

Macro-Basic: Die Unterprogramm-Bibliothek

Herkömmliche Basic-Erweiterungen sind in der Regel aus einem Guß, mit vielen nützlichen, aber auch unnötigen Befehlen. Und oft fehlen gerade die Funktionen, die man am dringendsten braucht. Damit will Macro-Basic ein für allemal Schluß machen.

An anderen Befehlserweiterungen fehlt immer etwas: Exbasic Level II hat keine Grafikmöglichkeiten, Simons Basic kein richtiges RENUMBER, um nur einige Beispiele zu nennen. Der Gedanke, ein flexibles Software-Instrument zu bauen, welches aus einer Vielzahl von Modulen besteht, die man — je nach aktuellem Bedarf — in den Computer lädt, drängt sich da geradezu auf. Verwirklicht wurde dieses einleuchtende Konzept mit Macro-Basic.

Packt man Macro-Basic aus, dann fällt dem in dieser Hinsicht von Commodore nicht gerade verwöhntem C 64-Besitzer zunächst einmal ein mehrere hundert DIN-A4-Seiten starkes Handbuch auf. Da merken Sie dann, was es heißt, seinen verfügbaren Basic-Befehlssatz in etwa zu vervierfachen. Das ist aber noch nicht alles, denn im Mai soll noch eine Grafikerweiterung auf den Markt kommen mit nochmal 18 neuen Befehlen. Daneben übersieht man fast die doppelseitig beschriebene Diskette und den Dongle, einen Kopierschutzstecker, der in den Kassettenport geschoben werden muß.

Wie ist Macro-Basic aufgebaut? Zwei Softwarepakete bilden derzeit das komplette System, von denen das erste aus einem Ladeprogramm und 260 Befehlen besteht, das zweite aber eine Dateiverwaltungs-Bibliothek mit weiteren 30 Befehlen bereithält. Das Ganze verteilt sich auf 160, größtenteils voneinander unabhängige, Module, die mittels eines Linkprogramms nach Belieben zusammengestellt werden können. Theoretisch verfügt man mit Macro-Basic nicht nur über eine, sondern über $160 \times 160 = 25600$ mögliche Befehlserweiterungen, je nachdem, wie man sich die Module zusammenbastelt. Jedes Modul enthält einen oder meist mehrere Befehle in Form von optimierten Maschinenprogrammen. Das kleinste Modul

beansprucht 7 Byte, das größte — wenn ich mich nicht vertan habe — 3670 Byte an Speicherplatz. Im Schnitt opfert man für jedes Modul etwa 100 bis 200 Byte. Wohin man im Speicher seine Befehlsbibliothek packen will, bleibt im eigenen Ermessen. Eine fertig erstellte Basic-Erweiterung kann unter einem Kennwort abgespeichert werden. Sollten Sie Zugang zu anderen Commodore-Computern haben, dann können Sie die Befehlserweiterung auch auf diese anwenden, denn Macro-Basic ist mit besonderem Augenmerk auf Kompatibilität gebaut worden.

Das Erstellen der individuellen Basic-Erweiterung geschieht relativ einfach im Dialog. Dazu lädt man das erste Programm auf der Macro-Basic-Diskette, gibt ein, auf welchem Commodore-System man arbeitet, steckt auf Anforderung den Datenstecker in den Kassettenport und gibt an, wo im Speicher die Maschinenprogramme abgelegt werden sollen. Man hat dann die Möglichkeit, sich auf dem Drucker ein Protokoll der weiteren Prozedur mitzuschreiben zu lassen. Das empfiehlt sich besonders bei größeren Erweiterungen, weil man nach einiger Zeit sicher nicht mehr weiß, welche Kombinationen man gewählt hat, und wohin man im Speicher das ganze Werk lädt. Die Startadresse braucht man nämlich, um die Erweiterung zu initialisieren. Von nun an ist alles ein Kinderspiel. Das Programm erfragt Stück für Stück die Kennzahlen der einzelnen Module bis man schließlich END eingibt. Dann erteilt man dem Ganzen einen Namen, legt eine formatierte Diskette ins Laufwerk, der Drucker schließt das Protokoll ab und das Ladeprogramm speichert alles ab, bevor es sich diskret aus dem Geschehen zurückzieht. Zum Testen habe ich mir mal eine Erweiterung aus 154 Modulen zusammengestellt, die etwa 23,6 KByte

beansprucht. Das Ganze dauert ungefähr eine halbe Stunde. Wenn man mehr Übung darin hat, reduziert sich der Zeitaufwand sicher noch erheblich. In diesen 23,6 KByte liegen immerhin 260 neue Befehle! Für den Normalgebrauch kommt man sicher leicht mit den 4 KByte im oberen RAM des Commodore 64 (\$C000 bis \$D000) aus.

Benötigt man nun irgendwann eines der auf diese Weise zusammengestellten Instrumente, dann lädt man es unter dem zugeteilten Namen, gibt danach NEW ein und startet mittels SYS Startadresse*. Die Startadresse holt man sich aus dem Protokoll als Startadresse Original. Automatisch wird nun auch der Basic-Speicher begrenzt und alle Befehle der Erweiterung stehen zur Verfügung.

Um welche Befehle es sich dabei handelt, sehen Sie aus der beigefügten Tabelle. Es ist allerdings unmöglich, in diesem Rahmen alle 290 Befehle vorzustellen. Deswegen habe ich nur eine — sicherlich etwas willkürliche — Auswahl der wichtigsten dort aufgeführt. Allen Macro-Basic-Befehlen ist ein »&« voranzustellen.

Sind Sie ein Verfechter der strukturierten Programmierung? Dann verfügen Sie über Befehle wie IF... THEN... ELSE... ENDIF, aber auch SELECT.. CASE.. OTHER.. ENDSSEL, DO..DO WHILE...DO UNTIL... mit END DO und EXIT. Label können gesetzt und mit GOSUB oder GOTO angesprungen werden. Echte Unterprogramme mit eigenen Variablen, die per CALL anzusteuern sind, sind möglich. Variable und Arrays können aber auch vom Hauptprogramm an das Unterprogramm übergeben werden. Mittels ' (Hochkomma) können Kommentare ins Programm geschrieben werden. Durch SUBMIT ist es möglich, auf einfache Weise mit Overlay-Technik zu arbeiten, was auch durch den OVERLAY-Befehl für einzelne Programme möglich ist. ON ERROR und RESUME erlauben das Auffangen von Fehlermeldungen, die in Variablen EC%, EL% und ET\$ als Fehlercode, fehlerhafte Basic-Zeile und Fehlertext abfragbar sind. Fast schon selbstverständlich sind Befehle wie AUTO (AUTONUMBER-

Funktion, die auch noch Texte wie zum Beispiel DATA schreiben kann), BLOAD oder BSAVE (ladender oder speichernder Zugriff auf beliebige Speicherbereiche), DELETE (Löschen beliebiger Zeilen), DLINK (Anhängen von Programmen) und DMERGE (Mischen von Programmen nach Zeilennummern!), DUMP-Funktionen, die auch die Auflistung von Arrays erlauben, LIM (zum Begrenzen des Basic-Speichers), LPAGE (zum seitenweisen Vorwärts- oder Rückwärts-Auflisten von Programmen), RENUM (das ist ein echtes RENUMBER), RLOAD (zum Laden und Starten eines Programmes), FIND und SEARCH (durchsucht Programme nach Variablen, Befehlen oder Konstanten und listet nach Speicherposition oder Basic-Zeilenummer auf) und TRACE-Möglichkeiten zur schrittweisen Bearbeitung von Basic-Programmen.

Der Verkehr mit Peripherie wird gefördert durch SPOOLING-Befehle, Anweisungen zur Druckeranpassung, DISK zum Ermitteln von Diskettenparametern, HCOPY zum Ausdrucken des Bildschirminhaltes. SCR LOAD oder SAVE erlaubt das Einladen oder Abspeichern von Bildschirmen. Mittels RLIST und SLIST lassen sich relative oder sequentielle Dateien auf dem Bildschirm zeigen. TURBO beschleunigt Diskettenoperationen.

Grafiken können — ab Mai angekündigt, auch mit Hochauflösung — durch Pixel oder Balken erstellt werden. FLASH-Funktionen lassen Bildschirmteile blinken, SOUND und BEEP sorgen für Lautuntermalungen. Eine Reihe von Befehlen greifen direkt auf den Bildschirm zu und erlauben beispielsweise das Füllen von Fenstern mit Zeichen.

Mathematisch Interessierte haben eine Vielzahl von neuen Möglichkeiten zur Verfügung, angefangen vom ACOS über Rechnungen mit mehrfach genauer Zahlendarstellung (bis zu 100 Ziffern!), Umrechnungen innerhalb der Zahlensysteme dezimal/sedezimal/binär, Winkelmaß-Umrechnungen etc. bis zur Berechnung des Gregorianischen Datums aus dem Julianischen und umgekehrt. Strings können wie mathematische Ausdrücke behandelt oder sogar als Basic-Anweisungen interpretiert werden.

Und beim Stichwort Strings sind wir bei der eigentlichen Stärke von Macro-Basic angelangt. Ich habe nie gewußt, was man alles mit Strings anstellen kann, bis ich Macro-Basic kennenlernte! Sie werden kaum eine String-Operation

Auswahl von Macro-Basic Befehlen

Befehl	Kurzbeschreibung
AUTO	Automatische Zeilennummerierung mit Textvorgabe-Option
BIN	Wandelt Binärstring in Dezimalzahl um
BIN\$	Wandelt Ganzzahl in 8- oder 16-stelligen Binärstring um
BLOAD	Datenblock ab bestimmter Adresse in Speicher laden
BSAVE	Speicherbereich abspeichern
BSPACE	Bildschirmfenster mit Zeichen füllen
CALL	Aufruf echter Unterprogramme
CEIL	Ermittelt den Nachkommawert
CHANGE	Durchsucht Speicher nach Stringvariable, teilt Position mit und ersetzt String
CONV	Umsetzen von Druckersteuerzeichen etc. in einem String
DEEK	Lesen von 2 Byte, die in 16-Bit-Wert umgesetzt werden
DEG	Umrechnung Altgrad in Bogenmaß
DELDIM\$	Löschen eines String-Arrays
DEZ	Umwandeln von Sedezimalzahlen in Dezimalzahlen
DLIST	Anzeige best. Records auf dem Bildschirm
DMERGE	Mischen von Basic-Programmen nach Zeilennummern
DOKE	Speichern von 16-Bit-Werten als 2 Byte
EDIT	Editieren bestimmter Bildschirmzeilen
EVAL	Ermittelt Zahlenwert aus Arithmetikstring
FAK	Berechnen der Fakultät
FIND	Durchsucht Speicher nach String und teilt Position mit
FLASH	Läßt Bildschirmfenster blinken
FREDIM	Hebt eine Dimensionierung auf
GET	Liest Zeichen von Tastatur und überträgt nur erlaubte Zeichen
INDEX	Durchsucht Stringvariable nach einer Stringvariablen und teilt Position mit
INFORM	Erlaubt formatierte Dateneingabe
INMSG	Überträgt Speicherbereich in Eingabefelder einer Bildschirm-Maske
INPUT #	Lädt maximal 65000 Zeichen von Diskette in angegebenen Speicherbereich
INS\$	Einfügen eines Strings in eine Stringvariable
KEYS\$	Durchsucht sequentielle Datei nach Suchbegriff
LIKE	Kopiert Speicherbereich in Stringvariable
LINE INPUT	Überträgt alle Zeichen (auch ;, etc.) in Stringvariable
LIST\$	Überträgt Basic-Zeile entschlüsselt in String
MASK	Erlaubt die Erstellung von Bildschirmmasken
MAX	Ermittelt Maximum aus numerischen Variablen
MOB DEF	Erlaubt die temporäre Anwendung weiterer Erweiterungen von Diskette
MIN	Ermittelt Minimum aus numerischen Variablen
MOVE	Kopiert Speicherbereich an vorgegebene Adresse
NONSTOP	Blockiert STOP-Taste
NSORT	Sortiert Gleitkomma-Array in aufsteigender Folge
PACK	Packt Gleitkommavariablen in 5-Byte-String
PLIST	Zeigt Basic-Programm von Diskette auf Bildschirm
QUIT	Liest nur erlaubte Zeichen von Tastatur
REP	Schaltet Repeat-Funktion ein
REPLC\$	Durchsucht String A\$ nach String B\$ und ersetzt B\$ durch C\$
RESTORE	Setzt DATA-Zeiger auf bestimmte Zeile oder Label
ROLLDE	Löscht Bildschirmzeile. Überträgt die Zeile in String und füllt sie dann mit einem angegebenen Stringinhalt
ROTL	Rollt Stringinhalt nach Bedingungen nach links
ROUND	Rundet Zahl auf angegebene Nachkommastellen
RUN\$	Ausführen von Basic-Befehlen in einem String
SBC	Subtrahiert Inhalt zweier Stringvariablen
SCR DO	Rollt Bildschirm nach unten
SET%	Setzt Bitposition eines Strings auf bestimmten Wert
SIM	Simuliert Eingabe einer Stringvariablen über Tastatur
SORT	Sortiert String-Array in aufsteigender Folge
STATE	Setzt Inhalt einer String-Variablen in ausführbare Basic-Befehle um und plaziert sie in angegebene Basic-Zeile während der Ausführung
SWAP	Vertauschen zweier Zahlenvariable
TEST%	Ermittelt Bit-Wert bestimmter Position eines Strings
TFRAME	Erzeugt Rahmen um Textstring
TIME	Initialisiert, stellt und zeigt Uhr an bestimmter Speicheradresse
TRANS	Formatiertes Umsetzen eines Strings nach Format und Positionsvariable
TRON	Aktivieren der TRACE-Funktion
USING	Ausgabe von Zahlen oder Strings nach Formatvariablen
WAIT STOP	Hält Programm an, bis STOP-Taste ein zweites Mal gedrückt wird
XRT	Ermittelt X-te Wurzel aus Y
ZADR	Ermittelt Adresse des 1.Byte einer bestimmten oder der gerade ausgeführten Basic-Zeile.
	Erlaubt Kommentar auch zwischen Befehlen. Vor jeden Befehl & setzen.

vermissen. Mir ist jedenfalls keine mehr eingefallen. Ob es sich um Suchen von ganzen Strings oder Teilen davon, Tauschen oder Verändern, Ein- oder Ausgabe, Positionieren, Prüfen, Umwandeln, Formatieren, Löschen... dreht, alles ist vorhanden.

Stringverarbeitung spielt beim Verwalten von Daten eine Rolle. Neben einer ganzen Reihe von neuen Befehlen, die den Datenzugriff auf Diskette oder im Speicher betreffen, gibt es sechs Module mit 30 Befehlen, die einen Dateityp schaffen, der ISAM/VSAM genannt wird. ISAM kennt vielleicht manch einer als IBM-Begriff: Indexed Sequential Access Method (also Index-sequentieller Dateizugriff). VSAM steht für Virtual Storage Access Method, also virtuelle Dateizugriffsmethode. Mittels ISAM/VSAM kann auf dreierlei Wegen Dateiverarbeitung geschehen: Sequentielle Verarbeitung in der Eingabereihenfolge, sequentielle Verarbeitung in Schlüsselfolgen und die gezielte Verarbeitung über Schlüssel. Datensätze dürfen variable Länge haben, die Datenspeicherung erfolgt blocküber-

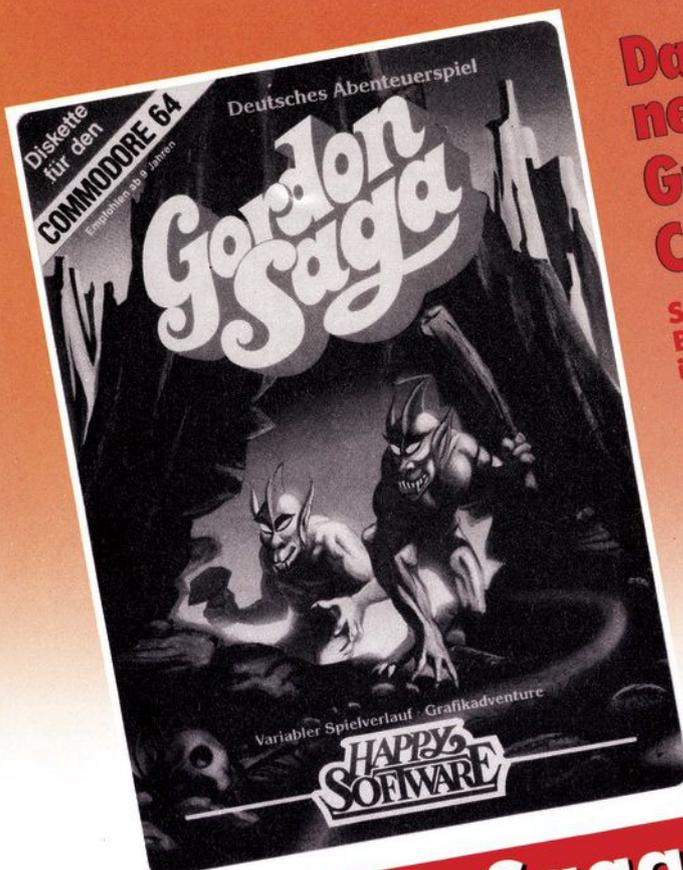
spannend, mehrere Schlüssel dürfen pro Datensatz verwendet werden, aber auch mehrere Datensätze pro Schlüssel sind möglich, wobei übrigens der Schlüssel nicht Bestandteil des Datensatzes sein muß. Bis zu fünf Diskettenstationen der unterschiedlichen CBM-Formate können parallel verwaltet werden. Außer den Befehlen zum Einrichten, Löschen und Sichern solcher Dateien stehen Befehle zum Verarbeiten von Diskettenfehlern, zum Update, zum Suchen von Datensätzen, zum Überschreiben, zum Kopieren, zum Reorganisieren der Datei beziehungsweise der Einträge zur Verfügung.

Sicherlich werden Sie in dieser Beschreibung Befehle vermissen. Das liegt aber nicht daran, daß Macro-Basic sie nicht zur Verfügung stellen würde, sondern daran, daß es unmöglich ist, hier alle aufzuführen. Kritik muß das Handbuch finden. Nicht, daß es vor Rechtschreibfehlern nur so strotzt stört, sondern, daß es offensichtlich für die großen Brüder des Commodore 64 geschrieben wurde, denn alle den Bildschirm betreffenden Operatio-

nen beziehen sich auf einen Bildschirmstart bei 32768. Einige Demoprogramme enthalten schlichtweg Fehler. Der Gebrauch der Bezeichnung BCD (binary coded decimal) als Abkürzung für Bildschirmcode verwirrt auch einigermaßen. Der Autor versichert aber, das Handbuch sei inzwischen überarbeitet worden und die neue Version enthalte diese Fehler nicht mehr. Der Preis hält sich im üblichen Rahmen: 298 Mark zahlt man für die Normalausführung (inklusive Dongle), dazu noch 34 Mark für das Handbuch. Als Zusatzpackung erhält man für 148 Mark die ISAM/VSAM Dateipackung. Nach Auskunft des Autors sind der Grafizusatz und weitere geplante Ergänzungen darin schon enthalten.

Fazit: Durch Macro-Basic wird der Commodore 64 zu einem derart vielseitigen Computer, daß dieses Software-Paket für ernsthafte Anwender guten Gewissens zu empfehlen ist. (Heimo Ponnath/gk)

Info: S&A Software, Bernd, Lambertusstr. 32, 5210 Troisdorf-Muellekoven



Das neueste deutsche Grafik-Adventure für Ihren Commodore 64

Suchen Sie die Pforte zu einer anderen Welt!
Beweisen Sie Ihren Spürsinn, denn der richtige Weg ist schwer zu finden, und überall lauern Gefahren!

- hochauflösende Grafik
- ausführliche Spielanweisungen
- riesiger Befehlsvorrat
- Eingabe von ganzen Sätzen möglich
- variabler Spielablauf

Markt & Technik
Verlag Aktiengesellschaft
Buchverlag

Hans-Pinsel-Straße 2, 8013 Haar bei München, ☎ (089) 4613-220
Schweiz: Markt & Technik-Vertriebs AG, Alpenstraße 14, CH-6300 Zug, ☎ 042/223155
Österreich: Rudolf-Lechner & Sohn, Heizerwerkstraße 10, A-1232 Wien, ☎ 0222/677526

Gordon Saga

Best.-Nr. MD 240 A

DM 39,-* (Sfr. 35,50 / öS 351,-)

* Inkl. MwSt. Unverbindliche Preisempfehlung.

Eine neue Dimension der Abenteuerspiele:
Kein Spiel gleicht dem anderen — Sie geraten in
Situationen, in denen Sie Ihre Spieltaktik völlig
ändern müssen.
Überzeugen Sie sich selbst!

Happy Software gibt's beim Buchhändler, bei Horten, Quelle und im
Computershop. Bestellkarten bitte an Ihren Buchhändler oder an eine
unserer Depotbuchhandlungen.
Adressenverzeichnis am Ende des Heftes!