

JOACHIM ERDWEG

ASSEMBLER- PROGRAMMIERUNG MIT DEM PC

**Eine schrittweise
und praxisnahe Einführung**

2., verbesserte und erweiterte Auflage



Erdweg, Joachim:

Assembler-Programmierung mit dem PC:
eine schrittweise und praxisnahe Einführung /
Joachim Erdweg. – 2., verb. und erw. Aufl. –
Braunschweig; Wiesbaden: Vieweg, 1992

- IBM ist ein geschütztes Warenzeichen der International Business Machines.
- TASM, TLINK, TLIB, Turbo Debugger und Turbo Pascal sind geschützte Warenzeichen von Borland.
- MS-DOS und MASM sind geschützte Warenzeichen von Microsoft.

Das in diesem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor und der Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

1. Auflage 1991

2., verbesserte und erweiterte Auflage 1992

Alle Rechte vorbehalten

© Springer Fachmedien Wiesbaden 1992

Ursprünglich erschienen bei Friedr. Vieweg & Sohn Verlagsgesellschaft 1992.



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Umschlaggestaltung: Schrimpf & Partner, Wiesbaden

Gedruckt auf säurefreiem Papier

ISBN 978-3-528-14791-4 ISBN 978-3-322-91940-3 (eBook)

DOI 10.1007/978-3-322-91940-3

Vorwort

In den letzten Jahren hat die Verbreitung von Personal Computern rasant zugenommen. In gleichem Maße ist die Zahl derjenigen gestiegen, die sich im Rahmen ihrer Ausbildung oder ihres Berufes mit der Programmierung dieser Rechner auseinandersetzen. Dies geschieht vorrangig in Programmiersprachen wie FORTRAN, COBOL, PASCAL, C oder BASIC.

Will man aber mehr über das erfahren, was bei in einem Computer vor sich geht, wenn ein Programm «läuft» oder wenn es zum Beispiel aus Zeitgründen nicht möglich ist, eine der erwähnten Sprachen zu verwenden, muß man sich auf der untersten Ebene mit dem Rechner auseinandersetzen. Hierzu muß man die Sprache kennen, die der Computer «spricht»: die Maschinen- oder auch Assemblersprache.

Das Ziel des vorliegenden Buches ist es, den Leser in die Assemblerprogrammierung von Personal Computern einzuführen. Dabei werden keinerlei Vorkenntnisse bezüglich des internen Aufbaus oder der Funktionsweise von Mikrocomputern oder deren Komponenten erwartet. Der Leser lernt schrittweise den Aufbau des Prozessors, die Mechanismen eines Betriebssystems und die Funktionsweise der einzelnen Maschinenbefehle kennen. Er erfährt, wie Programme aufgebaut sein müssen, damit sie vom Computer verstanden werden, wie man Assemblerrouitinen in höhere Programmiersprachen einbindet und wie man mit einem Debugger Fehler in einem Programm findet und beseitigt. An Hand von zahlreichen Programmbeispielen werden die erworbenen Kenntnisse praktisch dargestellt und vertieft.

Ich möchte nicht unerwähnt lassen, daß dieses Buch kein Ersatz zu den Originalhandbüchern der von mir eingesetzten Programme darstellt. Eine Alternative zu diesen Unterlagen, die im Programmpaket *Turbo Debugger & Tools* mittlerweile einen Umfang von mehr als 1000 Seiten besitzen, scheint mir wenig sinnvoll. Gedacht ist dieses Buch vielmehr als Ergänzung, um dem Anfänger auf dem Gebiet der Assemblerprogrammierung in übersichtlicher Form einen Weg zu lauffähigen Assemblerprogrammen aufzuzeigen.

Angeregt zu diesem Buch wurde ich während meines Studiums der Elektrotechnik, Schwerpunkt Technische Informatik, an der Fachhochschule Frankfurt. Zwar gibt es sehr gute Bücher über das Betriebssystem MS-DOS und auch über die 80x86-Prozessoren. Eine praxisbezogene Einführung in das Thema Assemblerprogrammierung konnte ich jedoch nicht ausfindig machen. Mit dem vorliegenden Buch hoffe ich, Abhilfe für alle Assemblerneulinge geschaffen zu haben.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ein kurzer Abriß der Computergeschichte	1
1.2	Die Entwicklung der Computersprachen	2
1.3	Einsatzgebiete von Programmiersprachen	2
2	Grundlagen der INTEL-Prozessoren	5
2.1	Die Geschichte der PC-Prozessoren	5
2.2	Aufbau des Prozessors	7
2.3	Prozessor-Register	8
2.3.1	Arbeitsregister	9
2.3.2	Zeigerregister	10
2.3.3	Indexregister	10
2.3.4	Segmentregister	11
2.3.5	Prozessor-Status-Register	12
2.4	Die Adressierung	17
2.4.1	Berechnung der physikalischen Adresse	17
2.4.2	Adressierung mittels Register	21
2.4.3	Adressierung mit Speicheroperand	22
2.5	INTEL-Konventionen	26
3	Grundelemente der Assemblersprache	29
3.1	Datentypen in Assembler	30
3.2	Der Aufbau einer Assemblerzeile	31
3.2.1	Namensfeld	31
3.2.2	Trennzeichen	35
3.2.3	Operationsfeld	37
3.2.4	Operandenfeld	37
3.2.5	Kommentare	39
3.3	Anweisungen an den Assemblierer	41
3.3.1	ORG-Anweisung	41
3.3.2	EQU-Anweisung	42
3.3.3	' \equiv '-Anweisung	43
3.3.4	DUP-Anweisung	43
4	Struktur eines Assemblerprogrammes	45
4.1	Segmentierung	47
4.1.1	Codesegment	48
4.1.2	Datensegment	51
4.1.3	Stapelsegment	51
4.1.4	Vereinfachte Segmentzuweisungen	52
4.1.5	Zugriff auf Segmente	53

4.2	'COM'- und 'EXE'-Programme	57
4.2.1	Die 'COM'-Struktur	59
4.2.2	Die 'EXE'-Struktur	62
5	Grundtechniken der Assembler-Programmierung	67
5.1	Elementare Befehle	68
5.1.1	Datentransport	68
5.1.2	Arithmetische Operationen	69
5.1.3	Boolesche Operationen	71
5.2	Sprungbefehle	74
5.2.1	Unbedingter Sprung	74
5.2.2	Bedingter Sprung	75
5.3	Unterprogramme, Bibliotheken und Makros	80
5.3.1	Aufruf von Unterprogrammen	81
5.3.2	Aufruf von Makros	98
5.3.3	Einbindung von INCLUDE-Dateien	101
5.4	Aufruf von MS-DOS-Funktionen	106
6	Assemblerroutinen in höheren Programmiersprachen	113
6.1	Trennung der Quellen	114
6.2	Von Turbo-Pascal zu Assembler	115
6.2.1	Registerbelegung	117
6.2.2	Deklaration der Unterprogramme	118
6.2.3	Aufruf von Assemblerroutinen	118
6.2.4	FAR und NEAR	120
6.2.5	Reihenfolge der Parameter	120
6.2.6	Turbo-Pascal-Variablen und -Unterprogramme	123
6.2.7	Ergebnisse von Turbo-Pascal-Funktionen	125
6.2.8	Zugriff auf spezielle Datentypen	129
6.2.9	Geschwindigkeitsgewinn	134
6.3	Von C zu Assembler	135
6.3.1	Registerbelegung	137
6.3.2	Deklaration und Aufruf der Unterprogramme	138
6.3.3	Reihenfolge der Parameter	139
7	Fehlervermeidung und -beseitigung	143
7.1	Syntaxfehler	143
7.2	Logische Fehler	144
7.3	Checkliste	145
7.4	Einsatz eines Debuggers	146
7.5	Debug-Strategie	148

Anhang	151
A Befehlssatz der 80x86-Prozessoren	153
A.1 Datentransport	154
A.2 Arithmetische Operationen	156
A.2.1 Addition	156
A.2.2 Subtraktion	157
A.2.3 Multiplikation	157
A.2.4 Division	158
A.2.5 Negation	158
A.2.6 Korrekturbefehle	158
A.3 Bit-orientierte Operationen	160
A.4 Konvertierungsbefehle	163
A.5 Flagsteuerung	163
A.6 Sprungbefehle	164
A.7 Unterprogrammaufrufe	168
A.8 Stringoperationen	170
A.9 Schleifensteuerung	172
A.10 Weitere Operationen	173
B Software-Interrupts	179
B.1 Interrupt 21h	179
B.1.1 Zeichenausgabe	179
B.1.2 Zeicheneingabe	181
B.1.3 Dateioperationen	183
B.2 Interrupt 10h	192
B.2.1 Bildschirm-Steuerung	192
B.2.2 Cursor-Steuerung	195
B.2.3 Textein- und ausgabe	196
B.2.4 Grafikein- und ausgabe	197
C Turbo Assembler	199
D Turbo Linker	203
E Turbo Library Manager	207
F Turbo Debugger	209
Literaturhinweise	213
Verzeichnis der Bilder	215
Verzeichnis der Tabellen	216
Sachwortverzeichnis	218