

Auf einen Blick

1	Einstieg in die Welt von C	21
2	Erste Schritte in C	37
3	Basisdatentypen in C	55
4	Rechnen mit C und Operatoren	85
5	Bedingte Anweisung und Verzweigung	115
6	Schleifen – Programmteile wiederholen	147
7	Funktionen erstellen	163
8	Präprozessor-Direktiven	193
9	Arrays und Zeichenketten (Strings)	219
10	Zeiger (Pointer)	251
11	Dynamische Speicherverwaltung	289
12	Komplexe Datentypen	305
13	Dynamische Datenstrukturen	343
14	Eingabe- und Ausgabefunktionen	361
15	Zeitroutinen (time.h)	411

Inhalt

Vorwort	17
1 Einstieg in die Welt von C	21
1.1 Die Sprache C	21
1.2 Die C-Standardbibliothek	22
1.2.1 Header-Dateien und Programmbibliothek	23
1.3 Die nötigen Werkzeuge für C	26
1.4 Übersetzen mit der Entwicklungsumgebung	29
1.5 Übersetzen mit GCC und Clang	33
1.6 Listings zum Buch	35
1.7 Kontrollfragen und Aufgaben im Buch	36
2 Erste Schritte in C	37
2.1 Das erste Programm in C	37
2.2 Die Funktion printf()	40
2.3 Zeichensätze in C	42
2.3.1 Basis-Ausführungszeichensatz	42
2.3.2 Einige wichtige Escape-Sequenzen	42
2.4 Symbole in C	45
2.4.1 Bezeichner	45
2.4.2 Reservierte Schlüsselwörter	46
2.4.3 Literale	48
2.4.4 Einfache Begrenzer	49

2.5	Kommentare	50
2.6	Kontrollfragen und Aufgaben	52
3	Basisdatentypen in C	55
3.1	Variablen	55
3.2	Deklaration und Definition	56
3.3	Initialisierung und Zuweisung von Werten	58
3.4	Datentypen für Ganzzahlen	59
3.4.1	Vorzeichenlos und vorzeichenbehaftet	61
3.4.2	Suffixe für Ganzzahlen	64
3.5	Datentypen für Zeichen	65
3.5.1	Der Datentyp char	65
3.5.2	Der Datentyp wchar_t	67
3.5.3	Unicode-Unterstützung	68
3.6	Datentypen für Fließkommazahlen	68
3.6.1	Suffixe für Fließkommazahlen	70
3.6.2	Komplexe Gleitkommatypen	70
3.7	Boolescher Datentyp	72
3.8	Speicherbedarf mit sizeof ermitteln	73
3.9	Wertebereiche der Datentypen ermitteln	75
3.9.1	Limits von Integer-Typen	76
3.9.2	Limits von Fließkommazahlen	78
3.9.3	Integer-Typen mit fester Größe verwenden	79
3.9.4	Sicherheit beim Kompilieren mit <code>_static_assert()</code> ...	80
3.10	Konstanten erstellen	81
3.11	Lebensdauer und Sichtbarkeit von Variablen	82

3.12	void – ein unvollständiger Typ	84
3.13	Kontrollfragen und Aufgaben	84
4	Rechnen mit C und Operatoren	85
4.1	Werte formatiert einlesen mit scanf()	85
4.2	Operatoren	89
4.3	Arithmetische Operatoren	93
4.4	Inkrement- und Dekrementoperator	95
4.5	Bitoperatoren	98
4.6	Implizite Typumwandlung	102
4.6.1	Arithmetische Umwandlung	103
4.6.2	Typpromotionen	105
4.7	Explizite Typumwandlung	106
4.8	Mathematische Funktionen in C	107
4.9	Kontrollfragen und Aufgaben	112
5	Bedingte Anweisung und Verzweigung	115
5.1	Bedingte Anweisungen	115
5.1.1	Bedingte Anweisungen mit if	116
5.1.2	Vergleichsoperatoren	118
5.1.3	Mini-Exkurs: Anweisungsblock	121
5.2	Die alternative Verzweigung	121
5.3	Der Bedingungsoperator ?:	124
5.4	Die mehrfache Verzweigung mit if und else if	125
5.4.1	Verschachteln von Verzweigungen	128

5.5 Mehrfache Verzweigung mit switch	130
5.5.1 Austritt aus der Fallunterscheidung mit break	133
5.6 Logische Verknüpfungen	136
5.6.1 Der !-Operator	137
5.6.2 Der &&-Operator – logisches UND	139
5.6.3 Der -Operator – logisches ODER	141
5.7 Kontrollfragen und Aufgaben	143
6 Schleifen – Programmteile wiederholen	147
<hr/>	
6.1 Die Zählschleife – for	147
6.2 Die kopfgesteuerte while-Schleife	152
6.3 Die fußgesteuerte do-while-Schleife	154
6.4 Kontrolliertes Verlassen von Schleifen	157
6.4.1 break	157
6.4.2 continue	159
6.5 Kontrollfragen und Aufgaben	161
7 Funktionen erstellen	163
<hr/>	
7.1 Funktionen definieren	164
7.2 Funktionen aufrufen	165
7.3 Funktionsdeklaration (Vorausdeklaration)	166
7.4 Funktionsparameter	169
7.5 Rückgabewert von Funktionen	171
7.5.1 Die return-Anweisung	173
7.6 Exkurs: Funktionen bei der Ausführung	175

7.7 Inline-Funktionen	176
7.8 Rekursionen	177
7.9 Die main()-Funktion	179
7.9.1 Der Rückgabewert von main() an das Betriebssystem	180
7.10 Programme mit exit() beenden	182
7.11 Globale, lokale und statische Variablen	183
7.11.1 Lokale Variablen	184
7.11.2 Globale Variablen	185
7.11.3 Speicherklasse static	188
7.11.4 Die Speicherklasse extern	190
7.12 Kontrollfragen und Aufgaben	191
8 Präprozessor-Direktiven	193
<hr/>	
8.1 Dateien einfügen mit #include	194
8.2 Konstanten und Makros mit #define und #undef	195
8.2.1 Symbolische Konstanten mit #define definieren ...	196
8.2.2 Makros mit #define definieren	198
8.2.3 Symbolische Konstanten und Makros aufheben (#undef)	202
8.3 Bedingte Kompilierung	202
8.3.1 Mehrfaches Inkludieren vermeiden	205
8.4 Programmdiagnose mit assert()	209
8.5 Generische Auswahl	211
8.6 Eigene Header erstellen	213
8.7 Kontrollfragen und Aufgaben	215

9	Arrays und Zeichenketten (Strings)	219
9.1	Was genau sind Arrays?	219
9.1.1	Arrays definieren	220
9.1.2	Arrays mit Werten versehen und darauf zugreifen	221
9.1.3	Initialisierung mit einer Initialisierungsliste	225
9.1.4	Bestimmte Elemente direkt initialisieren	227
9.1.5	Array mit Schreibschutz	227
9.1.6	Arrays mit fester und variabler Länge (VLA) – optional seit C11	227
9.1.7	Arrays mit scanf einlesen	229
9.1.8	Arrays an Funktionen übergeben	230
9.2	Mehrdimensionale Arrays	232
9.2.1	Zweidimensionalen Arrays Werte zuweisen und darauf zugreifen	233
9.2.2	Zweidimensionale Arrays an eine Funktion übergeben	236
9.2.3	Können wir noch mehr Dimensionen verwenden?	238
9.3	Strings (Zeichenketten)	239
9.3.1	Strings initialisieren	239
9.3.2	Einlesen von Strings	241
9.3.3	Unicode-Unterstützung	243
9.3.4	String-Funktionen der Standardbibliothek string.h	245
9.3.5	Sicherere Funktionen zum Schutz vor Speicherüberschreitungen	248
9.3.6	Umwandlungsfunktionen zwischen Zahlen und Strings	248
9.4	Kontrollfragen und Aufgaben	249

10	Zeiger (Pointer)	251
10.1	Zeiger vereinbaren	251
10.2	Zeiger verwenden	252
10.2.1	Explizite Typumwandlung für den byteweisen Zugriff	254
10.3	Zugriff auf den Inhalt von Zeigern	255
10.3.1	Der NULL-Zeiger	257
10.3.2	Deklaration, Adressierung und Dereferenzierung von Zeigern	259
10.4	Zeiger als Funktionsparameter	260
10.5	Zeiger als Rückgabewert	261
10.6	Zeigerarithmetik	264
10.7	Zugriff auf Array-Elemente über Zeiger	265
10.8	Arrays und Zeiger als Funktionsparameter	268
10.8.1	const-Array-Parameter	269
10.9	char-Arrays und Zeiger	271
10.10	Arrays von Zeigern	272
10.11	void-Zeiger	275
10.12	Typqualifizierer bei Zeigern	276
10.12.1	Konstanter Zeiger	277
10.12.2	Zeiger für konstante Daten	277
10.12.3	Konstante Zeiger und Zeiger für konstante Daten	278
10.12.4	Konstante Parameter für Funktionen	278
10.12.5	restrict-Zeiger	279
10.13	Zeiger auf Funktionen	281
10.14	Kontrollfragen und Aufgaben	285

11 Dynamische Speicherverwaltung	289
11.1 Neuen Speicher zur Laufzeit reservieren	290
11.2 Speicherblöcke vergrößern oder verkleinern	295
11.3 Speicherblöcke wieder freigeben	300
11.3.1 Memory Leaks (Speicherlecks)	300
11.4 Kontrollfragen und Aufgaben	302
12 Komplexe Datentypen	305
12.1 Strukturen	306
12.1.1 Strukturtypen deklarieren	306
12.1.2 Definition einer Strukturvariablen	308
12.1.3 Erlaubte Operationen auf Strukturvariablen	308
12.1.4 Deklaration und Definition zusammenfassen	309
12.1.5 Synonyme für Strukturtypen erstellen	310
12.1.6 Selektion von Komponenten einer Strukturvariablen	310
12.1.7 Strukturen initialisieren	314
12.1.8 Nur bestimmte Komponenten einer Strukturvariablen initialisieren	315
12.1.9 Zuweisung bei Strukturvariablen	316
12.1.10 Größe und Speicherausrichtung einer Struktur	317
12.1.11 Strukturen vergleichen	317
12.1.12 Strukturen, Funktionen und Strukturzeiger	317
12.1.13 Array von Strukturvariablen	322
12.1.14 Strukturvariablen als Komponenten in Strukturen	325
12.1.15 Zeiger als Komponenten in Strukturen	330
12.2 Unions	333

12.3 Der Aufzählungstyp enum	336
12.4 Eigene Typen mit typedef deklarieren	338
12.5 Kontrollfragen und Aufgaben	340
13 Dynamische Datenstrukturen	343
13.1 (Einfach) verkettete Listen	343
13.1.1 Ein neues Element in die Liste einfügen	350
13.1.2 Ein Element suchen und ausgeben	352
13.1.3 Ein Element aus der Liste entfernen	353
13.1.4 Das erste Element in der Liste löschen	354
13.1.5 Ein beliebiges Element in der Liste löschen	355
13.2 Doppelt verkettete Listen	357
13.3 Kontrollfragen und Aufgaben	358
14 Eingabe- und Ausgabefunktionen	361
14.1 Verschiedene Streams und Standard-Streams	361
14.1.1 Streams im Textmodus	362
14.1.2 Streams im binären Modus	363
14.1.3 Standard-Streams	363
14.2 Dateien	364
14.3 Dateien öffnen	365
14.3.1 Exklusiver Dateizugriff	368
14.3.2 Weitere Dateifunktionen	369
14.4 Dateien schließen	371
14.4.1 Limit maximal geöffneter Dateien	371

14.5 Fehler oder Dateiende prüfen	372
14.5.1 End-of-File Indicator	372
14.5.2 Error Indicator	373
14.5.3 Informationen zum Fehler ausgeben	373
14.5.4 Fehler- und EOF-Flag zurücksetzen – clearerr()	374
14.6 Weitere Funktionen für die Ein- und Ausgabe	374
14.6.1 Einzelne Zeichen aus einem Stream lesen	375
14.6.2 Zeichen in den Stream zurückstellen	375
14.6.3 Einzelne Zeichen in einen Stream schreiben	375
14.6.4 Zeilenweise aus einem Stream lesen	378
14.6.5 Zeilenweise in einen Stream schreiben	378
14.6.6 Lesen und Schreiben in ganzen Blöcken	383
14.7 Funktionen zur formatierten Ein-/Ausgabe	387
14.7.1 Funktionen zur formatierten Ausgabe	388
14.7.2 Umwandlungsvorgaben für die printf-Familie	389
14.7.3 Weite und Feldbreite	390
14.7.4 Flags	391
14.7.5 Genauigkeitsangaben für Zahlen bei printf()	393
14.7.6 Funktionen für die formatierte Eingabe	396
14.7.7 Umwandlungsvorgaben für die scanf()-Familie	397
14.7.8 Suchmengenkonvertierung	398
14.8 Wahlfreier Dateizugriff	400
14.8.1 Die aktuelle Dateiposition ermitteln	400
14.8.2 Die aktuelle Dateiposition ändern	400
14.9 Sicherere Funktionen mit C11	404
14.10 Datei löschen oder umbenennen	405
14.11 Pufferung	405
14.12 Kontrollfragen und Aufgaben	407

15 Zeitroutinen (time.h)	411
15.1 Die Funktion clock()	411
15.2 Erweiterte Zeitfunktionen	413
15.2.1 Eine Plattformunabhängige delay()-Funktion	415
15.2.2 Der Datentyp time_t	416
15.2.3 Der Datentyp struct tm	418
Anhang	427
A Übersichtstabellen wichtiger Sprachelemente	427
A.1 Operatorpriorität (Operator Precedence)	427
A.2 Reservierte Schlüsselwörter in C	429
A.3 Header-Dateien der Standardbibliothek	430
A.4 Kommandozeilenargumente	432
A.5 Weiterführende Ressourcen	435
B Lösungen der Übungsaufgaben	437
B.1 Antworten und Lösungen zu Kapitel 2	437
B.2 Antworten und Lösungen zu Kapitel 3	438
B.3 Antworten und Lösungen zu Kapitel 4	438
B.4 Antworten und Lösungen zu Kapitel 5	440
B.5 Antworten und Lösungen zu Kapitel 6	443
B.6 Antworten und Lösungen zu Kapitel 7	446
B.7 Antworten und Lösungen zu Kapitel 8	448
B.8 Antworten und Lösungen zu Kapitel 9	451
B.9 Antworten und Lösungen zu Kapitel 10	456
B.10 Antworten und Lösungen zu Kapitel 11	461
B.11 Antworten und Lösungen zu Kapitel 12	465