Contents

Preface **xvii**

1 Preliminaries **1**
- 1.1 Procedures . . . 1
- 1.2 Linux Distributions . . . 2
- 1.3 Standards and the **LSB** . . . 3

2 Compilers **5**
- 2.1 GCC . . . 5
- 2.2 Major gcc Options . . . 6
- 2.3 Preprocessor . . . 8
- 2.4 Other Languages . . . 9
- 2.5 Integrated Development Environments (IDE) . . . 9
- 2.6 Labs . . . 12

3 Libraries **13**
- 3.1 Static Libraries . . . 13
- 3.2 Shared Libraries . . . 14
- 3.3 Linking To Libraries . . . 16
- 3.4 Dynamic Linking Loader . . . 17
- 3.5 Labs . . . 19

4 Make **21**
- 4.1 Using make and Makefiles . . . 21
- 4.2 Building large projects . . . 22
- 4.3 More complicated rules . . . 23
- 4.4 Built-in rules . . . 25
- 4.5 Labs . . . 26

5 Source Control **27**
- 5.1 Source Control . . . 27

5.2 RCS and CVS 28
5.3 Subversion 29
5.4 git 30
5.5 Labs 31

6 Debugging 35

6.1 gdb (ddd, kdbg, Insight) 35
6.2 Electric Fence 36
6.3 Getting the Time 37
6.4 Profiling and Performance 37
6.5 valgrind 39
6.6 Labs 42

7 System Calls 45

7.1 System Calls vs. Library Functions 45
7.2 How System Calls are Made 46
7.3 Return Values and Error Numbers 46
7.4 Labs 47

8 Memory Management and Allocation 49

8.1 Memory Management 49
8.2 Dynamical Allocation 50
8.3 Tuning **malloc()** 52
8.4 Labs 53

9 Files and Filesystems in Linux 55

9.1 Files, Directories and Devices 55
9.2 The Virtual File System 57
9.3 The ext2/ext3 Filesystem 58
9.4 Journalling Filesystems 59
9.5 The ext4/ Filesystem 60
9.6 Labs 61

10 File I/O 63

10.1 Unix File I/O 63
10.2 Opening and Closing 64
10.3 Reading, Writing and Seeking 66
10.4 Positional and Vector I/O 68

10.5 Standard I/O Library 69
10.6 Large File Support (LFS) 70
10.7 Labs 71

11 Advanced File Operations 73
11.1 Stat Functions 73
11.2 Directory Functions 76
11.3 Inotify 77
11.4 Memory Mapping 79
11.5 **flock()** and **fcntl()** 80
11.6 Making Temporary Files 83
11.7 Other System Calls 84
11.8 Labs 87

12 Processes - I 89
12.1 What is a Process? 89
12.2 Process Limits 90
12.3 Process Groups 94
12.4 The **proc** Filesystem 94
12.5 Inter-Process Communication Methods 95
12.6 Labs 97

13 Processes - II 99
13.1 Using **system()** to Create a Process 99
13.2 Using **fork()** to Create a Process 100
13.3 Using **exec()** to Create a Process 101
13.4 Using **clone()** 102
13.5 Exiting 104
13.6 Constructors and Destructors 105
13.7 Waiting 105
13.8 Daemon Processes 106
13.9 Labs 107

14 Pipes and Fifos 109
14.1 Pipes and Inter-Process Communication 109
14.2 **popen()** and **pclose()** 110
14.3 **pipe()** 111
14.4 Named Pipes (FIFOs) 113

14.5 **splice(), vmsplice()** and **tee()** . . . 114
14.6 Labs . . . 116

15 Asynchronous I/O **117**
15.1 What is Asynchronous I/O? . . . 117
15.2 The Posix Asynchronous I/O API . . . 118
15.3 Linux Implementation . . . 119
15.4 Labs . . . 122

16 Signals - I **123**
16.1 What are Signals? . . . 123
16.2 Signals Available . . . 124
16.3 Dispatching Signals . . . 125
16.4 Alarms, Pausing and Sleeping . . . 126
16.5 Setting up a Signal Handler . . . 127
16.6 Signal Sets . . . 129
16.7 **sigaction()** . . . 131
16.8 Labs . . . 133

17 Signals - II **135**
17.1 Reentrancy and Signal Handlers . . . 135
17.2 Jumping and Non-Local Returns . . . 136
17.3 **siginfo** and **sigqueue()** . . . 138
17.4 Real Time Signals . . . 139
17.5 Labs . . . 140

18 POSIX Threads - I **141**
18.1 Multi-threading under Linux . . . 141
18.2 Basic Program Structure . . . 142
18.3 Creating and Destroying Threads . . . 142
18.4 Signals and Threads . . . 145
18.5 Forking vs. Threading . . . 146
18.6 Labs . . . 147

19 POSIX Threads - II **149**
19.1 Deadlocks and Race Conditions . . . 149
19.2 Mutex Operations . . . 150
19.3 Semaphores . . . 151

19.4 Futexes . . . 152
19.5 Conditional Operations . . . 152
19.6 Labs . . . 154

20 Networking and Sockets 155
20.1 Networking Layers . . . 155
20.2 What are Sockets? . . . 158
20.3 Stream Sockets . . . 159
20.4 Datagram Sockets . . . 159
20.5 Raw Sockets . . . 160
20.6 Byte Ordering . . . 160
20.7 Labs . . . 161

21 Sockets - Addresses and Hosts 163
21.1 Socket Address Structures . . . 163
21.2 Converting IP Addresses . . . 165
21.3 Host Information . . . 166
21.4 Labs . . . 168

22 Sockets - Ports and Protocols 171
22.1 Service Port Information . . . 171
22.2 Protocol Information . . . 173
22.3 Labs . . . 174

23 Sockets - Clients 177
23.1 Basic Client Sequence . . . 177
23.2 **socket()** . . . 178
23.3 **connect()** . . . 181
23.4 **close()** and **shutdown()** . . . 182
23.5 Unix Client . . . 184
23.6 Internet Client . . . 184
23.7 Labs . . . 186

24 Sockets - Servers 187
24.1 Basic Server Sequence . . . 187
24.2 **bind()** . . . 189
24.3 **listen()** . . . 191
24.4 **accept()** . . . 192
24.5 Unix Server . . . 193
24.6 Internet Server . . . 194

24.7 Labs 195

25 Sockets - Input/Output Operations **197**

25.1 **write(), read()** 197

25.2 **send(), recv()** 199

25.3 **sendto(), recvfrom()** 200

25.4 **sendmsg(), recvmsg()** 201

25.5 **sendfile()** 203

25.6 **socketpair()** 204

25.7 Labs 204

26 Sockets - Options **207**

26.1 Getting and Setting Socket Options 207

26.2 **fcntl()** 208

26.3 **ioctl()** 209

26.4 **getsockopt()** and **setsockopt()** 209

26.5 Labs 211

27 Netlink Sockets **213**

27.1 What are netlink Sockets? 213

27.2 Opening a netlink Socket 214

27.3 netlink Messages 215

27.4 Labs 217

28 Sockets - Multiplexing and Concurrent Servers **219**

28.1 Multiplexed and Asynchronous Socket I/O 220

28.2 **select()** 220

28.3 **poll()** 223

28.4 **pselect()** and **ppoll()** 225

28.5 **epoll** 225

28.6 Signal Driven and Asynchronous I/O 227

28.7 Concurrent Servers 228

28.8 Labs 229

29 Inter Process Communication **233**

29.1 Methods of IPC 233

29.2 System V IPC 234

29.3 Posix IPC 237

29.4 Labs 238

30 Shared Memory 241

30.1 What is Shared Memory? 241

30.2 System V Shared Memory 242

30.3 POSIX Shared Memory 244

30.4 Labs 246

31 Semaphores 247

31.1 What is a Semaphore? 247

31.2 System V Semaphores 248

31.3 POSIX Semaphores 251

31.4 Labs 253

32 Message Queues 255

32.1 What are Message Queues? 255

32.2 System V Message Queues 256

32.3 POSIX Message Queues 259

32.4 Labs 262