Counterfactual Debugging	46
Dereference Fixpoints	46
Data Ordering.....	48
Clean Raw Stack Execution Residue.....	64
Essential and Derived Properties	71
Software Defect Researcher: A New Profession	74

WinDbg Shortcuts	75
!imu and !mk	75
.opendump.....	80
Live Kernel Debugging of System Freeze	82
Mode-Independent WinDbg Scripts	91
PART 2: Crash Dump Analysis Patterns	93
Succession of Patterns	93
Ubiquitous Component.....	94
Nested Offender	120
Hunting for a Driver	124
Virtualized System.....	131
Effect Component	137
Well-Tested Function	144
Mixed Exception.....	145
Random Object	150
Not My Version (Hardware)	153
Missing Process	154
Platform-Specific Debugger	156
Value Deviation (Stack Trace)	159
CLR Thread	163

Insufficient Memory (Control Blocks)	166
PART 3: Crash Dump Analysis AntiPatterns	167
Habitual Reply	167
Part 4: Pattern Interaction	169
Null Data Pointer, Pass-Through Functions, and Platformmorphic Fault	169
Stack Trace Collection, Message Box, Hidden Exception, Nested Offender, Insufficient Memory, C++ Exception, Heap Leak and Ubiquitous Component	172
Blocked LPC Thread, Coupled Processes, Stack Trace Collection and Blocked GUI Thread	181
Virtualized Process, Incorrect Stack Trace, Stack Trace Collection, Multiple Exceptions, Optimized Code and C++ Exception.....	182
WOW64 Process, NULL Data Pointer, Stack Overflow, Main Thread, Incorrect Stack Trace, Nested Exceptions, Hidden Exception, Manual Dump, Multiple Exceptions and Virtualized System.....	189
NULL Data Pointer, Stack Trace, Inline Function Optimization and Platformmorphic Fault	201
Stack Trace Collection, Suspended Threads, Not My Version, Special Process, Main Thread and Blocked LPC Chain Threads	204
Truncated Dump, Stack Trace Collection, Waiting Thread Time and Wait Chains ...	212
ALPC Wait Chain, Missing Threads, Message Box, Zombie and Special Processes ...	214
CriticalSection High Contention and Wait Chains, Blocked Threads and Periodic Error: Memory Dump and Trace Analysis Pattern Cooperation	220
Statement Current, Coupled Processes, Wait Chain, Spiking Thread, Hidden Exception, Message Box and Not My Version	223

Stack Trace Collection, Missing Threads, Waiting Thread Time, Critical Section and LPC Wait Chains	226
Wait Chain, Blocked Thread, Waiting Thread Time, IRP Distribution Anomaly and Stack Trace Collection	231
PART 5: A Bit of Science and Philosophy.....	235
Memory Exponentiation (PowerSet)	235
Memory Dump View of Artificial Intelligence	236
Memoidealism as Monistic Aspect Pluralism.....	237
Memory Dumps as Posets.....	239
Metaphorical Bijectionism: A Method of Inquiry.....	241
Notes on Memoidealism	246
Panmemorism	247
Cubic Memory Representation	248
Manifold Memory Space.....	250
Ars Recordatio.....	252
Categories for the Working Software Defect Researcher	253
MemD Category	253
Operating Closure of Memory	256
Memoidealism Defined	258
Memuon: A Definition	259
PART 6: Fun with Crash Dumps.....	261

Music for Debugging	261
THE ALL MIGHTY DEBUGGER	261
Memory Space Music.....	262
The Duet of Threads.....	263
The Memory Dump of the Dead	264
Ancient Computations and a Vision of the New Dump	265
The Meaning of DUMP	266
Memory Analysis Ritual	267
The Intelligent Memory Movement.....	268
Moving towards the Psi Point	269
Experiments on Poor Bugs	270
Exception Processing Of Crash Hypothesis (EPOCH).....	271
Debugging Slang.....	272
SAD Events	272
BoBo Address	273
Mad Day	274
Bug-sistential and Bug-sistentialism	275
Debugging Spy Network.....	276
Games for Debugging: Go	277
The Tsar of Memory Dump Analysis	278

DNA and RNA of Ruptured Computation	279
BADOBODO Address: Childhood Memories.....	280
Bugs in Passing	281
Named Process: Vostokov.exe.....	283
Memory Analysts and Debuggers Day	286
After Volume 3	287
Crash, Core and Memory Dumps in Science Fiction and Fantasy	288
Reasoning with a Bug	301
PART 7: Software Troubleshooting	303
RADII and SDSD	303
Epistemic Troubleshooting and Debugging	304
RADII Process Illustrated	305
Debugware Patterns	307
Trace Expert	307
Troubleshooting Unit of Work	308
Checklist	309
Supporting Module	310
Span Differentiator	311
Self-Extractor	312
A Case Study.....	314

Can Software Tweet?	319
The Law of Simple Tools.....	320
Workaround Patterns	321
Hidden Output	321
Frozen Process	324
Axed Code	325
PART 8: Software Trace Analysis.....	327
CDFAnalyzer for Analysis of CDF (ETW) Traces	327
There ought to be a Planet at that Location!.....	328
Software Trace: Bird's Eye View.....	329
Extending Multithreading to Multibraiding (Adjoint Threading)	330
PART 9: Software Trace Analysis Patterns	335
Statement Density and Current	335
Exception Stack Trace	337
Thread of Activity.....	339
Discontinuity	341
Missing Component	342
Bifurcation Point	343
Characteristic Message Block.....	345
Activity Region	348

Vocabulary Index.....	349
Inter-Correlation	350
PART 10: The Origin of Crash Dumps	353
Full Page Heap Settings on x64 Windows	353
Memory Dumps from Hyper-Virtualized Windows	354
Fiber Bundle of Memory Space.....	357
On Self Dumps of Secure String API	358
PART 11: Memory Visualization.....	361
Pictures from Memory Space.....	361
Large-scale Structure of Memory Space	363
Advanced Memory Visualization	365
3D Memory Visualization	376
Memory Map Visualization Tools	389
PART 12: Art	391
Opcodism: The Art of Opcodes	391
Memory Dump and Minidumps.....	394
Hot Issues from Physicalist Artist Perspective	395
Memory Dumps from Physicalist Artist Perspective.....	396
Memory Hot Spot and the Illusion of Fix	397
Shared Section	398

Memory Space Road to the Ultimate Fix	399
Structure and Noise	400
PART 13: Miscellaneous	401
Assembling Code in WinDbg	401
Free Stack Traces	403
Stack Space and Program Database Types.....	405
The Longest Stack Trace.....	409
Software Victimology	414
Debugger as a Shut up Application	415
2 Great Windows Software Engineering Magazines	416
Appendix.....	417
Crash Dump Analysis Checklist.....	417
Index of WinDbg Commands	419
Notes.....	421
About the Author	422
Cover Images.....	423

Index of WinDbg Commands

!

!alpc, 215, 216, 418
!analyze, 418
!analyze -v, 23, 26, 27, 30, 36, 84, 125, 131, 145, 156, 157, 169, 172, 182, 183, 197, 201, 267, 417
!analyze -v -hang, 417
!apc, 418
!bugdump, 418
!chkimg, 23, 417
!cs, 220, 229, 417, 418
!devobj, 126, 128
!devstack, 130
!dh, 417
!dlls, 417
!dpcs, 418
!dumpobj, 157
!dumpstack, 27, 30, 156, 157
!envvar, 417
!error, 86, 223, 358
!exchain, 123, 148, 417
!exqueue, 418
!filecache, 166, 418
!fileobj, 71, 129
!for_each_thread, 91
!gflag, 417
!handle, 71
!heap, 173, 178, 179
!irp, 126, 128
!irpfind, 232, 418
!locks, 21, 231, 417, 418
!ipc, 208, 209, 230, 418
!ntsdeps.locks, 418
!pe, 145, 157
!peb, 417
!pool, 151, 418
!poolused, 418
!
!process, 75, 154, 205, 206, 207, 208, 212, 214, 226, 228, 230, 283, 418
!pte, 39
!qlocks, 418
!ready, 418
!runaway, 64, 160, 224, 239, 409, 417
!running, 133, 418
!session, 418
!sprocess, 418
!stacks, 150, 212, 226, 233, 418
!sysinfo, 153, 355, 417, 418
!teb, 31, 32, 48, 121, 146, 161, 190, 224
!thread, 37, 39, 91, 133, 137, 150, 151, 198, 216, 217, 229, 230, 231, 234
!uniqstack, 417
!vm, 154, 198, 212, 214, 218, 283, 418
!whattime, 233
!wow64exts.info, 33, 65, 67, 186, 193
.
.asm, 44, 121, 126, 132, 148, 187
.cxr, 39, 44, 86, 87, 122, 125, 147, 186, 187, 197, 205, 220, 221
.effmach, 32, 34, 182, 195
.enumtag, 418
.expr, 185, 358
.exr, 84, 122, 125, 145, 156, 182, 186, 187, 197, 199, 205, 206
.kframes, 91, 417
.load, 27, 182
.opendump, 8, 80
.process, 204, 220, 228, 284
.reboot, 87
.symfix, 417
.thread, 39, 44, 87, 122, 147, 187, 205, 220, 221, 229
.trap, 25, 39, 201, 202, 205, 206

|

dS, 418
dt, 72, 128, 129, 151, 152, 221

||, 80

~

K

~*k, 184
~*kc, 94, 191
~~, 22

B

kc, 22, 169, 170, 189, 191, 197
kcf, 190
kL, 27, 65, 68, 81, 120, 122, 146, 147, 156,
159, 163, 167, 201, 202, 220, 225, 325,
360, 391, 392, 409
kv, 36, 38, 43, 44, 87, 91, 132, 185, 187,
205, 221, 358, 404, 417

bc, 62, 326
bm, 325
bp, 42, 53, 56, 70, 326

D

da, 285
dds, 34, 40, 66, 67, 69, 70, 121, 133, 161,
194, 224
dp, 32, 34, 53, 56, 57, 60, 62, 127, 128, 132,
151
dps, 33, 34, 37, 137, 147, 186, 188
dpu, 38
dqs, 34, 195, 199

L

lm, 200, 354
lmk, 8, 75, 76
lmu, 8, 75, 76
lmv, 75, 206, 225, 284, 417
ln, 418

U

ub, 45, 121, 122, 126, 160, 162, 187, 193,
359, 391, 392
uf, 42, 49, 53, 127, 132, 148, 359, 401, 405,
406, 407, 408

Memory Dump Analysis Anthology

Volume 5

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2011 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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.

ISBN-13: 978-1-906717-96-4 (Paperback)

ISBN-13: 978-1-906717-97-1 (Hardback)

First printing, 2011

Revision 2 (June 2015)

Contents

Preface	17
Acknowledgements.....	19
PART 1: Professional Crash Dump Analysis and Debugging.....	21
Common Mistakes	21
Not Double-Checking Symbolic Output	21
Not Looking Past the First Found Evidence.....	24
Not Recognizing Data as UNICODE or ASCII Fragments	26
Common Questions.....	28
What Service is this?	28
Complete Stack Traces from x64 System	30
Software Behavior Patterns	32
Crash and Hang Analysis Audit Service	33
Case Study: Extremely Inconsistent Dump and CPU Spike	34
Raw Stack Dump of All Thread Stacks	39
Architecture of CARE.....	41
PART 2: Crash Dump Analysis Patterns.....	43
Succession of Patterns	43
Wait Chain (Process Objects)	49

Coincidental Frames.....	55
Fault Context.....	59
Coupled Processes (Weak).....	60
Hooked Functions (Kernel Space)	63
Hardware Activity.....	66
Incorrect Symbolic Information	71
Message Hooks	76
Blocked Thread (Hardware)	79
Coupled Machines.....	81
High Contention (Processors)	82
Thread Starvation (Normal Priority)	85
Coupled Processes (Semantics).....	87
Abridged Dump	88
Exception Stack Trace	93
Wait Chain (RPC)	95
Distributed Spike.....	99
Instrumentation Information.....	108
Template Module.....	112
Invalid Exception Information.....	116
Shared Buffer Overwrite	120

Pervasive System.....	125
Problem Exception Handler	126
Deadlock (Self)	127
Same Vendor.....	128
PART 3: Crash Dump Analysis AntiPatterns	129
Wild Explanations.....	129
PART 4: Pattern Interaction	133
Inconsistent Dump, Stack Trace Collection, LPC, Thread, Process, Executive Resource Wait Chains, Missing Threads and Waiting Thread Time.....	133
Fault Context, Wild Code, and Hardware Error	137
Main Thread, Critical Section Wait Chains, Critical Section Deadlock, Stack Trace Collection, Execution Residue, Data Contents Locality, Self-Diagnosis and Not My Version	145
Strong Process Coupling, Stack Trace Collection, Critical Section Corruption and Wait Chains, Message Box, Self-Diagnosis, Hidden Exception and Dynamic Memory Corruption.....	158
IRP Distribution Anomaly, Inconsistent Dump, Execution Residue, Hardware Activity, Coincidental Symbolic Information, Not My Version, Virtualized System	169
Spiking Thread, Main Thread, Message Hooks, Hooked Functions, Semantic Split, Coincidental Symbolic Information and Not My Version.....	180
Stack Trace Collection, Special Process, LPC and Critical Section Wait Chains, Blocked Thread, Coupled Machines, Thread Waiting Time and IRP Distribution Anomaly....	188
ALPC Wait Chains, Missing Threads, Waiting Thread Time and Semantic Process Coupling	200
Insufficient Kernel Pool Memory, Spiking Thread, and Data Contents Locality.....	201

Incorrect Stack Trace, Stack Overflow, Early Crash Dump, Nested Exception, Problem Exception Handler and Same Vendor	206
PART 5: A Bit of Science and Philosophy.....	213
Memory Systems Language	213
Categories for the Working Software Defect Researcher	214
Collective Pointer	214
Notes on Memoidealism	217
Archaeological Foundations for Memory Analysis.....	218
On God and Miracles.....	220
Psychoanalysis of Software Troubleshooting and Debugging	221
Ontological and Epistemological Memoidealism	222
On Unconscious	223
Ruminations on Automated Debugging.....	224
General Memory Analysis	225
Notation for Memory and Trace Analysis	226
Category Theory and Troubleshooting	227
Software Chorography and Chorology: A Definition.....	229
PART 6: Fun with Crash Dumps.....	231
Music for Debugging	231
Retry, Abort, Escape.....	231
Debugging Slang.....	232

STUPID.....	232
On the Same Page	233
.SYS.....	234
PLOT	235
Freedom.....	236
Free Verse	237
BCE, BC, and CE	238
HCI.....	239
Blog	240
Inherit a Fortune	241
Dr. Watson's Observational Patterns.....	242
Memory Dumps in Myths	245
Bus Debugging.....	246
Debugging the Debugger (16-bit)	247
Dr. DebugLove and Nature.....	249
Sailing Memory Spaces under an RGB Flag	253
Don't Name Your Driver a "Missile"	254
Notepad Debugging	255
!analyze -vostokov	263
Contemplating Crash Dumps in Unicode	264

Memory Dump Analysis Services Cap and T-Shirt	266
Troubleshooting Poem in Six Stanzas	267
On the Interpretation of M-Theory.....	268
Check the Name of Your Driver in Reverse.....	269
PART 7: Software Trace Analysis.....	271
Pattern Interaction.....	271
Adjoint Threads, Discontinuity, and Time Delta	271
Basic Software PLOTS	272
Two Readings of a Software Trace.....	274
CDFMarker Tool	276
The Extended Software Trace	277
Presenting a Software Story.....	278
Adjoint Threading in Process Monitor	279
PART 8: Software Trace Analysis Patterns	281
Significant Event.....	281
Time Delta	282
Adjoint Thread of Activity	283
Trace Acceleration	284
Incomplete History.....	286
Background and Foreground Components	287

Defamiliarizing Effect	290
Anchor Messages	293
No Trace Metafile	296
No Activity	297
Trace Partition	299
Truncated Trace	301
Diegetic Messages	302
False Positive Error	303
Guest Component	304
Message Change	305
Layered Periodization	306
PART 9: Models of Software Behaviour	311
Multiple Exceptions Pattern	311
Memory Leak (Process Heap) Pattern	315
Message Hooks Pattern	326
Modeling C++ Object Corruption	330
PART 10: The Origin of Crash Dumps	335
More on Demystifying First-chance Exceptions	335
PART 11: Structural Memory Patterns	343
Memory Snapshot	343

Aggregate Snapshot	345
Snapshot Collection	346
Memory Region.....	347
Region Boundary.....	348
Memory Hierarchy	350
Anchor Region.....	351
PART 12: Memory Visualization.....	353
Memory Map Visualization Tools (Revised).....	353
Decomposing Memory Dumps via DumpFilter	355
Can a Memory Dump be Blue?	359
Virtual to Physical Memory Mapping.....	360
The Memory Visualization Question	363
PART 13: Art	375
Sweet Oil of Memory	375
Night Sky	376
Component Trace.....	377
Ana-Trace-Log-Lyzer and Closed Session	378
Computer Memory Gardens	380
Debugging Venue	381
Inside a Memory File.....	382

Fabric of Memory Dumps	383
Race Condition in a Kernel Pool	394
Memory Interfaces.....	395
Bleeding Memory.....	396
Picture Frame for Memory Dumps	398
Front Cover Glitch	399
Chance Exceptions in a Turing Machine.....	400
PART 14: Security and Malware Analysis	401
Crash Dumps and Password Exposure	401
Crash Dump Analysis of Defective Malware	406
PART 15: Miscellaneous	411
Native Script Debugging	411
Component Heap	414
Attached Processes	416
User/Kernel Diagramming Styles	419
Appendix	423
Contention Patterns.....	423
Raw Stack Analysis Scripts	424
Crash Dump Analysis Checklist.....	425
Index of WinDbg Commands	427

Notes.....	429
About the Author	430
Cover Images.....	431

Index of WinDbg Commands

! <i>!address</i> , 317, 363, 372 <i>!alpc</i> , 50, 52 <i>!analyze</i> , 11, 21, 59, 93, 116, 120, 122, 137, 138, 139, 140, 141, 142, 145, 158, 206, 210, 263, 312 <i>!avrf</i> , 109 <i>!chkimg</i> , 63, 184, 185, 186 <i>!cmkd</i> , 348 <i>!cs</i> , 157, 159, 192, 193, 298 <i>!devobj</i> , 177 <i>!devstack</i> , 177 <i>!dh</i> , 254, 408 <i>!dpcs</i> , 176 <i>!exchain</i> , 126 <i>!fileobj</i> , 29, 198 <i>!for_each_thread</i> , 30 <i>!gflag</i> , 109, 110 <i>!heap</i> , 316, 321, 332 <i>!irp</i> , 25, 29, 177, 198 <i>!irpfind</i> , 169, 198 <i>!imi</i> , 115 <i>!locks</i> , 43, 48, 135 <i>!ipc</i> , 133, 189, 191 <i>!pool</i> , 121, 123, 204 <i>!poolused</i> , 202 <i>!process</i> , 28, 34, 53, 71, 73, 188, 197 <i>!pte</i> , 348 <i>!ready</i> , 35, 82, 85 <i>!runaway</i> , 88, 99, 104, 105, 180 <i>!running</i> , 34, 35, 47, 82, 85, 170, 202 <i>!stacks</i> , 48, 169 <i>!sysinfo</i> , 179 <i>!teb</i> , 39, 76, 152, 154, 165, 181, 207, 327, 338 <i>!thread</i> , 30, 36, 48, 56, 66, 80, 82, 83, 85, 88, 135, 169, 170, 190, 192, 197, 202, 203, 348, 349, 416, 417	! <i>!verifier</i> , 108 <i>!vm</i> , 201 <i>!wow64exts</i> , 39, 40 \$ <i>\$\$</i> , 365, 372, 412 . <i>.asm</i> , 103, 328, 331 <i>.cxr</i> , 116, 119, 120, 122, 167, 210, 330, 331 <i>.ecxr</i> , 118, 331, 338 <i>.effmach</i> , 30, 40 <i>.expr</i> , 94, 150, 314 <i>.exr</i> , 58, 116, 120, 122, 150, 158, 206, 312 <i>.formats</i> , 75, 123 <i>.frame</i> , 332, 411, 412 <i>.imgscan</i> , 408, 409 <i>.load</i> , 30, 39, 40 <i>.opendump</i> , 208, 210 <i>.process</i> , 30, 31, 71, 73, 192 <i>.reload</i> , 30, 72, 326 <i>.symfix</i> , 326 <i>.thread</i> , 30, 84, 119, 193, 194, 195, 196, 203, 330, 331 ~ <i>~*e</i> , 40 <i>~*kn</i> , 411 <i>~~</i> , 155, 298 <i>~0s</i> , 327	D <i>da</i> , 165, 264, 405
--	---	---------------------------------------

db, 121, 123, 264, 408

dc, 154, 155, 369

dd, 40, 91, 118, 332

dds, 26, 40, 57, 64, 66, 153, 154, 166

dp, 347, 407

dps, 76, 89, 170, 181, 204, 207, 318, 324,
327, 333, 405

dpu, 412

dqs, 322, 338

dt, 84, 332, 333

du, 22, 27, 81, 196, 264, 405

dv, 332

G

g, 28, 193, 209, 210, 248, 335, 405

K

k, 91, 152, 159, 316, 321, 326, 327, 330,
331, 408

kb, 30, 411, 412

kL, 55, 74, 78, 79, 110, 111, 145, 156, 164,
167, 180, 206, 209, 211, 262, 297, 313,
318, 401, 402, 406

kv, 81, 84, 93, 118, 119, 126, 149, 157, 165,
195, 196, 203, 210, 314

L

lm, 22, 27, 205, 363

lmt, 73, 113, 168

lmu, 72

lmv, 113, 114, 125, 128, 157, 183, 254

R

r, 30, 31, 40, 73, 79, 155, 192, 203, 207

U

u, 22, 23, 27, 58, 65, 103, 138, 141, 143,
178, 179, 183, 185, 186, 187, 247, 332,
335, 407

ub, 27, 55, 56, 69, 70, 77, 80, 104, 121, 139,
143, 176, 178, 182, 183, 184, 187, 324,
329, 331, 332, 404, 407

uf, 138, 139, 143

V

version, 80

Memory Dump Analysis Anthology

Volume 6

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2013 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-19-1 (Paperback)

ISBN-13: 978-1-908043-20-7 (Hardback)

First printing, 2013

Revision 2 (July 2015)

Contents

Preface	15
Acknowledgements.....	17
PART 1: Professional Crash Dump Analysis and Debugging.....	19
Memory Dump Analysis Best Practices.....	19
Windows Debugging Expert System WinDbg Extension	20
Common Mistakes	21
Not Comparing to Reference Debugger Output	21
From Bugchecks to Patterns	23
Raw Stack from Laterally Damaged Memory Dumps	24
WinDbg Tips and Tricks: Getting the Bottom of a Stack Trace	26
PART 2: Crash Dump Analysis Patterns.....	31
Divide by Zero (Kernel Mode)	31
Fat Process Dump	33
Blocked Queue	34
Crash Signature	37
Invalid Parameter (Process Heap).....	40
Hooking Level.....	43
Embedded Comments.....	47
Well-Tested Module.....	48

String Parameter	49
Environment Hint	51
Dual Stack Trace	52
Blocking Module	54
Wait Chain (Window Messaging)	55
Wait Chain (Named Pipes)	60
Top Module	62
Dialog Box	63
Technology-Specific Subtrace (COM Interface Invocation)	67
Livelock	70
Semantic Structure (PID.TID)	73
Instrumentation Side Effect	77
Directing Module	80
Stack Overflow (Software Implementation)	82
Data Correlation	84
Truncated Stack Trace	86
Least Common Frame	87
Self-Diagnosis (Kernel Mode)	89
Technology-Specific Subtrace (Dynamic Memory)	90
Module Hint	92

Custom Exception Handler (Kernel Space).....	94
No Data Types	96
Cloud Environment	97
Version-Specific Extension	99
Multiple Exceptions (Managed Space).....	104
Blocking File	105
Quiet Dump.....	109
Pleiades	110
Thread Age	111
Unsynchronized Dumps	113
Coupled Modules	114
Managed Stack Trace	115
Problem Vocabulary.....	116
Activation Context.....	117
Stack Trace Set	120
Special Thread (.NET CLR)	123
Dynamic Memory Corruption (Managed Heap)	124
Stack Trace Collection (Managed Space)	127
Duplicate Extension	131
Deadlock (Managed Space).....	135

Caller-n-Callee	138
Handled Exception (User Space)	141
Handled Exception (.NET CLR).....	144
Execution Residue (Managed Space)	149
Annotated Disassembly (JIT .NET code).....	151
Wait Chain (Mutex Objects).....	153
Inline Function Optimization (Managed Code)	155
Technology-Specific Subtrace (JIT .NET Code)	157
Double IRP Completion	160
PART 3: Pattern Interaction	163
Main Thread, Self-Diagnosis, Window Message Chain, Blocking Module, Ubiquitous Component, Dual Stack Trace, Pipe Wait Chain and Coupled Machines.....	163
Abridged Dump, Embedded Comment, Spiking Thread, Incorrect Stack Trace and Top Module.....	166
Stack Trace Collection, Message Box, Self-Diagnosis, Version-Specific Extension, Managed Stack Trace and Managed Code Exception	168
PART 4: Unified and Generative Debugging	171
A Periodic Table of Software Defects.....	171
Analysis, Architectural, Design, Implementation and Usage Debugging Patterns....	172
Generative Debugging	173
Metadefect Template Library	174
PART 5: A Bit of Science and Philosophy.....	175

On Memory Perspectives	175
Orbifold Memory Space	176
Notes on Memoidealism	177
M->analysis	178
Memiosphere	179
On Memory-Time vs. Space-Time	180
The Will to Be Memorized	181
The Trinity of Memory Worldview	182
Uses of Memoretics	183
Crossdisciplinary Memoretics as Interdisciplinary Science	184
Private Property on Memory Spaces	185
Coarse vs. Fine Grained DNA of Software Behavior	187
PART 6: Fun with Crash Dumps	189
Music for Debugging	189
555 Binary Threads	189
Out of Memory and Losing My Data (Comment Impact)	190
Navigating the Long List	191
Debugging Joke	192
Memory Dump Barcodes	193
MessageBox at Dublin Zoo	194

CDB for Kids.....	195
Snow Spike Residue	196
Second Snowfall Spike in Dublin	197
MMXI.....	198
Happy New Year and Decade of Debugging 0x7DB - 0x7E4!.....	199
Do Security Professionals Dream?	204
Debugging Slang.....	205
Golden Bug.....	205
Beer Time	206
Finger Exercise	207
Resolution Rush	208
The Window of Opportunity	209
Dump.....	210
Pre-analysis	211
Tapping.....	212
Having Fun	213
Adult Debugging.....	214
Second Eye	215
Abscess.....	216
Finction	217

Mad OS and other Publishing Blunders	218
The Ultimate Debugger's Desk.....	221
Memceptions: Flags and Handles are Everywhere!.....	222
Computer Memory Monsters	223
On President's Daily Briefs (PDBs)	226
The First Evidence for Process Resurrection.....	227
Vacuum Pages	228
WinDbg Command on Certificate	230
Pleasing WinDbg SOS Extension.....	231
Airport Terminal Services Incident.....	232
Philosophical Self-Interview.....	233
PART 7: A Bit of Religion	235
Memory Creates God	235
Morality and Memorianity.....	236
On Natural Theology	237
PART 8: Software Trace Analysis.....	239
Pattern Interaction.....	239
Basic Facts, Periodic Error, and Defamiliarizing Effect	239
Close and Deconstructive Readings of a Software Trace	240
Software Tracing Best Practices.....	241

No Longer Seeing Nothing: The Advantage of Patterns.....	242
PART 9: Software Trace Analysis Patterns	243
Focus of Tracing	243
Event Sequence Order	244
Implementation Discourse.....	245
News Value	246
Master Trace	247
Gossip.....	248
Impossible Trace	249
Glued Activity.....	250
Message Invariant.....	251
UI Message.....	252
Original Message.....	253
PART 10: Software Troubleshooting and Debugging	255
Debugware Patterns	255
System Description Snapshot.....	255
Debugging in 2021: Trends for the Next Decade	256
The Way of Philip Marlowe: Abductive Reasoning for Troubleshooting and Debugging	257
Workaround Patterns	258
Fake API.....	258

User Interface Problem Analysis Patterns.....	259
Message Box	259
PART 11: Software Victimology	263
Function Activity Theory	263
PART 12: Art	265
No E-numbers Software Product Sticker	265
Paleo-debugging: Excavated Minidump.....	266
Stack Trace Art	267
Debugger's Dream	268
Defect in Defect	269
Memorianity Cross.....	270
Memioart: The New Art Form.....	271
Clouded	272
Cloud Traces.....	273
What Is To Be Done?.....	274
PART 13: Miscellaneous	277
GI Index of Memory Dump Analysis.....	277
The New School of Debugging	279
TestWER Tool to Test Windows Error Reporting	280
Moving to ARM	283

The New School of Debugging: What's New.....	284
A.C.P. Root Cause Analysis Methodology	285
TestWAER Tool to Test Windows Azure Error Reporting.....	286
PART 14: Intelligence Analysis	289
Intelligence Analysis Patterns	289
The Birth of Memory Intelligence Agency.....	290
Appendix.....	291
Memory Analysis as a Service	291
Stack Overflow Patterns.....	292
.NET / CLR / Managed Space Patterns	293
Stack Trace Patterns.....	294
Symbol Patterns	295
Analysis Compass	296
Software Trace Analysis Checklist	297
Crash Dump Analysis Checklist.....	298
Index of WinDbg Commands	301
About the Author	304
Cover Images.....	305

Index of WinDbg Commands

!address, 38
!alpc, 35, 75, 303
!analyze, 303
!analyze -v, 31, 32, 37, 41, 82, 94, 99, 118, 133, 161, 282, 301
!analyze -v -hang, 301
!apc, 303
!bugdump, 303
!chkimg, 43, 44, 301
!CLRStack, 101, 102, 116, 128, 133, 134, 135, 152, 159, 305
!cs, 301, 303
!dh, 301
!dlk, 136, 138
!dlls, 301
!dpcs, 303
!DumpHeap, 127, 305
!DumpObj, 127, 151
!DumpStack, 116, 139, 140, 141, 149, 150, 170
!DumpStackObjects, 150, 305
!envvar, 301
!error, 109
!exchain, 95, 301
!exqueue, 303
!filecache, 303
!fileobj, 61, 107, 303
!flag, 40, 77, 301
!handle, 108
!help, 20
!IP2MD, 153, 157, 158, 159
!irp, 60, 83, 107, 162, 303
!irpfind, 303
!locks, 70, 212, 301, 303
!ipc, 303
!ntsdexts.locks, 303
!pe, 102, 105, 116, 170, 305
!peb, 51, 301
!pool, 91, 93, 303
!poolused, 115, 303
!PrintException, 102, 104, 105
!process, 58, 299, 303
!qlocks, 303
!ready, 303

!runaway, 78, 112, 301
!running, 70, 303
!search, 200
!session, 303
!spprocess, 303
!stacks, 303
!syncblk, 136, 305
!sysinfo, 301, 303
!teb, 24, 142
!thread, 52, 73
!Threads, 105, 124, 305
!U, 152, 157
!uniqstack, 121, 122, 301
!VerifyHeap, 126, 127, 305
!vm, 229, 303
.asm, 152, 157
.chain, 100, 101, 132, 133, 134, 305
.cordll, 99, 101, 102
.cxr, 32, 38, 84, 88, 108, 118
.dump, 232
.enumtag, 303
.formats, 85
.kframes, 28, 301
.load, 20, 101, 132, 232
.process, 108
.reload, 108
.symfix, 301
.thread, 84, 88, 108
.time, 113, 114
.ttime, 112
.unload, 100, 135
? , 84
~, 167
~*e, 105, 128, 305
~*kbL, 55, 56
da, 49, 50

g, 283
-hang, 303
k, 282
k L=, 29
kc, 26, 27, 77, 112, 121
kL, 24, 37, 40, 41, 54, 62, 68, 82, 84, 94, 125, 156, 158, 164, 167, 169, 262
kv, 31, 37, 42, 49, 56, 63, 64, 84, 108, 121, 301
lm, 97, 111, 224
lmv, 62, 97, 98, 100, 132, 301, 305
ln, 303
r, 87, 157
s-d, 200
u, 44, 46
ub, 42, 62, 83, 137, 138, 140, 143, 262
version, 228

Memory Dump Analysis Anthology

Volume 7

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2014 by Dmitry Vostokov

Copyright © 2014 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-51-1 (Paperback)

ISBN-13: 978-1-908043-52-8 (Hardback)

First printing, 2014

Revision 2 (July 2015)

Contents

Preface	23
Acknowledgements.....	25
PART 1: Professional Crash Dump Analysis and Debugging.....	27
WinDbg Shortcuts	27
.ecxr.....	27
!heap -x -v	29
!sw and !k.....	31
Two WinDbg Scripts That Changed the World.....	32
Raw Stack Dump of All Threads (Kernel Space)	37
The Design of Memory Dump Analysis: 7 Steps of Highly Successful Analysts.....	38
Postmortem Effects of -g	39
Event Owners	42
Improbable Occurrence	48
Pattern Cooperation	49
Page Heap Implementation	54
More Common Mistakes in Memory Analysis	60
Memory Dump Analysis Best Practices.....	63
PART 2: Crash Dump Analysis Patterns.....	65
FPU Exception	65

Hidden Parameter	67
Memory Leak (Page Tables)	69
Unrecognizable Symbolic Information	76
Network Packet Buildup.....	82
Disconnected Network Adapter.....	83
Problem Module	85
Empty Stack Trace	86
Debugger Bug.....	90
Value References	92
Self-Diagnosis (Registry).....	93
System Object	95
Module Variable.....	98
Stack Trace Collection (Predicate)	100
Stack Trace Collection (I/O Requests)	101
Regular Data.....	106
Translated Exception.....	107
Blocked DPC	108
Late Crash Dump	109
Blocked Thread (Timeout).....	110
Punctuated Memory Leak	111

Insufficient Memory (Reserved Virtual Memory)	114
Coincidental Error Code	117
Stored Exception	119
Activity Resonance	120
Value Adding Process	122
Memory Leak (I/O Completion Packets)	123
No Current Thread	124
Unloaded Module	126
Stack Trace Change	131
Spike Interval.....	132
Deviant Module.....	133
Hidden Exception (Kernel Space)	140
Handled Exception (Kernel Space)	141
High Contention (.NET CLR Monitors).....	142
Frozen Process	145
Incomplete Session	150
Error Reporting Fault	152
First Fault Stack Trace	155
Hidden Process.....	156
Disk Packet Buildup.....	158

Deviant Token	161
Module Collection.....	162
Handle Leak.....	164
Critical Stack Trace	165
Debugger Omission	166
Broken Link.....	168
Wait Chain (Pushlocks).....	170
Insufficient Memory (Session Pool)	172
Step Dumps.....	173
Reduced Symbolic Information.....	174
Injected Symbols	175
Glued Stack Trace.....	178
Distributed Wait Chain.....	182
Ubiquitous Component (Kernel Space).....	184
One-Thread Process	187
Module Product Process	189
Crash Signature Invariant.....	190
Small Values	191
Shared Structure	193
Wait Chain (CLR Monitors).....	194

Thread Cluster	195
Module Collection (Predicate)	196
False Effective Address	197
Screwbolt Wait Chain	198
PART 3: Core Dump Analysis Patterns (Mac OS X).....	201
GDB for WinDbg Users	201
Stack Trace	203
GDB Annoyances: Incomplete Stack Trace	205
NULL Pointer (Data)	206
Shared Buffer Overwrite	207
Multiple Exceptions.....	211
Double Free (Process Heap).....	213
Dynamic Memory Corruption (Process Heap)	214
Spiking Thread.....	216
NULL Pointer (Code).....	218
Execution Residue	220
Coincidental Symbolic Information	223
Paratext.....	225
Truncated Dump	227
C++ Exception.....	228

Local Buffer Overflow	229
Divide by Zero (User Mode)	231
Stack Overflow (User Mode)	232
Active Thread	236
PART 4: Malware Analysis Patterns	239
Malware: A Definition	239
Fake Module	240
RIP Stack Trace	244
Driver Device Collection	246
Pre-Obfuscation Residue.....	247
Packed Code.....	248
Raw Pointer.....	251
Out-of-Module Pointer	252
Patched Code	253
String Hint	254
Namespace.....	257
PART 5: A Bit of Science and Philosophy.....	259
On Matter	259
Commodities as Memories	260
Software as Means of Production	261

Notes on Memoidealism	262
The Confluence of Computers, Philosophy, and Religion	264
Analytic Memory Dump - A Mathematical Definition.....	265
Sorting and Early Greek Philosophers	266
General Abnormal Patterns of Structure and Behavior	267
On Matter and Substances.....	268
M-Memory	269
Ontology of Memoidealism	270
Philosophies of Persistence.....	273
Information as Arrow	275
Dialectical Triad in Memoidealism	276
PART 6: Software Trace Analysis Patterns	279
Software Trace Diagrams (STDiagrams).....	279
Macrofunction	283
Linked Messages	284
Marked Message.....	285
Trace Frames.....	286
Counter Value	288
Message Context.....	289
Error Distribution	290

Break-in Activity	291
Resume Activity.....	292
Fiber Bundle	294
Data Flow	296
Empty Trace	298
Error Message	299
Periodic Message Block.....	300
Visibility Limit.....	301
Relative Density	302
Sparse Trace	303
Opposition Messages	304
Split Trace.....	305
Message Interleave.....	306
Sheaf of Activities.....	307
Indexical Trace	310
Abnormal Value	311
Dominant Event Sequence.....	313
Pivot Message	314
Traces of Individuality	318
Indirect Facts.....	319

Hidden Error	320
Last Activity	322
State and Event	324
Dialogue	326
Motif	329
Exception Stack Trace (Java)	330
Correlated Discontinuity	332
Piecewise Activity.....	333
Density Distribution	335
Factor Group	336
Silent Messages.....	339
Shared Point.....	341
Meta Trace	343
Data Association.....	344
State Dump	346
Message Cover	347
Message Set	349
Error Thread	351
Activity Divergence	352
PART 7: Fun with Crash Dumps.....	355

Debugging Slang	355
LoL	355
Watching a Movie	356
PonOS.....	357
Typology, Typological.....	358
Memorandum	359
HELL.....	360
FBI	361
poo	362
STaMPs.....	363
A NoSQL Problem.....	364
Matrix.....	365
Fool	366
B2B, B2C, H2H	367
New Year Eve Debugging	368
Happy New Spiking Year of Software Trace Analysis	369
Happy New Year (from Windows 8).....	370
Music for Debugging	372
Going Romantic.....	372
Make It through This Trace	373

Fiction for Debugging	374
The Problem and The Solution.....	374
Pilgrimage to Harvard University	375
Welcome to Ki* and Ke*	376
I Memory Dump	377
A Blue Screen Watch.....	379
Poetry.....	380
Surfaces in Nature.....	381
PART 8: Software Narratology	383
Software Anti-Narrative	383
Software Narratology Helps Fiction Writers	384
Narremes in Software Narratology	386
Narralog - A Software Trace Modeling Language	387
What is a Software Narrative?	388
Software Narrative Planes	389
Software Narratology Square.....	391
Writing and Validation of Historical Narratives	392
Software Trace Analysis Patterns Domain Hierarchy.....	393
Process Monitor as Modeling Tool	394
Generalized Software Narrative and Trace.....	395

Unified Computer Diagnostics: Incorporating Hardware Narratology	396
Introducing Software Narratology of Things (Software NT)	397
PART 9: Software Diagnostics, Troubleshooting, and Debugging.....	399
Unified and Generative Debugging	399
Analysis, Architectural, Design, Implementation and Usage Debugging Patterns	399
Software Problem Description Language.....	401
What are Software Trace and Memory Dump Analysis? A One Sentence Definition	402
Software Problem Solving Tools as a Service	403
Software Problem Description Patterns	404
Software Behavior Pattern Prediction	405
Patterns of Software Diagnostics	406
First Fault	406
Highly Effective Diagnostics	407
Network Trace Analysis Patterns	408
Software Diagnostics Services.....	411
Architecture of Process Memory Dump Capture Done Right	412
An Introduction to General Systems Thinking (Book Review)	413
Software Diagnostics Institute Logo	414
User Interface Problem Analysis Patterns.....	415
Unresponsive Window	415

Pattern-Based Software Diagnostics	418
Software Diagnostics Discipline	419
Architecture of memCPU	420
Phenomenology of Software Diagnostics: A First Sketch.....	421
Software Diagnostics Report Schemes.....	422
Missing Cause Trace	422
Software Diagnostics Training: Two Approaches.....	423
Software Disruption Patterns.....	425
Space Precondition	425
Static Code Analysis Patterns.....	426
Loop Construct	426
The Structure of Software Problem Solving Organization	427
Bridging the Great Divide.....	428
Elementary Software Diagnostics Patterns.....	429
Zero Fault Software Diagnostics	430
Agile Software Diagnostics.....	432
ADDR Pattern Catalogue	433
Thinking-Based Software Diagnostics	434
Memory Acquisition Pattern Catalog.....	436
Trace Acquisition Pattern Catalog.....	437

Patterns of Software Diagnostics Architecture	438
Detecting and Predicting the Unknown	440
Software Diagnostics Metaphors	442
Software Diagnostics as Psychology	442
Software Diagnostics as Literary Criticism	443
Rapid Software Diagnostics Process (RSDP).....	444
Right First Time Software Diagnosis.....	445
Software Diagnosis Codes	446
Vulnerability Analysis Patterns (VAP).....	447
Versioned Namespace	449
PART 10: Art and Visualization.....	451
2012 (Pessimistic)	451
2012 (Optimistic).....	452
A Bug in a Bag (Collections, Ex-hi-bit 1)	453
A Bug Meets a Bug (The Clash of Civilizations)	454
A Bug Catcher.....	455
The Second Generation of CARE System (Trademark).....	456
RawStackGram	457
A Memory Window	458
Liquid Memory	459

Computer Brain	460
Computer Evolution	461
M Spaces	462
Happy Hellowin!.....	463
Pointers in Nature	464
Drink Sensibly Before The End Of The World!	465
MM=DD=YY	466
Process Monitor Log Visualized	468
Holes Infinity (HI OS)	472
Cyber Vostok Missions	473
A Dump Machine	474
The Power of Simplicity.....	475
Happy St. Patrick's Screen.....	476
Happy New Year 2014!	477
I Love Software Diagnostics	478
Puree Windows Cooking	479
Salad Winterminal.....	479
Kernel Soup	481
Neolithic Soup	482
Food Subsystems	483

An Accident of Creation	484
So Chi Salad, 2014	485
Self-Organized Window-ed soup	486
Political Computicarts	487
Needs Non-Invasive Debugging!	487
Russian Spaces	488
The Day I Quit.....	489
Hero of Dump Analysis, a Medal for Labor Day	490
Diagnosed by Vostokov ^{®TM}	491
Stack Trace Shapes.....	492
The Art of Internals	494
Threadinking	495
PART 11: Miscellaneous	497
C and C++ Programming Books That Made a Great Impression on the Author.....	497
Outside.....	499
After Debugging	500
Crash Dumps, Acquisitions, and Layoffs	501
Cadaver Worm: An Exercise in Malware Fiction	502
WinDbg as UNICODE to ASCII Converter	504
Appendix.....	505

Falsity and Coincidence Patterns	505
Process Patterns.....	506
Thread Patterns.....	507
Optimization Patterns	508
Exception Patterns	509
Module Patterns	510
RPC, LPC and ALPC Patterns and Case Studies.....	511
ERESOURCE Patterns and Case Studies.....	513
Meta-Memory Dump Patterns.....	515
Crash Dump Analysis Checklist.....	516
Index of WinDbg Commands	519
About the Author	521
Notes.....	522
Cover Images.....	523

Index of WinDbg Commands

!address, 54, 111, 112, 113, 115, 117, 118,
 125, 133, 137, 202, 377, 378
!alpc, 51, 511, 517
!analyze, 27, 38, 167, 516, 517
!bugdump, 518
!chkimg, 240, 253, 516
!CLRStack, 518
!cs, 52, 183, 516, 517
!dd, 69, 70
!dh, 133, 137, 138, 166, 243, 248, 516
!dlls, 516
!dpcs, 108, 517
!DumpHeap, 518
!DumpRuntimeTypes, 518
!DumpStackObjects, 518
!eeheap, 518
!envvar, 516
!error, 66
!exchain, 516
!exqueue, 517
!filecache, 517
!fileobj, 517
!FinalizeQueue, 518
!for_each_module, 92, 240, 242
!for_each_process, 85
!for_each_thread, 32, 34, 37, 163, 168
!GCHandleLeaks, 518
!GCHandles, 518
!gflag, 54, 516
!handle, 191, 192
!heap, 29, 115, 117, 516
!irp, 44, 45, 517
!irpfind, 45, 101, 517
!lk, 31
!lmi, 77, 78, 117, 118, 189, 243
!locks, 38, 513, 516, 517
!ipc, 511, 517
!ndiskd.miniport, 83
!ndiskd.miniports, 83, 517
!ndiskd.pkt pools, 82, 517
!object, 95, 96, 246
!pe, 518
!peb, 242, 516
!pool, 517
!poolfind, 123, 156, 168
!poolused, 123, 164, 172, 517
!process, 38, 50, 69, 77, 85, 97, 123, 146,
 152, 153, 154, 156, 164, 168, 169, 187,
 356, 517
!pte, 74, 75
!ptov, 70, 71, 72
!qlocks, 517
!ready, 517
!reg, 93
!runaway, 32, 38, 132, 142, 237, 416, 516
!running, 38, 120, 517
!scsikd.classext, 158, 159, 517
!session, 49, 150, 172, 517
!sprocess, 49, 122, 145, 150, 151, 517
!stacks, 184, 517
!sw, 31
!syncblk, 144, 518
!sysinfo, 516, 518
!teb, 86, 107
!thread, 32, 34, 37, 51, 108, 120, 121, 140
!Threads, 518
!token, 161
!uniqstack, 516
!VerifyHeap, 518
!vmt, 85, 90, 96, 98, 99, 168, 172, 517
.chain, 518
.cxr, 27, 52, 53, 65, 67, 79, 80, 81, 128, 129,
 140, 153, 197, 240
.echo, 32
.ecx, 27, 28, 119
.effmach, 31, 153, 163
.enumtag, 518
.exr, 66, 119, 125
.imgscan, 133, 166
.kframes, 516

.load, 31, 153, 158, 162, 163
.process, 52, 74, 76, 85, 153, 182
.reload, 80, 114, 162, 163, 176, 202
.symfix, 114, 516
.sympath+, 80, 176
.thread, 27, 37, 52, 53, 67, 79, 80, 81, 129,
 153, 162, 163, 183, 240
~, 28, 38, 60, 109, 124, 126, 142, 191, 202,
 204, 516, 518
~*k, 28, 109, 126, 142, 202
~*kv, 38, 516
Checklist, 516
d*, 38
dc, 56, 168, 174
dd, 69, 117, 129
dp, 29, 98, 118, 129, 191, 377
dps, 37, 65, 86, 107, 138, 140, 144
dpu, 68, 417
ds, 99, 516, 517, 518
dt, 43, 60, 61, 62, 93, 95, 145, 146, 158,
 174, 177
du, 504
eb, 504
g, 39, 40, 41, 115, 183, 202, 213, 385
k, 52, 53, 59, 89, 115, 124, 144, 153, 180,
 202, 240, 244, 415, 416
kc, 190
kL, 27, 39, 40, 67, 79, 80, 81, 86
kv, 60, 128, 129, 174, 176, 191, 202
kvL, 110
lm, 85, 175, 176, 252
lmp, 240
imu, 85, 162
lmv, 38, 77, 78, 79, 117, 118, 127, 133, 139,
 162, 189, 242, 417, 516, 517, 518
ln, 117, 118, 130, 517
rMF, 66
u, 48, 98, 117, 143, 154, 202, 253
ub, 67, 129, 141, 193, 201, 202

Memory Dump Analysis Anthology

Volume 8a

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2014 by Dmitry Vostokov

Copyright © 2014 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-53-5 (Paperback)

First printing, 2014

Revision 2.0 (July 2015)

Table of Contents

Preface	7
About the Author	9
PART 1: Professional Crash Dump Analysis and Debugging	11
Software Diagnostics Professional Certification	11
Three Roads to Kernel Space	13
PART 2: Crash Dump Analysis Patterns	15
Design Value	15
Hidden IRP.....	16
Tampered Dump	17
Wait Chain (RTL_RESOURCE)	29
Memory Fluctuation (Process Heap)	35
Last Object	37
Rough Stack Trace.....	39
Past Stack Trace	43
Stack Trace (I/O Request)	46
Stack Trace (File System Filters).....	48
Stack Trace (Database).....	51
Wait Chain (Modules)	56
Insufficient Memory (Stack Trace Database)	57
Insufficient Memory (Region)	63
Memory Leak (Regions)	65
Invalid Handle (Managed Space)	69
Ghost Thread	77
Dry Weight	79
Exception Module	80
PART 3: Memory Forensics	83
Memory Forensics Professional Certification	83
Native Memory Forensics	84

PART 4: A Bit of Science and Philosophy	85
Memory Symmetry Breaking	85
Memoevolutionism.....	86
Entropy as Memory and Memory as Entropy.....	87
Notes on Memoidealism.....	88
Welcome to Memorianism	89
United Memory Lands, Memorianites, EthnOS	90
Quotes from Memoriarch.....	91
Pattern-Oriented Philosophy	92
PART 5: Software Trace Analysis Patterns	93
Hidden Facts	93
Back Trace	95
Blackout	97
Missing Message.....	99
Use Case Trail.....	101
Event Sequence Phase	103
Milestones	105
File Size	107
Singleton Event	108
Visitor Trace	110
PART 6: Fun with Crash Dumps.....	111
Debugging Slang and Proverbs	111
<i>PUS</i>	111
<i>Coollect</i>	111
<i>Dump-out</i>	111
<i>LOGIC</i>	111
<i>DiagNose</i>	112
<i>Consolidation</i>	112
<i>No Pass a Run!</i>	112
<i>ID IoT Zone</i>	112
<i>Putty in Someone's Hands</i>	112
<i>DisPatched vs. DESPatched</i>	112
<i>Programmatica Nervosa</i>	113

<i>GOTCHA</i>	113
<i>Pan-o-RAM-ic</i>	113
<i>VLSI</i>	113
<i>Debugging Proverb</i>	113
Space Opera	114
If Programmers Were Writers	115
My Computer Celebrates Halloween	116
Look, there's a Bug!	117
Diagnostics in Science Fiction	118
Hard Copy Natives	119
PART 7: Software Narratology	121
Malnarratives	121
Higher-Order Pattern Narratives (Analyzing Diagnostic Analysis)	123
PART 8: Software Diagnostics, Troubleshooting, and Debugging	127
A Pattern Language for Performance Analysis	127
The Timeless Way of Diagnostics	128
Pattern-Oriented Debugging Process	130
PART 9: Art and Visualization	133
Café WoW	133
Bang Debugging	134
Bug Hunter	135
Glass of Water Dump	136
Memory Dump Analysis	137
Organic Incidents and Bad Stench	138
PART 10: Miscellaneous	139
Book Discovery	139
Quotes	140
Appendix	143

Crash Dump Analysis Checklist	143
Index of WinDbg Commands	147
Notes.....	149

Index of WinDbg Commands

!address, 52, 58, 63, 64, 65, 66, 67, 68,
 79
!alpc, 15, 144
!analyze, 143, 144, 145
!bugdump, 145
!chkimg, 144
!CLRStack, 73, 75, 76, 145
!cs, 78, 143, 144
!dh, 143
!dlls, 143
!do, 74, 76
!dpcs, 144
!dso, 73, 75
!DumpHeap, 145
!DumpRuntimeTypes, 145
!DumpStackObjects, 145
!eeheap, 145
!envvar, 143, 144
!exchain, 144
!enqueue, 144
!filecache, 144
!fileobj, 144
!FinalizeQueue, 145
!fltkd, 49
!GCHandleLeaks, 145
!GCHandles, 145
!gflag, 52, 74, 143
!handle, 33
!heap, 57, 59, 63, 64, 144
!irp, 16, 46, 49, 144
!irpfind, 16, 144
!locks, 143, 144
!ipc, 144
!ndiskd.miniports, 144
!ndiskd.pktpools, 144
!pe, 70, 75, 145
!peb, 143
!pool, 145
!poolused, 144
!process, 144
!qlocks, 144
!ready, 144
!runaway, 60, 143
!running, 144
!scsikd.classext, 144
!session, 144
!sprocess, 144
!stacks, 144
!syncblk, 145
!sysinfo, 143, 145
!teb, 39
!thread, 16, 33, 46, 78
!Threads, 145
!uniqstack, 143
!VerifyHeap, 145
!vmm, 144, 145
!process, 144
.chain, 145
.cxr, 18, 21, 28, 31
.ecxr, 18
.enumtag, 145
.for, 13
.frame, 31, 32, 72
.kframes, 143
.load, 70
.process, 77
.symfix, 143
.thread, 18, 28, 30, 31, 78
~, 143, 145
~*kv, 143
dp, 32
dps, 21, 52, 61
dpS, 39
dS, 143, 144, 145
dt, 31
k, 17, 18, 21, 28, 43, 51, 71, 74
kc, 60, 81
kn, 31, 72
lmv, 70, 143, 145
ln, 145

148 Index of WinDbg Commands

s, 143, 144

ub, 51, 73

uf, 30

Memory Dump Analysis Anthology

Volume 8b

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2015 by Dmitry Vostokov

Copyright © 2015 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-54-2 (Paperback)

First printing, 2015

Revision 1.02

Table of Contents

Preface	7
About the Author	9
PART 1: Professional Crash Dump Analysis and Debugging.....	11
Win32 Start Address Fallacy	11
Multidimensionality of Exceptions	13
PART 2: Crash Dump Analysis Patterns	15
Reference Leak.....	15
Origin Module	19
Hidden Call.....	21
Corrupt Structure.....	26
Software Exception	29
Crashed Process	30
Variable Subtrace	31
User Space Evidence	37
Technology-Specific Subtrace (COM Client Call).....	38
Internal Stack Trace	39
Distributed Exception (Managed Code).....	41
Thread Poset	43
PART 3: Pattern Interaction	45
Virtualized Process, Stack Trace Collection, COM Interface Invocation Subtrace, Active Thread, Spiking Thread, Last Error Collection, RIP Stack Trace, Value References, Namespace, and Module Hint	45
PART 4: A Bit of Science and Philosophy	57
Cantor Operating System.....	57
Metaphor of Memory as a Directed Container	57
Praxiverse.....	58
When Universe is Going to End?.....	58

Notes on Memoidealism.....	59
PART 5: Software Trace Analysis Patterns	61
Timeout.....	61
Activity Overlap.....	65
Adjoint Space	67
Indirect Message.....	70
Watch Thread	75
Punctuated Activity.....	77
Trace Mask.....	78
Trace Viewpoints	81
Data Reversal	83
Recovered Messages	85
Palimpsest Messages	87
Message Space.....	90
Interspace	92
Translated Message	94
Activity Disruption	96
PART 6: Fun with Debugging, Crash Dumps, and Traces.....	99
The Dump from the Future	99
Exchange Rate on 16.12.14.....	99
Check the Plug	100
Debugging Slang.....	101
<i>YAWE</i>	101
<i>Embedded Software Engineer</i>	101
<i>Minute-wise</i>	101
<i>Developer</i>	101
<i>Multidigitalist</i>	101
<i>KgB</i>	102
<i>CIQ (Crash IQ)</i>	102
<i>Pat Ching</i>	102
<i>Explosive Mixture</i>	102
<i>POEM</i>	102
<i>YearNormous Day</i>	103
<i>eNormous</i>	103

2015 - The Year of RAM	104
Diagnostics and Debugging in Science Fiction	105
Software and Hardware Exceptions.....	108
Logging for Kids.....	110
Find the Bug	111
Music for Debugging	112
Tracing and Counting Book	113
The Last Error	114
Patching the Hardware Defect.....	115
Pattern Match	116
PART 7: Software Narratology	117
Coding and Articoding.....	117
PART 8: Software Diagnostics, Troubleshooting, and Debugging	119
Special and General Trace and Log Analysis	119
Projective Debugging	123
Pattern! What Pattern?	132
I Didn't See Anything	135
PART 9: Art and Photography	137
Diagnostics Designer Glasses	137
Pattern Diagnostics Logo	138
Happy Valentine's Day	139
50 Shades of Crash Dump	140
Computer Universe	141
Failed Surveillance	142
Debugging Allegory on FEB 23	143
Object in Signaled State	144
Kernel Space Starts with 8	145
The Day of ST. P. The Elimination of Snakes	146
The Fifth Column.....	147
Proportionate Disproportionate Proportion.....	148
Autoportrait in 5 Objects	149
Kernel Works.....	150
Chip Forensics	151

Industrial Windows	152
The Meaning of Life	153
Hidden Bug.....	154
PART 10: Memory Forensics	155
Artifact-Malware and its Primary and Secondary Effects	155
PART 11: Miscellaneous.....	161
Quotes.....	161
Status Updates.....	163
Execution Residue.....	164
Appendix	165
Patterns are Weapons for Massive Debugging.....	165
Crash Dump Analysis Checklist	166
Index of WinDbg Commands	169

Index of WinDbg Commands

!address, 50, 55
 !alpc, 167
 !analyze, 41, 166, 167, 168
 !bugdump, 168
 !chkimg, 166
 !CLRStack, 23, 168
 !cs, 166, 167
 !dh, 166
 !dlls, 166
 !dpcs, 167
 !DumpHeap, 168
 !DumpObj, 41
 !DumpRuntimeTypes, 168
 !DumpStackObjects, 168
 !eeheap, 168
 !envvar, 166, 167
 !error, 49
 !exchain, 166
 !exqueue, 167
 !filecache, 167
 !fileobj, 167
 !FinalizeQueue, 168
 !GCHandleLeaks, 168
 !GCHandles, 168
 !gflag, 166
 !gle, 48
 !heap, 167
 !irp, 167
 !irpfind, 167
 !locks, 166, 167
 !ipc, 167
 !ndiskd.miniports, 167
 !ndiskd.pktpools, 167
 !object, 16
 !obtrace, 18
 !pe, 168
 !peb, 166
 !pool, 168
 !poolused, 16, 17, 167
 !process, 11, 15, 30, 92, 167
 !qlocks, 167
 !ready, 167
 !runaway, 47, 166
 !running, 167
 !scsikd.classext, 167
 !session, 99, 167
 !sprocess, 11, 167
 !stacks, 43, 167
 !sw, 45
 !syncblk, 168
 !sysinfo, 166, 168
 !teb, 48
 !thread, 11
 !Threads, 168
 !uniqstack, 166
 !VerifyHeap, 168
 !vm, 167, 168
 .asm, 50
 .chain, 168
 .enumtag, 168
 .kframes, 166
 .load, 45
 .reload, 45, 157
 .symfix, 45, 157, 166
 ~, 166, 168
 ~*kv, 166
 dc, 50, 155
 dps, 49, 55
 dS, 166, 167, 168
 dt, 26, 27, 28
 kL, 29, 46, 158
 lmV, 56, 155, 156, 158, 166, 168
 ln, 168
 s, 166, 167
 s-d, 55
 ub, 24, 25, 49

Memory Dump Analysis Anthology

Volume 9a

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2016 by Dmitry Vostokov

Copyright © 2016 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-35-1 (Paperback)

First printing, 2016

Table of Contents

Preface	7
About the Author	9
PART 1: Professional Crash Dump Analysis and Debugging	11
When realloc is not a realloc.....	11
WinDbg Shortcut !ddstack	12
PART 2: Crash Dump Analysis Patterns	15
Stack Trace Collection (CPUs)	15
Object Distribution Anomaly (.NET Heap)	19
Stack Trace Surface	22
Hidden Stack Trace	24
Evental Dumps	27
Active Thread (Windows).....	55
Clone Dump	59
Parameter Flow.....	63
Diachronic Module.....	67
PART 3: Pattern Interaction	69
Spiking Thread, Top Module, Module Hint, and Memory Fluctuation	69
PART 4: A Bit of Science and Philosophy	83
Quotes from Memoriarch	83
PART 5: Software Trace Analysis Patterns	85
Ruptured Trace	85
Sequence Repeat Anomaly	88
Adjoint Message	90
Coupled Activities	92
Error Powerset	94

Trace Dimension	96
Calibrating Trace	98
Data Interval	99
Identification Messages	101
PART 6: Fun with Debugging, Crash Dumps, and Traces.....	103
Dangerous Words	103
Debugging Slang.....	104
<i>MOAN</i>	104
<i>LOG</i>	104
<i>Diplodoc</i>	104
<i>pMud</i>	104
<i>HLL</i>	104
<i>Success</i>	105
<i>FOOD</i>	105
<i>Tor-mented</i>	105
<i>Obsession</i>	105
<i>Literature</i>	105
<i>CLERK</i>	105
<i>Analysis Paralysis</i>	106
<i>3D Dump</i>	106
<i>Star Wars</i>	106
<i>Daily Standup</i>	106
Debugging Curiosities	107
<i>Hung vs. Hanged</i>	107
<i>Trace Messages</i>	107
<i>13</i>	107
<i>Similar Cases</i>	107
<i>Error 1917</i>	108
Dump2Wave Update	109
Diagnostics and Debugging in Science Fiction	110
Suspicious Volume 9a	111
Music for Debugging	112

<i>Shpongle: Nothing Lasts But Nothing Is Lost</i>	112
PART 7: Linux Core Dump Analysis Patterns	113
NULL Pointer (Data)	113
Stack Trace	114
NULL Pointer (Code)	115
Spiking Thread	116
Dynamic Memory Corruption (process heap).....	118
Execution Residue.....	119
Coincidental Symbolic Information.....	121
Stack Overflow (user mode)	122
Divide by Zero (user mode).....	124
Local Buffer Overflow	125
C++ Exception	126
Paratext.....	127
Active Thread	129
Lateral Damage.....	130
Critical Region	131
PART 8: Software Diagnostics, Root Cause Analysis, Debugging	135
Workaround Patterns	135
<i>Axed Data</i>	135
Diagnostics, Forensics, Prognostics: The Copernican Revolution	137
Pattern Repertoire	140
Pattern-Oriented Software Internals: Pattern Paradigms and Software Internals	
Pattern Stack.....	142
Software Diagnostics Canvas	147
Software Traces and Logs as Proteins.....	149
Patterns-Based Root Cause Analysis Methodology	152
Teaching Complex Diagnostic Scenarios with Artificial Debugger (ArtDbg) and	
Pseudo-Memory Dumps	156
The Scope of Software Diagnostics	159
PART 9: Art and Photography	163

W - I'M DEBUGGIN' IT®	163
Coincidental Symbolic Information Pattern.....	164
Pisa Fault System Model	165
System Playing Tetris	166
A Pattern of Zeroes	167
Abnormal Structure	168
Control Your Software Emissions!	169
Component-Based Bug Architecture	170
PART 10: Miscellaneous.....	171
Quotes.....	171
World Software Diagnostics Day.....	173
Train Journey	174
Appendix	175
Crash Dump Analysis Checklist	175
Pattern Changes.....	178
Index of WinDbg Commands	179

Index of WinDbg Commands

!address, 60, 73, 76, 77
!alpc, 176
!analyze, 15, 25, 175, 176, 177
!bugdump, 177
!chkimg, 176
!CLRStack, 177
!cs, 175, 176
!ddstack, 3, 12
!dh, 175
!dlls, 175
!dpcs, 176
!DumpHeap, 19, 177
!DumpRuntimeTypes, 177
!DumpStackObjects, 177
!eeheap, 177
!envvar, 175, 176
!exchain, 176
!exqueue, 176
!filecache, 176
!fileobj, 176
!FinalizeQueue, 177
!GCHandleLeaks, 177
!GCHandlees, 177
!gflag, 175
!heap, 61, 74, 76, 77, 81, 176
!irp, 176
!irpfind, 176
!locks, 175, 176
!ipc, 176
!ndiskd.miniports, 176
!ndiskd.pkt pools, 176
!pe, 177
!peb, 175
!pool, 177
!poolused, 176
!process, 176
!qlocks, 176
!ready, 176
!runaway, 57, 69, 78, 175
!running, 15, 176
!scsikd.classext, 176
!session, 176
!sprocess, 176
!stacks, 176
!syncblk, 177
!sysinfo, 175, 177
!teb, 12
!Threads, 177
!uniqstack, 175
!VerifyHeap, 177
!vm, 176, 177
.chain, 177
.cxr, 25, 63
.ecxr, 26
.enumtag, 177
.exr, 25
.kframes, 175
.symfix, 175
~, 175, 177
~*k, 24, 59
~*kv, 175
dc, 64, 77
dd, 25
dp, 64
dps, 12, 65
dpS, 12, 13, 62
dS, 175, 176, 177
dt, 62
du, 78
kL, 64
kv, 24, 66
kvL, 63, 65
lmn, 59
lmv, 73, 175, 177
ln, 177
poi, 64
s, 175, 176
ub, 57, 64

Memory Dump Analysis Anthology

Volume 9b

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2016 by Dmitry Vostokov

Copyright © 2016 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-36-8 (Paperback)

First printing, 2016

Table of Contents

Preface	7
About the Author	9
PART 1: Crash Dump Analysis Patterns.....	11
Constant Subtrace.....	11
Wait Chain (Nonstandard Synchronization)	13
Not My Thread.....	16
Window Hint.....	17
Place Trace	20
Handle Limit (GDI, User Space)	22
Multiple Exceptions (Stowed)	28
Stack Trace Signature.....	35
Relative Memory Leak	37
JIT Code (Java).....	40
Wait Chain (C++11, Condition Variable)	42
PART 2: A Bit of Science, Philosophy, and Religion.....	45
Morality and Virtual Worlds	45
Quotes from Memoriarch.....	45
On Lives, Narratives, and Memory	45
Notes on Memoidealism.....	46
Worst Simulation World Hypothesis.....	46
Memory Ablution.....	46
PART 3: Software Trace Analysis Patterns	47
Data Selector.....	47
Declarative Trace	49
Trace Extension.....	50
Fourier Activity.....	51
Fiber of Activity.....	54
Missing Data	56
Message Pattern	57

Activity Theatre.....	58
Small DA+TA.....	59
Surveyor.....	61
Quotient Trace.....	62
PART 4: Fun with Debugging, Crash Dumps, and Traces	63
Debugging Slang.....	63
<i>Apology</i>	63
<i>MedioCriticalSection</i>	63
<i>SPASM</i>	63
<i>NoOO</i>	63
<i>AI</i>	63
<i>To Come Out of the Shell</i>	64
<i>3D Weekend</i>	64
<i>To Crawl into (One's Shell)</i>	64
<i>Bad Feeling</i>	64
<i>The Valley of Crash Dumps</i>	65
<i>Early Debugging</i>	65
<i>CHARLATAN</i>	65
Diagnostics and Debugging in Science Fiction	66
James Bond's Bugcheck and Error	66
Two-field System Agriculture.....	66
Bugs and InfoSec.....	66
Program Evolution	67
Roman + Hex.....	67
Debugging Curiosities	67
<i>Trace Messages</i>	67
Moscow Scare	68
Vacuum Needs PDB	69
My surname decomposed	69
Slavery.....	69
PORCA	70
Double Fee Request	70
Word Symmetry and Soviet History.....	70

Everything You Need for Debugging	71
Bugs in the System	72
Direct HR Reporting	72
The Devil at My Heels	72
PART 5: Software Narratology	73
PART 6: Software Diagnostics, Root Cause Analysis, Troubleshooting, and Debugging	75
Diagnostics of Things (DoT).....	75
Riemann Root Cause Analysis Language.....	76
Problem Solving as Code.....	80
Dia gram Graphical Diagnostic Analysis Language.....	82
Iterative Pattern-Oriented Root Cause Analysis	84
Theoretical Software Diagnostics and Education	86
PART 7: Art and Photography.....	89
Heap Corruption Explained by Lego Bricks	89
Linked List Illustrated by Lego Bricks	90
The Stack of Words	91
Packed and Unpacked Structures Illustrated by Lego Bricks	92
What Color is Your Instruction?.....	93
Sluggish System under Observation	104
Happy Debugging Card (Halloween Style)	105
PART 8: Structural Memory Patterns.....	107
Region Strata.....	107
PART 9: Miscellaneous.....	111
Quotes	111
English for Software Engineers (with UML)	114
Visual Learning Guide to Stack Traces	115
Real Programmers - No Impossible Code	116
Debugger Log Analyzer: Inception	118
Technical Books as Software.....	120

Job Forensic Archaeology	122
The Physical Spike	123
Software Experience Reuse through Generations	124
Fiber Bundle Reading	125
PART 10: Software Generalist	127
Sorting and Early Greek Philosophers.....	127
Software as Means of Production.....	127
MVC Worldview and the Origin of Economic Order	128
Software Generalist View of Religion	129
Mod N Reading System.....	130
Computational Collectives	133
Software Generalist Worldview	134
Event Tracing for Windows in UML	135
Empires of the Code.....	136
Standard Model and UML.....	137
Software Accommodation	138
Software Generalism	138
Software Labour and Alienation	139
Finite Sets.....	140
Computational Slotting Fees.....	141
On Facts about Software	141
Optimal Discrete Reading Chunks.....	142
On Software and Ethics.....	142
On Software Space-Time	143
Worship of Memory.....	143
Remembering d'Alembert	144
Software and Philosophical Beliefs	144
On Good Software	145
On Babbage-Chambers Paradox	145
On Abandonment	146
Cooperative Multireading Revisited and Started.....	147
Index of WinDbg Commands.....	149

Index of WinDbg Commands

!address, 17
!cs, 15
!error, 28, 30, 32, 33, 66
!handle, 42
!lmi, 69
.asm, 23
.cxr, 13, 14, 70
.exr, 28
.formats, 29
.lines, 30
.while, 25, 26
? , 26
~, 25
~*k, 16
dp, 23, 29
dps, 30, 32, 33, 113
dq, 23, 25
dt, 29
dw, 25
k, 40, 72
kc, 13, 14, 15, 22, 36, 68

Memory Dump Analysis Anthology

Volume 10

Dmitry Vostokov
Software Diagnostics Institute

OpenTask

Published by OpenTask, Republic of Ireland

Copyright © 2017 by Dmitry Vostokov

Copyright © 2017 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-908043-85-6 (Paperback)

First printing, 2017

Revision 1.03 (May 2017)

Table of Contents

Preface	7
About the Author	9
PART 1: Crash Dump Analysis Patterns	11
Quotient Stack Trace.....	11
Module Stack Trace	12
Foreign Module Frame.....	13
Unified Stack Trace	16
Mirror Dump Set	18
Memory Fibration	20
Aggregated Frames	21
Value Deviation (Structure Field).....	22
Stack Trace (I/O Devices)	24
High Contention (.NET Heap).....	26
Frame Regularity.....	30
Deadlock (.NET Finalizer)	34
Invalid Parameter (Runtime Function).....	38
Wait Chain (SRW Lock).....	40
Stack Trace Motif	42
PART 2: Linux Core Dump Analysis Pattern	43
Module Stack Trace	43
PART 3: Software Trace Analysis Patterns	45
Corrupt Message.....	45
Projective Space	47
Ornament.....	50
Poincaré Trace	52
De Broglie Trace Duality.....	55
Braid Group.....	57
Delay Dynamics.....	59
Activity Quantum	60

Trace Presheaf	61
Message Directory	63
Galois Trace.....	66
Singleton Trace	68
Braid of Activity.....	69
Tensor Trace	70
Unsynchronized Traces	72
Intrinsic ID	74
Combed Trace.....	75
Activity Packet.....	76
PART 4: Software Diagnostics, Root Cause Analysis, Troubleshooting, and Debugging	77
Topological Software Trace and Log Analysis	77
Is Your Security Healthy?	79
Software Diagnostic Space as a General Graph of Software Narratives.....	80
Software Diagnostics Metaphors.....	85
<i>Software Diagnostics as Archaeology</i>	85
Pattern-Oriented Diagnostic Analysis Process	86
Principles of Pattern-Oriented Software Data Analysis	88
Abstract Debugging Commands (ADC) Initiative	91
Reducing Analysis Pattern Complexity via Elementary Analysis Patterns	92
Categorical Foundations of Software Diagnostics	96
Existential Prognostics: Periodic Table of Diagnostic Patterns	98
Software Codiagnostics.....	100
The Unity of Pattern-Oriented Software Diagnostics	103
PART 5: A Bit of Science, Philosophy, and Religion	105
Quotes from Memoriarch	105
Notes on Memoidealism.....	105
PART 6: Fun with Debugging, Crash Dumps, and Traces.....	107
Debugging Slang.....	107

<i>SCANDAL</i>	107
<i>WTF</i>	107
<i>UOP</i>	107
<i>HOT</i>	107
<i>HOME</i>	107
<i>anOS</i>	108
<i>theOS</i>	108
<i>TCH</i>	108
<i>Top NoTCH</i>	108
 Problem Solving Techniques	109
 <i>Dissolution</i>	109
 Software Temperature.....	109
Watson.....	109
Cosmic Rays in Memory	110
Area 51.....	113
Measuring Software Diagnostics	114
Cash and Crash.....	114
Debugging Law.....	114
Suggested Pool Tags	114
TOR.....	114
Attitude to Debugging	115
British vs. American Spelling	116
How I Became a Grandmaster	117
Memory Dumps and VAT	117
Updatician.....	117
Traces of Cyrillic Alphabet.....	117
Prolific Letter.....	118
Code and Edoc	118
The Three-Software Vendor Body Problem.....	118
Modem Troubleshooting and Putin	118
Bugs for Fireworks	119
Critical SnowLOB.....	120
Apotypomamnismisophobia	121
Schadenfreude of AI	121
Diagnostics and Debugging in Science Fiction	121
The Mozart of Diagnostics	122

Keyboard Problems	124
PART 7: Art and Photography	125
(t)ra(c)in(g)	125
Development Muses	126
Pattern Inside and Outside	127
Between Diagnostic Activities	128
Patched Bug Construction Kit	129
Russian Binary	130
Secret Russian Binary	131
Postanalysm	132
PART 8: Miscellaneous.....	135
Quotes.....	135
Is There Any Life Inside Windows?	137
The Condition of My Productivity	138
CyberSpace and the Solution to CyberProblems	139
VAX/VMS Debugging Artefact.....	141
Observing Patterns of Cloud Structure and Behavior	143
Appendix	145
Crash Dump Analysis Checklist	145
Volume Index	149
Memory Analysis Patterns	149
Trace and Log Analysis Patterns	161
Index of WinDbg Commands	167

Index of WinDbg Commands

!alpc, 146
!analyze, 38, 39, 145, 146, 147
!bugdump, 147
!chkimg, 146
!CLRStack, 34, 147
!cs, 30, 145, 146
!dc, 113
!devobj, 25
!devobj, 24, 25
!devstack, 24
!dh, 145
!dlls, 145
!dpcs, 146
!DumpHeap, 147
!DumpRuntimeTypes, 147
!DumpStackObjects, 147
!eeheap, 147
!envvar, 145, 146
!exchain, 146
!exqueue, 146
!filecache, 146
!fileobj, 146
!FinalizeQueue, 147
!GCHandleLeaks, 147
!GCHandles, 147
!gflag, 145
!heap, 146
!irp, 24, 146
!irpfind, 146
!locks, 145, 146
!lpc, 146
!ndiskd.miniports, 146
!ndiskd.pktpools, 146
!pe, 147
!peb, 145
!pool, 147
!poolused, 146
!process, 23, 146
!qlocks, 146
!ready, 146
!runaway, 29, 145
!running, 146
!scsikd.classext, 146
!search, 113
!session, 146
!sprocess, 146
!stacks, 19, 146
!syncblk, 147
!sysinfo, 145, 147
!Threads, 147
!uniqstack, 145
!VerifyHeap, 147
!vm, 146, 147
.chain, 147
.enumtag, 147
.formats, 113
.kframes, 145
.symfix, 145
~, 37, 145, 147
~*k, 26, 40
~*kc, 16
~*kv, 145
dd, 37
dpp, 36
dS, 145, 146, 147
dt, 22
k, 13, 34, 35, 38
kL, 12
kvL, 36
lmv, 145, 147
ln, 147
s, 145, 146
ub, 33
version, 19

Memory Dump Analysis Anthology

Volume 11

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2018 by Dmitry Vostokov

Copyright © 2018 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-912636-11-2 (Paperback)

First printing, 2018

Revision 1.0 (October 2018)

Table of Contents

Preface	9
About the Author	11
PART 1: Crash Dump Analysis Patterns	13
System Call	13
Data Correlation (CPU Times)	15
Stack Trace Race	17
Hyperdump	19
Disassembly Ambiguity	24
Hidden Exception (Managed Space)	25
Insufficient Memory (Stack).....	28
Exception Reporting Thread	34
PART 2: Software Trace Analysis Patterns	35
Ultrasimilar Messages.....	35
Hedges	37
Trace Field.....	39
Script Messages	41
Working Set.....	42
Trace Homotopy	44
Signal.....	45
Renormalization.....	47
Motivic Trace	48
Significant Interval	50
Random Data	52
Truncated Data	53
Time Scale	55
Trace Sharding	56
Phantom Activity.....	57
PART 3: Software Diagnostics, Root Cause Analysis, Troubleshooting, and Debugging	59

The Most Important Skill in Software Diagnostics	59
Pattern-Oriented Data Analysis Example.....	61
Diagnostic Operads	63
Mathematical Concepts in Software Diagnostics and Software Data Analysis..	66
Software Diagnostics Engineering	69
Narrachain	71
Diagnostics-Driven Development	74
Integral Diamathics – Tracing the Road to Root Cause.....	75
Anolog.io.....	77
Meso-problem Solving using Meso-patterns.....	79
Lego Log Analysis	82
Artificial Chemistry Approach to Software Trace and Log Analysis	87
PART 4: Fun with Debugging, Crash Dumps, and Traces.....	93
Debugging Slang.....	93
<i>QUICK</i>	93
<i>DREAM</i>	93
<i>DOSE</i>	93
<i>Fex</i>	93
<i>DANCE</i>	94
<i>CORPSE</i>	94
<i>Hi</i>	94
<i>SOS</i>	94
<i>GUT</i>	94
<i>Autopsy</i>	95
<i>FILOsophy and FILOlogy</i>	95
<i>Timesheets and Timeshits</i>	95
<i>Software Logomancy</i>	95
<i>RhaPSODy</i>	95
<i>Developer</i>	96
Diagnostics and Debugging in Science Fiction	97
Russian-English Lexical Connection	98
Defects in Logs	98
A Space Makes a Difference.....	98
Digital Transformation	98
Pathology	98

Freemallocers.....	99
Double Layoff.....	99
Dump in Morse Code	99
Pushkin and Updates	99
Law of Misprints.....	99
SoftwareLog, MaintainLog, MountainLog.....	100
Cloud Patches	101
Puree Windows Cooking.....	102
 <i>An Edible CPU Chip</i>	102
 Trying on a Sherlock's Hat.....	103
The Reality is a Matrix.....	105
Problem Solving Exercises.....	106
Counting to 10	107
Music for Debugging	107
A Bug Climbs a Book	108
A Computer Crash.....	109
Traces on Roads	110
Lego Dump Analysis	112
 PART 5: Art and Photography	113
Cyberspace Diagnostics	113
Chasing a Beautiful Bug	114
Diagnostics and Poetry	116
Diagnostics in a Wild	117
Interview Preparation Deque	118
Practicing Sorting Algorithms.....	119
The Open/Closed Principle	120
Microsoft Campus in Redmond	121
Soviet Glasses and Apple Hardware	123
Design/Testing Proportion.....	124
 PART 6: Debugging Dictionary	125
7	125
8	128
Breakpoint.....	131

Crash	133
Hang.....	134
Kernel Space	135
Memory Dump.....	137
Memory Space.....	138
Physical Memory.....	139
Thread	140
User Space	141
Virtual Memory.....	143
PART 7: Tools.....	145
Dump2Picture Version 2.0.....	145
Window2Dump.....	149
WindowHistory	150
WindowHistory Mobile	155
MessageHistory	159
ScreenHistory.....	161
ProcessHistory	164
Using SSSL Principle to Design Support Tools	165
Repair Clipboard Chain	166
The Inception of Debugging Studio.....	168
Easter Egg.....	169
PART 8: Miscellaneous.....	171
Selected Crashes from My Computers.....	171
WinDbg Notes.....	211
Resume and CV as Memory Analysis Artifacts and General Traces.....	219
Quotes.....	220
My Road to Modern C++.....	222
Algorithms for Breakfast	225
Applying API Wrapper Pattern.....	229
Clipboard Issues Explained.....	233
Inside Citrix - November 2006	238
Looking at Software Problems from a Different Angle	243
Me and "Windows NT/2000 Native API Reference"	244
Appendix	245

Curriculum Vitae	245
Resume in WinDbg Style	254
Resume in GDB Style	256
Windows Internals Certificate	258
Dump2Picture 2.2.3 Source Code	259
Dump2Wave 1.3.3 Source Code	262
Window2Dump 1.0 Source Code	265
Index of WinDbg Commands	273

Index of WinDbg Commands

!address, 19, 172
!alpc, 215
!analyze -v, 17
!chkimg, 214
!CLRStack, 26
!dh, 20
!DumpStackObjects, 27
!error, 30, 204
!for_each_process, 214
!gle, 216
!heap, 184, 211
!runaway, 15, 177, 185, 206
!running, 218
!sprocess, 216
!stacks, 18
!teb, 26, 30, 33, 216
!thread, 215
!vm, 175
.chain, 212
.cxr, 191, 195, 196
.ecxr, 34, 209, 210, 215
.exr, 192, 215, 216
.formats, 191, 212
.frame, 33
.imgscan, 20
.lastevent, 28
.process, 214
.trap, 215, 218
~*k, 25, 208, 209
~<n>k, 29, 34, 172, 177, 199
~<n>s, 26, 30, 206, 215
a, 213
dps, 30
dpS, 31
g, 214
k, 17, 24, 176, 183, 190, 191, 195, 200,
201, 202, 210, 218
kc, 28, 32, 34, 171, 172, 173, 178, 180,
185, 187, 188, 192, 193, 196, 197,
203, 205, 206, 207, 213
lm, 126, 129
lmt, 174
imu, 218
lmv, 217, 218
r, 30, 188
s-a, 213
s-su, 33
ub, 14, 24
x, 189, 214

Memory Dump Analysis Anthology

Volume 12

Dmitry Vostokov
Software Diagnostics Institute

OpenTask

Published by OpenTask, Republic of Ireland

Copyright © 2019 by Dmitry Vostokov

Copyright © 2019 by Software Diagnostics Institute

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.

You must not circulate this book in any other binding or cover, and you must impose the same condition on any acquirer.

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: 978-1-912636-12-9 (Paperback)

First printing, 2019

Revision 1.01 (December 2019)

Table of Contents

Preface	7
About the Author	9
PART 1: Crash Dump Analysis Patterns	11
Active Space	11
Stack Overflow (Insufficient Memory)	13
Subsystem Modules	17
Region Profile	18
Region Clusters	19
Source Stack Trace	23
PART 2: Pattern Interaction	25
Abridged Dump, C++ Exception, Incorrect Stack Trace, Stack Trace Collection, Exception Stack Trace and Not My Version	25
Python Crash Dump Analysis Case Study	29
PART 3: Software Trace Analysis Patterns	37
Critical Point.....	37
Drone Message	39
Minimal Trace	41
Polytrace	43
Trace String	44
Equivalent Messages.....	47
Cartesian Trace	48
Message Annotations	50
CoTrace (CoLog, CoData)	51
Moduli Trace	54
Trace Similarity	57
Explanation Trace.....	59
Split Message	60
Phase Transition.....	62
Message Flow	65

Trace Constants	66
PART 4: Software Diagnostics, Root Cause Analysis, Troubleshooting, and Debugging	67
Introducing Software Pathology	67
10 Years of Trace and Log Analysis Patterns.....	69
Log's Loxels and Trace Message's Mexels Graphical Representation of Software Traces and Logs.....	74
Analysis Pattern Duality	80
10 Years of Software Narratology	81
Application of Trace and Log Analysis Patterns to Image Analysis: Introducing Space-like Narratology	86
Machine Learning Square and Software Diagnostics Institute Roadmap	88
PART 5: Fun with Debugging, Crash Dumps, and Traces.....	91
Debugging Slang.....	91
SANTA	91
<i>Call Center</i>	91
<i>Fu</i>	91
<i>ID.</i>	91
Clear Message.....	92
10-Year Challenge	93
Space Matters	97
Loop with a Bug	98
Dalkey	98
Ominous PID	98
Blue Screen at 32K Feet	99
Music for Debugging	100
Machine Learning	101
PART 6: Art and Photography	103
Crash Dump Collection.....	103
Failure Code	104
Automated Debugging	105

Real and Artificial Bugs	106
Two Windows	107
Tracing Tools	108
Log.....	109
Threads	110
Window.....	111
Linux Trace	112
Logs.....	113
Stop Code.....	114
Monolithic Architecture Leak	115
Visual Studios.....	116
Data Structures	117
Early Data Visualization.....	118
Early Data Science, Clustering, and Histogramming	119
Doing Exercise 0 in Sports Club.....	120
Burger Trace Frames and Adjoint Space of Chips	121
Patterns of Macro and Micro.....	122
Convex Programming Layout.....	123
PART 7: Book Covers.....	125
Writing Bad Code: Software Defect Construction, Simulation and Modeling of Software Bugs	125
Software Internals for Machine Learning	126
PART 8: Miscellaneous.....	127
Selected Crashes from My Computers.....	127
WinDbg Notes.....	129
Quotes.....	135
In a Chemical Laboratory	136
Historical Reminiscences	137
Baseplate Representation of Chemical Structure	138
What I'm passionate about?.....	148
C++ as a Scripting Tool	152
In Memory	155
The Road to Linux Kernel Space.....	156
Appendix	159

Stack Trace Patterns	159
Volume Index	161
Memory Analysis Patterns	161
Trace and Log Analysis Patterns	172
Index of WinDbg Commands	179

Index of WinDbg Commands

!address, 14
!analyze -v, 26, 94, 133
!blackboxbsd, 134
!blackboxntfs, 134
!blackboxpnp, 134
!dh, 129
!error, 14, 15
!mrt100sos, 130
!runaway, 11, 127
!teb, 15, 135
!vm, 133
.cxr, 26, 27, 32
.ecxr, 32
.exr, 13
.reload, 25, 135
.symfix, 25
.sympath, 32, 93
~*, 131
~*kc, 14
~s, 15, 27
dt, 135
k, 23, 26, 27, 30
kc, 13, 127, 128, 130
kL, 11, 32
lmf, 17
lmt, 129
lmv, 28, 31, 129
r, 13, 127

Volume Index

Memory Analysis Patterns

Abridged Dump	5
Accidental Lock	1
Activation Context	6
Active Space	12
Active Thread (Linux)	9
Active Thread (Mac OS X)	7
Active Thread (Windows)	9
Activity Resonance	7
Affine Thread	2
Aggregated Frames	10
Annotated Disassembly (JIT .NET code)	6
Blocked DPC	7
Blocked Queue (LPC/ALPC)	6
Blocked Thread (hardware)	5
Blocked Thread (software)	2
Blocked Thread (timeout)	7
Blocking File	6
Blocking Module	6
Broken Link	7
Busy System	1
C++ Exception	3
C++ Exception (Linux)	9
C++ Exception (Mac OS X)	7
Caller-n-Callee	6
Changed Environment	1
Clone Dump	9
Cloud Environment	6
CLR Thread	4
Coincidental Error Code	7
Coincidental Frames	5
Coincidental Symbolic Information	1
Coincidental Symbolic Information (Linux)	9
Coincidental Symbolic Information (Mac OS X)	7

Constant Subtrace	9
Corrupt Dump	2
Corrupt Structure	8
Coupled Machines	5
Coupled Modules	6
Coupled Processes (semantics)	5
Coupled Processes (strong)	1
Coupled Processes (weak)	5
Crash Signature	6
Crash Signature Invariant	7
Crashed Process	8
Critical Region (Linux)	9
Critical Section Corruption	2
Critical Stack Trace	7
Custom Exception Handler (kernel space)	6
Custom Exception Handler (user space)	1
Data Alignment (page boundary)	3
Data Contents Locality	2
Data Correlation (CPU times)	11
Data Correlation (function parameters)	6
Deadlock (critical sections)	1
Deadlock (executive resources)	1
Deadlock (.NET Finalizer)	10
Deadlock (LPC)	1
Deadlock (managed space)	6
Deadlock (mixed objects, kernel space)	3
Deadlock (mixed objects, user space)	1
Deadlock (self)	5
Debugger Bug	7
Debugger Omission	7
Design Value	8
Deviant Module	7
Deviant Token	7
Diachronic Module	9
Dialog Box	6
Directing Module	6
Disassembly Ambiguity	11
Disconnected Network Adapter	7

Disk Packet Buildup	7
Dispatch Level Spin	2
Distributed Exception (managed code)	8
Distributed Spike	5
Distributed Wait Chain	7
Divide by Zero (kernel mode)	6
Divide by Zero (user mode)	3
Divide by Zero (user mode. Linux)	9
Divide by Zero (user mode. Mac OS X)	7
Double Free (kernel pool)	1
Double Free (process heap)	1
Double Free (process heap, Mac OS X)	7
Double IRP Completion	6
Driver Device Collection (malware)	7
Dry Weight	8
Dual Stack Trace	6
Duplicate Extension	6
Duplicated Module	2
Dynamic Memory Corruption (kernel pool)	2
Dynamic Memory Corruption (managed heap)	6
Dynamic Memory Corruption (process heap)	1
Dynamic Memory Corruption (process heap, Linux)	9
Dynamic Memory Corruption (process heap, Mac OS X)	7
Early Crash Dump	1
Effect Component	4
Embedded Comments	6
Empty Stack Trace	7
Environment Hint	6
Error Reporting Fault	7
Evental Dumps	9
Exception Module	8
Exception Stack Trace	5
Exception Reporting Thread	11
Execution Residue (Linux)	9
Execution Residue (Mac OS X)	7
Execution Residue (managed space)	6
Execution Residue (unmanaged space)	2
Fake Module (malware)	7

False Effective Address	7
False Function Parameters	2
False Positive Dump	1
Fat Process Dump	6
Fault Context	5
First Fault Stack Trace	7
Foreign Module Frame	10
FPU Exception	7
Frame Pointer Omission	2
Frame Regularity	10
Frozen Process	7
Ghost Thread	8
Glued Stack Trace	7
Handle Leak	7
Handle Limit (GDI, kernel space)	2
Handle Limit (GDI, user space)	9
Handled Exception (.NET CLR)	6
Handled Exception (kernel space)	7
Handled Exception (user space)	6
Hardware Activity	5
Hardware Error	2
Hidden Call	8
Hidden Exception (kernel space)	7
Hidden Exception (managed space)	11
Hidden Exception (user space)	1
Hidden IRP	8
Hidden Module	2
Hidden Parameter	7
Hidden Process	7
Hidden Stack Trace	9
High Contention (.NET Heap)	10
High Contention (.NET CLR monitors)	7
High Contention (critical sections)	2
High Contention (executive resources)	1
High Contention (processors)	5
Historical Information	1
Hooked Functions (kernel space)	5
Hooked Functions (user space)	1

Hooked Modules	2
Hooking Level	6
Hyperdump	11
Incomplete Stack Trace (Mac OS X)	7
Incomplete Session	7
Inconsistent Dump	1
Incorrect Stack Trace	1
Incorrect Symbolic Information	5
Injected Symbols	7
Inline Function Optimization (managed code)	6
Inline Function Optimization (unmanaged code)	2
Instrumentation Information	5
Instrumentation Side Effect	6
Insufficient Memory (committed memory)	1
Insufficient Memory (control blocks)	4
Insufficient Memory (handle leak)	1
Insufficient Memory (kernel pool)	1
Insufficient Memory (module fragmentation)	2
Insufficient Memory (physical memory)	3
Insufficient Memory (PTE)	2
Insufficient Memory (region)	8
Insufficient Memory (reserved virtual memory)	7
Insufficient Memory (session pool)	7
Insufficient Memory (stack)	11
Insufficient Memory (stack trace database)	8
Internal Stack Trace	8
Invalid Exception Information	5
Invalid Handle (general)	2
Invalid Handle (managed space)	8
Invalid Parameter (process heap)	6
Invalid Parameter (runtime function)	10
Invalid Pointer (general)	1
JIT Code (.NET)	3
JIT Code (Java)	9
Last Error Collection	2
Last Object	8
Late Crash Dump	7
Lateral Damage	1

Lateral Damage (Linux)	9
Least Common Frame	6
Livelock	6
Local Buffer Overflow	1
Local Buffer Overflow (Linux)	9
Local Buffer Overflow (Mac OS X)	7
Lost Opportunity	2
Main Thread	1
Managed Code Exception	1
Managed Stack Trace	6
Manual Dump (kernel)	1
Manual Dump (process)	1
Memory Fibration	10
Memory Fluctuation (process heap)	8
Memory Leak (.NET heap)	1
Memory Leak (I/O completion packets)	7
Memory Leak (page tables)	7
Memory Leak (process heap)	1
Memory Leak (regions)	8
Message Box	2
Message Hooks	5
Mirror Dump Set	10
Missing Component (general)	2
Missing Component (static linking, user mode)	2
Missing Process	4
Missing Thread	1
Mixed Exception	4
Module Collection	7
Module Collection (predicate)	7
Module Hint	6
Module Product Process	7
Module Stack Trace (Linux)	10
Module Stack Trace (Windows)	10
Module Variable	7
Module Variety	1
Multiple Exceptions (kernel mode)	3
Multiple Exceptions (Mac OS X)	7
Multiple Exceptions (managed space)	6

Multiple Exceptions (stowed)	9
Multiple Exceptions (user mode)	1
Namespace (malware)	7
Nested Exceptions (managed code)	2
Nested Exceptions (unmanaged code)	2
Nested Offender	4
Network Packet Buildup	7
No Component Symbols	1
No Current Thread	7
No Data Types	6
No Process Dumps	2
No System Dumps	2
Not My Thread	9
Not My Version (hardware)	4
Not My Version (software)	2
NULL Pointer (code)	2
NULL Pointer (code, Linux)	9
NULL Pointer (code, Mac OS X)	7
NULL Pointer (data)	3
NULL Pointer (data, Linux)	9
NULL Pointer (data, Mac OS X)	7
Object Distribution Anomaly (.NET heap)	9
Object Distribution Anomaly (IRP)	1
OMAP Code Optimization	1
One-Thread Process	7
Optimized Code	1
Optimized VM Layout	2
Origin Module	8
Out-of-Module Pointer (malware)	7
Overaged System	2
Packed Code (malware)	7
Paged Out Data	3
Parameter Flow	9
Paratext	7
Paratext (Linux)	9
Pass Through Function	3
Passive System Thread (kernel space)	1
Passive Thread (user space)	1

Past Stack Trace	8
Patched Code (malware)	7
Pervasive System	5
Place Trace	9
Platform-Specific Debugger	4
Pleiades	6
Pre-Obfuscation Residue (malware)	7
Problem Exception Handler	5
Problem Module	7
Problem Vocabulary	6
Process Factory	3
Punctuated Memory Leak	7
Quotient Stack Trace	10
Quiet Dump	6
Random Object	4
Raw Pointer (malware)	7
Reduced Symbolic Information	7
Reference Leak	8
Region Clusters	12
Region Profile	12
Regular Data	7
Relative Memory Leak	9
RIP Stack Trace (malware)	7
Rough Stack Trace	8
Same Vendor	5
Screwbolt Wait Chain	7
Self-Diagnosis (kernel mode)	6
Self-Diagnosis (registry)	7
Self-Diagnosis (user mode)	2
Self-Dump	2
Semantic Split	3
Semantic Structure (PID.TID)	6
Shared Buffer Overwrite	5
Shared Buffer Overwrite (Mac OS X)	7
Shared Structure	7
Small Value	7
Software Exception	8
Source Stack Trace	12

Special Process	2
Special Stack Trace	1
Special Thread (.NET CLR)	6
Spike Interval	7
Spiking Thread	1
Spiking Thread (Linux)	9
Spiking Thread (Mac OS X)	7
Stack Overflow (insufficient memory)	12
Stack Overflow (kernel mode)	1
Stack Overflow (software implementation)	6
Stack Overflow (user mode)	2
Stack Overflow (user mode, Linux)	9
Stack Overflow (user mode, Mac OS X)	7
Stack Trace	1
Stack Trace (database)	8
Stack Trace (file system filters)	8
Stack Trace (I/O Devices)	10
Stack Trace (I/O request)	8
Stack Trace (Linux)	9
Stack Trace (Mac OS X)	7
Stack Trace Change	7
Stack Trace Collection (CPUs)	9
Stack Trace Collection (I/O requests)	7
Stack Trace Collection (managed space)	6
Stack Trace Collection (predicate)	7
Stack Trace Collection (unmanaged space)	1
Stack Trace Motif	10
Stack Trace Race	11
Stack Trace Set	6
Stack Trace Signature	9
Stack Trace Surface	9
Step Dumps	7
Stored Exception	7
String Hint (malware)	7
String Parameter	6
Subsystem Modules	12
Suspended Thread	2
Swarm of Shared Locks	3

System Call	11
System Object	7
Tampered Dump	8
Technology-Specific Subtrace (COM client call)	8
Technology-Specific Subtrace (COM interface invocation)	6
Technology-Specific Subtrace (dynamic memory)	6
Technology-Specific Subtrace (JIT .NET code)	6
Template Module	5
Thread Age	6
Thread Cluster	7
Thread Poset	8
Thread Starvation (normal priority)	5
Thread Starvation (realtime priority)	2
Top Module	6
Translated Exception	7
Truncated Dump	1
Truncated Dump (Mac OS X)	7
Truncated Stack Trace	6
Ubiquitous Component (kernel space)	7
Ubiquitous Component (user space)	4
Unified Stack Trace	10
Unknown Component	1
Unloaded Module	7
Unrecognizable Symbolic Information	7
Unsynchronized Dumps	6
User Space Evidence	8
Value Adding Process	7
Value Deviation (stack trace)	4
Value Deviation (structure field)	10
Value References	7
Variable Subtrace	8
Version-Specific Extension	6
Virtualized Process (WOW64)	1
Virtualized System	4
Wait Chain (C++11, condition variable)	9
Wait Chain (CLR monitors)	7
Wait Chain (critical sections)	1
Wait Chain (executive resources)	2

Wait Chain (general)	1
Wait Chain (LPC/ALPC)	3
Wait Chain (modules)	8
Wait Chain (mutex objects)	6
Wait Chain (named pipes)	6
Wait Chain (Nonstandard Synchronization)	9
Wait Chain (process objects)	5
Wait Chain (pushlocks)	7
Wait Chain (RPC)	5
Wait Chain (RTL_RESOURCE)	8
Wait Chain (SRW Lock)	10
Wait Chain (thread objects)	3
Wait Chain (window messaging)	6
Waiting Thread Time (kernel dumps)	1
Waiting Thread Time (user dumps)	2
Well-Tested Function	4
Well-Tested Module	6
Wild Code	2
Wild Pointer	2
Window Hint	9
Young System	2
Zombie Processes	2

Trace and Log Analysis Patterns

Abnormal Value	7
Activity Disruption	8
Activity Divergence	7
Activity Overlap	8
Activity Packet	10
Activity Quantum	10
Activity Region	4
Activity Theatre	9
Adjoint Message	9
Adjoint Space	8
Adjoint Thread of Activity	5
Anchor Messages	5
Back Trace	8
Background and Foreground Components	5
Basic Facts	3
Bifurcation Point	4
Blackout	8
Braid Group	10
Braid of Activity	10
Break-in Activity	7
Calibrating Trace	9
Cartesian Trace	12
Characteristic Message Block	4
Circular Trace	3
Combed Trace	10
Correlated Discontinuity	7
Corrupt Message	10
CoTrace (CoLog, CoData)	12
Counter Value	7
Coupled Activities	9
Critical Point	12

Data Association	7
Data Flow	7
Data Interval	9
Data Reversal	8
Data Selector	9
De Broglie Trace Duality	10
Declarative Trace	9
Defamiliarizing Effect	5
Delay Dynamics	10
Density Distribution	7
Dialogue	7
Diegetic Messages	5
Discontinuity	4
Dominant Event Sequence	7
Drone Message	12
Empty Trace	7
Equivalent Messages	12
Error Distribution	7
Error Message	7
Error Powerset	9
Error Thread	7
Event Sequence Order	6
Event Sequence Phase	8
Exception Stack Trace	4
Explanation Trace	12
Factor Group	7
False Positive Error	5
Fiber Bundle	7
Fiber of Activity	9
File Size	8
Focus of Tracing	6
Fourier Activity	9
Galois Trace	10

Glued Activity (ATID reuse)	6
Gossip	6
Guest Component	5
Hedges	11
Hidden Error	7
Hidden Facts	8
Identification Messages	9
Implementation Discourse	6
Impossible Trace	6
Incomplete History	5
Indexical Trace	7
Indirect Facts	7
Indirect Message	8
Inter-Correlation	4
Intra-Correlation	3
Intrinsic ID	10
Interspace	8
Last Activity	7
Layered Periodization	5
Linked Messages	7
Macrofunction	7
Marked Message	7
Master Trace	6
Message Annotations	12
Message Change	5
Message Context	7
Message Cover	7
Message Directory	10
Message Flow	12
Message Interleave	7
Message Invariant	6
Message Pattern	9
Message Set	7

Message Space	8
Meta Trace	7
Milestones	8
Minimal Trace	12
Missing Component	4
Missing Data	9
Missing Message	8
Moduli Trace	12
Motif	7
Motivic Trace	11
News Value	6
No Activity	5
No Trace Metafile	5
Opposition Messages	7
Original Message	6
Ornament	10
Palimpsest Messages	8
Periodic Error	3
Periodic Message Block	7
Phantom Activity	11
Phase Transition	12
Piecewise Activity	7
Pivot Message	7
Poincaré Trace	10
Polytrace	12
Projective Space	10
Punctuated Activity	8
Quotient Trace	9
Random Data	11
Recovered Messages	8
Relative Density	7
Renormalization	11
Resume Activity	7

Ruptured Trace	9
Script Messages	11
Sequence Repeat Anomaly	9
Shared Point	7
Sheaf of Activities	7
Signal	11
Significant Event	5
Significant Interval	11
Silent Messages	7
Singleton Event	8
Singleton Trace	10
Small DA+TA	9
Sparse Trace	7
Split Message	12
Split Trace	7
State and Event	7
State Dump	7
Statement Density and Current	4
Surveyor	9
Tensor Trace	10
Thread of Activity	4
Time Delta	5
Time Scale	11
Timeout	8
Trace Acceleration	5
Trace Constants	12
Trace Dimension	9
Trace Extension	9
Trace Field	11
Trace Frames	7
Trace Homotopy	11
Trace Mask	8
Trace Partition	5

Trace Presheaf	10
Trace Sharding	11
Trace Similarity	12
Trace String	12
Trace Viewpoints	8
Traces of Individuality	7
Translated Message	8
Truncated Data	11
Truncated Trace	5
Unsynchronized Traces	10
UI Message	6
Ultrasimilar Messages	11
Use Case Trail	8
Visibility Limit	7
Visitor Trace	8
Vocabulary Index	4
Watch Thread	8
Working Set	11