



Index

A

ABI (application binary interface), 53
 abort(3C) function, 106
 accept() function, 873
 Access control list (ACL), 764
 modifying, 766
 Accounting
 microstate, 48
 RM, 9
 zones, 411
 ACL (access control list), 764
 modifying, 766
 Adaptive locks, 829
 probes, 847–848
 Adaptive mutex lock implementation, 21
 Address spaces, 27
 core process components, 47
 HAT, 14
 HAT, implementation, 631–636
 HAT, ISM, 613–616
 HAT, overview of, 581–583
 HAT, pages, 506
 HAT, SPARC, 620–625
 HAT, support, 500–501
 HAT, synchronization, 616–620
 HAT, UltraSPARC layer, 583–625
 HAT, VM design, 457
 HAT, x64, 625–636
 process address space, 305

procfs, 123
 selection, 397
 space, callbacks, 472–473
 space, kernels, 528
 space, levels of memory, 448
 space, mapping, 650
 space, page faults in, 473–476
 space, pmap(1) command, 642–643
 space, processes, 13
 space, SPARC systems, 459–461
 space, VM, 467–476
 space, x64/x86 layouts, 461
 virtual, address spaces, 457–466
 virtual, aliasing, 592
 virtual, kernel maps, 965–969
 virtual, space layout, 628–631
 virtual, validation, 593
 Administration. *See also Management*
 RBACs, 11
 SMF, 5
 zones, 370
 Aggregation, GLDv3 link architecture, 888–889
 Aliasing virtual addresses, 592
 Allocation
 algorithms, 719–720
 blocks, HME, 598–599
 blocks, UFS, 754–760
 bmp_write function, 758
 cycles, 720–722
 dynamic resources, 32

- Allocation (*continued*)
 - files, descriptors, 662–665
 - kernels, 721–722
 - kernels, loadable modules, 531
 - large kernel page routines, 606–607
 - memory, file systems, 718–722
 - memory, global, 27–28
 - memory, heap mapping, 462
 - memory, kernels, 534–551
 - memory, logging, 570–572
 - memory, MPSS, 9
 - memory, NUMA, 803
 - memory, physical, 503–505
 - memory, procfs, 114
 - memory, TSB, 605
 - Modern memory allocators, 453
 - pages, new, 515
 - pages, sizes, 642–644
 - physical memory, 166
 - policies, 555–556, 558
 - resources, 555–556
 - scheduling classes, 9–10
 - segments, 557–558
 - slab allocators, 29
 - vnode interfaces, 696–697
 - ZFOD, 485
- Allocators
 - buddy, 537
 - linear-time performance, 552
 - logging, 576–577
 - slab, 537–551
 - tracing, 562–577
 - vmem, 552–562
 - vmem, implementation, 556–560
 - vmem, interfaces, 553–556
 - vmem, performance, 560–561
 - vmem, properties, 553
- AMD Opteron processor support, 8
- Anonymous memory, 449, 485–486
 - layers, 487–488
- Anonymous/process allocations, 721
- APIs (application programming interfaces), 45
 - LWP process model structures, 69
 - MPO, 807–811
 - projects/tasks, 419–420
 - semaphores, 306
 - shared memory, 286
 - system calls, 98–106
- Application binary interface (ABI), 53
- Application programming interfaces. *See APIs*
- Applications
 - continued backward binary compatibility, 327
 - DTrace, 7
 - interval timers, 12
- large page sizes, selection, 639
- libthread.so (threads library), 10
- LWP, 19, 20
- MPO, 9
- MPSS, 9
- optimization, 807–811
- performance, measurement, 640–642
- privileges, modifying state, 336–337
- process models, 48
- set-uid, 325
- zones, 368–371. *See also Zones*
- apctrace(1) command, 11
- Arbitrary resolution interval timers, 12
- Architecture
 - GLDv3, 883–889
 - kernels, 16–17
 - page scanners, 523
 - parallel systems, 816–819
 - real-time, 14
 - SPARC, 4
 - SPARC, system calls, 99–101
 - UFS, 749–750
- Arenas
 - creating, 553–555
 - importing, 556
 - locks, 559
 - structures, 557
 - vmem, 937–939
- Arrays
 - cyclic, 914
 - hat_lock, 617
 - page_hash, 507
 - process models, creating, 97
 - sclass, 194
 - sleepq_head, 845
 - turnstile_table[], 841
- as_add_callback() function, 472
- as_alloc() function, 469
- as_delete_callback() function, 472
- as_do_callbacks() function, 472
- as_fault() function, 470
- as_setat() function, 470
- Assignment, resources, 6
- Association, panic messages, 570
- Asynchronous file system transactions, 783
- Asynchronous signals, 145–148
 - implementation, 26
- async_request_size parameter, 524
- Atomic instructions, locks, 821
- Attributes
 - name-service, 433
 - namespaces, 433
 - process, 419
 - process, resources, 84–89

project, 419
 rctl, 433–435
 task, 419
 terminal, 957
 UFS, 767–768
 zone, 419

Auditing
 allocators, 576–577
 privileges, 362

Authentication, 10
 Authorization, RBAVs, 11
 Automaps, 390
 Automatic operation, devices, 398

Availability, 12
 predictive self-healing, 5

Awareness
 privileges, 330–331
 privileges, state transitions, 334–335

B

Backend interfaces, cyclic, 924–925
 Backing stores, 448
 Barriers, memory, 824
 Basic privileges, 342
 bcp (buffer control pointer), 570
 Berkeley Fast File System (FFS), 737
 Binary compatibility, 52
 Binary trees, file descriptor integer space, 662
 bind() function, 872
 Binding
 #pragma directives, 91
 processors, 33
 Bit sets, 345–346
 Blocks, 828
 booting, 746
 files, mapping to disk, 758–759
 HME, allocation, 598–599
 I/O, vnode pages on, 700
 interrupts, 185
 interrupts, cookies, 832
 metadata, buffering, 760
 physical-meta-data, 754
 rehash values, 596
 resource limits, 85
 shadow HME, 597–598
 signals, 132
 superblocks, 747–748
 tasks, 929
 threads, 202
 UFS, allocation, 754–760
 UFS, reading/writing, 760

bmap_read() function, 758
 bmap_write() function, 758
 Booting
 blocks, 746
 kernels, text, 528
 physical memory, 503
 zones, 375–379
 Bound threads, 21, 49
 bread_common() function, 760
 Breakpoint management, 961
 Buddy allocators, 537
 bufctl pointers, 571–572
 Buffer control pointer (bcp), 570
 Buffers
 blocks, metadata, 760
 freed, checking, 566
 global slab layer, 547
 kernels, semaphores, 845
 producer/consumer, 917–918
 TLB, 583, 639
 TLB, cpustat command, 641–642
 TLB, iTLB, 645
 TLB, trapset(1M) command, 640–641
 troubleshooting, 575
 TSB, 531, 583–584, 601–613
 Buftag data integrity, 570–571
 Bus management, 17
 busstat(1M) command, 11
 bwrite_common() function, 760

C

Cache coherent NUMA (ccNUMA), 795, 797
 Cachelists, 505, 509
 Caches, 28
 CPU layer, 545–546
 cyclic page, 718
 cyclic subsystem, 915
 directories, names, 669
 DNLC, 726–733
 file systems, 668–669
 file systems, VM, 450–451
 inodes, 752–754
 kstat(1) command, 80
 L2, 514
 memory in use column, 79
 objects, 540–543
 pages, 504
 pages, file systems, 721
 physical memory, 449
 physical page mapping, 513
 quantum, 559

Caches (*continued*)
 slab allocators, 538
 task queue implementation, 939–940
 tracing, enabling, 562–563
 viewing, 563–565
 vnode interfaces, 698–700
 vnode interfaces, traversing, 701–703
 warm, 162
 write-through, 822

Callbacks, address space, 472–473

Callouts, clocks, 904–910
callout_schedule() function, 907
callout_table structure, 905

Calls
switch() function, 243
 system. *See* System calls

Capabilities (privileges), 331
c_arg field, 907
 cas instructions, 820, 823

Categories
 of devices, 399–400
 of UFS locks, 768

ccNUMA (cache coherent NUMA), 795, 797

Central processing unit. *See* CPU

cfork() function, 92

c_func field, 907

Change flow, priority, 212
change_pri function, 827
change_priority functions, 843
change-pri dtrace probes, 232
check_page() function, 523
 Checksums, offload, 890–891
 Child size (CSIZE), 663
 CHIP_CMP_Shared_CACHE chip, 164
 CHIP_CMP_SPLIT_CACHE chip, 164
 CHIP_DEFAULT chip, 164
 Chip multiprocessing (CMP), 797
 Chip multithreading. *See* CMT
 Chips, types, 163
 CHIP_SMT chip, 164
chip_t, 162, 168
 chroot interactions, zones, 385–385
cl_active, 200
cl_admin, 199
cl_alloc, 199
 Clamps, page scanner CPU utilization, 521

Classes
 scheduling, 9–10, 22, 160. *See also* Dispatchers;
 Scheduling
 scheduling, dispatchers, 192–207
 scheduling, frameworks, 196
 scheduling, FSS, 10, 23, 160
 scheduling, functions, 198–202
 scheduling, FX, 10, 23, 160

scheduling, IA, 9, 23, 160
 scheduling, RT, 9, 23, 160
 scheduling, SYS, 23, 260
 scheduling, TS, 9, 160
 time-based class functions, 211–214

Classifiers, 864
 IP, 868–869
cl_canexit, 199
cl_donice, 201
cl_enterclass, 199
cl_exit, 200
cl_exitclass, 199
cl_fork, 199
cl_forkret, 199
cl_free, 199
cl_getclinfo, 199
cl_getclpri, 199
cl_globpri, 201

Clients
 address space callbacks, 472–473
 cyclic subsystems, 922–923

cl_inactive, 200

Clocks, 17
 callouts, 904–910
 cyclic subsystem, 912–925
 system time facilities, 910–911
 threads, 901–904
 tick processing, 903
clock_tick() function, 213, 903
close() function, 686, 874
cl_parmsget, 200
cl_parmsin, 199
cl_parmsout, 199
cl_parmsset, 200
cl_preempt, 200
cl_set_process_group, 201
cl_setrun, 201
cl_sleep, 201
cl_stop, 200
cl_swapout, 200
cl_tick, 201
CL_TICK(t) macro, 213
cl_trapret, 200
cl_vaparmsin, 199
cl_vaparmsout, 199
cl_wakeup, 201
cl_yield, 201

CMP (chip multiprocessing), 797
 CMT (chip multithreading), 795
 overview of, 797–799
 processors, 161

Code
elfexec, 98
exec, 93

fork, 83
 locks, 820
 paths, open() function, 661
 waiters, 825
Coexistence, 325
Collectives, resource controls, 425
Coloring pages, 512–516
Columns
 kmastat, 550
 memory in use, 79
 proc_sz, 83
Command-line interfaces, MPSS, 9
Commands
 apptrace(1), 11
 busstat(1M), 11
 coreadm(1M), 10
 cpio(1), 404
 cpustat, 641–642
 cpustat(1M), 11
 dtrace, 11
 dispadmin(1), 202
 elfdump(1), 53
 fstyp, 756
 ifconfig(1M), 395–396
 ipcrm(1), 406
 ipcs(1), 291, 406
 kill(1), 145
 kstat(1), 80
 ls(1), 54
 modinfo, 531
 pagesize, 644
 PCWATCH, 492
 pfiles, 668
 pgrep, 11
 pkill(1), 11, 387
 pmap, 465–466, 642–643
 poolstat(1), 39
 ppgsz, 646
 ppgsz(1M), 646
 ppriv(1), 383
 prctl, 285
 priocnt(1), 209
 proc(1), 120
 prstat(1), 11, 128, 231
 prstat -Lc, 47
 prstat(1M), 39
 ps(1), 74, 231
 psig(1), 132
 psradm(1M), 34
 trapset(1M), 640–641
 truss(1), 11
 zlogin(1), 403
 zoneadm, 373–374
 zsched, 374
 Compatibility, 52, 325
 continued backward binary, 327
 privileges, 328
 zones, 370
Compilation flags, 52
Compilers
 file system conversion, 734–736
 large pages, 648–649
Condition variables, 63, 253–255, 824
Configuration
 arenas, 553–555
 DLPIs, 14
 DRPs, 6
 IPC, creating modules, 280–282
 IPC, framework, 275–276
 IPC, tuneables, 285
 kernels, parameters, 971–974
 kmem_flags variable, 564
 memory, 6
 MMU, 625–626
 NICs, 10
 process models, 89–98
 RCM, 453
 RW locks, 837–838
 target directories, 10
 thread priority, 211–233
 zones, 401
 zones, file systems, 389–390
Configured state (zones), 371
CONN_DEF_REF macro, 870
connect() function, 872
Connections
 costs, 863
 queues, 865
 stacks, 858–859
 structures, 868
 TCP/IP, 8. *See also* TCP/IP
 teardowns, TCP, 398
CONN_INC_REF macro, 870
Consistency
 sequential models, 822
 TSB, 618–620
Consoles
 design, zones, 402–404
 I/O, 957–958
Consolidation, 32
Constants, 345–346
 HME hash tables, 600
 privileges, 346
 PRIV_NSET, 348
 PRIV_SETSIZE, 348
Constraints, dynamic task queues, 929
Containment, 325
Context switching, 159

Contiguous physical memory segments, 510
 Continued backward binary compatibility, 327
Controls
 access, UFS, 764–767
 bcp, 570
 files, procfs, 121
 messages, procfs, 121–122
 preemption, 218
 processes, zones, 386
 processors, 33–34
 resources, 87, 161, 423–432
 resources, global, 427–428
 resources, interfaces, 432–437
 resources, kernel interfaces, 437–444
 resources, local, 428
 resources, numeric values for, 426
 resources, policies, 428–429
 resources, shared memory, 288
 resources, signals, 430–431
 resources, System V, 282–284
 resources, tasks, 431–432
 resources, zones, 411
Conventions. See Names
Conversion, file systems (to Solaris 10), 734–736
Cookies, interrupt blocks, 832
Copy-on-write (COW), 58, 490
 faults, 497–498
Copy-on-write process, 484
coreadm(1M) command, 10
Core dumps
 blocks, 85
 privileges, 360–361
 processes, 493
Core files
 management, 10
 zones, 389
Core functions, dispatchers, 159
Core kernels, 18
Core process components, 47–48
Corruption, detecting memory, 565–566
Counters
 hardware, reading, 641–642
 kstat, 935–936
 lbolt, 902
 count() function, 232
Counting
 references, 423
 references, vnode interfaces, 698
COW (copy-on-write), 58, 490
 faults, 497–498
cpio(1) command, 404
CPU (central processing unit)
 groupings, 165
 large pages, support, 652–653

layer, 545–546
 mutex locks, 833
 page scanner utilization clamp, 521
 parallel system architectures, 816–819
 partitions, 165
 processor abstractions, 162–171
 queue insertion, 236
 selecting, 236
 SMP, 795
 tracking, 170
 utilization, 48
cpu_choose() function, 236
cpupart_move_cpu() function, 246
cpupart_t, 172–173
cpu_resched() function, 239
cpustat command, 11, 641–642
cpu_surrender(), 247
cpu_t structure, 162, 166, 172
Creating. See Configuration
Credentials, 47
 zones, 380
cred_getzoneid(3c) interface, 405
cred_t structure, 380
crgetref() function, 347
cr_groups field, 348
Cross-calls, interrupts, 268–270
Cross-zone communication, 405
cr_zone fields, 380
CSIZE (child size), 663
cv_timedwait() function, 258
cv_timedwait_sig() function, 258
cv_wait() function, 258
cv_wait_sig() function, 258
cv_wait_sig_swap() function, 258
Cycles
 allocation, 720–722
 physical memory, 503–505, 719
 vnode interfaces, 696–697
Cyclic subsystem, 912–925
 page caches, 28, 718
Cylinder groups, 748–749

D

Daemons
 devfsadmd, 405
 poold, 6, 36
 private, 356–357
 reapd, 6, 39
 system page scanner, 505
Databases, projects, 418–419
Data (kbytes), resource limits, 85
Data-Link Driver (DLD), 888

- Data-Link Provider Interfaces (DLPIs), 15, 882–883
 datapath, 874
 Data segments, kernels, 528–530
 Data structures. *See also* Structures
 anonymous memory, 486
 hash tables, 589
 in-kernel project, 421–423
 ISM, 614
 kernels, resource control interfaces, 438–439
 privileges, 346–349
 scheduling classes, 193–198
 TSB, 604
 VM, 530
 watchpoints, 494
 d commands, 11
 DDIs (device driver interfaces), 14
 task queues, 934–935
 Deallocation. *See also* Allocation
 files, descriptors, 662–665
 Deathrow lists, 109–110
 Debugger/Prom Interface (DPI), 959–960
 Debugging
 buftag, 571
 caches, 540
 KDI, 946–947
 kernels, memory, 563
 maps, 781
 mdb, 11, 65, 82, 271. *See also* mdb
 modular debuggers, 11
 modules, management, 945, 953–954
 privileges, 361–362
 with redzone indicators, 566–569
 Defining roles, 10–11
 Delay (normalize usage), 222
 Delete queue, 754
 Deletion of directory entries, 744
 Delivery, signals, 135, 269
 deltamap structure, 780
 Demand-page virtual memory systems, 14
 Demand paging, 450
 Demotion, pages, 651
 Depot layer, 546–547
 Descriptors
 files, 658, 660–661
 files, allocation/deallocation, 662–665
 files, limits, 665
 Destroying
 dispatcher locks, 186
 vnode interfaces, 698
 Detection
 memory corruption, 565–566
 uninitialized data, 569
 /dev file system, read-only mount, 402
 devsadmd daemon, 405
 Device driver interfaces (DDIs), 14
 task queues, 934–935
 Devices
 automatic operation, 398
 categories, 399–400
 drivers, 882–891. *See also* Drivers
 fully virtual, 400
 management, 17, 401
 modular I/O systems, 14
 namespaces, 400
 privileges, 402
 protection, 362–363
 security, 398
 unsafe, 399
 virtualization, 398
 zones, 398–404
 /dev/ip device node, 397
 Dictionaries, process resource controls, 88–89
 Dijkstra, E. W., 295
 DIRBLKSIZ, 743
 Direct I/O read/write concurrency, UFS, 10
 Directives, #pragma binding, 91
 Directories
 DNLC caches, 729–733
 entries, deletion of, 744
 names, caches, 669
 procfs, 114
 reading, 723–724
 searching, 723
 target, configuration, 10
 UFS, 742–744
 Directory name lookup cache (DNLC), 726–733
 Dirty pages, 449
 Disk blocks, mapping files to, 758–759
 DISM (Dynamic Intimate Shared Memory), 4, 9,
 11, 294–295
 dispadmin(1) command, 202
 Dispatchers, 9–10. *See also* Scheduling
 core functions, 159
 functions, 234–245
 initialization, 190–191
 interrupts, 262–270
 kernel sleep/wakeup facility, 253–262
 locks, 183–190, 824
 locks, functions, 186–187
 locks, statistics, 189–190
 locks, threads, 187–189
 mdb(1) kernel debugging facility, 271
 overview of, 157–162
 preemption, 246–253, 268
 processors, abstractions, 162–171
 processors, observability, 168–171

- queues, management, 234–242
 structures, 172–175
 structures, linkage, 175–177
 structures, queues, 176
 structures, viewing, 177–183
 switch() function, 242–246
 tables, 202–207
 threads, priorities, 207–233
 Dispatching, priority, 929
 dispdeq() function, 234
 disp_getbest() function, 189
 disp_lowpri_cpu() function, 237
 DISP_MUST_SURRENDER() function, 230
 Dispositions, signals, 130
 dispq_t, 173
 disp_queue_info, 174
 disp_t, 174
 Distribution, pages, 515
 DLD (Data-Link Driver), 888
 DLPIs (Data-Link Provider Interfaces), 15, 882–883
 DNLC (directory name lookup cache), 726–733
 dnlc_lookup() function, 727
 Domains, processors, 33–34
 door_call() function, 320
 Doors
 servers, 25
 zones, 405
 DPI (Debugger/Prom Interface), 959–960
 Drain models, 866
 Drivers
 DDIs, 14
 devices, 882–891
 DLD, 888
 GLDv3, 884–886
 HAT, 14
 kernels, memory segment, 534
 seg_kpm, 710
 seg_map, 710–718
 segments, VM, 476–485
 segments, kernel memory, 535–537
 DRPs (Dynamic Resource Pools), 6, 36
 DTrace, 7
 CPUs, tracking, 170
 lockstat providers, 846–851
 priority fields, tracking, 231
 privileges, tracking, 360
 SDT probes, 936–937
 tick processing, 904
 VM, tracing, 466–467
 vm.d script, 475
 vnode interfaces, 703–706
 zones, 413–414
 Dumps, ELF, 54
 dup() function, 660
 Dynamic Intimate Shared Memory (DISM), 4, 9, 11, 294–295
 Dynamic linking, 52
 Dynamic reconfiguration, 452
 Dynamic resource allocation, 32
 Dynamic Resource Pools (DRPs), 6, 36
 Dynamic system domains, 34
 Dynamic tasks
 pools (STREAMS subsystem), 940–941
 queues, 928–932
 Dynamic topology support, 799
 Dynamic tracing facility. *See* DTrace
- E**
- eat_signal() function, 142
 Effective sets (privileges), 324, 337
 elfdump(1) command, 53
 elfexec code, 98
 ELF (Executable and Linking Format), 53–55
 Embedded on-disk (UFS) inode, 739–742
 Enabling
 4-Mbyte pages, 647
 cache tracing, 562–563
 large pages, 646
 logging, 775
 microstate accounting, 126
 Ending transactions, 786–787
 Endpoints, TCP, 874
 Enqueueing packets, 866
 Environment arrays, 97
 Errors. *See also* Troubleshooting
 DTrace, 7
 Escalation
 prevention, 340
 privileges, 384
 /etc/name_to_sysnum, updating, 973
 /etc/project file, 418, 435–436
 Events
 base scheduling, 159
 hardware, measurement, 641–642
 IPC. *See* IPC
 process model termination, 106–110
 semaphores, 308
 Evolution
 of file system frameworks, 669–672
 process models, 48–52
 UFS, 737–738
 exec code, 93
 execsw structure, 94

Executable and Linking Format. *See* ELF

Executables

- data (memory mapping), 457
- objects, 52–55
- sharing, 458
- text (memory mapping), 457

Execution

- control, source analysis, 960–961
- processes, 16
- profiles, RBACs, 11
- threads of, 15
- `exit()` function, 107
- Exiting kernels, process model termination, 106–110

Expiry processing, 916–917

Explicit lgroup APIs, 810

Exposure, implementation, 553

Extended attributes, UFS, 767–768

Extending privileges, 327–328

F

Facilities

- kernels, 17
- mdb. *See* mdb
- signaling, 25–26

Failures. *See also* Troubleshooting

- panic messages, associating, 570
- UFS, 790

Fair Share (FSS) scheduling class, 10, 23, 160

- framework, 197
- tick processing, 219–220
- update processing, 220–227
- zones, 408

Fair-share schedulers, 222

Fallback, STREAMS, 879–880

fastscan pages, 519

Faults

- COW, 497–498
- large pages, 495–496
- pages, 28
- pages, address spaces, 473–476
- procfs, 122
- SEGOP_FAULT(), 478
- Solaris Fault Manager, 5

`fbread` function, 723

`fbwrite` function, 723

FFS (Berkeley Fast File System), 737

Fields, structures

- `c_arg`, 907
- `c_func`, 907
- `cr_groups`, 348

`cr_zone`, 380

hat structure, 587

`ic_db`, 741

`ic_ib`, 741

`ic_nlink`, 740–741

`ic_oeflflag`, 741

`ic_shadow`, 741

`ic_smode`, 740

`lrusage`, 124

`m_dummylock`, 833

`mem_total`, 84

`mo_cancel`, 674

`m_owner`, 831

`m_spinlock`, 832

parsers, mount options, 673

`p_mlreal`, 125

`p_mstart`, 124

priority, tracking, 231

PRIORITY LEVEL, 204

`pr_vaddr`, 492

RES, 204

signal set, 135

smap structure, 712

timestamp, 572

`ts_globpri`, 204–205

`ts_lwait`, 206

`tsmaxwait`, 206

`ts_quantum`, 205

`ts_slpret`, 206

`ts_tqexp`, 205

TTE, 590

`u_sigmask` [], 140

`u_signal` [], 140

`u_signodefer`, 140

`u_sigonstack`, 140

`u_sigresethand`, 140

`u_sigrestart`, 140

utilization, 63

FIFO (first-in, first-out), 24

Files, 29–30

access, 659

controls, procfs, 121

core, file management, 11

core, zones, 389

descriptors, 658, 660–661

descriptors, allocation/deallocation, 662–665

descriptors, limits, 665

ELF, 53–55

/etc/project, 418, 435–436

file systems. *See* File systems

logging, 775

mapping, to disk blocks, 758–759

memory mapped, 463–464, 481–484

- Files (*continued*)
 - methods, implementation, 686
 - physical memory, 448
 - /proc, 111–112. *See also* procfs
 - process-level file abstractions, 658–668
 - resource limits, 85
 - shared mapped, 464
 - structures, 666–668
- File systems, 17, 29–30, 31
 - caches, 28, 668–669
 - caches, VM, 450–451
 - conversion (Solaris 10), 734–736
 - /dev, read-only mount, 402
 - DNLC, 726–733
 - frameworks, 657–658
 - frameworks, Solaris, 668–672
 - fsflush process, 734
 - independent data, 685
 - I/O, 707–718
 - memory allocation, 718–722
 - modules, 672–675
 - mount method, 681–683
 - pages, caches, 504, 721
 - path-name management, 722–725
 - privileges, 344
 - proc(4) command, 421
 - process-level file abstractions, 658–668
 - process models, 110–129
 - process models, implementation, 113–123
 - summaries, 782–783
 - UFS. *See* UFS
 - umount method, 683
 - vfs interfaces, 675–685
 - zones, 389–393
- File-system-specific data, 686
- File-to-key interfaces, 281
- File Transfer Protocol (FTP) zones, 404
- _fini() function, 678
- FireEngine approach, 864
- First-in, first-out (FIFO), 24
- Fixed Priority (FX) scheduling class, 10, 23, 160
 - thread priorities, 227–228
 - tick processing, 228–229
- Flags
 - c, 74, 203
 - compilation, 52
 - ELF dumps, 54
 - HAT, 617–618
 - IPC_RMID, 290
 - JUSTLOOKING, 143
 - L, 74
 - LD_DEBUG, 55
 - mmap shared mapped file, 464
- NOCD, 337
- PG_WAIT, 512
- RCTL_LOCAL_DENY, 430
- RCTL_LOCAL_DEV, 429
- SA_SIGINFO, 138
- SUGID, 337
- TP_MSACCT, 126
- VM, 492–494
- Flow
 - priority change, 212
 - shuttle switching, 320
 - TCP, 871
- Fork(), pages, size preferences, 646
- fork code, 83
- Fragmentation, 559–560
- Frames, stacks, 97
- Frameworks
 - device drivers, 882–891
 - file systems, 657–658
 - file systems, Solaris, 668–672
 - FSS, 10, 197
 - kstat, HAT layers, 621–625
 - kstat, zones, 412–313
 - NUMA, 799–802
 - scheduling classes, 196
 - stacks, 863–870
 - System V, 274–282
 - TCP, 870–875
- free() function, 462
- Freeing
 - allocators, 537
 - dispatcher locks, 186
 - large pages, 499
 - resources, 555–556
 - segments, 557–558
- Free lists, 503, 509
 - pages, 495
- FREE state, 76
- fsflush process, 734
- FS methods. *See* fsops
- fsops (FS methods), 678
- fss_decay_usage() function, 224
- FSS (Fair Share) scheduling class, 10, 23, 160
 - framework, 197
 - tick processing, 219–220
 - update processing, 220–227
 - zones, 408
- fssproc_t structures, 194
- fss_update() function, 220
- fstyp command, 756
- FTP (File Transfer Protocol) zones, 404
- Full checksum offload, 890
- Fully preemptable kernels, 13

Fully virtual devices, 400
 Functionality, slab allocators, 538
 Functions, kernel. *See also* Commands
 abort(3C), 106
 accept(), 873
 address spaces, 470–472
 as_add_callback(), 472
 as_alloc(), 469
 as_delete_callback(), 472
 as_do_callbacks(), 472
 as_fault(), 470
 as_setat(), 470
 bind(), 872
 bmap_read(), 758
 bmap_write(), 758
 bread_common(), 760
 bwwrite_common(), 760
 callout_schedule(), 907
 cfork(), 92
 change_pri, 827
 change_priority, 843
 check_page(), 523
 clocks, 901–902
 clock_tick(), 213, 903
 close(), 686, 874
 connect(), 872
 count(), 232
 cpu_choose(), 236
 cpu_resched(), 239
 crgetref(), 347
 cv_timedwait(), 258
 cv_timedwait_sig(), 258
 cv_wait(), 258
 cv_wait_sig(), 258
 cv_wait_sig_swap(), 258
 dispatchers, 234–245
 dispatchers, initialization, 190–191
 dispatchers, locks, 186–187
 dispdeq(), 234
 disp_getbest(), 189
 disp_lowpri_cpu(), 237
 DISP_MUST_SURRENDER(), 230
 DNLC, 728
 dnlc_lookup(), 727
 door_call(), 320
 dup(), 660
 eat_signal(), 142
 exit(), 107
 fbread, 723
 fbwrite, 723
 _fini(), 678
 free(), 462
 fss_decay_usage(), 224
 fss_update(), 220
 getpage(), 475
 getpagesize(), 644
 getpagesizes(), 644–645
 getproc(), 93
 groupmember(), 347
 HAT, 582–583
 hat_map(), 482
 _init, 674
 init_mstate(), 126
 kmem_alloc(), 543
 kmem_cache_alloc(), 540
 kmem_cache_create(), 540
 kmem_cache_destroy(), 540
 kmem_cache_free(), 540
 kmem_freepages(), 536
 kmem_getpages(), 536
 kmem_update, 910
 lgroup_version(), 810
 lgrp_fini(), 811
 lgrp_init(), 810
 libraries, 436–437
 listen(), 873
 lufs_read_strategy(), 789
 lufs_write_strategy(), 789
 main(), 106
 mapelfexec(), 97
 memcntl(), 646
 mi_timer_fire, 909
 mlock(), 294
 mmap(), 464
 mq_open(), 310
 mutex_enter(), 830
 mutex_exit(), 830
 mutex_init(), 830
 new_mstate(), 126, 128
 open(), 660, 686
 open(), code path, 661
 owner, 827
 page_create(), 466
 page_create_va(), 510, 515
 page_find(), 507
 page_free(), 514
 page_lookup_nowait(), 507
 pipe(), 660
 poke_cpu(), 239
 polltime, 909
 prochasprocperm(), 347
 putpage(), 491
 read(), 686
 read(), file system I/O, 707–710
 realitexpire, 909
 restore_mstate(), 126
 rmalloc(), 560
 rw_exit(), 838

Functions, kernel (*continued*)
 rw_exit_wakeup(), 839
 sbrk(), 462
 schedpaging, 909
 scheduling classes, 198–202
 secpolicy_vnode_setattr(), 349
 segmap, 713
 seg_pupdate, 910
 sema_p(), 846
 semget(), 296
 setbackdq(), 234, 236
 setfrondq(), 234, 240
 setkpdq(), 234
 setppriv(), 337
 setrun(), 235, 909
 sigalarm2proc, 910
 sigkill(), 147
 socket(), 872
 softcall(), 908
 squeue_create(), 868
 supgroupmember(), 347
 support, vfs interfaces, 679–681
 switch(), 242–246
 taskq_create(), 931
 taskq_dispatch(), 931–932
 taskq_lock(), 932
 taskq_member(), 932
 taskq_resume(), 932
 taskq_suspend(), 932
 taskq_suspended(), 932
 taskq_wait(), 932
 term_mstate(), 126
 thread_create(), 234
 thread_high(), 188
 thread_lock(), 188
 time-based classes, 211–214
 timeout_common(), 907
 tod_set(), 911
 trans_roll(), 787
 ts_parmset(), 215
 ts_update(), 218, 910
 ts_wakeup(), 261–262
 turnstile_lookup(), 842
 turnstile_wakeup(), 843
 unsleep, 827
 vfork(), 469
 vfs_initopttbl(), 673
 vmem_add(), 553
 vmem_create(), 53
 vmem_free(), 558
 vn_alloc(), 696
 vnode interface, 696
 write(), 686
 write(), file system I/O, 707–710

FX (Fixed Priority) scheduling class, 10, 23, 160
 thread priorities, 227–228
 tick processing, 228–229

G

Generation, signals, 133–135
 Generic LAN driver (GLDv2) module, 882–883
 GET, 274–275
 getcpuid(3C) routine, 808
 get_high_resolution_time, 125
 getpage() function, 475
 getpagesize() function, 644
 getpagesizes() function, 644–645
 getproc() function, 93
 GET_TTE macro, 611
 GID (group ID), 59
 GLDv3 architecture, 883–889
 GLDv2 (generic LAN driver) module, 882–883
 Global hash lists, 507
 Global memory allocation, 27–28
 Global page replacement, 516
 Global priorities, threads, 208–209
 Global process priorities, 22–23
 Global resource controls, 427–428
 Global slab layer, 547–548
 Global zones, 6, 368. *See also* Zones
 visibility, 387
 Granularity, zones, 368
 Graphs, cyclic, 915
 Group ID. *See* GID
 groupmember() function, 347
 Groups
 CPU, 165
 cylinder, 748–749
 lgroups. *See* lgroups
 process models, 150–156
 Growing heaps, 461–462
 GRUB, enabling tracing, 562–563

H

Handlers
 PIL, 184
 traps, 108
 Handling traps, 961–962
 Hard links, UFS, 744–745
 Hard/soft rlimit interface, 431
 Hard swapping, 525
 Hardware
 counters, reading, 641–642
 hierarchies, 822

- locks, 819–824
- statistics utilities, 11
- synchronization, 819–824
- time-of-day clocks, 911
- Hardware Address Translation.** *See HAT*
- Hardware mapping entry (HME),** 500
- Hashed page table (HPT),** 589
- Hashed vmem arenas,** 938–939
- Hash lists,** pages, 507–508
- Hash tables**
 - data structures, 589
 - hme_blk* structure, 599–601
- Haslam, Jon,** 169
- HAT (hardware address translation),** 14, 292
 - implementation, 631–636
 - ISM, 613–616
 - layers, SPARC, 620–625
 - layers, synchronization, 616–620
 - layers, UltraSPARC, 583–625
 - layers, x64, 625–636
 - overview of, 581–583
 - pages, 506
 - support, 500–501
 - VM design, 457
- hat_lock array,** 617
- hat_map() function,** 482
- HAT mapping entry (HME),** 500, 591
 - blocks, allocation, 598–599
 - shadow blocks, 597–598
- Headers, ELF,** 54
- Heap**
 - growing, 461–462
 - kernels, 534–535
 - management, 913–916
 - physical memory, 449
 - size, 463
 - space (memory mapping), 457
- Hierarchies**
 - hardware, 822
 - lgroups, 800–812
 - memory, 796–799
 - time-of-day clocks, 911
 - UFS directories, 743
- High-priority interrupts,** 266–267
- High-resolution timers,** 910
- Hints, MADV_ACCESS_LWP,** 809
- hme_blk structure,** 594–597
 - hash tables, 599–601
- HME (HAT mapping entry),** 500, 591
 - blocks, allocation, 598–599
 - shadow blocks, 597–598
- Horizontal perimeters,** 864
- Housekeeping thread, DNLC,** 733
- HPT (hashed page table),** 589
- I**
- IA (Interactive) scheduling class,** 9, 23, 160
- ic_db field,** 741
- ic_ib field,** 741
- ic_nlink field,** 740–741
- ic_oeflflag field,** 741
- ic_shadow field,** 741
- ic_smode field,** 740
- ID**
 - callouts, 906
 - IPC, structure names, 281
 - root vnode, 683
- Identifiers,** 280
 - semaphores, 296
 - shared memory, calling, 288
- Idle queue,** 752–754
- IDL state,** 76, 158
- ifconfig(1M) command,** 395–396
- Implementation.** *See also Configuration*
 - adaptive mutex lock, 21
 - asynchronous signals, 26
 - cyclic subsystem, 913–922
 - DISM, 295
 - exposure, 553
 - files, methods, 686
 - file systems, 672
 - HAT, 631–636
 - kadb, 947
 - kernels, shared memory, 288–291
 - kmdb, 943–962
 - lgroups, 804–807
 - messages, queues, 301–303
 - modular, 17
 - mutex locks, 830–835
 - pages, scanners, 522–524
 - procfs, 113–123
 - signals, 135–148
 - slab allocators, 544–545
 - Solaris Doors, 314–320
 - swapfs, 489–491
 - task queues, 937–941
 - TCP, 870–875
 - turnstile, 841–843
 - VM, 451–453
 - vmem allocators, 556–560
- Importing arenas,** 556
- In-core log data structures,** 779–782
- In-core UFS inodes,** 751–752
- Indexes**
 - nodes. *See inodes*
 - procfs, 116
 - sleepq_head array, 845
 - slots, 275

Infinite time quantum, testing, 230
Inheritance
 priority, 840–843
 sets (privileges), 324
`_init` function, 674
Initialization
 callouts, 908
 dispatchers, 190–191
 dispatchers, locks, 186
 lgroup interfaces, 810–811
 modules, 674–675
Initial thread placement (NUMA), 802
`init_mstate()` function, 126
In-kernel project data structures, 421–423
Inodes
 caches, 752–754
 shadow, 745
 UFS, 751–764
 UFS, in-core, 751–752
 UFS, on-disk, 739–742
Insertion, queues, 235, 240–241
Instruction TLB (iTLB), 645
Integers, file descriptor allocation/deallocation, 662–665
Integration, networks, 15
Integrity, buftag data, 570–571
Intel x86 processor support, 8
Interactive (IA) scheduling class, 9, 23, 160
Interfaces
 ABI, 53
 APIs, 45. *See also* APIs
`cred_getzoneid(3c)`, 405
 cyclic subsystem, 912–925
 DDIs, 14
 DDIs, task queues, 934–935
 DLPIs, 15, 882–883
 DPI, 959–960
 dynamic task queues, 931–932
 file-to-key, 281
 hard/soft rlimit, 431
 IPClassifier, 869
 ipc module, 278
 KDI, 946–947
 kernels, 349–351
 kernels, resource controls, 437–444
 large page sizes, requests, 649–652
 least privileges, 344–363
 lgroups, initialization, 810–811
 libdl, wrappers, 956
 libraries, 353–355
 LWP process model structures, 69
 message queues, 309
 mount options for, 673–674
 MPO, 807–811
 MPSS, 9
 NICs, 10
 objects, 686–688
 pages, 510–512
 POSIX IPC, 304
 private kernel, system calls, 436
`proc(4)`, modifying privileges, 339
`proc(4)`, optimizing privileges, 360–361
 programming, task queues, 932–933
 projects, 419–420
`pthread_kill(3C)`, 145
 resource controls, 432–437
`seg_map` driver, 710–718
 segment, 478
 semaphores, 306
 set-uid, 356–357
 shared memory, 286
`shmdt(2)`, 290
`shm_open`, 304
`shm_unlink`, 304
 slab allocators, 542–543
 Solaris Doors, 313
 Solaris file system, 672
 system calls, 16, 98–106, 351–352
 tasks, 419–420
 TCP and IP, between, 872–874
 time-of-day clocks, 911
`timeout(9F)`, 904
`untimeout(9F)`, 908
 user credential library, 355–356
 versions, verification, 810
 vfs, 30, 668, 675–685
 vmem allocators, 553–556
 vnode, 30, 668, 685–706
 vnode, block I/O on pages, 700
 vnode, caches, 698–700
 vnode, DTrace probes, 703–706
 vnode, life cycles, 696–697
 vnode, mdb(1) kernel debugging facility, 701–703
 vnode, methods, 690–695
 vnode, reference counts, 698
 vnode, registration methods, 688–690
 vnode, support functions for, 696
 vnode, types, 688
 zones, 395–396
Internet Protocol. *See* IP
Internet Protocol Quality of Service. *See* IPQoS
Internet Protocol version 6 (IPv6), 10, 396–397
Interposing shared libraries, 647–648
Interprocess communication. *See* IPC
Interrupts
 block cookies, 832
 blocking, 185

- clocks, 901
- cross-calls, 268–270
- dispatchers, 262–270
- high-priority, 266–267
- interprocessor, 268
- load spreading, 894–895
- management, 33–34, 267, 958
- monitoring, 267–268
- PIL, 829
- priorities, 23, 264
- stacks, 891–895
- threads, 264–266
- threads, priorities, 266
- Inter-subsystem interfaces, cyclic kernel, 924
- Interval timers, applications, 12
- Intimate Shared Memory (ISM), 291–294, 613–616, 645
- Inverted page table (IPT), 588
- I/O
 - block, vnode pages, 700
 - console, 957–958
 - file systems, 707–718
 - management, 17
 - pages, locking, 498
 - parallel system architectures, 817
 - procfs, 120
 - terminal, 946, 956
 - VM file system caches, 450–451
- IP (Internet Protocol)
 - classifiers, 868–869
 - as multiplexers, 862
 - stacks, 880–882
 - structures, 861
- IPC (interprocess communication), 23
 - locks, 277–280
 - modules, creating, 280–282
 - objects, 274–275
 - overview of, 273
 - POSIX, 25, 303–312
 - resource limits, 86
 - Solaris Doors, 312–320
 - System V, 24–25
 - System V, framework, 274–282
 - System V, message queues, 299–303
 - System V, resource controls, 282–284
 - System V, semaphores, 295–298
 - System V, shared memory, 286–295
 - traditional Unix, 24
 - tunable, configuring, 285
 - zones, 405–407
- ipc modules, interfaces, 278
- ipc_perm structure, 281
- ipcrm(1) command, 406
- IPC_RMID flag, 290
- ipcs(1) command, 291, 406
- ipc_service structure, 276
- IP_HDRINCL option, 397
- IPPROTO_IP-level option, 397
- IPQoS (Internet Protocol Quality of Service), 38–39
- IPSec (IP Security), 10, 397
- IPT (inverted page table), 588
- IPv6 (Internet Protocol version 6), 10, 396–397
- IRIX privileges, 332
- ISM (Intimate Shared Memory), 291–294, 613–616, 645
- Isolation, zones, 368
- ISSIG_PENDING macro, 142
- ITLB (instruction TLB), 645
- J**
- Juggling cyclic subsystems, 922
- JUSTLOOKING flag, 143
- K**
- Kbytes, resource limits, 85
- KDI (Kernel/Debugger Interface), 946–947
- kernalmmap segment, 536
- Kernel/Debugger Interface (KDI), 946–947
- Keys
 - data structures, synchronization, 861–862
 - values, 280
- kill(1) command, 145
- kipc_perm_t member, 275
- kmastat columns, 550
- kmdbs
 - design, 946–949
 - implementation, 943–962
 - MDB components, implementation, 952–958.
 - See also* mdb
 - structures, 949–959
- kmem_alloc() function, 543
- kmem_cache_alloc() function, 540
- kmem_cache_create() function, 540
- kmem_cache_destroy() function, 540
- kmem_cache_free() function, 540
- kmem_flags variable, configuration, 564
- kmem_freepages() function, 536
- kmem_getpages() function, 536
- kmem_update function, 910
- kmutex_t lock, 768
- krwlock_t lock, 768

kstat(1) command, 80
 kstat
 counters, 935–936
 frameworks, HAT layers, 621–625
 frameworks, zones, 412–313
 kthread pointers, 831
 kthread_t, 174–175

L

Large kernel page support, 606–607
 Large pages. *See also* Pages
 applying, 639
 compilers, 648–649
 CPU support, 652–653
 enabling, 646
 requests, interfaces, 649–652
 support, changes to, 494–501
 Layers
 anonymous memory, 487–488
 CPU, 545–546
 depot, 546–547
 global slab, 547–548
 HAT, 14, 292
 HAT, SPARC, 620–625
 HAT, synchronization, 616–620
 HAT, UltraSPARC, 583–625
 HAT, VM design, 457
 HAT, x64, 625–636
 swapfs, 489–491
 target, 952
 target, mdb, 944–945
 vfs, 675
 VM, 456
 lbolt counter, 902
 L2 cache, 514
 LDAP (Lightweight Directory Access Protocol), 10
 LD_DEBUG flags, 55
 ldstub instructions, 820, 823
 Leaks, memory, 573
 Least privileges, 324–325, 328–329
 interfaces, 344–363
 Left ancestors, LPARENT, 663
 Levels
 of memory, 448
 of memory allocation, 535
 of PIL, 829
 lgroup_home(3C) routine, 808
 lgroups (locality groups), 165
 hierarchies, 800, 811–812
 implementation, 804–807

interfaces, initialization, 810–811
 observability, 168
 partitions, 167
 lgroup_version() function, 810
 lgrp_expand_proc_diff parameter, 807
 lgrp_fini() function, 811
 lgrp_init() function, 810
 lgrp_loadavg_tolerance parameter, 807
 lgrp_mem_default_policy parameter, 805–806
 lgrp_mem_pset_aware parameter, 806
 lgrp_privm_random_thresh parameter, 807
 lgrp_shm_random_thresh parameter, 806
 libdl interfaces, wrappers, 956
 libmpss.so library, 646–648
 libproc, 120
 Libraries
 functions, 436–437
 interfaces, 353–355
 libmpss.so, 646–648
 shared, interposing, 647–648
 shared, optimizations, 520–521
 sharing, 458
 threads, 10, 11
 unified process models, 50
 libthread.so (threads library), 10, 11
 Life cycles
 physical memory, 503–505, 719
 vnode interfaces, 696–697
 Lightweight Directory Access Protocol. *See* LDAP
 Lightweight process. *See* LWP
 Limitations
 file descriptors, 665
 pages, 521–522
 processes, 80–83
 resources, 85
 threads, 83–84
 zone privileges, 383–384
 Limit set (privileges), 338
 Linear-time performance, 552
 Linked lists
 multiple, 167
 pages, searching, 508
 Links
 dispatcher structures, 175–177
 dynamic linking, 52
 ELF, 53–55
 GLDv3 aggregation architecture, 888–889
 hard, UFS, 744–745
 processes, 47–48
 proc/hat structures, 587
 structures, 115
 Linux, privileges, 332

`listen()` function, 873

Lists

- ACL, 764
- ACL, modifying, 766
- deathrow, 109–110
- free lists, pages, 495
- hash lists, pages, 507–508
- linked, multiple, 167
- zones, 374–375

Loadable modules, 18, 668

 kernel allocation, 531

Load balancing, 799

Load spreading, interrupts, 894–895

Locality awareness, 799

locality groups. *See lgroups*

Local page replacement, 516

Local resource controls, 428

Locating pages, 507

Locks

 acquisition, 618, 837

 adaptive, 829

 adaptive, probes, 847–848

 arenas, 559

 cyclic subsystem, 918–919

 dispatchers, 183–190, 824

 dispatchers, functions, 186–187

 dispatchers, threads, 187–189

 hardware, 819–824

 HAT, 616

 IPC, 277–280

 ISM, 293

 mutex, 824, 827–835

 mutex, adaptive implementation, 21

 pages, 498

 projects, 423

 releasing, 834

 RW, 821, 824, 835–840

 RW, probes, 849–851

 spin, 828

 spin, probes, 848–849

 statistics, 834

 threads, 849

 UFS, 768–774

 UFS, protocols, 773–774

Lockstat providers, 846–851

Logging

 allocators, 576–577

 metadata, 783

 rolling, 787–788

 UFS, 10, 775–790

logmap structure, 781

Log-structured file systems, 775

Lookup, DNLC, 726–733

Loopback

 TCP, 874–875

 transport providers, 406

Loops

 callouts, 908

 mutex locks, 833

Losing privilege awareness, 335

lotsfree pages, 519

Lowest-set bit (LSB), 662

LPARENT node, 663

lrusage structure, 123–125

LSB (lowest-set bit), 662

ls(1) command, 54

lufs_read_strategy() function, 789

lufs_write_strategy() function, 789

LWP (lightweight process), 19, 20

 kernel thread exit, 108–109

 pools, 49

 process model structures, 69–73

 thread objects, 44

M

Macros

 CL_TICK(t), 213

 CONN_DEF_REF, 870

 CONN_INC_REF, 870

 GET_TTE, 611

 ISSIG_PENDING, 142

 PAGE_HASH_FUNC, 507

 PAGE_HASH_SEARCH, 508

 THREAD_SET_STATE, 187

 TRANS_BEGIN_ASYNC, 785

 TRANS_BEGIN_CSYNC, 786

 TRANS_BEGIN_SYNC, 785

 TRANS_TRY_BEGIN_ASYNC, 786

 TRANS_TRY_BEGIN_CSYNC, 786

 TS_NEWUMDPRI, 214

MADV_ACCESS_LWP hint, 809

madvice(3C) routine, 809

madv.so.1 routine, 809

Magazine sizes, 546

main() function, 106

Major page faults, 473

Management

 address space, VM, 467–476

 breakpoints, 961

 buses, 17

 core file, 11

 debugging, modules, 953–954

 devices, 16, 401

 heaps, 913–916

Management (*continued*)
 interrupts, 267, 958
 interrupts, processors, 33–34
 I/O, 16
 ipc_service structures, 276
 memory, 16, 26–29, 449–450
 memory, kernels, 28–29
 memory, pages, 506–516
 MMU. *See* MMU
 modules, debugging, 945
 path-name, 669
 paths, file system names, 722–725
 physical memory, 450
 process rights, 8
 queues, 159
 queues, dispatchers, 234–242
 RCM, 453
 resources, 3, 13, 16, 30–39, 160
 resources, core process components, 48
 resources, observability, 38–39
 resources, processors, 33–34
 resources, Solaris, 35–38
 resources, zones, 6, 370, 407–414
 rights, 8, 323. *See also* Privileges
 RM, 9
 SMF, 5
 Solaris Fault Manager, 5
 SVM, 775
 synchronization, 861–862
 watchpoints, 961
 MAPBLOCKSIZE, 781
 mapelfexec() function, 97
 mapentry structure, 781
 Mapping
 address space, 650
 debugging, 781
 files, to disk blocks, 758–759
 heaps, 461–462
 HME, 500, 591
 hme_blk structures, 594–597
 kernel virtual addresses, 965–969
 libraries, 458
 memory, files, 481–484
 memory, I/O, 708–709
 memory, pmap(1) command, 642–643
 pages, 506–516
 pages, physical, 643–644
 pages, seg_kpm driver, 710
 physical pages, caches, 513
 processes, pmap command, 465–466
 stacks, 462–463
 text, 530
 VM, 457
 Masks, signals, 132, 138

Massively parallel processor (MPP), 816
 matamap structure, 781
 max_nprocs value, 81
 max_percent_cpu parameter, 521
 maxusers variable, 81
 MC_HAT_ADVISE control operation, 649
 mdb(1) kernel debugging facility, 11, 65, 82
 caches, viewing, 563–565
 components, 943–962
 components, implementation in kmd, 952–958
 dispatchers, 271
 vfs interface information, 684–685
 vnode information, 701–703
 m_dummylock field, 833
 Measurement
 applications, performance, 640–642
 hardware events, 641–642
 microstate accounting, 127
 performance, NUMA, 803
 memcntl() function, 646
 meminfo(2) command, 643–644
 meminfo(2) routine, 808
 Memory, 447
 adding, 452
 allocation, 166
 allocation, file systems, 718–722
 allocation, global, 27–28
 allocation, kernels, 534–551
 allocation, NUMA, 803
 allocation, procfs, 114
 allocation, TSB, 605
 barriers, 824
 caches, 28
 demand-page virtual memory systems, 14
 DISM, 4, 9, 11, 294–295
 hierarchies, 796–799
 ISM, 291–294, 613–616, 645
 kernels, 527
 kernels, allocator logging facility, 576–577
 kernels, analyzing, 573–574
 kernels, debugging, 563
 kernels, detecting corruption, 565–566
 kernels, logging, 570–572
 kernels, segment drivers, 535–537
 kernels, slab allocators, 537–551
 kernels, tracing allocators, 562–577
 kernels, troubleshooting buffers, 575
 kernels, vmem allocators, 552–562
 kernels, VM layouts, 527–534
 leaks, 573
 levels of, 448
 management, 16, 26–29, 449–450
 management, kernels, 28–29
 management, pages, 506–516

- mapping, files, 463–464, 481–484
- mapping, I/O, 708–709
- MMU. *See* MMU
- models, 822
- MPSS, 9
- NUMA, 165, 166, 238. *See also* NUMA
- pageable, swapping, 532
- pages, physical memory, 448–449
- physical, 6. *See also* Physical memory
- physical, life cycles, 719
- pmap(1) command, 642–643
- protection, 448
- RAM, 81
- shared, ISM, 291–294
- shared, POSIX, 304–305
- shared, System V, 24, 286–295
- sharing, 448
- UMA, 166
- VM. *See* VM
- Memory in use column, 79
- Memory Management Unit, 292. *See* MMU
- Memory Placement Optimization. *See* MPO
- memseg lists, 509–510
- mem_total field, 84
- Messages
 - controls, procfs, 121–122
 - panic, associating, 570
 - POSIX, queues, 309–312
 - queues, System V, 299–303
 - System V queues, 24
- Metadata
 - blocks, buffering, 760
 - logging, 775, 783
- Microbenchmark performance, 560
- Microprocessors, CPU specific large page support, 652–653
- Microstate accounting, 48
 - process models, 125–129
- Migration, physical memory, 448–449
- Minor page faults, 473, 474
- min_percent_cpu parameter, 521
- Misses, TLB, 640
- mi_timer_fire function, 909
- mlock() function, 294
- ml_odunit_t structure, 778
- mmap() function, shared mapped file flags, 464
- MMU (Memory Management Unit), 27, 292
 - configuration, 625–626
 - file system I/O, 708
 - SRMMU, 588
 - virtual-to-physical translation, 449–450
- mntopts_t structure, 673
- mo_cancel field, 674
- MO_DEFAULT option, 674
- Models
 - drain, 866
 - interrupts, 891–895
 - latency, 800–801
 - memory, 822
 - privileges, 323–324, 325–333
 - privileges, superusers, 326
 - process. *See* Process models
 - processing, 866
 - protection, VM, 473
 - queues, 866
 - sequential consistency, 822
 - state, zones, 371–372
 - STREAMS, 856–859
 - tasks, pools, 930–931
 - threads, 20–21
- Modern memory allocators, 453
- Modes
 - plumbing, 877
 - polling, 892–893
- Modifying
 - ACLs, 766
 - pas, 339
 - privileges, state, 335–339
 - processors, state, 960
- modinfo command, 531
- Modular debuggers, 11
- Modular device I/O systems, 14
- Modular implementation, 17
- Modules
 - debugging, management, 953–954
 - file systems, 672–675
 - GLDv2, 882–883
 - initialization, 674–675
 - interfaces, 278
 - IPC, creating, 280–282
 - loadable, 18
 - loadable, file systems, 668
 - management, debugging, 945
 - shared memory kernel, 286
 - STREAMS, 859–862
 - UDP, 876–878
- MO_HASVALUE option, 674
- MO_IGNORE option, 674
- Monitoring
 - interrupts, 267–268
 - prstat(1M) command, 39
 - queues, 241–242
 - RM, 9
 - threads, priorities, 231–233
 - zones, 412–413
- MO_NODISPLAY option, 674
- mount method, 681–683
- Mount options, interfaces for, 673–674

m_owner field, 831
 MPO (Memory Placement Optimization), 9, 161, 452, 795
 APIs, 807–811
 parameters, 805–807
 statistics, 813–814
 MPP (massively parallel processor), 816
 MPSS (Multiple Page Size Support), 9, 452, 646
 mq_open() function, 310
 *m_spinlock fields, 832
 Multicasting, 881–882
 Multipathing, IP network, 881
 Multiplatform support, 13
 Multiple CPUs, latency models, 800
 Multiple file system support, 14
 Multiple linked lists, 167
 Multiple pages
 enabling, 646
 size, configuration, 645–653
 Multiple Page Size Support (MPSS), 9, 452, 646
 Multiple scheduler support, 14
 Multiple TSB probes, 610–611
 Multithreading, 795. *See also* CMT
 mutex_enter() function, 830
 mutex_exit() function, 830
 mutex_init() function, 830
 Mutex locks, 824, 827–835
 adaptive implementation, 21
 dispatchers, 183–190

N

Names
 conventions, core file management, 11
 directories, caches, 669
 DNLC, 726–733
 IP ID structures, 281
 path-name management, 669
 paths, file systems, 722–725
 privileges, 346
 process objects, 45
 searching, 723
 semaphores, 307
 zones, 372–373
 Name-service attributes, 433
 Namespaces
 attributes, 433
 devices, 400
 IPC, 276
 locks, 277
 Navigation, system calls, 101–106
 nc_hash entries, 727

Negative caches, DNLC, 729
 Network file system. *See* NFS
 Network interface cards (NICs), 10
 plumbing, 880–881
 speeds, 891–895
 Networks, 17
 integration, 15
 stacks, 855. *See also* Stacks
 throughput, 8
 zones, 393–398
 new_mstate() function, 126, 128
 New pages, allocation, 515
 NFS (network file system), 29, 30
 privileges, 343–344
 NICs (network interface cards), 10
 plumbing, 880–881
 speeds, 891–895
 NOCD flag, 337
 Nodes
 CSIZE, 663
 index. *See* inodes
 LPARENT, 663
 RPARENT, 663
 No fan-out defaults, 868
 Nofiles (descriptors), resource limits, 85
 Non-preemption points, 246
 Nonuniform memory access. *See* NUMA
 Normal callouts, 905
 Normalize usage (delay), 222
 Not-recently-used time, 520
 NUMA (nonuniform memory access), 165, 166, 238
 frameworks, 799–802
 initial thread placement, 802
 lgroups, hierarchies, 811–812
 lgroups, implementation, 804–807
 memory, allocation, 803
 memory hierarchies, 796–799
 MPO, APIs, 807–811
 MPO, statistics, 813–814
 overview of, 795
 parallel system architectures, 817
 scheduling, 802–803
 Numeric Ids, zones, 372–373
 Numeric values for resource controls, 426

O

Objects
 caches, 540–543
 chip_t, 168
 depot layer, 546–547
 executable, 52–55

- interfaces, 686–688
- IPC, 274–275
- memory, 455. *See also* Memory
- sclass_t, 195
- slab allocators, 538
- synchronization, 21, 824–827
- threads, 44–47
- vfs, 675
- vnode interfaces, life cycles, 697
- Observability**, 3, 12
 - lgroup, 168
 - networks, 863
 - processors, 168–171
 - resource management, 38–39
 - task queues, 935–937
 - tools, 640
 - zones, 407–414
- Observing signal activity**, 148–149
- On-disk formats, UFS, 739–750
- On-disk log data structures, 776–779
- ONPROC state, 76, 158
- Open Boot PROM, mapping text, 530
- open() function, 660, 686
 - code path, 661
- Opening files, 660
- Operations vectors, synchronization objects, 826–827
- Ops vector, 439–440
- Optimization**
 - large page sizes, selection, 639
 - lbthread.so (threads library), 10
 - MPO, 9, 452. *See also* MPO
 - privileges, 360–361
 - shared library, 520–521
 - UFS, 10
- Origination, signals, 133
- Out-of-the-box performance, 863
- owner function, 827
- Owners, locks, 827
- Ownership, resource controls, 432
- o zone option, 387

- P**
- Packets
 - enqueueing, 866
 - processing cost, 863
 - UDP, 876
- Pageable memory, swapping, 532
- page_create() function, 466
- page_create_va() function, 510, 515
- page_find() function, 507
- page_free() function, 514
- page_hash array, 507
- PAGE_HASH_FUNC macro, 507
- PAGE_HASH_SEARCH macro, 508
- Page-ins, 449, 496–497
- page_lookup_nowait() function, 507
- Page-outs, 448
 - algorithms, 518–520
 - physical swaps, 491
- Pages, memory
 - attaching, 463
 - cachelists, 509
 - caches, 28, 504
 - caches, file systems, 721
 - coloring, 512–516
 - compilers, 648–649
 - copy-on-write process, 484
 - CPU support, 652–653
 - cyclic page caches, 718
 - demotion, 651
 - distribution, 515
 - enabling, 646
 - fastscan, 519
 - faults, 28
 - faults, in address spaces, 473–476
 - free lists, 495, 509
 - global memory allocation, 27–28
 - hash lists, 507–508
 - HPT, 589
 - interfaces, 510–512
 - IPT, 588
 - ISM locks, 293
 - large kernel support, 606–607
 - limitations, 521–522
 - locating, 507
 - locking, 498
 - lotsfree, 519
 - mapping, seg_kpm driver, 710
 - memory, management, 27
 - memory, schedulers, 524–525
 - memseg lists, 509–510
 - MMU, 449–450, 452
 - MPSS, 452
 - new, allocation, 515
 - physical memory, 448–449, 506–516
 - physical memory, management, 450
 - placement, 512
 - protection, 484–485
 - protection, faults, 474
 - requests, interfaces, 649–652
 - scanners, 516–518
 - scanners, implementation, 522–524
 - selection, 639

Pages, memory (*continued*)
 semop(2), 298
 shared library optimizations, 520–521
 sizes, allocation, 642–644
 sizes, configuration, 645–653
 sizes, support, 644–645
 slowscan, 518
 structures, 506, 508–509
 support, changes to large, 494–501
 swapping, 491
 throttles, 512
 TSB, relocation, 607
 vnode interfaces, caches, 698–700

Page-size, 496

pagesize command, 644

Paging, 450

Panic messages, associating, 570

Parallel systems architecture, 816–819

Parameters
 async_request_size, 524
 GET_TTE macro, 611
 kernels, configuration, 971–974
 max_percent_cpu, 521
 min_percent_cpu, 521
 MPO, 805–807
 page-outs, 518–520
 pages, limitations, 521–522
 scan rate, 518–520
 slab cache, 548
 superblocks, 747–748
 throttlefree, 512

Parsers, mount options, 673

Partial checksum offload, 890

Partial Store Order (PSO), 823

Partitions
 CPU, 165
 lgroup, 167
 RM, 9
 zones, 5–6, 394–395

p_as, 56

pas, modifying, 339

Path-name management, 669

Paths
 code, open() function, 661
 names, file systems, 722–725

p_cred, 59

p_crlock, 59

PCWATCH command, 492

Pending signals, 135

Performance, 3. *See also* Optimization
 applications, measurement, 640–642
 DISM, 295
 kernels, text, 530
 libthread.so (threads library), 10

linear-time, 552

microbenchmark, 560

MPSS, 9

NUMA, 803

out-of-the-box, 863

pages, placement, 512

scalability, 12

slab allocators, 538

system-level, 560–561

TCP/IP, 8

threads, 21

UFS, 10

vmem allocators, 560–561

zones, 409–410

Perimeters, vertical, 864–868

Permissions
 ALTER, 298
 POSIX message queues, 311
 semaphores, 298

Permitted sets (privileges), 324

Per-process file tables, 659

Per-process state, 334

p_exec, 56

pgrep command, 11

PG_WAIT flag, 512

Physical memory. *See also* Memory
 allocation, 166, 503–505
 control, 6
 life cycles, 719
 management, 27, 450
 MPO, 9
 pages, 448–449, 506–516
 pages, schedulers, 524–525
 virtual-to-physical translation, 449–450

Physical-meta-data blocks, 754

Physical pages, mapping, 643–644

Physical swaps, page-outs, 491

PID (process ID), 46
 structure, 63

PIL (Priority Interrupt Level), 184, 829

PINNED state, 76, 158

pipe() function, 660

Pipes, 405

pkill(1) command, 11, 387

Placement
 MPO. *See* MPO
 pages, 512
 threads, 802
 turnstile, 834

p_lock, 277

p_lockp, 58–59

Plumbing
 modes, 877

NICs, 880–881

- pmap command, 465–466, 642–643
 p_mlreal field, 125
 p_mstart field, 124
 poke_cpu() function, 239
Policies
 allocation, 555–556, 558
 NUMA, 803
 resource controls, 428–429
 UFS layout, 754–758
Polling mode, 892–893
polltime function, 909
poold daemon, 6, 36
Pools
 dynamic task, STREAMS subsystem, 940–941
 LWP, 49
 physical memory, 503
 resources, 36
 resources, CPUs, 165
 resources, zones, 409, 412
 RM, 9
 tasks, 930–931
 tasks, troubleshooting, 940
poolstat(1) command, 39
Porting file systems (to Solaris 10), 734–736
POSIX
 IPC, 25, 303–312
 LWP process model structures, 69–70
 messages, queues, 309–312
 semaphores, 305–309
 shared memory, 304–305
 zones, 407
 p_pglink, 62
 p_pgpidp, 63
 ppgsz(1M) command, 646
 p_pidflag, 61
 p_pidg, 63
 p_ppglink, 62
 p_ppid, 61
 ppriv(1) command, 383
 #pragma binding directives, 91
 prcommon structure, 115
 prctl command, 285
 Predictive self-healing, 5
Preemption
 control, 218
 dispatchers, 246–253, 268
 kernel, 246
 threads, 235
Preferences. *See also Configuration; Options*
 page sizes, 646
priocnt(1) command, 209
Priorities
 change flow, 212
 dispatching, 929
 inheritance, 840–843
 interrupts, 264
 threads, 158
 threads, change, 235
 threads, configuration, 211–233
 threads, dispatcher, 207–233
 threads, FX, 227–228
 threads, global, 208–209
 threads, monitoring, 231–233
 threads, RT, 229
 threads, user, 209–211
 TS, 214–217
Priority Interrupt Level (PIL), 184, 829
PRIORITY LEVEL field, 204
Priority scheduling, 10
Privacy, 10
Private daemons, 356–357
Private kernel interfaces, system calls, 436
PRIV_FILE_DAC_WRITE privilege, 384
Privileges. *See also Security*
 auditing, 362
 awareness, 330–331
 awareness, state transitions, 334–335
 basic, 342
 constants, 346
 core dumps, 360–361
 debugging, 361–362
 devices, 402
 DTrace, tracking, 360
 escalation prevention, 340
 extending, 327–328
 kernels, 346–349
 least, 324–325, 328–329
 least, interfaces, 344–363
 library interfaces, 353–355
 models, 323–324, 325–333
 names, 346
 NFS, 343–344
 RBAC, 357–359
 resources, 86–87
 resources, controls, 426, 432
 runtime, 342–343
 semantics, 334–344
 state, modifying, 335–339
 superuser, 7–8
 superuser, models, 326
 superuser, RBACs, 11
 systems calls, 98–106
 tasks, 46
 third-party file systems, 344
 uid 0, troubleshooting, 340–341
 zones, 380–384

PRIV_NSET constant, 348
PRIV_PROC_MOUNT privilege, 383
PRIV_PROC_OWNER privilege, 387
PRIV_SETSIZE constant, 348
Probes
 adaptive locks, 847–848
 change-pri, 232
 DTrace, 7
 DTrace, SDT, 936–937
 DTrace, vnode interfaces, 703–706
 lockstat providers, 846–851
 multiple TSB, 610–611
 RW locks, 849–851
 sched tick, 904
 spin locks, 848–849
 VM, tracing, 466–467
/proc. *See* **procfs**
proc(1) command, 120
process attribute, 419
Processes, 15, 18–19
 address spaces, 13, 305
 address spaces, mappings, 650
 address spaces, SPARC systems, 459
 bash, 592
 copy-on-write, 484
 core dumps, 493
 core process components, 47–48
 execution, 16
 files, descriptors, 660–661
 files, mapping, 563
 fflush, file systems, 734
 global priorities, 22–23
 IPC, 23. *See also* IPC
 limits, 80–83
 links, 47–48
 LWP, 19, 20. *See also* LWP
 mapping, pmap command, 465–466
 models. *See* Process models
 objects, 44
 per-process file tables, 659
 profiles, 64
 rights management, 8, 323. *See also* Privileges
 /sbin/sh, 458
 scheduling, 16
 signaling, 25–26
 sleep, 257–261
 stacks, memory mapping, 457
 state, 159
 synchronization, 825–826
Process ID (PID), 46
 structure, 63

Processing
 expiry, 916–917
 models, 866
 tick, 212–214
 tick, DTrace, 904
 tick, FSS, 219–220
 tick, FX, 228–229
 tick, RT, 229–231
 tick, threads, 903
 tick, TS, 217–218
 update, 214
 update, FSS, 220–227
 update, TS, 218–219
Process-level file abstractions, 658–668
process.max-msg-messages, 300
process.max-msg-qbytes, 300
process.max-sem-nsems, 297
process.max-sem-ops, 297
Process models, 43
 components, 44–48
 creating, 89–98
 evolution, 48–52
 executable objects, 52–55
 file systems, 110–129
 file systems, implementation, 113–123
 groups, 150–156
 kernel process tables, 79–84
 microstate accounting, 125–129
 resources, attributes, 84–89
 resources, usage, 123–125
 sessions, 150–156
 signals, 129–149
 structures, 55–79
 structures, kernel threads, 73–79
 structures, LWPs, 69–73
 structures, proc, 56–66
 structures, user areas, 66–69
 system calls, 98–106
 termination, 106–110
 unified, 50–52
 zones, 386–389
Process Model Unification project, 48
Processors, 15
 addresses spaces, 459
 AMD Opteron support, 8
 binding, 33
 CMT, 161
 dispatchers, abstractions, 162–171
 dispatchers, observability, 168–171
 Intel x86 support, 8
 MPP, 816

- resource management, 33–34
- RISC, 819
- scheduling classes, 9–10
- selecting, 159
- sequential consistency models, 822
- sets, 165
- SPARC systems. *See* SPARC systems
- state, modifying, 960
- UltraSPARC. *See* UltraSPARC
- procfs (/proc file system), 11, 110–129
 - control messages, 121–122
 - files, 111–112
 - files, types, 118
 - implementation, 113–123
 - indexes, 116
 - I/O, 120
 - libproc, 120
 - reading, 119
 - references, 117
 - sample utility, 975–978
 - subdirectories, 113
 - visibility, 432
 - VM, large page support, 501
 - zones, 387–388
- prochasproperm() function, 347
- proc(4) interfaces
 - file systems, 421
 - privileges, modifying, 339
 - privileges, optimizing, 360–361
- proc_names.c, updating, 974
- proc_sz column, 83
- proc_t process state, 45
- Producer/consumer buffer, 917–918
- Profiles
 - processes, 64
 - RBACs, 11
- Program header (PHT) sections, 97
- Programming interfaces, task queues, 932–933
- project attribute, 419
- project.max-msg-ids, 299–300
- project.max-sem-ids, 296
- project.max-shm-ids, 288
- project.max-shm-memory, 288
- Projects, 35, 415–416
 - databases, 418–419
 - FSS update processing, 220
 - in-kernel project data structures, 421–423
 - interfaces, 419–420
 - kernels, 420–423
 - locks, 423
 - name-service attributes, 433
- Process Model Unification, 48
- resource controls, 423–432
- system calls, 420–421
- zones, 411
- Properties
 - global resource controls, 427–428
 - local resource controls, 428
 - vmem allocators, 553
- Protection. *See also* Security
 - devices, 362–363
 - faults, 473
 - memory, 448
 - models, VM, 473
 - pages, 484–485
- Protocols
 - FTP, zones, 404
 - IPSec, 10
 - IPv6, 10
 - LDAP, 10
 - TCP, connection teardown, 398
 - TCP/IP, performance, 8
 - UDP. *See* UDP
 - UFS locks, 773–774
- Providers
 - DLPIs, 15
 - lockstat, 846–851
 - sched, 904
 - vminfo, 703
- prstat(1) command, 11, 39, 47, 128, 231
- pr_vaddr field, 492
- ps(1) command, 74, 231
- p_sessp, 62
- Pseudo file systems, 30
- Pseudo-terminals, zones, 403–404
- psig(1) command, 132
- PSO (Partial Store Order), 823
- psradm(1M) command, 34
- p_stat, 60
- p_swapcnt, 60
- pthread_kill(3C) interface, 145
- putnext() routine, 858
- putpage() function, 491
- p_wcode, 61
- p_wdata, 61

Q

- QoS (Quality of Service), 10
 - IPQoS, 38–39
- QPAIR perimeters, 864
- Quality of Service. *See* QoS
- Quantum caches, 559

QUANTUM unit of time, 203
 Queries
 bmap_read() function, 758
 pmap(1) command, 642–643
 Queues
 Delete queue, 754
 dispatchers. *See Dispatchers*
 Idle queue, 752–754
 insertion, 235, 240–241
 management, 159
 management, dispatchers, 234–242
 messages, POSIX, 309–312
 messages, System V, 299–303
 models, 866
 monitoring, 241–242
 sleep, 255–257, 845
 STREAMS, 858
 System V messages, 24
 tasks, 927–928
 tasks, DDI, 934–935
 tasks, dynamic, 928–932
 tasks, implementation, 937–941
 tasks, observability, 935–937
 tasks, programming interfaces, 932–933
 tasks, troubleshooting, 940
 user-level sleep, 21
 WR, 954–956

R

RAM (random access memory), 81. *See also Physical memory*
 Random access memory. *See RAM*
 RBAC (role-based access control), 11, 324
 privileges, 357–359
 zones, 385
 rcpd daemon, 6, 39
 RCM (resource configuration manager), 453
 rctl
 attributes, 433–435
 consequences of exceeding, 429–430
 overview of, 424–425
 RCTL_LOCAL_DENY flag, 430
 RCTL_LOCAL_DEV flag, 429
 Reader/writer (RW) locks, 821, 824, 835–840
 probes, 849–851
 read() function, 686
 file system I/O, 707–710
 Reading
 directories, 723–724
 hardware counters, 641–642
 UFS blocks, 760

Read-only mount, /dev file system, 402
 READ permissions, 298
 Ready state (zones), 371
 realitexpire function, 909
 Real-time
 architecture, 14
 callouts, 905
 Real Time (RT) scheduling class, 9, 23, 160
 thread priorities, 229
 tick processing, 229–231
 Reclaim threads, 790
 Reconfiguration, dynamic, 452
 Records, bufctl_audit, 572
 Recovery, UFS, 790
 Redzone indicators, debugging with, 566–569
 References
 counts, 423
 counts, vnode interfaces, 698
 procfs, 117
 searching, 573–574
 siginfo structure, 137
 Registers, segments, 626
 Rehash values, blocks, 596
 Reinitialization, vnode interfaces, 698
 Relaxed Memory Order (RMO), 823
 Releasing
 dispatcher locks, 186
 locks, 834
 semaphores, 846
 Reliability, 3, 12
 threads, 21
 Relocation, pages, TSB, 607
 Removals, cyclic subsystems, 921–922
 Replacement, TSB, 607–609
 Requests
 large pages, compilers, 648–649
 large pages, interfaces, 649–652
 Reserving space in logs, 784–785
 Resizing
 cyclic subsystem, 919–921
 tables, 277
 Resource configuration manager (RCM), 453
 Resource Manager (RM), 9
 Resources
 allocation, 555–556
 assignment, 6
 controls, 87, 161, 423–432
 controls, global, 427–428
 controls, interfaces, 432–437
 controls, kernel interfaces, 437–444
 controls, local, 428
 controls, numeric values for, 426
 controls, policies, 428–429

- controls, shared memory, 288
- controls, signals, 430–431
- controls, System V, 282–284
- controls, tasks, 431–432
- controls, zones, 411
- DRPs, 6
- dynamic resource allocation, 32
- freeing, 555–556
- IPC, creating, 280–282
- limits, 85
- management, 3, 13, 16, 30–39, 160
- management, core process components, 48
- management, observability, 38–39
- management, processors, 33–34
- management, Solaris, 35–38
- management, zones, 6, 370, 407–414
- message queues, 299–301
- pools, 36
- pools, CPUs, 165
- privileges, 86–87
- process attributes, 84–89
- process models, usage, 123–125
- semaphores, 296–297
- restore_mstate() function, 126
- Restrictions, size, zones, 390
- Retired sets, 661
- Right ancestors, RPARENT, 663
- Rights management, 323. *See also* Privileges
 - processes, 7–8
- RISC processors, 819
- rlimit interface, 431
- rmalloc() function, 560
- RMID, 275
- RMO (Relaxed Memory Order), 823
- RM (Resource Manager), 9
- Role-based access control (RBAC), 11, 324
 - privileges, 357–359
 - zones, 385
- Roles, defining, 10–11
- Rolling logs, 787–788
- Root Set (privileges), 329
- Root vnode identification, 683
- Routines
 - ACLs, modifying, 766
 - getcpuid(3C), 808
 - lgroup_home(3C), 808
 - madvice(3C), 809
 - madv.so.1, 809
 - meminfo(2), 808
 - putnext(), 858
 - segvn_fault(), 708
- Routing zones, 398
- RPARENT node, 663
- RT (Real Time) scheduling class, 9, 23, 160
 - thread priorities, 229
 - tick processing, 229–231
- Running state (zones), 372
- Run queues, 157. *See also* Dispatchers
- RUN state, 76, 158
- Runtime
 - privileges, 342–343
 - zones, 371–375, 401–402
- rw_exit() function, 838
- rw_exit_wakeup() function, 839
- RW (reader/writer) locks, 821, 824, 835–840
 - probes, 849–851

S

- Safe privileges (zones), 381–382
- SA_SIGINFO flag, 138
- /sbin/sh process, 458
- sbrk() function, 462
- scaches, segmap, 505
- Scalability, 3
 - dynamic task queues, 929
 - performance, 12
 - stacks, 863
 - synchronization, 825
 - threads, 21
- Scanners
 - pages, 516–518
 - pages, implementation, 522–524
 - pages, parameters, 521–522
 - rate parameters, 518–520
- schedpaging function, 909
- sched provider, 904
- Schedulers, 15
 - activation, 217
 - fair-share, 222
 - FSS, 221
 - memory, 524–525
 - multiple, support, 14
 - zones, 408
- Scheduling, 18–19
 - callouts, 904–910
 - classes, 9–10, 22, 160
 - classes, dispatchers, 192–207
 - classes, frameworks, 196
 - classes, functions, 198–202
 - kernels, 22–23
 - NUMA, 802–803
 - processes, 16
 - threads, 49
 - workloads, 159

sclass array, 194
 sclass_t object, 195
 SDT probes, 936–937
Searching
 directories, 723
 references, 573–574
secpolicy vnode_setattr() function, 349
Security
 devices, 398
 DTrace, 7
 IP, 397
 IPSec, 10
 privileges, 326. *See also* Privileges
 zones, 6, 367, 370, 379–386
segkmem driver, 535
seg_kpm driver, 710
seg_kp segment, 532
segmap cache, 505
seg_map driver, 710–718
Segments
 allocation, 557–558
 contiguous physical memory, 510
 drivers, kernel memory, 535–537
 drivers, VM, 476–485
 freeing, 557–558
 kernelmap, 536
 kernels, 528–530
 kernels, address space, 533–534
 memory management, 27
 registers, 626
 seg_kp, 532
 seg_map driver, 710–718
 tracking, 556–557
SEGOP_FAULT(), 478
seg_pupdate function, 910
seg_vn driver, 481
segvn_fault() routine, 708
Selection
 addresses, 397
 CPUs, 236
 MPSS, 452
 pages, 639
 processors, 159
 syscall numbers, 971–972
 threads, 159
Semantics, privileges, 334–344
sema_p() function, 846
Semaphores
 APIs, 306
 events, 308
 kernels, 824, 844–846
 named, 307
 POSIX, 305–309
 releasing, 846
 System V, 24, 295–298
 unnamed, 305
semds_id structure, 297
semget() function, 296
semop(2) page, 298
sem_t structure, 308
Sensors, DTrace, 7
 Sequence numbers, slots, 275
 Sequential consistency models, 822
Servers, door, 25
Serviceability, 12
Service Management Framework (SMF), 5
Services
 APIs, 45
 DLPI, 888
 GLDv3 module, 886–888
 IOC framework design, 275–276
Sessions, process models, 150–156
setbackdq() function, 234, 236
Set fields, signals, 135
setfrontdq() function, 234, 240
setkpdq() function, 234
setpprив() function, 337
setrun function, 909
setrun() function, 235
Sets
 active, 661
 bits, 345–346
 LSB, 662
 ops vector, 439–440
 privileges, 324. *See also* Privileges
 processors, 33, 165
 resource controls, 425
 retired, 661
Set-uid applications, 325
Set-uid interfaces, 356–357
sf_hment structure, 591–594
SGA (System Global Area), 294
Shadow HME blocks, 597–598
Shadow inodes, 745
Shareable virtual devices, 400
Shared libraries
 interposing, 647–648
 optimization, 520–521
Shared mapped files, 464
Shared memory, 448
 DISM, 294–295
 ISM, 291–294, 613–616, 645
 NUMA, 803
 POSIX, 304–305
 System V, 286–295
Shared Memory Multiprocessor, 816

- Shares, scheduling classes, 9–10
- Sharing
 - executables, 458
 - libraries, 458
- shmfd(2) interface, 290
- shmid_ds structure, 287
- shm_open interface, 304
- shm_unlink interface, 304
- Shutting_down state (zones), 372
- Shuttle switching, 320
- SIGABRT signal, 106
- sigaction(2) system, 138
- sigalarm2proc function, 910
- SIGCLD signal, 107
- siginfo structure, 135–139
- Signals
 - activity, observing, 148–149
 - asynchronous, 145–148
 - behavior, 49
 - blocking, 132
 - core process components, 48
 - delivery, 269
 - generation, 133–134, 135
 - implementation, 135–148
 - kernels, 25–26
 - masks, 138
 - process models, 129–149
 - procfs, 122
 - resource controls, 430–431
 - SIGABRT, 106
 - SIGCLD, 107
 - SIGSEGV, 470
 - SIGTRAP, 493
 - SIGWAITING, 108
 - synchronous, 141–145
 - threads, 21
 - traps, 134
 - zones, 386
- sigprocmask(2) system, 132
- sigkill() function, 147
- SIGSEGV signal, 470
- SIGTRAP signal, 493
- SIGWAITING signal, 108
- Simplicity of privileges, 328
- Size
 - CSIZE, 663
 - heaps, 463
 - large pages. *See* Large pages; Multiple pages
 - magazines, 546
 - MPSS, 452
 - pages, allocation, 642–644
 - pages, configuration, 645–653
- pages, support, 644–645
- page-size, 496
- physical memory, 448
- restrictions, zones, 390
- Slab allocators, 29, 537–551
- Sleep
 - kernels, 253–262
 - processes, 257–261
 - queues, 255–257
 - threads, 235
- sleepq_head array, 845
- Sleep queues, 845
- SLEEP state, thread, 76, 158
- Slots
 - indexes, 275
 - locks, 277
 - pages, swapping, 491
 - seg_map driver, 714
- slowscan pages, 518
- smap structures, 711–712
- SMF (Service Management Framework), 5
- SMP (symmetric multiprocessor), 795
- socket() function, 872
- Sockets
 - UDP, 878
 - zones, 405
- softcall() function, 908
- Soft swapping, 524
- Software
 - time-of-day clocks, 911
 - TLB replacement strategies, 584
 - traps, 99
- Solaris
 - file system frameworks, 668–672
 - overview of, 3–4
 - resource management, 35–38
- Solaris 8 features, 10–11
- Solaris 9 features, 9–10
- Solaris 10
 - features, 5–8
 - file system conversion, 734–736
- Solaris Doors, 25, 312–320
 - implementation, 314–320
 - interfaces, 313
 - overview of, 313–314
- Solaris Fault Manager, 5
- Solaris Logical Volume Manager (SVM), 775
- Source analysis, execution control, 960–961
- Source compatibility, 52
- SPARC Reference MMU (SRMMU), 588
- SPARC systems, 4
 - address spaces, 459–461

SPARC systems (*continued*)

- HAT layers, 620–625
- kernels, text, 530
- locks, 820
- PIL, 829
- system calls, 99–101
- Speed, NICs, 891–895
- Spillover text, 532
- Spin locks, 828
 - probes, 848–849
- Spreading, interrupt load, 894–895
- squeue_create() function, 868
- squeues, 864–865
- SRMMU (SPARC Reference MMU), 588
- Stacks
 - checksums, offload, 890–891
 - design, 862–863
 - device drivers, 882–891
 - frames, 97
 - frameworks, 863–870
 - interrupts, 891–895
 - IP, 880–882
 - mapping, 462–463
 - networks, 855
 - physical memory, 449
 - processes, memory mapping, 457
 - resource limits, 85
 - STREAMS subsystem, 855–862
 - synchronization, 870
 - synchronous STREAMS, 878–880
 - TCP, 870–875
 - time-of-day clocks, 911
 - UDP, 875–878
- Starting
 - threads, 269
 - transactions, 785–786
- State
 - IDL, 158
 - models, zones, 371–372
 - ONPROC, 158
 - per-process, 334
 - PINNED, 158
 - privileges, modifying, 335–339
 - processes, 159
 - processors, modifying, 960
 - RUN, 158
 - RW locks, 837
 - SLEEP, 158
 - threads, 158
 - transitions, 334–335
 - TS_ONPROC, 188
- Static linked objects, 52

Statistics

- chips, 163
- cpustat command, 641–642
- dispatcher locks, 189–190
- DNLC, 733
- DTrace probe arguments, 703
- kernels, 170
- kstat counters, 935–936
- locks, 834
- lockstat providers, 846–851
- MPO, 813–814
- seg_map driver, 714–716
- slab cache, 548–551
- TLB misses, 640
 - utilities, hardware, 11
- Stevens, W. Richard, 273
- Stopping threads, 269
- STOP state, threads 76
- Storage
 - memory, 448
 - TLS, 50
 - TSB, 531, 583–584, 601–613
- Storage-based file systems, 30
- Store operations, 819
- STREAMS subsystem, 140
 - data block, 348
 - dynamic task pools, 940–941
 - network stacks, 855–862
 - removal of, 8
 - synchronous, 878–880
 - zones, 405
- struct mmu, 627–628
- Structures
 - anonymous memory, 486
 - arenas, 557
 - callout_table, 905
 - connections, 868
 - cpu_t, 166
 - cred_t, 380
 - deltamap, 780
 - dispatchers, 172–175
 - dispatchers, linkage, 175–177
 - dispatchers, queues, 176
 - dispatchers, viewing, 177–183
 - execsw, 94
 - files, 666–668
 - fssproc_t, 194
 - hash tables, 589
 - HAT, 585–588
 - hme_blk, 594–597
 - hme_blk, hash tables, 599–601
 - ID, IPC names, 281

- in-core log data, 779–782
- in-kernel project, 421–423
- IP, 861
- ipc_perm, 281
- ipc_service, 276
- ISM, 614
- kernels, resource control interfaces, 438–439
- kmdb, 949–959
- logmap, 781
- lrusage, 123–125
- mapentry, 781
- matamap, 781
- ml_odunit_t, 778
- mntopts_t, 673
- mutex locks, 830
- on-disk log data, 776–779
- pages, 506, 508–509
- PID, 63
- prcommon, 115
- privileges, 346–349
- process models, 55–79
- process models, kernel threads, 73–79
- process models, LWPs, 69–73
- process models, proc, 56–66
- process models, user areas, 66–69
- process resource control, 88
- projects, 418
- semds_id, 297
- sem_t, 308
- sf_hment, 591–594
- shmid_ds, 287
- siginfo, 135–139
- smap, 711–712
- Solaris Doors, 315
- TCP, 861
- TSB, 604
- UDP, 861
- uf_entry_t, 660
- ufs_acl, 765
- ufs_fsd, 765
- uio_resid, 723
- undo, 282
- VM, 530
- watchpoints, 494
- Subdirectories, /proc, 113
- SUGID flag, 337
- Summaries
 - file systems, 782–783
 - statistics, 640
- Sun Fire T2000, 169
- sun4u kernel, 599
- Superblocks, 747–748
- Superusers, 7–8
 - privilege models, 326
 - RBACs, 11
- supgroupmember() function, 347
- Support
 - AMD Opteron processor, 8
 - DNLC, 728
 - dynamic topology, 799
 - HAT, 500–501
 - Intel x86 processor, 8
 - large kernel pages, 606–607
 - large pages, changes to, 494–501
 - LDAP, 10
 - MPSS, 9, 646
 - multiplatform, 13
 - multiple file system, 14
 - multiple scheduler, 14
 - pages, CPU, 652–653
 - pages, sizes, 644–645
 - pseudo-terminals, 403–404
 - SPARC systems, 4
 - vfs interfaces, 679–681
 - vnode interface functions, 696
 - zone runtime, 373–374
- SVM (Solaris Logical Volume Manager), 775
- Swap files, 449
- swapfs layer, 489–491
- Swapping, 450
 - hard, 525
 - pageable memory, 532
 - page-outs, physical swaps, 491
 - soft, 524
- switch() function, 242–246
- Switching
 - context, 159
 - dispatchers. *See* Dispatchers
 - NIC mode, 893
- Symmetric multiprocessor (SMP), 795
- Synchronization, 815–816
 - DTrace lockstat providers, 846–851
 - hardware, 819–824
 - HAT layers, 616–620
 - IPC, 23. *See also* IPC
 - kernels, semaphores, 844–846
 - management, 861–862
 - mutex locks, 827–835
 - objects, 21, 824–827
 - parallel systems architecture, 816–819
 - processes, 825–826
 - reader/writer (RW) locks, 835–840
 - stacks, 870
 - turnstile, 840–843

Synchronous file system transactions, 783
 Synchronous signals, 141–145
 Synchronous STREAMS, 878–880
SYS_DEVICES privilege, 404
SYS (System) scheduling class, 23, 160
 System calls, 46
 adding, 971–974
 file system I/O, 709–710
 interfaces, 16, 351–352
 navigating, 101–106
 open() function, 661
 private kernel interfaces, 436
 process models, 98–106
 procfs, 122
 projects/tasks, 420–421
 testing, 974
 System Global Area (SGA), 294
 System-level performance, 560–561
 System page scanner daemon, 505
 System (SYS) scheduling class, 23, 160
 System time facilities, 910–911
 System V IPC, 24–25
 framework, 274–282
 IPC, zones, 406–407
 message queues, 299–303
 resource controls, 282–284
 semaphores, 295–298
 shared memory, 286–295

T

Tables
 callouts, 904–910
 dispatchers, 202–207
 headers, 54
 HPT, 589
 ipc_service structures, 276
 IPT, 588
 kernel process, 79–84
 mount options, 673
 per-process file, 659
 resizing, 277
 RT, 203
 syscall tables, adding entries, 972–973
 translation, 588–601
 TTE, 590–591
`t_affinitycnt`, 75
 Tags, TTE, 591
 Target directories, configuration, 10
 Target layer, 952
 mdb, 944–945
 task attribute, 419

`taskq_create()` function, 931
`taskq_dispatch()` function, 931–932
`taskq_lock()` function, 932
`taskq_member()` function, 932
`taskq_resume()` function, 932
`taskq_suspended()` function, 932
`taskq_suspend()` function, 932
`taskq_wait()` function, 932
 Tasks, 35, 416–417
 FSS update processing, 220
 interfaces, 419–420
 kernels, 420–423
 mount method, 682
 pools, 930–931
 privileges, 46
 queues, 927–928
 queues, DDI, 934–935
 queues, dynamic, 928–932
 queues, implementation, 937–941
 queues, observability, 935–937
 queues, programming interfaces, 932–933
 queues, troubleshooting, 940
 resource controls, 423–432, 431–432
 system calls, 420–421
 unmount method, 683
`t_astflag`, 78
`t_back`, 77
`t_bind_cpu`, 75
`t_bound_cpu`, 74
`t_cid`, 77
`t_cldata`, 77
`t_clfuncs`, 77
`t_clfuncs_pointer`, 198
 TCP/IP (Transmission Control Protocol/Internet Protocol)
 performance, 8
 stacks (as STREAMS modules), 860
 TCP (Transmission Control Protocol)
 connection teardown, 398
 flow, 871
 loopback, 874–875
 stacks, 870–875
 streams, creating, 859
 structures, 861
 synchronous STREAMS, 878–879
`t_cpu`, 77
`t_cred`, 77
`t_disp_queue`, 77
`t_disp_time`, 77
 Teardowns, TCP connections, 398
 Teer, Rich, 273
 Templates, options, 673
`t_epri`, 76

Terminal I/O, 946, 956
 Termination, process models, 106–110
 term_mstate() function, 126
 Test-and-set instructions, 819
 Testing
 infinite time quantum, 230
 ops vector, 439–440
 system calls, 974
 TS priorities, 214–215
 Text
 allocation, 531
 kernels, 528–530
 t_flag, 75
 t_forw, 77
 Third-party file system privileges, 344
 t_hold, 77
 thread_create() function, 234
 thread_high() function, 188
 Thread ID (TID), 46
 Thread local storage (TLS), 50
 thread_lock() function, 188
 Threads, 18–19, 795
 blocking, 202
 bound, 21, 49
 clocks, 901–904
 configuration, 211–233
 core process components, 48
 dispatchers. *See* Dispatchers
 DNLC, 733
 dynamic task queues, 929
 of execution, 15
 interrupts, 264–266
 interrupts, priorities, 266
 kernels, models, 20–21
 kernels, process model structures, 73–79
 library, 10, 11
 limits, 83–84
 locks, 849
 model evolution, 49–50
 mutex locks, 183–190
 objects, 44–47
 placement, 802
 preemption, 235
 priorities, change, 235
 priorities, FX, 227–228
 priorities, monitoring, 231–233
 priorities, RT, 229
 reclaim, 790
 scheduling, 49
 selecting, 159
 signals, 132
 sleep, 235
 starting, 269
 states, 158
 stopping, 269
 tick processing, 213, 903
 users, 19, 84
 waiters, 825
 wakeup, 235
 THREAD_SET_STATE macro, 187
 Three-way handshakes, 869
 throttlefree parameter, 512
 Throttles, pages, 512
 Throughput, networks, 8
 Thundering herd problem, 835
 Tick processing, 212–214
 DTrace, 904
 FSS, 219–220
 FX, 228–229
 RT, 229–231
 threads, 903
 TS, 217–218
 TID (thread ID), 46
 Time
 class functions, 211–214
 not-recently-used, 520
 resource limits, 85
 scheduling, 159
 Time-of-day clocks, 910–911
 timeout_common() function, 907
 timeout(9F) interface, 904
 Timeouts, 390
 Timers, 17
 arbitrary resolution interval, 12
 clocks, interactions, 902
 cyclic subsystem, 912–925
 system time facilities, 910–911
 Timeshare (TS) scheduling class, 9, 160
 priorities, 214–217
 tick processing, 217–218
 update processing, 218–219
 timestamp field, 572
 t_kpri_req, 77
 TLB (transaction lookaside buffer), 583, 639
 cpustat command, 641–642
 iTLB, 645
 trapset(1M) command, 640–641
 t_link, 74
 t_lockp, 77
 t_lpl, 77
 TLS (thread local storage), 50
 t_lwp, 77
 t_next, 77
 tod_set() function, 911

t_oldspl, 77
 Tools, 11
Dtrace. See DTrace
 lgroup observability, 168
 mdb, 11, 65, 82
 observability, 640
 /proc, 11
 Topologies
 dynamic topology support, 799
 UltraSPARC-I IV MMU, 584
 -T option, trapstat(1) (TLB misses), 640
 -t option, trapstat(1) (TLB misses), 640
 Total Store Ordering (TSO), 823
 TP_MSACCT flag, 126
 t_post_syscall, 78
 t_preempt, 75
 t_pre_sys, 77
 t_prev, 77
 t_pri, 76
 t_prioinv, 78
 t_proc_flag, 75
 t_procp, 77
 Tracing
 allocators, 562–577
 caches, enabling, 562–563
 DTrace, 7
 VM, 466–467
 Tracking
 CPUs, 170
 HME, 591
 hme_blk structures, 594–597
 priority fields, 231
 privileges, 360
 segments, 556–557
 transactions, 781
 Traditional Unix IPC, 24
 Transaction lookaside buffer. *See TLB*
 Transactions
 ending, 786–787
 logs, 576
 starting, 785–786
 tracking, 781
 UFS, 783–787
 TRANS_BEGIN_ASYNC macro, 785
 TRANS_BEGIN_CSYNC macro, 786
 TRANS_BEGIN_SYNC macro, 785
 Transitions, state, 334–335
 Translation
 HAT, implementation, 631–636
 HAT, ISM, 613–616
 HAT, overview of, 581–583
 HAT, pages, 506
 HAT, SPARC, 620–625
 HAT, support, 500–501
 HAT, synchronization, 616–620
 HAT, UltraSPARC layer, 583–625
 HAT, VM design, 457
 HAT, x64, 625–636
 tables, 588–601
 virtual memory, 269
 virtual-to-physical (memory), 449–450
 Translation Storage Buffer (TSB), 531, 583–584, 601–613
 Translation Table Entry (TTE), 590–591
 Transmission Control Protocol/Internet Protocol.
See TCP/IP
 Transmission Control Protocol. *See TCP*
 Transparency, zones, 368
 trans_roll() function, 787
 TRANS_TRY_BEGIN_ASYNC macro (UFS), 786
 TRANS_TRY_BEGIN_CSYNC macro (UFS), 786
 Traps
 handlers, 108
 handling, 961–962
 signals, 134
 system calls, handling, 100–101
 system calls, on SPARC, 99
 trapstat(1M) command, 640–641
 Traversals
 bmap_write() function, 759
 path-name functions, 724–725
 zone file systems, 392–393
 Trees
 binary, file descriptor integer space, 662
 CSIZE, 663
 LPARENT, 663
 RPARENT, 663
 Triggering signals, 25–26
 Troubleshooting. *See also Debugging*
 buffers, 575
 dynamic task queues, 929–930
 large pages, 499–500
 memory, 573–574
 memory, detecting corruption, 565–566
 panic messages, associating, 570
 STREAMS-based stacks, 862
 task queues, 940
 TSB, 609–613
 UFS, 790
 uid 0, 340–341
 truss(1) command, 11
 updating, 973–974
 Trusted Solaris privilege model, 329
 TSB (Translation Storage Buffer), 531, 583–584, 601–613
 t_schedflag, 75

- ts_globpri field, 204–205
- t_sig, 77
- t_sig_check, 78
- t_sigqueue, 77
- ts_lwait field, 206
- tsmaxwait field, 206
- TS_NEWUMDPRI macro, 214
- t_sobj_ops, 77
- TS_ONPROC state, 188
- TSO (Total Store Ordering), 823
- ts_parmsset() function, 215
- ts_quantum field, 205
- ts_slpret field, 206
- t_stack, 74
- t_state, 75
- TS (Timeshare) scheduling class, 9, 160
 - priorities, 214–217
 - tick processing, 217–218
 - update processing, 218–219
- ts_tqexp field, 205
- ts_update() function, 218, 910
- ts_wakeup() function, 261–262
- t_sysnum, 77
- TTE (Translation Table Entry), 590–591
- t_tid, 77
- t_trapret, 78
- t_ts, 78
- Tuneables
 - IPC, configuring, 285
 - semaphore kernel, 296
 - TSB, 620–621
- turnstile_lookup() function, 842
- Turnstiles, 825
 - implementation, 841–843
 - placement, 834
 - synchronization, 840–843
- turnstile_table[] array, 841
- turnstile_wakeup() function, 843
- t_wchan, 76
- t_wchan0, 76
- Two-handed clock algorithm, 517
- Types
 - of chips, 163
 - of terminals, 957
 - of vnode interfaces, 688
- U**
- UDP (User Datagram Protocol), 875–878
 - structures, 861
- uf_entry_t structure, 660
- ufs_acl structure, 765
- ufs_fsd structure, 765
- ufs_read method, 760
- UFS (Unix file system), 10, 11
 - access control, 764–767
 - architecture, 749–750
 - blocks, allocation, 754–760
 - blocks, booting, 746
 - blocks, reading/writing, 760
 - blocks, superblocks, 747–748
 - cylinder groups, 748–749
 - development history, 737–738
 - directories, 742–744
 - extended attributes, 767–768
 - failure recovery, 790
 - hard links, 744–745
 - inodes, 751–764
 - inodes, shadow, 745
 - locks, 768–774
 - locks, protocols, 773–774
 - logging, 775–790
 - on-disk formats, 739–750
 - summaries, 782–783
 - transactions, 783–787
- ufs_write method, 762–764
- uid 0, troubleshooting, 340–341
- UID (user ID), 59, 336
- uios_resid structure, 723
- UltraSPARC. *See also* SPARC systems
 - CMT, 797
 - CPU specific large page support, 652–653
 - HAT layer, 583–625
 - kernels, 531
 - locks, 820
 - trapset(1M) command, 640–641
- UMA (Uniform Memory Access), 166
- undo structure, 282
- Unified process models, 50–52
- Uniform Memory Access (UMA), 166
- Uninitialized data, detection, 569
- Units of physical memory, 448–449
- UNIX
 - IPC, 24
 - privileges, 325–333. *See also* Privileges
- Unix file system. *See* UFS
- umount method, 683
- Unnamed semaphores, 305
- unode, 30
- Unsafe devices, 399
- Unsafe privileges (zones), 382–383
- unsleep function, 827
- untimeout(9F) interface, 908

Update processing, 214
 FSS, 220–227
 TS, 218–219

Updating
`/etc/name_to_sysnum`, 973
`proc_names.c`, 974
`truss(1)`, 973–974

Usage, process resource, 123–125

User areas, 48
 process model structures, 66–69

User credential library interfaces, 355–356

User Datagram Protocol. *See UDP*

User ID. *See UID*

User-level sleep queues, 21

User preemption, 246

User priorities, threads, 209–211

User processes, DTrace, 7

User threads, 19, 44, 49, 84

`u_sigmask` [] field, 140
`u_signal` [] field, 140
`u_signodefer` field, 140
`u_sigonstack` field, 140
`u_sigresethand` field, 140
`u_sigrestart` field, 140

Utilities, 11. *See also Tools*
 DTrace. *See DTrace*
 statistics, hardware, 11

Utilization
 CPU, 48
 fields, 63
 page scanner CPU clamp, 521

V

Validation, virtual addresses, 593

Verification, interface versions, 810

Vertical perimeters, 864–868
 TCP, entry points, 871

`vfork()` function, 469

`VFS_FREEVFS` method, 677

`vfs` interfaces, 30, 668, 675–685

`VFS_MOUNT` method, 677

`VFS_MOUNTROOT` method, 677

`VFS_ROOT` method, 677

`VFS_STATVFS` method, 677

`VFS_SYNC` method, 677

`VFS_UNMOUNT` method, 677

`VFS_VGET` method, 677

VFS (virtual file system), 14, 29–30

`VFS_VNSTATE` method, 677

Viewing
 caches, 563–565
 dispatcher structures, 177–183

dispatch tables, 202
 large pages, 494–495

Virtual addresses
 address space, 457–466
 aliasing, 592
 kernel maps, 965–969
 space layout, 628–631
 validation, 593

Virtual devices, 400

Virtual file system. *See VFS*

Virtualization
 devices, 398
 zones, 368

Virtual memory. *See VM*

Visibility
 global zones, 387
`procfs`, 432
`v_maxup ttl` value, 81
`v_maxup` value, 81
`VM_BESTFIT` policy, 555, 558
`vmem_add()` function, 553
`Vmem` allocators, 552–562
 implementation, 556–560
 interfaces, 553–556
 performance, 560–561
 properties, 553
`vmem arenas`, 937–939
`vmem_create()` function, 53
`vmem_free()` function, 558
`vminfo` provider, 703
`VM_INSTANTFIT` policy, 555, 558
`VM_NEXTFIT` policy, 556, 558
`VM` (virtual memory), 26, 269
 address space, page faults in, 473–474
 address space, management, 467–476
 anonymous memory, 485–486
 anonymous memory, layers, 487–488
 data structures, 530
 design, 455–457
 file system caches, 450–451
 implementation, 451–453
 kernels, layouts, 527–534
 layers, 456
 levels of, 448
 overview of, 447
 protection, 448
 protection, models, 473
 resource limits, 85
 segment drivers, 476–485
 sharing, 448
 support large pages, changes to, 494–501
`swapfs` layer, 489–491
 tracing, 466–467
 translation, 269

virtual address space, 457–466
 virtual-to-physical translation, 449–450
 watchpoints, 492–494
vn_alloc() function, 696
vnode interfaces, 30, 668, 685–706
 caches, 698–700
 DTrace probes, 703–706
 life cycle, 696–697
 mdb(1) kernel debugging facility, 701–703
 methods, 690–695
 methods, registration, 688–690
 pages, block I/O, 700
 reference counts, 698
 root identification, 683
 support functions for, 696
 types, 688
VOP_INACTIVE() method, 697
vop_lookup() method, 723
vop_map method, 708
vop_readdir() method, 723–724
vop_read() method, 687
v_proc value, 81

W

Waiters, 825, 831
Wakeup
 kernels, 253–262
 threads, 235
Warm affinity, 161
Warm caches, 162
Watchpoints
 management, 961
 VM, 492–494
Workloads, scheduling, 159
Work request queue (WR), 954–956
Wrappers
 libdl interfaces, 956
 ppgsz, 646
 vnode functions, 687
write() function, 686
 file system I/O, 707–710
Write-through caches, 822
Writing
 syscall handlers, 972
 UFS blocks, 760
WR (work request queue), 954–956

X

X64 address space layout, 461
X86 address space layout, 461

X64 HAT layer, 625–636
-xpagesize_heap option, 648–649
-xpagesize option, 648
-xpagesize_stack option, 649
-xs option, 642
xxproc_t, 175

Z

Zero-fill-on-demand (ZFOD), 485
ZFOD (zero-fill-on-demand), 485
zlogin(1) command, 403
ZOMBIE state, 76, 107
zoneadm command, 373–374
zone attribute, 419
zonecfg file system configuration, 389–390
Zones, 5–6, 36
 accounting, 411
 administration, 370
 booting, 375–379
 chroot interactions, 385–385
 compatibility, 370
 configuration, 401
 console design, 402–404
 core files, 389
 credentials, 380
 devices, 398–404
 doors, 405
 DTrace, 413–414
 file systems, 389–393
 FSS update processing, 220
 FTP, 404
 granularity, 368
 interfaces, 395–396
 IPC, 405–407
 isolation, 368
 kstat framework, 412–313
 listing, 374–375
 names, 372–373
 networks, 393–398
 observability, 407–414
 overview of, 367–317
 partitions, 394–395
 performance, 409–410
 POSIX, 407
 privileges, 380–384
 process models, 386–389
 procfs, 387–388
 projects, 411
 pseudo-terminals, 403–404
 RBAC, 385
 resource management, 370, 407–414
 routing, 398

- Zones, 5–6, 36
 runtime, 371–375, 401–402
 security, 367, 370, 379–386
 signals, 386
 size restrictions, 390
 state models, 371–372
 transparency, 368
 virtualization, 368
 zsched command, 374