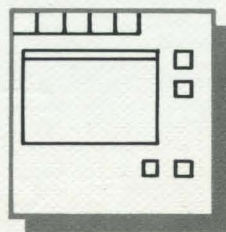
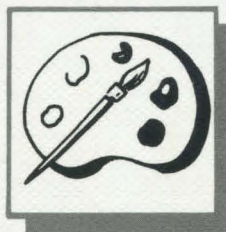


Kerkloh
Tornsdorf

für
Commodore
C64/128

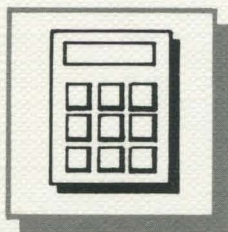


**Das
große**

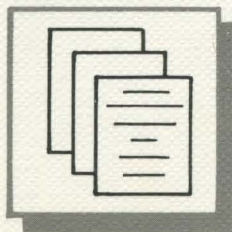


GEOGRAPH

inkl.
GEOS 1.2
GEOS 1.3
deutsch



Buch



DATA BECKER

Kerkloh
Tornsdorf

Das große GEOS-Buch

DATA BECKER

Karl
Taschen

Das große GEO-Buch

3. vollständig überarbeitete und erweiterte Auflage 1988

ISBN 3-89011-208-0

Copyright © 1986

DATA BECKER GmbH
Merowingerstr. 30
4000 Düsseldorf

Text verarbeitet mit Word 4.0, Microsoft
Ausgedruckt mit Hewlett Packard LaserJet II
Druck und Verarbeitung Graf und Pflügge, Düsseldorf

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der DATA BECKER GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

KAVRICH

Dieses Buch steckt zu voller Tiefe, Information und Hilfe für die Arbeit mit GECOS, daß ein Vorwort Sie eigentlich nur davon abhalten kann, den Rechner einzuschalten, das Buch aufzuschlagen und an das anzuprogrammieren und zu schauen, was wir für Sie herausgefunden und programmiert haben. Trotzdem möchten wir zumindest das an Sie weitergeben, was im Buch seinen Platz gefunden hat. Dieses Buch bezieht sich gleichzeitig auf den ersten in Deutschland verteilten Versionen GECOS V1.2 und V1.3 an. Die erste wird jedoch GECOS V1.2 konstruiert, die zweite wird gekauft werden. Die meisten Abbildungen in diesem Buch zeigen die Version 1.2, weil auf Grund der englischen Begriffe und Bedienführung unserer Meinung nach hier die meisten Abbildungen nötig sind. Im Anhang dieses Buches finden Sie eine Gegenüberstellung der wichtigsten Begriffe und Namen beider Versionen.

Besonders noch sind wir darauf, Ihnen für GECOS V1.2 und die folgende Version V1.3 eine Möglichkeit geschaffen zu haben, Selbstverständliches von der Originalität her zu machen. Deshalb können Sie nicht nur die Originalität wieder verwenden, sondern auch wichtige Änderungen an der Originalität vornehmen. Viele Änderungen werden ebenfalls mit dieser Originalität verbunden. Damit Sie einen Eindruck bekommen, mit welcher einfachen Mitteln Berkeley Software seine Abstraktion zu realisieren kann. Leider wir Ihnen im ersten Kapitel des Buches

Wichtiger Hinweis:

Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technischen Angaben und Programme in diesem Buch wurden von dem Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

Vorwort

Dieses Buch steckt so voller Tips, Informationen und Hilfen für die Arbeit mit GEOS, daß ein Vorwort Sie eigentlich nur davon abhalten kann, den Rechner einzuschalten, das Buch aufzuschlagen und all das auszuprobieren und zu nutzen, was wir für Sie herausgefunden und programmiert haben. Trotzdem möchten wir zumindest das an Sie weitergeben, was im Buch keinen Platz gefunden hat. Dieses Buch beschäftigt sich gleichzeitig mit den beiden in Deutschland verbreitetsten Versionen GEOS V1.2 und V1.3 deutsch. Die erste wird jedem C64 kostenlos beigelegt, die zweite muß gekauft werden. Die meisten Abbildungen in diesem Buch zeigen die Version 1.2, weil auf Grund der englischen Begriffe und Bedienung unserer Meinung nach hier am ehesten Abbildungen nötig sind. Im Anhang dieses Buches finden Sie eine Gegenüberstellung der wichtigsten Begriffe und Namen beider Versionen.

Besonders stolz sind wir darauf, Ihnen für GEOS V1.2 und die deutsche Version V1.3 eine Möglichkeit geschaffen zu haben, Sicherheitskopien von der Originaldiskette bootfähig zu machen. Dadurch können Sie nicht nur die Originale sicher verwahren, sondern auch beliebige Änderungen an der Startdiskette vornehmen. Viele Änderungen werden nämlich nur auf dieser Diskette wirksam. Damit Sie einen Eindruck bekommen, mit welchen wirksamen Mitteln Berkley Softworks solche Maßnahmen zu verhindern sucht, haben wir Ihnen im ersten Kapitel die Tricks verraten, die GEOS V1.2 und GEOS V1.3 so schwer kopierbar machen.

Damit Ihre Arbeit mit GEOS noch leichter und wirkungsvoller wird, enthält dieses Buch eine Fülle von Programmen für die tägliche Arbeit. Die wichtigsten sind vollständig in die GEOS-Umgebung eingebettet und könnten somit direkt von Berkley Softworks stammen. Bis auf das Programm zum Erstellen von bootfähigen Sicherheitskopien (nur V1.3 deutsch) funktionieren alle Programme anstandslos unter beiden GEOS-Versionen.

Die Autoren

Münster, den 27.4.1988

Inhaltsverzeichnis

1.	Die Besonderheiten von GEOS	13
1.1	Die Komponenten von GEOS	13
1.2	Die Versionen von GEOS	19
1.3	Bootvorgang und Kopierschutz	21
1.4	Seriennummer und Installationsvorgang	28
2.	Tips und Tricks zur Benutzeroberfläche	31
2.1	Laden	31
2.2	DESK TOP	34
2.3	Direktstart	59
2.4	GEOWRITE	61
3.	BASIC-Tools	69
3.1	Notizen	69
3.1.1	Das Programm "Notizendruck"	71
3.1.2	Dokumentation des Programms "Notizendruck"	73
3.2	GEOWRITE-Texte drucken	79
3.2.1	GEODRUCK	80
3.2.2	Das Programm "GEODRUCK"	81
3.2.3	Dokumentation des Programms "GEODRUCK"	84
3.3	Der Text-Konverter	89
3.3.1	Das Listing des Konverters	90
3.3.2	Die Bedienung des Konverters	94
3.3.3	Dokumentation des Konverters	98
3.4	Der Filemaster	101
3.4.1	Das Programm "Filemaster"	103
3.4.2	Erläuterung des Programms	111
3.4.3	Das Menü des Filemasters	124
3.4.4	Bedienung des Filemasters	130

4.	GEOS	135
4.1	Sicherheitskopien	135
4.2	Bleibende Änderungen	141
4.2.1	Die Problematik von Änderungen an GEOS	142
4.2.2	Der Modifikator	144
4.2.3	Bedienung des Modifikators	147
4.2.4	Erläuterung zur Funktionsweise	150
4.2.5	Anwendungen	159
4.3	Über die GEOS-Druckertreiber	185
4.3.1	Änderungen an den Treibern	189
4.3.2	Druck	193
5.	Neue GEOS-Programme	197
5.1	Die Uhrzeit ständig im Blick	198
5.1.1	Bedienung des Programms	210
5.2	Hardcops erstellen	213
5.2.1	Listing des Hardcopy-Programms	215
5.2.2	Bedienung des Programms	222
5.2.3	Dokumentation des Programms	224
5.3	Der Font-Editor	225
5.3.1	Bedienung des Font-Editors	227
5.3.2	Bedienungsreferenz	240
5.3.3	Listing des Programms	242
5.3.4	Beispiele mit dem Font-Editor	249
5.4	Der Maschinensprache-Monitor	259
5.4.1	Übersicht über die Befehle	268
5.4.2	Tips zur Arbeit mit dem EDMON	272
5.5	Der Einzelschrittsimulator	276
5.5.1	Listing des Steppers	279
5.5.2	Funktionsweise des Stepper	290
6.	GEOS intern	293
6.1	Die GEOS-Diskettenstrukturen	294
6.1.1	File-Verwaltung	295
6.1.2	Ein File-Eintrag	296

6.1.3	Der INFO-Sektor	298
6.1.4	Die Border und GEOS-Format V1.0	302
6.1.5	Filetyp	306
6.2	Wie GEOS funktioniert.....	312
6.3	Eigene Programme in GEOS	316
6.3.1	Wie eigene Programme geladen werden	317
6.3.2	Wie man GEOS durch eigene Programme erweitert	318
6.4	Die GEOS-Sprungtabelle	323
6.5	Speicher	390
6.5.1	Die GEOS-Variablen	390
6.5.2	Speicheraufbau	399
7.	Anhang	403
	Stichwortverzeichnis	421

1. Die Besonderheiten von GEOS

Als GEOS in den USA von Berkley Softworks entwickelt wurde, hatten die Entwickler vor, das scheinbar Unmögliche möglich zu machen. Praktisch alle Unzulänglichkeiten des bestehenden Betriebssystems sollten überwunden und zusätzlich fantastische Möglichkeiten bereitgestellt werden. In diesem Kapitel möchten wir Ihnen diese Besonderheiten von GEOS vorstellen und Ihnen einen Einblick davon verschaffen, wie dies möglich wurde und welche genialen Ideen dazu beitrugen.

Wir werden Ihnen in diesem Kapitel auch erklären, wie der Kopierschutz der unterschiedlichen GEOS-Versionen einschließlich GEOS V1.3 deutsch funktioniert. Bitte denken Sie daran, daß eine Weitergabe von Kopien nicht erlaubt ist und gegen das COPYRIGHT verstößt. Mit den hier wiedergegebenen Informationen sind einige vielleicht in der Lage, Kopien von GEOS zu erstellen. Trotzdem sind die Informationen über den Kopierschutz so lehrreich und interessant, daß wir Sie Ihnen nicht vorenthalten wollen.

1.1 Die Komponenten von GEOS

GEOS enthält gegenüber dem eingebauten Betriebssystem und im Vergleich zu anderen 8-Bit-Betriebssystemen eine ganze Reihe von Fähigkeiten, die ihresgleichen suchen:

- grafische Benutzeroberfläche mit Pull-Down-Menüs, Icons und Windows
- schnelle Diskettenroutinen
- Multitasking
- jederzeit ladbare Accessories
- proportionale Zeichensätze in verschiedensten Größen
- neues Dateiformat, das die Vorteile sequentieller und relativer Dateien verbindet
- unterstützt sehr viele Drucker

- unterstützt die wichtigsten Eingabegeräte (Maus, Joystick, Lightpen usw.)
- Datenaustausch zwischen verschiedenen Programmen durch Scrap-Konzept
- viele Grafikroutinen
- Routinen zum Packen und Entpacken

Ein vergleichbares Betriebssystem auf dem Atari ST benötigt 192 KByte Speicher. Beim C64 stehen aber maximal 64 KByte zur Verfügung, und davon soll natürlich noch etwas für die Programme übrigbleiben, die unter diesem Betriebssystem arbeiten. Den Entwicklern ist es gelungen, dieses Betriebssystem in einem annehmbaren Teil des Gesamtspeichers unterzubringen. Trotzdem ist der verbleibende Speicher für Anwenderprogramme knapp, doch dieses Manko wird dadurch wieder ausgeglichen, daß unter GEOS Programme ablaufen können, die wesentlich größer sind als der verbleibende Restspeicher. Auch GEOS lädt bestimmte notwendige Teile erst dann nach, wenn sie benötigt werden (Druckertreiber und Teile des DESK TOP).

Den Entwicklern bei Berkley Softworks war natürlich klar, daß es sehr schwer sein würde, im bestehenden C64-Markt ein völlig neues Betriebssystem durchzusetzen, sei es auch noch so leistungsfähig. Deshalb besteht GEOS auch aus erheblich mehr als nur dem Betriebssystem. Praktisch erhält man mit GEOS ein komplettes System zur Arbeit mit seinem C64, von der fertigen Oberfläche zur Arbeit mit der Floppy über Textverarbeitung und Malprogramm bis hin zum Taschenrechner und zur eingebauten Uhr.

GEOS besteht aus folgenden Komponenten:

GEOS

GEOS ist ein sehr kurzes Programm, das den Autostart ermöglicht. Dazu wird es beim Laden mit LOAD "GEOS",8,1 genau in den Stack des C64 geladen und startet sofort das eigentliche Boot-Programm "GEOS BOOT". Schon an der Programmierung des kleinen Programms "GEOS" zeigt sich, wie sorgfältig und

überlegt die Entwickler vorgegangen sind. Wird nämlich "GEOS" in den BASIC-Speicher geladen (LOAD "GEOS",8), so ist zwar kein Autostart möglich, zumindest steht aber sofort die richtige Programmzeile im BASIC-Speicher (LOAD "GEOS",8,1), die den korrekten Lade- und Startvorgang auslöst.

GEOS BOOT

Dieses Programm sorgt für den eigentlichen Ladevorgang des GEOS-Betriebssystems. Es schickt die Schnelladeroutinen (Speeder) zur Floppy, startet sie und lädt dann ca. 6mal schneller als sonst das GEOS KERNAL. Zusätzlich hat es noch einige weitere Aufgaben zu erledigen, über die normalerweise kaum etwas bekannt ist. Zum einen liegt es teilweise verschlüsselt auf Diskette vor, um eine Untersuchung des Programms zu erschweren. Daher muß es die notwendigen Teile des eigenen Programms erst einmal entschlüsseln. Wir werden im Zusammenhang mit dem GEOS-Kopierschutz noch genauer auf diesen Punkt eingehen.

Zum anderen stellt es eine Seriennummer in Form einer Zufallszahl für das GEOS-Betriebssystem bereit. Sollte GEOS noch nicht installiert sein, so trägt ein anderer Programmteil des GEOS-Betriebssystems diese Zufallszahl fest in das KERNAL ein. Die Zufallszahl wird übrigens durch die zufällige Position des Video-Rasterstrahls bestimmt. Das KERNAL wird nämlich in mehreren Teilen geladen, und am Ende des jeweiligen Teilladevorgangs bestimmt die Position des Rasterstrahls einen Teil der Zufallszahl.

In den Speeder-Routinen, die von GEOS BOOT zur Floppy geschickt werden, ist auch schon ein Teil der Kopierschutzabfrage enthalten. Findet diese Abfrage nicht den ordnungsgemäßen Schutz, so merkt GEOS BOOT das und leitet einen RESET des C64 ein. Der GEOS-Bootvorgang ist damit beendet.

GEOS KERNAL

Das GEOS KERNAL stellt das eigentliche Betriebssystem GEOS dar. Es besteht aus drei Teilen, die nacheinander von GEOS

BOOT geladen werden und GEOS initialisieren und starten. Ab GEOS 1.3 ist die Initialisierung so umfangreich geworden, daß diese teilweise in einen extra Programmbereich ausgelagert wurde, der anfangs von GEOS BOOT nach \$5000 geladen wird. Dort wird er dann später vom nachgeladenen DESK TOP überschrieben. Dieser Programmteil ermöglicht beispielsweise das automatische Laden und Starten von Programmen auf der Boot-Diskette, die einen speziellen Filetyp haben. Er besorgt auch die beim ersten Start notwendige Installation der Seriennummer im KERNAL.

DESK TOP

Das DESK TOP stellt die Verbindung zwischen Betriebssystem und Benutzer her. Es verwaltet den Disketteninhalt in Form von Icons, besorgt alle wichtigen Floppy-Operationen in einfacher, einprägsamer Art, die sonst nur über komplizierte Befehle möglich sind, und ermöglicht das Starten von Applications und Accessories. Der größte Teil des DESK TOP wird im Anschluß an das KERNAL geladen. Einige Funktionen bleiben aber auf Diskette und werden erst nachgeladen, wenn sie gebraucht werden. So muß sich beispielsweise das DESK TOP auf der aktuellen Diskette befinden, um den Info-Text zu einer Datei anzuzeigen.

Accessories

GEOS enthält einige nützliche Hilfsmittel, von denen jeweils eins jederzeit von Diskette nachgeladen werden kann. Dabei wird der Teil des Speichers, der von diesem Accessory belegt wird, auf Diskette gerettet und zum Schluß (wenn das Accessory fertig ist) wieder von Diskette geladen.

Applications

Die beiden mitgelieferten Anwendungsprogramme (Applications) GEOPaint und GEOWrite nutzen die neuen Möglichkeiten des GEOS-Betriebssystems voll aus. Da sie den Speicher benötigen, den auch das DESK TOP belegt, können sie nicht zusammen mit

diesem benutzt werden. In diesem Sinne ist auch das DESK TOP eine Application.

Input Devices

GEOS kann nicht nur mit Hilfe eines Joysticks gesteuert werden. Es werden auch die Commodore-Maus, Lichtgriffel und Koala-Pad unterstützt. Da für die Ansteuerung dieser Eingabegeräte unterschiedliche Routinen notwendig sind und Speicherplatz knapp ist, wird bei GEOS das DEVICE-Konzept eingesetzt. Dabei wird von den eigentlichen Betriebssystem-Routinen nicht die direkte Hardware, also das Gerät programmiert, sondern nur eine Art genormte Schnittstelle, der Gerätetreiber. Für jedes Gerät gibt es dann einen speziellen Gerätetreiber, und nur der gerade benötigte muß von Diskette geladen werden. Die anderen verbrauchen keinen Platz im Rechnerspeicher.

Printer

Auch bei der Ansteuerung der vielen unterschiedlichen Drucker nutzt GEOS das DEVICE-Konzept in Form unterschiedlicher Druckertreiber. Dadurch braucht das Betriebssystem (KERNAL) nicht verändert zu werden, wenn neue Drucker unterstützt werden sollen. Es reicht aus, einfach neue Druckertreiber zu erstellen. Der ausgewählte Druckertreiber ist bei GEOS normalerweise nicht geladen, dies geschieht erst direkt vor dem eigentlichen Druckvorgang. Deshalb sollten Sie auch auf den Arbeitsdisketten den Treiber für Ihren Drucker haben, wenn Sie von diesen Disketten aus drucken wollen. Bei GEOS 128 wird der Druckertreiber übrigens beim Booten geladen und kann während der Arbeit mit GEOS auf Grund des größeren Rechnerspeichers im Rechner bleiben.

Fonts

Im Unterschied zum Original-Betriebssystem des C64 und den üblichen Programmen unterstützt GEOS proportionale Zeichensätze. Dabei können die einzelnen Zeichen unterschiedlich breit sein. Da diese Zeichendarstellung von der Hardware des C64

nicht unterstützt wird, wird sie von GEOS softwaremäßig mittels der hochauflösenden Grafik erzeugt. GEOS ist aber nicht auf eine einzige Schrift festgelegt, sondern unterstützt die Verwendung verschiedener Schriften (FONTS) in unterschiedlichen Größen. Auch hier ist aus Platzmangel nur ein FONT in einer Größe (BSW 8 Point) im Speicher vorhanden, weitere FONTS müssen vor der Benutzung erst von Diskette nachgeladen werden. Die unterschiedlichen Schriftarten (Bold, Italic, Underline usw.) sind übrigens nicht fest im FONT vorhanden, sondern werden von den Ausgaberroutinen des Betriebssystems errechnet. So wird beispielsweise "Fettdruck" (Bold) dadurch erreicht, daß dasselbe Zeichen noch einmal um einen Punkt versetzt ausgegeben wird.

Album

GEOS enthält leistungsfähige Routinen zum Datenaustausch zwischen verschiedenen Programmen. Im Augenblick werden dabei Text- und Grafikdaten unterstützt. Aus einem Text oder einer Grafik können Teile ausgeschnitten werden und als Ausschnitt (Scrap) auf der Diskette abgelegt werden (Textscrap, Photoscrap). Dieses Verfahren wurde aus zweierlei Gründen eingerichtet: Zum einen muß jedes Anwendungsprogramm in der Lage sein, Daten zu kopieren, bei einer Textverarbeitung beispielsweise die Blockfunktionen und bei Malprogrammen das Kopieren von Ausschnitten. Da Speicherplatz unter GEOS extrem knapp ist, werden diese Daten nicht im Rechnerspeicher aufbewahrt, sondern auf Diskette. Durch den schnellen Zugriff auf Diskette bleibt der Kopiervorgang trotzdem im zeitlichen Rahmen.

Zum anderen wird die Leistungsstärke von GEOS erhöht, wenn die einzelnen Anwendungsprogramme zusammenarbeiten. Durch das einheitliche Scrap-Konzept wird diese Zusammenarbeit in Form von Datenaustausch sichergestellt. Da die Scraps auf der Diskette nur eine Art Zwischenspeicher darstellen und bei der nächsten Benutzung ihren bisherigen Inhalt verlieren, wurde zusätzlich noch das Album-Konzept eingeführt. Photo Manager und Text Manager sind in der Lage, eine Vielzahl von Ausschnitten in Alben zu verwalten. Damit dient das Scrap sozusa-

gen als Transportmittel zum Aufbewahrungsort, dem Album, und von dort zurück zum Anwendungsprogramm.

1.2 Die Versionen von GEOS

Es begann mit GEOS V1.0. Dies war eine praktisch fertige Vorversion, die in Deutschland überhaupt nicht vertrieben wurde. So war beispielsweise das Aussehen einiger Icons noch anders. Diese Version unterstützte das Umblättern zwischen verschiedenen Seiten des Inhaltsverzeichnisses per Tastendruck nicht, und mit GEOPaint konnten keine mehrfarbigen Grafiken erstellt werden. Man erkennt GEOS V1.0 beispielsweise am Aussehen des Mülleimers (Trashcan). Über eine ebenfalls in Deutschland nicht vertriebene Version GEOS V1.1 entstand dann die endgültige Version GEOS V1.2, die vielen C64ern kostenlos beigelegt wurde, wenn auch zuerst nur als Gutschein. Der größte Nachteil dieser Version war die ausschließliche Anpassung an den amerikanischen Markt. So fehlen GEOS V1.2 und seinen Applications die deutschen Umlaute, was den Einsatz als Textverarbeitung beispielsweise erheblich einschränkt. Deshalb haben wir den FONT-Editor entwickelt und in diesem Buch abgedruckt. Mit ihm ist eine einfache und komfortable Umstellung der Zeichensätze (Fonts) auf die deutschen Umlaute möglich.

Auf Grund des großen Erfolges entwickelte Berkley Softworks GEOS als Gesamtsystem ständig weiter, und zwar in zweifacher Hinsicht: Zum einen wurden immer neue Anwendungsprogramme und Hilfsmittel erstellt, so daß die fehlende Kompatibilität zum alten Betriebssystem einen immer kleineren Nachteil darstellt. Den Höhepunkt erreichte diese Entwicklung durch GEOPublish, ein komplettes Desktop-Publishing-Programm, das den C64 und GEOS bis an die Grenzen des möglichen ausreizt. Zum anderen wurde das GEOS-Betriebssystem weiterentwickelt, so daß die augenblicklich aktuellste Version GEOS 1.3 ist. Einige Besonderheiten sind die erheblich verstärkte Unterstützung der Tastatur, so daß jetzt wichtige Funktionen (Öffnen einer Diskette) auch durch Tastendruck möglich sind. Weiterhin unterstützt GEOS neben dem langsamen Grafikdruck jetzt auch

den Text-Modus, der erheblich schneller ist, bei dem aber auf Proportionalschrift und verschiedene Fonts verzichtet werden muß.

Durch die Entwicklung von Speichererweiterungen für C64 und C128 konnte zwar nicht die Speichernot unter GEOS beseitigt werden. Dafür war GEOS schon "zu fertig", und eine spezielle Version für Computer mit Speichererweiterung wollte Berkley Softworks aus Kompatibilitätsgründen wohl nicht entwickeln. Trotzdem unterstützt GEOS diese Speichererweiterungen und gewinnt damit enorm an Geschwindigkeit.

Ende 1987 kam dann für deutsche GEOS-Benutzer der entscheidende Durchbruch. GEOS V1.3 wurde komplett eingedeutscht. Dabei wurden nicht nur Fehlermeldungen und Menüs übersetzt, sondern auch die deutschen Umlaute implementiert. Dabei wurde sehr sorgfältig vorgegangen, so daß die Tastaturbelegung in etwa einer Schreibmaschinen-Tastatur entspricht. Diese GEOS-Version wird allerdings nicht zusammen mit dem C64 ausgeliefert, sondern muß extra erworben werden. Ein Update von GEOS V1.2 auf die deutsche Version V1.3 ist nach unserer Information zu einem günstigen Preis möglich.

Hatten C128-Besitzer schon immer auf eine Anpassung guter Produkte an die Fähigkeiten Ihres Rechners verzichten müssen und stellte oft das "GO 64" den ersten Schritt zur Nutzung des C128 dar, so wurden sie von den GEOS-Entwicklern angenehm überrascht. Diese entwickelten nämlich eine spezielle Version GEOS 128, die sowohl den 80-Zeichen-Bildschirm (doppelte Auflösung in waagerechter Richtung) als auch den größeren Speicher des C128 benutzt.

Wann die Weiterentwicklung von GEOS beendet ist, ist im Augenblick (glücklicherweise) nicht abzusehen. So kündigt sich aus den Staaten ein GEOS-BASIC an, und für eine GEOS-Version V1.4 haben die Entwickler bei Berkley Softworks sicherlich genug Ideen.

1.3 Bootvorgang und Kopierschutz

Den Entwicklern von GEOS war natürlich klar, daß GEOS nicht nur Begeisterung bei ehrlichen Käufern wecken würde. Somit mußte ein Schutzmechanismus entwickelt werden, der einerseits die einfache und problemlose Nutzung von GEOS mit seinen Zusatzprogrammen sicherstellt und andererseits das Erstellen von Raubkopien verhindert. Dieser Schutz teilt sich in zwei Komponenten. Zum einen wird das GEOS-Betriebssystem geschützt, zum anderen kann sich jedes Zusatzprogramm durch die Überprüfung einer Seriennummer vor illegaler Verbreitung schützen. Während wir die zweite Schutzkomponente im folgenden Kapitel erläutern wollen, soll in diesem Abschnitt der Bootvorgang mit seinen Schutzmechanismen erläutert werden.

Mit derselben Gründlichkeit und Sorgfalt, mit der das Betriebssystem GEOS entwickelt wurde, widmeten sich die Entwickler auch dem Schutz des GEOS Betriebssystems. Da dieser Schutz mit den verschiedenen Versionen weiterentwickelt und verbessert wurde, beginnen wir bei der Version GEOS V1.2. Das Grundprinzip des Schutzes besteht darin, ein Laden und Starten von GEOS zu verhindern.

Kopierschutz: GEOS V1.2

Der erste Schutzmechanismus besteht im Autostart. Durch

```
LOAD "GEOS",8,1
```

wird ein kleines Programm in den Rechnerspeicher geladen, das sofort die Kontrolle über den C64 übernimmt. Dies geschieht dadurch, daß es in einen speziellen Bereich geladen wird (Stack), in dem der Prozessor beim Aufruf von Unterprogrammen die Rücksprungadresse ablegt. Auch das Laden eines Programms (hier GEOS) stellt einen Unterprogramm-Aufruf dar. Das GEOS-Startprogramm überschreibt nun praktisch den gesamten Stack (256 Bytes) mit einer einzigen Rücksprungadresse, die genau dem Start des Ladeprogramms entspricht. Damit wird dem C64 die Möglichkeit genommen, nach dem Laden (Unterprogramm) zum eigentlichen Betriebssystem zurückzukehren. Er

wird vielmehr gezwungen, ab jetzt das Lade-Programm abzuarbeiten (Autostart). Dieses tut nun nichts anderes, als "GEOS BOOT" nach \$6000 zu laden und dort einzuspringen.

GEOS BOOT liegt teilweise kodiert vor. Die für das Übertragen des Speeders notwendigen Routinen wurden mit einem Wert EOR-verknüpft. Deshalb erfolgt vor der Benutzung dieser Routinen innerhalb eines Unterprogramms die Dekodierung. Interessanterweise wird die Dekodier-Routine nicht direkt aufgerufen, vielmehr wird die Adresse auf den Stack geschrieben, und der Rücksprung aus der Unterroutine landet nicht beim aufrufenden Hauptprogramm, sondern in der Dekodier-Routine. Diese reichlich komplizierte Vorgehensweise dient natürlich einzig und allein dem Zweck, ein Nachverfolgen dieser Vorgänge zu erschweren.

Sind die Übertragungsroutinen dekodiert, wird der Speeder zur Floppy geschickt und dort gestartet. Dieser Speeder enthält auch die Kopierschutzabfrage. Wird der Kopierschutz entdeckt, übertragen die Routinen das KERNAL in drei Teilen (VLIR-Format) zum Rechnerspeicher, andernfalls nichts. Daran merkt GEOS BOOT, daß der Kopierschutz nicht korrekt entdeckt wurde, und leitet einen RESET ein.

Beim eigentlichen Kopierschutz haben sich die Entwickler etwas ganz Besonderes einfallen lassen. Zum einen sollte dieser Schutz so gut (unkopierbar) wie eben möglich sein, zum anderen sollte eine Wiederherstellung der Originaldiskette von einer Sicherheitskopie aus möglich sein. Daraus folgte automatisch, daß der Schutz nicht im normalen Zugriffsbereich der Floppy (Track 1-35) liegen durfte. Denn dieser Bereich würde beim Zurückkopieren der Backup-Diskette auf das defekte Original überschrieben und ein möglicher Schutz zerstört. Der Kopierschutz wurde also auf Track 36 aufgetragen.

Bei diesem Schutz benutzte Berkley Softworks ein bis dahin kaum bekanntes und genutztes Verfahren, das Auftragen von Bytes ohne Sync-Markierung. Sollten Sie sich für dieses Schutzverfahren und seine Hintergründe näher interessieren, dann können Sie entsprechende Informationen in "Das Anti-Cracker-

Buch" von DATA BECKER im Kapitel 6.6.4 "Daten ohne SYNC-Markierung" nachlesen. An dieser Stelle wollen wir nur das Grundprinzip erklären.

Normalerweise teilt das Floppy-DOS die Diskette in Tracks und diese in Sektoren ein. Um nun genau den Beginn der Sektoren und deren Daten erkennen zu können, wird ein spezielles Byte auf die Diskette geschrieben, das SYNC-Byte (\$FF). Dieses Byte darf natürlich ansonsten auf dem Track nicht vorkommen. Da normale Daten aber auch dieses Byte enthalten, werden die Daten erst GCR-kodiert, so daß in gültigen Daten kein \$FF mehr vorkommen kann. Nun ist es mit geeigneten Routinen aber problemlos möglich, Daten ohne jede SYNC-Markierung auf die Diskette zu schreiben und auch wieder zu lesen.

Der GEOS-Kopierschutz der Version V1.2 besteht nun aus zwei Komponenten: Zum einen darf die Spur 36 keine einzige SYNC-Markierung enthalten, zum anderen muß eine Folge von 3 Bytes gefunden werden. Wird eine SYNC-Markierung entdeckt oder werden die drei Bytes nicht gefunden, weiß GEOS, daß es sich um eine Kopie handelt und reagiert dementsprechend. Übrigens waren die legendären Bytes der ersten Version GEOS V1.2: \$2F, \$53, \$77. Damals konnte man diese notwendige Byte-Folge noch direkt durch Analyse des Speeders mit seinen Abfrage-Routinen herausbekommen, in neueren Versionen ist das geschickt verhindert worden.

Sicherlich haben viele versucht, diesen Schutz (aus welchen Gründen auch immer) zu überlisten, und es vielleicht auch geschafft, die Abfrage aus den Speeder-Routinen zu entfernen. Der andere Weg, das Auftragen der richtigen Bytes, ist nämlich recht schwierig und erfordert intime Kenntnisse der Vorgänge. Trotz der geänderten Abfrageroutinen wird wohl vielen der Erfolg verwehrt geblieben sein. Zwar wurde GEOS anschließend komplett geladen, aber die Maus ließ sich nicht mehr bewegen, und auf die Tastatur reagierte GEOS auch nicht mehr. Wahrscheinlich ist so mancher nicht darauf gekommen, daß die Entwickler dies vorausgesehen haben und deshalb einen weiteren Schutzmechanismus installierten.

Nach dem Laden des KERNAL durch GEOS BOOT wird in die Initialisierungsroutine des KERNAL eingesprungen. An einer versteckten Stelle bildet sie eine Prüfsumme über den Speicherbereich ab \$6000 (dort liegt ja GEOS BOOT), und aus dieser Prüfsumme wird der Interrupt-Vektor berechnet und bei \$FFFE/\$FFFF eingetragen. Jede Änderung am Boot-Programm führt automatisch zu einer falschen Prüfsumme und damit zu einem falschen Interrupt-Vektor. Die dafür verantwortliche Prüfroutine findet man übrigens leicht, wenn man weiß, daß sie die Startadresse für den Prüfvorgang \$6000 in einen Zeiger in der ZERO-Page schreibt.

Erst wenn man statt der Prüfsumme gleich die richtigen Werte einträgt, wird GEOS nicht nur korrekt geladen, sondern funktioniert anschließend auch richtig.

Kopierschutz: GEOS V1.3

In den bisherigen Auflagen von "Das große GEOS Buch" und "GEOS Tips und Tricks" haben wir schon eine Reihe von Hilfsmitteln vorgestellt, um Sicherheitskopien der Original-Diskette, vereinfachtes Laden und einen Warmstart von GEOS zu ermöglichen. Natürlich waren wir äußerst gespannt auf die Version GEOS V1.3, um herauszufinden, ob die Entwickler nicht nur GEOS, sondern auch den Schutz weiterentwickelt haben. Die Antwort lautet: ja und zwar mit noch raffinierteren Methoden. Anscheinend haben unsere kleinen Utility-Programme (die ja auch in den USA veröffentlicht wurden) Berkley Softworks zu Höchstleistungen angespornt.

Schon die Tatsache, daß mit der neuen GEOS-Version zwei bootfähige Disketten ausgeliefert werden (SYSTEM DISK und BACKUP SYSTEM) gibt einem zu denken. Auch fehlt im Handbuch der neuen Version jeder Hinweis auf die anzufertigende Sicherungskopie, mit deren Hilfe man die Originaldiskette wieder restaurieren könnte. Dies ist auch nicht mehr möglich. Der Kopierschutz befindet sich nun bei der SYSTEM DISK mitten im Bereich der vom DOS genutzten Tracks und würde durch das "Draufkopieren" eines Backups zerstört werden. Damit sind Sie völlig von den beiden Disketten abhängig und müssen

sie sorgfältig behandeln, dürfen keine Veränderungen vornehmen ... außer Sie benutzen das in guter Tradition von uns auch für GEOS V1.3 entwickelte Hilfsmittel zum Anfertigen von Sicherungskopien.

GEOS V1.3 ist so gut geschützt, daß unserer Erfahrung nach bis zum heutigen Zeitpunkt keine illegalen Kopien in den entsprechenden Kreisen kursieren. Der Grund ist folgender: Der Kopierschutz befindet sich nicht mehr auf Track 36, sondern auf Track 19 mitten zwischen den normalen Daten. Auch bei diesem Schutz wurden Bytes ohne SYNC geschrieben und werden abgefragt, diese liegen allerdings genau in den Lücken zwischen den gültigen Daten. Damit ist es äußerst schwierig geworden, diese Bytes nachträglich auf die Diskette zu bekommen.

Nun könnte man natürlich versuchen, eine Veränderung des Speeders mit seiner Kopierschutzabfrage vorzunehmen, aber dem haben die Entwickler einen wirksamen Riegel vorgeschoben. Ein wichtiger Teil des KERNAL befindet sich nämlich in der neuen Version kodiert auf der Diskette (VLIR II und III). Dabei wird nicht etwa mit einem festen Wert kodiert (EOR) sondern der entscheidende Teil des Speeders selbst zur Kodierung herangezogen. Somit führt jede Änderung an diesen Routinen zu einem fehlerhaften Laden des KERNAL.

Nun kann man natürlich das KERNAL mit einem eigenen Programm dekodieren und anschließend die Speeder-Routinen entsprechend ändern. Anschließend wird zwar GEOS sichtbar bis zu Ende geladen, aber die Freude endet mit einem SYSTEM ERROR NEAR Wer zuerst noch glaubt, dies läge an einer fehlerhaften Dekodierung (was läge näher) oder an falsch geänderten Speeder-Routinen wird bald eines Besseren belehrt. Es gibt einen anderen Grund für dieses Verhalten, der in einer weiteren Schutzfunktion liegt, vergleichbar der Prüfsummenbildung durch das KERNAL in GEOS V1.2.

Nun kann man das gesamte GEOS BOOT mit dem übertragenen Speeder und ebenfalls KERNAL und DESK TOP bis ans Ende seiner Tage durchsuchen, ohne diesen Schutz jemals zu entdecken.

Dies ist wohl auch der Grund dafür, daß bisher keine Kopien von GEOS V1.3 aufgetaucht sind. Dieser zweite Schutz funktioniert so:

1. Die zweite Schutzroutine existiert an der Stelle, an der sie ausgeführt wird, nur für einen kurzen Augenblick und wird anschließend sofort gelöscht.
2. Es gibt keine spezielle Laderoutine für diese Schutzroutine, und sie wird nirgendwo sichtbar aufgerufen.

Nun haben wir Sie aber lange genug auf die Folter gespannt und wollen Ihnen diesen genialen Trick verraten. Der letzte VLIR-Record enthält hinter den gültigen Daten, die ganz normal zum C64 geschickt werden, die zweite Kopierschutzabfrage. Wahrscheinlich wissen Sie, daß die einzelnen Sektoren eines Files verkettet sind, indem jeweils die ersten beiden Bytes eines Sektors Track und Sektor des folgenden enthalten. Steht dort als erster Wert eine \$00 (0 als Track gibt es nicht), so hat man den letzten Sektor, und der zweite Wert enthält die Anzahl der gültigen Daten. Steht dort also bei einer Datei \$00, \$0A, so enthält der letzte Sektor 10 gültige Bytes, und nur diese werden auch von der Floppy an den Rechner übertragen. Das funktioniert unter GEOS genauso wie beim normalen Floppy-DOS.

Allerdings kümmert sich die Floppy nicht um die Daten, die in diesem Fall ab dem 11. Byte beginnen. Was dort steht, wird nicht berücksichtigt. Da das DOS aber immer ganze Sektoren in den Floppypuffer lädt, werden auch die restlichen (ungültigen) Bytes geladen. Diese Bytes stellen aber beim 3. VLIR-Record gerade die 2. Kopierschutzabfrage dar. Nachdem nun klar ist, woher die Routine kommt und wie sie (unbemerkt) geladen wird, bleibt noch die Frage, wer sie startet. Denn einen expliziten Sprung in einen Bereich des Floppypuffers, in dem kein Speeder ist, sondern Diskettendaten sind, würde möglicherweise auffallen.

Die zweite Abfrage wird absolut unauffällig vom GEOS KERNAL aus aktiviert. Dazu muß man wissen, daß GEOS für die 1541 zwei verschiedene Speeder hat. Der eine wird von

GEOS BOOT in der Floppy gestartet und enthält auch die erste Kopierschutzabfrage. Ist das KERNAL gestartet, überträgt es einen neuen, erheblich geänderten Speeder zur Floppy. Dieser Speeder hat an seinem Schluß kein Ende (etwa JMP, BNE oder RTS). Dadurch "läuft er über sein eigenes Ende" in einen Bereich hinein, in dem keine gültigen Daten liegen würden - wenn dort nicht genau die zweite Abfrage liegen würde. Da sie sich im letzten Sektor des dritten VLIR-Records befindet, wurde der Puffer seit dem Laden nicht mehr verändert.

Der vom KERNAL zur Floppy geschickte Speeder überschreibt nun gerade die gültigen Daten des KERNAL, aber nicht den dahinter liegenden Schutz, und "läuft genau in diesen hinein". Nachdem die Abfrage erfolgt ist, wird ein Zustand hergestellt, bei dem nichts mehr auf diesen genialen Vorgang schließen läßt. Das Ende des neuen Speeders wird von der Kopierschutzroutine mit einem RTS abgeschlossen, und anschließend zerstört sie sich selbst.

Schaut nun jemand nach dem Verlassen von GEOS im Floppypuffer nach, ob sich dort ein Hinweis auf einen Kopierschutz befindet, so findet er nur einen absolut sauberen Speeder ohne Kopierschutzroutinen und ohne Sprünge zu Stellen außerhalb des Speeders, die mißtrauisch machen könnten.

Lieber Leser! Wir hoffen, daß Sie mit uns der Meinung sind, daß es sich gelohnt hat, diese Geheimnisse von GEOS zu lüften. Die Programmierer haben eine ausgezeichnete Leistung erbracht und die Aufgabe, die sie dadurch stellen, ist wohl schwieriger als manches Adventure. Auch wenn dieser Schutzmechanismus nun enthüllt ist, werden es wohl nur ausgezeichnete Programmierer schaffen, die notwendigen Schritte vorzunehmen, um eine Kopie herstellen zu können, und schon dadurch ist einer schnellen, illegalen Verbreitung von GEOS V1.3 weiterhin ein wirksamer Riegel vorgeschoben. Wir aber gehen fest davon aus, daß Berkley Softworks sich durch unsere Bemühungen genötigt sieht, diesem Abenteuer eine neue, noch bessere Version folgen zu lassen, und wir sind schon mächtig gespannt darauf. Vielleicht gibt es bald eine Version GEOS V1.4, bei der eine Kopierschutzabfrage in den Sprite-Registern des VIC liegt oder

etwa völlig unlokalisierbar auf dem seriellen Bus nur im dünnen Verbindungskabel versteckt existiert. Na ja, lassen wir uns überraschen.

1.4 Seriennummer und Installationsvorgang

Nicht erst seit GEOS V1.3 enthält jede GEOS-Originaldiskette eine Seriennummer. Dies war auch schon bei GEOS V1.2 der Fall, aber erst seit GEOS V1.3 muß das System installiert werden, und dabei wird die Seriennummer auf die Diskette geschrieben. Wozu dient diese Nummer?

Natürlich wollte man bei Berkley Softworks die zusätzlich herausgebrachten Programme (GEOWRITE 2.0, GEOFILE usw.) vor unrechtmäßigem Kopieren schützen. Andererseits mußten sie für den ehrlichen Anwender kopierbar sein, damit er diese auf mehrere Arbeitsdisketten kopieren kann. Also erhielt jedes GEOS KERNAL eine Zufallsnummer zwischen 1 und 65535, und diese Nummer wird beim ersten Start des Zusatzprogramms in dieses selbst eingetragen. Anschließend kann es auf beliebige Disketten kopiert werden, fragt aber bei jedem Start die Seriennummer des KERNAL ab und vergleicht sie mit der gespeicherten. Stimmen beide nicht überein, wird das Programm abgebrochen. Wenn Sie ein so installiertes Programm weitergeben, kann kaum jemand etwas damit anfangen, er müßte schon zufällig dieselbe Seriennummer haben.

Wir haben schon bei der Beschreibung des Boot-Vorgangs darauf hingewiesen, daß die Zufallszahl, die als Seriennummer benutzt wird, von GEOS BOOT beim Ladevorgang erzeugt wird. Zusätzlich zum KERNAL wird ab GEOS V1.3 noch ein Programmteil vor "GEOS BOOT" nach \$5000 geladen und übernimmt wichtige Initialisierungsaufgaben. Es durchsucht beispielsweise die gesamte Diskette nach Dateien mit dem Filetyp AUTOSTART (Selbstaufführend) und startet diese. Ebenfalls prüft dieser Programmteil, ob schon eine Seriennummer existiert. Ist dies noch nicht der Fall, wird der Installationsvorgang eingeleitet und dabei die erzeugte Zufallszahl ins KERNAL von SYSTEM DISK und BACKUP SYSTEM eingetragen. Um her-

auszubekommen, wie die Seriennummer eines KERNAL lautet, gibt es eine Routine in der Sprungtabelle: \$C196, die die Seriennummer in \$02,\$03 ablegt.

Wie schützen sich nun die Zusatzprogramme mit Hilfe dieser Seriennummer vor unberechtigtem Kopieren? Diese Zusatzprogramme enthalten gewöhnlich einen Sektor auf der Diskette, der die Kopierschutzabfrage enthält. Nach dem Start holt das Programm diesen Sektor mit den GEOS-Routinen in den Puffer ab \$8000 startet und prüft ihn auf Veränderung. Dazu wird eine Prüfsumme gebildet durch Addition des Bereichs von \$8002 bis \$80FE und mit der richtigen Prüfsumme in \$80FF verglichen. Stimmen die Ergebnisse nicht überein, so wurde an diesem Sektor etwas geändert, und das Programm bricht ab. Andernfalls wird in die Prüfroutine ab \$8002 eingesprungen.

Diese prüft den Schutz und liefert im X-Register \$00 zurück, wenn der Schutz in Ordnung ist, sonst \$FF. Ist der Schutz korrekt erkannt worden, holt das Programm die Seriennummer mittels \$C196 und lädt einen Sektor mitten aus dem eigenen Programm, in dem diese Nummer dann eingetragen wird. Dieser Sektor wird anschließend zurückgeschrieben, so daß die Seriennummer fest im Programm eingetragen ist. Bei jedem weiteren Start des Programms prüft dieses, ob die Seriennummer ungleich \$0000 ist (dies würde bedeuten: noch nicht installiert und die Installation auslösen) und ob die Nummer mit der Nummer des KERNAL übereinstimmt.

Natürlich haben auch hier die Entwickler teilweise noch einige Tricks angewandt, um diese Routinen zu schützen. So liegen sie bei einigen Zusatzprogrammen verschlüsselt auf Diskette vor, bei anderen wird nicht direkt in die Sprungtabelle gesprungen (das läßt sich doch recht leicht finden), sondern Akku und X-Register mit der Adresse geladen und eine Betriebssystemroutine aufgerufen, die den Sprung an diese Adresse veranlaßt. Wenn Sie wie wir mit mehreren GEOS-Versionen arbeiten, die unterschiedliche Seriennummern haben, und es Sie immer wieder stört, wenn ein Programm plötzlich verlangt, ein anderes KERNAL zu booten, so können Sie folgendermaßen vorgehen, wenn das Programm noch nicht installiert ist.

Erstellen Sie eine ganz normale Kopie des nicht installierten Programms. Starten Sie dieses von der Kopie, und warten Sie, bis das Fenster mit der entsprechenden Meldung auftaucht. Lösen Sie einen RESET aus, und suchen Sie mit einem Monitor die Stelle, wo die Seriennummer des KERNAL geholt wird. Verfolgen Sie das Programmstück weiter, bis ein Einsprung in die Routine \$C1E4 (Block von Diskette lesen) erfolgt. Merken Sie sich Track und Sektor dieses Blocks, und laden Sie ihn mit einem Floppy-Monitor in den Floppypuffer ab \$0300. Ändern Sie ab \$8002 das Programm einfach in

```
0302 LDX #$00
0304 RTS
```

Nun müssen Sie noch die Prüfsumme von \$0302 bis \$03FE bilden und in \$03FF eintragen. Wenn Sie nun diesen Block auf Diskette zurückschreiben, wird sich diese Kopie anstandslos installieren lassen. Vorher erstellen Sie am besten eine Kopie dieser Diskette, so daß Sie bei einer anderen GEOS-Version erneut installieren können. Sie sollten dann allerdings genau notieren, welche Programmversion mit welcher GEOS-Version zusammenarbeitet. Wir müssen ja hoffentlich nicht extra darauf hinweisen, daß die COPYRIGHT-Bestimmungen die Weitergabe von Kopien unter Strafe stellen. Wir geben Ihnen diese Tips ausschließlich für die eigene Nutzung der Programme.

Erstellen Sie eine ganz normale Kopie des nicht installierten Programms. Starten Sie dieses von der Kopie, und warten Sie, bis das Fenster mit der entsprechenden Meldung auftaucht. Lösen Sie einen RESET aus, und suchen Sie mit einem Monitor die Stelle, wo die Seriennummer des KERNAL geholt wird. Verfolgen Sie das Programmstück weiter, bis ein Einsprung in die Routine \$C1E4 (Block von Diskette lesen) erfolgt. Merken Sie sich Track und Sektor dieses Blocks, und laden Sie ihn mit einem Floppy-Monitor in den Floppypuffer ab \$0300. Ändern Sie ab \$8002 das Programm einfach in

```
0302 LDX #$00
0304 RTS
```

Nun müssen Sie noch die Prüfsumme von \$0302 bis \$03FE bilden und in \$03FF eintragen. Wenn Sie nun diesen Block auf Diskette zurückschreiben, wird sich diese Kopie anstandslos installieren lassen. Vorher erstellen Sie am besten eine Kopie dieser Diskette, so daß Sie bei einer anderen GEOS-Version erneut installieren können. Sie sollten dann allerdings genau notieren, welche Programmversion mit welcher GEOS-Version zusammenarbeitet. Wir müssen ja hoffentlich nicht extra darauf hinweisen, daß die COPYRIGHT-Bestimmungen die Weitergabe von Kopien unter Strafe stellen. Wir geben Ihnen diese Tips ausschließlich für die eigene Nutzung der Programme.

2. Tips und Tricks zur Benutzeroberfläche

2.1 Laden und Starten

Was nützen Ihnen die schönsten Tips zur Benutzung, wenn Sie GEOS vielleicht gar nicht richtig laden können. Es gibt nämlich mehrere Punkte, an denen der Beginn Ihrer Arbeit mit GEOS schon scheitern kann:

GEOS hat einen ausgezeichneten Kopierschutz. Er wird während des Boot-Vorgangs abgefragt. Wird er innerhalb einer bestimmten Zeit nicht gefunden, so löst GEOS einen RESET des C64 aus. Auf dem Bildschirm erscheint wieder die Einschaltmeldung des C64. Was können Sie nun tun, wenn GEOS den Kopierschutz auf der Originaldiskette einfach nicht akzeptieren will?

Tip 1: Die BOOT-Pause

Wenn Sie gerade einen mißglückten Versuch erlebt haben, Booten Sie GEOS nicht sofort wieder. Warten Sie einen Augenblick, bis die Diskette im Laufwerk zum Stillstand gekommen ist. Oft genügt das schon, um einen erfolgreichen BOOT-Vorgang zu starten.

Tip 2: Floppy-Justierung mit dem "Bump"

Sollte GEOS dann noch nicht Booten, so kann es sein, daß der Kopf der Floppy etwas dejustiert ist. Dagegen hilft ein sogenannter "Bump". Das ist das Anschlagen des Kopfes gegen eine vorgesehene Begrenzung in der Floppy. Dieser "Bump" wird vor jedem Formatieren ausgeführt, um den Kopf genau zu positionieren. Mit einem kleinen Programm können Sie diesen "Bump" auch auslösen.

Achtung: Diesen "Bump" sollten Sie nicht bei sehr warmer Floppy starten. Wenn das Laufwerk mehrere Stunden eingeschaltet war und in den letzten Minuten viel geladen und gespeichert wurde, sollten Sie die 1541 für ca. 10 Minuten ausschalten, weil sonst der Tonkopf noch mehr dejustiert werden könnte. War das Laufwerk nur einfach eingeschaltet, so ist eigentlich nichts zu befürchten.)

```
10 REM "Bump"  
20 REM "Dieses Programm fährt den Kopf an den "  
30 REM "Anschlag und justiert ihn neu"  
40 OPEN 15,8,15,"I":REM Kanal 15 als Befehlskanal öffnen"  
50 PRINT#15, "M-W"CHR$(0)CHR$(0)CHR$(0)CHR$(192)  
60 CLOSE15
```

Speichern Sie dieses Programm, und starten Sie es anschließend mit RUN. Sie hören dann deutlich das Anschlagen des Kopfes. Nun sollte GEOS beim Booten den Kopierschutz ordnungsgemäß entdecken.

Tip 3: Letzte Rettung: "Sicherheitskopie"

Sollte auch ein "Bump" keine Lösung bringen (immer vorausgesetzt, Sie haben auch wirklich die Originaldiskette eingelegt), so finden Besitzer dieses Buches noch eine letzte Rettung. Wir haben nämlich für all die Benutzer von GEOS, die bei der täglichen Arbeit um ihre Originaldiskette fürchten, nach Mitteln und Wegen gesucht, ein Booten von einer Sicherheitskopie zu ermöglichen.

Die Lösung dieses Problems und damit eine echte Hilfe bei Ihrer Arbeit mit GEOS finden Sie in diesem Buch. Gleichzeitig können Sie damit eventuell auch das Problem, daß GEOS nicht Booten will, lösen. Wenn Sie GEOS V1.3 deutsch haben, dann erstellen Sie mit den Programmen aus Kapitel 4 eine Sicherheitskopie. Von dieser aus müßte dann das Booten eigentlich funktionieren.

Tip 4: Drucker ausschalten

Zum Abschluß noch ein letzter Tip, der vielleicht entscheidend dazu beitragen kann, daß Sie überhaupt anfangen können, mit GEOS zu arbeiten. Sollte GEOS nämlich kurz vor dem Ende des Ladevorgangs stehenbleiben, wenn schon oben am Bildschirmrand die Menüleiste aufgebaut wurde, so schalten Sie einfach den angeschlossenen Drucker aus. GEOS hatte insgesamt einige Schwierigkeiten mit angeschlossenen Druckern. Beispielsweise lief unser EPSON FX80 nicht mit der zugehörigen Druckeranpassung, sondern nur als COMMODORE kompatibler Drucker.

Am besten schalten Sie einen vorhandenen Drucker jeweils nur direkt zum Drucken ein und nach Beendigung des Druckens wieder aus. Wenn Ihr Drucker einen großen Puffer hat, so drucken Sie am besten nicht vom DESK TOP aus, sondern von der Anwendung (Application), mit der das Dokument erstellt wurde. Wenn Sie nämlich vom DESK TOP drucken und GEOS ist eigentlich fertig (weil der Rest des Dokumentes im Puffer des Druckers ist), wird ein RESET in GEOS ausgelöst. Der kann aber nicht zu Ende bearbeitet werden, weil der Drucker noch eingeschaltet ist. Sie müssen dann (trotz Druckerpuffer) so lange warten, bis der Drucker fertig ist und Sie ihn ausschalten können.

Übrigens wundern Sie sich nicht, wenn nach dem Ausschalten des Druckers gelegentlich ein zweites Diskettensymbol am rechten Rand mit einem Fragezeichen auftaucht. GEOS scheint den Drucker am seriellen Bus als zusätzliche Floppy zu erkennen und zeigt daher das zweite Diskettensymbol an. Da diese "Pseudo-Floppy" aber nicht richtig antwortet, wird das Fragezeichen ausgegeben.

Sollte Sie dieses Fragezeichen stören, so klicken Sie (bei ausgeschaltetem Drucker) unter SPECIAL das Feld RESET an. Dann verschwindet das merkwürdige "Disketten-Fragezeichen".

2.2 DESK TOP

Durch GEOS kann man auf sehr komfortable Weise mit dem C64 arbeiten. Damit die vielen alltäglichen Probleme gelöst werden können (Files kopieren, löschen, umbenennen, Disketten umbenennen, formatieren usw.), gibt es DESK TOP. Bei der (sehr bequemen) Arbeit mit DESK TOP kann man durch folgende Möglichkeiten noch mehr Leistung erzielen.

Tip 5: Verschiedene Namen für die Disketten wählen

Geben Sie bitte allen Disketten unterschiedliche Namen. GEOS erkennt diese nämlich daran und kann sie bei verschiedenen Namen unterscheiden. Vielleicht haben Sie schon gemerkt, daß GEOS beim Kopieren die gewünschte Diskette beim Namen nennt. Deshalb gibt es spätestens bei einem Kopierversuch zwischen zwei gleichnamigen Disketten Schwierigkeiten. Plötzlich verschwinden Icons von der BORDER (dem unteren Rand, auf dem man sie zum Kopieren ablegt), oder GEOS bricht den Kopiervorgang an einer Stelle ab.

Tip 6: Von der Originaldiskette nur starten

Nehmen Sie bitte die Originaldiskette sofort nach dem Booten aus dem Laufwerk, und legen Sie sie wieder an einen sicheren Ort. Es ist nicht nur verhängnisvoll, wenn Sie Kaffee darüber gießen, oder ein wichtiges File löschen. Es ist schon kein Booten mehr möglich, wenn Sie GEOS, GEOS BOOT oder GEOS KERNAL auf die BORDER setzen und GEOS dann verlassen.

Kopieren Sie also *kein* einzelnes File von der Originaldiskette. Stellen Sie sich vor, Sie haben es gerade auf der BORDER abgesetzt, und aus irgendeinem Grund fällt der Strom aus. Dann müßten Sie erneut Booten, aber dabei wird ein eventuell notwendiges File nicht gefunden (weil es auf der BORDER ist und damit außerhalb des normalen Inhaltsverzeichnisses).

Das allerbeste ist, Sie erstellen mit Hilfe des in diesem Buch abgedruckten Programms eine oder mehrere Sicherheitskopien, von denen Sie Booten können. Dann kommt die Originaldiskette in den abschließbaren Diskettenkasten, und das Problem tritt nie auf.

Tip 7: Nach Diskettenwechsel immer OPEN anklicken

Vergessen Sie bitte nicht, nach jedem Diskettenwechsel die neue Diskette mit OPEN anzumelden. Im harmlosesten Fall verlangt DESK TOP irgendwann in einer unpassenden Situation die Diskette, die Sie gerade weggelegt haben. Es können aber auch unangenehmere Dinge passieren.

Tip 8: Was Sie für die Arbeit mit einer Anwendung benötigen, muß auf einer Diskette sein.

Eine andere Diskette können Sie praktisch nur während Ihrer Arbeit mit DESK TOP einlegen (und mit OPEN aktualisieren). In den anderen Programmen ist ein Diskettenwechsel nicht vorgesehen. Wenn Sie einmal viel Zeit und eine Diskette haben, auf der absolut nichts Wichtiges ist (sie muß aber formatiert sein), so machen Sie einmal folgendes:

Erstellen Sie ein unwichtiges Dokument mit GEOPAINT. Nennen Sie es einfach "Unwichtig", und ziehen Sie ein paar Linien. Dabei sollten Sie das GEOPAINT-Fenster auf mehrere Stellen des gesamten Blattes setzen, damit GEOPAINT mehrfach speichern muß. Zum Schluß setzen Sie das GEOPAINT-Fenster wieder oben links in die Ecke und klicken die SCROLL-Pfeile an. Nehmen Sie die Diskette aus dem Laufwerk, und legen Sie die Diskette, auf der sich nichts Wichtiges befindet und die gelöscht werden kann, in das Laufwerk.

Achtung: Der Inhalt dieser Diskette ist anschließend wirklich unbrauchbar. Bewegen Sie das GEOPAINT-Fenster mit dem SCROLL-Zeiger nach rechts und anschließend nach unten. Das Ergebnis sind schöne

surrealistische Zeichnungen: Computeranimation. Zum Abschluß klicken Sie dann einmal. Das gibt es dann einen wunderschönen Systemabsturz mit der Meldung:

SYSTEM ERROR NEAR ...

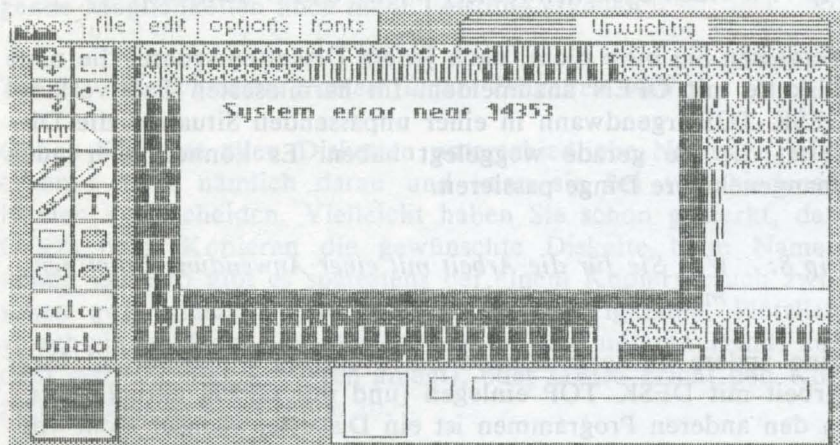


Abb. 1: Systemabsturz durch Wechseln der Diskette.

Das kommt daher, daß GEOPAINT für bestimmte Funktionen (zum Beispiel das Scrollen) Programmteile nachladen muß. Wenn Sie die Diskette vorher entnehmen, wird nicht dieses Programm-Modul, sondern irgend etwas nachgeladen. Dadurch stürzt GEOS dann natürlich ab. Kopieren Sie also für Ihre Arbeit mit Applications und Accessories alles, was Sie benötigen, auf eine Diskette, so daß Sie diese nicht zu wechseln brauchen.

Tip 9: VALIDATE und SCRATCH nur von GEOS aus

GEOS und das COMMODORE-DOS (das Disk Operating System, das Sie normalerweise von BASIC aus benutzen) sind einigermaßen kompatibel. So können Sie mit GEOS alle alten, ohne

GEOS erstellten Files bearbeiten und auf Disketten mit GEOS-Files Programme von BASIC aus speichern.

Wichtig ist aber in diesem Zusammenhang das Wörtchen "eini-germaßen". Wir raten Ihnen, nicht von BASIC aus auf GEOS-Disketten zu speichern. Irgendwann werden Sie nämlich versehentlich ein File löschen oder den VALIDATE-Befehl ausführen. Das hat aber auf einer GEOS-Diskette verheerende Folgen. Der Grund dafür ist ganz einfach: GEOS verwaltet auf einer Diskette mehr Informationen als das COMMODORE-DOS. Beispielsweise wird ein zusätzlicher Sektor für die BORDER und zu jedem GEOS-File ein INFO-Sektor angelegt. Solche Sektoren werden aber vom normalen DOS nicht erkannt und daher zum Beschreiben freigegeben. Wenn Sie nun etwas von BASIC aus abspeichern, werden diese Sektoren einfach mitbenutzt, und damit wird die GEOS-Diskette unbrauchbar.

Benutzen Sie also nie die Befehle SCRATCH und VALIDATE von BASIC aus bei GEOS-Disketten. Sollte das doch einmal versehentlich geschehen sein, so speichern Sie nichts mehr auf dieser Diskette, bevor Sie nicht von GEOS aus ein VALIDATE durchgeführt haben. Der GEOS-Befehl erkennt natürlich die zusätzlich benutzten Sektoren und belegt sie in der BAM. Damit sind sie vor Überschreiben geschützt.

Tip 10: Keine Systemnamen für eigene Dokumente

Geben Sie bitte niemals einem in GEOS erstellten Dokument einen Systemnamen. Damit meinen wir die Namen aller auf der Originaldiskette vorhandenen Files und zusätzlich die Namen der von GEOS-Programmen erstellten Datenfiles: NOTES, PREFERENCES und Swap-File. Sie riskieren damit nämlich entweder einen Systemabsturz (GEOS hängt sich auf) oder den totalen Verlust Ihrer Daten mit so einem Namen.

Wenn Sie beispielsweise einen Text "Notes" nennen, hängt sich das Accessory NOTE PAD beim ersten Steuerzeichen auf. (GEOWRITE markiert mit diesen Steuerzeichen die verschiedenen Schriften und Tabulatorsprünge.) Zusätzlich ist wahrscheinlich auch Ihr Text "Notes" danach zerstört. Wenn Sie eine

Grafik "Preferences" nennen, haben Sie zwar interessante Zeichen im PREFERENCE MANAGER (nach den Anklicken von LOAD im P.M.), aber jedes Abspeichern löscht Ihre Grafik.

Tip 11: VALIDATE schafft Platz

Benutzen Sie gelegentlich das Kommando VALIDATE (aber nur von GEOS aus). Manchmal können Sie dadurch einige zusätzliche Kilobyte auf der Diskette gewinnen. Es scheint, daß GEOS manchmal beim Löschen von Files nicht alle Sektoren freigibt. Wenn Sie also schon längere Zeit mit einer Diskette arbeiten, nur noch wenig Platz haben und sowieso einen Augenblick Pause machen wollen, klicken Sie VALIDATE an. Das dauert je nach Anzahl und Größe der vorhandenen Files mehrere Minuten, aber gelegentlich zeigt GEOS danach einige zusätzliche KBytes als verfügbaren Diskettenspeicher an.

Tip 12: Nie die Floppy ausschalten

Schalten Sie nie während Ihrer Arbeit mit GEOS die Floppy aus. Es gibt nur eine Ausnahme: Wenn Sie ein zweites Diskettenlaufwerk anmelden wollen, werden Sie ausdrücklich dazu aufgefordert. GEOS legt im Speicher der Floppy nämlich einen Teil des Speeders (Floppy-Beschleunigers) ab. Wenn Sie die Floppy ausschalten, wird dieser Speeder gelöscht. Das hat nicht etwa zu Folge, daß jetzt alle Diskettenoperationen gewohnt langsam ablaufen, sondern daß GEOS beim nächsten Diskettenzugriff einfach abstürzt. Kommen wir nun nach den vielen "Vorschriften" zu den Möglichkeiten, Ihre Arbeit mit GEOS zu vereinfachen und zu beschleunigen.

Tip 13: Preferences können für jede Diskette angelegt werden

Ihre eigene Einstellung der Farben, Mausgeschwindigkeit usw. können Sie durch Abspeichern aus dem PREFERENCE MANAGER heraus festlegen. Wichtig ist dabei, daß Sie diese Werte für jede einzelne Diskette festlegen können. Beim Öffnen einer Diskette mit dem GEOS-Befehl OPEN werden diese voreingestellten Werte gelesen und gesetzt. So können Sie also auf einer

DESK-TOP-Diskette die maximale Mausgeschwindigkeit einstellen, auf einer GEOPAINT-Diskette dagegen eine langsamere Geschwindigkeit für genaues Arbeiten.

Sie brauchen nicht zu befürchten, daß das von Ihnen eingestellte Datum durch das Öffnen einer Diskette mit anderen PREFERENCES verändert wird. Alle Einstellungen werden aus PREFERENCES übernommen bis auf Datum und Zeit, die unverändert erhalten bleiben.

Wenn Sie übrigens mit einer "geänderten Maus" experimentieren wollen, Ihnen also die Form des Pfeils nicht gefällt, so sollten Sie bei der Neuerstellung beachten, daß in jedem Fall die linke obere Ecke beim Klicken entscheidend ist. Selbst wenn Ihre neue "Maus" nur aus einem Punkt in der rechten unteren Ecke besteht, zählt die linke obere Ecke beim Klicken.

Tip 14: Automatisch den richtigen Drucker gewählt

Da GEOS mehrere Druckeranpassungen auf der Originaldiskette gespeichert hat, müssen Sie vor dem Drucken den richtigen Druckertreiber auswählen. Wenn Ihnen das zu lästig ist oder Sie es häufig vergessen, gibt es zwei Lösungsmöglichkeiten:

- Sie entfernen auf der Diskette alle Druckertreiber bis auf den einen, der zu Ihrem Drucker paßt. Dieser ist dann automatisch voreingestellt, so daß Sie die Anpassung nicht mehr wählen müssen.
- Wenn Sie aus irgendeinem Grund lieber mehrere Druckertreiber auf der Arbeitsdiskette haben möchten, muß der von Ihnen gewünschte einfach nur der erste sein. GEOS benutzt als Voreinstellung also den ersten Treiber, den es auf der Diskette findet.

Achtung: Die beiden vorgeschlagenen Lösungen gelten nur auf der Diskette, von der GEOS gebootet wird. Da es ein gewisses Risiko ist, irgend etwas auf der Originaldis-

kette zu ändern, sollten Sie diese Änderungen auf einer Kopie vornehmen und diese mit den Programmen aus Kapitel 4 bootfähig machen.

Tip 15: Gleich das richtige Eingabegerät

Dasselbe gilt übrigens auch für die sogenannten INPUT DRIVERS. Wenn Sie mit einem anderen Eingabegerät als dem Joystick arbeiten wollen, setzen Sie ihn einfach vor die anderen INPUT DRIVERS (gilt wieder nur für die Diskette, von der GEOS bootet).

Tip 16: Richtige Reihenfolge erspart Blättern

Nicht nur für die Wahl der Printer und INPUT DRIVER ist die Reihenfolge der Files (Icons) eine wichtige Sache. Sie können sich auch eine Menge Arbeit und Blättern sparen, wenn Sie Ihre Diskette aufräumen, also eine ganz gezielte Reihenfolge herstellen. Wenn Sie beispielsweise ein Diskette mit Bildern von GEOPAINT haben und auf dieser sind mehrere Bilder, so setzen Sie die, an denen Sie gerade arbeiten, auf die erste Seite. Dadurch brauchen Sie zur weiteren Arbeit gar nicht erst zu blättern, Sie können sie direkt doppelklicken und mit der Arbeit beginnen.

Tip 17: Icons auf eine andere Seite setzen

Um ein Icon auf irgendeine (beispielsweise die erste) Seite zu setzen, müssen Sie es auf der Border, also dem unteren Rand, absetzen. Dann schlagen Sie die gewünschte Seite auf und setzen es irgendwo auf dieser Seite wieder ab. Wenn dort noch Platz ist, wird es von GEOS dort abgelegt. Es ist nicht möglich, das Icon doppelzuklicken, dann die gewünschte Seite aufzuschlagen und es abzulegen. Beim Umblättern wird das Icon automatisch an seinem alten Platz abgelegt, so daß Sie damit nichts erreicht haben.

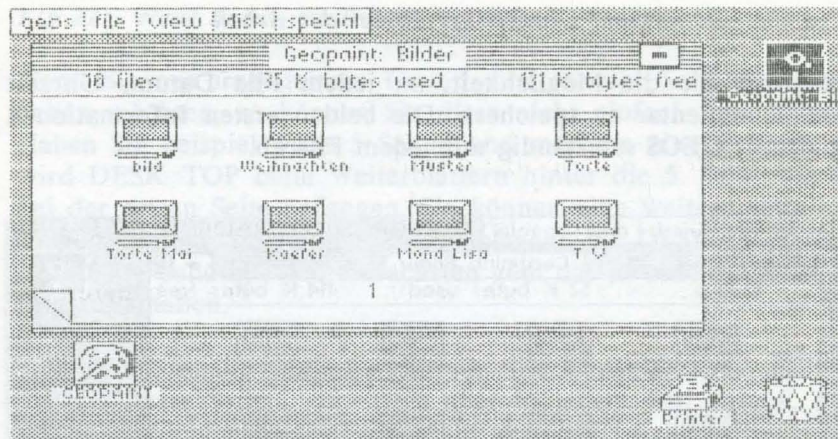


Abb. 2: Alle Bilder auf Seite 1

Tip 18: System-Files auf den letzten Seiten ablegen

Am sinnvollsten ist es, die FONTS, INPUT DRIVERS, PRINTERS, NOTES und PREFERENCES auf den hintersten Seiten abzulegen, da man auf diese Files nicht direkt zugreifen kann. Man kann also nicht etwa NOTES doppelklicken, um das NOTE PAD zu laden. Die Programme, die diese System-Files benötigen, finden Sie auch auf den hinteren Seiten.

Tip 19: Nicht zu lange Namen wählen

Geben Sie den Icons (also auch den von Ihnen erstellten Grafiken und Texten) nicht zu lange Namen. Diese werden von DESK TOP zentriert unter dem Icon angezeigt. Wenn Sie mehr als ca. 13 Zeichen benutzen (im Einzelfall hängt das wegen der Proportionalchrift von der Breite der Zeichen ab), sind diese nicht mehr ganz lesbar. Sie sollten lieber Datum, Uhrzeit und den Kommentar im INFO-Sektor benutzen, um Informationen zu den einzelnen Files festzuhalten.

Tip 20: Datum und Uhrzeit helfen bei der Arbeit

Benutzen Sie die Möglichkeit, zu jedem File Datum, Uhrzeit und Kommentar zu speichern. Die beiden ersten Informationen speichert GEOS selbständig mit jedem File ab.

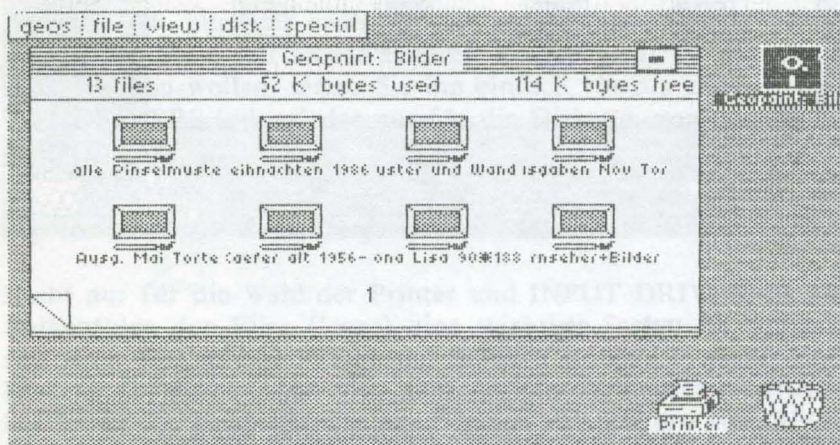


Abb. 3: Zu lange Namen der Files

Stellen Sie Datum und Uhrzeit also beim Beginn Ihrer Arbeit mit GEOS ein. Sie können sich dann ohne Schwierigkeiten die neueste Version Ihrer Grafik zeigen lassen, indem Sie unter VIEW "by DATE" wählen. Das zuletzt erstellte File steht jeweils ganz oben.

Tip 21: Kommentare im INFO-Fenster

Sie können sich viel Verwirrung ersparen, wenn Sie die Dokumente mit einem Kommentar versehen. Anstatt also den 4. Versuch eines Briefes an Peter vom 18.12.86, in dem Sie diesem frohe Weihnachten wünschen, "PETER18.12Weihnach4" zu nennen, nenne Sie den Brief einfach "Peter4", das Datum haben Sie ja wohl sowieso eingestellt, und im Kommentar (INFO-Sektor) schreiben Sie "Glückwünsche zu Weihnachten!".

Tip 22: Neue Seiten im Inhaltsverzeichnis erzeugen

Wenn Sie aus irgendeinem Grund eine weitere Seite für Icons haben möchten, so können Sie diese nicht einfach aufschlagen. Haben Sie beispielsweise 5 Seiten und möchten eine 6. Seite, so wird DESK TOP beim Weiterblättern hinter die 5. Seite wieder bei der ersten Seite anfangen. Sie können eine weitere Seite erzeugen, indem Sie solange Files ablegen, bis DESK TOP eine neue Seite benötigt. Am einfachsten geht das in unserem Beispiel folgendermaßen.

Kopieren Sie ein möglichst kurzes File (beispielsweise GEOS) auf die letzte Seite. Achten Sie aber in jedem Fall darauf, daß der Schreibschutz dieses Files nicht aktiviert ist. Wenn Sie nicht mehr wissen, wie das geht, so schauen Sie noch einmal bei Tip 17 nach. Vervielfältigen Sie dieses File (zum Beispiel GEOS) so oft, bis das neue File nicht mehr auf die 5. Seite paßt. DESK TOP erstellt dann automatisch eine neue Seite, auch wenn auf den vorherigen Seiten 1-4 noch Platz gewesen wäre.

Beim Vervielfältigen (FILE, DUPLICATE) müssen Sie aber jeweils neue Namen wählen. Hängen Sie einfach eine fortlaufende Nummer an. Nachdem die neue 6. Seite erstellt wurde, können Sie die überflüssigen Files auf der fünften Seite löschen (Sie haben doch hoffentlich ein File verwendet, das löscherbar war, sonst müssen Sie jetzt bei jedem einzelnen den Schreibschutz im INFO-Fenster entfernen. Sie merken doch hoffentlich, wie wertvoll unsere rechtzeitig gegebenen Tips und Warnungen sind?) Wenn Sie jetzt am Erstellen leerer Seiten so viel Freude haben sollten, daß Sie noch viele weitere erstellen möchten, so bedenken Sie bitte, daß es nicht so einfach möglich ist, leere Seiten wieder zu entfernen.

Tip 23: Mit DUPLICATE Vorlagen für verschiedene Texte oder Grafiken erstellen

Mit dem Befehl DUPLICATE können Sie sich auch bei Texten und Bildern viel Arbeit sparen. Angenommen, Sie haben eine Grafik "Landschaft", die Ihnen so gut gefällt, daß Sie diese in zwei verschiedene Richtungen weiterentwickeln möchten. Ver-

lassen Sie GEOPAINT, und duplizieren Sie die Grafik (neuer Name: "Landschaft2"). Die erste benennen Sie in "Landschaft1" um (RENAME). Nun können Sie mit "Landschaft1" in der einen Richtung weiterexperimentieren und mit "Landschaft2" in einer anderen. Auf diese Weise können Sie auch leicht eine Kopie herstellen, wenn Sie eine größere Änderung an einem Dokument ausprobieren wollen, aber den vorherigen Zustand nicht verlieren möchten (Sicher ist sicher).

Tip 24: Serienbriefe mit DUPLICATE

Mit dem Befehl DUPLICATE können Sie sogar problemlos so etwas wie "Serienbriefe" erstellen. Damit meint man in der Computerfachsprache Briefe, die sich beispielsweise nur in der Anrede und der Anschrift unterscheiden. Erstellen Sie einen Brief als Vorlage (beispielsweise eine "Einladung"), und lassen Sie die Stellen für Anrede, Vorname, Nachname usw. frei. Dann vervielfältigen Sie den Brief so oft, wie Sie ihn benötigen. Anschließend brauchen Sie mit GEOWRITE nur in jeden Brief die gewünschten Begriffe einzutragen. Wenn Sie den Texten beim Vervielfältigen auch noch entsprechende Namen geben (Einl. Müller, Einl. Meier...), sparen Sie nicht nur viel Arbeit, Sie wissen auch sofort, welche Einladung Sie gerade drucken.

Tip 25: Überflüssige Icons mit RESET entfernen

Wenn Sie Files von der Diskette A zur Diskette B kopieren, bleiben die Icons auf der Diskette A immer noch auf der BORDER, dem unteren Rand des Bildschirms. Wir sind selbst oft zu bequem, extra noch einmal "A" mit OPEN anzuwählen und die Icons zurückzusetzen. Wenn Sie aber beispielsweise einen Text auf mehreren Disketten suchen, diese jeweils einlegen und mit OPEN öffnen, und auf jeder sind mehrere Icons auf dem unteren Rand, so ist nach kurzer Zeit Ihre BORDER voll. Sie können diese Icons nicht einfach im DESK TOP Fenster absetzen, weil DESK TOP das als Aufforderung zum Kopieren mißversteht. Es gibt eine einfache Lösung.

Um alle Icons von vorherigen Disketten von der BORDER zu entfernen, wählen Sie einfach RESET unter SPECIAL. Nur die Icons, die auf der gerade eingelegten Diskette auf die BORDER gesetzt wurden, also zu dieser Diskette gehören, bleiben dort bestehen. Alle anderen verschwinden.

Tip 26: GEOS-Format V1.0 ist ohne Risiko

Wenn Sie eine Diskette mit GEOS benutzen wollen, die nicht mit GEOS erstellt (formatiert wurde), so fragt GEOS nach, ob die Diskette ins GEOS-Format umgewandelt werden soll. Vielleicht haben Sie sich an der Stelle schon geärgert, weil Sie einerseits Datenverlust befürchten (Format hört sich ziemlich gefährlich an), andererseits von einer nicht ins GEOS-Format geänderten Diskette keine Files kopieren können. GEOS bemängelt nämlich bei jedem Versuch, auf einer solchen Diskette ein Icon auf dem Rand abzulegen, daß dies nur bei GEOS-Disketten möglich ist.

Sie können selbsterstellte Disketten ohne Sorge ins GEOS-Format umwandeln lassen. GEOS benötigt einen zusätzlichen Sektor der Diskette, um dort Informationen für die BORDER ablegen zu können. Normalerweise wird dazu Track 19, Sektor 8 benutzt. Sollte dieser Sektor aber belegt sein, so sucht GEOS einen freien Sektor. Dadurch gehen also keine Daten verloren. Sie sollten diesen Vorgang allerdings nicht bei Disketten anwenden, die Sie mit Programmen gekauft haben. Diese Disketten haben oft einen Kopierschutz, der durch die Benutzung mit GEOS zerstört werden kann.

Tip 27: Swap-File - Wenn GEOS den leeren Platz nicht füllt

Vielleicht wissen Sie schon, daß GEOS bei jedem von Ihnen neu erstellten File (Text, Grafik, Notes) von vorne beginnend nach einer freien Stelle für das zugehörige Icon sucht. Eigenartigerweise scheint das nicht immer zu funktionieren. Wenn Sie beispielsweise auf der ersten Seite einen Platz frei haben und mit NOTE PAD ein Notizbuch NOTES neu erstellen, erscheint die-

ses nicht etwa auf der ersten Seite, sondern erst an einer späteren Stelle. Wie ist das möglich?

Bevor GEOS ein Accessory lädt, wird der Teil des Rechner-speichers, der verändert würde, auf die Diskette gerettet. Dafür benutzt GEOS einen eigenen Namen: "Swap-File". Für dieses "Austausch-File", das auf der Diskette abgelegt wird, wird natürlich von der ersten Seite an ein freier Platz gesucht und belegt. Wenn dann anschließend das Accessory (beispielsweise NOTE PAD) geladen wird und selbst ein File (NOTES) erzeugen will, ist dieser Platz natürlich belegt, und es wird ein weiterer freier Platz gesucht.

Wenn Sie das Accessory verlassen, wird der ursprüngliche Speicher des C64 wiederhergestellt. Dazu wird das Swap-File an die richtige Stelle geladen und anschließend gelöscht. Nun taucht im DESK TOP auf der ersten Seite eine leere Stelle auf. Dadurch bekommen Sie ein Swap-File normalerweise nie zu Gesicht.

Wenn Sie sich so ein geheimnisvolles File einmal anschauen wollen, so laden Sie ein Accessory und schalten den C64 einfach aus (Natürlich nicht bei der Bearbeitung wichtiger Daten und nicht während des Ladens oder Speicherns). Wenn Sie nun GEOS erneut Booten, können Sie zum erstenmal dieses geheimnisvolle Swap-File anschauen. Es hat stets dasselbe Icon wie das zugehörige Accessory. Mit diesem Swap-File können Sie alles machen: umbenennen, kopieren, duplizieren. Nur eines sollten Sie damit nicht machen: laden. Am besten werfen Sie es also nach dem Anschauen in den Papierkorb (Dieser Trick funktioniert nicht bei GEOS V1.3, da das DESK TOP automatisch das Swap-File löscht).

Tip 28: Wenn Accessories einfach nicht geladen werden

Diese Swap-Files sind auch der Grund dafür, daß genügend Platz auf der Diskette sein muß, um ein Accessory benutzen zu können. Ist dieser Platz nicht mehr vorhanden, kann GEOS nichts auf die Diskette retten und gibt (meistens) eine entsprechende Fehlermeldung aus. Sollte GEOS nach einigen Sekunden des Ladens einfach wieder ohne Fehlermeldung zum DESK TOP

zurückkehren, so haben Sie mit Sicherheit zu wenig freien Speicher auf der Diskette. Manchmal werden nämlich keine Fehlermeldungen ausgegeben, besonders bei Programmen, die selbst etwas speichern (z.B. PREFERENCE MANAGER: Preferences).

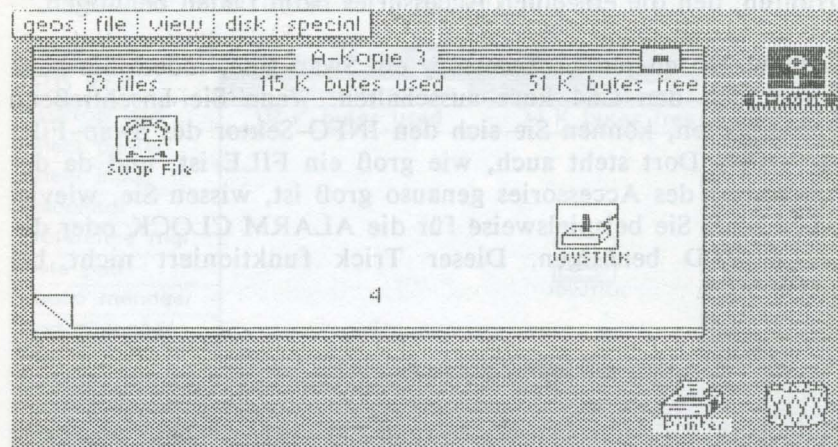


Abb. 4: Ein SWAP-File

Tip 29: Wieviel Platz benötigt ein Accessory

Den jeweiligen Speicherplatz herauszubekommen, den die einzelnen Accessories benötigen, ist nicht ganz einfach. Sie können nicht einfach INFO wählen. Dort wird zwar die Größe des Programms angezeigt, aber nicht der Platzbedarf. Woher weiß GEOS, wie groß ein Programm ist, wieviel Platz auf der Diskette frei ist und wieviel beim Laden eines Accessories als Swap-File auf der Diskette gespeichert werden muß? Die Größe eines Files, die im INFO-Sektor in KBytes angezeigt wird, holt sich GEOS aus dem Directory-Eintrag. Dort enthalten Byte 28 und 29 die Anzahl der Blocks. GEOS teilt diese Zahl durch vier und kann dann den Wert in KBytes ausgeben.

Die Anzahl der freien Blocks auf der Diskette holt sich GEOS aus der BAM. Die Größe des Swap-Files beim Laden eines Accessories erfährt GEOS hingegen aus dem INFO-Sektor. Dort geben die Bytes 71,72 die Ladeadresse und die Bytes 73,74 die Endadresse. Aus der Differenz berechnet GEOS den Platzbedarf. Mit einem einfachen Trick können Sie aber trotzdem den Platz erfahren, den die einzelnen Accessories beim Laden benötigen.

Erstellen Sie zu jedem Accessory ein Swap-File, indem Sie nach dem Laden den C64 kurz ausschalten. Wenn Sie anschließend GEOS booten, können Sie sich den INFO-Sektor des Swap-Files anschauen. Dort steht auch, wie groß ein FILE ist, und da der Platzbedarf des Accessories genauso groß ist, wissen Sie, wieviel freien Platz Sie beispielsweise für die ALARM CLOCK oder das NOTE PAD benötigen. Dieser Trick funktioniert nicht bei GEOS V1.3.

Tip 30: Blättern mit der Tastatur

Einiges an Zeit und Arbeit können Sie sparen, wenn Sie nicht mit Hilfe des "Eselohrs" die verschiedenen Seiten durchblättern, sondern einfach die zugehörige Zahl drücken. Wir meinen, daß die Benutzung der Tastatur oft wesentlich zur Vereinfachung von Arbeitsvorgängen beiträgt. Auch wenn eine Benutzeroberfläche mit "Maus" eine tolle Sache ist. Ab GEOS V1.3 gibt es übrigens noch mehr Tastenkombinationen, beispielsweise "C"+"o" für das Öffnen von Disketten.

Tip 31: CLOSE können Sie sich sparen

Wenn Sie die Diskette wie im Handbuch beschrieben wechseln, müssen Sie erst CLOSE und dann OPEN anklicken. Das CLOSE können Sie sich sparen. DESK TOP akzeptiert auch, wenn Sie nur OPEN anklicken. Dadurch sparen Sie einiges an Zeit und Mühe, weil nach CLOSE erst wieder das Pull-Down-Menü verschwindet und Sie DISK und dann OPEN klicken müssen. Genausogut können Sie das dunkle Diskettensymbol in der rechten oberen Ecke anklicken. Manchmal ist der Weg dahin einfach kürzer.

Tip 32: Alle Accessories unter GEOS

Sie brauchen nicht Seite für Seite durchzublättern, um das gewünschte Accessory zu finden. Klicken Sie einfach GEOS an, und DESK TOP zeigt Ihnen alle vorhandenen Accessories. Sie können das gewünschte dann direkt anklicken.

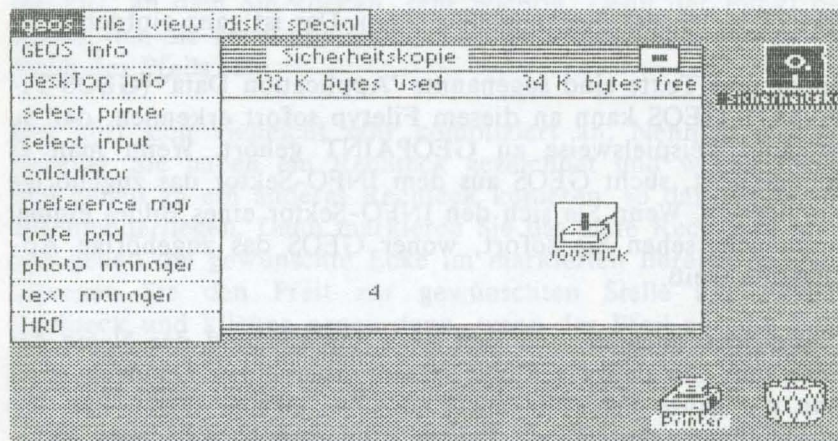


Abb. 5: Alle Accessories unter "geos"

Tip 33: Arbeitskopien von einer Vorlage

Viel Arbeit können Sie sich aber in jedem Fall sparen, wenn Sie einmal eine Arbeitsdiskette anfertigen und diese dann als Vorlage für weitere Arbeitsdisketten benutzen. Diese Diskette benutzen Sie dann nicht als Arbeitsdiskette, sondern legen sie beiseite. Wenn Sie nun eine Arbeitsdiskette benötigen, fertigen Sie mit BACKUP eine Kopie dieser Vorlage an. Für eine GEOWRITE-Arbeitsdiskette brauchen Sie dann beispielsweise nur noch GEOPAINT zu löschen, um genügend Platz für viele Bilder zu haben.

2.3 GEOPAINT

Tip 34: Bilder und Texte doppelklicken

Viel Zeit und Arbeit können Sie sparen, wenn Sie nicht erst das Application (beispielsweise GEOPAINT) laden und anschließend umständlich das gewünschte Bild wählen, sondern mit GEOPAINT gleich das passende Bild laden. Das ist ganz einfach.

Bilder und Texte sind sogenannte 'Application Data' (GEOS Filetyp=7). GEOS kann an diesem Filetyp sofort erkennen, daß so ein Bild beispielsweise zu GEOPAINT gehört. Wenn man es doppelklickt, sucht GEOS aus dem INFO-Sektor das zugehörige Application. Wenn Sie sich den INFO-Sektor eines Bildes einmal anschauen, sehen Sie sofort, woher GEOS das zugehörige Application weiß.

Dieses wird geladen, und ihm wird anschließend der Name des Files, das Sie doppelgeklickt haben, übergeben. Dadurch spart man die ganze Prozedur mit CREATE, OPEN, QUIT und das anschließende Auswählen. Leider kann man NOTES oder PREFERENCES nicht doppelklicken, um das zugehörige Accessory zu laden. Diese Files haben nämlich den Filetyp 4 (= System-File), und bei diesem GEOS-Filetyp läßt GEOS praktisch keine Aktionen zu (Kein Doppelklick, kein DUPLICATE, kein Umbenennen). Übrigens gibt es einen interessanten Effekt, wenn Sie ein Text- oder Photo-Album doppelklicken. Lassen Sie sich von der anschließenden Fehlermeldung

"PLEASE INSERT A DISK WITH THE PHOTO MANAGER"

nicht verwirren. Natürlich befindet sich der Photo Manager auf der Diskette. Aber das Photo Album ist mit Filetyp 7 ein Application Data. Daher sucht GEOS nach dem Application Photo Manager, und da dieser ein Accessory ist, findet GEOS ihn in so einem Fall nicht (Was man alles so entdeckt, wenn man lange mit einem Programm arbeitet!).

Tip 35: COPY bei GEOPAINT punktgenau

Wenn Sie in GEOPAINT einen Ausschnitt kopieren oder verschieben wollen, so markieren Sie ihn mit dem Ausschnittmarkierer und wählen dann in der Statuszeile unten rechts COPY oder MOVE. Danach müssen Sie innerhalb des markierten Bereichs klicken, um schieben oder kopieren zu können. Dabei ist der Ort, an dem Sie klicken, sehr wichtig. Denn der Punkt des Bildes, den Sie dabei anklicken, wird auch am Ziel genau an der Stelle des Pfeils sein.

Das hört sich vielleicht sehr kompliziert an. Nehmen wir ein Beispiel: Sie haben ein Rechteck gezeichnet und wollen dieses nun genau an ein anderes Rechteck kopieren, so daß die Ecken aufeinanderliegen. Dann markieren Sie das erste Rechteck, klicken genau die gewünschte Ecke im markierten Bereich an. Nun bewegen Sie den Pfeil zur gewünschten Stelle am zweiten Rechteck und klicken genau dann, wenn der Pfeil auf die Ecke zeigt. Das Rechteck 1 wird dann so kopiert, daß die vorher angeklickte Stelle genau an der Pfeilspitze beginnt.

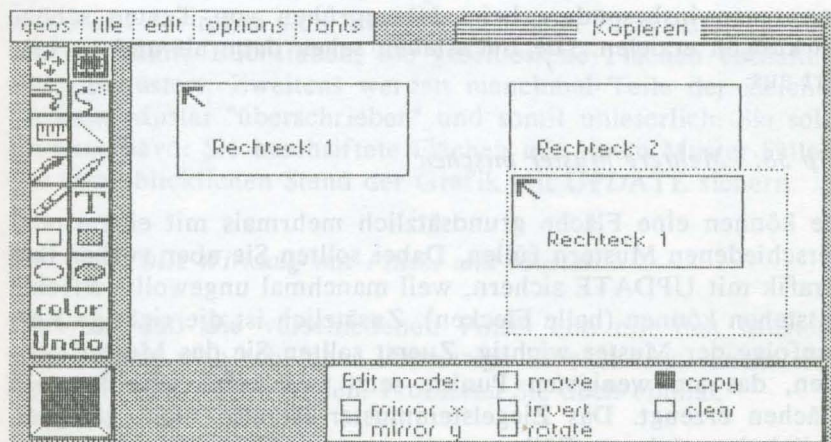


Abb. 6: Punktgenaues Kopieren in GEOPAINT

Tip 36: Schnelle Bewegung mit dem Seitenzeiger

Wenn Sie von der oberen Ecke eines Blattes schnell zur unteren Ecke wollen, so ist das SCROLLEN mit den Scroll-Pfeilen eine mühselige und zeitraubende Sache. Unten auf dem Bildschirm sehen Sie den Seitenzeiger, ein symbolisches Blatt, das Ihre ganze Seite darstellt. Darauf ist ein kleines Rechteck, das die Lage des GEOPAINT-Fensters auf dem Blatt anzeigt. Klicken Sie einfach das kleine Rechteck an, und setzen Sie es ganz nach unten. Sofort erscheint der untere Teil des Blattes im Fenster und kann bearbeitet werden.

Tip 37: Auch das leere Muster in der Musterleiste kann man benutzen

In der Musterleiste ist das erste Muster links völlig leer. Wenn Sie mit diesem eine Fläche füllen (Wasserhahn), in der schon etwas gezeichnet ist, so passiert nichts. Wenn Sie aber die Sprühdose mit diesem Muster verwenden, so wird der übersprühte Bereich "verwischt", das heißt, es werden immer mehr Punkte entfernt, bis die Fläche schließlich sogar ganz leer wird. Damit kann man insbesondere beim Übersprühen von Texten schöne Wirkungen erzielen. Die Buchstaben sehen dann alt und verwittert aus.

Tip 38: Mehrere Muster mischen

Sie können eine Fläche grundsätzlich mehrmals mit einem oder verschiedenen Mustern füllen. Dabei sollten Sie aber vorher Ihre Grafik mit UPDATE sichern, weil manchmal ungewollte Effekte entstehen können (helle Flecken). Zusätzlich ist die richtige Reihenfolge der Muster wichtig. Zuerst sollten Sie das Muster nehmen, das am wenigsten Punkte setzt *und* keine geschlossenen Flächen erzeugt. Das Ziegelsteinmuster ist also nicht geeignet, weil beim nächsten Füllvorgang nur jeweils ein einzelner Ziegelstein gefüllt wird. Übrigens hat das gefüllte Muster (2. oben links) ähnliche Eigenschaften wie das leere Muster daneben. Wenn eine Fläche mit diesem Muster gefüllt ist, kann man fast jedes andere Muster problemlos dazufügen.



Abb. 7: Mit dem leeren Muster "verwischen"

Tip 39: Probleme beim Mischen von Schrift und Muster

Manchmal vertragen sich Schrift und Muster nicht. Erstens werden bestimmte Buchstaben, die geschlossene Flächen enthalten, nicht gemustert. Zweitens werden manchmal Teile der Zeichen mit dem Muster "überschrieben" und somit unleserlich. Sie sollten also, bevor Sie beschriftete Flächen mit einem Muster füllen, den augenblicklichen Stand der Grafik mit UPDATE sichern.

Tip 40: Tolle Wirkung mit Pinsel und Muster

Dadurch, daß die verschiedenen Pinsel alle mit den Mustern malen, kann man scheinbar schwierige Effekte "mit einem einfachen Pinselstrich" erzielen. Probieren Sie doch einmal:

- den kleinen Pinselpunkt oder einen schmalen Strichpinsel zusammen mit einem gepunkteten Muster. Dadurch erzielt man verschieden gestrichelte Linien.
- die langen gepunkteten Pinsel. Diese ergeben fast mit jedem Muster einen schönen Effekt.

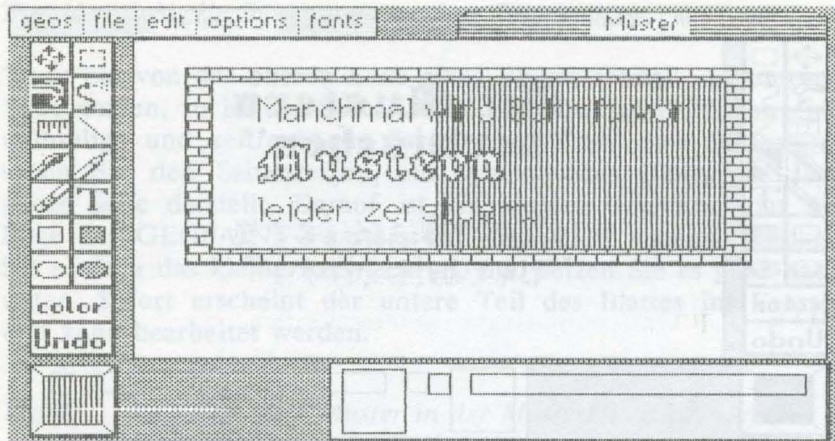


Abb. 8: Manchmal zerstören Muster eine Schrift

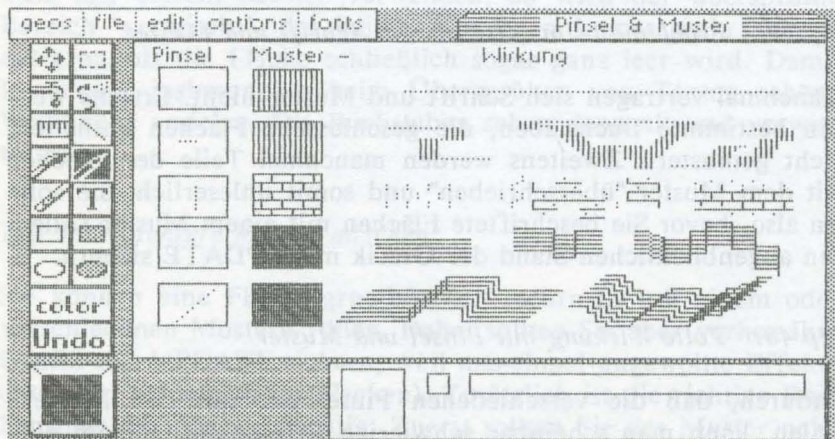


Abb. 9: Pinsel und Muster kombinieren

Tip 41: Doppelklicken in der Werkzeugleiste

Viele Funktionen der Werkzeugleiste haben eine Doppelfunktion. Wenn dieses Werkzeug schon aktiviert ist (einmal angeklickt und dunkel) und man klickt es zweimal schnell an, so ergeben folgende Werkzeuge noch eine weitere Funktion:

Pinself

CHANGE BRUSH - einen der 32 Pinsel auswählen

Bleistift

Umschalten zwischen PIXEL EDIT und NORMAL EDIT

Radiergummi

Löschen der gesamten Grafik im Fenster (also nicht des gesamten Blattes).

Tip 42: Mehr Werkzeuge im Pixeledit

Im Modus PIXEL EDIT können mehr Werkzeuge als im Handbuch erwähnt benutzt werden. Einige haben aber abweichende Funktionen.

Scroll-Pfeile

Umschalten in "Lupe setzen". Das Anklicken der Scroll-Pfeile ist der schnellste Weg, um schnell die Lupe auf eine andere Stelle zu setzen.

Wasserhahn

Füllt Flächen wie in NORMAL EDIT. Das Füllen endet an den Grenzen des PIXEL-EDIT-Fensters.

Sprühdose

Sprühfunktion; Es wird aber nicht in der Größe der angezeigten Sprühdose geändert, sondern in einem viel größeren Bereich. Die Sprühdose wirkt also weiterhin "im Modus NORMAL EDIT". Die Lupe zeigt die Auswirkung dadurch vergrößert.

Lineal

Normale Funktion

Gerade

Normale Funktion

Pinsel

Malt mit dem ausgesuchten Pinsel im gewählten Muster. Hier gilt dasselbe wie bei der Sprühdose. Das Ergebnis ist viel größer als der Pinsel.

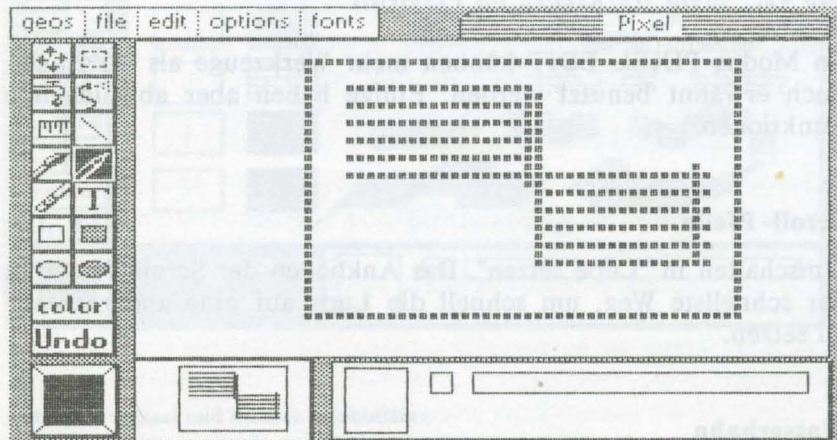


Abb. 10: So groß malt der Pinsel auch in PIXEL EDIT

Radiergummi

Normale LösCHFunktion

Bleistift

Setzt und löscht einzelne Punkte. In PIXEL EDIT hat er also drei Funktionen (keine Wirkung - das heißt nur Mausbewegung - setzen, löschen). Klickt man an einer leeren Stelle, wird "Punkte setzen" eingeschaltet, klickt man auf einem gesetzten Punkt, wird "Punkte löschen" aktiviert.

Rechteck

Normale Funktion

Gefülltes Rechteck

Normale Funktion

UNDO

Normale Funktion

Tip 43: UNDO sofort aktivieren

Mit der UNDO-Funktion können Sie den letzten Vorgang rückgängig machen. Dabei sollten Sie aber zwischen dem Vorgang, den Sie rückgängig machen wollen, und dem Anklicken von UNDO nichts anderes machen. Klicken Sie also vor UNDO kein Pull-Down-Menü an. UNDO ist nur jeweils einmal möglich. Man kann also nicht zwei Vorgänge rückgängig machen, und übrigens: UNDO UNDO ist auch nicht möglich.

Tip 44: Wie kann man die Muster- oder Pinselleiste wieder verlassen

Wenn Sie CHANGE BRUSH oder Musterwahl geklickt und es sich anders überlegt haben, so können Sie das Auswahlfeld nicht einfach verlassen. Bei einer Musterwahl können Sie einfach das Musterquadrat links unten anklicken, bei CHANGE BRUSH müssen Sie den schon eingestellten Pinsel anklicken, um das Feld verlassen zu können.

Tip 45: Objekte auf dem gesamten Papier verschieben

Leider können Sie keine Teile Ihrer Grafik einfach mit dem Ausschnittmarkierer umrahmen und aus dem GEOPAINT-Fenster hinausbewegen, um sie auf einer anderen Stelle des Blattes abzulegen. Wenn Sie nämlich den Bereich markiert haben und die Scroll-Pfeile anklicken, verschwindet die Markierung sofort. Trotzdem kann man Teile der Grafik frei auf dem gesamten Blatt bewegen. Dazu müssen sie nur im PHOTOSCRAP abgelegt werden (CUT, COPY). Anschließend bewegen Sie das GEOPAINT-Fenster an die gewünschte Stelle des Blattes und kleben den Ausschnitt dort wieder ein. Dazu wählen Sie am Ziel einen Ausschnitt, und ein anschließendes PASTE klebt dort den gespeicherten Ausschnitt ein.

Tip 46: Grafik-Bibliothek durch PHOTOSCRAP und Photo-Album

Mit dem PHOTOSCRAP können Sie sich viel Arbeit sparen, indem Sie häufig benutzte Objekte dort ablegen. Angenommen, Sie erstellen Ihre erste elektronische Schaltung. Jedes neu gezeichnete Bauelement kopieren Sie nun zu den anderen ins PHOTOSCRAP. Dabei dürfen Sie aber nicht einfach das Bauteil mit COPY herauskopieren, weil sonst der bestehende Inhalt des PHOTOSCRAP gelöscht würde.

Holen Sie den Inhalt des PHOTOSCRAP mit PASTE auf eine leere Stelle Ihres Blattes. Markieren Sie dann das Bauteil, und kopieren Sie es mit dem COPY-Befehl aus dem Status-Bereich

an diese Stelle. Nun können Sie Ihre um ein Bauteil erweiterte Bibliothek mit COPY ins PHOTOSCRAP kopieren. Am Ende Ihrer Arbeit kleben Sie alles in ein Foto-Album.

Tip 47: Achtung bei UNDO nach CUT, COPY und PASTE

Bei diesen drei Befehlen wirkt die UNDO-Funktion nicht, sie zerstört sogar mehr, als sie rettet. Zwar wird nach CUT durch UNDO der herausgeschnittene Teil der Grafik wiederhergestellt, aber das PHOTOSCRAP ist natürlich verändert. Wenn Sie nach PASTE die UNDO-Funktion anklicken, verschwindet das Eingeklebte nicht. Jetzt bleibt zusätzlich auch noch der gepunktete Rahmen bestehen. Wenn allerdings ein Pull-Down-Menü über diesen Rahmen klappt, verschwindet dieser nach dem Hochklappen an der Stelle.

2.3 Direktstart von Applications

Sind Sie ein Fan von GEOPAINT? Dann stört es Sie sicherlich, daß das Programm erst vom DESK TOP aus geladen werden kann. Schöner wäre es doch, wenn man eine Diskette erstellen könnte, bei der durch die Eingabe von 'LOAD "GEOS",8,1' sofort dieses Zeichenprogramm geladen und gestartet würde.

Da sich das DESK TOP normalerweise beim Ladevorgang der Applications selber überschreibt, ist es nicht möglich, daß diese anschließend auf irgendwelche Routinen des DESK TOP zurückgreifen. Somit ist es prinzipiell möglich, daß Applications völlig losgelöst vom DESK TOP funktionieren. Natürlich ließe sich auch eine Diskette realisieren, die ohne den Umweg über das DESK TOP sofort GEOWRITE installiert.

Verantwortlich dafür, welches Programm beim Booten von GEOS gestartet wird, ist eine Stelle im GEOS KERNAL. Hier steht in einem bestimmten Bereich der Filename dieses Programms. Im Normalfall steht hier also 'DESK TOP'.

Jetzt könnte man natürlich auf die Idee kommen, an dieser Stelle einen anderen Namen einzusetzen. Zufällig stimmt sogar

die Wortlänge von GEOPAINT und GEOWRITE mit der des DESK TOP überein, so daß es auch keine Platzprobleme gibt. Wenn Sie möchten, können Sie sich einmal davon überzeugen, daß dies tatsächlich funktioniert.

Spätestens beim Anklicken von 'QUIT' werden Sie aber den entscheidenden Nachteil dieser Methode feststellen: Das gewohnte Bild des DESK TOP wird nicht mehr erscheinen, da der hierfür notwendige Ladevorgang der gleiche ist wie bei der Neuinitialisierung von GEOS. Da jedoch an den entscheidenden Stellen im KERNAL nicht mehr der Name 'DESK TOP' steht, wird wieder das von Ihnen dort eingetragene Programm geladen. Die Anwendung kann also nicht mehr verlassen werden. Wenn Sie nun vermuten, wir hätten eine zweite Methode gefunden, um einen Sofortstart von Applications zu veranlassen, können wir Ihnen gratulieren: Sie haben Recht!

Richtige Tips und Tricks zeichnen sich auch dadurch aus, daß man sie relativ schnell und unkompliziert durchführen kann. Dafür ist der folgende Tip ein sehr gutes Beispiel. Uns ist eine Idee gekommen, wie man das GEOS KERNAL hinters Licht führen kann. Es ist nämlich nichts einfacher, als einem Application vom DESK TOP aus mittels RENAME einen anderen Namen zu geben. Die Idee besteht nun einfach darin, z.B. GEOPAINT umzubenennen in DESK TOP. Dies ist jedoch nur möglich, wenn sich das richtige DESK TOP nicht auch auf dieser Diskette befindet.

Wenn Sie also auf einer bootbaren GEOS-Kopie das DESK TOP auf den Papierkorb bringen und anschließend GEOPAINT umbenennen in DESK TOP, haben Sie eine Diskette, bei der zuerst das Malprogramm geladen wird. Dies funktioniert selbstverständlich auch bei GEOWRITE. Natürlich wird Ihnen beim Anklicken von 'QUIT' das gleiche widerfahren wie bei der ersten Möglichkeit. Da das KERNAL anstelle des richtigen DESK TOP wieder die Application lädt (die ja DESK TOP heißt), kann das Programm auf diese Art nicht verlassen werden. Es geht jedoch trotzdem.

In den Menüs der Anwendungen findet man unter dem Punkt FILE außer QUIT auch das Feld CLOSE. Wenn man nun seine Arbeit durch CLOSE beendet, erscheint ein Menü, das als dritten Punkt die Wahl zuläßt: 'QUIT TO DESK TOP'. Bevor nun dieses Feld angeklickt wird, muß eine Diskette eingelegt werden, die das tatsächliche DESK TOP enthält. Es erscheint nun wieder das gewohnte DESK-TOP-Bild.

Zum Schluß möchten wir noch darauf hinweisen, daß es nicht möglich ist, andere Programme außer den Applications in 'DESK TOP' umzubenennen. Diese Anwendungen sind nämlich die einzigen Programme, die nach Beendigung der Arbeit ein Nachladen des DESK TOP veranlassen.

2.4 GEOWRITE

Tip 48: Rechter Rand auf 5 spart Zeit und schafft Übersicht

Leider sind in GEOWRITE der linke und rechte Rand unserer Meinung nach sehr ungünstig voreingestellt. Schnell und komfortabel können Sie schreiben, wenn Sie gleich zu Beginn Ihrer Arbeit den rechten Rand auf '5' setzen. Dazu fahren Sie mit dem Pfeil so weit wie möglich nach rechts. Am rechten Rand neben der '7' auf der Positionsleiste befindet sich ein eigenartiges Symbol - der Markierer für den rechten Rand. Wenn Sie ihn anklicken, erhalten Sie ein bewegliches 'M', das Sie einfach auf die '5' bewegen und dort abklicken. Ab jetzt braucht GEOS nicht mehr zwischen dem linken und rechten Teil des Blattes umzuschalten. Der erneute Bildschirmaufbau entfällt damit.

Lesen Sie übrigens ruhig einmal den Text der Abbildung. Die deutschen Umlaute wurden mit dem FONT-EDITOR aus diesem Buch erstellt. Das gibt den Texten den letzten Schliff. Übrigens ist auch der Markierer für den rechten Rand ein 'M'. Er sieht nur deshalb so eigenartig aus, weil die restlichen 7 Tabulatoren auch auf dieser Position sitzen. Wenn Sie alle 8 Tabs zwischen linkem und rechtem Rand gesetzt haben, ist der Randmarkierer auch ein einfaches 'M' (Das haben wir in der Abbildung 12 einmal eingestellt).

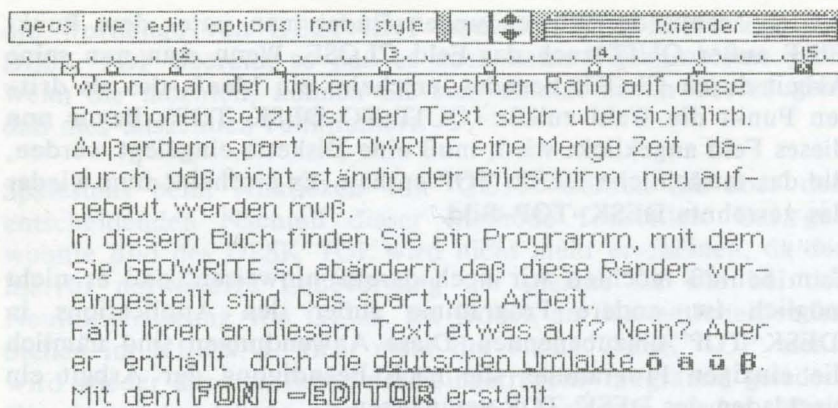


Abb. 11: Ränder gesetzt bei Geowrite

Tip 49: Schnelles Entfernen der Tabulatoren

Tabs werden gesetzt, indem einfach auf die gewünschte Stelle der Positionsleiste gezeigt und geklickt wird. Zum Verschieben klicken Sie den Tabulator an und bewegen ihn auf die neue Stelle. Dort klicken Sie ihn wieder ab. Zum Entfernen eines Tabs brauchen Sie ihn nur anzuklicken und nach oben zu verschieben. Wenn Sie nun noch einmal klicken, wird diese Position gelöscht.

Wenn Sie aber alle 8 gesetzten Tabulatoren auf diese Weise entfernen wollten, so hätten Sie viel zu tun. Einfacher geht es, indem Sie einfach kurzzeitig den rechten Rand so weit wie möglich nach links setzen. Alle Tabulatoren, die jetzt rechts vom rechten Rand wären, werden dadurch gelöscht, und Sie können den rechten Rand wieder auf die gewünschte Position setzen (In Wirklichkeit löscht GEOWRITE die Tabs gar nicht, sondern setzt sie einfach auf den rechten Rand. Wenn Sie diesen nun verschieben, bewegen sich ebenfalls alle "gelöschten" Tabulatoren. Die Wirkung ist wie ein Löschen).

Tip 50: BSW 9 Point spart Arbeit und Zeit

Schreiben Sie Ihren Text zuerst in BSW 9 Point. Zum einen ist das eine relativ kleine Schrift. Dadurch sehen Sie viel Text im GEOWRITE-Fenster und können dementsprechend viel bearbeiten und auf Rechtschreibung prüfen. GEOWRITE ist ein WYSIWYG-Programm (What you see is what you get = Sie sehen genau das, was später gedruckt wird). Das hat viele Vorteile, aber auch Nachteile. Was Sie nicht sehen, können Sie auch nicht markieren und damit auch nicht ausschneiden oder kopieren. Wenn Sie also bei einer großen Schrift nur 5 Zeilen auf dem Bildschirm sehen und 20 Zeilen verändern wollen, müssen Sie viel scrollen.

Zum anderen muß GEOWRITE für jede geänderte Schrift auf die Diskette zugreifen, um die Schrift anzeigen zu können. Wenn Sie also sofort beim Schreiben viele verschiedene Schriften benutzen, muß GEOWRITE diese für jeden Bildschirmaufbau kurz laden. Ganz extrem wird das, wenn Sie nach jedem Buchstaben eine andere Schrift benutzen. Dann können sie jedesmal, wenn GEOWRITE die Seite neu aufbaut, Kaffee trinken gehen.

Wenn Sie den Text geschrieben und korrigiert haben, markieren Sie alle gewünschten Stellen und ändern jeweils die Schrift. Für den Wechsel der Schriftarten (Bold, Underline ...) benötigt GEOWRITE keine Diskettenzugriffe. Diese Änderungen werden nicht geladen, sondern errechnet. Verschiedene Schriftarten können Sie also auch schon beim ersten Schreiben benutzen, um Ihren Text optisch aufzulockern.

Tip 51: Schriftwechsel: Besser von University 6 Point nach 24 Point als umgekehrt.

Wenn Sie größere Passagen des Textes von einer kleinen in eine große Schrift umändern möchten, so ist das ganz einfach. Meist sehen Sie den ganzen Text auf einmal im GEOWRITE-Fenster, können ihn markieren und die neue Schrift problemlos auswählen. Der umgekehrte Weg ist aber sehr problematisch. Sie sollten ihn daher vermeiden. Nehmen wir ein Beispiel.

Sie haben eine ganze Seite in University 24 Point und wollen diese in University 6 Point umwandeln. Dann markieren Sie soviel Text, wie Sie sehen können, und klicken die kleine Schrift an. Brav wandelt GEOWRITE den markierten Text um, aber daraus werden nur 1-2 Zeilen University 6 Point. Der restliche Text bleibt natürlich in der anderen Schrift, er war ja nicht markiert, und Sie müssen nun immer wieder Text markieren und in University 6 Point umwandeln. Das kann sehr lange dauern und sehr umständlich sein. Die richtige Reihenfolge spart Zeit und Nerven.

Tip 52: Textformatierung und das Einkleben von Bildern

In GEOWRITE-Texte können Sie mit GEOPAINT erstellte Grafiken einkleben. Leider gibt es dabei einige Einschränkungen:

- Eine eingeklebte Grafik sperrt die gesamte Breite. Links und rechts davon sind leider keine Texte möglich.
- Grafiken werden automatisch zentriert. Man kann sie nicht ganz nach links oder rechts setzen.

Wenn der Cursor mitten in einer Zeile steht und Sie kleben eine Grafik ein, fügt GEOWRITE selbständig ein RETURN ein und beginnt mit der Grafik in der nächsten Zeile.

Wenn Sie eine Grafik eingeklebt haben und den linken oder rechten Rand verschieben, verschiebt sich auch die Grafik (wegen der automatischen Zentrierung). Sie sollten daher die Grafik relativ früh einkleben, um den Text von vornherein so zu formatieren, daß Text und Grafik gut zueinander passen. Wählen Sie dann aber HIDE PICTURE unter OPTIONS, damit GEOWRITE die Grafik nicht jedesmal von der Diskette laden und zeichnen muß.

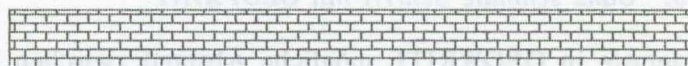
Tip 53: Bilder aus dem Text entfernen

Ein eingeklebtes Bild können Sie ganz einfach aus dem Text entfernen, weil im Text an der Stelle nur ein einzelnes Steuerzeichen steht, an dem GEOWRITE erkennt, daß hier ein Bild

eingeklebt wird. Wenn Sie dieses Steuerzeichen löschen, ist auch das Bild herausgeschnitten. Setzen Sie den Cursor auf die erste Position hinter dem angezeigten Bild (Wenn Sie ganz links einen riesigen Cursor blinken sehen, stimmt die Position noch nicht). Betätigen Sie dann einmal die DEL-Taste, und das Bild ist verschwunden. Dabei kommt es häufig vor, daß anschließend der Cursor ebenfalls verschwunden ist. Klicken Sie in so einem Fall einfach einmal, und er taucht wieder auf.



Nicht nur breite Graphiken belegen die gesamte Textbreite, wenn sie eingeklebt werden.



Auch sehr schmale Graphiken lassen keinen Platz mehr für Text links oder rechts der Graphik.



Deshalb sollte man die Textbreite von vornherein an die Breite der Graphik anpassen.

Abb. 12: Grafiken werden immer zentriert.

Tip 54: GEOWRITE speichert eingeklebte Bilder

Sie brauchen nicht zu befürchten, daß GEOWRITE ein eingeklebttes Bild verliert, wenn Sie den PHOTOSCRAP und das Photo-Album löschen. In dem Augenblick, in dem Sie es einkleben, speichert GEOWRITE es noch einmal selbständig ab und ist dann auf das Album nicht mehr angewiesen. Dadurch kann GEOWRITE aber nicht mehr 127 Seiten Text verwalten, sondern nur noch 64 Seiten Text und 63 eingeklebte Grafiken. Das sollte aber für die meisten Anwendungen völlig ausreichen.

Tip 55: Zwei Textseiten zusammenfügen

Genau so einfach, wie man eingeklebte Grafiken entfernen kann, kann man auch einen Seitenwechsel entfernen. Auch ein Seitenende besteht für GEOWRITE nur aus einem Steuerzeichen. Um dieses zu entfernen, positionieren Sie den Cursor auf der folgenden Seite obens links und drücken dann die DEL-Taste. GEOWRITE fragt nun nach, ob Sie das letzte Zeichen der vorherigen Seite löschen wollen (Delete the last character of the previous page?). Da das gerade das Steuerzeichen für den Seitenwechsel ist, können Sie unbesorgt "OK" anklicken, und die beiden Seiten sind wieder zusammengefügt.

Tip 56: Ganz schmale Schrift mit GEOPAINT

Der kleinste in GEOWRITE mögliche Abstand zwischen linkem und rechtem Rand ist '1', also beispielsweise von '2' - '3'. Schmalere Text ist auch kaum sinnvoll, weil dann fast kein Wordwrap mehr möglich ist. Wenn Sie doch einmal einen schmalen Textblock benötigen, so können Sie ihn in GEOPAINT problemlos schreiben und anschließend in GEOWRITE einkleben.

Tip 57: Schneiden Sie nicht zuviel Text auf einmal aus

Sie können nämlich später beim Einkleben im TEXTMANAGER einen Absturz bewirken. Normalerweise ist das kaum möglich, weil man ja nur soviel Text ausschneiden kann, wie im GEOWRITE-Fenster sichtbar ist. Wenn man aber alle Tricks benutzt (Schrift in University 6 Point, linken Rand auf '0', rechten Rand auf '7'), dann den gesamten Text markiert und später im TEXTMANAGER einklebt, gibt der auf und verabschiedet sich.

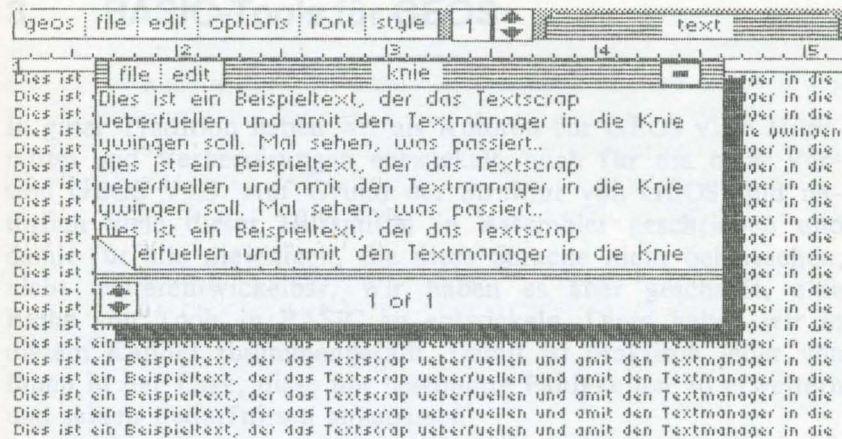


Abb. 13: TEXTSCRAP geht in die Knie

GEOS hat für viele Probleme ganz bestimmte Standards entwickelt. So werden beispielsweise die verschiedenen Fonts in einer Weise abgelegt, daß sowohl GEOPAINT als auch GEOWRITE (und auch unser Font-Editor) darauf zurückgreifen und neue erstellen können. Leider wurde aber häufig eine Kompatibilität zwischen den Files verschiedener Programme nicht vorgesehen. So ist es beispielsweise nicht möglich, mit NOTE PAD erstellte NOTES zu drucken. NOTES werden nämlich als System File abgelegt (Filetyp=04). Mit solch einem System File kann man als Benutzer aber praktisch nichts machen. Man kann es nicht duplizieren, umbenennen und auch nicht das Programm, das sie erstellt hat, direkt über ein Doppelklicken des System Files laden.

Mit dieser Gegebenheit haben wir uns nicht zufrieden gegeben. Deshalb haben wir ein Programm geschrieben, mit dem man die NOTES ausdrucken kann. Dabei muß dieses Programm zwei Probleme lösen:

- Es muß die GEOS-File im Format VLIR lesen.
- Es muß den GEOS-ASCII-Zeichensatz in CRM-ASCII umwandeln.

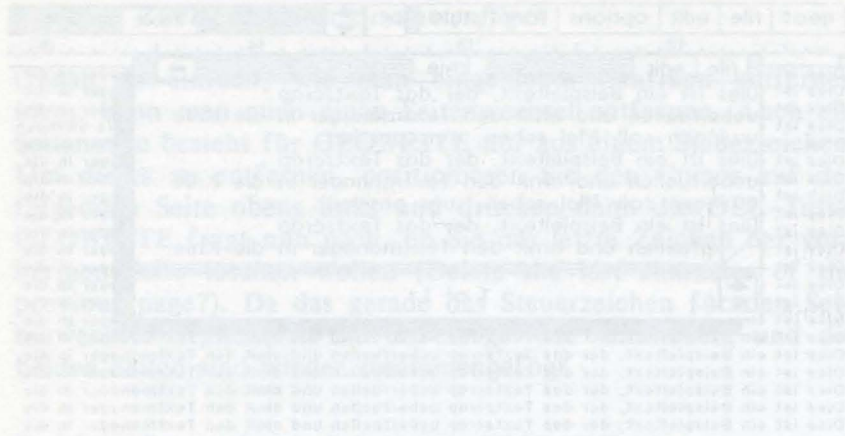


Fig. 36: Ganz schwere Schrift mit GEOPAIN

Der kleinste in GEOWRITE mögliche Abstand zwischen linker und rechtem Rand ist 1", wenn der Text nicht zu dünn ist. Schwächerer Text ist auch kaum sinnvoll, weil dann fast kein Word-wrap mehr möglich ist. Wenn Sie doch einmal einen schmaleren Textblock benötigen, so können Sie ihn in GEOPAIN problemlos schreiben und anschließend in GEOWRITE einkleben.

Fig. 37: Schneiden Sie nicht zuviel Text auf einmal aus

Sie können ähnlich später beim Einkleben im TEXTMANAGER einen Absatz bewirken. Normalerweise ist das kaum möglich, weil man ja nur soviel Text ausschneiden kann, wie im GEOWRITE-Fenster sichtbar ist. Wenn man aber alle Tricks benutzt (Schrift in University 6 Point, linken Rand auf 'V', rechten Rand auf 'T'), dann den gesamten Text markiert und später im TEXTMANAGER einleibt, gibt der auf und verabschiedet sich.

3. BASIC-Tools für GEOS

In guter Tradition haben wir als Autoren für GEOS viele Hilfsmittel und Verbesserungen entwickelt, auch für die neue Version GEOS V1.3. Auf Grund der Struktur von GEOS sind natürlich viele dieser Hilfsmittel in Assembler geschrieben und damit für alle diejenigen, die diese Sprache nicht beherrschen, nicht weiterentwickelbar. Wir haben es aber geschafft, eine Reihe von Tools in BASIC zu entwickeln. Diese haben wir in diesem Kapitel zusammengefaßt. Damit ist praktisch jeder von Ihnen in der Lage, diese um bestimmte Fähigkeiten zu erweitern und seinen Bedürfnissen anzupassen.

3.1 Notizen drucken

GEOS hat für viele Probleme ganz bestimmte Standards entwickelt. So werden beispielsweise die verschiedenen Fonts in einer Weise abgelegt, daß sowohl GEOPAINT als auch GEOWRITE (und auch unser Font-Editor) darauf zurückgreifen und neue erzeugen können. Leider wurde aber häufig eine Kompatibilität zwischen den Files verschiedener Programme nicht vorgesehen. So ist es beispielsweise nicht möglich, mit NOTE PAD erstellte NOTES zu drucken. NOTES werden nämlich als System File abgelegt (Filetyp=04). Mit solch einem System File kann man als Benutzer aber praktisch nichts machen. Man kann es nicht duplizieren, umbenennen und auch nicht das Programm, das sie erstellt hat, direkt über ein Doppelklicken des System Files laden.

Mit dieser Gegebenheit haben wir uns nicht zufrieden gegeben. Deshalb haben wir ein Programm geschrieben, mit dem man die NOTES ausdrucken kann. Dabei muß dieses Programm zwei Probleme lösen:

- Es muß ein GEOS-File im Format VLIR lesen.
- Es muß den GEOS-ASCII-Zeichensatz in CBM-ASCII umwandeln.

Das im folgenden abgedruckte Programm "Notizendruck" erledigt beides für Sie und ist zusätzlich noch sehr einfach in der Bedienung. Nach dem Abtippen und Speichern legen Sie einfach die Diskette ein, von der die NOTES gedruckt werden sollen. Nach dem Start und dem Druck einer Taste durchsucht es das Inhaltsverzeichnis nach den NOTES. Anschließend werden alle Pointer des VLIR-Sektors gelesen und in einem Feld gespeichert.

Nun werden zuerst die einzelnen Seiten auf dem Bildschirm angezeigt. Nach jeder Seite können Sie mit RETURN die nächste Seite anschauen oder mit "Q" direkt die Druckerausgabe anwählen. Eventuell müssen Sie die Druckerausgabe noch an Ihren Drucker anpassen. Bei unserer Einstellung werden am Ende jeder Zeile ein Carriage Return (CHR\$(13)) und ein Zeilenvorschub (CHR\$(19)) an den Drucker geschickt. Eventuell müssen Sie das CHR\$(10) bei Ihrem Drucker weglassen.

Jede Seite erhält automatisch eine Überschrift mit fortlaufender Numerierung. An dieser Stelle können Sie natürlich auch das zugehörige Datum ausgeben lassen, wenn Sie es nicht schon in den Notizen auf jede Seite schreiben.

Zum Schluß vor dem Abdruck noch ein kleiner Vorschlag für diejenigen, die das NOTE PAD ausgiebig nutzen und von daher alle Seiten mit Texten gefüllt haben. Sie können das Programm ohne Schwierigkeiten dahingehend erweitern, daß nicht mehr alle, sondern nur bestimmte Seiten ausgedruckt werden, beispielsweise:

- Von Seite 1 bis zur Seite X
- Ab Seite X bis zum Schluß
- Von Seite X bis Seite Y
- Nur die Seiten, die beim Zeigen auf dem Bildschirm markiert wurden.

Für die Fälle a, b, c brauchen Sie nur die zugehörigen Werte durch ein INPUT zu erfragen und die Ausgabeschleife in Zeile 740 an diese Werte anzupassen. Sie sollten aber vorher abfragen, ob die Werte für X und gegebenenfalls Y überhaupt im erlaubten Bereich zwischen "1" und "AN" liegen.

Im Fall d könnten Sie beispielsweise statt des Feldes NO(128,2) eine weitere Spalte, also NO (128,3) dimensionieren. Beim Anzeigen auf dem Bildschirm könnten Sie dann "J", "N" und "Q" als Eingaben erlauben. Bei "J" setzen Sie NO(I,3) auf "1", bei "N" auf "0". In der Druckschleife drucken Sie dann nur die Seiten, bei denen die dritte Spalte des Feldes eine "1" enthält.

3.1.1 Das Programm "Notizendruck"

```

10 DIM NO(128,2)
20 PRINT CHR$(147);CHR$(14)
30 PRINT"          NOTIZEN DRUCKEN!"
40 PRINT
50 PRINT"      BITTE NOTES-DISKETTE EINLEGEN!"
60 PRINT""
70 PRINT"          UND EINE TASTE DRUECKEN"
80 GET A$:IF A$="" THEN 80
90 OPEN5,8,15,"I"
100 OPEN3,8,3,"#"
110 T=18:S=1
120 SC$="NOTES" :SG$=""
130 FOR I=1 TO LEN(SC$)
140 : A=ASC(MID$(SC$,I,1))
150 : GOSUB 940:REM "C64 NACH GEOS"
160 : SG$=SG$+CHR$(A)
170 NEXT I
180 : REM
190 PRINT#5,"U1:"3;0;T;S
200 PRINT#5,"B-P:"3;0
210 GOSUB 910:NT=A
220 GOSUB 910:NS=A
230 BP=2:GE=0 :
240 FOR I=1 TO 8:
250 : PRINT#5,"B-P:"3;BP
260 : GOSUB 910:FT=A:
270 : GOSUB 910:TT=A:
280 : GOSUB 910:SS=A:
290 : FI$=""
300 : IF FT=0 THEN 370
310 : FOR J=1 TO 16:GOSUB 910
320 : . IF A=160 THEN J=16:GOTO 340
330 :   FI$=FI$+A$ :
340 : NEXTJ
350 : PRINTFI$

```

```

360 : IF F1$=SG$ THEN PRINT"GEFUNDEN",TT,SS:I=10:GE=1: GOTO380
370 : BP=BP+32
380 NEXT I
390 IF GE=0 THEN IF NT>0 THEN T=NT:S=NS:GOTO 190
400 IF GE=0 THEN PRINT"LEIDER KEINE NOTES":CLOSE5:END
410 REM "VLIR AUSLESEN"
420 IF TT<0 OR TT>35 THEN PRINT"FEHLER":END
430 PRINT#5,"U1:"3;0;TT;SS
440 ANZ=0:BP=2
450 : PRINT#5,"B-P:"3;BP
460 : GOSUB 910:NT=A:PRINTNT:REM "NOTETRACK"
470 : GOSUB 910:NS=A:PRINTNS:REM "NOTESSEKTOR"
480 : IF NT <1 OR NT>35 OR NS>21THEN520
490 : NO(AN,1)=NT:NO(AN,2)=NS:PRINT NT,NS
500 : AN=AN+1:BP=BP+2:
510 : IF AN <127 THEN 450
520 IF AN=0 THEN PRINT"KEINE SEITEN":CLOSE5:END
530 REM "NOTES AUSLESEN"
540 FOR I=0 TO AN-1 :PRINT"-----"
550 : PRINT"SEITE: ";I+1
560 : X=1
570 : T=NO(I,1):S=NO(I,2)
580 : IF T<1 OR T>35 THEN PRINT"FEHLER":CLOSE1:CLOSE5: END
590 : PRINT#5,"U1:"3;0;T;S
600 : PRINT#5,"B-P:"3;2
610 : GOSUB 910
620 : IF A=0 THEN 680
630 : GOSUB 980:REM "GEOS NACH C64"
640 : IF A>31 THEN PRINTCHR$(A);
650 : IF A=13 THEN X=1: PRINTCHR$(A);
660 : IF X>30 THEN IFA=32THEN:X=1: PRINTCHR$(13);
670 : X=X+1: GOTO 610
680 : PRINT:INPUT "NAECHSTE SEITE Q=ENDE";N$:IFN$="Q"THEN I=AN
690 NEXT I
700 INPUT "DRUCKEN J/N";N$
710 IF N$<>"J" THEN 890
720 REM "NOTES DRUCKEN "
730 OPEN1,4,7: :
740 FOR I=0 TO AN-1
750 : PRINT#1,"-----SEITE ";I+1;"-----";CHR$(10)
760 : X=1
770 : T=NO(I,1):S=NO(I,2)
780 : IF T<1 OR T>35 THEN PRINT"FEHLER": :CLOSE1:CLOSE5: STOP
790 : PRINT#5,"U1:"3;0;T;S
800 : PRINT#5,"B-P:"3;2
810 : GOSUB 910

```

```
820 : IF A=0 THEN 880
830 : GOSUB 980:REM "GEOS NACH C64"
840 : IF A>31 THEN PRINT#1,CHR$(A);
850 : IF A=13 THEN X=1: PRINT#1,CHR$(A);CHR$(10)
860 : IF X>50 THEN IFA=32 THEN X=1: PRINT#1,CHR$(A);CHR$(10)
870 : X=X+1: GOTO 810
880 NEXT I :PRINT#1:CLOSE1
890 CLOSE3:CLOSE5
900 END
910 GET#3,A$:A=ASC(A$+CHR$(0))
920 RETURN
930 REM COMO NACH GEOS
940 IF A>64 AND A<91 THEN A=A+32:RETURN
950 IFA>192ANDA<219THEN A=A-128:RETURN
960 RETURN
970 REM GEOS NACH COMO
980 IF A>64 AND A<91 THEN A=A+128:RETURN
990 IFA>96ANDA<123THEN A=A-32:RETURN
1000 RETURN
```

3.1.2 Dokumentation des Programms "Notizendruck"

In diesem Buch finden Sie eine ganze Reihe von Programmen, die Ihnen helfen und Ihre Arbeit erleichtern oder verbessern sollen. Vielleicht denken Sie aber gerade über eine Routine, die wir nützlich finden, nicht so, oder Sie vermissen eine, die wir überflüssig fanden. Unsere Programme sind daher eigentlich nur Vorschläge (natürlich funktionsfähige), und wir möchten Sie auffordern, diese Ihren eigenen Bedürfnissen anzupassen. Deshalb sind die meisten Programme in diesem Buch dokumentiert. So wird es Ihnen mit ein wenig Programmiererfahrung nicht schwerfallen, die Funktion der Programme zu verstehen und daraus vielleicht sogar noch einige Tricks zu lernen. Nun aber zur Dokumentation des Programms "Notizendruck":

10 Das Feld NO(128) enthält die Zeiger auf die möglichen Seiten des NOTE PAD. Dabei enthält NO(I,1) jeweils den Track und NO(I,2) den Sektor.

20-80 Nach der Bildschirmmeldung wartet das Programm, bis die Diskette mit den NOTES eingelegt und eine Taste gedrückt wurde.

- 90 Der Befehlskanal wird eröffnet und die eingelegte Diskette initialisiert.
- 100 Ein weiterer Kanal wird zur Floppy eröffnet und diesem ein Puffer zugewiesen.
- 110 T wird als Variable für den jeweils aktuellen Track und S für den aktuellen Sektor benutzt. Zu Beginn werden beide auf den ersten Sektor des Inhaltsverzeichnisses der Diskette gesetzt.
- 120 SC\$ erhält den gewünschten Filenamen, der gesucht werden soll. Dabei deutet das "C" an, daß die Zeichen "C"OMMODORE-Format haben. Die Variable SG\$ ("G" wie GEOS) wird als Leerstring initialisiert.
- 130-170 In einer Schleife konvertiert das Programm Zeichen für Zeichen. Am Ende enthält SG\$ dann den gesuchten File-Eintrag in der GEOS-Verschlüsselung. Dafür wird das Unterprogramm ab Zeile 940 benutzt, das einen CBM-ASCII Wert in A in GEOS-ASCII umwandelt. Der konvertierte Wert wird in A zurückgeliefert. Das Programm ist von Zeile 120 - Zeile 170 mit Absicht variabel gehalten, um auch grundsätzlich andere File-Einträge finden zu können.
- 190 Ab hier beginnt die Schleife zum Einlesen des gesamten Directory. In dieser Zeile wird der aktuelle Sektor in den Puffer geladen.
- 200 Der Buffer-Pointer wird auf das erste Byte gesetzt (Byte 0). Dies enthält den Track des folgenden Sektors (falls vorhanden).
- 210 Das Unterprogramm holt ein Zeichen aus dem Puffer der Floppy und übergibt in "A" den zugehörigen ASCII-Wert. Dieser wird hier der Variablen NT (Neuer Track) zugewiesen.
- 220 Das folgende Byte enthält den Sektor. Es wird der Variablen NS (Neuer Sektor) zugewiesen.

- 230 BP (Buffer-Pointer) wird auf den Filetyp des ersten File-Eintrags gesetzt. Die Variable für Gefunden (GE) wird auf "nicht gefunden" gesetzt.
- 240 Hier beginnt die Prüfschleife für jeweils 8 File-Einträge eines Sektors.
- 250-280 Filetyp, Track und Sektor des File-Eintrags werden geholt und in Variablen gesichert.
- 290 In FIS wird der Filename zusammengesetzt. Deshalb muß FIS vorher als Leerstring initialisiert werden.
- 300 Wenn der Filetyp "0" ist, ist das File gelöscht und darf nicht berücksichtigt werden.
- 310-340 In einer Schleife werden bis zu 16 Zeichen gelesen und zum Filenamens zusammengesetzt. Da kürzere Namen in der Floppy mit "SHIFT BLANK=160) aufgefüllt werden, wird bei diesem Zeichen die Schleife abgebrochen.
- 350 Bildschirmausgabe des Filenamens
- 360 Stimmen gesuchter und gefundener Name überein, wird eine entsprechende Meldung ausgegeben. Zusätzlich werden Starttrack und Startsektor des ersten Datenblocks angezeigt. Ge wird auf "gefunden" gesetzt und die Schleife verlassen.
- 370-380 Der Buffer-Pointer wird auf den Filetyp des nächsten File-Eintrags gesetzt und ein neuer Schleifendurchlauf begonnen.
- 390 Wurde der File-Eintrag (Notes) noch nicht gefunden und gibt es einen weiteren Directory-Block, so werden Track und Sektor auf diesen Block gesetzt, und das Programm springt zum erneuten Einlesen.
- 400 Gibt es keinen weiteren Block mehr und wurde nichts gefunden, wird eine entsprechende Meldung ausgegeben und das Programm beendet.

- 420 TT,SS enthalten Track und Sektor des VLIR-Pointer-Blocks. Sollte aus irgendeinem Grund ein ungültiger Wert eingelesen worden sein, endet das Programm mit einer Fehlermeldung.
- 430 Einlesen des Pointer-Blocks von NOTES
- 440 Die Anzahl der gefundenen Seiten wird auf "0" gesetzt, die Variable für den Buffer-Pointer wird auf das dritte Byte gesetzt, da die ersten stets \$00 und \$FF (255) enthalten.
- 450 Setzen des Buffer-Pointers auf den Wert, den die Variable enthält. Diese Zeile wird mehrmals benutzt.
- 460-470 Jeweils zwei Bytes enthalten Track und Sektor einer Notizenseite. Diese werden aus dem Block eingelesen.
- 480 Liegen die Werte nicht im erlaubten Bereich, wird das Einlesen weiterer Pointer abgebrochen. An NT=0 erkennt man auch, daß keine weiteren Seiten mehr folgen.
- 490 Track und Sektor werden im Feld gespeichert und angezeigt.
- 500 Die Anzahl der Seiten wird um "1" erhöht und der Buffer-Pointer um zwei Bytes weitersetzt.
- 510 Ist die maximal mögliche Anzahl Seiten noch nicht erreicht, wird der nächste Pointer gelesen.
- 520 An dieser Stelle wird stets geprüft, ob überhaupt eine einzige Seite vorhanden ist. Das ist wichtig, da man durch sofortiges Verlassen des NOTE PAD ohne eine Eingabe ein Notizbuch ohne eine Seite erhält.
- 540 Ausgabeschleife für alle vorhandenen Seiten
- 550 Jede Seite erhält eine fortlaufende Numerierung. Hier können Sie eventuell auch das Datum auf jede Seite drucken lassen.

- 560 X enthält später die Anzahl der bisher in der Zeile ausgegebenen Zeichen. Dadurch kann eine Art Wordwrap erreicht werden, da sonst der Bildschirm unübersichtlich vollgeschrieben würde. Hier wird X initialisiert.
- 570 Track und Sektor der aktuellen Seite werden aus dem Feld geholt.
- 580 Nochmal wird der Wert für den Track auf den gültigen Bereich überprüft.
- 590 Einlesen der Seite in den Puffer
- 600 Der Buffer-Pointer wird auf das erste Text-Byte gesetzt. Die beiden Bytes davor enthalten wieder \$00 und \$FF als Kennzeichnung dafür, daß es keinen Folge-Block gibt und der Block 255 gültige Daten enthält.
- 610 Ein Zeichen wird aus dem Puffer geholt. Das Zeichen wird in A\$, sein ASCII-Wert in A übergeben.
- 620 Eine '0' bedeutet bei NOTES das Ende der Seite.
- 630 GEOS-ASCII wird in CBM-ASCII umgewandelt.
- 640 Ist das Zeichen kein Steuerzeichen, wird es ausgegeben.
- 650 Ist das Zeichen ein Zeilenende, dann wird auch auf dem Bildschirm ein "RETURN" ausgegeben. Zusätzlich wird dann der Zähler für die Anzahl der Zeichen in der aktuellen Zeile zurückgesetzt.
- 660 Ab dem 30. Zeichen in der Zeile wird bei einem Leerzeichen (A=32) eine neue Seite begonnen. Damit wird weitgehend verhindert, daß Wörter unübersichtlich zerschnitten werden. Der Text bleibt gut lesbar.
- 670 Der Zeilenzähler wird um "1" erhöht und zum erneuten Einlesen gesprungen.

- 680 Beim Ende der Seite kann die nächste Seite durch RETURN auf dem Bildschirm ausgegeben werden. Mit "Q" kann man direkt zur Druckerausgabe gelangen.
- 690 Schleifenende für Ausgabe aller Notizenseiten.
- 700 Drucken muß mit "J" bestätigt werden.
- 730 Ein Ausgabekanal zum Drucker wird geöffnet.
- 740 Ausgabeschleife für alle vorhandenen Seiten.
- 750 Die Seiten werden auf dem Drucker durch einen Strich mit der Seitenangabe getrennt. Wir haben nicht für jede neue Notizenseite auch ein neues Blatt angefangen, um den Ausdruck zu beschleunigen und Papier zu sparen.
- 760-880 Dies ist im wesentlichen die gleiche Ausgabeschleife wie bei der Bildschirmausgabe. Allerdings geben wir bei unserem Drucker zusätzlich zum RETURN (CHR\$(13)) noch einen Zeilenvorschub (CHR\$(10)). Sollte Sie das stören, so entfernen Sie den Zeilenvorschub aus Zeile 850. Außerdem wird erst ab 50 Zeichen in einer Zeile nach einem Leerzeichen zum Beginn einer neuen Zeile gesucht. Wenn Sie diesen Wert ändern wollen, ersetzen Sie "50" in Zeile 860 durch einen anderen Wert.
- 910-920 Unterprogramm: Hier wird ein Zeichen aus dem Floppypuffer geholt und der gültige ASCII-Wert bestimmt. Es wird an jedes Zeichen zusätzlich ein CHR\$(0) angehängt, bevor der ASCII-Wert bestimmt wird. Andernfalls würde der Befehl ASC(A\$) bei einem Leerstring (A\$="") eine Fehlermeldung erzeugen.
- 930-960 Unterprogramm: Hier wird der CBM-ASCII-Wert in einen GEOS-ASCII-Wert umgewandelt. Diese Routine können Sie vielleicht auch in eigenen Programmen verwenden.
- 930-960 Unterprogramm: Ein GEOS-ASCII-Wert wird in einen CBM-ASCII-Wert umgewandelt.

3.2 GEOWRITE-Texte drucken

Das in der Grundausstattung zu GEOS enthaltene GEOWRITE bietet Ihnen völlig neue Möglichkeiten bei der Gestaltung Ihrer Texte auf dem C64. Dazu gehören Proportionalschrift, unterschiedliche Schriftarten und verschiedene Schriften. Allerdings haben diese Möglichkeiten auch ihren Preis. Wir meinen damit nicht den Verkaufspreis von GEOS, sondern die Druckgeschwindigkeit. Der Ausdruck des Textes muß nämlich als Grafik erfolgen, und dadurch sinkt die Druckgeschwindigkeit auf ein Minimum.

Schaut man sich das Ergebnis auf dem Papier an, wird man diesen Nachteil in den meisten Fällen wohl in Kauf nehmen. Leider kommt es aber häufig vor, daß man nach dem Ausdrucken feststellt, daß doch irgendwo noch ein paar Fehler übersehen worden sind und der Text noch einmal gedruckt werden muß. Dadurch geht dann allerdings weitere kostbare Zeit verloren, und wir können Ihnen nur wünschen, daß Sie beim zweiten Ausdruck keine Fehler mehr finden. Abhilfe schafft hier die Version GEOWRITE V2.0, die es erlaubt, die Texte mit dem Zeichensatz des angeschlossenen Druckers auszugeben. Dadurch erhöht sich die Ausgabegeschwindigkeit um ein Vielfaches.

Da aber sicher nicht jeder im Besitz dieser GEOWrite-Luxus-Version sein wird, haben wir für diesen Leserkreis nach einer Lösung gesucht und diese auch gefunden.

Schaut man sich die Art und Weise an, in der die Texte aller GEOWRITE-Versionen auf der Diskette abgespeichert werden, stellt man fest, daß der Text nicht als Grafik, sondern als ASCII-Fließtext abgelegt wird. Es liegt also Zeichen hinter Zeichen, und dazwischen befinden sich die Steuerzeichen für die Formatierung und die verschiedenen Schriften und Schriftarten. Erst auf dem Bildschirm und bei der Druckerausgabe werden diese Zeichen wieder in Grafik umgewandelt. Das brachte uns auf die Idee, ein Programm zu schreiben, mit dem ein mit GEOWRITE 1.2 oder 1.3 erstellter Text nicht als Grafik, son-

dern als Text ausgedruckt wird. Somit können Sie den Text schnell einmal ausdrucken lassen, korrigieren, und ihn, wenn er schließlich fehlerfrei ist, mit GEOWRITE endgültig drucken.

3.2.1 GEODRUCK: GEOWRITE-Texte als Text drucken

Grundsätzlich basiert das Programm GEODRUCK auf dem Programm "Notizendruck". Allerdings mußten einige entscheidende Änderungen und Erweiterungen berücksichtigt werden:

- Der Text wird von GEOWRITE etwas anders abgelegt als vom NOTE PAD.
- GEOWRITE speichert zu Beginn des Textes die Einstellungen für den linken und rechten Rand und die Tabulatoren ab. Diese Einstellungen müssen überlesen werden.
- In GEOWRITE-Texten kommen verschiedene Steuerzeichen für Schriftwechsel vor, die überlesen werden müssen.
- Ein GEOWRITE-Text kann maximal aus 63 Seiten bestehen, die NOTES dagegen aus 127 Seiten.

Bevor wir das Programm "GEODRUCK" abdrucken, wollen wir Ihnen noch einige Hinweise zur Bedienung geben. Nachdem Sie es gestartet haben, legen Sie bitte die Diskette mit dem zu druckenden Text ein. Nach einem Tastendruck werden Sie aufgefordert, den Namen des Textes einzugeben. Anschließend durchsucht das Programm das gesamte Directory nach diesem Namen. Dabei werden alle gültigen Namen im Inhaltsverzeichnis angezeigt, so daß Sie, falls Sie sich nicht mehr an die Schreibweise erinnern, diese am Bildschirm ablesen können.

Hat das Programm den File-Eintrag gefunden, werden die Pointer der VLIR-Struktur in das Feld gelesen. Anschließend können Sie entscheiden, ob die Ausgabe auf den Drucker oder den Bildschirm erfolgen soll. Nach jeder Seite drücken Sie bitte RETURN oder "Q", falls die Ausgabe abgebrochen werden soll.

"GEODRUCK" ist nicht besonders schnell. Wir benutzen es ausschließlich in kompilierter Form, und dann ist die Geschwindigkeit völlig ausreichend. Sollten Sie also einen BASIC-Compiler besitzen, so bringen Sie das Programm ruhig auf Trapp. Auch ohne Compiler gibt es sicherlich noch eine ganze Reihe von Möglichkeiten, das Programm zu beschleunigen. Wir haben das in der abgedruckten Version nicht getan, um es nicht unnötig kompliziert zu machen.

3.2.2 Das Programm "GEODRUCK"

```

20 DIM NO(64,2)
40 PRINT CHR$(147);CHR$(14)
60 PRINT"          GEOWRITE-TEXTE DRUCKEN!"
80 PRINT"          -----"
100 PRINT
120 PRINT
140 PRINT"          BITTE DISKETTE EINLEGEN!"
160 PRINT
180 PRINT"          UND EINE TASTE DRUECKEN"
200 GET A$:IF A$="" THEN 200
220 OPEN5,8,15,"I"
240 OPEN3,8,3,"#"
260 T=18:S=1
280 PRINT:PRINT
300 PRINT:PRINT
320 INPUT "          TEXTNAME:";SC$
340 SG$=""
360 FOR I=1 TO LEN(SC$)
380 : A=ASC(MID$(SC$,I,1))
400 : GOSUB 2480:REM "C64 NACH GEOS"
420 : SG$=SG$+CHR$(A)
440 NEXT I
460 :REM
480 PRINT#5,"U1:"3;0;T;S
500 PRINT#5,"B-P:"3;0
520 GOSUB 2420:NT=A
540 GOSUB 2420:NS=A
560 BP=2:GE=0 :
580 FOR I=1 TO 8:
600 : PRINT#5,"B-P:"3;BP
620 : GOSUB 2420:FT=A:
640 : GOSUB 2420:TT=A:

```

```

660 : GOSUB 2420:SS=A:
680 : FI$=""
700 : IF FT=0 THEN 840
720 : FOR J=1 TO 16:GOSUB 2420
740 : IF A=160 THEN J=16:GOTO 780
760 : FI$=FI$+A$ :
780 : NEXTJ
800 : PRINTFI$
820 : IF FI$=SG$ THEN PRINT"GEFUNDEN",TT,SS:I=10:GE=1: GOTO860
840 : BP=BP+32
860 NEXTI
880 IF GE=0 THEN IF NT>0 THEN T=NT:S=NS:GOTO 480
900 IF GE=0 THEN PRINT"LEIDER KEIN ";SC$;" GEFUNDEN":CLOSE5:END
920 REM "VLIR AUSLESEN"
940 IF TT<0 OR TT>35 THEN PRINT"FEHLER":END
960 PRINT#5,"U1:"3;0;TT;SS
980 ANZ=0:BP=2
1000 : PRINT#5,"B-P:"3;BP
1020 : GOSUB 2420:NT=A:PRINTNT:REM "TEXT-TRACK"
1040 : GOSUB 2420:NS=A:PRINTNS:REM "TEXT-SEKTOR"
1060 : IF NT <1 OR NT>35 OR NS>21THEN1140
1080 : NO(AN,1)=NT:NO(AN,2)=NS:PRINT NT,NS
1100 : AN=AN+1:BP=BP+2:
1120 : IF AN <64 THEN 1000
1140 IF AN=0 THEN PRINT"KEINE SEITEN":CLOSE5:END
1160 REM *****
1180 REM "TEXT AUSLESEN"
1185 : INPUT"BILDSCHIRM/DRUCKER B/D";N$
1186 IF N$<>"D" AND N$<>"B" THEN 1185
1187 : IF N$<>"D" THEN 1200
1190 : OPEN1,4,2:
1193 : PRINT#1,CHR$(27)CHR$(64)
1195 :
1200 FOR I=0 TO AN-1
1205 : IF N$="B" THEN PRINT"-----"
1210 : IF N$="D" THEN PRINT#1,"-----";CHR$(10)
1220 : IF N$="B"THEN PRINT"SEITE: ";I+1
1225 : IF N$="D"THEN PRINT#1,"SEITE: ";I+1;CHR$(10)
1240 : X=1 :PO=22
1260 : T=NO(I,1):S=NO(I,2)
1280 : IF T<1 OR T>35 THEN PRINT"FEHLER":CLOSE1:CLOSE5: END
1300 : PRINT#5,"U1:"3;0;T;S
1320 : PRINT#5,"B-P:"3;0
1340 : GOSUB 2420:TT=A
1360 : GOSUB 2420:SS=A
1380 : MA=255: REM "MAXIMALER POINTER"

```

```
1400 : PRINT#5,"B-P:"3;PO
1420 : IF TT=0 THEN MA=SS-1
1440 : GOSUB 2420:PO=PO+1
1460 : IF A<>23 THEN 1780
1480 : FOR K =1TO 3
1500 : PO=PO+1:IF PO>MA THEN 1580
1520 : GOSUB 2420
1540 : SA(K)=A :REM"SCHRIFFTART"
1560 : NEXT K :GOTO 1440
1580 : T=TT:S=SS:IF TT=0 THEN PRINT"FEHLER":CLOSE5:END
1600 : PRINT#5,"U1:"3;0;T;S
1620 : PRINT#5,"B-P:"3;0
1640 : GOSUB 2420:TT=A
1660 : GOSUB 2420:SS=A
1680 : MA=255: REM "MAXIMALER POINTER"
1700 : PO=2
1720 : PRINT#5,"B-P:"3;PO
1740 : IF TT=0 THEN MA=SS
1760 : GOTO 1520
1780 : IF A<>12 THEN 1840
1800 : IFN$="D"THEN PRINT#1,CHR$(13)CHR$(10)
1805 : IFN$= "B"THEN PRINTCHR$(13)CHR$(10)
1820 : IFN$="D"THEN PRINT#1,"-----
-";CHR$(10)
1825 : IFN$= "B"THEN PRINT"-----"
1840 : IF A=0 THEN PRINT"FEHLER":CLOSE5:END
1860 : GOSUB 2560:REM "GEOS NACH C64"
1880 : IF N$="B"THEN IF A>31 THEN PRINTCHR$(A);
1885 : IF N$="D"THEN IF A>31 THEN PRINT#1,CHR$(A);
1900 : IF N$="B"THEN IF A=13 THEN X=1: PRINTCHR$(A);
1905 : IF N$="D"THEN IF A=13 THEN X=1: PRINT#1,CHR$(A);
1920 : IFN$="B" THEN IF X>30 THEN IFA=32THEN:X=1: PRINTCHR$(13);
1925 : IFN$="D" THEN IF X>30 THEN IFA=32THEN:X=1: PRINT#1,CHR$(13);
CHR$(10)
1940 : X=X+1:IF PO<=MA THEN GOTO 1440
1945 IF TT>0 THEN T=TT:S=SS:PO=2:GOTO 1300
1947 IF N$="D" THEN PRINT#1,CHR$(13);CHR$(10)
1950 : IF N$="D" THEN 1980
1960 : PRINT:INPUT "NAECHSTE SEITE Q=ENDE";M$
1965 : IFM$="Q"THEN I=AN
1980 NEXTI
2380 CLOSE1: CLOSE3:CLOSE5
2400 END
2420 GET#3,A$:A=ASC(A$+CHR$(0))
2440 RETURN
2460 REM COMO NACH GEOS
```

```

2480 IF A>64 AND A<91 THEN A=A+32:RETURN
2500 IFA>192ANDA<219THEN A=A-128:RETURN
2520 RETURN
2540 REM GEOS NACH COMO
2560 IF A>64 AND A<91 THEN A=A+128:RETURN
2580 IFA>96ANDA<123THEN A=A-32:RETURN
2600 RETURN

```

3.2.3 Dokumentation des Programms "GEODRUCK"

- 20 Das Feld NO(128) enthält später die Zeiger auf die Seiten des Textes.
- 40-200 Nach der Bildschirmmeldung wartet das Programm, bis die Diskette mit dem Text eingelegt und eine Taste gedrückt wird.
- 220 Der Befehlskanal wird geöffnet und die eingelegte Diskette initialisiert.
- 240 Ein weiterer Kanal zur Floppy wird eröffnet und diesem ein Puffer zugewiesen.
- 260 T wird als Variable für den jeweils aktuellen Track und S für den aktuellen Sektor benutzt. Zu Beginn werden beide auf den ersten Sektor des Inhaltsverzeichnisses der Diskette gesetzt.
- 280-340 SC\$ erhält den gewünschten Filenamen, der gesucht werden soll. Dabei deutet das "C" an, daß die Zeichen "C"OMMODORE-Format haben. Die Variable SG\$ ("G" wie GEOS) wird als Leerstring initialisiert.
- 360-440 In einer Schleife konvertiert das Programm Zeichen für Zeichen. Am Ende enthält SG\$ dann den gesuchten File-Eintrag in der GEOS-Verschlüsselung.
- 480 Ab hier beginnt die Schleife zum Einlesen des gesamten Directorys. In dieser Zeile wird der aktuelle Sektor in den Puffer geladen.

- 500 Der Buffer-Pointer wird auf das erste Byte gesetzt (Byte 0). Dies enthält den Track des folgenden Sektors (falls vorhanden).
- 520 Das Unterprogramm holt ein Zeichen aus dem Puffer der Floppy und übergibt in "A" den zugehörigen ASCII-Wert. Dieser wird hier der Variablen NT (Neuer Track) zugewiesen.
- 540 Das folgende Byte enthält den Sektor des nächsten Directoryblocks. Es wird der Variablen NS (Neuer Sektor) zugewiesen.
- 560 BP (Buffer-Pointer) wird auf den Filetyp des ersten File-Eintrags gesetzt. Die Variable für "Gefunden" (GE) wird auf "nicht gefunden" gesetzt.
- 580 Hier beginnt die Prüfschleife für die 8 File-Einträge eines Directoryblocks
- 600-660 Filetyp, Track und Sektor des File-Eintrags werden geholt und in den Variablen FT, TT, SS gesichert.
- 680 In FI\$ wird der Filename zusammengesetzt. Deshalb muß FI\$ vorher als Leerstring initialisiert werden.
- 700 Wenn der Filetyp "0" ist, ist das File gelöscht und darf nicht berücksichtigt werden.
- 720-780 In einer Schleife werden bis zu 16 Zeichen gelesen und zum Filenamens zusammengesetzt. Da kürzere Namen in der Floppy mit "SHIFT BLANK=160) aufgefüllt werden, wird bei diesem Zeichen die Schleife abgebrochen.
- 800 Ausgabe des Filenamens auf dem Bildschirm. Dadurch können Sie prüfen, ob der gewünschte Text vielleicht überlesen wird, und die korrekte Schreibweise des Namens feststellen.
- 820 Stimmen gesuchter und gefundener Name überein, wird eine entsprechende Meldung ausgegeben. Zusätzlich werden Starttrack und Startsektor des ersten Datenblocks angezeigt. Die Variable "Ge" wird auf "gefunden" gesetzt und die Schleife verlassen.

- 840-860 Der Buffer-Pointer wird auf den Filetyp des nächsten File-Eintrags gesetzt und ein neuer Schleifendurchlauf begonnen.
- 880 Wurde der File-Eintrag des gewünschten Textes bisher nicht gefunden und gibt es einen weiteren Directory-Block, so werden Track und Sektor auf diesen Block gesetzt, und das Programm springt zum erneuten Einlesen.
- 900 Gibt es keinen weiteren Block mehr und wurde nichts gefunden, wird eine entsprechende Meldung ausgegeben und das Programm beendet.
- 940 TT,SS enthalten Track und Sektor des VLIR-Pointer-Blocks. Sollte aus irgendeinem Grund ein ungültiger Wert eingelesen worden sein, endet das Programm mit einer Fehlermeldung.
- 960 Einlesen des Pointer-Blocks des gewünschten Textes.
- 980 Die Anzahl der gefundenen Seiten wird auf "0" gesetzt, die Variable für den Buffer-Pointer wird auf das dritte Byte gesetzt, da die ersten beiden stets \$00 und \$FF (255) enthalten.
- 1000 Setzen des Buffer-Pointers auf den Wert, den die Variable enthält. Diese Zeile wird mehrmals benutzt.
- 1020-1040 Jeweils zwei Bytes enthalten Track und Sektor des ersten Datenblocks einer Textseite. Diese werden aus dem VLIR-Block eingelesen.
- 1060 Liegen die Werte nicht im erlaubten Bereich, wird das Einlesen weiterer Pointer abgebrochen. An NT=0 erkennt man auch, daß keine weiteren Seiten mehr folgen. Daran erkennt man, daß es in GEOWRITE nicht möglich ist, Seite 3 zu beschreiben, ohne daß es eine Seite 2 gibt.
- 1080 Track und Sektor des ersten Datenblocks werden im Feld gespeichert und angezeigt.

- 1100 Die Anzahl der Seiten wird um "1" erhöht und der Buffer-Pointer um zwei Bytes weitersetzt.
- 1120 Ist die maximal mögliche Anzahl Seiten noch nicht erreicht, wird der nächste Pointer gelesen.
- 1140 An dieser Stelle wird stets geprüft, ob überhaupt eine einzige Seite vorhanden ist.
- 1185 Soll auf Drucker oder Bildschirm ausgegeben werden? Dies wird in N\$ abgefragt und gespeichert. Für beide Ausgaben wird das gleiche Programm benutzt.
- 1187 Falls die Ausgabe auf den Bildschirm erfolgen soll, wird die Druckerinitialisierung übersprungen.
- 1190-1195 Vorbereitung zur Druckerausgabe. Ein Kanal zum Drucker wird geöffnet und dieser initialisiert. Die Initialisierungssequenz gilt nur für einen Epson FX 85. Haben Sie einen anderen Drucker, so schauen Sie bitte die entsprechenden Werte in Ihrem Handbuch nach oder lassen Zeile 1193 ganz weg.
- 1200 Ausgabeschleife für alle vorhandenen Seiten des Textes.
- 1205 Jede Seite beginnt mit einem trennenden Strich auf dem Bildschirm.
- 1210 Alle Ausgaben werden bei gewünschter Drucker- ausgabe auf den Drucker gegeben. Einfacher wäre es zwar mit dem Befehl CMD gewesen. Dabei traten aber Fehler beim Auslesen des Floppy-Puffers auf, so daß wir diesen umständlicheren Weg wählen mußten. Im folgenden finden Sie also jede Ausgabe einmal für den Bildschirm und einmal für den Drucker.
- 1220 Jede Seite erhält eine fortlaufende Numerierung.
- 1240 X enthält später die Anzahl der bisher in der Zeile ausgegebenen Zeichen. Dadurch kann eine Art Wordwrap erreicht werden, da sonst der Bildschirm

unübersichtlich vollgeschrieben würde. Wenn GEOWRITE den Text zeigt oder druckt, kümmert es sich ebenfalls um das Zeilenende. Hier wird X initialisiert.

PO wird im folgenden Programmteil als Zähler für die Anzahl der Bytes in einem Sektor benutzt. Das Programm muß wissen, wann die Daten eines Sektors zu Ende sind und ein neuer Sektor eingelesen werden muß. Da GEOWRITE vor dem eigentlichen Text den linken und rechten Rand und die Tabulatoren abspeichert, müssen diese mit PO=22 überlesen werden.

- 1260 Track und Sektor der aktuellen Seite werden aus dem Feld geholt.
- 1280 Der Wert für den Track wird auf den gültigen Bereich überprüft.
- 1300 Einlesen des ersten Sektors der aktuellen Seite in den Puffer
- 1320 Der Buffer-Pointer wird auf den Anfang des Puffers gesetzt. Dort stehen entweder Track und Sektor des nächsten Datenblocks oder eine "0" und die Anzahl der gültigen Bytes.
- 1340-1360 Die ersten beiden Bytes werden aus dem Puffer geholt und gesichert.
- 1380 Der größtmögliche Wert für den Pufferzeiger wird gesetzt. Er muß variabel sein, da der letzte Sektor einer Seite weniger gültige Bytes enthalten kann.
- 1400 Setzen des Buffer-Pointer auf den Wert der Variablen PO. Diese ist hier noch "22", um die Einstellungen zu überlesen.
- 1420 Wenn TT=0 ist, gibt es keinen Folge-Sektor, und in SS steht die Anzahl der gültigen Daten. Dann wird der maximale Wert "MA" für den Zeiger im Puffer "PO" entsprechend gesetzt.

- 1440 Ein Zeichen aus dem Puffer holen. Das Zeichen wird in A\$, sein ASCII-Wert in A übergeben.
- 1460 Bei A=23 wird der folgende Programmblock ausgeführt. Das ist das Steuerzeichen für Schriftwechsel, und die folgenden Bytes müssen überlesen werden.

3.3 Der Text-Konverter

Vielleicht haben Sie mit Ihrem C64 schon die Erfahrung gemacht, daß es zwei unterschiedliche Arten der Textverarbeitung gibt:

Die eine zeichnet sich durch ihre grafischen Fähigkeiten aus und ist aufgrund der zeitraubenden Bildschirmoperationen relativ langsam. Die andere Art der Textverarbeitung wiederum ist zwar recht flink, bietet dafür aber auch nicht die vielfältigen Möglichkeiten der hochauflösenden Grafik.

GEOWRITE gehört zu den Programmen der ersten Kategorie. Die Möglichkeiten, die es bietet, sind wirklich enorm. Jeden beliebigen Textabschnitt kann man z.B. auch nachträglich in einer völlig anderer Schriftart ausgeben lassen.

Doch jeder, der schon einmal mit einem Textverarbeitungsprogramm des Kalibers Textomat Plus von DATA BECKER gearbeitet hat, wird von der Geschwindigkeit der GEOS-Anwendung enttäuscht sein. Ein so von Geschwindigkeit verwöhnter Schreiber wird wohl kaum mehr für 'normale' Texte auf das langsame GEOWRITE zurückgreifen. Leider akzeptiert dieses Programm auch nur die Texte, die es selber abgespeichert hat, so daß es auch nicht möglich ist, einen Text z.B. mit Textomat Plus zu schreiben und anschließend in GEOWRITE einzuladen.

Doch genau hierfür bieten wir Ihnen das richtige Hilfsmittel an. Es gestattet Ihnen, einen Text auf einem nahezu beliebigen, schnellen Textverarbeitungsprogramm zu schreiben und ihn anschließend (mit einem Icon versehen) in GEOWRITE einzulesen. Hier kann GEOWRITE dann an dem Text seine Fähigkeiten voll

unter Beweis stellen. Für kleinere Änderungen am Text reicht dann auch die hier erzielte Geschwindigkeit völlig aus. Das Hilfsmittel ist natürlich nichts anderes als ein Programm. Es ist im nächsten Abschnitt abgedruckt.

3.3.1 Das Listing des Konverters

Das unten abgedruckte Programm "Konverter" ist vollständig in BASIC geschrieben. Bei der Eingabe sind die folgenden Punkte zu beachten:

- Das Semikolon ist in diesem Programm der wichtigste Befehl! Ein Fehlen wird nicht durch SYNTAX ERROR reklamiert.
- Die Data-Zeilen mit gleichem Inhalt ("32") können durch Voranstellen der nächsten Zeilennummer dupliziert werden. Dies vereinfacht die Eingabe.

```

0 REM GEOWRITE TEXTKONVERTER
5 REM AUTOR: RUEDIGER KERKLOH
10 DIM Z$(255),K(23)
15 GOSUB 655:REM STANDARD-TABELLE HOLEN
20 GOSUB 610:REM KOPFZEILE AUSGEBEN
25 PRINT" 1 = GEOS-TEXT ERZEUGEN":PRINT
30 PRINT" 2 = TABELLE AENDERN":PRINT
35 PRINT" 3 = TABELLE LADEN":PRINT
40 PRINT" 4 = TABELLE SPEICHERN":PRINT
45 PRINT:PRINT" BITTE WAEHLEN !"
50 GETW$:W=VAL(W$):IF W<1 OR W>4 THEN 50
55 GOSUB 610:REM KOPFZEILE AUSGEBEN
60 OPEN 1,8,15,"1:0":GOSUB 635
65 ON W GOSUB 85,465,520,565
70 CLOSE 1
75 GOTO 20:REM PROGRAMMSCHLEIFE
80 :
85 PRINT" BITTE GEBEN SIE DEN NAMEN DES ZU "
90 PRINT " KONVERTIERENDEN TEXTES EIN!"
95 PRINT:INPUT AT$
100 GOSUB 610
105 PRINT " WIEVIELE ZEICHEN PRO SEITE (1-4580) ":PRINT
110 INPUT AZ
115 IF AZ<1 OR AZ>4580 THEN 100
120 :

```

```
125 REM DATEN IN HILFSFILE ABLEGEN
130 OPEN 2,8,2,AT$+"R":GOSUB 635
135 M$="TEXT-DUMMY"
140 PRINT#1,"S:";M$+"*"
145 OPEN 3,8,3,M$+STR$(SZ+1)+"S,W":GOSUB 635
150 :
155 REM TEXT-KOPF ANLEGEN
160 FOR I=0 TO 23
165 PRINT#3,CHR$(K(I));
170 NEXT I
175 :
180 REM TEXT-DUMMYS ANLEGEN
185 P=0:REM ZEICHENZAehler PRO SEITE
190 GET#2,T$:IF T$="" THEN T$=CHR$(0)
195 S=ST AND 64:T=ASC(T$)
200 IF T<161 OR T>170 THEN 255
205 T1=T:T=69:REM KLEINES "E"
210 IF T1=161 THEN T1=79:GOTO 250
215 IF T1=165 THEN T1=65:GOTO 250
220 IF T1=166 THEN T1=85:GOTO 250
230 IF T1=168 THEN T1=207:GOTO 250
235 IF T1=169 THEN T1=193:GOTO 250
240 IF T1=170 THEN T1=213:GOTO 250
245 IF T1=167 THEN T1=83:T=T1
250 PRINT#3,Z$(T1);:REM KODIEREN
255 PRINT#3,Z$(T);:REM KODIEREN
260 IF S THEN 270:REM END OF FILE
265 P=P+1:IF P<AZ THEN 190
270 PRINT#3,CHR$(0);:REM TEXTABSCHLUSS
275 CLOSE 3:REM TEXT-DUMMY SCHLIESSEN
280 X$=M$+STR$(SZ+1):GOSUB 365:REM T & S HOLEN
285 E$=C$:F$=D$:REM TEXT-DUMMY MERKEN
290 X$="DUMMY":GOSUB 365:REM T & S
295 OPEN 5,8,5,"#":GOSUB 635
300 PRINT#1,"U1";5;0;ASC(C$);ASC(D$)
305 PRINT#1,"B-P";5;SZ*2+2
310 PRINT#5,E$;F$;:REM EINTRAGEN
315 PRINT#1,"U2";5;0;ASC(C$);ASC(D$)
320 CLOSE5
325 SZ=SZ+1
330 GOSUB 610:REM KOPFZEILE
335 PRINT" ANZAHL SEITEN: ";SZ
340 IF SZ=63 AND S=0 THEN PRINT:PRINT" TEXT ZU LANG !":GOTO 350
345 IF S=0 THEN 145
350 PRINT#1,"S:";M$;*"
355 CLOSE 2:RETURN
```

```
360 :
365 REM T & S DER DATEN HOLEN
370 OPEN 4,8,4,"#":GOSUB 635
375 A$=CHR$(18):B$=CHR$(1):REM DIR
380 A=ASC(A$):B=ASC(B$)
385 PRINT#1,"U1";4;0;A;B:REM LESEN
390 GET#4,A$,B$:REM NAECHSTER SEKTOR
395 FOR I=0 TO 7:REM ANZAHL EINTRAEGE
400 : K=0
405 : PRINT#1,"B-P";4;I*32+3
410 : GET#4,C$,D$:REM T & S DER DATEN
415 : IF D$="" THEN D$=CHR$(0)
420 : FOR J=1 TO LEN(X$)
425 : GET#4,W$
430 : IF W$=MID$(X$,J,1) THEN K=K+1
435 : NEXT J
440 : IF K=LEN(X$) THEN 450:REM FOUND
445 NEXT I:GOSUB 635:GOTO 380
450 CLOSE 4
455 RETURN
460 :
465 REM TABELLE AENDERN
470 INPUT " QUELLCODE-NUMMER ";Q
475 IF Q<0 OR Q>255 THEN 470
480 PRINT:PRINT" ZUGEORDNETER GEOS-CODE:";ASC(Z$(Q))
485 NC=ASC(Z$(Q))
490 PRINT:PRINT " NEUER GEOS-CODE:";PRINT
495 INPUT "(RETURN=UEBERNEHMEN, 0=ENDE)";NC
500 IF NC<1 OR NC>255 THEN 510
505 Z$(Q)=CHR$(NC):GOSUB 610:GOTO 465
510 RETURN
515 :
520 REM TABELLE LADEN
525 INPUT" NAME DER TABELLE";NT$
530 OPEN 2,8,2,NT$+"",S,R":GOSUB 635
535 FOR I=0 TO 255
540 GET#2,Z$(I)
545 NEXT I
550 CLOSE 2
555 RETURN
560 :
565 REM TABELLE SPEICHERN
570 INPUT " NAME FUER DIE NEUE TABELLE";NT$
575 OPEN 2,8,2,NT$+"",S,W":GOSUB 635
580 FOR I=0 TO 255
585 PRINT#2,Z$(I);
```

```
590 NEXT I
595 CLOSE 2
600 RETURN
605 :
610 PRINT CHR$(147):REM CLR HOME
615 PRINT TAB(9);"**** TEXT-KONVERTER ****"
620 PRINT:PRINT
625 RETURN
630 :
635 INPUT#1,F,FT$,FT,FS
640 IF F<>0 THEN PRINT:PRINT " ";FT$:STOP
645 RETURN
650 :
655 RESTORE
660 FOR I=0 TO 255
665 READ Z:Z$(I)=CHR$(Z)
670 NEXT I
675 READ Z:IF Z=-1 THEN 685
680 PRINT:PRINT" FEHLER IN DATAS":STOP
685 FOR J=0 TO 23
690 READ K(J)
695 NEXT J
700 RETURN
705 :
710 REM DEKODIER-TABELLE
715 DATA 32,32,32,32,32,32,32,32,32,32
720 DATA 32,32,32,13,32,32,32,32,32,32
725 DATA 32,32,32,32,32,32,32,32,32,32
730 DATA 32,32,32,33,34,35,36,37,38,39
735 DATA 40,41,42,43,44,45,46,47,48,49
740 DATA 50,51,52,53,54,55,56,57,58,59
745 DATA 60,61,62,63,64,97,98,99,100,101
750 DATA 102,103,104,105,106,107,108,109,110,111
755 DATA 112,113,114,115,116,117,118,119,120,121
760 DATA 122,91,92,93,94,13,96,65,66,67
765 DATA 68,69,70,71,72,73,74,75,76,77
770 DATA 78,79,80,81,82,83,84,85,86,87
775 DATA 88,89,90,123,124,125,126,127,32,32
780 DATA 32,32,32,32,32,32,32,32,32,32
785 DATA 32,32,32,32,32,32,32,32,32,32
790 DATA 32,32,32,32,32,32,32,32,32,32
795 DATA 32,32,32,32,32,32,32,32,32,32
800 DATA 32,32,32,32,32,32,32,32,32,32
805 DATA 32,32,32,32,32,32,32,32,32,32
810 DATA 32,32,32,65,66,67,68,69,70,71
815 DATA 72,73,74,75,76,77,78,79,80,81
```

```

820 DATA 82,83,84,85,86,87,88,89,90,123
825 DATA 124,125,126,127,32,32,32,32,32,32
830 DATA 32,32,32,32,32,32,32,32,32,32
835 DATA 32,32,32,32,32,32,32,32,32,32
840 DATA 32,32,32,32,32,32,-1
845 :
850 REM TEXTKOPF-DATEN (TABS ETC.)
855 DATA 24,0,48,1,64,0,144,0,224,0
860 DATA 48,1,48,1,48,1,48,1,48,1
865 DATA 23,9,0,0

```

3.3.2 Die Bedienung des Konverters

Für spätere Anwendungen speichern Sie das Programm bitte nach dem Abtippen sicher auf einer Diskette ab ('SAVE "KONVERTER",8'). Bevor Sie nun eine erste Anpassung eines vorhandenen Textes vornehmen, müssen noch ein paar Vorbereitungen getroffen werden.

Erstellen Sie zuerst eine Diskette, die nur GEOWRITE enthält. Von dieser Diskette muß GEOWRITE nun geladen werden. Bei dem dabei erscheinenden Menü ist der Punkt 'CREATE NEW DOCUMENT' auszuwählen. Der neue Text muß unbedingt den Namen "DUMMY" bekommen, da unser Programm später danach suchen wird. Achten Sie dabei auf Großschreibung des gesamten Namens.

Sobald die weiße Schreibfläche erschienen ist, kann das Programm durch 'Verlassen' auch schon wieder verlassen werden. Der Sinn dieser ganzen Aktion besteht einfach darin, eine Datei mit passendem Icon auf der Diskette zu erzeugen. Hätten wir dieses durch BASIC erreichen wollen, wäre unser Programm erheblich länger geworden.

Als nächstes muß der Text, der auf das GEOS-Format gebracht werden soll, auf die Diskette kopiert werden. Dies kann entweder unter GEOS geschehen oder mittels eines anderen, beliebigen Kopiervorgangs. Da das Betriebssystem aber gerade nach dem DESK TOP fragt, bietet sich die erste Möglichkeit an.

Wenn auch dieser Punkt abgeschlossen ist, kann endlich unser KONVERTER geladen werden. Er ist standardmäßig auf Texte, die mit Textomat Plus entstanden sind, "abgerichtet". Natürlich lassen sich auch viele andere Textformate konvertieren. Wie man die nötigen Änderungen vornimmt, ist weiter unten beschrieben. Gestartet wird das Programm durch RUN. Nach ein paar Sekunden erscheint dann das folgende Menü:

- 1 = GEOS-TEXT ERSTELLEN
- 2 = TABELLE AENDERN
- 3 = TABELLE LADEN
- 4 = TABELLE SPEICHERN

Im Augenblick interessiert uns nur der erste Punkt. Wird er ausgewählt, erscheint die Frage nach dem Namen des zu konvertierenden Textes. Geben Sie ihn bitte genau so ein, wie er bei 'LOAD "\$",8:(LIST)' im Inhaltsverzeichnis erscheinen würde. Er kann auch durch das Sternchen abgekürzt werden.

Anschließend werden Sie gefragt, wie viele Zeichen später auf einer Seite erscheinen sollen. Normalerweise ist '1500' ein recht guter Wert. Bei längeren Texten muß man aber aufpassen, daß die maximale Anzahl von 63 Textseiten, die GEOWRITE zuläßt, nicht überschritten wird. Die benötigte Anzahl Seiten kann man folgendermaßen berechnen:

$s = (254 / \text{Anzahl Zeichen pro Seite}) * \text{Anzahl Blöcke des Textes, die im Inhaltsverzeichnis der Diskette für den Text angegeben werden.}$

Das Ergebnis ist auf ganze Seiten aufzurunden. Hierzu ein Beispiel: Ein Text belegt laut Directory insgesamt 42 Blöcke. Es sollen 165 Zeichen auf einer Seite Platz finden. Daraus ergibt sich für die Anzahl der benötigten Seiten:

$$s = (254/165)*42$$

Aufgerundet also 65 Seiten. Diese stehen aber nicht zur Verfügung, so daß unser Programm nach einer halben Stunde harter Arbeit lapidar melden würde: "Text zu lang". Abhilfe schafft natürlich die Erhöhung der Anzahl Zeichen pro Seite z.B. von 165 auf 170.

Um solchen lästigen Pannen vorzubeugen, empfiehlt sich diese Berechnung vor jeder Konvertierung längerer Texte. Wenn die letzte Eingabe abgeschlossen ist, geht es auch schon los: Der Text wird konvertiert und die Anzahl der vollbeschriebenen Blätter auf dem Bildschirm angezeigt. Sollten während der Arbeit Fehler in Verbindung mit der Diskettenstation auftreten, unterbricht das Programm und gibt die entsprechende Meldung aus.

Leider nimmt der Konvertierungsvorgang relativ viel Zeit in Anspruch, da sowohl das BASIC als auch der serielle Datenbus des C64 sehr langsam sind. Aber für solche Fälle haben Sie ja mit diesem spannenden Buch vorgesorgt.

Das Ende der Prozedur wird durch das Erscheinen des schon bekannten Hauptmenüs mitgeteilt. Nun steht der künstlerischen Ausgestaltung des Textes durch GEOWRITE - fast - nichts mehr im Wege. Eine kleine Sache, die aber nun die Dringlichkeitsstufe 1 besitzt, muß unter GEOS noch gemacht werden: Ein Validate!

Klicken Sie dazu im DESK TOP das Feld AUFRÄUMEN an. Sie finden es dort in dem Menü mit der Überschrift DISKETTE. Wenn dieser Vorgang unterlassen wird, kommt das einem Transport des neuen Textfiles in den Mülleimer gleich. Aus der Dokumentation des Programms wird die Ursache für diesen Sachverhalt deutlich. Jetzt erst kann der Startschuß für ein Laden des neuen Textfiles fallen. Selbstverständlich kann nun auch der Name wieder beliebig gewählt werden.

Wir möchten uns nun mit der Anpassung unseres Programms an andere Textverarbeitungsprogramme auseinandersetzen. Leider meint jede Computerfirma, sie müsse ihren eigenen 'genormten' ASCII-Code für die Verschlüsselung der Zeichen entwickeln. Commodore hat hier wohl mit der Vertauschung von Groß- und Kleinbuchstaben den Vogel abgeschossen. Zusätzliche Verwirrung stiftet auch die Benutzung von Codes, deren Wert größer ist als 127. Solche Zeichen tauchen nämlich im Standard-ASCII-Code nicht auf.

GEOS hält sich lobenswerterweise an den echten ASCII-Code. Daß sich dabei in Verbindung mit 'alten' Texten Probleme ergeben müssen, liegt auf der Hand. In vielen Fällen werden Sie mit der standardmäßigen Konvertierung, die unser Programm vornimmt, bei Ihren Texten Erfolg haben. Einzige Bedingung ist nur, daß der Text sequentiell (als SEQ- oder PRG-File) auf der Diskette vorliegt.

Wenn falsche Zeichen in GEOWRITE erscheinen, ist eine manuelle Anpassung der Konvertierungstabelle notwendig. Dazu haben wir in unserem Programm einige Hilfsfunktionen installiert. Sie gestatten - bei Kenntnis der vom jeweiligen Programm verwendeten Verschlüsselung - eine entsprechende Umkodierung. Benutzt beispielsweise ein Programm zur Darstellung des Buchstabens 'A' den Wert 97, ordnen wir einfach dieser 97 den echten ASCII-Wert 65 zu. Wie die echten ASCII-Werte aussehen, können Sie einer Abbildung entnehmen, die sich im Anhang befindet.

Durchgeführt werden diese Änderungen mittels des zweiten Programmpunktes (Tabelle ändern). Dabei wird zuerst nach der 'Quellcode-Nummer' gefragt. Das ist der dezimale Wert des falsch dargestellten Zeichens in Ihrem Text. Benutzt Ihr Text z.B. für ein "B" den Wert 98, muß hier die Zahl 98 eingegeben werden.

Anschließend teilt Ihnen das Programm mit, welcher echte ASCII-Code zur Zeit diesem Wert zugeordnet ist. Wenn hier eine 66 ausgegeben wird, ist das in Ordnung, da dies genau einem 'B' entspricht (RETURN drücken). Ansonsten kann nun die richtige Zuordnung getroffen werden. Wenn Sie alle in GEOWRITE falsch dargestellten Zeichen korrigiert haben, beendet eine Null die Umkodierung.

Damit Sie nun nicht jedesmal bei der Konvertierung Ihrer Texte diese Umkodierung von neuem vornehmen müssen, können Sie die gesamte Tabelle abspeichern. Hierfür ist der vierte Menüpunkt (TABELLE SPEICHERN) vorgesehen. Dabei muß der Ta-

belle ein beliebiger Name gegeben werden. Vor einer erneuten Arbeit kann dann mittels des dritten Menüpunktes (TABELLE LADEN) die Tabelle schnell wieder geladen werden.

Eine besondere Behandlung erfordern die deutschen Umlaute sowie das 'ß'. In dem Programm haben wir ab der Zeile 200 bis zur Zeile 250 eine "manuelle" Konvertierung speziell für Textomat Plus vorgesehen. Sollten bei der Darstellung der Umlaute Ihres Textes Schwierigkeiten auftauchen, muß an diesen Stellen auch eine Anpassung vorgenommen werden.

3.3.3 Dokumentation des Konverters

Für Interessierte erläutern wir hier die Funktionsweise unseres Konverters. Besondere Voraussetzungen für ein genaues Verständnis sind Kenntnisse in der Handhabung der Floppy-Kanäle und der Struktur von VLIR-Files. Um die einzelnen Aktionen der Konvertierung leichter verstehen zu können, gehen wir zuerst auf das verwendete Prinzip ein:

Ein Textfile besitzt unter GEOWRITE die Struktur VLIR. Es ist also auf der Diskette ein Linkerblock vorhanden, der auf die verschiedenen Tracks und Sektoren der Texte zeigt. Dabei besitzt jede Seite in GEOWRITE einen Eintrag in diesem Linkerblock.

Da das normale DOS aber keine VLIR-Files kennt, müssen wir über einen Umweg eine solche Struktur erzielen. Dazu dient der DUMMY-Text, der zu Beginn jeder Konvertierung erstellt werden muß. Anschließend müssen in den nun vorhandenen Linkerblock die Tracks und Sektoren jeder konvertierten Seite eingetragen werden. Dazu gehen wir folgendermaßen vor:

Für jede Seite erzeugen wir ein normales, sequentielles File, das wir "TEXT-DUMMY" nennen. Es erhält zusätzlich in dem Filenamen die entsprechende Seitenzahl als eindeutiges Erkennungsmerkmal für uns. In dieser Datei können wir nun leicht genau eine Textseite konvertiert ablegen. Die eigentliche Konvertierung erfolgt dabei über das Array Z\$. Anschließend holt

sich das Programm aus dem Directory den Track und Sektor dieser Datei und trägt die gefundenen Werte in den Linker des VLIR-Files ein. Dieser Vorgang wird solange wiederholt, bis kein Text mehr vorhanden ist.

Am Schluß werden alle "TEXT-DUMMYS" gelöscht. Da hierbei aber in der BAM der Diskette die belegten Blöcke dieser Files wieder freigegeben werden, diese aber in Wirklichkeit nicht frei sind, da sie noch im VLIR-File benötigt werden, ist ein Aufräumen der Diskette (Validate) vom DESK TOP aus unbedingt erforderlich. Eine vorhergehende Schreiboperation würde sonst mit Sicherheit Teile des neuen Textes löschen. Hier nun eine Beschreibung der einzelnen Programmschritte:

- 10-15 Arrays für die Konvertierungstabelle und den Kopf des GEOS-Textes anlegen. Der Kopf enthält Angaben über die Ränder und Tabulatoren.
- 25-65 Menüauswahl erstellen, Eingabe auswerten und einspringen; vorher Befehlskanal öffnen und Diskette initialisieren.
- 70-75 Befehlskanal und Programmschleife schließen.
- 80-120 Eingabe der Daten für Menü 1.
- 130 Die Datei des alten Textes wird geöffnet.
- 140 Evtl. noch vorhandene "TEXT-DUMMYS" werden von der Diskette gelöscht.
- 145 Ein "TEXT-DUMMY SZ" wird geöffnet. 'SZ' steht dabei für die auch im Programm verwendete aktuelle Seitenzahl. Zu Beginn ist SZ=0.
- 160-170 Auf jeder Seite müssen zuerst Angaben über die Ränder und Tabulatoren eingetragen werden.
- 185-275 In diesen Zeilen erfolgt die eigentliche Konvertierung. Die Daten aus Kanal 2 werden dazu konvertiert auf den Kanal 3 (TEXT-DUMMY) geschickt. P ist dabei der Zeichenzähler. Er wird bei jeder neuen Seite auf Null gesetzt. S enthält den Diskettenstatus und wird bei Erreichen des Textendes 1.

- 280-285 Hier wird in ein Unterprogramm verzweigt, das den Track und Sektor des ersten Datenblocks des Files X\$ in die Variablen C\$ bzw. D\$ übergibt. In diesem Fall werden T & S des aktuellen TEXT-DUMMYS geholt und in E\$ bzw. F\$ abgelegt.
- 290 In das gleiche Unterprogramm wird nun mit dem 'VLIR-Dummy' eingesprungen. Dadurch wird der T & S des Linkerblocks in C\$ bzw. D\$ geholt.
- 295-320 Der Linkerblock wird in einen Floppypuffer eingelesen und der T & S des TEXT-DUMMYS an der richtigen Stelle eingetragen. Anschließend wird der Block zurückgeschrieben.
- 325-345 Auswertung der Seitenzahl 'SZ'. Ist der Quelltext zu Ende (S=1), ist die Konvertierung abgeschlossen.
- 350-360 Die TEXT-DUMMYS werden wieder gelöscht. Zurück zum Hauptmenü.
- 366-455 Unterprogramm:
Der Track und Sektor des ersten Datenblocks des Files in X\$ wird aus dem Directory in C\$ bzw. D\$ geholt. Dazu wird jeweils eine Directory-Seite in den Floppy-Puffer eingelesen und der gesuchte Filename mit allen acht Einträgen dieses Blocks verglichen. Wenn alle Zeichen übereinstimmen, ist das File gefunden. Die gesuchten Daten stehen vor dem Filenamen und werden daher auch vor dem Namensvergleich eingelesen (Zeile 410).
- 465-510 Hier steht das Änderungsprogramm für die Konvertierungstabelle.
- 520-550 Die Daten der Konvertierungstabelle werden hier eingelesen.
- 565-600 Speicherroutine für die Konvertierungstabelle
- 610-625 Die Meldung des Programms wird ausgegeben.
- 635-645 Sollte ein Fehler mit der Floppy auftreten, wird dieser hier ausgegeben und das Programm angehalten.

- 655-670 Die Konvertierungstabelle wird angelegt.
- 685-700 Die Daten für den Textkopf werden geholt.
- 710-840 Dekodiertabelle 'Textomat Plus'
- 850-865 Daten für den Textkopf (jeweils zwei Werte gehören zusammen):
linker Rand, rechter Rand, acht Tabulatoren, Angaben über den verwendeten Schrifttyp

3.4 Der Filemaster

Wenn Sie einige Zeit mit GEOS gearbeitet haben, wird es Sie vielleicht stören, daß alle Ihre eigenen Programme (BASIC oder Assembler) nur das einfache C64-Symbol erhalten und Sie im INFO-Bildschirm nicht Ihren Namen als Programmierer eintragen können. Vielleicht möchten Sie Ihrem Freund ein selbst erstelltes Haushaltsprogramm kopieren, und er soll stets daran erinnert werden, daß dieses Programm von Ihnen ist. Vielleicht hätten Sie auch eine tolle Idee für ein Icon, mit dem Sie dieses Programm auch optisch als Haushaltsprogramm markieren könnten, doch leider gibt es unterschiedliche Icons nur bei GEOS-Programmen, nicht aber für die selbst erstellten Files. Es gibt eine Lösung für Ihr Problem:

Als wir bei unserer Arbeit mit GEOS feststellten, welche fast unglaublichen Möglichkeiten dieses Programm enthält, haben wir immer wieder nach Mitteln und Wegen gesucht, um Ihnen als Anwender diese Möglichkeiten zugänglich zu machen. Wir könnten Sie nun auf das Kapitel GEOS INTERN verweisen, in dem Sie ausführlich nachlesen können, wie der INFO-Bildschirm unter GEOS aussieht und wie man ihn mit einem Disketten-Monitor erstellen kann. Vielleicht haben Sie aber gar keinen Disketten-Monitor, oder Sie haben noch gar keine Erfahrung im Umgang mit solch einem Werkzeug. Dann wären Sie an dieser Stelle sicherlich enttäuscht von diesem Buch. Wir haben uns deshalb die Arbeit gemacht, ein Programm zu schreiben, mit dem Sie auf sehr komfortable Weise eigene Files in das GEOS-Format umwandeln können: den Filemaster.

Bevor wir ihn aber vorstellen, möchten wie Ihnen ganz kurz und einfach erklären, wie ein eigenes Programm in das GEOS-Format umgewandelt wird. Für eine detailliertere und umfassendere Beschreibung des GEOS-Formats lesen Sie bitte den entsprechenden Abschnitt in Kapitel 4: GEOS INTERN. Normalerweise enthält das Inhaltsverzeichnis der Diskette für jedes File einen File-Eintrag mit folgenden Informationen:

- Filetyp
- Track und Sektor des ersten Datenblocks
- Name des Files, ergänzt mit 'Shift+Space' (ASCII=\$A0)
- Anzahl der Sektoren des Files

Genauere Informationen stehen in Ihrem Floppy-Handbuch. Dabei reserviert die Floppy zwar 30 Byte (=Speicherplätze) pro File-Eintrag, benutzt aber wesentlich weniger. Die anderen werden einfach auf '0' gesetzt. Hier greift GEOS ein und speichert an dieses Stellen zusätzliche Informationen:

- GEOS-Filetyp
- Track und Sektor des INFO-Bildschirms
- Datum und Uhrzeit.

Der INFO-Bildschirm enthält im wesentlichen die Texte, die unter INFO vom DESK TOP angezeigt werden, und das Aussehen des ICONs. Um also Ihr Programm ins GEOS-Format umzuwandeln, muß ein freier Sektor für den INFO-Bildschirm gesucht werden, dann muß der File-Eintrag im Inhaltsverzeichnis mit den zusätzlichen Informationen versehen werden. Anschließend wird das Icon erstellt und zusammen mit den Texteinträgen abgespeichert.

Alle diese Aufgaben erledigt dabei der Filemaster für Sie. Das Icon hat das gleiche Format wie ein Sprite, das Sie vielleicht schon aus Ihrem C64-Handbuch kennen. Daher enthält der Filemaster auch einen Sprite Generator.

3.4.1 Das Programm "Filemaster"

```

10 REM " FILE - MASTER "
15 REM
20 REM "**** BY MANFRED TORNSDORF ****"
25 REM
30 PRINTCHR$(14)
40 FOR I=0 TO 7:READ SP$(I):NEXT I
60 PRINT CHR$(147)
80 V=53248:POKEV+21,0
100 PRINT CHR$(147);"DISKETTE EINLEGEN + TASTE DRUECKEN"
120 GETA$:IF A$="" THEN 120
140 OPEN5,8,15,"I"
160 OPEN3,8,3,"#0"
180 PRINT CHR$(147):PRINT
200 PRINT" FILE-MASTER"
220 PRINT" -----"
230 PRINT
240 PRINT" WANDELT FILES INS GEOS-FORMAT"
260 PRINT:PRINT
280 PRINT" INHALTSVERZEICHNIS = I"
300 PRINT" SPRITE LADEN = SL"
320 PRINT" SPRITE ERSTELLEN = SE"
330 PRINT
340 PRINT" FILE EINLESEN = FE"
380 PRINT" FILE ZEIGEN = FZ"
400 PRINT" GEOS-FILE ERZEUGEN = GEOS"
420 PRINT" DATUM EINGEBEN/AENDERN = FD"
430 PRINT
440 PRINT" DISKETTE WECHSELN = DW"
460 PRINT" PROGRAMM BEENDEN = ENDE"
480 PRINT" INFO EINLESEN = INFO"
500 N$="XXXXXXX": PRINT
520 INPUT" EINGABE";N$
540 IF N$="I" THEN GOSUB 4060
560 IF N$="SL" THEN GOSUB 4060
580 IF N$="SE" THEN GOSUB 2320
600 IF N$="FE" THEN GOSUB 800
620 IF N$="FD" THEN GOSUB 3160
640 IF N$="FZ" AND Z1=1 THEN PRINTCHR$(147):GOSUB 980
660 IF N$="INFO" THEN GOSUB 5100
680 IF N$="DW" THEN CLOSE3:CLOSE5:GOTO100
700 IF N$="ENDE" THEN 760
720 IF N$="GEOS" THEN GOSUB 3540
740 GOTO 180
760 CLOSE3:CLOSE5:POKEV+21,0:END

```

```

780 STOP:REM*****
800 Z1=0:GOSUB 1380 :IF T=0 OR Z1=0 THEN RETURN
810 FOR I=1 TO 500:NEXTI:PRINTCHR$(147)
820 AT=T: REM DIR-TRACK GEFUNDEN
840 AS=S: REM DIR SEKTOR GEFUNDEN
860 AB=BP:REM BUFFERPOINTER FILETYP
880 TF=TS:SF=SS:REM TRACKFILE,SEKTORFILE
900 FI$=NA$ :
920 GOSUB2220:FS=A:GOSUB2220:GT=A:GOSUB2220:JA=A:JA$=STR$(A)
930 GOSUB2220:MO=A:MO$=STR$(A)
940 GOSUB2220:TA=A:TA$=STR$(A):GOSUB2220:HO=A:HO$=STR$(A)
950 GOSUB2220:MI=A:MI$=STR$(A)
960 GOSUB2220:LL=A:GOSUB2220:LH=A:
980 PRINTCHR$(145)CHR$(145)" ";FI$;CHR$(145);CHR$(145)
1000 PRINT"STARTTRACK  :";TF
1020 PRINT"STARTSEKTOR  :";SF
1040 PRINT
1060 PRINT"FILETYP      :";FT
1080 PRINT"INFO TRACK   :";IT
1100 PRINT"INFO SEKTOR  :";IS
1120 PRINT#5,"B-P:";3;BP+21
1140 PRINT"FILESTRUKTUR:";FS
1160 PRINT"GEOS F-TYP  :";GT;
1180 IF GT=1 THEN PRINT" =BASIC"
1181 IF GT=2 THEN PRINT" =ASSEMBLER"
1182 IF GT=3 THEN PRINT" =DATA FILE"
1183 IF GT=4 THEN PRINT" =SYSTEM FILE"
1184 IF GT=5 THEN PRINT" =ACCESSORY "
1185 IF GT=6 THEN PRINT" =APPLICATION"
1186 IF GT=7 THEN PRINT" =APPLICATION DATA"
1187 IF GT=8 THEN PRINT" =FONT FILE "
1188 IF GT=9 THEN PRINT" =PRINTER DRIVER"
1189 IF GT=10 THEN PRINT" =INPUT DRIVER"
1190 IF GT=0 THEN PRINT"KEIN GEOS FILE"
1220 IF GT< 0 AND GT >10 THEN PRINT
1240 PRINT"JAHR      :";JA
1260 PRINT"MONAT     :";MO
1280 PRINT"TAG       :";TA
1300 PRINT"STUNDE     :";HO
1320 PRINT"MINUTE    :";MI
1340 PRINT"FILELAENGE  :";LL+256*LH
1350 PRINT
1360 INPUT " WEITER";X$:RETURN
1380 REM *****
1400 PRINTCHR$(147):T=18:S=1:
1405 PRINT

```

```

1410 PRINT
1420 INPUT" GESUCHTER NAME:";SU$
1440 SU=LEN(SU$)
1460 PRINTCHR$(147)
1480 SV$="":FORI=1TOSU: SV=ASC(MID$(SU$,I,1)):IF SV>192 ANDSV<219 THEN
SV=SV-96
1500 SV$=SV$+CHR$(SV):NEXTI:
1520 PRINT#5,"B-R:"3;0;T ;S
1540 PRINT#5,"B-P:"3,0
1560 BP=2
1580 GOSUB 2220
1600 TN$=A$:TN=ASC(TN$)
1620 GOSUB 2220
1640 NA$=""
1660 SN$=A$:SN=ASC(SN$):IFN$<>"I"THEN PRINT"T=";T;" S=";S
1680 PRINT#5,"B-P:"3;2
1700 FOR J=1 TO 8
1720 PRINT#5,"B-P:"3;BP
1740 GOSUB 2220:FT$=A$:FT=ASC(FT$)AND 63
1760 GOSUB 2220:T$=A$:T=ASC(T$)
1780 GOSUB 2220:SS$=A$:SS=ASC(SS$)
1800 FOR I=1 TO 16:GOSUB2220
1820 IF ASC(A$)=160 THEN I=16:GOTO 1860
1840 NA$=NA$+A$:
1860 NEXTI:
1880 PRINT#5,"B-P:"3;BP+19
1900 GOSUB2220:IT$=A$:IT=ASC(IT$):
1920 GOSUB2220:ISS$=A$:IS=ASC(ISS$):
1940 IF FT=0 THEN 2080
1980 PRINT" ";FT;" ";T$;" ";SS;" ";CHR$(34);NA$;CHR$(34);
2000 PRINT" ";IT;" ";IS
2020 IF N$="I" THEN 2080
2030 PRINT
2040 IF SU$=MID$(NA$,1,SU) THEN PRINT" GEFUNDEN":Z1=1:RETURN
2060 IF SV$=MID$(NA$,1,SU) THEN PRINT" GEFUNDEN":Z1=1:RETURN
2080 NA$="":
2100 IF PEEK(203)<>64 THEN 2180
2120 BP=BP+32
2140 NEXTJ
2160 IF TN<>0 THEN PRINT" ":T=TN:S=SN:GOTO1520
2180 RETURN
2200 REM*****
2220 GET#3,A$:IFA$=""THENA$=CHR$(0)
2240 A=ASC(A$)
2260 RETURN
2280 REM*****

```

```

2300 FORK=0TO62:POKE832+K,0:NEXTK
2320 PRINTCHR$(147):PRINT"SPRITEMASTER"
2360 X1=0 :Y1=0:X=X1:Y=Y1:PS=43
2380 VX=0:VY=0:VT=0 :
2400 POKE2040,13:POKEV+21,1:POKEV,30:POKEV+1,200
2420 FOR J=X1+14 TO X1+23+14
2440 FOR I=Y1+1 TO Y1+21
2460 P=1024+I*40+J
2480 GOSUB 3000:VA%=VI%ANDVN%
2500 IF VA% <>0 THEN PS=42
2520 IF VA% = 0 THEN PS=46
2540 IF P>1024 AND P<2023 THEN POKEP,PS
2560 Y=Y+1:NEXT I : Y=0
2580 X=X+1:NEXTJ:X=0
2600 PS=43
2620 P=1024+(Y+1)*40+X+14
2640 PR=PEEK(P)
2660 IFP>1023ANDP<2024 THEN POKEP,PS
2680 GETA$:IF A$="" THEN2680
2700 IF A$=CHR$(29) ANDX<23 THENX=X+1
2720 IF A$=CHR$(157)ANDX>0 THENX=X-1
2740 IF A$=CHR$(17) ANDY<20 THENY=Y+1
2760 IF A$=CHR$(145)ANDY>0 THENY=Y-1
2780 IF A$="" THEN PR=42:
2800 IF A$=" " THEN PR=32:
2820 IF A$="E" THEN 3120
2840 IF A$=CHR$(147) THEN 2300
2860 POKEP,PR
2880 IF A$<>"*"AND A$<>" " THEN2620
2900 GOSUB 3000
2920 IF A$="" THEN POKEVB%,VI% OR VN%
2940 IF A$=" " THEN POKEVB%,VI%ANDSP%(VT%)
2960 GOTO 2620
2980 REM*****
3000 VY%=Y :VX%=INT((X)/8):VT%=X-VX%*8
3020 VB%=832+VX%+VY%*3 :REM BYTENUMMER
3040 VI%=PEEK(VB%) :REM INHALT
3060 VN%=2^(7-VT%) :REM NEUES BIT
3080 RETURN
3100 REM *****
3120 POKEV+21,0:PRINTCHR$(147):RETURN
3140 REM ***** DATUM ERSTELLEN ***
3150 PRINTCHR$(147)
3160 PRINTCHR$(147):PRINT" -----D A T U M-----"
3180 PRINT" ";TAS;". ";MOS;".19";JAS;" ";HOS;": ";MIS;" UHR"
3220 PRINT:PRINT:PRINT

```

```

3240 PRINT" JAHR      :";JA
3260 PRINT" MONAT    :";MO
3280 PRINT" TAG      :";TA
3300 PRINT" STUNDE   :";HO
3320 PRINT" MINUTE   :";MI
3330 PRINT
3340 PRINT" AENDERN = 2 ANF. BUCHST"
3360 PRINT" FERTIG  = ENDE"
3370 PRINT:PRINT
3380 INPUT " EINGABE";X$
3400 IF X$="ENDE" THEN RETURN
3410 PRINT:PRINT
3420 IF X$="JA" THEN INPUT" JAHR      :";JA$:JA=VAL(JA$):GOTO3160
3440 IF X$="MO" THEN INPUT" MONAT    :";MO$:MO=VAL(MO$):GOTO3160
3460 IF X$="TA" THEN INPUT" TAG      :";TA$:TA=VAL(TA$):GOTO3160
3480 IF X$="ST" THEN INPUT" STUNDE   :";HO$:HO=VAL(HO$):GOTO3160
3500 IF X$="MI" THEN INPUT" MINUTE   :";MI$:MI=VAL(MI$):GOTO3160
3520 GOTO 3160
3540 PRINTCHR$(147) :REM *****INFO FILE ERSTELLEN**
3560 IF IT<>0 THEN 3660 :REM SOLL GESETZT WERDEN
3580 IT=TF:IS=SF:PRINTIT,IS
3600 IF IT=18 THEN IT=19 :REM DIR-TRACK NICHT BENUTZEN
3610 PRINT#5,"B-A:"0;IT;IS
3620 INPUT#5,A,B$,C,D:PRINTA,B$,C,D:
3640 IF C<> 0 THEN IT=C:IS=D:GOTO 3600
3650 PRINT:PRINT:PRINT
3660 PRINT" BITTE GEOS-FILETYP EINGEBEN:"
3670 PRINT:
3680 INPUT" FILETYP";X$:X=VAL(X$)
3700 IF X< 1 AND X>15 THEN 3660
3720 FS=0:GT=X
3740 PRINT#5,"B-F:"0;IT;IS
3760 PRINT#5,"B-R:"3;0;AT;AS
3780 PRINT#5,"B-P:"3,AB+19
3800 PRINT#3,CHR$(IT);
3820 PRINT#3,CHR$(IS);
3840 PRINT#3,CHR$(FS);
3860 PRINT#3,CHR$(GT);
3880 PRINT#3,CHR$(JA);
3900 PRINT#3,CHR$(MO);
3920 PRINT#3,CHR$(TA);
3940 PRINT#3,CHR$(HO);
3960 PRINT#3,CHR$(MI);
3970 IF AT<1 OR AT>35 THEN PRINT" FEHLER:UNGUELTIGER TRACK":GOTO4040
3980 PRINT#5,"M-W"CHR$(6)CHR$(0)CHR$(2)CHR$(AT)CHR$(AS)
4000 PRINT#5,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$(144)

```

```

4020 PRINT:PRINT "      GESCHRIEBEN":
4040 INPUT " WEITER";X$:RETURN
4060 REM ***DIR UND SPRITE SUCHEN***
4080 PRINTCHR$(147) :T=18:S=1:IFN$="I"THEN 4200
4090 PRINT:PRINT:PRINT
4100 INPUT"  GESUCHTER NAME:";SU$
4120 SU=LEN(SU$)
4140 PRINTCHR$(147)
4160 SV$="":FORI=1TOSU:SV=ASC(MID$(SU$,I,1)):IF SV>192ANDSV<219THEN
SV=SV-96
4180 SV$=SV$+CHR$(SV):NEXTI:
4200 PRINT#5,"B-R:";3;0;T ;S
4220 PRINT#5,"B-P:";3;0
4240 BP=2
4260 GOSUB 2220
4280 TN$=A$:TN=ASC(TN$)
4300 GOSUB 2220
4320 NA$=""
4340 SN$=A$:SN=ASC(SN$):IFN$<>"I"THEN PRINT"T=";T;" S=";S
4360 PRINT#5,"B-P:";3;2
4380 FOR J=1 TO 8
4400 PRINT#5,"B-P:";3;BP
4420 GOSUB 2220:F1$=A$:F1=ASC(F1$)AND 63
4440 GOSUB 2220:
4460 GOSUB 2220:
4480 FOR I=1 TO 16:GOSUB2220
4500 IF ASC(A$)=160 THEN I=16:GOTO 4540
4520 NA$=NA$+A$:
4540 NEXTI:
4560 PRINT#5,"B-P:";3;BP+19
4580 GOSUB2220:I1$=A$:I1=ASC(I1$):
4600 GOSUB2220:I2$=A$:I2=ASC(I2$):
4620 IF F1=0 THEN 4740
4640 IF N$="I" THEN PRINT" ";NA$:GOTO4740
4660 PRINT" "; CHR$(34);NA$;CHR$(34)
4680 IF N$="I" THEN 4740
4700 IF SU$=MID$(NA$,1,SU) THEN PRINT"      GEFUNDEN":GOTO 4900
4720 IF SV$=MID$(NA$,1,SU) THEN PRINT"      GEFUNDEN":GOTO 4900
4740 NA$="":REM N$=""
4760 IF PEEK(203)<>64 THEN 4860
4780 BP=BP+32
4800 NEXTJ
4820 IF TN<>0 THEN PRINT" ":T=TN:S=SN:GOTO4200
4840 IF N$<>"I"THEN PRINT"      LEIDER NICHT GEFUNDEN"
4860 INPUT" WEITER";X$:RETURN
4900 REM *** SPRITE EINLESEN ***

```

```
4910 PRINTCHR$(147);
4920 PRINT"          SPRITE LESEN";CHR$(145)
4940 IF I1=0 THEN PRINT "KEIN SPRITE!":FOR I=0 TO 1000:NEXTI:RETURN
4960 PRINT#5,"B-R:"3;0;I1 ;I2
4980 PRINT#5,"B-P:"3,5
5000 FOR I=0 TO 62
5020 GOSUB 2220
5040 POKE832+I,A:PRINTI,A;"|"
5060 NEXTI:
5080 RETURN
5100 REM***INFO-SCHIRM EINLESEN*****
5120 IF IT=0 THEN RETURN:REM TRACK=0
5140 PRINT#5,"B-R:"3;0;IT;IS
5160 PRINT#5,"B-P:"3,0
5180 FORI=1TO6:A$(I,2)="" :NEXTI
5200 GOSUB2220:IFA<> 0 THEN 5600
5205 GOSUB2220:IFA<>255 THEN 5600
5210 GOSUB2220:IFA<> 3 THEN 5600
5215 GOSUB2220:IFA<>21 THEN 5600
5220 PRINT#5,"B-P:"3,71
5240 GOSUB 2220:X=A
5260 GOSUB 2220:X=X+256*A
5280 A$(1,2)=STR$(X)
5300 GOSUB 2220:X=A
5320 GOSUB 2220:X=X+256*A
5340 A$(2,2)=STR$(X)
5360 GOSUB 2220:X=A
5380 GOSUB 2220:X=X+256*A
5400 A$(3,2)=STR$(X)
5420 GOSUB 2220:IFA=0 THEN 5460
5440 A$(4,2)=A$(4,2)+A$:GOTO 5420
5460 PRINT#5,"B-P:"3,97
5480 GOSUB 2220:IFA=0 THEN 5520
5500 A$(5,2)=A$(5,2)+A$:GOTO 5480
5520 PRINT#5,"B-P:"3,160
5540 GOSUB 2220:IFA=0 THEN 5580
5560 A$(6,2)=A$(6,2)+A$:GOTO 5540
5580 REM
5600 REM **** INFOSCHIRM ERSTELLEN*****
5620 A$(1,1)=CHR$(17)+" 1 STARTADRESSE : "
5640 A$(2,1)=CHR$(17)+" 2 ENDADRESSE : "
5660 A$(3,1)=CHR$(17)+" 3 INITIALISIERUNG: "
5680 A$(4,1)=CHR$(17)+" 4 PROGRAMMNAME : "
5700 A$(5,1)=CHR$(17)+" 5 PERS. NAME : "
5720 A$(6,1)=CHR$(17)+" 6 HILFSTEXT : "
5740 REM*****
```

```

5760 PRINTCHR$(147)
5770 PRINT"      INFO-BILDSCHIRM ERSTELLEN"
5780 FOR I =1 TO 6:PRINTA$(I,1);A$(I,2):NEXTI
5790 PRINT:PRINT
5800 INPUT"  ZAHL/ENDE EINGEBEN";X$
5820 IF X$="ENDE" THEN PRINTCHR$(147) :GOTO6000
5840 IF VAL(X$)<1 OR VAL(X$)>6THEN5760
5850 PRINT:PRINT
5860 PRINT" ";A$(VAL(X$),1);" ";A$(VAL(X$),2)
5880 PRINTTAB(21);CHR$(145);
5900 IF LEN(A$(VAL(X$),2))>19 THENPRINTCHR$(145);
5920 INPUT A$(VAL(X$),2)
5940 GOTO 5760
5960 REM
5980 REM
6000 PRINT#5,"B-P:"3,0
6020 PRINT#3,CHR$(0);CHR$(255);CHR$(3);CHR$(21);CHR$(191);
6040 FOR I=0 TO 62
6060 PRINT#3,CHR$(PEEK(832+I));
6080 NEXTI
6100 PRINT#3,CHR$(FT+128);
6120 PRINT#3,CHR$(GT);
6140 PRINT#3,CHR$(FS);
6160 IF GT=1 THEN GOTO 6340
6180 REM X$=A$(1,2):GOSUB6720
6200 REM PRINT#3,CHR$(X2);CHR$(X1);
6220 X$=A$(1,2):GOSUB6720
6240 PRINT#3,CHR$(X2);CHR$(X1);
6260 X$=A$(2,2):GOSUB6720
6280 PRINT#3,CHR$(X2);CHR$(X1);
6300 X$=A$(3,2):GOSUB6720
6320 PRINT#3,CHR$(X2);CHR$(X1);
6340 X=LEN(A$(4,2)):IF X>19 THEN X=19
6360 PRINT#5,"B-P:"3,77
6380 FOR I=1 TO X:X$=MID$(A$(4,2),I,1):PRINT#3,X$;
6400 NEXTI :PRINT#3,CHR$(0);
6420 X=LEN(A$(5,2)):IF X>19 THEN X=19
6440 PRINT#5,"B-P:"3,97
6460 FOR I=1 TO X:X$=MID$(A$(5,2),I,1):PRINT#3,X$;
6480 NEXTI :PRINT#3,CHR$(0);
6500 X=LEN(A$(6,2)):IF X>40 THEN X=40
6520 PRINT#5,"B-P:"3,160
6540 FOR I=1 TO X:X$=MID$(A$(6,2),I,1):PRINT#3,X$;
6560 NEXTI :PRINT#3,CHR$(0);
6570 IF IT<1 OR IT>35 THEN PRINT" FALSCHER TRACK":GOTO6680
6580 PRINT#5,"M-W"CHR$(6)CHR$(0)CHR$(2)CHR$(IT)CHR$(IS)

```

```

6600 PRINT#5,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$(144)
6620 PRINT"      GESCHRIEBEN":
6640 PRINT#5,"B-A:"0;IT;IS
6660 INPUT#5,A,B$,C,D:PRINTA,B$,C,D:
6680 INPUT"  WEITER";X$
6700 RETURN
6720 X=VAL(X$):X1=INT(X/256):X2=INT(X-X1*256 )
6740 RETURN
6760 DATA127,191,223,239,247,251,253,254

```

3.4.2 Erläuterung des Programms

Wenn Sie den Filemaster nur einfach anwenden wollen, so können Sie diesen Abschnitt übergehen. Sollten Sie jedoch Änderungen vornehmen wollen oder Teile des Filemasters in eigenen Programmen verwenden wollen (Inhaltsverzeichnis ohne Programmverlust), so wird Ihnen dieser Abschnitt dabei sicherlich helfen. Sie sollten aber schon eigene Programme in BASIC erstellt und die üblichen Befehle kennengelernt haben.

Tip: Das Dollarzeichen entspricht dem Leiterzeichen über der '3')

Zeile	Erklärung
40	Einlesen von acht Werten in das Feld SP%. Diese Werte werden vom Spritemaster dazu benutzt, einen einzelnen Punkt des Sprites zu löschen.
60	Löschen des Bildschirms
80	V=Startadresse des Video-Controllers. Register 21 schaltet die Sprites ein oder aus. Mit '0' werden alle Sprites ausgeschaltet.
100-120	Wartet, bis eine Taste gedrückt wurde. Ermöglicht während des Programmablaufs einen Diskettenwechsel.

- 140 Öffnet den Befehlskanal der Diskette und initialisiert sie. Dadurch wird die Disketten-BAM in den Puffer des Laufwerks geschrieben. Näheres erfahren Sie in Ihrem Floppy-Handbuch.
- 160 Öffnet einen Kanal zur Floppy. Diesem Kanal wird der Puffer 0 (\$0300-\$03FF) zugewiesen.
- 180-480 Gibt das Menü auf dem Bildschirm aus.
- 500 Die Eingabevariable N\$ wird auf einen Wert gesetzt, der im Menü nicht vorkommt. Damit wird verhindert, daß durch versehentliches Drücken der Return-Taste ein Befehl erneut ausgeführt wird.
- 520-720 Eingabe und Sprung in das entsprechende Unterprogramm.
- 740 Nach der Rückkehr aus dem Unterprogramm oder falls kein gültiger Befehl eingegeben wurde, wird stets wieder das Menü ausgegeben.
- 760 Wird durch Eingabe von "ENDE" angesprungen. Schließt beide Kanäle und stoppt das Programm. Ein Neustart ohne Datenverlust ist mit GOTO 100 möglich.
- 800- Unterprogramm: File suchen und Daten anzeigen.
- 800 Z1=0 bedeutet, daß noch kein File-Eintrag gefunden worden ist. Erst nach erfolgreicher Suche im Unterprogramm ab Zeile 1380 wird Z1 auf 1 gesetzt. Wenn Z1=0 ist, bricht das Unterprogramm an dieser Stelle ab und kehrt zum Menü zurück. Das gleiche gilt für T=0, das heißt, wenn der aktuelle Track=0 ist. Das dient der Sicherheit, da ein Leserversuch mit T=0 zu einer Fehlermeldung der Floppy führt.
- 820 In dieser Variablen wird der Track gespeichert, indem der gefundene File-Eintrag steht. Eigentlich ist diese Variable überflüssig, da das Inhaltsverzeichnis der Diskette immer auf Spur (=Track) 18 liegt. Aus Gründen der Übersichtlichkeit wurde dennoch eine Variable angelegt.

- 830 Sektor des gefundenen File-Eintrags
- 860 Für jeden Puffer der Floppy gibt es einen Zeiger, der auf das nächste zu lesende Byte zeigt. AB enthält die Position des ersten Zeichens im gefundenen File-Eintrag (Das ist der Filetyp).
- 880 Der Starttrack des Programms wird in TF gesichert, ebenso der Startsektor in SF.
- 900 Die Variable Na\$ enthält den Namen des zuletzt eingelesenen File-Eintrags. Dieser vollständige Name wird in die Variable FI\$ übernommen. Wenn Sie also den File-Eintrag 'ALARM CLOCK' suchen lassen, reicht es, 'ALA' einzugeben. Trotzdem wird dann der vollständige Name 'ALARM CLOCK' angezeigt.
- 920-960 Die zusätzlichen Daten, die GEOS hinter dem Filenamen abspeichert, werden eingelesen. Dabei holt das Unterprogramm 2220 jeweils ein Zeichen und den zugehörigen ASCII-Wert und übergibt beide in A\$ und A.
- 980-1340 Teilprogramm, das die Daten des File-Eintrags zeigt. Alle Informationen des File-Eintrags werden angezeigt. Bei einem File-Eintrag, der nicht durch GEOS erstellt wurde, sind allerdings die meisten Werte =0. Die Variablen JA, MO, TA, HO, MI enthalten das Datum und die Uhrzeit. Leider konnte als Variable für die Stunden nicht die einleuchtendere Abkürzung ST verwendet werden, da dies eine vom C64 für eigene Zwecke verwendete Variable ist.
- 1360 Wartet, bis Sie alle Informationen gelesen haben, und kehrt dann zum Menü zurück.
- 1380- Unterprogramm: File-Eintrag suchen
- 1400 Bildschirm löschen und den aktuellen Track, Sektor auf 18,1 setzen.

- 1420-1500 Der Name des gesuchten File-Eintrags wird in `SUS` abgelegt. Gleichzeitig wird er in der Variablen `SV$` in geänderter Form gespeichert. Da GEOS die Zeichen unüblich codiert, könnten alle GEOS-File-Einträge nicht gefunden werden.
- 1520 Holt den aktuellen Track, Sektor in den Puffer 0.
- 1540 Setzt den Pufferzeiger auf den Anfang des Puffers. Dadurch können Track und Sektor des nächsten Blocks der Directory gelesen und gespeichert werden.
- 1560 Setzt die Variable für den Pufferzeiger auf den Anfang des ersten File-Eintrags.
- 1580-1600 In `TN` wird der Starttrack des nächsten Blocks gespeichert
- 1620-1660 Speichert den Startsektor in `SN`. Der aktuelle Filenamen wird gelöscht (`NA$=""`).
- 1680 Setzt den Pufferzeiger auf den Wert `BP`, in diesem Fall also auf den Anfang des ersten File-Eintrags (`BP=2`).
- 1700 Für die acht File-Einträge eines Sektors im Inhaltsverzeichnis.
- 1720 Pufferzeiger auf errechneten Wert. Dieser Wert wird 2120 berechnet.
- 1740 Holt den Filetyp. Da immer das erste Bit gesetzt ist (OR 128) und das zweite Bit (OR 64), falls das File vor Löschen geschützt ist, wird `FT` mit 63 AND-verknüpft. Dadurch werden die beiden ersten Bits nicht angezeigt.
- 1760-1780 Holt Track und Sektor des ersten Datenblocks. Dort beginnt also das eigentliche Programm, das zu diesem File-Eintrag gehört.
- 1800 Für die 16 möglichen Zeichen des File-Eintrags: Hole ein Zeichen.

- 1820 Wenn das Zeichen ein 'SHIFT + SPACE' (ASCII=160) ist, dann ist der Name zu Ende.
- 1840 Setzt den Filenamen aus den einzelnen Zeichen zusammen.
- 1860 Schleifenende
- 1880 Der Pufferzeiger wird auf das erste Zeichen hinter dem Namen gesetzt. Ab dort legt GEOS die weiteren Informationen ab.
- 1900-1920 Track und Sektor des INFO-Bildschirms aus dem File-Eintrag holen.
- 1940 Wenn FT=0 ist (Filetyp), dann ist das File gelöscht worden. Gelöschte Files sollen natürlich nicht angezeigt werden. An dieser Stelle können Sie das Programm erweitern, wenn Sie eine Möglichkeit benötigen, gelöschte Programme zu retten. Setzen Sie dann statt dieser Zeile ein: IF FT=0 then FT=2, und Ihr Programm ist, falls es noch nicht durch Abspeichern neuer Programme überschrieben wurde, wieder vorhanden. Speichern Sie es aber dann sofort auf eine andere Diskette.
- 1980-2000 Der Name und die wichtigsten Informationen werden auf dem Bildschirm ausgegeben. Dadurch können Sie das Unterprogramm bei der Suche beobachten und sofort feststellen, ob der gewünschte File-Eintrag vielleicht auf Grund falscher Schreibweise nicht gefunden wurde.
- 2040 Prüft, ob der augenblickliche Name mit dem gesuchten übereinstimmt. Dabei ist nur die Länge des eingegebenen Namens (SU) maßgebend. Bei Übereinstimmung wird Z1=1, und das Unterprogramm ist beendet.
- 2060 Prüft ebenfalls für die geänderten Zeichen eines GEOS-File-Eintrags.

- 2080 Der aktuelle Name wird gelöscht. Das ist wichtig, weil ja die einzelnen Zeichen stets dazugefügt werden. Würde er hier nicht gelöscht, so würde er immer länger.
- 2100 Sollte aus irgendeinem Grund ein Abbruch des Suchens gewünscht werden, genügt es, eine Taste zu drücken. Dann wird die Speicherstelle 203 ungleich 64, und das Unterprogramm ist beendet. Dies ist zum Beispiel dann wichtig, wenn Sie sich beim gesuchten Namen vertippt haben und nicht warten wollen, bis das ganze Inhaltsverzeichnis durchsucht worden ist.
- 2120 Setzt die Variable für den Pufferzeiger auf den Anfang des nächsten File-Eintrags.
- 2140 Ende der Schleife für alle acht File-Einträge
- 2160 Der gesuchte File-Eintrag ist in diesem Block des Inhaltsverzeichnisses nicht gefunden worden. Fall es noch einen weiteren Block gibt, enthalten TN und SN Spur und Sektor dieses Blocks. Dann werden die beiden Variablen zu aktuellen Variablen (T,S), und die Suche geht weiter. Andernfalls ist TN=0.
- 2180 Der File-Eintrag ist nicht gefunden worden. Das Inhaltsverzeichnis ist zu Ende und das Unterprogramm wird beendet.
- 2200-2280 Holt ein Zeichen aus dem Puffer der Floppy und übergibt es (A\$) und den ASCII-Wert (A)
- 2300- Unterprogramm: Spritemaster zum Erstellen der Sprites
- 2300 Löscht den Spriteblock.
- 2360-2380 Setzt die Variablen für Koordinaten auf den Anfangswert.
- 2400 Die Information für das Sprite 1 steht im Block 13 ($13 \cdot 64 = 832$). Sprite 1 wird eingeschaltet und die Koordinaten auf 30,200 (=linke untere Ecke) gesetzt.

- 2420- Das Sprite wird auf dem Bildschirm als großes Rechteck angezeigt. Dieses Rechteck wird von oben nach unten und links nach rechts gezeichnet.
- 2420 Für alle Spalten des Rechtecks
- 2440 Für jede Zeile
- 2460 P=aktuelle Position auf dem Bildschirm
- 2480 Holt den Inhalt des aktuellen Bytes (VI%) im Spritblock und die Position des zugehörigen Punktes (VN%). VA% ist '0', wenn dieser Punkt gesetzt ist.
- 2500 Wenn der Punkt gesetzt ist, wird das Zeichen für den Bildschirm ein '*'.
- 2520 Ist der Punkt nicht gesetzt, so ist das zugehörige Zeichen auf dem Bildschirm ein '.'.
- 2540 Wenn die Position innerhalb des Bildschirms liegt, wird das Zeichen in den Bildschirmspeicher geschrieben und damit sichtbar. Die Sicherheitsabfrage auf erlaubte Position soll Ihnen die Möglichkeit geben, die Lage des Rechtecks zu ändern, ohne daß bei einem möglichen Fehler gleich der Rechner abstürzt.
- 2560 nächste Zeile
- 2580 nächste Spalte
- 2600 '43' ist der Wert für '+'. Dieses Kreuz wird an der jeweils aktuellen Stelle ausgegeben.
- 2620 Berechnet die Speicherstelle im Bildschirmspeicher, die zur aktuellen Position gehört.
- 2640 Der Inhalt dieser Speicherstelle wird in PR gerettet, da dort ja die '43' für das Kreuz hineingeschrieben wird.
- 2660 Wenn die Position noch innerhalb des Bildschirmspeichers liegt, wird das '+' an der aktuellen Position gezeichnet.
- 2680 Wartet, bis eine Taste gedrückt wurde.

- 2700 'CURSOR rechts' - wenn noch nicht der rechte Rand erreicht wurde, wird die aktuelle Position um ein Zeichen nach rechts verschoben.
- 2720 'CURSOR links'
- 2740 'CURSOR nach unten'
- 2760 'CURSOR nach oben'
- 2780 Wenn '*' gedrückt wurde, soll an der aktuellen Stelle ein Punkt gesetzt werden, und das vorher gerettete Zeichen wird auf '*' gesetzt.
- 2800 Ebenso wird es durch Drücken der Leertaste auf das Leerzeichen (ASCII=32) gesetzt.
- 2820 Durch Drücken von 'E' wird die Arbeit mit dem Spritemaster beendet.
- 2840 Durch Drücken von 'SHIFT+CLEAR/HOME' werden der Spriteblock und das Rechteck gelöscht. Sie brauchen also nicht alle Punkte einzeln zu löschen.
- 2860 Das vorher gerettete Zeichen an der aktuellen Position wird wieder angezeigt. Es kann aber durch '*' oder ' ' geändert worden sein.
- 2880 Wenn die letzte Taste eine CURSOR-Bewegung war, dann wird die Abfrage erneut durchlaufen. Es muß ja im Spriteblock nichts geändert werden.
- 2900 Andernfalls werden im Unterprogramm 3000 der Inhalt des aktuellen Bytes im Spriteblock (VI%) und das aktuelle BIT (VN%) und seine Position (VT%) berechnet
- 2920 Soll ein Punkt gesetzt werden, so werden der vorherige Inhalt und das neue Bit OR-verknüpft.
- 2940 Zum Löschen eines Punktes wird die zugehörige Maske aus dem Feld SP%(Position des Bits) geholt und das Bit durch AND-Verknüpfen gelöscht. Die acht Werte für SP% stehen am Ende des Programms

- in den DATA-Zeilen und unterscheiden sich dadurch, daß jeweils ein anderes Bit nicht gesetzt ist.
- 2960 Zurück zur Abfrageschleife
- 2980-3100 Berechnet die Position des Punktes im Spriteblock, der zu der Position im Rechteck gehört
- 3120 Beendet die Arbeit mit dem Spritemaster. Das Sprite wird abgeschaltet und zum Menü zurückgekehrt
- 3140- Unterprogramm: Datum erstellen und anzeigen
- 3160-3180 Die Überschrift und das zusammengesetzte Datum werden angezeigt.
- 3200-3360 Die einzelnen Werte werden auf dem Bildschirm ausgegeben. Durch Eingabe der ersten beiden Buchstaben kann ein Wert geändert werden (z.B. 'TA' für 'TAG') und mit 'ENDE' wird das Unterprogramm beendet.
- 3380-3520 Nach erfolgter Eingabe wird die zu ändernde Variable abgefragt und das Datum erneut angezeigt.
- 3540- Unterprogramm: Umwandlung des File-Eintrags ins GEOS-Format
- 3560 Sicherheitsabfrage: Falls IT ungleich '0' ist, dann ist der INFO-Bildschirm schon erstellt, und die Umwandlung wird nicht durchgeführt.
- 3580 IT, IS sind Spur und Sektor des INFO-Bildschirms. Sie werden auf einen Anfangswert gesetzt und angezeigt.
- 3600 Dieser Sektor wird nun als belegt gekennzeichnet, damit die Floppy ihn nicht überschreibt. Falls er nicht mehr frei ist, gibt es eine Fehlermeldung der Floppy, die den nächsten freien Sektor enthält.
- 3620 Einlesen der Fehlermeldung

- 3640 Falls das Belegen geklappt hat, ist C=0. Andernfalls enthält C den Track und D den Sektor des nächsten freien Blocks. Dieser wird nun probeweise als belegt gekennzeichnet.
- 3660-3700 Der GEOS-Filetyp wird abgefragt. GEOS kennt noch wesentlich mehr Filetypen, um aber einen solchen zu erzeugen, benötigen Sie umfangreiche Programmierkenntnisse. Es genügt nicht, hier nur einfach die Zahl einzugeben. Deshalb haben wir an dieser Stelle bewußt nur die beiden Eingaben '1' und '2' zugelassen. Wie man andere Filetypen erstellt (beispielsweise ein eigenes ACCESSORY), erfahren Sie im Kapitel: GEOS INTERN.
- 3740 Der gefundene Sektor für den INFO-Bildschirm wird vorerst wieder freigegeben. Sonst müßte die Floppy stets neue freie Sektoren suchen, falls dieser Programmteil mehrmals für denselben File-Eintrag durchlaufen würde.
- 3760 Der Block des Inhaltsverzeichnisses, in dem der zu ändernde File-Eintrag steht, muß erneut in den Floppy-Puffer gelesen werden, da ja inzwischen durch das beispielsweise Anzeigen des Inhaltsverzeichnisses der Puffer geändert sein kann.
- 3780 Setzt den Pufferzeiger auf das erste Byte hinter dem Filenamem.
- 3800-3960 Schreibt die geänderten Werte in den Puffer.
- 3980-4000 Diese beiden Zeilen schreiben den Puffer auf die Diskette. Dadurch ist die Änderung auch beim Verlassen dieses Programms gesichert. Eigentlich wollten wir an dieser Stelle den Befehl '(B-W:);3;0;AT;AS' benutzen, der für Sie sicherlich verständlicher gewesen wäre. Allerdings funktionierte dieser Befehl nicht zuverlässig. Es wurde stets das erste Zeichen falsch abgespeichert. Nachdem wir lange in unserem eigenen Programm nach einem möglichen Fehler gesucht hatten, mußten wir diesen Befehl dann umgehen.

Warnung: Dieser Befehl aus den Zeilen 3980-4000 ist ein mächtiges Werkzeug. Mit ihm kann bei falscher Handhabung die Diskette gelöscht werden. Sollten Sie den Filemaster eigenen Bedürfnissen anpassen wollen, so achten Sie bitte auf folgendes:

1. Die beiden BASIC-Zeilen sollten auf keinen Fall in der Reihenfolge vertauscht werden.
2. Achten Sie bei einer Änderung bitte darauf, daß AT nicht größer als 35 wird. Mit diesem Befehl ist es nämlich möglich, Spuren oberhalb von Spur 35 zu schreiben. Dabei besteht aber die Gefahr, daß der Floppykopf bei einer zu großen Zahl gegen den hinteren Anschlag fährt und beschädigt wird.

4020-4040 Informiert über den Schreibvorgang und kehrt zum Menü zurück.

4060-4880 Dieses Unterprogramm entspricht im Aufbau dem Programmteil zur Filesuche ab Zeile 1400. Die Unterschiede bestehen nur aus 2 Punkten:

1. Wird dieses Unterprogramm vom Menü aus zur Ausgabe des Inhaltsverzeichnisses angesprungen, so ist N\$="I", und es wird jeweils nur der aktuelle Name NA\$ ausgegeben und nicht die anderen Informationen. Am Ende des Inhaltsverzeichnisses erfolgt die Rückkehr ins Hauptprogramm.
2. Beim Aufruf des Unterprogramms zwecks Einlesen eines Sprites wird der File-Eintrag gesucht, und falls er gefunden wird, verzweigt das Programm zur Einleseroutine 4900, andernfalls kehrt es zum Menü zurück.

4900 Unterprogramm: Sprite einlesen

4940 Falls der File-Eintrag keinen INFO-Bildschirm hat, ist IT=0, und das Sprite wird nicht eingelesen.

- 4960 Holt den Sektor des INFO-Bildschirms in den Puffer.
- 4980 Setzt den Pufferzeiger auf den Anfang des Sprites.
- 5000-5080 Liest die 64 Bytes des Sprites aus dem Puffer und schreibt sie in den Block 13. Danach erfolgt die Rückkehr zum Menü.
- 5100 Unterprogramm: INFO-Bildschirm einlesen, ändern und speichern
- 5120 Wenn der File-Eintrag keinen Zeiger auf einen INFO-Bildschirm hat, bricht das Unterprogramm hier ab und kehrt zum Menü zurück.
- 5140 Der Sektor des INFO-Bildschirms wird in den Puffer gelesen.
- 5160 Der Pufferzeiger wird auf das erste Byte des Puffers gesetzt.
- 5180 Die Texte werden auf den Wert '.' gesetzt.
- 5200-5215 Die ersten vier Bytes des Sektors werden getestet. Sie müssen ganz spezielle Werte haben, sonst enthält dieser Sektor nicht den INFO-Bildschirm. In einem solchen Fall würde aber der Versuch, die Texte bis zur abschließenden '0' einzulesen, zu Schwierigkeiten führen, falls keine '0' gefunden wird. Deshalb wird dann das Einlesen der Texte übersprungen.
- 5220 Setzt den Pufferzeiger auf den Anfang der Texte, die im INFO-Bildschirm ausgegeben werden.
- 5240-5560 Liest die sechs weiter unten angezeigten Variablen ein. Das Ende ist jeweils durch eine '0' gekennzeichnet. Bis zu dieser '0' wird gelesen. Das würde zu Problemen führen, wenn dieser Sektor gar kein INFO-Bildschirm ist.
- 5600-5720 Die auszugebenden Texte stehen in einem Feld, so daß sie in einer Schleife ausgegeben werden können.
- 5760-5780 Ausgabeschleife für Überschrift und Texte

- 5800 Eingabe der Zahl zum ändern oder 'ENDE', falls die Eingabe fertig ist.
- 5820 Bei 'ENDE' Puffer entsprechend ändern und zurückschreiben.
- 5840 Eingabe in Zahl verwandeln und auf gültigen Bereich prüfen.
- 5860-5920 Ermöglicht das Ändern eines Textes. Dabei wird der bisherige Text auf den Bildschirm geschrieben und der Cursor für den INPUT-Befehl so plaziert, daß die bisherige Eingabe teilweise oder ganz übernommen werden kann.
- 5940 Erneute Ausgabe der Texte
- 5980- Ändern der Werte und Zurückschreiben des Puffers.
- 6000 Pufferzeiger auf Pufferanfang setzen.
- 6020 Die ersten Bytes setzen. Die genaue Erklärung dieser Zahlen finden Sie im Kapitel: GEOS INTERN.
- 6040-6080 Die Daten für das Icon aus dem Spriteblock lesen und in den Puffer schreiben.
- 6100-6560 Die Texte in den Puffer schreiben und jeden Text mit '0' abschließen.
- 6570 Für die beiden folgenden Befehle wird IT auf den gültigen Bereich hin überprüft.
- 6580-6620 Puffer zurückschreiben.
- 6640 Sektor des INFO-Bildschirms als belegt kennzeichnen.
- 6660 Fehlermeldung auf dem Bildschirm ausgeben.
- 6680-6700 Rückkehr zum Menü
- 6720-6740 Wandelt X\$ in zwei Zahlen im Format HIGH-Byte, LOW-Byte um.
- 6760 Daten für das Feld SP%, mit dem ein einzelner Punkt im Sprite gelöscht werden kann

3.4.3 Das Menü des Filemasters

Nach dem Starten des Filemasters und dem Drücken einer Taste erscheint auf Ihrem Bildschirm ein Menü, das wir ausführlich erklären möchten, damit Sie genau wissen, was jeder einzelne Befehl bewirkt:

1. Inhaltsverzeichnis = "I"

Dieser Menüpunkt zeigt Ihnen das Inhaltsverzeichnis der eingelegten Diskette. Wir haben darauf verzichtet, auch den Namen und die ID der Diskette einzulesen und anzuzeigen, weil das zusätzlich Zeit benötigt hätte. Die Ausgabe des Inhaltsverzeichnisses können Sie durch Drücken einer Taste beenden. Am besten wählen Sie dazu eine Funktionstaste (F1-F8), da diese keine Zeichen auf dem Bildschirm erzeugen, und halten Sie die Taste bitte einen Augenblick fest. Der Filemaster fragt nämlich erst nach dem kompletten Einlesen eines Filenamens ab, ob eine Taste gedrückt ist. Dies dient dazu, daß das Einlesen schneller geschieht. Zur Rückkehr ins Menü drücken Sie einfach die RETURN-Taste.

2. Sprite laden = "SL"

Mit diesem Unterprogramm können Sie ein bestehendes Icon aus irgendeinem GEOS-File auslesen und in den Spriteblock laden. Es steht Ihnen danach als Vorlage und zur Änderung zur Verfügung.

Zu Beginn fragt der Filemaster nach dem Namen des Programms, aus dem das Sprite ausgelesen werden soll. Geben Sie diesen Namen so ein, wie er im Inhaltsverzeichnis der Diskette angezeigt wird. Sollte sich das File nicht auf der Arbeitsdiskette befinden, so wählen Sie bitte vorher im Menü den Punkt: Diskette wechseln, und legen Sie nach der Aufforderung die gewünschte Diskette ein. Wenn das Einlesen des Sprites erfolgt ist und der Filemaster wieder das Menü anzeigt, wechseln Sie bitte erneut auf die Arbeitsdiskette (bitte mit 'DW').

Wenn Sie den Namen eingegeben haben, sucht der Filemaster diesen File-Eintrag auf der Diskette und zeigt die gerade überprüften Namen an. Hat er das Programm gefunden, so erscheint die Schrift: Sprite lesen, und oben links werden die Zahlen und die Nummer (0-63) angezeigt. Danach erscheint wieder das Menü. Sollte der Name nicht gefunden werden, so macht Sie der Filemaster darauf aufmerksam und kehrt danach zum Menü zurück. Sie haben sich dann wahrscheinlich vertippt. Kontrollieren Sie den Namen noch einmal, indem Sie das Inhaltsverzeichnis anschauen.

3. Sprite erstellen = "SE"

Mit diesem Menüpunkt bieten wir Ihnen die Möglichkeit, auf einfache Weise Sprites zu erstellen. Dazu sollten Sie wissen, daß ein Sprite aus 64 Zahlen (Bytes) besteht. Jedes Byte wiederum besteht aus acht Bit, die bei der Sprite-Darstellung jeweils einem einzelnen Punkt entsprechen. Somit entscheiden also $63 \cdot 8 = 504$ Punkte über das Aussehen eines Sprites. Vielleicht haben Sie schon einmal nach der Anleitung Ihres C64-Handbuchs versucht, ein Sprite zu entwerfen. Das ist doch etwas mühsam. Mit dem Sprite-Generator, den dieses Programm enthält, geht es wesentlich einfacher. Bevor Sie weiterlesen, wählen Sie am besten den Menüpunkt "SE", dann können Sie das Gelesene gleich mit dem Bildschirm vergleichen.

Im rechten Bereich des Bildschirms entsteht eine Matrix (ein Rechteck), das eine große Abbildung des Sprites darstellt. Wenn der Spriteblock ab 832 leer ist, Sie also noch kein Sprite eingelesen haben, enthält dieses Rechteck 24 Punkte in der Breite und 21 Punkte von oben nach unten, also insgesamt 504 Punkte. Sollten Sie schon ein Sprite eingelesen haben, so sehen Sie es unten links in der Ecke, und für jeden Punkt des Sprites wird in der Matrix ein "*" gezeigt. Wenn das Rechteck vollständig gezeichnet ist, erscheint in seiner oberen linken Ecke ein Kreuz. Dieses Kreuz gibt immer die augenblickliche Position an und entspricht also dem Cursor, den Sie sonst zur Verfügung haben. Im Unterschied zum Cursor wird das Kreuz aber nicht um eine Stelle nach rechts bewegt, wenn Sie eine Taste drücken.

Um also beispielsweise ein '**' an die Position zu schreiben und das Kreuz um eine Stelle nach rechts zu bewegen, drücken Sie '**' und dann 'Cursor rechts'. Folgende Tasten zeigen eine Wirkung:

Cursor-Tasten

Mit ihnen wird der Cursor in der Matrix bewegt. Dabei bleiben die Punkte, über die er hinwegbewegt wird, unverändert.

''-Taste**

Mit ihr wird an der augenblicklichen Position ein '**' gesetzt. Gleichzeitig wird auch der entsprechende Punkt im Spriteblock gesetzt. Der Cursor ändert seine Position nicht.

Leertaste

Sie löscht einen Punkt an der Position des Kreuzes.

Shift+Clr

Mit dieser Tastenkombination löschen Sie normalerweise in BASIC den Bildschirm. Im Spritemaster wird dadurch das Sprite gelöscht und die Matrix auf den Anfangswert gesetzt. Wenn Sie also neu anfangen wollen, brauchen Sie nicht alle Punkte einzeln zu löschen.

'E'-Taste

Durch Drücken der Taste 'E' beenden Sie die Arbeit mit dem Spritemaster und kehren zum Menü zurück. Zwar wird das Sprite aus Block 13 dann nicht mehr angezeigt, es ist aber nicht gelöscht. Wenn Sie später den Spritemaster wieder benutzen, können Sie an der Stelle weitermachen, an der Sie aufgehört haben.

File einlesen = "FE"

Mit diesem Kommando lesen Sie einen File-Eintrag der Diskette ein. Nachdem Sie den gewünschten Namen eingegeben haben, sucht der Filemaster diesen Namen und zeigt die gerade über-

prüften File-Einträge auf dem Bildschirm an. Die Überprüfung bezieht sich nur auf die Länge des Namens, den Sie eingegeben haben. Um also die ALARM CLOCK einzulesen, reicht es, 'ALA' einzugeben, vorausgesetzt, es gibt keinen File-Eintrag vor der ALARM CLOCK, der ebenfalls mit diesen drei Buchstaben beginnt. Dann müssen Sie den vollen Namen richtig eingeben.

Wird der gewünschte File-Eintrag nicht im Inhaltsverzeichnis gefunden, so kehrt der Filemaster zum Menü zurück. Andernfalls werden alle Angaben auf dem Bildschirm angezeigt. Dies sind:

Starttrack, Startsektor

Spur und Block, an der das Programm auf der Diskette beginnt.

Filetyp

Normaler DOS-Filetyp

0=DELETED, 1=SEQUENTIAL, 2=PROGRAM, 3=USER, 4=RELATIVE.

Auf der Original-GEOS-Diskette sind bis auf die beiden Programme GEOS und GEOS BOOT alle vom Typ: USER.

INFO-Track+Sektor

Spur und Block des INFO-Bildschirms. Sind beide '0', ist er noch nicht erstellt.

Filestruktur

GEOS kennt zwei Typen. SEQUENTIAL=0 und VLIR=1. Hier muß für Ihre Programme immer eine '0' stehen.

GEOS-Filetyp

0=BASIC, 1=ASSEMBLER

Weitere Filetypen entnehmen Sie bitte dem Kapitel GEOS INTERN. Sie sollten aber ohne umfangreiche Programmierkenntnisse nur '0' oder '1' wählen.

Datum

Das Datum ist in den fünf Angaben: Jahr, Monat, Tag, Stunde, Minute gespeichert.

Filelänge

Gibt die Anzahl der Blöcke=Sektoren an, die das Programm auf der Diskette benötigt. Nach dem Drücken der RETURN-Taste kehrt der Filemaster ins Menü zurück.

5. File zeigen

Mit diesem Kommando werden noch einmal alle Informationen aus dem File-Eintrag angezeigt. Dieser Befehl wird nur ausgeführt, wenn ein File erfolgreich eingelesen wurde.

6. GEOS-File erzeugen = "GEOS"

Mit diesem Kommando werden die geänderten Daten des File-Eintrags auf die Diskette geschrieben. Dazu wird ein freier Block gesucht, in dem später der INFO-Bildschirm gespeichert werden soll. Vorher sollten Sie das Datum eingegeben haben. Nach dem erfolgreichen Schreiben erscheint die Meldung 'GESCHRIEBEN' auf dem Bildschirm, und Sie können mit der Return-Taste ins Menü zurückkehren.

7. Datum eingeben/ändern = "FD"

Hier wird das Datum des eingelesenen Files angezeigt und kann geändert werden. Bei von Ihnen erstellten Programmen werden die einzelnen Daten noch '0' sein. Durch Eingabe der ersten zwei Buchstaben kann jeder Eintrag geändert werden. Geben Sie also beispielsweise 'ST' ein, um die Stunden zu ändern. Durch Eingabe von 'ENDE' kehren Sie mit dem geänderten Datum ins Menü zurück. Diese Änderung können Sie sich auch mit 'FZ' vom Menü aus anschauen.

8. Diskette wechseln = "DW"

Dieses Programm kann nur erfolgreich arbeiten, wenn Sie für jeden Diskettenwechsel dieses Kommando benutzen. Das liegt daran, daß direkt von der Diskette gelesen und auf sie geschrieben wird. Wählen Sie zum Wechsel der Diskette also "DW", dann werden Sie aufgefordert, die Diskette einzulegen und eine Taste zu drücken. Danach befinden Sie sich wieder im Menü.

9. Programm beenden = "ENDE"

Bitte beenden Sie das Programm nicht, indem Sie die Taste 'STOP' betätigen. Wir haben sie absichtlich nicht abgeschaltet, weil Sie den Filemaster vielleicht ändern möchten. Nach erfolgreichem Schreiben auf die Diskette muß nämlich beim Programmende noch die geänderte BAM auf die Diskette geschrieben und die geöffneten Kanäle müssen geschlossen werden. Um die Arbeit mit dem Filemaster zu beenden, wählen Sie also bitte 'ENDE'.

10. INFO einlesen = "INFO"

Dieses Unterprogramm bereitet den INFO-Bildschirm auf. Bevor Sie es benutzen, sollten Sie auf jeden Fall ein Sprite eingelesen oder erstellt haben. Der Filemaster prüft, ob der Sektor, der im File-Eintrag für den INFO-Bildschirm bereitgestellt ist, schon Daten enthält. Ist dies der Fall, werden die Daten eingelesen und angezeigt, im anderen Fall wird das Einlesen übersprungen. Danach können Sie die sechs Angaben ändern, indem Sie die Zahl eingeben, die vor dem Text steht. Wenn ihr Programm in BASIC geschrieben ist, brauchen Sie die ersten drei Angaben nicht zu ändern. Sie werden von GEOS nicht benötigt und daher auch nicht auf die Diskette geschrieben. Wenn Sie mit der Eingabe fertig sind, geben Sie 'Ende' ein, und der erstellte oder geänderte INFO-Bildschirm wird auf die Diskette geschrieben.

3.4.4 Bedienung des Filemasters

Der Filemaster ist ein Programm, das sehr weitgehende Eingriffe in das Inhaltsverzeichnis der Diskette ermöglicht. Wir möchten Sie daher dringend bitten, die zu ändernden Programme auf einer gesonderten Diskette, der sogenannten Arbeitsdiskette, zu konvertieren. Diese Arbeitsdiskette sollten Sie möglichst nur zu diesem Zweck benutzen und keine wichtigen Programme darauf abspeichern. Es ist danach leicht, die geänderten Programme mit GEOS auf die gewünschte Diskette zu kopieren.

Zuerst kopieren Sie bitte das zu ändernde Programm auf die Arbeitsdiskette. Laden Sie dann den Filemaster, und starten Sie ihn mit RUN. Legen Sie die Arbeitsdiskette ein, und drücken Sie eine Taste. Danach zeigt Ihnen der Filemaster das Menü an. Um nun das eigene Programm ins GEOS-Format zu konvertieren, führen Sie bitte die folgenden Schritte aus:

1. Lesen Sie mit "FE" das zu ändernde Programm ein. Wir wollen einmal davon ausgehen, daß es ein BASIC-Programm ist.
2. Nun erstellen Sie bitte ein Sprite oder lesen eins aus einem GEOS-File ein. Falls Sie zum Einlesen die Diskette wechseln möchten, so wählen Sie bitte das Kommando 'DW'.
3. Wählen Sie "FD", um das File-Datum zu erstellen. Beginnen Sie mit dem Jahr, indem Sie "JA" eingeben. Anschließend erfragt der Filemaster das Jahr. Die Eingabe muß zweistellig erfolgen, da von GEOS und daher von diesem Programm die Zahl 1900 dazugezählt wird. Das Programm prüft nicht auf sinnvolle Eingaben. Wir haben bewußt darauf verzichtet, um den Filemaster nicht noch umfangreicher werden zu lassen. Geben Sie also bitte nicht beim Monat "123456789" ein. Eine Fehlermeldung wäre unvermeidlich, und Ihre bisherige Arbeit mit dem Filemaster wäre vergebens.

4. Wählen Sie "GEOS", um den File-Eintrag ins GEOS-Format zu konvertieren. Auf dem Bildschirm erscheint eine Meldung: " 65 NO BLOCK" und zwei Zahlen. Das ist völlig in Ordnung und zeigt, daß der Filemaster nach einem freien Block für den INFO-Bildschirm sucht. Anschließend sollte die Meldung "OK 0 0 " erscheinen. Geben Sie danach eine '1' für den Filetyp BASIC ein. Ein abschließendes 'GESCHRIEBEN' informiert Sie darüber, daß der File-Eintrag geändert und auf die Diskette geschrieben worden ist.
5. Wählen Sie "INFO", und geben Sie die Texte (4-6) ein. Den Hilfstext (6) sollten Sie nur kurz halten (maximal 50 Zeichen), da Sie ihn ja später unter GEOS leicht ändern und vervollständigen können. Nachdem Sie mit den Eingaben zufrieden sind, wählen Sie "ENDE", und der INFO-Bildschirm wird auf die Diskette geschrieben. Das Programm informiert Sie darüber mit dem Hinweis "GESCHRIEBEN".

Wenn der Schreibvorgang richtig durchgeführt wurde und vorher noch kein INFO-Bildschirm zu diesem Text bestand, erscheint die anschließende Meldung:

0 OK 0 0

Wenn der INFO-Sektor aber vorher schon beschrieben war, erscheint die Meldung

65 NO BLOCK

und zwei Zahlen. Der Block ist dann trotzdem richtig geschrieben worden, nur konnte er nicht mehr als belegt gekennzeichnet werden, weil er schon belegt war. Danach können Sie die Arbeit mit dem Filemaster beenden oder ein weiteres Programm konvertieren.

6. Bei Ihrer nächsten Arbeit mit GEOS kopieren Sie das geänderte Programm auf Ihre GEOS-Arbeitsdiskette und genießen Ihre Arbeit.

Abschließend möchten wir Ihnen noch einige Tips zur Arbeit mit dem Filemaster geben. Sie können nämlich noch einige zusätzliche Möglichkeiten ausnutzen, die im Programm schon vorgesehen sind. Dazu müssen Sie meist nur eine Zeile ändern. Wir haben das in dem abgedruckten Programm nicht gemacht, um die Arbeit mit dem Filemaster nicht unnötig kompliziert zu gestalten und einige Fehlermöglichkeiten auszuschließen.

Sollten Sie versehentlich auf der GEOS-Diskette ein File gelöscht haben, so können Sie es mit dem Filemaster retten. Das funktioniert allerdings nur sicher, wenn Sie nach dem versehentlichen Löschen noch keine Files gespeichert haben. Beim Löschen werden nämlich nicht etwa die Sektoren auf der Diskette gelöscht, sondern der Filetyp wird auf '0' gesetzt, und die Sektoren werden zum erneuten Beschreiben freigegeben. Wenn Sie den Filetyp wieder auf einen gültigen Wert setzen, erscheint das File wieder unter GEOS. Wenn Sie diese Möglichkeit in den Filemaster einbauen wollen, so ändern Sie ganz einfach folgende Programmzeilen um:

```
alt: 1940 IF FT=0 THEN 2080
neu: 1940 IF FT=0 THEN PRINT"****";
```

Nun werden gelöschte Files nicht mehr übergangen, sondern können ebenfalls eingelesen werden. Dabei werden vor ein gelöscht File zur Markierung drei Sternchen gesetzt. Nun müssen Sie noch eine Möglichkeit vorsehen, den Filetyp von '0' auf einen gültigen Wert zu setzen. Dazu ändern Sie noch:

```
neu: 3765 IF FT=0 THEN PRINT#5,"B-P:"3,AB:
neu: 3770 IF FT=0 THEN PRINT#3,CHR$(130);
alt: 3720 FS=0: GT=X
neu: 3720 GT=X
```

Damit ist der Filetyp wieder auf einen gültigen Wert gesetzt und wird durch Zurückschreiben des Puffers ab Zeile 3980 auch auf der Diskette geändert. Allerdings gibt es ein Problem. Ganz wichtig ist, daß Sie anschließend, also nachdem der File-Eintrag geändert (GEOS) und der INFO-Bildschirm erstellt wurde (INFO), GEOS laden und auf die geänderte Diskette den Befehl VALIDATE anwenden. Dadurch werden die Sektoren, die zu

dem geretteten File gehören, wieder als belegt gekennzeichnet, und damit ist das File endgültig gerettet. Wir wollen diese Rettungsaktion einmal kurz an einem Beispiel durchgehen.

Angenommen, Sie haben einen mit GEOWRITE erstellten Text 'Konzept' versehentlich gelöscht. Verlassen Sie GEOS mit 'BASIC'. Laden Sie den Filemaster, und nehmen Sie die beschriebenen Änderungen vor. Legen Sie die Diskette ein, auf der der gelöschte Text ist, und starten Sie den Filemaster. Geben Sie 'FE' und als Namen 'Konzept' ein. Wenn das Programm dann gefunden ist, werden die wichtigsten Daten angezeigt, unter anderem:

```
Filetyp      = 0
GEOS F-Typ  = 7
Filestruktur = 1
```

Wählen Sie danach im Hauptmenü 'GEOS', und geben Sie als GEOS-Filetyp die '7' ein. Anschließend wird die Änderung abgespeichert, und Sie können den Filemaster verlassen. Booten Sie dann GEOS, legen Sie die geänderte Diskette ein, und wählen Sie VALIDATE. Anschließend können Sie Ihren Text weiterbearbeiten oder drucken.

Als GEOS-Filetyp sind in der abgedruckten Version nur Werte für BASIC (=1) oder ASSEMBLER (=2) zugelassen. In Zeile 3700 wird die Eingabe daraufhin überprüft. Wenn Sie aber ein eigenes Programm beispielsweise in ein echtes Accessory umwandeln wollen (GT=5), so entfernen Sie bitte die Zeile 3700 oder erweitern Sie sie um weitere erlaubte Eingaben. Sie müssen dann aber selbst darauf achten, daß die GEOS-Filetypen auch einen vernünftigen Wert haben. Wenn Sie beispielsweise ein BASIC-Programm auf GT=5 setzen, wird GEOS es zwar laden und starten, danach hängt GEOS sich dann aber auf, weil ein BASIC-Programm unmöglich als Accessory funktionieren kann. Wie man wirkliche 'Hilfsmittel' erzeugen kann, steht im Kapitel INTERN ausführlich beschrieben.

Wenn Sie an einem GEOS-File etwas ändern möchten, beispielsweise das Datum oder den Namen des Programmierers, so beachten Sie bitte, daß beim Erstellen des INFO-Bildschirms kein

Sprite eingelesen wird. Wir haben bewußt darauf verzichtet, um ein möglicherweise schon von Ihnen erstelltes Sprite nicht versehentlich zu löschen. Lesen Sie dann zusätzlich auch das Sprite des Files ein. Wenn Sie also beispielsweise etwas an der ALARM CLOCK ändern wollen, so lesen Sie mit 'FE' die ALARM CLOCK ein und anschließend mit 'SL' auch das zugehörige Sprite. Danach können Sie Änderungen vornehmen und diese auch abspeichern (INFO).

4. GEOS viel einfacher und komfortabler

In diesem Kapitel wollen wir Ihnen einige sehr interessante Möglichkeiten vorstellen, GEOS einfacher und komfortabler zu machen. Wir stellen Ihnen dazu sogenannte Patches vor. Das sind Änderungen an einem Programm auf der Diskette. Im Gegensatz zu den Pokes, die man beispielsweise einsetzt, um in einem Spiel unendlich viele Leben zu bekommen, verändern Patches das Programm auf der Diskette. Damit muß dieser Vorgang nur ein einziges Mal erfolgen, anschließend kann einfach das geänderte Programm geladen werden.

4.1 Sicherheitskopien von GEOS V1.3 deutsch und V1.2

Anwender von Programmen beklagen oft, daß viele Softwarefirmen ihre Programme nur kopiergeschützt unter die Leute bringen. Dadurch ist es nicht möglich, Sicherheitskopien von den kostbaren Disketten zu machen. Wenn aber die Originaldiskette einmal aus irgendeinem Grund nicht mehr funktionsfähig ist, kann der Benutzer seine Texte oder Daten möglicherweise wochenlang nicht mehr benutzen, bis endlich Ersatz beschafft ist. Zusätzlich verhindern kopiergeschützte Programme, daß man sie seinen Bedürfnissen entsprechend anpassen kann.

GEOS hat einen besonders guten Kopierschutz. An diesem Programm werden wohl alle Kopierprogramme scheitern, wenn Sie nicht mindestens mit einem parallelen Kabel zwischen der Floppy und dem Rechner arbeiten. Aber wer hat schon einen Floppyspeeder mit parallelem Bus und das vielleicht einzige Kopierprogramm, das dann vielleicht GEOS bezwingen kann?

Andererseits ist aber gerade bei GEOS eine Änderung an der Diskette, von der man booten will, sehr nützlich. Viele Tips und Tricks beziehen sich gerade auf Änderungen an GEOS KERNAL, und das wird nur einmal beim Booten von der Originaldiskette geladen. Da wir schon seit einiger Zeit eine Sicher-

heitskopie von GEOS angefertigt haben und so nach Lust und Laune experimentieren konnten, haben wir uns überlegt, wie wir Ihnen dieselbe Möglichkeit verschaffen können.

Sicherheitskopie für GEOS V1.3 deutsch

Vielleicht haben Sie im 1. Kapitel schon eine Vorstellung davon gewonnen, wie schwierig das Erstellen einer Kopie von GEOS V1.3 deutsch ist. Trotzdem haben wir einen Weg für Sie gefunden, der allerdings aus drei kurzen Einzelprogrammen besteht. Bevor wir Ihnen diese Programme und deren Funktion allerdings vorstellen, möchten wir noch auf die rechtliche Seite des Vorhabens hinweisen. Die Weitergabe von Kopien von GEOS ist nicht erlaubt. Sie dürfen diese Kopien also ausschließlich selbst benutzen. Unsere Hilfsprogramme haben den ausschließlichen Zweck, Sicherheitskopien der Originaldiskette herstellen zu können und auf dieser Kopie gegebenenfalls gefahrlos Änderungen vorzunehmen.

Vorgehensweise:

Zunächst einmal benötigen Sie eine Kopie der installierten Originaldiskette SYSTEM DISK. Außerdem benötigen Sie noch eine leere formatierte Diskette, um die folgenden drei Programme zu speichern und die Daten, die von diesen Programmen angelegt werden. Geben Sie anschließend das folgende BASIC-Programm "getspeeder" ein, und speichern Sie es auf der leeren Diskette, die wir im folgenden "Programmdiskette" nennen wollen, unter dem Namen "getspeeder" ab.

```
10 DIM A(300)
20 OPEN 5,8,15,"IO:"
22 INPUT#5,A,B$,C,D
25 PRINT A,B$,C,D
30 FOR I = 40 TO 255
40 PRINT#5,"M-R"CHR$(I);CHR$(4)
50 GET#5,X$
60 X=ASC(X$+CHR$(0))
70 A(J)=X
80 J=J+1
90 NEXT I
```

```

100 FOR I = 0 TO 60
110 PRINT#5,"M-R"CHR$(I);CHR$(5)
120 GET#5,X$
130 X=ASC(X$+CHR$(0))
140 A(J)=X
150 J=J+1
160 NEXT I
170 CLOSE 5
180 OPEN 3,8,3,"WORKFILE,S,W"
200 FOR I=0 TO 260
210 PRINT#3,CHR$(A(I) );
220 NEXT I
230 CLOSE 3
240 PRINT "ALLES KLAR!"

```

Wenn Sie das Programm gespeichert haben, nehmen Sie bitte die Kopie der SYSTEM DISK und starten den Bootvorgang mit:

```
LOAD "GEOS",8,1
```

Nach wenigen Sekunden endet der Vorgang vorzeitig mit einem Reset. Legen Sie nun die "Programmdiskette" in das Diskettenlaufwerk, und geben Sie folgende Zeilen im Direktmodus ein:

```

OPEN 3,8,3,"#1"
OPEN 4,8,4,"#2"
LOAD "getspeeder",8
CLOSE 3
CLOSE 4

```

Starten Sie das Programm. Nach einigen Sekunden meldet es sich mit: ALLES KLAR. Das Programm hat auf der Diskette ein Datenfile mit dem Namen "WORKFILE" angelegt. Nun ist das zweite Programm an der Reihe, das die eben erzeugten Daten benötigt. Geben Sie das folgende Programm sorgfältig ein, und speichern Sie es unter "decode" auf der "Programmdiskette":

```

5 DIM A(300)
10 OPEN 3,8,3,"WORKFILE,S,R"
20 FOR I = 0 TO 260
30 GET#3,A$
40 A(I)=ASC(A$+CHR$(0))
50 NEXT I
60 CLOSE 3

```

```
100 OPEN 5,8,15,"IO:"
105 J = 0
110 FOR I = 40 TO 255
120 PRINT#5,"M-W"CHR$(I)CHR$(4)CHR$(1)CHR$(A(J))
130 J = J+1
140 NEXT I
250 FOR I = 0 TO 50
260 PRINT#5,"M-W"CHR$(I)CHR$(5)CHR$(1)CHR$(A(J))
270 J = J+1
280 NEXT I
300 I = 0
310 READ A
320 IF A = -1 THEN 360
330 X = X+A
340 PRINT#5,"M-W"CHR$(I)CHR$(6)CHR$(1)CHR$(A)
345 I = I+1
350 GOTO 310
360 IF X <> 9051 THEN CLOSE5:END
400 REM AUSFUEHREN
405 INPUT"DISKETTE EINLEGEN";N$
410 PRINT#5,"IO:"
420 PRINT#5,"M-E"CHR$(0)CHR$(6)
430 INPUT#5,A,B$,C,D
440 PRINT A,B$,C,D
450 REM
460 FOR I=64 TO 81
470 READ A
480 PRINT#5,"M-W"CHR$(I)CHR$(3)CHR$(1)CHR$(A)
490 NEXT I
500 PRINT#5,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$(144)
600 CLOSE5
610 PRINT "ALLES KLAR!": END
1000 REM
1001
DATA169,0,141,101,6,169,20,141,0,3,169,15,141,1,3,169,254,141,100,6,173
1002
DATA0,3,133,6,208,8,165,7,141,100,6,238,100,6,173,1,3,133,7,169,128,133
1003
DATA0,165,0,48,252,160,0,185,2,3,89,42,4,153,2,3,200,204,100,6,208,241
1004
DATA169,144,133,0,165,0,48,252,173,0,3,208,193,173,101,6,208,16,238,101
1005 DATA6,169,20,141,0,3,169,17,141,1,3,76,15,6,96,0,0,-1
1010 REM
1011 DATA136,208,250,132,62,169,255,76,78,6,0,0,0,0,133,72,96,96
```

Wenn Sie es gespeichert haben und die "Programmdiskette" noch eingelegt ist, starten Sie es bitte. Es liest dann die vorher erzeugten Daten ein und fordert Sie mit "DISKETTE EINLEGEN" auf, die Kopie der SYSTEM DISK einzulegen. Wenn das geschehen ist, drücken Sie bitte die ReRUN-Taste. Nun dauert es etwa eine Minute, bis sich das Programm mit ALLES KLAR! meldet. Nun ist noch ein dritter Schritt notwendig. Geben Sie dazu das dritte Programm ein, und speichern Sie es unter "basicpatch" auf der "Programmdiskette".

```

10 IF A=0 THEN A=1:LOAD"GEOS BOOT",8,1
20 OPEN 5,8,15,"IO:"
30 PRINT#5,"R:GEOS BOOT ALT=GEOS BOOT"
40 INPUT#5,A,B$,C,D
50 PRINTA,B$,C,D
60 CLOSE 5
100 IF PEEK(25759)<> 76THEN END:REM JMP
120 POKE 25759,96 :REM RTS
122 POKE 25748,44 :REM STA $0100,Y=>BIT
130 SYS 25730 :REM DECODIERE
132 POKE 25748,153:REM STA $0100
140 POKE 25759,76 :REM WIEDER JMP
150 IF PEEK(25745)<>201THEN END:REM EOR
160 POKE 25745,0:REM EOR #$00
170 REM $63F1 = JSR $F510 PATCHEN
180 IF PEEK(25585)<>245 THEN END
190 POKE 25585,4
200 POKE 25586,200
210 POKE 25587,76
220 REM NUN JSR $F510 -> JMP 04C8
230 REM $63C3 STA $0413 -> BIT $0413
240 IF PEEK(25541)<>141 THEN END
250 POKE 25541,44
260 REM *****
270 REM * NUN GEOS BOOT SPEICHERN *
280 REM *****
300 POKE 43,0 :POKE44,96:REM ANF
310 POKE 45,168:POKE46,100:REM ENDE
320 SAVE "GEOS BOOT",8
990 PRINT "ALLES KLAR"
995 END
998 REM *****

```

Wenn es gespeichert ist, legen Sie bitte die Kopie der SYSTEM DISK ins Diskettenlaufwerk und starten das Programm. Es benennt das Original-Programm GEOS BOOT um und erzeugt ein neues GEOS BOOT auf der Diskette. Wenn sich das Programm mit ALLES KLAR! meldet, haben Sie es geschafft. Sie haben nun eine Sicherheitskopie von der SYSTEM DISK. Sollte das Programm ohne diese Meldung abbrechen, so haben Sie entweder nicht GEOS V1.3 deutsch oder eine geänderte Version, bei der unsere Vorgehensweise leider nicht funktioniert. Problemlos funktioniert der Vorgang aber bei GEOS V1.3 deutsch mit dem Auslieferungszeitraum Anfang 1988.

EASY-ACCESS - Die Sicherheitskopie für GEOS V1.2

Bevor wir Ihnen das kurze Programm vorstellen, noch ein paar Hinweise zur Bedienung. Bitte benutzen Sie das Programm nur auf einer Kopie, die Sie mit dem Programm BACKUP von der Originaldiskette erstellt haben. Die Originaldiskette ist ja nur zum Booten da. Die Kopie benennen Sie dann um in BOOTDISK. Nun brauchen wir noch etwas Platz auf der Diskette, und zwar 2 KBytes. Am besten entfernen Sie einfach einige der Druckertreiber, die Sie sowieso nicht benötigen. Dann kann es losgehen. Verlassen SIE GEOS, und laden Sie EASY ACCESS. Legen Sie die Diskette BOOTDISK ein. Benennen SIE GEOS BOOT um in GEOS BOOTalt. Das geht ganz mit folgender Zeile im Direktmodus:

```
OPEN 3,8,15,"R:GEOS BOOTalt=GEOS BOOT": CLOSE3 (RETURN-TASTE)
```

Anschließend starten SIE EASY ACCESS mit RUN. Auf der BOOTDISK wird nun eine neue Version von GEOS BOOT erzeugt. Wenn EASY ACCESS fertig ist, können Sie sich das Inhaltsverzeichnis der Diskette BOOTDISK einmal anschauen. Dort sehen Sie nun zwei BOOT-Programme: GEOS BOOTalt und GEOS BOOT. Sie können jetzt das Ergebnis von EASY ACCESS sofort ausprobieren, indem Sie von dieser Diskette booten. Anschließend können Sie von GEOS aus das alte File GEOS BOOTalt löschen.

Da allerdings das neue Boot-Programm kein Icon besitzt, können Sie GEOS BOOTalt auch solange auf der Diskette lassen, bis Sie mit dem Filemaster ein Icon für GEOS BOOT erstellt haben. Dann können Sie nämlich das bisherige Icon als Vorlage verwenden. Wenn alles geklappt hat, können Sie die Originaldiskette für den Fall der Fälle an einen sicheren Ort verbannen und stets mit der BOOTDISK GEOS laden und starten.

```
1  REM *****
2  REM ***** EASY ACCESS *****
3  REM **** BY MANFRED TORNSDORF ****
4  REM *****
5  REM:
10 OPEN 3,8,3,"GEOS BOOTALT"
20 OPEN 4,8,4,"GEOS BOOT,P,W"
30 GET#3,A$:A=ASC(A$+CHR$(0))
40 IN=ST
41 IF A<>59 AND A<>7 AND A<>29 THEN FO=0+1:GOTO 50
42 IF A=59 THEN FO=FO+1
44 IF A=7 AND FO=1 THEN FO=FO+1
46 IF A=29 AND FO=2 THEN FO=3:PRINT#4,CHR$(162);:PRINT "OK":GOTO 70
50 PRINT#4,CHR$(A);
60 IF ST<>0 THEN CLOSE3:CLOSE4:END
70 PRINT#4;
80 IF IN=0 THEN 30
90 CLOSE3:CLOSE4:END
```

4.2 Bleibende Änderungen an GEOS

Nobody is perfect! Dies gilt sicherlich auch für die Erfinder von GEOS. Zwar sind ihnen bei der Programmierung keine schwerwiegenden Fehler unterlaufen (von der notorisch falschgehenden Uhr in GEOS V1.2 einmal abgesehen), doch sind uns bei der Arbeit mit dieser neuen Benutzeroberfläche einige Dinge aufgefallen, die man vielleicht hätte besser machen können. Zu nennen wäre da z.B. die zu lange Wartezeit beim erneuten booten von GEOS durch die RESTORE-Taste. Es sind oft auch kleine Dinge, die die Arbeit mit GEOS erheblich vereinfachen könnten. Sei es nun eine schneller ansprechende Wiederholfunktion aller Tasten oder sinnvollere Randbegrenzungen beim Neustart von GEOWRITE.

Einige unserer Vorschläge fallen auch einfach unter die Rubrik 'Kosmetik'. So störte es uns zum Beispiel, daß man die vom NOTE PAD erstellten NOTES außer durch den Filenamen optisch nicht von ihrem 'Mutterfile' unterscheiden kann, denn beide besitzen das gleiche Icon. Auch ein "Lifting" der Blinkfrequenz des Cursors hielten wir für angebracht, da dessen Schüchternheit GEOS häufig zu einem Versteckspiel ausarten läßt. Dies sind nur einige Punkte aus unserer langen Liste, denen wir dieses Kapitel gewidmet haben. Selbstverständlich bieten wir Ihnen außer den Vorschlägen an sich auch die Möglichkeit, jede Verbesserung tatsächlich zu verwirklichen.

4.2.1 Die Problematik von Änderungen an GEOS

In vielen Fällen genügte es in GEOS schon, eine oder zwei bestimmte Speicherstellen zu verändern, um einen sehr nützlichen Effekt zu bewirken. Genau hierbei ergibt sich das folgende Problem.

Man kann unter GEOS nicht einfach, wie im alten C64 Betriebssystem, in den Speicher 'poken' und womöglich sogar noch ein so modifiziertes Programm (z.B. GEOWRITE) wieder abspeichern (Sie kennen solche POKES vielleicht von Spielen, bei denen einem anschließend eine unbegrenzte Anzahl an Video-Helden oder Raumschiffen zur Verfügung steht).

Nun könnte man auf die Idee kommen, einfach vom C64 Betriebssystem das zu ändernde Programm mit LOAD "name",8 in den Rechner zu laden, anschließend unsere POKES einzugeben und das Ganze dann mit SAVE "name",8 wieder abzuspeichern. Das wäre auch tatsächlich möglich, wenn nicht GEOS ein vollständig anderes (besseres) File-Format benutzen würde.

Die Fehlermeldung, die bei einem solchen Versuch auftritt (Filetype mismatch) ist eine Folge dieser Inkompatibilität. Selbst wenn man den Filetyp ändern würde, ergeben sich zusätzliche Schwierigkeiten durch eine andere Verschlüsselung der Zeichen, die GEOS benutzt. Um nun doch gezielte Änderungen an GEOS-Files vornehmen zu können, haben wir ein Programm

geschrieben. Es besteht aus einem BASIC-Teil mit einer in DATAS gefaßten Maschinensprache-Routine. Die Funktion basiert dabei auf folgender Idee.

Da es anscheinend problematisch ist, ein GEOS-File ohne viel Aufwand zwecks Modifikation in den Rechner zu laden, gehen wir eben den umgekehrten Weg: Ein kleines Maschinensprache-programm wird von dem BASIC-Teil in die Floppy kopiert und 'poked' die zu ändernden Bytes dort gewissermaßen direkt auf die Diskette. Die neuen Werte werden dem Programm dabei sozusagen als 'Post' in den Floppy-Puffer mitgegeben. Diese Methode hat zusätzlich noch einen Geschwindigkeitsvorteil, weil die Übertragung von Daten über den normalerweise sehr langsamen seriellen Bus auf ein Minimum reduziert wird.

Allerdings ergibt sich ein weiteres Problem bei der neuen GEOS Version 1.3. Die Firma Berkley Softworks hat sich nämlich die Mühe gemacht, große Teile des GEOS Kernals auf der Diskette zu verschlüsseln. Damit ist es natürlich für unser Programm unmöglich, bestimmte Programmteile wiederzufinden, um dort Änderungen vornehmen zu können. Sie dürfen aus diesem Grunde Änderungen am Kernal nur auf solchen Disketten vornehmen, die mit Hilfe unseres Programms "Sicherheitskopie" (s. Kapitel 4.1) kopiert worden sind.

Das nun folgende Listing unseres 'Modifikators' sollten Sie mit äußerster Sorgfalt eingeben, da Fehleingaben nicht nur ein harmloses Nichtfunktionieren zur Folge haben, sondern auch reichlich Verwüstung auf Ihren Disketten stiften können. Natürlich bietet die vorhandene Prüfsumme schon einen recht guten Schutz vor Tippfehlern. Trotzdem halten Sie mit unserem Programm ein sehr mächtiges Werkzeug in Ihren Händen, so daß Sie vorsichtig sein sollten. Alle Änderungen sollten also ausschließlich auf Kopien und nie auf der Originaldiskette erfolgen.

4.2.2 Der Modifikator

```

10 REM *   GEOS-MODIFIKATOR   *
15 REM *   AUTOR: RUEDIGER KERKLOH *
20 REM
25 POKE 53280,0:POKE 53281,0
30 PRINT CHR$(147);CHR$(14);CHR$(8);CHR$(5)
35 PRINT TAB(8);"*** GEOS MODIFIKATOR ***":PRINT
40 PRINT"BITTE LEGEN SIE EINE SICHERHEITSKOPIE"
45 PRINT"DES ZU AENDERNDEN FILES EIN !"
50 GOSUB 440
55 :
60 REM * TRACK UND SEKTOR HOLEN *
65 PRINT CHR$(147):PRINT:PRINT
70 INPUT" FILENAME";X$
75 GOSUB 460:F$=Y$
80 FOR I=1 TO 16-LEN(Y$)
85 :   F$=F$+CHR$(160)
90 NEXT I
95 OPEN 1,8,15,"I:0"
100 OPEN 2,8,2,"#0"
105 PRINT:PRINT" SEARCHING ...":PRINT
110 T=18:S=1
115 PRINT#1,"U1";2;0;T;S
120 GET#2,T$,S$
125 T=ASC(T$+CHR$(0)):S=ASC(S$)
130 FOR I=0 TO 7
135 :   PRINT#1,"B-P";2;I*32+2
140 :   GET#2,Z$,P$,Q$
145 :   IF Z$="" THEN 195
150 :   PRINT" ";
155 :   FOR J=1 TO 16
160 :     GET#2,X$:GOSUB 460
165 :     M$=MID$(F$,J,1)
170 :     IF X$=M$ THEN K=K+1
175 :     PRINT Y$;
180 :   NEXT J
185 :   IF K=16 THEN 220
190 :   K=0:PRINT
195 NEXT I
200 IF T<>0 THEN 115
205 CLOSE 1
210 PRINT:PRINT" FILE EXISTIERT NICHT !"
215 END
220 PRINT:PRINT CHR$(145);">"
225 GET#2,IT$,ISS,SC$

```

```
230 PRINT:PRINT" FILESTRUKTUR: ";
235 IF SC$="" THEN SC$=CHR$(127):PRINT"SEQUENTIELL":GOTO 245
240 SC$=CHR$(0):PRINT"VLIR"
245 PRINT:PRINT
250 PRINT" 1 = TEXT ERSETZEN":PRINT
255 PRINT" 2 = BYTES ERSETZEN"
260 PRINT:PRINT:PRINT" BITTE WAEHLN"
265 GET W$:IF W$<>"1" AND W$<>"2" THEN 265
270 ON VAL(W$) GOSUB 280,320
275 CLOSE1:RUN
280 PRINT CHR$(147):PRINT
285 PRINT" GESUCHTEN TEXT EINGEBEN":PRINT
290 INPUT X$:GOSUB 460:AT$=Y$
295 PRINT:PRINT
300 PRINT" NEUEN TEXT EINGEBEN":X$="":PRINT
305 INPUT X$:X$=X$+CHR$(0):GOSUB460:NT$=Y$
310 GOTO 345
315 :
320 PRINT CHR$(147):PRINT
325 PRINT" GESUCHTE BYTES EINGEBEN:":PRINT
330 GOSUB 520:AT$=Y$:PRINT:PRINT
335 PRINT" NEUE BYTES EINGEBEN:":PRINT
340 GOSUB 520:NT$=Y$
345 IF LEN(AT$)+LEN(NT$)<57 THEN 355
350 PRINT:PRINT" EINGABE ZU LANG !":GOSUB 440:GOTO 245
355 REM * DATEN ZUR FLOPPY UEBERTRAGEN *
360 PRINT#1,"B-P";2;0
365 FOR I=1 TO 196
370 : READ D
375 : PRINT#2,CHR$(D);:W=W+D
380 NEXT I
385 IF W<22349 THEN PRINT:PRINT"FEHLER IN DATAS !":END
390 PRINT#2,SC$; :REM FILESTRUKTUR
395 PRINT#2,P$;Q$; :REM STARTSEKTOR
400 PRINT#2,CHR$(LEN(AT$));
405 PRINT#2,CHR$(LEN(NT$));
410 PRINT#2,AT$;NT$;
415 PRINT#1,"M-E";CHR$(0);CHR$(3)
420 INPUT#1,D,A$,A,A:CLOSE1:PRINT
425 IF D=0 THEN PRINT"DATEN GEFUNDEN UND GEAENDERT !":GOTO 440
430 PRINT"DATEN NICHT GEFUNDEN !"
435 :
440 PRINT:PRINT" <RETURN>"
445 GETW$:IFW$=""THEN445
450 RETURN
455 :
```

```

460 Y$=""
465 FOR A=1 TO LEN(X$)
470 : B=ASC(MID$(X$,A,1))
475 : IF B<192 THEN 485
480 : B=B-96
485 : IF B<65 THEN 500
490 : IFBAND32THENB=BAND223:GOTO500
495 : B=B OR 32
500 : Y$=Y$+CHR$(B)
505 NEXT A
510 RETURN
515 :
520 Y$="":I=1
525 : PRINT" BYTE NR.":"I;":";:INPUT D$
530 : IF VAL(D$)>255 THEN 525
535 : IF D$="" THEN 550
540 : Y$=Y$+CHR$(VAL(D$)):D$=""
545 : I=I+1:GOTO525
550 RETURN
555 REM
560 DATA 172,197,3,173,198,3,174,196,3,208,34,132,8,133,9,169,128, 133,
1,165
565 DATA 1,48,252,238,196,3,173,196,3,16,3,76,94,225,10, 170,189, 0,4,
240,238
570 DATA 168,189,1,4,133,11,132,10,169,128,133,2,165,2,48,252, 164,16,
162,2,
575 DATA 189,0,5,217,201,3,208,8,200,204,199,3,144,6,176,62,160,0,
132,16,232
580 DATA 208,233,152,240,37,132,16,189,0,5,157,0,6,232,208,247,
165,10,133
585 DATA12,165,11,133,13,136,152,73,255,168,174,199,3,189,
201,3,153,0,6,232
590 DATA 200,208,246,173,0,5,240,150,133,10,173,1,5,133,11,24,144,
166,232
595 DATA 138,56,237,199,3,24,101,16,170,173,199,3,24,101,16,168,
185,201,3
600 DATA 157,0,5,200,232,206,200,3,173,200,3,56,229,16,208,237,
162,144,134
605 DATA 2,165,2,48,252,165,16,240,6,134,3,165,3,48,252,76,188,230

```

4.2.3 Bedienung des Modifikators

Nachdem Sie unser Programm eingetippt haben, sollten Sie es zuerst durch SAVE "MODIFIKATOR",8 auf einer Diskette abspeichern. Anschließend fertigen Sie bitte unbedingt für die nun folgenden Arbeiten eine Kopie von GEOS mit dem Programm Sicherheitskopie aus dem Kapitel 4.1 an, damit Sie bei Fehlern, die immer einmal auftreten können, noch auf die Original-Programme zurückgreifen können. Außerdem ist hierdurch gewährleistet, daß die angegebenen Änderungen auch bei der deutschen GEOS Version 1.3 funktionieren, da hier ja das KERNAL in kodierter Form vorliegt. Wenn wir im folgenden von der GEOS-Diskette sprechen, meinen wir nur diese Kopie und niemals das Original. Erst jetzt dürfen Sie den Modifikator wieder laden:

```
LOAD "MODIFIKATOR",8
```

Er fordert Sie nach seinem Start durch RUN auf, eine Kopie von GEOS (die Sie ja soeben erstellt haben) einzulegen und dies durch RETURN zu bestätigen. Anschließend fragt das Programm nach dem Namen des Files, das verändert werden soll. Für unseren ersten Versuch geben Sie hier 'Hallo' ein. Nun wird dieses File auf der Diskette gesucht, indem jeder vorhandene Name eingelesen und mit unserem 'Hallo' verglichen wird. Gleichzeitig werden dabei die eingelesenen Namen in einer Liste auf dem Bildschirm ausgegeben.

In diesem Fall besteht die Liste natürlich aus allen Files, da ein Programm 'Hallo' auf der GEOS-Diskette nicht existiert. Prägen Sie sich bitte genau ein, wie in dieser Liste 'DESK TOP' ausgegeben wird, da wir diesen Namen bei unserem zweiten Versuch eingeben möchten und das Programm Wert auf Groß- bzw. Kleinschreibung legt. Tippen Sie also beim zweiten Anlauf den Namen dieses FONT-Files ein. Wenn es vom Programm gefunden wird, erscheint das folgende Menü:

```
1 = Text ersetzen  
2 = Bytes ersetzen
```

Sie können also nun entscheiden, ob innerhalb der ausgewählten Datei (die in diesem Fall 'Roma' heißt) bestimmte Texte oder aber bestimmte Bytefolgen gegen andere ausgetauscht werden sollen. Dazu fragt das Programm in beiden Fällen zuerst nach den Daten, die ersetzt werden sollen, und anschließend nach den Daten, die an deren Stelle treten sollen.

Es wird also nicht die Adresse einer bestimmten Speicherstelle übermittelt, in die ein neuer Wert geschrieben werden soll, sondern es wird eine Art Suchschlüssel übergeben, der innerhalb eines Programms nur einmal auftauchen darf. Wird dieser Suchschlüssel gefunden, werden genau an dieser Stelle die neuen Werte eingetragen.

Man muß also aufpassen, daß ein Suchschlüssel in einem Programm nicht noch einmal auftaucht (zumindest nicht vor der Stelle, an der die eigentliche Ersetzung stattfinden soll), da sonst die neuen Daten an einer völlig falschen Stelle abgelegt werden würden. Wir haben natürlich bei den in diesem Kapitel vorgeschlagenen Änderungen darauf geachtet, daß die vorgegebenen Texte oder Bytefolgen nur einmal vorkommen.

Für den ersten Versuch geben Sie bitte eine "1" ein für 'Text ersetzen'. Daraufhin werden Sie aufgefordert, den Suchschlüssel, also die zu ersetzende Zeichenkette, einzugeben. Lassen Sie das Programm z.B. nach 'Katze' suchen. Den neuen Text dürfen Sie sich selber aussuchen, denn der durch das anschließende Drücken der RETURN-Taste ausgelöste Suchvorgang wird ohne Erfolg bleiben, da es das Wort 'Katze' im DESK TOP nicht gibt. Somit wird auch kein neuer Text eingetragen. Das Programm "Modifikator" meldet in einem solchen Fall "Daten nicht gefunden".

Sollten Sie bei der Eingabe der DATA-Zeilen des Programms Fehler gemacht haben, wird der Suchvorgang nicht gestartet und eine entsprechende Fehlermeldung ausgegeben. Unser Programm berechnet nämlich, bevor es die Suchroutine in der Floppy aktiviert, eine Prüfsumme über alle DATAS und vergleicht diese

mit einem von uns ermittelten Sollwert. Sollten sich dabei Differenzen ergeben, müssen Sie Ihre abgetippten DATAS noch einmal überprüfen und mit den abgedruckten vergleichen.

Wenn Sie die Meldung 'Daten nicht gefunden!' erhalten haben, versuchen Sie in einem zweiten Anlauf einmal den Menüpunkt 'Bytes ersetzen'. Anschließend werden Sie wieder nach dem Suchschlüssel gefragt, der aber in diesem Fall die Form einzelner Bytes besitzt, die nacheinander eingegeben werden müssen. Geben Sie nun für das 1. Byte einmal eine Fünf ein. Wenn Sie die Frage nach dem 2. Byte einfach mit RETURN beantworten, gilt die Suchschlüssel eingabe als abgeschlossen, und es erscheinen die Fragen nach den neuen, zu ersetzenden Bytes.

Geben Sie auch hier einmal die Fünf für das 1. Byte ein. Wenn Sie bei der Frage nach dem 2. Byte einfach RETURN drücken, beginnt wieder der Suchvorgang, der aber diesmal erfolgreich sein wird. Die Meldung 'Daten gefunden und geändert' sollte dies bestätigen. In dem einige KByte langen Programm DESKTOP wird nämlich an einer Stelle bestimmt einmal eine Fünf auftauchen, die dann, aufgrund unserer Eingabe, durch eine Fünf ersetzt wird. Insgesamt haben wir das File dadurch also nicht verändert. Damit haben wir die Bedienung des Programms auch schon erklärt.

Wir hoffen, daß Sie sich nun vorstellen können, wie mächtig dieses Werkzeug ist. Daß es auch in der Lage ist, bei falschen oder gar willkürlichen Eingaben Programme zu zerstören, bedarf wohl keiner Erläuterung. Daher halten Sie sich bitte genau an die folgenden von uns herausgefundenen Verbesserungsvorschläge einschließlich der korrekten Eingabe der zu suchenden und zu ersetzenden Bytes. Solange das Programm in der Floppy noch nicht gestartet ist (rote Lampe leuchtet noch nicht), können Sie bei einer Fehleingabe das Programm mit RUN/STOP + RESTORE abbrechen und erneut starten.

4.2.4 Erläuterung zur Funktionsweise

Dieser Abschnitt richtet sich hauptsächlich an die Leser unter Ihnen, die Genaueres über die Funktionsweise unseres Modifikators wissen möchten. Sollten Sie daran nicht so sehr interessiert sein, können Sie hier getrost weiterblättern. Zuerst möchten wir den BASIC-Teil erläutern. Für ein genaues Verständnis sollten Sie schon ein wenig Programmiererfahrung in BASIC besitzen. Im zweiten Teil folgt dann die Dokumentation der Suchroutine, die in der Floppy abläuft. Da sie vollständig in Assembler geschrieben wurde, sind Kenntnisse in dieser Sprache notwendig, um den Erklärungen folgen zu können.

Dokumentation des BASIC-Teils

- 25 Bildschirmfarben auf Null (schwarz) setzen.
- 30 Bildschirm löschen, Großschreibung einschalten und arretieren, weiße Schrift.
- 50 Eine Tastatureingabe abwarten.
- 65 Bildschirm löschen.
- 75 Das Unterprogramm ab Zeile 460 wandelt die Zeichen eines Textes um in das unter GEOS benutzte Format. Das Ergebnis wird in Y\$ übergeben.
- 80-90 Der eingegebene Name wird mit "Shift Space" (=CHR\$(160)) auf die maximale Länge von 16 Zeichen gebracht.
- 95 Befehlskanal zur Floppy öffnen und Diskette initialisieren.
- 100 Übertragungskanal für Daten öffnen. Alle Daten, die über diesen Kanal geschickt werden, beziehen sich auf den Puffer Null in der Floppy (ab \$0300 im RAM).
- 110 Track und Sektor auf ersten Directory-Eintrag setzen.

- 115 Befehl "Block lesen" ausführen. Die Daten landen im Puffer Null (T=Track, S=Sektor).
- 120 Track und Sektor des Folgeblocks merken. Die Daten hierüber stehen immer an den ersten beiden Stellen eines Blocks.
- 125 Da bei der Umwandlung eines Tracks 0 in das Integer-Format ein "illegal quantity error" auftreten würde, wird hier ein CHR\$(0) angehängt.
- 130-195 Die 8 Filenamen des eingelesenen Blocks werden mit dem eingegebenen Filenamen verglichen. Ist der Name nicht dabei und noch ein Block vorhanden (T ungleich Null) wird dieser eingelesen, und der Vorgang beginnt von neuem.
- 135 Der Lesezeiger für den Puffer Null wird auf den Filetyp des nächsten Filenamens gesetzt.
- 140 Der Filetyp und Track u. Sektor der Daten werden eingelesen.
- 145 Falls der Filetyp gleich Null ist (= ""), nächsten Eintrag holen.
- 155-180 Die 16 Zeichen des Filenamens einlesen und jeweils mit einem Zeichen des eingegebenen Namens vergleichen.
- 185 "K" ist der Zähler für die Übereinstimmungen. Enthält er eine '16', ist das File gefunden.
- 190 Sonst den Zähler zurücksetzen und weiter suchen.
- 200 Falls noch eine Directory-Seite vorhanden ist, wird diese eingelesen.
- 205-215 File nicht gefunden.
- 225 Im GEOS-Format stehen hinter dem Filenamen weitere Informationen. Es folgen Track und Sektor des Infoblocks (IT\$ u. ISS) und die Filestruktur (SC\$). Der Infoblock enthält Angaben über das Icon

- und die Eintragungen, die in GEOS bei 'File Info' erscheinen. IT\$ und ISS werden im weiteren Verlauf nicht mehr benötigt.
- 235 Eine Null in dem Feld der Filestruktur steht für "SEQUENTIAL".
- 240 Filestruktur=1 VLIR-Format. Die Variable SC\$ wird später noch benutzt und auf den dort benötigten Wert gebracht.
- 245-275 Menüauswahl
- 280-290 Eingabe des alten Textes. Er wird in das GEOS-Format konvertiert und in AT\$ abgelegt.
- 300-310 Das gleiche geschieht mit dem neuen Text. Vorher wird dieser jedoch noch mit einer Null abgeschlossen. Das Ergebnis steht in NT\$.
- 320-330 Eingabe der alten Bytes, nach denen gesucht werden soll.
- 335-340 Eingabe der neuen Bytes.
- 345-350 Da im Floppy-Puffer hinter dem Programm nur noch Platz für insgesamt 56 Bytes ist, erfolgt hier eine Sicherheitsabfrage.
- 360-380 Das Maschinenprogramm wird in den Puffer Null (ab \$0300) kopiert. Gleichzeitig wird dabei eine Prüfsumme berechnet.
- 385 Fehlerbehandlung, falls die Prüfsumme nicht stimmt.
- 390-410 Es werden Daten übermittelt, die das Programm in der Floppy für den Suchvorgang benötigt. Außer der Filestruktur und dem Startsektor der Daten sind dies noch die Länge des alten und neuen Textes und natürlich auch der alte und der neue Text selbst.
- 415 Das Suchprogramm wird gestartet.
- 420-450 Aus dem Fehlerkanal wird die Meldung eingelesen und ausgewertet.

Das Suchprogramm stellt diese Meldung bereit. Sie lautet entweder "O.K." oder "RECORD NOT PRESENT". Es wird jedoch nur die zugehörige Fehlernummer ausgewertet.

- 460-510 Unterprogramm, das einen C64-Text in das unter GEOS benutzte Format umwandelt.
- 520-550 Dieses Unterprogramm übernimmt eine Eingabe von Bytes über die Tastatur in einen Textstring.
- 555-605 Hier steht das eigentliche Suchprogramm in Form von DATAS.

Dokumentation des Suchprogramms

Das Programm ist auf einem bildschirmorientierten Editor entstanden und besitzt somit keine Zeilennummern. Statt dessen befinden sich vor bestimmten Programmschritten sogenannte "Labels" (z.B. 'START'), auf die wir uns in den Erklärungen beziehen. Die Dokumentation erfolgt durch Kommentare innerhalb des Listings und durch anschließende Erläuterungen zur grundlegenden Funktionsweise des Programms.

```

* = $0300 ;Startadresse im Floppy-RAM
JOB1 = $01 ;Job f. Puffer 1
JOB2 = $02 ;Job f. Puffer 2
JOB3 = $03 ;Job f. Puffer 3
JOB4 = $04 ;Job f. Puffer 4
T1 = $08 ;Track f. Puffer 1
S1 = $09 ;Sektor f. Puffer 1
T2 = $0A ;Track f. Puffer 2
S2 = $0B ;Sektor f. Puffer 2
T3 = $0C ;Track f. Puffer 3
S4 = $0D ;Sektor f. Puffer 3
T4 = $0E ;Track f. Puffer 4
S4 = $0F ;Sektor f. Puffer 4
FLAG = $10 ;(s. Text)
;* BLOCK IN PUFFER 1 EINLESEN *
START LDY TRACK ;ZEIGER AUF BLOCK
LDA SEKTOR ;V. BASIC UEBERGEHEN
LDX COUNT ;STRUKTUR ?
BNE GET4 ;SEQUENTIELL!
    
```

```

        STY T1      ;SONST ZEIGER AUF
        STA S1      ;VLIR-LINKERBLOCK
        LDA #$80    ;JOB: SEKTOR LESEN
        STA JOB1    ;IN PUFFER 1
WAIT1  LDA JOB1    ;WARTEN BIS VORGANG
        BMI WAIT1  ;BEENDET IST
        ;* NAECHSTEN EINTRAG HOLEN
GET1   INC COUNT
        LDA COUNT  ;1-127 EINTRAEGE
        BPL GET2   ;KEIN ENDE
        JMP $E15E  ;"RECORD NOT PRESENT"
        ;AUF FEHLERKANAL
GET2   ASL A       ;MAL ZWEI (T & S !)
        TAX
        LDA $0400,X ;START: NEUE DATEI
        BEQ GET1   ;KEIN EINTRAG (T=0!)
        TAY
        LDA $0401,X
GET4   STA S2      ;T & S fuer Puffer 2
        STY T2
GET3   LDA #$80    ;BLOCK EINLESEN
        STA JOB2
WAIT2  LDA JOB2    ;WARTEN BIS VORGANG
        BMI WAIT2  ;BEENDET IST
        ;* DATEN IN PUFFER 2 SUCHEN *
        LDY FLAG   ;POINTER: ALTER TEXT
        LDX #2     ;POINTER IN PUFFER
SRCH2  LDA $0500,X ;GELESENER BLOCK
        CMP TEXT,Y ;ALTER TEXT
        BNE SRCH1  ;UNGLEICH !
        INY       ;ERHOEHEN
        CPY LENA   ;ALLE VERGLICHEN ?
        BCC SRCH3  ;NEIN!
        BCS SRCH4  ;JA
SRCH1  LDY #0     ;FLAG LOESCHEN
        STY FLAG   ;UEBERLAPP-FLAG
SRCH3  INX       ;PUFFERZEIGER
        BNE SRCH2  ;NOCH NICHT ZUENDE
        ;* PUFFER GELESEN *
        TYA       ;UEBEREINSTIMMUNGEN ?
        BEQ SRCH5  ;KEINE, WEITER
        ;* EVTL. UEBERLAPPUNG *
        STY FLAG   ;SETZEN
GRENZ1 LDA $0500,X ;PUFFER 2 -> 3
        STA $0600,X
        INX

```

```

BNE GRENZ1
LDA T2 ;PUFFER 3 DEN T & S
STA T3
;ZUORDNEN
LDA S2
STA S3
DEY
TYA ;ANZAHL IN POINTER
EOR #$FF ;UMWANDELN
TAY ;(Z.B. $01-> $FF)
; (ODER $02-> $FE)
LDX LENA ;NEUEN TEXT IN
SRCH6 LDA TEXT,X ;PUFFER 3 EINTRAGEN
STA $0600,Y
INX
INY
BNE SRCH6
SRCH5 LDA $0500 ;NAECHSTER TRACK
BEQ GET1 ;NICHT VORHANDEN
STA T2
LDA $0501 ;NAECHSTER SEKTOR
STA S2
CLC
;UNBEDINGTER SPRUNG
BCC GET3
SRCH4 ;* WERTE GEFUNDEN *
;
;X-REG: AUF LETZTES ALTES BYTE
INX ;POINTER AUF START-
TXA ;POSITION
SEC
SBC LENA
CLC ;PLUS DIE SCHON EIN-
ADC FLAG ;GETRAGENEN BYTES
TAX
LDA LENA ;AUF AKTUELLES NEUES
CLC ;BYTE ZEIGEN
ADC FLAG
TAY
SET1 LDA TEXT,Y ;NEUEN TEXT
STA $0500,X ;EINTRAGEN
INY
INX
DEC LENB
LDA LENB
SEC

```

```

SBC FLAG
BNE SET1
LDX #$90 ;SEKTOR SCHREIBEN
STX JOB2
WAIT3 LDA JOB2 ;WARTEN BIS VORGANG
      BMI WAIT3 ;BEENDET IST
      LDA FLAG ;UEBERLAPPUNG ?
      BEQ SET2 ;NEIN!
      STX JOB3 ;SONST PUFFER 3
      ;SCHREIBEN
WAIT4 LDA JOB3
      BMI WAIT4
SET2  JMP $E6BC ;"O.K." AUF FEHLER-
      ;KANAL LEGEN
;DATENBEREICH; WIRD VON BASIC VORBELEGT
COUNT * = * + 1 ;LINKER-POINTER
      ;BZW. VLIR FLAG
TRACK * = * + 1
SEKTOR * = * + 1
LENA * = * + 1 ;LÄNGE ALTER TEXT
LENB * = * + 1 ;LÄNGE NEUER TEXT
TEXT * = * + 56 ;ALTER & NEUER TEXT
.END

```

Speicheraufteilung

Innerhalb des Floppy-RAMs werden die Puffer folgendermaßen benutzt:

Puffer 0 (ab \$0300)

Enthält das Suchprogramm aus den DATAS mit den benötigten Variablen, die ebenfalls von BASIC übergeben werden.

Puffer 1 (ab \$0400)

Enthält bei VLIR-Files den ersten Datenblock (Linker). Dieser ist nicht, wie bei Sequential-Files, mit weiteren Blocks verknüpft. Er enthält max. 127 Zeiger auf Track und Sektor der Daten, die nun jeweils wieder wie Sequential-Files verkettet

sind. Nähere Angaben zum VLIR-Format finden Sie im Abschnitt 2.1. Soll ein Programm mit Sequential-Struktur durchsucht werden, wird dieser Puffer nicht benutzt.

Puffer 2 (ab \$0500)

In diesen Puffer werden nacheinander die Blocks des zu durchsuchenden Programms eingelesen und byteweise mit den von BASIC übergebenen Werten ("TEXTA") verglichen.

Puffer 3 (ab \$0600)

Zwischenspeicher für einen Programmblock. Sollte der Fall eintreten, daß sich die zu suchenden Daten genau am Ende des in Puffer 2 eingeladenen Programmteils befinden, jedoch noch nicht alle Bytes verglichen worden sind, kann noch nicht eindeutig entschieden werden, ob der eingelesene Block nun tatsächlich der richtige ist. Daher werden zwar die gewünschten Änderungen in Puffer 2 durchgeführt, aber der Block noch nicht auf die Diskette zurückgespeichert.

Vielmehr wird der Puffer 2 danach in den Puffer 3 kopiert und anschließend der Folgesektor des Programms in Puffer 2 eingelesen. Erst wenn sich hier bestätigt, daß die Stelle korrekt war, werden die neuen Bytes auch hier eingetragen und anschließend beide Puffer (2 und 3) zurückgeschrieben. Die Speicherstelle FLAG enthält bei solchen Überlappungen stets die Anzahl der Bytes, die schon in Puffer 3 ersetzt wurden. Eine Null signalisiert also, daß keine Überlappung stattgefunden hat.

Erklärungen zum Programmablauf

Man muß zwei Betriebsarten des Suchprogramms unterscheiden. In der ersten Betriebsart besitzt das zu untersuchende Programm die File-Struktur Sequential, in der zweiten ist die Struktur VLIR. Die Information, um welche File-Struktur es sich jeweils handelt, wird vom BASIC-Programm durch die Variable SC\$ übermittelt. Hierbei bedeutet '0' VLIR und '127' Sequential. Der Wert wird bei der Übertragung in die Variable COUNT übernommen.

Handelt es sich um ein Sequential-File, wird zu Beginn nach der Abfrage der Struktur ein Programmabschnitt übersprungen. Dieser Abschnitt, der bis GET4 reicht, dient bei VLIR-Files dazu, den sogenannten LINKER in den Puffer 1 (ab \$0400) einzuladen. Außerdem wird aus diesem Linker der Track und Sektor des nächsten Eintrags geholt.

Bei GET4 enthält das Y-Register bzw. der Akku den Track und Sektor des ersten Datenblocks. Bei Sequential-Files stammen diese Daten aus dem BASIC-Programm, bei VLIR-Files aus den Eintragungen des Linkers. Dieser Datenblock wird bei GET3 in den Puffer 2 (ab \$0500) eingelesen. Anschließend erfolgt der Vergleich mit den alten Daten (TEXTA). Die Funktion der Speicherstelle FLAG wurde oben bereits erwähnt.

Wenn der eingelesene Programmblock durchsucht worden ist (* Puffer gelesen *), wird getestet, ob es zu Überlappungen gekommen ist. Falls nicht, wird der nächste Programmblock eingelesen und weiter getestet (SRCH5). Existiert kein weiterer Block, wird bei GET1 versucht, ein neues RECORD aus dem Linker zu holen. Da dies jedoch nur bei VLIR-Files existiert, ist die Abfrage so ausgelegt, daß bei Sequential-Files sofort ein "RECORD NOT PRESENT" ausgegeben wird. Dies liegt daran, daß von BASIC bei solchen Files in die Speicherstelle COUNT nicht eine Null, wie bei VLIR-Files, sondern eine 127 übergeben wird. Ein inkrementieren auf 128 läßt die anschließende bedingte Abfrage natürlich sofort negativ ausfallen, und der Suchvorgang ist beendet.

Sollte es zu Überlappungen gekommen sein, erfolgt bei GRENZ1 die Kopie des Puffers 2 in den Puffer 3. Außerdem wird der zugehörige Track und Sektor gespeichert, um den Puffer 3 später evtl. abspeichern zu können. Da das Y-Register die Anzahl der Übereinstimmungen enthält, kann hieraus leicht ein Zeiger gebogen werden, der auf die in Frage kommenden Bytes im Puffer 3 zeigt. Diese Bytes werden anschließend durch die neuen ersetzt. Dazu wird durch LENA der notwendige Offset zum neuen TEXT erzeugt.

Sollten bei dem Vergleichsvorgang (SRCH2) alle alten Bytes gefunden werden, wird zu SRCH4 verzweigt. Hier erfolgt nun die Eintragung der restlichen neuen Bytes, die noch nicht im Puffer 3 eingetragen wurden. Falls keine Überlappung stattgefunden hat, sind dies natürlich alle neuen Bytes. Anschließend wird der Puffer 2 mit den geänderten Daten zurückgeschrieben. Sollten Überlappungen aufgetreten sein, muß auch der Puffer 3 zurückgeschrieben werden. Das Programm erkennt dies am Inhalt der Speicherstelle COUNT. Schließlich wird noch die O.K.-Meldung auf den Fehlerkanal gelegt.

4.2.5 Anwendungen für den Modifikator

Jetzt steht der ersten sinnvollen Anwendung des Modifikators eigentlich nichts mehr im Wege. Bevor es jedoch richtig losgeht, möchten wir noch kurz die 'Spielregeln' erläutern. Jeder von uns gebrachte Änderungsvorschlag bezieht sich auf ein ganz bestimmtes File auf der GEOS-Diskette (der Kopie!). Den Namen dieses Files werden wir natürlich jedesmal angeben und ihn dabei genau so schreiben, daß unser Programm es auch findet (Groß- bzw. Kleinschreibung!). Da außerdem verschiedene GEOS-Versionen im Umlauf sind, werden wir bei jedem Änderungsvorschlag angeben, auf welche GEOS-Versionen die angegebene Änderung angewendet werden darf. Teilweise ergibt sich dies auch schon aus der Überschrift der Änderung. So sind z.B. die deutschen Fehlermeldungen für GEOS V1.3 uninteressant, da diese Version schon vollständig eingedeutscht ist. Eine große Lesergruppe wird jedoch diesen Änderungsvorschlag für ihre Version 1.2 begrüßen.

Aus dem Zusammenhang heraus werden Sie außerdem feststellen, ob es sich bei den Ersetzungen um Texte oder um einzelne Bytes handelt. Dementsprechend müssen Sie den richtigen Menüpunkt im Modifikator auswählen. Wenn wir zum Beispiel 'Printer' gegen 'Drucker' austauschen möchten, geht dies natürlich am komfortabelsten durch das Menü 'Text ersetzen'.

Natürlich könnte man auch den Menüpunkt 'Bytes ersetzen' auswählen. Da das Programm aber in diesem Fall nur Dezimalzahlen annimmt, müßte man erst die entsprechenden ASCII-Werte der Zeichen im alten und neuen Text bestimmen. Sollten Sie dennoch einmal auf diese Weise einen Text ersetzen, denken Sie bitte daran, diesen durch eine Null zu beenden. Eine Null gilt in GEOS nämlich allgemein als Abschlußkennzeichen. Sie wird bei 'Text ersetzen' automatisch an den neuen Text angefügt.

Ferner sollten Sie beachten, daß ein neuer Text wohl kürzer, aber niemals länger sein darf als der alte, da sonst wichtige Daten innerhalb des Programms überschrieben werden können. Auf eine Besonderheit bei den von uns angegebenen dezimalen Suchschlüsseln möchten wir zuletzt noch hinweisen.

Die von uns angegebenen Suchbytes sind immer genau die Werte, die direkt vor den eigentlichen Werten stehen, die geändert werden sollen. Daher müssen diese jedesmal wieder mit in die Liste der neuen Bytes aufgenommen werden. Der Vorteil dieser Methode besteht darin, daß der Modifikator die Stelle jederzeit wiederfindet, auch wenn sie schon vorher einmal manipuliert wurde. Ein erfundenes Beispiel soll Ihnen beim Verständnis dieses etwas schwierigen Sachverhaltes helfen. Im Programm 'DESK TOP' sei eine Stelle mit der folgenden Bytefolge:

```
"11/123/17/90/240/21/11"
```

Ferner sei die 240 verantwortlich für die Blinkgeschwindigkeit des Cursors. Wenn hier also ein kleinerer Wert steht, blinkt der Cursor schneller. Nun könnte man mittels 'Bytes ersetzen' nach der Abfolge 240/21/11 suchen und durch 100/21/11 einen schnelleren Cursor erzielen. Der große Nachteil ist aber, daß man sich nun für eine evtl. spätere Änderung die 100 merken muß, da sie ja nun anstelle der 240 steht. Daher gehen wir den anderen Weg und lassen nach "11/123/17/90" suchen. Für die neuen Bytes muß man hier nun einfach den Cursorwert anhängen.

```
"11/123/17/90/100"
```

Man wird auf die Art diese Stelle immer wiederfinden, egal, welchen Cursor-Wert man einmal eingetragen hat. So, und nun viel Spaß bei den Änderungen.

Deutsche DESK TOP Meldungen (nur GEOS V1.2)

Mit Hilfe des Modifikators ist es möglich, beliebige Texte innerhalb eines Files gegen neue Texte zu ersetzen. Dies hat uns auf die Idee gebracht, die amerikanischen Fehlermeldungen der Version 1.2 im DESK TOP gegen deutsche Meldungen zu ersetzen. Daß dieses nur mit einer gewissen Kompromißbereitschaft möglich ist, wird sofort klar, wenn man einmal versucht, 'Turn on' in ein einigermaßen verständliches Deutsch zu übersetzen, ohne dabei die vorgegebene Maximal-Länge von sieben Buchstaben zu überschreiten.

In den meisten Fällen ist uns aber eine akzeptable Übersetzung eingefallen. Wenn Sie bei einigen Punkten bessere Vorschläge haben, können Sie natürlich auch diese nehmen. Einzige Bedingung ist nur, daß Ihr Text höchstens genauso lang ist, wie der im Original-DESK TOP. Bei der Eingabe des alten Textes müssen Sie auf die genaue Schreibweise achten, damit der Modifikator die Daten auch findet. Hier nun die im DESK TOP vorhandenen alten Meldungen, bei denen uns eine relativ sinnvolle Übersetzung möglich war.

1. *Operation canceled due to*
Vorgang abgebrochen
2. *Disk error*
Diskfehler
3. *This is a NON-GEOS disk*
Dies ist keine GEOS Disk
4. *Would you like it converted?*
Soll es eine werden?

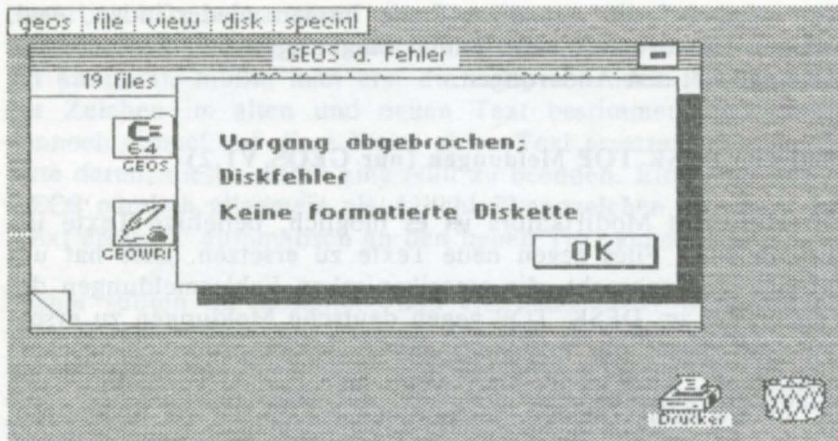


Abb. 14: Keine formatierte Diskette

5. *K bytes used*
K belegt
6. *K bytes free*
K frei
7. *Please insert disk*
Bitte legen Sie ein

Vielleicht wundern Sie sich an dieser Stelle, warum der neue Text länger als der alte ist. Eigentlich lautet der alte Text:

"Please insert disk:"

Da der INPUT-Befehl beim C64 aber keinen Doppelpunkt mag, haben wir ihn in der Eingabe des zu suchenden Textes nicht aufgenommen. Der Platz steht dadurch aber genau für den neuen Text zur Verfügung.

8. *A maximum*
Maximal acht Dateien dürfen

Leider können nur insgesamt 50 Daten-Bytes an die Floppy übergeben werden. Daher mußten wir den Rest des englischen Textes weglassen. Der englische Text

"A maximum of eight files may"

wird natürlich trotzdem gefunden und ist genau so lang wie der deutsche Text.

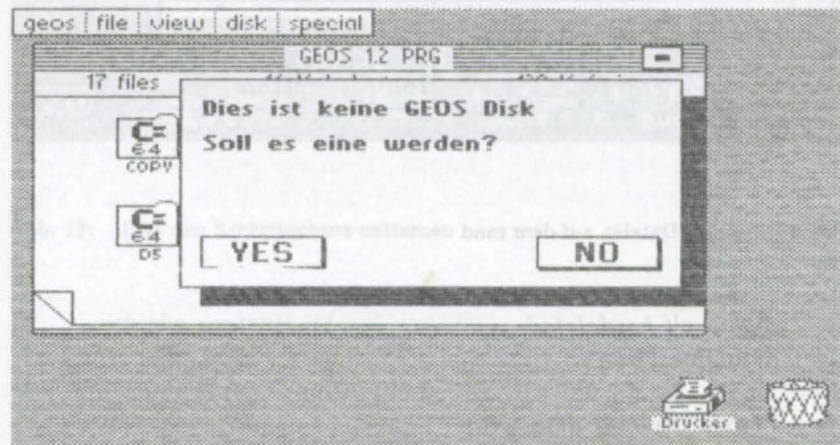


Abb. 15: ...soll es eine werden?

9. *be placed on the border*
am Rand abgelegt werden
10. *This file can't be printed*
Druck nicht moeglich
11. *from the deskTop*
vom DESK TOP aus
12. *This file is*
Geschützte Dateien kann man
Original:"This file is write protected"

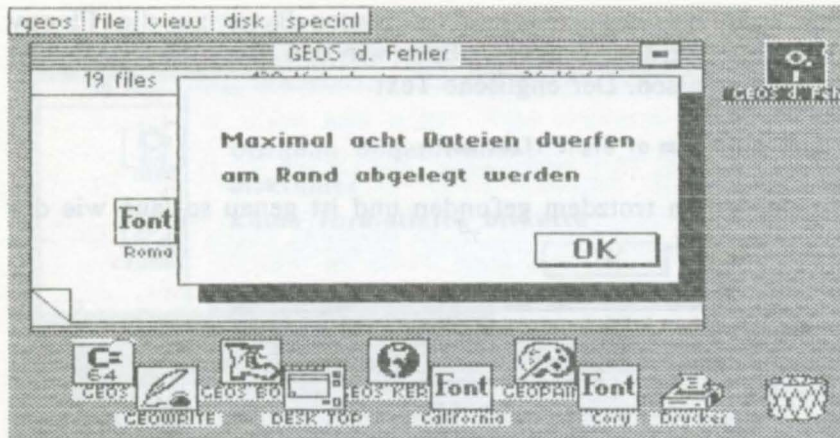


Abb. 16: Nur 8 Dateien auf dem rand

13. *and can't be deleted*
nicht loeschen
14. *Please insert a disk with*
Bitte eine Diskette mit
15. *already exists*
gibt es schon
16. *Please insert destination*
Zieldiskette einlegen
17. *Put disk to format*
Unformatierte Disk in Laufw. A
Original: "Put disk to format in drive: A"
18. *and enter a name for it*
einlegen und benennen
19. *Replace the contents of*
Ersetze den Inhalt von

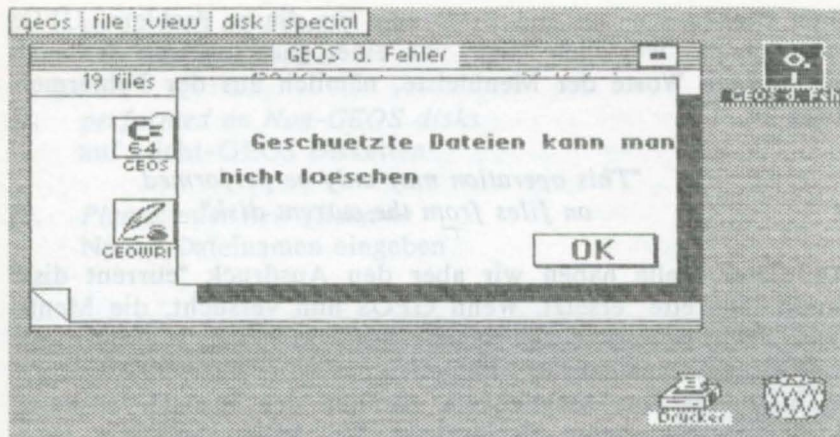


Abb. 17: Erst den Schreibschutz entfernen

- 20. *with the contents of*
mit dem Inhalt von
- 21. *Plug in*
Neues Laufwerk einschalten
Original: "Plug in & turn ON new drive."
- 22. *This operation may only be*
Dieser Vorgang geht nur
- 23. *performed on files from the*
bei Dateien der aktuellen
- 24. *current disk*
Scheibe

Vielleicht wundern Sie sich an dieser Stelle über die etwas laxe Schreibweise "Scheibe". Von der Länge hätte hier auch ohne weiteres "Diskette" gepaßt. Wir haben das an dieser Stelle auch probiert und uns dann sehr gewundert, als anschließend oben in der Menüleiste der Punkt "disk" völlig fehlte. Nach langem Su-

chen stießen wir auf die Erklärung für dieses Problem. GEOS holt sich das Wörtchen "disk" von einer ganz anderen Stelle als die übrigen Worte der Menüleiste, nämlich aus der Fehlermeldung 24:

*"This operation may only be performed
on files from the current disk".*

An dieser Stelle haben wir aber den Ausdruck "current disk" durch "Diskette" ersetzt. Wenn GEOS nun versucht, die Menüleiste aufzubauen und den Begriff "disk" zu holen, beginnt dieser mit dem Endezeichen "0", das genau das Wort "Diskette" beendet. So blieb nur eine Möglichkeit, nämlich den Begriff "Diskette" durch einen kürzeren zu ersetzen. Wir hätten natürlich lange nach einem Begriff suchen können, aber der Ausdruck "Scheibe" hat uns im Zusammenhang mit dem gerade beschriebenen Phänomen so fasziniert, daß wir ihn benutzt haben.

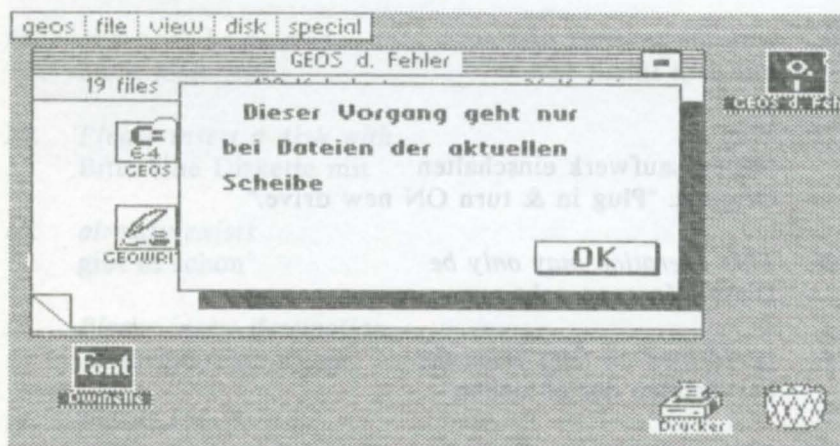


Abb. 18: Die Diskette als Scheibe

25. *This operation may not be*
Dieser Vorgang geht nicht

- 26. *performed on System files*
mit System-Dateien
- 27. *performed on Non-GEOS disks*
auf Nicht-GEOS Disketten
- 28. *Please enter new filename*
Neuen Dateinamen eingeben

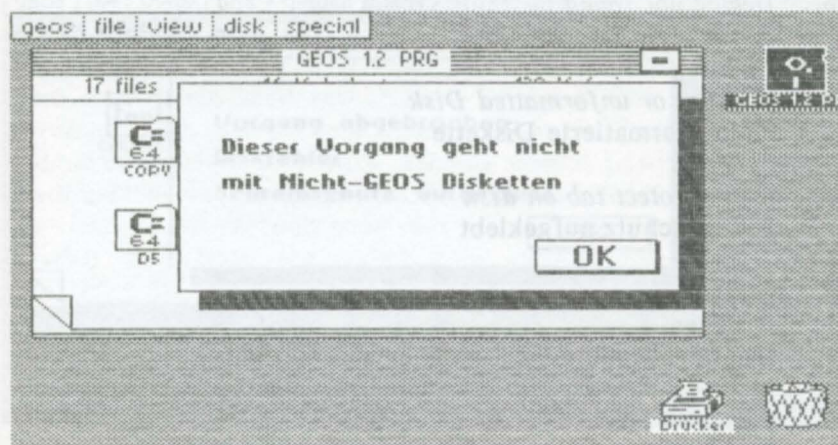


Abb. 19: GEOS-Disketten werden bevorzugt

- 29. *Please enter new disk name:*
Diskettennamen eingeben:
- 30. *desktop not on disk*
Desktop fehlt
- 31. *File from other disk*
File von anderer Disk
- 32. *Write Protect*
Schreibschutz

33. *Disk too full*
Diskette voll
34. *Directory full*
Directory voll
35. *File not found*
Datei nicht da
36. *Device not found*
Gerät nicht da
37. *Missing or unformatted Disk*
Keine formatierte Diskette
38. *Write protect tab on disk*
Schreibschutz aufgeklebt
39. *Printer*
Drucker
(Dieser Name muß zweimal ersetzt werden)
40. *printer*
Drucker
41. *This file can't be opened*
Datei ist nicht ladbar
42. *In drive*
In Laufwerk
Original: "In drive:"
43. *Canceled*
Abbruch!
44. *DRIVE A*
Laufwerk A
45. *DRIVE B*
Laufw.B

Eigentlich läßt sich im DESKTOP alles, was man irgendwann einmal zu lesen bekommt, ersetzen. Sie können sich z.B. auch ein ganz persönliches GEOS anfertigen, indem Sie anstelle der Namen, die in den Informationsfenstern (GEOS INFO, DESKTOP INFO) stehen, Ihren eigenen einsetzen (Für Raubkopierer ist dies vielleicht eine nicht ganz ernst zu nehmender Tip!). X4 für die Stunde (0 bis 24).

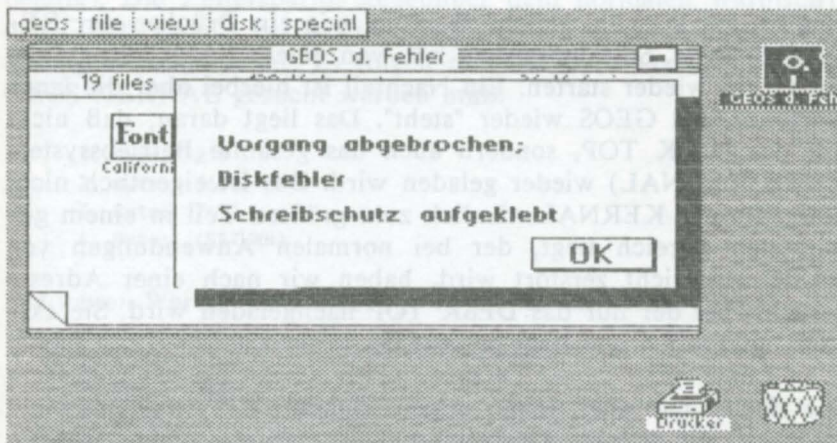


Abb. 20: Letzte Rettung Schreibschutz

GEOS Warmstart (nur GEOS V1.2)

Daß GEOS in Maschinensprache programmiert ist, wird für Sie bestimmt keine Neuigkeit mehr sein. Häufig kann man solche Programme an der einzig vorhandenen BASIC-Zeile mit dem SYS-Aufruf erkennen. Da steht dann z.B. nur:

```
10 SYS 2064
```

Die 2064 ist dabei die Startadresse des Maschinenprogramms. In GEOS wird man so etwas nicht finden, da BASIC von Anfang an komplett abgeschaltet wird. Trotzdem muß es aber irgendeine Startadresse geben, denn ein Drücken der RESTORE-Taste be-

wirkt nach einem Ausstieg ins BASIC ein erneutes Booten von GEOS. Diese Startadresse lautet dezimal:

49152

Wenn man also nach einem RESET

SYS 49152 (RETURN)

eingibt, hat dies die gleiche Wirkung wie ein Drücken der RESTORE-Taste. So könnte man z.B. von einem BASIC-Programm aus GEOS wieder starten. Ein Nachteil ist hierbei aber die lange Wartezeit, bis GEOS wieder "steht". Das liegt daran, daß nicht nur das DESK TOP, sondern auch das gesamte Betriebssystem (GEOS KERNAL) wieder geladen wird. Das ist eigentlich nicht nötig. Da das KERNAL nämlich zum größten Teil in einem geschützten Bereich liegt, der bei normalen Anwendungen von BASIC aus nicht zerstört wird, haben wir nach einer Adresse gesucht, bei der nur das DESK TOP nachgeladen wird. Sie existiert tatsächlich und liegt bei 49155.

Die Sache hat aber leider einen Haken. Wir staunten nicht schlecht, mit welcher Geschwindigkeit man aus einem BASIC-Programm wieder das gewohnte GEOS-Bild bekommen kann, wenn man durch SYS 49155 einen Warmstart auslöst. Umso größer war jedoch die Enttäuschung, als wir feststellen mußten, daß die Maus wohl in der Zeit ohne GEOS verhungert war. Sie ließ sich nicht mehr bewegen. Die Ursache hierfür war schnell gefunden. Bei jeder Initialisierung des KERNAL wird eine komplizierte Berechnung durchgeführt, deren Ergebnis die Startadresse der Bewegungsroutinen für die Maus ist (Für Profis: Der IRQ-Vektor wird berechnet).

Sollte bei dieser Berechnung ein falscher Wert herauskommen, wird GEOS zwar initialisiert, die Maus gibt jedoch keinerlei Lebenszeichen mehr von sich, da völlig falsche Routinen angesprungen werden. Grundlagen der Berechnung sind dabei Daten, die sich innerhalb des GEOS-BOOT-Programms befinden. Bei einem Warmstart, wie wir ihn vorhaben, ist jedoch das Boot-

Programm schon längst nicht mehr im Speicher vorhanden, da es im ungeschützten BASIC-Bereich gelegen hat.

Mit dem Modifikator ist es nun aber möglich, die ganze Berechnung im KERNAL abzukürzen, indem man hier von vornherein das richtige Ergebnis einträgt. Mit einem so veränderten KERNAL kann dann ohne weiteres ein Warmstart mit SYS 49155 durchgeführt werden (und die Maus ist danach quicklebendig). Die Zeitersparnis gegenüber dem normalen Warmstart, den GEOS bietet, ist dabei so groß, daß sich diese kleine Änderung wirklich lohnt. Es folgen nun die Bytes, nach denen im GEOS KERNAL gesucht werden muß:

1. Byte: 219
2. Byte: 165
3. Byte: 7
4. Byte: (RETURN)

Die neuen Werte, die das fertige Ergebnis enthalten, lauten:

1. Byte: 219
2. Byte: 165
3. Byte: 7
4. Byte: 169
5. Byte: 226
6. Byte: 141
7. Byte: 255
8. Byte: 255
9. Byte: 165
10. Byte: 6
11. Byte: 169
12. Byte: 220
13. Byte: (RETURN)

Mit einer weiteren kleinen Änderung muß nun noch der verfügbare BASIC-Speicher ein wenig eingeschränkt werden. Normalerweise steht dem Anwender für eigene Programme der Speicher bis \$A000 zur Verfügung. Nun liegen aber Teile des GEOS KERNAL von \$9000 bis \$A000, so daß dieser Bereich (also 4 Kbyte) vor BASIC geschützt werden muß. Dazu suchen Sie bitte im GEOS KERNAL nach den Bytes:

1. Byte: 130
2. Byte: 2
3. Byte: (RETURN)

Die folgenden Bytes bewirken dann automatisch bei einem Ausstieg aus GEOS durch BASIC eine Begrenzung des Speichers:

1. Byte: 130
2. Byte: 2
3. Byte: 169
4. Byte: 144
5. Byte: (RETURN)

Wenn anstelle der 144 eine 160 eingegeben wird, steht wieder der gesamte Speicherplatz zur Verfügung. In einem solchen Fall werden aber beim Anlegen von BASIC-Variablen Teile des KERNAL zerstört, und ein schnelles Booten ist nicht mehr möglich. Wenn Sie diese Änderungen mit Erfolg durchgeführt haben, sollten Sie die korrekte Funktion einmal überprüfen.

Nach dem Booten mit dieser Diskette wählen Sie anschließend unter SPECIAL den Punkt BASIC an. Nun sollte sich GEOS durch SYS 49155 (RETURN) in erheblich kürzerer Zeit als gewohnt in den ursprünglichen Zustand versetzen lassen. Vorher sollten Sie aber ein kleines BASIC-Programm schreiben, das einige Variablen und Arrays benutzt, damit auch die Funktion der Speicherbegrenzung kontrolliert werden kann. Diese Daten würden nämlich sonst sofort im Bereich des KERNAL abgelegt.

Cursor-Blinkfrequenz ändern (GEOS V1.2)

Eine Eigenschaft, die sich mit Hilfe des PREFERENCE MANAGER (Voreinstellungsmanager) nicht einstellen läßt, ist die ein wenig zu niedrig geratene Blinkfrequenz des Cursors unter GEOS. Da ihm ein bißchen mehr Vitalität sicher nicht schaden wird, haben wir im GEOS KERNAL die Stelle ausfindig gemacht, die für die Frequenz verantwortlich ist. Suchen Sie hier nach den Werten:

1. Byte 41
2. Byte 192

3. Byte 9
4. Byte (RETURN)

Die neuen Werte sind dann:

1. Byte 41
2. Byte 192
3. Byte 9
4. Byte X
5. Byte (RETURN)

Für "X" können Sie einen beliebigen Wert von 1 bis 63 einsetzen. Mit X=20 haben wir einen Wert gefunden, der unserem Geschmack entspricht.

Repeat-Frequenz einstellen (GEOS V1.2)

Wiederum handelt es sich um eine Änderung am Cursor, die demzufolge auch im GEOS KERNAL vorzunehmen ist. Sie bezieht sich auf die Wiederholungsfunktion, die immer dann in Aktion tritt, wenn eine Taste bei einer Texteingabe längere Zeit gedrückt wird. Wenn Sie z.B. in GEOWRITE länger auf das "A" drücken, wird der Buchstabe so oft ausgegeben, bis Sie die Taste wieder loslassen. Genau hier setzt nun unsere Kritik an.

Für eine sinnvolle Anwendung dieser Funktion ist der Zeitraum zwischen der Ausgabe der einzelnen Zeichen zu groß gewählt. Man kann durch mehrmaliges Drücken einer Taste sogar schneller arbeiten, als es diese Repeat-Funktion ermöglicht. Bei der Suche nach den dafür zuständigen Bytes im GEOS KERNAL sind wir auf zwei verschiedene Speicherstellen getroffen.

Die erste von ihnen bestimmt die Länge der Zeit, die verstreichen soll, bis die Repeat-Funktion überhaupt anspricht. Hier kann sogar ein Wert eingetragen werden, der die Wiederholungsfunktion komplett abstellt. In der zweiten Speicherstelle steht nun der Wert, der bei aktiver Repeat-Funktion für die langen Pausen zwischen der Ausgabe der einzelnen Zeichen verantwortlich ist. Steht hier ein extrem kleiner Wert, hat man im Nu eine ganze Zeile mit einem Zeichen vollgeschrieben. Hier

nun die Bytes, nach denen gesucht werden muß, wenn man die erste Eigenschaft (Zeitdauer bis zum Einsetzen der Wiederholungsfunktion) verändern möchte:

1. Byte 135
2. Byte 240
3. Byte 16
4. Byte 169
5. Byte (RETURN)

Die neuen Bytes lauten:

1. Byte 135
2. Byte 240
3. Byte 16
4. Byte 169
5. Byte X
6. Byte (RETURN)

Für das "X" können dabei Werte von 1 (kurze Zeit) bis 255 (Repeat ausgeschaltet) eingesetzt werden. Um die zweite Eigenschaft, also die eigentliche Wiederholungsfrequenz zu ändern, muß nach den folgenden Bytes gesucht werden:

1. Byte 248
2. Byte 228
3. Byte 169
4. Byte (RETURN)

Sie müssen ersetzt werden durch:

1. Byte 248
2. Byte 228
3. Byte 169
4. Byte X
5. Byte (RETURN)

Das "X" als 4. Byte stellt hier wieder den veränderbaren Wert dar. Es sind Werte von 1 (ganz schnell) bis 255 (extrem langsam) möglich. Als Anhaltspunkt für eigene Einstellungen sei hier noch erwähnt, daß in beiden Speicherstellen standardmäßig eine 15 eingetragen ist.

Tabs und Ränder bei GEOWRITE (GEOS V1.2 und GEOS V1.3)

Diesmal möchten wir einen sehr nützlichen Verbesserungsvorschlag machen, der sich auf das Textverarbeitungsprogramm GEOWRITE bezieht. Wenn man mit diesem Programm einen neuen Text erstellt, sind die beiden Randbegrenzungen und die acht verfügbaren Tabulatoren standartmäßig auf sehr ungünstige Werte gesetzt.

In den meisten Fällen wird man nämlich die Ränder zuerst einmal so legen, daß das gesamte Schriftstück auf den angezeigten Bildschirmbereich paßt. Dadurch entfällt das sonst notwendige Umblättern auf die zweite Bildschirmhälfte beim Erreichen des rechten Randes, und man hat so den gesamten Text im Auge. Wenn man fertig ist, kann man die Ränder anschließend immer noch mit der Maus auf die gewünschte Position bringen.

Nun ist jedoch der rechte Rand in seiner Grundeinstellung außerhalb des sichtbaren Bereichs, so daß jedesmal, wenn man einen neuen Text erstellt, auf der Einstelleiste das "M" gesucht und weiter nach links geholt werden muß. Diese Einstellung können Sie sich in Zukunft sparen, wenn Sie eine kleine Änderung in GEOWRITE vornehmen. Innerhalb dieses Programms haben wir nämlich die Stellen gefunden, die die Default-Werte, also die voreingestellten Werte enthalten. Von den hierfür reservierten 20 Speicherstellen entfallen vier auf die beiden Randpositionen und die restlichen 16 auf die acht möglichen Tabulatoren. Es sind also pro Markierung (Rand bzw. Tabulator) jeweils zwei Speicherstellen vorgesehen.

Der Grund hierfür liegt darin, daß man in einem Byte nur Werte von 0 bis einschließlich 255 darstellen kann, GEOWRITE sich aber die Positionen im Pixelformat merkt, bei dem der minimale Abstand von Tabulator zu Tabulator acht Punkte beträgt. Bei den 60 möglichen Tabulatorpositionen kommt man dabei auf Werte bis zu 479, die sich in einem Byte alleine nicht mehr darstellen lassen. Daher führt man ein weiteres Byte für den Über-

trag ein. Wenn also z.B. an der 53 Position (von null an gezählt) eine Markierung stehen soll, berechnet man die Werte für die Speicherstellen folgendermaßen:

$$53 * 8 = 424 \text{ (Pixelwert der Position)}$$

Da die 424 einmal durch 256 geteilt werden kann, erhält das höherwertige Byte eine eins, das niederwertige Byte den bei der Division aufgetretenen Rest von 168. Mit diesem Verfahren kann nun jeder der zwei Rand- bzw. der acht Tabulatorpositionen ein neuer Wert zugeordnet werden. Die Stelle in GEOWRITE, die die Default-Werte enthält, kann mit dem Modifikator bei der Suche nach den folgenden Bytes gefunden werden:

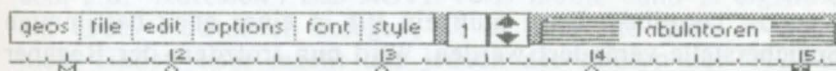
1. Byte: 251
2. Byte: 96
3. Byte: (RETURN)

Die nun folgende Tabelle enthält schon die von uns geänderten Werte für die Rand-Positionen. Außerdem sind die ersten drei Tabulatoren gesetzt. Damit Sie hier auch eigene Werte einsetzen können, sind die Bedeutungen der Bytes erklärt. Dabei steht "LO" für "niederwertiges Byte" und "HI" für "höherwertiges Byte".

1. Byte: 251
2. Byte: 96
3. Byte: 24 (linker Rand, LO)
4. Byte: 0 (linker Rand, HI)
5. Byte: 48 (rechter Rand, LO)
6. Byte: 1 (rechter Rand, HI)
7. Byte: 64 (1. Tabulator, LO)
8. Byte: 0 (1. Tabulator, HI)
9. Byte: 144 (2. Tabulator, LO)
10. Byte: 0 (2. Tabulator, HI)
11. Byte: 224 (3. Tabulator, LO)
12. Byte: 0 (3. Tabulator, HI)
13. Byte: 48 (4. Tabulator, LO)
14. Byte: 1 (4. Tabulator, HI)
15. Byte: 48 (5. Tabulator, LO)
16. Byte: 1 (5. Tabulator, HI)
17. Byte: 48 (6. Tabulator, LO)

- 18. Byte: 1 (6. Tabulator, HI)
- 19. Byte: 48 (7. Tabulator, LO)
- 20. Byte: 1 (7. Tabulator, HI)
- 21. Byte: 48 (8. Tabulator, LO)
- 22. Byte: 1 (8. Tabulator, HI)
- 23. Byte: (RETURN)

Wenn Sie hier eigene Werte einsetzen möchten, müssen Sie darauf achten, daß alle gesetzten Tabulatoren links von dem rechten Rand liegen, also kleine Werte bekommen. Nicht gesetzte Tabulatoren (wie bei unseren Werten die Tabs 4 - 8) werden auf den rechten Rand gesetzt. Sollte auch nur ein Tabulator größer sein als der rechte Rand, beantwortet GEOWRITE dieses nämlich mit einem gnadenlosen Absturz.



So setzt GEOWRITE die Ränder und Tabulatoren, wenn Sie die angegebenen Bytes benutzen. Wenn Sie die restlichen 5 Tabulatoren auch setzen wollen, ersetzen Sie "48" und "1" durch andere Werte. Mit dieser Voreinstellung läßt sich wirklich komfortabel arbeiten.

(Natürlich ebenso mit den deutschen Umlauten, die GEOWRITE mit Hilfe des FONT-EDITORS erhalten hat.)

Abb. 21: Neue Ränder und Tabulatoren

Die Korrektur der Uhr (GEOS V1.2)

Eine Frage, die viele GEOS-Fans immer wieder an die Redaktionen der Computer-Zeitschriften richten, lautet: "Warum läuft in GEOS die eingebaute Uhr viel zu langsam?" Die hierauf gegebenen Antworten hatten jedoch immer nur tröstenden Charakter. Da wurde dann z.B. eine neue GEOS-Version angekün-

digt (die mittlerweile in Form von GEOS V1.3 auf dem Markt ist), die diesen Fehler nicht mehr enthalten soll, oder es wurde zwar eine mögliche Ursache genannt, die jedoch so schwammig formuliert wurde, daß sie dem Leser auch nicht viel weiterhalf.

Wir haben die genaue Ursache jedoch herausgefunden: Die eigentliche Uhr befindet sich in einem vielseitigen Baustein des C64. Bevor Sie jedoch nun das Gehäuse Ihres Rechners aufschrauben und hoffnungsvoll nach einem Pendel suchen, lesen Sie noch die folgenden Sätze. Dieser Baustein nutzt die recht stabile Netzfrequenz von 50 Hz für die Ganggenauigkeit der Uhr. Nun ist es aber nicht etwa so, daß starke Schwankungen dieser Frequenz die Ungenauigkeit verursachen. Vielmehr ist eine falsche Programmierung des Bausteins daran schuld. Es besteht nämlich auch die Möglichkeit, die Uhr an einem 60-Hz-Netz zu betreiben. Dazu muß ein Bit in einer ganz bestimmten Speicherstelle verändert werden. Wird nun trotzdem der Rechner an 50 Hz betrieben, geht die Uhr zu langsam. Genau dies ist in GEOS der Fall.

Es genügt also, die Stelle zu finden, in der auf 60 Hz umgestellt wird, und hier einen anderen Wert einzutragen. Die folgenden Bytes ermöglichen es, innerhalb des GEOS KERNAL diese Veränderung vorzunehmen. Sie sollten anschließend auf allen Ihren bootbaren GEOS-Kopien das KERNAL gegen diese veränderte Version austauschen. Die zu suchenden Bytes lauten:

1. Byte: 13
2. Byte: 220
3. Byte: 3
4. Byte: 127
5. Byte: (RETURN)

Sie müssen ersetzt werden durch:

1. Byte: 13
2. Byte: 220
3. Byte: 3
4. Byte: 127
5. Byte: 128
6. Byte: (RETURN)

Mehrere Notes erstellen (GEOS V1.2 und GEOS V1.3)

Leider erlaubt das "NOTE PAD" nur die Erstellung eines einzigen Notizbuches pro Diskette. Es ist daher auch nicht notwendig, diesem einen Namen zu geben, wenn man auf einer Diskette zum ersten Mal das "NOTE PAD" benützt. Es wird dann automatisch eine Datei mit dem Namen "NOTES" angelegt. Diese enthält sämtliche Eintragungen, die mit dem Notizbuch auf dieser Diskette gemacht werden. Möchte man nun aber verschiedene Notizbücher anlegen, ist man gezwungen, diese auch auf verschiedenen Disketten abzulegen.

Nun könnte man aber auf die Idee kommen, den "NOTES" vom DESK TOP aus mittels RENAME einen anderen Namen zu geben, z.B. "Adressen". Das NOTE PAD müßte dann ja eigentlich ein neues Notizbuch anlegen, da es eine alte Datei "NOTES" nicht findet. Hier macht uns jedoch der Filetyp der "NOTES" einen Strich durch die Rechnung. Ein Versuch, diese Datei umzubenennen, wird nämlich vom DESK TOP mit der Begründung abgelehnt, daß dies bei "System-Files" nicht erlaubt sei. Sinnvoll ist dieses Verbot schon, da z.B. eine Umbenennung des System-Files GEOS KERNAL zur Folge hätte, daß man GEOS nicht mehr starten könnte, da GEOS BOOT diese wichtige Datei nicht mehr findet.

In unserem Fall ist dieser Schutz jedoch äußerst unpraktisch. Nun gibt es jedoch unter den zehn verschiedenen Filetypen, die unter GEOS vorhanden sind, auch sogenannte DATA FILES. Wenn man nun erreicht, daß das NOTE PAD anstelle der SYSTEM FILES diese DATA FILES für die Notizbücher benutzt, kann man diese auch später umbenennen. Genau diese Änderung können Sie mit dem Modifikator vornehmen. Dazu müssen Sie im "note pad" nach den folgenden Daten suchen:

1. Byte: 255
2. Byte: 131
3. Byte: (RETURN)

Die neuen Bytes lauten dann:

1. Byte: 255
2. Byte: 131
3. Byte: 3
4. Byte: (RETURN)

Die 3 steht hierbei für den neuen File-Typ. Wenn Sie an dieser Stelle eine 4 eingeben, erzeugt das "NOTE PAD" wieder die gewohnten System-Files. Der jeweilige Filetyp wird zur Kontrolle immer im Info-Fenster der "notes" unter "type:" eingetragen. Mit einem Beispiel möchten wir die neuen Möglichkeiten demonstrieren: Mit dem "note pad" sollen nun zwei unterschiedliche Notizblöcke auf einer Diskette erstellt werden. Auf dem einen möchten wir wichtige Telefonnummern notieren, der andere soll eine Sammlung von Kochrezepten werden.

Da noch keine Datei existiert, kann das "NOTE PAD" normal geladen werden. Es entsteht dabei die Datei "notes" auf der Diskette. Nun geben wir die ersten Telefonnummern ein. Wenn wir anschließend wieder ins DESK TOP zurückgehen, können wir mittels RENAME das File "NOTES" umbenennen in "Tel.Nr.". Anschließend laden wir das "NOTE PAD" wieder, um nun unsere Kochrezepte einzugeben. Natürlich heißt die dabei entstehende Datei wieder "notes". Wenn die Eingabe beendet ist, gehen wir zurück ins DESK TOP und benennen diesmal die "notes" um in "Rezepte".

Wir haben also nun zwei unterschiedliche Notizblöcke auf einer Diskette. Je nachdem, welchen Block man später "aufschlagen" möchte, muß vor dem Laden des "NOTE PADS" die entsprechende Datei in "notes" umbenannt werden. Dabei ist natürlich auf eine genaue Schreibweise zu achten.

Anderes Icon für die Notes (GEOS V1.2 und GEOS V1.3)

Leider sind die Programmierer des "NOTE PAD" nicht sehr konsequent gewesen. Normalerweise kann man nämlich in GEOS Datenfiles schon durch das Icon, also das kleine Bildchen im DESKTOP, von dem eigentlichen Programm unterscheiden. Zum

Beispiel versieht GEOWRITE die Textdateien durch das Symbol mit den beschriebenen Blättern. Dateien, die mit GEOPAINT erstellt wurden, zeigen hier eine Leinwand mit einem Pinsel.

Zuerst vermuteten wir, daß aus Geschwindigkeitsgründen einfach das Icon des "NOTE PAD" gleichzeitig für die "notes" mitbenutzt wird. Später fanden wir jedoch innerhalb des Programms eine Kopie der Icon-Daten wieder. Damit Sie in Zukunft die "notes" besser vom "NOTE PAD" unterscheiden können, haben wir einen Teil des Icons verändert und unten abgedruckt. Um diese Daten einzugeben, müssen Sie im "NOTE PAD" nach den folgenden Bytes suchen:

1. Byte: 249
2. Byte: 144
3. Byte: 0
4. Byte: 25
5. Byte: (RETURN)

Die neuen Daten für das Icon der "notes" lauten dann:

1. Byte: 249
2. Byte: 144
3. Byte: 0
4. Byte: 25
5. Byte: 148
6. Byte: 130
7. Byte: 25
8. Byte: 150
9. Byte: 105
10. Byte: 25
11. Byte: 144
12. Byte: 0
13. Byte: 25
14. Byte: 148
15. Byte: 0
16. Byte: 25
17. Byte: 150
18. Byte: 166
19. Byte: 153
20. Byte: 144
21. Byte: 83
22. Byte: 25
23. Byte: 144

- 24. Byte: 0
- 25. Byte: 25
- 26. Byte: 156
- 27. Byte: 34
- 28. Byte: 25
- 29. Byte: 148
- 30. Byte: 122
- 31. Byte: 153
- 32. Byte: (RETURN)

Position des Calculators ändern (GEOS V1.2 und GEOS V1.3)

Wenn Sie sich bei der Arbeit mit dem Taschenrechner manchmal darüber ärgern, daß dieser genau den Ausschnitt in Ihrem Text oder Bild verdeckt, aus denen Sie die Daten für Ihre Berechnung holen möchten, können Sie ihn mit dem Modifikator für immer in die linke obere Ecke verbannen. Suchen Sie hierzu im "calculator" nach den folgenden Bytes:

- 1. Byte: 96
- 2. Byte: 169
- 3. Byte: (RETURN)

Sie müssen ersetzt werden durch:

- 1. Byte: 96
- 2. Byte: 169
- 3. Byte: 0
- 4. Byte: 133
- 5. Byte: 4
- 6. Byte: 169
- 7. Byte: 0
- 8. Byte: (RETURN)

Wenn anstelle der 0 im 3. Byte eine 17 und anstelle der 0 im 7. Byte eine 56 eingetragen wird, erscheint der Taschenrechner wieder an der gewohnten Position.

Neue Grafikmuster (GEOS V1.2)

Man begegnet ihnen in GEOS überall: Bei dem Aufbau von Fenstern, Pull-Down-Menüs oder dem Hintergrund des DESK TOP. Gemeint sind die Muster, mit denen bestimmte Bildschirmbereiche ausgefüllt werden. So wird zum Beispiel eine normale Fehlermeldung in einem Fenster ausgegeben, dessen Hintergrund vorher mit einem 'Null-Muster' ausgefüllt wurde. Dies entspricht einfach einer weißen Fläche. Die Unterlage des DESK TOP wiederum ist mit einem sehr feinkörnigen Muster eingefärbt. Programmtechnisch wird dies erreicht, indem man in den Bytes des Musters nur jedes zweite Bit setzt. Ganz schwarze Muster werden z.B. benutzt, wenn um ein Fenster ein Schatten gelegt wird, der das Ganze dann dreidimensional wirken läßt. Die Bits eines solchen Musters sind alle gesetzt.

Zwischen Schwarz und Weiß gibt es nun eine Vielzahl von Mustern, mit denen Bereiche eingefärbt werden können. Da ein Betriebssystem wie GEOS geradezu von Grafik lebt, hat man diesen Mustern im GEOS KERNAL einen eigenen Platz gewidmet. Über ihr Aussehen können Sie sich leicht mit Hilfe von GEOPAINT informieren. Klickt man nämlich hier das Quadrat in der linken unteren Ecke an, erscheinen sofort alle 32 Muster, die das GEOS KERNAL beinhaltet. Die Daten hierüber werden also nicht, wie man vermuten könnte, von GEOPAINT zur Verfügung gestellt.

Wenn man nun über den Aufbau und die Lage der Muster im GEOS KERNAL informiert ist, kann man mit Hilfe des Modifikators sehr leicht erreichen, daß GEOPAINT genau das Muster mit in die Liste aufnimmt, das man schon immer bei seiner Arbeit vermißt hat. Zuerst wenden wir uns daher nun dem Aufbau eines Musters zu.

Benötigt werden insgesamt acht Bytes für die Darstellung eines Musters. Sie liegen im Speicher direkt hintereinander und werden in einer Grafik untereinander ausgegeben, so daß eine 8*8-Punktmatrix entsteht. Jedem gesetzten Bit wird dazu im Bild ein dunkler Punkt zugeordnet. Wenn Sie sich einmal ein Muster in GEOPAINT genau ansehen, werden Sie feststellen, daß hier aber

jeweils eine Matrix von 16*16 Punkten benutzt wird. Dies liegt einfach daran, daß das Muster hier gleich viermal ausgegeben wird und zwar zweimal nebeneinander und zweimal untereinander. Die Information steckt also tatsächlich auch nur in einer 8*8 Matrix. In vielen Fällen kommt aber die wahre Gestalt eines Musters erst durch eine großflächigere Ausgabe zum Ausdruck.

Mit dieser Information dürfte es Ihnen nun keine großen Schwierigkeiten bereiten, ein eigenes Muster selbst zu definieren. Am besten, Sie malen dazu in GEOPAINT ein 8 mal 8 Felder großes "Käsekästchenfeld", in dem Sie die Kästchen, die später dunkel erscheinen sollen, einfärben. Anschließend berechnen Sie aus jeder Reihe den entsprechenden Dezimalwert, indem Sie bei jedem ausgefüllten Feld den jeweiligen Stellenwert addieren. Die so ermittelten acht Werte können dann in den Modifikator eingegeben werden. Das folgende Bild enthält ein Beispiel für solch eine Berechnung.

Byte Nr.:	Wert:
1	0
2	56
3	68
4	68
5	68
6	124
7	68
8	68

Abb. 22: Ein neues Muster entsteht

Bevor Sie nun die neuen Werte eingeben, müssen wir noch auf einige wichtige Dinge aufmerksam machen: Die ersten Muster besitzen in GEOS einen Sonderstatus, da sie von den Program-

men selbst sehr häufig benutzt werden, um zum Beispiel einen Hintergrund auszufüllen. Ihr neues Muster erscheint daher erst an fünfter Position. Wenn Sie ein neues Muster ausprobieren möchten, müssen Sie im GEOS KERNAL nach den folgenden Bytes suchen, die noch zu dem vierten Muster gehören.

1. Byte: 36
2. Byte: 66
3. Byte: 153
4. Byte: (RETURN)

Bevor die neuen Daten für das fünfte Muster eingegeben werden können, müssen die drei Bytes wieder übernommen werden:

1. Byte: 36
2. Byte: 66
3. Byte: 153

Nun folgen die acht errechneten Dezimalzahlen. In dem Beispiel aus dem Bild lauten diese:

4. Byte: 0
5. Byte: 56
6. Byte: 68
7. Byte: 68
8. Byte: 68
9. Byte: 124
10. Byte: 68
11. Byte: 68
12. Byte: (RETURN)

Wenn Sie möchten, können Sie aber auch noch maximal fünf weitere Muster eingeben, bevor die Eingabe durch RETURN abgeschlossen wird. Diese erscheinen dann in der eingegebenen Reihenfolge in GEOPAINT. Ein neues Muster steht Ihnen dann uneingeschränkt zur Verfügung.

4.3 Über die GEOS-Druckertreiber

Wer schon einmal versucht hat, einen Treiber vom DESK TOP durch Anklicken zu laden, wird festgestellt haben, daß solch eine Datei offensichtlich kein eigenständiges Programm darstellt, da das DESK TOP das Vorhaben vorzeitig abbricht und eine

Fehlermeldung liefert. Tatsächlich handelt es sich bei den Druckertreibern um kleine Programme (max. 1 KByte) mit eigenem Filetyp (Printer Driver), die erst direkt vor einem Druckvorgang automatisch in den Speicher geladen werden.

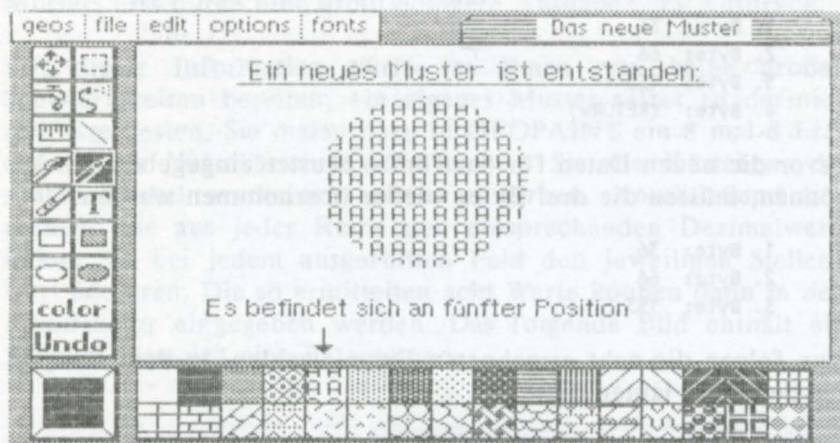


Abb. 23: Das erste Bild mit neuem Muster

Ihre Hauptaufgabe ist es, eine Grafik (und darunter fallen in GEOWRITE 1.2 bzw. 1.3 auch alle Texte) auf das für den jeweiligen Drucker richtige Format zu bringen. Die Treiber bestimmen außerdem, über welche Schnittstelle die Datenübertragung stattfinden soll, und organisieren den Datenaustausch. In den meisten Fällen wird ein Drucker wie die Floppy am seriellen Bus des C64 betrieben. Einige Treiber (z.B. für HP LaserJet oder ImageWriter) benutzen statt dessen auch eine serielle RS-232-Schnittstelle, die auf den Userport aufgesteckt werden muß.

Allen Treibern gemeinsam ist aber die Ladeadresse in den Speicher (ab \$7900) und eine kurze Sprungtabelle zu Beginn des Programms, die auf die einzelnen Routinen des jeweiligen Treibers zeigt. Dadurch ist gewährleistet, daß GEOS-Applications auch bei unterschiedlichen Längen der einzelnen Treiber-Unterprogramme immer den richtigen Einsprung finden. Im einzelnen

besitzen die derzeit auf dem Markt befindlichen Treiberprogramme maximal acht solcher Einsprungadressen. Die erste liegt bei \$7900 und endet normalerweise sofort mit RTS.

Die zweite Adresse liegt bei \$7903 und verzweigt in ein Unterprogramm, das den Drucker initialisiert. Hierbei wird der Papieranfang auf die derzeitige Position festgelegt und der Abstand zwischen den einzelnen Zeilen vorgegeben. Die dritte Adresse bei \$7906 wird am häufigsten angesprungen. Mit ihrer Hilfe kann eine gesamte Grafikzeile ausgegeben werden. Dazu wird ihr in den Speicherstellen \$02, \$03 ein Zeiger auf die zu druckenden Daten gegeben, die das normale Hires-Format (jedoch bis zu 640 Punkte) besitzen müssen.

Eine weitere, recht kurze Routine wird bei \$7909 aufgerufen. Sie beendet den Druckvorgang und erzeugt einen Papiervorschub bis zum nächsten Blattanfang. Bei \$790C liegt ein Sprung in einen 'Vierzeiler'. Dort werden nur feste Daten über den Treiber in die Register des Prozessors geholt. Sie enthalten z.B. Informationen über die mögliche Druckbreite des angeschlossenen Druckers. Leider dürfen wir keine Programme dokumentieren. Sie können aber selber einmal einen Treiber untersuchen, indem Sie ihn entweder mit dem EDMON aus dem Kapitel 5.4 laden (I "Epson FX-80") und disassemblieren (d 7900) oder mit Hilfe des untenstehenden Programms einen Treiber einlesen und sich diesen anschließend mit einem beliebigen Monitor ansehen.

Das nun folgende Programm kann aber nicht nur einen Treiber in den Rechner laden, sondern ihn auch wieder abspeichern. Zwischendurch können an ihm mit einem Monitor Änderungen vorgenommen werden. Dazu werden wir Ihnen im nächsten Abschnitt einige Anregungen geben.

```
10 OPEN 1,8,15,"1:0":GOSUB 190
15 PRINT CHR$(147);CHR$(142);CHR$(8)
20 PRINT " 1 = TREIBER LADEN"
25 PRINT" 2 = TREIBER SPEICHERN"
30 PRINT" 3 = ENDE"
35 GET W$:W=VAL(W$):IF W<1 OR W>3 THEN 35
40 PRINT CHR$(147)
45 ON W GOSUB 65,110,55
```

```

50 PRINT:GOTO 20
55 CLOSE1:END
60 :
65 OPEN 2,8,2,"*:U,R":GOSUB 190
70 K=0:PRINT:PRINT" LOADING"
75 GET#2,P$:IF P$="" THEN P$=CHR$(0)
80 POKE 30976+K,ASC(P$):K=K+1
85 IF(ST AND 64)=0 THEN 75
90 PRINT:CLOSE 2
95 PRINT:PRINT" LAENGE DES TREIBERS:";K" BYTES"
100 RETURN
105 :
110 OPEN 2,8,2,"DUMMY,S,W":GOSUB 190
115 INPUT " ANZAHL BYTES ";K
120 FOR I=0 TO K-1
125 PRINT#2,CHR$(PEEK(30976+I));:GOSUB190
130 NEXT I:CLOSE2
135 OPEN 2,8,2,"#"
140 PRINT#1,"U1";2;0;18;1
145 PRINT#1,"B-P";2;34
150 PRINT#2,CHR$(0);
155 GET#2,TD$,SD$
160 IF SD$="" THEN SD$=CHR$(0)
165 PRINT#1,"B-P";2;3
170 PRINT#2,TD$;SD$;
175 PRINT#1,"U2";2;0;18;1
180 CLOSE2:RETURN
185 :
190 INPUT#1,F,F$:REM FEHLERKANAL
195 IF F THEN PRINT:PRINTF$:STOP
200 RETURN

```

Nachdem Sie das Programm abgetippt haben, speichern Sie es bitte auf einer Diskette ab. Nun zur Bedienung des Programms. Bevor unser Programm in Aktion tritt, muß der zu ladende Druckertreiber mittels des DESK TOP an die erste Position des Inhaltsverzeichnisses gebracht werden. Außerdem darf sich im DESK TOP direkt neben dem Treiber (also an zweiter Position) kein anderes Icon befinden, da unser Programm genau hier einen eigenen Eintrag vornehmen wird, der allerdings am Ende wieder gelöscht wird. Nach dem Start unseres Programms erscheint dann ein Menü mit den folgenden drei Punkten:

1. Treiber laden
2. Treiber speichern
3. Ende

Wird der erste Menüpunkt ausgewählt, lädt das Programm den Treiber auf der Diskette in den Speicher ab \$7900. Anschließend wird die Länge des Files in Bytes angegeben. Diesen Wert müssen Sie sich unbedingt merken, wenn Sie den Treiber später wieder abspeichern möchten. Nach Auswahl des zweiten Menüpunktes werden Sie aufgefordert, die Länge des sich zur Zeit im Speicher befindlichen Treibers in Bytes einzugeben. Wenn der Treiber in der Länge nicht verändert worden ist, kann hier der unter 1. erhaltene Wert eingegeben werden. Andernfalls muß dieser Wert natürlich entsprechend korrigiert werden. Anschließend wird der Treiber auf der Diskette durch den neuen ersetzt. Wie immer sollten Sie für solche Arbeiten eine extra Diskette benutzen, damit bei eventuell auftretenden Schwierigkeiten keine wichtigen Daten zerstört werden können.

4.3.1 Änderungen an den Treibern

Leider haben viele Anwender von GEOS Schwierigkeiten, trotz des richtig gewählten Druckertreibers einen vernünftigen Ausdruck zu bekommen. In den meisten Fällen treten diese Probleme mit Druckern auf, die nachträglich mit einer Schnittstelle für den C64 ausgerüstet worden sind. So hatten wir z.B. Probleme mit unseren Epson-Druckern, die ab Werk mit einer parallelen Schnittstelle ausgestattet sind. Das angeschlossene DATA-BECKER-Interface erlaubt es nämlich, zwischen verschiedenen Betriebsarten zu wählen. Diese Auswahl geschieht dabei über die Sekundäradresse, die der angeschlossene Rechner vor einer Ausgabe an den Drucker schicken muß.

GEOS sendet nun auch solch eine Sekundäradresse. Leider ist es aber die falsche, so daß unser Interface einen Commodore-Drucker simuliert und nun auch nur entsprechende Zeichen erwartet. Da GEOS aber 'echte' Epson-Daten liefert, spielt der Drucker nun verrückt. Nur im Epson-Modus des Interface kann auch mit den entsprechenden Treibern gedruckt werden. Das Problem kann von zwei Seiten angegangen werden. Entweder

bringt man das Interface dazu, trotz der von GEOS gesendeten Sekundäradresse in den Epson-Modus zu gehen, oder man ändert im Druckertreiber selbst die Sekundäradresse auf den richtigen Wert.

Ob die erste Möglichkeit realisiert werden kann, hängt vom verwendeten Interface ab. Sind Schalter vorhanden, ist die Sache schnell erledigt: einfach den Direkt-(EPSON-) Modus einschalten. Unser Interface läßt sich softwaremäßig auf den Direktmodus fixieren. Dies kann durch die Eingabe der folgenden Zeilen (noch vor dem Booten von GEOS) erreicht werden:

```
OPEN 1,4,1
PRINT#1
OPEN 2,4,3
PRINT#2
CLOSE2
CLOSE1
```

Nun kann GEOS senden was es will, es wird immer der Epson-Modus eingeschaltet sein. In den meisten Fällen wird einem jedoch erst bei schon gebootetem GEOS einfallen, daß man vergessen hat, vorher diese Befehlsfolge einzugeben. Wir hatten jedenfalls schnell genug und haben den Druckertreiber gepatcht, uns also für die zweite Möglichkeit entschieden. Die Änderung ist recht einfach mit einem Monitor durchzuführen. Nachdem man den Treiber geladen hat (mit dem EDMON oder dem Programm aus 4.3) findet man ziemlich zu Beginn (ab \$792B ff.) den insgesamt dreimal vorhandenen Befehl, mit dem die Sekundäradresse an den Drucker gesendet wird. Er lautet:

```
"JSR $FF93"
```

Vor diesen Befehlen wird die Adresse in den Akku geladen. Man muß nun einfach anstelle der Fünf im niederwertigen Teil dieser Bytes eine Eins eintragen, damit unter GEOS ein problemloser Druck möglich ist. Bevor Sie diesen neuen Treiber wieder abspeichern, können Sie die Gelegenheit nutzen und bei Bedarf auch noch die nun folgenden Änderungen vornehmen. Zuerst eine kleine, aber feine Modifikation des Treibers. Wir fanden es sehr unschön, daß nach erfolgtem Druck immer ein

Vorschub bis zum nächsten Blattanfang ausgeführt wird. Besonders bei kurzen Probedrucken hilft hier nur ein schneller Griff zum Netzschalter des Druckers, um dieser Papierverschwendung Einhalt zu gebieten.

Es geht jedoch auch bequemer: Es genügt nämlich, den Wert \$60 (= 'RTS') in die Speicherstelle \$7909 zu schreiben, um einen kleinen Beitrag für die Reinhaltung der Umwelt zu leisten. Hierdurch wird die für das Übel verantwortliche Routine kurzerhand 'abgewürgt'. Der nun folgende Vorschlag wird Ihrem verbrauchten Farbband noch einmal ein bißchen auf die Sprünge helfen. Dazu pressen wir die letzte Farbe heraus, indem wir den Druckkopf einfach zweimal über die gleiche Position fahren lassen. Wenn man weiß, daß in den Treibern bei \$7906 in die eigentliche Druckroutine gesprungen wird, findet man schnell die Stelle, an der man unsere Idee verwirklichen könnte. Beim unserem Epson-FX-80-Treiber der GEOS-Version 1.2 ist dies bei \$79AA der Fall. Andere Treiber sind sehr ähnlich aufgebaut, so daß eine entsprechende Anpassung keine großen Schwierigkeiten bereiten dürfte.

Ab \$79AA werden in \$08/\$09 aus \$02/\$03 die Zeiger auf die Grafikzeile übergeben, die dann durch JSR \$79DE ausgedruckt wird. Für unsere Zwecke leiten wir diesen Sprung um an das freie Programmende des Treibers (ab \$7A93 ff., also: JSR \$7A93). Der EDMON teilt Ihnen das Programmende direkt nach dem Laden des Treibers mit. Hier haben wir genügend Platz für eigene Programmschritte zur Verfügung. Zuerst lassen wir auch das Unterprogramm ab \$79DE ausführen. Dadurch wird die Zeile ein erstes Mal gedruckt. Anschließend muß ein Wagenrücklauf ausgeführt werden (jedoch kein 'line feed'). Danach übergeben wir noch einmal die Parameter und drucken die Grafik ein zweites Mal durch JSR \$79DE aus. Mit einem RTS wird der kleine Ausflug schließlich beendet. Das Programm sieht etwa folgendermaßen aus:

\$7A93 JSR \$79DE	Zeile drucken
\$7A96 LDA #\$0D	'Carriage Return'
\$7A98 JSR \$FFA8	ausgeben
\$7A9B LDA \$02	Daten noch einmal

\$7A9D STA \$08	übergeben
\$7A9F LDA \$03	
\$7AA1 STA \$09	
\$7AA3 JSR \$79DE	Zeile drucken
\$7AA6 RTS	fertig

Diese Änderung läßt sich für alle Drucker ausführen, die bei einem Wagenrücklauf (\$0D) nicht auch automatisch einen Zeilenvorschub ausführen. Als letztes befassen wir uns mit der Möglichkeit, Grafiken in unterschiedlichen Breiten auszugeben. Manche Drucker bieten die Möglichkeit, bei Graphikausgaben die Anzahl der Punkte, die in waagerechter Richtung ausgegeben werden sollen, auszuwählen. Diese Auswahl geschieht durch bestimmte Steuerzeichen, die vor einem Druckvorgang vom Programm gesendet werden müssen.

Der Epson-Treiber wählt dabei eine Darstellung mit 640 Punkten, die auch der tatsächlich vorhandenen Anzahl Punkte z.B. in GEOPAINT-Bildern entspricht. Trägt man nun im Treiber ein anderes Steuerzeichen ein, kann die Breite eines Ausdrucks in weiten Grenzen variiert werden. Die Speicherstelle für das Steuerzeichen liegt im Epson-Treiber bei \$7A5A. Normalerweise steht hier \$04 für den 640-Punkte-Druck. In Ihrem Druckerhandbuch finden Sie meist unter 'ESC *' die möglichen Werte, die Sie hier eintragen können. Ein Epson RX 80 versteht z.B. die folgenden sechs Werte:

Wert	Punkte
0	480
1	960
2	960
3	1920
4	640 (voreingestellt)
6	720

Interessant ist es noch, daß bei den Werten Null und Zwei mit doppelter Geschwindigkeit (10 Zoll/sec.) gedruckt wird. Dadurch wird allerdings das Schriftbild etwas heller.

4.3.2 Drucken mit Centronics-Interface

In diesem Abschnitt zeigen wir Ihnen, wie Sie echtes Geld sparen können. Voraussetzung: Sie möchten unter GEOS drucken, haben sich aber noch keine teure Schnittstelle für ihren Epson-kompatiblen Drucker angeschafft. Auch bei Problemen mit dem seriellen Bus stellt die nun folgende Bauanleitung für eine Centronics-Schnittstelle die beste Lösung dar. Für unser Vorhaben benötigen Sie außer einem Lötkolben (ca. 25 Watt) die folgenden Dinge, die in jedem guten Elektronik-Geschäft zu bekommen sind:

- 1 Centronics-Stecker für den Drucker (2*18 Pole)
- 1 Userport-Stecker für den C64 (2*12 Pole)
- max. 1,5 Meter 11-adriges Kabel

Für weniger als 25 DM Materialkosten läßt sich so eine Schnittstelle bauen, die mit GEOS besser zusammenarbeitet als jede noch so komplizierte (und teure) käufliche. Die Hardware für das 'Interface' ist denkbar einfach. Es müssen nur die beiden Stecker nach dem folgenden Schema an das Kabel gelötet werden. Das untenstehende Bild zeigt Ihnen, wo Sie die entsprechenden Kontakte an den Steckern finden können. Kontaktzuordnung (In Klammern die Funktion)

Centronics (Drucker)	Userport (C64)
1	8 (-STROBE)
2	C (DATA 0)
3	D (DATA 1)
4	E (DATA 2)
5	F (DATA 3)
6	G (DATA 4)
7	H (DATA 5)
8	K (DATA 6)
9	L (DATA 7)
10	B (-Acknlg)
16	N (Masse)

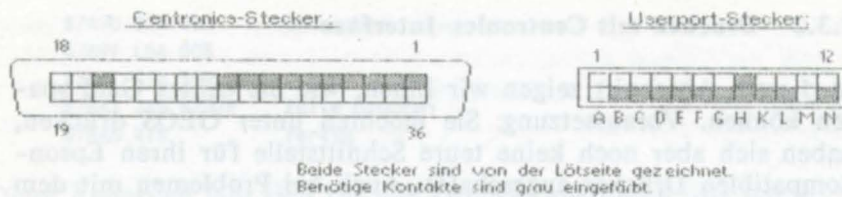


Abb. 24: Steckerzuordnung

Wenn Ihr Kabel fertig ist, bringen Sie bitte zumindest an dem Userportstecker eine Markierung an, an der Sie erkennen können, wo oben und wo unten ist. Ein verkehrtes Einstecken in den Rechner kann nämlich schlimme Folgen für beide Geräte haben! Durch die Trapezform des Centronics-Steckergehäuses für den Drucker kann Ihnen dies hiermit nicht passieren. Bei der Installation des Kabels achten Sie bitte darauf, daß die Geräte ausgeschaltet sind.

Was nun noch für den ersten Druck fehlt, ist die richtige Software für GEOS. Dazu haben wir ein Programm geschrieben, das einen neuen Treiber auf der eingelegten Diskette erzeugt. Er ist genauso wie allen anderen Druckertreiber zu behandeln, unterstützt jedoch keinen ASCII- und NLQ-Druck (wird unter GEOWRITE 2.0 angeboten). Eine genaue Bedienungsanleitung finden Sie im Anschluß an das Listing.

```

10 DIM D(341)
15 PRINT CHR$(147);CHR$(14);CHR$(8)
20 OPEN 1,8,15,"I:0":GOSUB 165
25 PRINT:INPUT N$:PRINT
30 OPEN 2,8,2,"DUMMY,S,W":GOSUB 165
35 INPUT "DRUCKMODUS (4 = STANDARD) ";DM$
40 INPUT "DOPPELDRUCK (J/N) ";DD$
45 INPUT "VORSCHUB AM ENDE (J/N) ";BV$
50 RESTORE
55 FOR I=0 TO 341
60 READ D(I):S=S+D(I)
65 NEXT I
70 IF S<>34965 THEN PRINT" FEHLER IN DATAS":STOP

```

```
75 IF DD$="J" THEN D(341)=255
80 D(262)=VAL(DM$)
85 IF BV$="N" THEN D(157)=7
90 FOR I=0 TO 341
95 PRINT#2,CHR$(D(I));:REM SEMIKOLON BEACHTEN !
100 NEXT I
105 CLOSE2
110 OPEN 2,8,2,"#"
115 PRINT#1,"U1";2;0;18;1
120 PRINT#1,"B-P";2;34
125 PRINT#2,CHR$(0);
130 GET#2,TD$,SD$
135 IF SD$="" THEN SD$=CHR$(0)
140 PRINT#1,"B-P";2;3
145 PRINT#2,TD$;SD$;
150 PRINT#1,"U2";2;0;18;1
155 CLOSE2
160 END
165 INPUT#1,N,T$,T,S:IF N=0 THEN RETURN
170 PRINT N;T$;T;S:STOP
175 REM
180 DATA 96,234,234,76,15,121,76,55,121,76,153,121,76,202,121, 32,68,
122,169
185 DATA 27,141,1,221,169,16,162,0,160,0,44,13,221,208,12, 136,
208,248,202
190 DATA 208,245,32,95,194,162,255,96,32,229,121,32,95,194, 162, 0,
96,32,68
195 DATA 122,165,3,133,9,165,2,133,8,32,97,121,44,85,122,240,16,
169,13,32
200 DATA 57,122,165,3,133,9,165,2,133,8,32,97,121,32,27,122,32,95,
194,96,165
205 DATA 9,72,165,8,72,32,165,121,176,7,104,133,8,104,133,9,96,32,
247,121
210 DATA 104,133,8,104,133,9,162,79,138,72,32,38,122,32,9,122,24,
169,8,101
215 DATA 8,133,8,169,0,101,9,133,9,104,170,202,16,230,96,32,68,
122,169,12
220 DATA 32,57,122,32,95,194,96,162,7,142,94,122,162,79,172,94,
122,177,8,208
225 DATA 21,136,16,249,24,169,8,101,8,133,8,169,0,101,9,133,9,202,
16,230,24
230 DATA 96,56,96,162,80,160,90,169,0,96,141,94,122,206,94,122,
172,94,122
235 DATA 177,8,32,57,122,206,94,122,16,243,96,169,121,133,9,169,
242,133,8
```

240 DATA 169,5,76,209,121,8,65,27,64,27,169,122,133,9,169,4,133,8,
169,5,76

245 DATA 209,121,2,128,4,42,27,162,0,138,72,189,86,122,32,57,122,
104,170,232

250 DATA 224,8,208,241,96,169,13,32,57,122,169,10,32,57,122, 96,120,
160,7,177

255 DATA 8,162,7,106,126,86,122,202,16,249,136,16,242,88,96,141,1,
221,169

260 DATA 16,44,13,221,240,251,96,32,92,194,16

265 DATA 221,173,13,221,96,0

Nun zur Bedienung des Programms

Wenn Sie unser Programm starten, muß sich auf der eingelegten Arbeitsdiskette an erster (!) Stelle ein beliebiger 'alter' Drucker-treiber befinden. Ihn benötigen wir für das Icon der Drucker-treiber und für die Informationen des Info-Fensters. Außerdem muß der Platz daneben frei sein für ein weiteres File, das unser Programm dort für kurze Zeit installiert. Die nun erscheinenden Fragen müssen Sie Ihren Wünschen entsprechend beantworten. Anschließend wird der neue Treiber auf die Diskette gebracht. Damit sind nun alle Voraussetzungen für einen ersten Probedruck erfüllt. Natürlich muß vorher von GEOS aus der neue Treiber mittels SELECT PRINTER ausgewählt werden.

5. Neue GEOS-Programme

Wenn man an GEOS denkt, fallen einem meistens sofort Begriffe aus der Benutzeroberfläche ein (Windows, Pull-Down-Menüs, Icons) und sicherlich stellt diese eine der ganz großen Besonderheiten und Neuerungen für den C64 dar. Dabei wird aber schnell eine wesentliche Neuerung vergessen, die GEOS auf dem C64 möglich macht: die Hilfsmittel (häufig auch Accessories genannt). Diese Programme können Sie praktisch jederzeit benutzen, ohne daß von Ihrer augenblicklichen Arbeit etwas verlorenggeht. In diesem Abschnitt stellen wir Ihnen vier neue Accessories für GEOS vor. Ein weiteres Programm, eine Uhr, stellt zwar kein wirkliches GEOS-Accessory dar, die deutsche Übersetzung "Hilfsmittel" trifft aber sicherlich dennoch zu. Von nun an entfällt das lästige Laden des Weckerprogramms, da die Uhrzeit ständig oben rechts auf dem Bildschirm ausgegeben wird. Außerdem finden Sie in diesem Abschnitt:

- Das Hardcopy-Programm, das es Ihnen erlaubt, den derzeitigen Bildschirminhalt jederzeit durch einen Tastendruck auf Diskette abspeichern zu können, ohne die Arbeit in GEOS hierfür unterbrechen zu müssen. Die hierbei entstehenden Dateien können von vielen Zeichenprogrammen direkt weiterverarbeitet und auch gedruckt werden.
- Den Font-Editor, mit dem Sie problemlos und mit Hilfe von Pull-Down-Menüs bestehende Zeichensätze in GEOS (FONT) verändern und neue erstellen können.
- Den Monitor, einen vollständig in GEOS eingebundenen Maschinensprache-Monitor mit den üblichen Möglichkeiten. Sie können jederzeit Speicherbereiche anschauen, Programmteile (auch das komplette GEOS KERNAL) disassemblieren und eigene Maschinensprache-Routinen erstellen und starten. Als besonderes Bonbon können Sie mit dem Monitor eigene Programme direkt erstellen und abspeichern. Damit brauchen Sie sich um Filetyp, Start- und Endadresse und INFO-Sektor nicht zu kümmern. Also einfach abspeichern, Monitor verlassen und das eigene Programm als Icon doppelklicken. Einfacher geht es nun wirklich nicht mehr.

- Der Einzelschrittsimulator, der den einfachen Einstieg in das Innenleben von GEOS ermöglicht. Jeder einzelne Programmschritt des Mikroprozessors kann hiermit verfolgt werden. Außerdem werden die Registerinhalte nach jedem Befehl auf dem Bildschirm ausgegeben. Für Programmierer und Eroberer des GEOS-Betriebssystems stellt der "Stepper" ein unentbehrliches Hilfsmittel dar.

Sollten Sie inzwischen auch auf den Geschmack gekommen sein und vielleicht eine eigene Idee für ein Programm unter GEOS haben, so finden Sie in diesem Kapitel nicht nur die Programme zum Abtippen, sondern auch ausführliche Dokumentationen, die Ihnen genau zeigen, wie man eigene Programme in GEOS einbinden kann und die vorhandenen Routinen sinnvoll nutzt. Wir meinen nämlich, daß GEOS um so beliebter wird, je mehr Programmierer die leistungsfähigen Möglichkeiten nutzen und entsprechende Programme schreiben.

Auf Ihren randvollen GEOS-Originaldisketten befindet sich nämlich nur eine Art Grundausrüstung. Diese müßte eigentlich um viele Programme erweitert werden. Während in den USA schon weitere Disketten zu GEOS herausgekommen sind, tut sich bei uns zum Thema GEOS nicht allzu viel. Mit diesem Kapitel wollen wir nicht nur zur Erweiterung von GEOS beitragen, sondern auch alle Programmierer, die noch zögern, auffordern, GEOS mit neuen Programmen zu unterstützen.

5.1 Die Uhrzeit ständig im Blick

Finden Sie es auch lästig, jedesmal das Programm "Preferences" (GEOS V1.3 deutsch: Voreinstellungen) oder die "Clock" (Wecker) laden zu müssen, nur um einen kurzen Blick auf die Uhr zu werfen? Damit dies endlich der Vergangenheit angehört, haben wir ein Programm geschrieben, das die Uhrzeit ständig oben rechts auf dem Bildschirm anzeigt und zwar im 24-Stunden-Format.

Bei der Implementierung des Programms waren einige Klümpchen notwendig, da im eigentlichen Arbeitsspeicher für derartige Anwendungen kein Bereich frei ist, der nicht über kurz oder

lang bei der Arbeit mit GEOS überschrieben wird. Daher haben wir im KERNAL nach einem Bereich gesucht, in dem Funktionen ausgeführt werden, die nicht unbedingt lebensnotwendig für das System sind. Wir sind dabei auf die Turbo-Disk-Routinen für die Diskettenstation gestoßen, die im Kernal die meiste Zeit nur wertvollen Speicherplatz verschwenden. Sie dienen bekanntlich dazu, die Diskettenzugriffe von GEOS zu beschleunigen. Normalerweise werden diese Routinen aber nur beim Neustart von GEOS einmal in den Speicher der angeschlossenen Diskettenlaufwerke kopiert. Aus diesem Grund darf man bei der Arbeit mit GEOS auch nicht einfach eine Floppy für einen Augenblick abstellen. Dabei wird natürlich das Speeder-Programm gelöscht und ein späterer Zugriff auf das Laufwerk mit einem Absturz des Rechners quittiert.

Grundsätzlich kann also das knappe Kilobyte Speicher im GEOS-Kernal für eigene Programme verwendet werden. Falls Sie jedoch ein zweites Laufwerk an Ihren 64er angeschlossen haben, sollten die folgenden Hinweise beachten. Nachdem unser Programm von GEOS aus gestartet wurde, ist es nicht mehr möglich, nachträglich ein zweites Laufwerk anzumelden. Bei einem solchen Versuch (Menüpunkt "Laufwerk anmelden") wird nämlich anstelle der Speeder-Routinen unser Uhrenprogramm in die Floppy kopiert. Was dies für Konsequenzen hat, können Sie sich sicher ausmalen.

Aus diesem Grund sollte man schon vor dem Booten von GEOS beide Floppy-Laufwerke einschalten. GEOS erkennt dies automatisch und kopiert den Speeder sofort in beide Stationen. Voraussetzung hierfür ist jedoch, daß eines der beiden Laufwerke die Gerätenummer Neun besitzt. Die Umstellung der Gerätenummer von Acht auf Neun sollte möglichst hardwaremäßig vorgenommen werden. Ein Programm, das diese Umstellung vornimmt, müßte andernfalls vor jedem Start von GEOS geladen und gestartet werden. Durch einen auch von Laien sehr leicht auszuführenden Eingriff in die Floppy kann man sich diese Arbeit jedoch ersparen. Ihr Floppy-Handbuch hilft Ihnen bei diesem Problem weiter.

Kommen wir nun endlich zu dem Programm-Listing. Zunächst drucken wir den Assembler-Source des Uhrenprogramms ab. Natürlich setzen wir nicht bei jedermann fundierte Assemblerkenntnisse voraus. Für Interessierte bietet das Listing aber eine Fülle von Informationen zur Programmierung unter GEOS. Im Anschluß an das Listing finden Sie noch ein gewöhnliches BASIC-Programm, das die vollständige Installation des Uhrenprogramms auf einer Diskette übernimmt. Sogar ein neues Icon für die Uhr wird dabei im DESK TOP erscheinen. Doch zunächst beschäftigen wir uns mit dem Source-Code:

Das Programm kann mit den meisten auf dem Markt erhältlichen Assemblern direkt übersetzt werden. Der assemblierte Code wird ab \$C000 abgelegt, obwohl er für den Bereich ab \$1000 geschrieben ist (Befehl: .OFFS \$C000 - *). Dies ist notwendig, da der von uns benutzte Assembler selber in dem Bereich ab \$1000 liegt. Falls Ihr Assembler ab \$C000 liegen sollte, können Sie das Programm natürlich direkt nach \$1000 übersetzen lassen. In diesem Fall ist der Offset-Befehl ersatzlos zu streichen und die korrekte Startadresse einzusetzen (* = \$1000). Nach der Übersetzung muß der Object-Code mit einem Monitor auf Diskette gespeichert werden. Nun fehlt nur noch die Konvertierung ins GEOS-Format. Dazu benötigen Sie den Filemaster aus dem Kapitel 3, der in der folgenden Reihenfolge bedient werden muß:

1. Zuerst muß ein geeignetes Icon (Sprite) erstellt werden (Befehl 'se'). Sie können auch ein schon vorhandenes Icon einlesen ('sl') und dies anschließend nach eigenen Wünschen modifizieren.
2. Lesen Sie nun das File ein (Befehl 'fe').
3. Geben Sie mit 'fd' das aktuelle Datum ein.
4. Mittels 'geos' das Programm konvertieren. Die Frage nach dem Filetyp mit 14 (= selbstausführend) beantworten.
5. Jetzt den Info-Bildschirm einlesen und hier die folgenden dezimalen Adressen eingeben:

STARTADRESSE : 4094
 ENDADRESSE : 0
 INITIALISIERUNG: 4096

Die Startadresse ist um zwei vermindert, da die ersten beiden Bytes im alten File-Format (in dem ein normales Monitorprogramm ja abspeichert) noch die Ladeadresse beinhalten. Diese wird aber für den Programmablauf nicht benötigt und daher vor dem eigentlichen Programm-Code abgelegt.

Die Angabe einer korrekten Endadresse ist nur bei dem Filetyp "Hilfsprogramm" interessant, da bei diesen vor dem eigentlichen Ladevorgang der von ihnen benötigte Arbeitsspeicherbereich auf Diskette gerettet wird (Swap File). Somit können Hilfsprogramme auch aus Anwendungen heraus geladen werden, da hierbei keine Daten verlorengehen. Beim Verlassen des Hilfsprogramms wird nämlich der Arbeitsspeicher durch das Laden des Swap-Files wieder restauriert. Das Anwendungsprogramm kann die Arbeit ungestört fortsetzen.

Unser Programm ist aber nur vom DESK TOP aus zu laden, und es lädt dies auch nach getaner Arbeit wieder nach. Aus diesem Grund wird ein Swap-File und damit auch die Angabe einer Endadresse nicht benötigt. Wenn Sie nun durch 'ende' den Konvertierungsvorgang abschließen, steht einem ersten Probelauf des neuen Programms nichts mehr im Wege.

Achtung: Das Programm Clock läuft nicht in Verbindung mit dem "Stepper" aus Kapitel 5.5

Hier nun das Listing des Programms:

```

;*****
;* PROGRAMM: CLOCK V1.3 *
;* AUTOR : RUEDIGER KERKLOH *
;* UPDATE : 07.04.88 *
;*****
* = $1000 ;STARTADRESSE
.OFFS $C000-* ;CODE AB $C000
;GEOS-MAKROS
DESKTOP = $C22C
GETFILE = $C208
    
```

```

CMPFSTRING = $C26E
MOVEDATA = $C17E
  MAIN = $849B
IRQVECTOR = $FFFE
  COUNT = $FF
  ;SPEEDER-ROUTINE SUCHEN
  ;(UM KOMPATIBILITAET ZU SPAETEREN
  ; GEOS-VERSIONEN HERZUSTELLEN)
INIT LDX #<DATEN      ;VERGLEICH-S-BYTES
      LDY #>DATEN      ;(SPEEDER-ROUTINE)
      STX $02
      STY $03
      JSR SEARCH        ;SPEEDER SUCHEN
      BNE CLOCK         ;GEFUNDEN !
      LDX #<NAME        ;VERGLEICH-S-BYTES
      LDY #>NAME        ;(TEXT:"CLOCK ")
      STX $02
      STY $03
      JSR SEARCH        ;CLOCK SUCHEN
      BEQ LEAVE         ;NICHT GEFUNDEN !
      LDA IRQVECTOR     ;ZEIGT ANSCHL. AUF
      SEC               ;DEN PROGRAMMSTART
      SBC #<[GO-NAME]   ;IM KERNAL
      STA $02
      LDA IRQVECTOR+1
      SBC #>[GO-NAME]
      STA $03
      ;* SPEEDER-DATEN WIEDER EINTRAGEN *
      LDY #5            ;(DAMIT DIE DATEN AUCH BEI
LOOP LDA DATEN,Y        ; EINEM ZWEITEN START GEFUNDEN
      STA ($02),Y        ; WERDEN)
      DEY
      BPL LOOP
      ;* ALTEN IRQ-VECTOR WIEDER EINTRAGEN *
      SEI
      LDY #<[OIRQ-NAME+1]
      LDA ($02),Y
      STA IRQVECTOR     ;EINTRAGEN
      INY
      LDA ($02),Y
      STA IRQVECTOR+1
      CLI
      CLC
      BCC LEAVE         ;UNBED. SPRUNG
      ;$04/$05 ENTHAELT STARTADRESSE
CLOCK LDX IRQVECTOR     ;ALTEN IRQ MERKEN

```

```

LDY IRQVECTOR+1
STX OIRQ+1
STY OIRQ+2
SEI
LDA $04 ;IRQ-VEKTOR UM-
CLC ;BIEGEN
ADC #<[GO-NAME]
STA IRQVECTOR
LDA $05
ADC #>[GO-NAME]
STA IRQVECTOR+1
LDX #<NAME ;PROGRAMM INS
LDY #>NAME ;KERNAL KOPIEREN
STX $02
STY $03
LDX #<[STOP-NAME] ;ANZAHL BYTES
LDY #>[STOP-NAME]
STX $06
STY $07
JSR MOVEDATA ;KOPIERROUTINE
CLI
LDX #<FNAME ;POINTER AUF
LDY #>FNAME ;FILENAMEN
STX $0E ;"PREF. MANAGER"
STY $0F
LDA #0 ;NORMAL LADEN
STA $02
STA $16 ;KEIN RAM RETTEN
JSR GETFILE ;LADEROUTINE
LEAVE JMP DESKTOP ;DESKTOP NACHLADEN
SEARCH ;* SUCHROUTINE *
LDX #$FF ;AB $9000 SUCHEN
LDY #$8F ;(WIRD NOCH ER-
STX $04 ;HOEHT!)
STY $05
CNT2 INC $04 ;POINTER ERHOEHEN
BNE CNT1
INC $05
BEQ FAIL ;NICHT GEFUNDEN
CNT1 LDA #6 ;ANZAHL BYTES
LDX #$02 ;ZEROPAGE POINTER
LDY #$04
JSR CMPFSTRING ;VERGLEICHEN
BNE CNT2 ;WEITERSUCHEN
LDA #$FF ;GEFUNDEN
FAIL RTS
    
```

```

;* FILENAME D. ACCESSORIES *
;*("PREF. MANAGER")
FNAME .BYTE $50,$72,$65,$66,$2E,$20,$4D
      .BYTE $61,$6E,$61,$67,$65,$72,$00
;* VERGLEICHSDATEN DES SPEEDERS *
DATEN .BYTE $48,$A5,$22,$A2,$FF,$38
;* EIGENTLICHES CLOCK-PROGRAMM *
NAME  .TEXT 'CLOCK ' ;6-BYTE-KENNUNG
GO    DEC COUNT      ;NUR ALLE 16 IRQ
      BEQ TIOUT      ;ERFOLGT AUSGABE
OIRQ  JMP $0000      ;(S.O.)
TIOUT PHA            ;REGISTER RETTEN
      TXA
      PHA
      TYA
      PHA
      LDX #$00      ;$01-$0D AUF STACK
RETTE LDA $01,X      ;RETTEN
      PHA
      INX
      CPX #$0D
      BNE RETTE
      LDA #$35      ;I/O EINBLENDEN
      STA $01
      LDA #58      ;DOPPELPUNKT
      STA $08
      STA $0B
      LDA $DC0B     ;STUNDEN
      AND #$1F
      CMP #$12
      BNE CNT
      LDA #$00      ;AUF NULL UHR SETZEN
CNT   BIT $DC0B     ;PM?
      BPL AM        ;NEIN
      SED
      CLC
      ADC #$12      ;SONST +12 STUNDEN
      CLD
AM    TAX          ;DIESER BLOCK ERGIBT
      LDA IRQVECTOR ;EINEN RELOKATIBLEN
      CLC          ;UNTERPROGRAMMAUFRUF
      ADC #<[CNT3-GO-1] ;(WIE "JSR CONV1")
      TAY
      LDA IRQVECTOR+1
      ADC #>[CNT3-GO-1]
      PHA

```

```

TYA
PHA
TXA
CLC ;NACH CONV VIA CONV1
BCC CONV1 ;(SONST: "BRANCH RANGE ERROR" !)
CNT3 STX $06
STA $07
LDA IRQVECTOR ;RELOKATIBEL (= "JSR CONV1")
CLC
ADC #<[CNT4-GO-1]
TAY
LDA IRQVECTOR+1
ADC #>[CNT4-GO-1]
PHA
TYA
PHA
LDA $DC0A ;MINUTEN
CLC
CONV1 BCC CONV
REL BCC OIRQ ;(SPRUNGHILFE)
CNT4 STX $09
STA $0A
LDA IRQVECTOR ;RELOKATIBEL (= "JSR CONV1")
CLC
ADC #<[CNT5-GO-1]
TAY
LDA IRQVECTOR+1
ADC #>[CNT5-GO-1]
PHA
TYA
PHA
LDA $DC09 ;SEKUNDEN
CLC
BCC CONV ;IN ZWEI POKES KONVERTIEREN

CNT5 STX $0C
STA $0D
LDA $DC08 ;UHR ENTRIEGELN
;* ZEICHENAUSGABE IN DIE BIT-MAP *
LDA #$31 ;CHARACTER-GENERATOR
STA $01 ;EINBLENDEN
LDX #$38 ;$A138 IST DIE OBERSTE
LDY #$A1 ;ADRESSE DES SEKUNDEN-
STX $04 ;BLOCKS IN DER BIT-MAP
STY $05
LDX #7 ;ZEICHENZAehler

```

```

NEXT2 LDA #0          ;CLR HI
      STA $03         ;(ADRESSE IM CHAR ROM)
      LDA $06,X       ;BILDSCHIRM-POKE
      ASL A           ;MAL 8
      ROL $03         ;
      ASL A           ;
      ROL $03         ;
      ASL A           ;
      ROL $03         ;
      STA $02         ;LO
      LDA #$00        ;ANFANGSADRESSE DES
      CLC             ;ZEICHENSATZES DAZU-
      ADC $03         ;ADDIEREN
      STA $03
      LDY #7         ;ZEILENZAEHLER
NEXT1 LDA ($02),Y
      STA ($04),Y
      DEY
      BPL NEXT1
      LDA $04         ;NAECHSTES ZEICHEN
      SEC
      SBC #8
      STA $04
      LDA $05
      SBC #0
      STA $05
      DEX             ;ALLE 8 ZEICHEN ?
      BPL NEXT2      ;NEIN, DANN WEITERMACHEN
      LDX #$0C        ;$0D - $01 VOM STACK
HOLE  PLA             ;HOLEN
      STA $01,X
      DEX
      BPL HOLE
      LDA #$10        ;NEUER STARTWERT
      STA COUNT
      PLA             ;REGISTER ZURUECKHOLEN
      TAY
      PLA
      TAX
      PLA
      CLC             ;ZUM GEOS IRQ
      BCC REL         ;VIA "REL"
      ;* BCD -> 2 POKEWERTE UMWANDELN *
CONV  PHA
      AND #$F0
      LSR A           ;LINKE HAELFTE

```

```

LSR A
LSR A
LSR A
CLC
ADC #48      ;OFFSET ZUM POKEWERT
TAX
PLA          ;RECHTE HAELFTE
AND #$0F
CLC
ADC #48      ;OFFSET ZUM POKEWERT
RTS
STOP
.END

```

Hier noch einmal das gleiche Programm, diesmal jedoch in Form von Datas, die durch einen kleinen BASIC-Vorspann auf die Diskette gebracht werden. Dabei bekommt die Uhr auch gleichzeitig ein neues Icon verpaßt. Den Filemaster aus dem Kapitel 3 benötigen Sie also in diesem Fall nicht. Das Programm übernimmt die vollständige Installation der Uhr. Nachdem Sie das BASIC-Programm eingegeben haben, sollten Sie es anschließend sofort auf eine Arbeitsdiskette abspeichern (SAVE "UHR BASIC",8). Falls beim Start nämlich irgend etwas schiefgehen sollte, war die doch etwas mühselige Eingabe des Listings wenigstens nicht umsonst.

Um die Uhr zu installieren, legen Sie nun eine GEOS-Diskette in das Laufwerk, die noch genügend Speicherplatz aufweist (ca. 5 KByte) und keine wichtigen Daten enthält. Starten Sie das Programm durch "RUN". Bei dem daraufhin erscheinenden Countdown wird eine Prüfsumme über alle Datas errechnet und diese mit einem von uns ermittelten, korrekten Wert verglichen. Sollten dabei Differenzen auftreten, müssen Sie die Data-Zeilen noch einmal sorgfältig überprüfen. Falls jedoch alles in Ordnung ist, erscheint nach ein paar Sekunden die Frage: "Autostart (J/N)". Beantworten Sie diese Frage zunächst einmal mit Ja (=J). Was es mit diesem "Autostart" auf sich hat, erfahren Sie im nächsten Abschnitt. Nun dauert es nicht mehr lange, bis die O.K.-Meldung ausgegeben wird. Vom DESK TOP aus können Sie nun Ihre neue Uhr bewundern. Hier nun das BASIC-Listing der Uhr.

Wichtig: Sämtliche Semikolons müssen eingegeben werden!

```

0 REM *****
2 REM * PROGRAMM: CLOCK V1.3 *
4 REM * AUTOR : RUEDIGER KERKLOH *
6 REM * UPDATE : 08/04/88 *
8 REM *****
10 :
12 OPEN 1,8,15,"I:0":REM FEHLERKANAL
14 RESTORE:PRINT CHR$(147):REM CLR HOME
16 REM * CHECKSUMME BERECHNEN *
18 FOR I=1 TO 539
20 READ A:S=S+A
22 PRINT CHR$(19);539-I;CHR$(157);" ";
24 NEXT:RESTORE
26 IF S<>57427 THEN PRINT "FEHLER IN DATAS!":STOP
28 REM * FILE ABSPEICHERN *
30 F$="CLOCK V1.3"
32 OPEN 2,8,2,F$+"U,W":GOSUB 112
34 FOR I=1 TO 426:READ A
36 PRINT#2,CHR$(A);:NEXT:GOSUB 112
38 CLOSE2
40 REM * INFO-SEKTOR ANLEGEN *
42 A=1:B=0
44 T=A:S=B:PRINT#1,"B-A:0":T;S
46 INPUT#1,N,T$,A,B:IF A=18 THEN A=19:B=0
48 IF N>0 THEN 44
50 OPEN 3,8,3,"#0":PRINT#3
52 PRINT#1,"U1";3;0;T;S:REM SEKTOR LESEN
54 FOR I=1 TO 113:READ B:PRINT#3,CHR$(B);:NEXT
56 FOR I=1 TO 141
58 PRINT#3,CHR$(0);:REM REST LOESCHEN
60 NEXT
62 PRINT#1,"U2";3;0;T;S:REM SCHREIBEN
64 REM * IN DIRECTORY EINTRAGEN *
66 :
68 PRINT#3
70 A$=CHR$(18):B$=CHR$(1):REM DIRECTORY
72 A=ASC(A$):B=ASC(B$):PRINT#1,"U1";3;0;A;B
74 GET#3,A$,B$:REM NAECHSTER SEKTOR
76 FOR I=0 TO 7:REM ANZAHL EINTRAEGE
78 PRINT#1,"B-P";3;I*32+5
80 FOR J=1 TO LEN(F$)
82 GET#3,W$:IF W$<>MID$(F$,J,1) THEN K=0:GOTO 86
84 K=K+1
86 NEXT J

```

```
88 IF K=LEN(F$) THEN 92:REM FOUND
90 NEXT I:GOSUB 112:GOTO 72
92 PRINT" AUTOSTART (J/N)";FT$
94 GET FT$
96 IF FT$="J" THEN FT=14:GOTO 102
98 IF FT$<>"N" THEN 94
100 FT=6:REM APPLICATION
102 PRINT#1,"B-P";3;I*32+21
104 PRINT#3,CHR$(T)CHR$(S)CHR$(0)CHR$(FT);
106 PRINT#3,CHR$(88)CHR$(4)CHR$(8)CHR$(14)CHR$(25)CHR$(2);
108 PRINT#1,"U2";3;0;A;B:REM DIR ZURUECK
110 CLOSE1:PRINT"O.K.":END
112 INPUT#1,N,T$:IF N THEN PRINT"ERROR: ";T$:STOP
114 RETURN
116 REM
118 DATA 162,181,160,16,134,2,132,3,32,137,16,208,56,162,187
120 DATA 160,16,134,2,132,3,32,137,16,240,108,173,254,255,56
122 DATA 233,6,133,2,173,255,255,233,0,133,3,160,5,185,181
124 DATA 16,145,2,136,16,248,120,160,11,177,2,141,254,255,200
126 DATA 177,2,141,255,255,88,24,144,65,174,254,255,172,255
128 DATA 255,142,198,16,140,199,16,120,165,4,24,105,6,141,254
130 DATA 255,165,5,105,0,141,255,255,162,187,160,16,134,2,132
132 DATA 3,162,239,160,0,134,6,132,7,32,126,193,88,162,167
134 DATA 160,16,134,14,132,15,169,0,133,2,133,22,32,8,194,76
136 DATA 44,194,162,255,160,143,134,4,132,5,230,4,208,4,230
138 DATA 5,240,13,169,6,162,2,160,4,32,110,194,208,237,169
140 DATA 255,96,80,114,101,102,46,32,77,97,110,97,103,101,114
142 DATA 0,72,165,34,162,255,56,67,76,79,67,75,32,198,255,240
144 DATA 3,76,0,0,72,138,72,152,72,162,0,181,1,72,232,224,13
146 DATA 208,248,169,53,133,1,169,58,133,8,133,11,173,11,220
148 DATA 41,31,201,18,208,2,169,0,44,11,220,16,5,248,24,105
150 DATA 18,216,170,173,254,255,24,105,72,168,173,255,255,105
152 DATA 0,72,152,72,138,24,144,23,134,6,133,7,173,254,255
154 DATA 24,105,99,168,173,255,255,105,0,72,152,72,173,10,220
156 DATA 24,144,117,144,160,134,9,133,10,173,254,255,24,105
158 DATA 124,168,173,255,255,105,0,72,152,72,173,9,220,24,144
160 DATA 90,134,12,133,13,173,8,220,169,49,133,1,162,56,160
162 DATA 161,134,4,132,5,162,7,169,0,133,3,181,6,10,38,3,10
164 DATA 38,3,10,38,3,133,2,169,208,24,101,3,133,3,160,7,177
166 DATA 2,145,4,136,16,249,165,4,56,233,8,133,4,165,5,233
168 DATA 0,133,5,202,16,207,162,12,104,149,1,202,16,250,169
170 DATA 16,133,255,104,168,104,170,104,24,144,139,72,41,240
172 DATA 74,74,74,74,24,105,48,170,104,41,15,24,105,48,96
174 :
176 REM DATEN INFO-SEKTOR
178 :
```

```

180 DATA 0,255,3,21,191,255,255,255,128,0,1,128,0,1,128,0,1
182 DATA 128,0,1,131,255,253,133,85,85,138,170,169,159,255
184 DATA 245,160,0,21,169,134,85,170,65,85,170,81,85,170,70
186 DATA 85,170,84,85,169,134,85,160,0,25,159,255,241,128,0
188 DATA 1,128,0,1,255,255,255,131,5,0,0,16,170,17,0,16,195
190 DATA 76,79,67,75,32,32,32,32,32,32,32,86,49,46,51,0,0,0
192 DATA 0,210,85,69,68,73,71,69,82,32,203,69,82,75,76,79,72

```

5.1.1 Bedienung des Programms

"Wozu eine Bedienungsanleitung für solch ein einfaches Uhrenprogramm", werden Sie sich vielleicht fragen. Auf den ersten Blick ist die Bedienung ja auch wirklich sehr einfach: Es genügt, das Icon der Uhr zweimal anzuklicken, um das Programm zu laden und zu starten. Nach der Installation wird das DESK TOP nachgeladen, und die Uhr erscheint oben rechts auf dem Bildschirm.

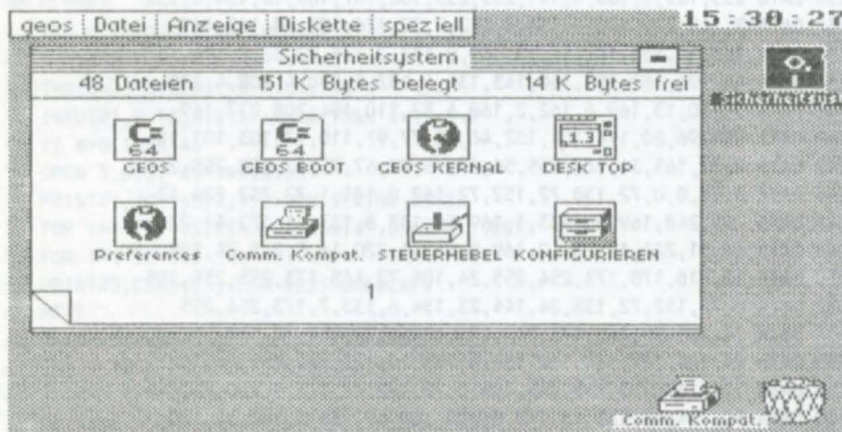


Abb. 25: Die eingblendete Uhr

Wie kann man nun aber die Uhr wieder verschwinden lassen? Dieses Problem haben wir ganz einfach gelöst: Versuchen Sie doch einmal, die Uhr ein zweites Mal zu laden. Natürlich taucht

jetzt nicht etwa oben links auf dem Bildschirm eine zweite Uhr auf. Statt dessen wird die Uhr oben rechts wieder inaktiviert. Wenn das DESK TOP nachgeladen ist, sieht alles wieder so aus wie vorher.

Wir haben aber noch ein ganzes Stück tiefer in die GEOS-Trickkiste gegriffen. Wenn Sie sich einmal im DESK TOP das Info-Fenster der Uhr ansehen, werden Sie unter der Eintragung "Typ" das Wörtchen "selbstaufführend" finden. Diesen neuen Filetyp gibt es erst seit der GEOS-Version 1.3, die mit einer erweiterten Boot-Routine ausgestattet worden ist. Wird GEOS neu geladen, so durchforstet ein Teil des Boot-Programms das gesamte Inhaltsverzeichnis der Diskette nach Programmen vom Typ "selbstaufführend". Wird eine solche Datei gefunden, so wird sie automatisch geladen und gestartet. Nach Beendigung des Programms sucht das Boot-Programm automatisch im Inhaltsverzeichnis der Diskette nach weiteren Autostart-Programmen. Erst wenn kein solches Programm mehr gefunden wird, lädt es das DESK TOP nach und startet es.

Bei der Erstellung eigener Autostart-Programme sollte man die folgenden Punkte besonders beachten:

1. Das Programm darf nur im Bereich von \$0400 bis \$4FFF liegen.
2. Es muß programmiert werden wie eine Anwendung.
3. Beendet wird solch ein Programm wie eine Anwendung durch das Nachladen des DESK TOP (JMP \$C22C). Das Boot-Programm sorgt aber schon dafür, daß evtl. erst noch weitere Autostart-Programme abgearbeitet werden.

An diese Regeln haben wir uns bei der Programmierung der Uhr gehalten. Wenn Sie also das Programm auf eine startbare GEOS-Diskette kopieren und hiermit anschließend GEOS neu laden, wird die Uhr automatisch installiert. Sicher wird es einen Erdteil geben, für den die nun angezeigte Uhrzeit einigermaßen stimmt. In den seltensten Fällen wird es aber die für uns gültige Zeit sein, da der 64er nicht über eine batteriegepufferte Echt-

zeituhr verfügt. Wir fanden es extrem lästig, nach jedem Ladevorgang das Voreinstellungsprogramm anklicken zu müssen, um hier die Korrektur des Datums und der Uhrzeit vornehmen zu können. Eine vernünftige Einstellung dieser Werte ist jedoch eine nützliche Sache, da man jederzeit die aktuellsten Dateiversionen anhand der Eintragungen im Info-Bildschirm erkennen kann.

Wer sich schon ein wenig den Source-Code der Uhr angesehen hat, wird unsere Lösung für dieses Problem kennen: Das Hilfsprogramm "Voreinstellungen" wird beim Neustart von GEOS selbstständig geladen, falls sich unsere Uhr als Autostart-Programm auf Ihrer Boot-Diskette befindet. Damit dies nicht auch beim Laden der Uhr von einer Arbeitsdiskette geschieht, ist an diese nette Funktion eine Bedingung geknüpft: Das Voreinstellungsprogramm muß auf der Boot-Diskette umbenannt werden in "Pref. Manager".

Dies ist die Abkürzung für "Preference Manager", also die amerikanische Bezeichnung für das Programm "Voreinstellungen". Dadurch ist es möglich, die Uhr auch zwischendurch einmal von einer Arbeitsdiskette laden zu können, ohne jedesmal auch die Voreinstellungen zu laden. Natürlich darf es auf solchen Disketten das Programm "Pref. Manger" nicht geben (wohl aber das Programm "Voreinstellungen!"). Eine Startsequenz (LOAD "GEOS",8,1) mit einer so behandelten Diskette sieht anschließend folgendermaßen aus: ("LOAD "GEOS",8,1")

1. Das Programm GEOS wird in den Prozessor-Stack geladen (ab \$0100), wodurch ein Autostart von GEOS geschieht.
2. Das Programm GEOS lädt nun GEOS BOOT nach und startet es. Hierbei handelt es sich um die eigentliche Boot-Routine, die im Speicher ab \$6000 abgelegt wird. GEOS BOOT lädt wiederum das gesamte Kernall nach und die schon oben erwähnte erweiterte Boot-Routine, die ab \$5000 abgelegt wird.

3. Nach dem Start der erweiterten Boot-Routine durchsucht diese das Diskettenverzeichnis nach selbstausführenden Programmen. Dabei wird unser Uhrenprogramm gefunden und nach \$1000 geladen.
4. Das Uhrenprogramm versucht nun, ein Programm mit dem Namen "Pref. Manager" zu laden und zu starten. Wird es nicht gefunden, so geht es bei 6. weiter.
5. Da das Voreinstellungsprogramm ein Hilfsprogramm ist, kann es geladen werden, ohne daß irgendwelche Daten (z.B. die des erweiterten Boot-Programms) zerstört werden. Bekanntlich wird bei Anwendungen der von ihnen benötigte Arbeitsspeicher vorher als sogenanntes Swap-File auf die Diskette gerettet. Nun können vom Anwender die korrekten Werte für Datum und Zeit eingegeben werden. Beim Verlassen des Programms werden die neuen Einstellungen übernommen, und das Swap-File wird zurückgeladen.
6. Das Uhrenprogramm sucht nun im KERNAL die Speeder-Routine und kopiert sich selbst dorthin. Die relokatable Programmiermethode ermöglicht es, die Uhr in beliebigen Speicherbereichen starten zu können. Die Speeder-Routinen in GEOS V1.2 liegen zum Beispiel in einem ganz anderen Speicherbereich. Trotzdem funktioniert die Uhr auch mit dieser Version einwandfrei.
7. Das erweiterte Bootprogramm sucht noch nach weiteren Autostart-Programmen. Sollten keine derartigen Files mehr gefunden werden, wird das DESK TOP geladen und gestartet. Der Boot-Vorgang ist damit abgeschlossen.

5.2 Hardcopys erstellen

Viele Rechner bieten die Möglichkeit, den aktuellen Bildschirminhalt auf dem Drucker auszugeben. Allerdings liegen diese Rechner meist in einer wesentlich höheren Preisklasse als der C64, der diese Möglichkeit nicht besitzt. Wie würden Sie

vorgehen, wenn Sie bei der Arbeit mit GEOS den augenblicklichen Bildschirminhalt (z.B. für einen späteren Ausdruck) auf Diskette retten möchten?

Eingefleischte Maschinensprache-Programmierer werden jetzt sagen: Das ist doch ganz einfach: RESET-Knopf drücken, Bildschirminhalt mit einem Monitor, der auch unter das ROM gelangt, von \$A000 nach \$2000 kopieren und die Daten anschließend durch 's "Bild",08,2000,3F40' auf die Diskette bringen. Wenn Sie mit dem letzten Satz nicht viel anfangen können, sind Sie in diesem Kapitel gut aufgehoben. Aber auch denjenigen, die den beschriebenen Vorgang nachvollziehen könnten, möchten wir hier eine auch von uns oft benutzte Erleichterung anbieten.

Wir haben für Sie ein Programm geschrieben, das bei der Arbeit mit GEOS die RESTORE-Taste des C64 mit einer Hardcopy-Funktion belegt. Man kann also jederzeit durch das Drücken dieser Taste den augenblicklichen Bildschirminhalt auf Diskette festhalten und anschließend in GEOS normal weiterarbeiten. Das abgespeicherte Bild besitzt das übliche DOS-Format und kann von gewöhnlichen Zeichenprogrammen (Hi-Eddi, Koala-Painter etc.) weiterverarbeitet oder auch gedruckt werden.

Zunächst einmal drucken wir den dokumentierten Source-Code des Hardcopy-Programms ab. Assembler-Experten können das Programm in dieser Form eingeben und anschließend mit dem Filemaster in ein echtes GEOS-File konvertieren. Eine genaue Bedienungsanleitung für den Filemaster finden Sie im Kapitel 3. Folgende Werte müssen dabei unter dem Menüpunkt "info" eingegeben werden:

Startadresse : 20478
Endadresse : 20940
Initialisierung: 20480

Wie auch schon bei dem Uhrenprogramm finden Sie im Anschluß an das Listing wieder einen BASIC-Lader, der das Programm samt passendem Icon (ein Fotoapparat) auf der Diskette installiert. Zunächst aber der Source-Code der Hardcopy-Routine:

5.2.1 Listing des Hardcopy-Programms

```

;*****
;* HARD-COPY V1.3 *
;* AUTOR : RUEDIGER KERKLOH *
;* UPDATE: 05.04.88 *
;*****
* = $5000 ;LADEADRESSE
.OFFS $C000-* ;(ASSEMBLERSPEZIFISCH)
;GEOS-MAKROS:
CMPFSTRING = $C26E
MOVEDATA = $C17E
IFILLRAM = $C1B4
SAVEFILE = $C1ED
OPENDISK = $C2A1
MAIN = $849B
NMIVECTOR = $FFFA
FLAG = $80
;RAM IN SPEEDER-ROUTINE SUCHEN
;(UM KOMPATIBILITAET ZU SPAETEREN
; GEOS-VERSIONEN HERZUSTELLEN)
START LDX #$FF ;AB $9000 SUCHEN
LDY #$8F ;(WIRD NOCH ER-
STX $04 ; HOEHT!)
STY $05
LDX #<STRING ;VERGLEICH-BYTES
LDY #>STRING
STX $02
STY $03
CNT2 INC $04 ;POINTER ERHOEHEN
BNE CNT1
INC $05
BEQ LEAVE ;NICHT GEFUNDEN
CNT1 LDA #7 ;ANZAHL BYTES
LDX #$02 ;ZEROPAGE POINTER
LDY #$04
JSR CMPFSTRING ;VERGLEICHROUTINE
BNE CNT2 ;WEITERSUCHEN
;$04/$05 ENTHAELT STARTADRESSE
LDX $04 ;NMI-VEKTOR
LDY $05 ;UMBIEGEN
STX NMIVECTOR
STY NMIVECTOR+1
LDX #<NMI ;PROGRAMM INS
LDY #>NMI ;KERNAL KOPIEREN
STX $02
    
```

```

    STY $03
    LDX #<[STOP-NMI] ;ANZAHL BYTES
    LDY #>[STOP-NMI]
    STX $06
    STY $07
    JSR MOVEDATA ;KOPIERROUTINE
    LDA #0 ;FLAG (VERHINDERT
    STA FLAG ; NMI-SCHACHTELUNG)
LEAVE LDX #$3E ;SWAP ZURUECK
    LDY #$C2 ;(JOB $C23E)
    STX MAIN
    STY MAIN+1
    RTS ;ENDE
;VERGLEICHSDATEN FUER SUCHVORGANG
STRING
.BYTE $84,$73,$84,$74,$C8,$84,$71
;EINSPRUNG BEI RESTORE
NMI PHA ;WERTE RETTEN
    LDA FLAG
    BNE EXIT ;WEITERER NMI !!!
    LDA #$FF ;SETZEN
    STA FLAG
    TYA
    PHA
    LDA $02
    PHA
    LDA $03
    PHA
INIT LDA NMIVECTOR
    CLC
    ADC #<[COPY-NMI]
    STA $02 ;NEUER JOB (LOW)
    LDA NMIVECTOR+1
    ADC #>[COPY-NMI]
    STA $03 ;NEUER JOB (HIGH)
    LDY #<[OLDJOB-COPY]
    LDA MAIN
    STA ($02),Y ;ALTE ROUTINE
    INY ;RETEN
    LDA MAIN+1
    STA ($02),Y
    INY ;2 BYTE DES GEOS-
    LDA $9FFE ;KERNAL RETTEN.
    STA ($02),Y ;(WERDEN BEIM AB-
    INY ; SPEICHERN DES
    LDA $9FFF ; BILDES BENOETIGT)

```

```

STA ($02),Y
LDA #00 ;START DER GRAPHIK
STA $9FFE ;EINTRAGEN: $2000
LDA #20 ;(ALTES DOS-FORMAT)
STA $9FFF
LDA $02 ;JOB EINTRAGEN
STA MAIN
LDA $03
STA MAIN+1
PLA ;ALTE WERTE HER-
STA $03 ;STELLEN
PLA
STA $02
PLA
TAY
EXIT PLA
RTI ;RESTORE ENDE
;ZEIT IN FILENAMEN EINTRAGEN
COPY CLC ;DATEN UEBER-
BCC SKIP ;SPRINGEN
;* DATENBEREICH *
OLDJOB .BYTE $00,$00 ;ALTE ROUTINE
.BYTE $00,$00 ;$9FFE/F DATEN
NAME .TEXT 'COPY ' ;FILENAME
TIME .TEXT 'XX-XX-XX'
.BYTE $0 ;ABSCHLUSS
INFO .BYTE $03,$15,$8F ;ICON-HEADER
;ICON-DEFINITION (63 BYTES)
.BYTE %11111111,%11111111,%11111111
.BYTE %10000000,%00000000,%00000001
.BYTE %10000000,%00000000,%00000001
.BYTE %10000000,%00000000,%00000001
.BYTE %10010010,%01100111,%00110001
.BYTE %10010010,%10010100,%10101001
.BYTE %10011110,%11110111,%00101001
.BYTE %10010010,%10010100,%10101001
.BYTE %10010010,%10010100,%10110001
.BYTE %10000000,%00000000,%00000001
.BYTE %10000000,%00000000,%00000001
.BYTE %10000000,%00000000,%00000001
.BYTE %10000000,%00000000,%00000001
.BYTE %10001100,%11001110,%01001001
.BYTE %10010001,%00101001,%01001001
.BYTE %10010001,%00101110,%00110001
.BYTE %10010001,%00101000,%00100001
.BYTE %10001100,%11001000,%01000001
.BYTE %10000000,%00000000,%00000001

```

```

.BYTE %10000000,%00000000,%00000001
.BYTE %10000000,%00000000,%00000001
.BYTE %11111111,%11111111,%11111111
    .BYTE $82      ;C64 FILETYP : PRG
    .BYTE $03      ;GEOS FILETYP: DATA
    .BYTE $00      ;STRUKTUR : SEQ
    .BYTE $FE,$9F ;START
    .BYTE $40,$BF ;ENDE
    .BYTE $00,$00 ;INITIALISIERUNG
;NAME FUER INFOSEKTOR
INAME .BYTE 'H
    .BYTE 'A OR $20 ;(KLEINSCHRIFT !)
    .BYTE 'R OR $20
    .BYTE 'D OR $20
    .BYTE 'C OR $20
    .BYTE 'O OR $20
    .BYTE 'P OR $20
    .BYTE 'Y OR $20
INFOEND
;EIGENTLICHE HARDCOPY-ROUTINE
SKIP JSR OPENDISK ;NEUE DISK
LDA MAIN ;START DER
CLC ;TIME-VARIABLEN
ADC #<[TIME-COPY] ;BERECHNEN UND
STA $02 ;NACH $02/$03
LDA MAIN+1
ADC #>[TIME-COPY]
STA $03
SED
LDY #0
LOOP2 LDA $8519,Y ;H,M,S
;* 1 HEX IN 2 DEZ. ASCII-WERTE *
TAX
LDA #0 ;LOESCHEN
CLC
LOOP1 DEX
BMI OK
ADC #1
BNE LOOP1
OK TAX ;ERGEBNIS RETTEN
TYA ;ZAEHLER RETTEN, IN
PHA ;ZEIGER UMWANDELN
LDY #7
STA ($02),Y ;PUFFER (SEKUNDEN)
ASL A ;* 2
ADC ($02),Y ;+ PUFFER = * 3

```

```

TAY ;ZEHNER IN ASCII
TXA ;NUR LINKE HAELFTE
LSR A
LSR A
LSR A
LSR A
ORA #$30 ;ZEHNER IN ASCII
STA ($02),Y
TXA
AND #$0F ;NUR RECHTE HAELFTE
ORA #$30 ;EINER IN ASCII
INY
STA ($02),Y
PLA ;ZAEHLER HOLEN
TAY
INY
CPY #3
BNE LOOP2
CLD ;NICHT MEHR NOTW.
;FILE-HEADER VORBEREITEN
LDA MAIN ;QUELLADRESSE
CLC ;BERECHNEN
ADC #<[INFO-COPY]
STA $02
LDA MAIN+1
ADC #>[INFO-COPY]
STA $03
LDX #$02 ;ZIELADRESSE
LDY #$81 ;(PUFFER AB $8102)
STX $04
STY $05
LDX #<[INFOEND-INFO] ;ANZAHL
LDY #>[INFOEND-INFO]
STX $06
STY $07
JSR MOVEDATA ;KOPIERROUTINE
LDA MAIN ;ADRESSE D. NAMENS
CLC ;EINTRAGEN
ADC #<[NAME-COPY]
STA $8100
LDA MAIN+1
ADC #>[NAME-COPY]
STA $8101
JSR IFILLRAM ;REST LOESCHEN
.WORD [256-77-[INFOEND-INAME]]
.WORD [$8100+77+INFOEND-INAME]

```

```

      .BYTE 0           ;LOESCHEN
      LDA #0           ;PAGE
      STA $16
      LDX #$81
      STA $14           ;$8100: FILE-HEADER
      STX $15
      JSR SAVEFILE     ;SPEICHERN
                       ;FEHLERAUSWERTUNG
                       ;ENTFAELT
      LDX MAIN         ;START DES H-COPY
      LDY MAIN+1       ;PROGRAMMS
      STX $02
      STY $03
      LDY #<[OLDJOB-COPY]
      LDA ($02),Y      ;ALTE ROUTINE
      STA MAIN         ;EINTRAGEN
      INY
      LDA ($02),Y
      STA MAIN+1
      INY              ;PATCH EINTRAGEN
      LDA ($02),Y
      STA $9FFE
      INY
      LDA ($02),Y
      STA $9FFF
      LDA #0           ;ZURUECKSETZEN
      STA FLAG
      RTS              ;FERTIG
STOP
      .END

```

Hier nun das BASIC-Listing des Hardcopy-Programms.

```

0 REM *****
2 REM * PROGRAM : HARDCOPY V1.3 *
4 REM * AUTHOR : RUEDIGER KERKLOH *
6 REM * UPDATE : 05.04.88 *
8 REM *****
10 :
12 OPEN 1,8,15,"I:0":REM FEHLERKANAL
14 RESTORE:PRINT CHR$(147):REM CLR HOME
16 REM * CHECKSUMME BERECHNEN *
18 FOR I=1 TO 573
20 READ A:S=S+A
22 PRINT CHR$(19);573-I;CHR$(157);" ";
24 NEXT:RESTORE

```

```
26 IF S<>59544 THEN PRINT "FEHLER IN DATAS !":STOP
28 REM * FILE ABSPEICHERN *
30 F$="HARDCOPY V1.3"
32 OPEN 2,8,2,F$+"U,W":GOSUB 104
34 FOR I=1 TO 460:READ A
36 PRINT#2,CHR$(A);:NEXT:GOSUB 104
38 CLOSE2
40 REM * INFO-SEKTOR ANLEGEN *
42 A=1:B=0
44 T=A:S=B:PRINT#1,"B-A:0";T;S
46 INPUT#1,N,T$,A,B:IF A=18 THEN A=19:B=0
48 IF N>0 THEN 44
50 OPEN 3,8,3,"#0":PRINT#3
52 PRINT#1,"U1";3;0;T;S:REM SEKTOR LESEN
54 FOR I=1 TO 113:READ B:PRINT#3,CHR$(B);:NEXT
56 FOR I=1 TO 141
58 PRINT#3,CHR$(0);:REM REST LOESCHEN
60 NEXT
62 PRINT#1,"U2";3;0;T;S:REM SCHREIBEN
64 REM * IN DIRECTORY EINTRAGEN *
66 :
68 PRINT#3
70 A$=CHR$(18):B$=CHR$(1):REM DIRECTORY
72 A=ASC(A$):B=ASC(B$):PRINT#1,"U1";3;0;A;B
74 GET#3,A$,B$:REM NAECHSTER SEKTOR
76 FOR I=0 TO 7:REM ANZAHL EINTRAEGE
78 PRINT#1,"B-P";3;I*32+5
80 FOR J=1 TO LEN(F$)
82 GET#3,W$:IF W$<>MID$(F$,J,1) THEN K=0:GOTO 86
84 K=K+1
86 NEXT J
88 IF K=LEN(F$) THEN 92:REM FOUND
90 NEXT I:GOSUB 104:GOTO 72
92 PRINT#1,"B-P";3;I*32+21
94 PRINT#3,CHR$(T)CHR$(S)CHR$(0)CHR$(5);
96 PRINT#3,CHR$(88)CHR$(4)CHR$(5)CHR$(20)CHR$(11)CHR$(2);
98 PRINT#1,"U2";3;0;A;B:REM DIR ZURUECK
100 CLOSE1:PRINT"O.K.":END
102 :
104 INPUT#1,N,T$:IF N THEN PRINT"ERROR: ";T$:STOP
106 RETURN
108 REM
110 DATA 162,255,160,143,134,4,132,5,162,79,160,80,134,2,132
112 DATA 3,230,4,208,4,230,5,240,44,169,7,162,2,160,4,32,110
114 DATA 194,208,237,166,4,164,5,142,250,255,140,251,255,162
116 DATA 86,160,80,134,2,132,3,162,118,160,1,134,6,132,7,32
```

```

118 DATA 126,193,169,0,133,128,162,62,160,194,142,155,132,140
120 DATA 156,132,96,132,115,132,116,200,132,113,72,165,128
122 DATA 208,80,169,255,133,128,152,72,165,2,72,165,3,72,173
124 DATA 250,255,24,105,87,133,2,173,251,255,105,0,133,3,160
126 DATA 3,173,155,132,145,2,200,173,156,132,145,2,200,173
128 DATA 254,159,145,2,200,173,255,159,145,2,169,0,141,254
130 DATA 159,169,32,141,255,159,165,2,141,155,132,165,3,141
132 DATA 156,132,104,133,3,104,133,2,104,168,104,64,24,144
134 DATA 101,0,0,0,0,67,79,80,89,32,88,88,45,88,88,45,88,88
136 DATA 0,3,21,191,255,255,255,128,0,1,128,0,1,128,0,1,146
138 DATA 103,49,146,148,169,158,247,41,146,148,169,146,148
140 DATA 177,128,0,1,128,0,1,128,0,1,140,206,73,145,41,73,145
142 DATA 46,49,145,40,33,140,200,65,128,0,1,128,0,1,128,0,1
144 DATA 255,255,255,130,3,0,254,159,64,191,0,0,72,97,114,100
146 DATA 99,111,112,121,32,161,194,173,155,132,24,105,12,133
148 DATA 2,173,156,132,105,0,133,3,248,160,0,185,25,133,170
150 DATA 169,0,24,202,48,4,105,1,208,249,170,152,72,160,7,145
152 DATA 2,10,113,2,168,138,74,74,74,74,9,48,145,2,138,41,15
154 DATA 9,48,200,145,2,104,168,200,192,3,208,207,216,173,155
156 DATA 132,24,105,21,133,2,173,156,132,105,0,133,3,162,2
158 DATA 160,129,134,4,132,5,162,83,160,0,134,6,132,7,32,126
160 DATA 193,173,155,132,24,105,7,141,0,129,173,156,132,105
162 DATA 0,141,1,129,32,180,193,171,0,85,129,0,169,0,133,22
164 DATA 162,129,133,20,134,21,32,237,193,174,155,132,172,156
166 DATA 132,134,2,132,3,160,3,177,2,141,155,132,200,177,2
168 DATA 141,156,132,200,177,2,141,254,159,200,177,2,141,255
170 DATA 159,169,0,133,128,96
172 :
174 REM DATEN INFO-SEKTOR
176 :
178 DATA 0,255,3,21,191,0,28,0,56,61,0,68,66,128,142,153,120
180 DATA 143,0,244,144,28,12,144,62,12,144,67,12,144,153,12
182 DATA 144,137,12,144,129,12,144,66,12,144,60,12,144,0,12
184 DATA 143,255,240,192,60,0,112,126,0,120,207,0,121,137,128
186 DATA 51,8,192,6,8,96,130,5,0,0,80,204,81,0,80,72,97,114
188 DATA 100,99,111,112,121,32,32,32,32,86,49,46,51,0,0,0,0
190 DATA 210,85,69,68,73,71,69,82,32,203,69,82,75,76,79,72

```

5.2.2 Bedienung des Programms

Wenn Sie das BASIC-Programm eingetippt haben, speichern Sie es bitte zuerst auf einer Diskette ab. Legen Sie anschließend eine GEOS-Arbeitsdiskette mit mindestens 20 KByte freiem Speicherplatz ein, und starten Sie unser Programm. Oben links auf

dem Bildschirm erscheint dann ein Countdown. In diesem Augenblick wird eine Prüfsumme über alle DATAS berechnet. Sollte sich bei der Eingabe der DATAS ein Fehler eingeschlichen haben, hat dies eine entsprechende Meldung zur Folge.

Wenn die DATAS in Ordnung sind, wird das Programm unter dem Namen "H-Copy V1.3" auf Ihrer GEOS-Diskette abgespeichert. Dabei erhält es auch gleichzeitig ein schönes Icon (Fotoapparat), an dem Sie es sofort erkennen können. Die eigentliche Bedienung ist nun ganz einfach: Sollen bei der Arbeit mit GEOS Hardcopies erstellt werden, muß vorher unser Programm geladen werden. Dies geschieht wie üblich in GEOS durch das Anklicken des Fotoapparates. Nach kurzer Zeit hat sich die Floppy wieder beruhigt, und das Programm ist startklar. Sie können nun aus jeder Situation heraus die RESTORE-Taste betätigen und dadurch eine Kopie des derzeitigen Bildes auf der Diskette erzielen. Pro Bild werden dazu 8 KByte Platz auf der Diskette benötigt. Sollte die Diskette zu voll sein, wird RESTORE einfach ignoriert.

Um anschließend den Eintrag des Bildes im Directory sehen zu können, muß die Diskette neu geöffnet werden, da das DESKTOP von dem Speichervorgang nicht informiert worden ist und somit auch kein neues Icon anzeigt. Die Bilder werden mit einem eigenen Icon versehen, das die Aufschrift "Hardcopy" trägt. Als Filenamen erhalten Sie den Text "Copy", der raffinierterweise noch um die aktuelle Uhrzeit ergänzt wird. Dadurch können Sie (genügend Platz auf der Diskette vorausgesetzt) mehrere Hardcopies erzeugen, die sich dann auch im Namen sinnvoll voneinander unterscheiden: An der Uhrzeit läßt sich später unschwer erkennen, welches Bild Sie zuerst und welches Sie später erstellt haben. Besonders angenehm ist es unserer Meinung nach, daß das Abspeichern der Bilder mit den GEOS-Turbo-Routinen erfolgt. Dadurch verdoppelt sich in etwa die Speichergeschwindigkeit (nur ca. 12 Sekunden).

5.2.3 Dokumentation des Programms

Besonders für diejenigen Leser unter Ihnen, die einmal selber Programme unter GEOS schreiben möchten und schon ein wenig Erfahrung mit Assemblern besitzen, haben wir den Source-Code des Hardcopy-Programms dokumentiert. Außer Kommentaren zu den einzelnen Programmschritten möchten wir Ihnen in diesem Abschnitt noch einige Hinweise zur grundsätzlichen Funktionsweise des Programms geben.

Nach dem Start des Programms wird zunächst einmal im Kernall ab \$9000 nach der Abfolge ganz bestimmter Bytes gesucht, die nur im Bereich der Speeder-Routinen vorhanden sind. Wie schon das Uhrenprogramm aus Kapitel 5.1, so wird auch das Hardcopy-Programm anschließend in diesen Bereich kopiert. Aus diesem Grund ist dieser Teil des Programms vollständig relokationstauglich geschrieben, was leider ein wenig das Verständnis des Listings erschwert. Sollte jedoch bei späteren GEOS-Versionen der Speeder in einem anderen Bereich liegen, funktioniert unser Programm trotzdem noch ohne Änderungen.

Die Speeder-Routinen bieten sogar genügend Platz, um gleichzeitig auch noch die Uhr aufnehmen zu können. Sollten die Vergleichsdaten aus irgendeinem Grund nicht gefunden werden (z.B. weil das Hardcopy-Programm schon installiert ist), wird das Programm sofort verlassen. Dies geschieht bei Hilfsprogrammen immer in der bei LEAVE angegebenen Art und Weise. Andernfalls wird der NMI-Vektor - die hierin enthaltene Adresse wird beim Betätigen von RESTORE angesprungen - umgebogen und anschließend mittels MOVEDATA das eigentliche Hardcopy-Programm in den gefundenen Bereich des Speeders kopiert. Damit ist die Initialisierung auch schon beendet.

Erst wenn die RESTORE-Taste betätigt wird, erfolgt die Abarbeitung des Programmteils ab 'NMI'. Hier wird zunächst einmal getestet, ob schon ein NMI vorliegt. Wir mußten nämlich feststellen, daß ein einziger Druck auf die RESTORE-Taste häufig eine ganze Salve von NMIs auslöst. Die Speicherstelle FLAG schützt uns in solchen Fällen vor der Anfertigung mehrerer Hardcopies, indem die NMI-Routine sofort wieder verlassen

wird. Nur ein einziger NMI wird beachtet. Im Anschluß daran werden die Speicherstellen \$9FFE/\$9FFF und \$849B/\$849C gerettet, da sie von uns mit neuen Werten belegt werden. In \$9FFE/\$9FFF wird die Anfangsadresse der Grafik eingetragen (\$2000), die für ein korrektes DOS-Format notwendig ist. Der Vektor \$849B/\$849C (s. Speicheraufteilung in Kapitel 6) erhält den Einsprung in die eigentliche Hardcopy-Routine. Beim nächsten Durchlauf der GEOS-Hauptschleife wird nun 'COPY' angesprochen.

Dabei wird zuerst die aktuelle Uhrzeit in einem für den File-Namen reservierten Speicherbereich abgelegt. Dazu muß jede Angabe (Stunde, Minute, Sekunde) in zwei entsprechende ASCII-Werte umgewandelt werden. Dies geschieht unter Zuhilfenahme des Dezimal-Modus (SED) des Prozessors. Weiterhin wird der bei INFO abgelegte Info-Sektor in den Puffer ab \$8100 kopiert. Die im Anschluß daran aufgerufene SAVE-Funktion entnimmt diesem Bereich sämtliche für den Speichervorgang notwendigen Informationen (Icon, Filetyp, Speicheradressen, etc.). Schließlich werden noch die Speicherstellen \$9FFE/\$9FFF und der Vektor \$849B/\$849C auf Ihren ursprünglichen Wert gebracht, und das Programm wird durch RTS verlassen.

5.3 Der Font-Editor

Einer der größten Vorzüge, den die Textverarbeitung unter GEOS bietet, besteht sicherlich darin, die aktuelle Schriftart eines Textes jederzeit variieren zu können. Dazu genügt es z.B. von GEOWRITE aus einen beliebigen Zeichensatz auszuwählen, mit dem dann ab der derzeitigen Cursor-Position weitergeschrieben werden soll. Zur Auswahl stehen dabei immer nur die Fonts (Zeichensätze), die sich auf der eingelegten Diskette befinden.

Für Berkley Softworks bietet das beschriebene Verfahren eine zusätzliche Einnahmequelle, da jederzeit weitere Fonts entworfen werden können, die dann zusätzlich zur GEOS-Grundausstattung käuflich zu erwerben sind. So dauerte es auch nicht lange, bis die erste Diskette mit 20 neuen Zeichensätzen auf dem Markt erschien ("Font Pack 1").

Wir stellten uns nun die Frage, warum die Programmierer von GEOS nicht ein kleines Programm entwickelt haben, mit dem der Anwender selbst an seinen Zeichensätzen Manipulationen vornehmen kann. Wer käme schon auf die Idee, mit diesem Hilfsmittel mühselig 20 gewöhnliche Zeichensätze neu zu definieren, wenn er sie für wenig Geld fertig beziehen kann? Viele Benutzer von GEOWRITE werden aber für ihre Texte bestimmte Sonderzeichen benötigen, die sie wohl niemals auf einer solchen "Font Pack"-Diskette finden werden. Außerdem gibt es da noch die vielen Besitzer einer GEOS-Version 1.2, die nicht einmal die deutschen Umlaute in den Fonts der im Lieferumfang enthaltenen Disketten finden. Aus diesem Grunde haben wir uns mit diesem Thema näher auseinandergesetzt.

Nachdem wir längere Zeit benötigt haben, um den recht komplizierten Aufbau eines Zeichensatzes zu verstehen, bestand für uns nun prinzipiell die Möglichkeit, einzelne Zeichen eines Satzes unter Zuhilfenahme eines Maschinensprache-Monitors (z.B. des EDMON aus dem Kapitel 5) zu modifizieren. So entstand unter GEOS V1.2 der erste deutsche Zeichensatz, der auch die Umlaute der deutschen Sprache enthielt.

Da dieses Verfahren natürlich viel zu umständlich ist, haben wir für Sie ein Programm entwickelt, mit dessen Hilfe Sie sämtliche Zeichensätze beliebig verändern (editieren) können. Es ist also nicht nur möglich, ein "o" durch zwei Pünktchen in ein "ö" zu verwandeln, sondern Sie können sogar spezielle Zeichensätze in Proportionalschrift (die ja unter GEOS immer benutzt wird) selber definieren.

Wie wäre es zum Beispiel mit einem griechischem Zeichensatz für alle Naturwissenschaftler unter Ihnen? Auch die tiefgestellten Atomzahlen der Chemie entziehen sich nun nicht mehr der Textverarbeitung durch GEOS. Natürlich können Sie auch beliebige Grafik-Symbole definieren, wie z.B. die auf der Vorderseite der Tasten am C64, die ja unter GEOS nicht mehr erreicht werden können.

Das Programm ist dabei so flexibel geschrieben, daß sowohl bestehende Zeichensätze modifiziert als auch vollständig neue Zeichensätze von 1-62 Point Höhe erstellt werden können. Bevor Sie nun loslegen und über den etwas mühseligen Weg der DATA-Zeilen unser Programm in den Rechner geben, möchten wir Ihnen schon ein wenig den Mund wäßig machen, damit Sie nicht nach halb getaner Arbeit doch noch die Flinte ins Korn werfen. Wir nehmen daher die Bedienung des Programms vorweg.

5.3.1 Bedienung des Font-Editors

Starten des Programms

Zuerst wieder einmal ein wichtiger Hinweis: Bei der Arbeit mit unserem Editor dürfen sich die Original-GEOS-Disketten nicht in Ihrem Aktionsradius befinden. Sollte nämlich aus irgendeinem Grund etwas schiefgehen, können Sie jederzeit wieder auf die Original-Zeichensätze zurückgreifen. Fertigen Sie also auf Ihrer Arbeitsdiskette eine Kopie der Zeichensätze an, die Sie modifizieren möchten. Wie dies geschieht, ist im Handbuch zu GEOS beschrieben.

Anschließend empfiehlt sich eine kleine Änderung des Font-Namens, um spätere Verwechslungen zu vermeiden. Da nach jeder Änderung eines Files automatisch das Datum und die Uhrzeit im zugehörigen Info-Feld abgelegt wird, sollten Sie diese Angaben mit Hilfe des Programms "Voreinstellungen" vor dem ersten Laden des Editors aktualisieren.

Da es sich bei dem Editor um ein GEOS-Hilfsprogramm (Accessory) handelt, kann er jederzeit, also auch von den Anwendungen GEOPAINT und GEOWRITE aus, geladen werden. Dazu genügt ein Anklicken seines Namens, der sich als Unterpunkt in dem GEOS-Menü (ganz oben links) befindet. Gleiches bewirkt auch ein Doppelklick des zugehörigen Icons vom DESK TOP aus.

Um mit dem Font-Editor arbeiten zu können, benötigen Sie relativ viel freien Platz auf Ihrer Diskette. Das Swap-File, das beim Laden erstellt wird, belegt 64 Blöcke, also etwa 16 KByte. Zusätzlich brauchen Sie natürlich noch Platz, wenn Sie einen neuen Zeichensatz oder eine neue Point-Größe erstellen wollen. Sie sollten beim Start des Font-Editors also etwa mindestens etwa 25 KByte freien Diskettenspeicher haben.

Nach dem Anklicken des Icons dauert es eine Weile, und wenn alles geklappt hat, erscheint in der Mitte Ihres Bildschirms ein schwarz umrahmtes Rechteck mit zwei Pull-Down-Menüs wie Sie sie auch vom DESK TOP her kennen.

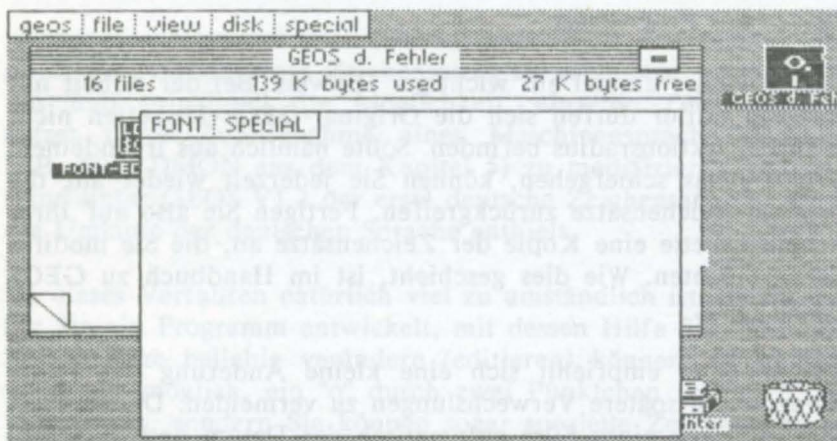


Abb. 26: Der Font-Editor

Beim Anklicken der beiden Menüpunkte FONT und SPECIAL muß jeweils ein Menü herunterklappen: FONT besitzt die Unterpunkte "LOAD", "SAVE", "UPDATE" und "CLR MEM". Die Unterpunkte von "SPECIAL" lauten "QUIT" und "NEW DISK". Der Einfachheit halber geben wir in den folgenden Erläuterungen nur den Unterpunkt an und nicht auch das Hauptmenü, unter dem Sie diesen Punkt finden. Sie brauchen übrigens keine Angst zu haben, daß unser Programm, falls es von GEOWRITE

oder GEOPAINT aus gestartet wird, irgendwelche Daten überschreibt: Nach dem Verlassen des Programms sieht alles wieder so aus wie vorher.

Ändern eines Zeichensatzes

In diesem Abschnitt erklären wir die Vorgehensweise, um ein schon vorhandenes Font-File den eigenen Wünschen anzupassen. Legen Sie dazu Ihre Arbeitdiskette, die den zu ändernden Zeichensatz enthält, in die Diskettenstation ein, und teilen Sie dies dem Programm mit, indem Sie mit der Maus NEW DISK anklicken. Dadurch werden die Namen der ersten fünf Fonts auf der Diskette in eine Liste übernommen. Diese Liste erscheint nach Anklicken des Feldes "Load".

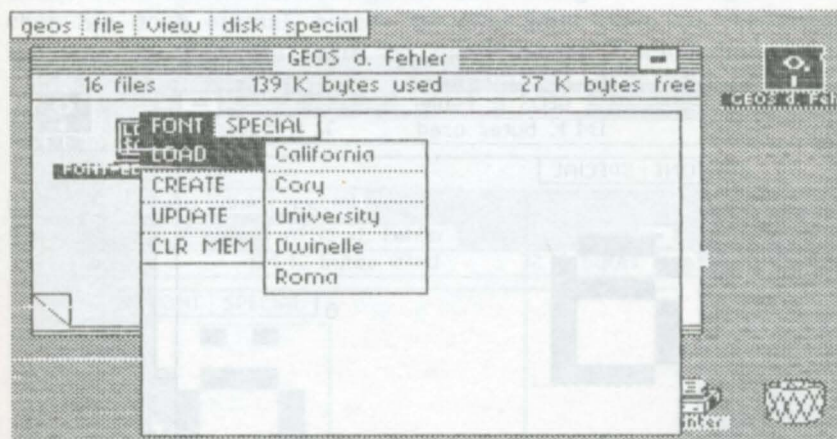


Abb. 27: Das Load-Menü

Die Begrenzung der Liste auf fünf Einträge war notwendig, da uns nur ein begrenzter Speicherplatz zur Verfügung stand. Sollte sich Ihr Font nicht darunter befinden, können Sie im DESK TOP die Reihenfolge der FONTS umändern. Klicken Sie dazu QUIT an. Nachdem in der Floppy wieder Ruhe eingekehrt und das gewohnte DESK-TOP-Bild erschienen ist, legen Sie das

nicht angezeigte Font und ein anderes, das Sie nicht unbedingt brauchen, auf der BORDER ab. Anschließend setzen Sie beide Icons in umgekehrter Reihenfolge wieder im DESK-TOP-Fenster ab. Die beiden Fonts sind nun in der Reihenfolge vertauscht. Anschließend muß der Font-Editor natürlich wieder geladen werden. Wenn Ihr Zeichensatz nun in der Load-Liste steht, muß nach dessen Auswahl noch die Point-Größe ausgewählt werden. Sie gibt die Höhe des Zeichens in Punkten an. Dann erst werden die entsprechenden Daten geladen. Dieser Vorgang dauert je nach Umfang des Files einige Sekunden.

Jetzt wird es spannend: Drücken Sie einmal einen beliebigen Buchstaben auf der Tastatur. Er wird prompt rechts im Programmfenster ausgegeben, und nach einer kurzen Verzögerung erscheint er noch einmal links daneben, diesmal jedoch viel größer und schwarz umrandet.

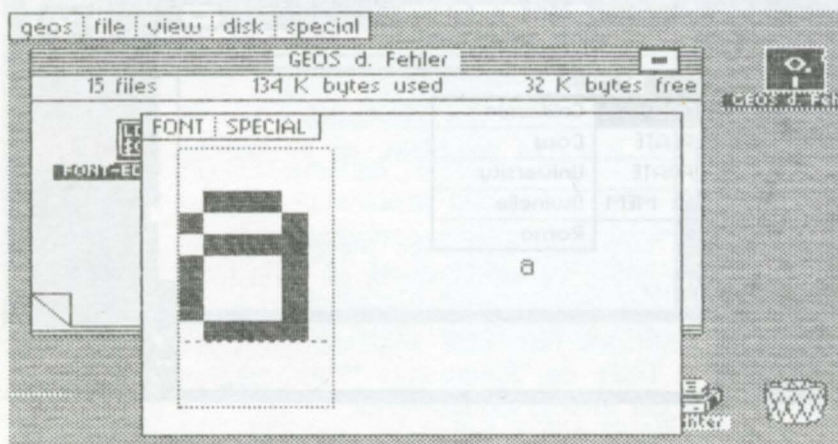


Abb. 28: Ein "a" in University 12 point

Innerhalb dieses begrenzten Feldes (EDIT-Feld) können Sie nun das Zeichen mit der Maus edieren. Ein Knopfdruck bewirkt dabei immer eine Umkehrung des aktuellen Zustandes. Eine helle

Fläche wird also dunkel und umgekehrt. Um das Ergebnis sofort "in natura" sehen zu können, wird immer auch das Original-Zeichen rechts daneben mit ausgegeben.

Eine Hilfslinie im Editierfeld zeigt außerdem noch an, wo sich bei einer späteren Ausgabe die Grundlinie des Zeichens befindet. Alle Punkte unterhalb dieser Linie werden nämlich als Unterlängen dargestellt. Beim "g" beispielsweise geht die Linie mitten durch den Buchstaben hindurch, da der untere Bogen des Zeichens etwas tiefer liegt als die Grundlinie des Zeichens. Besser wäre, man könnte die Einzelfelder kenntlich machen, z.B. durch ein Gitternetz oder durch einen Punkt in jeder Feldmitte.

Besonders Anwender der alten GEOS-Version 1.2 werden bei der Textverarbeitung die deutschen Umlaute vermißt haben. Unser Font-Editor ist hier genau das richtige Werkzeug. Versuchen Sie doch einmal, einem "o" zwei Pünktchen zu verpassen. Drücken Sie dazu die Taste "o", damit dieses Zeichen im EDIT-Feld angezeigt wird. Danach können Sie durch Klicken die passenden Punkte setzen.

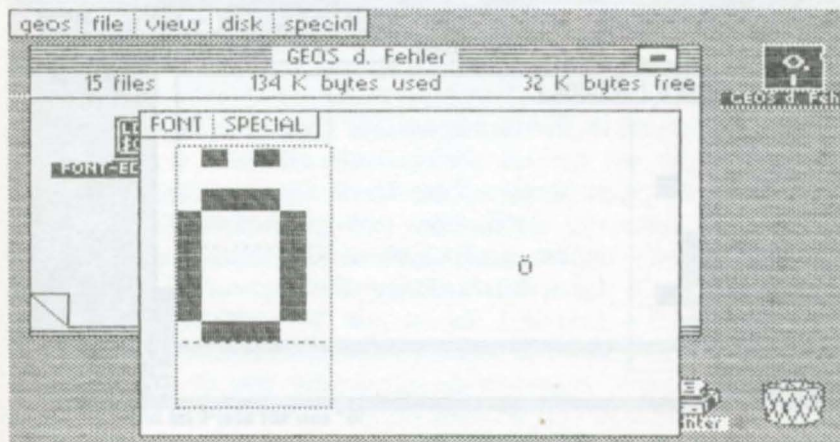


Abb. 29: Ein erstes "ö"

Wenn das "ö" fertig ist, sollten Sie aber nicht sofort den Font-Editor verlassen um voller Vorfreude das neuerstellte Zeichen zu benutzen, denn was nützt Ihnen das schönste "ö", wenn Sie stattdessen nun kein "o" mehr zur Verfügung haben. Wir sollten daher die neuen Zeichen auf Tasten legen, die zur Textverarbeitung nicht unbedingt benötigt werden. Hierbei bieten sich in GEOS V1.2 für Unlaute insbesondere Tasten an, die an der gleichen Stelle liegen wie bei einer Schreibmaschine. Versuchen wir daher, das "ö" auf den ":" der normalen C64-Tastatur zu legen.

Drücken Sie dazu erst einmal die Taste ":". Sofort erscheint der Doppelpunkt, und Sie können anfangen, ihn durch das "ö" zu ersetzen. Vielleicht werden Sie nun anfangen, mit dem Joystick Punkte zu setzen und zu löschen, aber nach kurzer Zeit werden Sie wahrscheinlich stirnrunzelnd auf den Bildschirm schauen. Irgendwas ist doch da nicht ganz in Ordnung. Das Zeichen ":" ist nämlich nur drei Spalten breit. In diese vorgegebenen Spalten paßt aber kaum ein "ö".

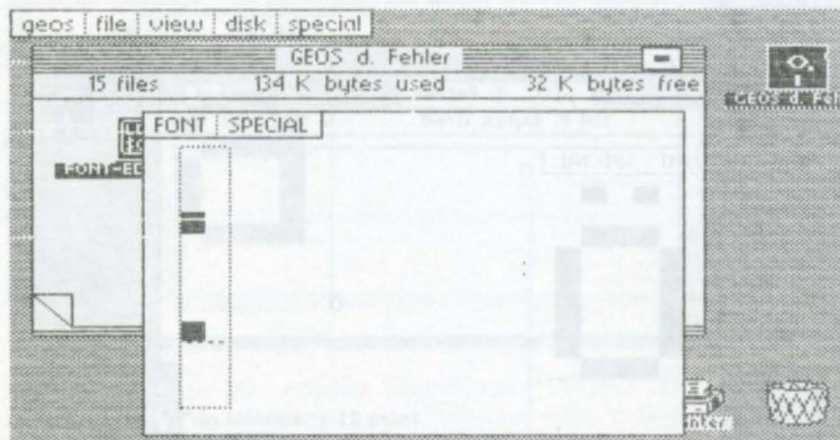


Abb. 30: Der Doppelpunkt ist nur 2 Spalten breit.

Dieses Problem taucht dadurch auf, daß die Schriftsätze aus Proportionalsschrift bestehen. Dadurch ist für jedes Zeichen eine unterschiedliche Breite vorgegeben. Wenn diese Vorgabe nicht zu ändern wäre, wäre unser Font-Editor nicht das Papier wert, auf dem er gedruckt ist. Es war zwar ein großes Programmier-Problem, die Breite der Zeichen veränderbar zu machen, weil die Zeichensätze äußerst kompliziert gespeichert sind, aber diese Fähigkeit mußte das Programm unbedingt erhalten. Drücken Sie die Taste, die Sie belegen möchten. Ist das hierbei erscheinende Feld zu schmal, positionieren Sie die Maus direkt rechts neben die rechte Feldbegrenzungslinie, und klicken Sie so oft, bis das Feld die gewünschte Größe erreicht hat. Sollte das Feld zu groß sein, gehen Sie analog vor, nur müssen Sie diesmal die Maus links neben die linke Feldbegrenzung setzen. Auf diese einfache Weise können Sie also die Breite eines Zeichens beliebig in weiten Grenzen verändern. Anschließend können Sie mit der Definition des Zeichens beginnen.

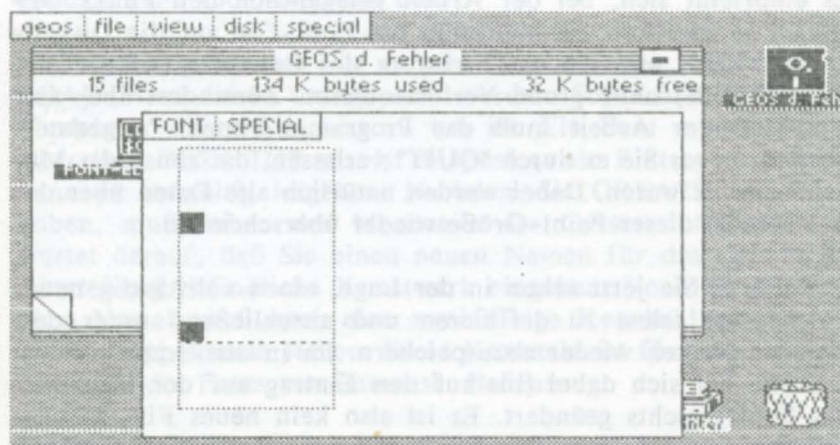


Abb. 31: Nun ist Platz für das "ö"

Selbstverständlich können jederzeit Änderungen der Zeichen-größe vorgenommen werden. Eine Verkleinerung bringt dabei jedoch immer Informationsverlust mit sich, da die hierbei "ab-

geschnittenen" Daten gelöscht werden. Wenn Sie also ein "a" bis auf eine Spalte verkleinern und anschließend wieder vergrößern, werden die "neuen alten" Spalten nur leer angehängt; das Zeichen ist verstümmelt.

An dieser Stelle möchten wir besonders die Besitzer der GEOS-Version 1.2 auf das Kapitel 5.3.4 "Beispiele mit dem Font-Editor" aufmerksam machen. Es ist nämlich gar nicht so einfach, die deutschen Umlaute zu erstellen. Damit die Pünktchen noch Platz finden, müssen die eigentlichen Zeichen nämlich etwas verkleinert werden. Deshalb zeigen wir Ihnen im Anschluß an das abgedruckte Listing an einigen Beispielen, wie die Änderungen der Zeichen am einfachsten vorgenommen werden. Dabei wird auch noch einmal die genaue Bedienung des Font-Editors für diese Zwecke beschrieben. Wenn Sie also auf Schwierigkeiten stoßen und etwas nicht sofort klappt, schauen Sie bitte ins Kapitel 5.3.4.

Es empfiehlt sich, bei der Arbeit gelegentlich den Punkt UPDATE anzuklicken, da hierdurch das Font-File auf die Diskette kopiert wird, und ein evtl. Absturz des Rechners (z.B. infolge Stromausfalls) ohne große Verluste bleibt. Zumindest aber zum Abschluß der Arbeit muß das Programm einmal "upgedated" werden, bevor Sie es durch "QUIT" verlassen, da sonst alle Mühen umsonst waren. Dabei werden natürlich alte Daten über den Zeichensatz dieser Point-Größe wieder überschrieben.

Damit sind Sie jetzt schon in der Lage, einen vollständig neuen Zeichensatz selbst zu definieren und anschließend unter dem gleichen Namen wieder abzuspeichern. Im Inhaltsverzeichnis der Diskette hat sich dabei (bis auf den Eintrag auf der Info-Seite des Fonts) nichts geändert. Es ist also kein neues File dazugekommen, sondern nur ein schon bestehendes modifiziert worden. Möchten Sie auch weiterhin den unveränderten Zeichensatz benutzen, können Sie vorher vom DESK TOP aus eine Kopie des Fonts anfertigen.

Erstellen eines neuen Zeichensatzes

Zu Beginn erwähnten wir, daß auch die Möglichkeit existiert, völlig neue Zeichensätze zu definieren, ohne vorher einen schon vorhandenen laden zu müssen. Den dafür vorgesehenen Befehl CREATE haben Sie sicher schon im Menü entdeckt. Jetzt dürfen Sie ihn zum ersten Mal anklicken. Falls Sie vorher schon einen Font modifiziert haben, werden diese Daten vorher noch ordnungsgemäß auf die Diskette geschrieben. Ein Anklicken von UPDATE ist also vor CREATE überflüssig.

Eine Voraussetzung muß nun jedoch erfüllt sein: Da ein neuer Eintrag im Directory erstellt wird, benötigt das Programm zumindest ein beliebiges Font-File auf der aktuellen Diskette. Der Editor ist nämlich nicht sehr kreativ und muß sich das Icon (also das zum neuen File gehörende Bildchen) von einem anderen Font abkupfern. Natürlich hätten wir auch diese Daten mit in das Programm aufnehmen können, aber die DATA-Liste unten genügt Ihnen auch so schon, oder?

Ob sich ein geeignetes File auf der Disk befindet, kann mit dem LOAD-Menü leicht nachgeprüft werden. Sollte hier "*" No File "*" stehen, wissen Sie, was zu tun ist: Ein Font muß her, und zwar vom DESK TOP aus. Ist mindestens ein Eintrag vorhanden, kann es endgültig losgehen: Nachdem Sie CREATE angeklickt haben, muß neben der Menüzeile der Cursor auftauchen. Er wartet darauf, daß Sie einen neuen Namen für das sich in der Enzstehung befindliche Kunstwerk eingeben. Vor dem eigentlichen Namen muß noch eine zweiziffrige Kennzahl gefolgt von einem Doppelpunkt stehen. Diese Kennzahl ist für eine korrekte Funktion des Fonts von immenser Bedeutung.

Da ein Rechner besser mit Zahlen als mit Namen umgehen kann, unterscheidet GEOS die Zeichensätze intern durch eine Kennziffer. So besitzt BSW die Nummer 0, California die Nummer 1 usw. in der Reihenfolge Cory, Dwinelle, Roma und University. Es sind also die Nummern 0 bis 5 schon vergeben. Theoretisch könnten Sie also mit 06 für eigene Zeichensätze weitermachen. Da aber wie gesagt inzwischen von Berkeley Softworks weitere Fonts angeboten werden, empfehlen wir, die

Numerierung für eigene Fonts ab 50 zu beginnen. Dadurch haben Sie genug "Luft" für künftige Ergänzungsfonts. Beim ersten Mal geben Sie also ein: "50:xxx". Hierbei steht "xxx" für einen beliebigen Namen.

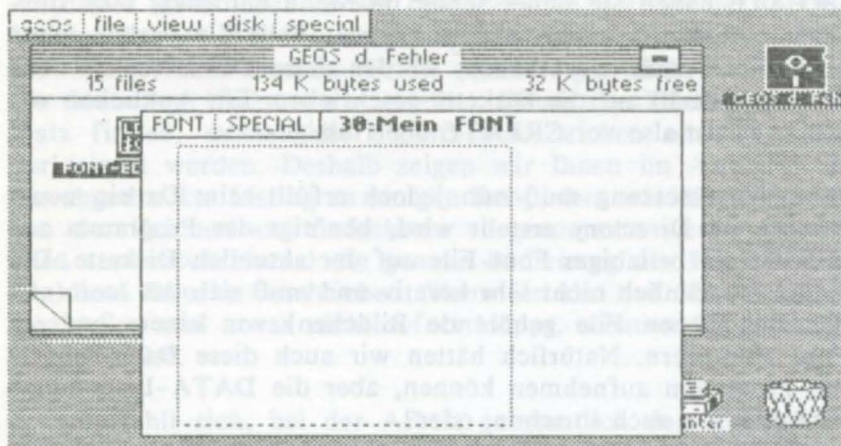


Abb. 32: "30:Mein FONT"

Sollte Sie jetzt die Phantasie im Stich lassen, macht das auch nichts: Die drei Kreuzchen genügen auch vorerst, denn im DESK TOP haben Sie später immer noch Gelegenheit, einen würdigen Namen zu finden. Wie immer beendet auch hier RETURN die Eingabe. Bei dem darauf folgenden Geräusch in der Floppy handelt es sich um den Icon-Raub für Ihr neues File.

Jetzt kann die Eingabe beginnen: Nach dem Drücken einer Taste erscheint... nichts, außer einem Rahmen. Dieser begrenzt wieder - wie beim Ändern der Zeichensätze - den editierbaren Bereich für das gedrückte Zeichen. Ein Druck auf den Feuerknopf innerhalb des Feldes bewirkt also eine Invertierung eines Bereiches. Bei einem Neustart des Font-Editors entsteht aber das Problem, daß das ganze Feld nur ein einziger Bereich ist, der sich bei Knopfdruck komplett schwarz färbt. Um einen lesbaren Zeichensatz zu definieren, muß man schon talentiert sein.

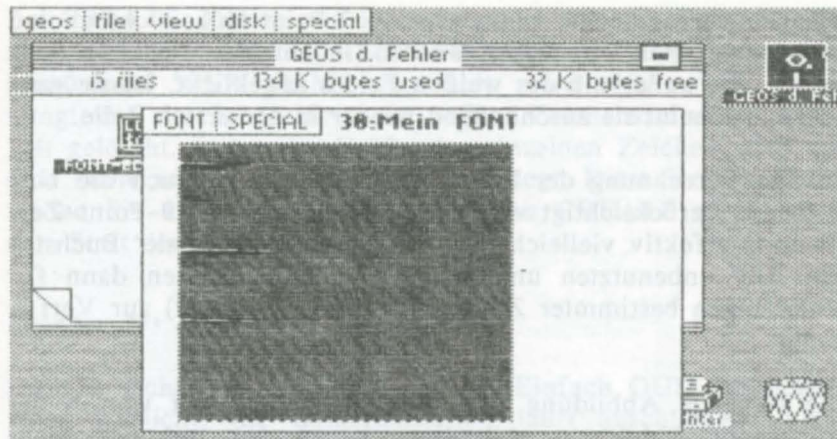


Abb. 33: Das erste Zeichen, ein riesiger Punkt

Da wir diese Begabung nicht besitzen, haben wir eine Funktion eingebaut, die es erlaubt, die Höhe des Zeichensatzes zu bestimmen. Wenn Sie den letzten Abschnitt aufmerksam gelesen haben (besonders die Möglichkeiten zur Breitenänderung eines Zeichens) und einmal scharf überlegen, könnten Sie darauf kommen, wie man im Font-Editor die Höhe der Zeichen festlegt. Richtig: Wird die Maus am unteren Bildschirmrand geklickt, vergrößert sich der Satz um eine Zeile, die unten angefügt wird. Analog kann die letzte Zeile entfernt werden, nur muß diesmal die Maus über dem oberen Rand des Fensters angeklickt werden.

Diese Auswahl muß vor jeder Erstellung eines neuen Zeichensatzes erfolgen. Sie entscheidet direkt, unter welcher Point-Größe das File später abgespeichert wird. Wenn Sie z.B. sofort UPDATE anklicken, befindet sich ein Font "xxx 1 Point" auf der Diskette. Es ist also eine wichtige Entscheidung, die zu Beginn getroffen werden muß.

Mindestens genauso wichtig ist die Festlegung der Nulllinie des Satzes. Sie gibt an, wie viele Punkte Unterlängen darstellen sol-

len. Diese waagerechte Linie wird ab 2 Point Höhe sichtbar und kann nach unten verschoben werden, indem man die Maus ganz rechts, also außerhalb des weißen Feldes angeklickt. Ist sie ganz unten, erscheint sie anschließend wieder in der ersten Zeile.

Bei der Berechnung der Point-Höhe müssen also auch die Unterlängen berücksichtigt werden. So bietet z.B. ein 9-Point-Zeichensatz effektiv vielleicht nur 7 Zeilen für "normale" Buchstaben. Die unbenutzten unteren beiden Zeilen stehen dann für Unterlängen bestimmter Zeichen, (z.B. Q, q oder y) zur Verfügung.

Die folgende Abbildung zeigt den gleichen Punkt wie in der vorherigen Abbildung. Allerdings haben wir mit der Maus fünfmal rechts, achtmal unten und sechsmal ganz rechts geklickt. Dadurch haben wir einen Zeichensatz mit 9 Point, davon sind 2 Point Unterlängen. Dieses Zeichen des Fonts besitzt 6 Spalten. Zu guter Letzt gehen wir noch auf den Menüpunkt "CLR MEM" (Abk. für CLEAR MEMORY, übersetzt: "Speicher löschen") ein.

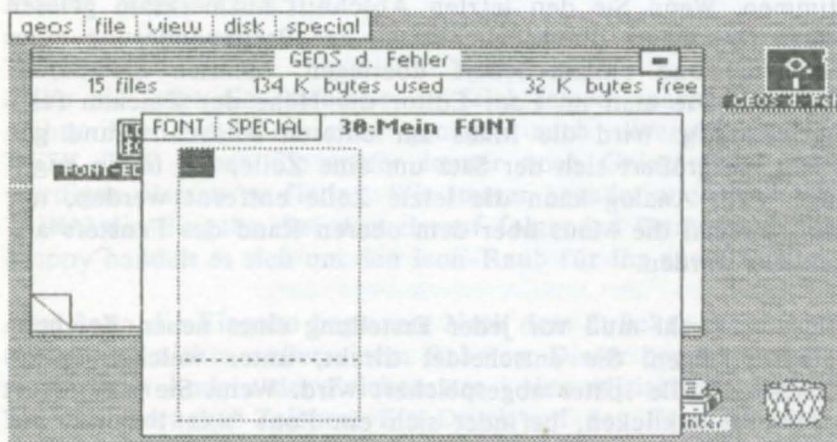


Abb. 34: 9 point mit 2point Unterlänge

Bei CREATE werden alle Informationen über die Zeichengröße vom vorher bearbeiteten Font übernommen. Wird jedoch der Menüpunkt "CLR MEM" angewählt, wird sofort (ohne Warnung!) der sich im Rechner befindliche Zeichensatz unwiderruflich gelöscht. Selbst die Größe der einzelnen Zeichen wird auf eine Höhe und Breite von Eins festgelegt. Diese Konfiguration finden Sie auch nach der Anwahl von CREATE direkt nach dem Start des Font-Editors vor.

Beenden des Programms

Dies ist sicher die leichteste Übung: Einfach QUIT anklicken. Aber Vorsicht, die Diskettenstation läuft anschließend nicht umsonst an: Als das Programm geladen wurde, sind dabei wichtige Daten auf der Disk abgelegt worden. GEOS-Profis wissen, was gemeint ist: Ein sogenanntes Swap-File wird wieder geladen. Dieses findet der Rechner natürlich nur auf der Diskette, von der unser Font-Editor geladen wurde. Sollte nicht die richtige Diskette eingelegt sein, passiert das, was dann bei allen GEOS-Hilfsprogrammen passiert: Harakiri des Rechners, aber natürlich nur softwaremäßig. Nach einem Neustart von GEOS ist wieder alles im Lot.

Auf eine Sache sollten Sie aber noch achten, wenn Sie den Font-Editor von GEOWRITE und GEOPAINT aus benutzen. Es kann vorkommen, daß der neue Font nicht sofort zu Ihrer Verfügung steht. Das liegt nicht etwa an einem Fehler des Editors, sondern an der Konzeption der Hilfsprogramme. Diese können nämlich etwas auf der Diskette ändern, ohne daß die zugehörigen Anwendungen davon informiert werden müssen. Sollte das bei Ihnen vorkommen, so verlassen Sie die Anwendung und starten sie anschließend neu. Dadurch können dann auch die Änderungen auf der Diskette, also auch neue Fonts, mitbenutzt werden.

An der folgenden Abbildung sehen Sie, wie der Info-Sektor eines neuerstellten Fonts aussieht. Der Eintrag unter "class" ist sehr wichtig, damit Sie im Nachhinein feststellen können, welche Kennzahlen Sie schon für Ihre Fonts vergeben haben.

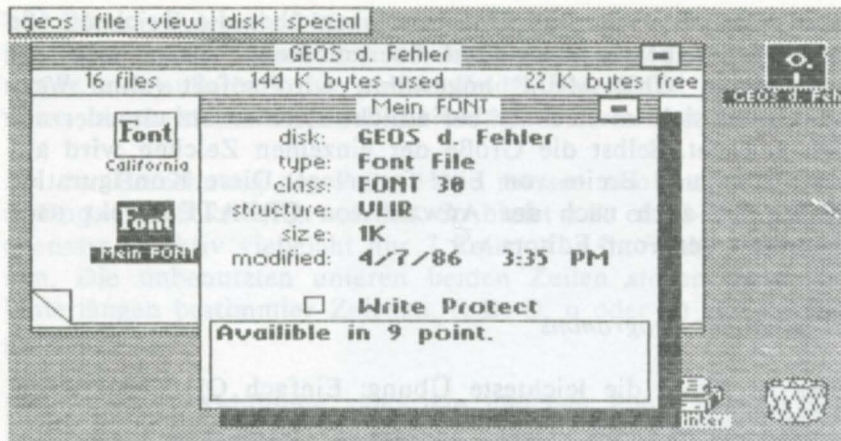


Abb. 35: class: FONT 30

5.3.2 Bedienungsreferenz

Damit Sie sich bei auftretenden Fragen nicht jedesmal durch die gesamte Bedienungsanleitung kämpfen müssen, haben wir in diesem Abschnitt noch einmal kurz die Menüpunkte und deren Funktion in alphabetischer Reihenfolge zusammengefaßt.

CLR MEM Löscht den Arbeitsspeicher ohne Vorwarnung vollständig. Alle Zeichen sind anschließend 1 Punkt hoch und 1 Punkt breit. Die Größe der einzelnen Zeichen kann anschließend mit der Maus festgelegt werden.

CREATE Dient zur Erstellung völlig neuer Zeichensätze. Nach dem Anklicken werden eventuelle Änderungen an einem Zeichensatz noch abgespeichert (UPDATE), und die aktuelle Arbeit wird beendet. Nun erscheint neben dem Menü der Cursor für die Eingabe einer zweiziffrigen Kennzahl gefolgt von einem Doppelpunkt und dem Namen für den neuen Font.

Wichtig!

Kennzahlen dürfen nicht mehrfach vergeben werden, da sonst GEOS die Zeichensätze nicht unterscheiden kann. Bei selbst erstellten Zeichensätzen finden Sie die Kennzahl später im Info-Bildschirm des Fonts unter KLASSE wieder (z.B. "KLASSE: Font 50"). Die Ziffern 00 bis 30 sind bereits vergeben. Da weitere Zeichensätze auf den Markt kommen werden, empfiehlt es sich, die Numerierung erst ab 50 zu beginnen. Es muß sich schon ein beliebiger Font auf der Diskette befinden. Außerdem muß auf ihr genügend Platz (je nach Point-Größe mind. 25 KByte) vorhanden sein.

FONT Zeigt das erste Untermenü.

LOAD Zeigt die ersten fünf Fonts der aktuellen Diskette.

Wird ein Font-File nicht angezeigt, muß es vom DESK TOP im Inhaltsverzeichnis weiter nach vorne gebracht werden.

Achtung!

Nach dem Anklicken von LOAD wird ein eventuell bisher in Arbeit befindlicher Zeichensatz aktualisiert abgespeichert. Die weitere Arbeit mit diesem Zeichensatz wird nun automatisch beendet. Ein Anklicken von UPDATE ist also vor LOAD nicht notwendig (schadet aber auch nicht). Wird der Name eines Font angeklickt, erscheinen die vorhandenen Größen. Nach dem Anklicken einer Größe wird der entsprechende Font geladen.

NEW DISK Muß nach jedem Wechsel der Diskette unbedingt angeklickt werden. Entspricht dem Befehl ÖFFNEN des Menüs DISKETTE im DESK TOP. Falls noch Änderungen an einem Zeichensatz gemacht worden sind, die übernommen werden sollen, muß vor dem Wechseln der Diskette das UPDATE-Feld angeklickt werden. Mit NEW DISK wird die derzei-

tige Arbeit beendet, und die Namen der neuen Zeichensätze werden in die LOAD-Liste übernommen.

UPDATE Speichert das gerade bearbeitete File auf Diskette ab. Reagiert nur bei tatsächlich vorangegangenen Änderungen. Alte Daten über das File gleicher Point-Höhe werden dabei überschrieben. Ein UPDATE des aktuellen Zeichensatzes wird automatisch ausgeführt bei LOAD und CREATE, nicht jedoch bei QUIT.

SPECIAL Zeigt das zweite Untermenü.

QUIT Korrektes Beenden des Programms. Vorher muß unbedingt die Ladediskette des Font-Editors eingelegt werden. QUIT führt keine UPDATE-Funktion aus, da sich das zugehörige Font-File nicht unbedingt auf der Ladediskette befinden muß. Eventuell muß daher bei noch eingelegter Arbeitsdiskette das UPDATE-Feld angeklickt werden, bevor mit der Ladediskette QUIT gewählt wird.

5.3.3 Listing des Programms

Geben Sie das nun folgende Listing bitte sehr sorgfältig ein, da schon eine einzige falsche Ziffer oder ein fehlendes Semikolon fatale Auswirkungen auf die Funktion hat. Wir haben zwar wie immer eine Prüfsummenfunktion installiert, die die Quersumme der DATAs berechnet und mit einem Sollwert vergleicht, aber $1+3$ ist nun einmal das gleiche wie $2+2$. Fehler können sich also durchaus wieder kompensieren und werden dann nicht erkannt. Sollte sich immer wieder ein Fehler einstellen, lassen Sie am besten einmal einen Bekannten Korrektur lesen, da man selbst gerne eigene Fehler überliest. Speichern Sie das Programm nach der Eingabe unbedingt erst ab, bevor Sie es starten. Legen Sie anschließend eine GEOS-Arbeitsdiskette ein, die noch mindestens ca. 10 KByte Speicher aufweist und keine wichtigen Daten enthält. Starten Sie nun das Programm durch die Eingabe von RUN. Es erscheint oben links auf dem Bildschirm ein Countdown. Falls keine Fehler auftreten, dauert es anschließend noch

eine Weile bis die "O.K."-Meldung erscheint. Der Font-Editor steht nun, mit einem schönen Icon versehen, zu Ihrer Verfügung.

```

0 REM *****
2 REM * PROGRAMM : FONT EDITOR V1.3 *
4 REM * AUTOR   : RUEDIGER KERKLOH *
6 REM * UPDATE  : 03/30/88 *
8 REM *****
10 :
12 OPEN 1,8,15,"I:0":REM FEHLERKANAL
14 RESTORE:PRINT CHR$(147):REM CLR HOME
16 REM * CHECKSUMME BERECHNEN *
18 FOR I=1 TO 2459
20 READ A:S=S+A
22 PRINT CHR$(19);2459-I;CHR$(157);" ";
24 NEXT:RESTORE
26 IF S<>264104 THEN PRINT "FEHLER IN DATAS !":STOP
28 REM * FILE ABSPEICHERN *
30 F$="FONT EDITOR"
32 OPEN 2,8,2,F$+"U,W":GOSUB 104
34 FOR I=1 TO 2346:READ A
36 PRINT#2,CHR$(A);:NEXT:GOSUB 104
38 CLOSE2
40 REM * INFO-SEKTOR ANLEGEN *
42 A=1:B=0
44 T=A:S=B:PRINT#1,"B-A:0";T;S
46 INPUT#1,N,T$,A,B:IF A=18 THEN A=19:B=0
48 IF N>0 THEN 44
50 OPEN 3,8,3,"#0":PRINT#3
52 PRINT#1,"U1";3;0;T;S:REM SEKTOR LESEN
54 FOR I=1 TO 113:READ B:PRINT#3,CHR$(B);:NEXT
56 FOR I=114 TO 256
58 PRINT#3,CHR$(0);:REM REST LOESCHEN
60 NEXT
62 PRINT#1,"U2";3;0;T;S:REM SCHREIBEN
64 REM * IN DIRECTORY EINTRAGEN *
66 :
68 PRINT#3
70 A$=CHR$(18):B$=CHR$(1):REM DIRECTORY
72 A=ASC(A$):B=ASC(B$):PRINT#1,"U1";3;0;A;B
74 GET#3,A$,B$:REM NAECHSTER SEKTOR
76 FOR I=0 TO 7:REM ANZAHL EINTRAEGE
78 PRINT#1,"B-P";3;I*32+5
80 FOR J=1 TO LEN(F$)

```

```
82 GET#3,W$:IF W$<>MID$(F$,J,1) THEN K=0:GOTO 86
84 K=K+1
86 NEXT J
88 IF K=LEN(F$) THEN 92:REM FOUND
90 NEXT I:GOSUB 104:GOTO 72
92 PRINT#1,"B-P";3;I*32+21
94 PRINT#3,CHR$(T)CHR$(S)CHR$(0)CHR$(5);
96 PRINT#3,CHR$(88)CHR$(3)CHR$(30)CHR$(13)CHR$(15)CHR$(7);
98 PRINT#1,"U2";3;0;A;B:REM DIR ZURUECK
100 CLOSE1:PRINT"O.K.":END
102 :
104 INPUT#1,N,T$:IF N THEN PRINT"ERROR: ";T$:STOP
106 RETURN
108 :
110 DATA 162,7,181,112,157,73,74,202,16,248,162,0,32,45,68,162,0,189,37,
73
112 DATA 149,6,232,224,6,208,246,169,0,32,57,193,32,36,193,169,255,32,
39,193
114 DATA 32,196,67,32,75,193,162,49,160,73,134,2,132,3,169,1,32, 32,81,
193,96
116 DATA 173,183,132,208,24,173,4,133,201,32,144,17,201,127,176,13,
162,90
118 DATA 160,65,142,169,132,140,170,132,32,162,70,96,173,183,132,208,
250,169
120 DATA 255,141,62,74,173,23,74,197,60,144,7,32,138,70,32,162,70,96,
160,0
122 DATA 166,41,109,63,74,197,60,176,11,200,202,208,245,32,103,70,32,
162,70
124 DATA 96,140,65,74,165,59,240,7,32,129,68,32,162,70,96,162,0,172,
64,74
126 DATA 173,22,74,10,10,10,197,58,144,7,32,220,69,32,162,70,96,109,
63,74
128 DATA 197,58,176,11,232,136,208,245,32,252,68,32,162,70,96,172,65,
74,173
130 DATA 4,133,56,32,147,68,32,162,70,96,72,32,36,72,104,174,61,74,
240,10
132 DATA 32,144,193,169,0,133,2,133,3,96,133,2,10,10,10,10,101,2,105,
82,133
134 DATA 2,169,0,105,82,133,3,160,0,177,2,153,45,74,240,3,200,208,246,
32,116
136 DATA 194,138,208,210,168,162,0,189,9,74,153,167,82,200,232,224,
11,208
138 DATA 244,192,110,208,238,169,0,141,202,73,141,69,74,32,122,194,224,
8,240
140 DATA 64,152,240,56,173,202,73,10,10,10,109,202,73,109,202,73,109,
202,73
```

142 DATA 105,2,168,174,69,74,232,169,0,248,24,105,1,202,208,251,216,
170,74
144 DATA 74,74,74,240,5,9,48,153,167,82,200,138,41,15,9,48,153,167,
82,238
146 DATA 202,73,238,69,74,208,185,32,119,194,174,202,73,173,196,73,
202,48
148 DATA 5,24,105,14,208,248,141,197,73,173,202,73,9,128,141,202,73,
162,196
150 DATA 160,73,134,2,132,3,96,72,32,144,193,162,45,160,74,134,2,132,
3,32
152 DATA 116,194,169,255,141,150,132,104,72,32,122,194,152,240,250,
104,56
154 DATA 233,1,16,243,162,21,160,83,134,16,132,17,162,0,160,20,134,6,
132,7
156 DATA 32,140,194,138,208,3,32,0,73,32,119,194,96,32,144,193,32,36,
72,173
158 DATA 61,74,208,244,32,141,193,32,205,71,32,232,67,32,75,193,169,
27,32
160 DATA 69,193,32,159,193,47,57,120,0,250,0,169,0,141,42,74,141,44,
74,133
162 DATA 4,162,125,160,0,169,49,134,24,132,25,133,5,169,15,133,6,162,
42,160
164 DATA 74,134,2,132,3,162,31,160,67,142,163,132,140,164,132,32,186,
193,96
166 DATA 173,44,74,201,58,208,192,32,138,193,162,82,160,82,32,14,73,
208,120
168 DATA 162,77,157,0,129,232,208,250,189,110,73,157,77,129,240,3,
232,208
170 DATA 245,169,32,157,77,129,173,42,74,157,78,129,173,43,74,157,79,
129,169
172 DATA 0,157,80,129,173,43,74,41,15,72,173,42,74,41,15,170,104,202,
48,5
174 DATA 24,105,10,208,248,141,128,129,162,45,160,74,142,0,129,140,1,
129,162
176 DATA 0,160,129,134,20,132,21,169,0,133,22,32,237,193,138,208,143,
162,45
178 DATA 160,74,134,2,132,3,32,116,194,138,208,13,32,137,194,138,240,
250,224
180 DATA 9,208,3,32,0,73,32,119,194,96,32,144,193,32,57,72,96,32,144,
193,173
182 DATA 163,132,13,164,132,240,59,32,196,67,96,162,0,189,26,74,157,
21,83
184 DATA 232,224,8,208,245,169,0,170,168,152,157,29,83,232,169,0,157,
29,83
186 DATA 232,200,192,98,208,240,32,205,71,166,44,164,45,134,4,132,5,
169,0

188 DATA 133,2,169,20,133,3,32,120,193,96,32,144,193,169,0,141,62,74,
 32,36
 190 DATA 72,32,225,193,162,7,189,73,74,149,112,202,16,248,162,194,160,
 62,142
 192 DATA 156,132,140,155,132,96,32,144,193,169,0,141,62,74,32,36,72,
 32,225
 194 DATA 193,169,5,133,17,138,208,49,162,82,160,82,134,14,132,15,169,
 8,133
 196 DATA 16,169,0,133,22,133,23,169,0,141,61,74,32,59,194,138,240,20,
 169,255
 198 DATA 141,61,74,189,253,73,157,82,82,232,224,12,208,245,169,1,208,
 7,169
 200 DATA 5,56,229,17,240,229,170,9,128,141,170,73,173,164,73,202,48,5,
 24,105
 202 DATA 14,208,248,141,165,73,96,174,21,83,232,236,24,83,208,2,162,
 0,142
 204 DATA 21,83,32,205,71,96,8,142,65,74,140,66,74,56,233,32,10,168,
 177,42
 206 DATA 24,109,65,74,133,112,200,177,42,105,0,133,113,165,112,41,7,73,
 7,133
 208 DATA 114,70,113,102,112,70,113,102,112,70,113,102,112,162,0,236,
 66,74
 210 DATA 240,14,165,112,24,101,39,133,112,144,2,230,113,232,208,237,
 165,112
 212 DATA 24,101,44,133,112,165,113,101,45,133,113,160,0,166,114,189,
 34,74
 214 DATA 133,114,40,144,4,81,112,145,112,49,112,240,2,56,96,24,96,
 160,192
 216 DATA 177,42,24,105,1,141,70,74,200,177,42,105,0,141,71,74,162,3,
 78,71
 218 DATA 74,110,70,74,202,208,247,238,70,74,165,39,205,70,74,176,60,
 164,41
 220 DATA 136,140,70,74,169,127,32,201,193,170,202,172,70,74,169,127,
 24,32
 222 DATA 147,68,166,39,160,0,177,112,172,70,74,145,112,165,112,56,233,
 1,133
 224 DATA 112,165,113,233,0,133,113,202,208,231,206,70,74,208,208,238,
 22,83
 226 DATA 32,205,71,165,44,24,101,39,141,70,74,174,64,74,160,0,173,4,
 133,24
 228 DATA 32,147,68,173,70,74,56,229,112,141,71,74,164,41,136,140,70,
 74,174
 230 DATA 64,74,202,172,70,74,173,4,133,24,32,147,68,160,0,166,114,
 202,138
 232 DATA 49,112,74,8,72,138,73,255,49,112,145,112,104,17,112,145,112,
 165,112

234 DATA 141,187,69,165,113,141,188,69,162,1,172,71,74,40,136,240,6,
126,255

236 DATA 255,232,208,247,206,70,74,16,189,173,4,133,56,233,31,10,170,
254,29

238 DATA 83,208,3,254,30,83,232,232,224,194,144,242,96,165,44,24,101,
39,141

240 DATA 70,74,174,64,74,202,240,241,160,0,173,4,133,24,32,147,68,173,
70,74

242 DATA 56,229,112,141,71,74,164,41,136,140,70,74,174,64,74,202,172,
70,74

244 DATA 173,4,133,24,32,147,68,165,112,141,34,70,165,113,141,35,70,
174,71

246 DATA 74,202,240,7,24,62,255,255,202,208,250,8,160,0,166,114,202,
138,49

248 DATA 112,40,42,72,165,114,10,56,233,1,73,255,49,112,145,112,104,
17,112

250 DATA 145,112,206,70,74,16,185,173,4,133,56,233,31,10,170,189,29,
83,233

252 DATA 0,157,29,83,201,255,208,3,222,30,83,232,232,224,194,144,235,
96,173

254 DATA 24,83,201,62,176,27,238,24,83,32,205,71,162,0,164,41,136,169,
32,24

256 DATA 32,147,68,169,0,168,145,112,200,196,39,208,249,96,174,24,83,
224,1

258 DATA 240,16,202,142,24,83,202,236,21,83,176,3,142,21,83,32,205,71,
96,162

260 DATA 0,189,43,73,149,6,232,224,6,208,246,32,36,193,162,195,160,0,
169,120

262 DATA 134,24,132,25,133,5,173,4,133,32,69,193,162,81,160,74,134,4,
132,5

264 DATA 162,1,160,8,134,2,132,3,32,120,193,173,4,133,32,201,193,240,
195,141

266 DATA 64,74,197,41,176,2,165,41,133,112,162,255,169,128,232,56,
229,112

268 DATA 176,250,142,63,74,169,0,141,66,74,174,64,74,202,142,65,74,
172,63

270 DATA 74,169,0,141,75,71,24,109,66,74,136,208,250,141,74,71,160,4,
14,74

272 DATA 71,46,75,71,136,208,247,169,82,24,109,74,71,141,74,71,133,
118,169

274 DATA 74,109,75,71,141,75,71,133,119,174,65,74,172,66,74,173,4,133,
24,32

276 DATA 147,68,172,63,74,8,8,162,0,40,126,255,255,8,232,224,16,208,
246,40

278 DATA 40,136,208,237,206,65,74,16,216,174,63,74,202,240,32,165,
118,133

280 DATA 116,165,119,133,117,165,116,24,105,16,133,118,165,117,105,0,
 133,119
 282 DATA 160,15,177,116,145,118,136,16,249,48,221,238,66,74,173,66,
 74,197
 284 DATA 41,240,3,76,251,70,32,156,71,32,217,71,169,223,37,57,133,57,
 96,160
 286 DATA 0,162,0,189,20,74,149,2,232,224,6,208,246,152,72,240,16,230,
 5,165
 288 DATA 2,24,105,16,133,2,144,2,230,3,136,208,240,169,144,145,2,32,
 66,193
 290 DATA 104,168,200,192,128,208,210,96,169,21,133,2,169,83,133,3,32,
 204,193
 292 DATA 96,173,23,74,133,6,174,63,74,24,202,48,4,101,41,208,249,233,
 0,133
 294 DATA 7,169,0,133,9,133,11,173,22,74,10,10,10,133,8,174,63,74,202,
 48,5
 296 DATA 109,64,74,208,248,233,0,133,10,169,85,32,39,193,165,6,166,38,
 24,109
 298 DATA 63,74,202,16,250,233,0,133,24,169,153,32,24,193,96,169,0,
 141,169
 300 DATA 132,141,170,132,141,163,132,141,164,132,32,57,72,32,75,193,
 96,173
 302 DATA 62,74,240,250,162,45,160,74,134,2,132,3,32,116,194,169,127,
 32,201
 304 DATA 193,170,164,41,136,169,127,24,32,147,68,165,112,56,233,20,133,
 6,133
 306 DATA 112,165,113,233,83,133,7,133,113,162,21,160,83,134,16,132,
 17,165
 308 DATA 41,141,150,132,32,143,194,32,119,194,162,45,160,74,32,14,73,
 162,254
 310 DATA 232,232,189,130,129,*41,31,240,42,197,41,240,38,144,241,160,
 28,185
 312 DATA 129,129,153,131,129,185,128,129,153,130,129,185,96,129,153,
 98,129
 314 DATA 185,95,129,153,97,129,136,136,232,232,224,28,144,224,152,
 170,169
 316 DATA 0,133,6,173,128,129,10,38,6,10,38,6,10,38,6,10,38,6,10,38,6,
 10,38
 318 DATA 6,5,41,157,130,129,165,6,157,131,129,165,112,157,97,129,165,
 113,157
 320 DATA 98,129,174,19,132,172,20,132,134,4,132,5,162,0,160,129,134,
 10,132
 322 DATA 11,32,231,193,169,0,141,62,74,96,162,60,160,65,142,163,132,
 140,164
 324 DATA 132,32,205,71,96,134,14,132,15,32,11,194,138,208,12,162,0,
 160,132

```
326 DATA 134,20,132,21,32,41,194,138,96,46,194,50,0,0,1,55,183,195,0,
255,0
328 DATA 46,58,50,0,118,0,2,110,73,128,66,73,115,73,128,93,73,59,115,
50,0
330 DATA 97,0,132,123,73,128,164,73,128,73,0,202,66,135,73,0,174,67,
142,73
332 DATA 0,181,67,59,87,79,0,129,0,130,150,73,0,252,67,155,73,0,31,68,
70,79
334 DATA 78,84,0,83,80,69,67,73,65,76,0,76,79,65,68,0,67,82,69,65,84,
69,0
336 DATA 85,80,68,65,84,69,0,67,76,82,32,77,69,77,0,81,85,73,84,0,78,
69,87
338 DATA 32,68,73,83,75,0,59,0,97,0,159,0,0,82,82,64,204,65,99,82,64,
204,65
340 DATA 116,82,64,204,65,133,82,64,204,65,150,82,64,204,65,59,255,159,
0,212
342 DATA 0,255,167,82,0,139,66,178,82,0,139,66,189,82,0,139,66,200,82,
0,139
344 DATA 66,211,82,0,139,66,222,82,0,139,66,233,82,0,139,66,244,82,0,
139,66
346 DATA 255,82,0,139,66,10,83,0,139,66,42,32,78,79,32,70,73,76,69,32,
42,0
348 DATA 32,32,32,88,32,80,79,73,78,84,0,81,74,8,62,16,1,0,12,0,1,8,
0,202
350 DATA 0,1,2,4,8,16,32,64,128
352 REM INFO-DATEN
354 DATA 0,255,3,21,191,255,255,255,128,0,1,191,128,1,152,128,9,152,0,
25,152
356 DATA 0,25,158,0,61,152,231,153,153,182,217,153,182,217,153,182,
217,188
358 DATA 230,205,128,0,1,128,0,1,152,164,1,144,132,1,155,174,237,146,
164,169
360 DATA 155,166,233,128,0,1,255,255,255,130,5,0,0,65,63,127,0,65,198,79
362 DATA 78,84,32,197,68,73,84,79,82,32,214,49,46,51,0,0,0,77,210,85,
69,68
364 DATA 73,71,69,82,32,203,69,82,75,76,79,72
```

5.3.4 Beispiele mit dem Font-Editor

Das folgende ausführliche Beispiel für die Nutzung des Font-Editors gilt nur für GEOS V1.2, da bei GEOS V1.3 die deutschen Umlaute schon vorhanden sind. In diesem Abschnitt möchten wir Ihnen bei der Änderung Ihrer Zeichensätze helfen. Wir wollen daher gemeinsam den Zeichensatz "University 12

point" mit deutschen Umlauten versehen. Bevor Sie den Font-Editor starten, sollten Sie überlegen, ob Sie nur den Zeichensatz mit den deutschen Umlauten oder beide benutzen wollen. Im zweiten Fall stellen Sie bitte eine Kopie des Zeichensatzes mit DUPLICATE her. Nennen Sie die Kopie beispielsweise "University.d". Dann erinnert Sie das ".d" daran, daß dieser Zeichensatz die deutschen Umlaute enthält.

Wir werden die Umlaute auf die Tasten "Doppelpunkt", "Semikolon", "Klammeraffe" und "Plus-Zeichen" legen. Dadurch werden zwar wichtige Zeichen vorerst nicht mehr zur Verfügung stehen, aber die Tastaturbelegung entspricht der einer Schreibmaschine. Wir haben nämlich immer wieder Schwierigkeiten mit dem Umdenken gehabt (deutsche Textverarbeitung - GEOWRITE). Das möchten wir Ihnen auf Dauer ersparen. Wenn die Umlaute fertig sind, können Sie den "Doppelpunkt" auf die Taste SHIFT+Komma=eckige Klammer links und das "Semikolon" auf SHIFT+Punkt=eckige Klammer rechts legen. Wenn Sie dann noch "z" und "y" vertauschen, brauchen Sie sich nicht mehr umzustellen.

Haben Sie auch genug freien Speicher auf der Diskette, und befindet sich "University.d" unter den ersten 5 FONTS? Gut, dann können Sie den Font-Editor starten. Laden Sie anschließend mit LOAD den Zeichensatz "University.d" in 12 point ein. Bevor Sie mit der Änderung beginnen, sollten Sie sich einfach einmal verschiedene Zeichen in Groß- und Kleinschrift anschauen. Sie haben dann eine bessere Vorstellung davon, wie diese im EDIT-Feld aussehen. Schauen Sie sich bitte zum Abschluß das große "O" an, indem Sie SHIFT + "o" drücken.

Sie sehen in der folgenden Abbildung, daß über dem "O" kein Platz mehr für die beiden Pünktchen ist, die wir für ein "Ö" hinzufügen müßten. Das heißt, daß wir für unsere Umlaute nicht direkt die Zeichen o, a, u als Vorlage benutzen können. Damit die neuen Zeichen gut aussehen, benötigen wir sogar zwei Zeilen über dem Zeichen. Dann bleibt zwischen den Pünktchen und dem Zeichen eine Zeile frei. Beginnen wir für die Ände-

rung mit dem kleinen "ö". Schauen wir uns dazu zuerst einmal das kleine "o" an. Es belegt sieben Zeilen und ist fünf Spalten breit.

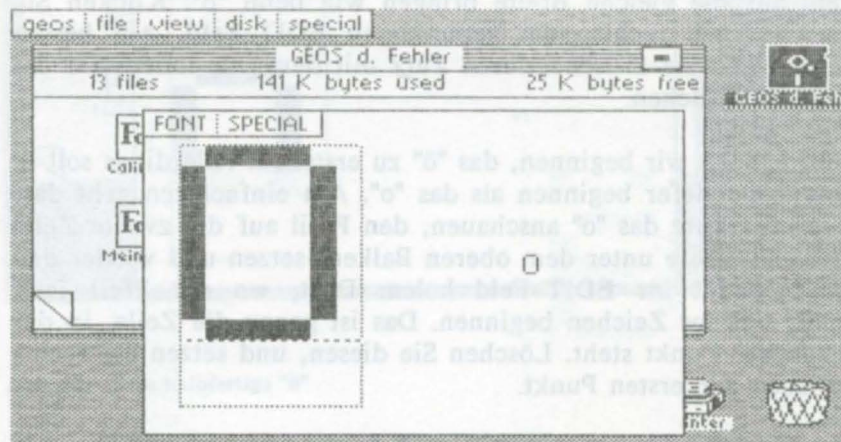


Abb. 35: Beim "O" ist kein Platz für die Pünktchen

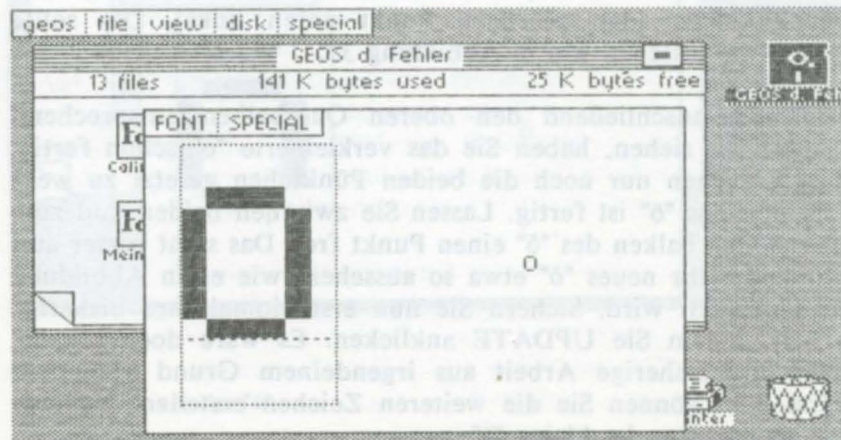


Abb. 36: Das "o" dient als Vorlage

Das EDIT-Feld ist allerdings sechs Spalten breit, weil jedes Zeichen rechts eine leere Spalte zur Trennung vom nächsten Zeichen im Text benötigt. Jetzt können wir beginnen, den Doppelpunkt in ein "ö" zu verwandeln. Zuerst wollen wir das EDIT-Feld auf die gleiche Breite bringen wie beim "o". Klicken Sie dazu viermal rechts vom gepunkteten EDIT-Feld. Am besten vergleichen Sie zwischendurch kurz mit dem "o", indem Sie die Taste "o" drücken.

Nun können wir beginnen, das "ö" zu erstellen. Allerdings soll es eine Zeile tiefer beginnen als das "o". Am einfachsten geht das, wenn Sie kurz das "o" anschauen, den Pfeil auf die zweite Zeile (die erste leere unter dem oberen Balken) setzen und wieder den Doppelpunkt ins EDIT-Feld holen. Dort, wo der Pfeil jetzt steht, soll das Zeichen beginnen. Das ist genau die Zeile, in der schon der Punkt steht. Löschen Sie diesen, und setzen Sie rechts daneben den ersten Punkt.

Da der obere Balken des "o" drei Punkte breit ist, setzen Sie noch zwei weitere Punkte rechts daneben. Ein prüfender Blick auf das "o": In der Zeile darunter müssen die senkrechten Balken in Spalte 1 und Spalte 5 sein. Ziehen Sie diese Balken nun bis zur zweituntersten Zeile, und löschen Sie den letzten Rest des Doppelpunktes (den gesetzten Punkt unten links). Das sollte dann etwa aussehen wie in Abbildung 37.

Wenn Sie anschließend den oberen Querbalken entsprechend nach unten ziehen, haben Sie das verkleinerte "o" schon fertig. Jetzt brauchen nur noch die beiden Pünktchen gesetzt zu werden, und das "ö" ist fertig. Lassen Sie zwischen beiden und zum oberen Querbalken des "ö" einen Punkt frei. Das sieht besser aus. Nun sollte Ihr neues "ö" etwa so aussehen, wie es in Abbildung 38 dargestellt wird. Sichern Sie nun erst einmal Ihre bisherige Arbeit, indem Sie UPDATE anklicken. Es wäre doch schade, wenn die bisherige Arbeit aus irgendeinem Grund vergebens wäre. Nun können Sie die weiteren Zeichen erstellen. Nehmen wir als nächstes das kleine "ä".

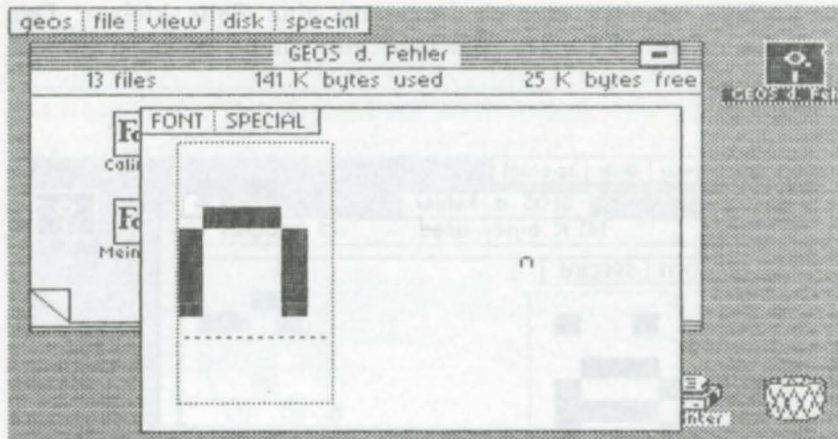


Abb. 37: Das halbfertige "ö"

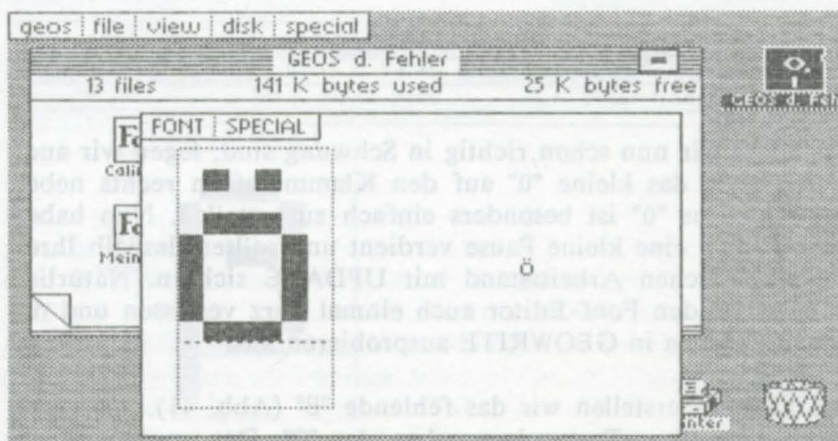


Abb. 38: Das fertige "ö"

Wir legen es auf das Semikolon. Dieses verbreitern wir solange, bis es die gleiche Breite hat wie das "a". Dann nehmen wir dieses

als Vorlage, machen es aber insgesamt eine Zeile kleiner. Das Ergebnis sollte dann so aussehen und dient gleichzeitig als Vorlage:

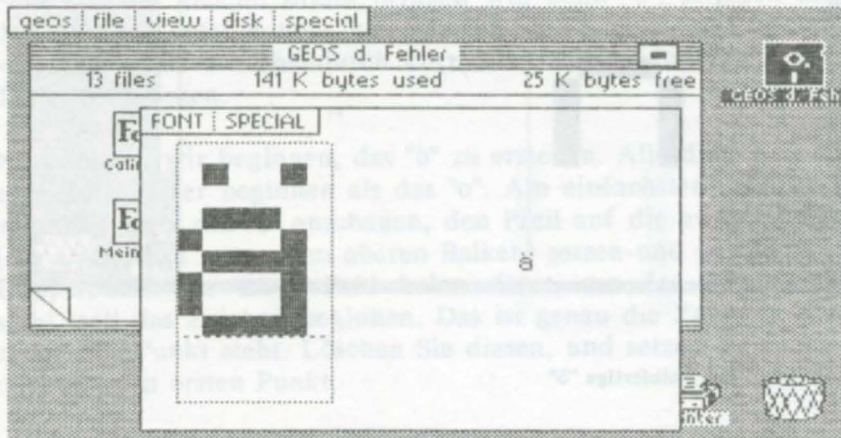


Abb.39: Das fertige "ä"

Nachdem wir nun schon richtig in Schwung sind, legen wir auch gleich noch das kleine "ü" auf den Klammeraffen rechts neben dem "p". Das "ü" ist besonders einfach zu erstellen. Nun haben Sie wirklich eine kleine Pause verdient und sollten deshalb Ihren augenblicklichen Arbeitsstand mit UPDATE sichern. Natürlich können Sie den Font-Editor auch einmal kurz verlassen und die neuen Zeichen in GEOWRITE ausprobieren.

Als nächstes erstellen wir das fehlende "ß" (Abb. 41). Wir legen es auf die PPlus-Taste oben neben der "0". Der vordere Strich bekommt eine Unterlänge von 2 Punkten, das heißt, wir setzen noch zwei Punkte unter den Strich. Unser abgedrucktes "ß" stellt sicherlich eine gute Vorlage dar.

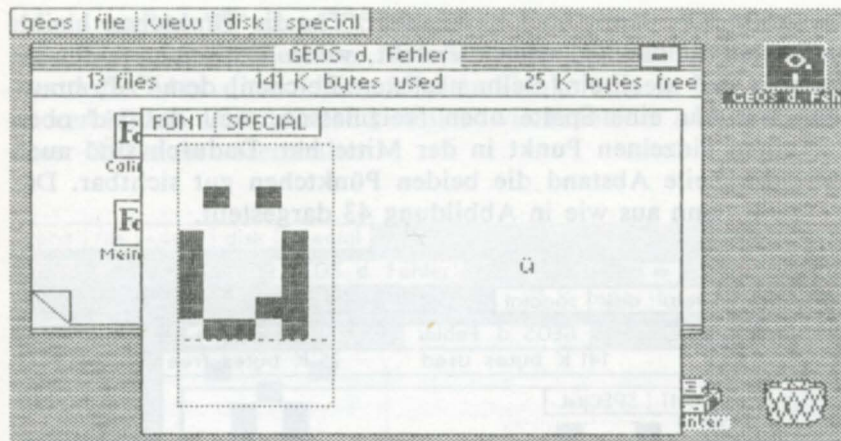


Abb. 40: Das fertig "ü"

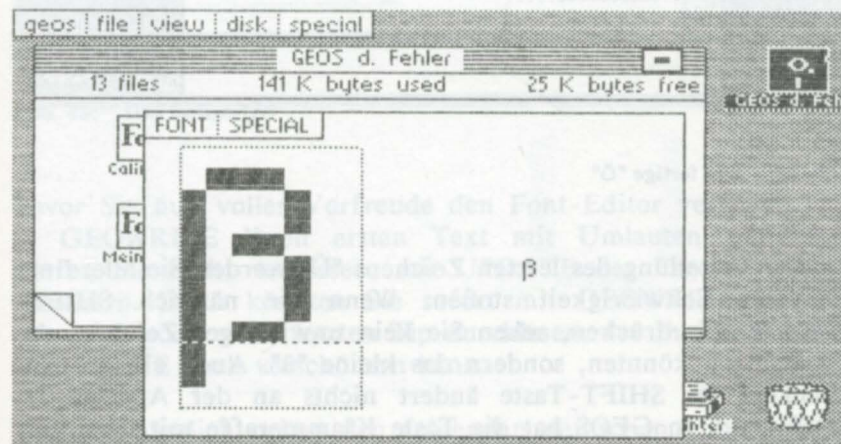


Abb. 41: Ein "ß" in University 12 point

Nun bleiben noch die drei großen Umlaute übrig, beginnen wir also mit dem "Ö". Wir lassen es zwei Punkte tiefer beginnen als

das große "O", damit noch genug Platz für die Pünktchen bleibt. Ansonsten ist das "Ö" schnell erstellt, wenn Sie immer mal wieder zum "O" wechseln. Beim nächsten Zeichen, dem "Ä", brauchen wir nur eine Spalte oben freizulassen, weil das "A" oben nur einen einzelnen Punkt in der Mitte hat. Dadurch sind auch ohne die Zeile Abstand die beiden Pünktchen gut sichtbar. Das "Ä" sieht dann aus wie in Abbildung 43 dargestellt.

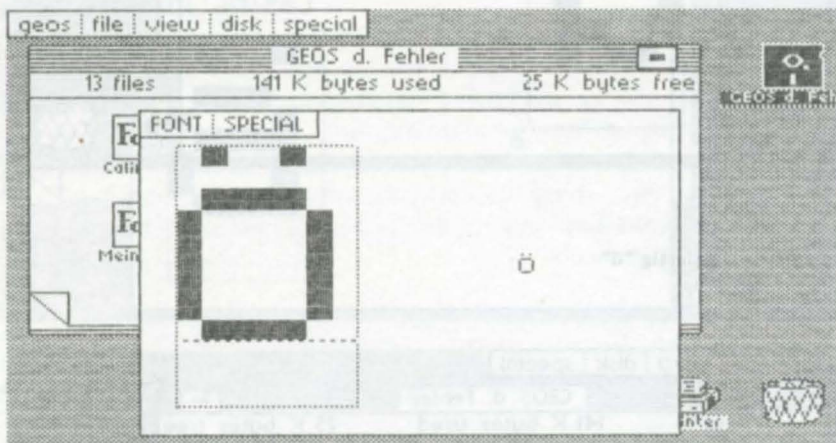


Abb. 42: Das fertige "Ö"

Bei der Erstellung des letzten Zeichens "Ü" werden Sie allerdings auf eine Schwierigkeit stoßen. Wenn Sie nämlich SHIFT+Klammeraffe drücken, sehen Sie kein unwichtiges Zeichen, das Sie ersetzen könnten, sondern das kleine "ü". Auch ein heftiges Drücken der SHIFT-Taste ändert nichts an der Anzeige im EDIT-Feld. In GEOS hat die Taste Klammeraffe mit und ohne SHIFT-Taste denselben Code. Wir haben daher nach einer anderen Möglichkeit gesucht und eine ganz annehmbare gefunden.

Wenn Sie nämlich COMMODORE+Klammeraffe drücken, erhalten Sie genau dasselbe Zeichen, als wenn Sie SHIFT + "7" drücken. Deshalb legen wir das "Ü" auf diese Tastenkombination. Wenn Sie später Texte schreiben, müssen Sie allerdings bei

einem großen "Ü" stets COMMODORE+Klammeraffe drücken, bei allen anderen Zeichen für die große Version SHIFT+Zeichentaste. Aber diese kleine Umständlichkeit ist in GEOS nicht zu vermeiden. Anschließend wollen wir Ihnen in Abbildung 44 unser "Ä" nicht vorenthalten.

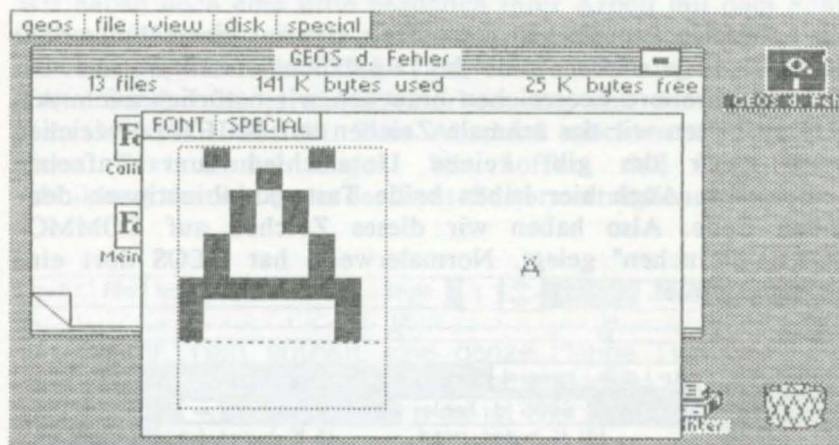


Abb. 43: Das fertige "Ä"

Bevor Sie nun voller Vorfreude den Font-Editor verlassen und in GEOWRITE Ihren ersten Text mit Umlauten schreiben, dürfen Sie nicht vergessen, mit UPDATE das Ergebnis abzuspeichern. Dann könnten Sie sofort mit GEOWRITE loslegen oder sich noch einen letzten Tip durchlesen, der Ihre Arbeit mit GEOWRITE noch verschönern kann.

Uns hat nämlich neben den fehlenden deutschen Umlauten an GEOWRITE noch gestört, daß es unmöglich war, Blocksatz zu erstellen. Dabei ist nämlich der rechte Rand bündig. Zwar muß das nicht unbedingt bei jedem Text geschehen, aber bei einigen Texten hätten wir das schon gerne. Bisher hatten wir uns immer dadurch beholfen, daß wir zwischen einige Worte ein Leerzeichen zusätzlich gesetzt haben. Das hat aber zwei Nachteile.

Erstens ist das Leerzeichen zu breit, so daß der rechte Rand dann trotzdem um mehrere Punkte verschieden war. Zweitens war die Lücke zwischen den Wörtern unterschiedlich groß, so daß diese Methode etwas behelfsmäßig aussah. Mit unserem Font-Editor haben wir nun eine komfortable Möglichkeit geschaffen, die wir Ihnen kurz vorstellen möchten.

Wir haben ein Leerzeichen geschaffen, das nur einen Punkt breit ist. Allerdings haben wir das zusätzlich erstellt, denn das vorherige breitere Leerzeichen brauchen wir natürlich auch. Am liebsten hätten wir das schmale Zeichen auf SHIFT+Leerzeichen gelegt, aber das gibt keinen Unterschied zum einfachen Leerzeichen. Auch hier haben beide Tastenkombinationen denselben Code. Also haben wir dieses Zeichen auf COMMODE+"Sternchen" gelegt. Normalerweise hat GEOS dort eine "Schlangenlinie".

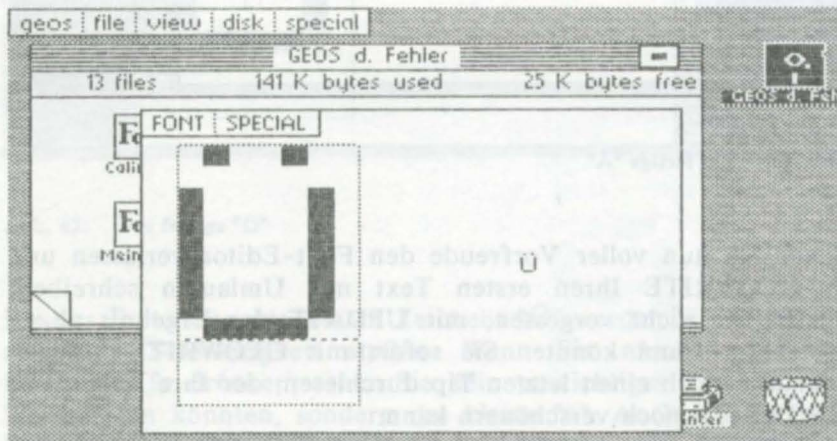


Abb. 44: Das fertige "Ü"

Anschließend konnten wir in allen Texten durch Drücken dieser Tastenkombination den Text dahinter um einen Punkt nach rechts verschieben. Wenn man dieses "Zeichen" zwischen mehreren verschiedenen Wörtern benutzt, bis der rechte Rand

bündig ist, fällt das nicht nur nicht auf, sondern der rechte Rand stimmt dann auch genau auf den Punkt. Wenn Sie so einen Text zusammen mit den Umlauten jemand zeigen, der keinen Font-Editor hat, wird der kaum den Mund wieder zubekommen und Ihnen kaum glauben, was er doch schwarz auf weiß sieht.

Wir haben noch eine Bitte bezüglich Ihrer Arbeit mit dem Font-Editor. Benutzen Sie bitte nicht die Funktionstasten, und drücken Sie nicht die CTRL-Taste zusammen mit einer anderen Taste. Sonst kann es nämlich vorkommen, daß der Font-Editor seine Mitarbeit einstellt. Zum Abschluß möchten wir Ihnen so einen Text einmal zeigen. Wir hoffen, daß Ihnen diese Abbildung das Abtippen des Font-Editors erträglicher macht.

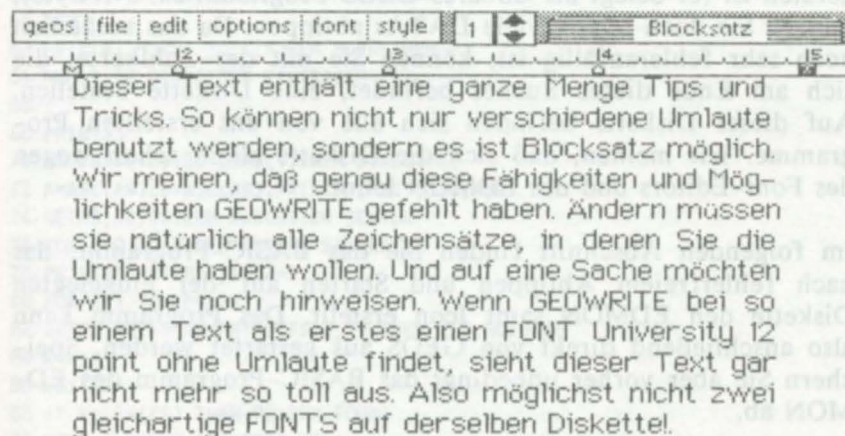


Abb. 45: Das erreicht man mit dem Font-Editor

5.4 Der Maschinensprache-Monitor (EDMON)

Wer GEOS intern kennenlernen will, der braucht dazu Hilfsmittel. Nachdem wir selbst lange Zeit mit einem gewöhnlichen Monitor gearbeitet haben, merkten wir immer mehr, wie nützlich ein vollständig an GEOS angepaßter Monitor sein könnte. Dieser

mußte jederzeit ladbar sein, durfte nichts von im Speicher befindlichen Programmteilen verändern und sollte die üblichen Befehle besitzen.

Genau dieses Hilfsmittel, das wir für unsere eigenen Zwecke ständig benutzen, finden Sie in diesem Kapitel. Obwohl wir das Programm sehr sorgfältig erstellt haben, sollten Sie den EDMON (von EDitor und MONitor) nicht benutzen, wenn Sie unersetzliche Daten im Arbeitsspeicher haben. Die Arbeit mit einem Maschinensprache-Monitor ist immer fehleranfällig, und deshalb sollten Sie vor der Benutzung des EDMON unbedingt Ihre Texte und Grafiken speichern.

Obwohl der EDMON für seine Leistungsfähigkeit recht kurz geraten ist (er belegt als ladbares GEOS-Programm ca. 5 KByte), müssen Sie doch relativ viele DATAs abtippen. Da das zusätzlich noch sehr fehleranfällig ist, können Sie mit der Zahlkarte, die sich am Ende dieses Buches befindet, eine Diskette bestellen. Auf dieser Diskette befinden sich alle von uns erstellten Programme. Wir meinen, daß sich die Diskette allein schon wegen des Font-Editors und des EDMON lohnt.

Im folgenden Abschnitt finden Sie das BASIC-Programm, das nach fehlerfreiem Abtippen und Starten auf der eingelegten Diskette den EDMON samt Icon erstellt. Das Programm kann also anschließend direkt von GEOS aus gestartet werden. Speichern Sie aber vorher unbedingt das BASIC-Programm des EDMON ab.

```

0 REM *****
2 REM * PROGRAMM: EDMON V1.3      *
4 REM * AUTOR   : RUEDIGER KERKLOH *
6 REM * UPDATE  : 04/04/88       *
8 REM *****
10 :
12 OPEN 1,8,15,"I:0":REM FEHLERKANAL
14 RESTORE:PRINT CHR$(147):REM CLR HOME
16 REM * CHECKSUMME BERECHNEN *
18 FOR I=1 TO 4299
20 READ A:S=S+A
22 PRINT CHR$(19);4299-I;CHR$(157);" ";
24 NEXT:RESTORE

```

```

26 IF S<>453247 THEN PRINT "FEHLER IN DATAS !":STOP
28 REM * FILE ABSPEICHERN *
30 F$="EDMON V1.3"
32 OPEN 2,8,2,F$+"U,W":GOSUB 104
34 FOR I=1 TO 4186:READ A
36 PRINT#2,CHR$(A);:NEXT:GOSUB 104
38 CLOSE2
40 REM * INFO-SEKTOR ANLEGEN *
42 A=1:B=0
44 T=A:S=B:PRINT#1,"B-A:0";T;S
46 INPUT#1,N,T$,A,B:IF A=18 THEN A=19:B=0
48 IF N>0 THEN 44
50 OPEN 3,8,3,"#0":PRINT#3
52 PRINT#1,"U1";3;0;T;S:REM SEKTOR LESEN
54 FOR I=1 TO 113:READ B:PRINT#3,CHR$(B);:NEXT
56 FOR I=1 TO 141
58 PRINT#3,CHR$(0);:REM REST LOESCHEN
60 NEXT
62 PRINT#1,"U2";3;0;T;S:REM SCHREIBEN
64 REM * IN DIRECTORY EINTRAGEN *
66 :
68 PRINT#3
70 A$=CHR$(18):B$=CHR$(1):REM DIRECTORY
72 A=ASC(A$):B=ASC(B$):PRINT#1,"U1";3;0;A;B
74 GET#3,A$,B$:REM NAECHSTER SEKTOR
76 FOR I=0 TO 7:REM ANZAHL EINTRAEGE
78 PRINT#1,"B-P";3;I*32+5
80 FOR J=1 TO LEN(F$)
82 GET#3,W$:IF W$<>MID$(F$,J,1) THEN K=0:GOTO 86
84 K=K+1
86 NEXT J
88 IF K=LEN(F$) THEN 92:REM FOUND
90 NEXT I:GOSUB 104:GOTO 72
92 PRINT#1,"B-P";3;I*32+21
94 PRINT#3,CHR$(T)CHR$(S)CHR$(0)CHR$(5);
96 PRINT#3,CHR$(88)CHR$(4)CHR$(4)CHR$(18)CHR$(35)CHR$(20);
98 PRINT#1,"U2";3;0;A;B:REM DIR ZURUECK
100 CLOSE1:PRINT"O.K.":END
102 :
104 INPUT#1,N,T$:IF N THEN PRINT"ERROR: ";T$:STOP
106 RETURN
108 REM
110 DATA 32,156,193,32,202,97,162,29,160,96,142,163,132,140
112 DATA 164,132,162,227,160,96,142,167,132,140,168,132,76
114 DATA 9,98,173,4,133,170,169,128,13,182,132,141,182,132
116 DATA 138,201,32,176,102,201,13,208,3,76,164,98,201,17,208

```

118 DATA 15, 174, 97, 112, 224, 14, 240, 4, 238, 97, 112, 96, 32, 181, 96
 120 DATA 96, 201, 30, 208, 11, 174, 96, 112, 224, 39, 240, 3, 238, 96, 112
 122 DATA 96, 201, 16, 208, 9, 174, 97, 112, 240, 3, 206, 97, 112, 96, 201
 124 DATA 8, 208, 9, 174, 96, 112, 240, 3, 206, 96, 112, 96, 201, 29, 208
 126 DATA 17, 174, 96, 112, 240, 11, 206, 96, 112, 169, 32, 32, 148, 96, 206
 128 DATA 96, 112, 96, 201, 18, 208, 3, 76, 0, 98, 201, 19, 208, 244, 76, 202
 130 DATA 97, 72, 32, 179, 97, 174, 97, 112, 202, 48, 6, 32, 188, 97, 76, 155
 132 DATA 96, 172, 96, 112, 104, 145, 144, 174, 96, 112, 224, 39, 240, 3
 134 DATA 238, 96, 112, 96, 32, 179, 97, 162, 14, 165, 144, 133, 146, 165
 136 DATA 145, 133, 147, 32, 188, 97, 160, 39, 177, 144, 145, 146, 136, 16
 138 DATA 249, 202, 208, 233, 169, 32, 160, 39, 145, 144, 136, 16, 251, 169
 140 DATA 64, 13, 182, 132, 141, 182, 132, 96, 44, 182, 132, 80, 68, 120
 142 DATA 162, 0, 32, 60, 193, 165, 12, 133, 144, 165, 13, 133, 145, 138
 144 DATA 72, 24, 105, 10, 170, 32, 60, 193, 162, 40, 160, 0, 177, 14, 145
 146 DATA 144, 165, 14, 24, 105, 8, 133, 14, 165, 15, 105, 0, 133, 15, 165
 148 DATA 144, 24, 105, 8, 133, 144, 165, 145, 105, 0, 133, 145, 202, 208
 150 DATA 223, 104, 170, 232, 224, 150, 144, 192, 88, 169, 27, 32, 69, 193
 152 DATA 169, 11, 32, 69, 193, 169, 13, 32, 69, 193, 32, 179, 97, 162, 0
 154 DATA 236, 97, 112, 240, 13, 169, 13, 32, 163, 97, 32, 188, 97, 232, 224
 156 DATA 15, 208, 238, 160, 0, 204, 96, 112, 208, 10, 165, 24, 141, 190
 158 DATA 132, 165, 25, 141, 191, 132, 177, 144, 201, 127, 240, 6, 32, 163
 160 DATA 97, 76, 134, 97, 140, 98, 112, 152, 10, 10, 24, 109, 98, 112, 72
 162 DATA 169, 32, 32, 163, 97, 104, 197, 24, 176, 245, 133, 24, 200, 192
 164 DATA 40, 208, 201, 165, 5, 56, 229, 41, 141, 192, 132, 169, 0, 141, 182
 166 DATA 132, 173, 180, 132, 41, 128, 9, 1, 141, 180, 132, 96, 142, 91, 112
 168 DATA 140, 92, 112, 32, 69, 193, 174, 91, 112, 172, 92, 112, 96, 162
 170 DATA 116, 160, 112, 134, 144, 132, 145, 96, 165, 144, 24, 105, 40, 133
 172 DATA 144, 165, 145, 105, 0, 133, 145, 96, 169, 160, 133, 47, 32, 75
 174 DATA 193, 169, 9, 133, 38, 165, 41, 32, 192, 193, 32, 179, 97, 162, 14
 176 DATA 160, 39, 169, 32, 145, 144, 136, 16, 251, 32, 188, 97, 202, 16
 178 DATA 241, 32, 168, 193, 1, 0, 0, 0, 5, 0, 3, 63, 1, 160, 2, 0, 0, 160, 0
 180 DATA 169, 0, 141, 96, 112, 141, 97, 112, 96, 169, 35, 141, 175, 132
 182 DATA 169, 98, 141, 176, 132, 173, 15, 192, 201, 19, 240, 6, 169, 0, 141
 184 DATA 195, 193, 96, 104, 104, 0, 216, 141, 2, 2, 169, 76, 141, 195, 193
 186 DATA 140, 5, 2, 142, 4, 2, 173, 11, 136, 141, 3, 2, 104, 170, 56, 233
 188 DATA 1, 141, 1, 2, 104, 72, 233, 0, 141, 0, 2, 138, 72, 173, 254, 255
 190 DATA 141, 8, 2, 173, 255, 255, 141, 7, 2, 186, 142, 6, 2, 142, 9, 2, 88
 192 DATA 224, 80, 176, 17, 162, 64, 32, 244, 101, 162, 132, 154, 169, 255
 194 DATA 72, 169, 34, 72, 76, 85, 98, 32, 45, 100, 162, 66, 169, 42, 32, 245
 196 DATA 99, 162, 0, 189, 189, 109, 8, 41, 127, 32, 48, 99, 232, 40, 16, 243
 198 DATA 169, 114, 208, 47, 169, 0, 133, 139, 174, 9, 2, 154, 169, 46, 162
 200 DATA 13, 32, 245, 99, 32, 155, 194, 76, 195, 193, 32, 158, 194, 169
 202 DATA 0, 141, 95, 112, 32, 179, 97, 174, 97, 112, 202, 48, 6, 32, 188
 204 DATA 97, 76, 178, 98, 32, 231, 98, 201, 0, 240, 213, 201, 46, 240, 245
 206 DATA 201, 32, 240, 241, 162, 14, 221, 19, 111, 240, 6, 202, 16, 248
 208 DATA 76, 55, 102, 134, 180, 138, 10, 101, 180, 170, 189, 5, 99, 72, 189

210 DATA 4,99,72,96,140,92,112,172,95,112,192,40,240,12,238
 212 DATA 95,112,177,144,201,127,208,5,169,32,44,169,0,172,92
 214 DATA 112,96,44,190,104,44,20,105,44,252,104,44,56,105,44
 216 DATA 82,105,44,131,105,44,183,105,44,209,105,44,215,105
 218 DATA 44,6,106,44,11,107,44,88,106,44,76,108,44,173,104
 220 DATA 44,22,106,142,93,112,140,94,112,201,13,208,27,169
 222 DATA 128,141,182,132,32,227,96,169,0,141,96,112,32,57,96
 224 DATA 44,182,132,80,3,32,227,96,76,88,99,32,32,96,174,93
 226 DATA 112,172,94,112,96,32,231,98,201,0,240,2,56,96,169
 228 DATA 32,24,96,32,101,104,176,5,32,101,104,144,48,32,171
 230 DATA 99,10,10,10,10,141,0,1,32,101,104,144,229,32,171,99
 232 DATA 13,0,1,56,96,32,95,99,32,108,99,144,13,72,32,113,99
 234 DATA 144,6,133,128,104,133,129,96,104,96,32,95,99,32,95
 236 DATA 99,24,96,201,58,8,41,15,40,144,2,105,8,96,32,141,99
 238 DATA 144,27,165,128,133,130,165,129,133,131,32,141,99,162
 240 DATA 2,181,127,72,181,129,149,127,104,149,129,202,208,243
 242 DATA 96,169,63,76,55,102,162,1,181,127,72,181,128,32,230
 244 DATA 99,104,72,74,74,74,74,32,254,99,170,104,41,15,32,254
 246 DATA 99,72,138,32,48,99,104,76,48,99,24,105,246,144,2,105
 248 DATA 6,105,58,96,165,128,141,1,2,165,129,141,0,2,96,133
 250 DATA 138,160,0,32,42,100,177,128,32,230,99,32,51,100,198
 252 DATA 138,208,241,96,32,42,100,169,127,44,169,13,32,48,99
 254 DATA 96,230,128,208,6,230,129,208,2,230,139,96,152,72,32
 256 DATA 45,100,104,162,46,32,245,99,76,39,100,32,113,99,144
 258 DATA 11,162,0,129,128,193,128,240,3,76,214,99,32,51,100
 260 DATA 198,138,96,169,3,133,128,169,2,133,129,169,4,96,72
 262 DATA 32,42,100,104,162,7,133,138,169,24,6,138,42,32,48
 264 DATA 99,202,48,8,152,240,242,32,42,100,208,237,96,133,138
 266 DATA 162,0,165,138,61,46,112,93,26,112,240,7,232,224,20
 268 DATA 144,241,162,14,189,66,112,72,74,74,74,74,170,104,41
 270 DATA 15,201,14,96,160,68,32,62,100,32,219,99,32,42,100
 272 DATA 160,0,177,128,32,138,100,134,138,32,42,100,177,128
 274 DATA 32,230,99,200,198,138,208,243,32,42,100,32,39,100
 276 DATA 200,192,4,144,245,160,0,177,128,162,31,221,34,111
 278 DATA 240,26,202,16,248,162,23,177,128,61,90,111,93,66,111
 280 DATA 240,6,202,16,243,76,174,101,138,24,105,32,170,138
 282 DATA 133,138,10,101,138,170,160,3,189,114,111,32,48,99
 284 DATA 232,136,208,246,177,128,32,138,100,176,222,168,169
 286 DATA 127,32,48,99,192,8,240,37,192,9,176,38,169,36,192
 288 DATA 2,208,5,169,35,32,48,99,169,36,192,0,208,2,169,127
 290 DATA 192,1,208,2,169,65,192,5,240,4,192,6,208,7,169,40
 292 DATA 32,48,99,169,36,32,48,99,192,12,208,11,32,185,101
 294 DATA 138,32,230,99,152,76,230,99,202,240,17,202,240,7,160
 296 DATA 2,177,128,32,230,99,160,1,177,128,32,230,99,160,0
 298 DATA 177,128,32,138,100,201,3,240,39,201,4,240,32,201,5
 300 DATA 240,17,201,6,240,21,201,10,240,23,201,11,240,16,201

302 DATA 8,240,4,96,32,167,101,169,41,76,48,99,32,156,101,169
 304 DATA 89,44,169,88,162,44,76,245,99,162,2,169,63,32,48,99
 306 DATA 202,16,250,96,166,129,160,1,177,128,16,1,202,24,105
 308 DATA 2,144,1,232,24,101,128,144,1,232,168,169,130,56,96
 310 DATA 160,0,177,128,32,138,100,176,51,201,12,240,217,201
 312 DATA 3,138,72,200,177,128,72,202,202,240,4,200,177,128
 314 DATA 170,104,168,104,96,32,45,100,169,63,32,48,99,32,1
 316 DATA 102,162,43,189,70,110,8,41,127,32,48,99,232,40,16
 318 DATA 243,96,162,73,32,244,101,162,49,32,1,102,32,219,99
 320 DATA 76,143,98,162,53,32,244,101,48,246,32,95,99,32,108
 322 DATA 99,144,9,133,133,32,113,99,133,132,176,88,169,63,32
 324 DATA 48,99,76,143,98,32,182,99,144,243,32,38,102,32,95
 326 DATA 99,201,45,240,33,201,43,208,205,165,132,170,165,133
 328 DATA 168,165,130,229,128,133,132,165,131,229,129,133,133
 330 DATA 138,56,229,132,133,132,152,229,133,133,133,165,132
 332 DATA 197,128,165,133,229,129,144,22,165,130,229,128,170
 334 DATA 165,131,229,129,168,138,24,101,132,133,132,152,101
 336 DATA 133,133,133,56,96,160,0,177,128,32,138,100,32,51,100
 338 DATA 202,208,250,96,72,165,130,197,128,165,131,229,129
 340 DATA 144,4,165,139,240,3,76,143,98,104,96,160,0,162,0,176
 342 DATA 18,32,158,102,161,128,145,132,32,51,100,200,208,243
 344 DATA 230,133,24,144,238,32,158,102,177,130,129,132,165
 346 DATA 130,208,2,198,131,198,130,165,132,208,2,198,133,198
 348 DATA 132,24,144,230,32,95,99,32,95,99,133,180,162,0,134
 350 DATA 128,134,129,201,36,240,85,201,37,240,62,201,33,208
 352 DATA 80,32,95,99,201,58,176,43,233,47,144,37,168,6,128
 354 DATA 38,129,165,128,166,129,10,38,129,10,38,129,24,101
 356 DATA 128,144,1,232,133,128,152,24,101,128,133,128,138,101
 358 DATA 129,133,129,76,254,102,105,48,96,32,95,99,201,32,208
 360 DATA 3,32,95,99,74,73,24,208,7,38,128,38,129,76,49,103
 362 DATA 73,24,42,96,32,95,99,170,201,71,176,30,233,47,144
 364 DATA 26,201,10,144,6,201,17,144,18,233,7,162,3,38,128,38
 366 DATA 129,202,16,249,5,128,133,128,76,75,103,138,96,169
 368 DATA 0,162,2,157,10,2,202,16,250,162,15,6,128,38,129,248
 370 DATA 160,2,185,10,2,121,10,2,153,10,2,136,16,244,202,16
 372 DATA 234,216,162,5,32,176,103,9,48,157,0,1,202,16,245,162
 374 DATA 4,189,0,1,201,48,208,3,202,208,246,96,169,16,14,12
 376 DATA 2,46,11,2,46,10,2,42,144,244,96,32,167,194,201,22
 378 DATA 208,248,76,143,98,169,0,141,96,112,169,68,32,68,100
 380 DATA 76,179,100,162,3,180,128,181,132,148,132,149,128,202
 382 DATA 16,245,96,196,132,138,229,133,144,11,24,152,229,134
 384 DATA 138,229,135,176,2,24,96,56,96,152,72,138,72,164,128
 386 DATA 166,129,32,228,103,104,170,104,168,96,152,56,229,128
 388 DATA 72,138,229,129,168,24,104,105,126,176,1,136,200,208
 390 DATA 6,56,233,128,24,200,96,162,93,32,244,101,162,49,32
 392 DATA 1,102,32,219,99,56,96,24,152,101,136,168,138,101,137

394 DATA 170,96,24,200,96,32,115,103,234,189,0,1,32,48,99,202
 396 DATA 16,247,96,32,95,99,162,7,32,95,99,201,32,240,249,74
 398 DATA 73,24,208,30,38,138,202,16,239,165,138,56,96,32,95
 400 DATA 99,201,48,144,15,201,58,144,243,201,65,144,7,41,223
 402 DATA 201,71,144,233,24,96,160,0,32,95,99,144,42,201,32
 404 DATA 240,247,201,34,240,15,32,104,104,32,116,99,144,23
 406 DATA 145,128,32,51,100,208,228,32,95,99,144,14,201,34,240
 408 DATA 219,145,128,32,51,100,208,240,76,55,102,96,32,141
 410 DATA 99,32,124,104,160,119,32,62,100,32,219,99,76,143,98
 412 DATA 162,0,32,1,102,32,45,100,162,46,169,82,32,245,99,32
 414 DATA 42,100,173,0,2,32,230,99,173,1,2,32,230,99,32,42,100
 416 DATA 173,7,2,32,230,99,173,8,2,32,230,99,173,2,2,160,255
 418 DATA 32,109,100,32,98,100,32,19,100,76,143,98,32,182,99
 420 DATA 32,158,102,32,191,103,160,77,32,62,100,32,219,99,169
 422 DATA 8,32,19,100,240,235,32,141,99,144,54,32,8,100,32,141
 424 DATA 99,144,46,165,128,141,8,2,165,129,141,7,2,32,75,104
 426 DATA 144,31,141,2,2,32,98,100,208,10,32,163,99,32,144,99
 428 DATA 144,15,169,8,133,138,32,95,99,32,76,100,208,248,76
 430 DATA 143,98,76,55,102,32,182,99,176,22,169,255,133,130
 432 DATA 133,131,208,14,32,167,194,201,0,240,249,201,22,208
 434 DATA 3,76,143,98,169,5,133,180,198,180,48,234,32,191,103
 436 DATA 32,158,102,32,174,100,32,144,102,240,238,32,141,99
 438 DATA 144,44,32,95,99,32,108,99,144,18,168,32,138,100,152
 440 DATA 160,0,145,128,200,202,240,5,32,108,99,176,245,32,201
 442 DATA 103,32,45,100,32,144,102,32,201,103,162,10,142,96
 444 DATA 112,96,76,55,102,32,182,99,144,248,32,95,99,32,108
 446 DATA 99,144,240,162,0,32,158,102,129,128,32,51,100,24,144
 448 DATA 245,32,63,102,76,178,102,32,141,99,144,3,32,8,100
 450 DATA 174,6,2,154,120,173,7,2,141,21,3,173,8,2,141,20,3
 452 DATA 173,0,2,72,173,1,2,72,173,2,2,72,173,3,2,174,4,2,172
 454 DATA 5,2,64,32,161,194,162,194,160,62,142,156,132,140,155
 456 DATA 132,76,143,98,32,182,99,32,214,103,162,11,169,2,134
 458 DATA 128,133,129,32,124,104,165,128,24,233,11,144,39,141
 460 DATA 10,2,32,214,103,32,158,102,32,191,103,172,10,2,177
 462 DATA 128,217,11,2,208,11,136,16,246,162,85,32,1,102,32
 464 DATA 219,99,32,51,100,24,144,223,76,55,102,32,161,194,32
 466 DATA 180,193,0,1,0,129,0,32,28,108,144,237,32,108,99,144
 468 DATA 232,201,0,240,228,201,15,176,224,141,69,129,169,130
 470 DATA 141,68,129,32,182,99,144,211,166,128,164,129,142,71
 472 DATA 129,140,72,129,142,75,129,140,76,129,166,130,164,131
 474 DATA 142,73,129,140,74,129,162,3,160,110,142,0,129,140
 476 DATA 1,129,162,3,160,21,142,2,129,140,3,129,169,191,141
 478 DATA 4,129,32,180,193,63,0,5,129,85,169,0,133,22,162,129
 480 DATA 133,20,134,21,32,237,193,138,240,8,201,3,208,44,162
 482 DATA 144,208,48,162,99,160,112,134,2,132,3,32,56,194,138
 484 DATA 240,4,201,5,208,22,162,99,160,112,134,2,132,3,162

486 DATA 3,160,110,134,14,132,15,32,89,194,138,240,11,138,9
 488 DATA 48,141,235,110,162,160,32,244,101,76,143,98,32,28
 490 DATA 108,144,33,32,161,194,138,208,231,162,99,160,112,134
 492 DATA 14,132,15,32,11,194,138,208,217,32,108,99,144,109
 494 DATA 201,0,240,105,201,1,240,3,76,55,102,32,141,99,144
 496 DATA 7,165,128,166,129,24,144,26,174,1,132,172,2,132,134
 498 DATA 4,132,5,32,241,107,138,208,67,173,2,128,174,3,128
 500 DATA 133,128,134,129,56,233,2,133,16,133,130,138,233,0
 502 DATA 133,17,133,131,174,1,132,172,2,132,134,4,132,5,160
 504 DATA 0,177,130,72,200,177,130,72,169,255,133,6,133,7,32
 506 DATA 255,193,160,1,104,145,130,136,104,145,130,138,208
 508 DATA 3,76,253,107,76,254,106,162,0,160,132,134,20,132,21
 510 DATA 32,41,194,138,208,239,173,70,129,201,1,208,31,160
 512 DATA 1,177,20,133,4,200,177,20,133,5,32,241,107,138,208
 514 DATA 49,162,8,173,2,128,133,4,240,40,173,3,128,133,5,32
 516 DATA 141,99,144,10,166,128,164,129,134,16,132,17,176,8
 518 DATA 166,16,164,17,134,128,132,129,169,255,133,6,133,7
 520 DATA 32,255,193,138,240,15,76,254,106,162,0,160,128,134
 522 DATA 10,132,11,32,228,193,96,165,16,72,165,17,72,162,167
 524 DATA 32,1,102,32,219,99,104,133,129,104,133,128,162,181
 526 DATA 32,1,102,32,219,99,76,143,98,162,0,32,95,99,144,37
 528 DATA 201,32,240,245,201,34,208,29,32,95,99,201,34,240,15
 530 DATA 157,99,112,232,224,16,208,241,32,95,99,201,34,208
 532 DATA 7,169,0,157,99,112,56,96,24,96,76,55,102,32,95,99
 534 DATA 144,248,201,32,240,247,32,118,99,32,147,99,144,236
 536 DATA 32,95,99,176,9,32,42,100,32,218,100,76,183,109,201
 538 DATA 32,240,238,41,223,141,12,2,32,95,99,41,223,141,11
 540 DATA 2,32,95,99,41,223,141,10,2,169,57,133,138,198,138
 542 DATA 48,189,162,2,165,138,10,101,138,168,189,10,2,217,114
 544 DATA 111,208,236,200,202,16,244,166,138,142,10,2,162,4
 546 DATA 169,32,157,11,2,202,16,250,162,4,32,95,99,144,88,201
 548 DATA 32,240,247,201,35,208,49,32,101,104,176,17,201,39
 550 DATA 208,6,32,95,99,24,144,23,201,36,208,10,32,101,104
 552 DATA 32,116,99,176,11,144,7,201,37,208,0,32,78,104,144
 554 DATA 106,133,130,162,2,134,180,208,65,32,95,99,201,36,240
 556 DATA 249,157,11,2,32,104,104,144,13,32,116,99,144,14,164
 558 DATA 130,132,131,133,130,169,0,157,11,2,202,16,222,169
 560 DATA 12,133,180,198,180,48,81,165,180,10,10,101,180,105
 562 DATA 4,168,162,4,189,11,2,217,10,110,208,233,136,202,16
 564 DATA 244,160,0,185,66,112,41,15,197,180,240,50,200,192
 566 DATA 20,144,242,165,180,208,4,230,180,208,232,201,7,208
 568 DATA 3,24,144,21,201,9,208,23,173,10,2,201,8,176,16,234
 570 DATA 164,130,166,131,32,8,104,133,130,169,12,133,180,144
 572 DATA 198,76,55,102,174,10,2,224,32,144,4,189,58,111,44
 574 DATA 169,255,73,255,57,26,112,29,34,111,32,138,100,197
 576 DATA 180,208,178,202,240,15,202,240,6,160,2,165,131,145

578 DATA 128,160,1,165,130,145,128,160,0,165,138,145,128,32
580 DATA 138,100,138,24,101,128,133,128,144,2,230,129,32,45
582 DATA 100,169,97,32,68,100,32,219,99,169,10,141,96,112,96
584 DATA 13,127,127,127,127,69,68,77,79,78,32,32,86,49,46,51
586 DATA 13,127,127,127,127,40,67,41,32,68,65,84,65,32,66,69
588 DATA 67,75,69,82,32,71,77,66,72,13,127,127,127,127,65,85
590 DATA 84,79,82,58,32,82,85,69,68,73,71,69,82,32,75,69,82
592 DATA 75,76,79,72,141,27,68,85,77,77,89,0,32,32,32,32,32
594 DATA 32,32,32,32,65,32,32,32,0,35,32,32,88,44,0,32,32,89
596 DATA 44,0,41,88,44,0,40,89,44,41,0,40,32,32,32,32,0,32
598 DATA 41,0,0,40,32,32,32,0,0,32,88,44,0,0,32,89,44,0,0,13
600 DATA 127,127,127,127,80,67,127,127,73,82,81,127,127,78
602 DATA 127,86,127,35,127,66,127,68,127,73,127,90,127,67,127
604 DATA 65,67,127,88,82,127,89,82,127,83,208,153,56,32,69
606 DATA 82,82,79,210,32,73,78,160,77,73,83,83,73,78,71,32
608 DATA 43,47,173,76,79,87,32,83,84,65,67,203,73,76,76,69
610 DATA 71,65,76,32,67,79,68,197,13,46,70,79,85,78,68,160
612 DATA 66,82,65,78,67,200,90,69,82,79,32,80,65,71,197,79
614 DATA 85,84,32,79,70,32,77,69,77,79,82,217,73,76,76,69,71
616 DATA 65,76,32,76,73,78,69,32,78,85,77,66,69,210,76,79,65
618 DATA 196,68,73,83,75,32,79,82,32,68,73,82,32,70,85,76,204
620 DATA 73,47,79,32,35,134,160,13,32,76,79,65,68,73,78,71
622 DATA 32,65,84,58,160,13,32,70,73,82,83,84,32,85,78,85,83
624 DATA 69,68,32,65,68,68,82,69,83,83,58,160,114,82,109,77
626 DATA 100,68,102,116,103,120,108,115,97,119,104,144,176
628 DATA 240,48,208,16,80,112,0,24,216,88,184,202,136,232,200
630 DATA 234,72,8,104,40,64,96,170,168,186,138,154,152,56,248
632 DATA 65,129,225,34,1,66,160,162,161,193,2,33,97,76,132
634 DATA 134,98,230,198,224,192,36,32,120,227,227,227,227,227
636 DATA 227,227,227,227,227,227,227,227,227,223,231,231,227,231
638 DATA 231,243,243,247,255,255,66,67,67,66,67,83,66,69,81
640 DATA 66,77,73,66,78,69,66,80,76,66,86,67,66,86,83,66,82
642 DATA 75,67,76,67,67,76,68,67,76,73,67,76,86,68,69,88,68
644 DATA 69,89,73,78,88,73,78,89,78,79,80,80,72,65,80,72,80
646 DATA 80,76,65,80,76,80,82,84,73,82,84,83,84,65,88,84,65
648 DATA 89,84,83,88,84,88,65,84,88,83,84,89,65,83,69,67,83
650 DATA 69,68,69,79,82,83,84,65,83,66,67,82,79,76,79,82,65
652 DATA 76,83,82,76,68,89,76,68,88,76,68,65,67,77,80,65,83
654 DATA 76,65,78,68,65,68,67,74,77,80,83,84,89,83,84,88,82
656 DATA 79,82,73,78,67,68,69,67,67,80,88,67,80,89,66,73,84
658 DATA 74,83,82,83,69,73,108,158,32,162,150,10,138,128,9
660 DATA 25,16,1,17,8,3,4,0,12,28,20,255,223,255,255,223,159
662 DATA 143,159,31,31,31,31,31,15,3,28,159,28,28,28,56,59
664 DATA 57,34,36,17,16,34,34,59,44,37,38,16,31,39,16,57,58
666 DATA 35,0,82,65,67
668 :

```

670 REM INFO-SEKTOR DATEN
672 :
674 DATA 0,255,3,21,191,255,255,255,128,0,1,159,255,157,160
676 DATA 0,65,168,198,93,168,169,65,168,175,93,168,169,65,174
678 DATA 201,93,160,0,65,164,0,65,166,68,89,172,170,89,164
680 DATA 170,89,166,170,89,172,68,89,164,0,89,159,255,129,128
682 DATA 0,25,128,0,1,255,255,255,130,5,0,0,96,64,122,0,96
684 DATA 197,68,77,79,78,32,32,32,32,32,32,32,86,49,46,51,0
686 DATA 0,0,0,210,85,69,68,73,71,69,82,32,203,69,82,75,76
688 DATA 79,72

```

5.4.1 Übersicht über die Befehle

Die Befehle des EDMON entsprechen weitgehend denen der üblichen C64 Monitore. Im folgenden haben wir den Befehlsvorrat und die jeweils notwendige Syntax in Form einer Liste abgedruckt. Hier können Sie vor allem schnell nachsehen, was an Ihrer Eingabe falsch ist, wenn der EDMON sich stur stellt und nur das Fragezeichen ausgibt. Allgemein gilt, daß hinter jeder "Eingabeeinheit" (Befehlszeichen oder Adresse) ein Blank (Leerzeichen) oder ein Komma stehen muß. In den Beispielen haben wir als Startadresse immer \$1000 und als Endadresse (falls erforderlich) immer \$2000 gewählt.

r

Die aktuellen Registerinhalte des Prozessors werden ausgegeben. Sie können anschließend verändert werden. Mit den veränderten Werten kann dann z.B. ein Programm gestartet werden.

g 1000

Startet ein Programm, das ab \$1000 liegt. Damit die Kontrolle anschließend wieder an den Monitor übergeben wird, muß das Programm mit einem 'BRK' enden (nicht 'RTS!').

d 1000 2000

Gibt den Speicherbereich disassembliert aus. Es werden immer fünf Zeilen ausgegeben. Durch Drücken einer Taste wird die Ausgabe fortgesetzt. Eine Endadresse ist nicht zwingend erforderlich. Die ausgegebenen Zeilen können im Hex-Teil verändert

werden. Die Änderungen werden durch RETURN übernommen. Abgebrochen wird die Funktion durch die RUN/STOP-Taste.

a 1000 LDA #\$FE

Der Assemble-Befehl erlaubt eine direkte Assemblierung in den Speicher. Es wird anschließend automatisch die nächste Adresse ausgegeben, um die Eingabe zu erleichtern. Wird nur die Adresse eingegeben, erscheint der Befehl, der zur Zeit an dieser Stelle steht. Er kann dann verändert oder einfach durch RETURN übernommen werden.

f 1000 2000 FF

Der Speicherbereich von \$1000 bis \$2000 wird mit dem Wert \$FF beschrieben.

m 1000 2000

Gibt den Speicherbereich als Hex-Dump aus. Wird keine Endadresse eingegeben, erscheinen nur acht Bytes. Die Speichereinhalte können verändert werden. RETURN übernimmt die veränderte Zeile wieder. Abbruch der Ausgabe erfolgt durch RUN/STOP.

t 1000 2000 4000-

Transportiert den Speicherbereich nach \$4000. Das Minuszeichen bewirkt, daß \$4000 die erste Adresse des neuen Datenblocks ist. In den meisten Fällen wird 't' in dieser Betriebsart eingesetzt.

t 1000 2000 4000+

Transportiert den Speicherbereich nach \$3000. Hierbei ist \$4000 jedoch die letzte Adresse des neuen Datenblocks. Die Daten werden also vor \$4000 abgelegt.

h 1000 2000 "Hallo"

Durchsucht den Speicherbereich nach dem Auftreten des Textes "Hallo". Alle gefundenen Anfangsadressen, ab denen der Text steht, werden dabei ausgegeben.

h 1000 2000 2F 1E

Hier dient die Hunt-Funktion dazu, innerhalb des Speicherbereiches nach Adressen zu suchen, die die Bytes '2F' und '1E' in

dieser Reihenfolge enthalten. Die Anzahl der Bytes in der Eingabezeile ist frei wählbar. Es werden alle gefundenen Adressen ausgegeben.

w 1000 "Hallo Maus"

Der Word-Befehl erlaubt es, einen Text direkt in den Speicher einzugeben. Anschließend wird die nächste, unbenutzte Speicherstelle ausgegeben. In diesem Falle würde also erscheinen: 'w 100A'.

w 1000 F5 E1 A1

In diesem Fall erlaubt 'Word' die direkte Eingabe von Bytes in den Speicher. Die angegebenen Daten werden dabei nacheinander abgelegt. Anschließend erscheint die erste unbenutzte Speicheradresse. In unserem Fall: 'w 1003'.

l "programm"

Lädt ein GEOS-Programm automatisch an die im Info-Sektor angegebene Adresse. So können Sie sich zum Beispiel die unter GEOS vorhandenen Anwendungen ansehen. Es empfiehlt sich aber, vorher den Arbeitsspeicher von \$0400 bis \$5FFF auf die Diskette zu retten (s "swap",0d,0400,5fff). Nach Ihren Studien können Sie nämlich diesen Bereich, der durch die Anwendungen mit Sicherheit zumindest teilweise zerstört worden ist, leicht wieder restaurieren (l "swap").

Sehr nützlich ist bei dem Load-Befehl die anschließende Ausgabe des Speicherbereiches, der von dem geladenen Programm belegt wird. Es wird die erste belegte und die erste, noch nicht benutzte Speicheradresse ausgegeben.

Beispiel: l "Mein Programm"

Ausgabe:

```
LOADING AT: $1200
FIRST UNUSED ADDRESS: $4570
l "programm",0,2000
```

Funktion wie oben. Die Adresse im Info-Sektor wird jedoch ignoriert. Die Daten werden in jedem Fall ab \$2000 abgelegt.

l "programm",1

Lädt ein Programm, das das alte DOS-Format besitzen muß. Die Anfangsadresse muß wie üblich zu Beginn des Datenblocks stehen. Die Daten werden ab der dort eingetragenen Adresse abgelegt. Auch bei dieser Betriebsart wird der Ladebereich angezeigt. Echte GEOS-Files dürfen so nicht geladen werden, da bei ihnen vor den eigentlichen Daten keine Informationen über die Ladeadresse vorhanden sind.

l "programm",1,2000

Lädt ein Programm nach \$2000. Die Daten müssen dabei das gewöhnliche DOS-Format besitzen (Bytes 1 und 2 der Daten müssen die Ladeadresse enthalten). Die Ladeadresse im File-Eintrag wird ignoriert. Anschließend wird der Ladebereich angezeigt.

s "programm",ft,1000,2000

Die Daten im Bereich von \$1000 (incl.) bis \$2000 (excl.) des Arbeitsspeichers werden auf Diskette gespeichert. 'ft' steht dabei für den gewünschten Filetyp, der hexadezimal, jedoch ohne das Dollarzeichen angegeben werden muß (01-0E). Wenn z.B. ein Hilfsprogramm erstellt werden soll, muß für "ft" der Wert "05" eingetragen werden. Bei Anwendungen steht hier "06". Weitere Filetypen finden Sie im Kapitel 6.

Beim Speichervorgang wird automatisch ein Infosektor erstellt, in den die Start- und Endadresse eingetragen wird. Zusätzlich wird die Startadresse als Einsprungadresse eingetragen. Die so erstellten Files erhalten ein Streifen-Icon und können (wenn der Filetyp es zuläßt) durch einen Doppelklick vom DESK TOP aus gestartet werden.

x

Mit diesem Befehl können Sie den EDMON wieder verlassen. Dazu müssen Sie die Diskette einlegen, von der Sie den EDMON auch geladen haben. Beim Ladevorgang sind nämlich hierauf ganz bestimmte Daten gerettet worden (in einem Swap File FILE), die nun wieder geladen werden müssen. Findet das Betriebssystem diese Daten nicht, da die falsche Diskette eingelegt ist, gibt es einen Rechnerabsturz.

5.4.2 Tips zur Arbeit mit dem EDMON

Der EDMON besitzt den Filetyp fünf, d.h. er ist ein Hilfsprogramm. Das hat den großen Vorteil, daß er nicht nur unter DESK TOP, sondern auch bei der Arbeit mit beliebigen Anwendungen geladen werden kann. Damit Sie schon einmal einen kleinen Vorgeschmack auf die Möglichkeiten erhalten, die der EDMON bietet, wollen wir Ihnen in diesem Kapitel einige Beispiele zeigen. Wenn Sie den EDMON geladen haben, sieht Ihr Bildschirm etwa folgendermaßen aus:

```
B*
PC  IRQ  NU # B D I Z C AC XR YR SP
.R 649D E2DC 1 0 1 1 0 0 0 0 07 F7 60 F5
```



Abb. 46: Die Einschaltmeldung des Edmon

Dabei zeigt der EDMON Ihnen in der Einschaltmeldung den Copyright-Hinweis und die Register des Prozessors. Natürlich überschreibt der EDMON einen Teil des aktuellen Speichers. Dieser wird aber vorher auf die Diskette gerettet, so daß Sie keinen Datenverlust zu befürchten haben. Trotzdem sollten Sie den EDMON nicht laden, wenn Sie etwas Unersetzliches im Speicher haben (beispielsweise einen noch nicht abgespeicherten Text). Die Arbeit mit einem Maschinensprache-Monitor kann nämlich immer einmal schiefgehen.

Der Speicher, den der EDMON belegt, ist sehr geschickt gewählt. Er beginnt nämlich bei \$6000. Damit liegt er genau im zweiten, unsichtbaren Bildschirm von GEOS. Dadurch haben Sie kein einziges Byte des Programms überschrieben und können alles anschauen, disassemblieren und gegebenenfalls wieder abspeichern. Sie haben also völligen Zugriff auf GEOS.

Schauen wir uns einmal einen Teil des Speichers an. Aus der Übersicht über die Speicherbelegung unter GEOS wissen Sie wahrscheinlich schon, daß von \$8000 - \$83FF 4 Puffer für die Zusammenarbeit mit der Floppy liegen. Schauen wir uns einmal den 2. Puffer ab \$8100 an, indem wir

M 8100 8140 (RETURN)

eingeben. Anschließend sehen Sie etwa folgendes Bild:

```

PC  IRQ  NU#BDIZCACXRVRSP
R 649D E2DC 1 0 1 1 0 0 0 0 07 F7 60 F5
.m 8100 8140
.M 8100 00 FF 03 15 BF FF FF FF
.M 8108 80 00 01 9F FF 9D A0 00
.M 8110 41 A8 C6 5D A8 A9 41 A8
.M 8118 AF 5D A8 A9 41 AE C9 5D
.M 8120 A0 00 41 A4 00 41 A6 44
.M 8128 59 AC AA 59 A4 AA 59 A6
.M 8130 AA 59 AC 44 59 A4 00 59
.M 8138 9F FF 81 80 00 19 80 00
.M 8140 01 FF FF FF 82 00 00 FE
    
```



Abb. 47: M 8100 8140

Wenn Sie schon häufiger mit einem Maschinensprache-Monitor gearbeitet haben, fällt Ihnen vielleicht auf, daß der EDMON zwar ein sogenanntes HEX-Dump ausgibt (das sind jeweils 8 Inhalte von Speicherstellen in einer Zeile), aber nicht die zugehö-

rigen ASCII-Zeichen. Das liegt daran, daß die Ausgabe des EDMON in hochaufgelöster Grafik erfolgt und daher schon recht langsam ist. Würden jetzt auch noch die zugehörigen ASCII-Zeichen ausgegeben, so würde die Ausgabe noch weiter verlangsamt. Daher haben wir darauf verzichtet.

Wenn Sie eine Weile mit GEOS arbeiten und sich schon mit dem Aufbau des INFO-Sektors beschäftigt haben, so fällt Ihnen vielleicht auf, daß die ausgegebenen Bytes eine große Ähnlichkeit mit so einem INFO-Sektor haben. Tatsächlich hat GEOS den INFO-Sektor meist im Puffer ab \$8100 abgelegt, und Sie können ihn so problemlos anschauen. Die mit dem M-Befehl ausgegebenen Bytes von \$8100 - \$8140 stellen im wesentlichen das Aussehen des zugehörigen Icons dar.

Aber der EDMON kann noch viel mehr, als nur Speicherbereiche mit ihrem Inhalt ausgeben. Wir wollen uns nun einmal einen Teil des GEOS-Kernals anschauen, und zwar die sogenannte "Jobschleife". Diese ist sehr interessant, weil von ihr alle Jobs (also die vielen kleinen Arbeitsgänge, die GEOS ständig erledigt), verwaltet werden. Die Jobschleife liegt in der GEOS Version 1.2 von \$C2C8 bis \$C2E3. Bei anderen Versionen kann man die Anfangsadresse der Schleife aus dem Inhalt der Speicherstellen \$C1C4/\$C1C5 auslesen. Die Adresse ist nämlich Bestandteil der Kernalsprungtabelle (s. Kapitel 6.4). Mit dem EDMON betrachtet sieht die "Main Loop" folgendermaßen aus, wie in Abbildung 48 dargestellt.

Aus Platzgründen hat man in der Jobschleife der GEOS-Version 1.3 - sie liegt hier von \$C0DF bis \$C0F7 - mehrere Unterprogramme zusammengefaßt und läßt sie durch nur einen Aufruf abarbeiten (Uhrzeit- und Alarmroutinen). Aus diesem Grund findet man hier nur vier Unterprogrammaufrufe.

Der EDMON ist auch deswegen so leistungsfähig, weil er vollkommen kompatibel zu GEOS ist. Das bedeutet, daß er GEOS möglichst wenig bei der Arbeit stört. Zum Beispiel gibt er die Zeichen genau wie GEOS in der hochaufgelösten Grafik aus. Grundsätzlich hätte man den EDMON natürlich auch so programmieren können, daß er erst auf Textmodus umschaltet (wie

im "normalen" C64) und anschließend alles als reinen Text ausgibt. Da aber normalerweise der Textbildschirm bei \$0400 liegt und dort schon die GEOS-Anwendungen beginnen, hätte man sich den ersten Teil dieser Anwendungen nicht anschauen können.

```
.d c2c8 c2e1
.d C2C8 20 8F E2 JSR 3E28F
.d C2CB 20 87 9E JSR 39EB7
.d C2CE 20 87 9F JSR 39F87
.d C2D1 20 30 F9 JSR 3F930
.d C2D4 20 ED F9 JSR 3F9ED
.d C2D7 AD 9B 84 LDA 3849B
.d C2DA AE 9C 84 LDX 3849C
.d C2DD 20 D8 C1 JSR 3C1D8
.d C2E0 58 CLI
.d C2E1 4C C8 C2 JMP 3C2C8
```



Abb. 48: Die Jobschleife mit dem EDMON disassembliert

Als Beispiel wollen wir Ihnen einmal zeigen, wie GEOWRITE gestartet wird. Die Startadresse ist \$0400. Dazu verlassen wir den EDMON, laden GEOWRITE und anschließend den EDMON erneut. (Das geht in der GEOWRITE-Menüzeile unter dem Oberbegriff "geos"). Nun können wir uns GEOWRITE in Ruhe anschauen, indem wir

```
D 0400 0427 (RETURN)
```

eingeben. Danach sehen wir die ersten Befehle, mit denen GEOWRITE initialisiert wird. Sie können sich sicherlich vorstellen, welch ein mächtiges Werkzeug der EDMON ist, wenn sie daran denken, daß Sie beliebige Speicherbereiche mit ihm ab-

speichern können. Sie können also GEOWRITE, ebenso aber auch das GEOS KERNAL speichern, und zwar im normalen DOS- oder im GEOS-Format.

```
.D 0400 4C 16 27 JMP 32716
.D 0403 4C 38 34 JMP 33438
.D 0406 20 B0 C1 JSR 3C180
.D 0409 20 FB 20 JSR 320FB
.D 040C 20 27 04 JSR 30427
.D 040F 20 9F 04 JSR 3049F
.D 0412 4C 9F 05 JMP 3059F
.D 0415 20 B0 C1 JSR 3C180
.D 0418 20 FB 20 JSR 320FB
.D 041B 20 27 04 JSR 30427
.D 041E 20 07 05 JSR 30507
.D 0421 20 09 05 JSR 30509
.D 0424 4C 2C C2 JMP 3C22C
.D 0427 20 50 06 JSR 30650
```

Abb. 49: Die Initialisierung von GEOWRITE

Zum Abschluß möchten wir Ihnen noch einen kleinen Tip geben. Mit dem EDMON ist es sehr einfach möglich, eigene Programme in GEOS zu schreiben. Diese können Sie dann z.B. als Anwendung (Filetyp = 6) oder als Hilfsprogramm (Filetyp=5) abspeichern und nach dem Verlassen von GEOS direkt durch einen Doppelklick starten und testen. So können Sie ein Programm aus kleinen Modulen entwickeln, einzeln austesten und jeweils speichern. Wenn Sie Schwierigkeiten mit dem Initialisieren und dem Ende Ihrer Module haben, so können Sie sich beispielsweise beim Hardcopy Programm anschauen, wie wir das gemacht haben.

5.5 Der Einzelschrittssimulator

Für unsere Nachforschungen im GEOS-Betriebssystem haben wir einen Einzelschrittssimulator entwickelt, der es ermöglicht, den Mikroprozessor bei seiner Arbeit genau zu verfolgen. Dazu wird nach jedem abgearbeiteten Befehl in eine Routine ver-

zweigt, die die derzeitigen Registerinhalte des Prozessors auf dem unteren Bildschirmrand ausgibt und erst wieder zum Hauptprogramm zurückkehrt, wenn eine bestimmte Taste betätigt wird. Gleichzeitig können dabei noch die aktuellen Inhalte vier beliebiger Speicherstellen angezeigt werden.

Wir haben dieses Hilfsmittel hauptsächlich zur Lokalisierung wichtiger Stellen des Kernals eingesetzt (Sprungtabelle, Haupt-Jobschleife, Seriennummer etc.). So konnten wir auch den Fehler ausfindig machen, der die Echtzeituhr in GEOS V1.2 stets auf 60 Hz setzt und damit ihre Ungenauigkeit verursacht. Natürlich können Sie das Programm auch benutzen, um Fehler in eigenen Programmen aufzuspüren, die Sie unter GEOS (z.B. mit dem EDMON) geschrieben haben.

Das abgedruckte Listing des SST (Single Step Simulator, kurz 'Stepper') kann mit jedem gewöhnlichen Assembler übersetzt und anschließend mit einem Monitorprogramm auf Diskette gebracht werden. Im Anschluß an das Listing drucken wir auch wieder ein BASIC-Listing ab, das den Stepper in Form von Daten enthält. Die Installation erfolgt dabei nach dem gleichen Prinzip wie bei den vorangegangenen Programmen.

Die abgedruckte BASIC-Version des Steppers funktioniert nur in Verbindung mit GEOS V1.3, wie es zum Zeitpunkt der Drucklegung dieses Buches ausgeliefert wurde. In GEOS V1.2 liegen die Speeder-Routinen, die der Stepper beim Laden überschreibt, in einem völlig anderen Bereich. Eine relokatable Programmierung, die den Stepper versionsunabhängig machen würde, ergäbe jedoch einen zu langen Code. Sollten Sie das Programm unter einer anderen GEOS-Version benutzen wollen, genügt es, die Startadresse im Source auf den jeweiligen Beginn des Speeders zu setzen (in GEOS V1.2: "*" = \$C5DE") und das Programm neu zu assemblieren. Bei der nun folgenden Beschreibung des Konvertierungsvorgangs müssen dann natürlich auch die hierbei auftretenden Adressen geändert werden.

Nach der Assemblierung speichern Sie bitte den Stepper von einem Monitorprogramm aus ab und wandeln ihn mittels des Filemasters aus dem Kapitel 3 in ein lauffähiges Accessory um

(Hilfsprogramm=Filetyp 5). Beachten Sie bei der Konvertierung bitte die folgenden Punkte:

1. Erstellen Sie zuerst ein passendes Icon (Sprite) für den Stepper (Befehl 'se' im Filemaster). Sie können auch ein schon vorhandenes Icon einlesen und dieses modifizieren (Befehl 'sl').
2. Lesen Sie nun den Stepper ein (Befehl 'fe').
3. Geben Sie mit 'fd' das aktuelle Datum ein.
4. Mittels 'geos' den Stepper konvertieren. Beantworten Sie die Frage nach dem Filetyp mit einer Fünf (=Hilfsprogramm).
5. Jetzt den Info-Bildschirm einlesen und die folgenden dezimalen Adressen eingeben (Werte in Klammern für GEOS VI.2):

STARTADRESSE	38446	(50652)
ENDADRESSE	38447	(50653)
INITIALISIERUNG	38448	(50654)

Die Startadresse ist um zwei Bytes kleiner als der Einsprung in die Initialisierung, da die ersten beiden Bytes im alten File-Format (und in diesem haben wir das Programm ja auf der Diskette vorliegen) noch die Ladeadresse beinhalten. Diese wird aber für den Programmablauf nicht benötigt. Dadurch, daß wir dem GEOS-Kernal die Programmlänge '1 byte' vorgaukeln (Endadresse minus Startadresse), kann unser Programm nach der Installation nicht durch ein langes Swap File wieder überschrieben werden. (Null Byte Programmlänge ist leider verboten!)

Bevor Sie nun durch 'ende' den Konvertierungsvorgang abschließen, können Sie noch die restlichen Eintragungen im INFO-Fenster vornehmen (Name des Autors, Datum, etc.). Anschließend befindet sich auf der Diskette der Stepper als echtes GEOS-Hilfsprogramm, das auch aus Anwendungen heraus geladen und gestartet werden kann. Beachten Sie aber, daß nun keine zweite Floppy mehr angemeldet werden kann, da diese

nun anstelle der Turbo-Disk-Routinen unser Stepper-Programm ins RAM geladen bekäme. Leider verträgt sich der Stepper aufgrund seiner Länge auch nicht mit unseren Programmen "Hard-copy" und der Uhr. Aufgrund des fehlenden Speicherplatzes mußten wir jedoch diesen Kompromiß eingehen. Hier nun das dokumentierte Assembler-Listing des Steppers.

5.5.1 Listing des Steppers

```

*****
;* PROGRAMM: SST V1.3      *
;* AUTOR   : RUEDIGER KERKLOH *
;* UPDATE  : 03/30/88      *
*****
*   = $9630 ;GEOS V1.3 (!)
;*   = $C5DE ;GEOS V1.2 (!)
IRQV = $FFFE ;IRQ-EINSPRUNG
VIA1  = $DC00 ;VIA
TALO  = VIA1+4 ;TIMER LO
TAHI  = VIA1+5 ; HI
ICRA  = VIA1+13 ;IRQ CONTROL
CRA   = VIA1+14 ;TIMER CONTROL
VIC   = $D000 ;VIC ANFANG
;*   INITIALISIERUNG      *
;* SINGLESTEP: CIA 1 A   *
START SEI
LDA $01
PHA
LDA #$35 ;$E000 -> RAM
STA $01 ;I/O EINSCHALTEN
LDA #$2C ;ALARMAUSWERTUNG
STA $C0E8 ;SPERREN
LDA #$EA ;STELLE IM KERNAL
STA $9067 ;DIE ICR AUSLIEST
STA $9068
STA $9069
LDA #$00 ;ZEILEN-IRQ SPERREN
STA VIC+26
LDX IRQV ;ORIGINAL IRQ MERKEN
LDY IRQV+1
STX OIRQ
STY OIRQ+1
LDX #<IRQ ;NEUEN IRQ EINTRAGEN
LDY #>IRQ

```

```

STX IRQV
STY IRQV+1
LDA #$00
STA CRA      ;TIMER A: STOP
LDA ICRA     ;CLR
LDA #$7F
STA ICRA     ;CLR MASKE
LDA #$81     ;SET MASKE: TIMER A
STA ICRA
LDX #27      ;ONE STEP TIME
LDY #0       ;27 CYCLES
STX TALO
STY TAHI
LDA #$09     ;START ONE SHOT
STA CRA
PLA
STA $01
LDX #$3E     ;LOAD SWAP FILE
LDY #$C2     ;AND RETURN TO
STX $849B   ;MAIN LOOP
STY $849C
RTS
;*** NEUER IRQ-EINSPRUNG ***
IRQ STA AKKU  ;RETEN
LDA $01
STA MEM
LDA #$35
STA $01
WAIT LDA #$FD ;TESTET SHIFT KEY
STA $DC00
LDA $DC01
AND #$80     ;GEDRUECKT ?
EOR #$80     ;(UMDREHEN)
ORA FLAG     ;DIREKT V. GEOS-IRQ ?
BEQ CONTJ    ;BEIDES NICHT :SST
JMP CONTM    ;SONST VERKUERZT
CONTJ LDA #$7F ;C= TASTE GEDRUECKT ?
STA $DC00
LDA $DC01
AND #$20
BEQ CONTK    ;JA -> SST
LDA #200     ;SONST: MAX WAIT
STA RPEAT
BNE WAIT     ;UNBEDINGTER SPRUNG
CONTK LDA RPEAT
BEQ CONTL    ;MIN WAIT!

```

```

NEXT SEC                                000 0100
SBC #5 ;WARTEZEIT ERNIEDR.             0100 000 001 010
STA RPEAT                                X,01+0AT 010
CONTL STX XREG ;X/Y REGISTER RETTEN    0100 000
STY YREG                                0100 000
TSX ;STACK KORREKTUR                   0100 000
INX ;STACK KORREKTUR                   0100 000
INX ;STACK KORREKTUR                   0100 000
INX ;STACK KORREKTUR                   0100 000
STX STAC                                0100 000
PLA ;STATUS-REGISTER                   0100 000
STA STAT                                0100 000
PLA ;P-COUNTER                         0100 000
STA PC+1 ;LO                            0100 000
PLA ;P-COUNTER                         0100 000
PHA ;P-COUNTER                         0100 000
STA PC ;HI                              0100 000
LDA PC+1 ;LO                            0100 000
PHA ;ZURUECK                           0100 000
LDA STAT                                0100 000
PHA ;ZURUECK                           0100 000
LDX #0
;* REGISTER IN B-CODE UMWANDELN *
LDA PC
JSR ASC ;PC 1+2.STELLE
STA TAB
STY TAB+1
LDY #0
NEXT LDA REG,Y ;PC LO
STY PC ;(FREI)
JSR ASC
STA TAB+2,X
INX
TYA
STA TAB+2,X
LDY PC
INX
INX
INY
CPY #6
BNE NEXT
;STATUS REGISTER SETZEN
LDX #7
NEXTS LSR STAT
BCS SET
LDA #46 ;PUNKT
    
```

```

        .BYTE $2C
SET    LDA #30      ;PFEIL
        STA TAB+17,X
        DEX
        BPL NEXTS
        LDA $8507   ;AKTUELLE MAUS SPEED
        JSR ASC
        STA TAB+26
        STY TAB+27
        LDA $84B6   ;STATUS MAUS
        JSR ASC
        STA TAB+29
        STY TAB+30
        LDA $8506   ;JOYSTICKWERT
        JSR ASC
        STA TAB+26
        STY TAB+27
        LDA $877D   ;ANZAHL PROGRAMMJOB
        JSR ASC
        STA TAB+29
        STY TAB+30
        ;*** AUSGABEROUTINE ***
        LDX #5      ;$07-$02 -> STACK
NEXTA  LDA $02,X
        PHA        ;$02/$03 CAROM
        DEX        ;$04/$05 BITMAP
        BPL NEXTA  ;$06/$07 TABELLE
        LDA #$8D
        STA $05    ;BIT MAP
        LDA #$80   ;OBERE ZEILE
        STA $04
        LDA #<TAB1 ;OBERE TABELLE
        STA $06
        LDA #>TAB1
        STA $07
        JSR OUT
        LDA #$BE   ;UNTERE ZEILE
        STA $05
        LDA #$F0
        STA $04
        LDA #<TAB ;UNTERE TABELLE
        STA $06
        LDA #>TAB
        STA $07
        JSR OUT
        LDX #0     ;STACK -> $02-$07

```

```

NEXTB PLA
      STA $02,X
      INX
      CPX #6
      BNE NEXTB
      LDX RPEAT ;WARTEN
      BEQ LEAVE ;MIN WAIT !
      LDY #$00
WAIT1 DEY
      BNE WAIT1
      DEX
      BNE WAIT1
LEAVE LDX XREG
      LDY YREG
CONTM LDA ICRA ;CLR
      LDA #$09 ;START
      STA CRA
      LDA #$01 ;GEOS IRQ? ; 2
      AND VIC+25 ; 4
      STA FLAG ;MERKEN ; 4
      BNE GIRQ ; 2
      LDA MEM ; 4
      STA $01 ; 3
      LDA AKKU ; 4
      RTI ; 6
      ;---
      ;29 CYCL.
      ;*** ZUM GEOS IRQ-PROGRAMM ***
GIRQ LDA MEM
      STA $01
      LDA AKKU
      JMP (OIRQ) ;ZUM GEOS IRQ
ASC ;*** UMWANDLUNG HEX->B.CODE ***
      PHA ;RETEN
      JSR CODE ;LINKS : AKKU
      TAY ;RECHTS: Y-REG
      PLA
      LSR A
      LSR A
      LSR A
      LSR A
      JSR CODE
      RTS
CODE AND #$0F
      CLD
      SEC
    
```

```

SBC #$0A
BCS HEX
ADC #57
HEX CLC
ADC #1
RTS
;*** ZEICHENAUSGABE ***
OUT LDA #$33 ;CHARACTER-GENERATOR
STA $01 ;EINBLENDEN
LDY #30 ;ZEICHENZAehler
NEXT2 LDA #0 ;CLR HI
STA $03 ;(ADRESSE IM CAR ROM)
LDA ($06),Y ;B-CODE
ASL A ;MAL 8
ROL $03
ASL A
ROL $03
ASL A
ROL $03
STA $02 ;LO
LDA #$D0 ;HI
CLC
ADC $03
STA $03
TYA
PHA
LDY #7
NEXT1 LDA ($02),Y
STA ($04),Y
DEY
BPL NEXT1
PLA
TAY
SEC ;NAECHSTES ZEICHEN
LDA $04
SBC #8
STA $04
LDA $05
SBC #0
STA $05
DEY ;ALLE ZEICHEN ?
BPL NEXT2
LDA #$35
STA $01
RTS
; ARBEITSSPEICHER

```

```

FLAG * = * + 1 ;BEI GEOS-IRQ
MEM * = * + 1
RPEAT * = * + 1
PC * = * + 1 ;HI
REG * = * + 1 ;LO
AKKU * = * + 1
XREG * = * + 1
YREG * = * + 1
STAC * = * + 1
STAT * = * + 1
OIRQ * = * + 2
;ANZEIGE BEREICH
TAB1 .BYTE 32,16, 3,32,32, 1, 3,32
      .BYTE 24,18,32,25,18,32,19,16
      .BYTE 32,14,22,35, 2, 4, 9,26
      .BYTE 3,32,32,32,32,32,32
TAB .BYTE 32,32,32,32,32,32,32,32
     .BYTE 32,32,32,32,32,32,32,32
     .BYTE 32,32,32,32,32,32,32,32
     .BYTE 32,32,32,32,32,32,32,32
     .END
    
```

Hier nun das BASIC-Listing des Steppers. Wie oben schon erwähnt, funktioniert dieses Programm nur in Verbindung mit GEOS V1.3. Die Installation des Steppers erfolgt wie bei dem Hardcopy-Programm nach der Eingabe von RUN selbständig. Natürlich ist auch diesmal eine Prüfsummenfunktion für die DATAs installiert. Falls die Installation geklappt hat, finden Sie anschließend den Stepper mit einem schönen Icon versehen auf dem DESK TOP wieder.

```

0 REM *****
2 REM * PROGRAMM: SST V1.3 *
4 REM * AUTOR : RUEDIGER KERKLOH *
6 REM * UPDATE : 03/30/88 *
8 REM *****
10 :
12 OPEN 1,8,15,"I:0":REM FEHLERKANAL
14 RESTORE:PRINT CHR$(147):REM CLR HOME
16 REM * CHECKSUMME BERECHNEN *
18 FOR I=1 TO 696
20 READ A:S=S+A
22 PRINT CHR$(19);696-I;CHR$(157);" ";
24 NEXT:RESTORE
26 IF S<>74062 THEN PRINT "FEHLER IN DATAS I":STOP
    
```

```

28 REM * FILE ABSPEICHERN *
30 F$="SST V1.3"
32 OPEN 2,8,2,F$+"U,W":GOSUB 104
34 FOR I=1 TO 582:READ A
36 PRINT#2,CHR$(A);:NEXT:GOSUB 104
38 CLOSE2
40 REM * INFO-SEKTOR ANLEGEN *
42 A=1:B=0
44 T=A:S=B:PRINT#1,"B-A:0";T;S
46 INPUT#1,N,T$,A,B:IF A=18 THEN A=19:B=0
48 IF N>0 THEN 44
50 OPEN 3,8,3,"#0":PRINT#3
52 PRINT#1,"U1";3;0;T;S:REM SEKTOR LESEN
54 FORI=583 TO 696:READ B:PRINT#3,CHR$(B);:NEXT
56 FORI=1 TO 142
58 PRINT#3,CHR$(0);:REM REST LOESCHEN
60 NEXT
62 PRINT#1,"U2";3;0;T;S:REM SCHREIBEN
64 REM * IN DIRECTORY EINTRAGEN *
66 :
68 PRINT#3
70 A$=CHR$(18):B$=CHR$(1):REM DIRECTORY
72 A=ASC(A$):B=ASC(B$):PRINT#1,"U1";3;0;A;B
74 GET#3,A$,B$:REM NAECHSTER SEKTOR
76 FOR I=0 TO 7:REM ANZAHL EINTRAEGE
78 PRINT#1,"B-P";3;I*32+5
80 FOR J=1 TO LEN(F$)
82 GET#3,W$:IF W$<>MID$(F$,J,1) THEN K=0:GOTO 86
84 K=K+1
86 NEXT J
88 IF K=LEN(F$) THEN 92:REM FOUND
90 NEXT I:GOSUB 104:GOTO 72
92 PRINT#1,"B-P";3;I*32+21
94 PRINT#3,CHR$(T)CHR$(S)CHR$(0)CHR$(5);
96 PRINT#3,CHR$(88)CHR$(4)CHR$(2)CHR$(13)CHR$(15)CHR$(4);
98 PRINT#1,"U2";3;0;A;B:REM DIR ZURUECK
100 CLOSE1:PRINT"O.K.":END
102 :
104 INPUT#1,N,T$:IF N THEN PRINT"ERROR: ";T$:STOP
106 RETURN
108 REM
110 DATA 120,165,1,72,169,53,133,1,169,44,141,232
112 DATA 192,169,234,141,103,144,141,104,144,141
114 DATA 105,144,169,0,141,26,208,174,254,255,172
116 DATA 255,255,142,54,152,140,55,152,162,146
118 DATA 160,150,142,254,255,140,255,255,169,0

```



```

212 REM * INFO-SEKTOR *
214 DATA 0,255,3,21,191,255,255,128,0,1,143
216 DATA 255,241,208,0,9,84,113,235,84,73,42,212
218 DATA 73,42,148,73,235,148,73,41,151,113,41
220 DATA 144,0,9,143,255,241,128,0,1,185,207,249
222 DATA 169,72,9,185,203,233,128,8,137,144,136
224 DATA 9,185,207,249,128,0,1,255,255,255,130
226 DATA 5,0,48,150,49,150,48,150,83,83,84,32,86
228 DATA 49,46,51,32,85,83,65,0,32,65,13,32,32
230 DATA 32,32,210,85,69,68,73,71,69,82,32,203
232 DATA 69,82,75,76,79,72,0

```

Geladen wird der Stepper, wie jedes andere Programm auch, durch zweimaliges Anklicken des Icons mit der Maus. Nach dem Ladevorgang werden Sie sicher den Eindruck haben, daß der Rechner abgestürzt ist, da er keinerlei Mausbewegungen mehr ausführt. Daß dem nicht so ist, werden Sie feststellen, wenn Sie kurz die Commodore-Taste antippen. Dabei muß sich das folgende Bild ergeben:

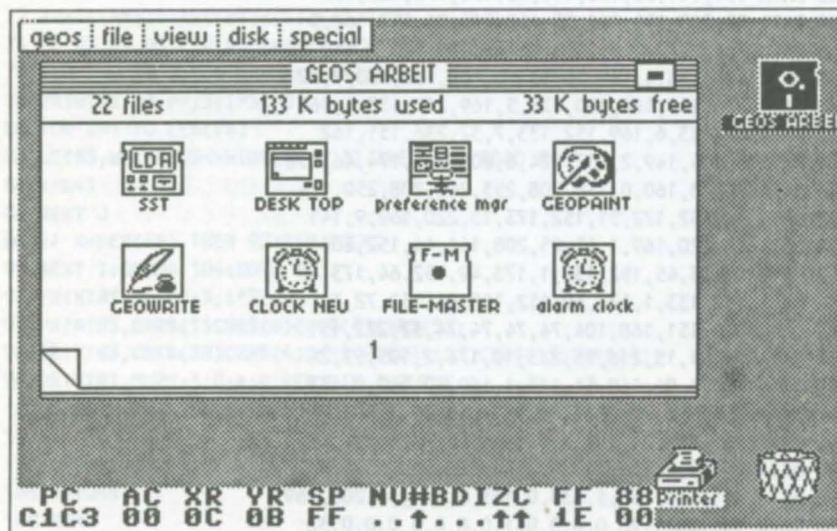


Abb. 50: Der Stepper

Am unteren Bildschirmrand wird eine aus zwei Zeilen bestehende Meldung eingeblendet. Die obere Zeile enthält die Namen der Register des 6510-Mikroprozessors. Dabei steht 'PC' für den Programmzähler (engl.: Program Counter), 'AC' für den Akkumulator, 'XR' bzw. 'YR' für das X- bzw. Y-Register und 'SP' für den Stapelzeiger (engl.: Stack Pointer). Die Flags des Statusregisters sind folgendermaßen gekennzeichnet:

<i>N</i>	Negative
<i>V</i>	Overflow
<i>#</i>	unbenutzt (immer gesetzt)
<i>B</i>	Break aufgetreten
<i>D</i>	Dezimalmodus
<i>I</i>	Interrupt
<i>Z</i>	Zero
<i>C</i>	Carry

Die untere Zeile enthält die aktuellen Registerinhalte des Prozessors nach der Abarbeitung des letzten Befehls. Die Ausgabe erfolgt hexadezimal, bis auf die des Statusregisters. Hier bedeutet der Pfeil ein gesetztes und der Punkt ein gelöscht Flag.

Außerdem finden Sie am Ende der beiden Zeilen noch die Inhalte von vier beliebigen Speicherstellen, die vor dem Start des Programms im Listing festgelegt werden müssen. Jedesmal, wenn Sie nun die Commodore-Taste betätigen, wird vom Prozessor genau der Befehl abgearbeitet, dessen Adresse sich im Programmzähler befindet. Anschließend wird die Anzeige aktualisiert.

Die Taste ist mit einer dynamischen Dauerfunktion ausgestattet, die aktiv wird, sobald sie längere Zeit betätigt wird. Die 'Stepzeit' wird dabei kürzer, je länger die Taste gedrückt wird. Dadurch können uninteressante Bereiche schneller abgearbeitet werden.

Sollte Ihnen die Geschwindigkeit noch nicht ausreichen, genügt ein Druck auf die linke SHIFT-Taste, um die Aktionen zu beschleunigen. Nun wird aber die Registeranzeige nicht mehr aktualisiert. Diesen Modus können Sie durch die SHIFT-LOCK-

Taste dauernd einschalten. Besonders nach dem Laden des Steppers muß diese Taste recht lange betätigt werden, da der Lade-prozeß des Steppers auch nach seinem Start noch nicht abgeschlossen ist (Neuaufbau des Bildschirms, Swap-File laden, etc.).

Natürlich arbeitet GEOS auch bei gedrückter SHIFT-Taste immer noch viel langsamer als im Normalzustand, da der Rechner nach jedem Programmschritt zumindest noch testen muß, welche Taste gedrückt ist.

5.5.2 Funktionsweise des Steppers

Hier noch einige ergänzende Anmerkungen zur Funktionsweise des Programms. Die Idee, die dem Programm zu Grunde liegt, ist recht einfach. Da der Mikroprozessor nach jedem abgearbeiteten Maschinen-Befehl anhalten soll, damit die Inhalte der Register ausgegeben werden können, haben wir uns der Interrupt-Technik bedient. Hardwarevoraussetzung ist hierfür nur ein Timer, der in der Lage ist, nach einer vorgegebenen Anzahl ausgeführter Taktzyklen eine Unterbrechung (=Interrupt) des Programms einzuleiten.

Diesen Timer stellt uns der 64er in einem der beiden Complex Interface Adapter (CIA) zur Verfügung. Er wird von uns nun genau so programmiert, daß der Mikroprozessor in der Lage ist, nur einen einzigen Befehl abzuarbeiten, bis eine solche Unterbrechung angefordert wird. Daraufhin wird in unsere Routine gesprungen, die die Ausgabe der Register in die Bit-Map erledigt, auf Tastendruck den Timer wieder startet und den nächsten Befehl des unterbrochenen Programms ausführt.

So einfach die Theorie auch klingen mag: In der Praxis haben sich einige Schwierigkeiten ergeben, da auch GEOS von der Interrupt-Technik Gebrauch macht, um z.B. die Tastatur und den Joystick abzufragen. Dazu wird noch nicht einmal unser Timer benötigt, da GEOS die Interrupts durch den Video-Chip erzeugen läßt, und zwar jedesmal dann, wenn eine bestimmte Zeile auf dem Bildschirm beschrieben wird.

Das so entstandene Problem der Interrupt-Kollisionen haben wir gelöst, indem wir den automatischen GEOS-Interrupt einfach abstellen und nur unser Programm durch Abfragen des Video-Chips entscheiden lassen, ob nach dem Stepper-IRQ zusätzlich noch ein GEOS-IRQ stattfinden soll. Dadurch ist gewährleistet, daß die Mausbewegungen und alle anderen Jobs des IRQ auch bei geladenem Stepper noch funktionieren.

Die meisten GEOS-Programme sind für den GEOS überlassen, beispielsweise Tastaturnachfrage und Maussteuerung. Trotzdem erscheinen in Deutschland keine Programme, die unter GEOS laufen und diese Möglichkeiten ausnutzen. Wir meinen, daß es zwei Gründe dafür gibt:

- Es gibt in Deutschland praktisch keine offiziellen Informationen über die GEOS-Routinen und deren Funktionen. Dadurch haben Programmierer natürlich keine Möglichkeiten, diese Routinen zu benutzen.
- Die Programmierung unter GEOS unterscheidet sich doch erheblich von der Programmierung der ähnlichen Core-KERNEL-Routinen. Zusätzlich erschwert das neue, von GEOS benutzte Fileformat die Einbindung eigener Programme.

In diesem Kapitel wollen wir Ihnen daher einige entscheidende Informationen zur Verfügung stellen. Zunächst einmal behandeln wir natürlich das neue Fileformat und zeigen die Vorteile dieser Änderung gegenüber dem veraltetem vom Happy-DOS benutzten Format auf. Weiterhin gehen wir auf den prinzipiellen Unterschied zwischen dem OS4- und dem GEOS-KERNEL ein. Dabei werden Sie feststellen, daß GEOS einen völlig neuen Programmiersatz von Ihnen verlangt; GEOS ist vollständig "event driven". Was dies bedeutet und vor allem welche Konsequenzen das für den Programmierer hat, erfahren Sie im Kapitel 6.2.

Mit der richtigen Einbindung neuer Programme in die GEOS-Umgebung setzen wir uns dann im nächsten Kapitel auseinander. Den Höhepunkt stellt aber sicherlich die vollständig dokumentierte GEOS-Sprungtabelle dar. Sie erlaubt den Zugriff auf alle über 170 fertigen Routinen, die das neue Kernel zur Verfügung stellt. Angefangen von der einfachen Erstellung von Pull-

Das 3D-Modellproblem des Text- und Bildschirms haben wir
 gelöst indem wir den gesamten GEOS-Flächeninhalt
 abhaken und dann unser Programm durch Abfragen des Video-
 Chips erstellen lassen. Das ist das GEOS-IRQ-System.
 noch ein GEOS-IRQ einfließen soll. Dadurch ist gewährleistet,
 daß alle Mikroprozessoren und alle Videokarten des IRQ auch
 im gleichen Schritt hochgelassen werden. Das ist ein
 Teil davon, zum Teil von Teilnehmern im GEOS-Forum
 ist.

5.5.2 Funktionsweise des Timers

Hier noch einige ergänzende Anmerkungen zur Funktionsweise
 des Timers. Die Idee, die dem Programm zu Grunde liegt,
 ist recht einfach. Da der Mikroprozessor nach jedem abgearbei-
 teten Maschinen-Befehl anhalten soll, damit die Inhalte der Re-
 gister ausgelesen werden können, haben wir uns der Interrupt-
 Technik bedient. Hardwarevoraussetzung ist hierfür nur ein Ti-
 mer, der in der Lage ist, nach einer vorgegebenen Anzahl aus-
 geführter Taktzyklen eine Unterbrechung (=Interrupt) des Pro-
 gramms einzuleiten.

Dieser Timer stellt uns der 64er in einem der beiden Comport
 Interface Adapter (CIA) zur Verfügung. Er wird von uns nun
 genau so programmiert, daß der Mikroprozessor in der Lage ist,
 nur einen einzigen Befehl abzuarbeiten, bis eine solche Unter-
 brechung angefordert wird. Daraufhin wird in unsere Routine
 gesprungen, die die Ausgabe der Register in die Bit-Map erlei-
 gert. Auf Tastendruck des Timers wieder startet und den näch-
 sten Befehl des unterbrochenen Programms ausführt.

So einfach die Theorie auch klingen mag: In der Praxis haben
 sich einige Schwierigkeiten ergab, da auch GEOS von der In-
 terrupt-Technik Gebrauch macht, um z.B. die Tastatur und das
 Joystick abzufragen. Dazu wird noch nicht einmal unser Timer
 benötigt, da GEOS die Interrupts durch den Video-Chip erzeu-
 gen läßt, und zwar jedesmal dann, wenn eine bestimmte Zeit
 auf dem Bildschirm beschrieben wird.

6. GEOS intern

GEOS stellt für die Programmierung eigener Applications und Accessories eine Vielzahl nützlicher Routinen zur Verfügung. Dadurch braucht sich der Programmierer um vieles nicht mehr zu kümmern, sondern kann das getrost GEOS überlassen, beispielsweise Tastaturabfrage und Maussteuerung. Trotzdem erscheinen in Deutschland keine Programme, die unter GEOS laufen und diese Möglichkeiten ausnutzen. Wir meinen, daß es zwei Gründe dafür gibt:

- Es gibt in Deutschland praktisch keine offiziellen Informationen über die GEOS-Routinen und deren Funktionen. Dadurch haben Programmierer natürlich keine Möglichkeiten, diese Routinen zu benutzen.
- Die Programmierung unter GEOS unterscheidet sich doch erheblich von der Programmierung der üblichen C64-KERNAL-Routinen. Zusätzlich erschwert das neue, von GEOS benutzte Fileformat die Einbindung eigener Programme.

Im diesem Kapitel wollen wir Ihnen daher einige entscheidende Informationen zur Verfügung stellen: Zunächst einmal behandeln wir ausführlich das neue Fileformat und zeigen die Vorteile dieser Neuerung gegenüber dem ursprünglich vom Floppy-DOS benutzten Format auf. Weiterhin gehen wir auf den prinzipiellen Unterschied zwischen dem C64- und dem GEOS-KERNAL ein. Dabei werden Sie feststellen, daß GEOS einen völlig neuen Programmierstil von Ihnen verlangt: GEOS ist vollständig "event driven". Was dies bedeutet und vor allem welche Konsequenzen dies für den Programmierer hat, erfahren Sie im Kapitel 6.2.

Mit der richtigen Einbindung neuer Programme in die GEOS-Umgebung setzen wir uns dann im nächsten Kapitel auseinander. Den Höhepunkt stellt aber sicherlich die vollständig dokumentierte GEOS-Sprungtabelle dar. Sie erlaubt den Zugriff auf die über 150 fertigen Routinen, die das neue Kernal zur Verfügung stellt. Angefangen von der einfachen Erstellung von Pull-

Down-Menüs über die Programmierung von Multi-Tasking-Prozessen bis hin zur Benutzung der Speeder-Routinen findet der Programmierer alles, was das Herz höher schlagen läßt.

Last but not least finden Sie im Anschluß daran eine vollständige Speicheraufteilung von GEOS. Denn was helfen die schönsten KERNAL-Routinen, wenn man nicht weiß, wo man das eigene Programm ungestört ablegen darf.

Aber auch wenn Sie keine Programme unter GEOS schreiben wollen, können Sie in diesem Kapitel viele wichtige Informationen finden: Welcher Bereich ist für BASIC-Programme nutzbar, wenn GEOS weiterhin im Speicher liegen soll? Mit welchen Speicherstellen kann man in GEOS entscheidende Veränderungen erzielen? Warum kann GEOS zwei Programme gleichzeitig in den Arbeitsspeicher laden (z.B. GEOPAINT und den Wecker). Wieso muß sich das DESK TOP auf der Diskette befinden, wenn man den Hilfstext eines Info-Fensters lesen möchte?

6.1 Die GEOS-Diskettenstrukturen

Eine der vielen Besonderheiten, die GEOS auszeichnet, ist die eigene Filestruktur. Sie ermöglicht erst die vielen zusätzlichen Möglichkeiten, die GEOS bei der File-Verwaltung bietet. Als wir uns zu Anfang mit GEOS beschäftigten und die vielen Änderungen gegenüber der normalen Filestruktur feststellten, hielten wir das für einen Versuch der Herstellerfirma, die internen Geheimnisse zu bewahren. Wir stellten dann aber bald fest, daß die vielen Informationen nicht Unkundige abschrecken sollen, sondern in GEOS eine wichtige Funktion haben. In diesem Kapitel wollen wir Ihnen die GEOS-Filestruktur in allen Einzelheiten erläutern. Dazu müssen wir aber noch einmal kurz darauf eingehen, wie das normale COMMODORE DOS die Files verwaltet.

6.1.1 File-Verwaltung beim COMMODORE DOS

Zu jedem File auf der Diskette gehört ein sogenannter File-Eintrag oder Directory-Eintrag, der alle wichtigen Informationen über das File enthält. Das Directory befindet sich auf Spur 18 der Diskette und beginnt bei Sektor 1

Format eines Directory-Eintrags:

Byte	Inhalt
0	Filetyp (*) geort mit # $\$80$
1,2	Spur und Sektor des ersten Datenblocks
3-18	Filename, ergänzt mit Shift-Space
19,20	Nur bei relativen Files benutzt
21	Nur bei relativen Files benutzt
22	Nicht benutzt
26,27	Spur und Sektor des neuen Files beim Überschreiben
28,29	Anzahl der Blocks im File (Low-High)

- (*) Filetypen:
- 0=Deleted
 - 1=Sequential
 - 2=Program
 - 3=User
 - 4=Relative

Ein Beispiel könnte so aussehen:

```

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
c2 11 00 F I L E N A M E A0 A0 A0 A0
16 17 18 19 20 21 22 23 24 25 26 27 28 29
A0 A0 A0 00 00 00 00 00 00 01 00 0A 00
    
```

Dieses File ist ein Programm, weil die untersten 4 Bit im Byte 01=2 sind. Außerdem ist es schreibgeschützt, weil das 6. Bit gesetzt ist ($\$C2$ statt $\$82$). Der erste Datenblock des Programms liegt auf Spur 17 (=11) und Sektor 0. Das File enthält einen Datenblock, also einen Sektor (Byte 26= $\$01$). An den vielen Nullen im File-Eintrag können Sie schon sehen, daß noch Platz

für zusätzliche Informationen vorhanden ist. Dieser Platz wird von GEOS konsequent genutzt. Wir wollen uns nun einmal einen File-Eintrag unter GEOS anschauen.

6.1.2 Ein File-Eintrag unter GEOS

Ein File-Eintrag unter GEOS-DOS enthält mehr Information. Diese zusätzliche Information steht an folgenden Stellen und hat die Bedeutung:

Byte 19,20	Sektor des INFO-Bildschirms
Byte 21	File-Struktur: 0=SEQUENTIAL 1=VLIR
Byte 22	Filetyp: 0 = Kein GEOS-File 1 = Basic 2 = Assembler 3 = Data File 4 = System File 5 = Desk Accessory 6 = Application 7 = Application Data 8 = Font File 9 = Printer Driver 10 = Input Device 11 = Disk Device
Byte 23-27	Datum und Zeit, und zwar mit dem Jahr beginnend

Wir wollen uns so ein GEOS-File einmal im Detail anschauen. Als Beispiel wählen wir das Programm GEOS BOOT und zeigen den entsprechenden Ausschnitt aus dem Directory. Ein GEOS-File-Eintrag:

```

0320 .....C2 01 10 47 45 4F ..b..geo
0328 53 20 42 4F 4F 54 A0 A0 s boot..
0330 A0 A0 A0 A0 A0 01 08 00 .....
0338 02 56 03 07 0F 00 06 00 .v.....
0340 00 00 .....
    
```

Der File-Eintrag beginnt bei \$0322. Die ersten Bytes entsprechen genau dem COMMODORE DOS. Das ist wichtig, damit die Files auch noch ohne GEOS geladen werden können. Die zusätzlichen Information, die GEOS im File-Eintrag ablegt, beginnen ab Byte \$0335 hinter dem letzten 'A0'. Der INFO-Sektor liegt auf Track 1 Sektor 8. Die File-Struktur ist

SEQUENTIAL

und der GEOS-Filetyp ist

ASSEMBLER-PROGRAMM (02)

Dahinter folgt dann das Datum, und zwar in der Reihenfolge:

Jahr, Monat, Tag, Stunde, Minute.

Dabei werden nur die beiden hinteren Ziffern des Jahres eingetragen, also \$56=86=1986. Die Stunden werden von 0-24 eingegeben, \$0F=15=03 Uhr PM. Damit Sie die GEOS-File-Einträge noch etwas besser kennenlernen und die zusätzlichen Informationen bald auf den ersten Blick erkennen, haben wir einen Teil des Directory abgedruckt. Schauen Sie sich die File-Einträge in Ruhe an. Die Informationen des GEOS DOS beginnen immer hinter dem letzten 'A0'.

Achten Sie dabei bitte darauf, daß GEOS im Gegensatz zum 'alten' C64 den Original-ASCII benutzt (s. Anhang). Dieser unterscheidet sich vom CBM-ASCII des 64er Kernal z.B. in der Groß- bzw. Kleinschreibung aller Buchstaben.

```

0300 12 09 C2 11 00 47 45 4F ..b..geo
0308 53 A0 A0 A0 A0 A0 A0 s.....
0310 A0 A0 A0 A0 A0 00 00 00 .....
0318 00 00 00 00 00 01 00 .....
0320 00 00 C2 01 10 47 45 4F ..b..geo
    
```

```

0328 53 20 42 4F 4F 54 A0 A0 s boot..
0330 A0 A0 A0 A0 A0 01 08 00 .....
0338 02 56 03 07 0F 00 06 00 .v.....
0340 00 00 C3 01 11 47 45 4F ..c..geo
0348 53 20 4B 45 52 4E 41 4C s kernal
0350 A0 A0 A0 A0 A0 01 09 01 .....
0358 04 56 03 07 0F 00 54 00 .v....t.
0360 00 00 C3 05 07 44 45 53 ..c..des
0368 4B 20 54 4F 50 A0 A0 A0 k top...
0370 A0 A0 A0 A0 A0 05 14 01 .....
0378 04 56 03 07 0F 00 47 00 .v....g.
0380 00 00 C3 08 0F 47 45 4F ..c..geo
0388 50 41 49 4E 54 A0 A0 A0 paint...
0390 A0 A0 A0 A0 A0 08 06 01 .....
0398 06 56 03 07 0F 00 5B 00 .v.....
03A0 00 00 C3 0D 02 47 45 4F ..c..geo

```

Zu jedem GEOS-File gehört ein INFO-Sektor, der viele Informationen über das File enthält. Daher wollen wir uns als nächstes den Aufbau so eines Sektors einmal anschauen.

6.1.3 Der INFO-Sektor

Die Adresse des INFO-Bildschirms steht im File-Eintrag hinter dem letzten 'A0', also im Byte 19,20. Diese Bytes werden vom DOS nur bei relativen Files benutzt. Im INFO-Sektor haben die Bytes folgende Bedeutung:

- 0-1 00,FF - normalerweise enthalten die ersten beiden Bytes den Zeiger auf den folgenden Sektor. Byte 0=Track, Byte 1=Sektor. Da es keinen Folgesektor gibt, ist Byte 0=\$00, und Byte 1 gibt die Anzahl der gültigen Bytes in diesem Sektor an. Die ersten beiden Bytes entsprechen damit dem üblichen COMMODORE-DOS-Format.
- 2-3 Diese beiden Bytes geben die Größe des Icons an, das auf dem Bildschirm dieses File symbolisiert. Das Byte \$02 gibt die Breite an und das Byte \$03 die Höhe. Dabei bedeutet die Breite '3' eine Breite

- von 3 Bytes und die Höhe '\$15= 21' 21 Zeilen. Damit hat das Icon genau das Format eines Sprites, das auch aus 63 Byte besteht.
- 4 Dieses Byte leitet die Grafikdaten des Icons ein. Da das 7. Bit gesetzt ist, sind die nun folgenden Bytes unkomprimiert abgelegt. In GEOS werden solche Daten nämlich häufig gepackt, um Speicherplatz zu sparen. Die restlichen Bits dieser Speicherstelle geben die Anzahl der nun folgenden Bytes an (\$3F=63).
- 5-67 Sprite-Darstellung des Icons. Hier stehen die 63 Bytes, die das Aussehen des Icons bestimmen. Sie haben genau Sprite-Format, also jeweils 3 Bytes ergeben eine Zeile des Sprites.
- 68 Filetyp (C 64) geort mit \$80. GEOS schaut hier bei Speichervorgängen nach, ob es sich um Daten (=1) oder ein Programm (=2) handelt. Für Programme muß hier also ein \$82=130 stehen. Bei aktivem Schreibschutz ist zusätzlich Bit 6=1.
- 69 GEOS-Filetyp (in Klammern die amerikanische Bezeichnung):
- 0 = altes C64 File (kein GEOS-File)
 - 1 = Basic
 - 2 = Assembler
 - 3 = reine Daten (Data File)
 - 4 = Systemdatei (System File)
 - 5 = Hilfsprogramm (Desk Accessory)
 - 6 = Anwendung (Application)
 - 7 = Anwendungsdaten (Application Data)
 - 8 = Schriftart (Font File)
 - 9 = Druckertreiber (Printer Driver)
 - 10 = Eingabetreiber (Input Driver)
 - 11 = Disk-Treiber (Disk Driver)
 - 12 = Startprogramm (System Boot File)
 - 13 = temporär (Temporary)
 - 14 = selbstausführend (Auto-Execute)

- 70 GEOS-File-Struktur (0=SEQ 1=VLIR). An dieser Stelle bedeutet SEQ=SEQUENTIAL nicht dasselbe wie im COMMODORE DOS, sondern die Sektoren sind hier jeweils über den Zeiger in Byte 0,1 verknüpft. SEQ meint also die normale Filestruktur. VLIR=VARIABLE LENGTH INDEXED RECORD stellt ein anderes Format der Files dar. Wir werden etwas später noch darauf eingehen. Eigene Programme, die in GEOS eingebunden werden sollen, erhalten normalerweise eine '0'.
- 71,72 Ladeadresse des Programms. An diese Stelle wird das Programm von GEOS geladen. ACHTUNG: Im alten DOS geben die Bytes 2,3 im ersten Datensektor des Programms die Startadresse an. Das ist bei GEOS anders. Wie man Schwierigkeiten vermeiden kann, werden wir etwas später erklären. Dort zeigen wir dann genau, wie man ein eigenes Programm in GEOS einbindet.
- 73,74 Diese beiden Bytes haben nur bei Accessories (Hilfsprogrammen) eine Bedeutung. Wenn Sie also als GEOS-Filetyp z.B. eine '2' eintragen, brauchen Sie sich um diese beiden Bytes nicht zu kümmern. Tragen Sie aber eine '5' für Accessory ein, so bestimmen diese Bytes das Ende des Programms (mind. 1 Byte lang!). Damit weiß GEOS, welcher Bereich des Arbeitsspeichers beim Laden des Accessories als Swap-File auf die Diskette gerettet werden muß. Genauere Informationen finden Sie im Abschnitt über das Einbinden eigener Programme.
- 75,76 Einsprungadresse des Programms. An der hier eingetragenen Stelle sollte sich die Initialisierung Ihres Programms oder ein Sprungbefehl dorthin befinden. Oft stimmt diese Adresse mit der Ladeadresse des Programms überein.
- 77-96 Hier steht der Text, der im Info-Fenster unter KLASSE erscheint.

- 97-116 Name des Autors. Bei Datenfiles steht hier jedoch der Name der Diskette, auf der sich die zugehörige Anwendung befindet.
- 117-136 Hier steht bei Anwendungsdaten der Name der zugehörigen Anwendung.
- 137-159 Frei für Anwendungen.
- 160-255 Hier steht die Information, die im unteren Teil des Info-Fensters ausgegeben wird (falls sich das DESKTOP auf der Diskette befindet). Der Text muß mit einer Null enden.

Wir wollen uns als Beispiel einmal den INFO-Sektor der Alarm Clock der GEOS-Version 1.2 anschauen. Er sieht, mit einem Disk-Monitor betrachtet, folgendermaßen aus (Beachten Sie, daß GEOS bei den Buchstaben und Zeichen die echte ASCII-Darstellung wählt):

```

0300 00 FF 03 15 BF FF FF FF .....
0308 80 00 01 83 3C C1 8C A5 .....
0310 31 89 7E 91 93 81 C9 96 .....
0318 08 69 8C 49 31 84 10 21 .....
0320 89 10 91 88 10 11 88 1F .....
0328 D1 88 00 11 89 00 91 84 .....
0330 00 21 84 49 21 86 10 61 .....
0338 87 81 E1 8E 7E 71 80 00 .....
0340 01 FF FF FF 83 05 00 00 .....
0348 54 D8 5F 00 54 41 6C 61 .....aLA
0350 72 6D 20 63 6C 6F 63 68 RM CLOCK
0358 20 56 31 2E 30 00 00 00 v1.0....
0360 00 44 61 76 69 64 20 44 ddAVID d
0368 75 72 72 61 6E 00 10 A9 URRAN...
0370 01 85 11 A9 00 85 17 85 .....
0378 16 20 3B C2 8A D0 0A A5 .....
0380 11 D0 09 20 55 08 B8 50 .....
0388 03 20 20 1A 60 A2 90 A9 .....
0390 00 9D DE 43 CA D0 FA A2 .....
0398 FF A9 00 9D 6E 44 CA D0 .....
03A0 53 65 74 20 74 68 65 20 SET THE
03A8 61 6C 61 72 6D 20 63 6C ALARM CL
03B0 6F 63 68 20 74 6F 20 68 OCK TO K
03B8 65 65 70 20 79 6F 75 72 EEP ZOUR
03C0 73 65 6C 66 20 74 69 6D SELF TIM
    
```

```

03C8 65 2D 63 6F 6E 73 63 69 E-CONSCI
03D0 6F 75 73 2E 00 3B C2 8A OUS.....
03D8 D0 24 A9 08 38 E5 11 18 .....
03E0 69 04 48 09 80 8D C0 05 .....
03E8 68 85 02 A9 0E 85 04 A0 .....
03F0 02 A2 04 20 60 C1 18 69 .....
03F8 0C 8D BB 05 A2 00 60 05 .....

```

Wir haben hinter den Zahlen nur die Bytes als Zeichen ausgegeben lassen, die die Texte enthalten. Ab \$0347 stehen im Format LOW-Byte, HIGH-Byte Ladeadresse (\$5400), Endadresse (\$5FD8) und Initialisierung-Einsprung (\$5400). Beim Laden des Weckers wird also vorher der Bereich \$5400-\$5FD8 als Swap-File auf der Diskette gespeichert. Wenn der Wecker wieder verlassen wird, lädt GEOS das SWAP FILE wieder an diese Stelle. Mit dieser Methode kann aus jeder Anwendung heraus ein Accessory (Hilfsprogramm) geladen werden, ohne daß in der Anwendung Daten verlorengehen.

6.1.4 Die Border und GEOS-Format V1.0

Es gibt noch zwei weitere Besonderheiten der GEOS-Filestruktur. Um diese zu erklären, müssen wir uns einmal die BAM einer Diskette anschauen. Sie liegt immer auf Track 18 Sektor 0. Normalerweise legt das COMMODORE DOS folgende Informationen in der BAM ab:

Die BAM des COMMODORE DOS:

0,1	Spur und Sektor des ersten Blocks der Directory
2	Format ("A"-1541-Format)
3	0
4-143	Bitmuster der belegten und nicht belegten Blocks (0=belegt). Jeweils 4 Byte gehören zusammen und charakterisieren eine Spur. Dabei gibt das erste Byte die Anzahl der freien Blöcke auf dieser Spur und die nächsten drei Byte die Bitmuster für die Sektoren 0-7, 8-15, 16-23

- 144-161 Name der Diskette ergänzt mit 'SHIFT + SPACE'='\$A0'
- 162-163 ID der Diskette
- 164 SHIFT + SPACE
- 165-166 "2A"-Format und DOS-Version
- 167-170 SHIFT + SPACE
- 171-255 Nicht benutzt und mit 0 aufgefüllt.

Auch hier bleibt also (wie im Directory) eine Menge ungenutzter Platz, den GEOS für eigene Zwecke nutzt. Dabei werden an zwei Stellen zusätzliche Informationen abgelegt:

Informationen des GEOS DOS:

- 171-172 Track und Sektor der Border. Wenn Sie eine nicht von GEOS formatierte Diskette unter GEOS benutzen, so erscheint eine entsprechende Meldung. Sie können die Diskette ins GEOS-Format umwandeln lassen. Dabei wird ein weiterer Block reserviert, und der Zeiger in 171,172 zeigt auf diesen Block. Er stellt also eine Erweiterung des Directory dar. Alle Files, die von GEOS auf der Border abgelegt werden, erhalten im normalen Directory das Zeichen für 'scratched' und werden in diesem Border-Block eingetragen. Da in einen Sektor nur acht File-Einträge passen, kann man maximal acht Files auf der Border ablegen.
- 173-188 "geos FORMAT v1.0" Gleichzeitig mit dem Anlegen des Borderblocks wird hier das GEOS-Format eingetragen. An dem Eintrag "v1.0" erkennt man, daß sich die GEOS-Filestruktur mit der neuen Version GEOS V1.2 nicht geändert hat. GEOS erkennt beim Öffnen an diesem Eintrag, ob die Diskette GEOS-Format hat.

Wir wollen uns jetzt einmal die BAM einer GEOS-Diskette anschauen. Es ist zwar die BAM einer GEOS-V1.2-Diskette, die Erklärungen gelten aber selbstverständlich genauso für die deutsche Version 1.3.

BAM einer GEOS-Diskette:

```

0300 12 01 41 00 00 00 00 00 ..A.....
0308 00 00 00 00 00 00 00 00 .....
0310 00 00 00 00 00 00 00 00 .....
0318 00 00 00 00 00 00 00 00 .....
0320 00 00 00 00 00 00 00 00 .....
0328 00 00 00 00 00 00 00 00 .....
0330 00 00 00 00 00 00 00 00 .....
0338 00 00 00 00 00 00 00 00 .....
0340 00 00 00 00 00 00 00 00 .....
0348 0E BC FD 05 00 00 00 00 .....
0350 00 00 00 00 00 00 00 00 .....
0358 00 00 00 00 00 00 00 00 .....
0360 00 00 00 00 00 00 00 00 .....
0368 00 00 00 00 00 00 00 00 .....
0370 00 00 00 00 00 00 00 00 .....
0378 00 00 00 00 00 00 00 00 .....
0380 00 00 00 00 00 00 00 00 .....
0388 00 00 00 00 02 10 08 00 .....
0390 47 45 4F 53 20 56 31 2E geos v1.
0398 32 A0 A0 A0 A0 A0 A0 A0 .....
03A0 A0 A0 44 46 A0 32 41 A0 ..df.2a.
03A8 A0 A0 A0 13 08 47 45 4F .....geo
03B0 53 20 66 6F 72 6D 61 74 s FORMAT
03B8 20 56 31 2E 30 00 00 00 ..v1.0...
03C0 00 00 00 00 00 00 00 00 .....
03C8 00 00 00 00 00 00 00 00 .....
03D0 00 00 00 00 00 00 00 00 .....
03D8 00 00 00 00 00 00 00 00 .....
03E0 00 00 00 00 00 00 00 00 .....
03E8 00 00 00 00 00 00 00 00 .....
03F0 00 00 00 00 00 00 00 00 .....
03F8 00 00 00 00 00 00 00 00 .....

```

Der Zeiger für den Border-Block zeigt auf Track \$13 Sektor \$08 (=19,8). Diesen Block wollen wir uns auch einmal anschauen, um zu zeigen, daß er genau wie ein Block des Directory aufgebaut ist.

Der Border-Block:

```

0300 00 FF 83 23 05 4D 50 53 .....mps
0308 2D 31 30 30 30 A0 A0 -1000...
0310 A0 A0 A0 A0 A0 23 0F 00 .....
0318 09 56 04 07 0F 06 04 00 .....
0320 00 00 83 22 03 4D 50 53 .....mps
0328 2D 38 30 33 A0 A0 A0 -803....
0330 A0 A0 A0 A0 A0 22 0D 00 .....
0338 09 56 04 07 0F 06 04 00 .....
0340 00 00 00 20 0D 55 6E 69 .....UNI
0348 76 65 72 73 69 74 79 A0 VERSITY.
0350 A0 A0 A0 A0 A0 20 06 01 .....
0358 08 56 04 07 0C 00 28 00 .....
0360 00 00 00 1A 00 43 61 6C .....cAL
0368 69 66 6F 72 6E 69 61 A0 IFORNIA.
0370 A0 A0 A0 A0 A0 1A 0B 01 .....
0378 08 56 04 07 0C 00 1A 00 .....
0380 00 00 00 1C 01 43 6F 72 .....cOR
0388 79 A0 A0 A0 A0 A0 A0 Y.....
0390 A0 A0 A0 A0 A0 1C 0C 01 .....
0398 08 56 04 07 0C 00 17 00 .....
03A0 00 00 00 1D 04 44 77 69 .....dWI
03A8 6E 65 6C 6C 65 A0 A0 NELLE...
03B0 A0 A0 A0 A0 A0 1D 0F 01 .....
03B8 08 56 04 07 0C 00 0D 00 .....
03C0 00 00 00 1E 05 52 6F 6D .....rOM
03C8 61 A0 A0 A0 A0 A0 A0 A.....
03D0 A0 A0 A0 A0 A0 1E 10 01 .....
03D8 08 56 04 07 0C 00 22 00 .....
03E0 00 00 00 23 05 4D 50 53 .....mps
03E8 2D 31 30 30 30 A0 A0 -1000...
03F0 A0 A0 A0 A0 A0 23 0F 00 .....
03F8 09 56 04 07 0F 06 04 00 .....
    
```

Auch hier haben wir nur die Zeichen für die Namen stehen gelassen und die restlichen ASCII-Zeichen rechts, die nur verwirren würden, durch '.' ersetzt. Beachten Sie aber bitte die eigenartige Schreibweise. Der erste Buchstabe erscheint klein geschrieben und die folgenden groß. Wir haben das absichtlich so übernommen, damit Sie die File-Einträge in Ihrem Disk-Monitor genau so wiederfinden. Trotzdem wird dann später in GEOS der erste Buchstabe groß- und die folgenden klein geschrieben.

Die ersten beiden Byte des Border-Blocks geben an, daß es keinen Folgeblock gibt und daß \$FF (=255) gültige Daten vorliegen. Nur die ersten beiden File-Einträge "mps-1000" und "mps-803" sind gültig, denn nur sie haben einen gültigen Filetyp (\$83). Die folgenden File-Einträge tauchen nicht auf der Border auf, weil der Filetyp auf '0' (=DELETED) gesetzt ist.

Achtung: Wenn Sie ein File auf die Border setzen und dann GEOS verlassen, können Sie es ohne GEOS nicht wieder laden. Das File wird von GEOS also im normalen Directory gelöscht. Sollten Sie also von der Originaldiskette GEOS BOOT auf die Border setzen und dann GEOS verlassen, so können Sie GEOS anschließend nicht mehr booten.

6.1.5 Filetyp - Filestruktur

Wenn Sie eigene Programme in GEOS einbinden wollen, müssen Sie wissen, welche Filetypen GEOS verwendet. Vielleicht ist Ihnen schon einmal aufgefallen, daß im INFO-Bildschirm zu den verschiedenen GEOS-Files unter TYP Begriffe wie Systemdatei, Hilfsprogramm, Schriftart usw. auftauchen. GEOS unterscheidet zur Zeit insgesamt vierzehn verschiedene Filetypen. Wir stellen Ihnen hier die verschiedenen Filetypen in einer Übersicht zusammen. Dabei nennen wir (falls vorhanden) jeweils ein Beispiel und geben an, ob dieses File die Filestruktur SEQ oder VLIR hat. Auf den Unterschied gehen wir dann allerdings weiter unten ein. Wir benutzen in der Auflistung die amerikanischen Bezeichnungen der GEOS-Version 1.2, da sie sich unter Programmierern durchgesetzt haben.

Abkürzungen

<i>GT</i>	GEOS-Filetyp
<i>FS</i>	Filestruktur
<i>Bez</i>	Bezeichnung im INFO-Fenster
<i>B</i>	Beispiel

GT=1,	FS=0,	Bez=BASIC,	B=BACKUP
GT=2,	FS=0,	Bez=ASSEMBLER,	B=GEOS BOOT
GT=3,	FS=?,	Bez=DATA FILE,	B=Harcopys unseres Programms
GT=4,	FS=1,	Bez=SYSTEM FILE,	B=GEOS KERNAL
GT=4,	FS=0,	Bez=SYSTEM FILE,	B=PREFERENCES
GT=5,	FS=0,	Bez=DESK ACCESSORY,	B=ALARM CLOCK
GT=6,	FS=1,	Bez=APPLICATION,	B=GEOPAINT
GT=7,	FS=1,	Bez=APPLICATION DATA,	B=NOTES
GT=8,	FS=1,	Bez=FONT FILE,	B=UNIVERSITY
GT=9,	FS=0,	Bez=PRINTER DRIVER,	B=MPS 1000
GT=10,	FS=0,	Bez=INPUT DRIVER,	B=JOYSTICK
GT=11,	FS=?,	Bez=DISK DEVICE	(Beispiel noch nicht vorhanden)
GT=12,	FS=0,	Bez=SYSTEM BOOT,	B=GEOS BOOT
GT=13,	FS=0,	Bez=TEMPORARY,	B=SWAP FILE
GT=14,	FS=0,	Bez=AUTO-EXEC,	B=unsere Clock

An diesem GEOS-Filetyp erkennt GEOS auch, was Sie mit diesem File machen können. Beispielsweise können Sie kein Hilfsprogramm (GT=5) drucken oder keine Systemdatei (GT=4) als Druckertreiber laden. Nun wollen wir noch den Unterschied zwischen FS=0 (SEQ) und FS=1 (VLIR) erklären:

Filestruktur: SEQ

Wenn Sie ein Programm ohne GEOS speichern, so wird ein Directory-Eintrag erstellt. Dabei wird hinter dem Filetyp ein Zeiger auf den ersten Datenblock gesetzt. Jeweils am Anfang des Datenblocks steht dann ein Zeiger, der auf den folgenden Datenblock zeigt. Der letzte Block der Daten hat als erstes Byte immer eine 0, und das zweite Byte gibt die Anzahl der gültigen Daten im letzten Sektor an. Dadurch sind die einzelnen Datenblöcke bildlich gesprochen zu einer langen Kette verknüpft. Deshalb nennt GEOS dies auch SEQUENTIAL.

Wenn man nun beispielsweise den letzten Datenblock lesen will, muß man sich an den Zeigern 'entlanghangeln', bis man zum letzten Block gelangt. Dazu müssen alle vorherigen Sektoren der Kette geladen werden. Außerdem kann man eigentlich nur etwas

hinzufügen, indem man die Kette nach hinten hin verlängert. Deshalb hat GEOS noch eine zweite Filestruktur.

Filestruktur: VLIR (VARIABLE LENGTH INDEXED RECORD)

Wenn Sie einmal einen File-Eintrag mit dem Disketten-Monitor anschauen, der die File-Struktur VLIR hat (beispielsweise DESK TOP), so werden Sie zunächst keinen Unterschied feststellen. Wenn Sie aber den ersten Datenblock laden (Track und Sektor stehen hinter dem Filetyp), so werden Sie erstaunt feststellen, daß es scheinbar nur einen einzigen Sektor Daten gibt. Dieser Sektor hat als erstes nämlich eine 0, es folgen also keine weiteren Sektoren. Außerdem steht in diesem Sektor gar nicht der Anfang des Programms oder der Daten, sondern nur einige Zahlen, und der Rest des Blocks ist abwechselnd mit '00' und 'FF' gefüllt.

Die ersten Zahlen in diesem Sektor stellen Zeiger auf Teile (RECORDS) des Programms oder der Daten dar. Diese Teile sind dann untereinander wieder SEQUENTIAL verknüpft, also mit einem Zeiger an jedem Blockanfang auf den folgenden Datenblock. Das hat zwei wichtige Vorteile:

- Die Anzahl der einzelnen Programm- oder Datenteile (RECORDS) ist veränderlich. Es können RECORDS angehängt, oder auch wieder gelöscht werden.
- Ein schnellerer Zugriff auf z.B. den letzten RECORD ist möglich, weil man nicht die ersten RECORDS lesen muß.

Das macht sich GEOS beispielsweise beim Notizbuch zunutze, indem dieses die Filestruktur VLIR erhalten hat. Dadurch gibt es bis zu 127 Zeiger auf die vorhandenen Blätter des Notizbuches. Jeder RECORD entspricht also einem Blatt und besteht nur aus einem Sektor.

Wenn Sie also 3 Blätter des Notizbuches beschrieben haben und mit dem Disk-Monitor den Sektor anschauen, der hinter dem Filetyp im File-Eintrag als erster Block angegeben ist, tauchen hinter den ersten beiden Bytes \$00 und \$FF sechs Bytes auf.

Dies sind die Zeiger auf die drei Sektoren, in denen der Text jeweils einer Seite steht. Die Zeiger haben das Format: Track, Sektor.

GEOS kennt aber nicht nur Daten, sondern auch Programme mit der Filestruktur VLIR. Hier sieht der Aufbau zwar genauso aus, auf jeden RECORD zeigt also ein Zeiger, aber das Laden erfolgt in einer anderen Weise. Der erste RECORD stellt nämlich in den meisten Fällen schon das ganze Programm dar. Die anderen Zeiger, die dahinter stehen, zeigen auf RECORDS, die von dem gestarteten Programm in OVERLAY-Technik nachgeladen werden können. Das heißt: Teile des bestehenden Programms werden durch nachgeladene Teile überschrieben. GEOPAINT hat beispielsweise 7 Zeiger:

<u>Track</u>	<u>Sektor</u>
\$08	\$03
\$10	\$11
\$10	\$0A
\$10	\$06
\$11	\$12
\$13	\$05
\$13	\$0F

Diese Zeiger können auf Ihrer Diskette durchaus andere Werte haben. Wenn Sie GEOPAINT vom DESK TOP aus anklicken, wird der erste RECORD (Track \$08, Sektor \$03) geladen. Die anderen RECORDS können dann von GEOPAINT bei bestimmten Arbeitsvorgängen nachgeladen werden. Wir wollen Ihnen einmal zwei Beispiele für die OVERLAY-Technik nennen:

1. DESK TOP besteht aus zwei RECORDS:

RECORD 1: Track \$05 Sektor \$10
RECORD 2: Track \$08 Sektor \$02

Normalerweise ist RECORD 1 geladen. Wenn Sie aber den INFO-Bildschirm laden, wird für die Ausgabe des Hilfstextes RECORD 2 nachgeladen. Das merken Sie schon daran, daß, falls

Sie DESK TOP nicht auf der aktuellen Diskette haben, kein Text ausgegeben wird, sondern die Meldung erscheint:

```
DESK TOP NOT ON DISK
```

Hier kann GEOS also die Textausgabe nicht nachladen und gibt den Hilfstext nicht aus.

2. Wenn Sie verschiedene Werkzeuge in GEOPAINT anklicken, werden von der Floppy Routinen nachgeladen, die zu diesem Werkzeug gehören. So gehört beispielsweise der RECORD 2 in GEOPAINT (Track \$10, Sektor \$11) zum 'Hahn', mit dem Sie Flächen füllen können.

Wie kann man nun einen RECORD laden? Dazu benötigen Sie einen Disk-Monitor. Schreiben Sie zuerst Track und Sektor des RECORDS auf, den Sie laden möchten. Nun erstellen Sie ein Dummy. Das ist in diesem Fall ein völlig nutzloses, kleines Programm, das zu nichts anderem dient, als einen File-Eintrag herzustellen. Sie können beispielsweise von BASIC aus ein Dummy erzeugen, indem Sie folgendes eingeben:

```
NEW  
10 GOTO 10  
SAVE"dummy",8
```

Nun benötigen Sie noch zwei Informationen:

1. Startadresse des RECORDS im C64-Speicher. Dazu schauen Sie im File-Eintrag nach, wo das File, dessen RECORD Sie laden wollen, den INFO-Sektor hat. Hinter dem letzten '\$A0' des File-Eintrags stehen Track und Sektor des INFO-Bildschirms. Laden Sie diesen Sektor in den Puffer 0 der Floppy (ab \$0300), und die Ladeadresse des RECORDS steht in den Speicherstellen \$0347 (LOW) und \$0348 (HIGH). Diese Adresse gilt in der Regel jedoch nur für den ersten RECORD, der den Hauptteil eines Programms enthält. Ist das Programm nämlich erst einmal gestartet, kann es selbst entscheiden, in welchen Bereich bei

Bedarf weitere RECORDS geladen werden. Bei GEO-PAINT lautet die Startadresse für den ersten Programmteil \$0400.

2. Sie müssen die beiden ersten Bytes des RECORDS ändern, da hier für das alte DOS-Format die Ladeadresse gespeichert werden muß. Um die Original-Werte nicht zu verlieren, müssen Sie sich diese nun merken. Holen Sie in unserem Beispiel den ersten Block (Track \$08, Sektor \$03) in den Puffer. Die ersten Bytes lauten dann:

```
$0300=Track des nächsten Blocks  
$0301=Sektor des nächsten Blocks  
$0302=$4C=JMP  
$0303=$04=Adresse: LOW  
$0304=$05=Adresse: HIGH
```

Damit Sie den RECORD an die richtige Adresse laden, schreiben Sie sich bitte die Bytes \$0302=\$4C und \$0303=\$04 auf. Danach ändern Sie diese in \$0302=\$02 und \$0303=\$04. Wenn Sie später diesen RECORD laden, muß nämlich in diesen beiden Bytes die Startadresse für das Laden stehen. Diese Adresse muß 2 Bytes höher sein als die Ladeadresse des RECORDS, weil die ersten beiden Bytes ja überschrieben werden und von Ihnen nachträglich eingefügt werden müssen.

Nun schreiben Sie den geänderten Sektor auf die Diskette zurück. Anschließend holen Sie den Teil des Directory der Diskette in den Puffer der Floppy, in dem der File-Eintrag "dummy" steht. In diesem File-Eintrag ändern Sie die Adresse des ersten Datenblocks in die Adresse des RECORDS um. Sie schreiben also hinter die \$82 (Filetyp im File-Eintrag) eine \$08 (Starttrack des RECORDS) und dahinter eine \$03 (Startsektor des RECORDS). Ihr File-Eintrag "dummy" beginnt jetzt also mit den drei Bytes:

```
$82 $08 $03
```

Nachdem Sie den geänderten Sektor des Inhaltsverzeichnisses auf die Diskette geschrieben haben, können Sie nun den RECORD laden, indem Sie einfach das "dummy" laden.

```
LOAD"dummy",8,1
```

Wenn Sie danach mit dem Monitor die beiden fehlenden Bytes einfügen, haben Sie GEOPAINT original im Speicher. Ändern Sie also \$0400 in \$4C und \$0401 in \$04 um.

Hier taucht allerdings die Schwierigkeit auf, daß ab \$0400 üblicherweise der Bildschirm liegt. Diese Schwierigkeit können Sie einfach umgehen, indem Sie GEOPAINT um \$1000 verschoben laden. Dazu schreiben Sie in den ersten Block des RECORDS nicht \$02, \$04, sondern \$02, \$14. Dadurch wird GEOPAINT nach \$1402 geladen, und Sie können sich das Programm ohne in Ruhe anschauen. Sollten Sie die Änderung später wieder rückgängig machen, dürfen Sie die Dummy-Datei anschließend nicht einfach durch den Scratch-Befehl des DOS löschen, da hierbei sämtliche Blocks freigegeben würden, die der erste RECORD belegt. Die einfachste Möglichkeit, die Dummy-Datei zu entfernen, besteht darin, den Filetyp im Directory auf Null zu setzen oder die Diskette von GEOS aus nach dem normalen Löschen der Datei neu zu validieren (aufzuräumen).

6.2 Wie GEOS funktioniert

Eigentlich ist die Bezeichnung GEOS auch in diesem Buch etwas ungenau. Wenn wir beispielsweise davon sprechen, daß GEOS ein Fenster mit einem bestimmten Inhalt zeigt oder GEOS eine Diskette kopiert, so ist das eigentlich nicht ganz richtig. Andererseits ist das für den, der GEOS wirklich nur benutzen will, auch ziemlich uninteressant. Wenn wir Ihnen aber hier einen Einblick darin verschaffen wollen, wie GEOS (immer noch ungenau) funktioniert, müssen wir uns etwas genauer ausdrücken.

Eigentlich ist GEOS "nur" das GEOS KERNAL. Zusätzlich gibt es ein weiteres System-File (DESK · TOP), Anwendungen (Applications) und Hilfsprogramme (Accessories). Dabei nimmt das DESK TOP eine gewisse Sonderstellung ein, obwohl es viel Ähnlichkeit mit einer Anwendung hat. Es gibt eine spezielle

Routine, um das DESK TOP nachzuladen, und beim Booten wird nach dem GEOS KERNAL automatisch ein File mit dem Namen DESK TOP geladen.

Das GEOS KERNAL stellt beispielsweise leistungsfähige Routinen zur Erstellung von Fenstern und zum Laden und Speichern von Sektoren von und auf Diskette zur Verfügung. Das DESK TOP nutzt diese Routinen, um vor bestimmten Aktionen zu warnen oder um ein File zu kopieren. Nun ist das KERNAL aber nicht einfach eine Ansammlung fertiger Routinen, die nur darauf warten, von DESK TOP oder einem anderen Programm aus angesprochen zu werden. Vielmehr ist das GEOS KERNAL selbst ein aktiver Teil des Programms. Wir wollen diesen kleinen, aber feinen Unterschied einmal kurz an einem Beispiel erläutern.

Beim C64 ohne GEOS gibt es eine Routine, die die Tastatur abfragt. Braucht ein Maschinensprache-Programmierer also eine Tastatureingabe, so springt er in diese Routine ein und erhält anschließend im Akku den Code des Zeichens oder eine "0", falls keine Taste gedrückt wurde, zurück. Bei GEOS ist das KERNAL nicht so passiv. Wenn ein Programm auf Eingaben über die Tastatur reagieren will, so fragt es nicht selbst in regelmäßigen Abständen die Tastatur ab, sondern übergibt an das KERNAL eine Adresse, zu der verzweigt werden soll, wenn ein Tastendruck erfolgt ist. Um alles weitere braucht das selbstgeschriebene Programm sich dann nicht mehr zu kümmern.

Dieser Unterschied ist so entscheidend, daß wir ihn noch einmal allgemein beschreiben wollen. Ein Programm, das unter GEOS läuft, ist mit dem KERNAL über Ereignisse (EVENTS) verbunden. Es übergibt an das KERNAL verschiedene Einsprungadressen, zu denen bei ganz bestimmten Ereignissen verzweigt werden soll. Für diese Ereignisse müssen natürliche fertige Routinen vorliegen.

Dieser Unterschied hat große Auswirkungen auf die Art und Weise, wie Programme unter GEOS geschrieben sein sollten. Daher sollten sich eigene Programme auch an ganz bestimmte Regeln halten, damit GEOS richtig funktionieren kann. Falsch

wäre es beispielsweise, ein Programm zu schreiben, das nach dem Laden und Starten in einer Schleife selbständig die Tastatur abfragt und bei gedrückter Taste "a" die Routine "DESK TOP laden und starten" anspringt. Durch so eine Programmierung werden die Möglichkeiten, die GEOS bietet, nicht genutzt und sogar blockiert.

Wir möchten daher kurz erklären, wie ein eigenes Programm beginnen und wie es enden sollte. Angenommen, wir schreiben ein Hilfsprogramm, beispielsweise mit dem EDMON. Dann sollte zu Beginn dieses Programms eine Initialisierungsroutine stehen. Die Startadresse wird vom EDMON beim Abspeichern des Programms automatisch im INFO-Sektor eingetragen. Dieser Initialisierungskopf unterscheidet sich erheblich von einer Initialisierung, die ein Maschinenprogramm im normalen C64-Betrieb durchführen würde.

Im letzteren Fall würden nämlich verschiedene Speicherstellen auf gewünschte Werte gesetzt werden, vielleicht noch einige Sprungvektoren "umgebogen" werden, und dann würde das Programm in das eigentliche Hauptprogramm verzweigen. Dieses Hauptprogramm würde dann den restlichen Ablauf steuern.

Unter GEOS muß eine Initialisierungsroutine die Adressen verschiedener Unterprogramme für bestimmte Ereignisse an das KERNAL übergeben und dann mit einem RTS beendet werden. Das ist ein entscheidender Unterschied. Daher wollen wir diesen Vorgang noch einmal an einem Beispiel erläutern. Wir schreiben diesmal eine Anwendung (Application). Diese soll Pull-Down-Menüs haben und Tastatureingaben verarbeiten. Wie würde die Initialisierung eines solchen Programms ablaufen? Wie würde GEOS dieses Programm starten?

Wenn unser selbsterstelltes Application die üblichen GEOS-Möglichkeiten nutzt, so taucht es als Icon auf der Benutzeroberfläche auf. Wenn nun ein Anwender dieses Programm doppelklickt oder mit OPEN lädt, schaut GEOS (genauer: hier das KERNAL) erst einmal nach, um was für eine Art Programm es

sich da handelt. Nachdem "Application" festgestellt wurde, wird eine Routine des KERNAL angesprungen, die für das Laden von Anwendungen zuständig ist.

Die Laderoutine des KERNAL holt die Ladeadresse aus dem INFO-Sektor und lädt das Programm an die eingetragene Adresse. Anschließend wird die Einsprungadresse geholt und mit einer "JSR Einsprungadresse" das Programm angesprungen. Ab dieser Einsprungadresse initialisiert sich unsere Application auf folgende Weise: Zuerst wird eine Routine angesprungen, die ein Menü aufbaut und dessen Benutzung vorbereitet (DoMenu, siehe Sprungtabelle). An diese Routine übergibt unser Programm eine Reihe von Parametern, die Aufbau, Lage und Texte des Menüs enthalten. Zusätzlich werden für jeden Menüpunkt Werte übergeben, die bestimmen, ob eine bestimmte Unterroutine beim Anklicken dieses Menüs angesprungen oder ein weiteres Untermenü aufgebaut werden soll.

Zusätzlich schreibt unser Programm die Adresse einer Unterroutine in zwei Speicherstellen, die beim Drücken einer Taste angesprungen werden soll. Anschließend endet unser Programm mit einem RTS. Dadurch wird also wieder zur Laderoutine zurückgekehrt und die Kontrolle an das GEOS KERNAL übergeben. Trotzdem ist unser Programm damit nicht beendet, weil es sich ja über ganz bestimmte Speicherstellen mit dem GEOS KERNAL verbunden hat.

Wenn der Anwender nun mit der Maus das Menü anklickt, baut das KERNAL es den übergebenen Parametern entsprechend auf. Wird ein Menüpunkt angeklickt, holt sich GEOS KERNAL aus den übergebenen Werten die Einsprungadresse und springt diese mit einem JSR an. Ebenso holt es bei einer gedrückten Taste die übergebene Adresse der gewünschten Unterroutine, schreibt den Tastencode in eine Speicherstelle und springt unser Unterprogramm an.

Nur wenn Programme in dieser Weise mit GEOS KERNAL zusammenarbeiten, kann GEOS seine Möglichkeiten voll entfalten. Beispielsweise kann jedes Accessory zugeladen werden, und anschließend macht unsere Anwendung ohne Datenverlust weiter.

Dazu liest unser Programm alle File-einträge von der Diskette und prüft diese auf Filetyp "5". Alle File-Einträge mit diesem Filetyp legen wir in einer Liste ab und zeigen die zugehörigen Filenamen in einem Untermenü an. Wenn nun ein Anwender den dritten Menüpunkt dieses Untermenüs anklickt, holen wir uns den dritten Filenamen aus der Tabelle und übergeben ihn an eine Routine des KERNAL.

Dieses rettet den Speicher, den das Accessory benutzt, auf die Diskette, lädt und startet es. Wenn die Arbeit mit dem Accessory beendet wird, holt das KERNAL den geretteten Speicher wieder von der Diskette und kehrt mit einem RTS zu unserer Application zurück.

Natürlich gibt es dabei eine Reihe von Funktionen, die zusätzlich ausgeführt werden. Beispielsweise rettet das KERNAL den kompletten Aufbau unseres Menüs, damit ein mögliches Menü des nachgeladenen Accessories nicht unser Menü zerstört. An dieser Stelle wollten wir Ihnen aber einen etwas vereinfachten und damit übersichtlichen Überblick über die Funktionsweise geben.

Um unser Programm zu beenden, sehen wir einen entsprechenden Menüpunkt QUIT mit zugehöriger Unterroutine vor. Wird dieser Menüpunkt angeklickt und unsere Routine angesprungen, so treffen wir alle Vorbereitungen für das Programmende (Datenspeicherung) und beenden unser Programm durch Aufruf der Routine "DESK TOP nachladen" aus der Sprungtabelle (JMP \$C22C). Den Rest übernimmt wieder das GEOS KERNAL.

6.3 Eigene Programme in GEOS

In diesem Abschnitt wollen wir verschiedene Möglichkeiten aufzeigen, einerseits eigene Programme von GEOS aus zu laden, andererseits Programm, in GEOS einzubinden und GEOS damit zu erweitern und noch attraktiver zu machen. Das Laden eigener Programme von GEOS aus ist deswegen besonders interessant,

weil dabei unter gewissen Umständen die schnellen Laderoutinen von GEOS benutzt werden können. Was dabei zu beachten ist, wollen wir im ersten Abschnitt behandeln.

6.3.1 Wie eigene Programme geladen werden

Mit GEOS kann man Programme laden, die in BASIC oder Assembler geschrieben sind. Dabei gibt es allerdings einige Unterschiede.

a) BASIC-Programme

Jedes BASIC-Programm können Sie von GEOS aus laden, indem Sie es doppelklicken oder OPEN wählen. GEOS erkennt BASIC-Programme am Filetyp und an der Startadresse \$0801=2049, die im ersten Datenblock hinter dem Zeiger, der auf den folgenden Datenblock zeigt, steht. Danach prüft GEOS, ob das Programm eine bestimmte Größe nicht überschreitet. Die schnellen Laderoutinen von GEOS benutzen den Speicher ab \$8000 im C64. Wenn das BASIC-Programm diesen Bereich nicht erreicht, wird es mit diesen Routinen geladen und automatisch gestartet. Das wird wohl in den meisten Fällen gewährleistet sein, so daß Sie in den Genuß des FAST LOAD kommen.

Sollte das Programm zu groß sein, so wird es vom normalen Betriebssystem des C64 geladen. Das geht natürlich langsamer. Aber auch in diesem Fall genügt es, das zugehörige Icon doppelzuklicken, und alles weitere macht GEOS.

b) Assembler-Programme

Programme, die in Assembler geschrieben sind, erkennt GEOS an einer Startadresse, die von \$0801 abweicht. Diese Programme können nur geladen werden, wenn die Startadresse unter \$0400 liegt. Solche Programme benutzen nämlich üblicherweise den Stack als Autostartbereich. Leider werden sie mit den langsamen DOS-Routinen geladen.

Mit einem Trick ist es aber möglich, trotzdem auch für Assembler-Programme die schnellen GEOS-Routinen zu verwenden. Dazu verwandeln Sie das Programm einfach in ein BASIC-Programm, indem Sie es mit einer SYS-Zeile versehen. Die dort angegebene Adresse muß dabei auf eine Verschieberoutine zeigen, die das Maschinenprogramm an die korrekte Adresse kopiert und anschließend startet.

\$0801=2049: 10 SYS 2080

\$0820=2080: Verschieberoutine, die das Programm in den Bereich verschiebt, für den es geschrieben wurde.

\$0850=2128: eigentliches Programm

Auf diese Weise können also auch Assembler-Programme mit den schnellen Laderoutinen geladen und anschließend automatisch gestartet werden.

6.3.2 Wie man GEOS durch eigene Programme erweitert

Es gibt zwei Möglichkeiten, eigenständige Programme in GEOS einzubinden. Beide Möglichkeiten verlangen natürlich eine ausreichende Kenntnis der internen Vorgänge in GEOS.

a) Anwendungen (Applications)

Beginnen wir mit der einfacheren Methode. GEOS soll ein von Ihnen geschriebenes Programm laden, starten und anschließend, wie z.B. bei GEOWRITE, zum DESK TOP zurückkehren. Um das zu erreichen, müssen Sie eine eigene Anwendung schreiben. Sie muß natürlich in Assembler geschrieben sein, weil unter GEOS der BASIC-Interpreter nicht verfügbar ist. Außerdem müssen die Bildschirmausgaben auf dem HIRES-Bildschirm (ab \$A000) erfolgen. Wie wird ein solches Programm nun in GEOS eingefügt?

Bei der Programmplanung sollten Sie den Bereich von \$0400-\$7FFF für Ihr Programm vorsehen. Der darüberliegende Bereich wird von GEOS benötigt. Folgende Schritte sind dann erforderlich, um das Programm in eine Anwendung zu verwandeln:

1. Wenn das Programm enden und die Kontrolle wieder an GEOS zurückgeben soll, können Sie dafür die Adresse \$C22C anspringen. Dadurch wird erreicht, daß das DESK TOP korrekt nachgeladen und gestartet wird.
2. Speichern Sie das Programm ab. Jetzt muß es in eine Anwendung umgewandelt werden. Am besten benutzen Sie dafür stets eine besondere Arbeitsdiskette, auf der nichts Wichtiges bei einem möglichen Fehler verlorenght. Von dieser Arbeitsdiskette können Sie es dann später auf die gewünschte Diskette kopieren.
3. Jetzt muß die eigentliche Umwandlung in das Format einer Anwendung erfolgen. Sie können das mit einem Disketten-Monitor oder einfacher mit dem Programm Filemaster, das in diesem Buch abgedruckt ist, erreichen.

Folgende Ergänzungen müssen vorgenommen werden:

- Ihr Programm muß einen INFO-Sektor erhalten. Track und Sektor müssen hinter dem letzten 'A0' des Filenamens stehen.
- Die nächsten beiden Bytes ändern Sie bitte in '0' und '6'. Dabei bedeutet das erste Byte Filestruktur=SEQ (\$00) und das zweite Byte GEOS-Filetyp=Anwendung (\$06).

(Mit dem Filemaster erreichen Sie diese Änderung, indem Sie 'FE' wählen, Ihr Programm einlesen lassen und danach vom Menü aus 'GEOS' wählen und als GEOS-Filetyp '6' wählen. Vorher können Sie noch ein Datum eingeben.)

- Im INFO-Sektor müssen die Bytes im Floppypuffer auf folgende Werte gesetzt werden (Puffer 0=\$0300-\$03FF):

```
Byte $0344=C64-Filetyp ($83=USER)
Byte $0345=GEOS-Filetyp (=6)
Byte $0346=Filestruktur (=0=SEQ)
Byte $0347=LOW-Byte-Ladeadresse
Byte $0348=HIGH-Byte-Ladeadresse
```

Dabei müssen Sie für die Ladeadresse einen um 2 kleineren Wert eintragen als die Startadresse des Programms. Beispiel: Tragen

Sie \$0FFE ein, wenn das Programm bei \$1000 beginnt. Das liegt daran, daß GEOS die Bytes 2,3 im ersten Block des Programms nicht wie das alte DOS als Ladeadresse, sondern schon als Programm ansieht. Wenn Sie die Startadresse des Programms als Ladeadresse eintragen würden, würde Ihre Anwendung um zwei Byte verschoben im Speicher stehen, die beiden ersten Bytes (\$1000, \$1001) wären dann die Bytes der Ladeadresse vom C64-DOS (\$00, \$10)

Byte \$0349=Endadresse des Programms (LOW)
Byte \$034A=Endadresse des Programms (HIGH)

Bei Anwendungen wird diese Adresse nicht benutzt. Tragen Sie hier aber trotzdem einen Wert ein, das machen die Anwendungen GEOWRITE und GEOPAINT auch. Dabei steht in diesen beiden Bytes nicht die wirkliche Endadresse, sondern die eingetragene Ladeadresse-1. In unserem Beispiel also \$0FFD. Die Differenz zwischen Ladeadresse und eingetragener Endadresse wird von GEOS nämlich bei Accessories als Swap-File auf der Diskette zwischengespeichert. Wenn Sie den eben beschriebenen Wert eintragen, unterbleibt das in jedem Fall.

Byte \$034B=Einsprungadresse (LOW)
Byte \$034C=Einsprungadresse (HIGH)

Auch diese Arbeit, die richtigen Werte in den INFO-Sektor einzutragen, erledigt der Filemaster für Sie, wenn Sie nach GEOS im Hauptmenü INFO wählen. Die Angaben erfolgen dezimal. Denken Sie bitte bei der Ladeadresse an den um 2 verminderten Wert.

Nachdem Sie die Änderungen jeweils auf der Diskette gespeichert haben, ist die Anwendung fertig. Sie können sie nun von GEOS aus benutzen. Wir möchten an dieser Stelle noch ein einfaches Beispiel abdrucken. Das folgende Programm 'Blinken' läßt den Bildschirmrahmen für einige Sekunden in immer schnellerer Folge blinken und lädt dann DESK TOP nach.

```
4590 LDA $01
4592 PHA
4593 LDA #$37
```

```

4595 STA $01
4597 LDA #$80
4599 TAX
459A TAY
459B STA $D020
459E DEX
459F BNE $459E
45A1 DEY
45A2 BNE $459E
45A4 TAX
45A5 DEX
45A6 BEQ $4580
45A8 TXA
45A9 TAY
45AA JMP $459B
45AD NOP
45AE NOP
45AF NOP
45B0 PLA
45B1 STA $01
45B3 CLI
45B4 JMP $C22C
    
```

In diesem Beispiel setzen Sie bitte im INFO-Sektor folgende Werte ein:

```

Ladeadresse      =$458E
Endadresse       =$458D
Einsprungsadresse=$4590
    
```

b) Hilfsmittel (Accessories)

Der große Vorteil von Hilfsmitteln ist, daß sie aus jeder Anwendung heraus geladen werden können. Wird die Arbeit mit dem Hilfsmittel beendet, arbeitet die Anwendung an der Stelle weiter, an der sie unterbrochen wurde. Das wird nur dadurch möglich, daß GEOS bestimmte Programmteile auf der Diskette als Swap-File (Austausch-File) ablegt. Wie funktioniert das?

Bevor ein Hilfsmittel geladen wird, ermittelt GEOS die Anfangs- und Endadresse. Dieser Bereich wird dann als Swap-File auf der Diskette gespeichert. Danach wird das Hilfsmittel geladen und kann gestartet werden. Soll nun das Hilfsmittel verlas-

sen werden, so lädt GEOS das Swap-File wieder an die richtige Stelle, und das vorherige Programm, die Anwendung, ist wieder im ursprünglichen Zustand.

Sie können das selbst einmal testen. Booten Sie GEOS und starten Sie das Weckerprogramm. Nachdem es geladen worden ist, verlassen Sie GEOS, indem Sie den Rechner abschalten oder (falls vorhanden) die RESET-Taste betätigen. Wenn Sie sich jetzt einmal das Inhaltsverzeichnis der Diskette ansehen (LOAD "\$,8 (RETURN) LIST (RETURN)) werden Sie eine neue Eintragung mit dem Namen "Swap-File" entdecken. Wenn Sie mit einem Disketten-Monitor den INFO-Sektor des Swap-Files anschauen, werden Sie feststellen, daß Lade- und Endadresse genau mit denen des Weckers übereinstimmen.

Anschließend löschen Sie bitte von GEOS aus das Swap-File wieder. Sollten Sie schon im Besitz einer Version GEOS V1.3 sein, werden Sie solch ein Swap-File im DESK TOP nie zu Gesicht bekommen, da diese Dateien bei jedem Öffnen der Diskette anhand des speziellen Filetyps (13=Temporary File) erkannt und sofort gelöscht werden. Wie erstellt man nun eigene Hilfsprogramme? Dazu sind grundsätzlich die gleichen Schritte wie beim Erstellen einer Anwendung notwendig. Folgende Änderungen müssen erfolgen:

- GEOS-Filetyp='5'
- Endadresse im INFO-Sektor auf Endadresse des Programms setzen, also nicht auf Ladeadresse-1.
- Ausstieg aus dem Hilfsmittel nicht mit "JMP \$C22C". Dieses Programmende ist aus zwei Gründen nicht möglich. Erstens soll das Hilfsmittel ja auch von GEOPAINT aus geladen werden können, und das Programmende mit JMP \$C22C würde einfach DESK TOP nachladen (und damit gleichzeitig GEOPAINT überschreiben). Zweitens würde jedesmal das vom Betriebssystem automatisch angelegte Swap-File ungenutzt auf der Diskette verbleiben.

Für das Beenden eines Accessories ist eine bestimmte Adresse der Sprungtabelle vorgesehen. Wird diese Adresse am Ende des Programms als Job in den Vektor der erweiterten Hauptschleife

geschrieben (\$849B/\$849C) und anschließend das Hilfsprogramm durch RTS verlassen, wird das Swap-File wieder von der Diskette geladen und anschließend gelöscht. Nun steht die Anwendung oder auch das DESK TOP unverändert zur Verfügung. Die Einsprungadresse für das KERNAL lautet \$C23E. Das Ende einer Anwendung muß damit folgendermaßen aussehen:

```
LDX #$C2      (Einsprungadresse)
LDY #$3E
STX $849C     (als Job übergeben)
STY $849B
RTS           (zurück zur Hauptschleife)
```

Der Rest wird automatisch vom KERNAL erledigt.

6.4 Die GEOS-Sprungtabelle

Erinnern Sie sich vielleicht noch daran, als Sie zum erstenmal eine Beschreibung der C64-KERNAL-Sprungtabelle in den Händen hielten. Plötzlich wurde das Programmieren wesentlich vereinfacht. Viele nützliche Routinen brauchten nicht mehr im eigenen Programm aufwendig programmiert zu werden, sondern wurden einfach im KERNAL angesprungen.

Zusätzlich bietet eine solche Sprungtabelle noch einen entscheidenden Vorteil: Verschiedene Versionen eines Betriebssystems beispielsweise bei COMMODORE-Rechnern sind insofern kompatibel, als die Einsprungadressen in der Sprungtabelle auf äquivalente Unterprogramme zeigen.

Eine solche Sprungtabelle gibt es auch in GEOS. Sie liegt im Speicher ab \$C100. Über diese Tabelle können alle wichtigen Aktionen des GEOS KERNAL benutzt werden. Zusätzlich besteht eine große Wahrscheinlichkeit, daß bei neuen Versionen des GEOS KERNAL die Routinen der Sprungtabelle problemlos weiterbenutzt werden können. So haben sich beispielsweise von GEOS V1.1 über V1.2 zu GEOS V1.3 viele Routinen geändert oder liegen an anderen Adressen. Die Tabelleneinträge zeigen aber immer noch auf die gleichen Routinen.

Auf den folgenden Seiten finden Sie nicht nur die Sprungtabelle abgedruckt, sondern auch die Funktion und Übergabeparameter der Routinen. Wenn Sie sich einen disassemblierten Ausdruck des GEOS KERNAL erstellen, können Sie sich diese Routinen genau anschauen und damit problemlos arbeiten. Halten Sie sich bei eigenen Programmen unbedingt an die Sprungtabelle, damit Ihr Programm auch bei neueren Versionen von GEOS voll funktionstüchtig bleibt.

C100 InterruptMain

Hier steht ein Sprung zum GEOS-IRQ. Normalerweise erledigt GEOS die IRQ-Abarbeitung von selbst. Ein solcher IRQ findet alle 1/50 Sekunde statt (Rasterzeilen-Interrupt). Dabei werden die folgenden Arbeiten erledigt:

1. Maus bewegen und deren Bereich testen.
2. Timergruppe A dekrementieren (Multi-Tasking-Jobs).
3. Timergruppe B dekrementieren (Delay-Jobs).
4. Cursor bedienen, falls das 7. Bit in \$84B4 gesetzt ist ("alphaFlag").
5. In \$850A (LOW) und \$850B (HIGH) eine Zufallszahl erzeugen. Diese Werte können durch JSR GetRandom ausgelesen werden (GetRandom=\$C187).

C103 InitProcesses

Jobs vom Typ A (s. InterruptMain) in Jobpuffer 1 schreiben.

- Akku : Anzahl der Jobs (maximal 20 !)
- \$02,\$03 : Adresse der Tabelle

Diese Tabelle muß folgenden Aufbau haben:

Jobadresse (Job 1)	LOW
Jobadresse	HIGH

Timerwert	LOW
Timerwert	HIGH
Jobadresse (Job 2)	LOW
Jobadresse	HIGH
Timerwert	LOW
Timerwert	HIGH

und so weiter. In diese Tabelle trägt man Jobs ein, die immer wieder ausgeführt werden sollen. Der zugehörige Timer-Wert entscheidet über die Häufigkeit der Ausführung. Dazu wird vom GEOS KERNAL jedem Job ein Status-Byte zugeordnet, mit dem Jobs aktiviert und inaktiviert werden können. Der IRQ zählt die zugehörigen Timer-Werte (sie bilden die Timer-Gruppe "A", siehe \$C100) auf "0" herunter und gibt den Job dann zur Abarbeitung durch die GEOS-Main-Loop (s. \$C1C3) frei. Somit können rein rechnerisch Timer-Werte zwischen 1/50 Sekunde (Timer-Wert=\$0001) und ca. 21 Stunden, 50 Minuten (Timer-Wert=\$FFFF) programmiert werden. Die einzelnen Bits im Status-Byte haben folgende Bedeutung:

- Bit 5* Wenn dieses Bit gesetzt ist, zählt der IRQ den zugehörigen Timer nicht weiter herunter. Der Job ist damit auf dem augenblicklichen Stand eingefroren (FREEZE).
- Bit 6* Verhindert, daß der Job ausgeführt wird, auch wenn der Timer-Wert abgelaufen ist (BLOCK).
- Bit 7* Wird vom IRQ als Zeichen dafür gesetzt, daß der zugehörige Timer abgelaufen ist. Dadurch kann der Job von der Hauptschleife abgearbeitet werden. Diese setzt sofort Bit 7 zurück (Disable) und startet den zugehörigen Prozeß. Durch das Zurücksetzen wird eine erneute Abarbeitung verhindert (siehe auch \$C109=EnableProcess).

Durch InitProcesses werden die Jobs nur eingetragen. Da aber hierbei Bit 5 (FREEZE) gesetzt wird, werden vom IRQ noch nicht die Timer gezählt. Dazu muß jeder Job erst zur Bearbeitung freigegeben werden (RestartProcess=\$C106).

Wichtig ist, daß man diese Jobs alle auf einmal eintragen muß. Ein nachträgliches Zufügen von Jobs ist nicht vorgesehen. Damit wird diese Routine immer zu dem Initialisierungsteil eines Programms gehören.

C106 **RestartProcess**

Löscht Bit 5 und Bit 6 im Status-Byte und trägt die Timer-Werte ein. Dadurch wird jede Blockierung aufgehoben (UNFREEZE und UNBLOCK). Die zugehörige Jobnummer wird im X-Register übergeben.

C109 **EnableProcess**

Obwohl die Jobs normalerweise durch heruntergezählte Timer gestartet werden, kann jeder Job durch EnableProcess auch "von Hand" gestartet werden. Der Timer-Wert des durch das X-Register bezeichneten Jobs wird sofort auf Null gesetzt. Beim nächsten Durchlauf der Hauptschleife wird dann dieser Job ausgeführt.

C10C **BlockProcess**

Setzt Bit 6 im Status-Byte des im X-Register übergebenen Jobs. Dadurch werden zwar die Timer-Werte weiter dekrementiert, wird dieser Job aber bei "0" dennoch nicht ausgeführt.

C10F **UnblockProcess**

Gegenstücke zu BlockProcess. Jobblockierung durch Löschen von Bit 6 im Status-Byte aufheben. Die Jobnummer muß im X-Register übergeben werden. Mit der Länge des blockierten Zeitraumes wächst die Wahrscheinlichkeit, daß inzwischen die

Timer-Werte vom IRQ heruntergezählt worden sind. In diesem Fall wird der Job beim nächsten Durchlauf der MainLoop abgearbeitet.

C112 FreezeProcess

Durch das Setzen des Bit 5 wird der Prozeß mit seinem aktuellen Timer-Stand eingefroren. Das X-Register muß die Jobnummer enthalten.

C115 UnfreezeProzess

Löscht Bit 5 im Statusbyte des im X-Register übergebenen Jobs. Dadurch werden die vorher gestoppten Timer-Werte weitergezählt. Auch beim Aufruf dieser Routine muß das X-Register die Jobnummer enthalten.

Grafik-Bibliothek:

Bei der Benutzung der Grafikroutinen hat das Byte \$2F entscheidend Einfluß darauf, wohin Informationen geschrieben oder von wo sie geholt werden. Abhängig von \$2F werden nämlich die Zeiger auf die einzelnen Bildschirme gesetzt. Folgende Werte für \$2F sind vorgesehen:

- \$80 = Ausgabe nur auf den sichtbaren Bildschirm (\$A000)
 - \$40 = Ausgabe nur auf den unsichtbaren Bildschirm (\$6000)
 - \$C0 = Ausgabe auf beide Bildschirme (\$C0=\$80 OR \$40 !)
-

C118 HorizontalLine

Zieht eine waagerechte Linie auf dem hochauflösenden Grafikbildschirm (vorher in \$2F den Bildschirm auswählen!).

Parameter

Akku: Die Bits bestimmen direkt das Strichmuster
 \$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)
 \$18: Y-Position der Linie

X-Werte: (0-319)

Y-Wert: (0-199)

C11B**InvertLine**

Invertiert eine waagerechte Linie auf dem Grafikbildschirm (vorher in \$2F den Bildschirm auswählen!).

Parameter

\$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)
 \$18: Y-Position der Linie

X-Wert: (0-319)

Y-Wert: (0-199)

C11E**RecoverLine**

Holt eine Linie aus dem zweiten Bildschirm (ab \$6000) auf den sichtbaren Bildschirm zurück. Wird beispielsweise benutzt, um beim Abbau von Menüs den ursprünglichen Bildschirm wiederherzustellen.

\$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)
 \$18: Y-Position der Linie

C121 VerticalLine

Zieht eine senkrechte Linie (vorher in \$2F den Bildschirm auswählen!).

Akku: Muster der Linie
 \$08: oberere Grenze
 \$09: untere Grenze
 \$0A: X-Position LOW
 \$0B: X-Position HIGH

C124 Rectangle

Füllt ein Rechteck mit einem Muster (vorher in \$2F den Bildschirm auswählen!).

\$06: obere Grenze
 \$07: untere Grenze
 \$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)
 \$22,\$23: Zeiger auf das Muster

Das Muster muß aus 8 Bytes bestehen und somit eine 8*8-Punkte-Matrix ergeben. Durch vorherigen Aufruf der Routine SetPattern (\$C139) kann auf einfache Art eins der 34 fertigen Muster des KERNAL vorgegeben werden. Dazu muß SetPattern im Akku die Nummer des gewünschten Musters (0 - 33) übergeben werden.

C127 FrameRectangle

Malt einen rechteckigen Rahmen (vorher in \$2F den Bildschirm auswählen!).

Akku: die Bits des Akkus ergeben das Muster
 \$06: obere Grenze

\$07: untere Grenze
 \$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)

C12A **InvertRectangle**

Invertiert ein Rechteck (bitte vorher in \$2F den Bildschirm auswählen!).

\$06: obere Grenze
 \$07: untere Grenze
 \$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)

C12D **RecoverRectangle**

Holt ein Rechteck aus dem zweiten Bildschirm zurück. Wird beispielsweise benutzt, um beim Abbau von Fenstern den ursprünglichen Bildschirminhalt wiederherzustellen.

\$06: obere Grenze
 \$07: untere Grenze
 \$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)

C130 **DrawLine**

Diese Routine zieht eine beliebig orientierbare Linie. Sie ist daher wesentlich flexibler als "HorizontalLine" oder "VerticalLine", aber auch erheblich langsamer. Vor ihrem Aufruf muß in \$2F der Bildschirm ausgewählt werden.

\$08: X-Position Punkt 1 LOW
 \$09: X-Position Punkt 1 HIGH
 \$0A: X-Position Punkt 2 LOW

\$0B: X-Position Punkt 2 HIGH
\$18: Y-Position Punkt 1
\$19: Y-Position Punkt 2

Zusätzlich bestimmen folgende Status-Bits beim Einsprung die Ausführung dieser Funktion:

Bit 7 (Negativ-Flag)=1: Dann ist Bit 0 (Carryflag) ohne Bedeutung. In diesem Fall wird die Linie vom zweiten, unsichtbaren Bildschirm auf den sichtbaren Bildschirm kopiert.

Mit dieser Funktion kann GEOPAINT ein UNDO realisieren. Die aktuelle Aktion erfolgt nur auf dem sichtbaren Bildschirm. Soll sie widerrufen werden, wird dieselbe Funktion noch einmal in diesem Modus (Bit 7=1) aufgerufen. Dadurch wird genau der veränderte Teil des Bildschirms vom zweiten Bildschirm geholt und in den sichtbaren kopiert.

Bit 7 (Negativ-Flag)=0: In diesem Fall entscheidet das Carryflag folgendermaßen:

Carryflag=1: Zieht eine Linie
Carryflag=0: Löscht eine Linie

Bei diesem Modus muß wieder der gewünschte Bildschirm in \$2F eingetragen werden.

C133**DrawPoint**

Diese Routine zeichnet einen Punkt oder kopiert ihn vom Hintergrundbildschirm auf den sichtbaren Bildschirm. Es gelten die gleichen Bedingungen wie bei "DrawLine" (Status-Bits, \$2F).

\$08: X-Position Punkt LOW
\$09: X-Position Punkt HIGH
\$18: Y-Position Punkt

C136 GraphicsString

Führt vorher definierte Grafik-Jobs aus einer Tabelle aus. In \$02,\$03 wird ein Zeiger auf eine Tabelle übergeben, die die gewünschten Jobnummern enthält. Hinter der Jobnummer folgen jeweils direkt die nötigen Daten für den Job. Das Ende der Tabelle muß durch die Jobnummer Null gekennzeichnet werden.

Jobnummer 0=RTS=Ende

Jobnummer 1=GetStartPoint

Funktion: 3 Bytes holen und nach \$87D4 (X- LOW),\$87D5(X-HIGH),\$87D6 (Y) schreiben. Diese Koordinaten werden bei den folgenden Jobs als Ausgangspunkt benutzt. Beispielsweise muß bei der folgenden Routine Jobnummer 2 zuerst Job (1) aufgerufen werden, dann kann DrawLineToPoint benutzt werden.

Jobnummer 2=DrawLineToPoint

Funktion: Die vorher von Job 1 eingetragenen Koordinaten nach \$08, \$09, \$18 übernehmen und Job 1 ausführen, um die Koordinaten eines weiteren Punktes zu holen (drei Bytes). Dann diese neuen 3 Bytes nach \$0A, \$0B, \$19 schreiben. Anschließend wird die Routine DrawLine mit gesetztem Carry- und gelöschtem V-Flag aufgerufen. (siehe \$C130)

Jobnummer 3=Rectangle

Funktion: Holt eine zweite Koordinate (3 Bytes). Berechnet aus den beiden Punkten ein Rechteck und füllt dieses mit dem derzeit aktuellen Muster.

Voraussetzung für den Aufruf von Job 3: Mit Job 1 die Koordinaten des ersten Eckpunktes festlegen und mit Job 5 das gewünschte Muster auswählen. Dann kann FillArea aufgerufen werden.

Beispiel für eine Tabelle, die mit Job 3 ein Rechteck von P1 zu P2 mit einem Muster füllt:

\$03,02=Zeiger auf Tabelle=\$1000
Anschließend: JSR GraphicsString

Daten in der Tabelle:

\$1000 - \$01 GetStartPoint (P1)
\$1001 - \$0A X-LOW von P1
\$1002 - \$00 X-HIGH von P1
\$1003 - \$10 Y von P1
\$1004 - \$05 SetPattern
\$1005 - \$01 Musternummer (schwarz)
\$1006 - \$03 FillArea
\$1007 - \$50 X-LOW von P2
\$1008 - \$00 X-HIGH von P2
\$1009 - \$60 Y von P2
\$100A - \$00 Tabellenende

Job 4=NOP

Dieser Job bewirkt nichts. Danach wird die Tabelle einfach weiter abgearbeitet.

Job 5=SetPattern

Funktion: Das folgende Byte gibt die Nummer des gewünschten Musters an (0-33). Über das Aussehen der Muster können Sie sich bei GEOPAINT in der Musterleiste informieren.

Job 6=PutString

Funktion: Gibt an einer bestimmten Position einen Text aus. Benötigt werden drei Bytes für die Position: X-LOW, X-HIGH, Y und ein folgender 2-Byte-Zeiger auf den Text. Der Text muß mit "0" abgeschlossen sein.

Job 7=FrameRectangle

Funktion: Einen rechteckigen Rahmen zeichnen. Benötigt werden wieder die drei Bytes, die die Position des zweiten Punktes P2 angeben. P1 muß durch Jobnummer 1 gesetzt sein.

Job 8=XOffset

Funktion: Erhöht die X-Koordinaten von P1 um den angegebenen Wert. Benötigt werden die zwei Bytes X-LOW, X-HIGH.

Job 9=YOffset

Funktion: Erhöht die Y-Koordinaten von P1 um den angegebenen Wert. Benötigt wird ein Byte (Y).

Job 10=XYOffset

Funktion: Wie Job 8 und Job 9. Benötigt werden drei Bytes X-LOW, X-HIGH, Y.

C139**SetPattern**

Zeiger auf gewünschtes Muster setzen. Die Nummer des Musters wird im Akku übergeben (0-33). Über das Aussehen der Muster können Sie sich bei GEOPAINT in der Musterleiste informieren.

C13C**GetScanLine**

Berechnet die Startadresse der Zeile, deren Nummer (0-199) im X-Register übergeben wird. Das Ergebnis steht in \$0C/\$0D für den sichtbaren und in \$0E/\$0F für den unsichtbaren Bildschirm (vorher in \$2F den/die Bildschirm(e) auswählen!).

C13F **TestPoint**

Testet einen Grafikpunkt.

\$08,\$09 = X-Position

\$18 = Y-Position

Ist der Punkt gesetzt, wird das Carryflag gesetzt, andernfalls gelöscht. \$2F entscheidet, auf welcher Bildschirmseite getestet wird.

C142 **BitmapUp**

Zeichnet ein Bild beliebiger Größe. Mit dieser Routine werden beispielsweise die Icons im DESK TOP gezeichnet.

Es müssen folgende Parameter übergeben werden:

\$02,\$03 = Adresse der Daten des Bildes

\$04 = X-Position (0-39)

\$05 = Y-Position (0-199)

\$06 = Breite des Bildes in Bytes

\$07 = Höhe des Bildes in Zeilen

Die X-Position kann also nur in 8-Bit-Schritten angegeben werden und wird automatisch mit 8 multipliziert. Die Grafikdaten müssen in einer für GEOS spezifischen Art und Weise komprimiert vorliegen. Ein typisches Beispiel hierfür sind die Daten eines Icons im Infosektor einer Datei. Hier die notwendigen Informationen zu dem benutzten Format:

Grafikdaten werden in GEOS nicht linear abgelegt, da dies viel zu viel Speicher in Anspruch nehmen würde. Statt dessen wird die Tatsache ausgenutzt, daß in den meisten Grafiken ganze Bytefolgen gleichartig aufgebaut sind. Diese Bytefolgen können nun aber zusammengefaßt werden, wodurch man je nach Bildinhalt einige Kilobyte Speicher einsparen kann. Dies funktioniert jedoch nur dann rentabel, wenn man sich von dem durch die

Hardware des 64er festgelegten Zusammenhang zwischen Grafikdaten und Speicherstellen löst und die Daten zeilenweise zusammenfaßt. Die Hardware des 64er faßt nämlich immer 8 im Speicher aufeinanderfolgende Bytes zu einem auf dem Bildschirm untereinanderliegenden Grafikblock zusammen. Man kann sich nun leicht vorstellen, daß die Grafikdaten einer waagerechten Grafik-Zeile eher ein sich wiederholendes Muster aufweisen, als die Daten mehrerer solcher untereinanderliegenden Achter-Blocks.

Um Daten komprimieren zu können, benötigt man bestimmte Steuerbytes, die sich mit den Grafikdaten abwechseln. In GEOS werden Grafikdaten durch ein Steuerbyte eingeleitet, das folgende Werte annehmen kann:

- 0:** Ende der Grafikdaten
- 1-127:** Der angegebene Wert entscheidet darüber, wie oft das als nächstes angegebene Byte wiederholt dargestellt werden soll. Im Extremfall kann hiermit also auch wieder eine unkromprimierte Grafik abgelegt werden. Dazu muß sich das Steuerbyte mit dem Wert "1" immer abwechseln mit einem Byte der Grafikdaten. Dadurch würde sich natürlich der benötigte Speicherplatz genau verdoppeln. Tauchen aber in einer Grafik große, ungemusterte Flächen auf, können mit diesem Steuerzeichen maximal 127 Bytes auf nur 2 Bytes komprimiert werden (also auf nur noch ca. 1.6% des ursprünglichen Speicherplatzbedarfes!).
- 128-220:** Der angegebene Wert, vermindert um 128, gibt an, wie viele Daten nun folgen, die direkt als Grafikdaten übernommen werden sollen. Hierdurch können komplizierte Grafiken, die keine sinnvolle Datenkompression erlauben, abgespeichert werden. Der Speicherplatzverlust durch Steuerzeichen wird hierdurch auf ein akzeptables Maß begrenzt. Schließlich wäre es ja auch möglich, solche Daten mit dem Steuerzeichen "1" zu speichern (s.o.).

221-255: Solche Steuerzeichen sind etwas komplizierter aufgebaut. In GEOS hat man es geschafft, sogar Grafikmuster komprimieren zu können, die sich nicht auf ein Byte beschränken. Um dieses Steuerzeichen richtig interpretieren zu können, muß man es erst einmal um 220 vermindern. Der Wert, der dabei übrigbleibt, gibt an, wie viele Bytes das sich wiederholende Muster insgesamt aufweist. Natürlich fehlt noch die Information darüber, wie oft dieses Muster wiederholt werden soll. Aus diesem Grund findet man im nächsten Byte noch keine Grafikdaten, sondern ein weiteres Steuerbyte. Dieses Steuerbyte gibt nun an, wie oft das Muster wiederholt werden soll. Erst jetzt folgen die Grafikdaten, die wiederum (Ihnen bleibt aber auch nichts erspart) in einem der ersten beiden Formate vorliegen müssen. Somit taucht das erste Grafik-Byte erst nach drei Steuerzeichen auf.

C145 **PutChar**

Gibt den Wert im Akku an der aktuellen Position als ASCII-Zeichen aus. Zeichen kleiner als dez. 32 (\$20) werden als Steuerzeichen interpretiert (s. ASCII-Tabelle im Anhang). Ungültige Werte führen zum Absturz des Rechners.

C148 **PutString**

PutString gibt eine Zeichenkette mit dem aktuellen Zeichensatz aus. Die Steuerzeichen zur Änderung des Zeichensatzes (siehe ASCII-Tabelle im Anhang) werden ignoriert. Alle anderen Steuerzeichen werden jedoch akzeptiert.

Parameter

\$02/\$03 Adresse des auszugebenden Textes, der mit Null enden muß.
 \$05 Y-Position für die Ausgabe
 \$18/18 X-Position für die Ausgabe

C14B **UseSystemFont**

Schaltet auf den residenten BSW-9-Point-Font des Betriebssystems um. Es werden keine Parameter benötigt.

C14E **StartMouseMode**

Initialisiert die Mausabfrage.

Ist das Carryflag gesetzt und steht in \$18/\$19 ein Wert ungleich "0", so wird \$18/\$19 als neue Mausposition (X) und der Inhalt des Y-Registers als neue Mausposition (Y) übernommen. Die Mausgeschwindigkeit wird auf 0 gesetzt.

Unabhängig vom Carryflag wird die Menüabfrage initialisiert.

C151 **DoMenu**

Diese Routine erstellt ein komplettes Pull-Down-Menü. Normalerweise gehört ein Aufruf von DoMenu zu jeder Initialisierungsroutine einer Application in GEOS. Dabei müssen die folgenden Parameter übergeben werden:

\$02,\$03 = Zeiger auf eine Datentabelle

Aufbau der Datentabelle (Format des Menüs):

<i>Byte 0</i>	= Y-Position oben
<i>Byte 1</i>	= Y-Position unten
<i>Byte 2</i>	= X-Position links LOW
<i>Byte 3</i>	= X-Position links HIGH
<i>Byte 4</i>	= X-Position rechts LOW
<i>Byte 5</i>	= X-Position rechts HIGH
<i>Byte 6</i>	= Statusbyte (siehe unten)
<i>Byte 7</i>	= Zeiger auf Menübegriff 1 LOW
<i>Byte 8</i>	= Zeiger auf Menübegriff 1 HIGH
<i>Byte 9</i>	= Aufbau-Flag (siehe unten)
<i>Byte 10</i>	= Pointer für Menüpunkt 1 LOW (siehe unten)
<i>Byte 11</i>	= Pointer für Menüpunkt 1 HIGH

Weiter wie ab Byte 7 für alle weiteren Menüpunkte (maximal 15).

Das Statusbyte (Byte 6) existiert nur für das Hauptmenü und hat den folgenden Aufbau:

<i>Bit 0-4:</i>	Anzahl der Hauptmenüpunkte
<i>Bit 6:</i>	=1, Die Maus kann die Menügrenzen nicht verlassen. =0, Maus ist frei beweglich.
<i>Bit 7:</i>	=1, Menüpunkte stehen untereinander, sonst nebeneinander wie im DESK TOP, GEOPAINT, etc.

Die Bits des Aufbau-Flags haben die folgende Bedeutung:

<i>Bit 7:</i>	=1, dann zeigt der Pointer dieses Menüpunktes (also z.B. Byte 10 und 11 für den 1. Hauptmenüpunkt) auf die Daten eines Submenü, das wiederum eine vollständige Menüstruktur aufweisen muß. Auf diese Weise können maximal 3 geschachtelte Untermenüs mit je 15 Einträgen erstellt werden.
<i>Bit 7:</i>	=0 und Bit 6=0, der Pointer dieses Menüpunktes zeigt auf die Einsprungadresse der ausgewählten Routine.

Wichtig!

Nach dem Einsprung in die eigene Routine baut sich das Menü nicht selbständig wieder ab. Dem Anwender wird so in seinem Programm die Möglichkeit gegeben, selbst zu entscheiden, was mit dem derzeitigen Menüstatus geschehen soll. Im einfachsten Fall kann man durch den Aufruf der Funktion GotoFirstMenu=\$C1BD den Abbau aller Untermenüs veranlassen. Soll jedoch das zuletzt angewählte Untermenü noch weitere Auswahlmöglichkeiten zulassen, kann dies durch den Aufruf der Funktion ReDoMenu=\$C193 erreicht werden.

Möchte man jedoch die Auswahl eines weiteren Menüpunktes im derzeit übergeordneten Menü zulassen, kann dies durch den Aufruf der Funktion DoPreviousMenu=\$C190 geschehen. In jedem Fall muß eine dieser Routinen vor dem Ende des eigenen Programms durch RTS aufgerufen werden. Nähere Angaben zu diesem Thema finden Sie auch bei der Behandlung der einzelnen Routinen.

Bit 7: =0 und Bit 6=1, der Pointer des Menüpunktes zeigt auch hier auf eine Einsprungsadresse. Nachdem die ab dort beginnende Routine abgearbeitet worden ist und mit RTS endet, wird ein weiteres Submenü erstellt. Der Zeiger auf die Daten dieses Submenüs muß vor dem RTS in \$02,\$03 eingetragen werden.

Hierdurch können Jobs ausgeführt werden, während das Menü aktiv ist. Damit ist der Programmierer nicht gezwungen, die komplette Menüstruktur bis zum letzten Submenü schon von vornherein festzulegen. Der Font-Editor benutzt diese Funktion zum Beispiel, um die vorhandenen Point-Höhen für den angewählten Zeichensatz zu laden, bevor er sie im Menü anzeigt. Die einzelnen Menübegriffe müssen alle mit "0" abgeschlossen sein.

C154 **RecoverMenu**

Sorgt dafür, daß beim Abbau des zuletzt ausgeklappten Menüs der ursprüngliche Bildschirmaufbau wiederhergestellt wird. Im Normalfall braucht sich der Anwender um diese Funktion nicht zu kümmern, da sie Bestandteil der Routinen ist, die ein Menü abbauen (DoPreviousMenu, GotoFirstMenu). Ist der Vektor \$84B1/\$84B2 gleich null, dann wird beim Abbau eines Menüs der Hintergrund einfach weiß eingefärbt (nicht sehr sinnvoll). Andernfalls muß in den beiden Speicherstellen die Adresse einer Routine stehen, die den Hintergrund wiederherstellt. Bei der normalen Warmstart-Konfiguration von GEOS steht hier ein Sprung auf die Routine RecoverRectangle (siehe auch dort). Diese holt aus dem unsichtbaren Bildschirm die original Grafikdaten des vom Menü überschriebenen Bereiches und kopiert sie in den sichtbaren Bildschirm.

Da ein Menü nur auf dem sichtbaren Bildschirm ausgegeben wird, kann das Bild auf diese Art restauriert werden. Falls aber nicht genügend Platz für eine vollständige Kopie des Bildschirms ab \$6000 vorhanden ist (wie z.B. bei GEOPAINT), kann man den Vektor \$84B1/\$84B2 auch auf eine eigene Routine umbiegen. Diese muß dafür sorgen, daß die vom Menü überschriebenen Daten vor dem Ausklappen in irgendeiner Form gerettet und beim Abbau wieder auf den sichtbaren Bildschirm geschrieben werden. Falls möglich, sollte man jedoch mit einem Hintergrundbildschirm arbeiten, da einem in diesem Fall die Programmierung der Rettungsroutinen erspart bleibt.

C157 **RecoverAllMenus**

Baut nacheinander alle heruntergeklappten Untermenüs ab. Dazu wird RecoverMenu (s.o.) als Unterprogramm benutzt.

Parameter

\$84B1/\$84B2 (siehe RecoverMenu)

C15A

DoIcons

Erstellt anklickbare Bilder wie beispielsweise das Diskettensymbol im DESK TOP. Genauso wie DoMenu wird auch DoIcons in den meisten Fällen Bestandteil einer Programm-Initialisierung sein. Die Icon-Funktion ist ebenso wie die Menüverwaltung in GEOS "event driven". Der Anwender teilt also dem Betriebssystem in einer Tabelle mit, welche Routinen beim Anklicken bestimmter Icons ausgeführt werden sollen. Nach dieser Initialisierung muß das Programm mit einem RTS enden, damit die GEOS-Hauptschleife die Kontrolle übernehmen kann.

Parameter

\$02,\$03 = Adresse der Tabelle (LOW/HIGH)

Aufbau der Tabelle:

Byte 0 = Anzahl der gewünschten Klickfelder
 Byte 1/2 = gewünschte X-Mausposition (LOW/HIGH)
 Byte 3 = gewünschte Y-Mausposition für jedes Klickfeld
 Byte 4/5 = Zeiger auf das Muster (LOW/HIGH)
 Byte 6 = X-Position des linken Randes des Icons (0-39)
 Byte 7 = Y-Position des oberen Randes des Icons (0-199)
 Byte 8 = Breite des Icons in Bytes
 Byte 9 = Höhe des Icons in Zeilen
 Byte 10/11 = Jobadresse bei Klick (LOW/HIGH)

Ab hier wie Byte 4-11 für alle weiteren Icons.

Nach dem Zeichnen der Bilder auf dem Bildschirm werden die Klickfelder aktiviert und ständig auf Anklicken getestet. Sollte ein bestimmtes Icon angeklickt worden sein, so ruft die GEOS-Hauptschleife das in den entsprechenden Stellen eingetragene Unterprogramm auf (für das erste Icon muß diese Adresse also in Byte 10 und 11 der Tabelle stehen). Hier wird nun der gewünschte Vorgang ausgeführt und anschließend mit RTS wieder zur Hauptschleife zurückgekehrt. Innerhalb der Routine können aus den Speicherstellen \$02 und \$03 die folgenden Informationen ausgelesen werden:

- \$02:** Nummer des angeklickten Icons (0 bis N). Dies ist sehr nützlich, falls alle Icons bis auf unterschiedliche Parameter die gleiche Funktion ausführen sollen. In einem solchen Fall kann man nämlich für alle Icons die gleiche Jobadresse eintragen und hier durch Auswertung der Speicherstelle \$02 Icon-spezifische Korrekturen der Parameter vornehmen.
- \$03:** = \$FF, dann ist das Icon zweimal kurz hintereinander angeklickt worden (Doppelklick).
- \$03:** = \$00, dann ist das Icon nur einmal oder aber mit zu großem zeitlichen Abstand zweimal angeklickt worden.

Ferner kann der Anwender durch das Eintragen bestimmter Werte in die Speicherstelle \$84B5 entscheiden, ob, und wenn ja, wie das Anklicken eines Icons dem Benutzer mitgeteilt werden soll:

Bit 6=0 und Bit 7=0: In diesem Fall wird einfach nur die angegebene Routine angesprungen. Am Aussehen des Icons ändert sich also beim Anklicken nichts.

Bit 6=0/1 und Bit 7=1, dann wird das Icon für einen kurzen Augenblick invertiert und anschließend zurückinvertiert. Dieses Verfahren findet zum Beispiel bei den OK- oder CANCEL-Icons Verwendung.

Bit 6=1 und Bit 7=0. Bei dieser Einstellung wird das angeklickte Icon invertiert. Beispiel: File-Icons im DESK TOP.

C15D**DShiftLeft**

Das X-Register muß auf eine 2-Byte-Adresse auf der Zeropage zeigen, deren Inhalt so oft um ein Byte nach links geschoben wird, wie der Inhalt des Y-Registers angibt (gibt jedesmal * 2).

C160 **BBMult**

Multipliziert 2 einzelne Bytes auf der Zeropage. Als Zeiger auf diese Bytes dienen das X- und das Y-Register. Das Low-Byte des Ergebnisses steht anschließend in der durch das X-Register bezeichneten Speicherstelle, das High-Byte befindet sich direkt dahinter.

C163 **BMult**

Multipliziert 2-Byte-Wert (Wort) mit einem Byte.

Parameter

X-Register: Zeigt auf die Zeropage-Adresse, die das Wort enthält.

Y-Register: Zeigt auf die Zeropage-Adresse, die das Byte enthält.

Das 16-Bit-Ergebnis steht anschließend in den durch das X-Register bezeichneten Speicherstellen (LOW/HIGH).

C166 **DMult**

Multipliziert einen 2-Byte-Wert (Wort) mit einem 2-Byte-Wert (Wort).

Parameter

X-Register: Zeigt auf die Zeropage-Adresse, die das erste Wort enthält

Y-Register: Zeigt auf die Zeropage-Adresse, die das zweite Wort enthält.

Das 16-Bit-Ergebnis steht anschließend in den durch das X-Register bezeichneten Speicherstellen (LOW/HIGH).

C169 **DDiv**

Dividiert einen 2-Byte-Wert (Wort) durch einen 2-Byte-Wert (Wort).

Parameter

- X-Register: Zeigt auf die Zeropage-Adresse, die das erste Wort enthält.
Y-Register: Zeigt auf die Zeropage-Adresse, die das zweite Wort enthält.

Das 16-Bit-Ergebnis steht anschließend in den durch das X-Register bezeichneten Speicherstellen (LOW/HIGH). Falls bei der Division ein Rest auftritt, wird dieser in \$12/13 (LOW/HIGH) abgelegt. Damit ist in GEOS auch eine Modulo-Funktion verfügbar.

C16C **DSdiv**

Diese Funktion unterscheidet sich von DDiv nur darin, daß die beiden Argumente als vorzeichenbehaftet (signed) angesehen werden. Damit kann ein 16-Bit-Wort Werte zwischen -32768 und +32767 annehmen, während ein "normales" 16-Bit-Wort (unsigned) zwischen 0 und 65535 liegt (jeweils einschließlich).

Parameter

siehe DDiv

C16F **Dabs**

Berechnet aus einer vorzeichenbehafteten 16-Bit-Zahl ihren Absolutwert.

Parameter

X-Register : Zeigt auf die Zeropage-Adresse, die das vorzeichenbehaftete Wort enthält.

Das 16-Bit-Ergebnis steht anschließend in den durch das X-Register bezeichneten Speicherstellen (LOW/HIGH).

C172 **Dnegate**

Bewirkt eine Vorzeichenumkehr bei einem vorzeichenbehafteten 16-Bit-Wort.

X-Register: Zeigt auf die Zeropage-Adresse, die das vorzeichenbehaftete Wort enthält.

Das 16-Bit-Ergebnis steht anschließend in den durch das X-Register bezeichneten Speicherstellen (LOW/HIGH).

C175 **Ddec**

Vermindert ein nicht vorzeichenbehaftetes 16-Bit-Wort um eins.

X-Register: Zeigt auf die Zeropage-Adresse, die das vorzeichenbehaftete Wort enthält.

Das 16-Bit-Ergebnis steht anschließend in den durch das X-Register bezeichneten Speicherstellen (LOW/HIGH).

C178 **ClearRam**

Füllt einen Speicherbereich mit "0".

Parameter

\$02,\$03 = Anzahl Bytes, die gelöscht werden sollen
\$04,\$05 = Startadresse des zu löschenden Bereiches

C17B **FillRam**

Füllt einen Speicherbereich mit dem Wert aus \$06.

Parameter

\$02,\$03 = Anzahl Bytes
\$04,\$05 = Startadresse
\$06 = Schreibwert

C17E **MoveData**

Transportiert einen Speicherbereich an die angegebene Adresse.

Parameter

\$02,\$03 = Ursprungsadresse
\$04,\$05 = Zieladresse
\$06,\$07 = Anzahl Bytes

C181 **InitRam**

Kopiert Speicherblöcke, die in einer Tabelle näher bezeichnet werden müssen. In \$02,\$03 muß der Zeiger auf diese Tabelle übergeben werden. Diese Routine ist sehr nützlich, um in einer

Initialisierungsroutine bestimmte Speicherstellen mit Startwerten vorzubelegen.

Aufbau der Tabelle

Byte 0/1 = Zieladresse (LOW/HIGH)
 Byte 2 = Anzahl der Bytes in diesem Block
 Byte 3 = ab hier steht der Datenblock

Hinter dem Datenblock geht es weiter wie bei Byte 0 für den nächsten Block. Das Ende der Tabelle muß durch die Zieladresse \$0000 gekennzeichnet werden.

C184

PutDecimal

Mit Hilfe dieser Routine kann eine 16-Bit-Dezimalzahl ausgegeben werden (vorher in \$2F den Bildschirm auswählen !).

Parameter

\$02/\$03: 16-Bit-Zahl, die ausgegeben werden soll (LOW/HIGH)

\$05: Y-Position, an der die Ausgabe stattfinden soll

\$18/\$19: dito für die X-Position (LOW/HIGH)

Akku: Bit 7=1, dann erfolgt die Ausgabe linksbündig, andernfalls rechtsbündig.

Bit 6=1, um führende Nullen zu unterdrücken

Bits 0-5 müssen Angaben über die Breite des gewünschten Ausgabefeldes (0 bis 63) enthalten, falls die Ausgabe rechtsbündig erfolgen soll (Bit 7=0).

C193 **ReDoMenu**

Baut das aktuelle Untermenü wieder auf. Dadurch ist es möglich, aus einem Untermenü mehrere Punkte auszuwählen.

C196 **GetSerialNumber**

Holt die 16-Bit-Serien-Nummer des GEOS-Kernals in die Speicherstellen \$02/\$03. Diese Zahl wird nur ein einziges Mal (und zwar beim ersten Booten von GEOS) erzeugt und fest in das KERNAL-File auf der Diskette eingetragen. In der GEOS-Version 1.3 findet man diese Nummer in den Speicherstellen \$9EA7 und \$9EA8. Nach der ersten Installation wird außerdem die Speicherstelle \$9EA9 auf \$FF gesetzt, damit der Installations teil bei späteren Bootvorgängen übersprungen wird.

Sämtliche käuflichen Zusatzprogramme (z.B. GEODEX) fragen bei ihrer ersten Installation mehr oder weniger direkt über die Routine GetSerialNumber die GEOS-Seriennummer ab und speichern sie fest auf der Diskette ab. Damit ist es nun so gut wie ausgeschlossen, daß dieses Zusatzprogramm in Verbindung mit einer anderen GEOS-System-Diskette funktioniert. Die Wahrscheinlichkeit, auf zwei gleiche Seriennummern zu treffen, beträgt nämlich genau 1 zu 65535 (Die Seriennummer \$0000 gibt es nicht!). Da jedes Programm zusätzlich auch noch mit einem Kopierschutz versehen ist, genügt es auch nicht, eine Kopie der Diskette vor der ersten Installation anzufertigen.

C199 **Sleep**

Bewirkt eine Verzögerung im Programmablauf. Die Verzögerungszeit (in 1/50 Sekunden) wird in \$02,\$03 übergeben und die Routine durch JSR Sleep aufgerufen. Nach Ablauf der Wartezeit wird das Programm automatisch hinter dem JSR fortgesetzt. Der

Vorteil gegenüber einer einfachen Warteschleife besteht darin, daß die Wartezeit anderen Jobs der Hauptschleife zur Verfügung gestellt wird.

Damit sieht ein typischer Programmausschnitt für eine Verzögerung in GEOS folgendermaßen aus: (Verzögerung: 20 Sekunden=1000/50 Sekunden -> \$03E8)

```

...
...
LDX #$E8
LDY #$03
STX $02
STY $03
JSR $C199
... (Fortsetzung des Programms nach 20 Sekunden)
...

```

Man sollte jedoch beachten, daß selbstverständlich in der Zwischenzeit von anderen Routinen, die von der Hauptschleife aufgerufen werden, Manipulationen an Speicherinhalten vorgenommen werden können.

C19C ClearMouseMode

Setzt das Maus-Flag (\$30) auf "0" und schaltet das Sprite ab.

Dadurch reduziert sich die Zeit, in der der Video-Chip zur Darstellung des Sprites auf das RAM zugreifen muß. Die Turbo-Disk-Routinen schalten bei Lade- und Speichervorgängen kurzzeitig die Maus ab, da die Datenübertragung durch den dauernden Zugriff des Video-Chips auf das RAM gestört würde. Normalerweise müßte hierbei sogar der gesamte Video-Chip ausgestellt werden, da zur Darstellung der Hires-Grafik auch Speicherzugriffe nötig sind. Die Turbo-Disk-Routinen nutzen jedoch geschickt die Zeitintervalle aus, in denen mit Sicherheit keine solchen Zugriffe zu erwarten sind. Das Gegenstück zu ClearMouseMode ist StartMouseMode (\$C14E).

C19F **iRectangle**

Im Gegensatz zur Funktion "Rectangle" können hier die Daten direkt hinter den Befehl JSR iRectangle gelegt werden. Das sieht dann folgendermaßen aus:

```
1000 JSR $C19F
1003 Byte 1 für $06 (oben)
1004 Byte 2 für $07 (unten)
1005 Byte 3 für $08 (links)
1006 Byte 4 für $09  "
1007 Byte 5 für $0A (rechts)
1008 Byte 6 für $0B  "
1009 hier wird das Programm fortgesetzt.
```

Für das Füllmuster gilt das unter "Rectangle" (\$C124) Gesagte. Die Fortsetzung des Programms erfolgt anschließend automatisch direkt hinter den Daten.

C1A2 **iFrameRectangle**

Funktion wie FrameRectangle (\$C127). Die Daten müssen jedoch direkt hinter dem Unterprogrammaufruf stehen.

C1A5 **iRecoverRectangle**

Funktion wie RecoverRectangle (\$C12D). Die Daten müssen jedoch direkt hinter dem Unterprogrammaufruf stehen.

C1A8 **iGraphicsString**

Funktion wie GraphicsString (\$C136). Jobs und Daten müssen jedoch direkt hinter dem Unterprogrammaufruf stehen.

C1AB **iBitmapUp**

Funktion wie BitmapUp (\$C142). Die Daten müssen jedoch direkt hinter dem Unterprogrammaufruf stehen.

C1AE **iPutString**

Funktion siehe \$C148; die Daten hinter dem Unterprogrammaufruf haben das folgende Format:

Byte 0/1 X-Position (LOW/HIGH)

Byte 2 Y Position

Byte 3 und folgende: Text für die Ausgabe. Der Text muß mit einer "0" abgeschlossen werden.

C1B1 **GetRealSize**

Holt Informationen über die Größe des übergebenen ASCII-Zeichens im aktuellen Zeichensatz. Berücksichtigt dabei die unterschiedlichen Schriftarten (Bold, Outline...).

Der ASCII-Wert des Zeichens muß im Akku übergeben werden und im X-Register die aktuelle Schriftart (aus \$2E). Nach dem Aufruf von GetRealSize steht im Y-Register die Breite, im X-Register die Höhe und im Akku die Höhe ohne Unterlängen (jeweils in Pixeln).

C1B4 **iFillRam**

Funktion wie FillRam (\$C17B). Die 5 Datenbytes müssen jedoch direkt hinter dem Unterprogrammaufruf stehen.

C1B7 **iMoveData**

Funktion wie ModeData (§C17E). Die 6 Datenbytes müssen jedoch direkt hinter dem Unterprogrammaufruf stehen.

C1BA **GetString**

Ausgabe eines Textes mit anschließender Möglichkeit, diesen zu modifizieren.

Parameter

- \$02,\$03 Zeiger auf einen Puffer, der den auszugebenden Text enthält. Der Text muß mit einer "0" abschließen. Soll nur eine Eingabe erfolgen, so muß der hier angegebene Puffer sofort mit einer Null beginnen.
- \$04 Zeilenende-Flag (s.u.)
- \$05 Y-Position der Ausgabe (0-199)
- \$06 maximale Länge des Textes
- \$0A,\$0B Job bei Überschreiten der max. Länge (LOW/HIGH)
- \$18,\$19 X-Position der Ausgabe (0-319) (LOW/HIGH)

Wird das Zeilenende erreicht und ist \$04 ungleich "0", wird der Job in \$0A,\$0B angesprungen. Andernfalls wird bei Überschreiten des Zeilenendes die Routine aufgerufen, deren Adresse sich in \$84AB/\$84AC befindet (Default). Hiermit besteht die Möglichkeit, auf Fehleingaben definiert reagieren zu können.

C1BD **GotoFirstMenu**

Baut die gesamte Menüstruktur bis zum Hauptmenü ab. Normalerweise sollte diese Routine zu Beginn jedes Programms stehen, das vom Menü aufgerufen wird. (s. auch DoMenu=§C151).

die Benutzung des Vektors sogar auf einen schlechten Programmierstil hin, da solch eine Schleife für Anwendungen in GEOS nur in den seltensten Fällen benötigt wird.

C1C6**DrawSprite**

Kopiert einen Spriteblock in einen der 8 verfügbaren Puffer. Von dort kann das Sprite direkt durch Einschalten (siehe EnableSprite=\$C1D2) sichtbar gemacht werden.

\$08 = Nummer des Blocks (0-7)

\$0A,\$0B = Zeiger auf die Sprite-Daten (LOW/HIGH)

C1C9**GetCharWidth**

Holt die Breite der sich im Akku befindlichen ASCII-Zeichen und übergibt es im Akku. Dabei werden evtl. aktivierte Schriftarten nicht berücksichtigt (Bold, Outline). War das Zeichen ein Steuercode (kleiner \$20), wird eine Null übergeben.

C1CC**LoadCharSet**

Schaltet einen neuen Zeichensatz ein. Der Zeichensatz muß sich schon im Speicher befinden. Die Startadresse wird in \$02,\$03 übergeben.

C1CF**PosSprite**

Das Sprite wird an der vorgegebenen Stelle positioniert.

\$08 = Nummer der Sprites
\$0A/\$0B = X-Position (0-319) (LOW/HIGH)
\$0C = Y-Position (0-199)

C1D2 **EnableSprite**

Schaltet das Sprite, dessen Nummer in \$08 steht, ein.

C1D5 **DisableSprite**

Schaltet das Sprite, dessen Nummer in \$08 steht, aus.

C1D8 **CallRoutine**

Ersetzt den im Befehlssatz des 6502-Prozessors fehlenden Befehl JSR (\$xxxx). CallRoutine ermöglicht also einen indirekten Einsprung in ein Unterprogramm. Dazu muß die Anfangsadresse der Routine im Akku (LOW) und im X-Register (HIGH) übergeben werden und JSR CallRoutine aufgerufen werden.

C1DB **CalcBlocksFree**

Berechnet die Anzahl der freien Blöcke aus der BAM. Der Zeiger auf die BAM muß sich in \$0C,\$0D befinden (normal liegt die BAM in GEOS ab \$8200). Das Ergebnis wird in \$0A,\$0B zurückgeliefert.

C1DE **ChkDkGEOS**

Testet die BAM auf GEOS-Format. Das Ergebnis wird im Akku und in \$848B übergeben. Ist der Akku=0, hat die Diskette kein

GEOS-Format, bei \$FF handelt es sich um eine GEOS-Diskette. Die aktuelle BAM befindet sich in GEOS normalerweise ab \$8200. Die Routine ChkDkGEOS erwartet in \$0C/\$0D einen Zeiger auf die BAM.

C1E1 **NewDisk**

Lädt die BAM einer neu eingelegten Diskette nach \$8200.

Parameter

keine

Rückgabe: Error-Status im X-Register (siehe Anhang)

C1E4 **GetBlock**

Liest einen vollständigen Sektor (256 Bytes) von der Diskette.

Parameter

\$04 = Track (1-35)

\$05 = Sektor (0-MaxSektor, siehe Floppy-Bedienungsanl.)

\$0A,\$0B = Zeiger auf einen freien Datenpuffer (meist \$8000)

Rückgabe: Error-Status im X-Register (siehe Anhang)

C1E7 **PutBlock**

Schreibt einen vollständigen Sektor (256 Bytes) auf die Diskette.

Parameter

- \$04 = Track (1-35)
\$05 = Sektor (0-MaxSektor, siehe Floppy-Bedienungsanl.)
\$0A,\$0B = Zeiger auf die Daten

Rückgabe: Error-Status im X-Register (siehe Anhang)

C1EA**SetGEOSDisk**

Bringt die eingelegte Diskette auf das GEOS-Format. Setzt zusätzlich das GEOS-Flag in \$848B. SetGEOSDisk trägt einen Identifizierungstext in die BAM ein und erzeugt die Border, damit Dateien am unteren Rand im DESK TOP abgelegt werden können.

Parameter

keine

C1ED**SaveFile**

Speichert eine Datei auf der Diskette ab. SaveFile bietet eine relative einfache Möglichkeit, um eigene Programme oder Daten im GEOS-Format abspeichern zu können. Außerdem kann SaveFile dazu benutzt werden, um ein VLIR-File neu zu erstellen.

Parameter

- \$16 Zeiger auf die Seite im DESK TOP, ab der nach einem freien Eintrag gesucht werden soll (0-143).
\$15,\$14 Zeiger auf einen Puffer, der den zukünftigen Info-Sektor enthalten muß (üblicherweise ab \$8100). Der Info-Sektor muß vollständig initialisiert sein, da SaveFile hieraus alle Parameter entnimmt. Hier die erforderlichen Parameter:

\$8100,\$8101	Hier muß ein Zeiger auf den Filenamen stehen. Der Filename darf jedoch nicht innerhalb des Info-Sektors abgelegt werden.
\$8102	= \$03, Breite des zukünftigen Icons in Bytes
\$8103	= \$15, Höhe des Icons in Zeilen
\$8104	= \$3F, Grafik-Steuerzeichen, s. BitmapUp=\$C142
\$8105-8143	63 Sprite-Bytes für das Icon
\$8144	DOS-Filetyp (\$81=SEQ, \$82=PRG, \$83=USR, \$84=REL)
\$8145	GEOS-Filetyp (1-14)
\$8146	Filestruktur (0=SEQ oder 1=VLIR)
\$8147,\$8148	Beginn der Daten
\$8149,\$814A	Ende der Daten
\$814B,\$814C	Initialisierungseinsprung nach dem Laden

Rückgabe : Error-Status im X-Register (siehe Anhang)

C1F0

SetGDirEntry

SetGDirEntry ist eine Unterroutine von SaveFile (\$C1ED) und wird daher sehr selten direkt angesprochen. Sie erstellt den Directory-Eintrag des Files. Die Parameter entsprechen denen von \$C1ED. Die Adresse der abzuspeichernden Daten wird nicht benötigt.

Parameter

In \$16 muß sich ein Zeiger auf die erste Directory-Seite befinden, ab der nach einem freien Eintrag gesucht werden soll (0-143).

In \$0E,\$0F muß sich ein Zeiger auf einen Puffer befinden (üblich \$8300), der Zeiger (Track, Sektor) auf genügend reservierte freie Blöcke der Diskette enthält (siehe BlkAloc=\$C1FC). Die Anzahl der benötigten Blöcke für ein File kann folgendermaßen berechnet werden:

Anzahl=2+INT((Programmumfang in Bytes)/254)

Falls die Datei im VLIR-Format gespeichert werden soll, muß noch ein weiterer Block für den VLIR-Linker addiert werden.

Der neue Directory-Block wird auf die Diskette geschrieben. Sinnvollerweise sollte kurz nach SetGDirEntry auch die BAM neu abgespeichert werden (PutDirHead=\$C24A), da evtl. eine weitere Directory-Seite beschrieben worden ist.

In \$0E,\$0F befindet sich anschließend ein Zeiger auf das erste, noch nicht benutzte Track/Sektor-Paar. Diese Angabe kann direkt für einen Speichervorgang durch WriteFile=\$C1F9 weiterverwendet werden.

C1F3 **BldGDirEntry**

Unterprogramm von SetGDirEntry (\$C1F0). Der Directory-Eintrag wird zwar in \$8400-\$841D erstellt, aber noch nicht auf die Diskette geschrieben. In \$0E,\$0F befindet sich anschließend ein Zeiger auf das erste, für den Speichervorgang verfügbare Track/Sektor-Paar.

C1F6 **GetFreeDirBlk**

Sucht einen freien Directory-Eintrag ab der Directory-Seite in \$16.

Das Y-Register zeigt anschließend in \$8000 auf den DOS-Filetyp des nächsten freien Eintrags.

Im X-Register wird die Fehlermeldung übergeben (s. Anhang).

C1F9 **WriteFile**

Speichert ein File auf der Diskette, ohne einen Eintrag im Directory zu erzeugen.

\$0E,\$0F = Zeiger auf einen Puffer, der die nötigen freien Blocks enthalten muß (Track, Sektor).

\$10,\$11 = Zeiger auf die zu speichernden Daten. Es wird solange in die angegebenen Blocks gespeichert, bis ein Track "0" auftritt. Dies dient als Endezeichen.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C1FC **BlkAlloc**

Belegt die für einen Speichervorgang benötigten Blöcke (max. 127=32258 Bytes) auf der Diskette. Dazu werden entsprechend viele freie Blöcke gesucht, als belegt gekennzeichnet und in einem Puffer abgelegt, dessen Adresse in **\$0E,\$0F** steht (üblich \$8300). Zusätzlich muß die Anzahl der zu speichernden Bytes in **\$06,\$07** übergeben werden.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C1FF **ReadFile**

Liebt eine verkettete Datei ein. Die Daten werden im RAM abgelegt.

Parameter

\$04 = Track des ersten Datenblocks

\$05 = Sektor des ersten Datenblocks

\$06,\$07 = Maximale Anzahl Daten, die gelesen werden sollen

\$10,\$11 = Adresse zum Speichern der Daten im RAM

Im X-Register wird die Fehlermeldung übergeben. X=0: kein Fehler aufgetreten, X=11: Es sind mehr Daten vorhanden als laut \$06,\$07 geladen werden sollten. In \$06,\$07 kann man also eine Obergrenze angeben, um nicht wichtige Programmteile zu überschreiben. Weitere Fehler: s. Anhang.

C202 SmallPutChar

Gibt ein Zeichen im Akku an der aktuellen Cursor-Position aus, ohne anschließend auf die nächste Schreibposition vorzurücken.

C205 FollowChain

Holt von einem verketteten File die verwendeten Tracks und Sektoren in einen Puffer, auf den \$08,\$09 zeigen muß (Üblich \$8100). In \$04,\$05 müssen Track und Sektor des ersten Datenblocks übergeben werden.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C208 GetFile

Dies ist die Hauptroutine zum Laden und evtl. Starten von Programmen.

Parameter

- \$0E,\$0F = Zeiger auf den gewünschten Filenamen
- \$02 =0: Das File wird an die in seinem Info-Sektor spezifizierte Stelle geladen. Dies ist der Normalfall.
- ftg =1: Die Daten werden ab der in \$10/\$11 angegebenen Adresse abgelegt. Da ein Programm in den seltensten Fällen relocatibel ist, wird diese Op-

tion hauptsächlich für reine Daten benutzt (z.B. beim Laden von Zeichensätzen).

\$16/\$17: Im Falle des Ladens eines Accessories (Filetyp 5) müssen hier noch diverse Eintragungen vorgenommen werden, die an das zu ladende Programm weitergeleitet werden. Das Accessory muß diese Angaben unbedingt auswerten und entsprechend reagieren.

Bit 7=1: Der sichtbare Bildschirm muß vom Accessory gerettet werden (z.B. in einem zweiten Swap-File). Wird z.B. von GEOPAINT aus ein Accessory geladen, so setzt GEOPAINT dieses Bit, da kein vollständiger Hintergrundbildschirm existiert.

Bit 7=0: Ein Retten des sichtbaren Bildschirms durch das Accessory ist nicht notwendig, da ein Hintergrundbildschirm verwaltet wird. Vom Accessory überschriebene Bildschirminformationen werden am Ende automatisch "von unten nach oben" kopiert.

Bit 6=1: Die Farben müssen gerettet werden.

Bit 6=0: Die Farben brauchen nicht gerettet zu werden.

GetFile erkennt anhand des Filetyps selbständig, mit welcher Laderoutine die angegebene Datei geladen werden muß. Handelt es sich um eine Application (GEOPAINT, GEOWRITE, etc.), so wird das Programm geladen und automatisch gestartet.

Gleiches gilt für Accessorys. Hierbei wird jedoch vorher noch der aktuelle System-Status (Event-Vektoren im \$84xx-Bereich, Angaben über Prozesse, Menus, usw.) gerettet. Außerdem wird ein Swap-File auf der Diskette erstellt, das den Inhalt des vom Accessory benötigten Arbeitsspeichers rettet. Dadurch wird es überhaupt erst möglich, daß nach Beendigung des Hilfsprogramms in der Anwendung weitergearbeitet werden kann.

Reine Datenfiles werden nur in den Arbeitsspeicher geladen.

C20B FindFile

Sucht im Directory und im Borderblock (das ist der Sektor, in dem GEOS die File-Einträge verwaltet, deren Icons auf dem Rand abgelegt sind) nach einem File-Eintrag. Dieser wird dann in \$8400 übergeben.

\$0E,\$0F =Zeiger auf den Filenamen (max. 16 Stellen plus eine Abschluß-Null)

Im X-Register wird die Fehlermeldung übergeben. (X=0: kein Fehler aufgetreten, X=5: File not Found, weitere Fehler: siehe Anhang).

C20E CRC

Berechnet eine Prüfsumme über einen definierten Bereich des Arbeitsspeichers. Diese Funktion ist zum Beispiel recht nützlich, um Kopierschutz-Routinen vor Manipulationen zu sichern.

Parameter

\$02/\$03 =Zeiger auf die Anfangsadresse der Daten

\$04/\$05 =Umfang der Daten in Bytes

Das Ergebnis steht anschließend in \$06/\$07.

C211 LdFile

Dies ist eine Laderoutine für GEOS-Files. Ihr muß in \$14,\$15 ein Zeiger auf den kompletten File-Eintrag übergeben werden (beispielsweise durch FindFile \$C20B nach \$8400 holen).

LdFile holt die notwendigen Informationen aus dem INFO-Sektor, lädt das File an die dort eingetragene Stelle und startet gegebenenfalls das Programm (je nach Filetyp). Eine alternative

Ladeadresse kann in dieser Low-Level-Ebene nicht mehr angegeben werden (s. GetFile).

C214**EnterTurbo**

Schaltet die schnellen Disk-Routinen ein. Falls sich die Routinen noch nicht im Floppy-RAM befinden, werden sie erst dorthin kopiert.

Parameter

\$8489 Nummer des Laufwerkes, auf das sich EnterTurbo bezieht (8 oder 9). Bei nur einem angeschlossenen Laufwerk steht hier automatisch eine Acht.

C217**LdDeskAcc**

Kompletter Ladevorgang für ein Accessory. LdDeskAcc wird beispielsweise automatisch aufgerufen, wenn man den Filenamen eines Accessories an GetFile=\$C208 übergibt. Diese Routine erstellt ein Swap-File, rettet den derzeitigen System-Status (Menus, Prozesse, alle Vektoren von \$849B bis einschl. \$84C1, aktueller Font, Style, Füllmuster), lädt und startet das Accessory. Trägt das Accessory den Job RstrAppl=\$C23E in den MainVector \$849B/\$849C ein und führt anschließend einen RTS aus, so wird der ursprüngliche Zustand der Application wiederhergestellt (s. auch GetFile=\$C208).

C21A**ReadBlock**

Low-Level-Routine zum schnellen Laden eines Sektors (256 Bytes) von der Diskette. Vor dem Aufruf der Routine muß InitForIO=\$C25C aufgerufen werden. Nach dem letzten Aufruf von ReadBlock muß der Diskettenzugriff durch den Aufruf von Do-

neWithIO=\$C25F beendet werden. Falls nur ein einziger Block geladen werden soll, ist hierfür GetBlock=\$C1E4 besser geeignet.

Parameter

\$04/\$05: Zeiger auf den zu ladenden Track und Sektor.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C21D **LdApplic**

Kompletter Ladevorgang für eine Application. LdApplic wird beispielsweise automatisch aufgerufen, wenn man den Filenamen einer Anwendung an GetFile=\$C208 übergibt.

C220 **WriteBlock**

Low-Level-Routine zum schnellen Speichern eines Sektors (256 Bytes) auf die Diskette. Vor dem Aufruf der Routine muß InitForIO=\$C25C aufgerufen werden. Nach dem letzten Aufruf von WriteBlock muß der Diskettenzugriff durch den Aufruf von DoneWithIO=\$C25F beendet werden. Falls nur ein einziger Block gespeichert werden soll, ist hierfür PutBlock=\$C1E7 besser geeignet.

Parameter

\$04/\$05: Zeiger auf den zu schreibenden Track und Sektor.

\$0A/\$0B: Zeiger auf die zu schreibenden 256 Bytes

Rückgabe: Error-Status im X-Register (siehe Anhang)

C223 VerWriteBlock

Funktion wie WriteBlock=\$C220. Zusätzlich wird anschließend ein Verify durchgeführt.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C226 FreeFile

FreeFile holt die BAM der aktuellen Diskette nach \$8200 und löscht ein File. Ein Pointer in \$14,\$15 muß auf den Directory-Eintrag des zu löschenden Files zeigen. Die Blocks des Files werden in der BAM freigegeben.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C229 GetFHdrInfo

Lädt den INFO-Sektor eines Files nach \$8100 und holt daraus die folgenden Daten:

\$04 =Starttrack des Files
 \$05 =Startsektor des Files
 \$10,11 =Ladeadresse

Der Routine muß in \$14,\$15 der Pointer auf den Directory-Eintrag übergeben werden. Gewöhnlich steht dieser in den ersten 30 Bytes ab \$8400.

C22C EnterDeskTop

Diese Routine muß zum Verlassen einer Anwendung (Filetyp=6) angesprungen werden. Anschließend wird ein GEOS-Warmstart durchgeführt, der Bildschirm gelöscht und die Aktion eingeleitet, die das DESK TOP lädt und startet. Befindet sich das DESK

TOP nicht auf der eingelegten Diskette, so erscheint ein Fenster mit der Aufforderung, eine Diskette einzulegen, die das DESK TOP enthält.

Parameter

keine

C22F

StartAppl

In diese Routine mündet EnterDeskTop=\$C22C. Es wird ein Warmstart durchgeführt, der Zeichensatz BSW-9-Point aktiviert und (falls noch nicht geschehen) das High-Byte der Seriennummer des Kernals in eine bestimmte Speicherstelle übertragen (in GEOS V1.3 deutsch ist dies die Adresse \$D82A). Stellt die Routine FindFTypes=\$C23B gelegentlich ihres Aufrufs Manipulationen an der Seriennummer fest, sabotiert sie die Routine LdDeskAcc, indem sie die Einsprungadresse im Vektor \$C217 um ein Byte erhöht. Ein mysteriöser Absturz des Rechners ist in diesem Fall im wahrsten Sinne des Wortes vorprogrammiert. Dies ist jedoch nur eine kleine Kostprobe des hervorragenden Kopierschutzes von GEOS, um Hackern das Handwerk nicht ganz so einfach zu machen.

Am Ende von StartAppl wird schließlich die Routine gestartet, die ab der in \$10/\$11 angegebenen Adresse liegen muß. Der eigentliche Ladevorgang muß also schon abgeschlossen sein.

C232

ExitTurbo

In der aktuellen Floppy (Nummer in \$8489) wird der Speeder abgeschaltet (jedoch nicht gelöscht). Es darf jeweils nur eine Floppy einen aktiven Speeder haben (Busproblem!). Durch EnterTurbo wird der Speeder wieder aktiviert.

C235 PurgeTurbo

Funktion ähnlich wie ExitTurbo=C232. Der Unterschied besteht darin, daß bei einem anschließenden EnterTurbo die Speeder-Routinen neu in die Floppy kopiert werden müssen, während nach ExitTurbo der sich noch im Floppy-RAM befindliche Speeder nur wieder aktiviert wird.

C238 DeleteFile

Löscht eine Datei vollständig von der Diskette und gibt alle durch sie belegten Blöcke wieder frei. Der Routine muß in \$02,\$03 ein Pointer auf den mit Null endenden Filenamen übergeben werden.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C23B FindFTypes

Holt die Namen der Files eines ganz bestimmten GEOS-Filetyps in eine Liste. Eine genauere Selektion ist durch eine Angabe für den gewünschten "TYP", der im Infosektor steht, möglich.

Übergabeparameter

\$0E,\$0F: Zeiger auf freien Datenbereich für die gefundenen Filenamen (je Name: 17 Bytes).

\$10: gesuchten GEOS-Filetyp (1-14)

\$11: maximal gewünschte Anzahl Files

\$16,\$17: hier kann ein Zeiger auf den gewünschten "TYP"-Eintrag der Files stehen. Falls dies unerwünscht ist, muß der Pointer auf \$0000 zeigen.

C23E RstrAppl

Dies ist die wichtigste Routine für die Programmierung von Accessories. Da beim Laden dieser Files ein Swap-File auf der Diskette erzeugt wird, muß dies nach Beendigung des Programms (QUIT) wieder zurückgespeichert werden. Außerdem wird der alte System-Status wiederhergestellt (Funktion der Pull-Down-Menüs, Icons, etc.). Dazu wird der Hauptschleife die Adresse dieser Routine übergeben. Das Ende eines Accessories sollte daher immer folgendermaßen aussehen:

```
LDX #$3E   Adresse von RstrAppl
LDY #$C2
STX $849B  als Job übergeben
STY $849C
RTS       Ende des Accessories
```

Dieses Programm stellt damit das kürzeste "funktionsfähige" Accessory dar.

C241 ToBasic

Routine lädt und startet ein BASIC-Programm von GEOS aus.

Parameter

\$10/\$11: Pointer auf einen auszuführenden BASIC-Befehl (z.B. LOAD "Puc Man",8,1).

C244 FastDelFile

Funktion wie DeleteFile=\$C238.

Außer dem Filenamen muß dieser Routine jedoch noch ein Zeiger (\$08,\$09) auf eine Tabelle übergeben werden, die die durch das File belegten Tracks und Sektoren enthält, die wieder frei-

gegeben werden sollen. Falls diese Tabelle nicht sowieso schon vorliegen sollte, ist ein Aufruf von DeleteFile bequemer.

C247 **GetDirHead**

Lädt die BAM nach \$8200.

C24A **PutDirHead**

Schreibt die BAM von \$8200 auf die Diskette. Diese Funktion muß immer dann aufgerufen werden, wenn neue Blöcke durch Low-Level-Routinen belegt worden sind.

C24D **NxtBlkAlloc**

Diese Routine unterscheidet sich von BlkAlloc=\$C1FC nur durch eine etwas höhere Einsprungadresse. Dadurch besteht die Möglichkeit, in \$08,\$09 den Track und Sektor selbst festzulegen, ab dem die Suche nach freien Blöcken beginnen soll. Für normale Anwendungen ist dieser Befehl jedoch nicht erforderlich.

C250 **ImprintRectangle**

Routine rettet ein Rechteck auf den 'unsichtbaren' zweiten Bildschirm (Gegenstück: RecoverRectangle= \$C12D).

\$06: obere Grenze
 \$07: untere Grenze
 \$08,\$09: linke Grenze (LOW,HIGH)
 \$0A,\$0B: rechte Grenze (LOW,HIGH)

C253 **iImprintRectangle**

Wie `ImprintRectangle=$C250`. Die acht Daten müssen jedoch direkt hinter dem Unterprogrammaufruf liegen.

C256 **DoDlgBox**

Erzeugt ein Fenster (Dialog Box) für Ein- und/oder Ausgaben. Der Routine muß hierzu nur in `$02/$03` ein Pointer auf eine Datentabelle übergeben werden, die nähere Informationen über das Fenster enthält. Solch eine Tabelle ist folgendermaßen aufgebaut:

1. Byte: Dieses Byte enthält im 7. Bit Informationen über die Größe der Dialogbox:

Bit 7=1: Fenstergröße und Position wie bei den üblichen DESK-TOP-Fenstern. In den meisten Fällen wird man diesen Modus wählen.

Bit 7=0: Die sechs folgenden Bytes der Tabelle enthalten Angaben über die Größe und Position der Dialogbox.

Bits 0-4: Hier kann die Nummer des Musters angegeben werden, mit dem die Dialogbox hinterlegt werden soll (Schattenfenster). Der Wert "0" ergibt kein Schattenfenster. DESK-TOP-Dialogboxen tragen hier eine Eins ein (schwarzer Schatten).

2. Byte: Position des oberen Fensterrandes (0-199)(nur, falls im 1. Byte Bit 7=0 ist)

3. Byte: Position des unteren Fensterrandes (0-199, > 2. Byte)(nur, falls im 1. Byte Bit 7=0 ist)

4./5. Byte: Position des linken Fensterrandes (0-319)(nur, falls im 1. Byte Bit 7=0 ist)

6./7. Byte: Position des rechten Fensterrandes (0-319, > 4. und 5. Byte)(nur, falls im 1. Byte Bit 7=0 ist)

Die weiteren Bytes der Fenstertabelle können aus den nun folgenden Steuerzeichen bestehen, die in beliebiger Reihenfolge auftreten dürfen und vom Betriebssystem nacheinander abgearbeitet werden:

- \$00:** *Funktion:* Endekennzeichen der Tabelle.
Parameter: keine
Kommentar: Mit diesem Zeichen muß jede Tabelle enden. Die Reaktion des Rechners hängt von der Art der vorausgegangenen Steuerzeichen ab. Durch das Endekennzeichen wird kein Abbau der Dialogbox veranlaßt. Hierfür existieren eigene Steuerzeichen.
- \$01:** *Funktion:* Das OK-Feld soll ausgegeben werden.
Parameter: 1. Byte: Abstand vom linken Rand der Dialogbox in ganzen Bytes
2. Byte: Abstand vom oberen Rand der Dialogbox in Zeilen
Kommentar: Es erscheint an der angegebenen Position das auch vom DESK TOP benutzte OK-Icon. Das Fenster wird automatisch abgebaut, sobald der Anwender das Feld mit der Maus anklickt. DoDlgBox kehrt anschließend zum aufrufenden Programm zurück. Hier kann nun aus der Speicherstelle \$02 die Nummer des Steuerzeichens ausgelesen werden, die den Abbau des Fensters verursacht hat (in diesem Fall also eine \$01).
- \$02:** *Funktion:* Das CANCEL-Feld (Abbruch) soll ausgegeben werden.
Parameter: wie bei Steuerzeichen \$01.
Kommentar: wie bei Steuerzeichen \$01.
- \$03:** *Funktion:* Das YES-Feld soll ausgegeben werden.
Parameter: wie bei Steuerzeichen \$01.
Kommentar: wie bei Steuerzeichen \$01.

- \$04:** *Funktion:* Das NO-Feld soll ausgegeben werden.
Parameter: wie bei Steuerzeichen \$01.
Kommentar: und wie bei Steuerzeichen \$01.
- \$05:** *Funktion:* Das OPEN-Feld (Öffnen) soll ausgegeben werden.
Parameter: wie bei Steuerzeichen \$01.
Kommentar: wie bei Steuerzeichen \$01.
- \$06:** *Funktion:* Das DISK-Feld soll ausgegeben werden.
Parameter: wie bei Steuerzeichen \$01.
Kommentar: wie bei Steuerzeichen \$01.
- \$07-\$0A:** Diese Steuerzeichen sind in GEOS noch nicht implementiert.
- \$0B:** *Funktion:* Gibt einen vorhandenen Text in dem Fenster aus.
Parameter: 1. Byte: Abstand des Textes vom linken Rand der Dialogbox in Pixeln
2. Byte: Abstand des Textes vom oberen Rand der Dialogbox in Zeilen
3. und 4. Byte: Adresse des Textes, der ausgegeben werden soll.
Kommentar: Der Text darf alle darstellbaren ASCII-Zeichen und Style-Kommandos enthalten (s. ASCII-Tabelle im Anhang). Als Endeckenzeichen dient eine Null.
- \$0C:** *Funktion:* Gibt einen vorhandenen Text aus, auf den über eine 0-Page-Adresse zugegriffen wird.
Parameter: 1. Byte: Abstand des Textes vom linken Rand der Dialogbox in Pixeln
2. Byte: Abstand des Textes vom oberen Rand der Dialogbox in Zeilen
3. Byte: Angabe einer 0-Page-Adresse, die einen Zeiger auf den auszugebenden Text enthalten muß.

- Kommentar:** Der Text darf alle darstellbaren ASCII-Zeichen und Style-Kommandos enthalten (s. ASCII-Tabelle im Anhang). Als Endekennzeichen dient eine Null. Erlaubter Bereich für den Pointer in der 0-Page: \$0C-\$16 (incl.)
- \$0D: Funktion:** Gibt einen vorhandenen Text aus, auf den über eine 0-Page-Adresse zugegriffen wird, und erlaubt eine Korrektur des Textes mittels Tastatur.
- Parameter:**
1. Byte: Abstand des Textes vom linken Rand der Dialogbox in Pixeln
 2. Byte: Abstand des Textes vom oberen Rand der Dialogbox in Zeilen
 3. Byte: Angabe einer 0-Page-Adresse, die einen Zeiger auf den auszugebenden Text enthalten muß.
 4. Byte: Maximale Anzahl Zeichen, die bei der Eingabe akzeptiert werden sollen
- Kommentar:** Der auszugebende Text sollte sinnvollerweise nur aus darstellbaren ASCII-Zeichen bestehen (s. ASCII-Tabelle im Anhang). Als Endekennzeichen dient eine Null. Erlaubter Bereich für den Pointer in der 0-Page: \$0C-\$16 (incl.)
- \$0E: Funktion:** Das Fenster wird abgebaut, sobald die Maus an einer beliebigen Position geklickt wird, an der sich kein anderes Klick-Feld des Fensters befindet.
- Parameter:** keine
- \$0F: Funktion:** Gibt einen Grafik-String aus.
- Parameter:** 1. und 2. Byte: Zeiger auf eine Tabelle, die Angaben über den Grafik-String enthalten muß. Für den Aufbau einer solchen Tabelle siehe GraphicsString= \$C136.

\$10: Funktion: Ausgabe eines Verzeichnisses, das die Dateinamen aller Files eines bestimmten Filetyps der aktuellen Diskette enthält. Eine solche Datei kann mit der Maus selektiert werden.

Parameter:

1. Byte: Abstand des Verzeichnisses vom linken Rand der Dialogbox in ganzen Bytes
2. Byte: Abstand des Verzeichnisses vom oberen Rand der Dialogbox in Zeilen

in \$10: Nummer des gesuchten GEOS-Filetyps
in \$16/\$17: Pointer auf den CLASS-Namen der gesuchten Dateien. Es werden nur die Dateinamen im Verzeichnis ausgegeben, deren Files im Info-Sektor die spezifizierte Eintragung besitzen. Sollte dies nicht erwünscht sein, muß der Vektor \$16/\$17 auf Null gesetzt werden.
in \$0C/\$0D: Pointer auf einen Puffer, der genügend Platz für den selektierten File-Eintrag (17 Bytes) aufweisen muß. In diesen Puffer überträgt DoDlgBox den selektierten Filenamen.

Kommentar: Zusätzlich zu dem Steuerzeichen \$10 sollten in der Tabelle der Dialogbox noch weitere Steuerzeichen auftreten, die ein Verlassen des Fensters ermöglichen (z.B. OK, CANCEL).

\$11: Funktion: Ermöglicht den Aufruf eines Unterprogramms des Anwenders, sobald die Maus an einer beliebigen Stelle außerhalb eines Fenster-Icons angeklickt wird.

Parameter: 1. und 2. Byte: Adresse der Routine des Anwenders, die angesprungen werden soll (LOW/HIGH).

Kommentar: Nach der Ausführung der angegebenen Routine kann die Dialogbox durch den Aufruf `JMP RstrFrmDialog=$C2BF` wieder

- abgebaut werden. Das Programm wird anschließend hinter JSR DoDlgBox fortgesetzt. Wird die angegebene Routine jedoch durch ein RTS beendet, erfolgt eine weitere Abarbeitung der in der Fenstertabelle enthaltenen Steuerzeichen.
- \$12: Funktion:** Ermöglicht die Einbindung selbst definierter Icons in das Fenster.
- Parameter:**
1. Byte: Abstand des Icons vom linken Rand der Dialogbox in ganzen Bytes
 2. Byte: Abstand des Icons vom oberen Rand der Dialogbox in Zeilen
 3. und 4. Byte: Pointer zu den Grafik-Daten des selbst definierten Icons (Format: siehe BitmapUp=\$C142).
 5. und 6. Byte: Null
 7. Byte: Breite des Icons in Bytes (Standard: \$06)
 8. Byte: Höhe des Icons in Zeilen (Standard: \$12)
 9. und 10. Byte: Pointer auf eine Routine, die beim Anklicken des Icons angesprungen werden soll.
- \$13: Funktion:** Ermöglicht die Einbindung einer eigenen Routine, die in jedem Fall beim Aufbau des Fensters aufgerufen wird.
- Parameter:**
1. und 2. Byte: Pointer auf eine Routine, die in jedem Fall beim Aufruf von DoDlgBox abgearbeitet werden soll.
- Kommentar:** Der Zeitpunkt der Abarbeitung wird direkt durch die Abfolge der Steuerzeichen innerhalb der Fenstertabelle bestimmt. Folgt also zuerst das Steuerzeichen \$01 und anschließend das Steuerzeichen \$13, so wird

erst das OK-Icon gezeichnet, bevor die angegebene Routine aufgerufen wird. Erst wenn diese mit einem RTS endet, werden alle weiteren Steuerzeichen abgearbeitet.

C259**RenameFile**

Ermöglicht das Umbenennen einer Datei auf der aktuellen Diskette.

Parameter

\$02,\$03: Zeiger auf den derzeitigen Filenamen

\$0C,\$0D: Zeiger auf den neuen Filenamen

Beide Filenamen müssen jeweils mit einer Null enden.

C25C**InitForIO**

Wird vor jeder Aktion mit dem schnellen Datenbus aufgerufen und rettet wichtige Parameter (IRQ, NMI etc.). Für Programmierer ist der Befehl nicht sehr interessant, da er automatisch von übergeordneten Routinen aufgerufen wird.

C25F**DoneWithIO**

Gegenstück zu InitTurboBus (\$C25C). Wird von übergeordneten Routinen, die auch InitForIO aufrufen, selbständig ausgeführt.

C262 **DShiftRight**

Das X-Register muß auf eine 2-Byte-Adresse auf der 0-Page zeigen, die Y-mal nach rechts geschoben wird (gibt jedesmal / 2). Gehört zur Mathe-Bibliothek (ab \$C15D).

C265 **CopyFString**

Kopiert einen String (abgeschlossen durch eine Null) in einen beliebigen Speicherbereich.

X-Register: Zeigt auf Pointer in der 0-Page, der auf den String zeigen muß.

Y-Register: Zeigt auf Pointer in der 0-Page, der auf die Zieladresse zeigt.

Maximal können 256 Bytes kopiert werden.

C268 **CopyString**

Funktion und Parameter wie CopyFString. Es können jedoch mehrere Texte, die hintereinanderliegen, gleichzeitig kopiert werden. Dazu muß im Akku die Anzahl der Texte übergeben werden (Jeder Text muß mit Null abschließen!).

C26B **CmpFString**

Vergleicht zwei Strings miteinander.

Parameter wie CopyFString. Falls im Akku anschließend '\$00' steht, sind beide Strings gleich. Ansonsten steht hier der Offset auf das ungleiche Element.

C26E**CmpString**

Wie CmpFString. Es können jedoch mehrere Texte, die direkt hintereinanderliegen, verglichen werden. Dazu muß der Routine im Akku noch die Anzahl der Texte übergeben werden. Ergebnis wie bei CmpFString. Das X-Register enthält hierbei die Nummer des Strings, bei der Ungleichheiten aufgetreten sind.

C271**FirstInit**

Führt einen Warmstart von GEOS durch und startet eine Anwendung, die ab der in \$I0/\$I1 eingetragenen Adresse liegen muß.

C274**OpenRecordFile**

Eröffnet eine schon existierende VLIR-Datei, die sich auf der Diskette befinden muß.

Parameter

\$02,\$03: Zeiger auf den Filenamen der VLIR-Datei

Rückgabe: Error-Status im X-Register (siehe Anhang)

C277**CloseRecordFile**

Schließt eine VLIR-Datei. Dabei wird der Dir-Eintrag aktualisiert (Datum, Uhrzeit). Keine Parameter notwendig.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C27A **NextRecord**

Positioniert auf den nächsten Record.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C27D **PreviousRecord**

Positioniert auf den vorherigen Record.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C280 **PointRecord**

Positioniert auf einen bestimmten Record. Der Akku muß dazu die Nummer des Records enthalten.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C283 **DeleteRecord**

Entfernt den aktuellen Record aus der Liste. Die Liste wird dazu aufgerückt, und die durch den Eintrag belegten Blöcke werden wieder freigegeben. Es sind keine direkten Parameter zu übergeben.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C286 **InsertRecord**

Fügt einen neuen Record vor den aktuellen Record ein. Dieser neue ist anschließend der aktuelle Record.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C289 **AppendRecord**

Fügt einen neuen Record hinter den aktuellen Record ein. Dieser ist anschließend der aktuelle Record.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C28C **ReadRecord**

Liest den aktuellen Record ein. In \$10,\$11 muß die gewünschte Ladeadresse stehen. In \$06,\$07 muß die maximale Anzahl gewünschter Daten enthalten sein.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C28F **WriteRecord**

Speichert den aktuellen Record ab.

\$06,\$07 Anzahl Bytes
\$10,\$11 Startadresse der Daten

Rückgabe: Error-Status im X-Register (siehe Anhang)

C292 **SetNextFree**

Sucht den nächsten freien Block auf der Diskette und markiert ihn in der BAM als belegt.

Parameter

\$08,\$09: Zeiger auf den ersten Block (Track und Sektor), ab dem gesucht werden soll.

Der Track 18 (Directory!) wird automatisch übersprungen, falls nicht ausdrücklich in \$08 eine 18 eingetragen wird.

C295 **UpdateRecordFile**

Die folgenden Daten eines geöffneten VLIR-Files werden auf der Diskette aktualisiert:

1. VLIR-Linker
2. Directory-Eintrag (Datum, Zeit)
3. BAM

C298 **GetPtrCurDkNm**

Setzt einen Zeiger auf den Namen der aktuellen Diskette. Das X-Register muß dazu auf einen Pointer in der 0-Page zeigen, der anschließend auf den Namen zeigt.

C29B **PromptOn**

Schaltet den Cursor an der aktuellen Schreibposition ein.

C29E**PromptOff**

Schaltet den Cursor aus.

C2A1**OpenDisk**

Initialisiert eine neue Diskette. Diese Routine muß nach jedem Diskettenwechsel aufgerufen werden, da das Betriebssystem einen Diskettenwechsel nicht selbst erkennt.

C2A4**DoInlineReturn**

Unterprogramm aller Inline-Routinen (das sind die Routinen der Sprungtabelle, die mit einem "i" beginnen, z.B. iRectangle).

C2A7**GetNextChar**

Holt ein Zeichen aus dem Tastaturpuffer in den Akku. Ist der Puffer leer, wird eine Null übergeben. Diese Routine ist als Ergänzung zu dem Event-Vektor \$84A3 zu verstehen, dessen Adresse jedesmal angesprungen wird, sobald eine Taste betätigt wird. Die dabei aufgerufene Routine kann sich durch GetNextChar weitere Zeichen aus dem Tastaturpuffer holen, bis ihr im Akku eine Null übergeben wird.

C2AA**BitmapClip**

BitmapClip ermöglicht es, über eine im nicht sichtbaren Bereich liegende Grafik eine Maske mit einem rechteckigen Ausschnitt zu legen und den Teil der Grafik, der durch die Maske nicht verdeckt wird, auf dem sichtbaren Bildschirm darzustellen. Hierdurch können Grafiken beliebiger Größe ausschnittsweise auf einem Teil des Bildschirms dargestellt werden.

Parameter

- \$02,\$03: Zeiger auf den Start der Grafikdaten
 \$04: Linker Rand des Ausgabe-Fensters (in Bytes)
 \$05: Oberer Rand des Ausgabe-Fensters (in Zeilen)
 \$06: Breite des Ausgabe-Fensters in Bytes
 \$07: Höhe des Ausgabe-Fensters in Zeilen

Mit diesen Angaben ist die Größe der ausmaskierten Grafik definiert. Es fehlen jetzt jedoch noch die Angaben, mit denen man den Bereich der Grafik markieren kann, der im Ausgabefenster erscheinen soll:

- \$18: Angabe, wie viele Bytes (= je 8 Pixel) auf der linken Seite der Gesamt-Grafik übersprungen werden sollen, bis die Ausgabe im Fenster beginnt. Der rechte Rand ergibt sich automatisch aus den Angaben in \$06.
 \$19: Hier muß die Anzahl Bytes der Grafik angegeben werden, die rechts neben dem Ausgabefenster ausmaskiert wird. Durch Addition der Werte in \$06, \$18 und \$19 kommt man damit auf die Breite der Gesamt-Grafik in Bytes.
 \$1A,\$1B: Angabe, wie viele Zeilen der Gesamt-Grafik bis zur Ausgabe im Fenster übersprungen werden sollen. Der untere Rand ergibt sich automatisch aus den Angaben in \$07.

C2AD**FindBAMBit**

Testet, ob ein bestimmter Block der Diskette in der BAM schon als belegt gekennzeichnet ist. In \$0E, \$0F muß dazu der Track und Sektor des Blocks enthalten sein. Ist der Akku anschließend gleich null, so ist der Block bereits belegt.

C2B0 **SetDevice**

Freigabe des seriellen IEC-Bus für ein beliebiges Gerät. Im Akku muß die gewünschte Gerätenummer enthalten sein (z.B. Nummer 4, falls der Drucker angesprochen werden soll). Falls das original Floppy-DOS angesprochen werden soll, muß vorher noch `PurgeTurbo=$C235` aufgerufen werden. Durch `InitForIO=$C25C` werden anschließend die 64er-Kernal-Routinen bereitgestellt. Nun können die gewünschten Bus-Aktionen stattfinden. Im Anschluß daran muß das GEOS-Kernal durch den Aufruf von `DoneWithIO=$C25F` wieder eingeblendet werden. Der Floppy-Speeder wird bei der Benutzung von High-Level-Routinen (z.B. `GetFile`, `SaveFile`) automatisch aktiviert. Lediglich bei den Routinen `ReadBlock`, `WriteBlock` und `VerWrBlock` muß er durch `EnterTurbo=$C214` "von Hand" wieder gestartet werden.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C2B3 **IsMseInRegion**

Mit dieser Routine kann getestet werden, ob sich die Maus derzeit in einem bestimmten Bildschirmbereich befindet.

Parameter

- \$06: obere Begrenzung (0-199)
- \$07: untere Begrenzung (größer als in \$06, max. 199)
- \$08,\$09: linke Begrenzung (0-319)
- \$10,\$11: rechte Begrenzung (größer als in \$08/\$09, max. 199)

Falls sich die Maus in dem angegebenen Bereich befindet, wird im Akku \$FF übergeben, andernfalls \$00.

C2B6**ReadByte**

Gehört zu den VLIR-Routinen (s. ab §C274). Holt ein Byte nach dem anderen aus dem aktuellen VLIR-Record in den Akku.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C2B9**FreeBlock (ab GEOS V1.3)**

Gibt einen in der BAM als belegt gekennzeichneten Block wieder frei.

Parameter

\$0E,\$0F: Track und Sektor des Blocks, der freigegeben werden soll.

In der GEOS-Version 1.2 ist FreeBlock noch nicht in der Sprungtabelle enthalten (drei NOP-Befehle). Man kann aber die Funktion durch einen direkten Einsprung ins KERNAL aufrufen. Die Adresse hierfür lautet \$9844. Die Parameterübergabe ändert sich gegenüber V1.3 nicht.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C2BC**ChangeDiskDevice**

Mit dieser Routine kann die Gerätenummer der derzeit aktuellen Floppy softwaremäßig auf einen anderen Wert gebracht werden. Hierzu muß ChangeDiskDevive die Nummer (8-11) im Akku übergeben werden.

Rückgabe: Error-Status im X-Register (siehe Anhang)

C2BF **RstrFrmDialog**

Wird beim Anklicken der Icons einer Dialogbox (z.B. OK, CANCEL etc.) automatisch aufgerufen. Hierdurch wird das Fenster abgebaut, die eingefrorenen und geretteten Jobs werden reaktiviert (Pull-Down-Menüs, Prozesse etc.), und in \$02 die Nummer des angeklickten Icons übergeben. Falls das Fenster jedoch von einer durch die Dialogbox aufgerufenen Unterroutine manuell abgebaut werden muß (s. DoDlgBox= \$C256), kann vor dem Aufruf von RstrFrmDialog in \$02 ein beliebiger Wert übergeben werden, der zur Auswertung an das Hauptprogramm weitergegeben wird.

C2C2 **Panic**

Ausgabe von 'System Error in \$xxxx'. Die Routine zieht sich die Adresse vom Stack, bei der der Prozessor in ein BRK gelaufen ist, und wandelt sie in eine vierstellige ASCII-Zahl um. Diese Zahl wird in einem Fenster als Abschiedsmitteilung ausgegeben.

C2C5 **BitOtherClip**

BitOtherClip besitzt eine ähnliche Funktion wie BitmapClip=\$C2AA, bietet jedoch den Vorteil, daß die Grafikdaten nicht direkt im Speicher vorliegen müssen. Statt dessen wird BitOtherClip die Einsprungadresse einer Routine mitgeteilt, die bei jedem Aufruf das nächste Byte der komprimierten Grafik (dazu s. BitmapUp=\$C142) in den Akku holt. Diese Routine wird in den meisten Fällen die Grafikdaten von der Floppy holen (z.B. mittels ReadByte=\$C2B6). Sie darf dabei die Speicherstellen \$02 bis \$1D nicht überschreiben. Außer den schon bei der Beschreibung von BitmapClip angegebenen Positionsangaben erwartet BitOtherClip noch die folgenden Informationen:

\$02,\$03: Zeiger auf einen 134 Bytes umfassenden Datenpuffer.

\$1C,\$1D: Adresse der oben beschriebenen Routine
\$1E,\$1F: Adresse einer Routine, die den Zeiger in \$02,\$03 wieder auf den Anfang des Datenpuffers setzt.

6.5 Speicher und Variablen

Wenn Sie die Funktionsweise des GEOS KERNAL an Hand eines disassemblierten Listings verstehen wollen und dabei unsere Sprungtabelle benutzen, so werden Sie immer wieder auf Speicherstellen stoßen, die das KERNAL benutzt. Eine enorme Erleichterung ist es dann, wenn Sie in einer Tabelle viele Speicherstellen und deren Funktion nachschauen können. Zusätzlich müssen Sie natürlich beim Schreiben eigener Programme über die von GEOS benutzten Speicherstellen informiert sein. Daher haben wir alle uns im Augenblick bekannten Speicherstellen und deren Bedeutung aufgelistet.

Beim Schreiben eigener Programme unter GEOS müssen Sie natürlich auch genau wissen, welche Speicherbereiche Sie verwenden dürfen. Deshalb enthält dieses Kapitel auch eine Speicher-aufteilung unter GEOS

6.5.1 Die GEOS-Variablen

Beginnen wir mit den von GEOS verwendeten Variablen und ihrer Bedeutung:

\$02-\$21

Pseudo-Register zur Kommunikation mit den KERNAL-Routinen. Werden für verschiedene Zwecke verwendet.

\$22-\$23

Pointer auf das aktuelle Muster

\$2E

Statusbyte für Schriftart (bold, invers, outline etc.)

\$2F

Bit 7=1: sichtbarer Bildschirm (\$A000) angewählt
 Bit 6=1: unsichtbarer Bildschirm (\$6000)
 Auch beide Bits=1 möglich!

\$0030

Maus-Flag

Bit 7=0: Maus abgeschaltet
 Bit 6=1: Menumodus: Bereichstest ein

\$0039

Joystick-Flag

Bit 7=0: keine Zeichen im Tastaturpuffer
 Bit 6=1: keine geänderte Joystick-Bewegung
 Bit 5=1: Feuerknopf gedrückt

\$003A

Aktuelle Maus-X-Positon (LOW)

\$003B

Aktuelle Maus-X-Position (HIGH)

\$003C

Aktuelle Maus-Y-Position

\$008E

Aktueller Wert bei Busoperationen

\$8400

Ab hier steht meistens der Directory-Eintrag (30 Bytes):

\$8400

DOS-Filetyp

\$8401

Track der Daten

\$8402	Adresse der oben beschriebenen Routine
Sektor der Daten	
\$8403-\$8412	Name der Datei, aufgefüllt mit \$A0
\$8413	Track des Info-Blocks
\$8414	Sektor des Info-Blocks
\$8415	Struktur (0 SEQ, 1 VLIR)
\$8416	GEOS-Filetyp (0-9)
\$8417	Jahr
\$8418	Monat
\$8419	Tag
\$841A	Stunde
\$841B	Minute
\$841C/D	Länge des Files in Blocks
\$8489	Nummer des aktuellen Ausgabegerätes
\$848A	Tabelle der Statuswerte der Ausgabegeräte (Speeder-Status)
\$848B	GEOS-FORMAT V1.0? 0 nein, \$FF ja
\$8496	Aktuelle Record-Nummer

\$8497

Jobadresse: Anzahl der vorhandenen Records

\$8498

Flag, entscheidet, ob beim Schließen eines Records ein Update notwendig ist.

\$849B/9

Vektor zur Erweiterung der MAIN LOOP um eigene Routine, die natürlich mit RTS enden muß.

\$849D/9E

IRQ-Vektor

Hier kann eine eigene Routine eingebunden werden, die bei jedem IRQ aufgerufen wird. Will man den GEOS-IRQ weiter ausführen, endet man seine IRQ-Routine nicht mit RTI, sondern mit JMP \$C100.

\$84A1/2

Jobadresse: Feuerknopf gedrückt

In diesen Vektor kann eine Adresse eingegeben werden, die angesprungen werden soll, falls der Feuerknopf gedrückt worden ist.

\$84A3/4

Jobadresse bei Tastatureingabe

Hier kann eine Adresse eingegeben werden, die angesprungen werden soll, falls ein Zeichen eingegeben wurde. Das Zeichen wird dieser Routine im Akku übergeben.

\$84A5/6

Jobadresse: Hebelbewegung Joystick

\$84A7/8

Jobadresse: Maus hat Bereich verlassen (\$84B6 nicht '0')

\$84A9/A

Jobadresse: Mausknopf wurde außerhalb des angegebenen Bereich betätigt.

\$84AF/B0

Indirekte Einsprungadresse bei BRK

Normalerweise zeigt dieser Vektor auf die Ausgaberroutine für die Meldung 'System error in \$XXXX'.

\$84B4

CURSOR-FLAG:

Bit 7=1: Cursor eingeschaltet

Bit 6=1: Cursor hell

Bit 6=0: Cursor dunkel

\$84B5

Flag für: Art der Invertierung eines Icons (Bit 7,6)

Bit 7=1: Icon invertiert nur kurz

Bit 7=0: Icon invertiert nicht

Bit 6=1: Icon bleibt invertiert

\$84B6

Statusregister Maus

Bit 7=1: Kontakt obere Grenze (Bereich 1)

Bit 6=1: Kontakt untere Grenze

Bit 5=1: Kontakt linke Grenze

Bit 4=1: Kontakt rechte Grenze

Bit 3=1: Maus außerhalb des markierten 2. Bereichs

\$84B7

Pull-Down-Menü: aktuelle Untertabelle

\$84B8

Obere Grenze Maus

\$84B9	
Untere Grenze	Code für aktuelle Horizontalposition
\$84BA/B	
Linke Grenze	300: rechts 301: hoch 302: links
\$84BC/D	
Rechte Grenze	300: runter 301: Neutralstellung
\$84C1	
Obere Grenze Maus (2. Bereich)	Aktuelle Winkelwert
\$84C2	
Untere Grenze (2. Bereich)	Zahlzahl, durch IRQ generiert
\$84C3/4	
Linke Grenze (2. Bereich)	Zykluszahl
\$84C5/6	
Rechte Grenze (2. Bereich)	38516 Jahr 38517 4. Quart 38518 Tag 38519 Stunde 38520 Minute 38521 Sekunde
\$8501	
Maximale Maus-Geschwindigkeit	38522 Sekunde
\$8502	
Minimale Maus-Geschwindigkeit	38523
\$8503	
Acceleration	38524 38525 38526
\$8504	
Aktuelles Zeichen, aus Tastaturpuffer gelesen	38527 38528
\$8505	
Code für aktuelle Feuerknopfposition	38529 38530
Bit 7=0: Knopf gedrückt	38531
Bit 7=1: Knopf nicht gedrückt	38532

\$8506

Code für aktuelle Hebelposition

\$00: rechts

\$02: hoch

\$04: links

\$06: runter

\$FF: Neutralstellung

\$8507

Aktuelle Mausgeschwindigkeit

\$850A/B

Zufallszahl, durch IRQ generiert

*Systemzeit:***\$8516** Jahr**\$8517** Monat**\$8518** Tag**\$8519** Stunde**\$851A** Minute**\$851B** Sekunde**\$86C0**

Pull-Down-Menü Status-Byte

Bit 7=1: Menü vertikal

Bit 7=0: Menü horizontal

Bit 6=0: Mausgrenzen werden auf ganzen Bildschirm
gesetzt

Bit 6=1: alte Mausgrenzen

\$86C1

Oberer Rand des Menüs

\$86C2

Unterer Rand des Menüs

- \$86C3/4**
Linker Rand des Menüs
- \$86C5/6**
Rechter Rand des Menüs
- \$86C7/8**
Adresse der Untertabelle
- \$86F1**
Timer-Werte (zählend/aktiv) für Puffer 1 (LOW/HIGH)
- \$8719**
Ab hier: Statusbytes der Programmjobs
- \$872D**
Ab hier: Jobadressen der Programmjobs
- \$8755**
Timer-Werte (für Initit/Neueintrag) der Programmjobs
- \$877D**
Anzahl der Programmjobs
- \$877E**
Anzahl der Delayjobs
- \$877F**
Ab hier: Timer-Werte für Verzögerung (LOW)
- \$8793**
Ab hier: Timer-Werte für Verzögerung (HIGH)
- \$87A7**
Adressen der Delay-Jobs (LOW)
- \$87BB**
Adressen der Delay-Jobs (HIGH)

\$87D7	Leserzeiger für Tastaturpuffer
\$87D8	Schreiberzeiger für Tastaturpuffer
\$87D9	Repeatflag für Tastatureingabe
\$87DA-\$87E9	Tastaturpuffer
\$87EA	Aktuelles Zeichen bei Eingabe durch IRQ in Tastaturpuffer
\$880B	Temporärer Speicher für Akku bei IRQ
\$8850/1	Accessory Rücksprung (Version 1.2)
\$8852	Accessory Rücksprung: STACK (Version 1.2)
\$8853/4	Fenster Rücksprung
\$8855	Fenster Rücksprung: STACK
\$8860	Temporärer Speicher für \$01
\$8861	Spritestatus: 8 bit für Sprite an/aus
\$8863/4	Einsprungsadresse für Floppy-RAM bei Übertragung

\$8865

Daten (Track) für Floppy-RAM bei Übertragung

\$8866

Daten (Sektor) für Floppy-RAM bei Übertragung

\$886E

Flag für: File aus BORDER gelesen (\$FF, \$00 sonst)

\$886F

Track: Eintrag in DIR gefunden

\$8870

Sektor: Eintrag in DIR gefunden

\$8871/2

File-Eintrag gefunden, zeigt auf Stelle in Puffer (\$8000)

\$8873

Track des gesuchten VLIR-Files

\$8874

Sektor des gesuchten VLIR-Files

\$8875

Anzahl der Leseversuche

6.5.2 Speicheraufbau

Natürlich reicht es für die Erstellung eigener Programme nicht, grundsätzlich über den Ablauf von Applications informiert zu sein. Damit eine Zusammenarbeit mit dem GEOS KERNAL reibungslos funktionieren kann, müssen wir wissen, welchen Speicher wir für eigene Programme benutzen können. Deshalb wollen wir Ihnen die Speicheraufteilung unter GEOS erklären.

\$0000-\$007F

Vom GEOS KERNAL benutzte Zeropage.

\$0080-\$00FF

Zeropage, die von den Applications benutzt werden kann. Ein Accessory sollte benutzte Speicherstellen vorher retten.

\$0100-\$01FF

Prozessor-Stack

\$0200-\$03FF

Zum größten Teil ungenutzt. Der Bereich von \$0300-\$0333 enthält die üblichen Vektoren des C64-Betriebssystems. Allerdings setzt GEOS diese Zeiger teilweise auf eigene Werte.

\$0400-\$5FFF

Platz für Applications. Wenn Sie zusätzlich zum DESK TOP Routinen laufen lassen wollen, so können Sie den Bereich von \$4570-\$496E benutzen.

Achtung: Bei der Autostart-Programmen müssen Sie bedenken, daß ab \$5000 ein Teil des Boot-Programms liegt!

\$6000-\$7F3F

2. Bildschirm. Hier rettet GEOS KERNAL den Hintergrund vor der Fensterausgabe und dem Herunterklappen eines Menüs.

\$7F40-\$7FFF

Platz für Application.

\$8000-\$80FF

Puffer 1 für Floppy-Daten

\$8100-\$81FF

Puffer 2 (meist aktueller INFO-Sektor)

\$8200-\$82FF

Puffer 3 (hier liegt normalerweise die BAM, die von GEOS verwaltet wird)

\$8300-\$83FF

Puffer 4 (oft benutzt für Ablage der Zeiger auf die Tracks und Sektoren, bei Lade- und Speichervorgängen.)

\$8400-\$8BFF

Daten des GEOS KERNAL (Jobadressen, Filename, Diskettenname, Spritedaten).

\$8C00-\$8FE9

Farb-RAM

\$8FF8-\$8FFF

Zeiger auf die Spritedaten

\$9000-\$9FFD

GEOS KERNAL (PRG)

\$A000-\$BF3F

1. Bildschirm. Dieser Speicher entspricht normalerweise dem, was Sie auf dem Bildschirm sehen.

\$BF40-\$BFFF

GEOS KERNAL (Daten : beispielsweise Aufbau der Klickfelder)

\$C000-\$FFFF

GEOS KERNAL (PRG)

Interessant ist, daß das GEOS KERNAL ab \$9000 beginnt. Wenn Sie mit SPECIAL: BASIC GEOS verlassen und später einen Warmstart machen wollen, müssen Sie unbedingt diesen Speicherbereich vor dem BASIC-Interpreter schützen, indem Sie die Obergrenze des Speichers für BASIC auf \$9000 setzen. Leider setzt GEOS KERNAL diese Werte auf \$A000. Sie können dieses Problem lösen, wenn Sie die zusätzlichen Änderungen im Kapitel "3.5.3 GEOS Warmstart" vornehmen. Dort wird dann das KERNAL so abgeändert, daß diese Adressen immer auf \$9000 gesetzt werden. Ohne diese Änderung müssen Sie direkt nach "SPECIAL: BASIC" folgende Werte im Direktmodus poken:

POKE 52,144

POKE 54,144

POKE 56,144

Auf keinen Fall dürfen vor diesen Pokes Stringoperation erfolgen.

7. Anhang

Fehlermeldungen des GEOS-Betriebssystems

Beim Aufruf von Diskettenroutinen in GEOS werden auftretende Fehlermeldungen von den niedrigsten Unterprogrammen im X-Register bis zur aufrufenden Routine des Anwenders hochgereicht (siehe Sprungtabelle in 6.4). Im folgenden finden Sie nun die zugehörigen Fehlermeldungen im Klartext.

Fehlernummer	Bedeutung
\$02	Es wurde versucht, einen ungültigen Track zu lesen (gleich 0 bzw. größer als 35).
\$03	Anzahl freier Blocks auf der Diskette reicht nicht aus. Hier hilft manchmal ein Validate der Diskette vom DESK TOP aus.
\$04	Kein freier Eintrag im Directory mehr vorhanden. Eventuell genügt es, vor dem Aufruf der entsprechenden Routine in \$16 einen kleineren Wert einzutragen, damit schon auf den ersten Seiten ein Eintrag gesucht wird.
\$05	Datei nicht gefunden (File not found).
\$06	Es sind Probleme mit der BAM aufgetreten (z.B. falls versucht worden ist, einen Block freizugeben, der schon frei ist).
\$07	Es ist versucht worden, auf einen VLIR-File zuzugreifen, das noch nicht geöffnet worden ist.
\$08	Es ist versucht worden, auf ein nicht existierenden Record einer VLIR-Datei zu positionieren.
\$09	Die maximale Anzahl Records (127) der VLIR-Datei ist bereits erreicht.

- \$0A Vorgang ist für eine Datei dieser Struktur nicht erlaubt.
- \$0B Die zu ladende Datei enthält mehr Werte, als durch die Parameter der Laderoutine zugelassen worden sind.
- \$0C Kann für eigene Routinen verwendet werden.
- \$0D Spezifiziertes Gerät ist nicht ansprechbar (Device not present).
- \$20 Header der Datei nicht gefunden.
- \$21 Keine Sync-Markierungen auf der Diskette vorhanden (keine oder unformatierte Diskette, Diskettenschacht offen).
- \$23 Defekter Block auf der Diskette.
- \$25 Verify Error (Diskette oder Laufwerk defekt).
- \$26 Die Diskette ist durch einen Aufkleber schreibgeschützt.
- \$27 Illegaler Headerblock.
- \$29 ID-Kennung stimmt nicht mit der aktuellen ID-Kennung, die das Kernal kennt, überein (tritt auf, falls OpenDisk vergessen wurde).
- \$2E Fehler bei der Dekodierung eines Bytes aufgetreten.
- \$73 Die DOS-Version der eingelegten Diskette ist nicht zulässig.

Der ASCII-Code unter GEOS

Die ASCII - Tabelle

L \ H	0	1	2	3	4	5	6	7
0	NUL	GrSt	SP	0	@	P	`	p
1		NOP	!	1	A	Q	a	q
2		RY1	"	2	B	R	b	r
3		RY0	#	3	C	S	c	s
4		GOx]	4	D	T	d	t
5		GOy	%	5	E	U	e	u
6		GOxy	&	6	F	V	f	v
7		CSet	'	7	G	W	g	w
8	BS	Bld1	(8	H	X	h	x
9	FS	It1cl)	9	I	Y	i	y
A	LF	Out11	*	:	J	Z	j	z
B	HM	Pln	+	;	K	[k	<
C	UP		,	<	L	\	l	
D	CR		-	=	M]	m	>
E	UL1		.	>	N	^	n	~
F	UL0		/	?	O	_	o	DEL

Für die deutsche GEOS-Version 1.3 gilt die folgende Zuordnung:

USA-ASCII	Deutsch-ASCII
-----------	---------------

#	°
[Ä
\	Ö
]	Ü
{	ä
	ö
}	ü*

Die Abkürzungen der Steuerzeichen (kleiner als \$20) haben die folgenden Bedeutungen:

(*) NUL	Endekennzeichen eines Textes
BS	Backspace (ein Zeichen zurück)
FS	Forwardspace (ein Zeichen vor; in GEOS V1.2 nicht korrekt implementiert)
LF	Line feed (Zeile nach unten)
HM	Home (Cursor nach links oben)
UP	Zeile nach oben
CR	Carriage Return ("Wagenrücklauf")
UL1	Underline On
UL0	Underline Off
(*) GrSt	(gefolgt von Daten) gibt einen 'Grafik-String' aus. Der Aufbau der Daten ist der gleiche wie bei der Funktion GraphikString (s. Kapitel 6)
NOP	bewirkt nichts
RV1	Reverse on
RV0	Reverse off
(*) GOx	plus 2 Byte: neue X-Position des Cursors festlegen (Lo/Hi)
(*) GOy	plus 2 Byte: neue Y-Position des Cursors festlegen
(*) GOxy	plus 3 Byte: neue X- und Y-Position festlegen
(*) Cset	wird z.B. in GEOPAINT dazu benutzt, einen neuen Zeichensatz zu initialisieren. Bei der normalen Textausgabe mittels 'PutString' werden die folgenden 3 Bytes, die Auskunft über den neuen FONT und die Schriftart geben, einfach ignoriert.
Bld1	Bold einschalten
Itlc1	Italics einschalten
Outl1	Outline einschalten
Pln	zurückschalten auf PLAIN TEXT

Das einzige Steuerzeichen, das größer ist als \$20, lautet DEL (Delete). Es löscht das vorherige Zeichen. Außerdem gibt es im System-Zeichensatz (BSW-9-Point) noch den Wert \$80, der das Commodore-Symbol für die Menu-Shortcuts codiert.

Wichtig: Die mit (*) gekennzeichneten Steuerzeichen dürfen nur in einem Textstring auftauchen, der mittels 'PutString' (\$C148) ausgegeben wird. Diese Routine erwartet in \$02/\$03 einen Zeiger auf den auszugebenden Text, der (wie immer in GEOS) mit NUL (s.o.) enden muß. Alle anderen Zeichen dürfen der Routine 'PutChar' (\$C145) einzeln im Akku übergeben werden. Die Ausgabe erfolgt dabei an der in \$18/\$19 (X) bzw. \$05 (Y) eingetragenen Position (dies gilt auch für 'PutString').

Unterschiedliche Begriffe und Namen bei GEOS V1.2 - GEOS V1.3 deutsch

Menüzeile GEOS V1.2

geos	file	view	disk	special
GEOS info	open	by icon	open	BASIC
deskTop info	duplicate	by size	close	RESET
select printer	rename	by type	rename	A-Link
select input	info	by date	copy	
preference mgr	print	by name	add drive	
			validate	
			format	

Menüzeile GEOS V1.3 deutsch

geos	Datei	Anzeige	Diskette	spezial
GEOS info	öffnen	Piktogramme	öffnen	BASIC
deskTop Info	Duplizieren	nach Größe	Schließen	RESET
Drucker wählen	Umbenennen	nach Typ	Umbenennen	
Input wählen	Info	nach Datum	Kopieren	
Voreinstellung	Drucken	nach Namen	Laufwerk anmelden	
			Aufräumen	
			Formatieren	

Programmnamen**GEOS V1.2**

GEOS
 GEOS
 GEOS
 GEOPAINT
 GEOWRITE
 DESK TOP
 BACKUP
 alarm clock
 preference mgr
 Preferences
 note pad
 Notes
 photo manager
 text manager
 calculator
 University
 California
 Cory
 Dwinelle
 Roma
 Joystick

GEOS V1.3 deutsch

GEOS
 BOOT GEOS BOOT
 KERNAL GEOS KERNAL
 GEOPAINT
 GEOWRITE
 DESK TOP
 Diskettenkopier
 Wecker
 Voreinstellung
 Preferences
 Notizblock
 Notes
 Foto-Manager
 Text-Manager
 Rechner
 University_GE
 California_GE
 Cory_GE
 Dwinelle_GE
 Roma_GE
 STEUERHEBEL
 COMM 1351
 INKWELL
 KOALA TASTATUR
 KONFIGURIEREN
 RBOOT

Schlagwortverzeichnis

Accessory

Hilfsmittel

Anwendung

Programm, das nur von DESK TOP aus geladen werden kann. Es nutzt die Möglichkeiten, die GEOS als fertige Programmteile bietet. (GEOPAINT, GEOWRITE)

Application

Anwendung

Arbeitskopie

Um mit GEOS arbeiten zu können, müssen Sie den Inhalt der Originaldiskette auf eine andere Diskette kopieren und einige der Programme löschen, um Platz für das Arbeiten zu haben. Eine so erstellte Kopie nennen wir Arbeitskopie. Ab GEOS V1.3 erkennt das DESK TOP die Originaldisketten und Kopien davon und verhindert einige Aktionen auf solchen Disketten. Da auch das mitgelieferte Programm "Diskettenkopier" keine Kopien der Originaldisketten zulässt, sollten Sie eine Diskette formatieren und über die Border die notwendigen Dateien auf die Diskette kopieren.

Assembler

Computersprache, die noch eine relativ große Nähe zur Maschinensprache hat. Während Maschinensprache aus einem wenig anschaulichen Code besteht, den der Rechner direkt abarbeiten kann, gibt es in Assembler für den Benutzer verständlichere Symbole und Abkürzungen.

Backup

Bezeichnung für eine Sicherheitskopie, auf der der Inhalt der Originaldiskette gespeichert ist. Ebenso heißt ein Programm auf der Diskette, mit dem man Kopien herstellen kann. Dieses Programm kann von GEOS aus geladen werden (Doppelklick) oder von BASIC aus mit:

```
LOAD"BACKUP",8  
RUN
```

Bei der deutschen Version von GEOS 1.3 heißt dieses Programm "Diskettenkopier".

BASIC

Eine der Sprachen, die ein Computer indirekt versteht. Ein Computer versteht direkt nur eine für uns wenig anschauliche Sprache (Maschinensprache), BASIC kann aber in diese Sprache übersetzt werden. Diese Arbeit erledigt der BASIC-Interpreter für Sie. BASIC ist in den C64 fest 'eingebaut', so daß diese Sprache jederzeit zur Verfügung steht.

BASIC-Interpreter

Programm, das die Umwandlung von BASIC in Maschinensprache vornimmt. Wenn Sie ein BASIC-Programm geschrieben haben und es mit RUN starten, werden die Befehle in kleinste Aufgaben zerlegt, die der Computer abarbeiten kann.

Benutzeroberfläche

Mit vielen Computern kann man nur in einer Programmiersprache (zum Beispiel BASIC) kommunizieren. Bevor der Rechner also etwas für Sie erledigen kann, müssen Sie die Sprache lernen. Mit einer Benutzeroberfläche hingegen können Sie auch ohne Computerkenntnisse einen Rechner bedienen. Dabei werden wichtige Elemente für die Arbeit mit dem Computer grafisch dargestellt, und die Arbeit mit diesen Elementen ist be-

kannten Abläufen nachempfunden. So werden beispielsweise unter GEOS Files als grafische Symbole dargestellt, und das Löschen eines solchen Files erfolgt, indem Sie das Symbol zum Papierkorb bewegen und dort ablegen.

Betriebssystem

Mit diesem Begriff beschreibt man ein Programm, das die wichtigsten Grundfunktionen des Computers ermöglicht. Ohne ein solches Programm kann ein Rechner nichts. Im C64 ist das Betriebssystem fest gespeichert, so daß es auch nach dem Ausschalten erhalten bleibt.

Bildschirmspeicher

Teil des Computerspeichers. In ihm werden die Informationen abgelegt, die das Aussehen des Bildschirms bestimmen. Ohne GEOS besteht der Bildschirmspeicher meist aus 1000 Speicherplätzen, die die Zahlen für die 40*25 möglichen Zeichen auf dem Bildschirm enthalten. Es genügt also, die richtige Zahl in den Bildschirmspeicher zu schreiben, damit an der entsprechenden Stelle auf den Bildschirm beispielsweise ein 'A' auftaucht. Im Gegensatz dazu gibt es noch den Hires-Bildschirm. Bei diesem Bildschirm ist man nicht auf die vorgefertigten Zeichen beschränkt, sondern man kann das Aussehen des Bildschirms aus einzelnen Punkten zusammensetzen. GEOS benutzt diesen Hires-Bildschirm.

Booten

So wird der Vorgang bezeichnet, bei dem ein Programm geladen und gestartet wird. Geos können Sie nur von der Originaldiskette booten.

Border

(Rand, Grenze) Damit ist ein bestimmter Bereich auf dem Bildschirm, den Sie im DESK TOP vor sich haben, gemeint. Dieser

Bereich liegt unter dem hellen GEOS-Fenster etwa in Höhe des Druckersymbols. Wenn man Icons auf der Border ablegt, kann man ihre Reihenfolge ändern oder sie kopieren.

Cursor

Meist blinkendes Symbol auf dem Bildschirm. Er zeigt dem Benutzer, daß der Computer Eingaben erwartet. Beim C64 ohne GEOS hat er die Form eines Quadrates, in GEOS erscheint er als senkrechter Strich. Der Cursor gibt in GEOWRITE die Stelle an, an der der nächste Buchstabe im Text erscheint.

DEL-Taste

Sie befindet sich oben rechts auf der Tastatur und löscht ein Zeichen links vom Cursor

DESK TOP

Programm, das eine Benutzeroberfläche erstellt, die einem Schreibtisch ähnelt. Auf diesem 'Schreibtisch' können Gegenstände bewegt und bearbeitet werden. Zusätzlich gibt es einen Drucker und einen Papierkorb. DESK TOP versucht, die Vorgänge auf dieser Benutzeroberfläche den Vorgängen an einem gewöhnlichen Schreibtisch nachzuempfinden.

Destination Disk

Zieldiskette; damit ist die Diskette gemeint, auf die Sie kopieren wollen.

Directory

Inhaltsverzeichnis der Diskette; damit die Floppy bei der Suche nach einem Programm nicht die ganze Diskette durchsuchen muß, wird zu jedem abgespeicherten File ein File-Eintrag im Inhaltsverzeichnis der Diskette erstellt, der wichtige Informationen (Name, Lage auf der Diskette) enthält.

Druckeranpassung

Da es unterschiedliche Druckertypen gibt, kann man nicht jeden Drucker an jeden Computer anschließen. Meist muß noch ein Programm geladen werden, das die Verständigung zwischen einem Computer und einem speziellen Drucker herstellt. Solche Programme nennt man Druckeranpassung. Sie müssen die für Ihren Drucker passende Anpassung wählen, bevor Sie etwas ausdrucken können.

Einzelstaptsimulator

Programm, das es ermöglicht, ein anderes Programm Schritt für Schritt abzuarbeiten und sich nach jedem Schritt wichtige Informationen anzuschauen. Im Kapitel 5 dieses Buches ist ein solches Programm abgedruckt, mit dem Sie GEOS Schritt für Schritt bei der Arbeit zuschauen können.

Fenster

Bezeichnung für einen Teil des Bildschirms, der durch einen Rahmen vom übrigen Teil abgetrennt ist. Dieser Teil wird dann für eine gewisse Zeit als eigenständiger Bildschirm behandelt. GEOS benutzt Fenster, um Informationen auszugeben (Fehlermeldungen) oder von Ihnen zu erfragen (Eingabe des Namens für eine Grafik). Das besondere an Fenstern ist, daß etwas auf dem Bildschirm ausgegeben werden kann und hinterher trotzdem der ursprüngliche Bildschirm erhalten ist. Dazu muß der Teil des Bildschirmspeichers, der durch das Erstellen des Fensters geändert wird, an einer anderen Stelle gerettet und später wieder dort hineingeschrieben werden.

File

Kartei, Ansammlung von Daten. Ein File ist ein zusammenhängender Datenblock, wobei auch ein Programm eigentlich aus Daten besteht, die erst vom Rechner als Befehle interpretiert werden. Unter GEOS wird jedes File nicht nur durch einen

Namen (Filename), sondern auch durch ein Symbol (Icon) dargestellt. Texte, Grafiken, Programme, Photoalben, Druckeranpassungen usw. sind Files.

File-Eintrag

Wichtige Daten zu einem File werden als File-Eintrag im Inhaltsverzeichnis der Diskette abgelegt. Dazu gehören: Name, Größe, Start auf der Diskette, Art des Files usw.

Floppy

Diskettenlaufwerk, ermöglicht das Abspeichern von Programmen und Daten, so daß diese auch nach dem Ausschalten des Rechners erhalten bleiben.

Floppypuffer

siehe Puffer

Formatieren

Damit bezeichnet man den Vorgang, mit dem eine Diskette für die Arbeit vorbereitet wird. Dabei wird sie völlig gelöscht. Sie sollten alle Disketten, die Sie für Ihre Arbeit mit GEOS benötigen, von GEOS formatieren lassen, weil sich dieser Vorgang etwas vom normalen C64-Formatier-Befehl unterscheidet. GEOS gibt der Diskette ein besonderes Format.

GEOS

(Graphic Environment Operating System) Betriebssystem mit grafischer Umgebung. GEOS ermöglicht eine Benutzeroberfläche mit Fenstern, Pull-Down-Menüs, Icons und einer Maus. Außerdem bietet es schnelle Diskettenoperationen und ein eigenes Fileformat.

GEOS-Fenster

Damit ist der helle Bereich etwa in der Mitte des Bildschirms bezeichnet, in dem DESK TOP bis zu acht Files gleichzeitig anzeigt.

GEOPAINT-Fenster

Damit ist der Teil des Bildschirms gemeint, in dem Sie Ihre Grafik verändern können. In diesem Fenster wird nur ein kleiner Ausschnitt der gesamten Grafik gezeigt. Diesen Ausschnitt können Sie verschieben.

Hardcopy

Mit Hardcopy wird ein auf dem Drucker erstelltes Bild bezeichnet, das den Bildschirm zu einem bestimmten Zeitpunkt festhält. Die gesamte Bildschirminformation (oft allerdings ohne Farbe, wenn der Drucker keine Farbmöglichkeit hat) wird dabei auf den Drucker übertragen.

Hilfsmittel

Programm, das praktisch jederzeit zur Verfügung steht. Allerdings können Sie nicht mehrere Hilfsmittel gleichzeitig benutzen. So können Sie beispielsweise mit GEOPAINT eine Grafik erstellen und zwischendurch mit dem Taschenrechner etwas ausrechnen.

Icon

Grafisches Symbol. GEOS stellt beispielsweise Ihre Texte und Grafiken auf dem Bildschirm als Icons dar, wenn Sie mit DESK TOP arbeiten. Dabei kann man einem Icon meist an der Form ansehen, was es repräsentieren soll. Auch der Drucker und der Papierkorb sind Icons.

Interrupt

Vorgang im Computer, bei dem der eigentliche Rechner seine Arbeit kurz unterbricht, etwas anderes ausführt und dann an der vorherigen Stelle weitermacht. In GEOS wird durch Interrupts die Maus gesteuert und die Tastatur abgefragt.

Kopierschutz

Disketten kann man grundsätzlich kopieren wie Kassetten. Da dies aber nicht dem Interesse der Hersteller entspricht, verhindern sie das erfolgreiche Kopieren der verkauften Diskette. Meist wird dabei etwas an der Diskette geändert, und diese Änderung wird vom Programm abgefragt. Ist sie auf der eingelegten Diskette nicht vorhanden, so arbeitet das Programm nicht weiter. Diese Änderungen, also der Kopierschutz, können normalerweise nicht kopiert werden. GEOS hat einen sehr guten Kopierschutz, der nur beim Booten überprüft wird.

Maus

Begriff, der eng mit der Benutzeroberfläche zusammenhängt. Die Maus stellt sozusagen Ihren Finger auf dem Bildschirm dar. Mit diesem Finger können Sie Symbole anfassen (anklicken) und transportieren. In GEOS kann die Maus verschiedene Formen annehmen. Üblicherweise wird sie durch einen Pfeil auf dem Bildschirm dargestellt. In GEOPAINT kann sie auch wie ein Fadenkreuz oder wie ein schmaler Stift aussehen. Die Maus wird mit dem Joystick bewegt. Der Feuerknopf bewirkt das Klicken (Anfassen) an der Stelle, an der die Spitze des Pfeils ist.

Menüzeile

Zeile auf dem Bildschirm (meist oben links), die verschiedene Begriffe enthält. (Im DESK TOP: GEOS, FILE, VIEW, DISK, SPECIAL) Jeder Begriff steht für ein Untermenü, das eine Reihe weiterer Befehle enthält. Wenn Sie einen Begriff in der Menüzeile anklicken, werden darunter in einem Fenster die zugehörigen Befehle angezeigt.

Originaldiskette

Damit meinen wir die Diskette, die Sie käuflich erworben haben oder die dem C64 beim Kauf beigelegt war. Bitte benutzen Sie die Originaldiskette nur zum Laden von GEOS oder um Kopien herzustellen. Benutzen Sie für die eigentliche Arbeit stets Arbeitskopien.

Pixel

Kleinster darstellbarer Punkt auf dem Bildschirm.

Proportionalschrift

Besondere Schrift, bei der die Buchstaben unterschiedlich viel Platz benötigen. Dadurch passen beispielsweise viel mehr 'i' in eine Zeile als 'W'

Puffer

Mit Puffer bezeichnet man grundsätzlich einen Speicherbereich, in dem Daten kurzfristig zwischengespeichert werden. Wir meinen damit meist die Floppypuffer. Dort werden Sektoren, die auf die Diskette geschrieben oder von ihr gelesen werden sollen, zwischengespeichert.

Pull-Down-Menü

siehe Menüzeile

Restore-Taste

Sie befindet sich oben rechts auf der Tastatur in der zweiten Reihe und wird von GEOS dazu benutzt, ein erneutes Booten auszulösen. Wenn Sie also GEOS durch 'BASIC' oder Laden eines BASIC-Programms verlassen haben, genügt es, die Originaldiskette einzulegen und die Restore-Taste zu drücken, um GEOS zu booten.

Scroll-Feld

Ein kleines Rechteck, das meistens zwei Pfeile enthält. Mit diesen beiden Pfeilen können Sie den Inhalt des Fensters in Pfeilrichtung verschieben (Scrollen). Dabei verschwindet ein Teil des Inhalts, und an der gegenüberliegenden Seite erscheinen weitere Zeilen. Das Scroll-Feld wird meistens benutzt, wenn nicht die gesamte Information im Fenster gleichzeitig darstellbar ist.

Scrollen

Vorgang, der etwa so funktioniert wie die Anzeige bei einem Geldspielautomaten. Dort dreht sich ja eine Walze an einem Fenster vorbei, so daß jeweils der Inhalt des Fensters nach oben verschwindet und von unten etwas Neues im Fenster gezeigt wird.

Seitenzeiger

Begriff aus GEOPAINT und GEOWRITE; in GEOPAINT meint er ein Rechteck unten in der Statuszeile. Dieses Rechteck entspricht der gesamten Seite, und das kleine Rechteck, das zu Beginn jeweils in der oberen linken Ecke erscheint, gibt die Lage des GEOPAINT-Fensters an. Bei GEOWRITE befindet er sich oben auf dem Bildschirm neben dem Namen des Textes und zeigt zusätzlich die Seitennummer an.

Sektor

Die Diskette ist in mehrere Tracks (Spuren, Ringe) unterteilt, die aus einer bestimmten Anzahl von Sektoren (Abschnitte, Segmente) bestehen. Die Floppy kann einen Sektor auf der Diskette mit Hilfe bestimmter Markierungen finden. Er enthält maximal 256 Daten. Wenn Sie ein Programm speichern, so wird es in passende Abschnitte aufgeteilt, die dann in jeweils einem Sektor gespeichert werden.

Sicherheitskopie

Kopie der Originaldiskette, die bis auf den Kopierschutz identisch ist. Von der Sicherheitskopie können Sie zwar GEOS nicht booten, aber Sie können damit die Originaldiskette im Notfall wiederherstellen (Nicht bei GEOS V1.3). Bitte arbeiten Sie nie mit der Originaldiskette. Benutzen Sie dafür Arbeitskopien.

Source Disk

Herkunftsdiskette; damit ist die Diskette gemeint, von der Sie kopieren wollen.

Sprite

Grafisches Symbol, das auf dem Bildschirm bewegt werden kann. Beim C64 bestehen sie aus einem 24*21 Punkte großen Block. Sprites sind gerade bei der Erstellung von Spielen sehr beliebt, weil sie sehr einfach bewegt werden können. GEOS benutzt Sprites, um beispielsweise die Maus auf dem Bildschirm dazustellen. Da Sprites vom Computer anders erstellt werden als der normale Bildschirm, tauchen sie bei einer Hardcopy nicht auf dem Papier auf.

Status-Zeile

Begriff in GEOPAINT für einen Bereich unten auf dem Bildschirm. Dort werden zu jedem Werkzeug, das Sie in GEOPAINT benutzen können, wichtige Informationen angezeigt. Nach dem Laden von GEOPAINT sehen Sie in der Status-Zeile den Seitenzeiger und die Farbleiste mit den 16 Farben.

Tabulator

Tabulatoren sind in GEOWRITE Symbole, mit denen man den Cursor schnell in eine bestimmte Spalte bewegen kann. Durch Drücken von 'CTRL + i' springt der Cursor nach rechts zur Spalte des nächsten Tabulators.

Track

siehe Sektoren

Untermenü

Anzahl von Befehlen, die inhaltlich zusammengehören. Diese werden sichtbar, wenn der zugehörige Oberbegriff in der Menüzeile angeklickt wird. (Beispiel: Oberbegriff FILE, Untermenü enthält die Befehle: OPEN, DUPLICATE, RENAME, INFO, PRINT, die sich alle auf ein ausgewähltes File beziehen.)

Wordwrap

Begriff aus der Textverarbeitung; Wordwrap sorgt dafür, daß jedes Wort, das nicht mehr in eine Zeile paßt, automatisch in die nächste Zeile gezogen wird.

Stichwortverzeichnis

Accessory	16, 47, 49, 312
Album	18
Anwendungen	159
Applications	16, 59, 312, 318
Arbeitskopien	49
ASCII-Code	96
Änderungen	141, 159
BACKUP	140
BAM	302
BASIC-Tools	69
Benutzeroberfläche	31
Blättern	48
BOOT-Pause	31
Bump	31
Calculator	182
CBM-ASCII	69
Centronics-Interface	193
CLOCK	208
CLOSE	48
COPY	59
Cursor-Blinkfrequenz	172
CUT	59
Datum	42, 128
DESK TOP	16, 34, 312
Direktstart	59
Diskette wechseln	35, 129
Diskettenstrukturen	294
Druck	193
Drucker	33, 39
Druckertreiber	188
DUPLICATE	43

EASY-ACCESS	140
EDMON	259, 268
Eingabegerät	40
Einzelschrittssimulator	198, 276
Epson-Modus	190
File erzeugen	128
File zeigen	128
File-Eintrag	296
File-Verwaltung	295
Filelänge	128
Filemaster	101, 124
Filestruktur	127, 306
Filetyp	127, 306
Floppy	38
Floppy-Justierung	31
Font-Editor	197, 225
Fonts	17
Format VLIR	69
GEODRUCK	80
GEOPAINT	50, 51
GEOPAINT+	66
GEOS	13, 14, 135, 312
GEOS BOOT	15, 22
GEOS KERNAL	15, 312
GEOS V1.0.	19
GEOS V1.2	19, 21, 140
GEOS V1.3	20, 24, 140
GEOS-ASCII-Zeichensatz	69
GEOS-Druckertreiber	185
GEOS-Filetyp	127
GEOS-Format	45
GEOS-Programme	197
GEOS-Variablen	390
GEOWRITE	61, 65, 79, 89, 98, 175
Grafik-Bibliothek	58
H-Copy V1.3	223
Hardcopy-Programm	197

Icon	40, 101, 180
INFO	129
INFO-Fenster	42
INFO-Track	127
Inhaltsverzeichnis	43
Input Devices	17
KERNAL	323
Kopierschutz	21, 24, 135
Laden	31
Maschinensprache-Monitor	259
Modifikator	144
Monitor	197
Muster	52, 53
Musterleiste	52
Namen	41
Notizen	69
Notizendruck	71
PASTE	59
Photo-Album	58
PHOTOSCRAP	58
Pinself	53
Pixeledit	55
Printer	17
Ränder	62, 175
Repeat-Frequenz	173
RESET	44
Schrift	53
Schriftwechsel	63
SCRATCH	36
Seitenzeiger	52
Sektor	127
SEQ	307
Serienbriefe	44

Seriennummer	28
Sicherheitskopie	32, 135, 140
Speicher	390
Speicheraufbau	399
Sprungtabelle	323
SST	277
Starten	31
Startsektor	127
Starttrack	127
Stepper	198, 277, 290
Swap-File	45
System-Files	41
Tab	175
Tabulatoren	62
TEXT-DUMMY	98
Text-Konverter	89
Textformatierung	64
Textomat Plus	89
Textseiten	66
Textverarbeitung	225
Treiber	189
Uhr	177
Uhrzeit	42, 198
UNDO	57, 59
VALIDATE	36, 38, 96
Variablen	390
VLIR	308
Warmstart	169
Werkzeuggestreife	55
WORKFILE	137

DAS STEHT DRIN:

Dieses Buch richtet sich vor allem an die Anwender von GEOS V1.2 und V1.3 deutsch. Deshalb enthält es eine Fülle von Tips und Tricks aus der täglichen Arbeit mit GEOS und eine Reihe von ausgezeichneten Programmen, die Ihre Arbeit mit GEOS entscheidend erleichtern können. Für Programmierer bietet dieses Buch alle wichtigen Informationen zur Erstellung eigener Programme unter GEOS.

Aus dem Inhalt:

- Die Besonderheiten von GEOS – Komponenten, Versionen, Kopierschutz
- Über 50 Tips und Tricks aus der täglichen praktischen Arbeit
- Viele Hilfsprogramme – Notizen drucken, Texte in GEOWRITE konvertieren, Icons für eigene Programme mit dem FILEMASTER
- Bootfähige Sicherheitskopien von GEOS V1.2 und V1.3 deutsch erstellen
- Hardcopy – jederzeit den aktuellen GEOS-Bildschirm im gängigen Format speichern
- Kompletter Font-Editor – bestehende Fonts ändern und neue erstellen
- EDMON – vollständiger Maschinensprache-Monitor unter GEOS als Accessory, dadurch jederzeit ladbar
- Einzelschritt-Simulator – GEOS in einzelnen Schritten abarbeiten, problemlose Fehlersuche in eigenen Programmen
- GEOS-Diskettenstrukturen – Fileformate, Filetypen und Info-Sektor
- GEOS-Sprungtabelle – über 150 fertige Routinen zur Nutzung in eigenen Programmen
- Für GEOS V1.2 und GEOS V1.3 deutsch

UND GESCHRIEBEN HABEN DIESES BUCH:

Manfred Tornsdorf und Rüdiger Kerkloh waren von Anfang an dabei, als GEOS den deutschen Markt eroberte. Ihr erstes Buch zu GEOS erschien praktisch mit der Auslieferung in Deutschland und half vielen GEOS-Besitzern beim Umgang mit diesem neuen Betriebssystem. Der in mehreren Ländern erschienene Titel fand auch in den USA große Beachtung.

ISB N 3-89011-208-0 DM +049.00

DM 49,-
sFr 47,-
ÖS 382,-

**DATA
BECKER**

