

M. O. S.

Mouse Operating System

Bedienungshandbuch

UELZENER STR. 12
2120 LÜNEBURG
FERNRUF (0 41 31) 4 71 22

STAR
DIVISION

Druck:

Nordland Druck GmbH, 2120 Lüneburg

AUTOR von M.O.S. Mouse Operating System :

=====

ANDREAS ZALLMANN (Programmierung & Handbuch)

Copyright 1986 by STAR-DIVISION GmbH
Uelzener Straße 12
D-2120 Lüneburg
04131/47122

Alle Rechte vorbehalten, kein Teil des Handbuches sowie des dazugehörigen Programmes darf in irgendeiner Form (Druck, Fotokopie oder einem sonstigen Verfahren) ohne schriftliche Genehmigung von STAR-DIVISION GmbH reproduziert oder vervielfältigt werden.

Wir werden jede uns bekannt werdende nichtautorisierte Weitergabe dieses Programmes mit allen uns zur Verfügung stehenden Mitteln verfolgen. Der Kunde erhält das Programm inklusive Diskette und Handbuch.

Für die Fehlerlosigkeit des Programmes wird keine Garantie übernommen, sollte ein Fehler entdeckt werden, ist STAR-DIVISION GmbH bestrebt, diesen so schnell wie möglich zu korrigieren.

ACHTUNG !

ATARI ST ist ein eingetragenes Warenzeichen der Fa. ATARI Inc.
COMMODORE, AMIGA sind eingetragene Warenzeichen der Fa. COMMODORE
APPLE, LISA, MACINTOSH sind eingetragene Warenzeichen der Fa. APPLE
GEM ist ein eingetragenes Warenzeichen der Fa. Digital-Research

INHALTSVERZEICHNIS

1.	Motivation für M.O.S.	1
1.1	Die neue Generation von Betriebssystemen	1
1.2	Was ist M.O.S.	2
1.3	An wen wendet sich M.O.S.	2
1.4	Was kann M.O.S.	2
1.5	Warum wurde M.O.S. gerade so gestaltet	2
1.5.1	Warum kein Betriebssystem, sondern eine Befehlserweiterung	2
1.5.2	Warum MODE 2	3
1.5.3	Warum keine Maus, sondern ein Joystick	3
1.6	Vergleich M.O.S. - andere Systeme	4
2.	Grundlagen von M.O.S.	5
2.1	Speicheraufteilung (Rameinplanung)	7
2.1.1	Textbildschirm	8
2.1.2	Matrix	8
2.1.3	Maussymbol	8
2.1.4	Zeichensatz	9
2.1.5	Iconspeicher	9
2.1.6	M.O.S.-MC-Programm	9
2.1.7	Systemvariablen	9
2.1.8	Buffer	9
2.2	Was befindet sich auf der mitgelieferten Diskette .	9
2.2.1	Das M.O.S.-System	9
2.2.2	Der Icondesigner	10
2.2.3	Der Zeichendesigner	10
2.2.4	Der Mousedesigner	10
2.2.5	Das Grafikprogramm	10
2.3	Einladen und Installieren vom M.O.S.	10
2.4	M.O.S. Programme in Verbindung mit MC und Basic ...	11
3.	Die neuen Basicbefehle	13
3.1	Grundlegende Syntax der Befehle	13
3.2	Unterschiede auf den verschiedenen CPC-Rechnern ...	13
3.3	Aufbau der Befehlsübersicht	13
3.4	Die Windowbefehle	14
3.5	Die Menübefehle	18
3.6	Die Iconbefehle	19
3.7	Die Mausbefehle	21
3.8	Sonstige Befehle	24
4.	Einsprünge der Befehle von MC	26
4.1	Aufbau der Übersicht	26
4.2	Einsprünge in die Windowbefehle	26
4.3	Einsprünge in die Menübefehle	28
4.4	Einsprünge in die Iconbefehle	28
4.5	Einsprünge in die Mausbefehle	29
4.6	Einsprünge in die sonstigen Befehle	31
5.	M.O.S. Routinen	33
6.	Systemvariablen	37

7.	Die Anwenderprogramme	39
7.1	Wozu die mitgelieferten Anwenderprogramme	39
7.2	Grundlagen der Anwenderprogramme	39
7.2.1	Laden der Anwenderprogramme	39
7.2.2	Identisches bei allen Anwenderprogrammen ..	39
7.3	Icondesigner	40
7.3.1	Der Bildschirmaufbau	40
7.3.2	Das Verändern des aktiven Icons	40
7.3.3	Die Funktion der Icons	41
7.3.4	Die Menüs	41
7.3.3.1	Das Bearbeitungsmenü	41
7.3.3.2	Das Ablagemenü	43
7.3.3.3	Das Diskettenmenü	43
7.3.3.4	Das Modusmenü	44
7.3.3.5	Das Utilitymenü	44
7.3.3.6	Das Abbruchmenü	45
7.3.5	Späteres benutzen der Icons	45
7.4	Zeichendesigner	46
7.4.1	Der Bildschirmaufbau	46
7.4.2	Das Verändern des aktiven Zeichens	46
7.4.3	Die Funktion der Icons	47
7.4.4	Die Menüs	47
7.4.3.1	Das Bearbeitungsmenü	47
7.4.3.2	Das Ablagemenü	47
7.4.3.3	Das Diskettenmenü	47
7.4.3.4	Das Modusmenü	47
7.4.3.5	Das Utilitymenü	48
7.4.3.6	Das Abbruchmenü	48
7.4.5	Späteres benutzen des Zeichensatzes	48
7.5	Mausdesigner	48
7.5.1	Der Bildschirmaufbau	48
7.5.2	Das Verändern des Mauzeigers	49
7.5.3	Die Funktion der Icons	49
7.5.4	Die Menüs	49
7.5.3.1	Das Bearbeitungsmenü	49
7.5.3.2	Das Diskettenmenü	49
7.5.3.3	Das Modusmenü	49
7.5.3.4	Das Abbruchmenü	49
7.5.5	Späteres benutzen der Icons	50
8.	Das Grafikprogramm	51
8.1	Laden des Grafikprogrammes	51
8.2	Funktionen auf dem Grafikbildschirm	51
8.3	Das Hauptmenü	51
8.3.1	Die Betriebsarten	51
8.3.2	Die Funktionen	54
9.	Ein Programmbeispiel unter M.O.S.	57

Anhang

1.	Begriffserklärungen	60
1.1	Benutzerfreundliche Oberfläche/ Desktop	60
1.2	Fenster	60
1.3	Menüs	61
1.4	Icons	61
1.5	Maus	61
1.6	Mauszeiger, Maussymbol	61
1.7	Anklicken	61
1.7.1	Tastendruck	62
1.7.2	Klick	62
1.7.3	Doppelklick	62
1.7.4	Press	62
1.7.5	Cancel	62
1.8	Kopfzeile/Headline	63
1.9	Closepoint	63
2.	Programmierhinweise	64
2.1	Programmierung einer Menüleiste	64
2.2	Anklicken von Icons	64

M.O.S. - Ein fantastisches System

Bedienungshandbuch

1. Motivation für M.O.S.

In dem ersten Kapitel wollen wir erläutern, was M.O.S. ist, was es kann, an wen es sich richtet, etc.

Außerdem wollen wir offenlegen, wieso das Programm gerade in dieser Form ausgeliefert wird und nicht in einer anderen mit Veränderungen an der einen oder anderen Stelle.

Ferner wird ein Überblick gegeben, wie sich die Landschaft der benutzerfreundlichen Oberflächen inzwischen entwickelt hat. Dabei wird auch auf ähnliche Produkte für den CPC aber auch für andere Rechner eingegangen.

Mir sei gleich eine Bemerkung gestattet. In der Anleitung werden manchmal bestimmte Ausdrücke, wie z.B. Icons oder Desktop, benutzt. Sollten Ihnen diese Ausdrücke unbekannt sein, oder sollten Sie sich über die Bedeutung der Ausdrücke nicht genau im klaren sein, so empfehle ich Ihnen die Lektüre des Anhangs 1, in dem diese Ausdrücke erklärt werden.

1.1 Die neue Generation von Betriebssystemen

Gerade in letzter Zeit entwickeln sich die Computer, die mit einer Oberfläche ausgeliefert werden zu einem wahren Verkaufsschlager, so der Atari ST und der Commodore Amiga.

Lassen Sie mich zuerst die Entwicklung zu diesen benutzerfreundlichen Oberflächen darlegen. Alles begann mit einer Programmiersprache mit dem sinnigen Namen Smalltalk, die von einer Tochterfirma des Xerox-Konzerns entwickelt wurde. Mit dieser Sprache war die Programmierung der sog. Icons, kleiner grafischer Symbole, sehr gut möglich. Smalltalk wurde dann zu Star weiterentwickelt. Star benutzte auch schon die Maus als Eingabegerät. Allerdings wurde Star nicht sehr weit verbreitet und gelangte bald in Vergessenheit.

Apple griff die Idee wieder auf. 1982 kam Apple mit Lisa auf dem Markt. Dieser Computer wurde mit Maus und einer Oberfläche ausgeliefert. Die Idee erregte zwar Aufsehen, dennoch wurde Lisa nicht gut verkauft, vermutlich, weil das System einfach zu ungewöhnlich benutzerfreundlich war.

Erst auf dem Macintosh von Apple, gelangte das System endlich eine weitere Verbreitung. Allerdings hatte es mit Star nicht mehr viel zu tun. Der Macintosh war leider sehr teuer, sodaß sich nur Betuchte ein solches Gerät leisten konnten.

Kürzlich kamen dann Atari und Commodore mit dem ST und dem Amiga auf dem Markt. Geräte, die erheblich billiger sind, als der Macintosh.

Das neueste Gerät mit benutzerfreundlicher Oberfläche ist der PC von Schneider.

1.2 Was ist M.O.S

Nun möchten allerdings viele Leute, die einen Schneider besitzen auch in den Genuß dieser benutzerfreundlichen Oberfläche, bzw. in den Genuß benutzerfreundlichen Programme zu kommen.

Wir haben uns gedacht, es müsste doch möglich sein, dies auch auf dem CPC zu verwirklichen.

Es handelt sich allerdings um kein Betriebssystem, sondern um eine Befehlserweiterung, deren Befehle per Basic oder per Maschinensprache aufgerufen werden können.

Statt einer Maus wird dabei allerdings ein Joystick benutzt.

M.O.S. läuft ausschließlich auf Mode 2.

M.O.S wird mit einem Grafikprogrammen und drei Anwendungsprogrammen zum Erstellen von Mauszeiger, Icons und Zeichensatz geliefert.

1.3 An wen wendet sich M.O.S.

M.O.S. wendet sich in erster Linie an den Basic-, oder Maschinenspracheprogrammierer, der anwenderfreundliche Programme erstellen will. Mit M.O.S. wird ihm eine überaus leistungsfähige Befehlserweiterung zur Verfügung gestellt, mit der die Erstellung anwenderfreundlicher Programme unter einer GEM-ähnlichen Oberfläche zum Kinderspiel wird.

1.4 Was kann M.O.S.

Unter M.O.S. kann man bis zu zwanzig unabhängige Textfenster öffnen. In diese kann man schreiben, Balken setzen, etc. Dabei kann in jedes Fenster, egal, ob es ganz oben liegt oder nicht, hineingeschrieben wird. Auch ist es jederzeit möglich ein beliebiges Fenster zu schließen. Der Bildschirm wird dann automatisch aktualisiert.

Ferner können bis zu 5 Menüs eröffnet werden. (siehe auch Anhang 1.3).

Dann können noch bis zu 20 Icons auf den Bildschirm gebracht werden, wobei die Icons entweder dem normalen Bildschirm oder einem der Fenster/Menüs zugeteilt werden können.

Außerdem ist im Programm eine Mausabfrage implementiert. Damit ist es möglich verschiedene Fenster oder Menüpunkte anzuklicken.

1.5 Warum wurde M.O.S. gerade so gestaltet?

An dieser Stelle soll erläutert werden, warum M.O.S. gerade so, wie es nun vorliegt, programmiert wurde und nicht anders.

1.5.1 Warum kein Betriebssystem, sondern eine Befehlserweiterung

So schön ein Desktop auch sein kann, so leicht man sich auch eingewöhnen kann, nach einiger Zeit jedoch fährt man mit einem Betriebssystem ohne Desktop besser. Da man das Betriebssystem im Allgemeinen auch oft benutzt, dürfte die Eingewöhnung auch ohne benutzerfreundliche Oberfläche gut vonstatten gehen.

Außerdem ist es beim Schneider auch problematisch das ganze Betriebssystem in eine Oberfläche einzubinden. Dazu fehlen einige wichtige Patches. Es ist also garnicht möglich ein völlig neues Betriebssystem auf dem CPC zu entwerfen, daß eine Oberfläche unterstützt.

Ferner ist der Speicherplatz beschränkt. Während die 16-Bit Computer sehr viel Speicherplatz haben, hat der Schneider nur ca. 40 Kilobyte an Speicherplatz zur Verfügung. Es müsste aber ein neues Betriebssystem mit Oberfläche in diesem Speicher abgelegt werden. Die Folge wäre, daß kaum noch Speicherplatz für andere Programme bliebe.

Aus diesen Gründen haben wir uns entschlossen, kein Betriebssystem, sondern eine Befehlserweiterung anzubieten. Sie haben weiterhin Ihren guten alten CPC vor sich, nur um viele, neue, extrem leistungsfähige Befehle erweitert.

Wird M.O.S. dann angeschaltet, werden einige Vektoren umgebogen und damit wird erreicht, daß zum Beispiel auch beim Listen von Programmen die Fenster an gleicher Position auf dem Bildschirm stehen bleiben, der Bildschirm wird also unter den geöffneten Fenstern gescrollt.

Eingetippter Text wird unter Fenstern natürlich ebenfalls nicht angezeigt. Sobald das Fenster geschlossen wird ist dieser aber wieder sichtbar.

1.5.2 Warum MODE 2

Wie Sie oben bereits gelesen haben, arbeitet M.O.S. nur in Mode 2. Sobald M.O.S. angeschaltet wird, so wird in Mode 2 geschaltet und der Mode-Befehl ist ab dann gesperrt.

Sollte M.O.S. auf allen Bildschirmmodi laufen, so wäre das Programm noch länger geworden und der frei Speicherplatz geringer. Dies wollten wir nicht und mußten uns folglich für einen Bildschirmmodus entscheiden.

Da M.O.S. vor allem bei Anwenderprogrammen zum Einsatz kommen dürfte und diese nun einmal im Allgemeinen im Modus 2 ablaufen, viel die Entscheidung leicht.

1.5.3 Warum keine Maus, sondern ein Joystick

Die Entscheidung das Programm mit einem Joystick und nicht mit einer Maus laufen zu lassen, war eigentlich nur logisch. Sehr viele Personen besitzen bereits einen Joystick und falls nicht, so ist ein Joystick nicht teuer und läßt sich zudem noch mit anderen Programmen nutzen.

Eine Maus ist im Vergleich zu einem Joystick jedoch erheblich

teurer. Da es sehr wenige Programme gibt, die eine Maus unterstützen, würden Sie die Maus praktisch nur für dieses Programm kaufen. Aber Sie sollen dieses Produkt ja möglichst billig nutzen.

Hinzu kommt, daß man zur sinnvollen Benutzung einer Maus einen großen Schreibtisch benötigt, was auch zum Teil auf Schwierigkeiten stoßen könnte.

Wer aber trotzdem mit der Maus arbeiten möchte, so ist dagegen nichts einzuwenden. Im Handel werden schon viele Mäuse angeboten, die die gleichen Signale wie ein Joystick liefern. Mit einer solchen Maus können Sie ohne Probleme mit M.O.S. arbeiten.

1.6 Vergleich M.O.S. - andere Systeme

Beginnen wir zuerst mit den größeren Computern. Es ist klar, daß M.O.S. von der Geschwindigkeit langsamer ist, als benutzerfreundliche Oberflächen auf 16-Bit-Computern. Immerhin ist das Herz des CPC's ja "nur" ein 8-Bit-Prozessor. Dennoch ist eine ausreichende Geschwindigkeit erreicht worden, besonders, wenn man im Grafikfenster arbeitet (dazu später).

Während bei den größeren Rechnern das Betriebssystem im Vordergrund steht, steht bei M.O.S. die einfache Programmierung im Vordergrund. Beim Amiga, Atari und dem Schneider PC ist die Programmierung von Fenstern, etc. bei weitem nicht so einfach, wie unter M.O.S.

Außerdem bietet M.O.S. zum Teil noch mehr Möglichkeiten. Beim Atari ist es z.B. nur möglich vier Fenster zu öffnen, unter M.O.S. können jedoch bis zu 20 Fenster geöffnet werden. Das außerdem in jedes Fenster geschrieben werden kann, auch wenn dieses nicht oben liegt, ist ebenfalls eine sehr leistungsfähige und schöne Möglichkeit.

Ein Produkt, was für den Schneider sehr bekannt ist, ist das AMX mouse System. Dieses wird mit einem sehr leistungsfähigem Grafikprogramm ausgeliefert. Die Befehlsweiterung ist jedoch bei weitem nicht so leistungsfähig, wie die von M.O.S. Besonders die Fenstertechnik ist nicht so ausgefeilt, wie unter M.O.S. Auch wird dort immer der Grafikbildschirm unter einem eröffneten Fenster abgespeichert, was gerade beim beschränkten Speicherplatz des CPCs eine problematische Sache ist. Bei M.O.S. werden im Allgemeinen aber nur die ASCII-Codes abgespeichert, was den benötigten Speicherplatz auf ein Achtel reduziert. Menüs sind bei dem AMX-Paket überhaupt nicht zu programmieren.

Das das AMX-Paket nur mit dürftiger englischer Anleitung ausgeliefert wird (10 Seiten für die Befehlsweiterung) ist ein weiterer Pluspunkt für M.O.S.

Wir sind also der Meinung, M.O.S. stellt nicht nur eine echte Alternative zu anderen Systemen dar, sondern ist etwas gänzlich neues.

2. Grundlagen von M.O.S.

In diesem Abschnitt der Anleitung sollen die Grundlagen, auf denen das gesamte M.O.S.-System basiert, vorgestellt werden. So kann sich der Benutzer auch ein Bild davon machen, wie das M.O.S.-Programm aufgebaut ist und was nun genau passiert, wenn er beispielsweise ein Fenster öffnet.

Zuerst einmal ist noch anzumerken, daß alle unter M.O.S. geschriebenen Programme kompatibel sind. Alle Befehle sind auf allen drei Rechnern vorhanden und die Einsprünge der Befehle und Routinen von M.O.S. liegen an den gleichen Stellen. Es muß nur die richtige M.O.S.-Version, die auch zu dem Rechner, auf dem das Programm ablaufen soll, paßt, geladen werden.

Beim Aufrufen der Befehle von Basic gibt es einen syntaktischen Unterschied (siehe 3.2). Er betrifft die Übergabe von Stringparametern. 464 Besitzer brauchen sich keine Sorgen zu machen, Ihre Befehle (per Stringdescriptor) funktionieren auch auf dem 664 und 6128. 664/6128 Besitzer dürfen jedoch Strings nicht direkt, sondern nur per Stringdescriptor übergeben, wenn die Programme auch auf einem 464 lauffähig seien sollen.

Wie Sie vielleicht wissen, besitzt der CPC einen 16k grossen Bildschirmspeicher. Dabei werden in MODE 2 zur Abspeicherung einer Bildschirmposition (1 Zeichen) acht Bytes benötigt. Will man also den von einem Fenster überlappten Bildschirmbereich zwischenspeichern, um ihn nachher wieder zu rekonstruieren, so müssten pro Bildschirmposition 8 Bytes gespeichert werden. Dies erschien uns, gerade wegen des begrenzten Speicherplatzes des CPCs, als keine befriedigende Lösung.

Deshalb haben wir uns entschlossen, jeweils nur die ASCII-Zeichen abzuspeichern. Zu diesem Zweck wurde ein 2000 Bytes grosser Textspeicher eingerichtet. In ihm ist vermerkt, auf welcher Stelle des Bildschirms welches ASCII-Zeichen steht. Grafik wird dann logischerweise nicht mit abgespeichert.

Werden nun Buchstaben auf den Bildschirm gebracht, so werden sie zunächst an entsprechender Position im Textspeicher eingetragen. Ist die Bildschirmposition nicht durch Fenster, oder ähnliches überlappt, so muß der Buchstabe jetzt auch noch auf dem Bildschirm ausgegeben werden.

Wie stellt der Rechner aber nun fest, an welcher Stelle der Bildschirm durch Fenster, Icons, etc. überlappt ist? Dies geschieht durch eine 250 Bytes grosse Matrix in der pro Bildschirmposition ein Bit angibt, ob diese Position überlappt ist oder nicht. In dieser Matrix "schaut" der CPC nach, ob die Position überdeckt ist und wenn das nicht der Fall ist, wird das entsprechende Zeichen auf den Bildschirm ausgegeben.

Was passiert nun, wenn ein Fenster eröffnet wird? Zunächst wird die Matrix in dem Bereich, den das Fenster überlappt auf eins gesetzt, als Zeichen, daß jetzt der Textbildschirm in dem Bereich nicht mehr sichtbar ist.

Das Aussehen des Fenster wird jetzt nicht etwa in den Textspeicher eingetragen, sondern in den Puffer. Dies ist auch nicht anders möglich, da ja auch beim Schreiben unter Fenstern der Textbildschirm aktualisiert wird (s. oben).

Das Fenster wird also im Puffer abgespeichert. Dabei werden ebenfalls nur die ASCII-Codes der Buchstaben, aus denen sich das Fenster zusammensetzt, gespeichert. In speziellen Fenstersystemvariablen wird natürlich festgehalten, wo das Fenster im Puffer abgelegt wurde. Damit der Rahmen und ähnliches nicht verlorengeht, wird all dies aus den Zeichen 0 bis 15 zusammengesetzt (zum Zeichensatz gleich mehr).

Anschließend wird das Fenster auf den Bildschirm gebracht, was kein größeres Problem ist, da das zuletzt geöffnete immer ganz oben liegt.

Wird in ein Fenster geschrieben, oder wird es gelöscht, so werden die Zeichen im Puffer an der richtigen Stelle verändert. Anschließend wird überprüft, ob an den entsprechenden Stellen das Fenster, in das gedruckt wurde, ganz oben liegt. Ist das der Fall, so müssen die Zeichen ausgegeben werden, da sie sichtbar sind. Wird das Fenster an bestimmten Positionen von anderen Fenstern höherer Priorität überdeckt, wird das Zeichen natürlich nicht ausgegeben.

Wird ein Fenster geschlossen, bekommt der Rechner Arbeit. Zunächst wird der Bereich im Puffer, der das Aussehen des Fensters bestimmte, herausgestrichen. Dies geschieht, indem alle folgenden Fensterbereiche hochgeschoben werden. Jetzt müssen natürlich die Zeiger in den Systemvariablen dieser Fenster auf die Pufferbereiche korrigiert werden.

Dann wird jede vom zu schliessenden Fenster überdeckte Bildschirmposition weiter untersucht. Und zwar wird zunächst festgestellt, ob an dieser Position der Bildschirm durch ein anderes Fenster, etc. überdeckt wird. Ist das nicht der Fall, dann ist der Textbildschirm sichtbar. Dann wird an das entsprechende Zeichen des Textspeichers an der Position ausgegeben und das entsprechende Matrixbit gelöscht, da der Textbildschirm jetzt nicht mehr überlappt wird.

Wird die Bildschirmposition doch noch von einem Fenster überdeckt, so bleibt das Matrixbit natürlich eins. Außerdem wird das Fenster ermittelt, welches an dieser Stelle die höchste Priorität hat. Das Zeichen, welches an dieser Bildschirmposition im Fenster steht, wird dann ausgegeben.

Die Verwaltung der Menüs verläuft analog.

Kommen wir zur Iconverwaltung. Es dürfen bis zu 20 Icons eröffnet werden. Am Anfang ist dem Icon 1 das Grafiksymbol 1 zugeordnet, dem Icon 2 das Grafiksymbol 2, etc. Dies können Sie aber durch den Befehl !SETGRAFIK beeinflussen.

Wird nun ein Icon auf den Textbildschirm gebracht, so läuft alles analog zu den Fenstern, bis auf eine Ausnahme: Die Icons müssen nicht im Puffer abgelegt werden, da ihr Aussehen bereits im Iconspeicher abgelegt ist.

Eine andere Möglichkeit ist noch, daß die Icons Fenstern zugeordnet werden. Dabei haben sie die gleiche Priorität wie das Fenster, dem sie zugeordnet sind. Allerdings überlappen die Icons das Fenster noch.

Beim Schließen der Fenster muß natürlich darauf geachtet werden, daß darunterliegende Icons ebfalls richtig aktualisiert werden.

Damit nicht vollständig auf Grafik verzichtet werden muß, haben wir uns entschlossen noch ein Grafikfenster zuzulassen, welches man mit dem Befehl !GRAFIKW definieren kann. Dabei kann man einen Teilbereich des Bildschirms definieren, indem nicht nur die ASCII-Zeichen, sondern auch Grafik zwischengespeichert wird.

Das Grafikfenster hat eine höhere Priorität als der Textspeicher und die Icons, die keinem Fenster zugeordnet sind. D.h. unter dem Grafikfenster sind Icons und andere Zeichen nicht mehr sichtbar.

Um ASCII-Zeichen ins Grafikfenster zu schreiben, kann man folgendermaßen vorgehen: Zuerst im Bereich vom Grafikfenster drucken, ohne das Grafikfenster anzuschalten. Dann das Grafikfenster einschalten.

Wird ein Fenster geöffnet, welches teilweise das Grafikfenster überlappt, so wird das Fenster im Puffer zuerst ganz normal abgespeichert. Hinter dem Aussehen des Fensters im ASCII-Code wird die Grafik abgespeichert. Allerdings nur der Bereich, der wirklich vom Fenster überlappt wird. Dafür werden pro Bildschirmposition 8 Bytes benötigt. Damit ergibt sich ein Speicherbedarf von 9 Bytes pro Bildschirmposition eines Fensters im Grafikfenster.

Da das Grafikfenster das Schließen der Fenster noch weiter kompliziert, gibt es eine Einschränkung von Fenstern im Bezug auf das Grafikfenster: Fenster, bzw. Menüs dürfen sich im Allgemeinen im Bereich des Grafikfensters nicht überlappen.

Dies kann allerdings mit dem Befehl !OVER,1 erlaubt werden. Dann muß aber das zuletzt geöffnete Fenster zuerst wieder geschlossen werden, sonst wird der darunterliegende Bildschirminhalt nicht richtig rekonstruiert.

Damit wären die Grundlagen der Fenstertechnik erklärt. Wenn Sie nicht alles verstanden haben, trösten Sie sich, so wichtig ist dies nicht zum erfolgreichen Benutzen des M.O.S.-Programmes.

Kommen wir zum Zeichensatz: Bei M.O.S. existieren nur 128 Zeichen. Die Charaktere von 128 bis 255 entsprechen denen von 0 bis 127, nur invertiert. M.O.S. benutzt einen vollständig neuen Zeichensatz. Natürlich sind dort nur 128 Zeichen gespeichert, die Inversen berechnet M.O.S. selbst.

Die Zeichen 32 bis 127 entsprechen dabei den normalen ASCII-Charakteren. Die Zeichen 0-15 sind Sonderzeichen für Fenster-rahmen, etc. Es gibt aber auch noch einen zweiten Satz Sonderzeichen, die von 16 bis 31 abgelegt sind. Durch den Befehl !SWAPSET kann zwischen den beiden Rahmenzeichensätzen umgeschaltet werden.

Wollen Sie die Zeichen verändern oder nur ansehen, so benutzen Sie dazu den Zeichendesigner (7.4).

2.1 Rameinplanung

M.O.S. belegt den Speicher oberhalb der Adresse 29399. Beim

Aufruf von M.O.S. wird gleich ein 2400 Bytes großer Puffer initialisiert, der bei Adresse 27000 beginnt. Unter Beibehaltung des Puffers können also die Adressen unterhalb von 27000 von Basic und MC genutzt werden.

Es folgt zunächst eine Übersicht aller Programmteile von M.O.S. und welche Speicherbereiche sie belegen. Anschließend werden die einzelnen Programmteile näher vorgestellt.

Start Ende Länge Kommentar

40620	42619	2000	Textbildschirm
40370	40619	250	Matrix (Bit gesetzt, Window, Menü über Textbildsch.
40242	40569	128	Maussymbol
39218	40241	1024	Zeichensatz
37938	39217	1280	Iconaussehen
30000	37937	7938	M.O.S. Maschinenprogramm
29400	29999	600	Systemvariablen
27000	29399	2400	Puffer (Voreinstellung)

2.1.1 Textbildschirm

Im Textbildschirm ist das Aussehen des Bildschirms in ASCII-Zeichen gespeichert. Zuerst die erste Reihe von links nach rechts, dann die zweite, etc.

Es ist allerdings nicht das Aussehen des Bildschirms, so wie er im Augenblick aussieht gespeichert, sondern lediglich das Aussehen des Textbildschirms. Darübergeklappte Fenster, oder ähnliches werden im Puffer gespeichert.

2.1.2 Matrix

In dieser Matrix ist gespeichert, an welchen Positionen der Textbildschirm von Fenstern, Menüs oder Icons überdeckt wird. Dabei wird pro Position ein Bit benötigt. Ist es zurückgesetzt, so wird der Textbildschirm nicht überdeckt, ist es gesetzt, so wird er überdeckt. Dabei ist natürlich nicht spezifiziert, durch was.

Jeweils 8 Positionen werden in einem Byte zusammengefasst und zwar die ganz linke Position in Bit 7, die ganz rechte in Bit 0.

Zuerst wird die erste Textzeile von links nach rechts in 80 durch 8 gleich 10 Bytes abgelegt, dann die zweite, etc.

2.1.3 Maussymbol

Die Maus ist eine Printposition groß. Es werden acht verschiedene Stellungen der Maus abgespeichert, jedesmal einen Plotpunkt weiter rechts. Daraus ergibt sich, daß obwohl das Maussymbol nur eine Printposition groß ist, zwei nebeneinanderliegende Printpositionen gespeichert werden müssen. Zuerst wird das oberste Byte der linken Printposition gespeichert, dann das oberste Byte der rechten Printposition. Anschließend werden die zweiten, dann die dritten Bytes, usw. abgespeichert. Dies geschieht dann acht Mal hintereinander.

2.1.4 Zeichensatz

Im Zeichensatz sind 128 Zeichen a 8 Byte abgespeichert. Die Charaktere über 128 sind nicht abgespeichert, da diese ohnehin nur die inversen Abbilder der Characteres unter 128 sind.

2.1.5 Iconaussehen

Hier ist das Aussehen der 20 Icons abgelegt. Pro Icon sind 64 Bytes vorgesehen. Zuerst wird die Printpositon in der oberen, linken Ecke des Icons abgelegt (8 Byte), dann die 3 Printpositionen rechts daneben, von links nach rechts und dann die darunterliegenden vier Printpositionen.

2.1.6 M.O.S.-Programm

Zum M.O.S.-Programm ist nicht viel zu sagen, mit CALL 30000 wird M.O.S. initialisiert.

2.1.7 Systemvariablen

Ein so umfangreiches Programmpaket wie M.O.S. benötigt sehr viele Systemvariablen, hier 600 Bytes. Die wichtigsten Systemvariablen werden unter 6. vorgestellt.

2.1.8 Puffer

Der Puffer ist auf 2400 Bytes Umfang voreingestellt, kann jedoch geändert werden. In ihm werden die geöffneten Fenster und Menüs gespeichert, nicht der von ihnen überdeckte Text, denn der ist im Textspeicher abgelegt. Weitere Informationen unter 3.8 beim Befehl !BUFFER.

2.2 Was befindet sich auf der Diskette

Unter diesem Punkt wollen wir Ihnen kurz vorstellen, was sich alles für Programme auf der mitgelieferten Diskette befinden. Diese haben wir in 6 Kategorien eingeteilt:

2.2.1 Das M.O.S.-System

- MOS.BAS - Basicprogramm zum Laden von M.O.S. und zum Verzweigen in die Unterprogramme. Fragt Rechnertyp ab und lädt danach die entsprechende M.O.S.-Version und installiert sie. Danach wird das Programm MOS.MEN geladen (siehe auch 2.3).
- MOS.DEM - Menüprogramm, welches in verschiedene Ebenen verzweigen kann, oder die Anwenderprogramme nachlädt (siehe auch 2.3 und 7).
- MOS464.BIN - 464 M.O.S.-Maschinenprogramm
- MOS664.BIN - 664 M.O.S.-Maschinenprogramm

MOS6128.BIN - 6128 M.O.S.-Maschinenprogramm

2.2.2 Der Icondesigner

Näheres zum Icondesigner siehe unter 7.3

ICON.DES - Hauptbasicprogramm zum Icondesigner
ICONDES1.MC - Maschinenprogramm 1 zum Icondesigner
ICONDES2.MC - Maschinenprogramm 2 zum Icondesigner
ICONDES.ICN - Beispiel File (siehe 7.3.4.3)

2.2.3 Der Zeichendesigner

Näheres zum Zeichendesigner siehe unter 7.4

ZEICHEN.DES - Hauptbasicprogramm zum Zeichendesigner
ZEICHEN.MC - Maschinenprogramm zum Zeichendesigner
ZEICHEN.CHR - Beispiel File (siehe 7.4.4.3)

2.2.4 Der Mousedesigner

Näheres zum Mousedesigner siehe unter 7.5

MOUSE.DES - Hauptbasicprogramm zum Mousedesigner
MOUSEDES.MC - Maschinenprogramm zum Mousedesigner
MAUS.MOU - Beispiel File (siehe 7.5.4.2)

2.2.5 Das Grafikprogramm

Näheres zum Grafikprogramm siehe unter 8.

PAINT.BAS - Vorprogramm zum Grafikprogramm. Es lädt es das
Hauptprogramm PAINT.PGM
PAINT.PGM - Hauptbasicprogramm zum Grafikprogramm
PAINT1.MC - Maschinenprogramm 1 zum Grafikprogramm
PAINT2.MC - Maschinenprogramm 2 zum Grafikprogramm
TEILBILD.TLB - Beispiel für ein Teilbild
TESTBILD.GSB - Beispiel für ein Gesamtbild

2.3 Einladen und Installieren von M.O.S.

Um M.O.S. einzuladen, müssen Sie zunächst M.O.S.-Programm und den Puffer mit dem MEMORY-Befehl vor Überschreiben durch Variablen,

oder ähnliches schützen. Da der Puffer bei Adresse 27000 beginnt, benötigen Sie einen MEMORY 26999 Befehl. Setzen Sie nachher durch den Befehl !BUFFER (s. 3.8) den Pufferstart auf eine andere Adresse, so müssen Sie ihn vorher natürlich ebenfalls wieder mit MEMORY sichern, falls Sie ihn nach unten verschieben. Falls der Puffer nach oben verschoben wird, so ist erst nach der Ausführung des !BUFFER-Befehls ein neuer MEMORY vorzunehmen. Natürlich ist es beim letzteren Fall nicht unbedingt erforderlich MEMORY anzugleichen, es ist jedoch empfehlenswert, da dann unter BASIC mehr Speicherplatz zur Verfügung steht.

Nach dem Setzen von MEMORY muß das Maschinenprogramm eingeladen werden. Welches der drei Versionen Sie von Diskette laden, kommt auf Ihren Rechner an:

```
LOAD "MOS464"      - Für den CPC 464
LOAD "MOS664"      - Für den CPC 664
LOAD "MOS6128"     - Für den CPC 6128
```

Anschließend brauchen Sie M.O.S. nur noch zu initialisieren, dies geschieht durch einen CALL 30000. Nach der Ausgabe einer Copyrichtmeldung können Sie beginnen.

Zusammenfassend nochmal den Lade- und Installierungsvorgang für M.O.S.:

```
1. MEMORY 26999
2. LOAD "MOSxxxx"   (xxxx CPC Version)
3. CALL 30000
```

Es gibt aber auch noch eine andere Möglichkeit M.O.S zu installieren: Dazu tippen Sie einfach RUN "MOS".

Es wird dann zunächst ein kleines Basicprogramm geladen, welches Sie fragt, welchen Rechner Sie besitzen. Nach erfolgter Antwort wird die richtige Version von Diskette geladen und installiert.

Dann wird ein anderes Basicprogramm nachgeladen, welches ein Menü aufbaut. Mit Joystick können Sie jetzt zwischen den drei Anwenderprogrammen (diese interessieren uns jetzt noch nicht, weiteres siehe Punkt 7) wählen, einem M.O.S.-Info (ausprobieren!) und den Menüpunkten M.O.S.-Ebene, BASIC-Ebene und RESET.

RESET setzt das System vollständig zurück, sodaß sich auch M.O.S. danach nicht mehr im Speicher befindet.

Durch den Menüpunkt M.O.S.-Ebene wird die Kontrolle an den Editor zurückgegeben, dabei bleibt M.O.S. eingeschaltet. Sie befinden sich jetzt also in dem installierten M.O.S.-System.

Durch den Menüpunkt BASIC-Ebene wird im Prinzip das gleiche erreicht, nur das M.O.S. durch den Befehl !MOSOFF ausgeschaltet wurde. M.O.S. ist aber trotzdem installiert, sodaß Sie jederzeit mit !MOSON das System wieder einschalten können.

Das Menüprogramm befindet sich nach der Auswahl von M.O.S.-Ebene oder BASIC-Ebene noch im Speicher, Sie können sich also mit LIST ein unter M.O.S. geschriebenes Programm ansehen.

2.4 M.O.S. Programme in Verbindung mit MC und Basic

Wie bereits oben erwähnt stehen Ihnen bei Beibehaltung des Puffers ca. 26k für Basicprogramme unter M.O.S. zur Verfügung. Wollen Sie auch in MC programmieren, so können Sie die MC-Programme vor den Puffer legen oder den Puffer heruntersetzen und das MC Programm zwischen Pufferende und Programmbeginn M.O.S. legen. Dies bleibt Ihnen überlassen. Vergessen Sie aber nicht Ihr Maschinenprogramm, bzw. den heruntergesetzten Puffer durch den MEMORY-Befehl zu schützen.

Schreiben Sie ein Basic- oder Maschinenprogramm, welches die Befehle unter M.O.S. oder die M.O.S. Routinen benutzt, so müssen Sie natürlich dafür sorgen, daß sich M.O.S. beim Ablauf Ihres Programm auch im Rechner befindet. Es müsste also gegebenenfalls von Diskette geladen und installiert werden (s.2.3). Zum Gebrauch der M.O.S.-Routinen muß M.O.S. nicht installiert, werden.

3. Die neuen Basicbefehle

Unter M.O.S. werden 36 neue Basicbefehle eingerichtet. Ferner wurden 24 neue Fehlermeldungen geschaffen, damit der Benutzer eventuelle Fehler optimal eingrenzen und beseitigen kann.

3.1 Die grundlegende Syntax der Befehle

Bei allen Befehlen handelt es sich um RSX-Befehle, d.h. es muß ihnen ein senkrechter Strich (zu erreichen über Shift+Klammeraffe) vorangestellt werden.

Folgt dem Befehl ein Parameter, so muß auch direkt hinter dem Befehl ein Komma stehen. Alle Parameter werden wie üblich ebenfalls durch Kommata getrennt.

Beispiel: In Basic braucht dem Befehl kein Komma zu folgen:

```
BORDER 10
```

Bei RSX-Befehlen jedoch muß, falls ein Parameter folgt, zwischen diesem und dem Befehl ebenfalls ein Komma stehen:

```
!WLOCATE,1,5,10
```

RSX-Befehlen können auch Strings übergeben werden:

Beispiel: !WPRINT,1,"Hallo"

3.2 Unterschiede auf den verschiedenen CPC-Rechnern

Der einzige Unterschied zwischen den verschiedenen CPC-Rechnern ist die Übergabe von Strings. Beim CPC 464 dürfen in RSX-Befehlen keine Strings direkt verwandt werden. Der String muß erst in eine Variable gespeichert werden, deren Stringdescriptor (Klammeraffe+Variablenname) dann übergeben werden muß.

Um obiges Beispiel auf dem 464 zu programmieren, ist folgende Befehlsfolge notwendig:

```
a$="Hallo":!WPRINT,1,%a
```

Wenn 664/6128 Ihre unter M.O.S. geschriebenen Programme auch auf einem 464 ablaufen lassen wollen, sollten Sie der Kompatibilität ebenfalls die Stringdescriptor verwenden.

3.3 Aufbau der Befehlsübersicht

Die Befehle sind in fünf Gruppen unterteilt: In Windowbefehle, Menübefehle, Iconbefehle, Mausbefehle und sonstige Befehle.

Zuerst folgt der Befehl und dessen Syntax. Parameter in Klammern sind optional, müssen also nicht übergeben werden.

Anschließend folgt eine genaue Beschreibung des Befehls und der zu übergebenden Parameter.

Zum Abschluß eines Befehls folgt manchmal ein Beispielprogramm. Diese sind so geschrieben, daß sie auch auf dem 464 laufen, es werden also Stringdescriptoren verwendet. Natürlich laufen die Programme nur, wenn vorher M.O.S. initialisiert wurde (s. 2.3).

Oft werden von den Programmen Fenster oder Icons geöffnet, die nach dem Abschluß des Programmes nicht geschlossen werden. Versuchen Sie einmal selber, diese Fenster, bzw. Icons mit den Befehlen zu schließen. Erst wenn alles geschlossen ist, können Sie das nächste Beispielprogramm ausprobieren. Sollten Sie es nicht schaffen, die Fenster/Menüs/Icons zu schließen, so tippen Sie !MOSOFF;!MOSON. Dann können Sie das nächste Beispielprogramm eintippen.

3.4 Die Windowbefehle

Die Windowbefehle dienen dazu, die zwanzig neuen Textfenster zu verwalten, zu öffnen, zu schließen ...

1. Der Befehl: !WOPEN,Winnr,X,Y,Breite,Höhe<,"Par"><,"Name">

Dieser Befehl eröffnet eines der zwanzig Fenster. Das gerade geöffnete Fenster hat immer die höchste Priorität und wird über alle vorhandenen geklappt.

Winnr gibt die Nummer des Windows an. An dieser Nummer wird das Fenster immer identifiziert.

X und Y geben die Textkoordinaten der oberen, linken Ecke des Fensters an. Wobei X die waagerechte Achse und Y die senkrechte Achse ist. Begonnen wird beim Zählen mit eins, sodaß die Koordinaten 1,1 die linke, obere Ecke des Bildschirms bedeuten.

Breite und Höhe geben die Größe des Fensters in Textkoordinaten an. Das Fenster muß sich aber vollständig im Bildschirm befinden, sonst wird die Fehlermeldung OUT OF SCREEN ausgegeben.

Damit könnte der Befehl schon abgeschlossen sein. Es ist aber mit dem nächsten Parameter möglich, daß Aussehen des Fensters näher zu spezifizieren.

Es gibt vier verschiedene Parameter, die in beliebiger Reihenfolge ohne Kommata getrennt hinter den Parameter Höhe angehängt werden können. Diese müssen aber in Anführungszeichen oder in einem String stehen (CPC 464: String vorher definieren und Descriptor übergeben).

- K - Das Fenster erhält eine Kopfzeile.
- C - Das Fenster erhält in der Kopfzeile einen Closepoint.
- R - Das Fenster erhält einen Rahmen
- I - Das Fenster wird invers aufgebaut

Ein Closepoint kann nur eingerichtet werden, wenn auch eine Kopfzeile eingerichtet wird. Wird versucht einen Closepoint ohne Kopfzeile aufzubauen, folgt die Fehlermeldung PARAMETER ERROR.

Der bedruckbare Bereich eines Fensters verkleinert sich entsprechend den Parametern:

K - Höhe verkleinert sich um 2 Zeilen
 R - Breite verkleinert sich um 4, Höhe um 2 Zeilen
 KR - Breite verkleinert sich um 4, Höhe um 3 Zeilen

Alle Fenster müssen mindestens so groß sein, daß es eine bedruckbare Position gibt. Daraus und aus obigen Verkleinerungen bei bestimmten Parametern ergeben sich folgende Mindestgrößen für die Fenster:

<u>Parameter</u>	<u>min. Breite</u>	<u>min. Höhe</u>
Keine Parameter	1	1
Kopfzeile	1	3
Kopfzeile und Closepoint	4	3
Rahmen	5	3
Kopfzeile und Rahmen	5	4
Kopfzeile und Closepoint und Rahmen	5	4

Wird die Mindestgröße unterschritten, wird die Fehlermeldung WINDOW TOO SMALL ausgegeben.

Ist ein Kopf vorhanden, kann in diesen auch noch eine Überschrift eingeblendet werden. Die Überschrift wird an die Parameter durch Komma getrennt angehängt. Sie muß aus einem String bestehen. Entweder setzt man die Überschrift in Anführungszeichen oder man schreibt die Stringvariable, in der sich die Überschrift befindet an Stelle des Strings in Anführungsstrichen. Verknüpfungen von Strings geschehen durch '+'. CPC 464 Besitzer müssen den String-descriptor verwenden, siehe 3.2.

Ist kein Kopf vorhanden oder ist der String Null Zeichen lang, wird die Fehlermeldung PARAMETER ERROR ausgegeben.

Das Fenster muß mindestens um vier Zeichen breiter sein, als der String lang ist. Hat das Fenster auch einen Closepoint muß das Fenster mindestens sieben Zeichen breiter sein. Ist das nicht der Fall, so wird die Fehlermeldung HEADLINE TOO LONG ausgegeben.

Haben Sie alles richtig gemacht, erscheint das Fenster nun blitzschnell auf dem Bildschirm (die Überschrift wird automatisch zentriert) und erhält die höchste Priorität, d.h. es liegt ganz oben. Jedes Fenster hat seine eigene Cursorposition, diese wird beim Eröffnen auf die erste beschreibbare Stelle oben links im Fenster (Rand und Kopf können ja nicht beschrieben werden) gesetzt.

Einige Besonderheiten: Normalerweise dürfen sich zwei Fenster beliebig überschneiden, allerdings nicht im Grafikfenster, sonst wird die Fehlermeldung WINDOW OVERWRITTEN IN GRAFIKWINDOW ausgegeben. Allerdings gibt es eine Möglichkeit, daß sich auf Fenster im Grafikfenster überlappen dürfen. Zu diesem Zweck wurde der Befehl !OVER eingerichtet. (s.3.8). Ist das Überlappen im Grafikfenster erlaubt, so ist darauf zu achten, daß auch wirklich das zuletzt geöffnete Fenster zuerst geschlossen wird, usw. Sonst kommt der Rechner durcheinander und der Bildschirm wird nicht richtig aktualisiert.

Sind Menüs geöffnet, dürfen ebenfalls keine neuen Windows mehr eröffnet werden, ansonsten wird die Fehlermeldung MENU OPEN ausgegeben.

Jede Textposition eines neu eröffneten Fensters kostet ein Byte Speicherplatz. Befindet sich die Textposition im Grafikfenster, so werden neun Bytes verbraucht. Steht im Puffer nicht mehr genug freier Speicherplatz zur Verfügung, so wird die Fehlermeldung NO ROOM FOR WINDOW ausgegeben.

Beispielprogramm:

```
10 A$="KR":B$="TEST"  
20 !wopen,1,10,10,10,10,%a$,%b$
```

2. Der Befehl: !WCLOSE<,Winnr,Winnr,...>

Dieser Befehl dient zum Schließen von Fenstern.

Werden beim Befehl WCLOSE keine Parameter angegeben, so werden alle geöffneten Fenster geschlossen. Ansonsten können beliebig viele Fenster angegeben werden, die geschlossen werden sollen. Sind die Fenster nicht geöffnet, wird die Fehlermeldung WINDOW NOT OPEN ausgegeben.

Wenn Menüs geöffnet sind, dürfen keine Fenster geschlossen werden.

Alle dem zu schließenden Fenster zugeordneten Icons werden ebenfalls geschlossen.

Beispielprogramm:

```
10 !WCLOSE,1,3,6 Die Fenster 1,3 und 6 werden geschlossen
```

3. Der Befehl: !WPRINT,Winnr,"Text"

Mit diesem Befehl kann ein Fenster mit Text beschrieben werden.

Der Parameter Winnr gibt das Fenster an, in dem der Text geschrieben werden soll.

Ist das letzte Zeichen im String ein Semikolon, so wird der nächste in diesem Fenster ausgedruckte Text direkt dahinter gedruckt. Ansonsten wird ein Zeilenvorschub im Fenster durchgeführt.

Ist das Fenster invers aufgebaut, bzw. gelöscht (!WCLS) so wird auch der Text invers ausgegeben, es sei denn die Inversfunktion ist zusätzlich auch noch angeschaltet (!INVERS).

Selbstverständlich darf man nicht in ungeöffnete Fenster schreiben.

Gelangt der Cursor beim Schreiben im Fenster auf den unteren Rand oder gelangt er unten aus dem Fenster, so wird die Fehlermeldung OUT OF WINDOW ausgegeben, der Fensterinhalt wird nicht gescrollt.

Um einen Zeilenvorschub zu erreichen, kann man die Befehlsfolge A\$=" ":!WPRINT,winnr,%a\$ (664/6128: !WPRINT," ") benutzen.

Beispielprogramm:

```

10 A$="KRC":B$="TEST"
20 !WOPEN,1,10,10,30,12,%A$,%B$
30 C$="HALLO ;":D$="M.O.S."
40 FOR I=1 TO 3:!WPRINT,1,%C$;!WPRINT,1,%D$:NEXT

```

4. Der Befehl: !WCLS,0 oder 1,<Winnr,Winnr,...>

Dieser Befehl dient zum Löschen des Fensterinhalts.

Der erste Parameter ist zwingend vorgegeben. Mit ihm wird festgelegt, ob die Fenster invers (1) oder normal (0) gelöscht werden sollen. Wird sonst kein Parameter mehr angegeben, werden alle geöffneten Fenster in der beschriebenen Weise gelöscht, ansonsten nur die angegebenen. Diese müssen aber geöffnet sein.

Beispielprogramm:

```

10 A$="KRC":B$="TEST"
20 !WOPEN,1,10,10,30,12,%A$,%B$
30 !WCLS,1,1:WHILE INKEY$="" :WEND:!WCLS,0,1

```

5. Der Befehl: !WLOCATE,Winnr,X,Y

Mit diesem Befehl wird der Fenstercursor positioniert.

Jedes Fenster hat ja einen eigenen Fenstercursor, der mit diesem Befehl positioniert werden kann. Winnr gibt die Nummer des Fensters an, und X und Y die neuen Koordinaten des Cursors, wobei die erste beschreibbare Positon des Fensters die Koordinaten (1,1) hat. Wird der Cursor außerhalb des Fensters positioniert, wird die Fehlermeldung OUT OF WINDOW ausgegeben. Der Cursor kann nur in geöffneten Fenstern positioniert werden.

Beispielprogramm:

```

10 A$="KR":B$="TEST"
20 !WOPEN,1,10,10,30,12,%A$,%B$
30 C$="+;":FOR x=10 TO 1 STEP -1:FOR y=10 TO 1 STEP -1
40 !WLOCATE x,y:!WPRINT,1,%C$:NEXT:NEXT

```

6. Der Befehl: !WSUB,Winnr,Y<,Y,Y,...>

Damit wird das angegebenen Fenster unterteilt. Dies geht allerdings nur, wenn das Fenster einen Rahmen hat. Ist das nicht der Fall, so wird die Fehlermeldung NO BORDER ausgegeben.

An den angegebenen Y-Positionen des Fensters wird eine Unterteilung vorgenommen. Die Y-Koordinate eins entspricht der ersten Reihe des Fensters. Die angegebene Koordinate darf natürlich nicht außerhalb des Fensters liegen.

Beispielprogramm:

```

10 A$="KR":B$="TEST"
20 !WOPEN,1,10,10,10,10,%A$,%B$
30 !WSUB,1,5,7,9

```

7. Der Befehl: !WBAR,Winnr,Barnr,Y,<Y,Y,...>

Auch dieser Befehl stellt ein Möglichkeit dar, ein Fenster zu unterteilen. Im angegebenen Fenster (welches diesmal keinen Rahmen haben muß) werden an den angegebenen Stellen Unterteilungen eingefügt, die durch den Parameter Barnr spezifiziert werden. Der seitliche Rahmen wird bei diesem Befehl nicht überschrieben, sondern nur der bedruckbare Bereich des Fensters. Es können aber selbstverständlich alle Reihen des Fensters durch !WBAR angesteuert werden, auch der Kopf und der Rahmen.

Als Barnummer sind Werte zwischen eins und fünf erlaubt:

<u>Barnr.</u>	<u>Zeichen</u>	<u>Funktion</u>
1		Es werden Gleichheitszeichen ausgegeben
2		Es werden Minuszeichen ausgegeben
3		Es wird das Grafikzeichen des Kopfes ausgegeben
4		Es wird ein gefüllter Kasten ausgegeben
5		Es wird das Borderzeichen 15 ausgegeben

Beispielprogramm:

```
10 A$="KR":B$="TEST"
20 !WOPEN,1,10,10,10,10,%A$,%B$
30 FOR i=1 TO 5:!WBAR,1,i,i+3:NEXT
```

3.5 Die Menübefehle

Die Menübefehle sind zum größten Teil ähnlich aufgebaut, wie die Windowbefehle, nur das im Allgemeinen die Angabe der Windownummer entfällt und immer auf das zuletzt geöffnete Menü zugegriffen wird. Es können bis zu fünf Menüs eröffnet werden.

8. Der Befehl: !MOPEN,X,Y,Breite,Höhe<,"Par","Text">

Die Syntax und der Befehlsaufbau sowie dessen Funktion ist mit dem des Befehls !WOPEN nahezu identisch. Sind bereits alle fünf Menüs geöffnet, wird die Fehlermeldung ALL MENUS OPEN ausgegeben.

Beispielprogramm

```
10 A$="KRCI":B$="TEST"
20 !MOPEN,10,10,15,10,%A$,%B%
```

9. Der Befehl: !MCLOSE

Mit diesem Befehl wird das zuletzt geöffnete Menü geschlossen.

Beispielprogramm

```
!MCLOSE
```

10. Der Befehl: !MPRINT,"Text"

Mit diesem Befehl wird das zuletzt eröffnete Menü beschrieben.
Alles weitere siehe !WPRINT.

Beispielprogramm

```
10 A$="KRC":B$="TEST"  
20 !MOPEN,10,10,15,10,%A$,%B$  
30 C$="HALLO":!MPRINT,%C$
```

11. Der Befehl: !MCLS,0 oder 1

Das zuletzt eröffnete Menü wird normal oder invers gelöscht (s. !WCLS).

Beispielprogramm

```
10 A$="KCR":B$="TEST"  
20 !MOPEN,10,10,15,10,%A$,%B$  
30 C$="HALLO":!MPRINT,%C$  
40 !MCLS,1:WHILE INKEY$="" :WEND:!WCLS,0
```

12. Der Befehl: !MLOCATE,X,Y

Der Cursor des zuletzt geöffneten Fensters wird positioniert (s. !WLOCATE)

Beispielprogramm:

```
10 A$="KR":B$="TEST"  
20 !MOPEN,10,10,30,12,%A$,%B$  
30 C$="+;":FOR x=10 TO 1 STEP -1:FOR y=10 TO 1 STEP -1  
40 !MLOCATE x,y:!MPRINT,%C$:NEXT:NEXT
```

13. Der Befehl: !MSUB,Y<,Y,Y,...>

Das aktuelle Menü wird unterteilt (s. !WSUB).

Beispielprogramm:

```
10 A$="KR":B$="TEST"  
20 !MOPEN,10,10,10,10,%A$,%B$  
30 !MSUB,5,7,9
```

14. Der Befehl: !MBAR,Barnr,Y<,Y,Y,...>

In das aktuelle Menü wird ein Balken eingeblendet (s. !WBAR).

Beispielprogramm:

```
10 A$="KR":B$="TEST"  
20 !MOPEN,10,10,10,10,%A$,%B$  
30 FOR i=1 TO 5:!MBAR,i,i+3:NEXT
```

3.6 Die Iconbefehle

Icons sind kleine Pictogramme, die auf den Bildschirm gesetzt werden können. Bis zu 20 Icons können gesetzt werden. Die Icons nehmen beim Setzen keinen Speicherplatz in Anspruch, trotzdem wird der darunterliegende Bildschirminhalt gespeichert. Icons sind vier mal zwei Printpositionen groß.

15. Der Befehl: !SETICON,Icnnr,X,Y,<,Winnr>

Das durch den Parameter Icnnr spezifizierte Icon wird an der durch die Parameter X und Y spezifizierten Bildschirmposition gesetzt. Ist kein weiteres Parameter angegeben, so wird das Icon auf den Textspeicher gesetzt, es sei denn, das Grafikfenster ist an, dann wird das Icon soweit es sich im Bereich des Grafikfensters befindet, unter das Grafikfenster und über das Textfenster gesetzt.

Ist noch ein weiterer Parameter angegeben, so wird das Icon in das durch diesen Parameter spezifizierte Fenster gesetzt. Die Koordinaten sind dann relativ von der linken, oberen Ecke des Fensters gerechnet, wobei die Koordinaten (1,1) der linken oberen Ecke des Fensters entspricht. Dort befindet sich dann auch die linke, obere Ecke des Icons.

Beispielprogramm:

```
10 !SETICON,1,10,10
```

16. Der Befehl: !ICONOFF<,Icnnr,Icnnr,...>

Mit dieser Funktion kann man angeschaltete Icons ausschalten. Wird kein Parameter angegeben, so werden alle auf dem Bildschirm befindlichen Icons gelöscht. Ansonsten werden nur die angegebenen Icons gelöscht, wobei man darauf achten muß, daß nur geöffnete Icons wieder geschlossen werden. Bei Nichtbeachtung wird die Fehlermeldung ICON NOT OPEN ausgegeben.

Beispielprogramm:

```
10 !SETICON,1,10,10:WHILE INKEY$="":WEND  
20 !ICONOFF
```

17. Der Befehl: !SETGRAFIK,Icnnr,Grafiknr

Normalerweise ist jedem Icon ein verschiedenes Pictogramm zugeordnet. Es ist aber auch möglich Icons ein gleiches Pictogramm zuzuordnen.

Durch diesen Befehl wird dem Icon mit der Nummer icnnr das Grafiksymboll des Icons grafiknr zugeordnet. Das neue Aussehen des Icons wird allerdings nicht gleich auf dem Bildschirm aktualisiert, sondern erst beim nächsten !SETICONbefehl, der sich auf dieses Icon bezieht, bzw. wenn das Icon von einem Window überlappt wird und dieses dann weggeklappt wird.

Beispielprogramm:

```
10 FOR I=1 TO 20:PRINT "ICON";I  
20 !SETGRAFIK,1,I:!SETICON,1,10,10
```

```
30 WHILE INKEY$="":WEND
40 !ICONOFF:NEXT
```

18. Der Befehl: !INVERSICON<,Icnnr,Icnnr,...>

Die angegebenen Icons werden auf dem Bildschirm invertiert. Zu beachten ist, daß das Aussehen der Icons im Speicher nicht verändert wird, sondern lediglich auf dem Bildschirm. Bei einem nochmaligen Aufruf des Befehls wird das gleiche Icon aber nicht nocheinmal invertiert, es wird immer nur das normale Aussehen des Icons invertiert ausgegeben. Dabei ist zu beachten, daß immer nur der sichtbare Teil des Icons invertiert wird. Werden keine Parameter angegeben, so werden alle sichtbaren Iconteile invertiert, ansonsten nur die sichtbaren Iconteile der angegebenen Icons. Die Icons müssen allerdings auch geöffnet werden.

Beispielprogramm:

```
10 !SETICON,1,10,10:WHILE INKEY$="":WEND
20 !INVERSICON
```

3.7 Die Mausbefehle

Der Benutzer hat ebenfalls die Möglichkeit ein Maussymbol über den Bildschirm zu steuern und damit bei entsprechender Programmierung verschiedene Fenster mit diesem Maussymbol anzu steuern.

Beispielprogramme folgen jetzt nicht mehr. Sollte richtig etwas demonstriert werden, würden die Programme recht lang. Ich verweise Sie also statt dessen auf unser grosses erläutertes Beispielprogramm (s.9)

19. Der Befehl: !LOCATEMOUSE,X,Y

Dieser Befehl setzt das Maussymbol auf die angegebenen Textkoordinaten. Das Maussymbol wird allerdings nicht auf dem Bildschirm angezeigt, es wird mit dem Befehl nur festgelegt an welcher Stelle das Maussymbol bei der nächsten Auswahl erscheinen soll.

20. Der Befehl: !LIMIT,X,Y,Breite,Höhe

Mit diesem Befehl kann man den Bereich festlegen, in dem sich der Mauszweiger bewegen darf. X und y geben die Textkoordinaten der linken, oberen Ecke des Bereiches an, in dem sich die Maus bewegen darf, Breite und Höhe die Größe. Befindet sich die aktuelle Mausposition nicht innerhalb des Bereiches, wird sie auf die linke, obere Ecke des Mausbereiches gesetzt.

21. Der Befehl: !LIMITOFF

Es kann wieder der ganze Bildschirm vom Maussymbol befahren werden. Das Maussymbol wird auf die linke, obere Ecke des Bildschirms gesetzt (nicht angezeigt).

22. Der Befehl: !SPEEDMOUSE<,Speed>

Auch die Geschwindigkeit der Maus lässt sich mühelos festlegen. Die Geschwindigkeit kann im Bereich von eins bis hundert gewählt werden, wobei hundert am Schnellsten und eins am Langsamsten ist. Wird kein Parameter angegeben, wird eine Geschwindigkeit von vierzig eingestellt. Dies entspricht der anfangs eingestellten Geschwindigkeit.

23. Der Befehl: !WKLICK<,Winnr,Winnr,Winnr,..)

Mit diesem Befehl definiert man ein Fenster als 'anklickbar', d.h. bei einer Abfrage mit dem Mausymbol mit Hilfe des Befehles !WAITMOUSE (s.u.) kann dieses Window angeklickt werden. Die Abfrage wird dann abgebrochen und die Kontrolle wird an den Basic-Interpreter zurückgegeben. Ist das Fenster nicht anklickbar und wird trotzdem mit der Maus angeklickt, passiert absolut überhaupt nichts.

Werden keine Parameter übergeben, werden alle offenen Fenster als anklickbar definiert. Ansonsten werden die bereits anklickbaren und die durch den Befehl als anklickbar gewählten Fenster als anklickbar definiert. Man kann also seine Fenster in mehreren Etappen als anklickbar definieren. Um z.B. die Fenster 1,2 und 3 anklicken zu können, können Sie den Befehl !WKLICK,1,2,3 oder auch !WKLICK,1:!WKLICK,2:!WKLICK,3 benutzen.

Einen Sonderfall stellt das Parameter Null dar. Entdeckt der Rechner eine Null in der Befehlsfolge, so werden die nächsten vier Parameter nicht als Windownummern angesehen, sondern als x,y-Koordinate, sowie als Breite und Höhe eines frei definierbaren Bereiches, der nun ebenfalls angeklickt werden kann, ohne das sich dort ein Fenster befindet. Es müssen aber nach der Null mindestens vier Parameter folgen um das frei definierbare Fenster eindeutig festzulegen. Danach können dann wieder andere als anklickbar definierte Fenster folgen. Ein Beispiel: Die Fenster 11 und 14 sollen als anklickbar definiert werden, sowie ein Feld mit den Startkoordinaten (5,8) und der Ausdehnung (Breite,Länge=30,10). Dazu genügt der Befehl !WKLICK,11,0,5,8,30,10,14, wobei die mittleren 5 Koordinaten der Festlegung des frei definierbaren Fensters entsprechen.

Selbstverständlich können nur offene Fenster als anklickbar definiert werden.

24. Der Befehl: !WKLICKOFF<,Winnr,Winnr,Winnr,...>

Der Befehl !WKLICKOFF ist, wie Sie sich sicher bereits gedacht haben, die genaue Gegenfunktion zum Befehl !WKLICK. Mit ihm werden Fenster als nicht anklickbar definiert. Werden keine Parameter angegeben, werden alle Fenster als nicht anklickbar definiert. Werden Parameter angegeben, werden die angegebenen als nicht anklickbar definiert, wobei es völlig egal ist, ob das Fenster überhaupt jemals als anklickbar definiert wurde und ob es überhaupt geöffnet wurde. Zugelassen sind Werte zwischen Null und zwanzig, die den entsprechenden Fenstern zugeordnet sind, wobei Null das frei definierbare Fenster als nicht mehr anklickbar definiert.

25. Der Befehl: !WAITMOUSE<,"Par">

Waitmouse dient zur Abfrage des Maus.

Die Maus wird an der vorher spezifizierten Stelle eingeblendet und kann mit Joystick oder Maus gesteuert werden.

Angehängt wird dem Befehl die Anklickarten, die zulässig sind:

- k - KLICK Drücken der Feuertaste höchstens 0.25 Sekunden und dann mindestens 0.25 kein Drücken der Feuertaste.
- d - DOUBLE KLICK Drücken der Feuertaste höchstens 0.25 Sekunden und nach höchstens 0.25 Sekunden nochmaliges Drücken.
- p - PRESS Mindestens 0.25 Sekunden Drücken der Feuertaste.

Um z.B. die Klickarten KLICK und DOUBLE KLICK zulassen, wäre der Befehl !WAITMOUSE,"kd" (464: A\$="KD":!WAITMOUSE,%A\$) notwendig.

Wird kein Parameter übergeben, wartet !WAITMOUSE auf irgendeinen Tastendruck.

Es gibt nun mehrere Arten, um den Befehl !WAITMOUSE zu beenden. Man kann z.B. die Abbruchtaste (DELETE) drücken, oder auf eine der beschriebenen Arten die Feuertaste drücken. Allerdings muß sich die Maus in einem als anklickbar definierten Fenster befinden.

Folgende Systemvariablen haben relevante Werte:

In der Systemvariablen KLKART (&7301) ist die Art des Anklickens enthalten:

- 1 - KLICK
- 2 - DOUBLE KLICK
- 3 - PRESS
- 255 - Abbruch (Delete)

In der Systemvariablen KLKWIN (&7300) ist die Nummer des Fensters, welches angeklickt wurde, enthalten. Wurde der Closepoint angeklickt, so wird zu der Nummer des Windows noch 32 addiert.

26. Der Befehl: !MCHOICE,Y,<,Y,Y,Y,...>

Diese Funktion ist sehr komplex. Mit ihr kann man den Benutzer einen Menüpunkt auswählen lassen. Zuerst sollte man ein Menü mit den oben genannten Befehlen aufbauen.

Der Benutzer kann durch diesen Befehl zunächst einmal festlegen, welche Textreihen im Menü er als Menüpunkte ansieht, denn es gibt ja auch Leerzeilen oder Unterteilungen (MSUB,MBAR), die der Benutzer nicht anwählen sollte. Mit den angegebenen Parametern wird dies festgelegt. Durch den Befehl !MCHOICE,2,3,4 werden z.B. die Reihen zwei, drei und vier als Menüpunkte definiert. Nur Sie darf der Benutzer nachher anwählen. Der Parameter eins entspricht der ersten Reihe des Fensters, dabei kann es sich z.B. auch um die

Kopfzeile handeln.

Keine Reihe eines Menüs darf zweimal angewählt werden. Versucht der Benutzer dies in seinem Befehl, wird die Fehlermeldung DOUBLE CHOICE ausgegeben.

Sind alle Menüpunkte ausgewählt, wird der Mauszeiger auf den ersten angegebenen gesetzt. Dann kann durch Bewegen des Mauszeigers innerhalb des Menüs ein Punkt ausgewählt werden. Der augenblicklich angewählte Punkt ist invers dargestellt. Nicht als Menüpunkte spezifizierte Punkte können nicht ausgewählt werden. Allerdings kann jederzeit mit der DELETE-Taste abgebrochen werden und auch der CLOSEPOINT angeklickt werden, sofern vorhanden.

Die Systemvariable ARTMEN (&74D6) gibt die Art an, wie ein Punkt ausgewählt wurde:

255 - DELETE (Abbruch)
0 - Normale Auswahl oder Closepoint

In der Systemvariablen KLKMEN (&74D7) ist die Nummer des angewählten Punktes abgelegt. Und zwar ist die erste im Befehl angegebene Y-Koordinate der Menüpunkt eins, die zweite angegebene Koordinate der Menüpunkt zwei, usw.

Wird der Closepoint eines Menüs angeklickt, so hat die Systemvariable den Wert 255.

3.8 Sonstige Befehle

27. Der Befehl: !MOSOFF

Dieser Befehl schaltet das MOS Programm ab. Der Benutzer hat nun den Eindruck wieder einen ganz normalen Schneider vor sich zu haben. Die Bildschirmausgabe läuft wieder wie üblich ab und alle M.O.S.-Befehle, bis auf !MOSON, laufen auf Syntaxerror.

28. Der Befehl !MOSON

Damit kann MOS wieder angeschaltet werden. Die Bildschirmausgabe ist wieder gepatcht und alle Befehle sind wieder erlaubt. Es wird automatisch in MODE 2 geschaltet.

29. Der Befehl: !CLOSEOFF

Nach Ausführung dieses Befehls werden Windows und Menüs nach Anklicken eines Closepointes nicht geschlossen.

30. Der Befehl: !CLOSEON

Nach Ausführung dieses Befehls werden Windows und Menüs ab sofort wieder beim Anlicken des Closepoints selbstständig geschlossen.

31. Der Befehl: !SWAPSET

Mit diesem Befehl wird der aktuelle Bordersatz durch den zweiten

ausgetauscht. Es werden aber nicht sofort alle auf dem Bildschirm vorhandenen Borderteile aktualisiert. Nur im normalen Arbeitsgang aktualisierte bzw. neu eröffnete Fenster werden im neuen Zeichensatz ausgegeben.

Der zweite Bordersatz entspricht den Zeichen 16 bis 31.

32. Der Befehl: !INVERS,0 oder 1

Dieser Befehl schaltet in den inversen oder in den normalen Modus. Bei !INVERS,1 wird in den inversen, bei !INVERS,0 wird in den normalen Modus geschaltet. Wird in den inversen Modus geschaltet, werden in Zukunft alle Texte invers ausgegeben, nur in inversen Fenstern wird der Text normal ausgegeben.

33. Der Befehl: !GRAFIKW,X,Y,Breite,Höhe

Mit diesem Befehl kann man in einem bestimmten Bereich auch Grafik beim Überlappen erhalten und zwischenspeichern. Bei Ausführung des Befehls darf allerdings weder ein Fenster noch ein Menü geöffnet sein. Im angegebenen Bereich wird dann Grafik beim Überlappen zwischengespeichert.

34. Der Befehl: !GRAFIKWOFF

Dieser Befehl schaltet die Zwischenspeicherung von Grafik in einem durch !GRAFIKW spezifizierten Bereich ab. Auch bei der Ausführung dieses Befehls müssen alle Menüs und Fenster geschlossen sein.

35. Der Befehl: !OVER,0 oder 1

Der Befehl !OVER erlaubt oder verbietet das Überlappen von Fenstern im Graifenster. !OVER,0 verbietet das Überlappen, !OVER,1 erlaubt es.

36. Der Befehl: !BUFFER,Start,Länge

Im Rechner ist unter MOS ein bestimmter Speicherbereich festgelegt, der für die Speicherung der Überlappten Teile dient. Dabei ist noch einmal zu Erwähnen, daß bei jedem Fenster und Menü pro Printposition ein Byte Speicherplatz und wenn sich diese Printposition im Grafikfenster befindet, dann neun Bytes Speicherplatz benötigt wird. Icons benötigen keinen Speicherplatz. Wenn jetzt der Raum nicht ausreicht, um z.B. noch ein erneutes Fenster zu öffnen meldet sich der Rechner mit NO ROOM FOR WINDOW. Um dem abzuhelpfen, kann man mit dem Befehl !BUFFER die Startadresse und die Länge dieses Zwischenspeichers frei wählen. Zunächst schaltet dieser Befehl MOS ab, um sicherzugehen, daß keinerlei Fenster und Menüs geöffnet sind. Dann wird der Befehl abgearbeitet, der neue Buffer eingerichtet. Dabei ist darauf zu achten, daß der Buffer nicht das MOSprogramm überlappt oder vor der durch den Befehl MEMORY gesetzten Grenze beginnt. Wird der Befehl ohne Fehlermeldung einwandfrei abgearbeitet, wird MOS selbstverständlich automatisch wieder eingeschaltet.

4. Einsprünge der Befehle von Maschinensprache

Unter M.O.S. hat man den grossen Vorteil, direkt von Maschinensprache aus, die M.O.S. Routinen und Befehle aufzurufen. In diesem Kapitel werden die genauen Aufrufverfahren für jeden der 36 Befehle vorgestellt.

Achtung!

Bei den Maschinenspracheeinsprünge muß besondere Sorgfalt darauf verwendet werden, daß die Parameter alle stimmen, da die Einsprünge zum Teil nach den Syntax und Parameterüberprüfungen einsetzen. Dadurch wird Zeit eingespart, die bei der Programmierung in MC ja besonders wichtig ist. Werden falsche Parameter übergeben, so kann dies eventuell zu einem Absturz führen, obwohl bei der Verwendung eines gleichgearteten Basicbefehl der Rechner sich nur mit einer Fehlermeldung gemeldet hätte.

4.1 Aufbau der Übersicht

Zuerst ist der Befehlsname aufgeführt. Dahinter sind die notwendigen Übergabeparameter aufgeführt. Zum Schluß folgt schließlich der CALL, mit dem der Befehl aufgerufen werden kann.

Bei manchen Befehlen ist allerdings kein Aufruf notwendig. Dort ist es absolut ausreichend bestimmte Systemvariablen auf entsprechende Werte zu setzen.

4.2 Einsprünge in die Windowbefehle

1. !WOPEN

&72E6 - Fensternummer

In der Systemvariablen &72EB muß der Status eingetragen werden:

- BIT 7 - Rahmen, wenn gesetzt
- BIT 6 - Kopf, wenn gesetzt
- BIT 5 - Closepoint, wenn gesetzt
- BIT 4 - Invers, wenn gesetzt
- BIT 3 - Überschrift, wenn gesetzt

&72E7 - X-Koordinate des Fensters links

&72E8 - Y-Koordinate des Fensters oben

&72E9 - Breite des Fensters

&72EA - Höhe des Fensters

&72DA - Startadresse der Überschrift (2 Byte), wenn durch &72EB ausgewählt

&72DC - Länge der Überschrift (1 Byte), wenn durch &72EB angew.

AUFRUF: CALL &7D91

2. !WCLOSE

a) Ein Fenster

&72E6 - Nummer des Fensters

AUFRUF: CALL &8047

b) Alle Fenster

AUFRUF: CALL &8070

3. !WPRINT

A - Nummer des Fensters

B - Länge des Strings

DE- Startadresse des Strings

Carryflag=1 ==> Zeilenvorschub

AUFRUF: CALL &92FD

4. !WCLS

a) Ein Fenster

&72E6 - Nummer des Fensters

&8AA7 - 0 = Nicht invers löschen

128 = Invers löschen

AUFRUF: CALL &8258

b) Alle Fenster

&8AA7 - siehe oben

AUFRUF: CALL &8244

5. !WLOCATE

A - Nummer des Fensters

L - X-Koordinate im Fenster

H - Y-Koordinate im Fenster

AUFRUF: CALL &92F2

6. !WSUB

&72E6 - Fensternummer

A - Y-Coordinate

AUFRUF: CALL &88E7

7. !WBAR

A - Y-Coordinate der Unterteilung

&894A - ASCII des Balkens (z.B. 45 für das Minuszeichen)

&72E6 - Nummer des Fensters

AUFRUF: CALL &89B5

4.3 Einsprünge in die Menübefehle

8. !MOPEN

s. !WOPEN, nur &72E6 entfällt

AUFRUF: CALL &92E0

9. !MCLOSE

AUFRUF: CALL &92CB

10. !MPRINT

s. !WPRINT, nur A entfällt

AUFRUF: CALL &92C6

11. !MCLS

&8AA7 - 0 - Nicht invers löschen
128 - Invers löschen

AUFRUF: CALL &92B8

12. !MLOCATE

H - Y-Coordinate
L - X-Coordinate

AUFRUF: CALL &92C1

13. !MSUB

A - Y-Coordinate

AUFRUF: CALL &9294

14. !MBAR

s. !WBAR, &72E6 entfällt

AUFRUF: CALL &9289

4.4 Einsprünge in die Iconbefehle

15. !SETICON

A - Iconnummer
D - Y-Coordinate
E - X-Coordinate

Carryflag=1 ==> Icon ist Fenster zugeordnet
C - Fensternummer (0 wenn Icon keinem Fenster zugeordnet)

AUFRUF: CALL &921E

16. !ICONOFF

a) Alle Icons aus

AUFRUF: CALL &8672

b) Ein Icon aus

A - Iconnummer

AUFRUF: CALL &9283

17. !SETGRAFIK

A - Iconnummer

B - Grafiknummer

AUFRUF: CALL &927B

18. !INVERSICON

a) Alle Icons invertieren

AUFRUF: CALL &868A

b) Ein Icon invertieren

A - Nummer

AUFRUF: CALL &86A1

4.5 Einsprünge in die Mausbefehle

19. !LOCATEMOUSE

&72FE - X

&72FF - Y

AUFRUF: CALL &8FD1

20. !LIMIT

&7305 - X

&7306 - Y

&7307 - Breite

&7308 - Höhe

AUFRUF: CALL &84F6

21. !LIMITOFF

AUFRUF: CALL &8510

22. !SPEEDMOUSE

&74B4 - Geschwindigkeit (1-100)

&74B5 - "

AUFRUF: Nicht erforderlich

23. !WKLICK

a) Alle Fenster die offen sind, als anklickbar definieren

AUFRUF: CALL &8434

b) Ein Textfenster als anklickbar definieren

A - Nummer des Fensters

AUFRUF: CALL &8463

c) Das frei definierbare Fenster

&7331 - X

&7332 - Y

&7333 - Breite

&7334 - Höhe

&730D - 255

AUFRUF: Nicht erforderlich

24. !WKLICKOFF

a) Alle Fenster

AUFRUF: CALL &849C

b) Ein Fenster

A - Nummer des Fensters (0-20)

AUFRUF: CALL &84BA

25. !WAITMOUSE

&870E - Klickart

Bit gesetzt ==> Klickart zulässig

Bit 0 - Klick

Bit 1 - Double Klick

Bit 2 - Press

AUFRUF: CALL &8756

26. !MCHOICE

A - Anzahl der Y-Coordinaten

DE - Startadresse, ab der die Coordinaten stehen (Ab DE müssen dann die Y-Coordinaten vom Programmierer eingetragen werden)

AUFRUF: CALL &929F

4.6 Einsprünge in die sonstigen Befehle

27. !MOSOFF

AUFRUF: CALL &782F

28. !MOSON

AUFRUF: CALL &780B

29. !CLOSEOFF

&74B3=0

AUFRUF: Nicht erforderlich

30. !CLOSEON

&74B3=1

AUFRUF: Nicht erforderlich

31. !SWAPSET

AUFRUF: CALL &888A

32. !INVERS

&72F3 - 128 - Inverse Ausgabe eingeschaltet

&72f3 - 000 - Inverse Ausgabe ausgeschaltet

AUFRUF: Nicht erforderlich

33. !GRAFIKW

H - Y

L - X

D - Höhe

E - Breite

AUFRUF: CALL &9264

34. !GRAFIKWOFF

AUFRUF: CALL &83CE

35. !OVER

&74DA - 0 - Überlappen nicht erlaubt.
&74DA - 1 - Überlappen erlaubt

36. !BUFFER

DE - Startadresse
BC - Länge

AUFRUF: CALL &933BP

5. Wichtige M.O.S.-Routinen

Im folgenden werden wichtige M.O.S.-Routinen vorgestellt. Zuerst wird die Funktion der Routine vorgestellt, dann die Einsprungsbedingungen, dann die Aussprungsbedingungen und abschließend die Adresse des Aufrufs.

1. Großbuchstaben in Kleinbuchstaben wandeln

EINSPRUNG:

A - ASCII-Zeichen

AUSSPRUNG:

A - Neues ASCII-Zeichen

AUFRUF: CALL &8C74

2. Diese Routine liefert die Adresse der Windowsystemvariablen, die ja bei jedem Window woanders liegt.

EINSPRUNG:

A - Nummer des Fensters

AUSSPRUNG:

HL - Adresse

AUFRUF: CALL &8C7A

3. Diese Routine berechnet die Bildschirm Speicheradresse einer Textkoordinate in MODE 2

EINSPRUNG:

H - Y-COORDINATE

L - X-COORDINATE

AUSSPRUNG:

HL - Adresse

AUFRUF: CALL &8C89

4. Ein ASCII-Zeichen ausdrucken

EINSPRUNG:

DE- Bildschirm Speicheradresse

A - ASCII-Zeichen

AUSSPRUNG:

AUFRUF: CALL &8CA5

5. Holt Buchstaben aus Textspeicher

EINSPRUNG:

H - Y-COORDINATE
L - X-COORDINATE

AUSSPRUNG:

A - ASCII des Zeichens

AUFRUF: CALL &8CE5

6. Berechnet Adresse einer Bildschirmposition im Textspeichers

EINSPRUNG:

H - Y-COORDINATE
L - X-COORDINATE

AUSSPRUNG:

HL- Adresse der Koordinaten im Textspeicher

AUFRUF: CALL &8CEC

7. Diese Routine blendet an einer Stelle des Bildschirms das Zeichen mit der an dieser Stelle höchsten Priorität ein.

EINSPRUNG:

H - Y-COORDINATE
L - X-COORDINATE

AUFRUF: CALL &8D0E

8. Kopiert die Systemvariablen eines Fensters in den festen Bereich ab der Systemvariable &7337.

EINSPRUNG:

A - Fensternummer (1-25, 21-25=Menüs)

AUSSPRUNG:

AUFRUF: CALL &8DB7

9. Kopiert die Systemvariablen eines Fensters aus dem festen Bereich ab der Systemvariablen &7337 zurück.

EINSPRUNG:

A - Fensternummer (s.o.)

AUSSPRUNG:

AUFRUF: CALL &8DC6

10. Textbildschirm auf dem Bildschirm ausgeben (ohne Rücksicht auf darüberliegende Fenster und ä.)

EINSPRUNG:

AUSSPRUNG:

AUFRUF: CALL &8DEC

11. Routine überprüft die Überschneidung zweier Bereiche

EINSPRUNG:

DE - Zeigt auf Bereich 1
HL - Zeigt auf Bereich 2

* Im Bereich sind in dieser Reihenfolge X,Y,Breite und Höhe des einen zu überprüfenden Bereiches abgelegt. *

AUSSPRUNG:

Carryflag=1 ==> Überschneidung
Carryflag=0 ==> Keine

AUFRUF: CALL &8E18

12. Routine überprüft und berechnet die Überschneidung zweier Bereiche

EINSPRUNG:

s. 11

AUSSPRUNG:

Carryflag wie oben

Wenn Überschneidung, dann ist ab &72F4 in dieser Reihenfolge X,Y,Breite und Höhe des Überschneidungsbereichs abgelegt.

AUFRUF: CALL &8E48

13. Routine überprüft, ob Textkoordinaten innerhalb eines Bereiches liegen.

EINSPRUNG:

D - Y
E - X
HL zeigt auf den Bereich (Format wie oben)

AUSSPRUNG:

Carryflag=1 - Koordinaten innerhalb
Carryflag=0 - Koordinaten außerhalb

AUFRUF: CALL &8E92

14. Multiplikationsroutine (Integer, 1 Byte)

EINSPRUNG:

H - Multiplikator A
L - " B

AUSSPRUNG:

HL - Ergebnis

AUFRUF: CALL &8EB0

6. Wichtige Systemvariablen

In diesem Kapitel werden wichtige Systemvariablen vorgestellt, deren Werte der Benutzer gebrauchen kann, z.B. um die genaue Position der Maus zu ermitteln.

Zum Format:

Zuerst wird der Name der Systemvariablen aufgeführt, dann die Adresse, wo diese abgelegt ist, anschließend die Anzahl der Bytes, die durch die Variable belegt wird und dann eine Funktionsbeschreibung der Systemvariablen.

Die Namen der Systemvariablen sind von mir gewählt und ansonsten irrelevant. Sie dienen lediglich zur Verdeutlichung der Funktion der Systemvariable.

Systemvariablen, die zwei Bytes umfassen, sind immer im Z80-Standard Format abgelegt, d.h. zuerst Low, dann Highbyte. Um z.B. den Wert der Systemvariable `adrwin` auszulesen, könnte folgende Befehlsfolge dienen:

```
PRINT PEEK(&72DE)+256*PEEK(&72DF)
```

1. `ADRWIN &72DE 2`

Startadresse des Fenster/Menüpuffers.

2. `LANWIN &72E0 2`

Länge des Puffers in Bytes.

3. `AKTWIN &72E2 2`

Aktueller Start des Puffers. Dieser Zeiger zeigt also auf den beschreibbaren, freien Teil des Puffers.

4. `AKTLAN &72E4 2`

Im Augenblick noch zur Verfügung stehender Speicherplatz im Buffer.

5. `INVFL &72F3 1`

Inversflag (128=Invers eingeschaltet, 0=Invers ausgeschaltet).

6. `XMAUS &72FE 1`

Aktuelle X-Text-Coordinate der Maus (1-80).

7. `YMAUS &72FF 1`

aktuelle Y-Text-Coordinate der Maus (1-25).

8. XGRA &7302 2

Aktuelle X-Grafik-Koordinate der Maus (0-639).

9. YGRA &7304 2

Aktuelle Y-Grafik-Koordinate der Maus (0-199).

10. AREAB &7305 4

In diesen 4 Bytes sind die Limitkoordinaten, die den Bewegungsraum der Maus begrenzen, gespeichert. Reihenfolge: X,Y,Breite, Höhe.

11. GRAON &732B 1

Flag, ob Grafikfenster aktiviert ist (1=aktiviert).

12. GRACOO &732C 4

Durch diese vier Bytes ist die Größe und Position des Grafikfensters spezifiziert. Bei den Coordinaten handelt es sich um Textkoordinaten (X 1-80, Y 1-25). Reihenfolge: X,Y,Breite,Höhe.

Es ist nicht angebracht irgendeine Daten durch POKE's zu ändern, sondern besser durch die dazu vorgesehenen Befehle.

7. Die Anwenderprogramme

Mit M.O.S. werden drei Anwenderprogramme ausgeliefert, ein Icondesigner, ein Zeichendesigner und ein Mousedesigner. Um die Bedienung dieser Programme geht es in diesem Kapitel. Dazu ist zu sagen, daß die Bedienung der Programme eigentlich durch Ausprobieren und Arbeiten mit dem Programm gelernt werden kann, da diese sehr benutzerfreundlich sind. Um keine Mißverständnisse aufkommen zu lassen, werden alle Funktionen der einzelnen Programme aber hier genau erklärt.

Die Programme sind grafisch sehr gut aufgemacht und brauchen sich nicht hinter Programmen auf dem Atari ST oder Commodore Amiga zu verstecken.

7.1 Warum die mitgelieferten Programme ?

In M.O.S. ist ja ein fester Satz Icons, ein fester Zeichensatz, der nicht durch SYMBOL geändert werden kann, und ein festes Maussymbol enthalten. Damit der Benutzer Icons, Zeichen und das Maussymbol umändern kann, geben wir ihm diese drei sehr komfortablen Anwendungsprogramme an die Hand.

So kann er sich seinen eigenen Zeichensatz, etc. erstellen. Er kann auch verschiedene Sätze erstellen, die dann bei Bedarf nachgeladen werden können.

Weiterhin sind die Programme einmal als Beispiel gedacht, was man unter M.O.S. machen kann.

7.2 Die Grundlagen der Anwenderprogramme

7.2.1 Das Laden der Programme

Alle drei Programme können nur über ein Menü geladen werden. Legen Sie die Systemdiskette ein und tippen Sie RUN "MOS".

Nach einer kurzen Zeit, werden Sie gefragt, welchen Rechner Sie besitzen. Drücken Sie dann bitte auf die entsprechende Taste (1 - CPC 464, 2 - CPC 664, 3 - CPC 6128). Der Rechner lädt und installiert die entsprechende M.O.S-Version und lädt ein Menüprogramm nach. Dort können Sie unter anderem die drei erwähnten Anwenderprogramme anwählen, indem Sie mit dem Mauszeiger den entsprechenden Menüpunkt anwählen und Feuer drücken. (Anmerkung: Der gerade angewählte Menüpunkt wird invers dargestellt)

Dann befinden Sie sich in Ihrem gewünschten Programm. Die Funktion der anderen Menüpunkte entnehmen Sie bitte Punkt 2.2.1.

7.2.2 Indentisches bei den Anwendungsprogrammen

Immer ist oben die sogenannte Menüleiste. Wenn Sie eines der Wörter anklicken, wird das entsprechende Menü heruntergeklappt. Sie können dann einen der Menüpunkte selektieren. Wünschen Sie das nicht, so können Sie entweder DELETE drücken oder den Closepoint anklicken.

Befinden Sie sich nicht in einem Menü hat die DELETE-Taste eine andere Funktion. Die Maus wird auf das erste Menü gesetzt und die Geschwindigkeit derselbigen wird erhöht. So kann man schnell auch die Menüs weiter rechts erreichen. Allerdings eignet sich die erhöhte Geschwindigkeit nicht mehr zum Verändern der Icons, Zeichen oder des Mauszeigers. Wurde ein Menü geöffnet, wird die Geschwindigkeit des Mauszeigers wieder auf den normalen Wert gesetzt.

Funktionen, die der Benutzer angewählt hat und die nicht wieder gutzumachende Schäden anrichten könnten, wie z.B. das Löschen aller Icons, hinterfragt der Rechner. Sie können entweder OK anklicken, um die Funktion ausführen zu lassen, oder QUIT um sie abzubrechen. Allerdings muß es sich dabei jeweils um einen Doppelklick (s. Anhang 1.7.3) handeln.

7.3 Der Icondesigner

Der Icondesigner dient zum Entwerfen neuer Icons, respektive zum Verändern der alten Icons.

Es gibt die 20 normalen Icons und ein aktives Icon. Nur dieses Icon können Sie verändern. Entspricht es Ihren Wünschen, so können Sie es in einem der 20 Icons ablegen. Das aktive Icon wird nämlich nicht mit auf Diskette abgespeichert.

7.3.1 Der Bildschirmaufbau

Oben befindet sich zunächst die Menüleiste. Durch Anklicken der entsprechenden Menüworte, wird das Menü geöffnet. Mehr dazu unter 7.3.4.

Links befindet sich eine Vergrößerung des aktiven Icons. In dieser Vergrößerung werden Sie auch später arbeiten, da man im Icon in Originalgröße nicht vernünftig arbeiten kann.

Damit Sie aber trotzdem noch wissen, wie das Icon in Wirklichkeit nachher aussieht, befindet sich unten rechts von der Vergrößerung die Darstellung des Icons in Originalgröße.

Darüber befinden sich vier Icons, die beim Anklicken diverse Funktionen haben. Mehr dazu unter 7.3.3.

Rechts befindet sich schließlich das Iconfile. Hier sind alle 20 Icons untergebracht. Bevor Sie nicht Icons geladen haben (s. 7.3.4.3) ist der Raum natürlich frei.

7.3.2 Das Verändern des aktiven Icons

Sie können das aktive Icon verändern, indem Sie auf die Vergrößerung des aktiven Icons fahren und die Feuertaste drücken. Dann wird der Punkt, über dem sich der Mauszeiger befindet, invertiert, gesetzt oder gelöscht. Das hängt vom Modus ab (s.7.3.4.4). Am Anfang wird aber immer invertiert.

Selbstverständlich können Sie das Icon noch durch diverse Menüpunkte und Anklicken von Icons verändern. Weiteres dazu siehe 7.3.3 und 7.3.4.

7.3.3 Die Funktion der Icons

Auf dem Bildschirm befinden sich vier anklickbare Icons: Die zwei Disketten, sie stehen für Laufwerk A und B, ein Drucker und ein Papierkorb.

Die beiden Disketten können Sie nur anklicken, wenn Sie im Menü DISKETTE sind, um das Laufwerk zu wählen (s. 7.3.4.3).

Wählen Sie das Druckericon an, so können Sie ein Icon aus dem Iconfile, nicht aber das aktive Icon, auf dem Drucker ausdrucken lassen. Der Rechner fordert Sie zunächst auf, sich ein Icon aus dem Iconfile auszusuchen. Drücken Sie DEL um abzuberechnen. Wollen Sie ausdrucken, dann fahren Sie mit dem Mauszeiger auf das Icon im Iconfile, welches Sie ausdrucken wollen, und drücken Sie Feuer. Der Rechner überprüft nun natürlich, ob der Drucker angeschlossen ist und ob er Online (druckbereit) ist. Ist das der Fall, so wird das Zeichen vergrößert ausgegeben. Dabei wird für ein gesetztes Bit ein 'X' und für ein zurückgesetztes Bit ein '.' ausgegeben.

Ansonsten wird die Meldung, daß der Drucker nicht empfangsbereit ist, ausgegeben. Sie können nun durch DEL oder durch Anklicken von QUIT den Druckversuch abbrechen oder, nachdem Sie Ihren Drucker betriebsbereit gemacht haben, OK anklicken, damit der Rechner erneut versuchen kann das Zeichen auszugeben.

Das letzte Icon, der Papierkorb, hat eine ganz radikale Funktion. Wird er angeklickt, so wird das gesamte Iconfile und das aktive Icon gelöscht. Man wirft seine ganzen Entwürfe sozusagen in den Papierkorb. Natürlich wird der Benutzer vom Programm darauf hingewiesen, daß die so gelöschten Icons unwiderruflich verloren sind und ob man diese wirklich löschen möchte. Durch DEL oder QUIT kann abgebrochen werden.

7.3.4 Die Menüs

Im Icondesigner befinden sich 7 Menüs mit insgesamt 17 Menüpunkten. Wollen Sie ein Menü anwählen, so fahren Sie auf das entsprechende Wort in der Menüleiste und drücken den Feuerknopf.

Dann wird das Menü sofort eröffnet und Sie können einen der Punkte auswählen. Dies tun Sie, indem Sie mit dem Mauszeiger auf den entsprechenden Punkt fahren und Feuer drücken.

Wollen Sie lieber doch keinen Menüpunkt auswählen, so drücken Sie entweder DEL oder klicken mit dem Mauszeiger den Closepoint (s. Anhang 1.8) des Fensters an. Es wird dann wieder geschlossen.

Die einzelnen Menüs mit Ihren Menüpunkten werden nun im folgenden vorgestellt.

7.3.4.1 Das Bearbeitungsmenü

Das Bearbeitungsmenü dient ausschließlich zum Verändern des aktiven Icons und umfaßt folgende sechs Menüpunkte:

a) INVERTIEREN

Bei Anwahl dieses Menüpunktes wird das aktive Icon invertiert, d.h. aus einem gesetzten Punkt wird ein nicht gesetzter und aus einem nicht gesetzten ein gesetzter.

b) SPIEGELN

Unter diesem Punkt wird das aktive Icon gespiegelt. Sie müssen allerdings noch wählen, ob an der horizontalen oder der vertikalen Achse des Icons gespiegelt werden soll. Sollten Sie kein Spiegeln wünschen, drücken Sie DEL.

c) Schieben

Nach Auswahl des Menüpunktes Schieben wird ein Fenster eröffnet, das Informationen enthält und darunter vier Icons mit Pfeilen in 4 Richtungen. Klicken Sie einen der Pfeile an, so wird das aktive Icon in die entsprechende Richtung geschoben. Dabei ist zu beachten, daß die Punkte, die auf der einen Seite herausfallen, nicht auf der anderen Seite wieder hereingeschoben werden, sondern verlorengehen. Dies ist aber Absicht, um so zu ermöglichen, daß man bequem äußere Spalten, bzw. Zeilen des Icons löschen kann. Haben Sie genug verschoben, drücken Sie DEL.

Es ist noch zu erwähnen, daß während des Verschiebens nur das Icon in Originalgröße verschoben wird, nicht die Vergrößerung, da das Informationsfenster darüberliegt. Die Vergrößerung wird erst aktualisiert, wenn DEL gedrückt wurde.

d) Löschen

Durch diese Funktion wird das aktive Icon gelöscht. Vor der Ausführung der Aktion wird diese natürlich hinterfragt. Um doch nicht zu löschen, drücken Sie DEL oder klicken Sie QUIT an.

e) Füllen

Diese Funktion ist der Funktion d) nicht unähnlich. Nur das nicht alle Bits zurückgesetzt, sondern gesetzt werden. Ansonsten ist die Bedienung identisch.

f) Rastern

Icon-, Zeichen- und Mousedesigner haben ja als Hintergrund ein Raster. Sollen Icons auf diesem Hintergrund dargestellt werden, so sieht es nicht sonderbar gut aus, wenn der Hintergrund des Icons völlig weiß ist. Wesentlich besser sieht es aus, wenn der Hintergrund des Icons genauso gerastert ist, wie der Bildschirmhintergrund. Dies ist ja auch bei den vier Icons (Diskettensymbole, Drucker, Papierkorb) auf dem Hintergrund der Fall.

Wollen Sie nun eines dieser vier Icons völlig umändern, so können Sie die Funktion RASTERN anwählen und erhalten ein gerastertes Hintergrund, in den Sie dann Ihr Icon einbetten können.

Da durch diese Funktion das aktive Icon zerstört wird, wird die Funktion natürlich hinterfragt.

7.3.4.2 Das Ablagemenü

Dieses Menü dient zum Übertragen des aktiven Icons in das Iconfile und umgekehrt. Es bietet zwei Menüpunkte:

a) Icon ablegen

Mit dieser Funktion ist es möglich das aktive Icon im Iconfile abzulegen. Zunächst müssen Sie mit dem Mauszeiger das Icon im Iconfile auswählen, in das das aktive Icon übertragen werden soll. Drücken Sie DEL, so wird die Funktion abgebrochen.

Da durch die Funktion ein Icon im Iconfile überschrieben wird, wird Sie hinterfragt.

b) Icon editieren

Mit Hilfe dieses Menüpunktes können Sie ein Icon aus dem Iconfile als aktives definieren. Zuerst wählen Sie das Icon aus dem Iconfile mit dem Mauszeiger aus. Durch Drücken von DEL wird die Funktion abgebrochen.

Da dadurch das aktive Icon zerstört wird, wird auch diese Funktion hinterfragt.

7.3.4.3 Das Diskettenmenü

Dieses Menü dient zum Speichern und zum Laden des Iconfiles.

a) Icons speichern

Wählen Sie diesen Menüpunkt an, so werden die Icons des Iconfiles auf Diskette abgelegt. Bitte geben Sie zunächst den Namen an, unter dem die Icons auf Diskette abgelegt werden sollen. Der Name darf höchstens acht Buchstaben umfassen und darf keine Extension beeinhalteln. Wird ein nicht zulässiger Name eingegeben, so wird nach Ausgabe einer entsprechenden Meldung die erneute Eingabe des Filenamens verlangt.

Anschließend müssen Sie wählen, auf welchem Laufwerk die Icons abgespeichert werden. Klicken Sie mit dem Mauszeiger das Laufwerk A oder B an (Icons). Drücken Sie DEL, so wird die Funktion abgebrochen. Es wird nun überprüft, ob die Diskette richtig eingelegt ist. Ist das nicht der Fall, oder ist die Diskette schreibgeschützt, so wird eine entsprechende Meldung ausgegeben.

Die Icons werden dann auf Diskette abgelegt. Dabei enthält der Name zur besseren Unterscheidung automatisch die Extension '.ICN'.

Sollte Ihr gewähltes Laufwerk überhaupt nicht angeschlossen sein, so passiert nichts. Sie haben dann keine andere Möglichkeit, als

den Rechner zurückzusetzen und den Icondesigner neu zu laden. Dabei gehen natürlich alle Icons verloren, also Vorsicht!

b) Icons laden

Durch diese Funktion können Sie Icons von Diskette laden. Da durch die Funktion sämtliche Icons zerstört werden (bis auf das aktive) wird die Aktion hinterfragt.

Danach wird wie schon unter 7.3.4.3 a) nach dem Filenamen gefragt. Auch hier ist keine Extension anzugeben, da diese wieder automatisch angehängt wird.

Danach ist das Laufwerk anzugeben und es wird versucht das entsprechende File zu laden. Auch hier muß die Diskette richtig eingelegt sein und darf nicht schreibgeschützt sein, sonst folgt eine entsprechende Meldung.

Ist das Laufwerk nicht angeschlossen oder befindet sich das File nicht auf Diskette, so müssen Sie den Icondesigner wie oben neu laden.

Beim erfolgreichen Laden werden die Icons anschließend im Iconfile dargestellt. Auf der Diskette befindet sich übrigens ein File mit den von M.O.S. verwandten Icons. Es ist unter dem Namen ICONDES.ICN abgelegt. Um dieses File zu laden ist also der Filename ICONDES anzugeben.

7.3.4.4 Das Modus-Menü

Hiermit wird festgelegt, ob beim Verändern eines Icons ein Punkt invertiert, gesetzt oder gelöscht werden soll. Außerdem hat dieser Modus auch noch Auswirkungen auf den Menüpunkt MERGE ICON im Utilitymenü (7.3.4.5).

a) Invertieren

Wird ein Punkt in der Vergrößerung des aktiven Icons angeklickt, dann wird er gesetzt, wenn er vorher nicht gesetzt war und zurückgesetzt, wenn er nicht gesetzt war.

b) Füllen

Ein angeklickter Punkt wird in jedem Fall gesetzt.

c) Löschen

Ein angeklickter Punkt wird in jedem Fall gelöscht (zurückgesetzt).

7.3.4.5 Das Utilitymenü

Durch diese Menü ist es möglich diverse Operationen mit den Icons im Iconfile durchzuführen.

a) Icon mergen

Das aktive Icon und eines vom Benutzer aus dem Iconfile auszuwählendes Icon werden auf bestimmte Art und Weise verknüpft und das Ergebnis wird wieder im aktiven File abgelegt.

Es gibt drei mögliche Verknüpfungsarten, die sich nach der gewählten Option im MODUS-Menü (7.3.4.4) richten:

1. Invertieren gewählt : Im Ergebnisicon wird ein Punkt gesetzt, wenn eines der beiden Operationsicons an der entsprechenden Stelle einen gesetzten Punkt hat. Sind an der Stelle die Punkte beider Icons gesetzt, wird der Punkt im Ergebnisicon nicht gesetzt (entspricht Verknüpfung durch XOR).
2. Füllen gewählt : Im Ergebnisicon wird nur dann kein Punkt gesetzt, wenn beide Icons an der entsprechenden Stelle keinen gesetzten Punkt haben (entspricht Verknüpfung durch OR).
3. Löschen gewählt : Identisch mit Invertieren gewählt.

b) Icon kopieren

Damit kann ein Icon aus dem Iconfile in ein anderes Icon des Iconfiles kopiert werden. Wählen Sie zuerst das Icon aus, welches Sie kopieren wollen, dann das Icon in welches es abgelegt werden soll. Während der Auswahl können Sie DEL drücken, um die Funktion abzubrechen. Die Aktion wird hinterfragt.

c) Icons tauschen

Mit dieser Funktion können zwei Icons aus dem Iconfile vertauscht werden. Wählen Sie die beiden Icons in bekannte Manier aus. Während der Auswahl können Sie DEL drücken, um die Funktion abzubrechen.

7.3.4.6 Das Abbruchmenü

Hier handelt es sich gar nicht um ein richtiges Menü, sondern eher um eine Funktion. Wählen Sie ABBRUCH an, so wird hinterfragt, ob der Icondesigner wirklich verlassen werden soll (dadurch werden alle Icons zerstört, die noch nicht auf Diskette gespeichert wurden). Klicken Sie OK durch Doppelklick an, so wird wieder das Menü zum Verzweigen in die verschiedenen Designer, etc. geladen.

7.3.5 Späteres Benutzen der Icons

Um die Icons später zu benutzen, müssen Sie diese zunächst einmal auf einer Diskette abspeichern. Wollen Sie diese nun in eigenen Programmen benutzen, so laden Sie im Programm nach dem Laden von M.O.S. die Icons (Extension ICN nicht vergessen!) und ab dann

stellt der Rechner die neuen Icons dar. Selbstverständlich können Sie im Laufe des Programmes wieder andere Icons nachladen.

Sie können sich aber auch eine neue M.O.S.-Version mit Ihren neuen Icons erstellen. Dazu ist folgende Befehlsfolge notwendig:

```
LOAD "MOSxxxx.BIN" - xxxx ist die M.O.S. Version (464,664,..)
LOAD "NAME.ICN" - Unter NAME stehen Ihre Icons auf Diskette
SAVE "MOSNEU.BIN",B,29400,10970
```

Diese neue MOS-Version können Sie dann direkt mit

```
LOAD "MOSNEU.BIN"
CALL 30000
```

laden und installieren. Sie können selbstverständlich auch einen anderen Namen wählen.

Wollen Sie die Icons auch in den Anwenderprogrammen verwenden, so müssen Sie zuerst auch wie oben beschrieben eine neue M.O.S.-Version erstellen, diese aber unter dem gleichen Namen wieder auf Ihre M.O.S.-Diskette abspeichern.

Sie müssen natürlich darauf achten, daß die Icons zumindest inhaltlich gleich bleiben. So ist es nicht sinnvoll das Diskettenicon plötzlich als Papierkorb zu definieren. Die Anwenderprogramme würden statt dem Diskettenicon immer den Papierkorb ausgeben.

7.4 Der Zeichendesigner

Der Zeichendesigner dient zum Entwerfen neuer Zeichen, respektive zum Verändern der alten Zeichen.

Es gibt ein aktives Zeichen. Nur dieses Zeichen können Sie verändern. Entspricht es Ihren Wünschen, so können Sie es im Zeichensatz ablegen. Das aktive Zeichen wird nämlich nicht mit auf Diskette abgespeichert.

7.4.1 Der Bildschirmaufbau

Oben befindet sich zunächst die Menüleiste. Durch Anklicken der entsprechenden Menüworte, wird das Menü geöffnet. Mehr dazu unter 7.4.4.

Links befindet sich eine Vergrößerung des aktiven Zeichens.

Unten rechts von der Vergrößerung ist das Zeichen in Originalgröße dargestellt.

Darüber befinden sich vier Icons, die beim Anklicken diverse Funktionen haben. Mehr dazu unter 7.4.3 und 7.3.3.

Rechts befindet sich schließlich Zeichensatz. Hier sind alle 128 Zeichen untergebracht. Bevor Sie keinen Zeichensatz geladen haben (s. 7.4.4.3) ist der Raum natürlich frei.

7.4.2 Das Verändern des aktiven Zeichens

Da dies absolut analog zum Icondesigner läuft, entnehmen Sie bitte alle Informationen von dort (7.3.2).

7.4.3 Die Funktion der Icons

Auf dem Bildschirm befinden sich vier anklickbare Icons: Die zwei Disketten, sie stehen für Laufwerk A und B, ein Drucker und ein Papierkorb.

Auch diese haben die gleiche Funktion wie im Icondesigner (s. 7.3.3).

7.4.4 Die Menüs

Im Icondesigner befinden sich 7 Menüs mit insgesamt 17 Menüpunkten. Wollen Sie ein Menü anwählen, so fahren Sie auf das entsprechende Wort in der Menüleiste und drücken den Feuerknopf.

Dann wird das Menü sofort eröffnet und Sie können einen der Punkte auswählen. Dies tun Sie, indem Sie mit dem Mauszeiger auf den entsprechenden Punkt fahren und Feuer drücken.

Wollen Sie lieber doch keinen Menüpunkt auswählen, so drücken Sie entweder DEL oder klicken mit dem Mauszeiger den Closepoint (s. Anhang 1.8) des Fensters an. Es wird dann wieder geschlossen.

Die einzelnen Menüs mit Ihren Menüpunkten werden nun im folgenden vorgestellt.

7.4.4.1 Das Bearbeitungsmenü

Das Bearbeitungsmenü dient ausschließlich zum Verändern des aktiven Zeichens und umfaßt folgende sechs Menüpunkte:

Die Funktionen Invertieren, Spiegeln, Löschen, Füllen und Schieben laufen genauso wie auf dem Icondesigner. Die Funktion Rastern entfällt. An ihre Stelle tritt die Funktion DREHEN:

Nach Aufruf dieser Funktion wird das aktive Zeichen um 90 Grad im Uhrzeigersinn gedreht.

7.4.4.2 Das Ablagemenü

Die Bedienung ist mit der des Icondesigners identisch, nur das Zeichen im Zeichensatz abgelegt werden können und nicht Icons im Iconfile.

7.4.4.3 Das Diskettenmenü

s. 7.3.4.3, nur werden die Zeichen unter der Extension '.CHR' abgespeichert. Das Demofile hat den Namen ZEICHEN.CHR.

7.4.4.4 Das Modus-Menü

Identisch mit Icondesigner.

7.4.4.5 Das Utilitiymenü

Identisch mit Icondesigner.

7.4.4.6 Das Abbruchmenü

Identisch mit Icondesigner.

7.4.5 Späteres Benutzen des Zeichensatzes

Um die Zeichen später zu benutzen, müssen Sie diese zunächst einmal auf einer Diskette abspeichern. Wollen Sie diese nun in eigenen Programmen benutzen, so laden Sie im Programm nach dem Laden von M.O.S. den Zeichensatz (Extension CHR nicht vergessen!) und ab dann stellt der Rechner die neuen Zeichen dar. Selbstverständlich können Sie im Laufe des Programmes wieder einen anderen Zeichensatz nachladen.

Sie können sich aber auch eine neue M.O.S.-Version mit Ihrem neuen Zeichensatz erstellen. Dazu ist folgende Befehlsfolge notwendig:

```
LOAD "MOSxxxx.BIN" - xxxx ist die M.O.S. Version (464,664,..)
LOAD "NAME.CHR" - Unter NAME stehen Ihre Zeichen auf Diskette
SAVE "MOSNEU.BIN",B,29400,10970
```

Diese neue MOS-Version können Sie dann direkt mit

```
LOAD "MOSNEU.BIN"
CALL 30000
```

laden und installieren. Sie können selbstverständlich auch einen anderen Namen wählen.

Wollen Sie die Zeichen auch in den Anwenderprogrammen verwenden, so müssen Sie zuerst auch wie oben beschrieben eine neue M.O.S.-Version erstellen, diese aber unter dem gleichen Namen wieder auf Ihre M.O.S.-Diskette abspeichern.

Sie müssen natürlich darauf achten, daß die Zeichen zumindest inhaltlich gleich bleiben. So ist es nicht sinnvoll das Zeichen "A" als "B" zu definieren.

7.5 Der Mousedesigner

Der Mousedesigner dient zum Entwerfen eines neuen Maussymbols, respektive zum Verändern des alten Mousesymbols.

7.5.1 Der Bildschirmaufbau

Oben befindet sich zunächst die Menüleiste. Durch Anklicken der entsprechenden Menüworte, wird das Menü geöffnet. Mehr dazu unter 7.5.4.

Links befindet sich eine Vergrößerung des Mousesymbols.

Unten rechts von der Vergrößerung ist das Mousesymbol in Originalgröße dargestellt.

Darüber befinden sich vier Icons, die beim Anklicken diverse Funktionen haben. Mehr dazu unter 7.5.3 und 7.3.3.

7.5.2 Das Verändern des Mousesymbols

s. 7.3.2

7.5.3 Die Funktion der Icons

Auf dem Bildschirm befinden sich vier anklickbare Icons: Die zwei Disketten, sie stehen für Laufwerk A und B, ein Drucker und ein Papierkorb.

Auch diese haben die gleiche Funktion wie im Icondesigner (s. 7.3.3).

7.5.4 Die Menüs

Im Mousedesigner befinden sich 5 Menüs mit insgesamt 12 Menüpunkten. Wollen Sie ein Menü auswählen, so fahren Sie auf das entsprechende Wort in der Menüleiste und drücken den Feuerknopf.

Dann wird das Menü sofort eröffnet und Sie können einen der Punkte auswählen. Dies tun Sie, indem Sie mit dem Mauszeiger auf den entsprechenden Punkt fahren und Feuer drücken.

Wollen Sie lieber doch keinen Menüpunkt auswählen, so drücken Sie entweder DEL oder klicken mit dem Mauszeiger den Closepoint (s. Anhang 1.8) des Fensters an. Es wird dann wieder geschlossen.

Die einzelnen Menüs mit Ihren Menüpunkten werden nun im folgenden vorgestellt.

7.5.4.1 Das Bearbeitungsmenü

Identisch mit Zeichendesigner.

7.5.4.2 Das Diskettenmenü

s. 7.3.4.3, nur werden die Zeichen unter der Extension '.MOU' abgespeichert. Das Demofile hat den Namen MAUS.MOU.

7.5.4.3 Das Modus-Menü

Identisch mit Icondesigner.

7.5.4.4 Das Abbruchmenü

Identisch mit Icondesigner.

7.5.5 Späteres Benutzen des Mousesymbols

Das Benutzen des Mousesymbols läuft analog zum Benutzen des Zeichensatzes und der Icons.

8. Das Grafikprogramm

Zum M.O.S. System wird ein Grafikprogramm ausgeliefert, das in diesem Kapitel vorgestellt und erklärt werden soll.

Es arbeitet in allen drei Bildschirmmodi.

8.1 Laden des Grafikprogrammes

Laden Sie das Programm mit RUN "PAINT" von der M.O.S.-Diskette.

8.2 Funktionen auf dem Grafikbildschirm

Nach dem Laden und Starten des Programmes gelangen Sie in den Grafikbildschirm. Dort sehen Sie sofort zwei Cursor:

Einen kleinen, der wie ein x aussieht, wir nennen ihn hier BASIS und einen großen, der aus zwei sich kreuzenden Linien besteht, die an allen 4 Enden auf den Bildschirmrand stoßen. Dieser Cursor ist der Grafikcursor.

Mit den Cursortasten können Sie nun den Grafikcursor bewegen. Mit ENTER setzen Sie die Basis auf den Grafikcursor.

Drücken Sie COPY wird die gerade im Hauptmenü ausgewählte Betriebsart ausgeführt (dazu später).

Mit SPACE gelangen Sie ins Hauptmenü.

Eine weitere und die letzte belegte Taste ist TAB. Manche Befehle arbeiten mit Bereichen. Der ganze Bildschirm ist unter M.O.S.-Paint in feste Bereiche unterteilt. Drücken Sie TAB, so wird der Bereich, in dem sich der Grafikcursor befindet, umrandet. Nach dem Druck einer Taste verschwindet dieser Rahmen wieder und Sie können ganz normal weiterarbeiten.

8.3 Das Hauptmenü

In das Hauptmenü gelangen Sie vom Grafikbildschirm durch den Druck von SPACE, der Leertaste.

Der gerade angewählte Menüpunkt ist invers dargestellt. Sie können den inversen Balken mit den Cursortasten bewegen und durch den Druck auf Copy wird die entsprechende Funktion ausgeführt.

Das Hauptmenü ist in zwei Bereiche unterteilt, die im folgenden dargestellt werden:

8.3.1 Die Betriebsarten

Wählen Sie eine Betriebsart an, so passiert nichts weiter, es wird in den Grafikbildschirm geschaltet. Entsprechend des angewählten Betriebsmodus wird dann beim Druck von COPY eine Funktion ausgeführt. Ein Betriebsmodus paßte leider nicht mehr unter die Betriebsmodi und mußte so ins Fenster der Funktionen ausweichen, der Betriebsmodus ZOOMEN.

a) PLOT

Beim Druck auf Copy wird ein Punkt auf die Position des Grafikcursors gesetzt.

b) DRAW

Zwischen der Basis und dem Grafikcursor wird eine Linie gezogen. Anschließend wird die Basis auf den Grafikcursor gesetzt.

c) RECHTECK

Basis und Grafikcursor bilden jeweils den Eckpunkt eines Rechtecks, welches durch diese Funktion gezeichnet wird.

d) KASTEN

Im Prinzip wie RECHTECK, nur daß dieses auch gleich ausgefüllt wird.

f) 3D - KASTEN

Zuerst muß wie bei Rechteck die Frontfläche des Kastens festgelegt werden. Durch weiteres Positionieren des Grafikcursors und anschließenden Druck auf COPY wird die 'Tiefe' des Kastens festgelegt.

g) KREIS/ELIPSE

Die Basis bildet den Mittelpunkt des Kreises, bzw. der Elipse. Die waagerechte Linie des Grafikcursors gibt den höchsten, bzw. tiefsten Punkt, die senkrechte Linie den Punkt ganz links, bzw. den Punkt ganz rechts des Kreises, bzw. der Elipse an.

h) SCHEIBE

Im Prinzip genauso wie KREIS/ELIPSE, nur das die Kreis/Elipsenfläche gleich ausgefüllt wird.

i) TEXT AUSGEBEN

An der Position des Grafikcursors wird ein vorher eingegebener Text (siehe Funktionen, 8.3.2) ausgegeben.

j) FILL

Vom Grafikcursor aus wird der Bildschirm ausgefüllt.

k) SPRAY

Der Bereich zwischen der Basis und dem Grafikcursor wird leicht

'gespränkelt'. Durch mehrmaligen Druck von COPY ist eine höhere Dichte der Punkte möglich.

l) LÖSCHEN

Der Bereich zwischen Basis und Grafikkursor wird gelöscht.

m) BEREICH LÖSCHEN

Der Bereich, in dem sich der Grafikkursor befindet (zu ermitteln per TAB) wird gelöscht.

n) BEREICH INVERTIEREN

Der Bereich, in dem sich der Grafikkursor befindet, wird invertiert.

o) BEREICHE TAUSCHEN

Der Bereich, in dem sich der Grafikkursor befindet und der Bereich in dem sich die Basis befindet, werden ausgetauscht.

p) BEREICHE KOPIEREN

Der Bereich, in dem sich die Basis befindet, wird in den Bereich, in dem sich der Grafikkursor befindet, kopiert.

q) TEXT GROSS AUSGEBEN

Hierbei handelt es sich um eine sehr leistungsfähige Funktion. Ein eingegebener Text kann in jeder Größe und in vier verschiedenen Arten ausgegeben werden.

Die Größe wird folgendermaßen bestimmt. Die waagerechte Ausdehnung eines Buchstabens entspricht dem waagerechten Abstand zwischen Basis und Grafikkursor und die senkrechte Ausdehnung eines Buchstabens entspricht dem senkrechten Abstand zwischen Basis und Grafikkursor.

In welcher Art die Schrift nun ausgegeben wird, hängt von der Stellung von Basis und Grafikkursor (GC) ab:

1. GC steht links oberhalb der BASIS:

Der Text wird normal von links nach rechts ausgegeben.

2. GC steht links unterhalb der BASIS:

Der Text wird von links nach rechts ausgegeben, allerdings auf dem Kopf.

3. GC steht rechts oberhalb der BASIS:

Der Text wird von rechts nach links ausgegeben.

4. GC steht rechts unterhalb der BASIS:

Der Text wird von rechts nach links und auf dem Kopf ausgegeben.

r) ZOOMEN

Zuerst wird mit dem Grafikcursor ausgewählt, welcher Bereich vergrößert dargestellt werden soll. Durch TAB kann man den aktuellen Bereich, in dem sich der Grafikcursor befindet, erfahren.

Wird COPY gedrückt, werden die PLOT-Punkte dieses Bereichs vergrößert dargestellt. Mit den Cursortasten können Sie dann einen Cursor bewegen, mit dessen Hilfe man Punkte setzen (COPY) oder löschen (DEL) kann.

Um aus dem Vergrößerungsmodus wieder in den normalen Grafikbildschirm zu kommen, ist ein zweimaliger (!) Druck von ENTER notwendig.

8.3.2 Die Funktionen

Wählen Sie eine Funktion, so wird im Allgemeinen über das Menü ein Fenster geklappt. Hier haben Sie dann die Möglichkeit diverse Parameter, wie Farben, etc. einzustellen. Anschließend wird wieder in den Grafikbildschirm geschaltet, wobei der momentane Betriebsmodus natürlich beibehalten wird. Grundsätzlich gilt für fast alle Funktionen, die eine Zahleneingabe haben: Mit der Cursortaste links wird die Zahl erniedrigt, mit der Cursortaste rechts wird die Zahl erhöht und mit COPY übernommen. Die Funktionen werden nun vorgestellt.

a) PENWAHL

Mit dieser Funktion wird die aktuelle Schreib- und Malfarbe festgelegt.

b) INKWahl

Zuerst muß die Farbe, die verändert werden soll mit den Cursortasten und COPY festgelegt werden, dann muß der neue Wert der Farbe dem Rechner in bekannter Weise übermittelt werden. Zur Hilfe wird eine Farbtabelle mit den Nummern der verschiedenen Farben (z.B. 0 SCHWARZ, ... 26 Leuchtendweiß) eingeblendet. Dies ist bei dem Befehl BORDER auch der Fall.

c) BORDER

Der gewünschte Border soll ausgewählt werden.

d) MODE

Der gewünschte Bildschirmmodus soll ausgewählt werden. Dabei ist allerdings zu beachten, daß der Grafikbildschirm durch diese

Funktion gelöscht wird.

e) MODUSWAHL

Mit dieser Funktion kann zwischen verschiedenen Arten der Grafikerstellung gewählt werden. (Normal, XOR, OR AND).

f) ZEICHENDESIGNER

Zuerst wählen Sie aus dem Zeichensatz mit den Cursortasten und COPY das Zeichen aus, welches Sie verändern wollen und geben dann mit Hilfe von Nullen und Einsen das neue Muster des Zeichens ein. Dabei steht die Eins für ein gesetztes Bit.

g) SCHRITTWEITE

Mit dieser Funktion können Sie die X- und Y-Schrittweite des Grafikkursors bestimmen. Sowohl X- als auch Y-Schrittweite sind am Anfang auf 4 gesetzt. Zuerst müssen Sie mit den Cursortasten und COPY zum Abschluß die X-Schrittweite wählen, dann die Y-Schrittweite.

h) TEXT EINGEBEN (1)

Da unter zwei Betriebsmodi Texte ausgegeben werden können, muß dieser ja auch irgendwie eingegeben werden. Dies geschieht durch diese oder folgende Funktion. Der gesamte Zeichensatz wird dargestellt und mit den Cursortasten können Sie ein Zeichen auswählen, um es durch COPY zu übernehmen. Durch DEL wird das letzte Zeichen gelöscht, durch ENTER die Eingabe abgeschlossen.

i) TEXT EINGEBEN (2)

Diese Texteingabe ist zwar schneller als obige, aber es können auch nicht alle Zeichen eingegeben werden. Es handelt sich nämlich hier um einen ganz normalen INPUT-Befehl.

j) SAVE

Mit dieser Funktion ist es möglich das gesamte Bild oder einen Teil des Bildes auf Diskette abzuspeichern.

Geben Sie zuerst den Namen an, unter dem der Bildschirm abgelegt werden soll. Dabei darf keine Extension angegeben werden. Anschließend fragt der Rechner, ob das gesamte Bild, oder ein Teil des Bildes, abgespeichert werden soll:

1. Das gesamte Bild

Das File erhält die Extension GSB (für GeSamtBild). Dann wird das ganze Bild auf Diskette ausgelagert. Dabei werden die Farben ebenfalls übertragen, sodaß bei einem späteren Einladen des Bildes auch die richtigen Farben aktiviert sind.

2. Das Teilbild

Das File erhält die Extension TLB (für TeilBild). Der Bereich zwischen Grafikcursor und Basis wird nun auf Diskette ausgelagert, Sie müssen also vorher die beiden Cursor richtig setzen. Die Farben werden nicht mit abgespeichert.

k) LOAD

Man kann natürlich auch Gesamt- und Teilbilder laden. Zuerst ist wieder der Name des Bildes anzugeben. Eine Extension ist wieder nicht erlaubt, da der Rechner diese automatisch anhängt.

Man wähle also wieder zwischen Gesamt- und Teilbild, daß entsprechende File wird von Diskette geladen und in den Grafikbildschirm gesprungen.

l) EXIT

Wird diese Funktion angewählt, fragt der CPC noch einmal nach, ob das Programm auch wirklich beendet werden soll. Drücken Sie bitte "J", wenn Sie das Programm verlassen wollen.

9. Ein Programmbeispiel unter M.O.S.

In diesem Kapitel wird ein unter M.O.S. geschriebenes Basicprogramm abgedruckt. Sie können es zur Anregung für eigene Programme verwenden.

Was soll das Programm können?

Nun, oben auf dem Bildschirm soll sich eine Kopfzeile befinden. Darin eingeblendet zwei Menüs mit den Namen Menue 1 und Menue 2. Ferner weiter rechts ein Menü mit Namen Abbruch. Rechts in der Mitte befindet sich ein Fenster.

Der Benutzer kann mit der Maus nun die drei Fenster in der Kopfzeile anklicken. Wird Abbruch angeklickt, kehrt das Programm in den READY-Modus zurück.

Wird eines der Menüfenster angeklickt, wird das entsprechende Menü eröffnet. Menü 1 soll 4 Menüpunkte und Menü 2 soll 2 Menüpunkte umfassen. Der Benutzer kann sich einen der Punkte aussuchen, den Closepoint anklicken, oder DEL drücken und im großen Fenster wird angeklicktes Menü, Art des Anklickens oder die Punktnummer angegeben.

Im folgenden ist das Programm abgedruckt, vielleicht tippen Sie es einmal ein. Die Erläuterungen zu den einzelnen Zeilen folgen nach dem Programm.

```
1000 '
1010 ' DEMO-PROGRAMM unter M.O.S.
1020 '
1030 ' Variableninitialisierung
1040 '
1050 DEF FN klkart=PEEK(&7301)
1060 DEF FN klkwin=PEEK(&7300)
1070 DEF FN klkmen=PEEK(&74D7)
1080 DEF FN artmen=PEEK(&74D6)
1090 '
1100 ' M.O.S. laden und initialisieren
1110 '
1120 MEMORY 26999
1130 LOAD "mosxxxx":CALL 3000
1140 '
1150 ' Bildschirmaufbau
1160 '
1170 CLS
1180 FOR i=0 to 3:MOVE 0,396-i*4:DRAWR 640,0:NEXT
1190 !WOPEN,1,3,1,9,1:!WOPEN,2,14,1,9,1
1200 w$=" Menue 1;":!WPRINT,1,%w$
1210 w$=" Menue 2;":!WPRINT,2,%w$
1220 !WOPEN,3,70,1,9,1
1230 w$=" Abbruch;":!WPRINT,3,%w$
1240 a$="kr":b$="Wahlanzeige":!WOPEN,20,50,10,20,5,%a$,%b$
1250 !WCLICK,1,2,3:!LOCATEMOUSE,1,3
1260 '
1270 ' Mausabfrage
1280 '
1290 !WAITMOUSE
1300 IF FN klkart=255 THEN w$="DEL gedrueckt.":GOTO 1640
1310 IF FN klkwin=3 THEN !MOSOFF:WHILE INKEY$<>"":WEND:MODE 2:
```

```

END
1320 IF FN klkwin=2 THEN 1520
1330 '
1340 ' Menue 1 gewaehlt
1350 '
1360 a$="krc":b$="Menue 1":!MOPEN,3,2,16,13,%a$,%b$
1370 a$="Bitte waeh- len":!MPRINT,%a$:!MPRINT,%ww$
1380 a$="Punkt 1":!MPRINT,%a$:!MPRINT,%ww$
1390 a$="Punkt 2":!MPRINT,%a$:!MPRINT,%ww$
1400 a$="Punkt 3":!MPRINT,%a$:!MPRINT,%ww$
1410 a$="Punkt 4":!MPRINT,%a$
1420 !MCHOICE,6,8,10,12
1430 IF FN artmen=255 THEN !MCLOSE:w$="Menue 1          DEL gedruec
kt;":GOTO 1640
1440 IF FN klkmen=255 THEN w$="Menue 1          Closepoint;":GOTO
1640
1450 !MCLOSE:IF FN klkmen=1 THEN w$="Menue 1          Punkt 1;":GO
TO 1640
1460 IF FN klkmen=2 THEN w$="Menue 1          Punkt 2;":GOTO 1640
1470 IF FN klkmen=3 THEN w$="Menue 1          Punkt 3;":GOTO 1640
1480 IF FN klkmen=4 THEN w$="Menue 1          Punkt 4;":GOTO 1640
1490 '
1500 ' Menue 2 gewaehlt
1510 '
1520 a$="krc":b$="Menue 2":!MOPEN,15,2,16,9,%a$,%b$
1530 a$="Bitte waeh- len":!MPRINT,%a$:!MPRINT,%ww$
1540 a$="Punkt 1":!MPRINT,%a$:!MPRINT,%ww$
1550 a$="Punkt 2;":!MPRINT,%a$
1560 !MCHOICE,6,8
1570 IF FN artmen=255 THEN !MCLOSE:w$="Menue 2          DEL gedruc
kt;":GOTO 1640
1580 IF FN klkmen=255 THEN w$="Menue 2          Closepoint;":GOTO
1640
1590 !MCLOSE:IF FN klkmen=1 THEN w$="Menue 2          Punkt 1;":GO
TO 1640
1600 w$="Menue 2          Punkt 2;":GOTO 1640
1610 '
1620 ' Ausgabe im Wahlanzeige-Fenster
1630 '
1640 !WCLS,0,20:!WPRINT,20,%w$
1650 GOTO 1290

```

Erläuterungen zum Demoprogramm:

Zeile	Erläuterungen
1050	Funktion für Klickart im Fenster
1060	Funktion für angeklicktes Fenster
1070	Funktion für Klickart im Menü
1080	Funktion für angeklickten Menüpunkt
1120-1130	M.O.S. laden und initialisieren Für xxxx Version (464/664/6128) eintragen
1170-1180	Kopflinie malen
1190-1210	Fenster für Menü 1 und Menü 2 öffnen
1220-1230	Fenster für Abbruch öffnen
1240	Grosses Fenster öffnen
1250	Fenster für Menü 1 und 2 und Abbruchfenster als an- klickbar definieren, Maus positionieren
1290	Mausabfrage
1300	Benutzer hat DEL gedrückt

1310 Abbruchfenster angeklickt ==> Programmende
1320 Verzweigung zu Menü 2
1360-1410 Aufbau Menü 1
1420 Abfrage Menü 1
1430 DEL gedrückt
1440 Closepoint angeklickt
1450-1480 Menüpunkt 1,2,3 oder 4 angeklickt
1520-1550 Aufbau Menü 2
1560 Abfrage Menü 2
1570 DEL gedrückt
1580 Closepoint angeklickt
1590-1600 Menüpunkt 1 oder 2 angeklickt
1640-1650 Unterprogramm: Ausgabe von Text im grossen Fenster

ANHANG

1. Begriffserklärungen

In keiner Anleitung kommt man ohne Fachausdrücke aus. Dies ist auch in dieser Anleitung nicht anders. Aber um den Benutzer, der einen häufig gebrauchten Begriff einfach nicht unterzubringen weiß, nicht im Regen stehen zu lassen, bieten wir ihm hier die Erklärung der wichtigsten Fachwörter im Text an.

1.1 Benutzerfreundliche Oberfläche/Desktop

Unter einer benutzerfreundlichen Oberfläche versteht man ein Programm respektive ein Betriebssystem, daß die Verbindung zwischen Rechner und Bediener möglichst leicht macht.

Dieses Programm fängt im Allgemeinen alle nur erdenklichen Fehler des Benutzers ab, bzw. hinterfragt gefährliche Aktionen. Dies ist wird durch mehrere überlappende Fenster und Icons (s.u.) dem Benutzer recht nahe gebracht, was den Vorteil hat, daß man innerhalb kürzester Zeit mit dem Programm umgehen kann.

Statt z.B. unter CP/M irgendwelche Befehle einzutippen, wird unter benutzerfreundlichen Oberflächen einfach ein entsprechendes Symbol mit einem kleinen elektronischen Mauszeiger (s.u.) angeklickt (s.u.).

Das eigentliche Betriebssystem verbirgt sich sozusagen unter dieser, grafisch sehr gut aufbereiteten, "Oberfläche".

Ein Desktop ist ein Programm, welches diese Oberfläche benutzt. Alles was beim Schneider z.B. über CP/M gemacht wird, wird dabei über das Desktop gemacht. Es beinhaltet eine komplette Diskettenverwaltung mit Formatieren, Kopieren, Inhaltsverzeichnis, etc. Ferner ist es möglich die Farben, die Geschwindigkeit des Mauszeigers und anderes einzustellen.

1.2 Fenster

Jeder CPC-Benutzer wird vermutlich Fenster kennen. Schließlich stellt das CPC-Basic schon acht Fenster zur Verfügung. Wenn wir hier allerdings von Fenstern sprechen, meinen wir nicht die Standart-Basic-Fenster. Im Prinzip wird bei diesen ja nur der beschreibbare Teil des Bildschirms eingeschränkt.

Wird eines der M.O.S.-Fenster geöffnet, so wird der alte Bildschirminhalt zwischengespeichert. Es können bis zu 20 Fenster geöffnet werden und in jedes der geöffneten kann geschrieben werden. Die Fenster können zusätzlich, der besseren Absetzung vom restlichen Bildschirm wegen, mit diversen Verzierungen ausgestattet werden, wie Rahmen, Kopfzeile, etc. Wird ein Fenster geschlossen, wird der alte Bildschirminhalt rekonstruiert und wieder dargestellt, wobei natürlich andere Fenster, die das geschlossene überlappen, beachtet werden.

Manchmal wird statt dem Ausdruck Fenster auch der gebräuchliche englische Ausdruck Window(s) verwandt.

1.3 Menüs

Menüs sind den Fenstern nicht unähnlich. Bis zu 5 können gleichzeitig eröffnet werden. Sie werden fast genauso wie die Fenster programmiert. Nur das die Möglichkeit besteht, einige Zeilen aus dem Menü als Menüpunkte zu definieren und dann mit Hilfe eines Mauszeigers einen der Menüpunkte auswählen zu lassen.

1.4 Icons

Icons sind kleine grafische Symbole, sogenannte Pictogramme. Oft durch Icons dargestellte Gegenstände sind Diskettenlaufwerke, Papierkörbe, Stopzeichen, oder ähnliches. Wird zum Beispiel vom Benutzer ein Icon, welches ein Diskettenlaufwerk darstellt, angeklickt, so wird auf vielen Rechnern das Inhaltsverzeichnis dieses Diskettenlaufwerkes eingeblendet.

1.5 Maus

Von einer Maus werden Sie sicher auch schon in Verbindung mit Computern gehört haben. Es handelt sich um ein kleines Gerät mit zwei oder drei Tasten, und einer im Gehäuse unten angebrachten Kugel. Durch Bewegen der Maus auf einer glatten Fläche kann man den Mauszeiger auf dem Bildschirm bewegen, durch Drücken einer Taste der Maus ein Fenster oder Icon anklicken.

1.6 Mauszeiger, Maussymbol

Es handelt sich dabei um einen kleinen elektronischen Pfeil, den man mit Hilfe einer Maus, oder eines Joysticks, auf dem Bildschirm bewegen kann.

1.7 Anklicken

Es wurde sooft von Anklicken gesprochen, ohne klarzumachen, was es überhaupt bedeutet. Wenn man mit dem Mauszeiger auf ein Fenster fährt und eine bestimmte Taste der Maus, bzw. des Joysticks drückt, so klickt man das Fenster an. D.h. man wählt es an. Wie das Programm darauf reagiert, kommt natürlich ganz auf die Funktion des Fensters oder Icons an. Ich erinnere an das Beispiel des Diskettenlaufwerkes (Anhang 1.4) um ein Inhaltsverzeichnis der Diskette zu erhalten.

Selbstverständlich kann es ein Programm verbieten, bestimmte Fenster oder Icons anzuklicken.

Es gibt jetzt noch fünf verschiedene Klickarten. Je nachdem, wie man die Taste des Joysticks drückt, kann dies eine verschiedene Funktion haben. Da M.O.S. mit Joystick betrieben wird, sprechen wir in Zukunft nur noch vom Joystick. Dessen Feuerknopf dient als "Anklickknopf".

Die verschiedenen Anklickarten werden nun im folgenden vorgestellt:

1.7.1 Tastendruck

Dies ist der Fall, wenn der Benutzer den Feuerknopf des Joysticks betätigt. Egal wie lange, wie oft, es handelt sich um einen Tastendruck. Es wird also nicht mehr zwischen KCLICK, DOPPELKCLICK und PRESS unterschieden, sondern sofort abgebrochen. Allerdings kann der Tastendruck nur zugelassen werden, wenn KCLICK, DOPPELKCLICK UND PRESS verboten werden. Mehr dazu beim Befehl #WAITMOUSE.

1.7.2 Klick

Bei einem Klick darf der Feuerknopf höchstens 0.25 Sekunden gedrückt gehalten werden. Dann darf mindestens 0.25 Sekunden der Knopf nicht mehr gedrückt werden, falls Doppelklick eingeschaltet ist (siehe auch Doppelklick). Ist ein Doppelklick jedoch nicht zugelassen, so wird ein Klick auch akzeptiert, wenn der Knopf kürzer als 0.25 Sekunden gedrückt wird und danach auch innerhalb von 0.25 Sekunden oder mehr erneut gedrückt wird.

Dies hört sich recht hektisch an, ist aber nach sehr kurzer Eingewöhnung kein Problem. Klick steht also für einen kurzen Druck des Feuerknopfes.

1.7.3 Doppelklick

Ein Doppelklick wird akzeptiert, wenn der Feuerknopf des Joysticks zunächst für weniger als 0.25 Sekunden gedrückt gehalten wird und nach einer Pause von weniger als 0.25 erneut gedrückt wird. Dabei ist es egal, wie lange der zweite Druck der Taste ist.

Ein Doppelklick ist also ein zweimaliges kurzes Drücken des Feuerknopfes mit kurzer Pause.

1.7.4 Press

Bei Press muß der Feuerknopf länger als 0.25 Sekunden betätigt werden. Press steht damit also für einen langen Tastendruck.

1.7.5 Cancel

Cancel hat nun mit dem Feuerknopf nichts zu tun. Bei einer Maus gibt es noch eine weitere Taste, deren Druck einen Abbruch einer Funktion bedeutet. Da aber der Joystick nur eine Taste hat, bzw. zwei parallel geschaltete Tasten, müssen wir auf die Tastatur ausweichen. Dabei wird bei Druck auf die 'DEL'-Taste ein Cancel ausgeführt.

Zur Veranschaulichung ein Beispiel. Wird ein Menü vom Benutzer angeklückt, so wird dieses geöffnet und der Benutzer kann einen der Menüpunkte auswählen. Hat er sich vertan, so kann er auch 'DEL' drücken, um keinen Menüpunkt anzuwählen, die Funktion also abzubrechen.

1.8 Kopfzeile

Menüs und Fenster können eine Kopfzeile (engl. Headline) haben. Diese besteht aus vier waagerechten Linien und ist ganz oben im Fenster, bzw. Menü angeordnet.

In diese Kopfzeile wird im Allgemeinen die Fenster, bzw. Menü-überschrift eingeblendet. Dies ist möglich, aber nicht zwingend vorgeschrieben. Allerdings kann man nur in ein Fenster/Menü mit Kopfzeile eine Überschrift, die gleich beim Eröffnen des Fensters übergeben werden muß, einblenden. Die Überschrift wird dabei automatisch zentriert.

Ferner kann in die Kopfzeile ein Closepoint (deutsch: Schließpunkt) eingeblendet werden.

1.9 Closepoint

Hat der Benutzer ein Menü oder Fenster eröffnet, welches einen Closepoint besitzt, so kann er das Fenster/Menü entweder durch einen Druck durch 'DEL' (siehe Anhang 1.7.5) schließen oder indem er mit dem Mauszeiger auf den Closepoint des Menüs fährt und eine Taste drückt. Das kann sich dann als praktisch erweisen, wenn der Benutzer keine Lust hat eine Hand vom Joystick zu nehmen, um 'DEL' zu drücken. Der Closepoint befindet sich immer links in der Kopfzeile und stellt eine Aussparung der Kopfzeile dar.

2.0 Programmierhinweise

An dieser Stelle werden dem Benutzer noch zwei wichtige Programmierhinweise gegeben:

2.1 Programmierung einer Menüleiste

Sicher möchte der Benutzer, wie schon in den Anwenderprogrammen, eine Menüleiste programmieren.

Dazu muß man zuerst mit DRAW-Befehlen die vier Linien erzeugen. Dann muß für jedes Menü ein Fenster eröffnet werden, in das die Überschrift des Menüs eingetragen wird.

Wird das Fenster dann angeklickt, so ist dies vom Benutzer abprüfbar und dann kann beispielsweise eine Bildschirmposition unter der Menüleiste das entsprechende Menü eröffnet werden.

2.2 Anklicken von Icons

Man möchte in Programmen sicher auch mal Icons anklicken können, z.B. wie in den Anwenderprogrammen den Papierkorb. Icons lassen sich aber nicht direkt anklicken.

Folgende Methode schafft Abhilfe:

Man definiert ein vier mal zwei Bildschirmpositionen grosses Fenster ohne Rahmen, etc. und setzt es an die Stelle, an der nachher das Icon stehen soll.

Anschließend setzt man das Icon in das Fenster. Wird nun vom Benutzer das Fenster angeklickt, so kann man dies abprüfen und weiß, daß er damit auch das Icon angeklickt hat.

