
Index

A

access-control lists (ACLs), 832
ACLs (access-control lists), 832
ACPI (advanced configuration and power interface), 862
address space layout randomization (ASLR), 832
admission-control algorithms, 286
advanced configuration and power interface (ACPI), 862
advanced encryption standard (AES), 677
advanced local procedure call (ALPC), 135, 854
ALPC (advanced local procedure call), 135, 854
AMD64 architecture, 387
Amdahl's Law, 167
AMD virtualization technology (AMD-V), 720
Android operating system, 85–86
API (application program interface), 63–64
Apple iPad, 60, 84
application containment, 713, 727–728
Aqua interface, 59, 84
ARM architecture, 388
arrays, 31
ASIDs (address-space identifiers), 374
ASLR (address space layout randomization), 832
assembly language, 77
asynchronous threading, 172

augmented-reality applications, 36
authentication:

multifactor, 689

automatic working-set trimming, 446

B

background processes, 74–75, 115, 296
balanced binary search trees, 33
binary search trees, 33
binary translation, 718–720
binary trees, 33
bitmaps, 34
bourne-Again shell (bash), 789
bridging, 732
bugs, 66

C

CFQ (Completely Fair Queueing), 817
children, 33
chipsets, 836
Chrome, 123
CIFS (common internet file system), 871
circularly linked lists, 32
client(s):
thin, 35
client-server model, 854–855
clock algorithm, 418–419
clones, 715
cloud computing, 41–42, 716
Cocoa Touch, 84

code integrity module (Windows 7), 832
COM (component object model), 873
common internet file system (CIFS), 871
Completely Fair Queueing (CFQ), 817
computational kernels, 835–836
computer environments:
 cloud computing, 41–42
 distributed systems, 37–38
 mobile computing, 36–37
 real-time embedded systems, 43
 virtualization, 40–41
computing:
 mobile, 36–37
concurrency, 166
Concurrency Runtime (ConcRT), 297, 880–881
condition variables, 879
conflict phase (of dispatch latency), 285
containers, 728
control partitions, 723
coupling, symmetric, 17
CPU scheduling:
 real-time, 283–290
 earliest-deadline-first scheduling, 288–289
 and minimizing latency, 283–285
 POSIX real-time scheduling, 290
 priority-based scheduling, 285–287
 proportional share scheduling, 289–290
 rate-monotonic scheduling, 287–288
 virtual machines, 729
critical-section problem:
 and mutex locks, 212–213

D

Dalvik virtual machine, 86
data parallelism, 168–169
defense in depth, 689
desktop window manager (DWM), 831
device objects, 855
Digital Equipment Corporation (DEC), 379
digital signatures, 832
DirectCompute, 835
discovery protocols, 39
disk(s):
 solid-state, 469
dispatcher, 294
DMA controller, 595
doubly linked lists, 32
driver objects, 855
DWM (desktop window manager), 831
dynamic configurations, 837, 838

E

earliest-deadline-first (EDF) scheduling, 288–289
EC2, 41
EDF (earliest-deadline-first) scheduling, 288–289
efficiency, 837
emulation, 40, 727
emulators, 713
encryption:
 public-key, 678
energy efficiency, 837
Erlang language, 241–242
event latency, 283–284
event-pair objects, 855
exit() system call, 120, 121

ext2 (second extended file system), 811
ext3 (third extended file system), 811–813
ext4 (fourth extended file system), 811
extended file attributes, 505
extensibility, 736

F

fast-user switching, 863–864
FIFO, 32
file info window (Mac OS X), 505
file replication, 767
file systems:
 Windows 7, *see* Windows 7
foreground processes, 115, 296
fork-join strategy, 172
fourth extended file system (ext4), 811

G

GCD (Grand Central Dispatch), 182–183
general trees, 33
gestures, 60
global positioning system (GPS), 36
GNOME desktop, 60
GPS (global positioning system), 36
Grand Central Dispatch (GCD), 182–183
granularity, minimum, 797
graphics shaders, 835
guard pages, 847
GUIs (graphical user interfaces), 59–62

H

Hadoop, 765
Hadoop distributed file system (HDFS), 767
handle tables, 844
hands-on computer systems, 20
hardware:
 virtual machines, 720–721
hash collisions, 471
hash functions, 33–34
hash maps, 471
HDFS (Hadoop distributed file system), 767
hibernation, 860–861
hybrid cloud, 42
hybrid operating systems, 83–86
 Android, 85–86
 iOS, 84–85
 Mac OS X, 84
hypercalls, 726
hypervisors, 712
 type 0, 723–724
 type 1, 724–725
 type 2, 725

I

IA-32 architecture, 384–387
 paging in, 385–387
 segmentation in, 384–385
IA-64 architecture, 387
IaaS (infrastructure as a service), 42
idle threads, 840
IDSs (intrusion-detection systems), 691–694
imperative languages, 241
impersonation, 853
implicit threading, 177–183

G
Grand Central Dispatch (GCD), 182–183
OpenMP and, 181–182
 thread pools and, 179–181
infrastructure as a service (IaaS), 42
Intel processors:
 IA-32 architecture, 384–387
 IA-64 architecture, 387
interface(s):
 choice of, 61–62
Internet Key Exchange (IKE), 682
interpretation, 40
interpreted languages, 727
interrupt latency, 284–285
interrupt service routines (ISRs), 840
I/O (input/output):
 virtual machines, 731–732
iOS operating system, 84–85
I/O system(s):
 application interface:
 vectored I/O, 603–604
IP (Internet Protocol), 681–683
iPad, see Apple iPad
ISRs (interrupt service routines), 840

J

Java Virtual Machine (JVM), 107, 726, 736–737
journaling file systems, 569–570
just-in-time (JIT) compilers, 727
JVM, see Java Virtual Machine

K

K Desktop Environment (KDE), 60
kernel(s):
 computational, 835
kernel code, 96
kernel data structures, 31–34
 arrays, 31

bitmaps, 34
 hash functions and maps, 33–34
 lists, 31–33
 queues, 32
 stacks, 32
 trees, 31–33
kernel environment, 84
Kernel-Mode Driver Framework (KMDF), 856
kernel-mode threads (KT), 844
kernel modules:
 Linux, 96–101
kernel transaction manager (KTM), 862
KMDF (Kernel-Mode Driver Framework), 856
KT (kernel-mode threads), 844
KTM (kernel transaction manager), 862

L

latency:
 in real-time systems, 283–285
 target, 797
left child, 33
LFH design, 883–884
LIFO, 32
Linux:
 kernel modules, 96–101
Linux system(s):
 obtaining page size on, 370
lists, 31–32
live migration (virtual machines), 716, 733–735
lock(s):
 mutex, 212–214
loosely-coupled systems, 17
love bug virus, 694
low-fragmentation heap (LFH) design, 883–884
LPCs (local procedure calls), 834

M

Mac OS X operating system, 84
main memory:

 paging for management of:
 and Oracle SPARC Solaris, 383

memory:

 transactional, 239–240

memory leaks, 101

memory management:

 with virtual machines, 730–731

memory-management unit (MMU),
 384

micro TLBs, 388

migration:

 with virtual machines, 733–735

minimum granularity, 797

mobile computing, 36–37

mobile systems:

 multitasking in, 115

 swapping on, 360, 407

module entry point, 97

module exit point, 97

Moore's Law, 6, 835

multicore systems, 14, 16, 166

multifactor authentication, 689

multiprocessor systems (parallel
 systems, tightly coupled
 systems), 166

multi-touch hardware, 863

mutant (Windows 7), 841

mutex locks, 212–214

N

namespaces, 793

NAT (network address translation),
 732

nested page tables (NPTs), 720

network address translation (NAT),
 732

non-uniform memory access

 (**NUMA**), 834

NPTs (nested page tables), 720

O

**OLE (object linking and
 embedding)**, 873

open-file table, 546–547

OpenMP, 181–182, 240–241

OpenSolaris, 46

operating system(s):

 hybrid systems, 83–86

 portability of, 836–837

Oracle SPARC Solaris, 383

Orange Book, 832

OSI model, 757–758

OSI Reference Model, 682

overcommitment, 729

P

PaaS (platform as a service), 42

page address extension (PAE), 396

page directory pointer table, 386

page-frame number (PFN) database,
 850–851

page-table entries (PTEs), 847

paging:

 and Oracle SPARC Solaris, 383

parallelism, 166, 168–169

paravirtualization, 713, 725–726

partition(s):

 control, 723

PC systems, 863

PDAs (personal digital assistants), 11

periodic processes, 286

periodic task rate, 286

personal computer (PC) systems, 863

personalities, 83

PFF (page-fault-frequency), 429–430

PFN database, 850–851
platform as a service (PaaS), 42
pop, 32
POSIX:
 real-time scheduling, 290
POST (power-on self-test), 862
power manager (Windows 7),
 860–861
power-on self-test (POST), 862
priority-based scheduling, 285–287
private cloud, 42
privilege levels, 23
procedural languages, 241
process(es):
 background, 74–75, 115, 296
 foreground, 115, 296
 system, 8
processor groups, 835
process synchronization:
 alternative approaches to, 238–242
 functional programming
 languages, 241–242
 OpenMP, 240–241
 transactional memory, 239–240
 critical-section problem:
 software solution to, 212–213
programming-environment
 virtualization, 713, 726–727
proportional share scheduling,
 289–290
protection domain, 721
protocols:
 discovery, 39
pseudo-device driver, 730–731
PTEs (page-table entries), 847
PTE tables, 847
Pthreads:
 thread cancellation in, 186–187
public cloud, 41
public-key encryption, 678
push, 32

R

RAID sets, 868
rate, periodic task, 286
rate-monotonic scheduling, 287–288
rate-monotonic scheduling algorithm, 287–288
RC4, 677
RDP, 717
real-time CPU scheduling, 283–290
 earliest-deadline-first scheduling,
 288–289
 and minimizing latency, 283–285
 POSIX real-time scheduling, 290
 priority-based scheduling, 285–287
 proportional share scheduling,
 289–290
 rate-monotonic scheduling, 287–288
red-black trees, 35
resume, 715
right child, 33
ROM (read-only memory), 93, 480
routers, 754
RR scheduling algorithm, 271–273

S

SaaS (software as a service), 42
Scala language, 241–242
scheduling:
 earliest-deadline-first, 288–289
 priority-based, 285–287
 proportional share, 289–290
 rate-monotonic, 287–288
 SSDs and, 478
SCM (Service Control Manager), 860
second extended file system (ext2),
 811
security identity (SID), 853
security tokens, 853
Service Control Manager (SCM), 860

- services, operating system, 115
 session manager subsystem (SMSS), 862
SID (security identity), 853
 singly linked lists, 32
 SJF scheduling algorithm, 267–270
 Skype, 40
 slim reader-writer (SRW) locks, 879
 SLOB allocator, 439
 SLUB allocator, 439
 SMB (server-message-block), 871
 SMSS (session manager subsystem), 862
 software as a service (SaaS), 42
 solid-state disks (SSDs), 11, 469, 478
 SPARC, 383
 SRM (security reference monitor), 858–859
 SRW (slim reader-writer) locks, 879
 SSTF scheduling algorithm, 474–475
 standard swapping, 358–360
storage:
 thread-local, 187
storage management:
 with virtual machines, 732–733
subsystems, 135
superuser, 688
Surface Computer, 863
suspended state, 715
swapping:
 on mobile systems, 360, 407
 standard, 358–360
switching:
 fast-user, 863–864
symmetric coupling, 17
symmetric encryption algorithm, 676
synchronous threading, 172
SYSGEN, 91–92
system daemons, 8
system-development time, 715
system hive, 861
system processes, 8, 844–845
system restore point, 861
- T**
- target latency, 797
 task parallelism, 168–169
 TEBs (thread environment blocks), 880
 terminal applications, 96
 terminal server systems, 864
 thin clients, 35
 third extended file system (ext3), 811–813
threads:
 implicit threading, 177–183
thread attach, 853
thread environment blocks (TEBs), 880
thread-local storage, 187
thread pools, 179–181
thunking, 834
time sharing (multitasking), 115
time slice, 796
timestamp counters (TSCs), 840–841
touch screen (touchscreen computing), 5, 60
transactions:
 atomic, 210
transactional memory, 239–240
Transmission Control Protocol/Internet Protocol (TCP/IP), 758–761
trap-and-emulate method, 717–718
trees, 33, 35
TSCs (timestamp counters), 840–841
type 0 hypervisors, 712, 723–724
type 1 hypervisors, 712, 724–725
type 2 hypervisors, 713, 725

U

UAC (User Account Control), 701
UI (user interface), 52–55
UMDF (User-Mode Driver Framework), 856
UMS, *see user-mode scheduling*
USBs (universal serial buses), 469
User Account Control (UAC), 701
user mode, 787
User-Mode Driver Framework (UMDF), 856
user-mode scheduling (UMS), 296–297, 835, 880–881
user-mode threads (UT), 844
UT (user-mode threads), 844

V

VACB (virtual address control block), 857
variables:
 condition, 879
VAX minicomputer, 379–380
VCPU (virtual CPU), 717
vectored I/O, 603–604
virtual CPU (VCPU), 717
virtualization, 40–41
 advantages and disadvantages of, 714–716
 and application containment, 727–728
 and emulation, 727
 and operating-system components, 728–735
 CPU scheduling, 729
 I/O, 731–732
 live migration, 733–735
 memory management, 730–731
 storage management, 732–733

para-, 725–726
 programming-environment, 726–727

virtual machines, 711–738. *See also virtualization*

advantages and disadvantages of, 714–716
 and binary translation, 718–720
 examples, 735–737
 features of, 715–717
 and hardware assistance, 720–721
 history of, 713–714
 Java Virtual Machine, 736–737
 life cycle of, 722–723
 trap-and-emulate systems, 717–718
 type 0 hypervisors, 723–724
 type 1 hypervisors, 724–725
 type 2 hypervisors, 725
 VMware, 735–736

virtual machine control structures (VMCSs), 721

virtual machine manager (VMM), 22–23, 41, 712

virtual machine sprawl, 723

VMCSs (virtual machine control structures), 721

VMM, *see virtual machine manager*

VM manager, 846–852

VMware, 714, 735–736

W

wait0 system call, 120–122

Win32 API, 875

Windows 7:

dynamic device support, 837, 838
 and energy efficiency, 837
 fast-user switching with, 863–864
 security in, 700–701

- synchronization in, 833–834, 878–879
terminal services, 863–864
user-mode scheduling in, 296–297
- Windows executive:**
booting, 862–863
power manager, 860–861
- Windows group policy**, 875
- Windows Task Manager**, 87, 88
- Windows Vista**, 830
security in, 700
symbolic links in, 869–870
- Windows XP**, 830
- Winsock**, 881
- Workstation (VMWare)**, 735–736
- X**
- x86-64 architecture**, 387
- Xen**, 714
- Z**
- zones**, 728

