

Inhaltsverzeichnis

| | |
|--|-----------|
| Vorwort des Herausgebers | XII |
| Vorwort der Autoren | XIII |
| 1 Softwareengineering und LISP | 1 |
| 1.1 LISP – eine Programmiersprache nur für die Künstliche Intelligenz (KI)? | 1 |
| 1.1.1 Eigenschaften effizienter Programmiersprachen | 2 |
| 1.1.2 Eigenschaften der KI-Programmierung | 2 |
| 1.1.3 LISP | 4 |
| 1.2 Der Verbreitungsgrad der Programmiersprache LISP | 5 |
| 1.3 Anforderungen an professionelle Software | 9 |
| 1.3.1 Benutzungsfreundlichkeit/Bedienungskomfort | 9 |
| 1.3.2 Wartungsfreundlichkeit/Wartbarkeit | 9 |
| 1.3.3 Zuverlässigkeit | 10 |
| 1.3.4 Funktionserfüllung und Funktionsumfang | 11 |
| 1.3.5 Zeitverhalten | 11 |
| 1.3.6 Übertragbarkeit/Portabilität | 11 |
| 1.3.7 Sonstiges | 11 |
| 1.4 Das Festlegen von Qualitätsanforderungen | 12 |
| 1.4.1 Das Softwarepaket WIZDOM | 12 |
| 1.4.2 Qualitätsanforderungen an WIZDOM | 13 |
| 1.5 Der Aufbau des Buches | 15 |
| 1.6 Terminologie und Notation | 17 |
| 1.7 Hinweise | 19 |
| 2 Komplexe LISP-Programme und Programmieretechniken ... | 21 |
| 2.1 Wiederholte Funktionsaufrufe | 21 |
| 2.1.1 Rekursion | 21 |
| 2.1.1.1 Lineare Rekursion | 23 |
| 2.1.1.2 Nicht-lineare Rekursion | 25 |
| 2.1.1.3 Rekursive Verarbeitung: Testverfahren | 26 |
| 2.1.2 Iteration | 30 |

| | | |
|----------|--|------------|
| 2.1.3 | Rekursion oder Iteration? | 32 |
| 2.2 | Gültigkeitsbereich und Ausdehnung von LISP Datenobjekten .. | 34 |
| 2.2.1 | Transfer der Programmkontrolle | 36 |
| 2.2.1.1 | Lokales Verlassen einer Funktion | 37 |
| 2.2.1.2 | Nicht-lokales Verlassen einer Funktion | 39 |
| 2.2.1.3 | Der Schutz vor ungewünschten Programmunterbrechungen ... | 39 |
| 2.3 | Makros | 42 |
| 2.4 | Unterschiede zwischen Common LISP und muLISP | 45 |
| 2.4.1 | Variablen | 45 |
| 2.4.2 | Implizite Auswertemechanismen | 49 |
| 2.4.3 | Sonstige Unterschiede | 50 |
| 2.5 | Zusammenfassung | 51 |
| 3 | Textverarbeitung mit LISP | 53 |
| 3.1 | Allgemeine Prinzipien der Textverarbeitung | 53 |
| 3.2 | Prinzipien der Textverarbeitung mit LISP | 56 |
| 3.3 | Der DHM-Editor | 58 |
| 3.3.1 | Initialisierung der DHM-Editor Konfiguration | 58 |
| 3.3.2 | Die Initialisierung der DHM-Editorfunktionen | 60 |
| 3.3.3 | Textbearbeitung im DHM-Editor | 73 |
| 3.3.4 | Sonstige Aktionen im DHM-Editor | 85 |
| 3.4 | Zusammenfassung | 89 |
| 4 | Der Aufbau eines natürlichsprachlichen Lexikons | 91 |
| 4.1 | Charakteristika natürlichsprachlicher Lexika | 92 |
| 4.2 | Datenstrukturen | 93 |
| 4.2.1 | Die Verkörperung von Datenstrukturen | 94 |
| 4.2.2 | Sucheffizienz und Dateistruktur | 95 |
| 4.2.3 | Auswahl einer lexikalischen Daten- und Dateistruktur | 96 |
| 4.3 | Das Lexikon als Diskriminationsnetzwerk | 97 |
| 4.3.1 | Buchstaben- und kohortenbasierte Diskriminationsnetze | 99 |
| 4.3.1.1 | Lexikonkompilierung | 100 |
| 4.3.1.2 | Dateipflege | 100 |
| 4.3.1.3 | Lexikonverwaltung | 101 |
| 4.3.1.4 | Sucheffizienz | 101 |
| 4.4 | Buchstaben- oder kohortenbasiertes Lexikon? | 102 |
| 4.4.1 | Grundlagen der Implementierung eines Kohortennetzes | 103 |
| 4.4.2 | Die Struktur der lexikalischen Datendateien in VIRTEX | 105 |
| 5 | Die Compilierung des Lexikons | 109 |
| 5.1 | Aufgaben der Lexikonkompilierung | 110 |
| 5.2 | Dateioperationen mit muLISP | 111 |
| 5.3 | Prinzipien der lexikalischen Suche | 114 |
| 5.4 | Sonderfälle der lexikalischen Suche in VIRTEX | 116 |

| | | |
|----------|---|------------|
| 5.5 | Der Compilierungsalgorithmus | 116 |
| 5.5.1 | Initialisierung | 119 |
| 5.5.2 | Netzwerkaktivierung | 122 |
| 5.5.3 | Das Lesen der Originaldatei | 123 |
| 5.5.4 | Die lexikalische Suche | 125 |
| 5.5.5 | Die lexikalische Eintragung | 129 |
| 5.5.6 | Das Lesen bestehender Subnetze | 133 |
| 5.5.7 | Das Schreiben von Subnetzen | 135 |
| 5.5.8 | Der Aufbau und die Verwaltung der Netzwerke | 139 |
| 5.5.9 | Das Schreiben der Lexikoneinträge | 142 |
| 5.5.10 | Die Sicherung der erzeugten Netzwerke | 144 |
| 5.5.11 | Utilities | 146 |
| 5.6 | Zusammenfassung | 146 |
| 6 | Der lexikalische Verarbeitungsalgorithmus und der DHM-Editor | 149 |
| 6.1 | Initialisierung | 150 |
| 6.2 | Lexikalische Suche und orthographische Anpassungen | 154 |
| 6.3 | VIRTEX und der DHM-Editor | 167 |
| 6.3.1 | Wortübergabe an VIRTEX | 168 |
| 6.3.2 | Aktivierung der lexikalischen Suche | 171 |
| 6.3.3 | Phonetische Transkription | 173 |
| 6.3.4 | VIRTEX Kommentare | 177 |
| 6.3.5 | Veränderte Compilierungsfunktionen und Utilities | 179 |
| 6.4 | Zusammenfassung | 180 |
| 7 | Der PC | 185 |
| 7.1 | Grundlagen | 185 |
| 7.1.1 | Die Eingabeeinheit | 186 |
| 7.1.2 | Die Ausgabeeinheit | 186 |
| 7.1.3 | Die externe Speichereinheit | 186 |
| 7.2 | Speicheraufbau und Speicherverwaltung | 187 |
| 7.3 | Adressierung | 189 |
| 7.4 | Register | 190 |
| 7.5 | Die Interrupts | 193 |
| 7.6 | Das BIOS | 195 |
| 7.7 | Der Bios-Datenbereich | 196 |
| 7.8 | Zusammenfassung | 199 |
| 8 | Kommunikation mit DOS | 201 |
| 8.1 | Einfache Kommunikation mit dem Betriebssystem | 201 |
| 8.1.1 | EXECUTE | 201 |
| 8.1.2 | GETSET | 204 |
| 8.1.3 | Die Abfrage des ERRORLEVEL | 206 |

| | | |
|-----------|---|------------|
| 8.2 | Direktzugriff auf Speicherbereiche | 207 |
| 8.2.1 | MEMORY | 208 |
| 8.2.2 | Computertyp und BIOS-Datum | 209 |
| 8.2.3 | System-Informationen | 212 |
| 8.3 | Der Zugriff auf DOS-Interrupts | 216 |
| 8.3.1 | REGISTER und INTERRUPT | 217 |
| 8.3.2 | Einfache Interrupt-Routinen | 218 |
| 8.3.2.1 | Hilfsfunktionen | 218 |
| 8.3.2.2 | Datum und Zeit durch Interruptaufrufe | 219 |
| 8.3.2.3 | Disketteninformationen | 224 |
| 8.3.2.4 | Die Verwaltung von Unterverzeichnissen | 231 |
| 8.4 | Zusammenfassung | 238 |
| 9 | Bildschirm und Graphik | 241 |
| 9.1 | Der Bildschirmspeicher | 242 |
| 9.1.1 | Der Aufbau des Bildschirmspeichers | 242 |
| 9.1.2 | MEMORY und das Attribut-Byte | 245 |
| 9.1.3 | SNAPSHOT | 248 |
| 9.2 | Der Videointerrupt | 253 |
| 9.2.1 | Die Anpassung von Programmen an die installierte Graphikkarte | 254 |
| 9.2.2 | Das Löschen des Bildschirms | 256 |
| 9.3 | Graphik, einfache Zeichnungen | 258 |
| 9.3.1 | Beispielfunktionen zur Graphikprogrammierung | 258 |
| 9.3.2 | Linien | 259 |
| 9.3.3 | Kästen und Rauten | 261 |
| 9.4 | Zusammenfassung | 264 |
| 10 | Benutzerfreundliche Menüs mit LISP | 267 |
| 10.1 | Einfache Menüs mit LISP | 267 |
| 10.2 | Das Menü des WIZDOM-Pakets | 268 |
| 10.2.1 | Die Programmierung des WIZDOM-Menüs | 270 |
| 10.2.1.1 | Die Auswahl der Menüsprache | 270 |
| 10.2.2.2 | Die Initialisierung des Menüs | 273 |
| 10.2.2.3 | Farboperationen im Menü | 276 |
| 10.2.2.4 | Das Auswerten der Tastatur | 278 |
| 10.2.2.5 | Die Steuerung des Menüs | 281 |
| 10.2.2.6 | Das Öffnen der Submenüs | 282 |
| 10.2.2.7 | Hilfsfunktionen | 283 |
| 10.3 | Zusammenfassung | 284 |
| 11 | Der Aufbau und die Struktur des muLISP-Interpreters | 287 |
| 11.1 | Der interne Aufbau von muLISP | 287 |
| 11.2 | Datenobjekte und Datenstrukturen in muLISP | 288 |
| 11.2.1 | Symbolische Datenobjekte – Symbols | 289 |

| | | |
|-----------|--|------------|
| 11.2.2 | Numerische Datenobjekte – Numbers | 293 |
| 11.2.3 | Listen – Conses | 296 |
| 11.3 | Die Initialisierung des Hauptspeichers | 296 |
| 11.4 | Das Laden und Linken eigener Routinen | 298 |
| 11.5 | Arbeitsmittel | 301 |
| 11.6 | Die Praxis der Unterprogrammtechnik unter muLISP | 304 |
| 11.6.1 | Die Erzeugung linkfähiger Maschinenprogramme | 304 |
| 11.6.2 | Der Systemstatus zum Zeitpunkt eines Unterprogrammaufrufes | 307 |
| 11.7 | Zusammenfassung | 309 |
| 12 | Einfache Assemblerrouitinen und LISP | 311 |
| 12.1 | Argumentübergabe an interne Prozeduren | 311 |
| 12.2 | Differenzierung in Datentypen | 313 |
| 12.3 | Die Auswertung der Zeigerstrukturen | 319 |
| 12.4 | Lineare Rekursion in Assembler | 321 |
| 12.5 | Labeling | 327 |
| 12.6 | Die Übergabe ausgewerteter Parameter an externe Prozeduren | 330 |
| 12.6.1 | Die Reservierung eines Datensegmentes | 331 |
| 12.6.2 | Die Nutzung des muLISP-Datensegmentes als Puffer für Eingabeparameter | 334 |
| 12.6.3 | Die Übergabe von Parametern an externe Prozeduren über den Stack | 337 |
| 12.6.4 | Die direkte Übergabe der Eingabeparameter in Registern | 346 |
| 13 | Die Integration komplexer Assemblerrouitinen in LISP | 351 |
| 13.1 | Benutzerschnittstellen in LISP | 352 |
| 13.1.1 | Grundlagen für die Einbindung einer Maus | 356 |
| 13.1.2 | Die Mauseinbindung mit LISP | 356 |
| 13.2 | Ein komplexes Assemblerprogramm | 358 |
| 13.2.1 | Der Programmablauf | 358 |
| 13.2.2 | Aufgaben und Aufbau der benötigten Dateien | 358 |
| 13.2.3 | Die Realisierung einer Programmunterbrechung durch externe Signale | 361 |
| 13.2.4 | Die Manipulation des LISP-Interpreters | 364 |
| 13.3 | Der muLISP-Initialisierungsteil GENERATE.LSP | 367 |
| 13.4 | Die Erzeugung des Hauptprogramms EDITM.EXE | 376 |
| 13.4.1 | Die Initialisierung des Hauptprogramms EDITM.EXE | 380 |
| 13.4.2 | Die Funktionen der Datei CALLER.LSP | 385 |
| 13.4.3 | Die Beispieldatei BEISP1.LSP | 387 |
| 13.4.4 | Die Beispieldatei BEISP2.LSP | 389 |
| 13.5 | Der Assemblerteil DRIVER.ASM | 390 |
| 14 | Der Aufbau eines Ladeprogramms | 411 |
| 14.1 | Die Initialisierung des Cacheprogramms | 413 |

| | | |
|----------------|--|------------|
| 14.2 | Das Listing der Datei WIZDOM.ASM | 414 |
| 14.3 | Die Interruptbibliothek INT680 | 421 |
| 14.4 | Der Aufruf der neuen Interruptprozeduren | 422 |
| 14.5 | Der WIZDOM Datei-Manager | 424 |
| 14.6 | Zusammenfassung | 431 |
| 15 | Die Integration der Komponenten des WIZDOM-Pakets ... | 433 |
| 15.1 | Die Module des WIZDOM-Pakets | 434 |
| 15.2 | Die Erzeugung des Gesamtpaketes | 434 |
| 15.2.1 | Aufbau der Datei WIZ-INIT.LSP | 435 |
| 15.2.2 | Modifikationen der Datei GENERATE.LSP | 439 |
| 15.2.3 | Aufbau des Generierungsverzeichnisses | 441 |
| 15.2.4 | Modifikationen der Datei MENU.LSP | 443 |
| 15.2.5 | Erweiterungen des DHM-Editors | 446 |
| 15.2.6 | Zusatzoptionen im DHM-Editor | 450 |
| 15.2.7 | Der Anschluß des Lexikons (VIRTEX) | 455 |
| 15.2.8 | Der Lexikoncompiler | 458 |
| 15.3 | Weitere Optionen und Operationen | 459 |
| 15.4 | Ausblick und weitere Module | 460 |
| 16 | Utilities | 461 |
| 16.1 | Programmbibliotheken | 461 |
| 16.1.1 | Das Erzeugen von .LIB-Dateien | 463 |
| 16.1.2 | Das Laden von Bibliotheksdateien | 464 |
| 16.1.3 | Das Bereitstellen von Funktionen und Dateien | 466 |
| 16.2 | Universelle LISP-Debugging Verfahren | 470 |
| 16.3 | Das Arbeiten mit Spezialzeichen | 471 |
| 16.3.1 | Der ASCII-Code im Graphikmodus | 471 |
| 16.3.2 | Die Darstellung einzelner Zeichen | 472 |
| 16.3.3 | Die Manipulation einzelner Zeichen | 474 |
| 16.3.4 | Das Bearbeiten der Matrix | 474 |
| 16.4 | Das Auslesen des Verzeichnisinhaltes (DIR) | 485 |
| 16.4.1 | Vorbereitung, Variableninitialisierung | 487 |
| 16.4.2 | Disk Transfer Area und Dateiattribute | 489 |
| 16.5 | Allgemeine Utilities | 494 |
| 17 | Zusammenfassung | 501 |
| 18 | Bibliographie | 503 |
| Anhänge | | |
| A | Tastaturcodes | 507 |
| A.1 | Der Scan-Code | 507 |
| A.2 | Die ASCII-Tabelle | 509 |

| | | |
|----------|---|-----|
| B | DHM-Editorfunktionen | 511 |
| B.1 | Bewegungen im Text | 511 |
| B.2 | Texteditierung | 514 |
| B.3 | Sonstige Aufgaben | 518 |
| C | Lexikalische Garbage Collection (VIRTEX) | 523 |
| D | Der Videointerrupt (INT 10h) | 527 |
| E | Der DOS-Interrupt (INT 21hex) | 541 |
| E.1 | Die BIOS-Interrupts | 541 |
| E.2 | Der DOS-Interrupt INT 21h | 544 |
| E.3 | muLISP-Beispielfunktionen | 548 |
| F | Die Einsprungradressen der muLISP-Primitiva | 553 |
| G | Der Maus-Interrupt INT 33h | 559 |
| H | BOX.ASM | 573 |
| I | Die Prozeduren der Datei DRIVER.ASM | 579 |
| K | Die muLISP BASE-PAGE, Systemvariablen und Speicher- adressen | 583 |
| K.1 | Die BASE-PAGE | 583 |
| K.2 | Die Global Storage Area | 588 |
| L | .PEK-Dateien | 591 |
| | Funktionsregister aller im Buch verwendeten LISP-Funktionen und Makros | 597 |
| | Index | 605 |