Contents

Foreword **vii**

Preface **ix**

1 **LINUX – the operating system** **1**

1.1 Main characteristics 3
1.2 Linux distributions 5

2 **Compiling the kernel** **6**

2.1 Where is everything? 6
2.2 Compiling 9
2.3 Additional configuration facilities 11

3 **Introduction to the kernel** **15**

3.1 Important data structures 20
3.1.1 The task structure 20
3.1.2 The process table 29
3.1.3 Files and inodes 31
3.1.4 Dynamic memory management 33
3.1.5 Queues and semaphores 34
3.1.6 System time and timers 36
3.2 Main algorithms 37
3.2.1 Signals 37
3.2.2 Interrupts 39
3.2.3 Booting the system 41
3.2.4 Timer interrupt 44
3.2.5 The scheduler 47

3.3 Implementing system calls 51
3.3.1 How do system calls actually work? 51
3.3.2 Examples of simple system calls 53
3.3.3 Examples of more complex system calls 55
3.3.4 Implementing a new system call 65

4 Memory management 70

4.1 The architecture-independent memory model in LINUX 72
4.1.1 Pages of memory 72
4.1.2 Virtual address space 72
4.1.3 Converting the linear address 73
4.1.4 The page directory 74
4.1.5 The page middle directory 75
4.1.6 The page table 76
4.2 The virtual address space for a process 79
4.2.1 The user segment 79
4.2.2 Virtual memory areas 81
4.2.3 The system call *brk* 85
4.2.4 Mapping functions 86
4.2.5 The kernel segment 86
4.2.6 Static memory allocation in the kernel segment 88
4.2.7 Dynamic memory allocation in the kernel segment 88
4.3 Block device caching 92
4.3.1 Block buffering 92
4.3.2 The update and bdflush processes 94
4.3.3 List structures for the buffer cache 95
4.3.4 Using the buffer cache 96
4.4 Paging under LINUX 98
4.4.1 Page cache and management 100
4.4.2 Finding a free page 101
4.4.3 Page errors and reloading a page 106

5 Inter-process communication 108

5.1 Synchronization in the kernel 110
5.2 Communication via files 114
5.2.1 Locking entire files 115
5.2.2 Locking file areas 116
5.3 Pipes 119
5.4 Debugging using *ptrace* 121
5.5 System V IPC 125

5.5.1 Access permissions, numbers and keys 125
5.5.2 Semaphores 126
5.5.3 Message queues 130
5.5.4 Shared memory 133
5.5.5 The ipcs and ipcrm commands 137
5.6 IPC with sockets 138
5.6.1 A simple example 139
5.6.2 The implementation of UNIX domain sockets 143

6 The LINUX file system 148

6.1 Basic principles 149
6.2 The representation of file systems in the kernel 152
6.2.1 Mounting 153
6.2.2 Superblock operations 155
6.2.3 The inode 158
6.2.4 Inode operations 161
6.2.5 The file structure 165
6.2.6 File operations 165
6.2.7 Opening a file 169
6.2.8 The directory cache 172
6.3 The *Proc* file system 173
6.4 The *Ext2* file system 179
6.4.1 The structure of the *Ext2* file system 180
6.4.2 Directories in the *Ext2* file system 182
6.4.3 Block allocation in the *Ext2* file system 183
6.4.4 Extensions of the *Ext2* file system 184

7 Device drivers under LINUX 186

7.1 Character and block devices 188
7.2 Polling and interrupts 190
7.2.1 Polling mode 190
7.2.2 Interrupt mode 191
7.2.3 Interrupt sharing 193
7.2.4 Bottom halves 194
7.2.5 Task queues 196
7.2.6 DMA mode 198
7.3 The hardware 200
7.3.1 Hardware detection 203
7.3.2 Automatic interrupt detection 205
7.4 Implementing a driver 207
7.4.1 The setup function 209
7.4.2 init 210
7.4.3 open and release 212

7.4.4 `read` and `write` 214
7.4.5 IOCTL 217
7.4.6 `select` 219
7.4.7 `lseek` 220
7.4.8 `mmap` 220
7.4.9 `readdir`, `fsync` and `fasync` 221
7.4.10 `check_media_change` and `revalidate` 221
7.5 An example of DMA operation 221

8 Network implementation 227

8.1 Introductory summary 228
8.1.1 The layer model of the network implementation 229
8.1.2 Getting the data from A to B 230
8.2 Important structures 234
8.2.1 The `socket` structure 234
8.2.2 The `sk_buff structure` – buffer management in the network 235
8.2.3 The INET socket – a special part of a socket 239
8.2.4 Protocol operations in the `proto` structure 243
8.2.5 The general structure of a socket address 246
8.3 Network devices under LINUX 247
8.3.1 Ethernet 254
8.3.2 SLIP and PLIP 255
8.3.3 The loopback device 255
8.3.4 The dummy device 255
8.4 ARP – the Address Resolution Protocol 256
8.5 IP 259
8.5.1 IP – general introduction 259
8.5.2 IP functions 260
8.5.3 Routing 263
8.5.4 IP multicasting and IGPM 265
8.5.5 IP packet filters 267
8.5.6 IP accounting and IP firewalling 268
8.6 UDP 269
8.6.1 Functions of UDP 269
8.6.2 Other functions 271
8.7 TCP 271
8.7.1 General notes on TCP 271
8.7.2 The TCP communication end-point – a finite state machine 272

8.7.3 Functions of TCP 273
8.8 The packet interface – an alternative? 277

9 Modules and debugging 279

9.1 What are modules? 279
9.2 Implementation in the kernel 280
9.2.1 Signatures of symbols 283
9.3 What can be implemented as a module? 283
9.4 Parameter passing 285
9.5 The kernel daemon 285
9.6 An example module 287
9.7 Debugging 289
9.7.1 Changes are the beginning of the end 289
9.7.2 The best debugger – `printk()` 290
9.7.3 Debugging with `gdb` 291

10 Multi-processing 293

10.1 The Intel multi-processor specification 293
10.2 Problems with multi-processor systems 295
10.3 Changes to the kernel 296
10.3.1 Kernel initialization 296
10.3.2 Scheduling 297
10.3.3 Message exchange between processors 298
10.3.4 Entering kernel mode 298
10.3.5 Interrupt handling 299
10.4 Compiling LINUX SMP 300

APPENDIX A – System calls 301

A.1 Process management 302
A.2 The file system 339
A.3 Communication 370
A.4 Memory management 373
A.5 Initialization 378
A.6 All that remains 379

APPENDIX B – Kernel-related commands 380

B.1 `free` – synopsis of the system memory 380
B.2 `ps` – display of process statistics 381
B.3 Additional kernel configuration 386
B.4 `top` – the CPU charts 387
B.5 `init` – primus inter pares 389
B.6 `shutdown` – shutting down the system 395
B.7 `strace` – monitoring a process 396

B.8 Configuring the network interface 400
B.9 traceroute – Ariadne's thread in the Internet 400
B.10 Configuring a serial interface 403
B.11 Configuring a parallel interface 405
B.12 mount 407

APPENDIX C – The *Proc* file system 415

C.1 The /proc/ directory 415
C.2 The net/ directory 421
C.3 The self/ directory 422
C.4 The sys/ directory 426

APPENDIX D – The boot process 428

D.1 Carrying out the boot process 428
D.2 LILO – the Linux loader 431
D.2.1 LILO started by MS-DOS MBR 431
D.2.2 LILO started by a boot manager 431
D.2.3 LILO in the master boot record 432
D.2.4 LILO files 432
D.2.5 LILO boot parameters 437
D.2.6 LILO start-up messages 439
D.2.7 Error messages 439

APPENDIX E – Useful kernel functions 441

References 457

Index 461