

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Warum gerade „C“?	1
1.2	Bevor es losgeht ...	2
1.2.1	Paketverwaltung unter SuSE-Linux	2
1.2.2	Paketinstallation bei Ubuntu	4
1.3	Die Werkzeuge	6
1.3.1	Der Editor – die Qual der Wahl	6
1.3.2	Der GNU C-Compiler gcc	8
1.3.3	Ablaufsteuerung mit GNU make	8
1.3.4	Für die Fehlersuche: Die Debugger	10
1.3.5	Integrierte Entwicklungsumgebungen	11
1.4	Der Umgang mit Compiler, Debugger und „make“ anhand von Beispielen	14
1.4.1	Primzahlen berechnen	14
1.4.2	Fehlersuche mit dem gcc	16
1.4.3	Fehlersuche mit dem GNU Debugger	17
1.4.4	Funktionsbibliotheken verwenden	19
1.4.5	Quelltexte aufteilen	21
1.5	Weiterführende Informationen	25
1.5.1	Die Unix-Online-Hilfen „man“, „xman“ und „tkman“	26
1.5.2	Ein Blick hinter die Kulissen: Die Include-Dateien	28
•		
<b>2</b>	<b>Arbeiten mit einer Entwicklungsumgebung</b>	<b>31</b>
2.1	Anjuta	31
2.1.1	Ein neues Projekt anlegen	31
2.1.2	Eingabe der Quelltexte	33

2.1.3	Kompilieren und Starten des Beispiels . . . . .	35
2.2	KDevelop . . . . .	36
2.3	Eclipse + C Development Tooling (CDT) . . . . .	39
2.3.1	Plug-ins einbinden . . . . .	40
2.3.2	Ein neues Projekt anlegen . . . . .	40
<b>3</b>	<b>Kommandozeilenprogramme . . . . .</b>	<b>43</b>
3.1	Parameter und Rückgabewert der Funktion <code>main()</code> . . . . .	43
3.1.1	Die Bedeutung des Rückgabewertes von <code>main()</code> . . . . .	44
3.1.2	Die Variablen <code>argc</code> und <code>argv</code> . . . . .	44
3.1.3	Auswerten der Kommandozeilenparameter . . . . .	45
3.1.4	Achtung: Platzhalter! . . . . .	47
3.2	Konventionen für Kommandozeilenprogramme . . . . .	48
3.2.1	Ein Muss: Die Hilfe-Option . . . . .	48
3.2.2	Fehlermeldungen . . . . .	50
3.2.3	Eigene <i>Manpages</i> erstellen . . . . .	51
3.3	Programme mehrsprachig auslegen . . . . .	53
3.4	Ausgabesteuerung im Terminal-Fenster . . . . .	60
3.4.1	ANSI-Steuersequenzen . . . . .	60
3.4.2	Die „ncurses“-Bibliothek . . . . .	61
<b>4</b>	<b>Dateien und Verzeichnisse . . . . .</b>	<b>67</b>
4.1	Die Arbeit mit Dateien . . . . .	67
4.1.1	Gepufferte Ein-/Ausgabe . . . . .	67
4.1.2	<code>stdin</code> , <code>stdout</code> und <code>stderr</code> . . . . .	68
4.1.3	Dateien öffnen und schließen . . . . .	69
4.1.4	Lesen aus und Schreiben in Dateien . . . . .	70
4.1.5	Ein Beispiel: Zeilen nummerieren . . . . .	74
4.2	Eigenschaften von Dateien oder Verzeichnissen auswerten . . . . .	75
4.3	Verzeichnisse einlesen . . . . .	77
<b>5</b>	<b>Interprozesskommunikation . . . . .</b>	<b>79</b>
5.1	Prozessverwaltung unter Linux . . . . .	79
5.2	Neue Prozesse starten . . . . .	80
5.2.1	Shell-Programme aufrufen mit <code>system()</code> . . . . .	80
5.2.2	Die Funktionen der <code>exec</code> -Familie . . . . .	81

5.2.3	Einen Kind-Prozess erzeugen mit <code>fork()</code> . . . . .	82
5.2.4	Warteschleifen . . . . .	85
5.3	Signale . . . . .	86
5.3.1	Die Weckfunktion <code>alarm()</code> . . . . .	87
5.3.2	Einen Signal-Handler einrichten . . . . .	88
5.3.3	Auf die Beendigung eines Kind-Prozesses warten . . . . .	89
5.3.4	Signale setzen mit <code>kill()</code> . . . . .	90
5.4	Datenaustausch zwischen Prozessen . . . . .	91
5.4.1	Pipes . . . . .	91
5.4.2	FIFOs . . . . .	95
5.4.3	Shared Memory . . . . .	97
5.5	Alternative Verfahren zur Erzeugung von Prozessen . . . . .	100
5.5.1	<code>popen()</code> und <code>pclose()</code> . . . . .	100
5.5.2	Die <code>fork()</code> -Alternative <code>clone()</code> . . . . .	101
5.5.3	POSIX-Threads . . . . .	103
<b>6</b>	<b>Devices – das Tor zur Hardware . . . . .</b>	<b>107</b>
6.1	Das Device-Konzept von Linux . . . . .	107
6.1.1	Devices öffnen und schließen . . . . .	108
6.1.2	Ungepuffertes Lesen und Schreiben . . . . .	109
6.1.3	Devices steuern mit <code>ioctl()</code> . . . . .	110
6.2	Das CD-ROM-Laufwerk . . . . .	111
6.2.1	Die CD „auswerfen“ . . . . .	111
6.2.2	Fähigkeiten des Laufwerks auslesen . . . . .	112
6.2.3	Audio-CDs abspielen . . . . .	114
6.3	Ansteuerung einer Soundkarte . . . . .	121
6.3.1	OSS, ALSA und ESOUND . . . . .	122
6.3.2	Der Mixer . . . . .	122
6.3.3	Audiodaten aufnehmen und wiedergeben . . . . .	126
6.4	„Video for Linux“ . . . . .	130
6.4.1	Eigenschaften des Devices . . . . .	130
6.4.2	Bilder aufzeichnen . . . . .	133
6.5	Die serielle Schnittstelle . . . . .	142
6.5.1	Terminal-Parameter einstellen . . . . .	143
6.5.2	Ein kleines Terminalprogramm . . . . .	145
6.6	Druckerausgaben . . . . .	149

6.7	Der <i>Universal Serial Bus</i> (USB) . . . . .	154
6.7.1	Ansteuerung von USB-Geräten anhand eines Beispiels . . . .	156
<b>7</b>	<b>Netzwerkprogrammierung . . . . .</b>	<b>163</b>
7.1	Einführung . . . . .	164
7.1.1	Begriffe . . . . .	164
7.1.2	Vorbereitung . . . . .	166
7.1.3	Das Client-Server-Prinzip . . . . .	169
7.1.4	Sockets . . . . .	170
7.2	Der TCP/IP-Client . . . . .	171
7.2.1	Aufbau einer Verbindung . . . . .	171
7.2.2	Ein „Universal“-Client . . . . .	173
7.2.3	Rechnernamen in IP-Adressen umwandeln . . . . .	176
7.3	Server-Programme . . . . .	178
7.3.1	Die Funktionsweise eines Servers . . . . .	178
7.3.2	Ein interaktiver TCP/IP-Server . . . . .	180
7.3.3	Ein kleiner Webserver . . . . .	184
7.4	Das <i>User Datagram Protocol</i> (UDP) . . . . .	191
7.4.1	UDP-Nachrichten senden . . . . .	191
7.4.2	Der UDP-Server . . . . .	194
7.4.3	Pakete an alle Teilnehmer senden: Broadcast . . . . .	197
7.4.4	Multicast-Sockets . . . . .	199
7.4.5	UPnP – <i>Universal Plug And Play</i> . . . . .	200
7.5	Noch ein Wort zur Sicherheit . . . . .	204
<b>8</b>	<b>Grafische Benutzeroberflächen . . . . .</b>	<b>205</b>
8.1	Die grafische Oberfläche X11 . . . . .	205
8.2	Das Toolkit GTK+ . . . . .	206
8.2.1	GTK 1.2 versus GTK 2.0 . . . . .	206
8.2.2	GTK-Programme übersetzen . . . . .	207
8.2.3	Ein erstes Beispiel . . . . .	208
8.2.4	Das Callback-Prinzip . . . . .	210
8.2.5	Schaltflächen (Buttons) . . . . .	213
8.2.6	Hinweistexte (Tipps) . . . . .	216
8.2.7	Widgets anordnen . . . . .	216
8.2.8	Text-Labels . . . . .	220

8.2.9	Dialogfenster . . . . .	221
8.2.10	Auswahlfelder . . . . .	224
8.2.11	Eingabefelder für Text und Zahlen . . . . .	228
8.2.12	Menüs . . . . .	233
8.2.13	Pixmap-Grafiken darstellen . . . . .	238
8.2.14	Zeichenflächen . . . . .	244
8.2.15	Zeichenfläche mit Rollbalken . . . . .	250
8.2.16	Dateiauswahlfenster . . . . .	252
8.2.17	Umlaute und Sonderzeichen . . . . .	255
8.2.18	Wie geht es weiter? . . . . .	255
8.3	Grafik ohne X11 mit der SVGALIB . . . . .	256
8.3.1	Besonderheiten beim Arbeiten mit der libvga . . . . .	256
8.3.2	Ein erstes Beispiel . . . . .	257
8.3.3	Mit Perspektive: 3D-Funktionen zeichnen . . . . .	260
8.3.4	Ein kleines Malprogramm . . . . .	262
8.3.5	Erweiterte Funktionen mit der libvgagl . . . . .	266
8.3.6	Weitere Informationsquellen . . . . .	268
<b>9</b>	<b>Hardware-Programmierung . . . . .</b>	<b>271</b>
9.1	Hardware-nahe Programme schreiben . . . . .	271
9.1.1	Eigene Programme mit <i>root</i> -Rechten ausstatten . . . . .	272
9.1.2	Zugriff auf I/O-Ports freischalten . . . . .	272
9.1.3	Zugriff auf die I/O-Ports . . . . .	273
9.2	Ansteuerung des Parallelports . . . . .	274
9.2.1	Beschreibung des Parallelports . . . . .	274
9.2.2	Die Adresse des Parallelports suchen . . . . .	275
9.2.3	Ein Beispiel: „LED-Lauflicht“ . . . . .	276
9.3	Modem-Steuerleitungen abfragen . . . . .	279
<b>10</b>	<b>Beispielprojekte . . . . .</b>	<b>283</b>
10.1	WebCam: Video-Übertragung per HTTP . . . . .	283
10.1.1	Wie die Bilder laufen lernen . . . . .	284
10.1.2	Strukturierung der Quelltexte . . . . .	284
10.1.3	Die HTTP-Authentifizierung . . . . .	298
10.2	Telefonbuch mit automatischer Anwahl . . . . .	300
10.2.1	Ziel des Projektes . . . . .	300

10.2.2	Strukturierung des Projektes . . . . .	301
10.2.3	Das Hauptprogramm . . . . .	301
10.2.4	Funktionen zur Ansteuerung des Modems . . . . .	304
10.2.5	Die Benutzerschnittstelle . . . . .	307
10.2.6	To Do . . . . .	312
<b>Anhang</b>	<b>. . . . .</b>	<b>315</b>
A1	– Daten zum Buch im Internet . . . . .	315
A2	– Das X11-Toolkit XView . . . . .	315
A3	– Aufbau einer WAV-Audiodatei . . . . .	316
A4	– Aufbau einer AU-Audiodatei . . . . .	317
A5	– Linux-Programmierung unter Windows: Cygwin . . . . .	317
<b>Literaturverzeichnis</b>	<b>. . . . .</b>	<b>319</b>
<b>Stichwortverzeichnis</b>	<b>. . . . .</b>	<b>321</b>