Vorwort XI

Teil 1 Die Umgebung 1

1 Richten Sie sich Ihre Umgebung ein 3

Einen Paketmanager einsetzen 4
C unter Windows kompilieren 6
POSIX für Windows 6
C mit POSIX kompilieren 8
C ohne POSIX kompilieren 8
Wo bitte geht es zur Bibliothek? 9
Ein paar meiner Lieblings-Flags 11
Pfade 12
Runtime-Linking 15
Makefiles verwenden 16
Variablen setzen 17
Die Regeln 19
Bibliotheken über ihren Quellcode nutzen 23
Bibliothek über ihren Quellcode nutzen – auch wenn Ihr Sysadmin das nicht will 24
C-Programme über Here-Dokumente kompilieren 26
Header-Dateien an der Befehlszeile einbinden 26
Der vereinheitlichte Header 27
Here-Dokumente 28
Von Stdin kompilieren 29

Bibliografische Informationen
http://d-nb.info/1046513710

digitalisiert durch DEUTSCHE NATIONAL BIBLIOTHEK

2 Debuggen, Testen, Dokumentieren ... 31
Einen Debugger verwenden ... 31
GDB-Variablen ... 35
Geben Sie Ihre Strukturen aus ... 37
Mit Valgrind auf Fehler prüfen ... 40
Unit-Tests 41
Ein Programm als Bibliothek verwenden ... 44
Abdeckung ... 46
Dokumentation einweben ... 47
Doxygen ... 47
Literaler Code mit CWEB ... 49
Fehlerprüfung ... 51
Wie ist der Anwender in den Fehler involviert? ... 51
Der Kontext, in dem der Anwender arbeitet ... 53
Wie sollte ein Hinweis auf einen Fehler zurückgegeben werden? ... 54

3 Verpacken Sie Ihr Projekt ... 55
Die Shell 56
Shell-Befehle durch ihre Ausgabe ersetzen ... 57
Die Shell für Schleifen nutzen, um auf einem Satz Dateien zu arbeiten ... 58
Dateien prüfen ... 60
fc 63
Makefiles vs. Shell-Skripten ... 65
Packen Sie Ihren Code mit den Autotools ... 67
Ein Autotools-Beispiel ... 69
Das Makefile durch Makefile.am beschreiben ... 72
Das configure-Skript ... 76

4 Versionsverwaltung ... 81
Änderungen per diff ... 82
Git-Objekte ... 83
Der Stash ... 87
Bäume und ihre Zweige ... 88
Merging ... 89
Der Rebase ... 91
Remote-Repositories ... 92

5 Mit anderen zusammenspielen ... 95
Das Vorgehen ... 95

Schreiben, damit es von anderen Sprachen gelesen werden kann 95
Die Wrapper-Funktion 96
Datenstrukturen über die Grenze schmuggeln 97
Linken 98
Python als Host 99
Kompilieren und linken 100
Das bedingte Unterverzeichnis für Automake 101
Distutils mit Unterstützung durch die Autotools 102

Teil II: Die Sprache 105

6 Ihr Weg zum Zeiger 107
Automatischer, statischer und manueller Speicher 107
Persistente Statusvariablen 110
Zeiger ohne malloc 111
Strukturen werden kopiert, Arrays werden als Alias weitergegeben 113
malloc und Speichertricks 115
Das Schicksal liegt in den Sternen 116
All die Zeigerarithmetik, die Sie kennen müssen 117

7 C-Syntax, die Sie ignorieren können 123
Kümmern Sie sich nicht darum, explizit aus main zurückzukehren 124
Lassen Sie Deklarationen fließen 124
Die Array-Größe zur Laufzeit setzen 126
Weniger Casting 127
Enums und Strings 128
Labels, goto, switch und break 130
Durchdachtes goto 131
switch 132
Veraltetes Float 135
Vorzeichenlose Integerwerte vergleichen 138

8 Hindernisse und Gelegenheiten 139
Robuste und ansprechende Makros schreiben 139
Präprozessortricks 143
Mit static und extern verlinken 146
Extern verlinkte Variablen in Header-Dateien 148

Das Schlüsselwort const 149
Nomen-Adjektiv-Form 150
Spannungen 151
Tiefe 152
Das Problem mit char const ** 153

9 Text **157**

Den Umgang mit Strings mithilfe von asprintf einfacher gestalten 157
Sicherheit 159
Konstante Strings 159
Strings mit asprintf erweitern 161
Ein Loblied auf strtok 162
Unicode 167
Das Kodieren für C-Code 169
Unicode-Bibliotheken 170
Der Beispielcode 171

10 Bessere Strukturen 175

Compound-Literale 176
Initialisierung per Compound-Literal 177
Variadische Makros 177
Listen sicher abschließen 179
Mehrfache Listen 180
foreach 181
Eine Funktion vektorisieren 182
Designated Initializers 183
Arrays und Structs mit Nullen initialisieren 185
Typedefs retten Ihnen den Tag 187
Über Stil 188
Mehrere Elemente aus einer Funktion zurückgeben 189
Fehler melden 190
Flexible Eingabewerte für Funktionen 192
Deklarieren Sie Ihre Funktion im printf-Stil 193
Optionale und benannte Argumente 195
Eine alte Funktion aufpolieren 197
Der void-Zeiger und die Strukturen, auf die er zeigt 203
Funktionen mit generischen Eingabewerten 203
Generische Strukturen 207

11 Objektorientierte Programmierung in C ... **213**
Was Sie nicht bekommen (und warum Sie es nicht vermissen werden) ... 214
Gültigkeitsbereich ... 214
Überladen mit Operator-Überladung ... 217
Strukturen und Dictionaries erweitern ... 222
Eine Struktur erweitern ... 223
Ein Dictionary implementieren ... 227
Lassen Sie Ihren Code auf Zeigern auf Objekte basieren ... 231
Funktionen in Ihren Structs ... 232
Referenzen zählen ... 237
Beispiel: Ein Substring-Objekt ... 237
Ein agentenbasiertes Modell der Gruppenbildung ... 241

12 Bibliotheken ... **249**
GLib 249
POSIX 250
Mit mmap riesige Datensätze verarbeiten ... 250
Einfaches Threading mit Pthreads ... 253
Die GNU Scientific Library ... 261
SQLite 263
Die Abfragen ... 264
libxml und cURL ... 266

Epilog ... **271**

Glossar ... **273**

Bibliografie ... **277**

Index ... **279**