

A von Neumann architecture - S1.1, 1.26  
 Affinity Processor - 5.43  
 Amdahl's Law - 4.12  
 Android - 2.59, S2.4  
 Atomic variable, operation - 6.21, 7.14, S6.1  
 Buffering - 3.53  
 Caching - 1.36, H1.2.2  
 Chip multithreading - 5.40  
 CLI/GUI - 2.8  
 Client-server communication - 3.59, S3.4, H2.1.2  
 Cloud Computing - 1.52, S1.3  
 Concurrency vs Parallelism - 4.10, S4.1  
 Context Switch - 3.20, H1.18  
 CPU Burst - S5.3  
 CPU scheduler - 5.6, 5.45, S5.1, H2.1.5  
 Critical section - 6.9, S6.1  
 Deterministic Modeling - 5.52  
 Direct memory access (DMA) - 1.40, H1.1.3  
 Dispatcher - 5.7  
 Dual-mode operation - 3.21, S3.2, H1.2.6  
 Earliest Deadline First scheduling (EDF) - 5.49  
 First come first served (FCFS) - 5.10, S5.2  
 Hybrid Structure - 2.53  
 Hyperthreading - 5.40  
 I/O, Computer system Operation- 1.28, 1.39, S1.1  
 Inter-process Communication (IPC) - S3.4  
 Interprocess communication - 3.40  
 Interrupt - 1.30, S1.2, H1.1.2  
 Kernel threads - 4.19  
 Layered Structure - 2.45  
 Lightweight process (LWP) - S4.3  
 Linkers & Loaders - 2.33, S2.2, H1.1.6  
 Linux synchronization - 7.13  
 Load balancing - 5.44  
 Loadable kernel (LKM) - 2.50  
 Long-term scheduling - 3.17  
 Mac OS, Darwin - 2.56, S2.4  
 Medium-term scheduling - 3.18  
 Memory - 1.34, 3.7  
 Microkernel Structure - 2.47, S2.3  
 Modular Structure - 2.44, 2.50  
 Monolithic Structure - 2.41  
 Multicore processors (CMT) - 5.38  
 Multicore programming - 4.9  
 Multilevel feedback queue (MLFQ) - 5.28, 5.30, S5.3, H2.1.8  
 Multiprocessor - S1.1  
 Multiprogramming - 1.47, S1.2, H1.2.1  
 Multithreaded multicore system - 5.39  
 Multithreaded process, models - 4.14, 4.20, H2.1.4, H2.1.6  
 Mutex locks - 6.22  
 NUMA - 1.45  
 OS Design Implementation - 2.36, S2.4, H1.1.5  
 OS General - 1.20, H1.1.1  
 OS services - 2.4, S2.1  
 OS structure - 2.39, 2.54, S2.2  
 Parallelism - 4.11  
 Parameter Passing - 2.2  
 Pipes - 3.54, S3.5, H2.1.3  
 Policy and Mechanism - S2.2  
 POSIX - 7.16  
 Priority Scheduling - 5.25, 5.46, S5.3, H2.1.10  
 Process - 3.5, 3.9, S3.1  
 Process control blocks (PCB) - 3.10  
 Process creation (fork) - 3.27  
 Process state - 3.9  
 Process termination - 3.37, S3.3, H2.1.1  
 Process-contention scope (PCS) - 5.35  
 Processor - 1.41, S1.2  
 Protection - 2.35, S2.2  
 Race condition - 6.7  
 Rate Monotonic Scheduling - 5.47, H2.1.9  
 Real-time scheduling - S5.4  
 Remote Producer Calls (RPC) - 3.62  
 Round Robin (RR) - 5.12, 5.27, S5.2  
 Run time environment - 2.18  
 Scheduling - 5.8, H1.1.6, 3.17  
 Scheduling Multiple-Processor 5.36  
 Scheduling Process - 3.14  
 Scheduling Thread - 5.34, S5.4  
 Semaphore - 6.24, S6.2  
 Shortest time remaining first (SRTF) - 5.22  
 Short's job first (SJF) - 5.17, S5.2  
 Signal handling - 4.28, S4.3  
 Solaris synchronization - 7.11, S6.3  
 Spinlock - 6.22  
 Storage - 1.33, S1.1  
 Symmetric multiprocessing (SMP) 5.37, S5.5  
 Synchronization - 3.51  
 Synchronization tools - 6.13  
 System calls (API) - 2.13, S2.1, 3.33, H1.1.4, H1.2.3  
 System programs - 2.22  
 System programs - 2.29  
 System-contention scope (SCS) - 5.35  
 Thread Cancellation - 4.31  
 Thread control block (TCB) - 4.15  
 Threads - 3.11, S4.4, 4.15, S4.1, S4.3, H1.2.5  
 Threads Linux - 4.36  
 Threads Windows - 4.34  
 Timesharing - 1.47  
 User threads - 4.19, H2.1.7  
 Virtualization (VM) - 1.48, S1.3  
 Windows synchronization - 7.13