Contents

About the Authors ..... *xviii*
Foreword ..... *xx*
Introduction ..... *xxiii*

Chapter 8 System mechanisms **1**

Processor execution model 2
Segmentation 2
Task state segments 6
Hardware side-channel vulnerabilities 9
Out-of-order execution 10
The CPU branch predictor 11
The CPU cache(s) 12
Side-channel attacks 13
Side-channel mitigations in Windows 18
KVA Shadow 18
Hardware indirect branch controls (IBRS, IBPB, STIBP, SSBD) 21
Retpoline and import optimization 23
STIBP pairing 26
Trap dispatching 30
Interrupt dispatching 32
Line-based versus message signaled–based interrupts 50
Timer processing 66
System worker threads 81
Exception dispatching 85
System service handling 91
WoW64 (Windows-on-Windows) 104
The WoW64 core 106
File system redirection 109
Registry redirection 110
X86 simulation on AMD64 platforms 111
ARM 113

Memory models 114
ARM32 simulation on ARM64 platforms 115
X86 simulation on ARM64 platforms 115
Object Manager 125
Executive objects 127
Object structure 131
Synchronization 170
High-IRQL synchronization 172
Low-IRQL synchronization 177
Advanced local procedure call 209
Connection model 210
Message model 212
Asynchronous operation 214
Views, regions, and sections 215
Attributes 216
Blobs, handles, and resources 217
Handle passing 218
Security 219
Performance 220
Power management 221
ALPC direct event attribute 222
Debugging and tracing 222
Windows Notification Facility 224
WNF features 225
WNF users 226
WNF state names and storage 233
WNF event aggregation 237
User-mode debugging 239
Kernel support 239
Native support 240
Windows subsystem support 242
Packaged applications 243
UWP applications 245
Centennial applications 246

The Host Activity Manager....249
The State Repository....251
The Dependency Mini Repository....255
Background tasks and the Broker Infrastructure....256
Packaged applications setup and startup....258
Package activation....259
Package registration....265
Conclusion....266

Chapter 9 Virtualization technologies 267

The Windows hypervisor....267
Partitions, processes, and threads....269
The hypervisor startup....274
The hypervisor memory manager....279
Hyper-V schedulers....287
Hypercalls and the hypervisor TLFS....299
Intercepts....300
The synthetic interrupt controller (SynIC)....301
The Windows hypervisor platform API and EXO partitions....304
Nested virtualization....307
The Windows hypervisor on ARM64....313
The virtualization stack....315
Virtual machine manager service and worker processes....315
The VID driver and the virtualization stack memory manager....317
The birth of a Virtual Machine (VM)....318
VMBus....323
Virtual hardware support....329
VA-backed virtual machines....336
Virtualization-based security (VBS)....340
Virtual trust levels (VTLs) and Virtual Secure Mode (VSM)....340
Services provided by the VSM and requirements....342
The Secure Kernel....345
Virtual interrupts....345
Secure intercepts....348

VSM system calls 349
Secure threads and scheduling 356
The Hypervisor Enforced Code Integrity 358
UEFI runtime virtualization 358
VSM startup 360
The Secure Kernel memory manager 363
Hot patching 368
Isolated User Mode 371
Trustlets creation 372
Secure devices 376
VBS-based enclaves 378
System Guard runtime attestation 386
Conclusion 390

Chapter 10 Management, diagnostics, and tracing 391

The registry 391
Viewing and changing the registry 391
Registry usage 392
Registry data types 393
Registry logical structure 394
Application hives 402
Transactional Registry (TxR) 403
Monitoring registry activity 404
Process Monitor internals 405
Registry internals 406
Hive reorganization 414
The registry namespace and operation 415
Stable storage 418
Registry filtering 422
Registry virtualization 422
Registry optimizations 425
Windows services 426
Service applications 426
Service accounts 433
The Service Control Manager (SCM) 446

Service control programs 450
Autostart services startup 451
Delayed autostart services 457
Triggered-start services 458
Startup errors 459
Accepting the boot and last known good 460
Service failures 462
Service shutdown 464
Shared service processes 465
Service tags 468
User services 469
Packaged services 473
Protected services 474
Task scheduling and UBPM 475
The Task Scheduler 476
Unified Background Process Manager (UBPM) 481
Task Scheduler COM interfaces 486
Windows Management Instrumentation 486
WMI architecture 487
WMI providers 488
The Common Information Model and the Managed Object Format Language 489
Class association 493
WMI implementation 496
WMI security 498
Event Tracing for Windows (ETW) 499
ETW initialization 501
ETW sessions 502
ETW providers 506
Providing events 509
ETW Logger thread 511
Consuming events 512
System loggers 516
ETW security 522

Dynamic tracing (DTrace) 525
Internal architecture 528
DTrace type library 534
Windows Error Reporting (WER) 535
User applications crashes 537
Kernel-mode (system) crashes 543
Process hang detection 551
Global flags 554
Kernel shims 557
Shim engine initialization 557
The shim database 559
Driver shims 560
Device shims 564
Conclusion 564

Chapter 11 Caching and file systems 565

Terminology 565
Key features of the cache manager 566
Single, centralized system cache 567
The memory manager 567
Cache coherency 568
Virtual block caching 569
Stream-based caching 569
Recoverable file system support 570
NTFS MFT working set enhancements 571
Memory partitions support 571
Cache virtual memory management 572
Cache size 574
Cache virtual size 574
Cache working set size 574
Cache physical size 574
Cache data structures 576
Systemwide cache data structures 576
Per-file cache data structures 579

File system interfaces 582
Copying to and from the cache 584
Caching with the mapping and pinning interfaces 584
Caching with the direct memory access interfaces 584
Fast I/O 585
Read-ahead and write-behind 586
Intelligent read-ahead 587
Read-ahead enhancements 588
Write-back caching and lazy writing 589
Disabling lazy writing for a file 595
Forcing the cache to write through to disk 595
Flushing mapped files 595
Write throttling 596
System threads 597
Aggressive write behind and low-priority lazy writes 598
Dynamic memory 599
Cache manager disk I/O accounting 600
File systems 602
Windows file system formats 602
CDFS 602
UDF 603
FAT12, FAT16, and FAT32 603
exFAT 606
NTFS 606
ReFS 608
File system driver architecture 608
Local FSDs 608
Remote FSDs 610
File system operations 618
Explicit file I/O 619
Memory manager's modified and mapped page writer 622
Cache manager's lazy writer 622
Cache manager's read-ahead thread 622
Memory manager's page fault handler 623
File system filter drivers and minifilters 623

Filtering named pipes and mailslots 625
Controlling reparse point behavior 626
Process Monitor 627
The NT File System (NTFS) 628
High-end file system requirements 628
Recoverability 629
Security 629
Data redundancy and fault tolerance 629
Advanced features of NTFS 630
Multiple data streams 631
Unicode-based names 633
General indexing facility 633
Dynamic bad-cluster remapping 633
Hard links 634
Symbolic (soft) links and junctions 634
Compression and sparse files 637
Change logging 637
Per-user volume quotas 638
Link tracking 639
Encryption 640
POSIX-style delete semantics 641
Defragmentation 643
Dynamic partitioning 646
NTFS support for tiered volumes 647
NTFS file system driver 652
NTFS on-disk structure 654
Volumes 655
Clusters 655
Master file table 656
File record numbers 660
File records 661
File names 664
Tunneling 666
Resident and nonresident attributes 667
Data compression and sparse files 670

Compressing sparse data 671
Compressing nonsparse data 673
Sparse files 675
The change journal file 675
Indexing 679
Object IDs 681
Quota tracking 681
Consolidated security 682
Reparse points 684
Storage reserves and NTFS reservations 685
Transaction support 688
Isolation 689
Transactional APIs 690
On-disk implementation 691
Logging implementation 693
NTFS recovery support 694
Design 694
Metadata logging 695
Log file service 695
Log record types 697
Recovery 699
Analysis pass 700
Redo pass 701
Undo pass 701
NTFS bad-cluster recovery 703
Self-healing 706
Online check-disk and fast repair 707
Encrypted file system 710
Encrypting a file for the first time 713
The decryption process 715
Backing up encrypted files 716
Copying encrypted files 717
BitLocker encryption offload 717
Online encryption support 719

Direct Access (DAX) disks 720
DAX driver model 721
DAX volumes 722
Cached and noncached I/O in DAX volumes 723
Mapping of executable images 724
Block volumes 728
File system filter drivers and DAX 730
Flushing DAX mode I/Os 731
Large and huge pages support 732
Virtual PM disks and storages spaces support 736
Resilient File System (ReFS) 739
Minstore architecture 740
B+ tree physical layout 742
Allocators 743
Page table 745
Minstore I/O 746
ReFS architecture 748
ReFS on-disk structure 751
Object IDs 752
Security and change journal 753
ReFS advanced features 754
File's block cloning (snapshot support) and sparse VDL 754
ReFS write-through 757
ReFS recovery support 759
Leak detection 761
Shingled magnetic recording (SMR) volumes 762
ReFS support for tiered volumes and SMR 764
Container compaction 766
Compression and ghosting 769
Storage Spaces 770
Spaces internal architecture 771
Services provided by Spaces 772
Conclusion 776

Chapter 12 Startup and shutdown 777

Boot process 777
The UEFI boot 777
The BIOS boot process 781
Secure Boot 781
The Windows Boot Manager 785
The Boot menu 799
Launching a boot application 800
Measured Boot 801
Trusted execution 805
The Windows OS Loader 808
Booting from iSCSI 811
The hypervisor loader 811
VSM startup policy 813
The Secure Launch 816
Initializing the kernel and executive subsystems 818
Kernel initialization phase 1 824
Smss, Csrss, and Wininit 830
ReadyBoot 835
Images that start automatically 837
Shutdown 837
Hibernation and Fast Startup 840
Windows Recovery Environment (WinRE) 845
Safe mode 847
Driver loading in safe mode 848
Safe-mode-aware user programs 849
Boot status file 850

Conclusion 850

Contents of Windows Internals, Seventh Edition, Part 1 ..... *851*

Index *861*