

Index

- \$(Q) variable, 550
 - /dev directory, 18
 - /etc/fstab files, 15
 - __builtin_expect() function, 70
 - __free_page() function, 194
 - __get_dma_pages() function, 193
 - __get_free_page() function, 193
 - __init macro, 68-69
 - __volatile__ modifier, 57-62
- A**
- ABI (Application Binary Interface), 40
 - absolute pathnames, 13, 299
 - Accelerated Graphics Port (AGP), 255
 - access
 - devices, 555-556
 - DMA, 281
 - rights, 15-20
 - activated field (task_struct structure), 90
 - active_list field (memory zones), 189
 - active_mm field (task_struct structure), 101
 - adding
 - caches, 213-222
 - code for system calls, 588-590
 - to wait queues, 136
 - addresses
 - devices, 255
 - fields (task_struct structure), 99-100
 - intervals, 226
 - linear spaces, 232-235, 454
 - logical, 454
 - memory, 433
 - mm_struct, 227-230
 - physical, 454
 - translation, 432, 438, 444
 - virtual, 182
 - vm_area_struct, 230-232
 - address_space structure, 331-333
 - add_wait_queue() function, 136
 - add_wait_queue_exclusive() function, 136
 - Advanced Programmable Interrupt Controller (APIC), 147
 - agetty programs, 12
 - AGP (Accelerated Graphics Port), 255
 - algorithms
 - big-o notations, 35
 - elevator, 263
 - aligning caches, 190
 - all_unreclaimable field (memory zones), 189
 - allocating memory
 - kmalloc() function, 224-225
 - kmem_cache_alloc() function, 225
 - allocators
 - global variables, 211-212
 - memory management, 200-209, 211-223
 - alloc_page() function, 192
 - alloc_pages() function, 192
 - anticipatory I/O schedulers, 263, 266
 - APIC (Advanced Programmable Interrupt Controller), 147
 - Application Binary Interface (ABI), 40
 - Application Specific Integrated Circuit (ASIC), 259
 - applications, 255
 - distributions, 5
 - Debian, 6
 - Fedora, 6-7
 - Gentoo, 7
 - Mandriva, 7
 - Red Hat, 6-7
 - SUSE, 7
 - Yellow Dog, 7
 - filesystems, 296-302
 - page caches, 330-334
 - VFS, 324-330. *See also* VFS
 - parallel ports, 285
 - ar command, 67
 - arch/ppc/ source code listings, 540-541
 - architecture
 - assembly language example, 46
 - PowerPC, 50-54
 - x86, 47-50
 - Big Endian/Little Endian, 42
 - CISC, 42
 - dependence, 26

- IHA, 257
- inline assembly, 55
 - asm keyword, 57
 - clobbered registers, 56
 - constraints, 56
 - input operands, 56
 - output operands, 55
 - parameter numbering, 56
 - __volatile__ modifier, 57-62
- memory initialization, 431
 - x86, 455
 - Intel-based, 444-454
 - PowerPC, 8, 431-432, 434-444
 - RISC, 39
 - source code, 538-541
 - UMA, 258
- areas, memory, 226, 232
- arithmetic instructions (x86), 45
- arrays
 - fields, 89
 - priority, 375
- ASIC (Application Specific Integrated Circuit), 259
- asm keyword, 57
- asmlinkage, 62
- assemblers, 526
- assembly
 - asm keyword, 57
 - inline, 55
 - clobbered registers, 56
 - constraints, 56
 - input operands, 56
 - output operands, 55
 - parameter numbering, 56
 - __volatile__ modifier, 57-62
 - languages, 38
 - example of, 46-54
 - PowerPC, 39-41
 - x86, 42-46
- asynchronous events, 143
- asynchronous execution flow, 142-152
- asynchronous I/O operations, 353
- atomic flags, 184. *See also* flags
- attributes
 - fields (task_struct structure), 84-86
 - files, 298
- B**
- Basic Input Output System (BIOS), 423
- BAT (Block Address Translation), 432, 436
- Bell Laboratories, 3
- Big Kernel Lock (BKL), 458
- Big-Endian, 42
- big-o notations, 35
- binary trees, 36-37
- binfmt field (task_struct structure), 86
- BIOS (Basic Input Output System), 423
- BKL (Big Kernel Lock), 458
- Block Address Translation (BAT), 432, 436
- blocks, 301
 - devices, 17, 260-261
 - disks, 301
 - state, 79, 116
- block_device_operations structure, 261
- blr (Branch to Link Register), 51
- boot loaders, 424
 - GRUB, 426-428
 - LILO, 429-430
 - PowerPC, 430-431
 - Yaboot, 430-431
- bottom-half interrupt handler methods, 150
- bouncing, 269
- Bourne shells, 4
- branch instructions (PowerPC), 41
- Branch to Link Register (blr), 51
- bridges, I/O, 255-258
- buddy systems (memory management), 194-199
- buffers
 - caches, 18
 - TLBs, 437
- buffer_head structure, 334
- buffer_init() function, calling, 497
- building
 - kernels
 - compilers, 527
 - cross compilers, 528
 - ELF object files, 529-535
 - linkers, 528
 - source build systems, 536-545, 547-551
 - toolchains, 526
 - parallel port drivers, 281-291
 - build_all_zonelists() function, calling, 474-475
- busses, I/O, 255-258
- C**
- C language usage, 62
 - asmlinkage, 62
 - const keyword, 64
 - inline keyword, 63
 - UL, 63
 - volatile keyword, 64
- caches
 - aligning, 190

- buffers, 18
- creating, 213-222
- descriptors, 203-208
- destroying, 222-223
- kmem_cache, 202
- pages, 261
 - address_space structures, 331-333
 - filesystems, 330-334
 - tracing, 357-365, 367-370
 - types of, 200
- cache_cache global variable, 211
- cache_chain global variable, 211
- cache_chain_sem global variable, 211
- cache_grow() function, 219-222
- cache_sizes descriptors, 202
- calculations, dynamic priority, 400
- calibrate_delay() function, calling, 494-495
- calling. *See also* system calls
 - buffer_init() function, 497
 - build_all_zonelists() function, 474-475
 - calibrate_delay() function, 494-495
 - console_init() function, 484
 - init_IRQ() function, 480-481
 - late_time_init() function, 493
 - local_irq_enable() function, 485
 - lock_kernel() function, 458-459
 - mem_init() function, 486-493
 - page_address_init() function, 460-463
 - page_alloc_init() function, 475-476
 - page_writeback_init() function, 509-511
 - parse_args() function, 476-478
 - pgtable_cache_init() function, 495-496
 - printk() function, 464
 - proc_root_init() function, 512-515
 - profile_init() function, 485
 - radix_tree_init() function, 508
 - rcu_init() function, 479
 - rest_init() function, 515
 - sched_init() function, 471-474
 - security_scaffolding_startup() function, 497
 - setup_arch() function, 464-469
 - setup_per_cpu_areas() function, 469-470
 - signals_init() function, 509
 - smp_prepare_boot_cpu() function, 470-471
 - softirq_init() function, 481
 - time_init() function, 482-483
 - trap_init() function, 479
 - vfs_cache_init() function, 498-507
- capabilities of task_struct structure, 94-96
- character devices, 18, 279, 298
- child processes, 21, 78
- children field (task_struct structure), 92
- chipsets, 255
- CHRP (Common Hardware Reference Platform), 423
- CISC (Complex Instruction Set Computing) architecture, 42
- clobbered registers, 56
- clocks, 411
 - devices, 280
 - real-time, 412-418
- clone() function, 105-106
- close() function, 345-350
- CML2, 547
- code
 - generation phases, 527
 - inline assembly, 55
 - asm keyword, 57
 - clobbered registers, 56
 - constraints, 56
 - input operands, 56
 - output operands, 55
 - parameter numbering, 56
 - __volatile__ modifier, 57-62
 - system calls, 588-590
 - traversing, 554-565, 567-575
 - writing, 575-585, 587-590
- coloring (slabs), 215
- comm field (task_struct structure), 86
- commands
 - ar, 67
 - hexdump, 66
 - objcopy, 67
- Common Hardware Reference Platform (CHRP), 423
- compilers, 526-527
 - asmlinkage, 62
 - cross, 528
- Complex Instruction Set Computing (CISC) architecture, 42
- components, MBR, 424
- compound pages, 187
- computer programs, 255. *See also* applications
- condition register (CR), 41
- configuration
 - caches, 213-222
 - devices, 575-585, 587-590
 - initrd, 486
 - kernel configuration tool, 543
- console_init() function, calling, 484
- const keyword, 64
- constants, marking, 63

constraints, 56
 context, 81, 150
 context_switch() function, 383-394
 control
 of bits, 438
 of files, 19
 terminals, 92
 transmitting information, 255
 controllers
 DMA, 281
 interrupts, 146
 count field (flags), 186
 count register (CTR), 41
 CPUs, yielding, 394-404
 cpus_allowed field (task_struct structure), 89
 CR (condition register), 41
 create_process program, 80-81
 credentials, 92-94
 cross compilers, 528
 cs_cachep field (cache descriptors), 207
 cs_dmacachep field (cache descriptors), 208
 cs_size field (cache descriptors), 207
 ctor field (cache descriptors), 206
 CTR (count register), 41
 current task structures, 387
 current variables, 82
 current working directories, 13

D

Data BAT (DBAT), 436
 data instructions (x86), 45
 data relocate (DR), 431
 data segments, 79
 data structures (VFS), 305-315, 317-322
 datatypes, 30
 linked lists, 30-34
 searching, 34-35
 trees, 35
 binary, 36-37
 red black, 38
 DBAT (Data BAT), 436
 deactivating tasks, 403
 dead processes, 23
 deadline I/O schedulers, 264
 deadlock, 409
 Debian, 6
 debugging device drivers, 590-591
 DECLARE_WORK() macro, 586
 declaring IOCTL numbers, 578-581
 decrementers, 155

defining execution contexts, 20
 defunct processes, 23
 dentry structures, 306, 315-317
 dependence, architecture, 26
 descriptors, 454
 caches, 203-208
 cache_sizes, 202
 files, 17, 300
 kmem_cache, 202
 memory zones, 187-189
 processes, 79-84
 address space fields, 99-100
 attribute fields, 84-86
 capabilities fields, 94-96
 credentials fields, 92-94
 filesystem fields, 99-100
 limitations fields, 97-99
 relationship fields, 90-92
 scheduling fields, 87-90
 destroying caches, 222-223
 devfs (Linux Device Filesystem), 260
 devices
 access, 555-556
 addressing, 255
 block, 17
 characters, 18, 298
 drivers, 25-26
 creating, 575-585, 587-590
 debugging, 590-591
 types of, 567-572
 files, 260, 298
 block devices, 260-261
 characters, 279
 clocks, 280
 DMA, 281
 generic block drivers, 274-276
 networks, 280
 operations, 277-278
 request queues, 263-273
 scheduling I/O, 263-273
 terminals, 280
 models (sysfs), 572-575
 pseudo, 18
 Direct Memory Access (DMA), 281
 direct store segments, 434
 directories, 13, 297
 current working, 13
 /dev, 18
 files, 17, 298-299
 fs/, 539

- home, 13
 - init/, 539
 - kernel/, 539
 - mm/, 539
 - Page Global Directory, 243
 - working, 299
 - dirty pages, flushing, 370-371
 - disks
 - blocks, 301
 - formatting, 301
 - initrd, 456
 - partitions, 301
 - distributions, 5
 - Debian, 6
 - Fedora, 6-7
 - Gentoo, 7
 - Mandriva, 7
 - Red Hat, 6-7
 - SUSE, 7
 - Yellow Dog, 7
 - DMA (Direct Memory Access), 281
 - dmesg tool, 68
 - do_exit() function, 117-120
 - do_fork() function, 106-109
 - do_page_fault() function, 238
 - DR (data relocate), 431
 - drivers
 - code, 554-565, 567-575
 - creating, 575-585, 587-590
 - debugging, 590-591
 - devices, 25-26
 - parallel ports, 281-291
 - tables, 261
 - types of, 567-572
 - wait queues, 559-564
 - work queues, 564-567
 - dtor field (cache descriptors), 206
 - dumb terminals, 280
 - dynamic libraries, 529
 - dynamic priority calculations, 400
- E**
- EA (effective address), 431
 - effective group IDs, 17
 - effective user IDs, 16
 - elevator algorithms, 263
 - ELF (Executable and Linking Format), 529-535
 - euid field (task_struct structure), 93
 - events, wait_event*() interfaces, 137-139
 - EXCEPTION() macro, 166
- exceptions
 - asynchronous execution flow, 143-151
 - page faults, 238
 - PowerPC page faults, 249
 - exec() system calls, 22
 - Executable and Linking Format (ELF), 529-535
 - executing
 - context of, 20, 81
 - processes, 80-81
 - adding to wait queues, 136
 - asynchronous execution flow, 142-151
 - clone() function, 105-106
 - creating, 101-102
 - do_exit() function, 117-120
 - do_fork() function, 106-109
 - fork() function, 103-104
 - lifespans, 109-116
 - sys_exit() function, 117
 - termination, 116
 - tracking, 124-133
 - vfork() function, 104
 - wait queues, 133-135
 - wait() function, 120-124
 - wait_event*() interfaces, 137-139
 - waking up, 140-142
 - schedulers, 375
 - context_switch() function, 383-394
 - selecting tasks, 376-383
 - yielding CPUs, 394-404
 - exit_code field (task_struct structure), 86
 - exit_signal field (task_struct structure), 86
 - exploration tools (kernels), 65
 - ar command, 67
 - hexdump command, 66
 - mm, 66
 - objcopy command, 67
 - objdump/readelf, 65
 - exporting symbols, 578
 - EXPORT_SYMBOL macro, 578
 - extensions, filenames, 297
 - external fragmentation, 194
 - external interrupts, 163
- F**
- faults (pages), 182, 237-249
 - fdatasync system calls, 18
 - Fedora, 6-7
 - fields
 - flags, 184-186
 - handlers, 153
 - memory zones, 187-189

- process descriptor
 - address space, 99-100
 - attributes, 84-86
 - capabilities, 94-96
 - credentials, 92-94
 - filesystem, 99-100
 - limitations, 97-99
 - relationship, 90-92
 - scheduling, 87-90
- superblock structures, 306-312
- task_struct structure
 - address space, 99-100
 - attribute, 84-86
 - capabilities, 94-96
 - credentials, 92-94
 - filesystem, 99-100
 - limitations, 97-99
 - relationship, 90-92
 - scheduling, 87-90
- filename extensions, 297
- files, 13, 296
 - attributes, 298
 - control, 19
 - descriptors, 17, 300
 - devices, 260, 298
 - block devices, 260-261
 - characters, 279
 - clocks, 280
 - DMA, 281
 - generic block drivers, 274-276
 - networks, 280
 - operations, 277-278
 - request queues, 263-273
 - scheduling I/O, 263-273
 - terminals, 280
- directories, 17, 298-299
- ELF, 529-535
- /etc/fstab, 15
- fields (task_struct structure), 100
- filenames, 297
- kernels, 13-20
- metadata, 17
- modes, 16
- operations, 300
- parameters, 350
- pathnames, 298-299
- processes
 - close() function, 345-350
 - files_struct structure, 326-330
 - fs_struct structure, 324
 - open() function, 337-344
 - regular, 17, 297
 - structures, 306, 318-322
 - types of, 17, 297-298
- filesystems, 13, 296
 - devfs, 260
 - fields (task_struct structure), 99-100
 - handlers, 273
 - hierarchies, 297
 - implementing, 301
 - kernels, 13-15
 - layers, 351-357
 - navigating, 555-556
 - overview, 296-302
 - page caches, 330-334
 - performance, 302
 - types of, 19
 - VFS, 273. *See also* VFS
- files_struct structure, 326-330
- first_time_slice field (task_struct structure), 90
- fixed-point instructions (PowerPC), 41
- flags, 153, 184-186, 428
 - cache descriptors, 205
 - task_struct structure, 85
- Flash, 423
- flips, 556-558
- floating-point instructions (PowerPC), 41
- flops, 556-558
- flushing dirty pages, 370-371
- fork() function, 103-104
- fork() system calls, 21
- forked processes, 21
- formatting. *See also* configuration
 - caches, 213-222
 - devices, 575-585, 587-590
 - disks, 301
- for_each_zone() function, 190
- fragmentation, external, 194
- frames
 - memory management, 191-199
 - pages, 182
- free field (slab descriptors), 209
- free software, 5
- Free Software Foundation (FSF), 4
- free_area field (memory zones), 189
- free_page() function, 194
- free_pages field (memory zones), 188
- front-side busses, 255
- fs field (task_struct structure), 100
- fs/ directory, 539
- FSF (Free Software Foundation), 4
- fsuid field (task_struct structure), 94

fsuid field (task_struct structure), 94
 fsync system calls, 18
 fs_struct structure, 324
 functions. *See also* commands
 add_wait_queue(), 136
 add_wait_queue_exclusive(), 136
 alloc_page(), 192
 alloc_pages(), 192
 __builtin_expect(), 70
 cache_grow(), 219-222
 close(), 345-350
 context_switch(), 383-394
 do_exit(), 117-120
 do_page_fault(), 238
 for_each_zone(), 190
 free_page(), 194
 __get_dma_pages(), 193
 __get_free_page(), 193
 helper (memory zones), 190
 is_highmem(), 190
 is_normal(), 190
 kmallocc(), 224-225
 kmem_cache_alloc(), 225
 kmem_cache_destroy(), 222-223
 likely(), 69-70
 list_del(), 32
 nice(), 24, 88
 open(), 337-344
 printk(), 67
 process creation, 101-102
 clone() function, 105-106
 do_fork() function, 106-109
 fork() function, 103-104
 vfork() function, 104
 releases, 193
 requests, 191-193
 scheduler_tick(), 394
 sched_fork(), 126-133
 start_kernel(), 456-457
 buffer_init() function, 497
 build_all_zonelists() function, 474-475
 calibrate_delay() function, 494-495
 console_init() function, 484
 init_IRQ() function, 480-481
 late_time_init() function, 493
 local_irq_enable() function, 485
 lock_kernel() function, 458-459
 mem_init() function, 486-493
 page_address_init() function, 460-463
 page_alloc_init() function, 475-476
 page_writeback_init() function, 509-511

 parse_args() function, 476-478
 pgtable_cache_init() function, 495-496
 printk() function, 464
 proc_root_init() function, 512-515
 profile_init() function, 485
 radix_tree_init() function, 508
 rcu_init() function, 479
 rest_init() function, 515
 sched_init() function, 471-474
 security_scaffolding_startup() function, 497
 setup_arch() function, 464-469
 setup_per_cpu_areas() function, 469-470
 signals_init() function, 509
 smp_prepare_boot_cpu() function, 470-471
 softirq_init() function, 481
 time_init() function, 482-483
 trap_init() function, 479
 vfs_cache_init() function, 498-507
 switch(), 392
 switch_to(), 26, 385
 synchronous, 261
 sys_exit(), 117
 unlikely(), 69-70
 wait(), 120-124

G

GEC (General Electric Company), 3
 general purpose caches, 200
 general-purpose registers (GPRs), 41
 generic block device layers, 261, 274
 generic block driver layers, 274-276
 Gentoo, 7
 geometry of hard drives, 301
 gfpflags field (cache descriptors), 206
 gfporder field (cache descriptors), 206
 gfp_mask integer value, 196
 GID (group ID), 13, 22
 global variables
 local list references, 322-324
 slab allocators, 211-212
 GMCH (Graphics and Memory Controller Hub), 257
 GPL (GNU General Public License), 5
 GPRs (general-purpose registers), 41
 Grand Unified Bootloader (GRUB), 426-428
 Graphics and Memory Controller Hub (GMCH), 257
 group ID (GID), 13, 22
 group_info field (task_struct structure), 94
 group_leader field (task_struct structure), 92
 GRUB (Grand Unified Bootloader), 426-428

H

handlers
 filesystems, 273
 page faults, 239-247
 hard drives, geometry of, 301
 hard links, 18

hardware
 I/O, 255-258
 parallel ports, 282

headers
 ELF, 530-531
 tables, 530
 programs, 534-535
 sections, 532

heads, 301
 heaps, 79
 helper functions, memory zones, 190
 Hertz (HZ), 156
 Hertz, Heinrich, 156
 hexdump command, 66
 hierarchies, filesystems, 297
 High Performance Event Timer (HPET), 482
 history of UNIX, 2-4
 home directories, 13
 host systems, 528
 HPET (High Performance Event Timer), 482
 hubs, 257
 hw_interrupt_type structure, 154
 hw_irq_controller structure, 154
 HyperTransport technology, 258
 HZ (Hertz), 156

I

I/O (input/output), 254
 asynchronous operations, 353
 devices
 block devices, 260-261
 characters, 279
 clocks, 280
 DMA, 281
 files, 260
 generic block drivers, 274-276
 networks, 280
 operations, 277-278
 request queues, 263-273
 scheduling, 263-273
 terminals, 280
 hardware, 255-258
 I/O Controller Hub (ICH), 257
 IBAT (Instruction BAT), 436

ICH (I/O Controller Hub), 257
 IDT (Interrupt Descriptor Table), 149, 165
 IHA (Intel Hub Architecture), 257
 images, building kernels, 542-545
 implementing filesystems, 301
 implicit kernel preemption, 407-409
 implicit user preemption, 405-406
 inactive_list field (memory zones), 189
 inb (read in a byte), 256
 index nodes, 17
 init process, 21-22, 456
 init threads (Process 1), 517-522
 init/ directory, 539
 initial RAM disk (initrd), 456, 486
 initialization
 architecture-dependent memory, 431
 PowerPC, 431-441
 x86, 455
 Intel-based, 444-454
 irqaction struct, 162
 kernels, 422
 systems, 422
 initrd (initial RAM disk), 456, 486
 init_IRQ() function, calling, 480-481
 inline assemblies, 55
 asm keyword, 57
 clobbered registers, 56
 constraints, 56
 input operands, 56
 output operands, 55
 parameter numbering, 56
 __volatile__ modifier, 57-62
 inline keyword, 63
 inode structures, 17, 301, 305-315
 input operands, 56
 input/output. *See* I/O
 Instruction BAT (IBAT), 436
 instruction relocate (IR), 431
 Intel Hub Architecture (IHA), 257
 interactive processes, 378
 interactive tasks, 398
 interactive_credit field (task_struct structure), 89
 interfaces, 4
 ABI, 40
 I/O, 255-258
 users, 12
 wait_event*(), 137-139
 Interprocess Communication (IPC), 298
 Interrupt Descriptor Table (IDT), 149, 165
 interrupt-acknowledge cycle, 147
 interrupts, 146-155, 423, 564-567

- context, 150
 - controllers, 146
 - polling and, 582-585
 - intervals, addresses, 226
 - inuse field (slab descriptors), 209
 - IOCTL numbers, declaring, 578-581
 - IPC (Interprocess Communication), 298
 - IR (instruction relocate), 431
 - IRQ structures, 151
 - irqaction struct, 153, 162
 - irq_desc_t structure, 151
 - IS_ERR macro, 71
 - is_highmem() function, 190
 - is_normal() function, 190
- J–K**
- jiffies, 398
- kernels**
- architecture-dependent memory
 - initialization, 431
 - PowerPC, 431-441
 - x86, 455
 - Intel-based, 444-454
 - boot loaders, 424
 - GRUB, 426-428
 - LILO, 429-430
 - PowerPC, 430-431
 - Yaboot, 430-431
 - configuration tool, 543
 - create_process program, 80-81
 - datatypes, 30
 - linked lists, 30-34
 - searching, 34-35
 - trees, 35-38
 - directory, 539
 - distributions, 5
 - Debian, 6
 - Fedora, 6-7
 - Gentoo, 7
 - Mandriva, 7
 - Red Hat, 6-7
 - SUSE, 7
 - Yellow Dog, 7
 - explicit kernel preemption, 405
 - exploration tools, 65
 - ar command, 67
 - hexdump command, 66
 - mm, 66
 - objcopy command, 67
 - objdump/readelf, 65
 - implicit kernel preemption, 407-409
 - init threads (Process 1), 517-522
 - initialization, 422
 - memory, 558-559
 - messages, 67
 - dmesg, 68
 - /var/log/, 68
 - mode, 80
 - organization, 11
 - overview of, 11
 - access rights, 15-20
 - device drivers, 25-26
 - files/filesystems, 13-15
 - processes, 20-23
 - schedulers, 24
 - system calls, 24
 - UID, 13
 - user interfaces, 12
 - printk() function, 67
 - release information, 8
 - source build systems, 536-537
 - architecture-dependent source code, 540-541
 - architecture-independent source code, 538-539
 - images, 542-545
 - Linux makefiles, 548-551
 - sub-makefiles, 545-548
 - space, 10
 - start_kernel() function, 456-457
 - buffer_init() function, 497
 - build_all_zonelists() function, 474-475
 - calibrate_delay() function, 494-495
 - console_init() function, 484
 - init_IRQ() function, 480-481
 - late_time_init() function, 493
 - local_irq_enable() function, 485
 - lock_kernel() function, 458-459
 - mem_init() function, 486-493
 - page_address_init() function, 460-463
 - page_alloc_init() function, 475-476
 - page_writeback_init() function, 509-511
 - parse_args() function, 476-478
 - phtable_cache_init() function, 495-496
 - printk() function, 464
 - proc_root_init() function, 512-515
 - profile_init() function, 485
 - radix_tree_init() function, 508
 - rcu_init() function, 479
 - rest_init() function, 515
 - sched_init() function, 471-474
 - security_scaffolding_startup() function, 497

setup_arch() function, 464-469
 setup_per_cpu_areas() function, 469-470
 signals_init() function, 509
 smp_prepare_boot_cpu() function, 470-471
 softirq_init() function, 481
 time_init() function, 482-483
 trap_init() function, 479
 vfs_cache_init() function, 498-507
 toolchains, 526
 compilers, 527
 cross compilers, 528
 ELF object files, 529-535
 linkers, 528
 keywords
 asm, 57
 const, 64
 inline, 63
 volatile, 64
 kmalloc() function, 224-225
 kmem_cache descriptors, 202
 kmem_cache_alloc() function, 225
 kmem_cache_destroy() function, 222-223

L

languages

assembly, 38
 example of, 46-54
 PowerPC, 39-41
 x86, 42-46

C, 62

asmlinkage, 62
 const keyword, 64
 inline keyword, 63
 UL, 63
 volatile keyword, 64

latency, 261

late_time_init() function, calling, 493

layers

filesystems, 351-357
 generic block device, 261, 274
 generic block drivers, 274-276

layouts, 537. *See also* configuration; formatting

li RT, RS, SI (Load Immediate), 51

libraries, 529

licenses, GPL, 5

lifecycles of slab allocators, 211-223

lifespans of processes, 109

descriptors, 82-84
 states, 109-116

likely() function, 69-70

LILO (LIInux LOader), 429-430

limitations of fields, 97-99

linear address spaces, memory management, 232-235

linear addresses, 454

link register (LR), 41

linked lists, 30-34

linkers, 528

links, 18-19, 298, 526

Linux

filesystems. *See* filesystems

makefiles, 548-551

process structures

linear address spaces, 232-235

memory management, 226-232

pages, 236-249

Linux Device Filesystem (devfs), 260

LIInux LOader (LILO), 429-430

Linux Power, 8

lists

cache descriptors, 203

clobber, 56

fields

flags, 186

slab descriptors, 204, 208

linked, 30-34

local references (global variables and), 322-324

searching, 34-35

work queues, 34

lists.next_reap, 204

lists.slabs_free, 204

lists.slabs_full, 204

lists.slabs_partial, 204

list_del() function, 32

Little Endian, 42

Load Immediate (li RT, RS, SI), 51

Load Word and Zero (lwz RT, D(RA)), 51

local list references, 322-324

local stacks, asmlinkage, 62

local_irq_enable() function, calling, 485

lock field (memory zones), 188

locking spinlocks, 409-411

lock_kernel() function, calling, 458-459

logical addresses, 454

logical disks, 301

login programs, 12

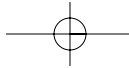
LR (link register), 41

lru field (flags), 186

lru_lock field (memory zones), 189

ls /usr/src/linux/arch, 538

lwz RT, D(RA) (Load Word and Zero), 51

**M**

Machine State Register (MSR), 431

macros

DECLARE_WORK(), 586

EXCEPTION(), 166

EXPORT_SYMBOL, 578

IS_ERR, 71

PTR_ERR, 68-71

makefiles

Linux, 548-551

sub-makefiles, 545-548

malloc_sizes[] global variable, 211

Mandriva, 7

mapping

fields (flags), 186

memory processes, 250-251

marking constants, 63

MBR (Master Boot Record), 424

MCH (Memory Controller Hub), 257

memory

addresses

mm_struct, 227-230

vm_area_struct, 230-232

addressing, 433

architecture-dependent initialization, 431

PowerPC, 431-441

x86, 455

Intel-based, 444-454

areas, 226, 232

buffer_head structures, 334

DMA, 281

initrd, 456

kernels, 558-559

kmalloc() function, 224-225

kmem_cache_alloc() function, 225

management, 180-183

linear address spaces, 232-235

Linux process structures, 226-232

request paths, 224-225

slab allocators, 200-209, 211-223

zones, 187-190

manager, 183

pages, 183-186

faults, 237-249

frames, 191-199

tables, 236-237

process mapping, 250-251

regions, 226

users, 558-559

virtual, 181

Memory Controller Hub (MCH), 257

Memory Management Unit (MMU), 183

memory-mapped I/O, 257

mem_init() function, calling, 486-493

messages (kernels), 67

dmesg, 68

printk() function, 67

/var/log/messages, 68

metadata files, 17

mingetty programs, 12

Minix, 4

MIT, 3

mm field (task_struct structure), 101

mm utility, 66

mm/ directory, 539

MMU (Memory Management Unit), 183

mm_struct structure, 227-230

models, sysfs and devices, 572-575

modes

files, 16

kernel, 80

sgid, 16

sticky, 16

suid, 16

user, 80

modifiers, __volatile, 57-62

modules, 281, 554-565, 567-575

monolithic systems, 10

mount points, 301

mount systems, 15

MSR (Machine State Register), 431

Multiboot Specification (GRUB), 427

MULTiplexed Information and Computing Service (MULTICS), 3

multiprogramming, 3

multiuser timesharing, 3

N

name field (cache descriptors), 207

named pipes, 19

navigating filesystems, 555-556

network devices, 280

next field (cache descriptors), 207

nice() function, 24, 88

nivcsw field (task_struct structure), 90

no-op, 157, 264

nodes, index, 17

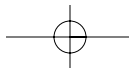
non-executable ELF file sections, 533

non-volatile storage, 296

Northbridge, 255-257

notations, big-o, 35

notification of parents, 120-124



notifier chains, 71
 num field (cache descriptors), 205
 numbering
 IOCTL, 578-581
 parameters, 56
 nvcsw field (task_struct structure), 90

O

O(1) schedulers, 375
 context_switch() function, 383-394
 CPUs, 394-404
 tasks, 376-383
 objcopy command, 67
 objdump utility, 65
 objects, 30
 create_process program, 80-81
 ELF, 529-535
 file formats, 297
 languages, 526
 linked lists, 30-34
 searching, 34-35
 trees, 35
 binary, 36-37
 red black, 38
 objsize field (cache descriptors), 205
 OF (Open Firmware), 423, 430
 offsetting descriptors, 454
 offsetting file parameters, 350
 Open Firmware (OF), 423, 430
 Open Programmable Interrupt Controller (OpenPIC), 148
 open source software, 5
 open() function, 337-344
 OpenPIC (Open Programmable Interrupt Controller), 148
 operating systems
 create_process program, 80-81
 overview of, 9-11
 operations
 asynchronous I/O, 353
 devices, 277-278
 files, 300
 superblock structures, 310-312
 optimizers, 527
 optimizing filesystems, 302
 organization of kernels, 11
 outb (write out a byte), 256
 output operands, 55
 overview of Linux, 2

P

padding zones, 190
 Page Directory Entry (PDE), 454
 Page Global Directory, 243
 Page Table Entry (PTE), 434, 454
 pages, 182-183
 caches, 261
 address_space structures, 331-333
 filesystems, 330-334
 tracing, 357-365, 367-370
 compound, 187
 dirty, 370-371
 faults, 182, 237-244, 246-249
 flags, 184-186
 frames, 182, 191-199
 memory management, 183-186
 tables, 183, 236-237
 page_address_init() function, calling, 460-463
 page_alloc_init() function, calling, 475-476
 page_writeback_init() function, calling, 509-511
 pages_high field (memory zones), 189
 pages_min, pages_low field (memory zones), 189
 pages_scanned, temp_priority field (memory zones), 189
 parallel port drivers, building, 281-291
 parameters
 asmlinkage, 62
 files, 350
 numbering, 56
 parents
 field (task_struct structure), 92
 notification, 120-124
 processes, 21, 78
 parse_args() function, calling, 476-478
 partitions, 13, 301
 pathnames, 13, 298-299
 path requests, 224-225
 PCI busses, 255
 PDE (Page Directory Entry), 454
 pdeath field (task_struct structure), 86
 performance, filesystems, 302
 pgtable_cache_init() function, calling, 495-496
 phases of compiling, 527
 physical addresses, 454
 PIC (Programmable Interrupt Controller), 154
 PID (process ID), 22
 pid field (task_struct structure), 85
 pipes, 19, 298
 PIT (Programmable Interval Time), 155
 pivoting the root, 456

- plugging, 268
 - policy field (task_struct structure), 89
 - polling and interrupts, 582-585
 - portability, 26
 - ports
 - I/O, 255-258
 - parallel drivers, 281-291
 - PowerPC
 - architecture-dependent memory initialization, 431-444
 - assembly languages, 39-41, 50-54
 - bootloaders, 430-431
 - Linux Power, 8
 - page fault exceptions, 249
 - x86 code convergence, 455
 - PowerPC Reference Platform (PreP), 423
 - preemption of tasks, 405
 - explicit kernel, 405
 - implicit kernel, 407-409
 - implicit user, 405-406
 - PreP (PowerPC Reference Platform), 423
 - prev_priority field (memory zones), 189
 - principle of locality, 228
 - printk() function, 67, 464
 - priority
 - arrays, 375
 - dynamic calculations, 400
 - processes, 24
 - Process 1 (init threads), 517-522
 - process ID (PID), 22
 - process status (ps), 23
 - processes, 20-23, 78, 456
 - asynchronous execution flow, 142-151
 - clone() function, 105-106
 - context, 150
 - create_process program, 80-81
 - creating, 21, 101-102
 - dead, 23
 - descriptors, 79-84
 - address space fields, 99-100
 - attribute fields, 84-86
 - capabilities fields, 94-96
 - credentials fields, 92-94
 - filesystem fields, 99-100
 - limitations fields, 97-99
 - relationship fields, 90-92
 - scheduling fields, 87-90
 - do_fork() function, 106-109
 - files
 - close() function, 345-350
 - files_struct structure, 326-330
 - fs_struct structure, 324
 - open() function, 337-344
 - fork() function, 103-104
 - init, 456
 - interactive, 378
 - lifespans, 109-111
 - Linux, 226-232
 - memory, 250-251
 - priority, 24
 - running, 79
 - schedulers, 376
 - sleeping, 23
 - spawning, 21
 - termination, 116
 - do_exit() function, 117-120
 - sys_exit() function, 117
 - wait() function, 120-124
 - tracking, 124-133
 - transitions, 111-116
 - types of, 78
 - vfork() function, 104
 - wait queues, 133-135
 - adding to, 136
 - wait_event*() interfaces, 137-139
 - waking up, 140-142
 - zombie, 115
 - proc_root_init() function, calling, 512-515
 - profile_init() function, calling, 485
 - Programmable Interrupt Controller (PIC), 154
 - Programmable Interval Time (PIT), 155
 - programming filesystems, 296-302
 - page caches, 330-334
 - VFS, 324-330. *See also* VFS
 - programs. *See* applications
 - protected mode (memory management), 449-450
 - protecting files, 15-20
 - ps (process status), 23
 - pseudo devices, 18
 - PTE (Page Table Entry), 434, 454
 - ptrace field (task_struct structure), 86
 - PTR_ERR macro, 71
- Q**
- queues
 - requests, 263-274
 - run, 22
 - system requests, 261
 - wait, 133-135, 559-564
 - adding to, 136

- wait_event*() interfaces, 137-139
- waking up, 140-142
- work, 564-567
 - lists, 34
 - tasklets, 586-588

R

- radix_tree_init() function, calling, 508
- RAM (random access memory), initrd, 456
- rcu_init() function, calling, 479
- readelf utility, 65
- reading
 - PPC real-time clocks, 414
 - real-time clocks (x86), 417
- ready state, 79
- ready to running state transitions, 112
- real addressing, 433
- real group IDs, 17
- real mode, 431
- real user IDs, 16
- real-time clocks, 412-418
- real_parent field (task_struct structure), 91
- receiving data from devices, 255
- red black trees, 38
- Red Hat, 6-7
- Reduced Instruction Set Computing (RISC)
 - architecture, 39
- references, 322-324
- regions, memory, 226
- registers
 - clobbered, 56
 - PowerPC, 41
 - segment, 434
 - SPRs, 436
- regular files, 17, 297
- relationships
 - fields, 90-92
 - makefiles, 548-551
- relative pathnames, 13, 299
- release information (kernels), 8
- releases, 193
- relocation, 528
- requests
 - functions, 191-193
 - paths, 224-225
 - queue utilities, 274
 - queues, 263-274
 - system queues, 261
- respawning programs, 12
- rest_init() function, calling, 515
- rights (access), 15-20

- RISC (Reduced Instruction Set Computing)
 - architecture, 39
- Ritchie, Dennis, 3
- rlim field (task_struct structure), 97
- root of users, 13
- root threads, 456
- rt_priority field (task_struct structure), 90
- rules, schedulers, 20
- run queues, 22, 125
- runnable states (processes), 22
- running processes, 79
- run_list field (task_struct structure), 88

S

- scanner phases, 527
- schedulers, 24, 79
 - anticipatory, 263
 - creating, 124-133
 - deadline I/O, 264
 - no-op I/O, 264
 - O(1), 375
 - context_switch() function, 383-394
 - selecting tasks, 376-383
 - yielding CPUs, 394-404
 - rules, 20
- scheduler_tick() function, 394
- scheduling
 - fields (task_struct structure), 87-90
 - I/O, 263-273
- sched_fork() function, 126-133
- sched_init() function, calling, 471-474
- scripts, 297
- SDR1 (Storage Description Register 1), 434
- searching datatypes, 34-35
- sections, 530
 - header tables, 532
 - non-executable ELF files, 533
- security_scaffolding_startup() function, calling, 497
- Segment Registers, 434
- Segmented Address Translation, 432
- segments, 79, 530
- selecting task schedulers, 376-383
- semantic attributes, 527
- semaphores, 409-411
- setup_arch() function, calling, 464-469
- setup_per_cpu_areas() function, calling, 469-470
- sgid
 - fields (task_struct structure), 94
 - modes, 16
- shared libraries, 529

- siblings
 - fields (task_struct structure), 92
 - processes, 78
- signals_init() function, calling, 509
- SIGSTOP, 115
- slabp_cache field (cache descriptors), 206
- slabs
 - allocators
 - global variables, 211-212
 - memory management, 200-209, 213-223
 - coloring, 215
- sleeping, processes, 23
- sleep_avg field (task_struct structure), 89
- smp_prepare_boot_cpu() function, calling, 470-471
- sockets, 19, 298
- soft links, 18
- softirq_init() function, calling, 481
- software. *See* applications
- source build systems, 536-537
 - architecture-dependent source code, 540-541
 - architecture-independent source code, 538-539
 - images, 542-545
 - Linux makefiles, 548-551
 - sub-makefiles, 545-548
- source code
 - generation phases, 527
 - inline assembly, 55
 - asm keyword, 57
 - clobbered registers, 56
 - constraints, 56
 - input operands, 56
 - output operands, 55
 - parameter numbering, 56
 - __volatile__ modifier, 57-62
 - system calls, 588-590
 - traversing, 554-565, 567-575
 - writing, 575-585, 587-590
- Southbridge, 255-257
- spaces
 - addresses, 232-235
 - kernels, 10
 - users, 10
 - virtual addresses, 182
- spawning processes, 21
- special purpose registers (SPRs), 436
- specialized caches, 200
- spinlocks, 409-411
- SPRs (special purpose registers), 436
- stacks, 62, 79
- standards, 4
- start_kernel() function, 456-457
 - buffer_init() function, 497
 - build_all_zonelists() function, 474-475
 - calibrate_delay() function, 494-495
 - console_init() function, 484
 - init_IRQ() function, 480-481
 - late_time_init() function, 493
 - local_irq_enable() function, 485
 - lock_kernel() function, 458-459
 - mem_init() function, 486-493
 - page_address_init() function, 460-463
 - page_alloc_init() function, 475-476
 - page_writeback_init() function, 509-511
 - parse_args() function, 476-478
 - pgtable_cache_init() function, 495-496
 - printk() function, 464
 - proc_root_init() function, 512-515
 - profile_init() function, 485
 - radix_tree_init() function, 508
 - rcu_init() function, 479
 - rest_init() function, 515
 - sched_init() function, 471-474
 - security_scaffolding_startup() function, 497
 - setup_arch() function, 464-469
 - setup_per_cpu_areas() function, 469-470
 - signals_init() function, 509
 - smp_prepare_boot_cpu() function, 470-471
 - softirq_init() function, 481
 - time_init() function, 482-483
 - trap_init() function, 479
 - vfs_cache_init() function, 498-507
- state, 78
 - fields (task_struct structure), 85
 - lifespans, 109, 111
 - ready, 79
 - transitions, 111-116
- static libraries, 529
- statically allocated major devices, 260
- static_prio field (task_struct structure), 88
- status, processes, 23
- sticky mode, 16
- Storage Description Register 1 (SDR1), 434
- Store Word with Update (stwu RS, D(RA)), 51
- structures
 - address_space, 331-333
 - block_device_operations, 261
 - buffer_head, 334
 - current task, 387
 - dentry, 315-317
 - files, 318-322
 - files_struct, 326-330

- fs_struct, 324
 - hw_interrupt_type, 154
 - hw_irq_controller, 154
 - inode, 301-315
 - IRQ, 151
 - irq_desc_t, 151
 - processes (Linux), 226-232
 - superblock, 306-312
 - task_struct, 82-84
 - address space fields, 99-100
 - attribute fields, 84-86
 - capabilities fields, 94-96
 - credentials fields, 92-94
 - filesystem fields, 99-100
 - limitations fields, 97-99
 - relationship fields, 90-92
 - scheduling fields, 87-90
 - VFS, 324-330
 - wait queues, 133-136
 - stwu RS, D(RA) (Store Word with Update), 51
 - sub-makefiles, 545-548
 - subdirectories, architecture-independent, 538
 - suid field (task_struct structure), 94
 - suid mode, 16
 - superblock structures, 305-312
 - Superio chips, 257
 - superusers, 13
 - super_operations structure, 310
 - SUSE, 7
 - switch() function, 392
 - switching tasks, 405
 - explicit kernel preemption, 405
 - implicit kernel preemption, 407-409
 - implicit user preemption, 405-406
 - switch_to() function, 26, 385
 - symbols
 - exporting, 578
 - links, 18
 - resolution, 529
 - sync system calls, 18
 - synchronous functions, 261
 - synchronous interrupts, 143
 - syntactical rules, 527
 - sysfs, device models and, 572-575
 - systems
 - calls, 24, 101-102, 143-146
 - clone() function, 105-106
 - code, 588-590
 - do_fork() function, 106-109
 - fork() function, 103-104
 - source code, 554-565, 567-575
 - vfork() function, 104
 - VFS, 336-342. *See also* VFS
 - clocks, 411-418
 - request queues, 261
 - timers, 155-163, 166-173
 - sys_exit() function, 117
- T**
- tables
 - drivers, 261
 - headers, 530
 - programs, 534-535
 - sections, 532
 - pages, 183, 236-237
 - Tanenbaum, Andrew, 4
 - target system, 528
 - tasklets, work queues and, 586-588
 - tasks
 - current task structure, 387
 - deactivating, 403
 - interactive, 398
 - preemption, 405
 - explicit kernel, 405
 - implicit kernel, 407-409
 - implicit user, 405-406
 - schedulers, 375
 - context_switch() function, 383-394
 - selecting, 376-383
 - yielding CPUs, 394-404
 - system clocks, 411-418
 - TASK_INTERRUPTIBLE state, 113
 - task_list, 82
 - TASK_RUNNING state, 113
 - TASK_STOPPED state, 116
 - task_struct structure, 82-84
 - address space, 99-100
 - attributes, 84-86
 - capabilities, 94-96
 - credentials, 92-94
 - filesystem, 99-100
 - limitations, 97-99
 - relationship, 90-92
 - scheduling, 87-90
 - TASK_UNINTERRUPTIBLE state, 114
 - TASK_ZOMBIE state, 115
 - terminals, 12, 280
 - termination processes, 116
 - do_exit() function, 117-120
 - sys_exit() function, 117
 - wait() function, 120-124
 - text segments, 79

- Thompson, Ken, 3
- threads, init (Process 1), 517-522
- timers, 411
 - real-time clocks, 412-418
 - system, 155-163, 166-173
- timesharing users, 3
- timeslices, 125, 374, 378
- timestamps
 - fields (task_struct structure), 89
 - schedulers, 378
- time_init() function, calling, 482-483
- time_slice field (task_struct structure), 89
- TLBs (Translation Lookaside Buffers), 437
- toolchains, 526
 - compilers, 527-528
 - ELF object files, 529-535
 - linkers, 528
- tools
 - distributions, 5
 - Debian, 6
 - Fedora, 6-7
 - Gentoo, 7
 - Mandriva, 7
 - Red Hat, 6-7
 - SUSE, 7
 - Yellow Dog, 7
 - dmesg, 68
 - exploration (kernels), 65
 - ar command, 67
 - hexdump command, 66
 - mm, 66
 - objcopy command, 67
 - objdump/readelf, 65
 - kernel configuration, 543
- top-half interrupt handler methods, 150
- Torvalds, Linus, 2
- tracing page caches, 357-365, 367-370
- tracking processes, 124-133
- tracks, 301
- transitions, process state, 110-116
- translation (addresses), 432
 - PPC, 438
 - x86 Intel-based, 444-454
- Translation Lookaside Buffers (TLBs), 437
- transmitting control information, 255
- traps, 157
- trap_init() function, calling, 479
- traversing source code, 554-565, 567-575
- trees, 35
 - binary, 36-37
 - red black, 38
- troubleshooting
 - device drivers, 590-591
 - filesystems, 302
- types
 - of drivers, 567-572
 - of files, 17, 297-298
 - of filesystems, 19
 - of interrupt handlers, 149
- U**
- UID (user ID), 13
- UL (unsigned long), 63
- UMA (Universal Motherboard Architecture), 258
- UNIX, history of, 2-4
- unlikely() function, 69-70
- unplugging, 268
- unsigned long (UL), 63
- user ID (UID), 13
- users
 - implicit user preemption, 405-406
 - interfaces, 12
 - mode, 80
 - space, 10
 - superusers, 13
- utilities. *See* tools
- V**
- VA (virtual address), 182, 434
- Values, flags, 184
- variables
 - \$(Q) variable, 550
 - current, 82
 - global
 - local list references, 322-324
 - slab allocators, 211-212
 - HZ, 155
- vectors, 149
- versions, kernels, 8
- vfork() function, 104
- VFS (virtual filesystem), 273
 - structures, 324-330
 - system calls, 336-371
- vfs_cache_init() function, calling, 498-507
- virtual address (VA), 182, 434
- virtual field (flags), 186
- virtual filesystem. *See* VFS
- virtual memory, 181
- virtual page number (VPN), 434
- virtual segment ID (VSID), 434
- virtual terminals, 12
- vm_area_struct structure, 230-232

volatile keyword, 64
VPN (virtual page number), 434
VSID (virtual segment ID), 434

W

wait queues, 133-135, 559-564
 adding to, 136
 wait_event*() interfaces, 137-139
 waking up, 140-142
wait() function, 120-124
wait() system calls, 21
wait_event*() interfaces, 137-139
wait_table, wait_table_size field (memory zones), 189
wait_table_bits field (memory zones), 189
waking up (wait queues), 140-142
window manager distributions, 5
 Debian, 6
 Fedora, 6-7
 Gentoo, 7
 Mandriva, 7
 Red Hat, 6-7
 SUSE, 7
 Yellow Dog, 7

WLAN (wireless LAN), 259
work queues, 564-567
 lists, 34
 tasklets, 586-588
working directories, 299
writing source code, 575-590

X-Z

x86, 163-165
 assembly languages, 42-50
 PowerPC code convergence, 455
 real-time clocks, 417

Yaboot, 430-431
Yellow Dog, 7
yielding CPUs, 394-404

zombie processes, 23, 79, 115
zones
 memory management, 187-190
 padding, 190