

Inhalt

Teil I: Eine solide Grundlage

1	Premiere in C	3
	Was ist ein Programm?	3
	Programmieren in C	3
	Alles klar für den Start?	6
	Starten Sie Ihr erstes C-Programm	7
	Das Entwickeln eines Programms	8
	Und wenn <i>CL</i> nicht funktioniert?	11
	Syntaxfehler lokalisieren und korrigieren	13
	Weitere Beispiele	14
	Nochmals: Programmierung	15
	Wichtige Hinweise des Kapitels	17
	Wichtige Begriffe des Kapitels	18
2	<i>printf</i> im Einsatz	21
	Anwendung von Steuerzeichen	21
	Experimente mit dem ANSI-Treiber	26
	<i>printf</i> -Formatspezifikationen	29
	Spezifikationen für Breite und Genauigkeit	34
	Wichtige Hinweise des Kapitels	41
	Wichtige Begriffe des Kapitels	43
3	Variablen, Typen und Operatoren	45
	Variablen im C-Programm	45
	<i>type qualifier</i> – Typspezifikation	51
	Übersichtliche Gestaltung des Programmtextes	54
	Erhöhen und Verringern eines Wertes	56
	Weiteres über Zuweisungen	58
	Welcher Operator hat Vorrang?	59
	Bit-Operatoren	61
	Die Speichererfordernisse der Datentypen	62
	Wichtige Hinweise des Kapitels	65
	Wichtige Begriffe des Kapitels	68

4	Strukturen eines C-Programmes	69
	Relationale Operatoren	69
	Die <i>if</i> -Anweisung	70
	Logische Operatoren	74
	Mit <i>while</i> Statements wiederholen	74
	Die <i>for</i> -Anweisung	78
	<i>if-else</i> verschachtelt	81
	Wichtige Hinweise des Kapitels	84
	Wichtige Begriffe des Kapitels	85
5	Erweiterte Programmstrukturen	87
	Schleifenkonstruktion mit <i>do-while</i>	87
	Die Anweisung <i>continue</i>	88
	Schleifen beenden mit <i>break</i>	90
	Die Schleife in der Schleife in der Schleife..	91
	Auswahl mit <i>switch</i>	92
	Die Sprunganweisung <i>goto</i>	96
	Die bedingte Zuweisung	96
	Lange Kontrollketten für <i>printf</i>	98
	Wichtige Hinweise des Kapitels	100
	Wichtige Begriffe des Kapitels	102
6	Konstanten und Makros definieren	103
	Die Aufgabe des Präprozessors	103
	<i>#define</i> definiert	103
	Mit <i>#define</i> Makros einrichten	107
	Temporäre Definition mit der CL-Option /D	112
	Include-Dateien	112
	<i>putchar</i> gibt Zeichen aus	115
	<i>getchar</i> liest Zeichen ein	116
	Wichtige Hinweise des Kapitels	118
	Wichtige Begriffe des Kapitels	120

7	Zeichenketten	121
	Deklarieren einer Zeichenkette	121
	Zeichenketten-Konstanten	126
	Zeichenketten initialisieren	127
	Ein- und Ausgabe einer Zeichenkette mit <i>gets</i> und <i>puts</i>	128
	Ist 'A' gleich "A" ?	129
	Wichtige Hinweise des Kapitels	130
	Wichtige Begriffe des Kapitels	131
8	Selbstgeschriebene C-Funktionen	133
	Grundlagen der C-Funktionen	133
	An Funktionen Werte als Argumente übergeben	137
	Mehrere Argumente	139
	Das Statement <i>return</i>	140
	Die Angabe <i>void</i>	143
	Include-Dateien	144
	Zeichenketten an Funktionen übergeben	144
	<i>call by value</i>	146
	Rekursion — eine Funktion ruft sich selber auf	147
	Wichtige Hinweise des Kapitels	152
	Wichtige Begriffe des Kapitels	154
9	Die Laufzeitbibliothek	155
	Routinen zur Manipulation von Verzeichnissen	155
	Mathematische Funktionen	158
	Klassifizieren von ASCII-Zeichen	160
	Routinen zum Konvertieren von Zeichen	162
	Numerische Konvertierung	163
	Die Manipulation von Zeichenketten	165
	Die Bedeutung der Laufzeitbibliothek	165
	Wichtige Hinweise des Kapitels	167
	Wichtige Begriffe des Kapitels	168

Teil II: Ausbau der Grundlagen

10 Zeiger in C	173
Adressen und Inhalte von Speicherstellen anzeigen	173
Deklarieren einer Zeigervariable	175
Zeiger und Funktionen	178
Die Grundlagen von <i>scanf</i>	183
Ein Nachtrag zu <i>printf</i>	193
Die Rückgabewerte von <i>printf</i> und <i>scanf</i>	193
<i>scanf</i> und Zeichenketten	194
Weitere Steuerungsmöglichkeiten mit <i>scanf</i>	197
Wichtige Hinweise des Kapitels	199
Wichtige Begriffe des Kapitels	202
11 Vektoren	203
Vektoren anlegen	203
Vektoren an Funktionen übergeben	207
Vektorelemente initialisieren	211
Mehrdimensionale Vektoren	212
Mehrdimensionale Vektoren initialisieren	215
Vereinfachte Verarbeitung eines Vektors	216
Wichtige Hinweise des Kapitels	218
Wichtige Begriffe des Kapitels	219
12 Zeiger für Fortgeschrittene	221
Per Zeiger auf Zeichenketten zugreifen	221
Zeiger-Arithmetik	227
Der Gebrauch von Zeiger-Vektoren	231
Funktionen können Adressen zurückgeben	235
Zeiger auf Funktionen	236
Typ-Konvertierung	239
Funktionen zum Suchen und Sortieren	240
Suchen und Sortieren in einem Vektor von Zeichenketten	247
<i>near pointer / far pointer</i>	249
Wichtige Hinweise des Kapitels	256
Wichtige Begriffe des Kapitels	258

13 Vom Umgang mit Dateien	259
Sinn und Zweck von Dateizeigern	259
Formatierte Datei-Ein- und Ausgabe	268
Zeichen- oder Binärmodus?	269
Noch eine Funktion: <i>ungetc</i>	271
Wichtige Hinweise des Kapitels	272
Wichtige Begriffe des Kapitels	274
14 Weitere Möglichkeiten des Präprozessors	275
Präprozessor-Direktiven für bedingte Verarbeitung	275
<i>#undef</i> hebt Definitionen auf	278
Die Direktive <i>#error</i>	280
Vordefinierte Präprozessor-Konstanten	281
Include-Dateien mit Quelltext	284
Wichtige Hinweise des Kapitels	287
Wichtige Begriffe des Kapitels	288
15 Datenstrukturen	289
Der Nutzen von Strukturen	289
Die Verwendung von Strukturen	293
In Funktionen auf Strukturen zugreifen	295
Strukturen und Vektoren	297
Das Initialisieren von Strukturen	298
<i>unions</i> — spezielle C-Strukturen	298
Verschachtelte Strukturen	300
Bitfelder	301
Wichtige Hinweise des Kapitels	305
Wichtige Begriffe des Kapitels	306

16	Dies & Das	309
	Unabhängig kompilierte Objekt-Dateien	309
	Aufzählungstypen	312
	Mit <i>typedef</i> eigene Datentypen schaffen	314
	Variablen: <i>static</i> oder <i>automatic</i> ?	316
	Noch etwas über globale Variablen	318
	Automatische Typkonvertierung	321
	Die Spezifikationen <i>const</i> und <i>volatile</i>	321
	Komplexe Deklarationen verstehen	324
	Kommentare – einmal anders	325
	Wichtige Hinweise des Kapitels	327
	Wichtige Begriffe des Kapitels	329

Teil III: C voll ausschöpfen

17	Auswerten der Befehlszeile	333
	Erste Schritte	333
	<i>argv</i> als Zeiger	338
	wildcard-Angaben entschlüsseln	339
	Noch einmal: <i>_dos_findfirst</i> und <i>_dos_findnext</i>	341
	Wichtige Hinweise des Kapitels	343
	Wichtige Begriffe des Kapitels	343
18	Zugriff auf die Umgebung	345
	Anzeigen der aktuellen Umgebung	345
	Zugriff auf die Umgebung mit <i>getenv</i>	347
	<i>putenv</i> schreibt in den Umgebungsspeicher	348
	Anwendungen des Umgebungsspeichers	349
	Wichtige Hinweise des Kapitels	351
	Wichtige Begriffe des Kapitels	352

19	Ein Programm beenden	353
	Beenden eines Programms mit <i>exit</i>	353
	Beenden eines Programms mit <i>abort</i>	356
	Testen mit <i>assert</i>	357
	Die Exitliste	359
	<i>main</i> mit <i>return</i> verlassen	360
	Wichtige Hinweise des Kapitels	362
	Wichtige Begriffe des Kapitels	363
20	Umleitung von Ein- und Ausgabe	365
	Operatoren zum Umleiten	365
	Wie man E/A-Umleitung ermöglicht	369
	FehlerNachrichten über <i>stderr</i>	373
	<i>stdprn</i> und <i>stdaux</i>	376
	Die Eintragung <i>FILES</i> in <i>config.sys</i>	376
	Wichtige Hinweise des Kapitels	378
	Wichtige Begriffe des Kapitels	379
21	Dateizugriff mit Pfiff	381
	Bestehende Dateien verändern	381
	Pfadnamen: Synthese und Analyse	389
	Mit schnellen Schritten durch die Datei	391
	Temporäre Dateien	396
	Wichtige Hinweise des Kapitels	400
	Wichtige Begriffe des Kapitels	402
22	Vordefinierte globale Variablen	403
	Angaben über das Betriebssystem	403
	Die Umgebung	404
	Der Standard-Datei-Modus	406
	Unter OS/2: Real- oder Protected-Modus?	407
	Wurde der ANSI-Standard eingehalten?	408
	<i>Program Segment Prefix</i>	408
	Die Zeitmaschine	411
	Fehler erkennen und darstellen	415
	Wichtige Hinweise des Kapitels	417
	Wichtige Begriffe des Kapitels	419

23	Argumente in variabler Anzahl	421
	Die Grundlagen	421
	Auf Argumente im Stack zugreifen	423
	Argumente unterschiedlichen Typs	427
	Eine variable Anzahl von Zeichenketten-Argumenten	430
	<i>vprintf</i> und <i>vfprintf</i>	432
	Von Funktionen, die eine variable Anzahl	
	von Argumenten unterstützen	433
	Wichtige Hinweise des Kapitels	435
	Wichtige Begriffe des Kapitels	436
24	low level-I/O — Elementare Dateizugriffe	437
	Ein Vergleich: <i>stream-I/O</i> und <i>low level-I/O</i>	437
	<i>low level-I/O</i> — ein erstes Anwendungsbeispiel	438
	Informationen über die Dateien	442
	Der Typ einer Dateihandle	445
	<i>lseek</i> und <i>tell</i>	448
	Eine Dateihandle duplizieren	449
	Wichtige Hinweise des Kapitels	451
	Wichtige Begriffe des Kapitels	453
25	Dynamische Speicherzuordnung	455
	Grundsätzliches	455
	<i>fastcopy</i> : Ein erstes Beispiel	458
	Andere Arten der dynamischen Zuordnung von Speicherplatz	460
	Riesige Vektoren	464
	Eine Speicherzuordnung nachträglich ändern	465
	<i>linked lists</i> — verkettete Listen	467
	Ein Vergleich zwischen verketteter Liste und Vektor	473
	Doppelt verkettete Listen	473
	Den Heap prüfen	478
	Überlegungen zum Speichermodell	484
	Wichtige Hinweise des Kapitels	485
	Wichtige Begriffe des Kapitels	487

Teil IV: Professionelles Programmieren

26 Die Compiler-Optionen	491
Anweisungen für den Präprozessor	494
Die Ausgabedateien	498
Die Namen der Objekt-Datei und der ausführbaren Datei ändern	506
Optionen für die Liste	507
Die Wahl des Speichermodells	509
Die Verarbeitung von Gleitkommazahlen	509
Verschiedene Optionsschalter	511
Die Code-Erzeugung	513
Sprach-Spezifisches	515
Die Linker-Optionen	518
Optimieren zwecks besserer Performance	520
Die Umgebungsvariable <i>CL</i>	529
<i>#pragma</i> Compiler-Direktiven	530
Wichtige Hinweise des Kapitels	535
Wichtige Begriffe des Kapitels	540
27 LIB und MAKE vereinfachen das Programmieren	541
Das Arbeiten mit <i>LIB</i>	541
Die Direktive <i>/Zg</i>	548
<i>MAKE</i> hilft beim Ändern von Programmen	550
Unterschiede zwischen <i>NMAKE</i> und <i>MAKE</i>	557
Wichtige Hinweise des Kapitels	559
Wichtige Begriffe des Kapitels	560
28 Die Speichermodelle	561
Das Speichermodell <i>Small</i>	561
Das Speichermodell <i>Medium</i>	565
Das Speichermodell <i>Compact</i>	568
Das Speichermodell <i>Large</i>	571
Das Speichermodell <i>Huge</i>	574
Das Speichermodell <i>Tiny</i>	574
Maßgeschneidertes	574
<i>near heap – far heap</i>	574
Wichtige Hinweise des Kapitels	576
Wichtige Begriffe des Kapitels	577

29	Die Tastatur fest im Griff	579
	Warten auf den Anwender	579
	Direkte Tastaturabfrage	582
	Die Funktion <i>_bios_keybrd</i>	584
	Wichtige Hinweise des Kapitels	592
	Wichtige Begriffe des Kapitels	593
30	Kritische Fehler.....	595
	Die Grundlagen	595
	Die übliche Vorgehensweise	596
	Ein einfaches Beispiel	598
	Wichtige Hinweise des Kapitels	606
	Wichtige Begriffe des Kapitels	606
31	Sprachen-Vielfalt.....	607
	Binär — die Urform	607
	Aufruf einer FORTRAN-Unterroutine	610
	Beehelfsmäßiges Testen der Schnittstelle	612
	AssemblerROUTINEN erstellen und benutzen	612
	Der Inline-Assembler	620
	Wichtige Hinweise des Kapitels	622
	Wichtige Begriffe des Kapitels	622
32	Der Aufruf von DOS-Funktionen.....	623
	DOS-Funktionen	623
	Die DOS-Funktionen der Laufzeitbibliothek	630
	Wichtige Hinweise des Kapitels	650
	Wichtige Begriffe des Kapitels	653
33	Das ROM-BIOS	655
	Der Zugriff auf das BIOS	655
	Die BIOS-Routinen der C-Laufzeitbibliothek	657
	Wichtige Hinweise des Kapitels	667
	Wichtige Begriffe des Kapitels	668

34 Bildschirmgestaltung — nur vom Feinsten	669
Welche Video-Möglichkeiten bietet der Rechner?	669
Den Bildschirm löschen	672
Aber bitte mit Farbe: <i>_outtext</i>	673
Die Farbe des Hintergrundes	676
Mehrere Bildschirmseiten benutzen	677
FensterIn im Textmodus	681
Text innerhalb des Fensters positionieren	682
Cursor an — Cursor aus	685
Wichtige Hinweise des Kapitels	688
Wichtige Begriffe des Kapitels	690
35 Prozeßkontrolle und Signalbehandlung	691
Der Aufruf von DOS-Befehlen aus einem Programm	691
Signal-Behandlung	698
Wichtige Hinweise des Kapitels	702
Wichtige Begriffe des Kapitels	704
Anhang	707
ASCII-Zeichensatz	707
Was ist neu an Microsoft C Version 6.0?	710
Liste der Beispielprogramme	714
Tabellen dieses Buches	724
Register	727