

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>11</b>
<b>1 Einführung</b>	<b>13</b>
1.1 Warum gerade „C“? . . . . .	13
1.2 Die Werkzeuge . . . . .	14
1.2.1 Der Editor – die Qual der Wahl . . . . .	14
1.2.2 Der C-Compiler: <code>gcc</code> . . . . .	15
1.2.3 Ablaufsteuerung mit <code>make</code> . . . . .	16
1.2.4 Die Debugger: <code>gdb</code> , <code>xxgdb</code> und <code>ddd</code> . . . . .	17
1.2.5 Integrierte Entwicklungsumgebungen . . . . .	19
1.3 Der Umgang mit <code>gcc</code> , <code>gdb</code> und <code>make</code> an Beispielen . . . . .	22
1.3.1 Primzahlen berechnen . . . . .	22
1.3.2 Fehlersuche mit dem <code>gcc</code> . . . . .	23
1.3.3 Fehlersuche mit <code>gdb</code> , <code>xxgdb</code> und <code>ddd</code> . . . . .	24
1.3.4 Funktionsbibliotheken verwenden . . . . .	27
1.3.5 Quelltexte aufteilen . . . . .	28
1.4 Weiterführende Informationen . . . . .	31
1.4.1 Die Unix-Online-Hilfen „ <code>man</code> “ und „ <code>xman</code> “ . . . . .	32
1.4.2 Ein Blick hinter die Kulissen: die Include-Dateien .	33
<b>2 Arbeiten mit einer Entwicklungsumgebung</b>	<b>37</b>
2.1 Die „X-Window Workbench“ . . . . .	37
2.1.1 Ein kleines Beispielprogramm . . . . .	38
2.1.2 Projektdateien . . . . .	40
2.1.3 Auswählen und Verändern der Konfiguration . . . . .	40
2.1.4 Anpassen der Konfigurationsdatei . . . . .	41
2.2 „Anjuta“ . . . . .	42
2.2.1 Ein neues Projekt anlegen . . . . .	42
2.2.2 Eingabe der Quelltexte . . . . .	44

2.2.3	Kompilieren und Starten des Beispiels . . . . .	46
<b>3</b>	<b>Shell-Programme</b>	<b>47</b>
3.1	Parameter und Rückgabewert der Funktion <code>main()</code> . . . . .	47
3.1.1	Die Bedeutung des Rückgabewertes von <code>main()</code> . . . . .	48
3.1.2	Die Variablen <code>argc</code> und <code>argv</code> . . . . .	48
3.1.3	Auswerten der Kommandozeilenparameter . . . . .	49
3.1.4	Achtung: Platzhalter! . . . . .	51
3.2	Konventionen für Shell-Programme . . . . .	53
3.2.1	Ein Muss: Die Hilfe-Option . . . . .	53
3.2.2	Fehlermeldungen . . . . .	54
3.2.3	Eigene <code>man</code> -Seiten erstellen . . . . .	56
3.3	Programme mehrsprachig auslegen . . . . .	57
3.4	Ausgabesteuerung im Terminal-Fenster . . . . .	64
3.4.1	ANSI-Steuersequenzen . . . . .	64
3.4.2	Die „ <code>curses</code> “-Bibliothek . . . . .	66
<b>4</b>	<b>Dateien und Verzeichnisse</b>	<b>69</b>
4.1	Das Arbeiten mit Dateien . . . . .	69
4.1.1	Gepufferte Ein-/Ausgabe . . . . .	69
4.1.2	<code>stdin</code> , <code>stdout</code> und <code>stderr</code> . . . . .	70
4.1.3	Dateien öffnen und schließen . . . . .	71
4.1.4	Lesen aus und Schreiben in Dateien . . . . .	72
4.1.5	Ein Beispiel: Zeilen nummerieren . . . . .	76
4.2	Eigenschaften von Dateien oder Verzeichnissen auswerten	78
4.3	Verzeichnisse einlesen . . . . .	80
<b>5</b>	<b>Interprozesskommunikation</b>	<b>83</b>
5.1	Prozessverwaltung unter Linux . . . . .	83
5.2	Neue Prozesse starten . . . . .	84
5.2.1	Shell-Programme aufrufen mit <code>system()</code> . . . . .	84
5.2.2	Die Funktionen der <code>exec</code> -Familie . . . . .	85
5.2.3	Einen Kind-Prozess erzeugen mit <code>fork()</code> . . . . .	87
5.2.4	Warteschleifen . . . . .	90
5.3	Signale . . . . .	91
5.3.1	Die Weckfunktion <code>alarm()</code> . . . . .	92
5.3.2	Einen Signal-Handler einrichten . . . . .	93
5.3.3	Auf die Beendigung eines Kind-Prozesses warten .	94
5.3.4	Signale setzen mit <code>kill()</code> . . . . .	95
5.4	Datenaustausch zwischen Prozessen . . . . .	96
5.4.1	Pipes . . . . .	96

5.4.2	FIFOs . . . . .	101
5.4.3	Shared Memory . . . . .	103
5.5	Alternativen zur Erzeugung von Prozessen . . . . .	106
5.5.1	popen() und pclose() . . . . .	106
5.5.2	Die fork()-Alternative clone() . . . . .	108
5.5.3	POSIX-Threads . . . . .	110
<b>6</b>	<b>Devices – das Tor zur Hardware</b>	<b>115</b>
6.1	Das Device-Konzept unter Linux . . . . .	115
6.1.1	Devices öffnen und schließen . . . . .	116
6.1.2	Ungepuffertes Lesen und Schreiben . . . . .	117
6.1.3	Devices steuern mit ioctl() . . . . .	118
6.2	Das CD-ROM-Laufwerk . . . . .	119
6.2.1	Die CD „auswerfen“ . . . . .	119
6.2.2	Fähigkeiten des Laufwerks auslesen . . . . .	120
6.2.3	Audio-CDs abspielen . . . . .	122
6.3	Ansteuerung einer Sound-Karte . . . . .	130
6.3.1	Der Mixer . . . . .	131
6.3.2	Audio-Daten aufnehmen und wiedergeben . . . . .	135
6.4	„Video for Linux“ . . . . .	140
6.4.1	Eigenschaften des Devices . . . . .	140
6.4.2	Bilder aufzeichnen . . . . .	143
6.5	Die serielle Schnittstelle . . . . .	148
6.5.1	Terminal-Parameter einstellen . . . . .	149
6.5.2	Ein kleines Terminalprogramm . . . . .	152
<b>7</b>	<b>Netzwerkprogrammierung</b>	<b>157</b>
7.1	Einführung . . . . .	158
7.1.1	Begriffe . . . . .	158
7.1.2	Das Client-Server-Prinzip . . . . .	161
7.1.3	Sockets . . . . .	161
7.2	Der TCP/IP-Client . . . . .	163
7.2.1	Aufbau einer Verbindung . . . . .	163
7.2.2	Ein „Universal“-Client . . . . .	165
7.2.3	Rechnernamen in IP-Adressen umwandeln . . . . .	168
7.3	Server-Programme . . . . .	171
7.3.1	Die Funktionsweise eines Servers . . . . .	171
7.3.2	Ein interaktiver TCP/IP-Server . . . . .	173
7.3.3	Ein minimaler HTTP-Server . . . . .	177
7.4	Noch ein Wort zur Sicherheit . . . . .	184

<b>8 Grafikprogrammierung</b>	<b>187</b>
8.1 Die grafische Oberfläche X11 . . . . .	187
8.2 XView und OpenLook . . . . .	188
8.2.1 Ein Fenster öffnen . . . . .	189
8.2.2 Attribute abfragen und ändern . . . . .	194
8.2.3 Bedienfelder einrichten . . . . .	194
8.2.4 Schaltflächen (Buttons) einrichten . . . . .	196
8.2.5 Daten an eine Callback-Funktion weiterleiten . . . . .	201
8.2.6 Anpassen der Fenster- und Bedienfeldgröße . . . . .	204
8.2.7 Das Message-Objekt . . . . .	205
8.2.8 Auswahlfelder . . . . .	206
8.2.9 Icons (Piktogramme) . . . . .	209
8.2.10 Eingabefelder für Text und Zahlen . . . . .	214
8.2.11 Pegelanzeige und Schieberegler . . . . .	217
8.2.12 Menüs . . . . .	219
8.2.13 Command Frames . . . . .	224
8.2.14 Hinweis- und Abfragefenster . . . . .	226
8.2.15 Farben einrichten . . . . .	231
8.2.16 Zeichenflächen . . . . .	234
8.2.17 Ein kleines Zeichenprogramm . . . . .	250
8.2.18 Dateiauswahlfenster . . . . .	256
8.3 Grafik ohne X11 mit der libvga . . . . .	261
8.3.1 Besonderheiten beim Arbeiten mit der libvga . . . . .	261
8.3.2 Ein erstes Beispiel . . . . .	262
8.3.3 Mit Perspektive: 3D-Funktionen zeichnen . . . . .	265
8.3.4 Ein kleines Malprogramm . . . . .	268
8.3.5 Erweiterte Funktionen mit der libvgagl . . . . .	273
8.3.6 Weitere Informationsquellen . . . . .	275
<b>9 Hardware-Programmierung</b>	<b>277</b>
9.1 Hardware-nahe Programme schreiben . . . . .	277
9.1.1 Eigene Programme mit root-Rechten ausstatten . . . . .	278
9.1.2 Zugriff auf I/O-Ports freischalten . . . . .	278
9.1.3 Zugriff auf die I/O-Ports . . . . .	279
9.2 Ansteuerung des Parallelports . . . . .	280
9.2.1 Beschreibung des Parallelports . . . . .	280
9.2.2 Die Adresse des Parallelports suchen . . . . .	280
9.2.3 Ein Beispiel: „LED-Lauflicht“ . . . . .	283
9.3 Modem-Steuerleitungen abfragen . . . . .	285

<b>10 Beispielprojekte</b>	<b>289</b>
10.1 Ein Mischpult für X11: <code>xvmixer</code>	289
10.1.1 Ziel des Projektes	289
10.1.2 Aufteilung des Quelltextes	290
10.1.3 Das Hauptprogramm	291
10.1.4 Die Ansteuerung des Audio-Mixers	293
10.1.5 Die Funktionen der grafischen Benutzeroberfläche	297
10.1.6 Das Ergebnis	305
10.1.7 To Do	306
10.2 Telefonbuch mit automatischer Anwahl	307
10.2.1 Ziel des Projektes	307
10.2.2 Strukturierung des Projektes	308
10.2.3 Das Hauptprogramm	308
10.2.4 Funktionen zur Ansteuerung des Modems	312
10.2.5 Die Benutzerschnittstelle	315
10.2.6 To Do	321
<b>Anhang</b>	<b>323</b>
A1 – Daten zum Buch im Internet	323
A2 – Das Paket <code>gettext</code>	323
A3 – Deutsche Sprachunterstützung für XView	324
A4 – Aufbau einer WAV-Audiodatei	324
A5 – Aufbau einer AU-Audiodatei	326
A6 – Linux-Programmierung unter Windows: Cygwin	326
A7 – Informationen aus dem Internet	328
<b>Index</b>	<b>329</b>