

ISM 64 —  
ohne Fleiß kein Preis



Das Auswahlmü von ISM 64.

ISM stellt eine Ausnahme unter den Datenverwaltungsprogrammen dar. Es unterscheidet sich in einigen wesentlichen Punkten von Superbase 64, Multidata oder Datamat.

Im Gegensatz zu den anderen Dateiprogrammen wie Superbase 64 oder Multidata ist ISM 64 kein fertiges Anwendungsprogramm. Es handelt sich vielmehr um eine Systemerweiterung, die, erst eingearbeitet in ein eigenes Programm, richtig nutzbar ist.

Diese Konzeption hat Vor- und Nachteile: Zum einen bietet sie die Möglichkeit der individuellen Problemlösung, zum anderen kann aber insbesondere der Anfänger vor so große programmtechnische Probleme gestellt werden, daß er ISM 64 am Ende gar nicht nutzen kann.

Der hier vorliegende Testbericht geht daher nicht nur auf die Möglichkeiten ein, die ISM 64 bietet, sondern gibt auch Hinweise zur Programmierung.

Grundlage dieses Berichts ist die mehrmonatige Arbeit mit ISM 64 bei der Erstellung eines umfassenden Anwenderprogramms.

## Allgemeines

ISM 64 selbst ist ein Maschinenprogramm von 15 KByte Länge und belegt normalerweise den Basic-Speicher ab dez. 27000. Für das An-

wenderprogramm stehen dann noch etwa 24 KByte zur Verfügung. Auch wenn man noch einen gewissen Bedarf an Speicherplatz für Daten abzieht, sollte dies ausreichend sein. Das Maschinensprache-RAM (dez. 49152 bis 53247) bleibt frei; hier lokalisierte Programme (zum Beispiel Interfaces oder Programme zur Unterstützung der Floppy) sollten normal verwendbar sein.

Nach dem Laden von ISM 64 stehen eine Reihe spezieller Befehle zur Einrichtung und Bearbeitung von Dateien zur Verfügung. Diese Befehle sind sehr mächtig und nicht mit jenen zu vergleichen, die man von den üblichen Interpretererweiterungen wie zum Beispiel Simons Basic kennt. Das Schreiben, Ändern und Suchen von Datensätzen wird dann sehr bequem und effektiv.

## Was bietet ISM 64?

ISM 64 beeindruckt zunächst durch die Möglichkeit, praktisch unbegrenzt lange Datensätze verwalten zu können. Ein einzelner Satz darf bis zu 31875 Zeichen lang sein, er kann in bis zu 125 Felder unterteilt werden, wobei jedes Feld bis zu 255 Zeichen enthalten darf. Die Felder können von fester oder variabler

Länge sein, bis zu 40 Felder lassen sich als Schlüsselfelder definieren. Angesichts dieser Zahlen fragt man sich natürlich, ob dies überhaupt sinnvoll ist. Welcher Privatanwender hat schon Datensätze mit Zehntausenden von Zeichen oder Dutzenden von Schlüsseln? Hier muß man wissen, daß der ISM nicht speziell für den C 64 entwickelt wurde. Es handelt sich vielmehr um die angepaßte Version eines Programms, das in erster Linie zum gewerblichen Einsatz auf den CBM-Computern der 8000er-Serie gedacht ist. Aber davon abgesehen, auch bei der privaten Anwendung sind Satz-längen von um die 1000 Zeichen durchaus nicht unrealistisch. Wenn man zum Beispiel Literaturdaten verwaltet und jedem Eintrag eine kurze Zusammenfassung der Literaturstelle hinzufügt, sind 500 bis 1000 Zeichen schnell erreicht.

Entsprechend weitgehend sind auch die Möglichkeiten, die dem Anwender bei der Einrichtung der Datei zur Verfügung stehen. Wie bereits erwähnt, kann ein Feld des Satzes als variabel definiert werden, das heißt die Länge des Feldes kann von Satz zu Satz unterschiedlich sein. Hierbei wird auf der Floppy aber immer nur der Platz belegt, der der tatsächlichen Länge des Satzes entspricht. Dies ist keinesfalls selbstverständlich, andere Systeme sind da nicht so flexibel. Dem Anwender bleibt auch überlassen, welche Felder er wirklich mit Daten füllen will. Auf diese Weise ist es möglich, ohne Speicherplatzverschwendung den Datensatz großzügig zu definieren und erst nach und nach alle Felder auszufüllen. Ein Datensatz kann so im Laufe der Zeit dynamisch wachsen.

Bis zu 40 Felder eines Satzes können als Schlüssel definiert werden.



# ISM 64 — ohne Fleiß kein Preis

Von diesen Feldern werden bis zu 48 Zeichen linksbündig in die Schlüsseldatei eingetragen. Bei der Suche nach einem Satz wird dann zunächst auf diese Datei zugegriffen (siehe unten). Bei ISM 64 können die Schlüssel als ein- oder mehrdeutig definiert werden. Ein mehrdeutiger Schlüssel (zum Beispiel Nachname) darf mehrfach vorkommen, durch sequentielle Suche (siehe unten) können dann alle Sätze mit diesem Schlüssel gefunden werden. Hat man dagegen einen Schlüssel als eindeutig definiert (zum Beispiel sinnvoll bei einer laufenden Nummer), dann weist der Dateiverwalter diesen Satz mit einer entsprechenden Fehlermeldung zurück. Der Satz geht hierbei natürlich nicht verloren, sondern kann vom Anwenderprogramm zur Korrektur weiterverarbeitet werden. Überhaupt ist die Fehlerbehandlung bei ISM 64 sehr gelungen. Systemabstürze lassen sich bei geschickter Programmierung weitgehend ausschließen.

An dieser Stelle erscheint es sinnvoll, etwas ausführlicher auf die Theorie der Dateiverwaltung einzugehen. ISM 64 verwendet hier keine spezielle Lösung, sondern verwaltet die Sätze nach dem Prinzip der relativen Datei, wobei Schlüssel und Sätze in unterschiedlichen Dateien abgelegt werden. Relativdateien werden vom Basic V 2.0 ja leider nicht unterstützt, und sind daher sonst nur durch Kunstgriffe zugänglich.

## Das Prinzip der relativen Datei

Wie ist nun eine relative Datei aufgebaut? Das Prinzip ist so einfach wie wirkungsvoll: Der Speicherplatz auf der Floppy wird zunächst in kleine Häppchen, die sogenannten Records, zerlegt. Die Länge der einzelnen Records ist bei ISM 64 frei wählbar. Dies ist wichtig, wenn man den bei der Floppy VC 1541 ja nicht gerade üppig bemessenen Speicherplatz optimal nutzen will. Die einzelnen Records werden intern durchnummeriert. Ein Datensatz wird nun in diese Records geschrieben, ein Satz von beispielsweise 300 Byte Länge belegt dann bei einer gewählten Recordlänge von 100 Byte gerade drei Records. Dem Satz wird dann vom

Programm intern zur eindeutigen Unterscheidung von anderen Sätzen die Nummer zugewiesen, die der Nummer des ersten belegten Records entspricht. Gleichzeitig werden die Schlüssel des Satzes — dies sind die Einträge, über die man später auf den Satz zugreifen möchte — in eine eigene Datei, die Schlüsseldatei, in einen sogenannten Schlüsselbaum eingetragen. Die Schlüssel werden von vornherein alphabetisch beziehungsweise numerisch sortiert. Bei jedem Schlüssel findet sich dann als Verweis auf den zugehörigen Satz die Nummer des Records, mit dem der Satz beginnt. Diese Nummer ist naturgemäß umso höher, je weiter der Satz relativ vom Anfang der Datei entfernt ist, je mehr Records zum Zeitpunkt des Eintrags also schon belegt waren. So erklärt sich die Bezeichnung Relativdatei. Bei der Suche nach einem bestimmten Satz sucht der Dateiverwalter im Schlüsselbaum nach dem angegebenen Suchbegriff. Ist dieser vorhanden, dann merkt sich der Dateiverwalter die beim Schlüssel eingetragene Recordnummer. Diese gibt die relative Lage des entsprechenden Satzes zum Dateian-

fang an, und der Dateiverwalter kann auf diesen Satz zugreifen. Das Prinzip der Datenspeicherung in Relativdateien gilt momentan als optimal bei der Arbeit mit Kleincomputern.

### Datensätze eintragen

Jeder Datensatz kann und sollte unmittelbar nach der Erstellung auf die Floppy geschrieben werden, um nicht das Risiko eines etwaigen Datenverlustes, beispielsweise infolge eines Stromausfalls, einzugehen. Die Zeit, die zum Eintrag benötigt wird, steigt mit der Anzahl der Schlüssel stark an. Allerdings werden die Schlüssel beim Eintrag bereits alphabetisch sortiert, ein Arbeitsgang, der bei der Ausgabe von sortierten Listen bei den meisten anderen Programmen später bewältigt werden muß. Vierzig Schlüssel sind bei der bekannten »Schnelligkeit« der VC 1541 sicher kaum realisierbar. ISM 64 bietet hier allerdings die Möglichkeit der Stapelverarbeitung, das heißt man trägt zunächst nur den Satz ein. Dies geht sehr schnell, der zeitaufwenige Eintrag

### Ausdruck von Datensätzen

```

1000 dim dn$(1):dn$(1)="Dateiname"
1010 !iopen,10,dn$(1),5:rem Datei öffnen
1020 !iread,m,1,df$(1),rn,fe:rem Datensatz lesen
1030 !iclose:rem Datei schließen
1040 open4,4:rem Druckerkanal öffnen
1050 for i=1 to n
1060 print#4,df$(i)
1070 next:rem Felder des Satzes drucken
1080 close4:rem Druckerkanal schließen
1090 open15,8,15:rem Befehlskanal der Floppy öffnen
1100 print#15,"i":rem Floppy initialisieren
1110 close15:rem Befehlskanal der Floppy schließen
1120 return:rem z.B. nächsten Satz lesen

```

Nach jedem Ausdruck muß die Floppy initialisiert werden. Diese kleine Routine gibt auch einen Eindruck, wie die ISM-Befehle in ein Basicprogramm eingebaut werden.

der Schlüssel in den sortierten Baum erfolgt erst später. Dies kann durch ein kleines Programm geschehen, ohne daß man dabei sein muß. Die Realisierung dieser Möglichkeit, Satz und Schlüssel getrennt voneinander einzutragen, ist allerdings eine Sache des Anwenderprogramms. Auf diese Weise ist es dann möglich, zu einem Satz mehr

der Schlüssel in den sortierten Baum erfolgt erst später. Dies kann durch ein kleines Programm geschehen, ohne daß man dabei sein muß. Die Realisierung dieser Möglichkeit, Satz und Schlüssel getrennt voneinander einzutragen, ist allerdings eine Sache des Anwenderprogramms. Auf diese Weise ist es dann möglich, zu einem Satz mehr



Schlüssel einzutragen, als bei der Definition der Datei vorgesehen war.

#### Datensätze ändern

Das Ändern eines Satzes ist möglich, indem man den Satz zunächst liest, die Änderungen vornimmt, und den Satz dann mit einem speziellen Schreibbefehl wieder einträgt. Auch hier ist der Komfort von der Qualität des Anwenderprogramms abhängig. Zusätzlich ist es möglich, einen Satz komplett mit allen Schlüsseln, nur den Satz, nur die Schlüssel, oder aber nur einzelne Schlüssel des Satzes zu löschen (und gegebenenfalls durch neue zu ersetzen).

#### Datensätze suchen

Sicher am wichtigsten bei einem Dateiverwaltungssystem sind die Möglichkeiten, die man zur Suche nach Datensätzen hat. ISM 64 bietet hier Befehle zur Suche mit voller Übereinstimmung eines Schlüssels mit einem Suchbegriff, sowie zur Suche mit verkürztem Schlüssel (zum Beispiel Suche nach Sätzen, bei denen ein Schlüssel mit A beginnt). Zusätzlich kann über die Recordnummer direkt auf einen bestimmten Satz zugegriffen werden. Eine Suche mit Joker (zum Beispiel nach einem »M??er« ist nicht von vornherein vorgesehen, kann aber im Anwenderprogramm leicht realisiert werden. Außerdem erlaubt ISM die sequentielle Suche, das heißt, man liest den ersten Satz, auf den ein bestimmtes Kriterium zutrifft, und kann dann alle weiteren Sätze lesen. Diese Option erlaubt auch das schnelle Durchblättern der Datei. Oft findet man so einen interessierenden Satz am schnellsten. Das Zeitverhalten bei der Suche ist noch befriedigend, die Grenzen der Floppy VC 1541 werden hier mehr als offenbar. Ein kleiner Nachteil ist sicherlich, daß vom Programm Groß- und Kleinschreibung unterschieden werden. Wenn man nach einem Mayer mit mayer sucht, dann wird der entsprechende Satz nicht gefunden.

#### Drucken von Datensätzen

Das Drucken von Datensätzen wird von ISM 64 nicht unterstützt, die Druckroutine ist allein Sache des Anwenderprogramms. Dies sollten insbesondere alle C 64-Anwender, die keinen Commodore-Drucker

haben, nicht bedauern. Oft muß man ja feststellen, daß bei fertigen Programmlösungen andere als Commodore-Drucker, wenn überhaupt, nur unvollständig unterstützt werden. Die Möglichkeit einer ganz individuellen Druckroutine ist in solchen Fällen eher ein Vorteil. Allerdings muß man bei der Programmierung einiges unbedingt beachten (siehe unten).

## Die Programmierung von ISM 64

Wie bereits oft erwähnt, ist ISM nur zusammen mit einem Anwenderprogramm voll nutzbar. Wie einfach oder schwierig ist nun die Erstellung eines solchen Programms? Ein möglicher Erfolg beziehungsweise Mißerfolg hängt erstens davon ab, wie gut man das Basic V 2.0 beherrscht und zweitens, wie hoch die eigene Frustrationsschwelle liegt. Wenn man gut programmieren kann, und nicht so schnell aufgibt, wird man sicherlich zu einer befriedigenden Lösung kommen. Das mitgelieferte Handbuch ist umfassend, auch im Detail. Allerdings ist es nicht einfach zu lesen. Viele Kapitel wird auch der Fortgeschrittene mehrfach studieren müssen, vieles wird auch erst im Laufe der Arbeit mit dem Programm klar werden. Hilfreich ist es, daß ein fertiges, kommentiertes Beispielprogramm auf der Diskette mitgeliefert wird. Dieses Programm kann normal gelistet werden, und ist durchaus nützlich zur Inspiration bei der Programmerstellung. Leider enthält es mindestens einen programmtechnischen Fehler. Ein anderes, mitgeliefertes Programm erlaubt die Anlage beliebiger Dateien. Diese Routine ist sehr brauchbar, so daß keine Notwendigkeit besteht, eine eigene Anlage-Routine zu schreiben.

Wie bereits eingangs ausgeführt, liegt eine der Hauptstärken des ISM 64 darin, praktisch unbegrenzt lange Datensätze verwalten zu können. Unter anderem darf ein einzelnes Feld bis zu 255 Zeichen lang sein. Leider ist im Basic 2.0 eine adäquate Eingabemöglichkeit nicht vorgesehen. Das INPUT-Statement erlaubt nur Eingaben bis zu real 78 Zeichen, wobei keine Kommata vorkommen

dürfen. Einziger Ansatzpunkt ist das GET-Statement.

Ein besonderes Kapitel ist der Ausdruck von Datensätzen. Dies ist wie gesagt allein Sache des Anwenderprogramms. In einem solchen Programm ist nun folgendes zu berücksichtigen: Nach jedem Ausdruck muß die Floppy initialisiert werden. Wird die Floppy nach einem Ausdruck nicht initialisiert, dann führt der nächste Dateizugriff zu einem Systemabsturz. Da dieser Effekt auftrat:

1. Bei einem über ein Software-Interface am User-Port angeschlossenen Epson-Drucker
2. bei einem normal angeschlossenen VC 1526 und
3. auch wenn überhaupt gar kein Drucker angeschlossen war, scheint es sich um ein grundsätzliches Problem zu handeln, dessen Ursache wohl in der Unzulänglichkeit des seriellen Busses liegt. Wie programmtechnisch beim Ausdruck verfahren werden kann, ist aus dem abgedruckten Listing ersichtlich. Leider wird in der ISM-Anleitung nichts von dieser Problematik erwähnt, obwohl dies meiner Ansicht nach unbedingt notwendig wäre. Denn was nutzt einem letztendlich ein Dateiverwaltungsprogramm, wenn man nicht drucken kann.

## Fazit

ISM 64 ist ein sehr leistungsfähiges Programm mit professioneller Konzeption. Die Notwendigkeit, ein eigenes Anwenderprogramm schreiben zu müssen, kann ein Vorteil oder Nachteil sein, je nach Art des geplanten Einsatzes und den Kenntnissen des Anwenders. Der Preis von zirka 140 Mark ist sicherlich günstig zu nennen.

Am Ende sei noch erwähnt, daß für alle, die ISM 64 einsetzen möchten, — ohne ein eigenes Anwenderprogramm zu schreiben — die Möglichkeit besteht, ein fertiges Programm mit Anleitung vom Autor dieses Berichtes zu beziehen. Der Autor ist auch gerne bereit, anderen ISM-Anwendern bei aufgetretenen Problemen im Rahmen seiner Möglichkeiten weiterzuhelfen. Schreiben sie an H. J. Schlicht, Kirschgartr. 7, 6900 Heidelberg 1

(H. J. Schlicht/rg)