Inhaltsverzeichnis

Vorwort

TE	IL 1	Die Sprache C	1
1	Ein	Überblick über C	3
		Der Ursprung von C	3
		C ist eine mittlere Sprache	3
		C ist eine strukturierte Sprache	5
		C ist eine Sprache für Programmierer	7
		Compiler im Vergleich zu Interpretern	8
		Die Form eines C-Programms	9
		Bibliothek und Binden	10
		Getrennte Übersetzung	12
		Übersetzen eines C-Programms	13
		Übersetzen von Programmen mit Stapelbefehlen	13
		Das Speicherabbild von C	14
		Überblick über wichtige Begriffe	14
2	Var	iable, Konstante, Operatoren und Ausdrücke	17
		Bezeichnernamen	17
		Datentypen	18
٠		Typabwandlung	19
		Zugriffsabwandlung	21
		Erklärung von Variablen	22
		Lokale Variable	22
		Formale Parameter	24
		Globale Variable	25
		Speicherklassenangaben	27
		Extern	28
		Static-Variable	29
		Lokale Static-Variable	29
		Globale Static-Variable	30
		Register-Variable	32
		Zuweisungsbefehle	33
		Typumwandlung in Zuweisungen	33
		Variableninitialisierung	35
		Konstante	35
		Backslash-Zeichenkonstante	36



	Operatoren	37
	Arithmetische Operatoren	3
	inkrement und Dekrement	3
	Verhältnis- und logische Operatoren	40
	Bitweise Operatoren	4:
	Der ?-Operator	40
	Die &- und *-Zeigeroperatoren	4(
	Der Übersetzungszeit-Operator sizeof	4
	Das Kornma als Operator	49
	Der Punkt- (.) und der Pfeil-Operator(->)	5
	Klammern und eckige Klammern als Operatoren	5
	Zusammenfassung der Prioritätsreihenfolge	5
	Ausdrücke	5 [.]
	Typumwandlung in Ausdrücken	5
	Cast	5
	Leerstellen und Klammern	5-
	C-Kurzschreibweise	5
3	Befehle zur Steuerung des Programmflusses	
	True und false	57
	C-Befehle	57
	Bedingungsbefehle	58
	if-Befehl	58
	Geschachtelte if-Befehle	59
	Die if-else-if-Leiter	6
	Die ?-Alternative	6
	switch	64
	Geschachtelte switch-Befehle	67
	Schleifen	` 68
	for	68
	Variationen von for-Schleifen	69
	Die Endlos-Schleife	7:
	for-Schleifen ohne Körper	7:
	while	74
	do-while	76
	break	77
	exit()	79
	goto und Sprungmarken	8

4	Funktionen	85
	Der return-Befehl	85
	Rücksprung aus einer Funktion	85
	Rückgabe von Werten	87
	Regeln zum Geltungsbereich von Funktionen	88
	Funktionsargument	89
	Aufruf über den Wert, Aufruf über Referenz	90
	Erstellen eines Aufrufs über Referenz	91
	Aufruf von Funktionen mit Feldern	92
	argc und argv, die Argumente von main()	96
	Funktionen mit nicht ganzzahligem Rückgabewert	99
	Rückgabe von Zeigem	101
	Rekursion	102
	Zeiger auf Funktionen	104
	Implementierung	107
	Parameter und allgemeingültige Funktionen	107
	Effizienz	107
	Bibliotheken und Dateien	108
	Wie groß sollte eine Programmdatei sein?	108
	. Aufteilen von Dateien	109
	Bibliotheken	109
5	Felder	111
•	Eindimensionale Felder	111
	Die Übergabe von eindimensionalen Feldem an Funktionen	112
	Eindimensionale Felder als Zeichenketten	114
	Zweidimensionale Felder	116
	Felder von Zeichenketten	119
	Mehrdimensionale Felder	121
	Felder und Zeiger	122
	Zugewiesene Felder	124
	Initialisierung von Feldern	128
	Initialisierung von Feldem unbestimmter Größe	129
	Tic-Tac-Toe - ein Beisniel	130

6	Zeiger	135
	Zeiger sind Adressen	135
	Zeigervariable	135
	Die Zeigeroperatoren	136
	Zeigerausdrücke	138
	Zeigerzuweisungen	138
	Zeigerarithmetik	138
	Zeigervergleiche	140
	Dynamische Zuweisungsfunktionen	142
	Zeiger und Felder	143
	Zeiger auf Zeichenfelder	145
	Felder von Zeigem	146
	Zeiger auf Zeiger	148
	Zeigerinitialisierung	149
	Zeiger auf Funktionen	151
	Zeiger sind keine ganzen Zahlen	154
	Probleme mit Zeigern	154
7	Strukturen, Varianten u. benutzerdefinierte Variat	ole 159
	Strukturen	159
	Der Zugriff auf Strukturelemente	161
	Felder von Strukturen	162
	Eine Adressliste - ein Beispiel	162
	Übergabe von Strukturen an Funktionen	168
	Übergabe von Strukturelementen an Funktionen	169
	Übergabe von vollständigen Strukturen an Funktionen	
	one game ver rememberger or arranger arranger	170
	Zeiger auf Strukturen	170 171
	Zeiger auf Strukturen	171
	Zeiger auf Strukturen Erklären eines Zeigers auf eine Struktur	171 171
	Zeiger auf Strukturen Erklären eines Zeigers auf eine Struktur Die Verwendung von Zeigem auf Strukturen	171 171 171
	Zeiger auf Strukturen Erklären eines Zeigers auf eine Struktur Die Verwendung von Zeigem auf Strukturen Felder und Strukturen in Strukturen	171 171 171 175
	Zeiger auf Strukturen Erklären eines Zeigers auf eine Struktur Die Verwendung von Zeigem auf Strukturen Felder und Strukturen in Strukturen Bit-Felder	171 171 171 175 176
	Zeiger auf Strukturen Erklären eines Zeigers auf eine Struktur Die Verwendung von Zeigem auf Strukturen Felder und Strukturen in Strukturen Bit-Felder Varianten	171 171 171 175 176 179

getchar() und putchar() gets() und puts() Formatierte Konsolein-/ausgabe printf() scanf() Das gepufferte Dateisystem Der Dateizeiger 197 Text/Binär fopen() putc() getc() felose() feof(), ferror() und rewind() Die Benutzung von fopen(), getc(), putc() und felose() fesek() und wahlfreie Ein-/Ausgabe getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() 215 read() und write() 216 217 218 219 219 219 210 210 211 211 211 211 212 213 214 215 215 215 216 216 217 217 218 218 218 219 219 210 210 210 211 211 211 211 212 213 214 215 215 216 216 217 217 218 218 218 219 219 219 210 210 210 210 210 210 210 211 211 211	8	Eingabe, Ausgabe und Diskettendateien	187
Septe Description 189		Konsolein-/ausgabe	188
Formatierte Konsolein-/ausgabe printf() scanf() Das gepufferte Dateisystem Der Dateizeiger Text/Binär fopen() putc() getc() folose() foor(), ferror() und rewind() Die Benutzung von fopen(), getc(), putc() und fclose() fesek() und wahlfreie Ein-/Ausgabe getw() und putw() fread() und twrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 9 Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare pedingte Übersetzungsanweisungen ##include Bedingte Übersetzungsanweisungen ##ifdef und #findef #undef #undef #line #pragma Vordefinierte Makronamen			
printf() 191 scartf() 193 Das gepufferte Dateisystem 196 Der Dateizeiger 197 Text/Binär 197 fopen() 198 getc() 198 getc() 199 folose() 200 feof(), ferror() und rewind() 201 Die Benutzung von fopen(), getc(), putc() und folose() 201 fesek() und wahlfreie Ein-/Ausgabe 204 getw() und putw() 205 fread() und fwrite() 206 Lesen und Schreiben anderer Datentypen 212 stdin, stdout und stderr 213 fprintf() und fscartf() 215 Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen 215 open(), creat() und close() 215 read() und write() 217 unlink() 218 Auswahl einer Ein-/Ausgabemethode 221 4uswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare 223 #define #error 226 #error #include 226 Bedingte Übersetzungsanweisungen 227 #fidef und #findef 226 #undef #undef 227 #line #pragma 232 Vordefinierte Makronamen 232			
Das gepufferte Dateisystem 196		_	
Das gepufferte Dateisystem Der Dateizeiger Text/Binär fopen() putc() getc() felose() feof(), ferror() und rewind() Die Benutzung von fopen(), getc(), putc() und fclose() getw() und wahlfreie Ein-/Ausgabe getw() und write() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 9 Der C-Preprozessor und Kommentare per C-Preprozessor und Kommentare per C-Preprozessor und Kommentare getw() ### ### ### ### ### ### ### ### ### #		. •	
Der Dateizeiger 197 Text/Binär 197 Text/Binär 197 fopen() 197 putc() 198 getc() 199 getc() 199 folose() 200 feof(), ferror() und rewind() 201 fosek() und wahlfreie Ein-/Ausgabe 204 getw() und wahlfreie Ein-/Ausgabe 205 getw() und wite() 205 fread() und fwrite() 206 getw() und schreiben anderer Datentypen 212 stdin, stdout und stderr 197 foprintf() und fecanf() 213 Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen 215 open(), creat() und close() 215 read() und write() 217 unlink() 218 Dateien mit wahlfreiem Zugriff und Iseek() 219 Auswahl einer Ein-/Ausgabernethode 221 405 4		_	
Text/Binär 197 fopen() 197 putc() 198 getc() 199 felose() 200 feof(), ferror() und rewind() 201 Die Benutzung von fopen(), getc(), putc() und felose() 201 fesek() und wahlfreie Ein-/Ausgabe 204 getw() und putw() 205 fread() und fwrite() 205 Lesen und Schreiben anderer Datentypen 212 stdin, stdout und stderr 213 fprintf() und fscanf() 213 Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen 215 open(), creat() und close() 215 read() und write() 217 unlink() 219 Dateien mit wahlfreiem Zugriff und Iseek() 215 Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare 223 #define 224 #error 226 #error 226 #error 227 #include #error 226 #error 227 #include #error 228 #eliine 224 #undef 231 #undef 231 #undef 231 #undef 231 #undef 231 #undef 231 #pragma Vordefinierte Makronamen 232		Das gepufferte Dateisystem	
fopen() 197 putc() 198 getc() 199 fclose() 200 feof(), ferror() und rewind() 201 Die Benutzung von fopen(), getc(), putc() und fclose() 201 fseek() und wahifreie Ein-/Ausgabe 204 getw() und putw() 205 getw() und fwrite() 205 Lesen und Schreiben anderer Datentypen 212 stdin, stdout und stderr 213 fprintf() und fscanf() 213 Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen 215 open(), creat() und close() 215 read() und write() 217 unlink() 218 Auswahl einer Ein-/Ausgabemethode 221 Poer C-Preprozessor und Kommentare 223 #define #error 226 #error 226 #include Bedingte Übersetzungsanweisungen 227 #iff, #else, #elif und #endif #line 231 #undef #line 231 #undef #line 231 #pragma Vordefinierte Makronamen 232		Der Dateizeiger	
putc() getc() fclose() fclose() feof(), ferror() und rewind() Die Benutzung von fopen(), getc(), putc() und fclose() fseek() und wahitreie Ein-/Ausgabe getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahitreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 9 Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare 223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #tidef und #findef #undef #undef #line #pragma Vordefinierte Makronamen		Text/Binär	
getc() 198 198 199 1		fopen()	197
fclose() feof(), ferror() und rewind() Die Benutzung von fopen(), getc(), putc() und fclose() fseek() und wahlfreie Ein-/Ausgabe getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und iseek() Auswahl einer Ein-/Ausgabemethode 9 Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare 1223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #fidef und #ifndef #undef #undef #undef #undef #line #pragma Vordefinierte Makronamen		pute()	199
feof(), ferror() und rewind() Die Benutzung von fopen(), getc(), putc() und fclose() fseek() und wahifreie Ein-/Ausgabe getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahifreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare 223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #itdef und #ifindef #undef		getc()	199
Die Benutzung von fopen(), getc(), putc() und fclose() fseek() und wahlfreie Ein-/Ausgabe getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 9 Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare 223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #lidef und #lindef #undef #undef #undef #undef #pragma Vordefinierte Makronamen		fclose()	200
fseek() und wahlfreie Ein-/Ausgabe getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und leek() Auswahl einer Ein-/Ausgabemethode 221 Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare 223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #itdef und #findef #undef #undef #undef #undef #pragma Vordefinierte Makronamen		feof(), ferror() und rewind()	201
getw() und putw() fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 9 Der C-Preprozessor und Kommentare Der C-Preprozessor und Kommentare 223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #tidef und #ifndef #undef #undef #undef #undef #ine #pragma Vordefinierte Makronamen		Die Benutzung von fopen(), getc(), putc() und fclose()	201
fread() und fwrite() Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare Der C-Preprozessor 223 #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #titdef und #ifndef #undef #undef #undef #undef #undef #undef #pragma Vordefinierte Makronamen 223 242 243 244 246 246 246 247 248 249 240 240 240 240 241 241 241 242 244 245 246 246 246 247 246 247 247 248 248 249 240 240 240 240 240 240 240		fseek() und wahlfreie Ein-/Ausgabe	204
Lesen und Schreiben anderer Datentypen stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 221 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #titdef und #findef #undef #undef #undef #undef #pragma Vordefinierte Makronamen 213 215 215 215 215 216 217 217 218 218 219 219 220 221 221 222 223 223 224 226 226 227 227 227 227 228 227 227		getw() und putw()	205
stdin, stdout und stderr fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und lseek() Auswahl einer Ein-/Ausgabemethode 221 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #ifdef und #ifndef #undef #undef #undef #undef #ine #pragma #pragma #pragma #open UNIX-ähnliche Dateiroutinen 215 215 215 216 217 217 218 218 218 218 218 218 218 218 218 218		fread() und fwrite()	206
fprintf() und fscanf() Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 221 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #ifdef und #findef 227 #undef #undef #undef #undef #pragma Vordefinierte Makronamen 232		Lesen und Schreiben anderer Datentypen	212
Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen open(), creat() und close() 215 read() und write() 217 unlink() 219 Dateien mit wahlfreiem Zugriff und Iseek() 219 Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare 223 Der C-Preprozessor 223 #define 224 #error 226 #error 226 #include 226 Bedingte Übersetzungsanweisungen 227 #if, #else, #elif und #endif 227 #ifdef und #ifndef 231 #line 231 #pragma 232 Vordefinierte Makronamen 232		stdin, stdout und stderr	213
open(), creat() und close() read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare Der C-Preprozessor #define 224 #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #tidef und #lindef #undef #undef #line #pragma Vordefinierte Makronamen 215 227 228 229 229 220 221 221 221 222 223 224 226 226 227 227 227 228 228 229 229 220 220 220 220		fprintf() und fscanf()	213
read() und write() unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #tidef und #lindef #undef #undef #pragma #pragma Vordefinierte Makronamen 217 228 229 221 221 223 224 226 226 227 227 227 228 229 229 220 220 221 221 222 223 224 226 226 227 227 227 228 228 229 220 220 220 220 220		Ungepufferte Ein-/Ausgabe - UNIX-ähnliche Dateiroutinen	215
unlink() Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 223 9 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #ifdef und #lindef #undef #undef #line #pragma Vordefinierte Makronamen 219 220 221 222 223 224 226 226 227 227 227 228 229 220 220 221 221 222 223 224 226 226 227 227 227 228 228 229 220 220 220 220 220		open(), creat() und close()	215
Dateien mit wahlfreiem Zugriff und Iseek() Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endlf #itdef und #lindef #undef #undef #undef #line #pragma Vordefinierte Makronamen 219 221 223 224 226 226 227 227 227 228 228 229 229 220 220 220 221 220 220		read() und write()	217
Auswahl einer Ein-/Ausgabemethode 221 9 Der C-Preprozessor und Kommentare 223		unlink()	219
9 Der C-Preprozessor und Kommentare Der C-Preprozessor #define #error #include Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #tifdef und #lindef #undef #undef #line #pragma Vordefinierte Makronamen 223 224 226 227 231 231 232 232 232 232 2332		Dateien mit wahlfreiem Zugriff und Iseek()	219
Der C-Preprozessor 223 #define 224 #error 226 #include 226 Bedingte Übersetzungsanweisungen 227 #if, #else, #elif und #endif 227 #fidef und #ifndef 231 #undef 231 #pragma 232 Vordefinierte Makronamen 232		Auswahl einer Ein-/Ausgabemethode	221
Der C-Preprozessor 223 #define 224 #error 226 #include 226 Bedingte Übersetzungsanweisungen 227 #if, #else, #elif und #endlf 236 #undef 231 #line 231 #pragma Vordefinierte Makronamen 232	9	Der C-Preprozessor und Kommentare	223
#error 226 #include 226 Bedingte Übersetzungsanweisungen 227 #if, #else, #elif und #endlf 227 #itdef und #lindef 231 #undef 231 #line 231 #pragma Vordefinierte Makronamen 232		_ •	223
#include 226 Bedingte Übersetzungsanweisungen 227 #if, #else, #elif und #endif 230 #ifdef und #ifndef 231 #undef 231 #line 231 #pragma 232 Vordefinierte Makronamen 233		#define	224
Bedingte Übersetzungsanweisungen #if, #else, #elif und #endif #tifdef und #lindef #undef #line #pragma Vordefinierte Makronamen 227 ##ifdef und #ifndef 230 ##ifdef und #ifndef 231 ##ifdef und #ifndef 231 ##ifdef und #ifndef 232 ##ifdef und #ifndef 233 ##ifdef und #ifndef 234 ##ifdef und #ifndef 235 ##ifdef und #ifndef 236 ##ifdef und #ifndef 237 ##ifdef und #ifndef 237 ##ifdef und #ifndef 238 ##ifdef und #ifndef 239 ##ifdef und #ifndef 230 ##ifdef und #ifndef 231 ##ifdef und #ifndef 231 ##ifdef und #ifndef 233 ##ifdef und #ifndef 233 ##ifdef und #ifndef 234 ##ifdef und #ifndef 235 ##ifdef und #ifndef 236 ##ifdef und #ifndef 237 ##ifdef und #ifndef 237 ##ifdef und #ifndef 238 ##ifdef und #ifndef 238 ##ifdef und #ifndef 239 ##ifdef und #ifndef 230 ##ifdef und #ifndef 230 ##ifdef und #ifndef 231 ##ifdef und #ifndef 231 ##ifdef und #ifndef 231 ##ifdef und #ifndef 231 ##ifdef und #ifndef 233 ##ifdef und #ifndef 234 ##ifdef und #ifndef 235 ##ifdef und #ifndef 236 ##ifdef und #ifndef 237 ##ifdef und #ifndef 237 ##ifdef und #ifndef 238 ##ifdef und #ifndef ##ifdef und #ifndef 238 ##ifdef und #ifndef ##ifdef und #ifnd		#error	226
#if, #else, #elif und #endif 227 #ifdef und #lfndef 230 #undef 231 #line 231 #pragma 232 Vordefinierte Makronamen 232		#include	226
#if, #else, #elif und #endif 227 #ifdef und #lfndef 230 #undef 231 #line 231 #pragma 232 Vordefinierte Makronamen 232		Bedingte Übersetzungsanweisungen	227
#Ifdef und #Ifndef 230 #undef 231 #line 231 #pragma 232 Vordefinierte Makronamen 232			227
#line 231 #pragma 232 Vordefinierte Makronamen 232		• •	230
#line 231 #pragma 232 Vordefinierte Makronamen 232		#undef	231
#pragma 232 Vordefinierte Makronamen 232			231
Vordefinierte Makronamen 232			232

TE	L 2	Die C-Standardbibliothek	235
10	Bin	den, Bibliotheken und Definitionsdateien	237
		Der Binder	237
		Getrennte Übersetzung	237
		Relozierbarer Code	238
		Binden mit Overlays	240
		Die C-Standardbibliothek	241
		Bibliotheksdateien / Objektdateien	242
		Definitionsdateien	242
		Makros in Definitionsdateien	243
		Neudefinition von Bibliotheksfunktionen	245
		stddef.h und limits.h	245
11	Ein	-/Ausgabefunktionen	247
		Ströme	247
		Textstrom	248
		Binärstrom	248
		Dateien	248
		Konzept / tatsächliche Implementierung	249
		Ein-/Ausgabefunktionen	250
12	Zei	chenketten- und Zeichenfunktionen	309
13	Ma	thematische Funktionen	341
14	Zei	t-, Datums- u. a. systembezogene Funktionen	359
15	Dyı	namische Speicherzuweisung	375
16	Bild	dschirm- und Grafikfunktionen	393
-			
17	Vei	schiedene Funktionen	407

TEI	L 3	Algorithmen und Anwendungen	437
18	Sor	tieren und Suchen	439
		Sortieren	439
		Klassen von Sortieralgorithmen	440
		Bewertung von Sortieralgorithmen	441
		Bubble-Sort	442
		Sortieren durch Auswahl	446
		Sortieren durch Einfügen	447
		Verbesserte Sortierverfahren	449
		Shell-Sort	449
		Quick-Sort	451
		Auswahl eines Sortierverfahrens	455
		Sortieren von anderen Datentypen	455
		Sortieren von Zeichenketten	455
		Sortieren von Strukturen	457
		Sortieren von Diskettendateien	458
		Sortieren von Dateien mit wahlfreiem Zugriff	458
		Sortieren von sequentiellen Dateien	461
		Suchen	465
		Suchverfahren	465
		Sequentielle Suche	466
		Binärsuche	466
19	Wa	rteschlangen, Stapel, verknüpfte Listen & Warteschlangen	Bäume 469 470
		Die kreisförmige Warteschlange	475
		Stapelspeicher	479
		Verknüpfte Listen	483
		Die einfach verknüpfte Liste	484
		Die doppelt verknüpfte Liste	490
		Eine Adreßliste	494
		Binärbäume	501
20	Spa	ırsam besetzte Felder	511
	- 1	Die verknüpfte Liste als sparsam besetztes Feld	512
		Bewertung der verknüpften Liste	515
		Der Binärbaum als sparsam besetztes Feld	515
		Bewertung des Binārbaums	518
		Das Feld von Zeigern als sparsam besetztes Feld	518
		Bewertung des Feldes von Zeigem	521
		Hash-Verfahren	521
		Bewertung des Hash-Verfahrens	525
		Auswahl eines Verfahrens	526
21	Par	sen und Auswerten von Ausdrücken	529
		Ausdrücke	529
		Zerlegung eines Ausdrucks	531
		Parsen von Ausdrücken	534
		Parsen eines einfachen Ausdrucks	536
		Erweiterung des Parsers um Variable	542
		Syntaxprüfung in einem rekursiven Parser	551

22	Problemlösung mit künstlicher Intelligenz	553
	Darstellung und Terminologie	554
	Kombinatorische Explosion	555
	Suchverfahren	557
	Beurteilung eines Suchverfahrens	557
	Eine grafische Darstellung	559
	Die Suche in die Tiefe	560
	Beurteilung der Suche in die Tiefe	570
	Die Suche in die Breite	571
	Beurteilung der Suche in die Breite	573
	Erweiterung um Erfahrungswerte	574
	Die Suche nach dem Maximum	575
	Beurteilung der Suche nach dem Maximum	580
	Die Suche nach dem Minimum	581
	Beurteilung der Suche nach dem Minimum	583
	Auswahl eines Suchverfahrens	583
	Auffinden mehrerer Lösungen	584
	Entfernen von Pfaden	584
	Entfernen von Knoten	585
	Finden der 'optimalen' Lösung	592
	Zurück zu den 'verlorenen Schlüsseln'	598
23	Verwendung der Systemhilfsmittel	603
	Die 8088-Prozessorfamilie	603
	Die 8088-Unterbrechungen und PC-DOS	605
	Zugriff auf die Systemhilfsmittel im ROM-BIOS	607
	Ändern des Bildschirmmodus	613
	Löschen des Bildschirms	614
	Cursorpositionierung	615
	Verwendung der Scancodes der PC-Tastatur	616
	DOS zum Zugriff auf Systemfunktionen	618
	Prüfen des Tastaturstatus	621
	Benutzung des Druckers	622
	Lesen und Schreiben über die serielle Schnittstelle	622
	Anzeigen des Inhaltsverzeichnisses	623
	Letzte Gedanken über die Benutzung der Systemi	nilfsmittel 623
24	Grafik	625
	Modus und Palette	626
	Pixel schreiben	627
	Pixel mit dem ROM-BIOS schreiben	627
	Pixel direkt schreiben	629
	Linien zeichnen	631
	Rechtecke zeichnen und ausfüllen	636
	Alle Techniken zusammengefaßt	638
	una rasimuan sasamittenäelön	030

ر سرج		. Oothers as Embeddeling a mit O	054
	L 4	Software-Entwicklung mit C	651
25	Sch	nnittstelle zu Assemblerroutinen	653
		Aufrufkonventionen eines C-Compilers	654
		Die Aufrufkonventionen von Microsoft C	655
		Aufrufkonventionen von Aztec C	656
		Erstellen einer Assemblerfunktion	657
		Benutzung von #asm und #endasm	668
		Wann in Assembler codiert werden sollte	669
26	Sof	tware-Engineering mit C	671
		Das Top-Down-Verfahren	671
		Gliedern ihres Programms	672
		Auswahl einer Datenstruktur	673
		Sichere Funktionen	674
		Funktionsprototypen	676
		lint und make	677
		lint	677
		make	679
27	Effi	zienz, Portierbarkeit und Fehlersuche	685
		Effizienz	685
		Die Inkrement- und Dekrement-Operatoren	686
		Verwendung von Registervariablen	687
		Zeiger im Vergleich zur Feldindizierung	689
		Verwendung von Funktionen	690
		Portierung von Programmen	693
		Verwendung von #define	694
		Betriebssystemabhängigkeiten	695
		Unterschiede in der Datengröße	696
		Fehlersuche	697
		Fehler in der Ausführungsreihenfolge	697
		Probleme mit Zeigern	698
		Umdefinieren von Funktionen	700
		Bizarre Syntaxfehler	701
		Fehler durch Abweichung um 1	702
		Bereichsfehler	. 703
		Vergessen von Funktionserklärungen	705
		Fehler bei Aufrufargumenten	706
		Heap-Stack-Kollision	707
		Allgemeine Vorgehensweise bei der Fehlersuche	707
		Die Kunst der Programmpflege	709
		Beheben von Fehlern	709
		Schützen des Quellcodes	710

TEIL 5	Neuere Entwicklungen	713
28 C+	- +	715
	Datenabstraktion	716
	Objekte	717
	Erklärung von Funktionsparametern	718
	Kommentare	719
	Klassen	719
	Erklären von Mitglieder-Funktionen	721
	Konstruktoren und Destruktoren	724
	friend	727
	Overload von Funktionen	727
	Overload von Operatoren	730
	Datenabstraktion und Overload von Operatoren	73 3
	Andere Eigenschaften von C++	734
	new und delete	734
	inline	735
ANHAN	NG A Der Unterschied zwischen UNIX C und	l
	ANSI-Standard C	737
	Weggefallene Schlüsselwörter	737
	Zusätzliche Schlüsselwörter	737
	const	738
	enum	738
	signed	739
	void	739
	volatile	739
	Übergabe von Strukturen	740
	Funktionsprototypen	740
	Standardbibliotheken	741
	Zusätzliche Preprozessoranweisungen	741
Inc	lex	743