

Inhaltsverzeichnis

	Vorwort	19
	Einleitung	21
E.1	Was ist Assembler?	21
E.2	Warum Assembler?	21
E.3	Anmerkungen zum Buch	22
	Teil I – Assembler unter DOS	23
I	Theoretische Grundlagen	25
I.1	Eine Einführung in den 8086er Prozessor	25
	I.1.1 Funktionsweise eines PC	25
	I.1.2 Der Speicher	26
	I.1.3 Bits, Bytes & Co.	26
I.2	Die Register	27
	I.2.1 Universalregister	28
	I.2.2 Die Segmentregister	29
	I.2.3 Index- und Zeigeregister	31
	I.2.4 Das Flagregister	31
I.3	Der Stack	34
I.4	Übungsaufgaben	35
2	Mathematisches	37
2.1	Zahlensysteme	37
	2.1.1 Dezimalsystem	37
	2.1.2 Das Dualsystem	38
	2.1.3 Das Hexadezimalsystem	40
	2.1.4 Logische und physikalische Adressen	42
	2.1.5 Wieso kann der Computer rechnen?	42
2.2	Übungsaufgaben	50
3	Erste Programmierschritte	51
3.1	Wir schreiben das erste Programm	51
	3.1.1 Assembler & Linker	51
	3.1.2 Hallo Welt!	52
	3.1.3 Genauere Erläuterungen zum Programm selbst	57
3.2	Übungsaufgaben	59

4	Möglichkeiten des Assemblers	61
4.1	Grundlagen	61
4.1.1	Deklaration der Segmente	61
4.1.2	Der Aufbau eines Assemblerprogramms	63
4.2	Der Umgang mit Daten	64
4.2.1	Variablen	64
4.2.2	Konstanten	68
4.3	Prozeduren und Makros	69
4.3.1	Prozeduren	69
4.3.2	Makros	72
4.4	Übungsaufgaben	75
5	Befehle und Adressierungsarten	77
5.1	Die wichtigsten Befehle	77
5.1.1	Transportbefehle	77
5.1.2	Mathematische Befehle	79
5.1.3	Schiebebefehle	85
5.1.4	Stringbefehle	86
5.1.5	Sprünge und Schleifen	89
5.1.6	Verschiedene Befehle	92
5.1.7	Stackbefehle	94
5.2	COM-Programme	95
5.3	Adressierungsarten	97
5.4	Übungsaufgaben	100
6	DOS unter der Lupe	101
6.1	Interrupts	101
6.1.1	Wie ein Software-Interrupt funktioniert	101
6.2	Ausführbare Dateien	104
6.2.1	COM & EXE	104
6.2.2	Das PSP	107
6.3	Übungsaufgaben	109
7	Programmieren – Die Grundlagen	111
7.1	Mit Texten und Zahlen arbeiten	111
7.1.1	Einlesen eines Textes	111
7.1.2	Zahlenausgabe	112
7.1.3	Zahleneingabe	114
7.1.4	Externe FAR-Prozeduren	116
7.2	Zugriff auf Dateien	119
7.2.1	Das Erstellen und Öffnen von Dateien	120

7.2.2	Daten schreiben und lesen	121
7.2.3	Dateien schließen	122
7.2.4	Beispiel	123
7.3	Die Kommandozeile	124
7.4	Ports	126
7.5	Übungsaufgaben	127
8	Grafikprogramme	129
8.1	Grundwissen	129
8.1.1	Speicher reservieren	129
8.1.2	VGA & Co	130
8.2	Etwas Geschwindigkeitsoptimierung	133
8.3	Das Feuer-Demo	134
8.3.1	Wie das Feuer entsteht	135
8.4	Der Bildbetrachter	142
8.5	Übungsaufgaben	143
9	Speicherresidente Programme	145
9.1	Das Programm	145
10	Der Coprozessor	149
10.1	Grundlagen	149
10.1.1	Beispiele	152
10.2	Wichtige Funktionen	158
10.2.1	Sinus und Cosinus	158
10.2.2	Potenzen	158
10.2.3	Die Exponentialfunktion	159
10.2.4	Logarithmus zur Basis e und 10	160
10.3	Wichtige FPU-Befehle	161
10.3.1	Befehle zur Kontrolle der FPU	161
10.3.2	Stackbefehle	161
10.3.3	Mathematische Befehle	162
10.3.4	Vergleiche	166
10.4	Übungsaufgaben	167
11	Fehlersuche mit dem Debugger	169
11.1	Grundlagen	169
11.2	Ein Beispielprogramm	169
11.2.1	Die Fehlersuche	170
11.2.2	Mehr Komfort	175

II.3	Übersicht über Turbo-Debugger.....	176
II.4	Übungsaufgaben	176
I2	Assembler in Hochsprachen	177
I2.1	Assembler und Pascal	177
I2.1.1	Der externe Assembler	177
I2.1.2	Der integrierte Assembler	179
I2.2	Assembler und C	181
I2.3	Übungsaufgaben	184
I3	Der Befehl CPUID	185
I3.1	Grundlagen.....	185
I3.2	Den CPUID-Befehl anwenden	186
I3.3	Die erweiterten Funktionen	189
I4	MMX und 3DNow!	193
I4.1	Was steckt dahinter	193
I4.2	Wo ist der Unterschied?	194
I4.3	Die neuen Register.....	194
I4.4	Datentypen	195
I4.5	Vorbereitungen zum Einsatz der Multimedia-Extensionen	195
I4.5.1	Verfügbarkeit von MMX testen	195
I4.5.2	Verfügbarkeit von 3DNow! testen	196
I4.5.3	Vorbereitung der Register	196
I4.6	Beispiel einer Addition mit MMX.....	198
I4.7	Befehle	201
I4.7.1	MMX-Befehle	202
I4.7.2	3DNow!-Befehle	210
I4.7.3	DSP-Befehle	218
I4.8	Übungsaufgaben	219
Teil II – Assembler unter Windows		221
I5	Grundlagen zur Windows-Programmierung	223
I5.1	Allgemeines	223
I5.1.1	Der ewige Krieg	223
I5.1.2	Welchen Assembler sollten Sie benutzen?	224
I5.1.3	Was ist ein API?	225
I5.1.4	ANSI- und Unicode-Funktionen	226
I5.1.5	Extended-Funktionen	226
I5.1.6	Was sind DLLs?	226

15.1.7	Was sind Libraries?	227
15.1.8	Was bleibt noch zu sagen?	227
15.2	Die MessageBox	228
15.2.1	Eine MessageBox (TASM-Version)	228
15.2.2	MASM-Version	231
15.3	Was ist neu?	233
15.3.1	32-Bit	233
15.3.2	Das Ring-Prinzip	234
15.3.3	Interrupts	234
15.3.4	Das FLAT-Modell und der Protected-Mode	234
15.4	Ausführliche Erläuterung des ersten Programms	234
15.4.1	Die Parameter beim Kompilieren und Linken	236
15.4.2	MessageBoxA	237
15.4.3	ExitProcess	239
15.4.4	Symbolische Namen und weitere Include-Dateien	240
15.5	Übungsaufgaben	241
16	Fenster, Dialogboxen und Ressourcen	243
16.1	Ein Fenster in Assembler	243
16.2	Ein Fenster in C	247
16.3	Erläuterungen	250
16.3.1	Objekte und Module	250
16.3.2	Client-Area	250
16.3.3	Gerätekontext	250
16.3.4	Handles	250
16.3.5	Nachrichten	250
16.3.6	Aufbau der MSG-Struktur	251
16.3.7	Verschiedene Nachrichten	252
16.3.8	Nachrichten explizit senden	254
16.3.9	Threads	254
16.4	Beschreibung des Listings	254
16.4.1	Ablauf	255
16.4.2	Menüs	262
16.4.3	Ressource-Dateien	265
16.4.4	Steuerelemente	268
16.4.5	Anweisungen	272
16.4.6	Beispieldialogfenster	273
16.4.7	Funktionen	279
16.5	Multilingualismus	287

16.6	Standard-Dialogfenster	288
16.6.1	GetOpenFileName	288
16.6.2	GetSaveFileName	290
16.7	Übungsaufgaben	290
17	Tools	291
17.1	MASM32 Prostart.	291
17.1.1	Benutzung von Prostart	291
17.1.2	Fazit	296
17.2	Wie man eigene Libraries erzeugt	296
17.2.1	Unter MASM	296
17.2.2	Unter TASM	299
17.2.3	Fazit	300
17.3	Übungsaufgaben	300
18	Debugging und Exception-Handling	301
18.1	Welche Fehler gibt es?	301
18.2	SEH	302
18.2.1	Was ist eine Exception?	304
18.3	Fehler lokalisieren	305
18.3.1	Mit Debugger	305
18.3.2	Ohne Debugger	306
18.4	DbgWin.	306
18.4.1	Die Oberfläche von DbgWin	307
18.4.2	Benutzung von DbgWin	307
18.5	Exception-Handler	309
18.5.1	Finale Exception-Handler	309
18.5.2	Per-Thread-Exception-Handler	313
18.5.3	Ein eigener Exception-Handler	314
18.6	Weitere Informationen zum Thema.	316
18.7	Übungsaufgaben	316
19	Prozesse, Threads und Timer	317
19.1	Prozesse	317
19.1.1	Prüfen der Internetverbindung	317
19.1.2	Benutzte Funktionen	319
19.1.3	Prozesse aus laufenden Prozessen starten	320
19.2	Threads	321
19.3	Thread-Programm	322
19.4	Prioritäten.	323

19.5	Thread-Funktionen	323
19.5.1	CloseHandle	323
19.5.2	CreateThread	324
19.5.3	Sleep	324
19.6	Synchronisation	324
19.6.1	WaitForSingleObject	325
19.6.2	Synchronisation des Internetverbindungs-Beispiels	325
19.7	Timer	328
19.8	Übungsaufgabe	331
20	Dateien	333
20.1	Ein einfacher TextViewer	333
20.1.1	GetWindowText	340
20.1.2	SetWindowText	340
20.2	Verwendete Dateifunktionen	340
20.2.1	CopyFile	340
20.2.2	CreateFile	341
20.2.3	DeleteFile	342
20.2.4	ReadFile	343
20.2.5	WriteFile	343
20.3	Weitere Dateifunktionen	343
20.3.1	GetFileSize	343
20.3.2	MoveFile	344
20.3.3	SetFilePointer	344
20.4	Übungsaufgaben	345
21	DLLs	347
21.1	Grundlegendes über DLLs	347
21.2	Load Time Dynamic Linking und Run Time Dynamic Linking	348
21.3	Run Time Dynamic Linking in der Praxis	349
21.3.1	LoadLibrary	350
21.3.2	GetProcAddress	350
21.4	Eigene DLLs erstellen	351
21.4.1	Funktionen	351
21.4.2	Das DLL-Modul	353
21.4.3	Das Assembler-Testprogramm	356
21.4.4	Das C-Testprogramm	356
21.4.5	Das Delphi-Testprogramm	357
21.4.6	Das Visual-Basic-Testprogramm	358
21.5	Übungsaufgaben	359

22	Registry	361
22.1	Aufbau der Registry	361
22.2	Funktionen	362
22.2.1	Überblick	362
22.2.2	RegCloseKey	363
22.2.3	RegCreateKeyEx	363
22.2.4	RegDeleteKey	365
22.2.5	RegDeleteValue	365
22.2.6	RegOpenKeyEx	365
22.2.7	RegQueryValueEx	366
22.2.8	RegSetValueEx	367
22.3	Übungsaufgabe	368
23	Behind the Scenes	369
23.1	Der Protected-Mode	369
23.1.1	Privileg-Level	370
23.2	Speicherverwaltung	371
23.2.1	DPL & RPL	373
23.3	Task-Switch.	373
23.3.1	Multitasking	373
23.4	Datenzugriffe	374
23.5	Gates	374
23.5.1	Call-Gate	374
23.5.2	Task-Gate	374
23.5.3	Interrupt-Gate	374
23.5.4	Trap-Gate	374
23.6	Interrupts	375
23.7	Shadow-Register.	375
23.8	Paging	375
23.9	VMM	376
23.10	Fazit.	377
23.11	Übungsaufgaben	377
Teil III – Optimieren		379
24	Optimieren	381
24.1	Von Fakten und Mythen	381
24.1.1	Assembler in Computerspielen	381
24.1.2	Das Pareto-Prinzip	381
24.1.3	The root of all evil	382

24.1.4	Die 2 Regeln des Optimierens	382
24.1.5	Sünden im Namen der Dummheit	383
24.1.6	Schnelligkeit ist wichtig	384
24.1.7	Weniger Codezeilen = Schnellerer Code	384
24.1.8	Einmal schnell, immer schnell	385
24.1.9	Compiler ist gleich Compiler!?	385
24.1.10	Und nun?	387
25	Vor dem Code-Tuning	389
25.1	Anforderungen	389
25.1.1	Welche Hardware wird vorausgesetzt?	389
25.1.2	Welches System wird vorausgesetzt?	389
25.1.3	Akzeptanz des Benutzers	390
25.1.4	Das ‚Not invented here‘-Syndrom	390
25.1.5	Größe vs. Geschwindigkeit	391
25.1.6	Design	391
25.2	Profiling	391
25.2.1	Der subjektive Eindruck	392
25.2.2	Den Anwender ablenken	392
25.2.3	Code-Fragmente messen	393
25.2.4	Die gesamte Applikation messen	393
25.2.5	Der Profiler	393
25.2.6	Flaschenhalse entschärfen	396
26	Klassische Flaschenhalse	397
26.1	Eingabe- und Ausgabe-Operationen	397
26.1.1	Dateien	397
26.1.2	Netzwerkzugriffe	398
26.1.3	Bildschirm-/Loggingausgaben	398
26.2	Das System.	398
26.2.1	Systemaufrufe	399
26.2.2	Paging	399
26.3	Berechnungen	399
26.3.1	Caching	399
26.3.2	Multithreading	399
26.3.3	Näherungswerte vorberechnen	399
26.3.4	Shiften	400
26.4	Boolesche Tests	400
26.5	Schleifen.	400

26.6	Variablen.....	401
26.6.1	Abhängigkeiten	401
26.6.2	Casts	402
26.6.3	Datentypen	402
26.6.4	Alignment	402
26.6.5	Deklaration	402
27	Beispiele der Verschwendung	403
27.1	Verschwendete Rechenzeit 1	403
27.2	Verschwendete Rechenzeit 2.	403
27.3	Verschwendeter Speicherplatz	404
27.4	Verschwendeter Speicherplatz 2.	405
27.5	Verschwendetes Geld.....	406
27.6	Fazit.....	408
28	Vorgehensweise	409
29	Ein praktisches Beispiel	411
29.1	Der Code.....	412
29.2	Die Funktionen	418
29.3	Eine optimierte Version.....	418
29.3.1	Wo anfangen?	419
29.3.2	Die Funktion intersection	421
29.3.3	Die Funktion nearest	427
29.3.4	Die Funktion SkalarProdukt	428
29.3.5	Die Funktion isEqualPoint	431
29.3.6	Die restlichen Funktionen	431
29.3.7	Der abschließende Profiler-Durchgang	432
29.3.8	Veränderung in Zahlen	433
29.3.9	isEqualPoint im Disassembler	433
29.4	Fazit.....	437
30	Reverse Engineering	439
30.1	Trojaner-Infektion im MASM32-Package?.....	439
30.1.1	Analyse der Datei	440
30.1.2	Was das Programm wirklich macht	453
30.1.3	Ist das immer so leicht?	453
30.2	Eigenen Code reversen	454
30.2.1	Assembler-Listing erstellen	454
30.3	Letzte Anmerkung	456

Teil IV – Anhang	457
A MASM32-Installation	459
A.1 Meldungen von Virenscannern	460
A.2 Pfad einrichten	460
A.3 DOS-Dateien erzeugen	463
A.4 Probleme mit älteren Versionen	464
B NABFBFSM	465
B.1 Unterschiede	465
B.2 Hello World	466
C AT&T-Syntax	469
D Linux	471
D.1 Was ist neu?	471
D.1.1 Was bei NASM zu beachten ist	472
D.2 Aufbau eines Assembler-Programms	472
D.2.1 .data	473
D.2.2 .bss	473
D.2.3 .text	473
D.3 Systemaufrufe	473
D.4 Hello World	474
D.5 Funktionen	474
D.5.1 Programm beenden	475
D.5.2 Eingabe	475
D.5.3 Ausgabe	475
D.5.4 Öffnen	476
D.5.5 Schließen	476
D.5.6 Verzeichnis wechseln	476
D.5.7 Verzeichnis erzeugen	477
D.5.8 Verzeichnis löschen	477
D.5.9 Weitere Funktionsaufrufe	477
E Wichtige Interruptfunktionen	479
E.1 Zeichen-Ein- und Ausgabe	480
E.2 Verzeichnis und Dateiverwaltung	481
E.3 Speicher-Verwaltung	484
E.4 System- und Programmfunktionen	485
E.5 Grafikfunktionen	486
E.6 Mausfunktionen	487

F	Win32-API-Funktionen	491
F.1	Beep	491
F.2	CheckDlgButton	491
F.3	CheckRadioButton	492
F.4	CloseHandle	493
F.5	CopyFile	493
F.6	CreateDialog	494
F.7	CreateDialogParam	494
F.8	CreateFile	495
F.9	CreateThread	497
F.10	CreateWindowEx	499
F.11	DefDlgProc	510
F.12	DefWindowProc	511
F.13	DeleteFile	511
F.14	DisableThreadLibraryCalls	512
F.15	DispatchMessage	512
F.16	DlgDirList	513
F.17	DlgDirListComboBox	514
F.18	DlgDirSelectComboBoxEx	514
F.19	DlgDirSelectEx	514
F.20	EndDialog	515
F.21	ExitProcess	516
F.22	ExitThread	516
F.23	FreeLibrary	516
F.24	FreeLibraryAndExitThread	517
F.25	GetDlgCtrlID	517
F.26	GetDlgItem	518
F.27	GetDlgItemText	518
F.28	GetExitCodeThread	519
F.29	GetFileSize	519
F.30	GetMessage	520
F.31	GetModuleHandle	521
F.32	GetOpenFileName	522
F.33	GetProcAddress	522
F.34	GetSaveFileName	523
F.35	GetThreadPriority	523
F.36	GetWindowText	524
F.37	InternetCloseHandle	524

F.38	InternetGetConnectedState	525
F.39	InternetOpen	525
F.40	InternetOpenUrl	526
F.41	IsDlgButtonChecked	527
F.42	IsDialogMessage	527
F.43	KillTimer	528
F.44	LoadCursor	528
F.45	LoadIcon	530
F.46	LoadLibrary	530
F.47	LoadMenu	531
F.48	MessageBox	531
F.49	MoveFile	535
F.50	PostQuitMessage	536
F.51	ReadFile	536
F.52	RegCloseKey	537
F.53	RegCreateKeyEx	537
F.54	RegDeleteKey	539
F.55	RegDeleteValue	540
	F.55.1 RegisterClassEx	541
F.56	RegOpenKeyEx	541
F.57	RegQueryValueEx	543
F.58	RegSetValueEx	544
F.59	ResumeThread	545
F.60	SetDlgItemText	546
F.61	SendDlgItemMessage	546
F.62	SendMessage	547
F.63	SetFilePointer	547
F.64	SetMenu	548
F.65	SetThreadPriority	549
F.66	SetTimer	549
F.67	SetWindowText	550
F.68	ShellExecute	551
F.69	ShowWindow	553
F.70	Sleep	554
F.71	SuspendThread	554
F.72	UpdateWindow	555
F.73	WaitForSingleObject	555
F.74	WriteFile	556

G	Strukturen	557
G.1	MSG	557
G.2	CREATESTRUCT.....	558
G.3	OPENFILENAME.....	558
G.4	WNDCLASSEX	560
H	Tools, Links und Literatur	563
H.1	IDEs.....	563
H.2	Foren.....	563
H.3	Assembler.....	564
H.4	Debugger/Disassembler	565
H.5	Sonstige Tools	566
H.6	Linux Links.....	566
H.7	Sonstige Links	566
H.8	Bücher.....	567
I	Nachrichten	569
J	Lösungen zu den Übungsaufgaben	573
J.1	Kapitel 1.....	573
J.2	Kapitel 2	573
J.3	Kapitel 3	573
J.4	Kapitel 4	574
J.5	Kapitel 5	574
J.6	Kapitel 6	575
J.7	Kapitel 7	575
J.8	Kapitel 8	579
J.9	Kapitel 10	580
J.10	Kapitel 12	580
J.11	Kapitel 14	581
J.12	Kapitel 15.....	581
J.13	Kapitel 16	582
J.14	Kapitel 17.....	583
J.15	Kapitel 18	583
J.16	Kapitel 19	584
J.17	Kapitel 21	584
J.18	Kapitel 22	584
K	Über die CD-Rom	587
	Stichwortverzeichnis	589