

```

JJJJJ      A      DDDDDD      000      SSSSS
JJJJJ      A A    DD D      0 0      SS S
JJ         AA AA   DD DD     00 00     SS
JJ         A A     DD D      00 00     SSSSS
JJ         AAAAAA  DD DD     00 00     SS
JJ JJ      AA AA   DD D      0 0      S SS
JJJJJ      AA AA   DDDDDD     000      SSSSS
  
```

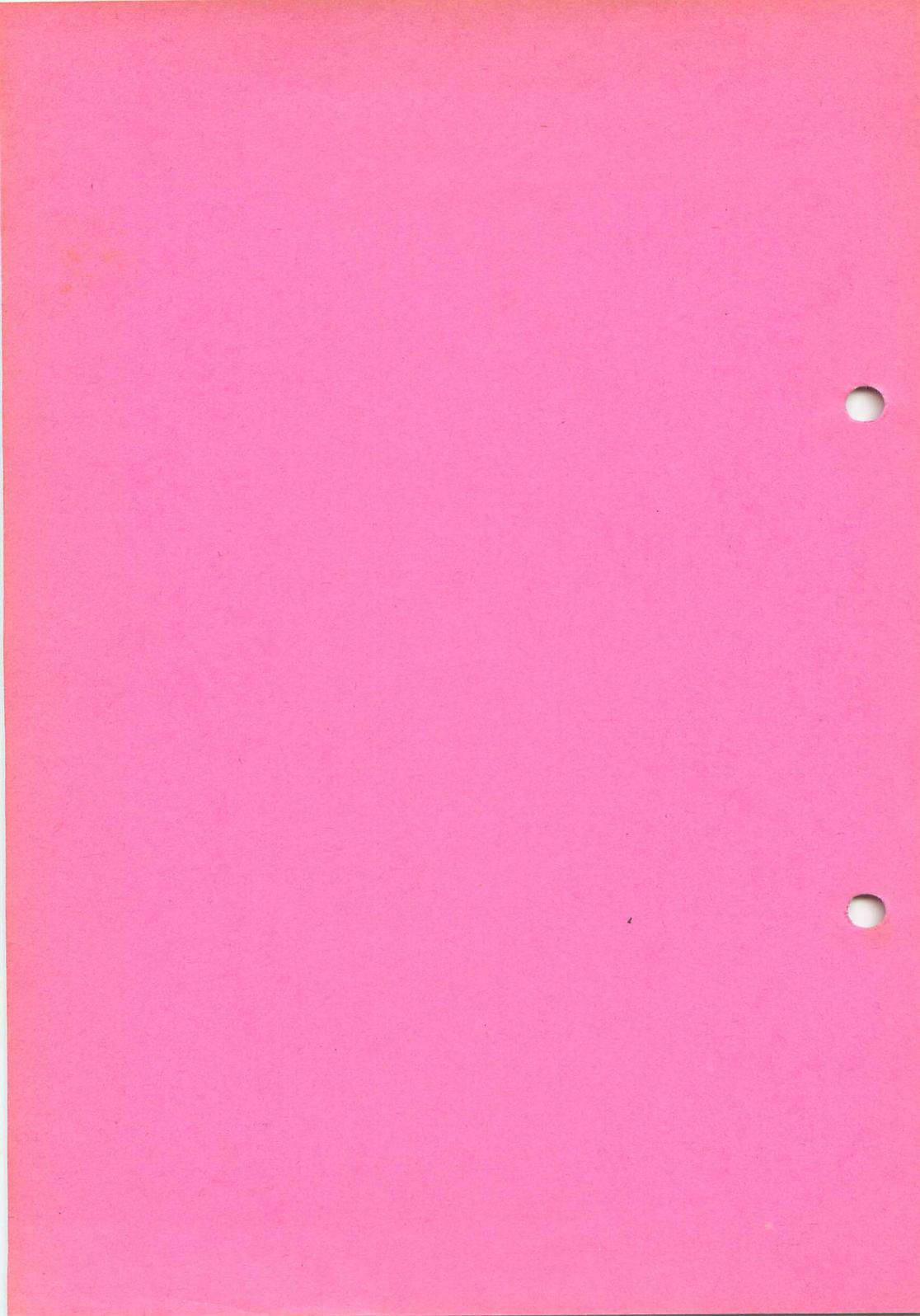
DAS DISKETTEN-BETRIEBSSYSTEM
 FÜR DIE PROZESSOREN
 68008 , 68000 UND 68020
 IM NDR-KLEIN-COMPUTER

(C) 1985, 1986
 KLAUS JANSEN
 KREFELD

VERSION 2.0

Graf Elektronik Systeme GmbH





JJJJ	A	DDDDD	000	SSSS
JJJJ	A A	DD D	0 0	SS S
JJ	AA AA	DD DD	00 00	SS
JJ	A A	DD D	00 00	SSSS
JJ	AAAAAA	DD DD	00 00	SS
JJ JJ	AA AA	DD D	0 0	S SS
JJJJ	AA AA	DDDDD	000	SSSS

DAS DISKETTEN-BETRIEBSSYSTEM
 FÜR DIE PROZESSOREN
 68008 , 68000 UND 68020
 IM NDR-KLEIN-COMPUTER

(C) 1985, 1986
 KLAUS JANSEN
 KREFELD

VERSION 2.0

VORWORT

Das Diskettenbetriebssystem JADOS wurde von mir auf einem NDR - Klein - Computer mit 68008-CPU entwickelt. Als Hauptspeicher standen mir 200 KByte statisches RAM zur Verfügung und als Massenspeicher 2 Floppydisklaufwerke TEAC FD 55 F.

Als Entwicklungshilfsmittel für die Version 2.0 standen mir nur die JADOS-Version 1.2 sowie das Grundprogramm von R.-D. Klein zur Verfügung; ein Beweis für die Leistungsfähigkeit von JADOS und des Grundprogramms.

Mein besonderer Dank gilt meiner Ehefrau Claudia, die die Korrekturlesung des Handbuchs vornahm und an vielen Stellen für verständlichere Formulierungen sorgte.

Sehr herzlich bedanke ich mich auch bei Herrn Axel Granel, der mir viele wertvolle Anregungen und Verbesserungsvorschläge gab.

Krefeld, im Juni 1986

Klaus Janßen

INHALTSVERZEICHNIS

1) ALLGEMEINES	7
1.1) EINLEITUNG	7
1.2) ÄNDERUNGEN ZU VERSION 1.20	7
1.3) LEISTUNGSUMFANG	8
2) INBETRIEBNAHME	9
2.1) GRUNDVORAUSSETZUNGEN	9
2.1.1) Konfiguration ohne Bootkarte	9
2.1.2) Konfiguration mit Bootkarte	9
2.2) SINNVOLLE ERWEITERUNGEN	9
2.3) STARTVORGANG	10
2.3.1) Floppy Start	10
2.3.2) Bibliotheksstart	10
2.3.3) Der AUTOSTART-Mechanismus	11
2.3.4) Fehlersituationen	11
3) DIE SPEICHERVERWALTUNG	12
4) DIE DATEIVERWALTUNG	14
4.1) DATENTRÄGERFORMAT	14
4.2) DISKETTENKAPAZITÄT	14
4.3) INHALTSVERZEICHNIS	15
4.4) SPURTABELLE	16
4.5) DATEISTEUERBLOCK	17

5) DIE KOMMANDOS	18
5.1) EINFÜHRUNG	18
5.1.1) Der Kommandointerpreter	18
5.1.2) Dateinamen	18
5.1.3) Zeileneditor	19
5.1.4) Ladeadresse	19
5.2) DIE INTERNEN KOMMANDOS	20
5.2.1) Übersicht	20
5.2.2) Ausstieg mit Ctrl A	20
5.2.3) Warmstart mit Ctrl C	21
5.2.4) Laufwerkumschaltung mit 1: oder 2:	21
5.2.5) ASS	21
5.2.6) BELL	24
5.2.7) CLS	24
5.2.8) COPY	24
5.2.9) DIR / DIRP	25
5.2.10) ECHO	25
5.2.11) EDIT	26
5.2.12) ERA	27
5.2.13) FF	27
5.2.14) PAUSE	27
5.2.15) PRINT	28
5.2.16) REM	28
5.2.17) REN	28
5.2.18) SYS	29
5.2.19) TLOAD	30
5.2.20) TSAVE	31
5.2.21) TYPE	31
5.3) DIE EXTERNEN KOMMANDOS	32
5.3.1) Programmdateien	32
5.3.1.1) Programme vom Typ .68K	32
5.3.1.2) Programme vom Typ .COM	33
5.3.2) Stapelverarbeitung	33

6) BESCHREIBUNG DER UNTERPROGRAMME	36
6.1) NUMERISCHE ÜBERSICHT	36
6.2) ALPHABETISCHE ÜBERSICHT	37
6.3) BESCHREIBUNG DER EINZELNEN UNTERPROGRAMME	38
6.3.1) getleng	0
6.3.2) getname	1
6.3.3) getstadd	2
6.3.4) lese	3
6.3.5) loadtext	4
6.3.6) motoroff	5
6.3.7) response	6
6.3.8) schreibe	7
6.3.9) strgcomp	8
6.3.10) tload	9
6.3.11) tsave	10
6.3.12) uppercas	11
6.3.13) wrblank	12
6.3.14) writ	13
6.3.15) close	14
6.3.16) copyfile	15
6.3.17) create	16
6.3.18) erase	17
6.3.19) fillfcb	18
6.3.20) open	19
6.3.21) readrec	20
6.3.22) rename	21
6.3.23) setdta	22
6.3.24) writerec	23
6.3.25) getversi	24
6.3.26) getparm	25
6.3.27) hardcopy	26
6.3.28) bell	27
6.3.29) beep	28
6.3.30) errnoise	29
6.3.31) sound	30
6.3.32) inport	31
6.3.33) outport	32
6.3.34) movelbin	33
6.3.35) moveltxt	34
6.3.36) moverbin	35
6.3.37) wrtcmd	36
6.3.38) gtretcod	37
6.3.39) stretcod	38
6.3.40) moveline	39
6.3.41) wraddr	40
6.3.42) ci	41
6.3.43) getparm1	42
6.3.44) getparm2	43
6.3.45) fileload	44
6.3.46) filesave	45
6.3.47) getparm3	46
6.3.48) getparm4	47
6.3.49) loadpart	48
6.3.50) savepart	49
6.3.51) catalog	50

A) A N H A N G	75
A.1) HILFSPROGRAMME VON JADOS	75
A.1.1) ÜBERSICHT	75
A.1.2) COPY	76
A.1.3) BRUCK	77
A.1.4) DSAVE	79
A.1.5) FORMAT	80
A.1.6) MORE	81
A.1.7) KONSTART	82
A.2) ABHILFE BEI FEHLERMELDUNGEN	83

1) ALLGEMEINES

1.1) EINLEITUNG

JADOS ist ein Betriebssystem für die Prozessorfamilie 680xx im NDR-Klein-Computer und unterstützt den Betrieb mit ein oder zwei Minidiskettenlaufwerken (5 1/4 " oder 3 1/2 ") im Format NDR80.

Die vorliegende Version 2.0 ist vollständig kompatibel zur bisherigen Version 1.2. Das bis dato schon recht leistungsfähige JADOS wurde stark verbessert, sowohl im Funktionsumfang als auch in den Punkten Schnelligkeit und Bedienungskomfort. Der Umfang des JADOS - Systems erhöhte sich dabei um 50%, während das Handbuch sogar um fast 100% erweitert werden mußte, um alle Neuerungen zu dokumentieren.

1.2) ÄNDERUNGEN ZU VERSION 1.20

Folgende Funktionserweiterungen und Änderungen wurden durchgeführt:

- An jeder beliebigen Stelle, an der eine Tastatureingabe möglich ist, kann mit der Tastenkombination Ctrl P eine Hardcopy des Textinhalts auf den Drucker erfolgen.
- Es wird die Baugruppe SOUND unterstützt, sowohl im Kommandomodus als auch für den Programmierer.
- Die Eingabe von Kommandos und Dateinamen wurde vereinfacht. Mit der Tastenkombination Ctrl J kann an jeder beliebigen Stelle der vorige Pufferinhalt übernommen werden.
- Sämtliche internen Kommandos können jetzt sowohl mit als auch ohne Parameter aufgerufen werden. Fehlende Parameter werden von JADOS automatisch erfragt.
- Der Startvorgang von JADOS ist nun wesentlich bedienungsfreundlicher. Doppelinstallationen des Systems oder des Variablenbereichs können nicht mehr auftreten.
- Es wurde ein Autostartmechanismus eingebaut. Nach dem Umladen von der Systemdiskette wird die Datei AUTOEXEC gestartet, sofern sie vorhanden ist.
- Einzelschrittbearbeitung ist nun uneingeschränkt möglich.
- Das Laden mehrerer Textmodule kann nun automatisch über eine sogenannte Antwortdatei erfolgen.
- Programme können einen 4-Byte-Code absetzen, der von anderen Programmen verarbeitet werden kann.
- Programme können ein JADOS-Kommando in einen Kommandopuffer schreiben, das nach Rückkehr in den Kommandointerpreter automatisch abgearbeitet wird.

- Es werden 25 neue Unterprogramme für den Programmierer angeboten, darunter Routinen für die SOUND-Baugruppe, Speichertransfer, Laden und Speichern binärer Dateien, Inhaltsverzeichnis usw....
- Es stehen viele neue Kommandos zur Verfügung: Bildschirm löschen, Tonsignal, Drucken mit Seitenvorschub, Seitenvorschub usw....
- Das eingebaute Kommando COPY arbeitet nun etwa 20 mal schneller als vorher.
- Das Laden, Speichern und Löschen von Dateien wird nun 2 bis 8 mal so schnell ausgeführt wie vorher.
- Eine Folge von JADOS-Kommandos kann nun in einer Kommandodatei abgelegt werden, die wie ein Programm aufgerufen werden kann. Damit lassen sich immer wiederkehrende Kommandoabläufe sehr bequem automatisieren.
- Das Inhaltsverzeichnis wurde stark verbessert. So können nun fast 70 Dateien auf einem Bildschirm angezeigt werden. Sehr hilfreich ist auch, daß jetzt Dateien selektiert werden können.
- Die externen Kommandos wurden ebenfalls überarbeitet. So steht jetzt ein Formatierprogramm für alle CPUs zur Verfügung.

1.3) LEISTUNGSUMFANG

- * Laden, Speichern und Löschen von Dateien, Texten und Programmen
- * Die Bereiche gelöschter Dateien stehen wieder uneingeschränkt zur Verfügung
- * Koppeln von Texten; manuell oder über Antwortdatei
- * Ändern von Dateinamen
- * Selektives Inhaltsverzeichnis
- * Laufwerksumschaltung über Tastatur
- * Automatische Abarbeitung einer Startdatei nach dem Booten
- * Volle Unterstützung des Assemblers von R.-D. Klein
- * Volle Unterstützung des Editors von R.-D. Klein
- * Hardcopy des Textinhalts
- * Superschnelle Textausgabe mit Vor- und Rückwärtsblättern
- * Formatieren im NDR80-Format
- * Sehr komfortable Druck-Funktion
- * Laden von Programmen mit Autostart
- * Unterstützung der CPUs 68008, 68000 und 68020 !!
- * Läuft mit und ohne BANKBOOT-Karte !!
- * Komplette Kommandoabläufe können mit Hilfe von Kommandodateien automatisiert werden; ähnlich SUBMIT bei CP/M
- * Deutsche Bedienerführung und Fehlermeldungen
- * Umfangreiches, klar verständliches deutsches Handbuch
- * Sehr günstiger Preis !!!!!

2) IN BETRIEBNAHME

2.1) GRUNDVORAUSETZUNGEN

Für den Betrieb von JADOS sind folgende Baugruppen unbedingt erforderlich:

2.1.1) Konfiguration ohne Bootkarte

- CPU-Karte mit 68008-, 68000- oder 68020-Prozessor.
- Speicherkarten mit Grundprogramm von R.-D. Klein ab Version 4.3 ab Adresse \$00000.
- RAM-Speicher von mindestens 64 KByte lückenlos ab Speicherplatz \$08000.
- Grafikcontroller GDP64.
- Monitor für Darstellung von 80 Zeichen.
- Diskettencontroller FLO2 oder FDC.
- Tastaturinterface KEY.
- ASCII-Tastatur
- Ein Mini-Diskettenlaufwerk doppelseitig, 80 Spuren pro Seite (z.B. TEAC FD 55F).

2.1.2) Konfiguration mit Bootkarte

- CPU-Karte mit 68008-, 68000- oder 68020-Prozessor.
- Speicherkarten mit Grundprogramm von R.-D. Klein ab Version 4.3 auf einer Adresse ungleich 0.
- RAM-Speicher von mindestens 16 KByte lückenlos hinter dem Grundprogramm und 64 KByte ab Adresse \$00000.
- Bankbootkarte(n) mit Booteprom
- Grafikcontroller GDP64.
- Monitor für Darstellung von 80 Zeichen.
- Diskettencontroller FLO2 oder FDC.
- Tastaturinterface KEY.
- ASCII-Tastatur
- Ein Mini-Diskettenlaufwerk doppelseitig, 80 Spuren pro Seite (z.B. TEAC FD 55F).

2.2) SINNVOLLE ERWEITERUNGEN

Für den Betrieb empfehlenswert sind folgende Erweiterungen:

- Centronicsschnittstelle für einen Drucker.
- Matrixdrucker (vorteilhaft ist EPSON RX 80 oder dazu Kompatibler).
- Speichererweiterung auf 128 KByte im ersten zusammenhängenden Speicherbereich.
- Zweites Diskettenlaufwerk.
- SOUND-Baugruppe.

2.3) STARTVORGANG

Es gibt zwei Möglichkeiten JADOS zu starten, nämlich über den Menüpunkt Floppy Start im Grundprogramm oder über die Bibliotheksfunktion.

2.3.1) Floppy Start

Nach dem Einschalten des Computers, einem RESET, einer Einzelschrittbearbeitung im Grundprogramm oder einer Fehlersituation wie z.B. ILLEGAL INSTRUCTION, LINEF oder ADDRESS ERROR muß JADOS von der Systemdiskette geladen werden.

Wurde JADOS bereits installiert und soll erneut gestartet werden, so kann es mit Floppy Start gestartet werden, ohne daß es zu einer Mehrfachinstallation kommt, wie in Version 1.2. Es wird dann lediglich der Bootsektor geladen und anschließend in das bereits vorhandene JADOS verzweigt.

Es muß die Systemdiskette in das Laufwerk 1 eingelegt und folgendes Menü im Grundprogramm angewählt werden:

- 1 = Speichern CAS
- 2 = Laden CAS
- 3 = Prüfen CAS
- 4 = Floppy Start
- W = weiter

Mit dem Menüpunkt 4 wird JADOS gestartet.

nach erfolgreichem Startvorgang meldet sich JADOS mit:

```
J A D O S  VERSION 2.00  CPU 680XX
(C) 1985, 1986 BY KLAUS JANBEN
```

```
FREIER BENUTZERSPEICHER (KBYTE) : XXX
1>
```

Es werden die jeweils vorhandene CPU und der tatsächlich vorhandene freie Benutzerspeicher angezeigt.

Das Zeichen 1> mit dem blinkenden Cursor zeigt an, das JADOS bereit ist, Kommandos entgegenzunehmen.

2.3.2) Bibliotheksstart

JADOS besitzt die Bibliothekskennung und kann daher auch über die Bibliotheksfunktion gestartet werden, sofern es sich im Hauptspeicher befindet. Dies erfolgt sehr viel schneller als der Ladevorgang von Diskette. Nach einem RESET, einer Einzelschrittbearbeitung oder einer Fehlersituation ist jedoch der Stackbereich nicht mehr in Ordnung, sodaß Floppy Start benutzt werden muß.

2.3.3) Der AUTOSTART-Mechanismus

Nach jedem kompletten Ladevorgang von der Systemdiskette sucht JADOS nach einer Datei AUTOEXEC.BAT auf Laufwerk 1. Diese Kommandodatei (siehe auch Kapitel 5.3.2 - Stapelverarbeitung) wird, falls sie existiert, automatisch gestartet. Damit kann der Benutzer Systeminitialisierungen durchführen oder ein Anwendungsprogramm starten.

Dieser Mechanismus wird nur ausgeführt, wenn JADOS komplett von der Systemdiskette geladen wurde, d.h nicht nach einem Bibliotheksstart und auch nicht nach einem Floppy Start, wenn dabei das System nicht geladen wurde.

2.3.4) Fehlersituationen

Es gibt verschiedene Fehlersituationen, die beim Laden oder Starten von JADOS auftreten können.

Beim Urladevorgang können folgende Ereignisse eintreten:

- Das Grundprogramm meldet sich wieder. Dann war kein System auf der Diskette.
- Der Urlader meldet: FALSCHES FORMAT. Dann liegt keine Minidiskette mit Double Density - Formatierung vor.
- Der Urlader meldet: ZU WENIG SPEICHER. Dann ist der erste zusammenhängende Speicherbereich zu klein. Der Speicherbereich muß 64 KByte umfassen.
- Der Urlader meldet: LESEFEHLER BEIM BOOTEN. Dann ist die Systemdiskette nicht mehr in Ordnung. Deshalb ist es wichtig, sich eine Sicherheitskopie der Systemdiskette anzulegen !

Beim Start des JADOS können folgende Meldungen vorkommen:

- JADOS meldet: ZU WENIG SPEICHER !! Dann ist kein freier Benutzerspeicher vorhanden.
- JADOS meldet: WEGEN STACKFEHLER BITTE MIT FLOPPY START BOOTEN !! Dies tritt auf, wenn ein Bibliotheksstart versucht wurde, obwohl ein RESET, eine Einzelschrittbearbeitung oder eine Fehlersituation vorlagen.

3) DIE SPEICHERVERWALTUNG

JADOS verwaltet den ersten zusammenhängenden Speicherbereich. Dabei werden zwei verschiedene Speichermodelle unterschieden, abhängig davon, ob eine Bankbootkarte verwendet wird oder nicht.

Die folgende Abbildung zeigt das Speichermodell einer typischen Konfiguration mit 96 KByte RAM ohne Einsatz der Bankbootkarte:

Adresse	-----	Belegung
\$00000	-----	
	I	I
	I	I >>>> Grundprogramm 4.3
	I	I
\$07FFF	I	I
\$08000	I-----I	-I
	I	I >>>> Variablen und Symboltabelle
\$0BFFF	I	I
\$0C000	I-----I	-I >>>> Beginn des Benutzerspeichers
	I	I
	I	I
	I	I
	I	I
	I	I
	I	I
	I	I ^ Stackbereich
\$1B7E0	I-----I	-I
	I	I >>>> JADOS - System
\$1FFFF	I	I
	-----	>>>> Ende des 1. Speicherbereichs

Auf Adresse \$00000 bis \$07FFF liegt das Grundprogramm. Es belegt 32 KByte Speicherplatz.

Auf Adresse \$08000 bis \$0BFFF werden die Variablen des Grundprogramms sowie die sogenannte Symboltabelle verwaltet. Dafür stehen insgesamt 16 KByte RAM zur Verfügung. Für die Symbole allein sind etwa 13 KByte reserviert, was etwa 720 Symbolen entspricht.

Ab Adresse \$0C000 beginnt der Benutzerspeicher. Die Adresse \$0C000 ist gleichzeitig die Ladeadresse für die .68K-Dateien.

Am Ende des 1. zusammenhängenden Speicherbereichs liegt das JADOS mit seinen Variablen. Davor beginnt der Stack, welcher in Richtung niedrigerer Adressen wächst. Kollisionen zwischen dem Stack und dem Benutzerspeicher werden von JADOS rechtzeitig erkannt und gemeldet. In der derzeitigen Version 2.0 belegt das JADOS-System etwa 18 KByte freien Speicher.

Die folgende Abbildung zeigt das Speicherabbild einer typischen Konfiguration mit Bankbootkarte, 128 KByte Ram ab Adresse \$00000, dem Grundprogramm auf Adresse \$E0000 sowie 16 KByte RAM ab Adresse \$E8000.

Adresse	-----	Belegung
\$00000	-----	>>>> Interruptvektoren
\$00400	I - - - - - I	>>>> Beginn des Benutzerspeichers
	I	I
	I	I
	I	I
	I	I
	I	I
	I	I
	.	
	.	
	I	I
	I	I
	I	I
	I	I
	I	I ^ Stackbereich
\$1B7E0	I - - - - - I	
	I	I >>>> JADOS - System
\$1FFFF	I	I
	-----	>>>> Ende des 1. Speicherbereichs
\$E0000	-----	
	I	I
	I	I >>>> Grundprogramm 4.3
	I	I
\$E7FFF	I	I
\$E8000	I - - - - - I	
	I	I >>>> Variablen und Symboltabelle
\$EBFFF	I	I

Bei dieser Konfiguration sind zwei Speicherbereiche interessant. Im 1. Speicherbereich liegen ab Adresse \$00000 die Interruptvektoren. Ab Adresse \$00400 beginnt der JADOS-Benutzerspeicher. \$00400 ist die Ladeadresse für .68K-Dateien. Das JADOS-System selbst belegt wieder den hinteren Teil des ersten Speicherbereichs und umfaßt derzeit etwa 18 KByte.

Im zweiten Speicherbereich liegt das Grundprogramm. Daran schließt sich der Bereich der Grundprogrammvariablen und der Symboltabelle an. Wer übrigens mit den weiter oben erwähnten 720 Symbolen nicht auskommt, der kann den RAM-Speicher hinter dem Grundprogramm auf 32 KByte ausbauen und hat dann Platz für fast 1500 Symbole.

4) DIE DATEIVERWALTUNG

Die in den Abschnitten 'INHALTSVERZEICHNIS' und 'SPURTABELLE' angegebenen Informationen richten sich vor allem an diejenigen Benutzer, die sich für den internen Aufbau von JADOS interessieren. Der nicht daran Interessierte kann diese Abschnitte überspringen.

4.1) DATENTRÄGERFORMAT

Datenträger sind doppelseitige Disketten mit 96 Spuren pro Zoll. Das Diskettenformat ist wie folgt. (mc - Format) :

- doppelseitig
- doppelte Dichte
- Miniformat 5 1/4 Zoll oder 3 1/2 Zoll
- 80 Spuren pro Seite
- 5 Sektoren pro Spur
- 1024 Byte pro Sektor

Andere Formate werden nicht unterstützt.

4.2) DISKETTENKAPAZITÄT

JADOS verwaltet 160 logische Spuren pro Diskette. Dabei entsprechen den Spuren 0 - 79 die physikalischen Spuren 0 - 79 auf Seite 0 und den Spuren 80 - 159 die physikalischen Spuren 79 - 0 auf Seite 1.

Es gilt folgende Einteilung:

<u>Spuren</u>	<u>Sektoren</u>	<u>Inhalt</u>	<u>Kapazität</u>
0 - 2	1 - 5	Betriebssystem	15 KByte
3	1	Spurtabelle	1 KByte
3	2 - 5	Inhaltsverzeichnis.	4 KByte
4 - 159	1 - 5	Benutzerdateien	780 KByte

4.3) INHALTSVERZEICHNIS

Pro Datei werden 32 Bytes Informationen abgelegt. Da für das Inhaltsverzeichnis 4 Sektoren zur Verfügung stehen, kann JADOS 128 Dateien pro Diskette verwalten.

Für jede Datei ist folgender Datensatz eingetragen:

Byte-Nr	Inhalt
0 - 1	Kennung, z.B. \$E5E5 -> kein Eintrag \$0000 -> gültiger Eintrag \$FFFF -> gelöschter Eintrag
2 - 9	Dateiname
12 - 14	Namenserweiterung bzw. Dateityp
16 - 17	Startspur der Datei
18 - 19	Endesektor, d.h. letzter Sektor der letzten Dateispur
22 - 25	Ladeadresse
26 - 27	Dateilänge, gemessen in Sektoren

Die Bytes Nummer 10,11 und 15 sind intern reserviert. Die Bytes Nummer 20 bis 21 und 28 bis 31 sind für eine spätere Erweiterung reserviert und werden bisher nicht ausgenutzt. Eine sinnvolle Erweiterung wäre z.B. das Datum des letzten schreibenden Zugriffs auf die Datei.

4.4) SPURTABELLE

Bei sehr einfachen Betriebssystemen werden Dateien fortlaufend Spur für Spur abgespeichert. Dies ermöglicht zwar ein schnelles Laden und Speichern der Dateien, hat aber einen wesentlichen Nachteil:

Wenn eine schon vorhandene Datei vergrößert werden soll, so paßt die Ergänzung häufig nicht mehr auf den der Datei zugewiesenen Bereich. Für die gesamte Datei (einschließlich Erweiterung) muß in diesem Fall ein Bereich gesucht werden, der ausreichend groß ist. Meistens liegt ein solcher Bereich hinter der letzten Datei. Der der ursprünglichen Datei (vor der Erweiterung) zugewiesene Bereich wird in der Regel gelöscht und steht einer neuen Datei zur Verfügung. Da Dateien nun aber immer nur fortlaufend, d.h. an einem Stück gespeichert werden können, darf diese neue Datei auf keinen Fall größer als der frei gewordene Bereich sein. Da es in der Praxis aber nicht immer möglich ist, eine Datei "herbeizuzaubern", die genau in die frei gewordene Lücke paßt oder nur wenig kleiner ist, kann es häufig vorkommen, daß ein gelöschter Bereich von einer sehr viel kleineren Datei belegt wird oder überhaupt nicht mehr belegt wird, da man zufällig nur noch größere Dateien zu speichern hat. Dabei geht dann sehr viel Diskettenkapazität verloren, was eine recht unökonomische Speicherplatznutzung darstellt. In ungünstigen Fällen (nach vielen Löschvorgängen) paßt keine Datei mehr auf die Diskette, obgleich vielleicht nur wenige Prozent der Diskettenkapazität ausgenutzt sind.

JADOS verwaltet die Diskettenkapazität wesentlich effizienter. Jede Datei belegt eine oder mehrere komplette Spuren. Umfangreichere Dateien müssen dabei nicht unbedingt auf aufeinanderfolgenden Spuren untergebracht sein. Soll z.B. eine Datei vergrößert werden, so sucht JADOS automatisch nach freien Bereichen (z.B. einen durch Löschen einer Datei freigewordenen Bereich) und speichert die Dateiergänzung darauf ab. Auf diese Weise können alle auf der Diskette vorhandenen Spuren auch wirklich ausgenutzt werden.

Im Inhaltsverzeichnis ist eingetragen, auf welcher Spur eine Datei beginnt. Da größere Dateien an verschiedenen Stellen auf der Diskette gespeichert sein können, wird in einer sogenannten *Spurtabelle* festgehalten, wo es weitergeht. Und zwar werden hier für jede Spur einer Datei jeweils die Nachfolgerspur (also in Richtung auf das Dateiende) und die Vorgängerspur (also in Richtung auf den Datei-anfang) eingetragen.

Die Spurtabelle ist folgendermaßen aufgebaut:

Byte	Inhalt	Zusatzbedeutung
0 - 1	Vorgängerspur	\$ESE5 -> freie Spur \$FFFF -> kein Vorgänger
2 - 3	Nachfolgerspur	4 - 159 -> Nummer der Vorgängerspur \$ESE5 -> freie Spur \$FFFF -> kein Nachfolger 4 - 159 -> Nummer der Nachfolgerspur

Der Vorteil dieses Verfahrens liegt darin, daß vorhandene Dateien problemlos wachsen können und daß die durch Löschen einer Datei frei gewordenen Spuren wieder beliebig zur Verfügung stehen. Ein kleiner Nachteil besteht darin, daß das Laden und Speichern etwas länger dauern kann.

4.5) DATEISTEUERBLOCK

Jede Datei wird von einem Dateisteuerblock, im folgenden mit FCB (engl. file control block) abgekürzt, verwaltet. Es ist dies ein Datensatz, der beim Öffnen oder Anlegen einer Datei gebildet wird, und beim Schließen der Datei wieder unwirksam wird.

Der Datensatz ist folgendermaßen aufgebaut:

Byte	Inhalt	Zusatzbedeutung
0 - 1	Laufwerksnummer	1 oder 2
2 - 9	Dateiname	
10 - 11	Intern reserviert	
12 - 14	Namenserweiterung bzw. Dateityp	
15	Intern reserviert	
16 - 17	Startspur der Datei	
18 - 19	Endesektor, d.h. letzter Sektor der letzten Dateispur	
20 - 21	Für spätere Erweiterungen reserviert	
22 - 25	Ladeadresse	
26 - 27	Dateilänge, gemessen in Sektoren	
28 - 31	Für spätere Erweiterungen reserviert	
32 - 33	Sektor des Inhaltsverzeichnisses	
34 - 35	Startbyte im Sektor des Inhaltsverzeichnisses	
36 - 37	Dateistatus	0 -> geschlossen 1 -> geöffnet
38 - 39	Aktuelle Dateispur	
40 - 41	Aktueller Dateisektor	
42 - 43	Letzte Dateispur	
44 - 47	Aktuelle Lade- bzw. Speicheradresse	

5) DIE KOMMANDOS

5.1) EINFÜHRUNG

5.1.1) Der Kommandointerpreter

Der Benutzer kommuniziert mit JADOS über Kommandos, die in der Regel über die Tastatur eingegeben werden. Die Auswertung besorgt der im JADOS eingebaute Kommandointerpreter. Dieser wird sofort nach dem Start von JADOS aktiv und zeigt mit einem Prompt die Bereitschaft des Systems an, Kommandos entgegenzunehmen.

Das Prompt besteht aus der Nummer des aktuellen Laufwerks und dem Zeichen ">". Nach dem Start erscheint also

1>

Viele Kommandos benötigen Parameter - in der Regel Dateinamen - um ihre Aufgabe zu erfüllen. Diese Parameter können von erfahrenen Anwender direkt hinter dem Kommando, abgesetzt durch ein Leerzeichen, eingegeben werden.

Wer sich die Anzahl und Bedeutung der Parameter aber nicht merken will, braucht nur die Kommandos eingeben. JADOS fragt dann alle benötigten Parameter mit erklärenden Hinweisen ab.

JADOS unterscheidet zwischen internen und externen Kommandos. Die Internen sind fest in JADOS eingebaut. Die externen Kommandos sind Namen von Programmdateien, die nach dem Aufruf von der Diskette geladen und dann automatisch abgearbeitet werden.

5.1.2) Dateinamen

Ein Dateiname besteht aus drei Komponenten, der Laufwerksnummer, dem eigentlichen Namen und dem Dateityp. Zwischen der Laufwerksnummer und dem Namen muß das Trennzeichen ":" und zwischen dem Namen und dem Dateityp das Trennzeichen "." stehen.

Wird die Laufwerksnummer nicht angegeben, dann nimmt JADOS an, daß das aktuelle Laufwerk verwendet werden soll. Das aktuelle Laufwerk wird durch das Systemprompt angezeigt. Mögliche Laufwerksnummern sind 1: und 2:.

Der Name umfaßt 1 bis 8 Zeichen, wobei kein Unterschied zwischen Klein- und Großbuchstaben besteht. Die meisten Sonderzeichen werden zwar akzeptiert, sollten aber nicht verwendet werden.

Der Dateityp umfaßt 0 bis 3 Zeichen, ansonsten gilt das für den Namen Besagte.

Die Dateitypen .68K, .COM und .BAT sind von JADOS mit festen Bedeutungen belegt.

Beispiele für gültige Dateinamen:

```
FORMAT.68K
2:FORMAT.68K
FORMAT
```

5.1.3) Zeileneditor

Bei der Eingabe von Kommandos und Dateinamen über die Tastatur sind folgende Möglichkeiten gegeben:

- * Mit der <- bzw. der DEL-Taste kann das Zeichen links vom Cursor gelöscht werden. Der Cursor rückt dabei um eine Stelle nach links.
- * Mit der RETURN-Taste wird die Eingabe ordnungsgemäß beendet.
- * Mit Ctrl C kann die Eingabe abgebrochen werden. Geschieht dies während der Eingabe eines Kommandos, so wird ein "Warmstart" ausgeführt, d.h. JADOS initialisiert sich neu. Während der Eingabe eines Dateinamens bewirkt Ctrl C den Abbruch des Kommandos.
- * In JADOS gibt es Zeichenpuffer für Kommandos und Dateinamen. Mit Ctrl J kann der Pufferinhalt abgerufen werden. Wenn man z.B. ein vorher eingegebenes Kommandos wiederholen möchte, so braucht man nur Ctrl J und RETURN eingeben.
- * Mit Ctrl P kann eine Hardcopy des Textinhalts des Bildschirms auf den Drucker ausgegeben werden.

5.1.4) Ladeadresse

Der Begriff "Ladeadresse" wird im folgenden häufiger vorkommen. Darunter wird eine Speicheradresse verstanden, ab der ein Text oder ein Programm von Diskette geladen wird.

Bei Systemen ohne Bankbootkarte ist die Ladeadresse für Programme gleich \$C000 und bei Systemen mit Bankbootkarte gleich \$0400.

5.2) DIE INTERNEN KOMMANDOS

5.2.1) Übersicht

In der folgenden alphabetischen Übersicht sind alle internen JADOS-Kommandos aufgelistet. Dabei wird folgende Nomenklatur verwendet:

- [...] -> Die in den eckigen Klammern angegebenen Parameter sind optionell, d.h. man kann sie beim Kommandoaufruf mit eingeben.
- xx | yy -> Die durch den senkrechten Strich getrennten Parameter sind alternativ.

Kommando	Parameter	Bedeutung
Ctrl A	--	Ausstieg aus JADOS
Ctrl C	--	Warmstart
1:	--	Umschaltung auf Laufwerk 1
2:	--	Umschaltung auf Laufwerk 2
ASS	[dateiname]	Assemblierung
BELL	--	Glockenton
CLS	--	Bildschirm löschen
COPY	[datei1 datei2]	Datei kopieren
DIR	[dateiname]	Inhaltsverzeichnis anzeigen
DIRP	[dateiname]	Inhaltsverzeichnis ausdrucken
ECHO	ON OF	Stapelverarbeitungskommandos anzeigen nicht anzeigen
EDIT	[dateiname]	Textdatei eingeben oder ändern
ERA	[dateiname]	Datei löschen
FF	--	Seitenvorschub am Drucker
PAUSE	[beliebiger Text]	Stapelverarbeitung anhalten
PRINT	[dateiname]	Textdatei ausdrucken
REM	[beliebiger Text]	Kommentar in Stapelverarbeitung
REN	[datei-alt datei-neu]	Datei umbenennen
SYS	[1 2]	System kopieren
TLOAD	[adresse dateiname]	Textdateien auf bestimmte Adresse laden
TSAVE	[dateiname adresse]	Text ab bestimmter Adresse abspeichern
TYPE	[dateiname]	Textdatei auf Bildschirm anzeigen

5.2.2) Ausstieg mit Ctrl A

Mit der Tastenkombination Ctrl A wird JADOS verlassen. Der Benutzer befindet sich dann wieder in der Systemumgebung des Grundprogramms von R.-D. Klein.

Der erneute Einsprung in JADOS wird in der Regel mit der Bibliotheksfunktion eingeleitet oder alternativ mit FLOPPY START.

5.2.3) Warmstart mit Ctrl C

Mit der Tastenkombination Ctrl C wird ein sogenannter Warmstart des Betriebssystems veranlaßt. JADOS reagiert darauf wie folgt:

- Der Bildschirm wird gelöscht.
- Das Copyright, die Versionsnummer, die CPU und der freie Benutzerspeicher werden angezeigt.
- Das Laufwerk 1 wird zum aktuellen Laufwerk
- Alle Tastaturpuffer werden gelöscht.

5.2.4) Laufwerkumschaltung mit 1 = oder 2 =

Das Systemprompt zeigt die Nummer des aktuellen Laufwerks an. JADOS bietet zwei Kommandos zur Umschaltung des aktuellen Laufwerks an:

- ```
1: --> Laufwerk 1 wird aktuell
2: --> Laufwerk 2 wird aktuell.
```

### 5.2.5) ASS

ASS ist die Abkürzung für *assemble*. Damit kann ein Quellprogramm in ablauffähigen Maschinencode übersetzt werden. JADOS verwendet den im Grundprogramm enthaltenen Assembler.

Die Arbeitsweise des Kommandos hängt davon ab, ob beim Aufruf ein Parameter übergeben wird.

#### 5.2.5.1) Mit Parameter

Diese Kommandoart wird hauptsächlich bei kleinen Programmen verwendet, die nur aus einem Quellprogrammmodul bestehen.

Der übergebene Parameter ist der Name der Quellprogrammdatei. Diese Textdatei wird auf die Ladeadresse für Programme plus \$1000 geladen. Damit bleiben 4 KByte Speicherplatz für den Maschinencode. Nach einem fehlerfreien Assemblerlauf wird der Maschinencode unter dem eingegebenen Namen abgespeichert, wobei als Dateityp automatisch .68K eingesetzt wird.

#### 5.2.5.2) Ohne Parameter

Diese Kommandoart bietet dem Anwender wesentlich mehr Möglichkeiten.

Nach der Kommandoeingabe erscheint folgender Text:

```
A S S E M B L E R

!! KEINE ORG- ODER OFFSETANWEISUNG BENUTZEN !!
!! STARTADRESSE IST KONFIGURATIONSABHANGIG !!
!! NORMAL: $C000 - MIT BOOTKARTE: $0400 !!
```

Beim Assembliervorgang müssen drei Speicherbereiche verwaltet werden, die Symboltabelle, der erzeugte Maschinencode und der Quelltext. Der Maschinencode wird ab Adresse \$C000 bzw. \$0400 abgelegt, daher dürfen keine ORG- oder OFFSET-Anweisungen benutzt werden. Die Startadresse der Symboltabelle liegt hinter dem Grundprogramm und umfaßt in der Normalkonfiguration etwa 13 KByte und in der Bootkartenkonfiguration je nach Speicherausbau zwischen 5 KByte und 29 KByte. Dabei entsprechen 13 KByte etwa 700 Symbolen !!

### ACHTUNG !!

Wenn die Symboltabelle in der Normalkonfiguration die Adresse \$C000 überschreitet, dann wird sie durch den abgelegten Maschinencode zerstört.

Nach dem Hilfetext wird der Benutzer aufgefordert, die Startadresse des Quelltextes einzugeben:

QUELLPROGRAMME LADEN  
WELCHE STARTADRESSE ? (DEFAULT = \$00XXXX)

Der Quelltext wird hinter den Maschinencode geladen. Bei einer Fehleingabe oder wenn kein Wert eingegeben wurde, dann wählt JADOS automatisch den DEFAULT - Wert. Dieser liegt 4 KByte hinter dem Start des Maschinencodes, also \$D000 bzw. \$1400.

Die folgende Übersicht zeigt die Lage von Symboltabelle, Maschinencode und Quelltext in Abhängigkeit von der Konfiguration:

|                     | NORMALKONFIGURATION | BOOTKARTENKONFIGURATION |
|---------------------|---------------------|-------------------------|
| Symboltabelle       | \$008A4E            | \$0E8A4E                |
| Maschinencode       | \$00C000            | \$000400                |
| Quelltext (Default) | \$00D000            | \$001400                |

Nach Eingabe der Startadresse wird der Benutzer aufgefordert, das Quellprogramm zu laden. Dabei können mehrere Quellmodule in Folge geladen werden, die dann als Ganzes assembliert werden. Damit kann der Programmierer kleine Quellmodule wie Bausteine zu einem großen Programm verbinden.

Das Laden der Quellmodule geschieht menügeführt:

- 1 - TEXT LADEN
- 2 - automatisch
- 9 - beenden

Mit 9 kann der Ladevorgang abgebrochen werden.

Mit 1 oder Return wird eine Textdatei geladen:

WELCHE TEXTDATEI LADEN ?

Der Benutzer kann hier den Namen des Quellprogrammes eingeben. Nach jedem Ladevorgang erscheint obiges Menü erneut.

Mit 2 kann eine automatische Antwortdatei geladen werden, die die Namen aller Quellmodule des Programms enthält. Diese Antwortdatei wird mit dem normalen Editor erstellt und benötigt keinen speziellen Dateityp. Beim Ladevorgang werden alle Namen und zusätzlich die Endadressen angezeigt. Diese Funktion ist besonders bei großen Programmen mit vielen Quellmodulen sehr hilfreich.

Danach erscheint ein Ausgabemenü, in dem der Bediener die Art der Ausgabe des Assemblers festlegen kann:

A U S G A B E  
1 - NUR FEHLER  
2 - Bildschirm  
3 - Drucker  
9 - beenden

Mit 1 oder Return wird festgelegt, daß nur Fehlermeldungen auf dem Bildschirm erscheinen.

Mit 2 wird festgelegt, daß das komplette Listing einschließlich Fehlermeldungen auf dem Bildschirm ausgegeben wird. Das Listing kann mit CTRL S angehalten und mit CTRL Q fortgesetzt werden.

Mit 3 wird festgelegt, daß das komplette Listing einschließlich Fehlermeldungen auf dem Drucker ausgegeben wird.

Mit 9 kann das gesamte Kommando abgebrochen werden.

Nun erfolgt der Assembliervorgang. Falls Fehler aufgetreten sind, wird das Maschinenprogramm nicht gespeichert. Bei fehlerfreien Maschinenprogrammen wird der Bediener aufgefordert:

NAMEN DER PROGRAMMDATEI (.68K) EINGEBEN:

Es muß nur der Name eingegeben werden. Der Dateityp .68K wird automatisch vergeben. Dateien dieses Typs können dann im Kommando-interpretier von JADOS durch Eingeben des Namens geladen und gestartet werden.

**5.2.6) BELL**

Hiermit kann ein Glockenton erzeugt werden.

Das Kommando ist für den Stapelverarbeitungsbetrieb gedacht, um den Benutzer auf Eingaben oder dergleichen aufmerksam zu machen.

Voraussetzung dafür sind die Baugruppe **SOUND** und ein Lautsprecher. Die Baugruppe wird vom Grundprogramm unterstützt.

Leider ergeben sich Schwierigkeiten mit dem Drucker, wenn die **SOUND**-Baugruppe auf die vom Grundprogramm vorgesehene Adresse **FFFFFF40** eingestellt wird. Deshalb adressiert **JADOS** diese Baugruppe auf der Adresse **FFFFFF50**.

Die genannten I/O-Adressen beziehen sich auf das System mit **68008**-CPU. Die entsprechenden Adressen für die anderen CPUs der **68000**-Familie werden von **JADOS** automatisch angepaßt.

**5.2.7) CLS**

Mit diesem Kommando werden der Bildschirm gelöscht und der **80**-Zeichen-Modus eingestellt. Dies ist auch für Stapelverarbeitungsbetrieb interessant.

**5.2.8) COPY**

**COPY** dient zum Kopieren von Dateien. Das Kommando kann mit oder ohne Parameter aufgerufen werden.

a) mit Parameter:

1. Parameter: Name der Quelldatei
2. Parameter: Name der Zieldatei

b) ohne Parameter:

Nach der Kommandoeingabe wird gefragt:

**WELCHE DATEI KOPIEREN ?**

Hier ist der Name der zu kopierenden Datei einzugeben. Danach wird der Name der Kopie erfragt:

**NAME DER KOPIE ?**

Nach erfolgreichem Kopiervorgang meldet **JADOS**:

**DATEI KOPIERT .**

### 5.2.9) DIR / DIRP

Mit DIR bzw. DIRP wird das Inhaltsverzeichnis einer Diskette ausgegeben.

DIR --> Anzeige auf dem Bildschirm.  
DIRP --> Ausdruck auf dem Drucker.

Es werden jeweils der Name, der Dateityp und die Größe einer Datei angegeben. Am Schluß werden die Anzahl der aufgelisteten Dateien, deren Gesamtgröße und die Anzahl der freien Spuren und Sektoren der Diskette angegeben. Das folgende Beispiel zeigt einen Ausschnitt aus der Entwicklungsdiskette für dieses Handbuch:

| INHALTSVERZEICHNIS VON LAUFWERK 2 |     |        |                  |     |        |
|-----------------------------------|-----|--------|------------------|-----|--------|
| --NAME--                          | TYP | LAENGE | --NAME--         | TYP | LAENGE |
| TRAP0005                          | JHB | 18     | TRAP0613         | JHB | 14     |
| TRAP2230                          | JHB | 10     | TRAP3141         | JHB | 14     |
| COMASS                            | JHB | 15     | COMREST          | JHB | 1      |
| ANZAHL DATEIEN:                   |     | 8      | ANZAHL SEKTOREN: |     | 103    |
| FREIE SEKTOREN:                   |     | 375    | FREIE SPUREN:    |     | 53     |

a) ohne Parameter

Es wird das komplette Inhaltsverzeichnis der im aktuellen Laufwerk eingelegten Diskette ausgegeben.

b) mit Parameter

Hier können das Laufwerk sowie bestimmte Dateien selektiert werden. Der Parameter wird wie ein Dateiname eingegeben. Alle auf den Namen passende Dateien werden dann aufgelistet. Es können auch sogenannte Jokerzeichen ( im EDV-Jargon "Wildcards" genannt ) eingegeben werden. Dabei ist das Zeichen "?" ein Joker für einen Buchstaben und das Zeichen "\*" ein Joker für alle Buchstaben.

Beispiele:

|               |                                                                      |
|---------------|----------------------------------------------------------------------|
| DIR *.68K     | --> Alle Dateien vom Typ .68K                                        |
| DIR DOS*.*    | --> Alle Dateinamen, die mit "DOS" anfangen                          |
| DIR BRIEF.TXT | --> Alle Dateinamen, die mit "BRIEF" anfangen und vom Typ "TXT" sind |
| DIR 2:*.?68   | --> Alle Dateien auf Laufwerk 2, deren Typname mit "68" endet        |

### 5.2.10) ECHO

Bei der Stapelverarbeitung werden die einzelnen Kommandos normalerweise angezeigt. Mit diesem Kommando kann die Anzeige wahlweise ein- oder ausgeschaltet werden.

ECHO ON --> Anzeige einschalten  
ECHO OF --> Anzeige abschalten

Normalerweise ist die Anzeige eingeschaltet.

## 5.2.11) EDIT

EDIT dient zum Eingeben und/oder Ändern von Textdateien. Dabei wird der im Grundprogramm enthaltene bildschirmorientierte Editor von R.-D. Klein verwendet.

Die Arbeitsweise von EDIT hängt davon ab, ob ein Dateiname als Parameter übergeben wird.

### 5.2.11.1) Mit Parameter

Falls die Datei bereits existiert, wird sie automatisch von der Diskette geladen. Der Benutzer kann dann den Text ändern. Nach dem Editieren wird der Text automatisch unter dem gleichen Namen abgespeichert.

Falls die Datei noch nicht existiert, kann der Benutzer einen neuen Text eingeben. Nach der Eingabe wird der Text automatisch unter dem eingegebenen Namen abgespeichert.

### 5.2.11.2) Ohne Parameter

Nach der Eingabe von EDIT erscheint folgendes Menü:

```

T E X T E D I T O R
 1 - neuen Text erstellen
 2 - ALTEN TEXT ÄNDERN
 9 - beenden

```

Mit dem Menüpunkt 1 kann ein völlig neuer Text erstellt werden. Nach der Eingabe verläßt der Bediener den Editor wie gewohnt mit CTRL KX und erhält folgende Aufforderung:

NAMEN DER NEUEN TEXTDATEI EINGEBEN:

Jetzt kann dem erstellten Text ein Name zugewiesen werden, unter dem er auf der Diskette abgespeichert wird. Mit CTRL C oder CTRL A kann das Abspeichern verhindert werden.

Mit dem Menüpunkt 2 oder Return kann ein bereits erstellter Text geändert werden. Dazu wird der Bediener gefragt:

WELCHE TEXTDATEI ÄNDERN ?

Der Bediener kann nun den Namen der zu ändernden Textdatei eingeben. Diese wird von der Diskette geladen und kann dann mit dem Editor bearbeitet werden. Nach Verlassen des Editors mit CTRL KX wird dem Benutzer der Name der ursprünglichen Textdatei angezeigt und dann gefragt:

NAMEN DER GEÄNDERTEN TEXTDATEI EINGEBEN:

Gibt der Benutzer den gleichen Namen ein, so wird die ursprüngliche Textdatei durch die geänderte ersetzt. Gibt er einen anderen Namen ein, so wird der geänderte Text unter diesem Namen abgespeichert, während die ursprüngliche Textdatei erhalten bleibt. Gibt er Ctrl A oder Ctrl C ein, so wird nichts abgespeichert.

Mit dem Menüpunkt 9 kann das Kommando beendet werden.

**ACHTUNG !!**

Folgende Hinweise sind unbedingt zu beachten:

- Der Editor von R.-D. Klein kann nur Texte bis 64 KByte bearbeiten. Wird diese Grenze überschritten, so produziert der Editor nur noch "Schrott". Die Reaktionen von JADOS können dann auch fehlerhaft sein.
- Der Benutzer ist dafür verantwortlich, daß der Text nicht in den Arbeitsspeicher von JADOS wächst. Als Faustregel gilt, daß die als Tor angezeigte Adresse nicht größer wird als die Endadresse des ersten zusammenhängenden Speicherbereichs minus 22 KByte.  
 Beispiel: Speicherausbau von \$8000 - \$17FFF (= 64 KByte)  
 Textstart ist \$C000  
 Tor muß kleiner sein als \$12800.
- Falls bei der Eingabe des Namens der geänderten Textdatei ein Fehler auftritt, so wird der Text nicht gespeichert. Hier kann sich der Benutzer mit dem TSAVE-Kommando helfen, wobei die Frage nach der Ladeadresse mit RETURN zu quittieren ist.

**5.2.12) ERA**

Mit ERA kann eine Datei gelöscht werden. Die freigewordenen Bereiche auf der Diskette können von anderen Dateien wieder belegt werden. Seit Version 2.0 wird der Löschvorgang sehr schnell ausgeführt.

a) mit Parameter

Die als Parameter übergebene Datei wird gelöscht.

b) ohne Parameter

Der Benutzer wird nach dem Namen der zu löschenden Datei gefragt:

WELCHE DATEI LÖSCHEN ?

Der Löschvorgang wird anschließend bestätigt:

DATEI GELÖSCHT

**5.2.13) FF**

Hiermit kann ein Seitenvorschub auf dem Drucker ausgelöst werden. Dies ist vor allem beim Ausdruck mehrerer Textdateien über eine Stapelverarbeitung nützlich.

**5.2.14) PAUSE**

Mit diesem Kommando kann der Ablauf einer Stapelverarbeitung angehalten werden. JADOS gibt dabei folgende Meldung aus:

Wenn fertig, bitte eine Taste betätigen ...

JADOS wartet dann, bis der Benutzer eine beliebige Taste betätigt.

**5.2.15) PRINT**

Mit PRINT kann eine Textdatei auf den Drucker ausgegeben werden. Dabei wird automatisch nach jeweils 65 Zeilen ein Seitenvorschub erzeugt.

Dem Kommando kann der Name der Textdatei als Parameter übergeben werden. Bei Aufruf des Kommandos ohne Parameter fragt JADOS:

WELCHE TEXTDATEI AUSGEBEN ?

Der Name der Textdatei kann dann eingegeben werden.

**5.2.16) REM**

Dieses Kommando dient zum Einbringen von Kommentarzeilen in Stapelverarbeitungsdateien.

**5.2.17) REN**

REN dient als Abkürzung für *rename*. Mit diesem Kommando kann der Name einer schon existierenden Datei geändert werden. Die nötigen Parameter können wieder wahlweise beim Aufruf mitgegeben werden oder werden von JADOS erfragt.

a) mit Parameter:

1. Parameter: Alter Name
2. Parameter: Neuer Name

b) ohne Parameter:

Nach der Kommandoeingabe fragt JADOS:

WELCHE DATEI UMBENENNEN ?

Hier wird der alte Name eingegeben. Danach erscheint die Frage nach dem neuen Namen:

NEUER NAME ?

Nach der Umbenennung bestätigt JADOS den Vorgang

DATEI UMBENANNT .

5.2.18) **SYS**

SYS ist die Abkürzung von *system* und dient zum Kopieren des Betriebssystems auf eine andere Diskette. Dabei werden die Systemspuren von der Systemdiskette geladen und anschließend auf die neue Diskette übertragen.

Dieses Kommando erfordert mindestens 15 KByte freien Benutzerspeicher !!

Das Kommando arbeitet wahlweise mit einem Laufwerk oder mit zweien.

Wird SYS ohne Parameter aufgerufen, so erscheint ein Menü. Gibt man gleich SYS 1 ein, so wird der Menüpunkt 1 ausgeführt, bei SYS 2 der Menüpunkt 2.

KOPIEREN DES JADOS AUF DIE SYSTEMSPUREN

- 1 - LAUFWERK 1: AUF 1:
- 2 - Laufwerk 1: auf 2:
- 9 - Kommando abbrechen

Mit 1 oder RETURN wird JADOS veranlaßt, die Systemkopie nur auf Laufwerk 1 durchzuführen. JADOS fordert den Benutzer dann auf:

SYSTEMDISKETTE IN LAUFWERK 1 EINLEGEN  
LEERTASTE DRÜCKEN .

Der Benutzer muß nun die JADOS-Systemdiskette in Laufwerk 1 einlegen und danach die Leertaste betätigen. JADOS kopiert nun die Systemspuren in den Hauptspeicher. Danach erscheint die Aufforderung:

DISKETTE WECHSELN !!  
LEERTASTE DRÜCKEN .

Der Benutzer muß nun die Systemdiskette entnehmen und die neue Diskette in Laufwerk 1 einlegen. Danach ist die Leertaste zu betätigen. JADOS kopiert nun das im Hauptspeicher befindliche System auf die Systemspuren der neuen Diskette.

Mit dem Menüpunkt 2 können beide Diskettenlaufwerke ausgenutzt werden, sofern darüber verfügt. Der Benutzer wird nun aufgefordert:

SYSTEMDISKETTE IN LAUFWERK 1: UND KOPIE IN 2: EINLEGEN  
LEERTASTE DRÜCKEN .

Nach Drücken der Leertaste kopiert JADOS die Systemspuren von Laufwerk 1 auf Laufwerk 2.

Nach erfolgreicher Kopie meldet JADOS:

SYSTEM ABGESPEICHERT .

## 5.2.19) TLOAD

TLOAD dient zum Laden einer oder mehrerer Textdateien in den Hauptspeicher zwecks weiterer Bearbeitung mit dem Grundprogramm. Eine mögliche Anwendung wäre das Laden eines Quellprogramms, um es im Grundprogramm mit der Option DEBUG EIN zu assemblieren und dann eine Einzelschrittverarbeitung durchzuführen.

a) mit Parameter

1. Parameter: Speicheradresse, ab der der Text geladen werden soll
2. Parameter: Name der Textdatei

b) ohne Parameter

Dies bietet dem Benutzer mehr Möglichkeiten als der Aufruf mit Parametern. Zuerst wird die Startadresse erfragt.

WELCHE STARTADRESSE ? (DEFAULT = \$00XXXX)

Die angegebene Adresse kann mit RETURN bestätigt werden oder aber man gibt die gewünschte Adresse ein.

Nach Eingabe der Startadresse wird der Benutzer aufgefordert, den Quelltext zu laden. Dabei können mehrere Quelltexte in Folge geladen werden, die dann als ein Text zusammengebunden werden. Damit kann man große Texte erzeugen.

Das Laden der Quelltexte geschieht menügeführt:

- 1 - TEXT LADEN
- 2 - automatisch
- 9 - beenden

Mit 9 kann der Ladevorgang abgebrochen werden.

Mit 1 oder Return wird eine Textdatei geladen:

WELCHE TEXTDATEI LADEN ?

Der Benutzer kann hier den Namen der Textdatei eingeben. Nach jedem Ladevorgang erscheint obiges Menü erneut.

Mit 2 kann eine automatische Antwortdatei geladen werden, die die Namen aller Quelltexte enthält. Diese Antwortdatei wird mit dem normalen Editor erstellt und benötigt keinen speziellen Dateityp. Beim Ladevorgang werden alle Namen und zusätzlich die Endadressen angezeigt.

### 5.2.20) TSAVE

TSAVE dient zum Abspeichern von Texten aus dem Hauptspeicher auf Diskette.

a) mit Parameter

1. Parameter: Name, unter dem der Text gespeichert werden soll
2. Parameter: Speicheradresse, ab der der Text gespeichert werden soll

b) ohne Parameter

Der Benutzer wird zuerst nach dem Namen der Textdatei gefragt:

NAMEN DER TEXTDATEI EINGEBEN:

Nach der Eingabe des Namens erscheint die Frage nach der Startadresse:

WELCHE STARTADRESSE ? (DEFAULT = \$00XXXX)

### 5.2.21) TYPE

Mit TYPE kann eine Textdatei auf den Bildschirm ausgegeben werden.

Dem Kommando kann der Name der Textdatei als Parameter mitgegeben werden. Bei Aufruf des Kommandos ohne Parameter fragt JADOS:

WELCHE TEXTDATEI AUSGEBEN ?

Der Name der Textdatei kann dann eingegeben werden.

### 5.3) DIE EXTERNEN KOMMANDOS

Bei den externen Kommandos handelt es sich um Namen von Dateien. Dabei muß zwischen Programmdateien (Typ .68K oder .COM) und Kommandodateien (Stapelverarbeitung ; Typ .BAT) unterschieden werden.

Daß es sich um ein externes Kommando handelt, erkennt JADOS am Dateityp, d.h. an der Namenserweiterung .68K, .COM oder .BAT. Diese Namenserweiterung braucht der Benutzer jedoch nicht einzugeben. Angenommen, der Benutzer hat die Programmdatei DRUCKE.68K entwickelt und abgespeichert und möchte diese nun aufrufen. Er gibt dazu lediglich DRUCKE ein. JADOS sucht nun nach der Datei DRUCKE.68K, lädt diese und startet sie automatisch. Selbstverständlich darf der Benutzer einem externen Kommando nicht den Namen eines internen Kommandos geben, wie z.B. PRINT. JADOS würde sogleich das interne Kommando starten.

Die Reihenfolge, in der JADOS externe Kommandos sucht, ist:  
**.68K --> .COM --> .BAT**

JADOS sucht also zuerst nach einer Datei vom Typ .68K, danach vom Typ .COM und zuletzt vom Typ .BAT.  
 Erst wenn alle drei Dateien nicht existieren, meldet JADOS

FALSCHES KOMMANDO .

Selbstverständlich darf der Benutzer auch keine gleichen Namen für externe Kommandos benutzen, z.B. DRUCKE.68K und DRUCKE.BAT, denn JADOS würde als Kommando nur DRUCKE.68K finden und ausführen.

#### 5.3.1) Programmdateien

Programmdateien bestehen aus ausführbarem Maschinencode. JADOS erkennt solche Dateien ausschließlich am Dateityp.

##### 5.3.1.1) Programme vom Typ .68K

Diese Programmdateien werden von JADOS auf die Ladeadresse \$C000 bzw. \$0400 geladen und automatisch gestartet. Die Startadresse entspricht dabei der Ladeadresse.

Falls ein Programm den sogenannten Bibliotheksvorspann besitzt, so wird die Startadresse vom entsprechenden Bibliothekseintrag bestimmt. Es werden jedoch nur solche Programme gestartet, die als relokativ gekennzeichnet sind, die also nicht an eine feste Startadresse gebunden sind.

Jedes Programm muß mit einem "RTS"-Befehl beendet werden, damit JADOS wieder die Programmkontrolle erhält.

### 5.3.1.2) Programme vom Typ .COM

Seit Version 2.0 steht der Programmtyp .COM zur Verfügung. Diese Programmdateien werden aber nicht auf die Ladeadresse geladen, sondern möglichst weit hinten im Benutzerspeicher. Dadurch bleibt der Speicher ab der Ladeadresse frei.

.COM-Dateien müssen voll relokativ sein, um fehlerfrei ablaufen zu können. Die Ladeadresse liegt nämlich nicht fest, sondern wird von JADOS automatisch gewählt.

Der Dateityp .COM wird durch Umbenennen einer .68K-Datei erzeugt.

### 5.3.2) Stapelverarbeitung

Bisher konnten Kommandos nur über die Tastatur eingegeben werden. Seit Version 2.0 ist nun ein Mechanismus verfügbar, der eine Reihe von Kommandos, die in einer Textdatei abgelegt sind, automatisch abarbeitet. Immer wiederkehrende Kommandofolgen, z.B. bei einer Datensicherung, können auf diese Weise bequem und sicher ablaufen.

Dies entspricht der SUBMIT-Funktion des bekannten Betriebssystems CP/M 68K.

Die Kommandodateien sind ganz normale Textdateien, die mit dem Editor erstellt werden. Sie müssen vom Dateityp **.BAT** sein, damit JADOS sie als Kommandodateien erkennen kann.

Jede Textzeile darf ein JADOS-Kommando enthalten. Dabei können sämtliche interne und externe Kommandos verwendet werden. Lediglich die Verwendung von Kommandodateien ist eingeschränkt, da eine Verschachtelung von Kommandodateien nicht zulässig ist. Nur das letzte Kommando einer Stapelverarbeitungsdatei darf eine Kommandodatei sein! Auf diese Weise können Stapelverarbeitungsdateien verkettet werden.

Die Kommandos werden der Reihe nach abgearbeitet, bis das letzte Kommando bearbeitet worden ist. Der Benutzer kann die Stapelverarbeitung jederzeit unterbrechen, indem er eine beliebige Taste betätigt. JADOS fragt dann:

STAPELVERARBEITUNG ABBRECHEN (J/N) ?

Durch Eingabe von "J" wird die Stapelverarbeitung komplett abgebrochen, ansonsten wird sie korrekt fortgeführt.

Eine Kommandodatei darf Kommandos mit formalen Parametern enthalten. Diese formalen Parameter heißen X1, X2, X3 und X4. Beim Aufruf einer Kommandodatei können bis zu 4 Parameter mitgegeben werden, die dann die formalen Parameter ersetzen. Dadurch wird eine hohe Flexibilität erreicht.

Ein Beispiel soll dies verdeutlichen:

Eine Textdatei (z.B. mit dem Namen dateixyz) soll viermal ausgedruckt werden. Man könnte nun viermal das Kommando PRINT dateixyz über die Tastatur eingeben und dazwischen jedesmal den Drucker auf eine neue Seite einstellen.

Eine Kommandodatei erfüllt dies wesentlich eleganter. Um die Kommandodatei allgemeingültig zu halten, d.h. auf Textdateien unterschiedlichen Namens anwendbar zu machen, muß die Kommandodatei mit einem formalen Parameter arbeiten. Die folgende Kommandosequenz erfüllt alle Anforderungen:

```
FF
PRINT Z1
FF
PRINT Z1
FF
PRINT Z1
FF
PRINT Z1
FF
FF
FF
```

Angenommen, die Kommandodatei habe den Namen DRUCKE4.BAT. Wird nun DRUCKE4 dateixyz eingegeben, so ersetzt JADOS den formalen Parameter Z1 durch den aktuellen Parameter dateixyz.

Soll nun eine Textdatei mit dem Namen dateiabc viermal ausgedruckt werden, so muß man lediglich DRUCKE4 dateiabc eingeben. Die Kommandodatei konnte dabei unverändert übernommen werden. Damit hat sich der kleine Arbeitsaufwand, die Kommandodatei zu erstellen, bereits gelohnt!!

Die folgende Kommandodatei, mit deren Hilfe eine Datensicherung aller Dateien, die zur Entwicklung von JADOS dienen, ausgeführt wird, möge abschließend als Beispiel einer nützlichen Anwendung der Stapelverarbeitung dienen:

```
bell
rem Quelle-in-1: und-Ziel-in-2:
pause
copy 1:dosmain.asm 2:dosmain.200
copy 1:dostabs.asm 2:dostabs.200
copy 1:dostool.asm 2:dostool.200
copy 1:dosfile.asm 2:dosfile.200
copy 1:dosbatch.asm 2:dosbatch.200
copy 1:dosgrund.asm 2:dosgrund.200
copy 1:doslink.ans 2:doslink.ans
copy 1:dossave.bat 2:dossave.bat
copy 1:dosinit.asm 2:dosinit.200
copy 1:dosinit.68k 2:dosinit.68k
copy 1:jados200.68k 2:jados200.68k
bell
rem Ende des Backups
```

Nach Aufruf der Kommandodatei "DOSSAVE" ertönt ein akustisches Signal. Danach fordert JADOS den Benutzer auf, die Quellendiskette in Laufwerk 1 und die Zieldiskette in Laufwerk 2 einzulegen sowie die Leertaste zu betätigen. Danach werden alle Dateien, die zur JADOS-Entwicklung dienen von Laufwerk 1 auf Laufwerk 2 kopiert.

Der 1. große Vorteil von Kommandodateien ist, daß sie immer wieder verwendet werden können. Eine Kommandofolge muß nur einmal erstellt werden. Dadurch erspart man sehr viel "Tipparbeit". Der 2. große Vorteil ist, daß die Kommandos automatisch abgearbeitet werden. In dieser Zeit braucht der Benutzer nicht unbedingt am Computer zu sitzen. Der 3. große Vorteil ist, daß man sich über die Kommandoabfolge keine Gedanken mehr machen muß. Dies hat man ja bei der Erstellung der Kommandodatei bereits getan.

## 6) BESCHREIBUNG DER UNTERPROGRAMME

In JADOS sind 51 Unterprogramme für den Programmierer verfügbar. Bei geladenem Betriebssystem können Anwendungsprogramme diese Unterprogramme mit dem Befehl TRAP #6 aufrufen. In Register D7 muß die Nummer des Unterprogramms übergeben werden.

### ACHTUNG !!

Der Variablenbereich von JADOS bezieht sich auf das Register A6. Dieses wird beim Aufruf eines JADOS-Unterprogramms automatisch auf den richtigen Wert gebracht. Der ursprüngliche Wert von A6 geht dabei verloren.

### 6.1) NUMERISCHE ÜBERSICHT

| Nummer | Name     | Register           |
|--------|----------|--------------------|
| 0      | getleng  | A0 -> D1           |
| 1      | getname  | A0, A1 -> D0.B     |
| 2      | getstadd | A2 -> A0           |
| 3      | lese     | D2, A1 -> D0.B     |
| 4      | loadtext | A2 -> D0.B         |
| 5      | motoroff |                    |
| 6      | response |                    |
| 7      | schreibe | A0                 |
| 8      | strgcomp | A0, A1 -> D0.B     |
| 9      | tload    | A0, A1 -> D0.B, A0 |
| 10     | tsave    | A0, A1 -> D0.B     |
| 11     | uppercas | A0                 |
| 12     | wrblank  | D1                 |
| 13     | wrint    | D0                 |
| 14     | close    | A1                 |
| 15     | copyfile | A0, A1, A2 -> D0.B |
| 16     | create   | A1 -> D0.B         |
| 17     | erase    | A1 -> D0.B         |
| 18     | fillfcb  | A0, A1 -> D0.B     |
| 19     | open     | A1 -> D0.B         |
| 20     | readrec  | A1 -> D0.B         |
| 21     | rename   | A1, A2 -> D0.B     |
| 22     | setdta   | A0, A1             |
| 23     | writerec | A1 -> D0.B         |
| 24     | getversi | -> D0.L            |
| 25     | getparm  | -> A0              |
| 26     | hardcopy |                    |
| 27     | bell     |                    |
| 28     | beep     | D1.L, D2.L         |
| 29     | errnoise |                    |
| 30     | sound    | A0                 |
| 31     | inport   | D1.L -> D0.B       |
| 32     | outport  | D0.B; D1.L         |
| 33     | movebin  | D1, A1, A2         |
| 34     | movetxt  | A1, A2             |
| 35     | moverbin | D1, A1, A2         |

| Nummer | Name     | Register               |
|--------|----------|------------------------|
| 36     | wrtcmd   | A0                     |
| 37     | gtretcod | -> D0.L                |
| 38     | stretcod | D0.L                   |
| 39     | moveline | D2, A0, A2 -> D0.B, A0 |
| 40     | wraddr   | D0.L                   |
| 41     | ci       | -> D0.B                |
| 42     | getparm1 | -> A0                  |
| 43     | getparm2 | -> A0                  |
| 44     | fileload | A0, A1 -> D0.B         |
| 45     | filesave | D1, A0, A1 -> D0.B     |
| 46     | getparm3 | -> A0                  |
| 47     | getparm4 | -> A0                  |
| 48     | loadpart | A0, A1 -> D0.B, D2     |
| 49     | savepart | D2, A0, A1 -> D0.B     |
| 50     | catalog  | A0                     |

## 6.2) ALPHABETISCHE ÜBERSICHT

| Name     | Nummer | Kurzbeschreibung                                                                                           |
|----------|--------|------------------------------------------------------------------------------------------------------------|
| beep     | 28     | Erzeugt Ton mit wählbarer Frequenz und Dauer (Soundbaugruppe !)                                            |
| bell     | 27     | Erzeugt Glockenton (Soundbaugruppe !)                                                                      |
| catalog  | 50     | Schreibt Disketteninhaltsverzeichnis über die CO2-Schnittstelle                                            |
| ci       | 41     | Liest ASCII-Wert von Tastatur; Ctrl P erzeugt Text-Hardcopy vom Bildschirm                                 |
| close    | 14     | Schließt eine geöffnete Datei                                                                              |
| copyfile | 15     | Kopiert eine Datei                                                                                         |
| create   | 16     | Legt eine Datei an                                                                                         |
| erase    | 17     | Löscht eine Datei                                                                                          |
| errnoise | 29     | Erzeugt Fehlerton (Soundbaugruppe !)                                                                       |
| fileload | 44     | Liest komplette Datei von Diskette                                                                         |
| filesave | 45     | Schreibt komplette Datei auf Diskette                                                                      |
| fillfcb  | 18     | Trägt Dateinamen in Dateisteuerblock ein                                                                   |
| getleng  | 0      | Ermittelt Länge eines Textes                                                                               |
| getname  | 1      | Erfragt Dateinamen und trägt diesen in den Dateisteuerblock ein                                            |
| getparm  | 25     | Liefert Parameter in der Kommandozeile                                                                     |
| getparm1 | 42     | Liefert 1. Parameter der Kommandozeile                                                                     |
| getparm2 | 43     | Liefert 2. Parameter der Kommandozeile                                                                     |
| getparm3 | 46     | Liefert 3. Parameter der Kommandozeile                                                                     |
| getparm4 | 47     | Liefert 4. Parameter der Kommandozeile                                                                     |
| getstadd | 2      | Erfragt Startadresse zum Laden/Speichern in/von Hauptspeicher von/auf Diskette                             |
| getversi | 24     | Informiert über die Version von JADDS                                                                      |
| gtretcod | 37     | Liest 4-Byte Usercode                                                                                      |
| hardcopy | 26     | Druckt Bildschirminhalt (nur Text)                                                                         |
| inport   | 31     | Liest Byte von Port (gilt für alle CPUs der 68000-Familie)                                                 |
| lese     | 3      | Liest einen Textstring von der Tastatur                                                                    |
| loadpart | 48     | Liest Teil einer Datei von Diskette                                                                        |
| loadtext | 4      | Lädt menügesteuert Texte von Diskette in den Hauptspeicher, wobei die Texte hintereinander abgelegt werden |

| <u>Name</u> | <u>Nummer</u> | <u>Kurzbeschreibung</u>                                             |
|-------------|---------------|---------------------------------------------------------------------|
| motoroff    | 5             | Schaltet Diskettenmotoren ab                                        |
| movelbin    | 33            | Verschiebt binären Speicherbereich von links                        |
| moveline    | 39            | Verschiebt Textzeile in Speicherpuffer                              |
| moveltxt    | 34            | Verschiebt Textbereich von links                                    |
| movebin     | 35            | Verschiebt binären Speicherbereich von rechts                       |
| open        | 19            | Öffnet eine Datei                                                   |
| outport     | 32            | Gibt Byte auf Port aus (gilt für alle CPUs der 68000-Familie)       |
| readrec     | 20            | Liest einen Dateisektor                                             |
| rename      | 21            | Benennt eine Datei um                                               |
| response    | 6             | Wartet auf Tastendruck                                              |
| savepart    | 49            | Schreibt Teil einer Datei auf Diskette                              |
| schreibe    | 7             | Schreibt einen Text über die C02-Schnittstelle                      |
| setdta      | 22            | Verändert die aktuelle Lade/Speicher - Transfer-Adresse             |
| sound       | 30            | Steuert Soundbaugruppe auf Adresse \$FFFFFF50 an                    |
| stretcod    | 38            | Setzt 4-Byte Usercode ab                                            |
| strgcomp    | 8             | Vergleicht zwei Textstrings                                         |
| tload       | 9             | Lädt eine komplette Textdatei                                       |
| tsave       | 10            | Speichert eine komplette Textdatei                                  |
| uppercas    | 11            | Wandelt alle Kleinbuchstaben in einem Text in Großbuchstaben        |
| wraddr      | 40            | Schreibt eine 24-Bit Adresse hexadezimal über die C02-Schnittstelle |
| wrblank     | 12            | Schreibt eine Anzahl Leerzeichen über die C02-Schnittstelle         |
| wrint       | 13            | Schreibt eine 16-Bit Dezimalzahl über die C02-Schnittstelle         |
| writerec    | 23            | Schreibt einen Dateisektor                                          |
| wrtcmd      | 36            | Schreibt JADOS-Kommando in Kommandopuffer                           |

### 6.3) BESCHREIBUNG DER EINZELNEN UNTERPROGRAMME

#### 6.3.1) getleng 0

Hiermit kann die Länge eines Textes ermittelt werden.

| <u>Parameter:</u> | <u>Register</u> | <u>Einq./Ausq.</u> | <u>Bedeutung</u>              |
|-------------------|-----------------|--------------------|-------------------------------|
|                   | A0.l            | x                  | Startadresse des Textes       |
|                   | D1.w            | x                  | Anzahl der Zeichen des Textes |

**Hinweis:** Das Textende muß durch eine binäre Null gekennzeichnet sein. Die Länge des Textes sollte nicht größer als 65535 Zeichen sein, da D1 nur als Wortregister verwendet wird. Es wird nur Register D1 verändert.

```

Beispiel : * TEXTLANGENERMITTLUNG
*
START:
LEA TEXT(PC),A0
MOVE #0,D7
TRAP #6
RTS

TEXT: DC.B 'Hallo',0
END

```

Im Beispiel wird die Länge des Textes 'Hallo' ermittelt. Das Ergebnis 5 steht nach dem TRAP-Aufruf in Register D1.

### 6.3.2) **getname** 1

Hiermit steht dem Programmierer eine Funktion zur Bedienerführung zur Verfügung. Es wird ein Text ausgegeben, der vom Bediener verlangt, einen Dateinamen einzugeben. Der Bediener kann nun einen bis zu 14 Zeichen umfassenden Namen eingeben. Die Eingabe kann mit CTRL A oder CTRL C abgebrochen werden. Ein solcher Abbruch wird im Register D0 mit dem Wert #FF angezeigt. Nach korrekter Eingabe werden alle Klein- in Großbuchstaben umgewandelt. Anschließend wird ein Dateisteuerblock initialisiert und mit dem eingegebenen Dateinamen ausgefüllt. Falls hierbei ein Fehler festgestellt wird, so wird ein Fehlertext ausgegeben und der Wert #FF in das Register D0 geschrieben.

| Parameter: | Register | Eing./Ausg. | Bedeutung                                                                                   |
|------------|----------|-------------|---------------------------------------------------------------------------------------------|
|            | A0.1     | x           | Adresse des Aufforderungstextes                                                             |
|            | A1.1     | x           | Adresse eines Dateisteuerblocks                                                             |
|            | D0.b     | x           | Fehlercode:<br>00 - kein Fehler<br>#FF - Name falsch oder Abbruch<br>mit CTRL A oder CTRL C |

Hinweis: Alle Ausgaben werden über die CO2-Schnittstelle getätigt. Es wird nur Register D0 verändert.

```

Beispiel: * DATEINAMEN ERFRAGEN
*
START:
LEA TEXT(PC),A0
LEA USERFCB(PC),A1
MOVE #1,D7
TRAP #6
RTS

TEXT: DC.B 'Bitte Dateinamen eingeben: ',0
 DS 0
USERFCB: DS.B 48
END

```

Im Beispiel wird der Text 'Bitte Dateinamen eingeben: ' ausgegeben. Der Cursor blinkt neben dem Text, und der Benutzer kann nun einen Dateinamen eingeben. Der Dateisteuerblock *USERFCB* enthält nach dem TRAP-Aufruf den eingegebenen Dateinamen.

6.3.3) **getstadd 2**

Hiermit steht dem Programmierer eine Funktion zur Bedienerführung zur Verfügung. Es wird ein Text ausgegeben, der vom Bediener verlangt, eine Startadresse einzugeben. Der Bediener kann nun einen bis zu 10 Zeichen umfassenden Wert eingeben. Die Eingabe kann mit CTRL A oder CTRL C abgebrochen werden. Ein solcher Abbruch oder wenn der Wert kleiner ist als ein vorgegebener Wert führt dazu, daß als Startadresse der vorgegebene Wert zurückgegeben wird.

| Parameter: | Register | Eing./Ausg. | Bedeutung                |
|------------|----------|-------------|--------------------------|
|            | A0.1     | x           | Eingegebene Startadresse |
|            | A2.1     | x           | Vorgegebene Startadresse |

Hinweis: Alle Ausgaben werden über die CO2-Schnittstelle getätigt. Es wird nur Register A0 verändert.

```

Beispiel: * STARTADRESSE ERFRAGEN
 *
 START:
 LEA $C000,A2
 MOVE #2,D7
 TRAP #6
 RTS
 END

```

Im Beispiel wird der Text 'WELCHE STARTADRESSE ? (DEFAULT = \$00C000)' ausgegeben.

Der Cursor blinkt neben dem Text, und der Benutzer kann nun einen Wert eingeben. Dies kann eine Zahl oder ein Symbol sein. Falls die die Eingabe nicht abgebrochen wurde und der Wert nicht kleiner als \$C000 ist, steht nach dem TRAP-Aufruf im Register A0 der eingegebene Wert.

6.3.4) **lese 3**

Diese Funktion dient zum Einlesen eines Textes. Mit der RETURN-Taste wird die Eingabe korrekt abgeschlossen. Mit der <- Taste oder der DEL-Taste kann die Eingabe korrigiert werden.

Die Tastenkombination Ctrl J gibt den Textpufferinhalt ab der Cursorposition aus.

Mit CTRL A oder CTRL C kann die Eingabe abgebrochen werden. Dies erzeugt einen Fehlercode.

| Parameter: | Register | Eing./Ausg. | Bedeutung                                                                              |
|------------|----------|-------------|----------------------------------------------------------------------------------------|
|            | A1.1     | x           | Adresse des Textpuffers                                                                |
|            | D2.w     | x           | Zeichenzahl                                                                            |
|            | D0.b     | x           | Zustandscode:<br>00 - Eingabe ok<br>01 - Abbruch mit CTRL A<br>03 - Abbruch mit CTRL C |

Hinweis: Alle Ausgaben werden über die CO2-Schnittstelle getätigt. Alle Steuerzeichen (\$00 - \$1F) werden ignoriert außer CTRL A, CTRL C, CTRL H (<-), CTRL M (RETURN) und CTRL J (LINEFEED). Es wird nur Register D0 verändert.

Die Funktion CTRL J steht seit Version 2.0 zur Verfügung.

```

Beispiel: * TEXTZEILE EINLESEN
 *
 START:
 LEA TEXTBUF(PC),A1
 MOVE #10,D2
 MOVE #3,D7
 TRAP #6
 RTS

 TEXTBUF: DS.B 11
 END

```

Im Beispiel kann ein bis zu 10 Zeichen umfassender Text eingegeben werden. Dieser Text steht nach dem TRAP-Aufruf in *TEXTBUF*. Dabei wird das Textende automatisch mit \$00 markiert. Deshalb muß der Textbuffer auch um ein Zeichen größer sein als die vorgegebene Textlänge.

### 6.3.5) loadtext 4

Dies ist eine sehr komplexe Funktion zum benutzergeführten Laden von Textmodulen von Diskette in den Speicher. Dies geschieht menügeführt. Die Startadresse des ersten Textes kann vom Benutzer eingegeben werden, wobei der Programmierer eine Mindeststartadresse vorgegeben kann.

- \* Seit Version 2.0 steht auch ein automatischer Textlademechanismus zur Verfügung. Hierbei wird der Name einer Textdatei eingegeben, die
- \* wiederum die Namen einer oder mehrerer Textdateien enthält, die
- \* letztendlich geladen werden sollen.
- \* Neu ist auch die Ausgabe der jeweiligen Endadresse.

| Parameter: | Register | Eing./Ausg. | Bedeutung                                                         |
|------------|----------|-------------|-------------------------------------------------------------------|
|            | A2.1     | *           | Mindeststartadresse                                               |
|            | D0.b     | x           | Zustandscode:                                                     |
|            |          |             | 00 - Texte geladen                                                |
|            |          |             | 01 - Funktion abgebrochen, ohne daß ein Text geladen wurde        |
|            |          |             | \$FF - Fehler beim Laden von Diskette (z.B. Datei nicht gefunden) |

Hinweis: Alle Ausgaben werden über die CO2-Schnittstelle getätigt. Der geladene Text kann im Grundprogramm bearbeitet werden. Z.B. können mehrere Programmodule zu einem Quellprogramm zusammengefaßt und komplett assembliert werden. Dies geht besonders bequem mit der neuen automatischen Textladefunktion !!  
Es werden keine Register außer D0 verändert.

Beispiel: \* MEHRERE TEXTDATEIEN EINLESEN UND DRUCKEN

```

*
START:
 LEA $C000,A2
 MOVE #4,D7 * loadtext
 TRAP #6
 CMP.B #0,D0
 BNE.S FEHLER

 JSR @GETSTX
 MOVEA.L D0,A0
 JSR @LST
 MOVE #7,D7 * schreibe
 TRAP #6
FEHLER:
 RTS

```

Im Beispiel können mehrere Textdateien von Diskette in den Hauptspeicher geladen werden. Nach dem Laden wird der Fehlercode abgefragt. Ist dieser Null, so wird der komplette Text über die Druckerschnittstelle ausgegeben.

### 6.3.6) motoroff 5

Hiermit können die Laufwerksmotoren abgeschaltet werden.

Parameter: keine !

Beispiel : \* MOTOREN ABSCHALTEN

```

*
START:
 MOVE #5,D7
 TRAP #6
 RTS
END

```

Im Beispiel werden die Motoren der angeschlossenen Laufwerke abgeschaltet.

A) A N H A N GA.1) DIE HILFSPROGRAMME VON JADOS

Zum Lieferumfang von JADOS gehören zur Zeit folgende Hilfsprogramme, die als .68K - Dateien auf der Systemdiskette gespeichert sind.

A.1.1) Übersicht

Die folgende Übersicht informiert kurz über die bereitgestellten Hilfsprogramme:

| <u>Dateiname</u> | <u>Bedeutung</u>                                                                                                         |
|------------------|--------------------------------------------------------------------------------------------------------------------------|
| ICOPY            | Kopierprogramm für Systeme mit nur einem Laufwerk                                                                        |
| BRDRUCK          | Programm zum Ausdrucken von Briefen und Dokumentationen unter Ausnutzung der Steuerodes des Standarddruckers EPSON RX 80 |
| DSAVE            | Speichern eines Bereichs des Hauptspeichers auf Diskette                                                                 |
| FORMAT           | Formatierprogramm                                                                                                        |
| MORE             | Programm zur Anzeige von Textdateien mit superschneller Bildschirmausgabe                                                |
| ROMSTART         | Programm zum Auffinden und Starten von Bibliotheksprogrammen, die im RAM oder im EPROM gespeichert sind                  |

A.1.2) 1COPY

Zweck: Programm zum Kopieren von Dateien für Systeme mit  
nur einem Laufwerk  
Aufruf: 1COPY  
Codeumfang: 1016 Byte  
Startadresse: relocativ

## Bedienungsablauf:

Das Programm meldet sich mit seinem Namen, der Versionsnummer und dem Copyright.

```

Kopierprogramm fuer EIN-LAUFWERK-SYSTEME
(C) 1986 by Klaus Janßen, Version 2.0

```

Anschließend erscheint folgendes Menü:

```
1 - KOPIEREN
9 - beenden
```

Mit Eingabe einer "1" oder Return wird der Kopiervorgang eingeleitet, mit Eingabe einer "9" wird das Programm beendet.

Ein normaler fehlerfreier Kopiervorgang läuft folgendermaßen ab. Erst wird nach dem Namen der Quelldatei, dann nach der Zieldatei gefragt:

```
Quelldateinamen eingeben:
Zieldateinamen eingeben:
```

Sind die Laufwerksangaben unterschiedlich, dann verhält sich 1COPY genauso wie das interne Kommando COPY. Sind die Laufwerksangaben dagegen gleich, fordert 1COPY dazu auf, die Quelldiskette einzulegen:

```
Quell-Diskette einlegen -->
LEERTASTE DRÜCKEN
```

Anschließend wird die Quelldatei komplett in den Hauptspeicher geladen. Danach kommt die Aufforderung, die Zieldiskette einzulegen:

```
Ziel-Diskette einlegen -->
LEERTASTE DRÜCKEN
```

Nach dem Kopieren erscheint in der Regel eine Erfolgsmeldung:

```
Datei kopiert
```

Danach erscheint erneut das obige Menü.

Folgende Fehlersituationen werden erkannt und gemeldet:

- Quelldatei nicht gefunden
- Diskette ist voll
- Inhaltsverzeichnis ist voll
- undefinierter Diskettenfehler
- falscher Dateiname

A.1.3) BRDRUCK

Zweck: Druckprogramm, welches die SteuerCodes des EPSON RX 80 Druckers ausnutzt  
 Aufruf: BRDRUCK  
 Codeumfang: 2362 Byte  
 Startadresse: relocativ

Bedienung: Nach dem Programmstart meldet sich folgender Bildschirm:

-----  
 DRUCKPROGRAMM FÜR EPSON RX80 . VERSION 2.0 (C) 1985 BY KLAUS JANGEN

## O P T I O N E N

## A K T I O N E N

|                  |              |
|------------------|--------------|
| Seitenzahlen : N | Text laden : |
| Startseite : 1   | Drucken :    |
| Seitenlänge : 65 | Beenden :    |

## MELDUNGEN:

## I N F O

Anwählen eines Feldes mit:  
 ^E Feld nach oben - ^X Feld nach unten  
 ^S Feld nach links - ^D Feld nach rechts  
 ^C Drucken abbrechen  
 Auslösen einer Aktion mit: RETURN (EINGABE)

-----  
 Ein blinkender Cursor zeigt den gerade aktuellen Menüpunkt an. Mit ^S und ^D kann zwischen OPTIONEN und AKTIONEN gewählt werden und mit ^E und ^X innerhalb von OPTIONEN und AKTIONEN.  
 Durch die Betätigung der RETURN - Taste wird der angewählte Menüpunkt aktiv. Bei "Seitenzahlen", "Startseite", "Seitenlänge" und "Text laden" erscheint dann die Meldung "EINGABE MACHEN", während bei "Drucken" der Druckvorgang beginnt (vorausgesetzt, es wurde bereits ein Text geladen !) und bei "Beenden" das Druckprogramm verlassen wird.  
 Bei "Text laden" muß ein Dateiname eingegeben werden.  
 Bei "Seitenzahlen" kann man "J" für JA oder "N" für NEIN eingeben.  
 Bei "Startseite" kann man eingeben, mit welcher Zahl die Seitennummerierung beginnen soll.  
 Bei "Seitenlänge" kann man eingeben, wieviel Zeilen eine Seite haben soll. Es wird dann jeweils ein Seitenvorschub erzeugt.

Wenn man z.B einen einfachen Brief ausdrucken will, so betätigt man nach dem Programmstart die RETURN - Taste.  
 Dann gibt man den Dateinamen ein. Nach dem Laden des Textes bewegt man den Cursor mit ^X auf "Drucken" und betätigt erneut die RETURN - Taste. Nach dem Drucken bewegt man den Cursor mit ^X auf "Beenden" und ist dann wieder im Kommandomodus von JADOS.

WICHTIG !! Wenn man den Druckvorgang unterbrechen will, so muß man ^C eintippen.

Steuercodes: Im Editor können nur ASCII-Zeichen zwischen \$20 und \$7E eingegeben werden. Zur Druckersteuerung wird hier eine Zeichenkombination, bestehend aus "#" und einem Steuerbuchstaben, verwendet. Die folgende Tabelle zeigt die verwendbaren Steuerbuchstaben und die entsprechenden Druckeraktionen:

| <u>Steuerbuchstaben</u> | <u>Druckeraktion</u>           |
|-------------------------|--------------------------------|
| B                       | Breitschrift EIN               |
| b                       | Breitschrift AUS               |
| C                       | Kursivschrift EIN              |
| c                       | Kursivschrift AUS              |
| D                       | Doppeldruck EIN                |
| d                       | Doppeldruck AUS                |
| E                       | Elite - Schriftart             |
| F                       | Fettdruck EIN                  |
| f                       | Fettdruck AUS                  |
| H                       | Halbe Druckgeschwindigkeit EIN |
| h                       | Halbe Druckgeschwindigkeit AUS |
| I                       | Indizierung EIN                |
| i                       | Indizierung AUS                |
| N                       | Normal - Schriftart (PICA)     |
| P                       | Potenzierung EIN               |
| p                       | Potenzierung AUS               |
| R                       | Ein Zeichen zurück             |
| S                       | Schwaalschrift EIN             |
| s                       | Schwaalschrift AUS             |
| U                       | Unterstreichen EIN             |
| u                       | Unterstreichen AUS             |
| V                       | Seitenvorschub                 |

Kombinationen sind ebenfalls möglich.  
Alle Steuercodes beziehen sich auf den Drucker RX 80 von EPSON oder dazu kompatibler Drucker.

A.1.4) DSAVE

Zweck: Programm zum Abspeichern eines Speicherbereichs  
(z.B. im EPROM) auf Diskette  
 Aufruf: DSAVE  
 Codeumfang: 0860 Byte  
 Startadresse: relocativ

## Bedienungsablauf:

Das Programm meldet sich mit seinem Namen, der Versionsnummer und dem Copyright.

```

Data Save Version 2.0
(C) 1986 by Klaus Janßen

```

## a) mit Parametern:

1. Parameter: Name der Datei
2. Parameter: Startadresse
3. Parameter: Anzahl KByte

Nach dem Speichern meldet DSAVE:

Abgespeichert !!

Beispiel für einen Aufruf mit Parametern:

```
DSAVE GRUND43.ROM $E0000 32
```

## b) ohne Parameter:

Erst wird nach dem Dateinamen gefragt:

Dateiname eingeben

Danach wird die Startadresse festgelegt:

WELCHE STARTADRESSE ? (DEFAULT = \$000000)

Daran anschließend muß noch festgelegt werden, wieviel KByte abgespeichert werden sollen:

wieviel KByte speichern ?

Ist auch dies festgelegt, wird der so definierte Speicherbereich auf die Diskette geschrieben. DSAVE meldet dann:

Abgespeichert !!

Folgende Fehlermeldungen können auftauchen:

```
Kein Platz auf Diskette
Diskettenfehler beim Schreibzugriff
Fehler bei der Namenseingabe
Anzahl KByte muß zwischen 1 und 780 sein
```

A.1.5) FORMAT

Zweck: Programm zum Formatieren von Disketten  
 Aufruf: FORMAT  
 Codeumfang: 1862 Byte  
 Speicherbedarf: 16 KByte freien Benutzerspeicher  
 Startadresse: relativ

## Bedienungsablauf:

Nach dem Programmstart erscheint wie üblich das Copyright:

```

Formatierprogramm V 2.0 - 68008/00/20
fuer Standardformat NDR80
(C) 1986 by Klaus Janßen

```

Danach wird dazu aufgefordert, die alte Diskette zu entnehmen und das Laufwerk auszuwählen:

```
Alte Diskette bitte entnehmen
Laufwerk auswählen
1 - Laufwerk 1
2 - Laufwerk 2
9 - Beenden
```

Nach Wahl des Laufwerks wird die Laufwerksnummer bestätigt und dann der Benutzer aufgefordert, eine neue Diskette in das gewählte Laufwerk einzulegen:

Bitte neue Diskette einlegen

Formatieren starten (J/N) ?

Mit Eingabe von "J" für JA beginnt der Formatiervorgang, ansonsten erscheint wieder das Laufwerkmenü:

Der Formatiervorgang wird durch Anzeige der aktuellen Spurnummer dokumentiert:

Diskette wird formatiert -- Spur: XX

Die Spurnummer beginnt mit 00 und endet mit 79. Es werden beide Seiten formatiert.

An das Formatieren schließt sich ein Prüflösevorgang an:

Diskette wird geprüft -- Spur: XX

Wurde beim Prüflösen kein Fehler entdeckt, so erscheint die Meldung

Formatierung ok

Im Fehlerfall können folgende Meldungen erscheinen:

Fehler beim Formatieren  
 Laufwerksfehler  
 Zu wenig Speicher  
 Fehler beim Prüflauf entdeckt

Nach einem solchen Fehlerfall erscheint erneut das Eingangsmenü.

#### A.1.6) MORE

Zweck: Programm zur superschnellen Anzeige von Textdateien  
 Aufruf: MORE  
 Codeumfang: 1136 Byte  
 Startadresse: relokativ

Bedienungsablauf:

a) mit Parameter:

1. Parameter: Name der Textdatei

b) ohne Parameter:

Der Benutzer wird aufgefordert:

Bitte Dateinamen eingeben:

Danach wird die Textdatei in den Hauptspeicher geladen und die erste Seite Text angezeigt. Die Textausgabe ist "superschnell", kann aber keine deutschen Sonderzeichen ausgeben.

Mit jeder beliebigen Taste kann die Anzeige der nächsten Seite (etwa 20 Zeilen) erreicht werden. Am unteren Ende des Bildschirms erscheint jeweils eine Meldung, ob die Anfangsseite, die Schlußseite oder eine beliebige Seite dazwischen angezeigt wird. Mit der Taste "-" kann man eine halbe Seite (10 Zeilen) zurück !!

Wenn am unteren Bildschirmrand \*\*\* Ende \*\*\* angezeigt wird, so kann man mit dem nächsten Tastendruck wieder nach JADOS zurück.

A.1.7) ROMSTART

Zweck: Programm zum Suchen und Starten von Bibliotheksprogrammen, die im Hauptspeicher stehen  
 Aufruf: ROMSTART  
 Codeumfang: 0890 Byte  
 Startadresse: relokativ

## Bedienungsablauf:

Nach dem Programmstart wird folgender Bildschirm angezeigt:

-----  
 \*\*\*\*\*  
 BIBLIOTHEKSFUNKTION VERSION 2.0  
 (C) COPYRIGHT 1986 BY KLAUS JANSEN  
 \*\*\*\*\*

- 1 - Suchen rückwärts
- 2 - SUCHEN VORWARTS
- 3 - Starten
- 9 - Beenden

| NAME  | LÄNGE  | ADRESSE |
|-------|--------|---------|
| JADOS | 002EE0 | 027020  |

-----  
 Die Suche nach Bibliotheksprogrammen erfolgt wie im Grundprogramm in Schritten von 1 KByte.

Mit dem Menüpunkt 1 wird die Suche von höheren nach niedrigeren Adressen durchgeführt; mit dem Menüpunkt 2 oder RETURN von niedrigeren nach höheren Adressen.

Mit Punkt 3 wird das unter NAME angezeigte Programm gestartet.  
 Mit Punkt 9 kann die Bibliotheksfunktion beendet werden.

## A.2) ABHILFE BEI FEHLERMELDUNGEN

### - ABRUCH WEGEN LESEFEHLER

-> Im Kommando ASS und TLOAD erscheint diese Meldung, falls beim menügesteuerten Laden von Texten ein Fehler auftritt.

Abhilfe: Neu versuchen !

### - ACHTUNG !! DIESER TEXT IST GRÖßER ALS 64 KBYTE ER KANN NICHT MIT DEM RDK-EDITOR BEARBEITET WERDEN

-> Im Kommando EDIT erscheint diese Meldung, falls Textdateien bearbeitet werden sollen, die größer als 64 KByte sind.

Abhilfe: Textdatei mit TLOAD laden und im Grundprogramm durch Eingabe einer sedezialen Null an geeigneter Stelle verkleinern.

### - BIBLIOTHEKSPROGRAMM NICHT RELOKATIV

-> Beim Laden eines ausführbaren Programms mit Bibliotheksvorspann erscheint diese Meldung, falls das Programm als nicht relokativ gekennzeichnet ist.

Abhilfe: Programm relokativ schreiben und dies im Bibliotheksvorspann kennzeichnen.  
Programm mit TLOAD auf die absolute Adresse laden, die im Bibliotheksvorspann steht und im Grundprogramm starten.

### - DATEINAME EXISTIERT BEREITS

-> Im Kommando REN erscheint diese Meldung, falls der neue Name bereits existiert.

Abhilfe: Datei mit neuem Namen umbenennen oder löschen.

### - DATEI NICHT GEFUNDEN

-> Bei vielen Kommandos, die eine Datei bearbeiten, erscheint diese Meldung, falls die Datei nicht existiert.

Abhilfe: Eine andere Diskette nehmen und den Versuch wiederholen.

### - DISKETTENFEHLER ODER SCHREIBSCHUTZ GESETZT !

-> Falls ein Schreibzugriff auf eine schreibgeschützte Diskette stattfindet oder beim Lesen ein CRC- oder Datenfehler festgestellt wurde, erscheint diese Meldung.

Abhilfe: Schreibschutz entfernen oder Diskette überprüfen.

**- FALSCHER ORG-ANWEISUNGEN BENUTZT**

-> Im Kommando ASS erscheint diese Meldung, falls man ORG-Anweisungen benutzt.

Abhilfe: Alle ORG-Anweisungen aus der Programmquelle entfernen.

**- FALSCHES KOMMANDO**

-> Diese Meldung erscheint, wenn ein eingegebenes Kommando weder ein Internes ist, noch als .68K-, .COM- oder .BAT-Datei existiert.

Abhilfe: Eine andere Diskette einlegen und Kommando wiederholen.

**- FEHLER BEI DER NAMENSEINGABE**

-> Bei vielen Kommandos, die die Eingabe eines Dateinamens erfordern, erscheint diese Meldung. Meist wurde dann gar kein Name eingegeben oder eine falsche Laufwerksangabe gemacht.

Abhilfe: Namen korrekt eingeben.

**- FEHLER BEIM SPEICHERN DER PROGRAMMDATEI**

-> Diese Meldung erscheint im Kommando ASS, falls die Abspeicherung des Maschinencodes fehlschlägt.

Abhilfe: Diskette überprüfen.

**- KEIN PLATZ MEHR AUF DER DISKETTE**

-> Wenn die Diskette keine freien Spuren mehr enthält und eine neue Spur angelegt werden soll, dann erscheint diese Meldung.

Abhilfe: Eine neue Diskette nehmen oder nicht mehr gebrauchte Dateien löschen.

**- KEIN PLATZ MEHR IM INHALTSVERZEICHNIS**

-> Wenn die Diskette bereits 128 Dateieinträge enthält und eine neue Datei angelegt werden soll, dann erscheint diese Meldung.

Abhilfe: Eine neue Diskette nehmen oder nicht mehr gebrauchte Dateien löschen.

**- LADEFehler**

-> Falls im Kommando ASS beim Laden eines Quellprogramms ein Fehler auftritt, erscheint diese Meldung.

Abhilfe: Diskette prüfen.

- PARAMETERFEHLER

-> Falls das Kommando ASS mit einem ungültigen Parameter aufgerufen wird, erscheint diese Meldung.

Abhilfe: Parameter korrekt eingeben.

- SPEICHER VOLL

-> Diese Meldung erscheint immer dann, wenn eine Datei nicht mehr in den freien Benutzerspeicher paßt.

Abhilfe: Benutzerspeicher vergrößern oder Datei verkleinern.

- STAPELVERARBEITUNG ABBRECHEN (J/N) ?

-> Falls bei der Abarbeitung einer Kommandodatei eine Taste gedrückt wird, erscheint diese Meldung. Der Benutzer kann sich dann entscheiden, ob er die Stapelverarbeitung abbrechen oder fortsetzen will.

- SYSTEM WEGEN FEHLER NICHT ABGESPEICHERT

-> Diese Meldung tritt auf, wenn im Kommando SYS ein Diskettenfehler auftritt.

Abhilfe: Neue Diskette nehmen.

- TEXT WEGEN SCHREIBFEHLER NICHT GESPEICHERT

-> Diese Meldung erscheint im Kommando TSAVE, falls beim Speichern des Textes ein Fehler auftritt.

Abhilfe: Diskette prüfen.

- UMBENENNUNG NUR AUF GLEICHEM LAUFWERK !

-> Im Kommando REN erscheint diese Meldung, falls für den alten und neuen Namen unterschiedliche Laufwerksangaben gemacht werden.

Abhilfe: Gleiche Laufwerksangaben machen.

- UNDEFINIERTER DISKETTENFEHLER

-> Diese Meldung kann erscheinen, falls ein undefinierter Diskettenfehler auftritt.

Abhilfe: Diskette prüfen.

- UNDEFINIERTER LESEFEHLER

-> Diese Meldung kann erscheinen, falls beim Lesen von einer Diskette ein Fehler auftritt.

Abhilfe: Diskette prüfen.

- WEGEN STACKFEHLER BITTE MIT FLOPPYSTART BOOTEN !!

-> Diese Meldung erscheint, falls man versucht, JADOS nach einem Reset, einer Einzelschrittbearbeitung oder einer Fehlersituation mit der Bibliotheksfunktion zu starten.

Abhilfe: Systemdiskette einlegen und mit FLOPPY START booten.

- ZU WENIG RAM !! -> ABRUCH

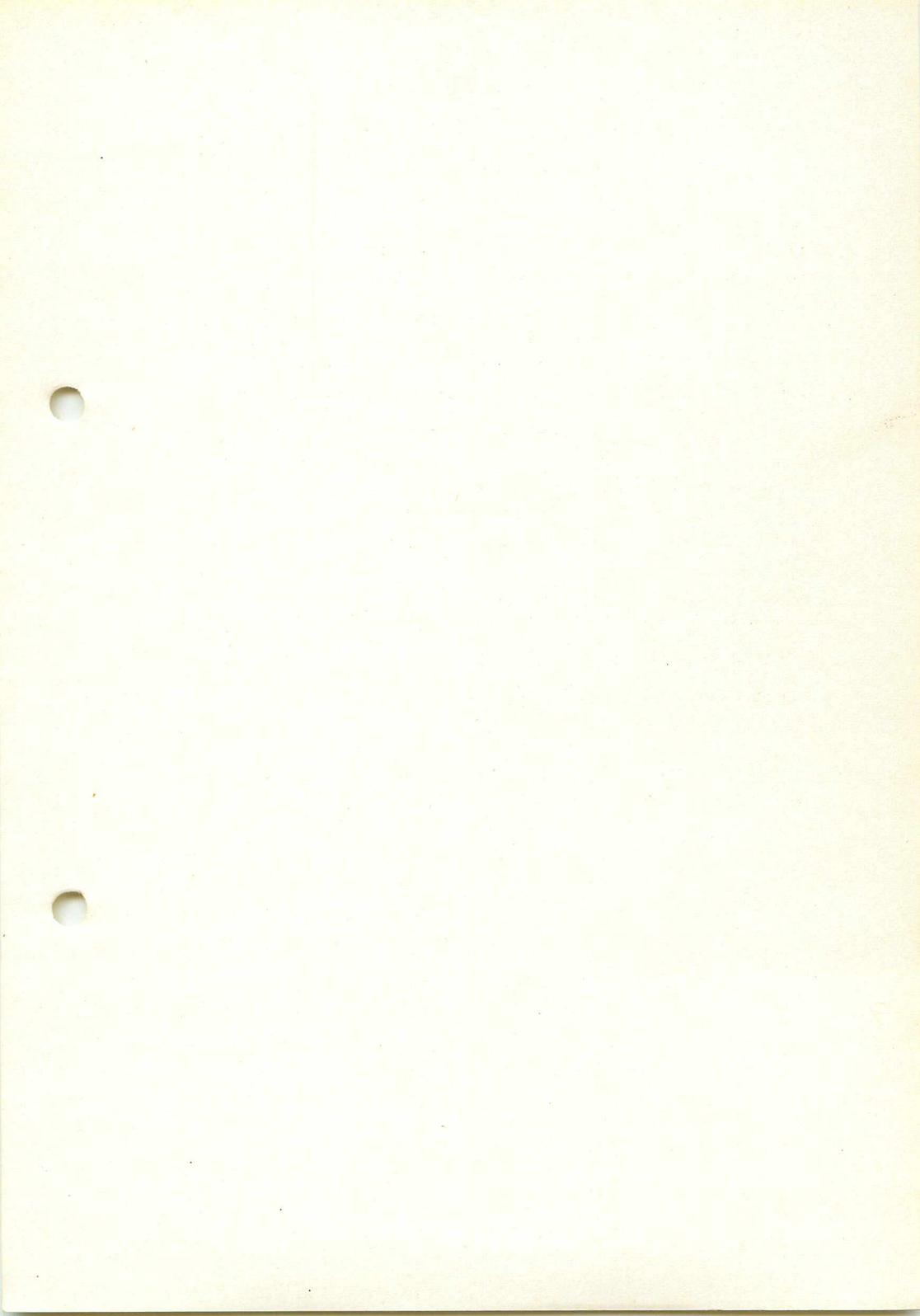
-> Das Kommando SYS benötigt 16 KByte Benutzerspeicher. Falls dieser nicht vorhanden ist, erscheint diese Meldung.

Abhilfe: Benutzerspeicher vergrößern.

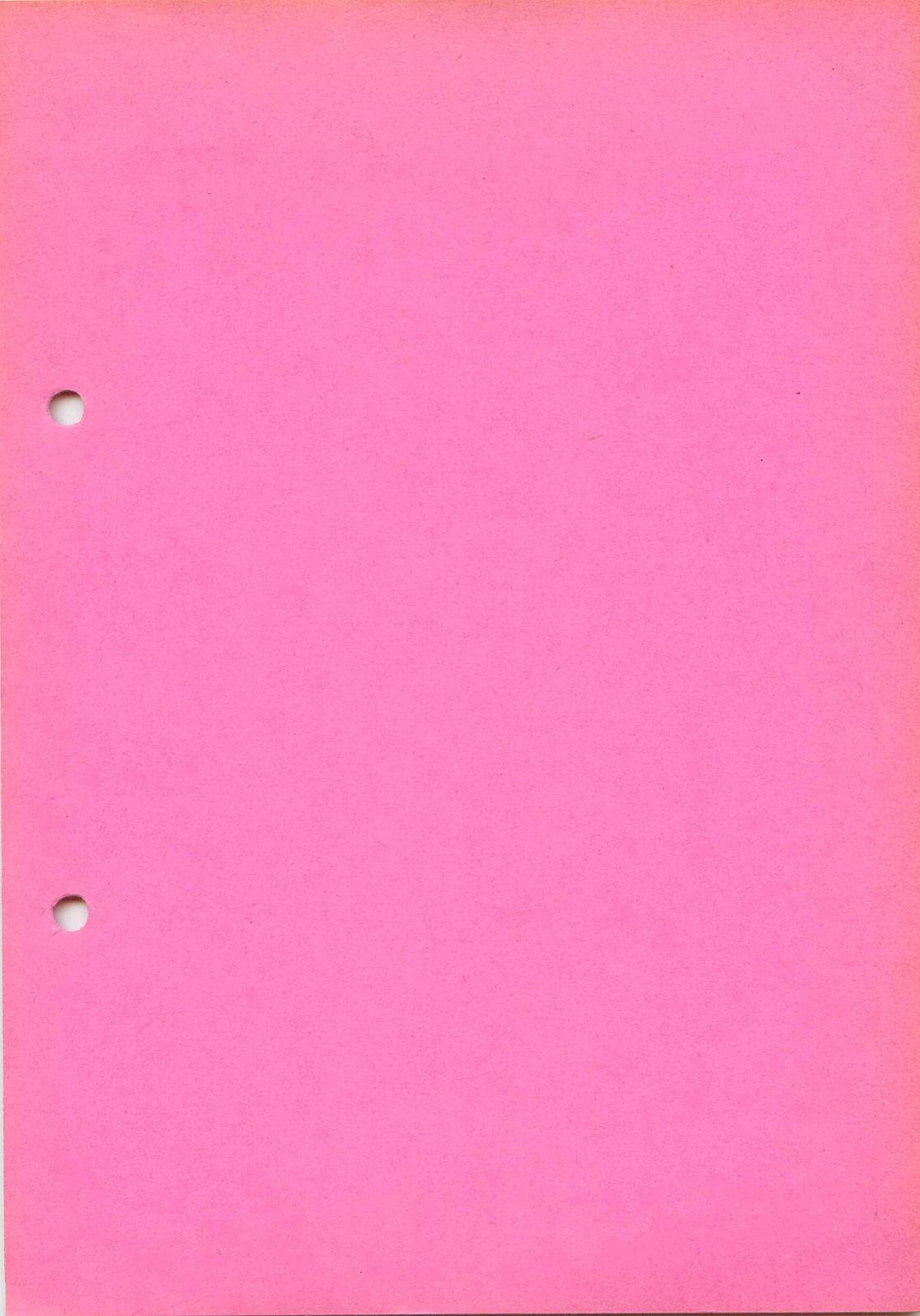
- ZU WENIG SPEICHER !!

-> Diese Meldung erscheint nach dem Start von JADOS, falls kein freier Benutzerspeicher vorhanden ist.

Abhilfe: Ersten zusammenhängenden Speicherbereich ausbauen.







**Neu!**



**Telefonservice**

**08 31- 62 11**

**jeden Mittwochabend  
bis 20.00 Uhr**

**Graf Elektronik Systeme GmbH**

Magnusstraße 13 · Postfach 1610  
8960 Kempten (Allgäu)  
Telefon: (08 31) 62 11  
Teletex: 831804 = GRAF  
Telex: 17 831804 = GRAF  
Datentelefon: (08 31) 6 93 30

**Verkauf:**

Computervilla  
Ludwigstraße 18 b  
(bei Möbel-Krügel)  
8960 Kempten-Sankt Mang  
Telefon: 08 31 / 6 93 00

**Geschäftszeiten: GES GmbH + Verkauf**

Mo. - Do. 8.00 - 12.00 Uhr, 13.00 - 17.00 Uhr  
Freitag 8.00 - 12.00 Uhr  
Telefonservice

**Filiale Hamburg**

Ehrenbergstraße 56  
2000 Hamburg 50  
Telefon: (0 40) 38 81 51

**Filiale München:**

Georgenstraße 61  
8000 München 40  
Telefon: (0 89) 2 71 58 58

**Öffnungszeiten der Filialen:**

Montag - Freitag  
10.00 - 12.00 Uhr, 13.00 - 18.00 Uhr  
Samstag 10.00 - 14.00 Uhr

**ges**