

---

# Index

[ ' ], 321  
\ character, 282, 463  
: character, 463, 1388, 1533  
#! character, 826  
\* character, 349, 459  
/ character, 1156, 1533  
^ character, 274–275  
\_ character, 237n, 545

## A

ABI (application binary interface), 224–225, 231–232  
Absolute- and clock-time intervals, 782–784  
Absolute branching, 201  
Abstract Windowing Toolkit, 102n  
Accelerate framework, 213

ACPI (Advanced Configuration and Power Interface), 1598  
ACPI plane, 1598  
Activation/shuttle, 714–716  
Active data, 1036  
Active matching, 1256, 1280  
Active queue, 871  
Actual parameters, 231  
AD (address and data or control information), 165  
Address Book, and Spotlight, 1440–1441  
Address translation, 181–183, 404–406  
Adler checksum, 335  
Advanced Micro Devices (AMD), 162  
Advisory-mode file locking, 1230–1232  
ADVLOCK, 1581  
AES encryption, 920  
Affinities, 151–152  
AFP (Apple Filing Protocol), 60, 1393–1395  
Agent (daemon), 472, 491  
Aggregation, Ethernet ports, 455  
AGP (Accelerated Graphics Port), 157  
AIFF files, 1398  
AIM (Apple-IBM-Motorola), 160n  
aio\_init(), 449  
Alarm, 524–526  
Aleph, 16  
Aliases, 1554–1556  
ALLOCATE, 1581  
Allocation block consumption, 1508  
Allocation blocks, 1477  
Allocation file, 1505  
Allocation strategies (Xsan), 152

- Allocation zone, 1561–1562
- AltiVec, 212, 402
  - faltivec, 214–215
  - instructions, 218–219
  - 970FX AltiVec implementation, 215–218
  - prefetching, 187
  - vector computing, 213–215
- Amazon architecture, 168
- amber, 131, 256–261
- Amelio, Gil, 6
- Analysis tools, 1474–1477
- AND masks, mouse pointer demo, 317
- Animation, Forth code, 306–312
- Annotated output, 241
- AOL Instant Messaging (AIM), 38
- Apache web server, 31
- APIs, 116, 726
  - and ABIs, 224–225
  - and aliases, 1554–1556
  - Authorization Services, 135–138
  - BIOS, 364
  - Bootstrap Server, 1068–1079
  - BSD, 102–103, 502–504
  - C Library, 74–75
  - Carbon, 27, 31–33, 103–107, 1555
  - Carbon File Manager, 1407, 1430, 1516
  - Carbon Multiprocessing Services, 769–771
  - Cocoa, 107–113, 953, 1555
  - comm area, 597
  - Common Security Services Manager (CSSM), 135
  - Core Foundation notification, 1197
  - Core Services, 89–90
  - diagnostics, 358
  - Disk Arbitration subsystem, 1353–1355
  - dyld (dynamic link editor), 933
  - file system, 1387
  - graphics and multimedia, 91–101
  - Java, 113–114
  - launch, 476, 1072
  - Mac OS X file system, 1387
  - Mach, 519–529, 948
  - Mach task, 702–705, 727
  - Mach thread, 707–710, 727
  - Mach VM, 880–893, 948
  - MDQuery, 1431–1434
  - memory, 949, 952, 966
  - notify(3), 1184, 1191
  - plug-in, 81
  - POSIX, 759, 1407, 1442
  - processor, 693–696
  - processor set, 690–691
  - PThreads, 759–760, 802
  - Quartz Services, 1293
  - QuickTime, 114
  - security, 135, 1442
  - Spotlight, 120, 1431–1432
  - Uniform Type Identifiers (UTI), 830–833
  - universal page lists (UPLs), 912
  - Unix, 1555–1556
  - WebObjects, 113
  - X Window System, 103
  - .app, 78
  - AppKit.framework, 107–108
  - Apple Cluster File System (ACFS), 1387
  - Apple Disk Image Controller (hdix), 463
  - Apple EFI Runtime kernel extension, 1589–1590
  - Apple Events (AE), 101, 1172–1181
  - Apple Expo, 34
  - Apple Filing Protocol (AFP), 60, 1393–1395
  - Apple Motion Sensor. *See* Sudden Motion Sensor (SMS)
  - Apple partitioning scheme, 1348–1352
  - Apple Public Source License (APSL), 47
  - Apple RAID header, 344–346
  - Apple Remote Desktop
    - and Security Agent, 136*n*
  - Apple Search Kit, 1412
  - Apple Session Protocol (ASP), 1394
  - Apple System Logger (ASL), 1077, 1325
  - Apple Type Services (ATS), 101
  - appleCLUT8, 437
  - AppleDouble format, 1557
  - AppleFileServer.app, 60
  - AppleRAID, 344, 345, 1362
  - AppleScript, 106, 115, 125–126, 1173–1175
  - AppleShare, 1393
  - AppleTalk, 27
  - AppleWorks, 37
  - Application bundle, 76–78
    - and AppleScripts, 1175
    - Classic Startup.app, 114
    - and UTIs, 832
    - See also* Bundles
  - Application environments, 101–117
    - BSD, 102–103
    - Carbon, 103–107

- Classic, 114–116
- Cocoa, 107–113
- Java, 113–114
- QuickTime, 114
- Rosetta, 116–117
- WebObjects, 113
  - X Window System, 103
- Application scaling mode, 118
- Application Services, 106
  - Core Audio, 100
  - Core Image and Core Video, 96–99
  - OpenGL, 96
  - Quartz, 91–95
  - QuickDraw 2D, 95–96
  - QuickTime, 99–100
  - subframeworks, 101
- Application-defined signals, 1132
- Aqua user interface, 33–34, 37, 45, 117
- Arbitration, 164*n*
- Architecture
  - Core Image, 97
  - EFI, 368
  - graphics, 91
  - Mach VM, 850
  - Power Mac G5, 158, 164
  - PowerPC 970FX, 188, 222–224
  - QuickTime, 99
  - runtime, 61–74
  - security, 133–135
  - Spotlight, 1410–1415
  - system, 164
  - x86-based Mac computers, 1587–1588
  - Xgrid, 147–149
  - Xsan, 151
- Argument munging
  - BSD system calls, 563–566
- Argument types
  - Fsevents, 1418
- Argument (use of term), 231
- as (GNU-based assembler), 62
- ASCII PPM (portable pixmap), 322
- ASSERT(), 579
- assert(), 579
- Associativity, 179
- Asterisk character, 349, 459
- Asymmetric copy-on-write, 864
- Asynchronous I/O
  - signals, 1132
  - subsystem, 449
- Asynchronous system trap (AST), 465, 532
- ATAPI (ATA Packet Interface), 298
- Atlas supervisor, 835–836
- Atomic compare-and-store function, 244–246
- Atomic operations, 1212–1213
- attach(), 442, 1281
- Attributes file, B-Tree, 1524–1530
- ATTRLIST, 1581
- Audio, 100
  - audit, 143
- Audit control port, 620–621
- Audit daemon, 620
- Audit logging, 449, 620–623
- Audit macros, 623–624
- Audit token, 1038
- Audit trail, 143
  - auditd, 143, 449, 620
- Audit-event-to-event-class mapping, 449
- Auditing support, 619–625
- Auditing system, 134
  - audit control file, 144–145
  - components, 143
  - audit\_init(), 449, 619
  - audit\_worker(), 711
- AULab.app, 100
- Authorization services, 135–138
- Automator, 40, 126
  - Automator action, 78

## B

- background-color, 302
- Backing storage, 853*n*
- Backing store, 837, 853
- Backslash character, 282, 463
- Backtrace print mechanism, 419
- Backup port, 1069
- Balance strategy (Xsan), 152
- Balanced coding method (BCM), 165
- Base set, Forth words, 282
- bash shell, 125
- Basic Security Module (BSM), 134, 449
- BAT registers, 183–185, 405
- bclear\_buf\_thread(), 712
- BeFS, 8
- Behavior values (Mach VM), 889–890
- BeOS, 7–9
- Berkeley FFS (Fast File System), 462, 1391

- Berkeley Packet Filter (BPF), 454
- Berners-Lee, Timothy John, 12
- BezelUI directory, 60
- BFS compared with Spotlight, 1415
- Bidirectional pipes, 1146
- Big real mode, 363
- Big-endian, 225–226
- BigTop.app, 130
- Binary format, 378
- Binding styles, 71
- Bindings Controller, 112
- BIOS (basic input/output system), 264
  - limitations, 362–364
- BIOS Data Area (BDA), 367
- BLAS, 125
- bless, 1589
- bless(8), 270–271
- Block address translation (BAT), 183–185, 405
- Block device, 1349
- Block transfer (BLT), 377*n*
- BlockAllocate(), 1572
- BlockDeallocate(), 1572
- Blocking thread, 718–720
- Blue Box, 28, 29, 33, 538, 591–594
  - system calls, 554
  - trap value, 594
- Blue project, 2
- Blue thread, 591
- Bluetooth, 38, 60
- BMP (Windows Bitmap), 93
- Bonjour, 39*n*
- Boot behavior, 355–356
- Boot chooser, 329
- Boot devices, 326, 367
- Boot Integrity Services (BIS), 363–364
- Boot manager, 370–371
- Boot modes, 339–340
- Boot ROM, 46, 268
  - PowerPC, 269–271
- Boot sequence, 324–328
- Boot Server Discovery Protocol, 347
- Boot services, EFI, 366–367
- Bootable (blessed) volume, 1589
- BootCache history list, 353–354
- BootCache optimization, 353–354
- BootCache play list, 353–354
- BootCacheControl, 354
- BootCache.kext, 1282
- BootCacheOverride, 358
- BootCLUT, 435–437
- boot-device, 326
- boot.efi, 46, 60, 371, 1499, 1589
- bootinfo file, 270, 329
- bootinfo format, 328–329
- Booting, 263, 419–422
  - an alternate kernel, 341–343
  - over a network, 346–349
  - from a software RAID, 344–346
- Bootloader, 46, 60
  - bootinfo file, 270, 329
  - choosing an alternate, 1589
  - See also* boot.efi; BootX
- BOOTP, 463
- bootp(), 463
- Boot-screen information, 643–644
- Bootstrap context, 1065–1066
- Bootstrap Server, 1065–1079
  - bootstrap\_check\_in(), 1069
  - bootstrap\_create\_server(), 1068
  - bootstrap\_info(), 1072–1074
  - bootstrap\_parent(), 1071
  - bootstrap\_status(), 1070
  - bootstrap\_subset(), 1071
- Boot-time daemons, other than launchd, 498
- Boot-time kernel arguments, 355–361
- Boot-time snag keys, 327–328
- boot-uuid, 334
- BootX, 46, 60
  - and AppleRaid, 344
  - BootCLUT, 435–437
  - bootinfo, 328–330
  - cache data, 338
  - and Client IP, 348–349
  - ELF support, 330
  - kernel launch, 388
  - logical memory map, 333
  - operation, 331–340
  - signature, 338
  - structure, 330
  - third-party bootloader, 328*n*
  - and volume header, 1499
- Branch history tables, 211–212
- Branch instructions, 201, 247–256
- Branch prediction, 201, 211–212
- Branch reachability, 248
- Breakpoint trap, 405

**brk()**, 949  
**Brunner, John**, 146  
**BSD**, 50–53
 

- application environment, 44–45, 102–103
- condition variables, 1218
- hardware clock, 466
- proc** structure, 747
- system call return values, 571
- system calls, 554
- unified buffer cache (UBC), 52
- uthread** structure, 721, 748
- vndevice** interface, 463
  - vnode disk driver, 1370–1371
- bsd/ directory**, 502–504
- BSD initialization**, 443–447
  - data structures, 445–451
  - file system, 447–448
  - mounting the root file, 455–464
  - networking subsystem, 450–453
  - shared memory regions, 466–468
- BSD processes**, 746–748
  - fork()** system call, 748–756
  - vfork()** system call, 756–759
- BSD system calls**, 557–576
  - argument munging, 563–566
  - data structures, 558–563
  - kernel processing, 566–572
  - user processing, 572–576
- bsd\_ast()**, 465
- bsd\_bufferinit()**, 446
- bsd\_hardclock()**, 466, 787
- bsd\_init()**, 445–449
- bsdinit\_task()**, 1141
- bsdmake**, 124
- bsd\_utaskbootstrap()**, 465
- B-Tree-based catalog file**, 1388
- B-Trees**, 1481, 1488
- bufqscan\_thread()**, 712
- Bundles and frameworks**, 76
  - bundle structure, 83
  - bundle types, 78–80
  - frameworks, 82–86
  - iTunes application bundle, 77
  - load commands, 87
  - OS X standard frameworks, 85–86
  - prebinding, 86–89
  - printing libraries, 88
  - property list files, 81–82

**BUPOR** (bring-up power-on reset), 393  
**Bus drivers**, 368–369  
**Bus interface unit**, 165  
**Bus master**, 161  
**Bus speed**, 159–162  
**Bus standards**, 160–162  
**Bush, Vannevar**, 1410  
**Byte ordering**, 225–226, 1594

## C

**C++**, 53, 1237  
**C library**, 74–76  
**Cache(s)**

- associativity, 179
- coherency, 180
- direct-mapped, 179
- lines, 175
- management, 403–404
- MERSI, 180
- 970FX, 176–178
- RAS, 180
- set-associative, 179
- store policy, 179
- store-through, 403*n*

**Cacheability**, 858  
**CacheBasher.app**, 130  
**cacheInit()**, 403  
**CAD** (command, address, data), 163  
**Callback-related functions**

- CHUD support, 606–611
- Callbacks, 134, 484–485
- Call-by-name, 231
- Call-by-reference, 231
- Call-by-value, 231
- Callee-save register, 226
- Caller-save register, 226
- CAM (content-addressable memory), 224
- Canon, 10
- Capabilites array, 1575
- Carbon, 103–107
  - APIs, 27, 31, 33, 105–107
  - CFM binaries, 104
  - compared with Cocoa, 106–107
- Carbon Event Manager, 106
- Carbon File Manager, 1407, 1430, 1516
- Carbon Managers, 90

Carbon Multiprocessing Services, 769–771  
 Carbon Process Manager, 767–769  
 Carbon Thread Manager, 772–774  
`Carbon.framework` (Mac OS X), 103  
`CarbonLib` (Mac OS 9), 103  
`CarbonSound`, 105  
`CardBus`, 162  
 Caret character, 274–275  
 Carnegie Mellon University (CMU), 17  
 Case sensitivity  
     HFS+, 1473, 1531–1532  
     HFSX, 39  
     UFS, 1392  
 Cast-conversion, 89  
 Catalog, I/O Kit, 1253  
 Catalog B-Tree, 1566  
 Catalog B-Tree header node, 1518  
 Catalog file, HFS+, 1510–1519  
 Catalog node (`cnode`), 1569  
 Catalog node IDs, 1513  
`catch_exception_raise()`, 1117  
`catch_exception_raise_identity()`, 1117  
`catch_exception_raise_state_identity()`, 1117  
`CCacheServer.app`, 60  
`cddafs` (Compact Disc Digital Audio file system), 1398–1400  
 CD/DVD, 127, 1388–1389  
`CDIS/Installation Log.app`, 496  
`CDIS/instlogd`, 496  
`CDIS/LCA.app`, 496  
`CDIS/preheat.sh`, 496  
`CDSA`, 134  
     plug-ins, 135  
 Cells, Forth, 279  
`cerror()`, 572, 751  
`CFBundle`, 78  
`CFDictionaryAddValue()`, 1359  
 CFM (Code Fragment Manager), 103, 104, 115  
`CFMachPort`, 1203  
`CFMessagePort`, 1203–1205  
`CFRunLoopTimer`, 1209  
`CFSocket`, 1206–1208  
`CGDisplayBaseAddress()`, 1293  
 Change logger  
     Fsevents, 1422–1428  
 Character device, 1349  
`checkForWork()`, 1246  
`chkfac()`, 548  
`chmod()`, 1444  
`chmodx_np()`, 1445  
 Choke handlerr, 553  
 chosen properties, 334–335  
`chroot()`, 929  
`CHUD`, 130, 604  
     callback-related functions, 606–611  
     CPU-related functions, 605–606  
     `kdebug_chudhook()`, 625  
     memory-related functions, 605  
     task-related functions, 605  
     thread-related functions, 605  
 CHUD tools  
     `acid`, 131  
     `amber`, 131  
     `BigTop.app`, 130  
     `CacheBasher.app`, 130  
     `hwprefs`, 1587–1588  
     `MONster.app`, 131  
     `PMC Index.app`, 131  
     `Reggie SE.app`, 131  
     `Saturn.app`, 131  
     `Shark.app`, 131, 1338  
     `simg4`, 131  
     `simg5`, 131  
     `Skidmarks GT.app`, 131  
     `SpindownHD.app`, 131  
`CHUDCall`, 586  
`CIContext`, 97  
`CIFilter`, 97  
 CIFS (Common Internet File System), 1397  
`CIImage`, 97  
`CIKernel`, 98  
`CISampler`, 98  
 Class hierarchy, I/O Kit, 1239–1240  
 Class matching, 1256  
 Classes, I/O Kit, 1243–1246  
 Classic, 114–116  
     environment, 29  
     sharing of OS X resources, 115  
`Classic Startup.app`, 60, 114  
`Classic.app`, 29*n*  
 Class-modeling tool, Xcode, 107  
 Cleartext storage, 1320  
 Client interface, Open Firmware, 299  
 Client IP, and BootX, 348–349  
 Client systems, Xsan, 152

- Clock frequencies, 220
- Clock implementation, 320–321
- Clock services, 522–524
- Clock speeds, 163
  - `clock_config()`, 430, 780
  - Clock-cycle-per-instruction (CPI), 168
  - `clock_init()`, 431
  - Clock-time intervals, 782–784
  - `cloneproc()`, 465, 751
  - `clone_system_shared_regions()`, 929
  - `closeGate()`, 1246
  - Close-on-exec flag, 824
  - Clumps, HFS+, 1481
  - `cluster_io()`, 1364–1366
  - CMOS (Complementary Metal Oxide Semiconductor), 157
  - CMU Mach 2.0, 10
  - `cngetc()`, 469
  - CNID (catalog node ID), 1510, 1551
    - assignments, 1515
    - converting to a Unix pathname, 1515
    - and exchangedata(), 1570
  - cnode, 1569
  - cnputc(), 469
  - Cocoa, 107
    - and Carbon, 33
    - compared with Carbon, 106–107
    - Core Data, 108–113
    - malloc zones, 953
    - and NeXT, 107
    - nib files (NeXT Interface Builder), 108
      - NSTask, 763–764
      - NSThread, 764–766
        - Yellow Box, 33
    - Coherency point, 180
    - Colon character, 463, 1388
    - Color lookup table (CLUT), 302, 437
    - Colors, 302–303
    - ColorSync, 5, 101
    - commandGateHandler(), 1248
    - Command-line editing, 275
    - Command-line support, 126–129
    - Command-line tools, 141
    - Common Criteria Tools package, 143n
    - Common Object File Format, 378
    - Common Security Services Manager (CSSM) API, 135
    - CommonPanels, 105
    - CommonPoint, 4
    - commpage area, 434, 554, 594–601, 922–924
    - `com.osxbook.dummyd.plist` configuration file, 483–484
    - Compare-and-store function, 245–246, 1211
    - Compare-and-swap function, 1211
    - Compatibility Support Module (CSM), 367
    - Compilers
      - GNU C, 124
      - and libraries, 124–125
      - version dependency, 677
    - Compiling dynamic and static libraries, 69–70
    - Compiling the kernel, 676–681
    - `complete()`, 1258
    - Complex Instruction-Set Computer (CISC), 157
    - Complex message, 1034
    - Composite value, 240
    - Compositing, 93–94
    - Compressed kernel, 337
    - Computation value, 805
    - Computer Hardware Understanding Development.
      - See* CHUD; CHUD tools
    - Condition Register (CR), 392
    - Conditional branch, 201
    - `config/`, 502
    - Configuration agents, 142
    - Configurations, 111
    - Console initialization, 416–419
    - Console I/O, Forth words, 286
    - Console operations table, 418
    - Constraint value, 805
    - Constructors, 440
    - Context, 542
      - Context communications area, 660
      - Context switching, 774
      - Context-altering instruction, 537
      - Context-synchronizing, 537
    - Contiguous allocation, 996–997, 1259
    - Continuation functions, 22, 431, 717–726
    - Control events, 143
    - Control flow, Forth words, 286–287
    - Control key character, 274–275
    - Control transfer
      - external hardware interrupts, 532
      - processor traps, 532
      - software traps, 532
      - system calls, 533
    - Control-flow instructions, 200–201

- Conventional PCI, 161  
 Conversion factor, thread processor usage, 796–799  
 Cooked devices, 1349  
 Cooperative multitasking, 771  
 Copland, 5–7  
 Copy and paste, 115  
 Copy objects, 864  
 Copy strategy, 858  
 COPYFILE, 1581  
`copyfile()`, 1557, 1582  
`copyin()`, 908, 909  
 Copy-on-write, 17, 20, 516, 862–864  
`copyout()`, 908, 909  
`CopyProcessName()`, 767  
`copypv()`, 909  
 Core Audio, 100  
 Core Data, 40, 108–113  
 Core file, 63  
 Core Foundation, 89, 106  
   data types, 81  
   dictionary, 1294  
   distributed notifications, 1198–1199  
 Core Foundation IPC, 1197–1210  
   notifications, 1197  
   run loop, 1201–1210  
 Core Graphics, 101  
 Core Image, 40  
   plug-in architecture, 97  
 Core Image and Core Video, 96–99  
 Core Services, 89, 106  
   directory, 60  
 Core Video, 40  
 Cox, Brad, 11  
`cpu_uptime`, 699  
`cpu_doshutdown()`, 551  
`cpu_type()`, 467–468  
 Cracked instructions, 206  
 Crash Reporter daemon, 497, 1323  
`Crash_Reportert.app`, 60–61  
 Crash reports, 61  
 Crash-resistant server example, 1074–1076  
`create_unix_stack()`, 820  
 cron daemon, 475–476  
`curctx`, 538  
`current_act()`, 716  
`current_thread()`, 716  
 Cursor pmap, 866  
`CutTrace()`, 657  
 CutTrace low-level tracing, 655–658  
 Cyberdog, 24  
 Cycle-accurate core simulator, 970FX, 256–261  
 Cylinder-Head-Sector (CHS) addressing model, 1346
- ## D
- D0, 206  
 D1, D2, D3, 206–207  
`DACopyDiskDescription()`, 1355–1357  
 DADisk functions, 1353–1355  
 Daemon  
   agent, 472  
   audit, 620  
   cron, 475–476  
   group membership resolution, 1451  
   launchd, 472–485  
   on-demand launching, 473  
   SSH, 474–475  
   and SystemStarter, 498  
   update, 921  
 DART (Device Address Resolution Table), 157, 1257–1259  
 Darwin, 3, 46  
   benefits, 47–48  
   notify center, 1197  
   packages, 47, 676  
   as subset of OS X, 48  
   32-bit ABI runtime stack, 232  
   xnu package, 381  
 Darwin kernel. *See* xnu kernel  
 DarwinBuild, 681  
 Dashboard, 39–40, 119–120  
 Dashboard Widgets, 40, 78, 119  
 Data, Open Firmware, 297–298  
 Data caching, disabling, 914  
 Data fork, 1557, 1567  
 Data link interface layer (DLIL), 153  
   initializing, 450  
 Data manager (memory manager), 851  
 Data partition, 1350  
 Data stream manipulation instructions, 219  
 Data structure(s)  
   BSD initialization, 445–451  
   BSD system calls, 558–563  
   EFI, 1589–1590

- file descriptors, 1148, 1374
- global profile cache, 944
- HFS+, 1477–1482
- and `hfsdebug`, 1474
- IPC-related, 1042, 1048
- `kauth`, 1450
- `kmod_info`, 1264
- `kqueue`, 1192
- locking, 407, 449–451, 1214, 1218
- low-memory global, 650–655
- Mach, 687, 714–716, 791, 1333
- Mach locks, 421, 1213–1215, 1218
- Mach port, 1044–1045, 1054
- Mach trap table, 579
- Mach VM, 849–855, 865–866, 896
- Mach zone allocator, 985–992
- memory, 410
- and mutexes, 1224
- Open Firmware, 289–298
- pageable memory, 995–997
- panic-related, 1322
- patch tables, 398–400
- per-processor, 390, 652, 778
- and physical memory, 410
- pipe, 449
- private, 428
- `pthread`, 751
- resource allocation, 356
- scalable zone, 955–966
- scheduler, 408, 774, 791
- serialized, 441
- shared region mapping, 434, 466–468
- shuttle and thread, 715
- UNIX system call, 572
- UPL (universal page lists), 840
- virtual machine monitor (VMM), 538, 659–660
- `vnode`, 448, 455, 1374, 1383
- Data-cache-block-touch instruction, 186–187
- Data-modeling tool
  - Xcode, 107
- `dd`, 841–843, 1350
- DDR (Double Data Rate), 157
- Dead names, 1029, 1084–1086
- Dead port, 1048
- Dead-Code Stripping, 123
- `deadfs`, 1400–1401
- `-debug`, 1368
- debug kernel argument, 360–361
- Debugger(s) and debugging
  - and annotated code, 241
  - auditing support, 619–625
  - bundle, 78
  - Forth programs, 288–289
  - kernel arguments, 358–361
  - kernel event tracing (`kdebug`), 625–638
  - kernel panics, 1322–1324
  - logging, 1325–1328
  - Mach VM interfaces, 896
  - memory, 976
  - per-process kernel tracing, 617–619
  - remote, 122, 602–603
  - remote core dumps, 1324–1325
  - and signals, 1144
  - stabs, 1338–1344
  - tools, 129–130
  - tracing, 649–659
  - xnu kernel, 50
- See also* GDB (GNU debugger); KDB (Kernel Debugger); KDP (Kernel Debugging Protocol)
- `debugger_lock`, 419
- Decrementer Register (DEC), 777
- Dedicated register, 226
- Dedicated threads, 710
- `.def` extension, 578
- Default environment (system and `fs_root`), 467–468
- Default init program, 469
- Default pager, 517–518, 853
- Default processor set, 689
- `deferctx`, 538
- Deflate compression, 367–368
- Defragmentation, HFS+, 1558–1561
- Delayed detachment, 450
- Demand paging, 836
- Denning, Peter J., 837
- Dennis, Jack B., 683
- Denormal numbers, 219*n*
- Desktop (Finder), 34
- Destructors, 440
- `detach()`, 1281
- Detached pthreads, 751
- Detaching a disk, 1368, 1370
- Detachment of protocols, 450
- Developer previews, 31–34
- Developer’s Interface Guide (DIG64), 365
- `devfs` (device file system), 1401

Device Driver Kext, 1280–1288  
 Device driver(s), 369, 1233–1235  
   Boot ROM (PowerPC), 270  
   matching, 1255–1257  
 Device interface, Open Firmware, 300  
 Device Interface I/O Kit, 1269–1271  
 Device node naming, 1367  
 Device pager, 518, 854  
 Device special files, 1235  
 Device tree, 289  
   data, 297–298  
   dealias, 291  
   methods, 296–297  
   properties, 292–296  
 Device Tree plane, I/O Kit, 1252  
`device_page_bootstrap()`, 879  
`/dev/kmem`, 841–843, 1597–1598  
`/dev/mem`, 1597–1598  
`dgBootScreen()`, 643  
`dgLRA()`, 644  
`dgpcpy()`, 645–646, 650  
 DHCP, 463  
`dhcpol_parse_packet()`, 463  
 diag, 638  
`diagCall()`, 641  
`diagCommon.h`, 642  
 Diagnostic features, 639  
 Diagnostics system calls, 642–644  
   interface, 641–649  
 Dictionary, Forth, 282–288  
 Digital imaging, 38  
 Direct function calls, 230  
 Direct memory access (DMA), 94, 157  
 Direct-mapped cache, 179  
 Directory services, and security, 140–141  
 Dirty pages, 872  
`DiscRecording.framework`, 127  
 Disk and partitions, listing, 1348  
 Disk arbitration, 1353–1361, 1497  
 Disk Arbitration subsystem API, 1353–1355  
 Disk image, 1320  
   BSD vnode disk driver, 1370–1371  
   `hdiutil`, 1367–1369  
   RAM disks, 1370  
   virtual disks, 1366–1374  
 Disk Images framework, 1574  
 Disk mounting dissenter object, 1357  
 Disk object, 1354  
 Disk supersets, 1362  
 Disk Utility, and security, 139  
 Disk (Xsan), 150  
`diskarbitrationd`, 1353  
`DiskImages.framework`, 127  
`diskimages-helper`, 1368  
 Diskless booting, 347  
 Disks and partitions, 1345–1349  
   Apple partitioning scheme, 1348–1352  
   GUID-based partitioning, 1352  
   PC-style partitioning, 1352  
`diskutil command`, 1349, 1352, 1353  
 Dispatch group, 207  
 Display PostScript, 10  
 Dissenter object, 1357  
`distcc`, 122  
 Distributed builds, 122  
 Distributed notification center, 1197  
 Distributed Objects, 1164–1172  
`ditto`, 1557  
`dkstrategy()`, 1365–1366  
`dladdr()`, 75n, 236  
`dlil_call_delayed_detach_thread()`, 712  
`dlil_input_thread()`, 711  
`dlopen()`, 75  
 DLPAR (Dynamic LPAR), 169–170  
 DMA Address Relocation Table, 157  
 Dock, 33  
 Domain initialization, 451–452  
`domaininit()`, 451  
`doServiceMatch()`, 1255  
 Double-precision floating-point values, 240  
 Drag and drop, 11, 115  
 Draw protocol, 377  
 Drawing images, 322–323  
 Drawing rectangles, 303, 314–317  
 Driver(s)  
   EFI, 367–369  
   I/O Kit, 1242  
   matching, 1255–1257  
   methods, 1287  
   partitions, OS 9, 1350  
   personalities, 1255, 1281–1283  
   xnu kernel, 48  
 Driver side (data transmission), 165  
`drutil`, 127  
`dscl command`, 141  
 DSN, 17  
 DSP, 207  
 DSP, WRT, GCT, MAP, 207–208

dssall, 403  
 Dual-processor architecture, 158  
 Dual-processor bus, 164  
 dummyd (trivial echo server), 479–482  
 DummyDriver I/O Kit driver, 1284–1286  
 DummySysctl kernel extension, 1273  
 DVD player, 38  
 dyld (dynamic link editor), 62  
     environment variables, 933  
     interposing, 73–74  
     loading a non-split-segment file, 934  
     loading a split-segment file, 935  
         Mac OS X 10.4, 932–933  
 .dylib, 63  
 Dynamic branch prediction, 212  
 Dynamic computation modes, 222  
 Dynamic libraries, compiling, 69  
 Dynamic linker, 64, 817–818, 932–935  
 Dynamic linking, 67  
     binding styles, 71  
     dyld interposing, 73–74  
     two-level namespaces, 71  
     weakly linked symbols, 72  
     ZeroLink, 123  
 Dynamic pager program, 918–921  
 Dynamic shared library, compiling, 69–71  
 Dynamically allocated memory, 887  
 dynamic\_pager, 517–518, 918

## E

EEPROM, 269  
 Effective address (EA), 181  
 EFI (Extensible Firmware Interface), 46, 362  
     architecture, 368  
     benefits, 378–379  
     binary format, 378  
     boot manager, 370–371  
     boot services, 366–367  
     driver types, 368–369  
     drivers, 367–369  
     GUID partition table, 374  
     and preboot software, 364  
     protocol, 369  
     runtime services, 367  
     services, 366–367  
     shell, 370–374, 372–374  
     Universal Graphics Adapter (UGA), 376–377

EFI Byte Code (EBC), 367, 377–378  
 EFI Byte Code (EBC) Virtual Machine, 377–378  
 EFI NVRAM, 369–370, 1589–1590  
 EFI Runtime kernel extension, 1589–1590  
 EFI System Partition (ESP), 376  
 EFI System Table, 1590  
 eieio, 1212  
 EINVAL error, 759  
 eip register, 65  
 Elastic I/O interconnect, 164–166  
 ELF support, 330  
 Embedded C++, 1237  
     and I/O Kit, 1237–1238  
 EMM external memory management, 517  
 Emulation registers, 197  
 Emulators, 278  
 enable\_bluebox(), 591  
 Encoding names, 1532–1534  
 Encrypted storage, 1321  
 Encrypted virtual memory, 134, 139  
 Encrypted volumes, 1310–1321  
 Encryption, swap files, 920  
 Encryption keys, 135  
 Entities, 111  
 Entity inheritance, 111  
 Entity-relationship (ER) diagram, 111  
 Entropy collection, 637–638  
 Epilogue, 233  
 EPS (Encapsulated PostScript), 93  
 ERATs, 183  
 errno, 751  
 Error correction code (ECC), 159  
 Error-correcting code, 180*n*  
 escc (serial port), 417*n*  
 escc-legacy (serial port), 417*n*  
 /etc/rc, 486–487  
 ether\_family\_init(), 454–455  
 Ethernet connections, 276–278  
 Ethernet interface family, 454–455  
 Ethernet interface MAC address, 1309–1310  
 Ethernet port aggregation, 455  
 Event tracing (kdebug), 625–638  
 Exception(s), 201, 518  
     counters, 654–655  
     and exception vectors, 383–385  
     filtering, 546–547  
     Mach, 1112–1129  
     ports, 515, 1049, 1112  
     PowerPC, 383–385

- Exception(s), (*continued*)  
 processing, 543–553  
 save areas, 539–542  
 vectors, 535, 536
- Exception-handling chain, 1118
- Exception-handling registers, 536–537
- Exception-handling routine, 385
- `exception_triage()`, 1120
- EXCHANGEDATA, 1581
- `exchangedata()`, 1570, 1582
- `exec()`, 116, 684
- `exec_copyout_strings()`, 822
- `exec_handle_suid()`, 819
- `exec_mach_imgact()`, 757, 815, 819, 822, 824
- `exec_shell_imgact()`, 827
- `execsigs()`, 824
- Execution environments, 686–687
- `execve()`, 469, 757
- `execve()` system call, 62, 812–827  
   Task Working Set (TWS), 947
- Existence map, 921
- `exit()`, 572
- Exposé, 39, 119
- Extended file system (HFS Plus), 25
- Extended memory addresses, 363
- Extended partitions, 363, 1352
- EXTENDED\_SECURITY, 1581
- Extending a volume, 1572
- Extensible Firmware Interface. *See* EFI
- Extensions, 59
- Extent descriptor data structure, 1480
- Extents, 1480
- Extents overflow file, 1519–1524
- eXternal booter, 344
- External hardware interrupts, 532
- External memory management, 427*n*
- External memory manager, 854
- F**
- `facctx`, 538
- Facility Status UFT, 589–590
- Factorials, 241–244
- Families, I/O Kit, 1239–1241
- Family Superclasses, 1239–1240
- Fan control unit (FCU), 221, 1308
- Fans (temperature management), 220–222
- Fast IPC space, 1044
- Fast Logout and Autosave, 40
- Fast Path, 171
- Fast traps, 590
- Fast user switching, 39, 119
- Fastpath call, 554
- Fastpath system calls, 590
- Fat binaries, 12*n*, 66–68
- Fat Darwin kernel, 66
- Fat (Universal) binaries, 825
- Fax capability, 39
- `fcntl()`, 1231, 1430
- FCode, 273
- fd (file descriptor), 1374–1375
- fdesc file system, 1402
- `fdexec()`, 824
- fdisk, 1352
- Feldman, Jerry, 16
- F\_FULLFSYNC, 1539–1540
- Fibre Channel Multipathing, 153
- Fictitious extensions (aka pseudo-extensions), 441
- Fifo file system, 1147
- FIFO with Second Chance, 871
- fifofs (named pipes), 1403
- Fifos, 1147
- File descriptor passing, 1148–1155
- File descriptors, 1374–1375
- File forks, HFS+, 1480
- File format, BootX, 328–330
- File sharing, 1395
- File system APIs, Mac OS X, 1387
- File system initialization, 447–448
- File system interfaces, 1581
- File system types, 1386–1409  
   AFP (Apple Filing Protocol), 1393–1395  
   cddafs, 1398–1400  
   deadfs, 1400–1401  
   devfs, 1401  
   fdesc, 1402  
   fifofs, 1403  
   FTP, 1395–1396  
   HFS Plus and HFS, 1388  
   ISO 9660, 1388–1389  
   MS-DOS, 1390  
   NFS, 1396–1397  
   NTFS, 1391  
   SMB/CIFS, 1397  
   specfs, 1403

- synthfs, 1404
- UDF (Universal Disk Format), 1391
- UFS, 1391–1393
- union, 1405–1407
- vofls, 1407–1409
- WebDAV, 1397–1399
- File system user-space view, 57–61
- File systems
  - access control lists (ACL), 1441–1445
  - disk arbitration, 1353–1361
  - disk devices, 1362–1366
  - disk images, 1366–1367
  - disks and partitions, 1345–1349
  - file system types, 1386–1409
  - files and file descriptors, 1374–1375
  - kauth authorization subsystem, 1445–1469
  - Spotlight, 1409–1441
  - VFS layer, 1376–1386
- Files and file descriptors, 1374–1375
- `filesec_init()`, 1445
- Filesystems, 60
- filetype, 63
- FileVault, 39, 139
- Fill strategy (Xsan), 152
- FindByContent, 101
- Finder (Desktop), 34
- `finderInfo` array, 1498
- Finding files, mdfind, 128
- Fine-grained kernel event tracing (kdebug), 625–638
  - kdebug entropy collection, 637–638
  - kdebug tracing, 625–637
- FireWire plane, I/O Kit, 1252
- Firewire target disk mode, 328
- FireWire-based debugging, 601–603
- Firmware, 46, 263–271
  - boot sequence, 324–328
  - BootCache optimization, 353–354
  - booting an alternate kernel, 341–343
  - booting from a software RAID device, 344–346
  - booting over a network, 346–349
  - boot-time kernel arguments, 355–361
  - BootX, 328–340
  - call interface, 640–641
  - device tree, 289–298
  - Extensible Firmware Interface (EFI), 362–379
    - firmware boot sequence, 324–328
    - Forth programming language, 279–289
    - launching the kernel, 352–353
  - Open Firmware, 272–278
  - Open Firmware interfaces, 298–300
  - Open Firmware programming, 300–324
  - power-on reset (POR), 271–272
  - power-on self-test (POST), 264
  - security, 349–352
  - variables, 1294–1296
- `FirmwareCall()`, 640–641
- Fix and Continue, 123
- `fixBufferUserRead()`, 1313–1315
- `fixBufferUserWrite()`, 1313–1318
- Fixed-point instructions, 198
- `FixupResourceForks`, 1557
- Flash invalidation, L1 D-cache, 403
- Floating-point initialization value, 401
- Floating-point instructions, 199
- Floating-point parameters, 239–240
- Floating-point register (FPR), 401
- Floating-point-estimate instructions, 199
- FLOCK, 1581
- `flock()`, 1231
- Focus-follows-mouse, 103
- Font demo, 318–320
- Fonts, 115, 317–320
- `foreground-color`, 302
- `fork()`, 52, 116, 748–756
  - early UNIX, 684
  - and Mach ports, 755
- `fork1()`, 751–753
- `forkproc()`, 751–753
- Formal parameters, 231
- Forth code, 279
  - for animation, 306–312
  - cells, 279
  - for core logic, 312
  - debugging, 288–289
  - dictionary, 282–288
  - shell, 274–275
  - stacks, 279
  - words, 281–282
- Foundation notifications, 1181
- Foundation.framework, 107–108
- 4.2BSD, 1131
- 4.3BSD, 10, 19, 1131
- 4.4BSD, 153
- Fragile base-class, 55
- Fragment, 104, 1479
- Fragmentation, 1519, 1520

- Fragmented files, 1558–1562  
 Framebuffer, 1288–1292  
 Framebuffer memory, 1293  
 Framework scaling mode, 118  
 Frameworks, 4–5, 59–60
  - OS X standard, 85
  - private, 84–85
  - `System.framework`, 87–88
  - See also* Bundles and frameworks`free()`, 1280  
 Free list (save area), 414, 540  
 Free pool (save area), 414  
 Free queue, 871  
 Free Software Foundation (FSF), 47*n*  
 Free space, 1507  
 Free space partition, 1350  
 FreeBSD
  - NFS, 1396
  - UFS2, 1392
  - virtual memory, 21
 Freezing a volume, 1571  
 Fsevents, 51
  - argument types, 1418
  - change logger, 1422–1428
  - event generation, 1421
  - implementation, 1419
  - supported event types, 1416`fs_usage`, 129  
 FTP file system, 1395–1396  
`function()`, 247  
 Function calls, 228–230
  - stack trace, 237–238
 Function rerouting, 246
  - constructing branch instructions, 247–256
  - instruction patching, 247
 Function (use of term), 231  
`function_new()`, 247  
`function_stub()`, 248  
`funnel_alloc()`, 445  
 Funnels, 52, 1223–1229
- G**
- `g++`, 62  
 Gassée, Jean-Louis, 7–9  
 GB (metric definition), 1347  
`gcc`, 62
- GCC 4.0.0, 66*n*  
 GCT (global competition table), 207–208  
 GD, 207  
 GD, DSP, WRT, GCT, MAP, 207–208  
 GDB (GNU debugger), 50*n*, 60
  - FPR, 401
  - script, crash reports, 61
  - and stabs, 1338
  - using for debugging, 1328–1331
  - Xcode, 122, 601–603
 General I/O Kit Classes, 1239–1240  
 General OS Classes, 1239–1240  
 Gershwin, 6–7  
`getattrlist()`, 1570, 1575–1582  
`getchar()`, 469  
`get-colors`, 302  
`getcontext()`, 1138  
`getdirententriesattr()`, 1571  
`getfwpather`, 341  
`get_ip_parameters()`, 463  
`getmntinfo()`, 1348*n*  
`getOrientationUC()`, 1299–1302  
`getpid()`, 575  
`GetProcessForPID()`, 767  
`gettimeofday()`, 594  
`get_user_regs()`, 466  
`getWorkLoop()`, 1246  
 Giant mutex, 52*n*  
 GiB (definition), 1347  
 GIDs (group identifiers), 133  
 GIF (Graphics Interchange Format), 93  
 Gigabyte (definitions), 1347  
 Giuse, Dario, 20  
 Global data structure, 1374  
 Global history vector, 211  
 Global profile cache, 944  
 Global variables, GUID, 369–370  
 Global volume lock, 1571  
 Globe data, 336  
 GNU assembler, 193  
 GNU C compiler, 124, 1338  
 GNU debugger. *See* GDB (GNU debugger)  
 GNU-Darwin, 35  
`gnumake`, 124  
 GPROF, 537  
`gprof`, 612–617  
`gpt`, 376, 1352–1353, 1373  
 GPT-partitioned disk, 374–376

GPU (graphics processing unit), 94  
`grade_binary()`, 816  
 Graphics adapter, 376–377  
 Graphics Address Remapping Table (GART), 157  
 Graphics and multimedia APIs, 91–101  
 Graphics and multimedia architecture, 91  
 Group membership resolution daemon, 1451  
 GUID, 369–370, 1373  
 GUID Partition Table (GPT), 374, 1352–1353  
 x86-based Macintosh computers, 1590–1591

## H

Hamming code, 180–181  
 Handler thread, 518, 1115  
 Handlers, 201, 750  
 Handoff scheduling, 708, 791, 1058  
 Hangup signal (HUP), 489  
 Hard disk, 1345n  
 Hard limits, 685  
 Hard links, 1547–1551  
 Hard reset exception, 271  
 Hardware abstraction layer (HAL), 5, 7, 44–45, 1236  
 Hardware exception counters, 654–655  
 Hardware interrupt offset register (HIOR), 271  
 Hardware interrupts, 549–551  
 Hardware streams, 186  
 Hardware-Implementation-Dependent (HID) registers, 194  
 Harvard Architecture, 178  
 Hash tables, 411–413, 869  
 hdutil, 127, 1320, 1367–1369  
   sparse disk images, 1575  
 hdix (Disk Image Controller), 463  
 Header bitmap, 957  
 heap, 129  
 Heat sensors, 221–222, 1305–1308  
 Help (Carbon subframework), 105  
`hertz_tick()`, 780, 785, 787  
 HFS, 462, 1386  
 HFS+, 462, 1386  
   aliases, 1554–1556  
   allocation blocks, 1477  
   allocation file, 1505–1509  
   analysis tools, 1474–1477  
   attributes file, 1524–1530  
   B-Trees, 1481–1491

case sensitivity, 1531–1532  
 catalog file, 1510–1519  
 clumps, 1481  
 comparisons, 1575–1582  
 extents, 1480  
 extents overflow file, 1519–1524  
 file forks, 1480  
 filename encodings, 1532–1534  
 freezing and thawing a volume, 1571  
 hard links, 1547–1551  
 HFS wrapper, 1501–1505  
 journaling, 1538–1546  
 metadata zone layout, 1562  
 and NTFS comparison, 1583  
 optimizations, 1558–1570  
 permissions, 1534–1538  
 quotas, 1546–1547  
 reserved areas, 1493  
 resizing a volume, 1572  
 resource forks, 1556–1558  
 sparse disk images, 1574–1575  
 special files, 1492, 1505–1530  
 special system calls, 1570–1571  
 startup file, 1530  
 symbolic links, 1553–1554  
 Technical Note TN1150, 1477  
 unlinking open files, 1552–1553  
 volume header, 1493–1501  
 volume notifications, 1572  
 volume structure, 1491–1493  
 volumes, 1515  
`hfsdebug`, 1474–1475, 1558–1560, 1565  
`hfs_extendefs()`, 1572  
`hfs_generate_volume_notifications()`, 1574  
`hfs_hotfilesync()`, 1567  
`hfs_recording_init()`, 1566  
`hfs_recording_start()`, 1568  
`hfs_relocate()`, 1558  
`hfs_truncatefs()`, 1572  
 HFSX, 39  
 HID (hardware-implementation-dependent), 403  
 HID (human interface device), 1241  
 High-level architecture, 45  
 High-level processor initialization, 405–421  
   after virtual memory, 415–421  
   low-level VM initialization, 409–415  
   before virtual memory, 407–409

- High-level VM subsystem, 423–428  
 High-water mark, 918  
 HIServices, 101  
 History list, BootCache, 353–354  
 Hit rate, caching, 353–354  
 HIToolbox, 106  
 HIView, 106  
 Host information, Mach, 519–522  
 Host ports, 1050  
 Host statistics, Mach APIs, 526–529  
`host_get_boot_info()`, 520  
`host_info()`, 519–520  
`host_kernel_version()`, 520  
`host_notify_init()`, 1060  
`host_page_size()`, 520  
`host_processors()`, 693, 1050  
`host_processor_set_priv()`, 690  
`host_processor_sets()`, 690  
`host_request_notification()`, 1060  
`host_statistics()`, 526–529, 892  
`host_zone_info()`, 985  
 Hot File area, 1562  
 Hot File Clustering (HFC), 1562–1564, 1566–1570  
 Hot File thread, 1565  
 Hot Files B-Tree, 1565–1566  
`hotfiles_adopt()`, 1570  
`hotfiles_evict()`, 1570–1571  
 HotSpot Java Virtual Machine (JVM), 113  
 Howarth, David J., 835–836  
 HPC (high-performance computing) applications, 183  
 HTML rendering, 106  
 Huge allocations, 966  
 Human-interface services, 101  
 Hupp, Jon A., 146  
`hwprefs`, 178, 1587  
 Hybrid drivers, 369  
 HyperTransport (HT), 157–164  
 HyperTransport Technology Consortium, 162
- |
- IA-64 architecture, 365  
 IA-32 computers, and EFI, 367  
 IA-32 real mode, 362  
 IBM  
   Jikes (Java compiler), 113n  
   S/390 G5, 157  
   Taligent, 4–5
- iCal, Spotlight search results, 1440–1441  
 iChat, 38–40  
`.icns`, 118  
 Icon Browser application, 118  
 Icon Composer, 118  
 Icon status indication, 119  
 IDE (integrated device electronics), 298  
 iDisk, 38  
 Idle kernel thread, 432–435  
`idle_thread()`, 711  
 IEEE, 273  
 IEEE 802 address, 375  
 Ignore reset, 394  
`ihandler()`, 549–550  
 Illegal instructions, 257, 1595  
 iMac, 3  
 Image activator table, 815  
 Image formats, 93  
 Image processing, 98  
 ImageCapture, 106  
 iMovie 2, 37  
 Inactive queue, 871  
 Indirect function calls, 228–229  
 Indirect system call, 558  
`Info.plist`, 81, 84  
 Information Access Toolkit (V-Twin), 90  
 Information property list, 81  
 Infrastructure initialization, 775–789  
 Inheritance, entity, 111  
 Inheritance, memory, 884  
`init()`, 1280  
`init_domain()`, 451  
 Initialization  
   BSD, 443–469  
   console, 416–419  
   custom routine, 70–71  
   data link interface layer (DLIL), 450  
   domain and protocol, 451–452  
   file system, 447–448  
   high-level processor, 405–421  
   high-level vm subsystem, 423–428  
   infrastructure, 775–789  
   I/O Kit, 435–443, 1253–1255  
   IPC, 430  
   `kprintf()`, 416–417  
   Mach subsystem, 421–432  
   networking subsystem, 450–453  
   page frames, 425–426  
   Pmap, 411–415

- Processor-specific, 400–401
- protocol, 451–453
- Scheduler, 421–423
  - Virtual memory, 409–415, 430
- `initialize()`, 1254
- `inittoaddr()`, 455
- Ink (handwriting recognition), 106
- Inkwell, 38
- Inline assembly, 193
- Inline data, message, 516
- Input handler, 450
- Installation log viewer, 496
- Installation progress bar, 498
- Installation startup, 496–499
- `install_name_tool`, 129
- Instruction Cache Access (ICA), 206
- Instruction Decode Unit, 185
- Instruction Fetch Address Register (IFAR), 206
- Instruction Fetch Unit, 185, 210
- Instruction Match CAM (IMC), 224
- Instruction patching, 247, 249–256
- Instruction pipeline, 970FX, 205
- Instruction Prefetch Queue, 185
- Instruction set, 197–202
- Integrated chip (IC), 157
- Intel Boot Initiative (IBI), 366
- Intel Core Duo, 156, 173–174
- Intel *Wired for Management* specification, 375
- Interconnect mechanism, 161–162
- Interface Builder, 108
- Interfaces, file system, 1581
- Interfaces, Open Firmware, 298–300
- Internal objects, 853
- Internet, 24–25
- Internet Systems Consortium, Inc. (ISC), 35
- Interposition, 73–74, 1087
- Interpreter(s), 125–129
  - AppleScript, 125–126
  - Automator, 126
  - command-line support, 126–129
  - scripts, 825–827
- Interrupt, 201
- `interrupt()`, 550–551
- Interrupt line, 549
- Interrupt model, 718
- Interrupt vector relocation, 271
- `interruptEventSource()`, 1248
- `interruptOccurred()`, 1248
- In-use bitmap, 957
- I/O addresses, 1257
- I/O Kit, 53–54, 1235
  - class hierarchy, 1239–1240
  - classes, 1243–1246
  - Device Interface, 1269–1271
  - disk device and partitions, 1362–1363
  - driver matching, 1255–1257
  - driver methods, 1287
  - driver personality, 1281–1283
  - drivers, 1242
  - Embedded C++, 1237–1238
  - families, 1239–1241
  - initialization, 435–443, 1253–1255
  - I/O Catalog, 1253
  - I/O Registry, 1251–1252
  - memory allocation, 995–997
  - nubs, 1242–1243
  - planes, 1252
  - traps, 554, 584
  - Work Loop, 1246–1251
- I/O Kit Device Driver Kext, 1280–1288
- I/O Media Filter, 1362–1364
- I/O protocol (UGA), 377
- I/O Registry, 54
  - `IOCatalogue`, 1243
  - `IOCreateThread()`, 711–713
  - `ioctl` (I/O control), 1346
  - `IODiskMatching()`, 459–460
  - `IOExternalMethod`, 1300
  - `IOFindBSDRoot()`, 457–461
  - `IOFindMatchingChild()`, 460
  - `IOKernelDebugger`, 1245
  - `iokit/`, 502, 504
  - `IOKitResetTime()`, 447
  - `IOLog()`, 1325
  - `IOMalloc()`, 995–997
  - `IOMallocContiguous()`, 1259
  - `IONetworkInterface`, 153
  - `IONetworkNamePrefixMatching()`, 459–460
  - `IOPanicPlatform`, 1253
  - `IOPlatformExpertDevice`, 1598
  - iops (internal operations), 206
  - `IORecursiveLockLock()`, 1247
  - `IORecursiveLockUnlock()`, 1247
  - `ioreg` command-line utility, 269, 1252, 1338
  - `IORегистernetworkinterface()`, 460
  - `IORegistry`, 442–443
  - `IORegistryEntry`, 1243
  - `IORegistryExplorer.app`, 1252

- I**  
 IORRegistryIterator, 1243  
 IOService, 1243  
     connection handle, 1302  
 iostat, 1338  
 IOWorkLoop class, 1246–1248  
 IPC (interprocess communication), 16, 1021–1025  
     advisory-mode file locking, 1230–1232  
     Apple Events, 1172–1181  
     bootstrap servers, 1066–1079  
     BSD condition variables, 1218  
     CFMachPort, 1203  
     CFMessagePort, 1203  
     CFSocket, 1206–1208  
     client-server example, 1080–1084  
     Core Foundation, 1197–1210  
     dead names, 1084  
     Distributed Objects, 1164–1172  
     entry splay tree, 1043–1044  
     entry table, 1043  
     exceptions, 1112–1129  
     file descriptors, 1148–1155  
     Foundation notifications, 1181–1184  
     funnels, 1223–1229  
     initialization, 429  
     interposition, 1087–1090  
     I/O kit, 1223  
     kernel event notification, 1191  
     lock sets, 1218  
     low-level locking, 1213–1217  
     Mach IPC messages, 1031–1040  
     Mach ports, 514–516, 1027–1031, 1044–1048  
     messages, 516, 1031–1040  
     messaging, 1057–1060  
     MIG, 1094–1111  
     msg traps, 578  
     name servers, 1068–1079  
     named pipes (fifos), 1147  
     notifications, 1181–1197  
     notify(3) API, 1184–1186  
     out-of-line memory, 1090–1092  
     pipes, 1145–1146  
     port allocation, 1052–1057  
     port rights, 1093–1094  
     port sets, 1085–1087  
     POSIX IPC, 1156–1164  
     Pthreads synchronization interfaces, 1222  
     ptrace() system call, 1144–1145  
     run loop, 1201–1210  
     semaphores, 1219–1222  
     signals, 1129–1145  
     spaces, 1041–1044  
     SPLs, 1230  
     subsystem initialization, 1060–1062  
     synchronization, 1210–1232  
     tasks, 1048–1051  
     threads, 1051–1052  
     XSI IPC, 1155–1156  
 IPC-related data structures associated with a Mach task, 1048  
 iPhoto, 39  
 iPod driver, 1283  
 ISO 9660 file system, 1388–1389  
 ISS, 208  
 issignal(), 1141  
 iSync, 38  
 isync, 1212  
 iTunes, 37  
 iTunes application bundle, 76–77
- J**
- Jaguar, 36, 38–39  
 Java, 113–114  
     Abstract Windowing Toolkit, 102*n*  
     and Cocoa, 107  
 java, 113  
 Java 2 Standard Edition (J2SE), 37  
 Java archives (jar files), 113  
 Java Development Kit (JDK), 33, 113  
 Java Native Interface (JNI), 113  
 Java threads, 762–763  
 Java Virtual Machine framework, 113  
 javac, 113  
 JavaVM.framework, 113  
 JBOD (Just a Bunch of Disks), 151  
 Jikes (Java compiler), 113*n*  
 Jobs, Steve, 9–10  
 Journal info block, 1541  
 Journaling, HFS+, 1538–1546  
 journal\_init(), 448  
 JPEG (Joint Photographic Experts Group), 93  
 Just-in-time (JIT) compiler, 33, 113  
 Just-in-time (lazy) binding, 71  
 JVM, 762

**K**

- `kalloc()`, 992–993, 995
- `kalloc_init()`, 879
- `kau_init()`, 449, 619
- Kauth, 134
  - Kauth authorization subsystem, 1445–1446
    - activity monitor, 1453–1469
    - implementation, 1450–1453
    - listeners and authorization, 1449
    - scopes and actions, 1446–1448
    - Vnode-level activity monitor, 1453–1469
  - kauth framework, 51
  - `kauth_authorize_action()`, 814
  - `kauth_init()`, 445
- KDB (Kernel Debugger), 50*n*, 603–604
  - commands, 1332–1336
  - show all commands, 1336
  - show commands, 1336
  - and stabs, 1338
- kdebug, 625–638
- KDP (Kernel Debugging Protocol), 50*n*, 602–603, 1245, 1324, 1329
- kdmpd, 1324
- Kerberos, 37
- Kernel, 48–57
  - address space, 907–911, 1594–1595
  - auditing support, 619–625
  - bootstrap thread, 431–432
  - CHUD support, 603–612
  - debugging support, 601–658
    - `diag`, 638
    - event queues, 484–485
    - functions, 382
    - funnel, 445
    - `kdebug`, 625–638
    - kext caches, 59, 335–338
    - logging, 416
    - Mach-O load commands, 337
    - mode, 531
    - modules (kmods), 1260–1266
    - notes, 1192
    - panic, 1322–1324
    - port, 1049
    - process tracing, 129, 617
    - save areas, 538
    - stack pointer, 538
    - subsystems, 419–422
    - symbols, 385–387
  - task, 513
  - tracing (`ktrace(2)` and `kdump`), 129, 617–619
  - traps, 1121
  - version string, 650
  - See also* xnu kernel
- Kernel arguments, 355–361
  - for boot behavior, 355–356
  - for locking behavior, 357
  - for properties and debugging, 358–361
  - for resource allocation, 356
  - for root devices, 357
  - for scheduling behavior, 357–358
- Kernel Debug Kit, 1329
- Kernel Event Agent, 1572
- Kernel extension(s), 57, 79, 1259, 1296
  - auto loading, 1267–1269
  - bundle, 1260
  - caches, 335
  - creating, 1271–1288
  - generic, 1271–1279
  - managing, 1266–1267
  - static linking, 69
  - Universal, 1261
- Kernel memory, 841–845
  - allocator, 428
  - `dd` and `/dev/kmem`, 841–843
  - `kvm(3)` interface, 843–845
- Kernel profiling (`kgmon` and `gprof`), 612
  - Mach task and thread sampling, 616–617
  - per-process profiling (`profil(2)`), 614–616
- Kernel programming interfaces (KPIs), 51*n*, 153–154, 1235, 1466
- Kernel startup, 382–388
  - BSD initialization, 443–469
  - first thread, 432–435
  - high-level processor initialization, 405–421
  - I/O kit initialization, 435–443
  - launching the user-space program, 469–470
  - low-level processor initialization, 388–405
    - Mach subsystem initialization, 421–432
    - slave processors, 470–472
- Kernel threads, Mach, 710–712
- Kernel User Notification Center (KUNC), 508
- `kernel_bootstrap()`, 431–432
- `kernel_bootstrap_thread()`, 711
- `kernel_debug()`, 638
- `kernel_memory_allocate()`, 983
- Kernel-space security, 133–134
- `kernel_thread()`, 712–713

- k**  
 kernel\_trap(), 581  
 kevent(), 1192–1196  
 Kext, 1259–1269  
 .kext, 79  
 Kext management, 1268  
 Kext repository cache, 59*n*, 335–338  
 kextcache, 348  
 kextd, 55–56, 1266–1267  
 kextstat command, 49  
 Key combinations  
     command-line editing (Open Firmware), 275  
     for nvedit command-line editing, 327  
     for Open Firmware startup, 274  
 Keychain, 135, 140  
 Keychain (OS 9), 27  
 Keynote file, 79  
 kgmon, 612–617  
 kHasBundle, 78  
 Kilburn, Tom, 835–836  
 kill(), 1129, 1138  
 kmem\_alloc, 983–985  
 kmeminit(), 445  
 kmem\_init(), 878, 879  
 kmmod\_dump\_log(), 1326  
 kmmod\_info, 1254, 1264, 1266  
 kmmod\_init(), 431  
 knote\_init(), 449  
 kprintf() initialization, 416–417, 604  
 kqueue, 51, 484–485, 1572–1574  
     filters, 1193  
 kqueue(2), 1191  
 ksh shell, 125  
 ktrace(), 129, 617–619  
 K2 I/O device controller, 159  
 kudos, 287  
 kvm(3) interface, 843–845  
 kvm\_getargv(), 843–844  
 kvm\_getenvv(), 843–844  
 kvm\_getprocs(), 843–844  
 kvm\_nlist(), 843–844  
 kvm\_read(), 843–844  
 kvm\_write(), 843–844
- L**  
 L1 and L2 caches, 176–178  
 L2 cache, flushing, 404
- L1 D-cache, 403  
 LangAnalysis, 101  
 Language Chooser Application, 496  
 Languages  
     Forth, 279–289  
     scripting, 125  
     supported in Xcode, 124  
 LAPACK, 125  
 Large allocations, 964–966  
 Large pages, 184  
 lastcomm command, 134  
 Last-in first-out (LIFO), 280–281  
 latency, 1338  
 Launch Services, 116  
     framework, 828–833  
     registration database, 829–830  
 LaunchAgents, 60  
 LaunchApplication(), 767  
 LaunchCFMApp, 104  
 launchd, 60, 472–485  
     -compliant daemons, 478  
     daemon configuration, 473–476  
     daemon creation, 476–484  
     debug messages, 1077–1078  
     operation, 484–485  
     session launching, 488–490  
 LaunchDaemons, 60  
 Launching applications, 828  
     mapping entities to handlers, 828–830  
     Uniform Type Identifiers (UTI), 830–833  
 Launching the kernel, 352–353  
 LaunchServices, 101  
 Lazy binding, 71  
 LBA (logical block address), 374  
 LCA, 498  
     LC\_LOAD\_DYLINKER, 817–818  
     LC\_PREBOUND\_DYLIB load command, 941  
     LC\_SEGMENT, 817–818  
     LC\_SEGMENT\_64, 817–818  
     LC\_THREAD, 817–818  
     LC\_UNIXTHREAD, 817–818  
 ld (static link editor), 62  
 LDAP (Lightweight Directory Access Protocol), 140  
 LDT (Lightning Data Transport), 162  
 Leaf function, 234  
 leaks, 129  
 ledger\_init(), 431

libc, 74  
 libdbm, 74  
 libdl, 74–75  
 libdyldapis, 75  
 libinfo, 74  
 libkern/, 502  
     classes and routines, 505–506  
     C++ environment, 1000–1001  
     directory, 505  
     library, 54–55  
 libkeymgr, 75  
 libkmod.a, 1265  
 libkmodc++.a, 1263  
 libkvm, 74  
 liblaunch, 75  
 libm, 75  
 libmacho, 75  
 libnotify, 75  
 libpoll, 75  
 libpthread, 75  
 Library Directory, 59–61  
 libpcsvs, 75  
 libsa/, 502, 506–507  
 libsa library, 55–56  
 libstreams, 75  
 libSystem, 74  
 libtool, 63  
 libunc, 75  
 Link Aggregation, 455  
 Link register (LR), 248  
 Link widths, 163  
 Linkage area, 231–232  
 Linking, 67  
     binding styles, 71  
     dyld interposing, 73–74  
     two-level namespaces, 71  
     weakly linked symbols, 72  
     ZeroLink, 123  
 Linux, 23, 845  
     and Xgrid, 147n  
 lipo, 66, 129–130  
 Liquid cooling, 220–222  
 listDisplays(), 1292–1294  
 Listeners (callbacks), 134  
 Listeners (kauth), 1449, 1457  
 Little-endian, 226  
 Load commands, 337, 817–818  
     listing, 65  
 Load Miss Queue (LMQ), 185  
 Load Reorder Queue (LRQ), 185  
 load\_context(), 784  
 load\_machfile(), 816–818  
 load\_shared\_file(), 927, 930  
 Load/Store Unit (LSU), 185, 186  
 Load-time binding, 71  
 Local busses, 160  
 Local domain, 58  
 Local file systems, 1386  
 Local lock implementation, 1232  
 Local notification center, 1197  
 Lock attributes, 1217  
 Lock groups, 1217  
 lockf(), 1231  
 Locking behavior  
     kernel arguments, 357  
 Locking data structures, 407, 449, 450, 451, 1214, 1218  
 Log buffer daemon, 496–497  
 Log messages, 1327–1328  
 Logging  
     audit, 449  
     debugging, 1325–1328  
     kernel, 416  
     in Mac OS X, 1326  
     malloc operations, 976–978  
 Logical Block Addressing (LBA), 1346  
 Logical partitions, 363  
 Logical unit number (LUN), 150–151  
 Logical-to-real address, 644–645  
 Login  
     hook, 491–493  
     multiuser startup, 490–493  
 loginwindow, 488, 490–495  
     system restart, 494  
     system shutdown, 494  
     user logout, 494  
 loginwindow.app, 60  
 Lookup function, 870  
 lookupd, 132  
 Loopback interface, 450, 454  
 Love, Tom, 11  
 Low tracing, 655–658  
 Low-level diagnostics, 638–649  
     firmware call interface, 640–641  
     system call interface, 641–649  
 Low-level kernel tracing, 649–659

Low-level locking, 1213–1217  
 lock groups, 1217  
 mutexes, 1215  
 read/write locks, 1216  
 spinlocks, 1213  
 Low-level processor initialization, 388–405  
 memory patching, 398–400  
 processor types, 395–398  
 processor-specific initialization, 400–401  
 reset types, 392–394  
 Low-memory global data area, 651  
 Low-memory global data structures, 650–655  
 Low-trace record generation, 658  
 Low-water mark, 918  
 LPAR (Logical Partitioning), 169–170  
 LRU (least recently used), 841  
 lsosf, 130  
 LSOpenCFURLSpec(), 828  
 LSOpenFromRefSpec(), 828  
 LSOpenFromURLSpec(), 828  
 LSOpenFSRef(), 828  
 LSPrefersPPC, 1594  
 lsregister, 829–830  
 lwsync, 1212  
 LZ77 compression, 368  
 LZSS compression, 337*n*

## M

.Mac account, 37, 1397  
 Mac OS 7.6, 6, 24  
 Mac OS 8, 24–27  
 Mac OS 9, 25–27, 114–116  
   driver partitions, 1350  
 Mac OS X  
   auditing system, 621  
   block and character device, 1364  
   Bootstrap Server, 1068  
   command-line tools, 1476  
   creation, 31–36  
   Disk Arbitration subsystem, 1353–1361  
   Disk Images framework, 1367  
   file descriptors, 1375  
   file system APIs, 1386–1387  
   file system comparisons, 1575–1582  
   IPC, 1024, 1041–1060  
   logging, 1326  
   memory management, 838–846

process subsystem, 688  
 scheduler priorities, 792  
 Spotlight, 1410–1441  
 supported file systems, 1386–1409  
 system call processing, 553–557  
 system startup, 388  
 and Unicode filenames, 1533  
 virtual memory subsystem, 837  
 10.x, 36–41  
 Mac OS X 10.0, 36–37  
   shuttle and thread data structures, 715  
 Mac OS X 10.3  
   current thread and activation, 716  
   shuttle and thread, 715  
 Mac OS X 10.4, 39–41  
   current thread and activation, 716  
   dyld, 932–933  
   prebinding, 942  
 Mac OS X Server, 31, 145  
   Xgrid, 146–150  
   Xsan, 150–153  
 MacBook Pro, 3  
 Mach, 18–22, 49–50, 510–519  
   activation and shuttle, 714–716  
   continuations, 717–726  
   exceptions, 518–519, 1112–1129  
   kernel fundamentals, 511–518  
   kernel threads, 710–712  
   ledgers, 431  
   lock sets, 1218  
   memory objects, 516–518  
   nomenclature, 20  
   page map, 921  
   pager interface, 857  
   processor sets, 689–691  
   processors, 692–702  
   remote procedure call (RPC), 712–714  
   semaphores, 1219–1222  
   subsystem initialization, 421–432  
   system calls, 554  
   task and thread sampling, 616–617  
   and task API, 702–705  
   tasks and threads, 727–746  
   threads, 705–712, 717, 730–732  
   traps, 53, 576–584  
   zone allocator, 985–992  
 Mach APIs, 519  
   alarm, 524–526  
   clock services, 522–524

- host information, 519–522
- host statistics, 526–529
- processor, 693–696
- task, 702–705
- thread, 707–710
- Mach Emulation Daemon (machd), 30
- Mach Interface Generator (MIG), 18, 429, 578
- Mach IPC. *See* IPC
- Mach Object File Format, 62
- Mach port(s), 689–690, 1027–1031
  - allocation, 1031
  - communication, 1027
  - and `fork()`, 755
  - internal structure, 1045
  - as objects, 1030
  - rights, 134, 1028
- Mach VM, 846–868
  - architecture, 850
  - backing stores, 853
  - map entries, 851
  - maps, 849–851
  - objects, 851–853
  - pager, 428*n*
  - pgers, 853–861
  - protection values, 884
  - task address space, 849
- Mach VM interfaces, 893
  - controlling memory inheritance, 893
  - debugging, 896
  - naming and sharing memory, 901–907
  - protecting memory, 897–898
  - user-space, 878–893
- `mach_absolute_time()`, 782, 807
- Machine check exception (MCE), 272
- Machine State Register (MSR), 188, 194–195, 401
- Machine-dependent thread state, 538–539
- `machine_init()`, 430
- `machine_startup`, 419–421
- `machine_thread_create()`, 751–753
- `mach_msg()`, 1025, 1031
- `mach_msg_overwrite()`, 1025, 1031, 1058
- `mach_msg_overwrite_trap()`, 1032
- Mach-O, 62–66
  - filetypes, 63–64
  - header load commands, 65
  - header structure (32-bit version), 63
  - listing the load commands, 87
  - prebinding, 89
- `mach_port_allocate()`, 1031, 1052
- `mach_port_allocate_full()`, 1052
- `mach_port_get_attributes()`, 1055–1056
- `mach_port_names()`, 1055
- `mach_task_self()`, 703
- `MachTen`, 23
- `mach_thread_self()`, 707
- `MACH_TRAP()`, 579–582
- `mach_vm_allocate()`, 885, 893–896, 948
- `mach_vm_behavior_set()`, 889–890
- `mach_vm_copy()`, 888
- `mach_vm_deallocate()`, 885
- `mach_vm_inherit()`, 884, 886
- `mach_vm_map()`, 881–884
- `mach_vm_msync()`, 891
- `mach_vm_protect()`, 886, 897
- `mach_vm_read()`, 887, 899–901
- `mach_vm_read_overwrite()`, 887
- `mach_vm_region()`, 893
- `mach_vm_region_info()`, 896–897
- `mach_vm_remap()`, 884–885
- `mach_vm_wire()`, 889
- `mach_vm_write()`, 888, 899–901
- Macintosh File System (MFS), 1386
- Macworld Conference, 3
- MacWorld Expo, 33–34
- `macx_swapoff()`, 919
- `macx_swapon()`, 919
- `macx_triggers()`, 919–920
- Magic number, 323*n*
- Magnified mode, 118
- `makecontext()`, 1138
- `malloc`
  - implementation, 954
  - zones, 951–956
- `malloc()`, 966–967, 1000
- `MallocDebug.app`, 130
- `malloc_history`, 130
- Managed Object Context, 112–113
- MAP, 207–208
- Mapping tables, 414–415
- `mapping_adjust()`, 433
- Master boot record (MBR), 363, 1352
- Master CPU, 264*n*
- Master directory block, 1503
- Master processor, 388–391, 397, 405, 407
- Matching dictionaries, 1268–1269
- Matching dictionary, 1359–1361
- Matchmaker (language), 18
- `maxmem`, 296

MAXUSERS, 686  
`mbuf_expand_thread()`, 711  
`mdcheckschema`, 1439  
`mdfind`, 1439  
`mdfind` command, 128–129  
`mdimport`, 1414, 1439  
`.mdimporter`, 79  
`MDItem` attributes, 1412  
`mdls` command, 128  
`MDQuery`, 1431–1434  
`mdutil`, 1439  
`memberd`, 132  
`Memex`, 1410  
`Memory`, 835  
  addresses, 182  
  allocated, 414  
  classes, 1244  
  controller, 157, 171  
  copy-on-write, 862–864  
  data structures, 410  
  deallocation, 982  
  debugging, 976  
  debugging Mach VM, 896  
  dynamic pager program, 918–921  
  inheritance, 893–895  
  kalloc zones, 992–993  
  kernel and user address space, 907–911  
  `kmem_alloc`, 983–985  
  Mach VM, 846–868  
  Mach VM interfaces, 893–907  
  Mach VM user-space interface, 878–893  
  `mach_vm_allocate()`, 893–896  
  `mach_vm_map()`, 881–884  
  `mach_vm_protect()`, 897–898  
  `mach_vm_remap()`, 884–885  
  manager, 851  
  map, 333  
  memory management, 838–846  
  memory-mapped files, 1001–1005  
  NVRAM, 266  
  objects, 20, 516–518  
  OSMalloc, 993–994  
  page faults, 876–877  
  page replacement, 871–872  
  pagers, 853–861  
  patching, 398–400  
  PRAM, 265–266  
  protection, 6–7, 34  
  reading kernel memory, 841–845  
  regions, 409, 466–468, 517  
  resident memory, 868–877  
  shared memory, 902–907, 922–942  
  64-bit computing, 1005–1020  
  subsystem, 839  
  swap files, 918–921  
  synchronization, 202  
  Task Working Set (TWS), 942–948  
  transfer, 888  
  unified buffer cache (UBC), 913–918  
  universal page lists (UPLs), 912  
  update daemon, 921–922  
  virtual memory initialization, 877  
  `vm_page` structure, 868–869  
  *See also* Virtual Memory  
`Memory allocation` in the kernel, 980  
  BSD portion, 997–1000  
  I/O kit, 995–997  
  `kalloc`, 992–993  
  `kmem_alloc`, 983–985  
  `libkern`'s C++ environment, 1000–1001  
  Mach zone allocator, 985–992  
  OSMalloc, 993–994  
  page-level allocation, 980  
`Memory allocation` in user space, 948–951  
  enumerating all pointers, 971–974  
  intercepting malloc layer, 979  
  logging malloc operations, 976–978  
  `malloc()` routine, 966–968  
  memory allocator internals, 951, 957–964  
  scalable zone statistics, 974–976  
`Memory Management Unit (MMU)`  
  address translation, 181–183  
  block address translation, 184–185  
  ERATs, 183  
  large pages, 184  
  segment lookaside buffer (SLB), 181–182  
  translation lookaside buffer (TLB), 181–182  
`Memory properties`  
  Power Mac G5, 295  
  PowerBook G4, 294  
`memory_manager_default_init()`, 879  
`Memory-mapped files`, 1001–1005  
`memory_object_control_bootstrap()`, 879  
`memory_object_data_request()`, 859, 877  
`memory_object_data_return()`, 860  
`memory_object_destroy()`, 861

- `memory_object_get_attributes()`, 859
- `memory_object_init()`, 856–858
- `memory_object_lock_request()`, 860
- `memory_object_server()`, 856
- `memory_object_terminate()`, 861
- MergePef, 130
- MERSI, 180
- Message header fields, 1034–1035
- Message passing. *See* IPC
- Message trailer, 1038–1039
- Messages, 20, 516, 1031–1040
- Messaging implementation, 1057
- Metadata
  - allocation zone, 1561–1562
  - attributes, 1432
  - controllers, 151–152
  - framework, 1411–1412
  - importer, 79, 1428–1430
- Metadata server (mds), Spotlight, 1414
- Methods, Open Firmware, 296–297
- Microcoded instructions, 206
- Micropartitioning (processor), 170
- Microsoft
  - Active Directory, 140
  - Chicago, 2
  - Windows 95, 2, 24
- Microsoft Windows
  - NTFS, 1582–1585
- `microtime()`, 455
- MIDI System Services, 99
- MIG, 1094–1111
  - and client-server systems, 1100
  - and kernel, 1108
  - specification files, 1095
- Migrating threads model, 717
- MiniPCI, 161–162
- mkext cache (multiextension cache), 335–338
- `mkfifo()`, 1147
- MkLinux, 23
- `mmap()`, 1001, 1004, 1349
- Mobile Motion Module. *See* Sudden Motion Sensor (SMS)
- MODE32 software, 267
- Model (Core Data), 111
- Model-View-Controller (MVC), 108, 110*n*
- MONster.app, 131
- Moore, Charles, 279
- Morphemes, 101
- mount command, 1348
- `mount_ftp`, 1395–1396
- Mounting, root file system, 455–464
- Mouse pointer demo, 314–317
- MP Services. *See* Carbon Multiprocessing Services
- .mpkg, 79
- Mr. Registry.app, 54, 1252
- MS-DOS file system, 1390
- MsgType, 1099
- `msleep()`, 1218
- `msync()`, 922
- multicast DNS (mDNS), 148
- Multichip module (MCM), 170
- MULTICS IPC, 1024
- Multiextension (mkext) cache, 59
- Multimedia extension sets, 214
- Multiple users, fast user switching, 119
- Multiprocessing, 170, 771
  - Multiprocessor system processes, 700–702
  - Multiprocessor system startup, 388–395
  - Multitasking, 771
  - Multithreading, 169–170
    - and signals, 1135
  - Multouser startup, 485–495
    - user login, 490–493
  - Munging, 562–566
  - Musical names, 24
  - Mutexes, 52, 1215, 1224–1225
  - `my_thread_setup()`, 730–732, 1596–1597

## N

- Naked rights, 1050
- Name servers, 1060–1079
- Named entry handle, 883
- Named Pipes (Fifos), 1147
- Named POSIX semaphores, 1160
- Nanokernel, 25–26
- NaN<sub>s</sub> (Not a Number), 402
- Natural unit, 378
- `natural_t`, 1043
- Navigation Services, and UTIs, 831
- NavigationServices, 106
- `nchinit()`, 448
- Nested pmap, 867
- `netboot_mountroot()`, 462–463
- NetInfo, 140

Netname server (nmserver), 30  
**netstat**, 276  
 Network booting, 346–349, 463–464  
 Network Data Representation (NDR), 1123  
 Network Diagnostics.app, 60  
 Network domain, 58  
 Network file systems, 1386  
 Network funnel, 445  
 Network Kernel Extensions (NKE), 51, 153–154  
 Network Message Server, 515, 1060, 1063–1064  
 Network Name Server, 1063  
 Networking, 153–154  
 Networking, user-level, 90  
 Networking subsystem initialization, 450–453  
 Network-Transparent Ports, 515  
 New World architecture, 268  
 NeWS, 10  
`new_system_shared_regions()`, 927, 931  
 NeXT, 9–10  
     and Cocoa, 107  
     NEXTSTEP, 10–13  
     OPENSTEP, 13–15  
     Plan A, 9  
     and Ross Perot, 10  
     startup history, 9  
     StepStone, 11  
 NeXT Interface Builder, 108  
 NEXTSTEP Driver Kit, 1234  
 NFS, 463  
     asynchronous I/O server, 723–724  
     buffer cache, 916–918  
     file system, 1396–1397  
`nfs_bind_resv_thread()`, 711  
 nfsd, 724  
 NFSEXPORT, 1581  
 nfsiod program, 723–724  
`nfssvc()`, 724  
 nib files, 108  
 nibtool, 109  
 niutil, 141  
 nm (object file symbol table), 63  
 Node identifier, 375  
 Nodes, B-Tree, 1484  
 Nonexecutable stack, 1595–1596  
 Non-Java (NJ) bit, 402  
 Nonleaf class, 55  
 Nonmaskable interrupt (NMI), 394  
 Nonvolatile memory. *See* NVRAM  
 Nonvolatile register, 226

North Bridge/South Bridge, 161  
 Not Quite NFS (NQNFS), 1397  
 Notifications, 1181–1197  
     `kqueue(2)`, 1191  
     `notify(3)` API, 1184, 1191  
     `notify_post()`, 1185–1186  
 Novell, 3  
 np (nonportable) calls, 759–761, 929  
 NSBundle, 78  
 NSManagedObject, 111  
 NSMovie, 114  
 NSMovieView, 114  
 NSNotification, 1181  
 NSTask Cocoa Class, 763–764  
 NSThread Cocoa Class, 764–766  
 NSUndoManager, 110  
 NSWorkspace class, 116  
 NTFS file system, 1391  
 Nubs, 56, 1242–1243  
 NuBus, 269  
 NuKernel, 4  
 Null mount file system, 1405  
 Null port, 1048  
`nvedit`, 326, 327  
 NVRAM  
     dumping contents, 300–301  
     EFI, 369–370  
     and Open Firmware, 266  
     panic log, 343  
     Platform Expert, 57  
     the script, 325  
     variables, 296, 343, 1590

## O

OBEXAgent.app, 60  
 Obfuscation scheme, password, 491  
 Object graph, 110  
`ObjectAlloc.app`, 130  
 Objective-C, 11  
     and Cocoa, 107  
     protocols, 1165  
 Objects, Mach VM, 851–853  
 OEA registers, 194  
`open()`, 1002–1009  
 Open Firmware, 46, 264, 272  
     ASCII password, 350  
     character display, 317–320

- clock implementation, 320–321
- command-line editing, 275
- console, 325
  - creating windows, 323–324
  - drawing images, 322–323
  - emulators, 278
  - features, 273
  - font demo, 318–320
  - and Forth, 279–289
  - Forth shell, 274–275
  - `getfwpath`, 341
  - interfaces, 298–300
  - Mach-O binaries, 326
  - mouse pointer, 314–317
  - mouse pointer demo, 315–317
  - password, 349–352
  - pixel-addressable printing, 318–320
  - programming, 300–324
  - serial download, 278
  - startup key combination, 274
  - TELNET, 275–276
  - TFTP, 276–278
  - Towers of Hanoi problem, 303–314
  - window, 324
  - See also* Firmware
- Open Firmware Working Group, 272–273
- Open Group, 21*n*
- Open Scripting Architecture (OSA), 106
- Open Software Foundation (OSF), 21
  - and Apple, 23
- Open source compilers, 124*n*
- Open source development, 35
- Open source software, 43
- Open Transport, 90
- OpenAL, 100
- OpenBoot, 272
- OpenDarwin, 35
- OpenDoc, 5, 7
- `openGate()`, 1246
- OpenGL, 37, 96
- OpenGL interfaces, 97
- OpenGL Profiler.app, 130
- OpenScripting, 106
- OPENSTEP, 13–15
- Operators, Forth words, 285
- Option ROM, 363
- Organick, Elliott L., 1024
- OS Picker, 330
- `osacompile`, 126
- `osascript`, 125
- `OSBacktrace()`, 1326
- `osfmk/`, 502
  - directory, 508–509
  - `osfmk`, MLinux, 23
  - `OSlibkerninit()`, 1254
  - `OSMalloc` functions, 993–994
  - `OSReportWithBacktrace()`, 1326
  - `OSRuntimeFinalizeCPP()`, 1264
  - `OSRuntimeInitializeCPP()`, 1254, 1264
  - `otool`, 65, 69, 87, 89, 130
  - Out-of-line data, 699–700
    - message, 516
  - Out-of-line memory, 1090
  - Out-of-order execution, 169
  - Overwrite call, 888

## P

- `PackageMaker.app`, 79
- `Packages/`, 496
- Pad word, 957
- Page deactivation, 890
- Page faults
  - resident memory, 876–877
  - and TWS, 947
- Page frame, 835
- Page frames, initializing, 425–426
- Page queues, 870
- Page table, 181, 403
- Page table entry group (PTEG), 411–413
- Pageable memory, 995–997
- Pageable resident page, 870–871
- Page-in, 857
- Page-in operation, 1006–1007
- Page-level allocation, 980
- Page-out, 857, 1008–1009
- Pageout daemon, 435, 872–874
- Pager
  - Mach VM, 428*n*, 853–861
  - port, 855–856
  - problems, 860
  - reply port, 858
- Pages (address translation), 181
- pagestuff, 130
- Palette bundle, 79
- Palm, Inc., 9
- Panic image, 437–438

- Panic log, 343
- Panic server, 1324
- Panic UI, testing, 439
- Panther, 36, 39
- Parameter area, 231–234
- Parameter memory. *See* PRAM
- Parameter passing, 231, 239–240
- Parameter (use of term), 231
- Parity bits, 180
- `parse_bsd_args()`, 445
- `parse_machfile()`, 818
- Partition map, 1350, 1371–1372
- Partition map entries, 1350
- Partition table editor, 1348
- Partitioning, GPT, 374–376
- Partitioning, x86 computers, 1590–1591
- Partitioning (drive), 363
- Partitioning (processor), 170
- Passing parameters, 239–240
- Passive data, 1036
- Passive matching, 1256
- Password
  - obfuscation scheme, 491
  - Open Firmware, 349–352
- `password` command, 349–350
- Pasteboard Manager, 831
- Pasteboard Server (pbs), 31, 60
- Patch partition, 1350
- Patch tables, 398–400
- Pathnames, and kernel functions, 382
- Payne, R. Bruce, 835–836
- `pbdprojectdump` command-line program, 124
- PC Card Standard, 162
- PCA PTEG Control Area, 411
- `pcb`, 538
- PCBs process control blocks, 414
- PCI bus bridge, 157
- PCI devices, listing, 1303–1304
- PCI Express, 160–162
- PCI-X, 159
- PCMCIA, 162*n*
- PC-style partitioning, 1352
- PDF, and Quartz 2D, 92–93
- PDF Kit, 40
- `pdisk` command, 1348–1350, 1352
- PE32 binary format, 378
- PEF files, merging, 130
- PEFViewer, 130
- `perfmon_init()`, 430
- Perforce Software, Inc., 124
- Performance Monitoring Counter (PMC), 131
- Perl, 125, 1559–1560
- Permissions, HFS+, 1534–1538
- Perot, Ross, 10
- Per-process kernel tracing, 617–619
- Per-process profiling, 614–616
- Per-processor data, 390–392
- Per-processor information areas, 650–655
- Perq computers, 17
- Persistent Store Coordinator, 112
- Personalities, driver, 1255–1257, 1282
- `pexpert/`, 502
  - directory, 509–510
- PHP, 125
- Physical addresses, 1257
- Physical memory, examining, 645–648
- Physical memory window, 865
- PI (processor-interconnect), 164
- `pid_for_task()`, 581–584, 703
- Pink project, 2, 4–5
- `pipeinit()`, 449
- Pipes, 1145–1146
- PIR Processor ID Register, 394
- Pixel-addressable printing, Open Firmware, 318–320
- `.pkg`, 79
- `PkgInfo` file, 78
- PKZip archives, 1557
- Plan A (NeXT), 9
- Platform, 155–156
- Platform Expert, 56–57, 408, 509–510, 1255
  - computer serial number, 1304–1305
  - temperature sensor readings, 1305–1308
- Platinum-look user interface, 25
- Play list (BootCache), 353
- `plist` files, 81–82
- Plug-and-play, 53
- Plug-in API, 81
- Plug-in architecture, Core Image, 97
- Plug-in bundle, 79
- `plutil`, 82
- `pmap`, 411, 865–868
  - initialization, 411–415
- `pmap_create()`, 865–866
- `pmap_destroy()`, 866
- `pmap_enter()`, 866–867
- `pmap_init()`, 877, 879

`pmap_map()`, 865–868  
`pmap_map_iohole()`, 865  
`pmap_page_protect()`, 867  
`pmap_protect()`, 867  
`pmap_reference()`, 866  
`pmap_remove()`, 867  
`pmap_switch()`, 868  
`pmap_vm_init()`, 410  
`PMC_Index.app`, 131  
 Pointer enumeration, 972  
 Polled-mode, 602  
 Port(s)  
     allocation, 1031, 1052, 1053  
     descriptor, 905–907  
     Mach, 134, 514, 689, 755, 1027, 1045  
     pager, 855–856  
     renaming, 1047  
     rights, 1027–1030, 1053, 1088–1094  
     set, 515, 620–621, 1029, 1085  
 Portable Executable (PE), 378  
 POSIX, 533  
     BSD init, 449  
     threads (Pthreads), 75, 759–774  
 POSIX IPC, 1156  
     semaphores, 1156  
     shared memory, 1160  
 Postfix notation, 279  
`postModLoad()`, 1265  
`postsig()`, 1141  
 POWER4+, 169–172  
 POWER6, 171  
 Power Mac G5, 156–166  
     amber, 256–261  
     architecture and platform, 164  
     atomic compare-and-store function, 244–246  
     comparison with other processors, 166–174  
     dual-processor architecture, 158  
     elastic I/O interconnect, 164–166  
     function rerouting, 246–256  
     HyperTransport, 162–164  
     Intel Core Duo, comparison with, 173–174  
     K2 I/O device controller, 159  
     out-of-order execution, 169  
     PCI Express, 160–162  
     PCI-X, 159–160  
     POWER 4 processors, 166, 169  
     POWER5 processors, 169  
     PowerPC 970, 970FX, and 970MP, 171–224  
     recursive factorial function, 241–244  
     sensors, 222, 1297  
     64-bit processor, 167–168  
     software conventions, 224–240  
     speculative execution, 168–169  
     split transactions, 163  
     superscalar, 168  
     thermal and power management, 220–222  
     U3H system controller, 157–159  
 Power Management Unit (PMU), 265–266  
 Power plane, 1252  
 POWER4 processors, 166, 169  
 POWER4+ processor specs, 172–174  
 POWER5 processors, 169–171  
 POWER5+ processors, 171  
 Power-on reset (POR), 271–272  
 Power-on self-test (POST), 264, 366–369  
 PowerPC  
     bootloader, 46  
     exceptions, 383–385  
     executables, 68, 116–117  
     970s, 171–173  
     processors, 155–156  
     registers, 189–191  
     system calls, 554, 584–586  
     unconditional branch instruction, 249  
 PowerPC 970FX, 174–224  
     AltiVec, 212–219  
     branch prediction, 211–212  
     caches, 175–181  
     core, 202–212  
     cycle-accurate core simulator, 256–261  
     D0, 206  
     D1, D2, D3, 206–207  
     GD, DSP, WRT, GCT, MAP, 207–208  
     IFAR, ICA, 206  
     instruction pipeline, 205–210  
     instruction set, 197–202  
     internal buffers and queues, 185  
     ISS, RF, 208–210  
     maximum instructions, 210  
     Memory Management Unit, 181–185  
     power management, 219–222  
     prefetching, 186–187  
     processor-type table entry, 395–397  
     registers, 188–194  
     rename registers, 194–197  
     64-bit architecture, 222–224  
     softpatch facility, 224  
     XFER, 207

PowerPC 970MP, 395*n*  
 PowerPC OEA registers, 195  
 PowerShare, 5, 7  
 PowerTalk, 5, 7  
 PowerTune, 220  
`ppc_init()`, 405–409, 470  
`ppc_init_cpu()`, 470–471  
`ppc_vm_init()`, 410, 877  
 PPM (portable pixmap) image data, 322  
 PRAM, 57, 265
 

- resetting, 266
- resetting, and password security, 352*n*

 Prebinding, 71, 86–89, 941
 

- Mac OS X 10.4, 942

 Preboot Execution Environment (PXE), 363–364  
 Preboot software, and EFI, 364  
 Precious pages, 891  
 Precompiled headers, 123  
 Predefined queries, 111  
 Predictive compilation, 122  
 Predictor tables, 211–212  
 Preemptive multitasking, 7, 771  
 Preference pane bundle, 80  
 Preferred Executable Format (PEF), 104  
 Prefetch Filter Queue (PFQ), 185, 186  
 Prefetch Request Queue (PRQ), 186  
 Prefetching, 187  
 Prefix matching, 459–460  
`preModLoad()`, 1264  
`prepare()`, 1258  
 Prerequisite packages, 677–680  
 Primary partitions, 363  
 Print (Carbon subframework), 106  
 PrintCore, 101  
`printf()`, 1325
 

- without using malloc(), 971–972

 Printing libraries, 88  
 printlibs, 88  
 Priority ranges, 791–792  
 Private frameworks, 60, 84–85  
 Privileged state mode, 188  
`probe()`, 1256, 1280  
 Probe score, 1256  
 Problem state mode, 188  
 /proc file system, 1409  
 proc structure, BSD, 747

`procdup()`, 465, 751  
 Procedure (use of term), 231  
 Process accounting, 134  
 Process control block (PCB), 689  
 Process model, 718  
 Process serial number (PSN), 767  
 Processes, 683–684
 

- activation and shuttle, 714–716
- BSD processes, 746–759
- continuations, 717–726
- in early UNIX, 684–685
- execution environments, 686–687
- `execve()` system call, 812–827
- Launch Services, 828–830
- Mach tasks and threads, 727–746
- POSIX threads (Pthreads), 759–774
- process limits, 685–686
- processors, 689–702
- remote procedure call, 712–714
- scheduling, 774–811
- task structure, 702–705
- thread migration, 717
- thread structure, 705–712
- Uniform Type Identifiers (UTI), 830–833

 Processor Abstraction Layer (PAL), 365  
 Processor API, 693–696  
 Processor cache information, retrieving, 178  
 Processor event labels, 260  
 Processor initialization, 388–421  
 Processor interconnect, 158, 165  
 Processor set API, 689–691
 

- processor structure, 692
- Processor traps, 532
- Processor types, 395–398
- Processor usage, conversion factor, 796–799
- `processor_exit()`, 693–696, 700
- `processor_get_assignment()`, 696
- `processor_info()`, 693
- `processor_set_create()`, 690
- `processor_set_default()`, 690
- `processor_set_destroy()`, 690–691
- `processor_set_info()`, 691
- `processor_set_stack_usage()`, 691
- `processor_set_statistics()`, 691
- `processor_set_tasks()`, 691

 Processor-specific initialization, 400–401  
`processor_start()`, 693–696, 700  
`processor_start_thread()`, 711  
 Processor-type table entry, 395–397

- `processor_up()`, 784  
`procinit()`, 445  
Productivity features, 119  
  Dashboard, 119–120  
  fast user switching, 119  
  Spotlight, 120  
`profil()`, 614–616  
  .profile bundle, 80  
Programmable read-only memory (PROM), 264  
Programming, 121–131  
  compilers and libraries, 124–125  
  Disc Recording framework, 127  
  Disk Images framework, 127  
  interpreters, 125–129  
  precompiled headers, 123  
  sips command, 128  
  tools, 129–131  
  trivial AppleScript program, 125  
  Xcode, 122–124  
Prologue, 233  
Properties, kernel arguments, 358–361  
Properties, Open Firmware, 292–296  
Property List Editor, 82  
Property list (plist) files, 81–82  
Protected mode, 363  
Protection values  
  Mach VM, 884  
  `vm_protect()`, 1596  
Protocol, EFI, 369  
Protocol initialization, 451–453  
`proto_input_run()`, 711  
`proto_kpi_init()`, 450  
ps command, 829  
Pseudo file systems, 1386–1387  
Pseudo-devices, 297  
Pseudo-extensions, 441  
PseudoKernel, 591–594  
PTEG (Page table entry group), 411–413  
  hash table size, 412–413  
ptesync, 403  
Pthread data structure, 751  
`pthread_cond_signal_thread_np()`, 760  
`pthread_cond_timedwait_relative_np()`, 760  
`pthread_create_suspended_np()`, 760  
`pthread_get_stackaddr_np()`, 760  
`pthread_get_stacksize_np()`, 760  
`pthread_is_threaded_np()`, 760  
`pthread_mach_thread_np()`, 760  
`pthread_main_np()`, 760  
Pthreads, 1222  
Pthreads API, 759–760, 802  
Pthreads library, 707–710  
`pthread_yield_np()`, 760  
`ptrace()` system call, 1144  
Public beta, 35  
Puma, 36–38  
PVR Processor Version Register, 397  
Python, 125  
  resolving an alias, 1555
- ## Q
- QNaN (Quiet NaN), 402  
QNIX, 17–18  
/qtif, 437  
QTKit.framework, 114  
Quality of Service structure, 1054  
Quartz, 91  
Quartz Composer, 40  
Quartz Compositor, 92–94  
Quartz 2D, 37, 92–93  
Quartz 2D Extreme, 40  
Quartz Debug.app, 119, 130  
Quartz Extreme, 38, 94–95  
Quartz Extreme with Accelerated 2D, 94–95  
Quartz Services, 94, 1293  
quartz-wm (window manager), 103  
Queues, 970FX, 185, 209  
QuickDraw, 101  
QuickDraw GX, 5, 7  
QuickDraw 2D, 95–96  
QuickDraw 3D, 5, 7  
QuickTime, 36–41, 114  
  Core Image and Core Video, 98  
  RAW file, 437  
  video-rendering pipeline, 98  
QuickTime Virtual Reality (QTVR), 99  
Quotas, HFS+, 1546–1547
- ## R
- RAID  
  AppleRAID, 344–345  
  booting from a software RAID, 344  
  and BootX, 346n  
  disk supersets, 1362

RAID, (*continued*)  
 headers, 345–346  
 Xsan, 150–151  
 Xserve, 150, 153  
 RAM, 223, 268, 294, 354  
 RAM disks, 459, 1370  
 Random number generator (RNG), 637  
 Raptor, 4, 152  
 RAS, 180  
 Rashid, Richard, 17, 18, 22  
 Raster graphics, 93  
 Raster image formats, 93  
 Raw devices, 1349  
 Raw kernel memory access, 845  
 Rcsid, 1099  
 Reachability (of a branch), 248  
 READDIRATTR, 1581  
 readdirattr(), 1582  
 Read/write locks, 1216  
 Real address (RA), 181  
 Real mode, 363  
 Real-time clock, 778–780  
 Receive rights, 514, 1027–1030  
 Receive side (data transmission), 165  
 receivePacket(), 602  
 Reconnect UAM, 1394  
 Recorder function, 972  
 Records, B-Trees, 1487  
 Recursion, 305  
 Recursive factorial function, 241–244  
 Recursive invocation, 73  
 Red project, 2, 4  
 Reggie SE.app, 131, 194  
 Register contents, viewing, 194  
 Registers, 227–228  
 direct calls, 230  
 OEA, 194  
 32-bit, 222*n*  
 VEA, 188–191  
 registerService(), 443  
 Registry, I/O, 1251–1252  
 Relative branching, 201  
 Remote core dumps, 1324–1325  
 Remote debugging, 122  
 Remote disk mounting, 463  
 Remote procedure call (RPC)  
   Mach, 712–714, 717  
   MIG, 1094  
 remove\_all\_shared\_regions(), 931  
 Rename registers, 194–197  
 Rendezvous, 39  
 Repairing permissions, 1536  
 Request, split transactions, 163  
 Resampling an image, 128  
 Reserved areas, 1493  
 Reset  
   exception, 271  
   handler, 392  
   interrupt, 271  
   the PRAM, 266  
   types, 392–394  
 ResetHandler, 394  
 reset\_shared\_file(), 927, 931  
 Resident memory, 868  
 module, 423–425  
 page faults, 876–877  
 page queues, 870  
 page replacement, 871  
 resident pages, 869–870  
 vm\_page structure, 868–869  
 vm\_page\_grab(), 872–875  
 Resolution-independent user interface, 118–119  
 Resource allocation, kernel arguments, 356  
 Resource forks, 1556–1558, 1567  
 Resource ledgers, 729  
 Response, split transactions, 163  
 Restartable system call, 570  
 Return from interrupt, 537  
 Return stack, 280  
 Reverse DNS, 82, 136*n*, 831  
 Reverse Polish notation (RPN), 279  
 revoke(), 1400  
 RF (register file), 208–210  
 rfid, 401, 537  
 Rhapsody, 24, 27–31  
   Blue Box, 29  
   Yellow Box, 29–31  
 Rich Site Summary (RSS), 99  
 RIG. *See* Rochester’s Intelligent Gateway (RIG)  
 RISC (reduced instruction-set computing), 5–6  
 Rochester’s Intelligent Gateway (RIG), 16  
 ROM-in-RAM design, 268  
 Root device, 457–458  
   kernel arguments, 357  
 Root file system, 456  
   mounting, 455–464

- rootdev, 456–457
- rootpath, 334
- Rosetta, 116–117, 1591*n*
  - translate, 1593
  - x86-based Mac computers, 1592–1594
- Rotating a framebuffer, 1288–1292
- Round-robin strategy (Xsan), 152
- Roving pointer, 1507
- RPC. *See* Remote procedure call
- /rsrc, 1556
- rtclock\_intr(), 550–551, 780, 785
- RTP (Real-Time Transport Protocol), 99
- RTSP (Real-Time Streaming Protocol), 99
- r12, 229
- Ruby, 125
- Run loop, 1201–1203
  - sources, 1203–1210
- Run queues, 791–793
- Runtime architecture, 61
  - fat binaries, 66–67
  - linking, 67–74
  - Mach-O files, 62–66
- Runtime recording list, 1568
- Runtime services, EFI, 367
  
- S**
- sa (stand-alone), 55
- Safari web browser, 39, 41
  - Spotlight search, 1440–1441
- Safe mode, 332, 339, 469
- Samba, 1397
- sample (process profiler), 130
- Sampler.app, 130
- Sampling tasks and threads, Mach, 616–617
- Saturate (SAT) bit, 402
- Saturn.app, 131
- Save area(s), 414
  - argument munging, 563
  - BSD system calls, 557
  - exception, 539
  - LowTraceRecord structure, 658
  - pointers (pcb and upcb), 538
  - vector, 333
- Save area free list, 540
- Save area free pool, 540
- .saver, 80
- say command, 125, 128
- /sbin/init, 469
- /sbin/launchd, 469
- /sbin/mach\_init, 469
- sbrk(), 949
- Scalable malloc allocation categories (32-bit), 956
- Scalable malloc allocation categories (64-bit), 956
- Scalable zone allocator, 953
- Scalable zone data structures, 955
- Scalable zone statistics, 974–976
- Scalar, 168
- Scaling modes, 118
- sched\_init(), 421–423
- sched\_pri\_shift, 796–797
- sched\_startup(), 785
- sched\_tick\_continue(), 786–789, 801
- sched\_tick\_interval, 796–797
- sched\_tick\_thread(), 711
- Scheduler initialization, 421–423
- Scheduler operation, 789–802
- Scheduler services, 432–435
- Scheduler startup, 784–786
- Scheduler tick, 786
- Scheduler tick thread, 432
- Scheduling, 774–775
  - absolute- and clock-time intervals, 782–784
  - behavior, 357–358
  - policy, 774, 802–811
  - statistics, 527–529
  - timeslicing quantum, 775–777
  - timing and clocks, 777–782
- SCOM (Scan Communications) facility, 403–404
- SCOMC (control) registers, 404
- SCOMD (data) registers, 404
- Scopes (kauth), 1446–1448
- Screen dimensions, 301–302
- Screensaver bundle, 80
- Screenshot, text console, 648–649
- Script, the, 326–327
- Scriptable applications, 1172
- Scriptable Image Processing System. *See* sips command
- Scripting languages, 125
  - AppleScript, 106, 115, 125–126, 1173–1175
  - Perl, 125, 1559–1560
  - PHP, 125
  - See also* Open Scripting Architecture (OSA)
- sc\_usage, 130

**s**  
 scutil command, 142  
**SEARCHFS**, 1581  
 searchfs(), 1571, 1582  
 Searching. *See* Spotlight  
 SearchKit.framework, 90  
 Sector, hard drive, 1345–1346  
 Secure empty trash, 39, 132, 139  
 Secure transport, 135  
 Security, 131–133  
     architecture, 133  
     auditing system, 143–145  
     authorization services, 137  
     Basic Security Module (BSM), 134  
     command-line tools, 141  
     and daemons, 132  
     firmware, 349–352  
     kernel-space security, 133–134  
     Security Agent, 136  
     security command, 140  
     Security Server, 135–136  
     snag keys, 351  
     system administration, 139–142  
     token, 1038–1040  
     user-space security, 134–139  
 Security Agent, 135–136  
     and Apple Remote Desktop, 136*n*  
     security command, 139–140  
 Security Server, 135–136  
 Security.framework, 139–140  
 SecurityHI, 106  
 security-mode, 350–351  
 Segment descriptors, 181  
 Segment lookaside buffer (SLB), 181–182, 405  
 Segment registers (SRs), 404  
 Segment table entries (STEs), 181–182, 404  
 Segments (address translation), 181  
 select() system call, 722  
 Self port, 577, 1049  
 semaphore\_create(), 759, 1219  
 Semaphores  
     Mach, 1219–1222  
     Named POSIX, 1160  
 semaphore\_signal(), 759  
 semaphore\_signal\_all(), 759  
 semaphore\_signal\_thread(), 759  
 semaphore\_wait(), 759, 1219  
 semaphore\_wait\_signal(), 759  
 semexit(), 824  
     sem\_open(), 1156  
     sem\_unlink(), 1156  
     Send rights, 514, 1027–1030, 1088  
     Send-once rights, 1029  
     sendPacket(), 602  
     Sequence number, 1038  
     Serial console, 417–418  
     Serial download, Open Firmware, 276–278  
     Serial port names, 417*n*  
     serial\_keyboard\_poll(), 711  
     Serial-line-based debugging, 603–604  
     Server. *See* Mac OS X Server  
     Server 1.x, 31  
     ServerPrefix, 1099  
     Service bundle, 80  
     Service plane, I/O Kit, 1252  
     Set-associative cache, 179  
     setattrlist(), 1570  
     set-colors, 302  
     setconf(), 455–456  
     setcontext(), 1138  
     Setgid, 819  
     SETI project, 147*n*  
     setthetime(), 455  
     Setuid, 819  
     Shadow chain, 862–863, 896  
     Shadow objects, 848, 862–863  
     Shadowed clone, 929  
     Shadowing, 1370  
     shandler(), 557, 590  
     Shared bus, 159  
     Shared memory  
         client, 903  
         client-server example, 902  
         regions, 466–468  
         server, 905, 922–942  
         setup, 923  
     Shared object files, 932–935  
     shared\_file\_init(), 468  
     shared\_region\_make\_private\_np(), 927–928  
     shared\_region\_map\_file\_np(), 927–929,  
         936–940  
     Shark.app, 131, 1338  
     Shell, EFI, 370–374  
     Shell scripts, 825–827, 842–843  
     Shells, 125  
     Sherlock, 25  
     Sherlock version 3, 39

**shmexec()**, 824  
**shm\_open()**, 1160  
 Shoch, John F., 146  
 show commands, KDB, 1336  
 show all commands, KDB, 1336  
 Shuttle/activation, 714–716  
 Sideband signals, 161  
 sifting, 287  
 sifting (Forth dictionary), 287–288  
**sigalstack()**, 1136  
 Signaling NaN (Not a Number), 402  
 Signals, 1129–1131  
     application-defined signals, 1132  
     asynchronous I/O, 1132  
     and debugging, 1144  
     default action, 1130  
     delivery, 1130, 1138  
     generation, 1130, 1138  
     and IPC, 1132  
     and Mach exceptions, 1141  
     maximum number, 1131–1132  
     and multithreading, 1135  
     ptrace() system call, 1144  
     reliability, 1131  
     signal actions, 1135–1136  
 Signature, BootX, 338  
**SIGTRAP**, 819  
 Silicomp, 21*n*  
 SIMD (single-instruction, multiple data), 213  
 simg4, 131  
 simg5, 131, 256–261  
 Simple integers, 204*n*  
 Simple message, 1034  
 SimpleCryptoDisk, 1311, 1321  
     I/O Kit driver, 1311–1320  
 Single Partition UDIF (SPUD), 1369  
 Single-precision floating-point values, 240  
 Single-user mode, 332, 339, 349, 469  
 Single-user startup, 495–496  
     bootstrap through launchd, 495  
 sips command, 128  
 64-bit architecture, 222–223  
 64-bit comm area, 594  
 64-bit computing, 1005–1020  
 64-bit processor, 167–168  
 64-bit support, 40  
 Sizing memory, 409–411  
 Skidmarks GT.app, 131  
 Slash character, 1156  
 Slave processor, 388–391, 401, 405, 470–472  
**slave\_machine\_init()**, 471  
**slave\_main()**, 470–471  
 Sleep, 1179–1181  
     and reset handlers, 393–394  
 Slice (disk partition), 1348  
     .slideSaver, 80  
 Small allocations, 964–965  
 SMB/CIFS, 38, 1397  
 SMP (symmetric multiprocessing), 170  
 SMT (simultaneous multithreading), 169–170  
 Snag keys, 327–328, 340  
     and security mode, 351  
 Snoop response (SR), 165  
 SOAP (Simple Object Access Protocol), 90  
**socketinit()**, 450  
 Soft limits, 685  
 Soft reset, system reset exception, 271  
 Software conventions, 224  
     byte ordering, 225–226  
     passing parameters, 239–240  
     register usage, 226–228  
     returning values, 240  
     stack usage, 230–238  
 Software debouncing, 394  
 Software dependencies, 679  
 Software traps, 532  
 Source code editor, Xcode, 122  
 Spam filtering, 38  
 Sparse disk images, 1574–1575  
 Sparse file, 1392  
 specfs (special files), 1403  
 Speculative execution, 168  
 SpeechRecognition, 106, 121  
 SpeechSynthesis, 101  
 SPICE, 17  
 Spider Systems, 17–18  
 Spier, Michael J., 1024  
 Spin Control.app, 130  
 SpindownHD.app, 131  
 Spinlocks, 1213–1215  
 Spinning globe data, 336  
 SPL (set-priority-level) call, 1230  
 Splay tree, 1043–1044  
 Split transactions, 163  
**SplitFork**, 1557  
 Split-segment library, 924–925, 927–931, 939

Spotlight, 39–40, 90, 119, 120, 1409  
 architecture, 1410–1415  
 command-line tools, 1439–1440  
 compared with BFS, 1415  
 Fsevents, 1416–1428  
 limitations, 1440–1441  
 metadata importers, 1428–1430  
 metadata server (mds), 1414  
 queries, 1431–1438  
 search, 51  
 temporary files, 1430  
`.spreporter`, 80  
 SPRG1 register, 432  
 SPRG registers, 537  
 Spring-loaded folders, 25  
 SQLite, 40  
 SRR0, 536–537  
 SRR1, 537  
 SSB (source-synchronous bus), 164–165  
 SSH daemon, 474–475  
`ssh-agent`, 122  
 SSL (Secure Socket Layer), 135  
 Stabs, 1338–1344  
 Stack allocation queue, 433  
 Stack usage, 230–238  
   in functions, 235–236  
   printing stack frames, 235–238  
 Stack-based language, 279–280  
 Stacks, Forth, 279–280, 283  
 Stallman, Richard, 533  
 Star Trek, 3  
`start()`, 1280  
`startCPU()`, 394  
`StartIOKit()`, 442–443, 1253–1255  
`startMatching()`, 1255  
 Startup extensions dictionary, 443  
 Startup file, HFS+, 1530  
 Static archive libraries, 69  
 Static branch prediction, 212  
 Static link editor, 62  
 Static linking. *See* Dynamic linking  
 Static prediction, 212  
 Statistics, Mach VM, 892–893  
 StepStone Corporation, 11  
 Sticky Keys, 121  
`stop()`, 1280  
 Storage area network. *See* Xsan  
 Storage Description Register (SDR1), 194

Storage pools, 151  
 Store Data Queue (SDQ), 185  
 Store policy, caches, 179  
 Store Queue (SQ), 185  
 Store Reorder Queue (SRQ), 185  
 StorNext, 150  
 Streaming SIMD Extensions 2 (SSE2), 174  
 Streaming SIMD Extensions 3 (SSE3), 174  
 Streams, and prefetching, 186  
`struct machine_thread`, 687  
`struct proc`, 687  
`struct processor`, 687  
`struct processor_set`, 687  
`struct run_queue`, 687  
`struct task`, 687  
`struct thread`, 687  
`struct uthread`, 687  
 Subframeworks, 101  
 Subnormal numbers, 219n  
 Subprocessor partitioning, 170  
 Subroutine (use of term), 231  
 Sudden Motion Sensor (SMS), 1297–1303  
 Sun Microsystems  
   Network Information Service (NIS), 140  
 NeWS, 10  
   and Open Firmware, 272–273  
 OpenStep, 13  
 Solaris, 7  
 Superscalar, 168  
 Support packages, 297–298  
 Suspend count, 704, 708  
 SVG (Scalable Vector Graphics), 93  
 Swap files, 518  
   memory, 918–921  
`swapcontext()`, 1138  
 Swing, 102, 113  
 Symbol table. *See* Stabs  
 Symbolic link, 1390  
 Symbolic links (symlinks), 1553–1554  
 Symmetric copy-on-write, 862–864  
 Symmetric multiprocessing (SMP), 771  
`sync()`, 921–922, 1212, 1567  
 Sync flags, 891  
 Synchronization, 1210–1232  
   advisory-mode file locking, 1230–1232  
   atomic operations, 1213  
   BSD condition variables, 1218  
   classes, 1244

funnels, 1223–1229  
 low-level locking, 1213–1217  
 Mach lock sets, 1218  
 Mach semaphores, 1219–1222  
 Pthreads, 1222  
 SPLs, 1230  
 Synchronous bus, 165  
 synthfs, 1404  
`sysctl()`, 178, 845–846, 1279  
 sysctl hierarchy, 1277  
 sysctl nodes, 1277  
`sysctlbyname()`, 845–846, 1594  
`sysctl_handle_init()`, 1277  
`sysctl_handle_string()`, 1277  
`sysctl_register_fixed()`, 450  
 sysfs, 845  
 syslogd, 1077  
 System Abstraction Layer (SAL), 365  
 System administration, and security, 139–142  
 System architecture, 164  
 System call, 201, 533, 537  
     cancellation, 562  
     categories, 557–601  
     processing, 553–557  
     return values, BSD, 571  
     x86-based Mac computers, 1597  
 System call data structures  
     Third Edition UNIX, 572  
 System clock configuration, 781  
 System Configuration, 142  
 System domain, 59  
 System Enabler, 267  
 System entry mechanisms, 533–543  
 /System/Library/ Directory, 59–61  
 System linkage instructions, 537  
 System Management Unit (SMU), 265–266  
 System preferences, 34  
 System Profile reporter, 80  
 System reset exception, 271  
 System reset interrupt (SRI), 271  
 System restart, 493–495  
 System shutdown, 493–495  
 System size configuration, 686  
 System startup, 388–395  
 System V IPC, 449  
`system_profiler`  
     x86-based Mac computers, 1588  
 SystemStarter  
     and daemons, 498

## T

Taligent Object Services (TalOS), 4–5  
 Target disk mode, 328  
`tart`, 237*n*  
 Task(s)  
     address spaces, 849  
     API, 702–705  
     functions, CHUD, 605  
     and IPC, 1048  
     Mach, 431  
     port, 514  
     roles, 810–811  
     sampling, Mach, 616–617  
     and threads, 512–514, 727–746  
 Task (Mach), 19  
 task structure, 702  
 Task Working Set (TWS), 942–948  
     `execve()` system call, 947  
     global user profile cache, 945  
     and page-faults, 947  
`task_assign()`, 696  
`TASK_BACKGROUND_APPLICATION`, 811  
`TASK_CONTROL_APPLICATION`, 811  
`task_create()`, 704, 727–729  
`TASK_FOREGROUND_APPLICATION`, 811  
`task_for_pid()`, 704, 899  
`task_get_exception_ports()`, 704  
`task_get_special_port()`, 704  
`TASK_GRAPHICS_SERVER`, 811  
`task_halt()`, 818  
`task_info()`, 704  
`task_init()`, 431  
`task_policy_get()`, 705  
`task_policy_set()`, 705  
`task_resume()`, 704  
`task_set_exception_ports()`, 704  
`task_set_64bit`, 816  
`task_set_special_port()`, 705  
`task_suspend()`, 704  
`task_swap_exception_ports()`, 704  
`task_terminate()`, 704  
`task_threads()`, 704  
`TASK_UNSPECIFIED`, 810  
`tbxi` (Toolbox image), 270  
 Tcl, 125  
`tsh` shell, 125  
 TELNET, 275–276  
 telnet support package, 275–276

Temperature, Hot File Clustering, 1563–1564, 1567, 1569  
 Temperature sensor readings, 1305–1308  
 Templates, Xcode, 124  
 Temporary files, 1430  
 Tenon Systems, 23  
`terminal-emulator`, 317  
 Test-and-set, 1211  
 Tevanian, Avadis, 20, 22  
 Text-to-Speech, 121  
 Textual console, 648  
 TFTP (trivial file transfer protocol), 276–278  
     daemon, 276  
`thandler()`, 548, 551–552  
 Thermal and power management  
     Power Mac G5, 220–222  
 Thermal monitoring, 421  
 Thermometers, 221–222  
 32-bit comm areas, 594–601  
 32-bit processor, exception processing, 544–545  
 32-bit registers, 222n  
 32-bit system enabler, 267  
 Thrashing, 837  
 Thread(s)  
     call graph, 789–790  
     creation, x86 computers, 1596–1597  
     functions, 605  
     info UFT, 587–589  
     interface functions, 730–732  
     and IPC, 1051  
     Mach, 20, 431, 512–514, 705–712  
     migration, 717  
     port, 514  
     sampling, 616–617  
 thread structure, 706–707  
`Thread Viewer.app`, 130  
`thread_abort()`, 709, 1118  
`thread_bind()`, 435  
`thread_block()`, 718  
`thread_block_reason()`, 718  
`thread_bootstrap()`, 407  
`thread_create()`, 707, 730, 759  
`thread_create_running()`, 708, 759  
`thread_dup()`, 751  
`THREAD_EXTENDED_POLICY`, 802–803  
`thread_funnel_set()`, 1228  
`thread_funnel_switch()`, 1228  
`thread_get_exception_ports()`, 709  
`thread_get_special_port()`, 710  
`thread_get_state()`, 707, 759  
`thread_info()`, 707  
`thread_init()`, 431  
`thread_invoke()`, 718  
`threadMain()`, 1246  
`thread_policy_get()`, 710  
`thread_policy_set()`, 809  
`THREAD_PRECEDENCE_POLICY`, 803–805  
`thread_resume()`, 465, 708, 759  
`thread_set_exception_ports()`, 710  
`thread_set_special_port()`, 710  
`thread_set_state()`, 707, 759  
`thread_stack_daemon()`, 711  
`THREAD_STANDARD_POLICY`, 802  
`thread_suspend()`, 708, 1118  
`thread_switch()`, 708  
`thread_switch_continue()`, 708  
`thread_terminate()`, 708, 759  
`thread_terminate_daemon()`, 711  
`THREAD_TIME_CONSTRAINT_POLICY`, 805–809  
`thread_update_scan()`, 801  
`thread_wakeup()`, 453  
`thread_wire()`, 709  
 Three Rivers Corporation, 17–18  
 Throughput, 160, 165  
 TIFF (Tagged Image File Format), 93  
 Tiger, 36, 39–41  
 Timebase Register (TB), 192, 407, 470, 777, 782  
`timeout()`, 453  
 Timer call processing, 788  
`timer_call_interrupt()`, 786–788  
`timer_call_setup()`, 775  
 Timeslicing quantum, 775–777  
 Timing and clocks, 777–782  
 Tiny allocations, 957–964  
 Tiny block, 958  
 Tiny quanta, 957  
 Tiny regions, 957  
 tlbie, 403  
 tlbsync, 403  
 TLS (Transport Layer Security), 135  
 Toll-free bridging, 89n  
 Toolbox ROM, 267  
 Tools, 129  
     CHUD, 130–131  
     debugging and analysis, 129–130  
     programming, 129–131  
     Visual Tools, 131  
 Towers of Hanoi problem, 303–314

Trace points, 617–618  
 Tracing, 256–261, 649–659  
     using *amber*, 257–259  
 Trailer attributes, 1038  
 Trailer formats, 1038–1039  
 Transfer-handshake (TH) packets, 165  
 Transient attribute, 111  
 Transistors, 167  
     translate (Rosetta), 1593  
 Translation lookaside buffer (TLB), 181–182  
`trap()`, 466, 548, 876  
 Trap instruction, 201  
 Trap table, 593  
 Trap vectors, 536  
 Traps and exception data, 1119  
 Trash can, secure empty, 39, 139  
 Trigger events, 143  
 Trivial AppleScript program, 125  
 Trivial executable, 64  
 TruBlueEnvironment, 33, 114, 591  
 Trust policies, security, 135  
 Two-level namespaces, 71  
`tws_handle_startup_file()`, 816

## U

U-area (user structure), 52*n*, 563, 685, 720  
 UDF (Universal Disk Format) file system, 1391  
 UFS file system, 1391–1393  
 UFS2, 1392  
 UIDs (user identifiers), 133  
 UI/SA registers, 188–191  
 ulocks, 1218  
 Ultra-fast traps (UFTs), 554, 587–594  
     Blue Box calls, 591–594  
     fast traps, 590  
 Umbrella frameworks, 84–86  
 Unconditional branch, 201, 249  
 Underscore, removed, 237*n*  
 Underscore character prefix, 545  
 Unicode filenames, 1533  
 Unidirectional links, 162–163  
 Unified buffer cache (UBC), 52, 447, 913–918  
     routines, 915–916  
 Uniform Type Identifiers (UTI), 830–833  
     declaration, 832–833  
 union mount file system, 1405–1407  
 Uniprocessor patch table, 400

Universal access support, 40, 120–121  
 Universal Binaries, 12*n*  
     x86-based computers, 1591–1592  
     and Xcode, 122  
 Universal Binary format, 66–69  
 UNIVERSAL CD, 1350  
 Universal Disk Image Format (UDIF), 1369  
 Universal Graphics Adapter (UGA), 376–377  
 UNIVERSAL HD, 1349–1352  
 Universal kernel extension, 1261  
 Unix, 33  
     exception handler, 466, 1114, 1141  
     shells, 125  
 UNIX  
     processes, 684  
     system call data structures, 572  
     and virtual memory, 836  
`unix_syscall()`, 557, 568  
 Unlinking open files, 1552–1553  
 Unnamed pipes, 1146  
 Unnamed POSIX semaphores, 1160  
 Unresolved kernel traps, 1121–1122  
 Unstable write, 916  
 Untyped data, 1034  
 Upcall, 856  
`upcb`, 538  
 Update daemon, 921  
 UPL (universal page lists), 840, 912  
`upl_abort()`, 912  
`upl_commit()`, 912  
`upl_create()`, 912  
 Uptime, 699  
 URLMount, 1395  
 USB plane, I/O Kit, 1252  
 User area (u-area), 52  
 User Authentication Method (UAM), 1394  
 User domain, 58  
 User interface, 117  
     Open Firmware, 299  
     productivity features, 119–120  
     resolution-independent, 118–119  
     universal access support, 120–121  
     visual effects, 118  
 User login, multiuser startup, 490–493  
 User logout, 493–495  
 User memory window, 909  
 User mode, 531  
 User save areas, 538  
 User stack configuration, `execve()`, 820–824

User structure, 563  
**USERACCESS**, 1581  
 User-level startup, 472  
   installation startup, 496–499  
   multiuser startup, 485–495  
   single-user startup, 495–496  
**UserPrefix**, 1099  
 User space, 530  
 User-space interface, Mach VM, 878–893  
 User-space pagers, 854  
 User-space program, 469–470  
 User-space security, 134–139  
 User-space system call stub, 573–574  
*/usr/bin/ditto*, 1557  
*/usr/include*, 103  
*/usr/lib*, 103  
 UTF-8, 1533  
 uthread structure, 721, 748  
 U3H system controller, 157–159  
 UUID (universally unique identifier), 334  
 UVM, 23  
*ux\_handler()*, 711  
*ux\_handler\_init()*, 466

**V**

Valid array, 1575  
 Valid ports, 1048  
 Van Horn, Earl C., 683  
**vBigNum**, 125  
**vCards**, 38  
*vclean()*, 861, 1400  
**vDSP**, 125  
 VEA registers, 188–191  
*vecdst()*, 187  
**vecLib**, 213  
 Vector graphics, 93  
 Vector image formats, 93  
 Vector instructions, 199–200  
 Vector parameters, 239–240  
 Vector save area, 333  
 Vector streams, 186  
 -verbose, 1368  
 Verbose mode, 332, 340, 349, 469  
 Version control systems, 124  
 Versioning, 84  
*vfork()* system call, 756–759

**vfs** (virtual file system), 1376–1386  
   functions, 1376–1378  
**VFS layer**, 1376–1386  
   journaling, 1541  
**vfscnf**, 1379  
**vfs\_event\_signal()**, 1574  
**vfsinit()**, 1384–1385  
**vfs\_mountroot()**, 455–456, 461–462  
**vfs\_op\_init()**, 1384  
**vfstable**, 1379  
**VGA**, 363, 376–377  
 Victim thread, 518, 1115  
 Video, 96–100  
 Video console, 418  
 Video-rendering pipeline, 98  
**vImage**, 125, 213  
 Virtual address, 181, 644–645  
 Virtual address space, 529–530, 910–911  
 Virtual disks, 1371–1374  
 Virtual file system (VFS), 51  
 Virtual I/O Server Partition, 170  
 Virtual machine interface, 511  
 Virtual machine monitor (VMM), 659  
   control data structures, 659–660  
   running code in a virtual machine, 662–676  
   using the VMM facility, 661–662  
 Virtual memory, 836  
   contiguous, 20  
   encrypted, 134  
   initialization, 409–415, 877  
   and IPC initialization, 430  
   Mach, 49, 516–518  
   map, 817  
   map entries, 847–848  
   and memory objects, 516–518  
   and MMU, 181, 836  
   and Mode32, 267  
   object, 847–848  
   pages, 847–848  
   regions, 468  
   statistics, 130, 527–529  
   subsystem, 52, 408, 837  
   system limits, 908–909  
   and UNIX, 836  
 Virtual page number (VPN), 182  
 Virtual PC, 673  
 Virtual segment ID (VSID), 182  
 Virtualizer, Classic Startup, 115

Virtual-to-physical (VP) table, 869  
 Visual effects, 118  
 Visual tools, 131  
 VLANs (Virtual LANs), 455  
 VLIW (very-large instruction-word), 168  
`vmachmon32` program, 674–676  
`vm_allocate()`, 663, 930–931  
`vMathLib`, 125  
`vm_deallocate()`, 856, 931  
`vm_fault()`, 876  
`vm_fault_init()`, 879  
`vm_initialized`, 408–409  
`vmmmap`, 130  
`vm_map()`, 856  
`vm_map_copyin()`, 931  
`vm_map_copy_overwrite()`, 931  
`vm_map_create()`, 817  
`vm_map_deallocate()`, 818  
`vm_map_fork()`, 878  
`vm_map_get_upl()`, 912  
`vm_map_init()`, 879  
`vm_map_lookup_entry()`, 851  
`vm_map_protect()`, 931  
`vmm_dispatch()`, 662  
`vm_object_bootstrap()`, 879  
`vm_object_iopl_request()`, 912  
`vm_object_update()`, 860  
`vm_object_upl_request()`, 912  
`vm_page` structure, 868–869  
`vm_page_bootstrap()`, 879  
`vm_page_free_reserve()`, 874–875  
`vm_page_grab()`, 872–875  
`vm_page_module_init()`, 879  
`vm_protect()`, 1596  
`vm_stat`, 130  
`vn` device nodes, 1370–1371  
`vndevice` interface, BSD, 463  
`vn_getpath()`, 1457  
 Vnode (virtual node), 840, 1376–1386  
     activity monitor, 1453–1469  
     pager, 518, 854, 1002–1009  
     structures, 447–448, 1383  
     tag, 1457  
`vnode_authorize()`, 814, 1452  
`vnode_pagein()`, 859–860  
`vnode_pager_init()`, 858  
`vnode_pager_zone`, 455  
`vnode/vfs` layer, 1377  
`VNOP_BLOCKMAP()`, 1366  
`vn_rdwr_64()`, 920  
`vn_read()`, 920  
`vn_read_swapfile()`, 921  
 VoiceOver, 121  
 Volatile register, 226  
`volfs` (volume ID file system), 1407–1409  
`VOL_RENAME`, 1581  
 Voltmeters, 222  
 Volume  
     extending and shrinking, 1572  
     formats, 1581  
     freezing and thawing, 1571  
 Volume (disk partitions), 1348  
 Volume header, HFS+, 1493–1501  
 Volume notifications, 1572  
 Volume Status Database, 1355  
 Volume status database (vsdb), 1355  
 Volume UUIDs, 1355  
 Volume (Xsan), 150  
`VRSAVE`, 216–217, 402  
`vsdbutil`, 1355

## W

`WaitTime`, 1099  
`wakeup()`, 1218  
`wakeup_one()`, 1218  
 Walden, David C., 1026  
 Watcher (fsevents), 1416  
`.wdgt`, 78  
 Weakly linked symbols, 72  
 Web browser, origins, 12  
 Web Kit, 41  
 Web plug-in bundle, 80  
 WebDAV, 1397–1399  
 WebObjects, 15, 31, 113  
 White, Jim E., 1094  
 Widgets, 40, 78, 119  
 WIMG, 854  
 Window backing-store, 94  
 Window management (Quartz), 91–92  
 Window ports, 515  
 Window project, Open Firmware, 324  
 Window Server, 31  
 Windows compatibility, 39  
 Windows NT, Yellow Box, 30  
 WindowServer application, 92  
`wm_ffm`, 103

- Words, Forth, 281–288  
**Work Loop**  
 I/O Kit, 1246–1251  
**Working Set Model**, 942–943  
**Working Set Principle**, 837, 943  
**Workspaces**, Xcode, 122  
**Worldwide Developers Conference (WWDC)**,  
 28, 31  
**Worms**, 146  
**Wrapper**, HFS, 1501–1505  
**Write-after-read (WAR)**, 196  
**Write-after-write (WAW)**, 196  
**Write-back**, 179  
**Write-through**, 179  
**WRT**, 207  
**WRT, GCT, MAP**, 207–208
- X**
- X Window System**, 39, 103  
**X Window System and Aqua**, 117  
**Xcode**, 121–124, 1271–1274  
 class-modeling tool, 107  
 command line program, 124  
 data-modeling tool, 107  
 plug-in, 80  
 supported languages, 124  
 template, 1281–1282, 1311  
**x86 computers**  
 ACPI plane, 1598  
 architecture, 1587–1588  
*boot.efi*, 46, 60, 1589  
 byte ordering, 1594  
 Darwin, 3  
*/dev/mem* and */dev/kmem*, 1597–1598  
**Extensible Firmware Interface (EFI)**, 362–379,  
 1589–1590  
**GUID Partition Table (GPT)**, 1352–1353,  
 1590–1591  
*hwprefs*, 1588  
 kernel address space, 1594–1595  
 nonexecutable stack, 1595–1596  
 partitioning, 1590–1591  
 Rosetta, 1591*n*, 1592–1594  
 Star Trek, 3  
 system calls, 1597  
 thread creation, 1596–1597  
 Universal Binaries, 1591–1592
- x86 Mach-O executables, 68  
 Xerox worms, 146  
**XFER**, 207  
**Xgrid**, 146  
 architecture, 147–149  
 software, 149–150  
**XML-RPC**, 90  
**xnu kernel**, 48  
 BSD, 50–53  
 compiling, 676–681  
 debugging, 601–658  
 debugging (KDB), 50  
 directories, 502–510  
 drivers, 48  
 entering the kernel, 529–543  
 exception processing, 543–553  
 extensions, 49  
 I/O kit, 53–54  
 kernel extensions, 57  
 libkern library, 54–55  
 libsa library, 55–56  
 Mach, 49–50, 510–519  
 Mach APIs, 519–529  
 Platform Expert, 56–57  
 source, 501–510  
 system call categories, 557–601  
 system call processing, 553–557  
 virtual machine monitor (VMM), 659–676  
**xnu package**, 680–681  
**xnu source**, 501–510  
**XOR masks**, mouse pointer demo, 317  
**Xsan**, 150  
 allocation strategies, 152  
 client systems, 152  
 command-line utilities, 153  
 communication infrastructure, 153  
 metadata controllers, 152  
 storage, 150–152  
**Xserve**, 145*n*  
**XSI IPC**, 1155–1156
- Y**
- Yellow Box**, 28, 29–31  
 Cocoa, 33  
 Windows NT, 30  
**YieldToAnyThread()**, 772  
**YieldToThread()**, 772

**Z**

`zalloc()`, 987  
`zalloc_async()`, 988  
`zalloc_canblock()`, 990–991  
`zalloc_noblock()`, 989  
Zero page, 910  
Zero-filled memory, 853  
ZeroLink, 123  
`zfree()`, 990, 991  
`zget()`, 989  
`zget_space()`, 987  
`zinit()`, 877, 987–988

Zone allocator functions, 988–989  
Zone map size, 428  
Zone of zones, 987  
Zone page table, 990  
zone structure, 986  
`zone_bootstrap()`, 879  
`zone_change()`, 989  
`zone_gc()`, 991  
`zone_init()`, 879, 990  
`zone_page_init()`, 991  
zone\_zone, 427  
`zprint`, 986  
zsh shell, 125