

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

```
K K SSSS PPPP RRRR 000 U U L
K K S P P R R 0 0 U U L
K K S P P R R 0 0 U U L
KKK SSS PPPP RRRR 0 0 U U L
K K S P R R 0 0 U U L
K K S P R R 0 0 U U L
K K SSSS P R R 000 UUUUU LLLLL
```

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

```
N N SSSS 999 1 000 H H 000 L EEEEE SSSS
N N t S 9 9 11 0 0 H H 0 0 L E S
NN N ooo ttt eee ## S 9 9 1 0 00 H H 0 0 L E S
N N N o o t e e e ## SSS 9999 1 0 0 0 ----- HHHH 0 0 L EEEE SSS
N NN o o t eeee S 9 1 00 0 H H 0 0 L E S
N N o o t e ## S 9 1 0 0 H H 0 0 L E S
N N ooo tt eeee ## SSSS 999 111 000 H H 000 LLLLL EEEEE SSSS
```

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:31:08 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

*
* * * * * * N O T I C E * * * * *
*
* ATTENTION!! NEW BIN NUMBERS AT HILL CENTER!
* -STARTS AUGUST 29- CHECK CAMPUS MAIL
* FOR DESCRIPTIVE MEMO OR PICK ONE UP AT HILL
* I/O COUNTER OR AID STATION-
* RESET BIN IN WYLBUR TO CONFORM!
*

DDDDDDDDDDDD		SSSSSSSSSSSS	KKK	KKK	FFFFFFFFFFFFFF	TTTTTTTTTTTTTT	PPPPPPPPPPPP
DDDDDDDDDDDD		SSSSSSSSSSSS	KKK	KKK	FFFFFFFFFFFFFF	TTTTTTTTTTTTTT	PPPPPPPPPPPP
DDDDDDDDDDDD		SSSSSSSSSSSS	KKK	KKK	FFFFFFFFFFFFFF	TTTTTTTTTTTTTT	PPPPPPPPPPPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDD	DDD	SSS	KKK	KKK	FFF	TTT	PPP PPP
DDDDDDDDDDDD	SSSSSSSSSSSS	TTTTTTTTTTTTTT			FFF	TTT	PPP
DDDDDDDDDDDD	SSSSSSSSSSSS	TTTTTTTTTTTTTT			FFF	TTT	PPP
DDDDDDDDDDDD	SSSSSSSSSSSS	TTTTTTTTTTTTTT			FFF	TTT	PPP

LLL		SSSSSSSSSSSS	TTTTTTTTTTTTTT			111
LLL		SSSSSSSSSSSS	TTTTTTTTTTTTTT			111
LLL		SSSSSSSSSSSS	TTTTTTTTTTTTTT			111
LLL	SSS		TTT			111111
LLL	SSS		TTT			111111
LLL	SSS		TTT			111111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLL	SSS		TTT			111
LLLLLLLLLLLLLLLL	SSSSSSSSSSSS	TTTTTTTTTTTTTT				111111111
LLLLLLLLLLLLLLLL	SSSSSSSSSSSS	TTTTTTTTTTTTTT				111111111
LLLLLLLLLLLLLLLL	SSSSSSSSSSSS	TTTTTTTTTTTTTT				111111111

1952-1953
 1954-1955
 1956-1957
 1958-1959
 1960-1961
 1962-1963
 1964-1965
 1966-1967
 1968-1969
 1970-1971
 1972-1973
 1974-1975
 1976-1977
 1978-1979
 1980-1981
 1982-1983
 1984-1985
 1986-1987
 1988-1989
 1990-1991
 1992-1993
 1994-1995
 1996-1997
 1998-1999
 2000-2001
 2002-2003
 2004-2005
 2006-2007
 2008-2009
 2010-2011
 2012-2013
 2014-2015
 2016-2017
 2018-2019
 2020-2021
 2022-2023
 2024-2025

Year	Value	Value	Value
1952	100	100	100
1953	100	100	100
1954	100	100	100
1955	100	100	100
1956	100	100	100
1957	100	100	100
1958	100	100	100
1959	100	100	100
1960	100	100	100
1961	100	100	100
1962	100	100	100
1963	100	100	100
1964	100	100	100
1965	100	100	100
1966	100	100	100
1967	100	100	100
1968	100	100	100
1969	100	100	100
1970	100	100	100
1971	100	100	100
1972	100	100	100
1973	100	100	100
1974	100	100	100
1975	100	100	100
1976	100	100	100
1977	100	100	100
1978	100	100	100
1979	100	100	100
1980	100	100	100
1981	100	100	100
1982	100	100	100
1983	100	100	100
1984	100	100	100
1985	100	100	100
1986	100	100	100
1987	100	100	100
1988	100	100	100
1989	100	100	100
1990	100	100	100
1991	100	100	100
1992	100	100	100
1993	100	100	100
1994	100	100	100
1995	100	100	100
1996	100	100	100
1997	100	100	100
1998	100	100	100
1999	100	100	100
2000	100	100	100
2001	100	100	100
2002	100	100	100
2003	100	100	100
2004	100	100	100
2005	100	100	100
2006	100	100	100
2007	100	100	100
2008	100	100	100
2009	100	100	100
2010	100	100	100
2011	100	100	100
2012	100	100	100
2013	100	100	100
2014	100	100	100
2015	100	100	100
2016	100	100	100
2017	100	100	100
2018	100	100	100
2019	100	100	100
2020	100	100	100
2021	100	100	100
2022	100	100	100
2023	100	100	100
2024	100	100	100
2025	100	100	100

Year	Value	Value	Value	Value
1952	100	100	100	100
1953	100	100	100	100
1954	100	100	100	100
1955	100	100	100	100
1956	100	100	100	100
1957	100	100	100	100
1958	100	100	100	100
1959	100	100	100	100
1960	100	100	100	100
1961	100	100	100	100
1962	100	100	100	100
1963	100	100	100	100
1964	100	100	100	100
1965	100	100	100	100
1966	100	100	100	100
1967	100	100	100	100
1968	100	100	100	100
1969	100	100	100	100
1970	100	100	100	100
1971	100	100	100	100
1972	100	100	100	100
1973	100	100	100	100
1974	100	100	100	100
1975	100	100	100	100
1976	100	100	100	100
1977	100	100	100	100
1978	100	100	100	100
1979	100	100	100	100
1980	100	100	100	100
1981	100	100	100	100
1982	100	100	100	100
1983	100	100	100	100
1984	100	100	100	100
1985	100	100	100	100
1986	100	100	100	100
1987	100	100	100	100
1988	100	100	100	100
1989	100	100	100	100
1990	100	100	100	100
1991	100	100	100	100
1992	100	100	100	100
1993	100	100	100	100
1994	100	100	100	100
1995	100	100	100	100
1996	100	100	100	100
1997	100	100	100	100
1998	100	100	100	100
1999	100	100	100	100
2000	100	100	100	100
2001	100	100	100	100
2002	100	100	100	100
2003	100	100	100	100
2004	100	100	100	100
2005	100	100	100	100
2006	100	100	100	100
2007	100	100	100	100
2008	100	100	100	100
2009	100	100	100	100
2010	100	100	100	100
2011	100	100	100	100
2012	100	100	100	100
2013	100	100	100	100
2014	100	100	100	100
2015	100	100	100	100
2016	100	100	100	100
2017	100	100	100	100
2018	100	100	100	100
2019	100	100	100	100
2020	100	100	100	100
2021	100	100	100	100
2022	100	100	100	100
2023	100	100	100	100
2024	100	100	100	100
2025	100	100	100	100

1	Documentation
49	Development History
114	Zero Page Storage
122	System Equates
182	Main Program
272	COMMAND PARSER
280	Command Table
335	Information Commands
475	Commands
574	Type-Print Commands
643	Memory Transfer Commands
681	File Transfer Command
701	Sub-Programs
767	Sub-Pgm Sector Read
918	Sub-Pgm Disk Duplicate
973	Control Transfer Commands
1038	OPEN Routines
1153	CLOSE Routine
1235	Main-Program Subroutines
1362	Sub-Pgm Subroutines
1383	Driver Tables
1456	Driver Routine Specifications
1495	Directory Subroutines
1528	CODOS Disk Drivers
1692	FLEX Disk Drivers
2096	CP/M Disk Drivers
2744	IBM Disk Drivers
3099	RSX Disk Drivers
3120	HDE Disk Drivers
3146	TRS-80 Disk Drivers
3281	UCSD Disk Drivers
3322	USER Disk Drivers
3373	CHROMATICS Disk Drivers
3720	N Device Drivers
3736	C Device Drivers
3787	P Device Drivers
3831	M Device Drivers
3892	U Device Drivers
3907	System I/O
4032	Disk Drivers
4084	Disk Drivers - Read Sector Routine
4218	Disk Drivers - Write Sector Routine
4342	Disk Drivers - Subroutines
4349	Disk Drivers - Command Tables
4401	Error Routine
4472	Text Strings
4513	Conversion Routines
4532	EBCDIC => ASCII Conversion Tables
4826	ASCII => EBCDIC Conversion Tables
4979	Variables & Buffers
5008	Non-Zero Page Storage

```
1 .SBTTL Documentation
2 .TITLE Disk File Transfer Program -- by Keith Sproul
3
4 ; Written by: Copyright by:
5 ; Keith A. Sproul Micro Technology Unlimited
6 ; 1368 Noah Road 2806 Hillsborough St.
7 ; North Brunswick, P.O. Box 12106
8 ; New Jersey 08902 Raleigh, North Carolina 27605
9 ; (201) 246-3749 (919) 833-1458
10 ; April 1980
11 ;
12 ;*****
13 ;* This program is being developed by Keith Sproul on the LCSR *
14 ;* DEC-20 at Rutgers University. It was written for specific micro- *
15 ;* computer hardware and so it is of very little use to anyone that *
16 ;* does not have that hardware. However, anyone wanting the data on *
17 ;* the different disk formats that went into writing this pgm are *
18 ;* welcome to it at copying charges. Send system or netmail to me *
19 ;* (KSproul@Rutgers) and I will give you a copy of all of the disk *
20 ;* formats and other information that I have put together in the *
21 ;* process of writing this program when the project is complete and *
22 ;* I have all of the information put together in one place. *
23 ;* All rights reserved. This program may not be copied, used or *
24 ;* re-distributed in any form, in whole or in part, without explicit *
25 ;* written permission of M.T.U. & Keith Sproul. *
26 ;*****
27 ; This program will read and write Disks from other
28 ; systems as long as the bottom level Disk Format is:
29 ;
30 ; 8 INCH FLOPPY (OBVIOUSLY)
31 ; SOFT SECTORED
32 ; SINGLE or DOUBLE DENSITY
33 ;
34 ; All Disks MUST be Initialized on their respective Host System.
35 ; Write protect all Disks unless actually wanting to write on them.
36 ; Good quality disks are strongly suggested since DSKFTP does not
37 ; do re-trys on some of the disk I/O. (Writing Especially)
38 ; This program was not optimized for speed in that it waits for each
39 ; disk command to complete before proceeding to the next step.
40 ; DSKFTP was designed to be completely in memory during use.
41 ; Maximum # of files 'OPEN' at one time by DSKFTP is 2.
42 ; As much stuff as possible is done in the general routines
43 ; before control is passed to the specific routines.
44 ; DSKFTP does not support double sided drives (yet).
45 ; Except for CODOS.
46
47
```

```
49          .SBTTL  Development History
50
51
52          ;      04/??/80      STARTED DSKFTP.
53          ;      07/26/80      STARTED DOING DETAILED HISTORY
54          ;      07/28/80      GOT IBM INPUT WORKING
55          ;      07/29/80      Added more detail to TRS-80 Directory.
56          ;      08/04/80      Added LPT: Drivers
57          ;      08/07/80      ADDED RETRYS TO DISK READ
58          ;
59          ;      08/11/80      Got IBM Write working.
60          ;      08/25/80      APEX => CODOS
61          ;      09/30/80      FLEX Tab Un-Compression Added
62          ;      10/10/80      GOT FLEX INPUT WORKING
63          ;      10/21/80      STARTED CP/M DIRECTORY PARSING
64          ;      10/24/80      ADDED ZEROING OUT THE REST OF FLEX FILES
65          ;      10/28/80      ADDED BOOT COMMAND
66          ;
67          ;      11/17/80      STARTED DUP COMMAND
68          ;      12/24/80      GOT CODOS WORKING
69          ;      12/29/80      DID MORE WORK ON CP/M
70          ;      01/03/81      GOT IBM EOD STUFF FIXED
71          ;
72          ;      01/05/81      * IN RALEIGH, N.C.
73          ;      01/06/81      CHANGED SYNTAX & SEPERATED DISK TYPES FROM DEVICES
74          ;      01/07/81      ADDED DEFINE, WORKED ON CP/M
75          ;      01/13/81      MOVED BUFFERS OUT OF SYSRAM
76          ;      01/16/81      IMPLIMENTED A LOT OF BRUCES STUFF
77          ;      01/17/81      PUT BUFFERS BACK INTO DMA RAM
78          ;      01/18/81      GOT DEVICES WORKING
79          ;      01/19/81      GOT CP/M MULTIPLE EXTENTS WORKING
80          ;
81          ;
82          ;      01/22/81      CHANGED INIT - STATUS,
83          ;
84          ;
85          ;      01/26/81      & ADDED SOME STUFF TO IT
86          ;      01/29/81      CP/M ^Z EOF ADDED
87          ;
88          ;
89          ;
90          ;      01/29/81      CP/M IN/OUT NAME BUFFERS ADDED
91          ;
92          ;      01/29/81      IBM NAME BUFFER ADDED
93          ;
94          ;
95          ;      01/26/81      ADDED WRITE RE-TRY
96          ;      01/29/81      ADDED READ/WRITE -RTRN
97          ;
98          ;
99          ;
100          ;      01/29/81      CLEANED UP SOME STUFF ON CP/M
101          ;
102          ;      01/29/81      CLEANED UP ERROR HANDLER
103          ;
104          ;
105          ;      01/29/81      ADDED DRIVE CMD
106          ;
107          ;      01/30/81      ADDED CPM TAB EXPANSION
108          ;
109          ;      31-JAN-81      ADDED ECHO CONTROL FOR CONSOLE
110          ;
111          ;      02-FEB-81      RE-DID SECTOR READ/WRITE
112          ;
113          ;      02-FEB-81      RE-DID DENSITY CMD
114          ;
115          ;
116          ;      03-FEB-81      ADDED MORE INFO TO SIZE CMD
117          ;
118          ;      04-FEB-81      MOVED STORAGE OUT OF ZERO PAGE
119          ;
120          ;
121          ;      04-FEB-81      ADDED CONDITIONAL EQUATES
122          ;
123          ;
124          ;      03-MAR-81      SENT COPY TO MTU
125          ;
126          ;      16-MAR-81      ADDED FSIZE CMD
127          ;
128          ;
129          ;      03-MAR-81      GOT COMMENTS FROM BRUCE/MTU
130          ;
131          ;      16-MAR-81      ADDED MEMBOT/MEMTOP CMDS
132          ;
133          ;
134          ;      03-MAR-81      NAME CHANGED TO 'DISKEX'
135          ;
136          ;      23-MAR-81      CHANGED DRIVE CMD A LITTLE
137          ;
138          ;
139          ;      23-MAR-81      ADDED LINE-FEED SUPPRESSION OPTION
```

104	:	24-MAR-81	SENT COPY TO BRUCE, ALONG WITH SOME TEST DISKS
105	:	31-MAR-81	WORKED ON LF OPTION AGAIN
106	:	31-MAY-81	STARTED WORK ON CP/M AGAIN
107	:	25-JUL-81	BRUCE WANTS PASCAL FORMAT
108	:	02-SEP-81	STARTED GETTING READY FOR SENDING TO MTU
109	:	13-SEP-81	U.C.S.D. PASCAL
110	:	23-SEP-81	DID MOST OF CONVERTING TO MTU-ASM
111	:	11-JUL-82	STARTED ADDING CHROMATICS DISK FORMAT
112	:		


```
114 .SBTTL Zero Page Storage
115
116      0000      . =      $0000      ; START AT LOCATION 0
117
118 0000 0002      MEMPTR: . = .+2      ; MEMORY POINTER
119 0002 0004      PRTPTR: . = .+2      ; PRINT POINTER (TEMP POINTER)
120
```

```

122          .SBTTL  System Equates
123
124
125          .LIST   CND
126
127          0001    KSV      =      1          ; KSPROUL VERSION
128          0000    KIM      =      0          ; KIM VERSION
129          0000    AIM      =      0          ; AIM VERSION
130          0000    MTU      =      0          ; MTU-100 VERSION
131
132          01      .IFEQ    KSV-1          ; K-SPROUL VERSION
133          .IFT
134          C000    SYSRAM   =      $C000
135          8000    USRRAM   =      $8000
136          1000    ORG      =      $1000
137          00      .ENDC
138
139          01      .IFEQ    KIM-1          ; STANDARD KIM VERSION
140          .IFT
141          SYSRAM   =      $C000
142          USRRAM   =      $6000
143          ORG      =      $2000
144          00      .ENDC
145
146          01      .IFEQ    AIM-1          ; STANDARD AIM VERSION
147          .IFT
148          SYSRAM   =      $8000
149          USRRAM   =      $4000
150          ORG      =      $0200
151          00      .ENDC
152
153          01      .IFEQ    MTU-1          ; MTU-100 VERSION
154          .IFT
155          SYSRAM   =      $E000
156          USRRAM   =      $C000
157          ORG      =      $1000
158          00      .ENDC
159
160          ; DISK BUFFER(S)
161
162          8000     . =      USRRAM          ; (COULD BE 256 BYTES)
163          8000 8200 DIRBUF: . = .+512      ; DIRECTORY/TEMP BUFFER
164          8200 8400 INPBUF: . = .+512     ; INPUT BUFFER
165          8400 8600 OUTBUF: . = .+512    ; OUTPUT BUFFER
166
167          8600     USREND:
168

```

```

170
171          .MCALL DSKEQU, SVCDEF, MACDEF
172          .LIST ME
173          ;          .READ DSKEUQ.A
174 8600          DSKEQU
(1)          ;***** FLOPPY DISK CONTROLLER REGISTER ADDRESSES *****
(1)
(1)          C603          CODOS =          SYSRAM+$0603          ; CODOS WARM START ADDRESS
(1)          DFOO          DSKIPL =          SYSRAM+$1F00          ; DISK BOOT ENTRY POINT
(1)          DFE8          FDCIRQ =          SYSRAM+$1FE8          ; K-1013 INTERRUPT REG.          (READ-ONLY)
(1)          DFE8          FDCHWC =          SYSRAM+$1FE8          ; DMA MODE (B0) WRITE PROT (B1) (WRITE-ONLY)
(1)          DFEA          FDCDMA =          SYSRAM+$1FEA          ; FDC DMA ADDRESS REGISTER
(1)          DFEE          FDCMSR =          SYSRAM+$1FEE          ; FDC NEC-765 MAIN STATUS REGISTER
(1)          DFEF          FDCDR =          SYSRAM+$1FEF          ; FDC NEC-765 DATA REGISTER
(1)
(1)          ;*****
175          ;          .READ SVCDEF.A
176 8600          SVCDEF
(1)
(1)          .MCALL SVC, SVCBYT
(1)          OOBO          .=          $OOBO
(1)          OOBO 0001          UO:          .BLKW 1          ; NUMERIC VALUES
(1)          OOB2 0001          U1:          .BLKW 1          ; ADDRESSES
(1)          OOB4 0001          U2:          .BLKW 1          ; ADDRESSES / SIZE
(1)          OOB6 0001          U3:          .BLKW 1          ; FILE NAME POINTER
(1)          OOB8 0001          U4:          .BLKW 1          ;
(1)          OOBA 0001          U5:          .BLKW 1          ; POINT TO START OF INPUT LINE BUFFER
(1)          OOBc 0001          U6:          .BLKW 1          ; POINT TO START OF OUTPUT LINE BUFFER
(1)          OOBE 0003          U7:          .BLKB 3          ; 24 BIT FILE ORDINAL POINTER
(1)          OOEE          SVCENA =          $OOEE          ; MUST BE SET TO $80 TO ENABLE SVCS
(1)
177          ;          .NLIST ME
178          .NLIST BEX
179
180

```

```

182          .SBTTL  Main Program
183
184          1000          . =      ORG          ; (NORMALLY $2000)
185
186          ;          .READ  DISKEX_MAIN.A
187
188 1000  4C  47  10  START:  JMP  COLD          ; COLD START VECTOR
189 1003  4C  D0  10  JWARM:  JMP  WARM          ; WARM START VECTOR
190 1006  4C  06  C6  SYSIN:  JMP  SYSRAM+$0606    ; CODOS CONSOLE TTY INPUT
191 1009  A9  20          CTYSPA: LDA  #$20          ; PRINT a <space>
192 100B  EA          ;          NOP          ; JUST FALL THROUGH
193 100C  4C  09  C6  SYSOUT: JMP  SYSRAM+$0609    ; CODOS CONSOLE TTY OUTPUT
194 100F  4C  0C  C6  KEYDWN: JMP  SYSRAM+$060C    ; CODOS CHECK IF KEY DOWN
195          ;          ;          ; IF NO KEYDWN ROUTINE,
196          ;          ;          ; REPLACE WITH
197          ;          ; LDA  #$80  A9 80
198          ;          ; RTS      60
199 1012  4C  2A  FO  PRINTR: JMP  $F02A          ; PRINTER OUTPUT ROUTINE
200
201 1015  10          DMPBPL: .BYTE  16          ; DUMP BYTES PER LINE
202
203 1016  FF          ECHO:   .BYTE  $FF          ; ZERO = NO ECHO
204          ;          ;          ; OTHER = ECHO
205 1017  FF          NOLFCR: .BYTE  $FF          ; ZERO = NO LF AFTER CR
206          ;          ;          ; OTHER = LF
207
208
209 1018  OD  OA          .BYTE  $OD,$OA          ; INCASE THE FILE IS 'TYPED'
210 101A  43  6F  70  .ASCIZ  'Copyright by Keith A. Sproul & M.T.U. 1981'
211 1045  OD  OA          .BYTE  $OD,$OA

```

```

213
214
215 1047 A2 FF          COLD:  LDX    #$FF          ; COLD START
216 1049 9A            TXS                    ; INIT STACK POINTER
217 104A D8            CLD
218 104B 78            SEI
219 104C A9 02        LDA    #$02          ; BIT 1
220 104E 8D E8 DF     STA    FDCHWC       ; WRITE PROTECT SYSRAM
221 1051 A2 32        LDX    #TTLMS1^    ; TITLE & COPYRIGHT
222 1053 A0 1E        LDY    #TTLMS1&$FF
223 1055 20 AF 2B     JSR    PRTSTR
224 1058 20 E0 2B     JSR    CRLF
225 105B A2 32        LDX    #TTLMS2^
226 105D A0 2E        LDY    #TTLMS2&$FF
227 105F 20 AF 2B     JSR    PRTSTR
228 1062 20 E0 2B     JSR    CRLF
229 1065 A2 32        LDX    #TTLMS3^
230 1067 A0 44        LDY    #TTLMS3&$FF
231 1069 20 AF 2B     JSR    PRTSTR
232 106C 20 2C 2E     JSR    DSKINI      ; INIT the FDC & SYSTEM
233 106F A9 00        LDA    #0          ; CHECK DRIVE 0
234 1071 20 47 2C     JSR    CHKDRV
235 1074 A9 01        LDA    #1          ; CHECK DRIVE 1
236 1076 20 47 2C     JSR    CHKDRV
237 1079 A9 02        LDA    #2          ; CHECK DRIVE 2
238 107B 20 47 2C     JSR    CHKDRV
239 107E A9 03        LDA    #3          ; CHECK DRIVE 3
240 1080 20 47 2C     JSR    CHKDRV
241 1083 20 E0 2B     JSR    CRLF        ; NEW LINE
242 1086 A2 32        LDX    #DI.MSG^
243 1088 A0 6D        LDY    #DI.MSG&$FF
244 108A 20 AF 2B     JSR    PRTSTR
245 108D AD C8 35     LDA    DRIVEI
246 1090 09 30        ORA    #$30        ; MAKE INTO ASCII #
247 1092 20 95 2B     JSR    CTYOUT
248 1095 A2 32        LDX    #DO.MSG^
249 1097 A0 85        LDY    #DO.MSG&$FF
250 1099 20 AF 2B     JSR    PRTSTR
251 109C AD C9 35     LDA    DRIVED
252 109F 09 30        ORA    #$30        ; MAKE INTO ASCII #
253 10A1 20 95 2B     JSR    CTYOUT
254 10A4 20 E0 2B     JSR    CRLF
255 10A7 A2 32        LDX    #BEGMSG^    ; BEGGININGM OF MEMORY MSG
256 10A9 A0 9D        LDY    #BEGMSG&$FF
257 10AB 20 AF 2B     JSR    PRTSTR
258 10AE AD BA 35     LDA    BEGMEM+1    ; HIGH BYTE
259 10B1 20 C3 2B     JSR    PRTBYT
260 10B4 AD B9 35     LDA    BEGMEM+0    ; LOW BYTE
261 10B7 20 C3 2B     JSR    PRTBYT
262 10BA A2 32        LDX    #ENDMSG^    ; END OF MEMORY MSG
263 10BC A0 C3        LDY    #ENDMSG&$FF
264 10BE 20 AF 2B     JSR    PRTSTR
265 10C1 AD BC 35     LDA    ENDMEM+1    ; HIGH BYTE
266 10C4 20 C3 2B     JSR    PRTBYT
267 10C7 AD BB 35     LDA    ENDMEM+0    ; LOW YTE
    
```

268 10CA 20 C3 2B
269 10CD 20 EO 2B
270

JSR PRTBYT
JSR CRLF

```

272          .SBTTL  COMMAND PARSER
273
274          ;          .READ  CMDPRS.A
275          .MCALL  CMDPRS
276  10D0          CMDPRS
(1)
(1)  10D0  A2  FF          WARM:  LDX    #$FF          ; WARM START
(1)  10D2  9A          TXS          ; INIT STACK POINTER
(1)  10D3  D8          CLD          ; INSURE BINARY
(1)  10D4  AD  E8  DF          LDA    FDCIRQ          ; CHECK IF INTERRUPT PENDING?
(1)  10D7  30  03          BMI    WRMST1          ; ACTIVE LOW
(1)  10D9  20  59  2E          JSR    SINTST          ; SENSE INTERRUPT STATUS
(1)
(1)  10DC  A9  00          WRMST1: LDA    #0          ; ZERO OUT INPUT BUFFER POINTER
(1)  10DE  8D  44  36          STA    BUFPTR
(1)  10E1  20  AB  2B          JSR    PROMPT          ; PRINT THE PROMPT
(1)  10E4  20  21  19          JSR    INSTRG          ; GET THE COMMAND LINE
(1)  10E7  B0  F3          BCS    WRMST1          ; IF ANYTHING WRONG, RE-DO
(1)  10E9  20  B3  19          JSR    GSTATOM          ; GET THE MAIN COMMAND
(1)  10EC  AD  00  38          LDA    ATMBUF+0          ; CHECK 1ST CHAR
(1)  10EF  C9  3B          CMP    #' ;
(1)  10F1  FO  E9          BEQ    WRMST1          ; IGNORE REST OF LINE IF COMMENT
(1)  10F3  A2  00          LDX    #0
(1)  10F5  AO  00          CMDLPO: LDY    #0          ; INIT THE INDEX POINTERS
(1)  10F7  BD  26  11          CMD.LP: LDA    CMDTAB,X
(1)  10FA  D9  00  38          CMP    ATMBUF,Y
(1)  10FD  DO  09          BNE    NXTCMD
(1)  10FF  E8          INX
(1)  1100  BD  26  11          LDA    CMDTAB,X          ; IF IT IS TRAILING NULL
(1)  1103  FO  18          BEQ    FNDCMD          ; THEN CMD MATCHES
(1)
(1)  1105  C8          INY
(1)  1106  DO  EF          BNE    CMD.LP          ; WILL ALWAYS BRANCH
(1)
(1)  1108  E8          NXTCMD: INX          ; SKIP OVER REST
(1)  1109  BD  26  11          LDA    CMDTAB,X          ; OF CURRENT COMMAND
(1)  110C  DO  FA          BNE    NXTCMD
(1)  110E  E8          INX          ; SKIP OVER NUL
(1)  110F  E8          INX          ; SKIP OVER ADDRESS
(1)  1110  E8          INX
(1)  1111  BD  26  11          LDA    CMDTAB,X          ; CHECK IF AT END OF TABLE
(1)  1114  DO  DF          BNE    CMDLPO          ; WILL BRANCH UNLESS AT END OF TABLE
(1)  1116  48          PHA          ; SO STACK WON'T WRAP AROUND
(1)  1117  48          PHA          ; IN ERROR ROUTINE
(1)  1118  A9  01          LDA    #CMDERR          ; COMMAND ERROR
(1)  111A  20  82  2F          JSR    ERROR
(1)
(1)  111D  BD  28  11          FNDCMD: LDA    CMDTAB+2,X          ; GET HIGH BYTE (ADDR-1)
(1)  1120  48          PHA
(1)  1121  BD  27  11          LDA    CMDTAB+1,X          ; GET LOW BYTE (ADDR-1)
(1)  1124  48          PHA
(1)  1125  60          RTS          ; RETURN TO THE APPROPRIATE ROUTINE
(1)
277
278

```

```

280          .SBTTL Command Table
281
282 1126 53 54 41  CMDTAB: .ASCIZ 'STAT'          ; STATUS (ALSO RE-INITs STUFF)
283 112B DD 11      .ADDR STATUS-1
284 112D 52 45 53  .ASCIZ 'RESULT'        ; PRINT THE FDC RESULTS
285 1134 ED 11      .ADDR PRESLT-1
286 1136 53 49 5A  .ASCIZ 'SIZE'          ; PRINT THE SIZE OF FILE IN MEMORY
287 113B 7E 12      .ADDR PSIZE-1
288 113D 4D 45 4D  .ASCIZ 'MEMBEG'        ; SET BEGGINING OF MEMORY
289 1144 51 12      .ADDR MEMBEG-1
290 1146 4D 45 4D  .ASCIZ 'MEMEND'        ; SET TOP OF MEMORY
291 114D 65 12      .ADDR MEMEND-1
292 114F 52 45 53  .ASCIZ 'RESET'        ; RESET the FDC
293 1155 5D 13      .ADDR RESET-1
294 1157 44 52 49  .ASCIZ 'DRIVE'        ; SET DEFAULT DRIVE #s
295 115D 27 13      .ADDR SDRIVE-1
296 115F 44 45 46  .ASCIZ 'DEF'          ; DEFINE DISK TYPE
297 1163 88 13      .ADDR DEFINE-1
298 1165 55 4E 44  .ASCIZ 'UND'          ; 'CLOSE' (UN-DEFINE) A DISK
299 1169 D4 13      .ADDR UNDFIN-1
300 116B 44 49 52  .ASCIZ 'DIR'          ; DIRECTORY
301 116F E6 13      .ADDR DIRECT-1
302 1171 54 59 50  .ASCIZ 'TYPE'        ; TYPE FILE
303 1176 37 14      .ADDR TYPE-1
304 1178 50 52 49  .ASCIZ 'PRINT'        ; PRINT FILE TO PRINTER
305 117E 4F 14      .ADDR PRINT-1
306 1180 46 53 49  .ASCIZ 'FSIZE'        ; DETERMINE THE SIZE OF A FILE
307 1186 67 14      .ADDR FSIZE-1
308 1188 52 45 41  .ASCIZ 'READ'          ; READ A FILE INTO MEMORY
309 118D B6 14      .ADDR READF-1
310 118F 57 52 49  .ASCIZ 'WRITE'        ; WRITE A FILE FROM MEMORY
311 1195 D3 14      .ADDR WRITEF-1
312 1197 43 4F 50  .ASCIZ 'COPY'          ; COPY FILE
313 119C F3 14      .ADDR COPYF-1
314 119E 44 45 4E  .ASCIZ 'DEN'          ; SET DENSITY PARMS
315 11A2 9E 15      .ADDR DENSTY-1
316 11A4 53 45 43  .ASCIZ 'SECTOR'        ; SECTOR READ SUB-PROGRAM
317 11AB E4 15      .ADDR SECTOR-1
318 11AD 4E 45 58  .ASCIZ 'NEXT'        ; READ THE 'NEXT' SECTOR
319 11B2 6D 16      .ADDR NEXT-1
320 11B4 44 55 4D  .ASCIZ 'DUMP'        ; HEX DUMP of MEMORY
321 11B9 11 15      .ADDR DUMP-1
322 11BB 44 55 50  .ASCIZ 'DUP'          ; DISK DUP SUB-PROGRAM
323 11BF DF 16      .ADDR D_DISK-1
324 11C1 47 4F 00  .ASCIZ 'GO'          ; JUMP TO ADDRESS
325 11C4 40 17      .ADDR GO-1
326 11C6 45 58 49  .ASCIZ 'EXIT'        ; EXIT TO SYSTEM MONITOR
327 11CB 4E 17      .ADDR EXIT-1
328 11CD 43 4F 44  .ASCIZ 'CODOS'        ; EXECUTE CODOS CMD OR EXIT TO CODOS
329 11D3 54 17      .ADDR ECODOS-1
330 11D5 42 4F 4F  .ASCIZ 'BOOT'        ; BOOT THE DISK SYSTEM
331 11DA 8B 17      .ADDR DBOOT-1
332 11DC 00 00      .ADDR O          ; END OF LIST
333

```



```

335 .SBTTL Information Commands
336
337 11DE AD EE DF STATUS: LDA FDCMSR ; CHECK Main Status Register
338 11E1 29 10 AND #$10 ; CHECK FDC BUSY BIT
339 11E3 FO 06 BEQ STATOK
340 11E5 EA NOP ; WAIT
341 11E6 EA NOP
342 11E7 EA NOP
343 11E8 AD EF DF LDA FDCDR ; TRY RESETTING THE FDC (WORKS IN SOME CASES)
344 11EB 4C 00 10 STATOK: JMP START ; GO BACK & RE-INIT EVERYTHING
345
346
347 ; .READ PRESLT.A
348 .MCALL PRESLT ; PRINT THE FDC RESULTS
349 11EE PRESLT
(1)
(1) 11EE 20 EO 2B PRESLT: JSR CRLF ; NEW LINE
(1) 11F1 A9 4D LDA #'M ; PRINT 'MSR='
(1) 11F3 20 95 2B JSR CTYOUT
(1) 11F6 A9 53 LDA #'S
(1) 11F8 20 95 2B JSR CTYOUT
(1) 11FB A9 52 LDA #'R
(1) 11FD 20 4A 12 JSR RSLEQU ; PRINT 'R='
(1) 1200 AD EE DF LDA FDCMSR ; GET THE MAIN STATUS REGISTER
(1) 1203 20 C3 2B JSR PRTBYT
(1) 1206 20 40 2C JSR SPACE2 ; 2 SPACES
(1) 1209 A0 03 LDY #3 ; 3 STATUS REGISTERS (0-2)
(1) 120B A2 00 LDX #0 ; INIT INDEX POINTER TO BEGINING OF DSKSTS
(1) 120D A9 53 RSLSTX: LDA #'S ; RESULT ST-X
(1) 120F 20 95 2B JSR CTYOUT
(1) 1212 A9 54 LDA #'T
(1) 1214 20 95 2B JSR CTYOUT
(1) 1217 A9 2D LDA #'-
(1) 1219 20 95 2B JSR CTYOUT
(1) 121C 8A TXA ; GET REG #
(1) 121D 09 30 ORA #$30 ; CONVERT TO ASCII NUMBER
(1) 121F 20 3C 12 JSR RSLDAT
(1) 1222 88 DEY
(1) 1223 DO E8 BNE RSLSTX ; DO ALL 3 REGS
(1)
(1) 1225 A9 43 LDA #'C
(1) 1227 20 3C 12 JSR RSLDAT
(1) 122A A9 48 LDA #'H
(1) 122C 20 3C 12 JSR RSLDAT
(1) 122F A9 52 LDA #'R
(1) 1231 20 3C 12 JSR RSLDAT
(1) 1234 A9 4E LDA #'N
(1) 1236 20 3C 12 JSR RSLDAT
(1) 1239 4C 03 10 JMP JWARM ; RETURN TO TOP LEVEL
(1)
(1) 123C 20 4A 12 RSLDAT: JSR RSLEQU ; RESULT PRINT DATA
(1) 123F BD 00 36 LDA DSKSTS,X
(1) 1242 20 C3 2B JSR PRTBYT
(1) 1245 20 40 2C JSR SPACE2
(1) 1248 E8 INX ; BUMP UP POINTER TO NEXT REGISTER
    
```

```

(1) 1249 60          RTS
(i)
(1) 124A 20 95 2B   RSLEQU: JSR   CTYOUT      ; RESULT PRT EQUAL
(1) 124D A9 3D      LDA   #' =
(1) 124F 4C 95 2B   JMP    CTYOUT
(i)
350
351 1252 20 06 1A   MEMBEG: JSR   GETNUM      ; GET THE ADDRESS
352 1255 B0 23      BCS   SMEMER
353 1257 AD 42 36   LDA   NUMBER+0      ; LOW BYTE
354 125A 8D B9 35   STA   BEGMEM+0
355 125D AD 43 36   LDA   NUMBER+1      ; HIGH BYTE
356 1260 8D BA 35   STA   BEGMEM+1
357 1263 4C 03 10   JMP    JWARM
358
359
360 1266 20 06 1A   MEMEND: JSR   GETNUM      ; GET THE ADDRESS
361 1269 B0 0F      BCS   SMEMER
362 126B AD 42 36   LDA   NUMBER+0      ; LOW BYTE
363 126E 8D BB 35   STA   ENDMEM+0
364 1271 AD 43 36   LDA   NUMBER+1      ; HIGH BYTE
365 1274 8D BC 35   STA   ENDMEM+1
366 1277 4C 03 10   JMP    JWARM
367
368
369 127A A9 15      SMEMER: LDA   #NOPARM
370 127C 20 82 2F   JSR   ERROR
371
372 127F AD B9 35   PSIZE:  LDA   BEGMEM+0      ; ADDRESS OF BOTTOM OF MEMORY
373 1282 85 00      STA   MEMPTR+0
374 1284 AD BA 35   LDA   BEGMEM+1
375 1287 85 01      STA   MEMPTR+1
376 1289 A0 00      LDY   #0
377 128B B1 00      PSIZLP: LDA   (MEMPTR),Y
378 128D C9 1A      CMP   #'Z&$3F      ; WILL BE FOUND EVENTUALLY (EVEN IF THIS ONE)
379 128F FO 09      BEQ   PSZEND      ; EXIT WHEN END FOUND
380 1291 E6 00      INC   MEMPTR
381 1293 DO F6      BNE   PSIZLP
382 1295 E6 01      INC   MEMPTR+1
383 1297 4C 8B 12   JMP    PSIZLP
384
385 129A 38          PSZEND: SEC          ; GET READY FOR SUBTRACT
386 129B A5 00      LDA   MEMPTR+0      ; LOW BYTE
387 129D ED B9 35   SBC   BEGMEM+0
388 12A0 8D 4C 36   STA   WORD+0      ; TEMP SAVE
389 12A3 A5 01      LDA   MEMPTR+1      ; HIGH BYTE
390 12A5 ED BA 35   SBC   BEGMEM+1
391 12A8 8D 4D 36   STA   WORD+1
392 12AB A2 32      LDX   #PSZMSG^      ; HIGH BYTE
393 12AD A0 E9      LDY   #PSZMSG&$FF  ; LOW BYTE
394 12AF 20 AF 2B   JSR   PRTSTR
395 12B2 AE 4D 36   LDX   WORD+1      ; PRINT NUMBER OF CHARS (IN HEX)
396 12B5 AC 4C 36   LDY   WORD+0
397 12B8 20 1A 13   JSR   PSZSBR
398 12BB 20 40 2C   JSR   SPACE2

```

```

399 12BE A9 28          LDA #'(
400 12C0 20 95 2B      JSR CTYOUT
401 12C3 AD 4D 36      LDA WORD+1          ; HIGH BYTE
402 12C6 4A           LSR A                ; / 2
403 12C7 4A           LSR A                ; / 4
404 12C8 AA           TAX                ; DO ROUNDING CORRECTLY
405 12C9 AD 4D 36      LDA WORD+1
406 12CC 29 03        AND #$03
407 12CE OD 4C 36      ORA WORD+0
408 12D1 FO 01        BEQ PSIZEK
409 12D3 E8           INX                ; BUMP IF OVER EVEN K
410 12D4 8A           PSIZEK: TXA
411 12D5 20 BA 1A      JSR HXTDCP          ; PRINT # of K (IN BCD)
412 12D8 20 09 10      JSR CTYSPA
413 12DB A9 4B        LDA #'K
414 12DD 20 95 2B      JSR CTYOUT
415 12E0 A9 29        LDA #' )
416 12E2 20 95 2B      JSR CTYOUT
417 12E5 20 3A 2C      JSR SPACE4
418 12E8 A9 40        LDA #'@
419 12EA 20 95 2B      JSR CTYOUT
420 12ED 20 09 10      JSR CTYSPA
421 12FO AE BA 35      LDX BEGMEM+1       ; BOTTOM OF BUFFER
422 12F3 AC B9 35      LDY BEGMEM+0
423 12F6 20 1A 13      JSR PSZSBR
424 12F9 20 09 10      JSR CTYSPA
425 12FC A9 2D        LDA #' -
426 12FE 20 95 2B      JSR CTYOUT
427 1301 20 09 10      JSR CTYSPA
428 1304 A6 01        LDX MEMPTR+1       ; END OF FILE
429 1306 A4 00        LDY MEMPTR+0
430 1308 20 1A 13      JSR PSZSBR
431 130B 20 40 2C      JSR SPACE2         ; COUPLE OF SPACES
432 130E AE BC 35      LDX ENDMEM+1       ; ADDRESS OF END OF BUFFER
433 1311 AC BB 35      LDY ENDMEM+0
434 1314 20 1A 13      JSR PSZSBR
435 1317 4C 03 10      JMP JWARM
436
437 131A A9 24         PSZSBR: LDA #' $          ; PRINT '$'
438 131C 20 95 2B      JSR CTYOUT
439 131F 8A           TXA                ; GET HIGH BYTE
440 1320 20 C3 2B      JSR PRTBYT
441 1323 98           TYA                ; GET LOW BYTE
442 1324 20 C3 2B      JSR PRTBYT
443 1327 60           RTS
444
445
446
447 1328             SDRIVE:                ; SET DEFAULT DRIVE #s
448 1328 20 06 1A      JSR GETNUM
449 132B BO 15         BCS MSDRNO         ; MISSING DRIVE #
450 132D C9 04        CMP #4              ; 0-3
451 132F BO 16         BGE SDRVER
452 1331 8D CA 35      STA DRVTMP         ; DRIVE TEMP SAVE
453 1334 20 B3 19      JSR GTATOM        ; GET 'I' OR 'O'

```

```
454 1337 AD 00 38 LDA ATMBUF+0
455 133A C9 49 CMP #'I ; INPUT ?
456 133C FO OE BEQ SDRVIN
457 133E C9 4F CMP #'0 ; OUTPUT ?
458 1340 FO 13 BEQ SDRVOU
459
460 1342 A9 15 MSDRNO: LDA #NOPARM ; MISSING PARAMETER
461 1344 20 82 2F JSR ERROR
462
463 1347 A9 03 SDRVER: LDA #DRVERR
464 1349 20 82 2F JSR ERROR
465
466 134C AD CA 35 SDRVIN: LDA DRVTMP ; RESTORE DRIVE #
467 134F 8D C8 35 STA DRIVEI
468 1352 4C 03 10 JMP JWARM
469
470 1355 AD CA 35 SDRVOU: LDA DRVTMP ; RESTORE DRIVE #
471 1358 8D C9 35 STA DRIVEO
472 135B 4C 03 10 JMP JWARM
473
```

```

475 .SBTTL Commands
476
477 ; .READ FDCRST.A
478 .MCALL FDCRST
479 135E FDCRST
(1)
(1) ; RESET THE FLOPPY DISK CONTROLLER CHIP &
(1) ; RECALIBRATE ALL ACTIVE DRIVES.
(1)
(1) 135E AD EE DF RESET: LDA FDCMSR ; CHECK MAIN STATUS REGISTRE
(1) 1361 10 FB BPL RESET ; WAIT UNTIL REQUEST FOR MASTER GOES TRUE
(1) 1363 29 40 AND #$40 ; CHECK DIO
(1) 1365 FO 05 BEQ RSTRCL ; IF NOT WANTING TO TALK
(1) 1367 EA NOP
(1) 1368 EA NOP
(1) 1369 20 E4 2E JSR RESULT ; ELSE LISTEN
(1)
(1) 136C A2 03 RSTRCL: LDX #3 ; FOUR DRIVES (0-3)
(1) ; RESET-RECAL
(1) 136E 8A RRCLLP: TXA ; PUT DRIVE # INTO ACC
(1) 136F 48 PHA ; SAVE X
(1) 1370 20 41 2E JSR SDRVST ; SENSE DRIVE STATUS
(1) 1373 68 PLA ; RESTORE CURRENT DRIVE #
(1) 1374 AA TAX
(1) 1375 AD 00 36 LDA DSKSTS+0 ; GET ST-3
(1) 1378 29 20 AND #$20 ; CHECK READY BIT
(1) 137A FO 07 BEQ RSTDNR ; SKIP OVER RECAL IF NOT READY
(1) 137C 8A TXA ; SAVE X & PUT IT IN ACC
(1) 137D 48 PHA
(1) 137E 20 61 2E JSR RECAL
(1) 1381 68 PLA ; RESTORE X
(1) 1382 AA TAX
(1) 1383 CA RSTDNR: DEX ; RESET DRIVE NOT READY
(1) 1384 10 E8 BPL RRCLLP ; DO ALL FOUR DRIVES
(1)
(1) 1386 4C 03 10 JMP JWARM ; RETURN TO TOP LEVEL
480
481
482 1389 20 06 1A DEFINE: JSR GETNUM
483 138C B0 3D BCS DEFPER ; PARM ERROR
484 138E C9 04 CMP #4
485 1390 B0 34 BGE DEFDER ; DRIVE ERROR
486 1392 8D 33 36 STA TMPDRV
487 1395 20 59 1A JSR DSK.NO ; DISK TYPE #
488 1398 AE 33 36 LDX TMPDRV ; GET THE DRIVE #
489 139B 9D C4 35 STA DRIVES,X ; USE AS OFF-SET
490 139E 0A ASL A ; * 2
491 139F AA TAX ; USE AS INDEX
492 13A0 BD 39 1B LDA DEFTAB+0,X
493 13A3 8D 30 36 STA TMPJMP+0
494 13A6 BD 3A 1B LDA DEFTAB+1,X
495 13A9 8D 31 36 STA TMPJMP+1
496 13AC BD 0B 1C LDA DBSTAB+0,X
497 13AF 8D 3A 36 STA TMPDEN
498 13B2 BD 0C 1C LDA DBSTAB+1,X
    
```

```

499 13B5 8D 39 36          STA  TMBYSC
500 13B8 AD 33 36          LDA  TMPDRV
501 13BB 20 61 2E          JSR  RECAL
502 13BE 20 B4 1A          JSR  EXETMP
503 13C1 B0 OD             BCS  DEF.ER
504 13C3 4C O3 10          JMP  JWARM
505
506 13C6 A9 O3             DEFDER: LDA  #DRVERR          ; ILLEGAL DRIVE #
507 13C8 20 82 2F          JSR  ERROR
508
509 13CB A9 15             DEFPER: LDA  #NOPARM         ; MISSING PARAMETER
510 13CD 20 82 2F          JSR  ERROR
511
512 13D0 A9 18             DEF.ER: LDA  #DEFERR
513 13D2 20 82 2F          JSR  ERROR
514
515
516
517 13D5 20 O6 1A          UNDFIN: JSR  GETNUM          ; GET DRIVE #
518 13D8 B0 F1             DEFPER  BCS  #4             ; PARM ERROR
519 13DA C9 O4             CMP   #4
520 13DC B0 E8             BGE  DEFDER          ; DRIVE ERROR
521 13DE AA                TAX
522 13DF A9 O0             LDA  #0             ; USE AS INDEX
523 13E1 9D C4 35          STA  DRIVES,X
524 13E4 4C O3 10          JMP  JWARM
525
526

```

```

528
529 13E7 20 06 1A   DIRECT: JSR   GETNUM       ; GET DRIVE #
530 13EA 8D 13 36   STA   INPDRV       ; DEFAULTS TO DRIVE 0
531 13ED C9 04      CMP   #4
532 13EF B0 3D      BGE   DIRDER       ; DRIVE ERROR
533 13F1 20 8B 1A   JSR   DSKTYP       ; CONVERT TO DISK TYPE
534 13F4 FO 3D      BEQ   DIRUND       ; ERROR IF ZERO
535 13F6 8D 12 36   STA   INPDEV       ; SAVE DEVICE #
536 13F9 OA          ASL   A             ; A <= A*2
537 13FA AA          TAX           ; USE AS INDEX
538 13FB BD 0B 1C   LDA   DBSTAB+0,X   ; GET BYTES/SECTOR
539 13FE 8D 19 36   STA   INBYSC
540 1401 BD 0C 1C   LDA   DBSTAB+1,X   ; GET DENSITY
541 1404 8D 1A 36   STA   INPDEN
542
543 1407 BD 57 1B   LDA   DIRTAB+0,X   ; GET LOW BYTE
544 140A 8D 40 36   STA   CMDJMP+0
545 140D BD 58 1B   LDA   DIRTAB+1,X   ; GET HIGH BYTE
546 1410 8D 41 36   STA   CMDJMP+1
547
548 1413 BD ED 1B   LDA   HDRTAB+0,X
549 1416 85 02      STA   PRTPTR+0     ; SET UP FOR PRINT STRING SUBR
550 1418 BD EE 1B   LDA   HDRTAB+1,X
551 141B 85 03      STA   PRTPTR+1
552 141D 20 B3 2B   JSR   PRTEXT       ; PRINT THE SYSTEM HEADER
553 1420 20 E0 2B   JSR   CRLF         ; ANOTHER NEW LINE
554 1423 A9 80      LDA   #DIRBUF^    ; SET UP BUFFER ADDRESS
555 1425 8D 1B 36   STA   INPAGE
556 1428 20 B7 1A   JSR   EXECMD       ; GO EXECUTE APPROPRIATE DIRECTORY CMD
557 142B 4C 03 10   JMP   JWARM        ; RETURN TO TOP LEVEL
558
559 142E A9 03      DIRDER: LDA   #DRVERR ; ILLEGAL DRIVE #
560 1430 20 82 2F   JSR   ERROR
561
562 1433 A9 19      DIRUND: LDA   #UNDFDT ; UNDEFINED DISK TYPE (DRIVE NOT DEFINED)
563 1435 20 82 2F   JSR   ERROR
564
565 ; ON TRANSFER:
566
567 ; INPDEV = DISK TYPE #
568 ; INPDRV = DRIVE #
569 ; INPAGE = ADDRESS OF DIRECTORY BUFFER
570 ; INBYSC = BYTES/SECTOR of DISK TYPE
571 ; INPDEN = DENSITY of DISK TYPE
572

```

```

574 .SBTTL Type-Print Commands
575
576 ; TYPE FILE
577 1438 20 BF 17 TYPE: JSR OPENFI ; OPEN FILE for INPUT
578 143B 20 91 2A JSR C.OPN ; OPEN CONSOLE (CRLF)
579 ; TYPE LOOP
580 143E 20 AE 1A TYPELPL: JSR EXEINP ; EXECUTE APPROPRIATE INPUT CHAR ROUTINE
581 1441 B0 OA BCS TYPEXT ; EXIT ON END OF FILE
582 1443 20 A1 2A JSR C.GUT ; OUTPUT CHAR TO CONSOLE (CRLFs & IGNORE LFs)
583 1446 90 F6 BCC TYPELPL ; WILL ALWAYS BRANCH (UNLESS ERROR)
584 1448 A9 OC LDA #FIOERR ; FILE I/O ERROR
585 144A 20 82 2F JSR ERROR
586
587 144D 4C O3 10 TYPEXT: JMP JWARM ; RETURN TO TOP LEVEL
588
589
590
591
592
593 ; PRINT FILE
594 1450 20 BF 17 PRINT: JSR OPENFI ; OPEN FILE for INPUT
595 1453 20 DE 2A JSR P.OPN ; OPEN PRINTER
596 ; PRINT LOOP
597 1456 20 AE 1A PRNTLPL: JSR EXEINP ; EXECUTE APPROPRIATE INPUT CHAR ROUTINE
598 1459 B0 OA BCS PRTEXT ; EXIT ON END OF FILE
599 145B 20 ED 2A JSR P.OUT ; OUTPUT CHAR TO PRINTER (CRLFs & IGNORE LFs)
600 145E 90 F6 BCC PRNTLPL ; WILL ALWAYS BRANCH (UNLESS ERROR)
601 1460 A9 OC LDA #FIOERR ; FILE I/O ERROR
602 1462 20 82 2F JSR ERROR
603
604 1465 4C O3 10 PRTEXT: JMP JWARM ; RETURN TO TOP LEVEL
605
606
607
608 ; PRINT THE SIZE (IN HEX) OF THE SPECIFIED FILE
609 ; ADDED 03-MAR-81
610
611 1468 20 BF 17 FSIZE: JSR OPENFI ; OPEN FILE for INPUT
612 146B A9 00 LDA #0 ; ZERO OUT THE COUNTER
613 146D 8D C1 36 STA FSZCTR+0
614 1470 8D C2 36 STA FSZCTR+1
615 1473 8D C3 36 STA FSZCTR+2
616 1476 8D C4 36 STA FSZCTR+3
617 1479 20 AE 1A FSIZLPL: JSR EXEINP
618 147C B0 14 BCS FSZEXT
619 147E EE C1 36 INC FSZCTR+0 ; BUMP UP FILE-SIZE-COUNTER
620 1481 D0 F6 BNE FSIZLPL
621 1483 EE C2 36 INC FSZCTR+1
622 1486 D0 F1 BNE FSIZLPL
623 1488 EE C3 36 INC FSZCTR+2
624 148B D0 EC BNE FSIZLPL
625 148D EE C4 36 INC FSZCTR+3
626 1490 D0 E7 BNE FSIZLPL
627
628 1492 20 E0 2B FSZEXT: JSR CRLF

```


629	1495	A2	32		LDX	#FSZMSG^	; HIGH BYTE
630	1497	AO	F4		LDY	#FSZMSG&\$FF	; LOW BYTE
631	1499	20	AF	2B	JSR	PRTSTR	
632	149C	AD	C4	36	LDA	FSZCTR+3	; HIGH BYTE
633	149F	20	C3	2B	JSR	PRTBYT	
634	14A2	AD	C3	36	LDA	FSZCTR+2	
635	14A5	20	C3	2B	JSR	PRTBYT	
636	14A8	AD	C2	36	LDA	FSZCTR+1	
637	14AB	20	C3	2B	JSR	PRTBYT	
638	14AE	AD	C1	36	LDA	FSZCTR+0	
639	14B1	20	C3	2B	JSR	PRTBYT	
640	14B4	4C	03	10	JMP	JWARM	
641							

```

643          .SBTTL  Memory Transfer Commands
644
645
646 14B7 20 BF 17 READF: JSR  OPENFI      ; OPEN THE FILE FOR INPUT
647 14BA A9 O1      LDA  #1            ; OUTPUT FLAG
648 14BC 20 OB 2B      JSR  M.OPN      ; OPEN MEMORY FOR OUTPUT
649          ; READ LOOP
650 14BF 20 AE 1A RDLF: JSR  EXEINP     ; GET THE INPUT CHAR
651 14C2 B0 OA      BCS  RDEXIT      ; EXIT
652 14C4 20 40 2B      JSR  M.OUT      ; OUTPUT TO MEMORY
653 14C7 90 F6      BCC  RDLF        ; WILL ALWAYS BRANCH (UNLESS ERROR)
654 14C9 A9 OC      LDA  #FIOERR     ; FILE I/O ERROR
655 14CB 20 82 2F      JSR  ERROR      ;
656
657 14CE 20 29 2B RDEXIT: JSR  M.CLO     ; CLOSE MEMORY ON OUTPUT
658 14D1 4C O3 10      JMP  JWARM      ; RETURN TO TOP LEVEL
659
660
661
662
663
664
665 14D4 A9 O0      WRITEF: LDA  #O        ; INPUT FLAG
666 14D6 20 OB 2B      JSR  M.OPN      ; OPEN MEMORY FOR INPUT
667 14D9 20 28 18      JSR  OPENFO     ; OPEN THE FILE FOR OUTPUT
668          ; WRITE LOOP
669 14DC 20 31 2B WRLF: JSR  M.INP      ; GET THE INPUT CHAR FROM MEMORY
670 14DF B0 OD      BCS  WREXIT      ; EXIT
671 14E1 20 B1 1A      JSR  EXEOUT     ; OUTPUT TO DEVICE
672 14E4 90 F6      BCC  WRLF        ; WILL ALWAYS BRANCH UNLESS ERROR
673 14E6 20 A2 18      JSR  CLOSEF     ; CLOSE FILE ANYWAY
674 14E9 A9 OC      LDA  #FIOERR     ; FALL THROUGH TO ERROR
675 14EB 20 82 2F      JSR  ERROR      ; FILE I/O ERROR
676
677 14EE 20 A2 18 WREXIT: JSR  CLOSEF     ; CLOSE FILE (OUTPUT ONLY)
678 14F1 4C O3 10      JMP  JWARM      ; RETURN TO TOP LEVEL
679

```

```
681 .SBTTL File Transfer Command
682
683
684 14F4 20 BF 17 COPYF: JSR OPENFI ; OPEN the file for INPUT
685 14F7 20 28 18 JSR OPENFO ; OPEN the file for OUTPUT
686
687 ; COPY FILE LOOP
688 14FA 20 AE 1A CPYFLP: JSR EXEINP ; INPUT A CHAR (BYTE)
689 14FD B0 OD CPYEXT ; EXIT ON EOF
690 14FF 20 B1 1A JSR EXEOUT ; OUTPUT A CHAR (BYTE)
691 1502 90 F6 BCC CPYFLP ; WILL ALWAYS BRANCH UNLESS ERROR
692 1504 20 A2 18 JSR CLOSEF ; CLOSE ANYWAY ON ERROR
693 1507 A9 OC LDA #FIOERR
694 1509 20 82 2F JSR ERROR
695
696 150C 20 A2 18 CPYEXT: JSR CLOSEF ; CLOSE FILE (OUTPUT ONLY)
697 150F 4C 03 10 JMP JWARM ; GO BACK TO THE MAIN ROUTINE
698
699
```

```

701          .SBTTL  Sub-Programs
702
703
704 1512 20 06 1A  DUMP:  JSR    GETNUM      ; ADD <FROM> [<TO>]
705 1515 90 0A          BCC    DMPCC        ; GET THE ADDRESS
706 1517 A9 00          LDA    #USRRAM&$FF  ; IF NO ADDR, DEFAULT TO USER-RAM
707 1519 8D 42 36      STA    NUMBER+0
708 151C A9 80          LDA    #USRRAM^
709 151E 8D 43 36      STA    NUMBER+1
710 1521 AD 42 36      DMPCC: LDA    NUMBER+0  ; LOW BYTE
711 1524 85 02          STA    PRTPTR+0
712 1526 AD 43 36      LDA    NUMBER+1  ; HIGH BYTE
713 1529 85 03          STA    PRTPTR+1
714 152B 20 06 1A      JSR    GETNUM      ; GET END ADDRESS (IF PRESENT)
715 152E 90 1B          BCC    DUMPLP     ; BRANCH IF NUMBER PRESENT
716 1530 20 5D 15      JSR    DMPLIN     ; ONLY DO 8 LINES IF NO NUMBER
717 1533 20 5D 15      JSR    DMPLIN
718 1536 20 5D 15      JSR    DMPLIN
719 1539 20 5D 15      JSR    DMPLIN
720 153C 20 5D 15      JSR    DMPLIN
721 153F 20 5D 15      JSR    DMPLIN
722 1542 20 5D 15      JSR    DMPLIN
723 1545 20 5D 15      JSR    DMPLIN
724 1548 4C 03 10      JMP    JWARM
725
726 154B 20 5D 15      DUMPLP: JSR    DMPLIN  ; DISPLAY ONE LINE
727 154E A5 02          LDA    PRTPTR+0  ; TEST IF AT END
728 1550 CD 42 36      CMP    NUMBER+0
729 1553 A5 03          LDA    PRTPTR+1
730 1555 ED 43 36      SBC    NUMBER+1  ; CARRY SET IF END
731 1558 90 F1          BCC    DUMPLP     ; FALL THROUGH WHEN DONE
732 155A 4C 03 10      JMP    JWARM     ; GO BACK TO MAIN
733
734 155D 20 FO 2B      DMPLIN: JSR    CRLFSQ  ; NEW LINE (^S/^Q & ^C)
735 1560 A5 03          LDA    PRTPTR+1  ; PRINT HIGH BYTE OF ADDRESS
736 1562 20 C3 2B      JSR    PRTBYT
737 1565 A5 02          LDA    PRTPTR+0  ; PRINT LOW BYTE OF ADDRESS
738 1567 20 C3 2B      JSR    PRTBYT
739 156A A9 3A          LDA    #' :
740 156C 20 95 2B      JSR    CTYOUT
741 156F 20 09 10      JSR    CTYSPA
742 1572 A0 00          LDY    #0
743 1574 B1 02          DMPLP2: LDA    (PRTPTR),Y
744 1576 20 C3 2B      JSR    PRTBYT
745 1579 20 09 10      JSR    CTYSPA
746 157C C8            INY
747 157D CC 15 10      CPY    DMPBPL     ; 16 or 8
748 1580 90 F2          BLT    DMPLP2
749 1582 20 09 10      JSR    CTYSPA     ; PRINT A 2ND SPACE
750 1585 A0 00          LDY    #0
751 1587 B1 02          DMPLP3: LDA    (PRTPTR),Y
752 1589 20 2B 2C      JSR    CHRPER     ; PRINT VALID CHAR or '.'
753 158C C8            INY
754 158D CC 15 10      CPY    DMPBPL     ; 16 or 8
755 1590 90 F5          BLT    DMPLP3

```

756	1592	18			CLC	
757	1593	A5	O2		LDA	PRTPTR
758	1595	6D	15	10	ADC	DMPBPL ; 16 or 8
759	1598	85	O2		STA	PRTPTR
760	159A	90	O2		BCC	DMPLRT
761	159C	E6	O3		INC	PRTPTR+1
762	159E	60			DMPLRT:	RTS
763						
764						
765						

767					.SBTTL	Sub-Pgm	Sector	Read		
768										
769	159F	20	06	1A	DENSTY:	JSR	GETNUM		:	GET DENSITY
770	15A2	BO	3C			BCS	MS.PRM			
771	15A4	C9	03			CMP	#3			
772	15A6	BO	22			BGE	DNSERR			
773	15A8	29	03			AND	#\$03			
774	15AA	8D	C7	36		STA	SUBDEN			
775	15AD	20	06	1A		JSR	GETNUM		:	GET BYTES/SECTOR FLAG
776	15B0	BO	2E			BCS	MS.PRM			
777	15B2	29	0F			AND	#\$0F			
778	15B4	8D	C8	36		STA	SUBYSC			
779	15B7	20	06	1A		JSR	GETNUM		:	GET THE FIRST SECTOR #
780	15BA	BO	24			BCS	MS.PRM			
781	15BC	8D	C9	36		STA	SBFRST			
782	15BF	20	06	1A		JSR	GETNUM		:	GET THE LAST SECTOR #
783	15C2	BO	1C			BCS	MS.PRM			
784	15C4	8D	CA	36		STA	SBLAST			
785	15C7	4C	03	10		JMP	JWARM			
786										
787	15CA	A9	0A		DNSERR:	LDA	#DENERR		:	DENSITY SET ERROR
788	15CC	20	82	2F		JSR	ERROR			
789										
790										
791										
792	15CF	AD	C7	36	DENCHK:	LDA	SUBDEN		:	DENSITY CHECK
793	15D2	C9	01			CMP	#1			
794	15D4	FO	09			BEQ	DEN.OK			
795	15D6	C9	02			CMP	#2			
796	15D8	FO	05			BEQ	DEN.OK			
797	15DA	A9	0A			LDA	#DENERR		:	INVALID DENSITY
798	15DC	20	82	2F		JSR	ERROR			
799										
800	15DF	60			DEN.OK:	RTS				
801										
802										
803	15E0	A9	15		MS.PRM:	LDA	#NOPARM		:	MISSING PARAMETER ERROR
804	15E2	20	82	2F		JSR	ERROR			
805										
806										

```

808
809 15E5 20 CF 15 SECTOR: JSR DENCHK ; CHECK DENSITY
810 15E8 AD C7 36 LDA SUBDEN ; MOVE THE DENSITY
811 15EB 8D 3A 36 STA TMPDEN
812 15EE AD C8 36 LDA SUBYSC ; GET BYTES/SECTOR FLAG
813 15F1 8D 39 36 STA TMBYSC
814 15F4 20 B3 19 JSR GTATOM
815 15F7 AD 00 38 LDA ATMBUF+0
816 15FA 8D CB 36 STA SUBFLG ; FLAG TO READ OR WRITE (R W)
817 15FD C9 57 CMP #'W
818 15FF FO 09 BEQ SECTOK
819 1601 C9 52 CMP #'R
820 1603 FO 05 BEQ SECTOK
821 1605 A9 1D LDA #INVPRM
822 1607 20 82 2F JSR ERROR
823
824 160A 20 06 1A SECTOK: JSR GETNUM ; GET THE DRIVE #
825 160D BO D1 BCS MS.PRM ; MISSING PARM
826 160F C9 04 CMP #4 ; 0-3
827 1611 BO 56 BGE IVN.ER ; INVALID DRIVE #
828 1613 8D 33 36 STA TMPDRV
829 1616 8D CC 36 STA SUBDRV
830 1619 20 06 1A JSR GETNUM ; GET THE TRACK #
831 161C BO C2 BCS MS.PRM ; MISSING PARM
832 161E C9 4D CMP #77 ; 0-76
833 1620 BO 47 BGE IVN.ER ; INVALID DRIVE #
834 1622 8D 35 36 STA TMPTRK
835 1625 8D CD 36 STA SUBTRK
836 1628 20 06 1A JSR GETNUM ; GET SECTOR #
837 162B BO B3 BCS MS.PRM ; MISSING PARM
838 162D C9 21 CMP #33 ; NEVER MORE THAN 32
839 162F BO 38 BGE IVN.ER ; INVALID SECTOR #
840 1631 8D 36 36 STA TMPSTCT ; SET SECTOR TO READ
841 1634 8D 38 36 STA TMLSTCT ; SET AS LAST SECTOR ALSO
842 1637 8D CE 36 STA SUBSTCT
843 163A A9 80 LDA #USRRAM^ ; SET DMA PAGE
844 163C 8D 3B 36 STA TMPAGE
845 163F AD CB 36 LDA SUBFLG
846 1642 C9 52 CMP #'R
847 1644 FO 09 BEQ RSCTR ; GO READ
848 1646 C9 57 CMP #'W
849 1648 FO 08 BEQ WSCTR ; GO WRITE
850 164A A9 1D LDA #INVPRM
851 164C 20 82 2F JSR ERROR
852
853 164F 20 ED 2C RSCTR: JSR RDTEMP ; READ THE SECTOR
854 1652 4C 03 10 JMP JWARM
855
856 1655 A2 33 WSCTR: LDX #CONFIRM^ ; CONFIRM MSG
857 1657 A0 03 LDY #CONFIRM&$FF
858 1659 20 AF 2B JSR PRTSTR
859 165C 20 77 2B JSR CTYIN
860 165F C9 59 CMP #'Y
861 1661 DO 03 BNE WRSEND ; ABORT IF NOT
862 1663 20 AA 2D JSR WRTEMP ; WRITE THE SECTOR
    
```

```

863 1666 4C 03 10 WRSEND: JMP JWARM
864
865 1669 A9 1C IVN.ER: LDA #INVNUM ; INVALID NUMBER ERROR
866 166B 20 82 2F JSR ERROR
867
868
869
870 166E NEXT: ; READ THE 'NEXT' SECTOR
871 166E EE CE 36 INC SUBSCT
872 1671 AD CE 36 LDA SUBSCT
873 1674 CD CA 36 CMP SBLAST
874 1677 FO 13 BEQ NEXTOK
875 1679 90 11 BLT NEXTOK
876 167B AD C9 36 LDA SBFRST
877 167E 8D CE 36 STA SUBSCT
878 1681 EE CD 36 INC SUBTRK
879 1684 AD CD 36 LDA SUBTRK
880 1687 C9 4D CMP #77
881 1689 90 01 BLT NEXTOK
882 168B 60 RTS
883
884 168C 20 CF 15 NEXTOK: JSR DENCHK
885 168F AD C7 36 LDA SUBDEN ; SET DENSITY
886 1692 8D 3A 36 STA TMPDEN
887 1695 AD CC 36 LDA SUBDRV ; SET DRIVE
888 1698 8D 33 36 STA TMPDRV
889 169B AD C8 36 LDA SUBYSC ; SET BYTES/SECTOR
890 169E 8D 39 36 STA TMBYSC
891 16A1 AD CD 36 LDA SUBTRK ; SET TRACK #
892 16A4 8D 35 36 STA TMPTRK
893 16A7 AD CE 36 LDA SUBSCT ; SET SECTOR #
894 16AA 8D 36 36 STA TMPSCCT
895 16AD 8D 38 36 STA TMLSCT ; LAST SECTOR # TOO
896 16B0 A9 80 LDA #USRRAM^ ; SET DMA PAGE
897 16B2 8D 3B 36 STA TMPAGE
898
899 16B5 20 ED 2C JSR RDTEMP ; READ THE SECTOR
900 16B8 20 E0 2B JSR CRLF
901 16BB A9 54 LDA #'T
902 16BD 20 D7 16 JSR NXTEQU
903 16C0 AD 35 36 LDA TMPTRK
904 16C3 20 C3 2B JSR PRTBYT
905 16C6 20 3A 2C JSR SPACE4
906 16C9 A9 53 LDA #'S
907 16CB 20 D7 16 JSR NXTEQU
908 16CE AD 36 36 LDA TMPSCCT
909 16D1 20 C3 2B JSR PRTBYT
910 16D4 4C 03 10 JMP JWARM
911
912 16D7 20 95 2B NXTEQU: JSR CTYOUT
913 16DA A9 3D LDA #' =
914 16DC 20 95 2B JSR CTYOUT
915 16DF 60 RTS
916

```



```

918          .SBTTL  Sub-Pgm Disk Duplicate
919
920
921          ; DO SOME WORK ON THIS!!!!
922          ; COULD DO 2 LOOPS/ 1-SINGLE 2-DOUBLE DEN
923 16E0 A9 0B          D.DISK: LDA  #NOTIMP
924 16E2 20 82 2F          JSR  ERROR
925 16E5 20 FO 1A          JSR  GETDRV
926 16E8 B0 50          BCS  D.DERR          ; NO DRIVE # ERROR
927 16EA 8D C5 36          STA  FRMDRV          ; FROM DRIVE
928 16ED 20 FO 1A          JSR  GETDRV
929 16F0 B0 48          BCS  D.DERR          ; NO DRIVE # ERROR
930 16F2 8D C6 36          STA  TODRV          ; TO DRIVE
931 16F5 20 O6 1A          JSR  GETNUM
932 16F8 B0 40          BCS  D.DERR
933 16FA 8D C9 36          STA  SBFRST          ; First SECTOR
934 16FD 20 O6 1A          JSR  GETNUM
935 1700 B0 38          BCS  D.DERR
936 1702 8D CA 36          STA  SBLAST          ; Last SECTOR
937
938 1705 A9 00          LDA  #0          ; START WITH TRACK 0
939 1707 8D 35 36          STA  TMPTRK
940 170A AD C9 36          LDA  SBFRST
941 170D 8D 36 36          STA  TMPSCCT
942 1710 AD CA 36          LDA  SBLAST
943 1713 8D 38 36          STA  TMLSCCT
944
945 1716 A9 80          LDA  #USRRAM^          ; SET BUFFER ADDRESS
946 1718 8D 3B 36          STA  TMPAGE
947
948 171B AD C5 36          DDUPLP: LDA  FRMDRV
949 171E 8D 33 36          STA  TMPDRV          ; SET 'FROM' DRIVE
950 1721 20 3F 17          JSR  DPREAD
951 1724 AD C6 36          LDA  TODRV
952 1727 8D 33 36          STA  TMPDRV          ; SET 'TO' DRIVE
953 172A 20 40 17          JSR  DPWRIT
954 172D EE 35 36          INC  TMPTRK
955 1730 AD 35 36          LDA  TMPTRK
956 1733 C9 4D          CMP  #77
957 1735 90 E4          BLT  DDUPLP
958 1737 4C O3 10          JMP  JWARM
959
960 173A A9 15          D.DERR: LDA  #NOPARM          ; MISSING PARM
961 173C 20 82 2F          JSR  ERROR
962
963 173F          DPREAD:
964 173F 60          RTS
965
966 1740          DPWRIT:
967 1740 60          RTS
968
969
970
971

```

```

973          .SBTTL Control Transfer Commands
974
975          ;          .READ GOEXIT
976          .MCALL GOEXIT
977 1741      GOEXIT
(1)
(1) 1741    20 06 1A    GO:      JSR      GETNUM      ; GET THE ADDRESS
(1) 1744    B0 03      BCS      GABORT      ; ABORT IF NO NUMBER
(1) 1746    20 4C 17    JSR      GOJSR      ; ALLOW TERMINATION WITH RTS
(1) 1749    4C 03 10    GABORT:  JMP      JWARM
(1)
(1) 174C    6C 42 36    GOJSR:  JMP      (NUMBER)
(1)
(1) 174F    20 E0 2B    EXIT:   JSR      CRLF
(1) 1752    6C FC FF    JMP      ($FFFC)      ; WILL JUMP TO RESET ON ANY SYSTEM
(1)
978
979
980
981 1755    AD 44 36    ECODOS: LDA      BUFPTR      ; CHEAT
982 1758    48          PHA          ; SAVE BUFFER POINTER
983 1759    20 B3 19    JSR      GTATOM
984 175C    B0 26      BCS      JCODOS
985 175E    AD 00 38    LDA      ATMBUF+0      ; CHECK FIRST CHAR
986 1761    C9 3B      CMP      #' ;
987 1763    F0 1F      BEQ      JCODOS
988 1765    68          PLA          ; RESTORE BUFFER POINTER
989 1766    8D 44 36    STA      BUFPTR
990 1769    18          CLC
991 176A    69 00      ADC      #INBUFF&$FF
992 176C    85 BA      STA      U5+0          ; MAKE U5 POINT TO REST OF LINE
993 176E    A9 37      LDA      #INBUFF^
994 1770    69 00      ADC      #0
995 1772    85 BB      STA      U5+1
996 1774    20 E0 2B    JSR      CRLF          ; NEW LINE
997 1777    A9 80      LDA      #$80          ; ENABLE SVCs
998 1779    85 EE      STA      SVCENA
999 177B      SVC      13          ; EXECUTE CODOS COMMAND
(1) 177B    00 OD      .BYTE    0,13
1000 177D    A9 00      LDA      #0          ; DISABLE SVCs
1001 177F    85 EE      STA      SVCENA
1002 1781    4C 03 10    JMP      JWARM
1003
1004 1784    68          JCODOS:  PLA          ; RESTORE STACK, DISCARD
1005 1785    A9 00      LDA      #0          ; DISABLE SVCs
1006 1787    85 EE      STA      SVCENA
1007 1789    4C 03 C6    JMP      CODOS        ; TRANSFER BACK TO CODOS
1008
1009

```

```
1011
1012 178C          DBOOT:          ; MAKE SURE DISK IS CODOS
1013 178C          LDA      #USRRAM^
1014 178E      A9      80          STA      TMPAGE
1015 1791      A9      00          LDA      #0
1016 1793      8D      33      36  STA      TMPDRV
1017 1796      8D      35      36  STA      TMPTRK
1018 1799      8D      36      36  STA      TMPSCCT
1019 179C      8D      38      36  STA      TMLSCT
1020 179F      A9      02          LDA      #2          ; DENSITY
1021 17A1      8D      3A      36  STA      TMPDEN
1022 17A4      A9      01          LDA      #1          ; BYTES/SECTOR CODE
1023 17A6      8D      39      36  STA      TMBYSC
1024 17A9      20      F3      2C  JSR      RDRTRN          ; READ & RETURN
1025 17AC      B0      0C          BCS      DBT.ER
1026
1027 17AE      AD      3F      80  LDA      USRRAM+63          ; ALSO CHECK ENTRY ADDRESS
1028 17B1      29      F0          AND      #$FO          ; GET HIGH NIBBLE
1029 17B3      C9      C0          CMP      #SYSRAM^
1030 17B5      D0      03          BNE      DBT.ER
1031
1032 17B7      4C      00      DF  JMP      DSKIPL
1033
1034 17BA      A9      1B          DBT.ER: LDA      #DBTERR          ; DISK BOOT ERROR
1035 17BC      20      82      2F  JSR      ERROR
1036
```

```

1038          .SBTTL  OPEN Routines
1039
1040 17BF 20 D7 18  OPENFI: JSR  GFNAME      ; GET (SORT-OF) FILE-NAME
1041 17C2 30 50      BMI  OPNDVI     ; OPEN DEVICE INPUT
1042 17C4 90 03      BCC  OPNFIO     ; BRANCH IF DRIVE # IN ACC
1043 17C6 AD C8 35      LDA  DRIVEI     ; ELSE GET DEFAULT INPUT DRIVE #
1044 17C9 8D 13 36  OPNFIO: STA  INPDRV     ; SAVE INPUT DRIVE #
1045 17CC 8D 33 36      STA  TMPDRV     ; IN TEMP TOO **
1046 17CF 20 8B 1A      JSR  DSKTYP     ; GET DEVICE # & DRIVE # for INPUT
1047 17D2 8D 12 36      STA  INPDEV     ; SAVE INPUT DEVICE #
1048 17D5 8D 32 36      STA  TMPDEV     ; IN TEMP TOO **
1049 17D8 0A          ASL  A           ; A <= A*2
1050 17D9 AA          TAX           ; USE AS INDEX
1051 17DA BD B1 18      LDA  INPTAB+0,X  ; SET UP INPUT INDIRECT JUMP
1052 17DD 8D 10 36      STA  INPJMP+0   ; LOW BYTE
1053 17E0 BD B2 1B      LDA  INPTAB+1,X
1054 17E3 8D 11 36      STA  INPJMP+1   ; HIGH BYTE
1055
1056 17E6 BD 0B 1C      LDA  DBSTAB+0,X  ; GET BYTES/SECTOR
1057 17E9 3C 06      BMI  OPNF11     ; IF NEGATIVE, IGNORE
1058 17EB 8D 19 36      STA  INBYSC
1059 17EE 8D 39 36      STA  TMBYSC
1060 17F1 BD 0C 1C  OPNF11: LDA  DBSTAB+1,X  ; GET DENSITY
1061 17F4 3C 06      BMI  OPNF12     ; IF NEGATIVE, IGNORE
1062 17F6 8D 1A 36      STA  INPDEN
1063 17F9 8D 3A 36      STA  TMPDEN
1064 17FC A9 82      OPNF12: LDA  #INPBUF^  ; SET BUFFER ADDRESS
1065 17FE 8D 1B 36      STA  INPAGE
1066 1801 A9 00      LDA  #0         ; INPUT FLAG for OPENF
1067 1803 20 8B 18      JSR  OPENF      ; OPEN the FILE
1068 1806 B0 7E      BCS  OPN_ER     ; ERROR IF FILE NOT FOUND
1069 1808 AD 4A 36      LDA  DELIMIT    ; CHECK DELIMITER
1070 180B C9 3A      CMP  #' :       ; IF COLON, GET RID OF DRIVE #
1071 180D D0 03      BNE  OPNF13
1072 180F 20 B3 19      JSR  GTATOM     ; GET THE DRIVE # %%%
1073 1812 18      OPNF13: CLC
1074 1813 60      RTS
1075
1076 1814 8D 12 36  OPNDVI: STA  INPDEV     ; SAVE THE DEVICE CHAR
1077 1817 20 04 19      JSR  OPNDEV     ; OPEN THE DEVICE
1078          ; (PASS THE DEV IN ACC)
1079 181A BD 4D 1C      LDA  DEVTAB+6,X
1080 181D 8D 10 36      STA  INPJMP+0
1081 1820 BD 4E 1C      LDA  DEVTAB+7,X
1082 1823 8D 11 36      STA  INPJMP+1
1083 1826 18      CLC
1084 1827 60      RTS
1085
    
```

```

1087
1088 1828 20 D7 18 OPENFO: JSR GFNAME ; GET (SORT-OF) FILE-NAME
1089 182B 30 45 BMI OPNDVO ; OPEN DEVICE OUTPUT
1090 182D 90 03 BCC OPNFOO ; BRANCH IF DRIVE # IN ACC
1091 182F AD C9 35 LDA DRIVEO ; ELSE GET DEFAULT OUTPUT DRIVE #
1092 1832 8D 23 36 OPNFOO: STA OUTDRV ; SAVE OUTPUT DRIVE #
1093 1835 8D 33 36 STA TMPDRV ; IN TEMP TOO **
1094 1838 20 8B 1A JSR DSKTYP ; GET DEVICE # & DRIVE # for OUTPUT
1095 183B 8D 22 36 STA OUTDEV ; SAVE OUTPUT DEVICE #
1096 183E 8D 32 36 STA TMPDEV ; IN TEMP TOO **
1097 1841 OA ASL A ; A <= A*2
1098 1842 AA TAX ; USE AS INDEX
1099 1843 BD CF 1B LDA OUTTAB+0,X ; SET UP OUTPUT INDIRECT JUMP
1100 1846 8D 20 36 STA OUTJMP+0 ; LOW BYTE
1101 1849 BD DO 1B LDA OUTTAB+1,X
1102 184C 8D 21 36 STA OUTJMP+1 ; HIGH BYTE
1103
1104 184F BD OB 1C LDA DBSTAB+0,X ; GET BYTES/SECTOR
1105 1852 30 06 BMI OPNFO1 ; IF NEGATIVE, IGNORE
1106 1854 8D 29 36 STA OUBYSC
1107 1857 8D 39 36 STA TMBYSC
1108 185A BD OC 1C OPNFO1: LDA DBSTAB+1,X ; GET DENSITY
1109 185D 30 06 BMI OPNFO2 ; IF NEGATIVE, IGNORE
1110 185F 8D 2A 36 STA OUTDEN
1111 1862 8D 3A 36 STA TMPDEN
1112 1865 A9 84 OPNFO2: LDA #OUTBUF^ ; SET BUFFER ADDRESS
1113 1867 8D 2B 36 STA OUPAGE
1114 186A A9 01 LDA #1 ; OUTPUT FLAG for OPEN
1115 186C 20 8B 18 JSR OPENF
1116 186F B0 15 BCS OPN.ER ; ERROR IF FILE NOT FOUND
1117 1871 60 RTS
1118
1119 1872 8D 22 36 OPNDVO: STA OUTDEV ; SAVE THE DEVICE CHAR
1120 1875 20 04 19 JSR OPNDEV ; OPEN THE DEVICE
1121 ; (PASS THE DEV IN ACC)
1122 1878 BD 4F 1C LDA DEVTAB+8,X
1123 187B 8D 20 36 STA OUTJMP+0
1124 187E BD 50 1C LDA DEVTAB+9,X
1125 1881 8D 21 36 STA OUTJMP+1
1126 1884 18 CLC
1127 1885 60 RTS
1128

```

```

1130
1131 1886 A9 11      OPN.ER: LDA      #OPNERR      ; OPEN ERROR
1132 1888 20 82 2F      JSR      ERROR
1133
1134
1135                                ; OPEN A FILE
1136 188B 8D 3C 36      OPENF: STA      TMPIOF      ; SAVE INPUT/OUTPUT FLAG
1137 188E A9 80          LDA      #DIRBUF^     ; SET BUFFER ADDRESS
1138 1890 8D 3B 36      STA      TMPAGE
1139 1893 BD 75 1B      LDA      OPNTAB+0,X    ; OPEN FILE ROUTINE LOW BYTE
1140 1896 8D 40 36      STA      CMDJMP+0
1141 1899 BD 76 1B      LDA      OPNTAB+1,X    ; OPEN FILE ROUTINE HIGH BYTE
1142 189C 8D 41 36      STA      CMDJMP+1
1143 189F 6C 40 36      JMP      (CMDJMP)      ; OPEN THE FILE
1144
1145
1146
1147                                ; THE OPEN ROUTINES USE THE RDTEMP READ SECTOR ROUTINE
1148                                ; SO THEY DO NOT INTERFERE WITH EITHER INPUT or OUTPUT.
1149
1150
1151
1152
1153      .SBTTL  CLOSE Routine
1154
1155 18A2                                CLOSEF:                                ; CLOSE A FILE (OUTPUT ONLY)
1156 18A2 AD 22 36      LDA      OUTDEV      ; GET OUTPUT DEVICE #
1157 18A5 C9 41          CMP      #'A
1158 18A7 BO 14          BGE      CLODEV
1159 18A9 OA            ASL      A              ; A <= A*2
1160 18AA AA            TAX
1161 18AB BD 93 1B      LDA      CLOTAB+0,X    ; CLOSE FILE ROUTINE LOW BYTE
1162 18AE 8D 30 36      STA      TMPJMP+0
1163 18B1 BD 94 1B      LDA      CLOTAB+1,X    ; CLOSE FILE ROUTINE HIGH BYTE
1164 18B4 8D 31 36      STA      TMPJMP+1
1165 18B7 20 B4 1A      JSR      EXETMP      ; CLOSE THE FILE
1166 18BA BO 16          BCS      CLO.ER      ; CLOSE ERROR
1167 18BC 60
1168
1169 18BD 20 90 1A      CLODEV: JSR      DEV.NO      ; GET THE DEVICE INDEX
1170 18C0 BD 4B 1C      LDA      DEVTAB+4,X
1171 18C3 8D 30 36      STA      TMPJMP+0
1172 18C6 BD 4C 1C      LDA      DEVTAB+5,X
1173 18C9 8D 31 36      STA      TMPJMP+1
1174 18CC 20 B4 1A      JSR      EXETMP
1175 18CF BO 01          BCS      CLO.ER
1176 18D1 60
1177
1178 18D2 A9 12          CLO.ER: LDA      #CLOERR      ; CLOSE FILE ERROR
1179 18D4 20 82 2F      JSR      ERROR
1180

```

```

1182
1183 18D7 AD 44 36 GFNAME: LDA BUFPTR ; CHEAT
1184 18DA 48 PHA ; SAVE BUFFER POINTER
1185 18DB 20 B3 19 JSR GTATOM
1186 18DE CD 01 38 CMP ATMBUF+1
1187 18E1 DO 07 BNE GFNMOK ; IF TRUE, FALL THROUGH
1188 18E3 68 PLA ; PULL BUFFER POINTER BUT IGNORE
1189 18E4 AD 00 38 LDA ATMBUF+0 ; GET THE DEVICE CHARACTER
1190 18E7 09 80 ORA #$80 ; MAKE NEGATIVE
1191 18E9 60 RTS
1192
1193 18EA C9 3A GFNMOK: CMP #' : ; IF COLON THEN GET DRIVE #
1194 18EC DO 10 BNE GFNDRV
1195 18EE 20 B3 19 JSR GTATOM
1196 18F1 AD 00 38 LDA ATMBUF+0 ; GET FIRST CHAR
1197 18F4 29 03 AND #$03
1198 18F6 AA TAX ; SAVE IN X REG
1199 18F7 68 PLA ; RESTORE BUFFER POINTER
1200 18F8 8D 44 36 STA BUFPTR
1201 18FB 8A TXA ; GET DRIVE #
1202 18FC 18 CLC
1203 18FD 60 RTS
1204
1205 18FE 68 GFNDRV: PLA ; NO DRIVE
1206 18FF 8D 44 36 STA BUFPTR ; RESTORE BUFFER POINTER
1207 1902 38 SEC
1208 1903 60 RTS ; RETURN C = 0 IF DRIVE # GIVEN
1209 ; C = 1 IF NOT
1210 ; A = DEVICE OR DRIVE #
1211
1212
1213 1904 20 90 1A OPNDEV: JSR DEV.NO ; GET THE DEVICE #
1214 1907 BD 49 1C LDA DEVTAB+2,X
1215 190A 8D 30 36 STA TMPJMP+0
1216 190D BD 4A 1C LDA DEVTAB+3,X
1217 1910 8D 31 36 STA TMPJMP+1
1218 1913 8A TXA
1219 1914 48 PHA ; SAVE THE DEVICE INDEX
1220 1915 20 B4 1A JSR EXETMP
1221 1918 B0 04 BCS OPNDVE ; OPEN DEVICE ERROR
1222 191A 68 PLA
1223 191B AA TAX ; RESTORE IT
1224 191C 18 CLC
1225 191D 60 RTS
1226
1227 191E 68 OPNDVE: PLA ; PULL BUT DISCARD
1228 191F 38 SEC
1229 1920 60 RTS
1230
1231
1232
1233
    
```

```

1235          .SBTTL  Main-Program Subroutines
1236
1237          .MCALL  INSTRG, GTATOM, GETNUM
1238          ;      .READ  INSTRING.A
1239 1921      INSTRG
(1)          ;      INPUT STRING ROUTINE
(1)
(1) 1921      INSTRG:          ; ROUTINE TO INPUT A STRING OF TEXT
(1)          ; ENDING WITH A C/R
(1) 1921  A0  50          LDY    #80          ; INIT Y INDEX POINTER
(1) 1923  A9  0D          LDA    #$0D         ; <CR>S
(1) 1925  99  00  37     INSFIL: STA  INBUFF,Y  ; PUT <CR>S THROUGHOUT THE BUFFER
(1) 1928  88          DEY
(1) 1929  DO  FA          BNE    INSFIL      ; Y = 0 WHEN DONE
(1)
(1) 192B  20  77  2B     INSTR1: JSR  CTYIN      ; GET A CHARACTER
(1)
(1) 192E  C9  00          CMP    #$00          ; = ^@ ? <NUL>
(1) 1930  FO  F9          BEQ    INSTR1       ; IGNORE NULLS
(1) 1932  C9  03          CMP    #'C&$3F      ; = ^C ?
(1) 1934  FO  2C          BEQ    CANLIN      ; ABORT (CANCEL LINE)
(1) 1936  C9  08          CMP    #'H&$3F      ; = ^H ?
(1) 1938  FO  3B          BEQ    CTRLH      ; BACK-SPACE
(1) 193A  C9  0A          CMP    #'J&$3F      ; = ^J ? <LF>
(1) 193C  FO  ED          BEQ    INSTR1       ; IGNORE LINE FEEDS
(1) 193E  C9  OD          CMP    #'M&$3F      ; = ^M ? <CR>
(1) 1940  FO  43          BEQ    CTRLM      ; CARRIAGE RETURN
(1) 1942  C9  12          CMP    #'R&$3F      ;
(1) 1944  FO  51          BEQ    CTRLR      ;
(1) 1946  C9  15          CMP    #'U&$3F      ; = ^U ?
(1) 1948  FO  18          BEQ    CANLIN      ; ABORT (CANCEL LINE)
(1) 194A  C9  18          CMP    #'X&$3F      ; = ^X ?
(1) 194C  FO  14          BEQ    CANLIN      ; ABORT (CANCEL LINE)
(1) 194E  C9  7F          CMP    #$7F        ; = <RUBOUT> ?
(1) 1950  FO  1E          BEQ    RUBOUT      ; BACK-SPACE
(1) 1952  C9  20          CMP    #$20        ; < $20 ?
(1) 1954  90  D5          BLT    INSTR1      ; IGNORE ALL UNDEFINED CTRL CHARS
(1)
(1) 1956  C9  60          CMP    #$60        ; CHECK IF LOWER CASE
(1) 1958  90  02          BLT    UPPER      ; ALREADY UPPER CASE
(1) 195A  29  5F          AND    #$5F        ; MAKE UPPER CASE IF LOWER CASE
(1) 195C  99  00  37     UPPER: STA  INBUFF,Y  ; SAV IT IN THE INPUT BUFFER ($0200)
(1) 195F  C8          INY
(1) 1960  DO  C9          BNE    INSTR1      ; WILL ALWAYS BRANCH
(1)
(1)          ; ABORT (CANCEL LINE)
(1) 1962  48          CANLIN: PHA          ; SAVE CHAR
(1) 1963  A9  5E          LDA    #'^
(1) 1965  20  95  2B     JSR  CTYOUT
(1) 1968  68          PLA          ; RESTORE CHAR
(1) 1969  09  40          ORA    #$40        ; CONVERT TO NORMAL CHARACTER
(1) 196B  20  95  2B     JSR  CTYOUT
(1) 196E  38          SEC          ; SET ERROR FLAG (C = 1)
(1) 196F  60          RTS
(1)

```



```

(1) 1970 A9 08 RUBOUT: LDA #$08 ; PRINT A ^H (BACK-SPACE)
(1) 1972 20 95 2B JSR CTYOUT
(1) ; FALL THROUGH TO CTRLH
(1) 1975 CTRLH: ; BACK-SPACE
(1) 1975 88 DEY ; DECREMENT BUFFER POINTER
(1) 1976 30 1D BMI CTRLM1 ; RE-DO (REPRINT PROMPT IF BACK TOO FAR)
(1) 1978 A9 20 LDA #$20 ; SPACE
(1) 197A 20 95 2B JSR CTYOUT
(1) 197D A9 08 LDA #$08 ; GET BACKSPACE
(1) 197F 20 95 2B JSR CTYOUT
(1) 1982 4C 2B 19 JMP INSTR1 ; BACK FOR ANOTHER CHARACTER
(1)
(1)
(1) 1985 99 00 37 CTRLM: STA INBUFF,Y ; PUT AT END OF LINE
(1) 1988 AD 00 37 LDA INBUFF+0
(1) 198B C9 0D CMP #$0D ; IF <CR> THEN RE-DO
(1) 198D FO 06 BEQ CTRLM1
(1) 198F C9 3B CMP #' ;
(1) 1991 FO 02 BEQ CTRLM1 ; IF WHOLE LINE THEN RE-DO ALSO
(1) 1993 18 CLC ; SET OKAY FLAG
(1) 1994 60 RTS ; RETURN
(1)
(1) 1995 38 CTRLM1: SEC ; SET RE-DO FLAG
(1) 1996 60 RTS
(1)
(1) ; REPINT CURRENT LINE
(1)
(1) 1997 98 CTRLR: TYA ; SAVE Y
(1) 1998 48 PHA
(1) 1999 A9 00 LDA #$00
(1) 199B 99 00 37 STA INBUFF,Y ; MARK END OF LINE
(1) 199E 20 AB 2B JSR PROMPT ; PRINT SYSTEM PROMPT (MUST BE DEFINED IN PGM)
(1) 19A1 A0 00 LDY #$00
(1) 19A3 B9 00 37 CTRLR1: LDA INBUFF,Y
(1) 19A6 FO 06 BEQ CTRLR2
(1) 19A8 20 95 2B JSR CTYOUT
(1) 19AB C8 INY
(1) 19AC DO F5 BNE CTRLR1 ; WILL ALWAYS BRANCH
(1) 19AE 68 CTRLR2: PLA
(1) 19AF A8 TAY
(1) 19B0 4C 2B 19 JMP INSTR1 ; CONTINUE
(1)
(1)
1240 ; .READ GETATOM.A
1241 19B3 GTATOM
(1)
(1)
(1) ; GET ATOM: RETURNS NEXT ATOM
(1) ; '.' ARE NOT CONSIDERED DELIMITERS
(1) ; SO IT WILL RETURN A FULL FILENAME
(1) ; AS ONE ATOM
(1) ; GET THE NEXT ATOM IN THE INPUT BUFFER
(1) ; GET POINTER INTO THE INPUT BUFFER
(1) 19B3 AE 44 36 GTATOM: LDX BUFPTR
(1)
(1) 19B6 BD 00 37 GATOM1: LDA INBUFF,X ; GET THE CHAR
    
```

```

(1) 19B9 C9 OD          CMP    #$OD          ; EOL?
(1) 19BB FO 23         BEQ    ENDLIN
(1) 19BD 20 F3 19     JSR    QDELIM          ; IS IT A DELIMITER?
(1) 19CO DO 04         BNE    GATOM2
(1) 19C2 E8           INX
(1) 19C3 DO F1        BNE    GATOM1          ; POINT TO NEXT CHAR
(1) 19C5 OO           BRK
(1)
(1) 19C6 AO 00         GATOM2: LDY    #O
(1) 19C8 BD 00 37     GATOM3: LDA    INBUFF,X
(1) 19CB 99 00 38     STA    ATMBUF,Y
(1) 19CE C8           INY
(1) 19CF E8           INX
(1) 19D0 20 F3 19     JSR    QDELIM          ; IS IT A DELIMITER?
(1) 19D3 DO F3        BNE    GATOM3          ; CONTINUE (WILL ALWAYS BRANCH)
(1) 19D5 8D 4A 36     STA    DELIMIT
(1) 19D8 8E 44 36     STX    BUFPTR
(1) 19DB 8C 45 36     STY    ATMLEN
(1) 19DE 18           CLC
(1) 19DF 60           RTS
(1)
(1) 19E0 AO 00         ENDLIN: LDY    #O
(1) 19E2 BD 00 37     LDA    INBUFF,X          ; GET NEXT CHAR
(1) 19E5 99 00 38     STA    ATMBUF,Y
(1) 19E8 8D 4A 36     STA    DELIMIT          ; SAVE THE DELIMITER
(1) 19EB 8E 44 36     STX    BUFPTR          ; SAVE BUFFER POINTER
(1) 19EE 8C 45 36     STY    ATMLEN          ; SAVE LENGTH OF ATOM
(1) 19F1 38           SEC
(1) 19F2 60           RTS
(1)
(1) 19F3 C9 3A         QDELIM: CMP    #'          ; LOOK FOR DELIMITER
(1) 19F5 FO OE        BEQ    QDLMEQ
(1) 19F7 C9 20        CMP    #$20          ; <SPACE>
(1) 19F9 FO 0A        BEQ    QDLMEQ
(1) 19FB C9 OD        CMP    #$OD          ; <CR>
(1) 19FD FO 06        BEQ    QDLMEQ
(1) 19FF C9 3B        CMP    #'          ;
(1) 1A01 FO 02        BEQ    QDLMEQ
(1) 1A03 C9 2C        CMP    #'          ;
(1) 1A05 60         QDLMEQ: RTS          ; X = X Y = Y A = A Z=1 IF DELIMITER
(1)
(1)
(1)
1242 ; .READ GETNUM.A
1243 1A06 GETNUM
(1)
(1) 1A06 A9 00         GETNUM: LDA    #O
(1) 1A08 8D 42 36     STA    NUMBER          ; ZERO OUT THE NUMBER
(1) 1A0B 8D 43 36     STA    NUMBER+1
(1) 1A0E 20 B3 19     JSR    GTATOM          ; GET NEXT ATOM
(1) 1A11 AD 00 38     LDA    ATMBUF+0        ; GET 1ST CHAR
(1) 1A14 C9 OD        CMP    #$OD
(1) 1A16 FO 3D        BEQ    GTNMEX
(1) 1A18 A2 00        LDX    #O
(1) 1A1A BD 00 38     GNUMLP: LDA    ATMBUF,X          ; ZERO OUT X REG

```

```

(1) 1A1D CD 4A 36      CMP      DELIMIT
(1) 1A20 FO 2E        BEQ      NUMEND      ; EXIT WHEN DELIMITER REACHED
(1) 1A22 C9 30        CMP      #'O          ; LESS THAN O ?
(1) 1A24 90 OE        BLT      NUM.ER
(1) 1A26 C9 3A        CMP      #'<'9 +1>    ; ASCII 9 +1
(1) 1A28 90 OF        BLT      NUMOK
(1) 1A2A C9 47        CMP      #'<'F +1>    ; ASCII F +1
(1) 1A2C BO 06        BGE      NUM.ER
(1) 1A2E E9 06        SBC      #$06
(1) 1A30 C9 3A        CMP      #'$3A        ; 9 < CHR < A ?
(1) 1A32 BO 05        BGE      NUMOK      ; FALL THROUGH TO NUM.ER
(1) 1A34 A9 14        NUM.ER: LDA     #NUMERR   ; NUMBER ERROR
(1) 1A36 20 82 2F    JSR      ERROR
(1)
(1) 1A39 29 OF        NUMOK: AND     #$0F      ; MASK NUMBER TO LOW NIBBLE
(1) 1A3B A0 04        LDY      #4          ; SHIFT # FROM RIGHT (LOW) NIBBLE - LEFT (HIGH)
(1) 1A3D 0A          GNMLP1: ASL     A
(1) 1A3E 88          DEY
(1) 1A3F DO FC        BNE      GNMLP1
(1)
(1) 1A41 A0 04        LDY      #4          ; SHIFT NIBBLE INTO NUMBER
(1) 1A43 0A          GNMLP2: ASL     A
(1) 1A44 2E 42 36    ROL     NUMBER
(1) 1A47 2E 43 36    ROL     NUMBER+1
(1) 1A4A 88          DEY
(1) 1A4B DO F6        BNE      GNMLP2
(1) 1A4D E8          INX
(1) 1A4E DO CA        BNE      GNUMLP     ; POINT TO NEXT NUMBER IN ATOM
(1)                  ; WILL ALWAYS BRANCH
(1)
(1) 1A50 AD 42 36    NUMEND: LDA     NUMBER    ; RETURN WITH A = LOW BYTE OF NUMBER
(1) 1A53 18          CLC
(1) 1A54 60          RTS
(1)
(1) 1A55 A9 00        GTNMEX: LDA     #0          ; RETURNS ZERO IF NO NUMBER PRESENT
(1) 1A57 38          SEC          ; INIT TO ZERO
(1) 1A58 60          RTS          ; SET EOL CONDITION
(1)

```

1244
 1245
 1246
 1247

```

1249
1250 1A59 20 B3 19 DSK.NO: JSR GTATOM ; RETURN WITH DISK-TYPE # IN %X
1251 1A5C A2 00 LDX #0
1252 1A5E AD 00 38 DEV.LP: LDA ATMBUF+0
1253 1A61 DD FB 1A CMP DSKTAB,X ; 1ST CHAR
1254 1A64 DO 17 BNE NXDEV4
1255 1A66 E8 INX
1256 1A67 AD 01 38 LDA ATMBUF+1
1257 1A6A DD FB 1A CMP DSKTAB,X ; 2ND CHAR
1258 1A6D DO 0F BNE NXDEV3
1259 1A6F E8 INX
1260 1A70 AD 02 38 LDA ATMBUF+2
1261 1A73 DD FB 1A CMP DSKTAB,X ; 3RD CHAR
1262 1A76 DO 07 BNE NXDEV2
1263 1A78 E8 INX
1264 1A79 BD FB 1A LDA DSKTAB,X ; 4TH CHAR
1265 1A7C 60 RTS
1266
1267 1A7D E8 NXDEV4: INX ; ADVANCE TO NEXT DEVICE IN DEV TABLE
1268 1A7E E8 NXDEV3: INX
1269 1A7F E8 NXDEV2: INX
1270 1A80 E8 INX
1271 1A81 BD FB 1A LDA DSKTAB,X ; BRANCH UNLESS AT END OF TABLE
1272 1A84 DO D8 BNE DEV.LP
1273
1274 1A86 A9 02 LDA #DSKERR ; DISK TYPE ERROR
1275 1A88 20 82 2F JSR ERROR
1276
1277
1278 1A8B DSKTYP: ; ENTER WITH DRIVE IN A
1279 1A8B AA TAX
1280 1A8C BD C4 35 LDA DRIVES,X
1281 1A8F 60 RTS ; RETURN
1282 ; DISK TYPE # IN A
1283 ; DRIVE # IN X
1284
1285
    
```

```

1287                                     ; GET THE INDEX OF THE DEVICE IN ACC
1288 1A90 29 7F          DEV.NO: AND    # $7F          ; TURN OFF 'DEVICE' FLAG
1289 1A92 A2 00          LDY           # 0            ; INIT INDEX
1290 1A94 DD 47 1C      DEVNLP: CMP    DEVTAB,X      ; CHECK
1291 1A97 FO 14          BEQ           DEVNOK
1292 1A99 E8            INX
1293 1A9A E8            INX                       ; SKIP OVER THIS DEVICE
1294 1A9B E8            INX
1295 1A9C E8            INX
1296 1A9D E8            INX
1297 1A9E E8            INX
1298 1A9F E8            INX
1299 1AA0 E8            INX
1300 1AA1 E8            INX
1301 1AA2 E8            INX
1302 1AA3 BC 47 1C      LDY           DEVTAB,X
1303 1AA6 DO EC          BNE           DEVNLP        ; IF ZERO THEN END OF LIST
1304 1AA8 A9 17          LDA           # DEVERR
1305 1AAA 20 82 2F      JSR           ERROR
1306 1AAD 60            DEVNOK: RTS              ; RTNS X = POINTER TO DEVICE
1307
1308
1309
1310
1311
1312
1313                                     ; ALLOW INDIRECT JSR's
1314
1315
1316 1AAE 6C 10 36      EXEINP: JMP    (INPJMP)        ; INPUT INDIRECT JSR
1317 1AB1 6C 20 36      EXEOUT: JMP   (OUTJMP)         ; OUTPUT INDIRECT JSR
1318 1AB4 6C 30 36      EXETMP: JMP   (TMPJMP)         ; TEMPORARY INDIRECT JSR
1319 1AB7 6C 40 36      EXECMD: JMP   (CMDJMP)         ; COMMAND INDIRECT JSR
1320
1321
1322
1323

```

```

1325
1326 1ABA 20 CO 1A HXTDCP: JSR HEXDEC ; CONVERT IT
1327 1ABD 4C C3 2B HXTDCP: JMP PRTBYT ; PRINT IT
1328 1ACO HEXDEC:
1329 1ACO C9 00 CMP #0 ; CONVERT HEX # IN ACC TO BCD IN ACC
1330 1AC2 FO OE BEQ H.D.EX ; EXIT IF ZERO
1331 1AC4 8D 4B 36 STA ACC ; TEMP SAVE ACC
1332 1AC7 F8 SED ; SET DECIMAL MODE
1333 1AC8 A9 00 LDA #0
1334 1ACA 18 H.D.LP: CLC
1335 1ACB 69 01 ADC #1
1336 1ACD CE 4B 36 DEC ACC
1337 1ADO DO F8 BNE H.D.LP
1338 1AD2 D8 H.D.EX: CLD ; RESET TO BINARY MODE
1339 1AD3 60 RTS ; RETURN X = X Y = Y A = DEC (A)
1340
1341
1342
1343
1344
1345
1346
1347 1AD4 DECHEX:
1348 1AD4 C9 00 CMP #0
1349 1AD6 FO OA BEQ D.H.EX
1350 1AD8 F8 SED ; SET DECIMAL MODE
1351 1AD9 A2 00 LDX #0
1352 1ADB E8 D.H.LP: INX
1353 1ADC 38 SEC ; SET UP FOR SUBTRACT
1354 1ADD E9 01 SBC #1
1355 1ADF DO FA BNE D.H.LP
1356 1AE1 8A TXA ; PUT HEX VALUE INTO ACC
1357 1AE2 D8 D.H.EX: CLD ; RESET TO BINARY MODE
1358
1359 1AE3 60 RTS
1360

```

```
1362 .SBTTL Sub-Pgm Subroutines
1363
1364 1AE4 20 FO 1A SUBINI: JSR GETDRV ; GET DRIVE #
1365 1AE7 20 61 2E JSR RECAL ; RECALIBRATE DRIVE #O
1366 1AEA A9 80 LDA #USRRAM^ ; SET BUFFER ADDRESS
1367 1AEC 8D 3B 36 STA TMPAGE
1368 1AEF 60 RTS
1369
1370
1371
1372 1AFO 20 06 1A GETDRV: JSR GETNUM ; GET DRIVE #
1373 1AF3 08 PHP ; SAVE CARRY STATUS
1374 1AF4 29 03 AND #$03
1375 1AF6 8D 33 36 STA TMPDRV ; SET TEMP DRIVE
1376 1AF9 28 PLP ; RESTORE CARRY STATUS
1377 1AFA 60 RTS
1378
1379
1380
1381
```

```

1383          .SBTTL  Driver Tables
1384
1385          ; DISK TYPE TABLE          ; DISK TYPES $00 - $3F (64 MAX)
1386          ;                          ; MAXIMUM OF 127 DISK TYPES
1387 1AFB  2D  2D  2D  DSKTAB: .ASCII  '----'<0>      ; ERROR CONDITON
1388 1AFF  43  44  53      .ASCII  'CDS'<1>      ; CODDOS
1389 1B03  46  4C  58      .ASCII  'FLX'<2>      ; FLEX
1390 1B07  43  50  4D      .ASCII  'CPM'<3>      ; CP/M
1391 1B0B  49  42  4D      .ASCII  'IBM'<4>      ; IBM
1392 1B0F  52  53  58      .ASCII  'RSX'<5>      ; RSX-11
1393 1B13  48  44  45      .ASCII  'HDE'<6>      ; HDE DOS
1394 1B17  54  52  53      .ASCII  'TRS'<7>      ; TRS-80
1395 1B1B  50  53  31      .ASCII  'PS1'<8>      ; UCSD PASCAL  SINGLE DENSITY
1396 1B1F  50  53  32      .ASCII  'PS2'<9>      ; UCSD PASCAL  MTU-DOUBLE DENSITY
1397 1B23  55  53  52      .ASCII  'USR'<10>     ; USER DEVICE
1398 1B27  43  4D  58      .ASCII  'CMX'<11>     ; Expansion
1399 1B2B  2D  2D  2D      .ASCII  '----'<12>     ; Expansion
1400 1B2F  2D  2D  2D      .ASCII  '----'<13>     ; Expansion
1401 1B33  2D  2D  2D      .ASCII  '----'<14>     ; Expansion
1402 1B37  00  00          .ADDR  0
1403
1404          .MACRO  TABLE  LB
1405 LB'TAB: .ADDR  ERR'LB      ; ERROR          0
1406          .ADDR  CDS'LB      ; CODDOS         1
1407          .ADDR  FLX'LB      ; FLEX           2
1408          .ADDR  CPM'LB      ; CP/M           3
1409          .ADDR  IBM'LB      ; IBM            4
1410          .ADDR  RSX'LB      ; RSX-11        5
1411          .ADDR  HDE'LB      ; H.D.E.        6
1412          .ADDR  TRS'LB      ; TRS-80        7
1413          .ADDR  PS1'LB      ; PASCAL DEN-1  8
1414          .ADDR  PS2'LB      ; PASCAL DEN-2  9
1415          .ADDR  USR'LB      ; USER         10
1416          .ADDR  CMX'LB      ; Expansion     11
1417          .ADDR  -1          ; Expansion     12
1418          .ADDR  -1          ; Expansion     13
1419          .ADDR  -1          ; Expansion     14
1420
1421          .ENDM
1422
1423          ; TABLES
1423 1B39          TABLE  DEF      ; DEFINE
1424 (1) 1B39  D6  27  DEFTAB: .ADDR  ERRDEF      ; ERROR          0
1425 (1) 1B3B  A9  1C      .ADDR  CDSDEF      ; CODDOS         1
1426 (1) 1B3D  AC  1D      .ADDR  FLXDEF      ; FLEX           2
1427 (1) 1B3F  6A  20      .ADDR  CPMDEF      ; CP/M           3
1428 (1) 1B41  E3  24      .ADDR  IBMDEF      ; IBM            4
1429 (1) 1B43  FC  26      .ADDR  RSXDEF      ; RSX-11        5
1430 (1) 1B45  00  27      .ADDR  HDEDEF      ; H.D.E.        6
1431 (1) 1B47  04  27      .ADDR  TRSDEF      ; TRS-80        7
1432 (1) 1B49  B7  27      .ADDR  PS1DEF      ; PASCAL DEN-1  8
1433 (1) 1B4B  BB  27      .ADDR  PS2DEF      ; PASCAL DEN-2  9
1434 (1) 1B4D  BF  27      .ADDR  USRDEF      ; USER         10
1435 (1) 1B4F  DA  27      .ADDR  CMXDEF      ; EXPANSION     11
1436 (1) 1B51  FF  FF      .ADDR  -1          ; EXPANSION     12
1437 (1) 1B53  FF  FF      .ADDR  -1          ; EXPANSION     13

```


(1)	1B55	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1424	1B57				TABLE	DIR		; DIRECTORY	
(1)	1B57	D6	27	DIRTAB:	.ADDR	ERRDIR		; ERROR	0
(1)	1B59	C5	1C		.ADDR	CDSDIR		; CODOS	1
(1)	1B5B	AE	1D		.ADDR	FLXDIR		; FLEX	2
(1)	1B5D	6C	20		.ADDR	CPMDIR		; CP/M	3
(1)	1B5F	E5	24		.ADDR	IBMDIR		; IBM	4
(1)	1B61	FC	26		.ADDR	RSXDIR		; RSX-11	5
(1)	1B63	00	27		.ADDR	HDEDIR		; H.D.E.	6
(1)	1B65	07	27		.ADDR	TRSDIR		; TRS-80	7
(1)	1B67	B7	27		.ADDR	PS1DIR		; PASCAL DEN-1	8
(1)	1B69	BB	27		.ADDR	PS2DIR		; PASCAL DEN-2	9
(1)	1B6B	C2	27		.ADDR	USRDIR		; USER	10
(1)	1B6D	DC	27		.ADDR	CMXDIR		; EXPANSION	11
(1)	1B6F	FF	FF		.ADDR	-1		; EXPANSION	12
(1)	1B71	FF	FF		.ADDR	-1		; EXPANSION	13
(1)	1B73	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1425	1B75				TABLE	OPN		; OPEN DEVICE	
(1)	1B75	D8	27	OPNTAB:	.ADDR	ERROPN		; ERROR	0
(1)	1B77	54	1D		.ADDR	CDSOPN		; CODOS	1
(1)	1B79	42	1E		.ADDR	FLXOPN		; FLEX	2
(1)	1B7B	DD	20		.ADDR	CPMOPN		; CP/M	3
(1)	1B7D	1E	25		.ADDR	IBMOPN		; IBM	4
(1)	1B7F	FE	26		.ADDR	RSXOPN		; RSX-11	5
(1)	1B81	02	27		.ADDR	HDEOPN		; H.D.E.	6
(1)	1B83	B5	27		.ADDR	TRSOPN		; TRS-80	7
(1)	1B85	B9	27		.ADDR	PS1OPN		; PASCAL DEN-1	8
(1)	1B87	BD	27		.ADDR	PS2OPN		; PASCAL DEN-2	9
(1)	1B89	C5	27		.ADDR	USROPN		; USER	10
(1)	1B8B	6C	28		.ADDR	CMXOPN		; EXPANSION	11
(1)	1B8D	FF	FF		.ADDR	-1		; EXPANSION	12
(1)	1B8F	FF	FF		.ADDR	-1		; EXPANSION	13
(1)	1B91	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1426	1B93				TABLE	CLO		; CLOSE DEVICE	
(1)	1B93	D8	27	CLOTAB:	.ADDR	ERRCLO		; ERROR	0
(1)	1B95	A2	1D		.ADDR	CDSCLC		; CODOS	1
(1)	1B97	29	20		.ADDR	FLXCLO		; FLEX	2
(1)	1B99	55	24		.ADDR	CPMCLO		; CP/M	3
(1)	1B9B	DC	25		.ADDR	IBMCLC		; IBM	4
(1)	1B9D	FE	26		.ADDR	RSXCLO		; RSX-11	5
(1)	1B9F	02	27		.ADDR	HDECLO		; H.D.E.	6
(1)	1BA1	B5	27		.ADDR	TRSCLO		; TRS-80	7
(1)	1BA3	B9	27		.ADDR	PS1CLO		; PASCAL DEN-1	8
(1)	1BA5	BD	27		.ADDR	PS2CLO		; PASCAL DEN-2	9
(1)	1BA7	CE	27		.ADDR	USRCLO		; USER	10
(1)	1BA9	2A	2A		.ADDR	CMXCLO		; EXPANSION	11
(1)	1BAB	FF	FF		.ADDR	-1		; EXPANSION	12
(1)	1BAD	FF	FF		.ADDR	-1		; EXPANSION	13
(1)	1BAF	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1427	1BB1				TABLE	INP		; INPUT FROM DEVICE	
(1)	1BB1	D8	27	INPTAB:	.ADDR	ERRINP		; ERROR	0

(1)	1BB3	97	1D		.ADDR	CDSINP		; CODOS	1
(1)	1BB5	8F	1F		.ADDR	FLXINP		; FLEX	2
(1)	1BB7	E8	22		.ADDR	CPMINP		; CP/M	3
(1)	1BB9	2F	26		.ADDR	IBMINP		; IBM	4
(1)	1BBB	FE	26		.ADDR	RSXINP		; RSX-11	5
(1)	1BBD	O2	27		.ADDR	HDEINP		; H.D.E.	6
(1)	1BBF	B5	27		.ADDR	TRSINP		; TRS-80	7
(1)	1BC1	B9	27		.ADDR	PS1INP		; PASCAL DEN-1	8
(1)	1BC3	BD	27		.ADDR	PS2INP		; PASCAL DEN-2	9
(1)	1BC5	C8	27		.ADDR	USRINP		; USER	10
(1)	1BC7	2C	2A		.ADDR	CMXINP		; EXPANSION	11
(1)	1BC9	FF	FF		.ADDR	-1		; EXPANSION	12
(1)	1BCB	FF	FF		.ADDR	-1		; EXPANSION	13
(1)	1BCD	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1428	1BCF				TABLE	OUT		; OUTPUT TO DEVICE	
(1)	1BCF	D8	27	OUTTAB:	.ADDR	ERRROUT		; ERROR	0
(1)	1BD1	9C	1D		.ADDR	CDSOUT		; CODOS	1
(1)	1BD3	FO	1F		.ADDR	FLXOUT		; FLEX	2
(1)	1BD5	C5	23		.ADDR	CPMOUT		; CP/M	3
(1)	1BD7	95	26		.ADDR	IBMOUT		; IBM	4
(1)	1BD9	FE	26		.ADDR	RSXOUT		; RSX-11	5
(1)	1BDB	O2	27		.ADDR	HDEOUT		; H.D.E.	6
(1)	1BDD	B5	27		.ADDR	TRSOUT		; TRS-80	7
(1)	1BDF	B9	27		.ADDR	PS1OUT		; PASCAL DEN-1	8
(1)	1BE1	BD	27		.ADDR	PS2OUT		; PASCAL DEN-2	9
(1)	1BE3	CB	27		.ADDR	USROUT		; USER	10
(1)	1BE5	88	2A		.ADDR	CMXOUT		; EXPANSION	11
(1)	1BE7	FF	FF		.ADDR	-1		; EXPANSION	12
(1)	1BE9	FF	FF		.ADDR	-1		; EXPANSION	13
(1)	1BEB	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1429									
1430	1BED				TABLE	HDR		; HEADER	
(1)	1BED	2D	34	HDRTAB:	.ADDR	ERRHDR		; ERROR	0
(1)	1BEF	13	33		.ADDR	CDSHDR		; CODOS	1
(1)	1BF1	2A	33		.ADDR	FLXHDR		; FLEX	2
(1)	1BF3	51	33		.ADDR	CPMHDR		; CP/M	3
(1)	1BF5	71	33		.ADDR	IBMHDR		; IBM	4
(1)	1BF7	76	33		.ADDR	RSXHDR		; RSX-11	5
(1)	1BF9	7E	33		.ADDR	HDEHDR		; H.D.E.	6
(1)	1BFB	86	33		.ADDR	TRSHDR		; TRS-80	7
(1)	1BFD	A5	33		.ADDR	PS1HDR		; PASCAL DEN-1	8
(1)	1BFF	C5	33		.ADDR	PS2HDR		; PASCAL DEN-2	9
(1)	1C01	E5	33		.ADDR	USRHDR		; USER	10
(1)	1C03	F2	33		.ADDR	CMXHDR		; EXPANSION	11
(1)	1C05	FF	FF		.ADDR	-1		; EXPANSION	12
(1)	1C07	FF	FF		.ADDR	-1		; EXPANSION	13
(1)	1C09	FF	FF		.ADDR	-1		; EXPANSION	14
(1)									
1431	1COB				TABLE	DBS		; Density & Bytes/Sector	
(1)	1COB	FF	FF	DBSTAB:	.ADDR	ERRDBS		; ERROR	0
(1)	1COD	O1	O2		.ADDR	CDSDBS		; CODOS	1
(1)	1COF	O1	O1		.ADDR	FLXDBS		; FLEX	2
(1)	1C11	O0	O1		.ADDR	CPMDBS		; CP/M	3

(1)	1C13	00	01	.ADDR	IBMDBS	; IBM	4
(1)	1C15	00	01	.ADDR	RSXDBS	; RSX-11	5
(1)	1C17	FF	FF	.ADDR	HDEDBS	; H.D.E.	6
(1)	1C19	01	02	.ADDR	TRSDBS	; TRS-80	7
(1)	1C1B	00	01	.ADDR	PS1DBS	; PASCAL DEN-1	8
(1)	1C1D	01	02	.ADDR	PS2DBS	; PASCAL DEN-2	9
(1)	1C1F	FF	FF	.ADDR	USRDBS	; USER	10
(1)	1C21	00	01	.ADDR	CMXDBS	; EXPANSION	11
(1)	1C23	FF	FF	.ADDR	-1	; EXPANSION	12
(1)	1C25	FF	FF	.ADDR	-1	; EXPANSION	13
(1)	1C27	FF	FF	.ADDR	-1	; EXPANSION	14
(1)							
1432	1C29			TABLE	SCT	; First/Last Sector	
(1)	1C29	FF	FF	SCTTAB: .ADDR	ERRSCT	; ERROR	0
(1)	1C2B	19	00	.ADDR	CDSSCT	; CODOS	1
(1)	1C2D	0F	01	.ADDR	FLXSCT	; FLEX	2
(1)	1C2F	1A	01	.ADDR	CPMSCT	; CP/M	3
(1)	1C31	1A	01	.ADDR	IBMSCT	; IBM	4
(1)	1C33	1A	01	.ADDR	RSXSCT	; RSX-11	5
(1)	1C35	FF	FF	.ADDR	HDESCT	; H.D.E.	6
(1)	1C37	1A	01	.ADDR	TRSSCT	; TRS-80	7
(1)	1C39	1A	01	.ADDR	PS1SCT	; PASCAL DEN-1	8
(1)	1C3B	1A	01	.ADDR	PS2SCT	; PASCAL DEN-2	9
(1)	1C3D	FF	FF	.ADDR	USRST	; USER	10
(1)	1C3F	0A	01	.ADDR	CMXSCT	; EXPANSION	11
(1)	1C41	FF	FF	.ADDR	-1	; EXPANSION	12
(1)	1C43	FF	FF	.ADDR	-1	; EXPANSION	13
(1)	1C45	FF	FF	.ADDR	-1	; EXPANSION	14
(1)							
1433							

```

1435
1436
1437 .MACRO DEVICE LB
1438 .BYTE 'LB ; DEVICE CHARACTER
1439 .BYTE O ; MAKE EVEN
1440 .ADDR LB'.OPN ; OPEN ROUTINE
1441 .ADDR LB'.CLO ; CLOSE ROUTINE
1442 .ADDR LB'.INP ; INPUT ROUTINE
1443 .ADDR LB'.OUT ; OUTPUT ROUTINE
1444 .ENDM
1445 ; DEVICES $40 - $5A (26 MAX)
1446 ; 'A' - 'Z'
1447 ; MAXIMUM OF 25 DEVICES
1448 DEVTAB: DEVICE N
1449 (1) 1C47 4E .BYTE 'N ; DEVICE CHARACTER
1450 (1) 1C48 00 .BYTE O ; MAKE EVEN
1451 (1) 1C49 8F 2A .ADDR N.OPN ; OPEN ROUTINE
1452 (1) 1C4B 8F 2A .ADDR N.CLO ; CLOSE ROUTINE
1453 (1) 1C4D 8B 2A .ADDR N.INP ; INPUT ROUTINE
1454 (1) 1C4F 8F 2A .ADDR N.OUT ; OUTPUT ROUTINE
1455 1449 1C51 DEVICE C ; CONSOLE
1456 (1) 1C51 43 .BYTE 'C ; DEVICE CHARACTER
1457 (1) 1C52 00 .BYTE O ; MAKE EVEN
1458 (1) 1C53 91 2A .ADDR C.OPN ; OPEN ROUTINE
1459 (1) 1C55 94 2A .ADDR C.CLO ; CLOSE ROUTINE
1460 (1) 1C57 96 2A .ADDR C.INP ; INPUT ROUTINE
1461 (1) 1C59 A1 2A .ADDR C.OUT ; OUTPUT ROUTINE
1462 1450 1C5B DEVICE P ; PRINTER
1463 (1) 1C5B 50 .BYTE 'P ; DEVICE CHARACTER
1464 (1) 1C5C 00 .BYTE O ; MAKE EVEN
1465 (1) 1C5D DE 2A .ADDR P.OPN ; OPEN ROUTINE
1466 (1) 1C5F E6 2A .ADDR P.CLO ; CLOSE ROUTINE
1467 (1) 1C61 EB 2A .ADDR P.INP ; INPUT ROUTINE
1468 (1) 1C63 ED 2A .ADDR P.OUT ; OUTPUT ROUTINE
1469 1451 1C65 DEVICE M ; MEMORY
1470 (1) 1C65 4D .BYTE 'M ; DEVICE CHARACTER
1471 (1) 1C66 00 .BYTE O ; MAKE EVEN
1472 (1) 1C67 0B 2B .ADDR M.OPN ; OPEN ROUTINE
1473 (1) 1C69 29 2B .ADDR M.CLO ; CLOSE ROUTINE
1474 (1) 1C6B 31 2B .ADDR M.INP ; INPUT ROUTINE
1475 (1) 1C6D 40 2B .ADDR M.OUT ; OUTPUT ROUTINE
1476 1452 1C6F DEVICE U ; USER DEVICE
1477 (1) 1C6F 55 .BYTE 'U ; DEVICE CHARACTER
1478 (1) 1C70 00 .BYTE O ; MAKE EVEN
1479 (1) 1C71 66 2B .ADDR U.OPN ; OPEN ROUTINE
1480 (1) 1C73 6F 2B .ADDR U.CLO ; CLOSE ROUTINE
1481 (1) 1C75 69 2B .ADDR U.INP ; INPUT ROUTINE
1482 (1) 1C77 6C 2B .ADDR U.OUT ; OUTPUT ROUTINE
1483 1453 1C79 00 00 .WORD O
1484 1454
    
```

```
1456      .SBTTL Driver Routine Specifications
1457
1458      ; All routines MUST end in a RTS.
1459
1460      ; INPUT routines must return with the Carry flag clear
1461      ; until the End of File is reached, then carry set indicating EOF.
1462      ; (Not when the last char in the file is reached).
1463      ;
1464      ; Open routines clear Carry if file found.
1465      ; OPEN:          TMPIOF = 0 == INPUT.
1466      ;                TMPIOF = 1 == OUTPUT.
1467      ;
1468      ; A new file must be CREATED on the host system
1469      ; and only WRITTEN to with DSKFTP.
1470      ;
1471      ; Close routines clear Carry if successfully closed.
1472      ; Input files are NOT 'CLOSED'; only output files.
1473      ;
1474      ;
1475      ; Common Labels:
1476      ;     xxx      =>      Disk Type
1477      ;     y        =>      Sub-Section
1478      ;     xxxOzz   =>      Open
1479      ;     xxxOFF   =       Open File Found
1480      ;     xxxONF   =       Open File NOT Found
1481      ;     xxxIzz   =       Input
1482      ;     xxxIEF   =       INPUT: END of FILE
1483      ;     xxxOzz   =       Output
1484      ;     xxxyNS   =       Next Sector
1485      ;     xxxyNT   =       Next Track
1486      ;     xxxLPn   =       Loop # n
1487      ;     xxxyEX   =       Exit
1488      ;     xxxCTS   =       Calculate Track Sector
1489      ;     xxxySR   =       SubRoutine
1490      ;     xxxyER   =       ERROR CONDITION
1491      ;     xxxyCC   =       CARRY CLEAR RETURN
1492      ;     xxxyCS   =       CARRY SET RETURN
1493
```

```
1495          .SBTTL Directory Subroutines
1496
1497          ; SUBROUTINES TO GET, PRINT, & ADVANCE POINTER
1498          ; TO NEXT BYTE OF THE DIRECTORY BUFFER
1499
1500 1C7B BD 00 80 DIRCHR: LDA DIRBUF,X ; PRINT CHAR
1501 1C7E 20 1C 2C      JSR PRVCHR
1502 1C81 E8          INX
1503 1C82 60          RTS
1504
1505 1C83 BD 00 80 DIRBYT: LDA DIRBUF,X ; PRINT HEX BYTE
1506 1C86 20 C3 2B      JSR PRTBYT
1507 1C89 E8          INX
1508 1C8A 60          RTS
1509
1510 1C8B 20 83 1C DIRBT4: JSR DIRBYT
1511 1C8E 20 83 1C      JSR DIRBYT
1512 1C91 20 83 1C      JSR DIRBYT
1513 1C94 20 83 1C      JSR DIRBYT
1514 1C97 20 09 10      JSR CTYSPA
1515 1C9A 60          RTS
1516
1517 1C9B BD 00 80 DIRDEC: LDA DIRBUF,X ; PRINT DECIMAL BYTE
1518 1C9E 20 BA 1A      JSR HXTDCP ; CONVERT AND PRINT
1519 1CA1 E8          INX
1520 1CA2 60          RTS
1521
1522
1523 1CA3 A9 2E DIRPER: LDA #'
1524 1CA5 20 95 2B      JSR CTYOUT
1525 1CA8 60          RTS
1526
```

```

1528 .SBTTL CODOS Disk Drivers
1529
1530 ; CODOS Chanel Oriented Disk Operating System
1531
1532 ; NEC-765 FDC CONTROLLER CHIP
1533 ; DOUBLE DENSITY
1534 ; 256 BYTES/SECTOR
1535 ; 26 SECTORS/TRACK (0-25).
1536
1537
1538 ; SYSTEM OVERHEAD:
1539 ; TRACK 12 DIRECTORY
1540 ; SECTOR 1-16?
1541 ;
1542
1543
1544 0201 CDSDBS = $0201 ; DOUBLE DENSITY, 256 BYTES/SECTOR
1545 0019 CDS SCT = $0019 ; SECTOR INFO
1546
1547 1CA9 A9 BD CDSDEF: LDA #CDSTXT&$FF
1548 1CAB 85 BA STA U5+0
1549 1CAD A9 35 LDA #CDSTXT^
1550 1CAF 85 BB STA U5+1
1551 1CB1 AD 33 36 LDA TMPDRV ; GET DRIVE #
1552 1CB4 09 30 ORA #$30 ; MAKE INTO ASCII #
1553 1CB6 8D C2 35 STA CDSTXT+5
1554 1CB9 A9 80 LDA #$80 ; ENABLE SVCs
1555 1CBB 85 EE STA SVCENA
1556 1CBD SVC 13
(1) 1CBD 00 OD .BYTE 0,13
1557 1CBF A9 00 LDA #0 ; DISABLE SVCs
1558 1CC1 85 EE STA SVCENA
1559 1CC3 18 CLC
1560 1CC4 60 RTS
1561
1562
1563 1CC5 A9 OC CDSDIR: LDA #12 ; DIRECTORY TRACK IS #12
1564 1CC7 8D 15 36 STA INPTRK ; [UNCHANGED, = 12]
1565 1CCA A9 00 LDA #0 ; SECTOR ZERO
1566 1CCC 8D 16 36 STA INPSCT
1567 1CCF 20 AF 2C JSR RDSCTR ; GET BAT SECTOR TO EXTRACT V.S.N.
1568 1CD2 AD FB 80 LDA DIRBUF+$FB ; VSN HIGH BYTE
1569 1CD5 20 C3 2B JSR PRTBYT
1570 1CD8 AD FA 80 LDA DIRBUF+$FA ; VSN LOW BYTE
1571 1CDB 20 C3 2B JSR PRTBYT
1572 1CDE AD FD 80 LDA DIRBUF+$FD ; # OF FILES
1573 1CE1 8D 4E 36 STA CDSTMP
1574
1575 1CE4 A9 10 LDA #16 ; SECTOR 16 (DISK NAME?)
1576 1CE6 8D 16 36 STA INPSCT
1577 1CE9 20 AF 2C JSR RDSCTR
1578 1CEC A9 01 LDA #1 ; FIRST DIRECTORY INFO SECTOR
1579 1CEE 8D 16 36 STA INPSCT
1580 1CF1 20 AF 2C CDSLPO: JSR RDSCTR
1581 1CF4 A2 00 LDX #0 ; INIT INDEX POINTER
    
```

```

1582 1CF6 BD 00 80 CDSL1: LDA DIRBUF,X ; CODOS DIRECTORY LOOP
1583 1CF9 FO 47 CDSL1: BEQ CDSDEX ; CODOS DIR EXIT
1584 1CFB E8 INX
1585 1CFC BD 00 80 LDA DIRBUF,X ; CHECK IF ACTIVE ENTRY
1586 1CFF DO 09 BNE CDSDE ; BRANCH IF ACTIVE
1587 1D01 8A TXA ; FALL THROUGH IF NOT
1588 1D02 18 CLC
1589 1D03 69 OF ADC #16-1
1590 1D05 AA TAX
1591 1D06 DO F1 BNE CDSL1 ; WILL ALWAYS BRANCH
1592 1D08 FO 2E BEQ CDSDE1
1593 ; CODOS DON'T SKIP ENTRY
1594 1DOA 20 FO 2B CSDSE: JSR CRLFSQ ; NEW LINE
1595 1DOD 20 40 2C JSR SPACE2 ; TWO SPACES
1596
1597 1D10 AO OE LDY #14 ; 14 CHARS IN NAME
1598 1D12 BD 00 80 CDSL2: LDA DIRBUF,X ; CODOS DIRECTORY NAME PRINT LOOP
1599 1D15 20 1C 2C JSR PRVCHR ; PRINT VALID CHAR
1600 1D18 C9 2E CMP #' ; CHECK IF END OF NAME
1601 1D1A FO 06 BEQ CDSEXT ; IF SO GO DO EXTENSION (DONE SO EXTRA CHRS
1602 1D1C E8 INX ; ADVANCE POINTER DON'T GET PRINTED)
1603 1D1D 88 DEY
1604 1D1E DO F2 BNE CDSL2
1605 1D20 00 BRK ; IF IT GETS HERE THEN INVALID FILENAME
1606 1D21 00 BRK
1607 1D22 BD 01 80 CDSEXT: LDA DIRBUF+1,X ; Get File Extension
1608 1D25 20 1C 2C JSR PRVCHR ; PRINT THE FILE NAME EXTENSION
1609 1D28 E8 CDSL3: INX ; FILL THE REST OF THE SPACE WITH SPACES
1610 1D29 20 09 10 JSR CTYSPA ; <space>
1611 1D2C 88 DEY
1612 1D2D DO F9 BNE CDSL3
1613
1614 1D2F 20 40 2C JSR SPACE2
1615 1D32 20 83 1C JSR DIRBYT ; GET & PRINT (& INX) BAT ENTRY FOR FILE
1616 1D35 8A TXA ; JUST FOR TEST
1617 1D36 DO BE BNE CDSL1 ; IF NOT CONTINUE WITH DIRECTORY
1618 1D38 EE 16 36 CDSDE1: INC INPSCT ; ADVANCE TO NEXT SECTOR
1619 1D3B AD 16 36 LDA INPSCT
1620 1D3E C9 11 CMP #17 ; CHECK IF DONE
1621 1D40 90 AF BLT CDSLPO ; GO BACK IF NOT
1622 1D42 20 EO 2B CSDSEX: JSR CRLF ; ELSE FALL THROUGH IF DONE
1623 1D45 A9 23 LDA #'#
1624 1D47 20 95 2B JSR CTYOUT
1625 1D4A 20 09 10 JSR CTYSPA
1626 1D4D AD 4E 36 LDA CDSTMP
1627 1D50 20 BA 1A JSR HXTDCP ; PRINT IN DECIMAL
1628 1D53 60 RTS
1629

```



```

1631
1632 1D54                CDSOPN:
1633 1D54 20 B3 19      JSR   GTATOM      ; GET THE FILE NAME
1634 1D57 AD 45 36      LDA   ATMLN
1635 1D5A C9 03         CMP   #3
1636 1D5C 90 34         BLT   CDSOIF      ; ILLEGAL FILE NAME
1637 1D5E A9 80         LDA   #$80       ; ENABLE SVCs
1638 1D60 85 EE         STA   SVCENA
1639 1D62 A9 00         LDA   #ATMBUF&$$FF ; GET POINTER TO FILE NAME
1640 1D64 85 B6         STA   U3+0
1641 1D66 A9 38         LDA   #ATMBUF^
1642 1D68 85 B7         STA   U3+1
1643
1644 1D6A AD 3C 36      LDA   TMPIOF
1645 1D6D DO 02         BNE   CDSOOU      ; IF OUTPUT
1646 1D6F FO 09         BEQ   CDSOIN      ; IF INPUT
1647
1648 1D71 A2 06         CDSOOU: LDX   #6       ; CHANNEL 6 FOR OUTPUT
1649 1D73 AD 33 36      LDA   TMPDRV      ; DRIVE #
1650 1D76                SVC   21          ; OPEN THE FILE
    (1) 1D76 00 15      .BYTE 0,21
1651 1D78 18          CLC
1652 1D79 60          RTS
1653
1654 1D7A A2 05         CDSOIN: LDX   #5       ; CHANNEL 5 FOR INPUT
1655 1D7C AD 33 36      LDA   TMPDRV      ; DRIVE #
1656 1D7F                SVC   21          ; OPEN THE FILE
    (1) 1D7F 00 15      .BYTE 0,21
1657 1D81 10 02         BPL   CDSONF      ; IF FILE EXISTS THEN OKAY
1658 1D83 18          CLC
1659 1D84 60          RTS               ; LEAVE SVCs ENABLED
1660
1661 1D85 A2 05         CDSONF: LDX   #5       ; FREE CHANNEL 5, (DELETE CREATED FILE)
1662 1D87                SVC   22          ; FRE CHANNEL SVC
    (1) 1D87 00 16      .BYTE 0,22
1663 1D89 A9 00         LDA   #0          ; DISABLE SVCs
1664 1D8B 85 EE         STA   SVCENA
1665 1D8D A9 11         LDA   #OPNERR
1666 1D8F 20 82 2F      JSR   ERROR
1667
1668 1D92 A9 13         CDSOIF: LDA   #INVFNM ; ILLEGAL FILE NAME ERROR
1669 1D94 20 82 2F      JSR   ERROR
1670
1671 1D97 A2 05         CDSINP: LDX   #5
1672 1D99                SVC   3           ; INPUT FROM CHANNEL 5
    (1) 1D99 00 03      .BYTE 0,3
1673 1D9B 60          RTS               ; SVC 3 RTNs WITH CARRY SET APPROPRIATELY
1674
1675
1676 1D9C A2 06         CDSOUT: LDX   #6
1677 1D9E                SVC   4           ; OUTPUT ON CHANNEL 6
    (1) 1D9E 00 04      .BYTE 0,4
1678 1DA0 18          CLC
1679 1DA1 60          RTS
1680
    
```

1681	1DA2	A2	06	CDSCL0:	LDX	#6	
1682	1DA4				SVC	22	; FREE (CLOSE) OUTPUT CHANNEL
(1)	1DA4	00	16		.BYTE	0,22	
1683	1DA6	A9	00		LDA	#0	; DISABLE SVCs
1684	1DA8	85	EE		STA	SVCENA	
1685	1DAA	18			CLC		; IF NECESSARY
1686	1DAB	60			RTS		
1687							
1688							
1689							
1690							

```

1692          .SBTTL  FLEX  Disk Drivers
1693
1694          ; FLEX  FLEXible Operating System
1695
1696          ;          1771/1791 FDC CONTROLLER CHIP
1697          ; 256  BYTES/SECTOR
1698          ; 15  SECTORS/TRACK  (1-F)
1699
1700          ; SYSTEM OVERHEAD:
1701          ;          TRACK 0 SCTR 3  DISK INFO TRACK
1702          ;          REST OF TRACK 0 IS DIRECTORY
1703
1704          ;          1ST 4 BYTES OF NORMAL SECTOR ARE LINKAGE INFO
1705
1706          ;          00-07  FILE NAME
1707          ;          08-0A  EXT
1708          ;          0B    PROTECTION ATTRIBUTES
1709          ;          0C    RESERVED
1710          ;          0D-0E  START TRCK/SCTR
1711          ;          0F-10  END TRCK/SCTR
1712          ;          11-12  SIZE
1713          ;          13    FILE SECTOR MAP INDICATOR
1714          ;          14    RESERVED
1715          ;          15    MONTH
1716          ;          16    DAY
1717          ;          17    YEAR
1718
1719          ;          24 BYTES/ENTRY
1720          ;          10 ENTRYS/SECTOR
1721
1722          0101  FLXDBS =          $0101          ; SINGLE DENSITY, 256 BYTES/SECTOR
1723          010F  FLXSCT =          $010F          ; SECTOR INFO
1724
1725          1DAC  FLXDEF:
1726          1DAC  18          CLC
1727          1DAD  60          RTS
1728
    
```

```

1730
1731 1DAE A9 00          FLXDIR: LDA #0          ; TRACK 0
1732 1DB0 8D 15 36      STA INPTRK
1733 1DB3 A9 03          LDA #3          ; SECTOR 3
1734 1DB5 8D 16 36      STA INPSCT
1735 1DB8 20 AF 2C      JSR RDSCTR      ; GET DISK NAME SECTOR
1736 1DBB A2 10          LDX #16         ; SKIP OVER 1st 16 BYTES
1737 1DBD A0 OA          LDY #10         ; 10 CHARS IN DISK NAME
1738 1DBF 20 7B 1C      FLXDPN: JSR DIRCHR ; PRINT VALID CHAR & INC POINTER
1739 1DC2 88            DEY
1740 1DC3 DO FA          BNE FLXDPN      ; PRINT DIR FILE NAME
1741
1742 1DC5 A9 05          LDA #5          ; SECTOR 5
1743 1DC7 8D 16 36      STA INPSCT
1744 1DCA 20 AF 2C      FLXDNS: JSR RDSCTR ; READ THE (NEXT) SECTOR
1745 1DCD A2 10          LDX #16         ; SKIP 1ST 16 BYTES OF SECTOR
1746 1DCF 20 FO 2B      FLXDLP: JSR CRLFSQ ; NEW LINE
1747 1DD2 A0 08          FLXDL1: LDY #8    ; 8 CHARS IN FILE NAME
1748 1DD4 BD 00 80      LDA DIRBUF,X    ; GET 1st CHAR OF NAME
1749 1DD7 FO 5C          BEQ FLXDEX      ; IF ZERO THEN END OF DIRECTORY
1750 1DD9 10 0C          BPL FLXLPO     ; IF POSITIVE THEN VALID ENTRY
1751 1ddb AO 18          LDY #24        ; IF NEGATIVE THEN DELETED ENTRY
1752 1DDD E8            FLXDSE: INX      ; ADVANCE OVER DELETED DIRECTORY
1753 1DDE 88            DEY
1754 1DDF DO FC          BNE FLXDSE
1755 1DE1 EO 00          CPX #0         ; CHECK IF AT END OF SECTOR
1756 1DE3 DO ED          BNE FLXDL1     ; IF NOT DO NEXT ENTRY (DON'T PRINT CRLF)
1757 1DE5 FO 3D          BEQ FLXDES     ; ELSE GO
1758
1759 1DE7 20 7B 1C      FLXLPO: JSR DIRCHR ; PRINT VALID CHAR
1760 1DEA 88            DEY
1761 1DEB DO FA          BNE FLXLPO     ; PRINT ALL 8 CHARS of FILE NAME
1762
1763 1DED 20 A3 1C      JSR DIRPER     ; PRINT '.'
1764 1DF0 A0 03          LDY #3         ; 3 CHARS IN EXTENSION
1765 1DF2 20 7B 1C      FLXLP2: JSR DIRCHR ; PRINT DIRECTORY CHAR
1766 1DF5 88            DEY
1767 1DF6 DO FA          BNE FLXLP2
1768 1DF8 20 40 2C      JSR SPACE2     ; 2 <space>
1769 1DFB 20 83 1C      JSR DIRBYT    ; PRINT PROTECTION ATTRIBUTES
1770 1DFE E8            INX      ; SKIP (RESERVED or SYSTEM EXPANSION)
1771 1DFF 20 36 1E      JSR FLXDSR     ; PRINT STARTING TRACK/SECTOR
1772 1E02 20 36 1E      JSR FLXDSR     ; PRINT ENDING TRACK/SECTOR
1773 1E05 20 36 1E      JSR FLXDSR     ; PRINT FILE SIZE
1774 1E08 20 40 2C      JSR SPACE2     ; 2 <space>
1775 1E0B E8            INX      ; SKIP OVER FILE SECTOR MAP INDICATOR
1776 1E0C E8            INX      ; SKIP (BYTE RESERVED FOR FUTURE SYSTEM USE)
1777 1E0D 20 9B 1C      JSR DIRDEC     ; PRINT MONTH (DECIMAL)
1778 1E10 A9 2F          LDA #'/'
1779 1E12 20 95 2B      JSR CTYOUT
1780 1E15 20 9B 1C      JSR DIRDEC     ; PRINT DAY (DECIMAL)
1781 1E18 A9 2F          LDA #'/'
1782 1E1A 20 95 2B      JSR CTYOUT
1783 1E1D 20 9B 1C      JSR DIRDEC     ; PRINT YEAR (DECIMAL)
1784 1E20 EO 00          CPX #0
    
```

1785	1E22	DO	AB		BNE	FLXDLP		; PRINT NEXT DIR ENTRY IN SECTOR
1786								; IF ZERO THEN END OF THIS SECTOR
1787	1E24				FLXDES:			; FLEX DIR END OF SECTOR
1788	1E24	AD	00	80	LDA	DIRBUF+0		; GET LINKAGE BYTES
1789	1E27	8D	15	36	STA	INPTRK		; TRACK
1790	1E2A	AD	01	80	LDA	DIRBUF+1		
1791	1E2D	8D	16	36	STA	INPSC		; SECTOR
1792	1E30	OD	15	36	ORA	INPTRK		; IF BOTH ZERO THEN QUIT
1793	1E33	DO	95		BNE	FLXDNS		; FLEX DIR NEXT SECOTR
1794	1E35	60			FLXDEX:	RTS		; FLEX DIR EXIT
1795								
1796	1E36	20	09	10	FLXDSE:	JSR	CTYSPA	; <space>
1797	1E39	20	83	1C	JSR	DIRBYT		; PRINT STARTING TRACK
1798	1E3C	20	83	1C	JSR	DIRBYT		; PRINT STARTING SECTOR
1799	1E3F	60			RTS			
1800								

```

1802
1803 1E40 38          FLXONF: SEC          ; SET FILE NOT FOUND FLAG
1804 1E41 60          RTS
1805
1806 1E42 A9 00      FLXOPN: LDA #0          ; FLEX OPEN:
1807 1E44 8D 4F 36   STA FLXTFL          ; ZERO TAB FLAG & COUNTER
1808 1E47 8D 50 36   STA FLXTCT          ; TAB FLAG
1809 1E4A A9 00      LDA #0          ; TAB COUNTER
1810 1E4C 8D 00 80   STA DIRBUF+0        ; TRACK 0
1811 1E4F A9 05      LDA #5          ; SECTOR 5
1812 1E51 8D 01 80   STA DIRBUF+1
1813 1E54 20 B3 19   JSR GTATOM          ; GET THE FILE NAME
1814
1815 1E57 AD 00 80   FLXONS: LDA DIRBUF+0 ; GET NEXT TRACK #
1816 1E5A 8D 35 36   STA TMPTRK
1817 1E5D AD 01 80   LDA DIRBUF+1        ; GET NEXT SECTOR #
1818 1E60 8D 36 36   STA TMPSCT
1819 1E63 8D 38 36   STA TMLSCT          ; SAVE AS LAST SECTOR ALSO
1820 1E66 OD 35 36   ORA TMPTRK          ; CHECK IF ZERO
1821 1E69 FO D5      BEQ FLXONF          ; FLEX Open Not Found
1822
1823 1E6B 20 ED 2C   JSR RDTEMP
1824 1E6E A2 10      LDX #16             ; SKIP FIRST 16 BYTES OF SECTOR
1825 1E70 AO 00      FLXONE: LDY #0
1826 1E72 B9 00 38   LDA ATMBUF,Y        ; 1st CHAR
1827 1E75 DD 00 80   CMP DIRBUF,X
1828 1E78 DO 73      BNE FLXON8
1829 1E7A E8        INX
1830 1E7B C8        INY
1831
1832 1E7C B9 00 38   LDA ATMBUF,Y        ; 2nd CHAR
1833 1E7F C9 2E      CMP #'.'            ; END OF FILE NAME?
1834 1E81          JEQ FLXOX7          ; THEN DO EXTENSION
1835 (1) 1E81 DO 03   BNE .+5
1836 (1) 1E83 4C 07 1F JMP FLXOX7
1837 1E86 DD 00 80   CMP DIRBUF,X
1838 1E89 DO 63      BNE FLXON7
1839 1E8B E8        INX
1840 1E8C C8        INY
1841 1E8D B9 00 38   LDA ATMBUF,Y        ; 3rd CHAR
1842 1E90 C9 2E      CMP #'.'            ; END OF FILE NAME?
1843 1E92 FO 74      BEQ FLXOX6          ; THEN DO EXTENSION
1844 1E94 DD 00 80   CMP DIRBUF,X
1845 1E97 DO 56      BNE FLXON6
1846 1E99 E8        INX
1847 1E9A C8        INY
1848
1847 1E9B B9 00 38   LDA ATMBUF,Y        ; 4th CHAR
1848 1E9E C9 2E      CMP #'.'            ; END OF FILE NAME?
1849 1EA0 FO 67      BEQ FLXOX5          ; THEN DO EXTENSION
1850 1EA2 DD 00 80   CMP DIRBUF,X
1851 1EA5 DO 49      BNE FLXON5
1852 1EA7 E8        INX
1853 1EA8 C8        INY
1854 1EA9 B9 00 38   LDA ATMBUF,Y        ; 5th CHAR
    
```

```

1855 1EAC C9 2E          CMP      #'          ; END OF FILE NAME?
1856 1EAE FO 5A          BEQ      FLXOX4      ; THEN DO EXTENSION
1857 1EB0 DD 00 80      CMP      DIRBUF,X
1858 1EB3 DO 3C          BNE      FLXON4
1859 1EB5 E8            INX
1860 1EB6 C8            INY
1861 1EB7 B9 00 38      LDA      ATMBUF,Y    ; 6th CHAR
1862 1EBA C9 2E          CMP      #'          ; END OF FILE NAME?
1863 1EBC FO 4D          BEQ      FLXOX3      ; THEN DO EXTENSION
1864 1EBE DD 00 80      CMP      DIRBUF,X
1865 1EC1 DO 2F          BNE      FLXON3
1866 1EC3 E8            INX
1867 1EC4 C8            INY
1868 1EC5 B9 00 38      LDA      ATMBUF,Y    ; 7th CHAR
1869 1EC8 C9 2E          CMP      #'          ; END OF FILE NAME?
1870 1ECA FO 40          BEQ      FLXOX2      ; THEN DO EXTENSION
1871 1ECC DD 00 80      CMP      DIRBUF,X
1872 1ECF DO 22          BNE      FLXON2
1873 1ED1 E8            INX
1874 1ED2 C8            INY
1875 1ED3 B9 00 38      LDA      ATMBUF,Y    ; 8th CHAR
1876 1ED6 C9 2E          CMP      #'          ; END OF FILE NAME?
1877 1ED8 FO 33          BEQ      FLXOX1      ; THEN DO EXTENSION
1878 1EDA DD 00 80      CMP      DIRBUF,X
1879 1EDD DO 15          BNE      FLXON1
1880 1EDF E8            INX
1881 1EE0 C8            INY
1882 1EE1 B9 00 38      LDA      ATMBUF,Y    ; HAD BETTER BE A '.'
1883 1EE4 C9 2E          CMP      #'          ;
1884 1EE6 FO 26          BEQ      FLXOXO
1885 1EE8 A9 13          LDA      #INVFNM     ; INVALID FILE NAME
1886 1EEA 20 82 2F      JSR      ERROR
1887
1888 1EED E8            FLXON8: INX          ; 1
1889 1EEE E8            FLXON7: INX          ; 2
1890 1EEF E8            FLXON6: INX          ; 3
1891 1EFO E8            FLXON5: INX          ; 4
1892 1EF1 E8            FLXON4: INX          ; 5
1893 1EF2 E8            FLXON3: INX          ; 6
1894 1EF3 E8            FLXON2: INX          ; 7
1895 1EF4 E8            FLXON1: INX          ; 8
1896 1EF5 E8            FLXOE3: INX          ; 9
1897 1EF6 E8            FLXOE2: INX          ; 10
1898 1EF7 E8            FLXOE1: INX          ; 11
1899 1EF8 AO 0C          LDY      #24-11-1
1900 1EFA E8            FLXONL: INX
1901 1EFB 88            DEY
1902 1EFC DO FC          BNE      FLXONL
1903 1EFE E8            INX          ; DO LAST ONE FOR TESTING
1904
1905 1EFF          JNE      FLXONE
(1) 1EFF FO 03          BEQ      .+5
(1) 1FO1 4C 70 1E      JMP      FLXONE
1906 1FO4 4C 57 1E      JMP      FLXONS
1907
    
```



```

1963 1F67 68          PLA          ; RESTORE FIRST SECTOR
1964 1F68 8D 26 36    STA          OUTSCT
1965 1F6B 68          PLA          ; RESTORE FIRST TRACK
1966 1F6C 8D 25 36    STA          OUTTRK
1967 1F6F A9 84          LDA          #OUTBUF^  ; SET BUFFER ADDRESS
1968 1F71 8D 3B 36    STA          TMPAGE
1969 1F74 20 ED 2C    JSR          RDTEMP  ; READ THE 1st SECTOR
1970 1F77 18          CLC          ; SET OPENED FLAG
1971 1F78 60          RTS
1972
1973 1F79 68          FLXOIN: PLA      ; RESTORE POINTER
1974 1F7A 8D 14 36    STA          INPPTR
1975 1F7D 68          PLA          ; RESTORE FIRST SECTOR
1976 1F7E 8D 16 36    STA          INPSCT
1977 1F81 68          PLA          ; RESTORE FIRST TRACK
1978 1F82 8D 15 36    STA          INPTRK
1979 1F85 A9 82          LDA          #INPBUF^  ; SET BUFFER ADDRESS
1980 1F87 8D 1B 36    STA          INPAGE
1981 1F8A 20 AF 2C    JSR          RDSCTR  ; READ THE SECTOR
1982 1F8D 18          CLC          ; SET OPENED FLAG
1983 1F8E 60          RTS
1984
1985
1986 1F8F AD 4F 36    FLXINP: LDA      ; CHECK IF TAB
1987 1F92 FO OE          BEQ          FLXINT  ; BRANCH IF NOT
1988 1F94 CE 50 36    DEC          FLXTCT  ; DEC TAB COUNTER IF SO
1989 1F97 DO 05          BNE          FLXIMR
1990 1F99 A9 00          LDA          #0      ; ZERO TAB FLAG IF LAST
1991 1F9B 8D 4F 36    STA          FLXTFL
1992 1F9E A9 20          FLXIMR: LDA      ; <space> FLX INPUT MORE (SPACES)
1993 1FA0 DO 2C          BNE          FLXICC  ; WILL ALWAYS BRANCH
1994
1995          ; FLX INPUT NO TAB
1996 1FA2
1997 1FA2 20 C3 1F    FLXINT: JSR          FLXINB ; FLEX NEXT BYTE
1998 1FA5 BO 47          BCS          FLXIEX
1999 1FA7 C9 00          CMP          #$00
2000 1FA9 FO F7          BEQ          FLXINT  ; IGNORE ALL NULLS!
2001 1FAB C9 09          CMP          #'I&$3F ; TAB?
2002 1FAD DO 1F          BNE          FLXICC  ; RETURN IF NOT
2003 1FAF A9 FF          LDA          #$FF
2004 1FB1 8D 4F 36    STA          FLXTFL  ; MAKE TAB FLAG NON ZERO
2005 1FB4 20 C3 1F    JSR          FLXINB ; GET NEXT BYTE
2006 1FB7 BO 35          BCS          FLXIEX
2007 1FB9 8D 50 36    STA          FLXTCT
2008 1FBC CE 50 36    DEC          FLXTCT ; DEC FOR THIS SPACE
2009 1FBF A9 20          LDA          #$20  ; <space>
2010 1FC1 DO 0B          BNE          FLXICC  ; RETURN (WILL ALWAYS BRANCH)
2011
2012
2013 1FC3          FLXINB:          ; FLEX INPUT NEXT BYTE
2014 1FC6 EE 14 36    INC          INPPTR ; BUMP UP POINTER
2015 1FC6 FO 08          BEQ          FLXINS  ; NEXT SECTOR IF ZERO
2016 1FC8 AE 14 36    LDX          INPPTR
2017 1FCB BD 00 82    LDA          INPBUF,X
    
```

```

2018
2019 1FCE 18          FLXICC: CLC
2020 1FCF 60          RTS
2021
2022 1FDO AD 00 82    FLXINS: LDA INPBUF+0      ; GET NEXT TRACK #
2023 1FD3 8D 15 36          STA INPTRK
2024 1FD6 AD 01 82          LDA INPBUF+1      ; GET NEXT SECTOR #
2025 1FD9 8D 16 36          STA INPSCT
2026 1FDC OD 15 36          ORA INPTRK      ; IF BOTH ZERO THEN END OF FILE
2027 1FDF FO OD          BEQ FLXIEX
2028 1FE1 20 AF 2C          JSR RDSCTR      ; GET THE NEXT SECTOR
2029 1FE4 A2 04          LDX #4
2030 1FE6 8E 14 36          STX INPPTR      ; RESET INPUT POINTER BACK TO BEGINNING
2031 1FE9 BD 00 82          LDA INPBUF,X
2032 1FEC 18          CLC
2033 1FED 60          RTS
2034
2035 1FEE 38          FLXIEX: SEC      ; SET END OF FILE
2036 1FEF 60          RTS
2037
2038 1FF0 EE 24 36    FLXOUT: INC OUTPTR
2039 1FF3 FO 08          BEQ FLXONS
2040 1FF5 AE 24 36          LDX OUTPTR
2041 1FF8 9D 00 84          STA OUTBUF,X
2042 1FFB 18          CLC
2043 1FFC 60          RTS
2044
2045 1FFD 48          FLXONS: PHA      ; SAVE CHAR
2046 1FFE 20 6F 2D          JSR WRSCTR      ; WRITE CURRENT SECTOR
2047 2001 AD 00 84          LDA OUTBUF+0    ; GET NEXT TRACK #
2048 2004 8D 25 36          STA OUTTRK
2049 2007 8D 35 36          STA TMPTRK
2050 200A AD 01 84          LDA OUTBUF+1    ; GET NEXT SECTOR #
2051 200D 8D 26 36          STA OUTSCT
2052 2010 8D 36 36          STA TMPSCT
2053 2013 8D 38 36          STA TMLSCT
2054
2055 2016 A9 84          LDA #OUTBUF^    ; SET BUFFER ADDRESS
2056 2018 8D 3B 36          STA TMPAGE
2057 201B 20 ED 2C          JSR RDTEMP
2058 201E A2 04          LDX #4
2059 2020 8E 24 36          STX OUTPTR      ; SKIP PAST LINKAGE INFO
2060 2023 68          PLA
2061 2024 9D 00 84          STA OUTBUF,X    ; RESTORE CHAR
2062 2027 18          CLC
2063 2028 60          RTS
2064
2065 2029 AE 24 36    FLXCLO: LDX OUTPTR      ; FLEX CLOSE FILE ROUTINE
2066 202C FO 08          BEQ FLXCL2
2067 202E A9 00          LDA #0
2068 2030 9D 00 84    FLXCLP: STA OUTBUF,X    ; FILL THE REST OF THE FILE WITH $00 (NULS)
2069 2033 E8          INX
2070 2034 DO FA          BNE FLXCLP      ; DO THE REST OF THIS SECTOR
2071
2072 2036 AD 00 84    FLXCL2: LDA OUTBUF+0
    
```

```
2073 2039 OD 01 84 ORA OUTBUF+1 ; IF BOTH ZERO THEN AT THE END
2074 203C FO 27 BEQ FLXCEN
2075 203E 20 6F 2D JSR WRSCTR ; ELSE WRITE IT & GET THE NEXT ONE
2076 2041 AD 00 84 LDA OUTBUF+0
2077 2044 8D 35 36 STA TMPTRK
2078 2047 8D 25 36 STA OUTTRK
2079 204A AD 01 84 LDA OUTBUF+1
2080 204D 8D 26 36 STA OUTSCT
2081 2050 8D 36 36 STA TMPSTC
2082 2053 8D 38 36 STA TMLSTC
2083
2084 2056 A9 84 LDA #OUTBUF^ ; SET BUFFER ADDRESS
2085 2058 8D 3B 36 STA TMPAGE
2086 205B 20 ED 2C JSR RDTEMP ; READ INTO OUTPUT BUFFER
2087 205E A2 03 LDX #4-1
2088 2060 8D 24 36 STA OUTPTR
2089 2063 D0 C4 BNE FLXCLO ; CONTINUE CLOSING (WILL ALWAYS BRANCH)
2090
2091 2065 20 6F 2D FLXCEN: JSR WRSCTR ; FLEX CLOSE END
2092 2068 18 CLC ; WRITE THE LAST SECTOR
2093 2069 60 RTS
2094
```

```

2096          .SBTTL CP/M   Disk Drivers
2097
2098          ; CP/M Control Program / Microcomputer
2099
2100          ; CP/M STANDARD CONFIGUREATION:
2101          ; 128 BYTES/SECTOR
2102          ; 26 SECTORS/TRACK
2103          ; SINGLE DENSITY SINGLE SIDED
2104          ; CP/M (1.3) STAGGERING OF SECTORS
2105
2106          ; SYSTEM OVERHEAD:
2107          ; TRACK 0 DISK BOOT
2108          ; TRACK 1 CP/M
2109          ; TRACK 2 DIRECTORY
2110
2111          ;      OO      1      $OO = ACTIVE      $E5 = DELETED
2112          ;      O1-08   8      FILENAME (PADDED WITH SPACES)
2113          ;      O9-0B   3      FILE TYPE (EXTENSION)
2114          ;      OC      1      WHICH EXTENT (O FOR INITIAL ENTRY, 1 FOR REST)
2115          ;      OD      1      ? NORMALLY $OO
2116          ;      OE      1      ? NORMALLY $OO
2117          ;      OF      1      LENGTH
2118          ;      10-1F  16      BLOCK MAP
2119
2120          O100          CPMDBS =      $O100          ; SINGLE DENSITY, 128 BYTES/SECTOR
2121          O11A          CPMSCCT =     $O11A          ; SECTOR INFO
2122
2123          206A          CPMDEF:
2124          206A      18          CLC
2125          206B      60          RTS
2126
2127          206C      A9      02          CPMDIR: LDA      #2          ; TRACK 2
2128          206E      8D      15      36          STA      INPTRK
2129          2071      A9      01          LDA      #1          ; SECTOR 1
2130          2073      8D      65      36          STA      CPMEXT          ; USED FOR SOMETHING DIFERENT THAN NORMAL
2131
2132          2076      AD      65      36          CPMLPO: LDA      CPMEXT          ; GET CURRENT LOGICAL SECTOR #
2133          2079      18          CLC          ; BUMP DOWN BY ONE
2134          207A      69      FF          ADC      #-1
2135          207C      20      CO      24          JSR      CPMLTP          ; CONVERT TO PHYSICAL
2136          207F      8D      16      36          STA      INPSCT
2137          2082      20      AF      2C          JSR      RDSCTR
2138          2085      A2      00          LDX      #0          ; INIT INDEX POINTER
2139          2087          CPMLP1:
2140          2087      BD      00      80          LDA      DIRBUF,X          ; CHECK IF FILES ACTIVE (OO = ACTIVE)
2141          208A      FO      09          BEQ      CPMDOK          ; BRANCH IF SO
2142          208C      8A          TXA          ; FALL THROUGH IF NOT
2143          208D      18          CLC          ; AND SKIP THIS ENTRY
2144          208E      69      20          ADC      #32          ; 32 BYTES PER DIRECTORY ENTRY
2145          2090      AA          TAX          ; RESTORE BACK TO X
2146          2091      10      F4          BPL      CPMLP1          ; IF LESS THAN 128 CONTINUE
2147          2093      30      3D          BMI      CPMDNS          ; OTHERWISE GO GET NEXT SECTOR
2148
2149          2095      20      FO      2B          CPMDOK: JSR      CRLFSQ          ; NEW LINE (^S/^Q & ^C)
2150          2098      E8          INX          ; BUMP PAST ACTIVE FLAG
    
```

```

2151 2099 AO 08          LDY      #8          ; 8 CHARS IN NAME
2152 209B 20 7B 1C      CPMLP2: JSR     DIRCHR       ; PRINT VALID CHAR
2153 209E 88           DEY
2154 209F DO FA          BNE     CPMLP2
2155 20A1 20 A3 1C      JSR     DIRPER       ; PRINT '.'
2156 20A4 AO 03          LDY     #3          ; 3 CHARS IN FILE EXTENSION
2157 20A6 20 7B 1C      CPMLP3: JSR     DIRCHR       ; PRINT DIR CHAR
2158 20A9 88           DEY
2159 20AA DO FA          BNE     CPMLP3
2160 20AC 20 09 10      JSR     CTYSPA
2161 20AF BD 00 80      LDA     DIRBUF,X
2162 20B2 09 30          ORA     #$30        ; TURN INTO ASCII #
2163 20B4 20 95 2B      JSR     CTYOUT
2164 20B7 20 09 10      JSR     CTYSPA
2165 20BA E8           INX
2166 20BB E8           INX
2167 20BC E8           INX
2168 20BD 20 83 1C      JSR     DIRBYT       ; PRINT FILE LENGTH
2169 20C0 20 40 2C      JSR     SPACE2
2170
2171 20C3 AO 10          LDY     #16         ; 16 BYTES IN BAT
2172 20C5 20 83 1C      CPMLP4: JSR     DIRBYT       ; PRINT BAT ENTRY
2173 20C8 20 09 10      JSR     CTYSPA       ; <space>
2174 20CB 88           DEY
2175 20CC DO F7          BNE     CPMLP4
2176 20CE EO 80          CPX     #128
2177 20D0 90 B5          BLT     CPMLP1       ; DOES THE SAME AS BPL
2178
2179 20D2 EE 65 36      CPMDNS: INC     CPMEXT       ; ADVANCE TO NEXT SECTOR
2180 20D5 AD 65 36      LDA     CPMEXT
2181 20D8 C9 1B          CMP     #27
2182 20DA 90 9A          BLT     CPMLPO       ; CONTINUE UNTILL ALL DIR SECTORS DONE
2183 20DC 60           RTS
2184

```

```

2186
2187          0009          CPMTSP =          9          ; # OF SPACES IN A TAB (SORT-OF)
2188
2189 20DD 20 B3 19          CPMOPN: JSR          GTATOM          ; GET THE FILE NAME
2190 20EO A0 OF          LDY          #16-1
2191 20E2 B9 00 38          CPMOPL: LDA          ATMBUF,Y
2192 20E5 99 7D 36          STA          CPMBUF,Y
2193 20E8 88          DEY
2194 20E9 10 F7          BPL          CPMOPL
2195 20EB A9 00          LDA          #0          ; FIRST EXTENT
2196 20ED 8D 65 36          STA          CPMEXT
2197 20FO A9 09          LDA          #CPMTSP          ; INIT CP/M TAB COUNTER
2198 20F2 8D 67 36          STA          CPMTCT
2199 20F5 A9 00          LDA          #0          ; CLEAR TAB FLAG
2200 20F7 8D 66 36          STA          CPMTFL
2201 20FA 20 00 21          JSR          CPMOPF          ; CP/M OPEN FILE
2202 20FD 60          RTS
2203
2204
2205 20FE 38          CPMONF: SEC          ; SET FILE NOT FOUND FLAG
2206 20FF 60          RTS
2207
2208 2100 A9 02          CPMOPF: LDA          #2          ; SET DIR TRACK TO 2
2209 2102 8D 35 36          STA          TMPTRK
2210 2105 A9 00          LDA          #1-1          ; FIRST SECTOR # (GETS BUMPED UP)
2211 2107 8D 36 36          STA          TMPSCT
2212 210A 8D 38 36          STA          TMLSCT
2213
2214 210D EE 36 36          CPMONS: INC          TMPSCT          ; BUMP UP SECTOR NUMBER
2215 2110 EE 38 36          INC          TMLSCT          ; BUMP UP THE LAST SECTOR TOO
2216 2113 AD 36 36          LDA          TMPSCT
2217 2116 C9 1B          CMP          #27          ; CHECK IF AT END OF DIRECTORY TRACK
2218 2118 B0 E4          BGE          CPMONF          ; FILE NOT FOUND
2219 211A 20 ED 2C          JSR          RDTEMP
2220
2221 211D A2 00          LDX          #0          ; X DIRECTORY BUFF POINTER
2222 211F A0 00          LDY          #0          ; Y ATOM BUFF POINTER
2223 2121 BD 00 80          CPMONE: LDA          DIRBUF,X          ; CHECK ACTIVE FLAG
2224 2124          JNE          CPMOSE          ; SKIP ENTRY IF DELETED
(1) 2124 FO 03          BEQ          .+5
(1) 2126 4C 42 22          JMP          CPMOSE
2225 2129 E8          INX          ; POINT TO 1st CHAR OF NAME IN DIR FIELD
2226 212A BD 00 80          LDA          DIRBUF,X          ; 1st CHAR
2227 212D D9 7D 36          CMP          CPMBUF,Y
2228 2130          JNE          CPMON8
(1) 2130 FO 03          BEQ          .+5
(1) 2132 4C 43 22          JMP          CPMON8
2229 2135 E8          INX          ; BUMP INDEX REGISTERS
2230 2136 C8          INY
2231 2137 BD 00 80          LDA          DIRBUF,X
2232 213A C9 20          CMP          #$20          ; IF SPACE THEN FOUND 1ST PART
2233 213C DO OA          BNE          CPMNX7
2234 213E B9 7D 36          LDA          CPMBUF,Y
2235 2141 C9 2E          CMP          #'.'          ; MUST BE A PERIOD FOR MATCH
2236 2143          JEQ          CPMOX7
    
```

```

(1) 2143 DO 03 BNE .+5
(1) 2145 4C 03 22 JMP CPMOX7
2237
2238 2148 BD 00 80 CPMNX7: LDA DIRBUF,X ; 2nd CHAR
2239 214B D9 7D 36 CMP CPMBUF,Y
2240 214E JNE CPMON7
(1) 214E FO 03 BEQ .+5
(1) 2150 4C 44 22 JMP CPMON7
2241
2242 2153 E8 INX ; BUMP INDEX REGISTERS
2243 2154 C8 INY
2244 2155 BD 00 80 LDA DIRBUF,X
2245 2158 C9 20 CMP #$20 ; IF SPACE THEN FOUND 1ST PART
2246 215A DO 0A BNE CPMNX6
2247 215C B9 7D 36 LDA CPMBUF,Y
2248 215F C9 2E CMP #'
2249 2161 JEQ CPMOX6
(1) 2161 DO 03 BNE .+5
(1) 2163 4C 04 22 JMP CPMOX6
2250
2251 2166 BD 00 80 CPMNX6: LDA DIRBUF,X ; 3rd CHAR
2252 2169 D9 7D 36 CMP CPMBUF,Y
2253 216C JNE CPMON6
(1) 216C FO 03 BEQ .+5
(1) 216E 4C 45 22 JMP CPMON6
2254
2255 2171 E8 INX ; BUMP INDEX REGISTERS
2256 2172 C8 INY
2257 2173 BD 00 80 LDA DIRBUF,X
2258 2176 C9 20 CMP #$20 ; IF SPACE THEN FOUND 1ST PART
2259 2178 DO 0A BNE CPMNX5
2260 217A B9 7D 36 LDA CPMBUF,Y
2261 217D C9 2E CMP #'
2262 217F JEQ CPMOX5
(1) 217F DO 03 BNE .+5
(1) 2181 4C 05 22 JMP CPMOX5
2263
2264 2184 BD 00 80 CPMNX5: LDA DIRBUF,X ; 4th CHAR
2265 2187 D9 7D 36 CMP CPMBUF,Y
2266 218A JNE CPMON5
(1) 218A FO 03 BEQ .+5
(1) 218C 4C 46 22 JMP CPMON5
2267
2268 218F E8 INX ; BUMP INDEX REGISTERS
2269 2190 C8 INY
2270 2191 BD 00 80 LDA DIRBUF,X
2271 2194 C9 20 CMP #$20 ; IF SPACE THEN FOUND 1ST PART
2272 2196 DO 07 BNE CPMNX4
2273 2198 B9 7D 36 LDA CPMBUF,Y
2274 219B C9 2E CMP #'
2275 219D FO 67 BEQ CPMOX4
2276
2277 219F BD 00 80 CPMNX4: LDA DIRBUF,X ; 5th CHAR
2278 21A2 D9 7D 36 CMP CPMBUF,Y
2279 21A5 JNE CPMON4
    
```

```

(1) 21A5 FO 03 BEQ .+5
(1) 21A7 4C 47 22 JMP CPMON4
2280
2281 21AA E8 INX ; BUMP INDEX REGISTERS
2282 21AB C8 INY
2283 21AC BD 00 80 LDA DIRBUF,X
2284 21AF C9 20 CMP #$20 ; IF SPACE THEN FOUND 1ST PART
2285 21B1 DO 07 BNE CPMNX3
2286 21B3 B9 7D 36 LDA CPMBUF,Y
2287 21B6 C9 2E CMP #'
2288 21B8 FO 4D BEQ CPMOX3
2289
2290 21BA BD 00 80 CPMNX3: LDA DIRBUF,X ; 6th CHAR
2291 21BD D9 7D 36 CMP CPMBUF,Y
2292 21C0 JNE CPMON3
(1) 21C0 FO 03 BEQ .+5
(1) 21C2 4C 48 22 JMP CPMON3
2293
2294 21C5 E8 INX ; BUMP INDEX REGISTERS
2295 21C6 C8 INY
2296 21C7 BD 00 80 LDA DIRBUF,X
2297 21CA C9 20 CMP #$20 ; IF SPACE THEN FOUND 1ST PART
2298 21CC DO 07 BNE CPMNX2
2299 21CE B9 7D 36 LDA CPMBUF,Y
2300 21D1 C9 2E CMP #'
2301 21D3 FO 33 BEQ CPMOX2
2302
2303 21D5 BD 00 80 CPMNX2: LDA DIRBUF,X ; 7th CHAR
2304 21D8 D9 7D 36 CMP CPMBUF,Y
2305 21DB DO 6C BNE CPMON2
2306
2307 21DD E8 INX ; BUMP INDEX REGISTERS
2308 21DE C8 INY
2309 21DF BD 00 80 LDA DIRBUF,X
2310 21E2 C9 20 CMP #$20 ; IF SPACE THEN FOUND 1ST PART
2311 21E4 DO 07 BNE CPMNX1
2312 21E6 B9 7D 36 LDA CPMBUF,Y
2313 21E9 C9 2E CMP #'
2314 21EB FO 1C BEQ CPMOX1
2315
2316 21ED BD 00 80 CPMNX1: LDA DIRBUF,X ; 8th CHAR
2317 21FO D9 7D 36 CMP CPMBUF,Y
2318 21F3 DO 55 BNE CPMON1
2319
2320 21F5 E8 INX ; BUMP INDEX REGISTERS
2321 21F6 C8 INY
2322 21F7 B9 7D 36 LDA CPMBUF,Y
2323 21FA C9 2E CMP #'
2324 21FC FO 0C BEQ CPMOX0
2325
2326 21FE A9 13 LDA #INVFNM ; INVALID FILE NAME ERROR
2327 2200 20 82 2F JSR ERROR
2328
2329 2203 E8 CPMOX7: INX ; BUMP OVER THE REST OF THE
2330 2204 E8 CPMOX6: INX ; SPACE FOR THE FILE NAME
    
```



```

2331 2205 E8          CPMOX5: INX
2332 2206 E8          CPMOX4: INX
2333 2207 E8          CPMOX3: INX
2334 2208 E8          CPMOX2: INX
2335 2209 E8          CPMOX1: INX
2336 220A          CPMOX0:
2337 220A C8          INY          ; SKIP OVER PERIOD IN THE ATOM BUFFER
2338 220B BD 00 80    LDA DIRBUF,X      ; 1st CHAR OF EXTENSION
2339 220E D9 7D 36    CMP CPMBUF,Y
2340 2211 DO 38          BNE CPMOE3
2341 2213 E8          INX
2342 2214 C8          INY
2343 2215 BD 00 80    LDA DIRBUF,X      ; 2nd CHAR OF EXTENSION
2344 2218 D9 7D 36    CMP CPMBUF,Y
2345 221B DO 2F          BNE CPMOE2
2346 221D E8          INX
2347 221E C8          INY
2348 221F BD 00 80    LDA DIRBUF,X      ; 3rd CHAR OF EXTENSION
2349 2222 D9 7D 36    CMP CPMBUF,Y
2350 2225 DO 26          BNE CPMOE1
2351
2352 2227 E8          INX
2353 2228 BD 00 80    LDA DIRBUF,X      ; GET THE EXTENT NUMBER
2354 222B CD 65 36    CMP CPMEXT        ; CHECK EXTENT #
2355 222E DO 1E          BNE CPMOEE
2356
2357 2230 48          PHA          ; FILE FOUND IF GOT TO HERE
2358 2231 E8          INX
2359 2232 E8          INX
2360 2233 E8          INX
2361 2234 BD 00 80    LDA DIRBUF,X      ; LENGTH BYTE (WHAT IS IT USED FOR?)
2362 2237 48          PHA          ; GET LENGTH (# OF SECTORS)      ??? 13-SEP-81
2363 2238 E8          INX          ; SAVE (IS NUMBER OF SECTORS USED)
2364 2239 AO 00          LDY #0          ; NOW POINTING TO BLOCK MAP
2365 223B AD 3C 36    LDA TMPIOF        ; INIT INDEX POINTER (GETS USED BELOW)
2366 223E FO 1D          BEQ CPMOIN        ; CHECK if INPUT or OUTPUT
2367 2240 DO 62          BNE CPMOOT        ; IF ZERO THEN INPUT
2368          ; ELSE OUTPUT
2369          ; WILL ALWAYS BRANCH
2370 2242 E8          CPMOSE: INX      ; CP/M OPEN SKIP ENTRY
2371
2372 2243 E8          CPMON8: INX      ; 1
2373 2244 E8          CPMON7: INX      ; 2
2374 2245 E8          CPMON6: INX      ; 3
2375 2246 E8          CPMON5: INX      ; 4
2376 2247 E8          CPMON4: INX      ; 5
2377 2248 E8          CPMON3: INX      ; 6
2378 2249 E8          CPMON2: INX      ; 7
2379 224A E8          CPMON1: INX      ; 8
2380 224B E8          CPMOE3: INX      ; 9
2381 224C E8          CPMOE2: INX      ; 10
2382 224D E8          CPMOE1: INX      ; 11
2383 224E AO 13          CPMOEE: LDY      #32-12-1
2384 2250 E8          CPMONL: INX
2385 2251 88          DEY
    
```

2386	2252	DO	FC	BNE	CPMONL	
2387	2254	E8		INX		; DO LAST ONE FOR TESTING
2388						
2389	2255			JNE	CPMONE	; DO NEXT ENTRY IN THIS SECTOR
(1)	2255	FO	O3	BEQ	.+5	
(1)	2257	4C	1F	JMP	CPMONE	
2390	225A	4C	OD	JMP	CPMONS	; DO NEXT SECTOR
2391						
2392						

```

2394
2395 225D 68          CPMOIN: PLA
2396 225E 8D 54 36    STA CPMNSI      ; CP/M # OF SECTORS
2397 2261 68          PLA
2398 2262 8D 52 36    STA CPMEXI      ; SAVE EXTENT FLAG
2399 2265 BD 00 80    CPMOIL: LDA DIRBUF,X
2400 2268 99 55 36    STA CPMBTI,Y   ; CP/M BLOCK TABLE (INPUT)
2401 226B E8          INX
2402 226C C8          INY
2403 226D CO 10      CPY #16         ; 16 ENTRIES IN BLOCK TABLE
2404 226F 90 F4      BLT CPMOIL
2405 2271 A9 00      LDA #0
2406 2273 8D 53 36    STA CPMBPI     ; BLOCK TABLE POINTER (INPUT)
2407 2276 A9 00      LDA #0         ; NR (RECORD INTO BLOCK)
2408 2278 8D 51 36    STA CPMIRC
2409 227B AE 51 36    LDX CPMIRC
2410 227E AD 55 36    LDA CPMBTI+0   ; GET FIRST BLOCK NUMBER
2411 2281 20 63 24    JSR CPMCTS
2412 2284 8E 15 36    STX INPTRK
2413 2287 8C 16 36    STY INPSCT
2414 228A A9 82      LDA #INPBUF^   ; SET BUFFER ADDRESS
2415 228C 8D 1B 36    STA INPAGE
2416 228F A9 FF      LDA #-1
2417 2291 8D 14 36    STA INPPTR
2418 2294 20 AF 2C    JSR RDSCTR     ; READ THE SECTOR
2419
2420 2297 AO OF      LDY #16-1      ; SAVE NAME BUFFER
2421 2299 B9 7D 36    CPMOIM: LDA CPMBUF,Y ; MOVE LOOP
2422 229C 99 8D 36    STA CPMBFI,Y
2423 229F 88          DEY
2424 22A0 10 F7      BPL CPMOIM
2425
2426 22A2 18          CLC           ; SET OKAY FLAG
2427 22A3 60          RTS
2428

```

```

2430
2431 22A4 68          CPMOOL: PLA
2432 22A5 8D 6C 36          STA CPMNSO          ; CP/M # OF SECTORS
2433 22A8 68          PLA
2434 22A9 8D 6A 36          STA CPMEXO          ; SAVE EXTENT FLAG
2435 22AC BD 00 80          CPMOOL: LDA DIRBUF,X
2436 22AF 99 6D 36          STA CPMBTO,Y       ; CP/M BLOCK TABLE (OUTPUT)
2437 22B2 E8          INX
2438 22B3 C8          INY
2439 22B4 C0 10          CPY #16             ; 16 ENTRIES IN BLOCK TABLE
2440 22B6 90 F4          BLT CPMOOL
2441 22B8 A9 00          LDA #0
2442 22BA 8D 6B 36          STA CPMBPO         ; BLOCK TABLE POINTER (OUTPUT)
2443 22BD A9 00          LDA #0             ; NR (RECORD INTO BLOCK)
2444 22BF 8D 69 36          STA CPMORC
2445 22C2 AE 69 36          LDX CPMORC
2446 22C5 AD 6D 36          LDA CPMBTO+0      ; GET FIRST BLOCK NUMBER
2447 22C8 20 63 24          JSR CPMCTS
2448 22CB 8E 25 36          STX OUTTRK
2449 22CE 8C 26 36          STY OUTSCT
2450 22D1 A9 84          LDA #OUTBUF^      ; SET BUFFER ADDRESS
2451 22D3 8D 2B 36          STA OUPAGE
2452 22D6 A9 FF          LDA #-1
2453 22D8 8D 24 36          STA OUTPTR
2454
2455 22DB A0 0F          LDY #16-1         ; SAVE NAME BUFFER
2456 22DD B9 7D 36          CPMOOL: LDA CPMBUF,Y ; MOVE LOOP
2457 22E0 99 9D 36          STA CPMBFO,Y
2458 22E3 88          DEY
2459 22E4 10 F7          BPL CPMOOL
2460
2461 22E6 18          CLC             ; SET OKAY FLAG
2462 22E7 60          RTS
2463

```

```

2465
2466 22E8 AD 66 36 CPMINP: LDA CPMTFL ; CHECK IF TAB
2467 22EB FO 13 BEQ CPMINT ; GO GET NEXT CHAR IF NO TAB
2468 22ED CE 67 36 DEC CPMTCT ; KEEP TRACK POSITION
2469 22FO DO 0A BNE CPMITD
2470 22F2 A9 00 LDA #0
2471 22F4 8D 66 36 STA CPMTFL ; ZERO TAB FLAG
2472 22F7 A9 09 LDA #CPMTSP ; # OF SPACES IN TAB
2473 22F9 8D 67 36 STA CPMTCT
2474 22FC A9 20 CPMITD: LDA #$20 ; SPACE
2475 22FE 18 CLC
2476 22FF 60 RTS
2477
2478 2300 EE 14 36 CPMINT: INC INPPTR
2479 2303 30 04 BMI CPMINS
2480 2305 20 81 23 JSR CPMISR ; RTNS WITH EOF COND
2481 2308 60 RTS
2482
2483
2484 2309 CE 54 36 CPMINS: DEC CPMNSI ; CP/M # OF SECTORS
2485 230C 30 40 BMI CPMIEF ; END OF FILE
2486 230E EE 51 36 INC CPMIRC ; CP/M INPUT RECORD COUNTER (1-8)
2487 2311 AD 51 36 LDA CPMIRC
2488 2314 C9 08 CMP #8 ; 8 IF STARTED WITH 0, 9 IF 1
2489 2316 90 17 BLT CPMISB ; BRANCH IF STILL IN SAME BLOCK
2490 ; FALL THROUGH IF NEXT BLOCK
2491 2318 A9 00 LDA #0 ; MUST BE THE SAME AS ABOVE
2492 231A 8D 51 36 STA CPMIRC ; ZERO RECORD COUNTER
2493 231D EE 53 36 INC CPMBPI ; BLOCK TABLE POINTER
2494 2320 AE 53 36 LDX CPMBPI
2495 2323 EO 10 CPX #16 ; 16 BLOCKS/EXTENT
2496 2325 B0 29 BGE CPMINE ; GET NEXT EXTENT
2497 2327 BD 55 36 LDA CPMBTI,X
2498 232A FO 22 BEQ CPMIEF ; END IF ZERO
2499 232C AD 51 36 LDA CPMIRC
2500
2501 232F AA CPMISB: TAX ; PUT NR INTO X REG
2502 2330 AC 53 36 LDY CPMBPI ; BLOCK TABLE POINTER
2503 2333 B9 55 36 LDA CPMBTI,Y ; BLOCK NUMBER
2504 2336 20 63 24 JSR CPMCTS ; CALC TRACK/SECTOR
2505 2339 8E 15 36 STX INPTRK
2506 233C 8C 16 36 STY INPSCT
2507 233F 20 AF 2C JSR RDSCTR
2508 2342 A9 FF LDA #-1
2509 2344 8D 14 36 STA INPPTR ; ZERO OUT INPUT POINTER
2510 2347 EE 14 36 INC INPPTR ; POINT TO FIRST CHAR
2511 234A 20 81 23 JSR CPMISR ; GET CHAR AND EOF CONDITIONS
2512 234D 60 RTS
2513

```

```

2515
2516 234E 38          CPMIEF: SEC          ; PUT HERE SO WONT CAN STILL BRANCH TO
2517 234F 60          RTS
2518
2519
2520
2521 2350          CPMINE:          ; INPUT NEXT ENTRY,
2522 2350 EE 65 36    INC CPMEXT          ; BUMP UP & LOOK FOR NEXT EXTENT
2523 2353 A9 00      LDA #0
2524 2355 8D 3C 36    STA TMPIOF          ; SET INPUT FLAG
2525 2358 AD 13 36    LDA INPDRV          ; RE-SET DRIVE #
2526 235B 8D 33 36    STA TMPDRV
2527 235E AD 19 36    LDA INBYSC          ; RE-SET BYTES/SECTOR
2528 2361 8D 39 36    STA TMBYSC
2529 2364 AD 1A 36    LDA INPDEN          ; RE-SET DENSITY
2530 2367 8D 3A 36    STA TMPDEN
2531
2532 236A AO OF          LDY #16-1          ; MOVE FILENAME BACK INTO THE BUFFER
2533 236C B9 8D 36    CPMIML: LDA CPMBFI,Y
2534 236F 99 7D 36    STA CPMBUF,Y
2535 2372 88          DEY
2536 2373 10 F7      BPL CPMIML
2537
2538 2375 20 00 21    JSR CPMOPF          ; OPEN FILE
2539 2378 B0 D4          BCS CPMIEF          ; END OF FILE IF NO NEXT EXTENT
2540
2541 237A EE 14 36    INC INPPTR          ; ADVANCE PAST FIRST CHAR
2542 237D 20 81 23    JSR CPMISR          ; GET CHAR AND EOF CONDITION
2543 2380 60          RTS
2544

```

```

2546
2547 2381 AE 14 36 CPMISR: LDX INPPTR ; CP/M INPUT SUB-ROUTINE
2548 2384 BD 00 82 LDA INPBUF,X
2549 2387 C9 1A CMP #'Z&$3F ; ^Z ? (CPM EOF CHAR)
2550 2389 FO C3 BEQ CPMIEF
2551 238B C9 OD CMP #$OD ; CR ?
2552 238D DO OA BNE CPMIN1 ; BEGGING OF LINE
2553 238F 48 PHA ; SAVE THE CR
2554 2390 A9 09 LDA #CPMTSP ; # OF SPACES
2555 2392 8D 67 36 STA CPMTCT ; INIT TAB COUNTER
2556 2395 68 PLA ; RESTORE THE CR
2557 2396 4C C3 23 JMP CPMICC ; RTN OKAY
2558
2559 2399 C9 09 CPMIN1: CMP #'I&$3F ; TAB ?
2560 239B DO 1A BNE CPMIN2
2561 239D A9 FF LDA #$FF ; NON-ZERO
2562 239F 8D 66 36 STA CPMTFL ; TAB FLAG
2563 23A2 A9 20 LDA #$20 ; SPACE
2564 23A4 CE 67 36 DEC CPMTCT ; BUMP TAB COUNTER
2565 23A7 DO 1A BNE CPMICC
2566 23A9 A9 09 LDA #CPMTSP ; # OF SPACES
2567 23AB 8D 67 36 STA CPMTCT
2568 23AE A9 00 LDA #0 ; CLEAR TAB FLAG
2569 23B0 8D 66 36 STA CPMTFL
2570 23B3 A9 20 LDA #$20
2571 23B5 DO OC BNE CPMICC ; RTN OKAY
2572
2573 23B7 CE 67 36 CPMIN2: DEC CPMTCT
2574 23BA DO 07 BNE CPMICC
2575 23BC 48 PHA
2576 23BD A9 09 LDA #CPMTSP ; # OF SPACES
2577 23BF 8D 67 36 STA CPMTCT
2578 23C2 68 PLA
2579
2580 23C3 18 CPMICC: CLC
2581 23C4 60 RTS
2582
2583
2584
2585
    
```

```

2587
2588           ;.ERROR ;           THIS PART IS NOT FINISHED
2589
2590 23C5  EE  24  36  CPMOUT: INC  OUTPTR           ; CP/M OUTPUT
2591 23C8  30  04           BMI  CPMONS
2592 23CA  20  4D  24           JSR  CPMOSR           ; RTNS WITH EOF COND
2593 23CD  60           RTS
2594
2595
2596
2597           ; CP/M OUTPUT NEXT SECTOR
2598 23CE  8D  AD  36  CPMONS: STA  CPMOTC           ; SAVE CHAR
2599 23D1  20  6F  2D           JSR  WRSCTR           ; WRITE THE SECTOR..
2600           ; NOW SET UP FOR THE NEXT ONE
2601 23D4  CE  6C  36           DEC  CPMNSO           ; CP/M # OF SECTORS
2602 23D7  30  40           BMI  CPMOEF           ; END OF FILE
2603 23D9  EE  69  36           INC  CPMORC           ; CP/M OUTUT RECORD COUNTER (1-8)
2604 23DC  AD  69  36           LDA  CPMORC
2605 23DF  C9  08           CMP  #8           ; 8 IF STARTED WITH 0, 9 IF 1
2606 23E1  90  17           BLT  CPMOSB           ; BRANCH IF STILL IN SAME BLOCK
2607           ; FALL THROUGH IF NEXT BLOCK
2608 23E3  A9  00           LDA  #0           ; MUST BE THE SAME AS ABOVE
2609 23E5  8D  69  36           STA  CPMORC           ; ZERO RECORD COUNTER
2610 23E8  EE  6B  36           INC  CPMBPO           ; BLOCK TABLE POINTER
2611 23EB  AE  6B  36           LDX  CPMBPO
2612 23EE  EO  10           CPX  #16           ; 16 BLOCKS/EXTENT
2613 23FO  B0  29           BGE  CPMONE           ; GET NEXT EXTENT
2614 23F2  BD  6D  36           LDA  CPMBTO,X
2615 23F5  FO  22           BEQ  CPMOEF           ; END IF ZERO
2616 23F7  AD  69  36           LDA  CPMORC
2617
2618 23FA  AA           CPMOSB: TAX           ; PUT NR INTO X REG
2619 23FB  AC  6B  36           LDY  CPMBPO           ; BLOCK TABLE POINTER
2620 23FE  B9  6D  36           LDA  CPMBTO,Y           ; BLOCK NUMBER
2621 2401  20  63  24           JSR  CPMCTS           ; CALC TRACK/SECTOR
2622 2404  8E  25  36           STX  OUTTRK
2623 2407  8C  26  36           STY  OUTSCT
2624 240A  A9  FF           LDA  #-1
2625 240C  8D  24  36           STA  OUTPTR           ; ZERO OUT OUTUT POINTER
2626 240F  EE  24  36           INC  OUTPTR           ; POINT TO FIRST CHAR
2627 2412  AD  AD  36           LDA  CPMOTC           ; RESTORE THE CHAR
2628 2415  20  4D  24           JSR  CPMOSR           ; GET CHAR AND EOF CONDITIONS
2629 2418  60           RTS
2630
2631 2419  38           CPMOEF: SEC           ; PUT HERE SO WONT CAN STILL BRANCH TO
2632 241A  60           RTS
2633
2634 241B           CPMONE:           ; OUTUT NEXT ENTRY,
2635 241B  EE  65  36           INC  CPMEXT           ; BUMP UP & LOOK FOR NEXT EXTENT
2636 241E  A9  01           LDA  #1           ; SET OUTPUT FLAG
2637 2420  8D  3C  36           STA  TMPIOF           ; SET OUTUT FLAG
2638 2423  AD  23  36           LDA  OUTDRV           ; RE-SET DRIVE #
2639 2426  8D  33  36           STA  TMPDRV
2640 2429  AD  29  36           LDA  OUBYSC           ; RE-SET BYTES/SECTOR
2641 242C  8D  39  36           STA  TMBYSC
    
```



```

2642 242F AD 2A 36 LDA OUTDEN ; RE-SET DENSITY
2643 2432 8D 3A 36 STA TMPDEN
2644
2645 2435 AO OF LDY #16-1 ; MOVE FILENAME BACK INTO THE BUFFER
2646 2437 B9 9D 36 CPMOML: LDA CPMBFO,Y
2647 243A 99 7D 36 STA CPMBUF,Y
2648 243D 88 DEY
2649 243E 10 F7 BPL CPMOML
2650
2651 2440 20 00 21 JSR CPMOPF ; OPEN FILE
2652 2443 BO D4 BCS CPMOEF ; END OF FILE IF NO NEXT EXTENT
2653
2654 2445 EE 24 36 INC OUTPTR ; ADVANCE PAST FIRST CHAR
2655 2448 68 PLA ; RESTORE THE CHAR
2656 2449 20 4D 24 JSR CPMOSR ; PUT CHAR INTO BUFFER
2657 244C 60 RTS
2658
2659
2660 244D AE 24 36 CPMOSR: LDX OUTPTR ; CP/M OUTUT SUB-ROUTINE
2661 2450 9D 00 84 STA OUTBUF,X
2662 2453 18 CLC
2663 2454 60 RTS
2664
2665

```

```

2667
2668 2455 A9 1A          CPMCLD: LDA    #'Z&$3F      ; EOF CHAR
2669 2457 20 C5 23          JSR    CPMOUT
2670 245A A9 00          CPMCLP: LDA    #0
2671 245C 20 C5 23          JSR    CPMOUT
2672 245F 90 F9          BCC    CPMCLP      ; WAIT TILL EOF HIT
2673 2461 18             CLC
2674 2462 60             RTS
2675
2676
2677
2678
2679
2680
2681 2463          CPMCTS:
2682 2463 8D 4C 36          STA    WORD+O      ; THIS ROUTINE WAS WRITTEN WITH
2683 2466 A9 00             LDA    #0           ; THE HELP OF AN ARTICLE IN
2684 2468 8D 4D 36          STA    WORD+1      ; ACG-NJ DECEMBER 1980 PAGE 12
2685 246B OE 4C 36          ASL    WORD+O      ; ABOUT CP/M
2686 246E 2E 4D 36          ROL    WORD+1      ; CALCULATE TRACK/SECTOR
2687 2471 OE 4C 36          ASL    WORD+O      ; TEMP SAVE (LOW BYTE)
2688 2474 2E 4D 36          ROL    WORD+1
2689 2477 OE 4C 36          ASL    WORD+O      ; ZERO HIGH BYTE
2690 247A 2E 4D 36          ROL    WORD+1      ; * 2
2691 247D 18             CLC                    ; * 4
2692 247E AD 4C 36          LDA    WORD+O      ; * 8
2693 2481 69 34             ADC    #52
2694 2483 8D 4C 36          STA    WORD+O
2695 2486 AD 4D 36          LDA    WORD+1
2696 2489 69 00             ADC    #0
2697 248B 8D 4D 36          STA    WORD+1
2698 248E 18             CLC                    ; + NR MOD 8 (X = RECORD # (1-8))
2699 248F 8A             TXA                    ; X = NR (RECORD INTO BLOCK)
2700 2490 6D 4C 36          ADC    WORD+O
2701 2493 8D 4C 36          STA    WORD+O
2702 2496 AD 4D 36          LDA    WORD+1
2703 2499 69 00             ADC    #0
2704 249B 8D 4D 36          STA    WORD+1
2705 249E A2 00             LDX    #0           ; USE X TO HOLD ANSWER
2706 24A0 38             CPMCTO: SEC
2707 24A1 E8             INX
2708 24A2 AD 4C 36          LDA    WORD+O      ; / 26
2709 24A5 E9 1A             SBC    #26
2710 24A7 8D 4C 36          STA    WORD+O
2711 24AA AD 4D 36          LDA    WORD+1
2712 24AD E9 00             SBC    #0
2713 24AF 8D 4D 36          STA    WORD+1
2714 24B2 10 EC             BPL    CPMCTO
2715 24B4 CA             DEX                    ; X = TRACK NUMBER
2716 24B5 18             CLC
2717 24B6 AD 4C 36          LDA    WORD+O      ; GET REMAINDER
2718 24B9 69 1A             ADC    #26
2719 24BB 20 CO 24          JSR    CPMLTP      ; CONVERT LOGICAL -> PHYSICAL
2720 24BE A8             TAY
2721 24BF 60             RTS
    
```

```
2722
2723
2724 24C0          CPMLTP:          ; ROUTINE TO CONVERT LOGICAL -> PHYSICAL
2725 24C0  A8          TAY          ; USE AS INDEX INTO TABLE
2726 24C1  B9  C5  24  LDA  CPMLPT,Y
2727 24C4  60          RTS
2728
2729
2730 24C5          CPMLPT:          ; FOR CP/M 1.3 AND LATER
2731 24C5  01  07  0D  .BYTE 01,07,13 ; CP/M LOGICAL - PHYSICAL TABLE
2732 24C8  13  19  05  .BYTE 19,25,05 ; ACG-NJ DECEMBER 1980 PAGE 12
2733 24CB  0B  11  17  .BYTE 11,17,23
2734 24CE  03  09  0F  .BYTE 03,09,15
2735 24D1  15  02  08  .BYTE 21,02,08
2736 24D4  0E  14  1A  .BYTE 14,20,26
2737 24D7  06  0C  12  .BYTE 06,12,18
2738 24DA  18  04  0A  .BYTE 24,04,10
2739 24DD  10  16  00  .BYTE 16,22,00
2740 24E0  00  00  00  .BYTE 00,00,00
2741
2742
```

```

2744 .SBTTL IBM Disk Drivers
2745
2746 ; IBM International Buisness Machines 3740/3742
2747 ; IBM DOES NOT USE <cr>s or <lf>s
2748 ; 1 Physical Line per Physical Sector (Record)
2749
2750 ; 128 BYTES/SECTOR
2751 ; 26 SECTORS/TRACK (1-26)
2752
2753 ; SYSTEM OVERHEAD:
2754 ; TRACK 0 INDEX (DATASET LABELS)
2755 ; TRACK 74 NOT USED
2756 ; TRACK 75 /REPLACEMENTS
2757 ; TRACK 76 \FOR DEFECTIVE TRACKS
2758
2759 ; MAXIMUMS:
2760 ; FILES/DISK 19
2761 ; SPACE/DISK 148.28K (151840 BYTES)
2762 ; RECORDS/DISK 1898 (CARD IMAGES)
2763
2764 ; READ/WRITE:
2765 ; BOTH IGNORE ALL NULLS
2766 ; READ UNPAD SPACES
2767 ; WRITE PAD WITH SPACES UP TO COLUMN 80
2768
2769
2770 ; IBM VOLUME NAME
2771 ; 123456789 1.....80
2772 ; VOL1DATANM W
2773
2774 ; 1 - 4 VOL1 LABEL ID
2775 ; 5 RESERVED
2776 ; 6 - 22 DATA-SET-NAME 6 - 13 USED STANDARD
2777 ; 23 - 27 BLOCK LENGTH
2778 ; 28 RECORD ATTRIBUTE
2779 ; 29 - 30 1ST TRACK # BOE Begining of Extent
2780 ; 31 HEAD #
2781 ; 32 - 33 1ST SECTOR #
2782 ; 34 PHYSICAL RECORD LENGTH
2783 ; 35 - 36 LAST TRACK # EOE End of Extent
2784 ; 37 HEAD #
2785 ; 38 - 39 LAST SECTOR
2786 ; 40 RECORD/BLOCK FORMAT
2787
2788 ; 75 - 76 END TRACK # EOD End of Data
2789 ; 77 HEAD #
2790 ; 78 - 79 END SECTOR #
2791 ; 80 W ?
2792
2793 0100 IBMDBS = $0100 ; SINGLE DENSITY, 128 BYTES/SECTOR
2794 011A IBMSCT = $011A ; SECTOR INFO
2795

```

```

2797
2798 24E3          IBMDEF:          ; NOTHING NEEDED FOR IBM
2799 24E3      18          CLC
2800 24E4      60          RTS
2801
2802
2803 24E5      A9      00          IBMDIR: LDA      #0          ; TRACK 0
2804 24E7      8D      15      36          STA      INPTRK
2805 24EA      A9      07          LDA      #7          ; SECTOR 7
2806 24EC      8D      16      36          STA      INP SCT
2807 24EF      20      AF      2C          IBMDLP: JSR      RDSCTR
2808 24F2      AD      00      80          LDA      DIRBUF+0 ; GET 1st CHAR OF HDR (D or H)
2809 24F5      20      2F      34          JSR      E.TO.A   ; CONVERT EBCDIC TO ASCII
2810 24F8      C9      44          CMP      #'D      ; IF NOT ACTIVE, DON'T PRINT IT
2811 24FA      FO      17          BEQ      IBMDNS   ; SKIP IF INVALID ENTRY
2812 24FC      20      FO      2B          JSR      CRLFSQ  ; NEW LINE
2813 24FF      A2      00          LDX      #0          ; INIT INDEX POINTER
2814 2501      8A          IBMLP1: TXA          ; SAVE X
2815 2502      48          PHA
2816 2503      BD      00      80          LDA      DIRBUF,X
2817 2506      20      2F      34          JSR      E.TO.A   ; CONVERT EBCDIC - ASCII
2818 2509      20      1C      2C          JSR      PRVCHR  ; PRINT VALID CHARS
2819 250C      68          PLA          ; RESTORE X
2820 250D      AA          TAX
2821 250E      E8          INX
2822 250F      EO      50          CPX      #80      ; 80 CHARS/DIR ENTRY
2823 2511      90      EE          BLT      IBMLP1
2824
2825 2513      EE      16      36          IBMDNS: INC      INP SCT ; IBM DIR NEXT SECTOR
2826 2516      AD      16      36          LDA      INP SCT
2827 2519      C9      1B          CMP      #27
2828 251B      90      D2          BLT      IBMDLP  ; GET NEXT ENTRY
2829 251D      60          RTS
2830

```

```

2832
2833 251E 20 B3 19 IBMOPN: JSR GTATOM ; GET THE FILE-NAME
2834 2521 CE 45 36 DEC ATMLEN ; ????*??? MAYBE *???*??
2835
2836 2524 A9 00 IBMOPN: LDA #0 ; TRACK 0
2837 2526 8D 35 36 STA TMPTRK
2838 2529 A9 07 LDA #8-1 ; SECTOR 8 (GETS INCREMENTED TO 8)
2839 252B 8D 36 36 STA TMPST
2840 252E 8D 38 36 STA TMLSCT ; SAVE AS LAST SECTOR TOO
2841 2531 EE 36 36 IBMONS: INC TMPST ; BUMP POINTER TO NEXT SECTOR
2842 2534 EE 38 36 INC TMLSCT ; BUMP LAST SECTOR TOO
2843 2537 AD 36 36 LDA TMPST
2844 253A C9 1B CMP #27
2845 253C B0 1F BGE IBMONF ; FILE NOT FOUND
2846 253E 20 ED 2C JSR RDTEMP ; READ DIRECTORY SECTOR
2847
2848 2541 A0 00 LDY #0 ; INIT INDEX TO ZERO TO POINT TO FILENAME
2849 2543 B9 05 80 IBMOP2: LDA DIRBUF+5,Y
2850 2546 20 2F 34 JSR E.TO.A ; CONVERT EBCDIC to ASCII
2851 2549 D9 00 38 CMP ATMBUF,Y
2852 254C D0 E3 BNE IBMONS ; OPEN NEXT SECTOR
2853 254E C8 INY
2854 254F CC 45 36 CPY ATMLEN ; IF LENGTH OF ATOM THEN FOUND
2855 2552 90 EF BLT IBMOP2
2856 2554 B9 05 80 LDA DIRBUF+5,Y ; IF NEXT CHAR NOT A SPACE THEN NOT VALID MATCH
2857 2557 C9 40 CMP #$40 ; EBCDIC SPACE
2858 2559 D0 D6 BNE IBMONS
2859 255B F0 02 BEQ IBMOFF ; Open File Found
2860
2861 ; IBM Open File NOT Found
2862 255D 38 IBMONF: SEC ; SET FILE NOT FOUND FLAG
2863 255E 60 RTS
2864
2865
2866 255F IBMOFF: ; IBM Open File Found
2867 255F AD 36 36 LDA TMPST ; SAVE THE DIRECTORY SECTOR #
2868 2562 8D AE 36 STA IBMDSN
2869 2565 A2 1C LDX #29-1 ; INIT INDEX TO POINT TO 1st TRACK #
2870 2567 20 C5 25 JSR IBMOSR
2871 256A 48 PHA ; *** SAVE ON STACK
2872 256B A2 1F LDX #32-1 ; INIT INDEX TO POINT TO 1st SECTOR #
2873 256D 20 C5 25 JSR IBMOSR
2874 2570 48 PHA ; *** SAVE ON STACK
2875
2876 2571 AD 3C 36 LDA TMPIOF ; TEST IF INPUT or OUTPUT
2877 2574 F0 24 BEQ IBMOIN ; 0 = INPUT
2878 2576 D0 00 BNE IBMOOT ; 1 = OUTPUT
2879
2880
    
```

```

2882
2883 2578 A2 22          IBM00T: LDX    #35-1          ; INIT INDEX TO POINT TO LAST TRACK #
2884 257A 20 C5 25          JSR    IBMOSR
2885 257D 8D 27 36          STA    OULTRK
2886 2580 A2 25          LDX    #38-1          ; INIT INDEX TO POINT TO LAST SECTOR #
2887 2582 20 C5 25          JSR    IBMOSR
2888 2585 8D 28 36          STA    OULSCT
2889 2588 68              PLA          ; *** RESTORE 1st SECTOR #
2890 2589 8D 26 36          STA    OUTSCT
2891 258C 68              PLA          ; *** RESTORE 1st TRACK #
2892 258D 8D 25 36          STA    OUTTRK
2893
2894 2590 EA              NOP
2895 2591 EA              NOP
2896 2592 EA              NOP
2897
2898 2593 A9 00          LDA    #0
2899 2595 8D 24 36          STA    OUTPTR          ; ZERO OUTPUT POINTER
2900 2598 18              CLC
2901 2599 60              RTS
2902
2903
2904 259A A2 4A          IBM0IN: LDX    #75-1          ; INIT INDEX TO POINT TO LAST TRACK #
2905 259C 20 C5 25          JSR    IBMOSR
2906 259F 8D 17 36          STA    INLTRK
2907 25A2 A2 4D          LDX    #78-1          ; INIT INDEX TO POINT TO LAST SECTOR #
2908 25A4 20 C5 25          JSR    IBMOSR
2909 25A7 8D 18 36          STA    INLSCT
2910
2911 25AA 68              PLA          ; *** RESTORE 1st SECTOR #
2912 25AB 8D 16 36          STA    INPSCT
2913 25AE 68              PLA          ; *** RESTORE 1st TRACK #
2914 25AF 8D 15 36          STA    INPTRK
2915
2916 25B2 EE 16 36          INC    INPSCT          ; BUMP UP SECTOR POINTER (IGNORE 1st LINE)
2917 25B5 EE 18 36          INC    INLSCT          ; BUMP UP LAST SECTOR POINTER
2918
2919 25B8 A9 00          LDA    #0
2920 25BA 8D 14 36          STA    INPPTR          ; ZERO INPUT POINTER
2921 25BD 20 AF 2C          JSR    RDSCTR
2922 25C0 20 82 26          JSR    IBMUNP          ; UN-PAD THE SPACES FROM THE IBM INPUT BUFFER
2923 25C3 18              CLC
2924 25C4 60              RTS
2925
2926
2927
2928 25C5 BD 00 80          IBMOSR: LDA    DIRBUF,X          ; GET 1ST DIGIT
2929 25C8 29 OF          AND    #$0F          ; MASK OFF THE HIGH BYTE
2930 25CA 0A              ASL    A              ; SHIFT TO LEFT NIBBLE
2931 25CB 0A              ASL    A
2932 25CC 0A              ASL    A
2933 25CD 0A              ASL    A
2934 25CE 8D 4B 36          STA    ACC          ; SAVE IT
2935 25D1 BD 01 80          LDA    DIRBUF+1,X          ; GET 2ND DIGIT
2936 25D4 29 OF          AND    #$0F
    
```

```

2937 25D6 OD 4B 36 ORA ACC
2938 25D9 4C D4 1A JMP DECHX ; CONVERT BCD TO HEX
2939
2940 ; SINCE CLOSE IS !ONLY! USED FOR OUTPUT,
2941 ; THIS IS OKAY TO DO
2942 25DC AD 23 36 IBMCLD: LDA OUTDRV ; RE-SET DRIVE #
2943 25DF 8D 33 36 STA TMPDRV
2944 25E2 AD 2A 36 LDA OUTDEN ; RE-SET DENSITY
2945 25E5 8D 3A 36 STA TMPDEN
2946 25E8 AD 29 36 LDA QUBYSC ; RE-SET BYTES/SECTOR
2947 25EB 8D 39 36 STA TMBYSC
2948 25EE A9 00 00 LDA #0 ; TRACK 00
2949 25FO 8D 35 36 STA TMPTRK
2950 25F3 AD AE 36 LDA IBMDSN ; GET THE DIR SECTOR #
2951 25F6 8D 36 36 STA TMPSTC
2952 25F9 A9 80 80 LDA #DIRBUF^ ; RE-SET THE DMA PAGE
2953 25FB 8D 3B 36 STA TMPAGE
2954 25FE 20 ED 2C JSR RDTEMP ; GET THE DIRECTORY SECTOR
2955 ; UPDATE THE EOD TO LAST TRK/SCT USED
2956 2601 AD 25 36 LDA OUTTRK ; GET LAST TRACK #
2957 2604 20 1F 26 JSR IBMCSR ; COVERT TO EBCDIC-BCD
2958 2607 8E 4A 80 STX DIRBUF+75-1 ; X = HIGH BYTE
2959 260A 8C 4B 80 STY DIRBUF+76-1 ; Y = LOW BYTE
2960
2961 260D AD 26 36 LDA OUTSCT
2962 2610 20 1F 26 JSR IBMCSR ; COVERT TO EBCDIC-BCD
2963 2613 8E 4D 80 STX DIRBUF+78-1 ; X = HIGH BYTE
2964 2616 8C 4E 80 STY DIRBUF+79-1 ; Y = LOW BYTE
2965
2966 2619 20 AA 2D JSR WRTEMP ; WRITE OUT THE DIRECTORY ENTRY
2967 261C 4C B1 26 JMP IBMONS ; DO SAME THING AS Output Next Sector
2968
2969
2970 261F 20 CO 1A IBMCSR: JSR HEXDEC ; CONVERT ACC TO BCD
2971 2622 48 PHA ; SAVE IT
2972 2623 4A LSR A ; GET HIGH NIBBLE
2973 2624 4A LSR A
2974 2625 4A LSR A
2975 2626 4A LSR A
2976 2627 09 FO ORA #$FO ; CONVERT TO EBCDIC NUMBER
2977 2629 AA TAX ; SAVE HIGH BYTE IN %X
2978 262A 68 PLA ; LOW NIBBLE
2979 262B 09 FO ORA #$FO ; CONVERT TO EBCDIC NUMBER
2980 262D A8 TAY ; SAVE LOW BYTE IN %Y
2981 262E 60 RTS ; RTS X = HIGH Y = LOW (BOTH IN EBCDIC BCD)
2982

```



```

2984
2985 262F AE 14 36 IBMINP: LDX INPPTR ; GET INPUT INDEX POINTER
2986 2632 BD 00 82 LDA INPBUF,X
2987 2635 FO 08 BEQ IBMINS
2988 2637 20 2F 34 JSR E.TO.A ; CONVERT TO ASCII
2989 263A EE 14 36 INC INPPTR ; BUMP UP POINTER
2990 263D 18 CLC ; CLEAR CARRY
2991 263E 60 RTS
2992
2993 ;.ERROR ; NOT WORKING COMPLETELY CORRECT (MAYBE IT IS)
2994
2995 263F EE 16 36 IBMINS: INC INPSCT ; BUMP UP SECTOR NUMBER
2996 2642 AD 15 36 LDA INPTRK ; CHECK END TRACK
2997 2645 CD 17 36 CMP INLTRK ; CHECK IF END TRACK REACHED
2998 2648 90 0C BLT IBMIOK ; OKAY IF TRK # <
2999 264A AD 16 36 LDA INPSCT
3000 264D CD 18 36 CMP INLSCS ; CHECK IF AT END OF FILE
3001 2650 90 04 BLT IBMIOK ; BLE
3002 2652 FO 02 BEQ IBMIOK
3003 2654 BO 2A BGE IBMIEF ; WILL ALWAYS BRANCH
3004
3005 2656 AD 16 36 IBMIOK: LDA INPSCT ; CHECK IF END OF TRACK REACHED
3006 2659 C9 1B CMP #27
3007 265B 90 14 BLT IBMIST ; BRANCH IF SAME TRACK
3008 265D A9 01 LDA #1 ; SET SECTOR #1
3009 265F 8D 16 36 STA INPSCT
3010 2662 EE 15 36 INC INPTRK ; BUMP UP TRACK #
3011 2665 AD 15 36 LDA INPTRK
3012 2668 CD 17 36 CMP INLTRK ; CHECK IF AT EOF
3013 266B 90 04 BLT IBMIST ; BLE
3014 266D FO 02 BEQ IBMIST
3015 266F BO 0F BGE IBMIEF ; EOF IF GREATER.
3016
3017 2671 20 AF 2C IBMIST: JSR RDSCTR ; READ NEXT SECTOR
3018 2674 20 82 26 JSR IBMUNP ; UN-PAD THE SPACES FROM THE INPUT BUFFER
3019 2677 A2 00 LDX #0 ; ZERO %X
3020 2679 8E 14 36 STX INPPTR
3021 267C A9 0D LDA #0D ; INDICATE END OF LINE
3022 267E 18 CLC
3023 267F 60 RTS
3024
3025 2680 38 IBMIEF: SEC ; SET EOF FLAG
3026 2681 60 RTS
3027
    
```

```
3029
3030 2682 A2 4F IBMUNP: LDX #79 ; UN-PAD THE SPACES
3031 2684 BD 00 82 IBMUN1: LDA INPBUF,X
3032 2687 C9 40 CMP #$40 ; EBCDIC <space>
3033 2689 D0 03 BNE IBMUN2
3034 268B CA DEX
3035 268C 10 F6 BPL IBMUN1 ; WILL ALWAYS BRANCH UNLESS LINE IS ALL BLANK
3036
3037 268E E8 IBMUN2: INX ; POINT TO 1st SPACE (IF ALL BLANK, THEN X = 0)
3038 268F A9 00 LDA #$00 ; INDICATE END OF RECORD
3039 2691 9D 00 82 STA INPBUF,X
3040 2694 60 IBMUNX: RTS
3041
3042
```

```

3044
3045 2695 29 7F          IBMOUT: AND    #$7F
3046 2697 C9 OD          CMP    #$0D          ; WRITE RECORD ON <cr>
3047 2699 FO 16          BEQ    IBMONS
3048 269B C9 OA          CMP    #$0A
3049 269D FO OE          BEQ    IBMOEX        ; IGNORE LINE-FEEDS.
3050 269F 20 34 34      JSR    A.T.O.E        ; CONVERT TO EBCDIC
3051 26A2 AE 24 36      LDX    OUTPTR
3052 26A5 9D 00 84      STA    OUTBUF,X
3053 26A8 EE 24 36      INC    OUTPTR
3054 26AB FO 02          BEQ    IBMOER        ; ERROR ON OVERFLOW
3055 26AD 18             IBMOEX: CLC
3056 26AE 60             RTS
3057
3058 26AF             IBMOEF:             ; END OF FILE
3059 26AF 38           IBMOER: SEC        ; ERROR
3060 26B0 60             RTS
3061
3062 26B1 EE 26 36      IBMONS: INC    OUTSCT    ; BUMP UP SECTOR NUMBER
3063 26B4 AD 26 36      LDA    OUTSCT
3064 26B7 CD 28 36      CMP    OULSCT        ; CHECK IF AT END OF FILE
3065 26BA 90 OC          BLT    IBMOOK
3066 26BC AD 25 36      LDA    OUTTRK        ; CHECK END TRACK
3067 26BF CD 27 36      CMP    OULTRK        ; CHECK IF END TRACK REACHED
3068 26C2 90 04          BLT    IBMOOK        ; BLE
3069 26C4 FO 02          BEQ    IBMOOK
3070 26C6 BO E7          BGE    IBMOER        ; BGT ERROR
3071
3072 26C8 AD 26 36      IBMOOK: LDA    OUTSCT
3073 26CB C9 1B          CMP    #27
3074 26CD 90 14          BLT    IBMOST        ; NEXT TRACK IF > 27
3075 26CF A9 01          LDA    #1            ; SET SECTOR 1
3076 26D1 8D 26 36      STA    OUTSCT
3077 26D4 EE 25 36      INC    OUTTRK        ; BUMP UP TRACK #
3078 26D7 AD 25 36      LDA    OUTTRK
3079 26DA CD 27 36      CMP    OULTRK        ; CHECK IF AT EOF
3080 26DD 90 04          BLT    IBMOST        ; BLE
3081 26DF FO 02          BEQ    IBMOST
3082 26E1 BO CC          BGE    IBMOEF        ; EOF IF GREATER.
3083
3084
3085 26E3 A9 84          IBMOST: LDA    #OUTBUF^ ; BRANCH IF SAME TRACK
3086 26E5 8D 2B 36      STA    OUPAGE        ; SET BUFFER ADDRESS
3087 26E8 20 6F 2D      JSR    WRSCTR
3088 26EB A2 00          LDX    #0
3089 26ED A9 40          LDA    #$40          ; EBCDIC SPACE
3090 26EF 9D 00 84      IBMPAD: STA    OUTBUF,X    ; ZERO OUT THE BUFFER
3091 26F2 CA             DEX                ; (SPACE)
3092 26F3 DO FA          BNE    IBMPAD
3093 26F5 A9 00          LDA    #0
3094 26F7 8D 24 36      STA    OUTPTR
3095 26FA 18             CLC
3096 26FB 60             RTS
3097

```

```
3099          .SBTTL  RSX      Disk Drivers
3100
3101          ; RSX  RSX-11 FOR PDP-11
3102
3103          0100      RSXDBS =      $0100          ; SINGLE DENSITY, 128 BYTES/SECTOR
3104          011A      RSXSCT =      $011A
3105
3106
3107  26FC          RSXDEF:
3108  26FC  18      RSXDIR: CLC
3109  26FD  60      RTS
3110
3111
3112  26FE          RSXOPN:
3113  26FE          RSXINP:
3114  26FE          RSXOUT:
3115  26FE          RSXCLO:
3116  26FE  38      SEC
3117  26FF  60      RTS
3118
3119
3120          .SBTTL  HDE      Disk Drivers
3121
3122          ; HDE  Hudson Digital Electronics inc.
3123
3124          ;      1771 FDC CONTROLLER CHIP
3125
3126          FFFF      HDEDBS =      $FFFF          ;
3127          FFFF      HDESCT =      $FFFF          ; ?
3128
3129  2700          HDEDEF:
3130  2700          HDEDIR:
3131  2700  18      CLC
3132  2701  60      RTS
3133
3134
3135
3136
3137  2702          HDEOPN:
3138  2702          HDEINP:
3139  2702          HDEOUT:
3140  2702          HDECLO:
3141  2702  38      SEC
3142  2703  60      RTS
3143
3144
```

```

3146 .SBTTL TRS-80 Disk Drivers
3147
3148 ; TRS-80 The Radio Shack 80 TRS-DOS
3149 ; 1771/1791 CONTROLER CHIP
3150
3151 ; DISK FORMAT:
3152 ; TRACK 0 SINGLE DENSITY 128 BYTES/SECTOR
3153 ; TRACK 1-76 DOUBLE DENSITY
3154 ; 26 SECTORS $01 - $1A
3155 ; 256 BYTES/SECTOR
3156 ; TRACK 44 DIRECTORY
3157 ; SECTOR 3-26
3158
3159 ; BYTE(s) VALUE MEANING
3160 ; 00 ATTRIBUTES
3161 ; 01 0 ?
3162 ; 02 0 ?
3163 ; 03 FILE SIZE LOW BYTE
3164 ; 04 0/1 RECORD LENGTH 0=256 1=1
3165 ; 05-0F FILE NAME
3166 ; 10-13 PASSWORD PROTECTION ?
3167 ; 14 FILE SIZE HIGH BYTE
3168 ; 15
3169 ; 16
3170 ; 17
3171 ; 18-35 FF ?
3172 ; 36 37 38 DATE YY DD MM
3173 ; 39 0-6 PROTECTION LEVEL
3174 ; 3A 3B 3C DATE YY DD MM
3175 ; 3D 3E FF ?
3176 ; 3F 00 ?
3177
3178 0201 TRSDBS = $0201 ; DOUBLE DENSITY, 256 BYTES/SECTOR
3179 011A TRSSCT = $011A ; SECTOR INFO
3180
3181 2704 TRSDEF: ; CHANGE
3182 2704 18 CLC
3183 2705 60 RTS
3184
    
```

```

3186
3187 2706 60          TRSDEX: RTS          ; DIR EXIT
3188
3189 2707 A9 2C      TRSDIR: LDA      #44          ; DIR TRACK = 44 ($2C)
3190 2709 8D 15 36      STA      INPTRK
3191 270C A9 02          LDA      #3-1          ; DIR STARTS IN SECTOR 3
3192 270E 8D 16 36      STA      INPSTC
3193 2711 EE 16 36      TRSDLP: INC      INPSTC          ; BUMP UP TO NEXT SECTOR
3194 2714 AD 16 36      LDA      INPSTC
3195 2717 C9 1B          CMP      #27
3196 2719 B0 EB          BGE      TRSDEX
3197 271B 20 AF 2C      JSR      RDSCTR
3198 271E A2 00          LDX      #0
3199
3200 2720 BD 10 80      TRSDNE: LDA      DIRBUF+16,X  ; DIR NEXT ENTRY
3201 2723 D0 05          BNE      TRSDOK          ; CHECK ACTIVE FLAG
3202 2725 A0 40          LDY      #64          ; IF FILE NOT ACTIVE, THEN SKIP ENTIRE ENTRY
3203 2727 4C A2 27      JMP      TRSDSE          ; DIR SKIP ENTRY
3204
3205 272A 20 FO 2B      TRSDOK: JSR      CRLFSQ          ; NEW LINE
3206 272D E8            INX
3207 272E E8            INX
3208 272F E8            INX
3209 2730 E8            INX
3210 2731 E8            INX
3211 2732 A0 0B          LDY      #11          ; 11 CHARS IN NAME FIELD
3212 2734 20 7B 1C      TRSLP2: JSR      DIRCHR
3213 2737 88            DEY
3214 2738 D0 FA          BNE      TRSLP2
3215 273A 20 3A 2C      JSR      SPACE4          ; 4 SPACES
3216
3217 273D BD 27 80      LDA      DIRBUF-16+$37,X ; GET MONTH
3218 2740 20 AF 27      JSR      TRSDSR          ; PRINT IT IN BCD & a space
3219 2743 BD 28 80      LDA      DIRBUF-16+$38,X ; GET DAY
3220 2746 20 AF 27      JSR      TRSDSR          ; PRINT IT IN BCD & a space
3221 2749 BD 26 80      LDA      DIRBUF-16+$36,X ; GET YEAR
3222 274C 20 AF 27      JSR      TRSDSR          ; PRINT IT IN BCD & a space
3223 274F 20 09 10      JSR      CTYSPA
3224
3225 2752 BD 2B 80      LDA      DIRBUF-16+$3B,X ; GET 2nd MONTH
3226 2755 20 AF 27      JSR      TRSDSR          ; PRINT IT IN BCD & a space
3227 2758 BD 2C 80      LDA      DIRBUF-16+$3C,X ; GET 2nd DAY
3228 275B 20 AF 27      JSR      TRSDSR          ; PRINT IT IN BCD & a space
3229 275E BD 2A 80      LDA      DIRBUF-16+$3A,X ; GET 2nd YEAR
3230 2761 20 AF 27      JSR      TRSDSR          ; PRINT IT IN BCD & a space
3231 2764 20 40 2C      JSR      SPACE2
3232
3233 2767 BD FO 7F      LDA      DIRBUF-16+$00,X ; GET ATTRIBUTE BYTE (BYTE 0)
3234 276A 48            PHA          ; SAVE IT
3235 276B 29 20          AND      #$20          ; MASK OUT Data/Pgm BIT
3236 276D D0 04          BNE      TRSDAP
3237 276F A9 44          LDA      #'D
3238 2771 D0 02          BNE      TRSDSP          ; SKIP OVER LDA 'P'
3239 2773 A9 50          TRSDAP: LDA      #'P          ; TRS Dir Attribute 'P'
3240 2775 20 95 2B      TRSDSP: JSR      CTYOUT
    
```

```

3241 2778 68          PLA          ; RESTORE ATTRIBUTE BYTE
3242 2779 30 04      BMI          TRSDAS   ; CHECK IF 'S' or '*'
3243 277B A9 2A      LDA          #'*
3244 277D DO 02      BNE          TRSDSS   ; SKIP OVER LDA 'S
3245 277F A9 53      TRSDAS: LDA   #'S     ; TRS Dir Attribute 'S'
3246 2781 20 95 2B   TRSDSS: JSR   CTYOUT
3247
3248 2784 A9 3F          LDA          #'?          ; PRINT '?' FOR NOW FIGURE OUT LATER
3249 2786 20 95 2B   JSR          CTYOUT
3250 2789 BD 29 80    LDA          DIRBUF-16+$39,X ; GET PASSWORD LEVEL NUMBER
3251 278C 09 30      ORA          #$30      ; CONVERT TO ASSCI NUMBER
3252 278E 20 95 2B   JSR          CTYOUT
3253 2791 20 40 2C   JSR          SPACE2
3254 2794 BD 04 80    LDA          DIRBUF-16+$14,X ; GET # OF CHARS HIGH BYTE
3255 2797 20 C3 2B   JSR          PRTBYT      ; PRINT IT
3256 279A BD F3 7F    LDA          DIRBUF-16+$03,X ; GET # OF CHARS LOW BYTE
3257 279D 20 C3 2B   JSR          PRTBYT      ; PRINT IT
3258
3259 27A0 A0 30      LDY          #64-16      ; INCREMENT UP TO NEXT ENTRY
3260 27A2 E8          TRSDSE: INX      ; TRS DIR SKIP (REST OF) ENDTRY
3261 27A3 88          DEY
3262 27A4 DO FC      BNE          TRSDSE
3263 27A6 8A          TRSDNS: TXA      ; JUST TO TEST X
3264 27A7          JNE          TRSDNE   ; DO NEXT ENTRY IF NOT END OF SECTOR
(1) 27A7 FO 03      BEQ          .+5
(1) 27A9 4C 20 27   JMP          TRSDNE
3265 27AC 4C 11 27   JMP          TRSDLP
3266
3267 27AF 20 BA 1A    TRSDSR: JSR   HXTDCP   ; PRINT THE BYTE IN BCD
3268 27B2 4C 09 10   JMP          CTYSPA   ; PRINT a <space>
3269
3270
3271
3272
3273 27B5          TRSOPN:
3274 27B5          TRSINP:
3275 27B5          TRSOUT:
3276 27B5          TRSCLO:
3277 27B5 38          SEC
3278 27B6 60          RTS
3279

```

```
3281          .SBTTL UCSD   Disk Drivers
3282
3283          ; USC          UCSD PASCAL SINGLE DENSITY
3284
3285          0100          PS1DBS =          $0100          ; SINGLE DENSITY, 128 BYTES/SECTOR
3286          011A          PS1SCT =          $011A          ; SECTOR INFO
3287
3288          27B7          PS1DEF:
3289          27B7          PS1DIR:
3290          27B7          18          CLC
3291          27B8          60          RTS
3292
3293          27B9          PS10PN:
3294          27B9          PS1INP:
3295          27B9          PS1OUT:
3296          27B9          PS1CLO:
3297          27B9          38          SEC
3298          27BA          60          RTS
3299
3300
3301          ; USC          M.T.U. UCSD PASCAL DOUBLE DENSITY
3302
3303          0201          PS2DBS =          $0201          ; DOUBLE DENSITY, 256 BYTES/SECTOR
3304          011A          PS2SCT =          $011A          ; SECTOR INFO
3305
3306          27BB          PS2DEF:
3307          27BB          PS2DIR:
3308          27BB          18          CLC
3309          27BC          60          RTS
3310
3311          27BD          PS20PN:
3312          27BD          PS2INP:
3313          27BD          PS2OUT:
3314          27BD          PS2CLO:
3315          27BD          38          SEC
3316          27BE          60          RTS
3317
3318
3319
3320
```



```

3322 .SBTTL USER Disk Drivers
3323
3324 ; USER DISK TYPE
3325
3326 ; USER DEVICE JUMP TABLE
3327 ; GIVES ERROR IF UNDEFINED (NOT CHANGED by USER)
3328
3329 FFFF USRDBS = $FFFF ; USER DEFINED (SEE TABLE DBSTAB, ABOVE)
3330 FFFF USRSCT = $FFFF
3331
3332 27BF 4C D1 27 USRDEF: JMP USR.ER
3333
3334
3335 27C2 4C D1 27 USRDIR: JMP USR.ER ; USER DIRECTORY
3336 27C5 4C D1 27 USROPN: JMP USR.ER ; USER OPEN FILE
3337 27C8 4C D1 27 USRINP: JMP USR.ER ; USER READ FILE
3338 27CB 4C D1 27 USROUT: JMP USR.ER ; USER WRITE FILE
3339 27CE 4C D1 27 USRCLO: JMP USR.ER ; USER CLOSE FILE
3340
3341
3342 27D1 A9 08 USR.ER: LDA #USRERR ; USER ERROR
3343 27D3 20 82 2F JSR ERROR
3344
3345
3346 ; ERROR DRIVERS JUST INCASE THEY GET INVOKED
3347 ; (ERROR DEFINE IS VALID)
3348
3349 FFFF ERRDBS = $FFFF
3350 FFFF ERRSCT = $FFFF
3351
3352 27D6 ERRDEF:
3353 27D6 ERRDIR:
3354 27D6 18 CLC
3355 27D7 60 RTS
3356
3357 27D8 ERROPN:
3358 27D8 ERRCLO:
3359 27D8 ERRINP:
3360 27D8 ERROUT:
3361 27D8 38 SEC
3362 27D9 60 RTS
3363
3364
3365 ; OTHER POSSIBLE TYPES
3366 ; NAME DENSITY BYTES/SECTOR
3367 ;
3368 ; XEROX SINGLE 512
3369 ; CPT SINGLE 256
3370
3371
  
```

```

3373      .SBTTL  CHROMATICS      Disk Drivers
3374
3375      ;          CHROMATIC 7900 68000 SYSTEM
3376
3377      0100      CMXDBS  =      $0100      ; DOUBLE DENSITY, 256 BYTES/SECTOR
3378      010A      CMXSCT  =      $010A      ; SECTOR INFO
3379
3380      27DA      CMXDEF:          ; NOTHING NEEDED FOR CHROMATICS
3381      27DA      18          CLC
3382      27DB      60          RTS
3383
3384      27DC      A9      00      CMXDIR: LDA      #0          ; TRACK 0
3385      27DE      8D      15      36      STA      INPTRK
3386      27E1      A9      01          LDA      #1          ; SECTOR 1
3387      27E3      8D      16      36      STA      INPST
3388      27E6      20      AF      2C      CMXDLP: JSR      RDSCTR
3389      27E9      AD      00      80      LDA      DIRBUF+0
3390      27EC      FO      6C          BEQ      CMXD99      ; END OF DIRECTORY
3391      27EE      A2      00          LDX      #0
3392      27FO      AO      00      CMXLP1: LDY      #0
3393      27F2      20      FO      2B      JSR      CRLFSQ      ; NEW-LINE
3394      27F5      BD      00      80      LDA      DIRBUF,X
3395      27F8      FO      60          BEQ      CMXD99      ; END OF DIRECTORY
3396      27FA      20      7B      1C      CMXLP2: JSR      DIRCHR
3397      27FD      C8          INY
3398      27FE      CO      08          CPY      #8          ; PRIMARY FILE NAME
3399      2800      90      F8          BLT      CMXLP2
3400      2802      20      A3      1C      JSR      DIRPER      ; PRINT A PERIOD
3401      2805      20      7B      1C      JSR      DIRCHR      ; 9 EXTENSION
3402      2808      20      7B      1C      JSR      DIRCHR      ; 10 EXTENSION
3403      280B      20      7B      1C      JSR      DIRCHR      ; 11 EXTENSION
3404      280E      E8          INX          ; NOT USED
3405      280F      20      09      10      JSR      CTYSPA
3406      2812      20      7B      1C      JSR      DIRCHR      ; PASS-WORD
3407      2815      20      7B      1C      JSR      DIRCHR
3408      2818      20      09      10      JSR      CTYSPA
3409      281B      20      8B      1C      JSR      DIRBT4      ; START-OF-FILE
3410      281E      20      8B      1C      JSR      DIRBT4      ; LENGTH-OF-FILE
3411      2821      20      8B      1C      JSR      DIRBT4      ; ORIGIN DATE
3412      2824      20      8B      1C      JSR      DIRBT4      ; LAST ACCESS DATE
3413
3414      2827      BD      00      80          LDA      DIRBUF,X
3415      282A      8D      BF      36      STA      CMXTMP+0
3416      282D      E8          INX          ; POINT TO SECOND BYTE OF STATUS
3417      282E      BD      00      80      LDA      DIRBUF,X
3418      2831      8D      CO      36      STA      CMXTMP+1
3419      2834      AO      00          LDY      #0
3420      2836      B9      5C      28      CMXDL3: LDA      CMXTXT,Y      ; FILE-STATUS LOOP
3421      2839      OE      CO      36      ASL      CMXTMP+1
3422      283C      2E      BF      36      ROL      CMXTMP+0
3423      283F      BO      02          BCS      CMXDL4
3424      2841      A9      2E          LDA      #'
3425      2843      20      95      2B      CMXDL4: JSR      CTYOUT
3426      2846      C8          INY
3427      2847      CO      10          CPY      #16          ; 16 BITS IN FILE-STATUS WORD

```

3428	2849	90	EB		BLT	CMXDL3
3429						
3430	284B	E8			INX	
3431	284C	EO	80		GPX	#128
3432	284E	90	AO		BLT	CMXLP1
3433	2850	EE	16	36	INC	INPSC
3434	2853	AD	16	36	LDA	INPSC
3435	2856	C9	1A		CMP	#26
3436	2858	90	8C		BLT	CMXDLP
3437	285A	18			CMXD99: CLC	
3438	285B	60			RTS	
3439					:	'FEDCBA9876543210'
3440	285C	49	2D	2D	CMXTXT: .ASCII	'I-----S?????OK'
3441						
3442					:	BIT MEANING
3443					:	-----
3444					:	15 INVISIBLE FILE
3445					:	14-9 UNUSED
3446					:	8 SYSTEM
3447					:	7
3448					:	6
3449					:	5
3450					:	4
3451					:	3
3452					:	2
3453					:	1 ODD FILE LENGTH
3454					:	0 KILLED
3455					:	

```

3457
3458 286C 20 B3 19 CMXOPN: JSR GTATOM ; GET THE FILE-NAME
3459 286F A9 00 LDA #0 ; TRACK 0
3460 2871 8D 35 36 STA TMPTRK
3461 2874 A9 01 LDA #1 ; SECTOR 1
3462 2876 8D 36 36 STA TMPST
3463 2879 8D 38 36 STA TMLSCT ; SAVE AS LAST SECTOR TOO
3464 287C 20 ED 2C CMXONS: JSR RDTEMP ; READ DIRECTORY SECTOR
3465 287F A0 00 LDY #0*32
3466 2881 20 A3 29 JSR CMXCHK
3467 2884 F0 24 BEQ CMXOFF
3468 2886 A0 20 LDY #1*32
3469 2888 20 A3 29 JSR CMXCHK
3470 288B F0 1D BEQ CMXOFF
3471 288D A0 40 LDY #2*32
3472 288F 20 A3 29 JSR CMXCHK
3473 2892 F0 16 BEQ CMXOFF
3474 2894 A0 60 LDY #3*32
3475 2896 20 A3 29 JSR CMXCHK
3476 2899 F0 0F BEQ CMXOFF
3477 289B EE 36 36 INC TMPST
3478 289E EE 38 36 INC TMLSCT
3479 28A1 AD 36 36 LDA TMPST
3480 28A4 C9 1A CMP #26
3481 28A6 90 D4 BLT CMXONS
3482 28A8 38 SEC ; SET NOT FOUND
3483 28A9 60 RTS
3484
3485 28AA CMXOFF: ; FILE FOUND
3486 28AA E9 OE 80 LDA DIRBUF+14,Y ; SAVE STAR-OF-FILE
3487 28AD 8D AF 36 STA CMXSOF+0
3488 28B0 B9 OF 80 LDA DIRBUF+15,Y
3489 28B3 8D B0 36 STA CMXSOF+1
3490 28B6 B9 10 80 LDA DIRBUF+16,Y
3491 28B9 8D B1 36 STA CMXSOF+2
3492 28BC B9 11 80 LDA DIRBUF+17,Y
3493 28BF 8D B2 36 STA CMXSOF+3
3494 28C2 B9 12 80 LDA DIRBUF+18,Y ; SAVE LENGTH-OF-FILE
3495 28C5 8D B3 36 STA CMXLOF+0
3496 28C8 B9 13 80 LDA DIRBUF+19,Y
3497 28CB 8D B4 36 STA CMXLOF+1
3498 28CE B9 14 80 LDA DIRBUF+20,Y
3499 28D1 8D B5 36 STA CMXLOF+2
3500 28D4 B9 15 80 LDA DIRBUF+21,Y
3501 28D7 8D B6 36 STA CMXLOF+3
3502
3503 28DA 38 SEC
3504 28DB AD B1 36 LDA CMXSOF+2
3505 28DE E9 OD ; 26*128=$0D00
3506 28E0 8D B1 36 STA CMXSOF+2
3507 28E3 AD B0 36 LDA CMXSOF+1
3508 28E6 E9 00 SBC #0
3509 28E8 8D B0 36 STA CMXSOF+1
3510 28EB AD AF 36 LDA CMXSOF+0
3511 28EE E9 00 SBC #0
    
```

```

3512 28F0 8D AF 36 STA CMXSOF+0
3513
3514 28F3 A9 00 LDA #0 ; ZERO OUT TRACK/SECTOR
3515 28F5 8D 35 36 STA TMPTRK
3516 28F8 8D 36 36 STA TMPSC
3517 28FB 38 CMXCTS: SEC
3518 28FC AD B1 36 LDA CMXSOF+2
3519 28FF E9 1A SBC #$1A ; 26*256=$1A00
3520 2901 8D B1 36 STA CMXSOF+2
3521 2904 AD B0 36 LDA CMXSOF+1
3522 2907 E9 00 SBC #0
3523 2909 8D B0 36 STA CMXSOF+1
3524 290C AD AF 36 LDA CMXSOF+0
3525 290F E9 00 SBC #0
3526 2911 8D AF 36 STA CMXSOF+0
3527 2914 EE 35 36 INC TMPTRK
3528 2917 AD AF 36 LDA CMXSOF+0
3529 291A 10 DF BPL CMXCTS
3530 291C 18 CLC ; ADD BACK ONCE TO GET REMAINDER
3531 291D AD B1 36 LDA CMXSOF+2
3532 2920 69 1A ADC #$1A ; 26*256=$1A00
3533 2922 8D B1 36 STA CMXSOF+2
3534 2925 AD B0 36 LDA CMXSOF+1
3535 2928 69 00 ADC #0
3536 292A 8D B0 36 STA CMXSOF+1
3537 292D AD AF 36 LDA CMXSOF+0
3538 2930 69 00 ADC #0
3539 2932 8D AF 36 STA CMXSOF+0
3540 2935 AD B1 36 LDA CMXSOF+2 ; GET LOW-MIDDLE BYTE
3541 2938 8D 36 36 STA TMPSC
3542 293B EE 36 36 INC TMPSC
3543 293E AD B2 36 LDA CMXSOF+3 ; GET LOW BYTE (BYTE POINTER)
3544 2941 8D 34 36 STA TMPPTR
3545
3546 2944 AD 3C 36 LDA TMPIOF ; TEST IF INPUT or OUTPUT
3547 2947 FO 02 BEQ CMXOIN ; 0 = INPUT
3548 2949 DO 2C BNE CMXOOT ; 1 = OUTPUT
3549
3550 294B AD 35 36 CMXOIN: LDA TMPTRK ; CHROMATICS OPEN IN
3551 294E 8D 15 36 STA INPTRK
3552 2951 AD 36 36 LDA TMPSC
3553 2954 8D 16 36 STA INPSCT
3554 2957 AD 34 36 LDA TMPPTR
3555 295A 8D 14 36 STA INPPTR
3556 295D AO 03 LDY #4-1
3557 295F B9 B3 36 CMXOIL: LDA CMXLDF,Y ; MOVE THE FILE LENGTH
3558 2962 99 B7 36 STA CMXIL,Y
3559 2965 88 DEY
3560 2966 10 F7 BPL CMXOIL
3561
3562 2968 A9 01 LDA #$01 ; SET DENSITY & BYTES/SECTOR
3563 296A 8D 19 36 STA INBYSC ; 256 BYTES/SECTOR
3564 296D A9 02 LDA #$02 ; DOUBLE DENSITY
3565 296F 8D 1A 36 STA INPDEN
3566 2972 20 AF 2C JSR RDSCTR
    
```

```
3567 2975 18 CLC
3568 2976 60 RTS
3569
3570 2977 AD 35 36 CMXOOT: LDA TMPTRK ; CHROMATICS OPEN OUT
3571 297A 8D 25 36 STA OUTTRK
3572 297D AD 36 36 LDA TMPSCCT
3573 2980 8D 26 36 STA OUTSCT
3574 2983 AD 34 36 LDA TMPPTR
3575 2986 8D 24 36 STA OUTPTR
3576 2989 AO 03 LDY #4-1
3577 298B B9 B3 36 CMXOOL: LDA CMXLOF,Y ; MOVE THE FILE LENGTH
3578 298E 99 BB 36 STA CMXOL,Y
3579 2991 88 DEY
3580 2992 10 F7 BPL CMXOOL
3581 ; SET DENSITY & BYTES/SECTOR
3582 2994 A9 01 LDA #$01 ; 256 BYTES/SECTOR
3583 2996 8D 29 36 STA OUBYSC
3584 2999 A9 02 LDA #$02 ; DOUBLE DENSITY
3585 299B 8D 2A 36 STA OUTDEN
3586 299E 20 AF 2C JSR RDSCTR
3587 29A1 18 CLC
3588 29A2 60 RTS
3589
3590
```

```

3592
3593 29A3          CMXCHK:          ; CHROMATICS DIRECTORY CHECK
3594 29A3 A2 00          LDX      #0
3595 29A5 BD 00 38        LDA      ATMBUF,X
3596 29A8 D9 00 80        CMP      DIRBUF+0,Y ; 1ST CHAR
3597 29AB DO 5D          BNE      CMXCKN
3598
3599 29AD E8          INX
3600 29AE BD 00 38        LDA      ATMBUF,X ; 2ND CHAR
3601 29B1 C9 2E          CMP      #'
3602 29B3 FO 58          BEQ      CMXCPF ; PRIMARY NAME FOUND
3603 29B5 D9 01 80        CMP      DIRBUF+1,Y
3604 29B8 DO 50          BNE      CMXCKN
3605 29BA E8          INX ; 3RD CHAR
3606 29BB BD 00 38        LDA      ATMBUF,X
3607 29BE C9 2E          CMP      #'
3608 29C0 FO 4B          BEQ      CMXCPF
3609 29C2 D9 02 80        CMP      DIRBUF+2,Y
3610 29C5 DO 43          BNE      CMXCKN
3611 29C7 E8          INX ; 4TH CHAR
3612 29C8 BD 00 38        LDA      ATMBUF,X
3613 29CB C9 2E          CMP      #'
3614 29CD FO 3E          BEQ      CMXCPF
3615 29CF D9 03 80        CMP      DIRBUF+3,Y
3616 29D2 DO 36          BNE      CMXCKN
3617 29D4 E8          INX ; 5TH CHAR
3618 29D5 BD 00 38        LDA      ATMBUF,X
3619 29D8 C9 2E          CMP      #'
3620 29DA FO 31          BEQ      CMXCPF
3621 29DC D9 04 80        CMP      DIRBUF+4,Y
3622 29DF DO 29          BNE      CMXCKN
3623 29E1 E8          INX ; 6TH CHAR
3624 29E2 BD 00 38        LDA      ATMBUF,X
3625 29E5 C9 2E          CMP      #'
3626 29E7 FO 24          BEQ      CMXCPF
3627 29E9 D9 05 80        CMP      DIRBUF+5,Y
3628 29EC DO 1C          BNE      CMXCKN
3629 29EE E8          INX ; 7TH CHAR
3630 29EF BD 00 38        LDA      ATMBUF,X
3631 29F2 C9 2E          CMP      #'
3632 29F4 FO 17          BEQ      CMXCPF
3633 29F6 D9 06 80        CMP      DIRBUF+6,Y
3634 29F9 DO 0F          BNE      CMXCKN
3635 29FB E8          INX ; 8TH CHAR
3636 29FC BD 00 38        LDA      ATMBUF,X
3637 29FF C9 2E          CMP      #'
3638 2A01 FO 0A          BEQ      CMXCPF
3639 2A03 D9 07 80        CMP      DIRBUF+7,Y
3640 2A06 DO 02          BNE      CMXCKN
3641 2A08 FO 03          BEQ      CMXCPF
3642 2A0A A9 FF          CMXCKN: LDA      #$FF ; NON-ZERO
3643 2A0C 60          RTS
3644
3645 2A0D          CMXCPF:          ; PRIMARY NAME FOUND
3646 2A0D E8          INX ; EXT 1ST CHAR
    
```

3647	2AOE	BD	00	38	LDA	ATMBUF,X	
3648	2A11	D9	08	80	CMP	DIRBUF+8,Y	
3649	2A14	DO	F4		BNE	CMXCKN	
3650	2A16	E8			INX		; EXT 2ND CHAR
3651	2A17	BD	00	38	LDA	ATMBUF,X	
3652	2A1A	D9	09	80	CMP	DIRBUF+9,Y	
3653	2A1D	DO	EB		BNE	CMXCKN	
3654	2A1F	E8			INX		; EXT 3RD CHAR
3655	2A20	BD	00	38	LDA	ATMBUF,X	
3656	2A23	D9	0A	80	CMP	DIRBUF+10,Y	
3657	2A26	DO	E2		BNE	CMXCKN	
3658	2A28	18			CLC		; SET FOUND
3659	2A29	60			RTS		
3660							
3661	2A2A				CMXCLO:		
3662	2A2A	18			CLC		
3663	2A2B	60			RTS		
3664							


```

3666
3667 2A2C 38          CMXINP: SEC
3668 2A2D AD BA 36    LDA CMXIL+3
3669 2A30 E9 01      SBC #1
3670 2A32 8D BA 36    STA CMXIL+3
3671 2A35 AD B9 36    LDA CMXIL+2
3672 2A38 E9 00      SBC #0
3673 2A3A 8D B9 36    STA CMXIL+2
3674 2A3D AD B8 36    LDA CMXIL+1
3675 2A40 E9 00      SBC #0
3676 2A42 8D B8 36    STA CMXIL+1
3677 2A45 AD B7 36    LDA CMXIL+0
3678 2A48 E9 00      SBC #0
3679 2A4A 8D B7 36    STA CMXIL+0
3680 2A4D 2D B8 36    AND CMXIL+1
3681 2A50 2D B9 36    AND CMXIL+2
3682 2A53 2D BA 36    AND CMXIL+3
3683 2A56 C9 FF      CMP #$FF ; CHECK IF = -1
3684 2A58 FO 2C      BEQ CMXEOF
3685
3686 2A5A AE 14 36    LDX INPPTR
3687 2A5D BD 00 82    LDA INPBUF,X
3688 2A60 EE 14 36    INC INPPTR
3689 2A63 FO 02      BEQ CMXINS
3690 2A65 18          CLC
3691 2A66 60          RTS
3692
3693 2A67 48          CMXINS: PHA ; INPUT NEXT SECTOR
3694 2A68 EE 16 36    INC INPSCT
3695 2A6B AD 16 36    LDA INPSCT
3696 2A6E C9 1B      CMP #26+1
3697 2A70 B0 06      BGE CMXINT
3698 2A72 20 AF 2C    JSR RDSCTR
3699 2A75 68          PLA
3700 2A76 18          CLC
3701 2A77 60          RTS
3702
3703 2A78 EE 15 36    CMXINT: INC ; INPUT NEXT TRACK
3704 2A7B A9 01      LDA #1
3705 2A7D 8D 16 36    STA INPSCT ; SET SECTOR # BACK TO 1
3706 2A80 20 AF 2C    JSR RDSCTR
3707 2A83 68          PLA
3708 2A84 18          CLC
3709 2A85 60          RTS
3710
3711 2A86 38          CMXEOF: SEC
3712 2A87 60          RTS
3713
3714 2A88 18          CMXOUT: CLC
3715 2A89 38          SEC
3716 2A8A 60          RTS
3717
3718
    
```

```
3720 .SBTTL N Device Drivers
3721
3722 ; N NUL
3723
3724 2A8B A9 00 N.INP: LDA #0 ; NULL
3725 2A8D 38 SEC ; SET END OF FILE
3726 2A8E 60 RTS
3727
3728 2A8F N.OPN:
3729 2A8F N.OUT:
3730 2A8F N.CLO:
3731 2A8F 18 CLC
3732 2A90 60 RTS
3733
3734
```

```

3736          .SBTTL C      Device Drivers
3737
3738          ; C          CONSOLE
3739
3740 2A91 20 E0 2B C.OPN: JSR CRLF          ; NEW LINE (FALL THROUGH)
3741 2A94 18 C.CLO: CLC          ; SET OKAY FLAG
3742 2A95 60 RTS
3743
3744 2A96 20 77 2B C.INP: JSR CTYIN          ; GET A CHAR FROM THE CONSOLE
3745 2A99 C9 1A CMP #'Z&$3F          ; ^Z EOF FROM CONSOLE
3746 2A9B FO 02 BEQ C.IEX          ;
3747 2A9D 18 CLC          ; NOT END OF FILE
3748 2A9E 60 RTS
3749 2A9F 38 C.IEX: SEC          ; SET END OF FILE CONDITION
3750 2AA0 60 RTS
3751
3752 2AA1 48 C.OUT: PHA          ; SAVE ACC
3753 2AA2 20 OF 10 JSR KEYDWN          ; SEE IF KEY DOWN
3754 2AA5 30 13 BMI C.NKDN
3755 2AA7 C9 03 CMP #'C&$3F          ; ^C ?
3756 2AA9 FO 22 BEQ CABORT          ; ABORT ON ^C
3757 2AAB C9 13 CMP #'S&$3F
3758 2AAD DO 0B BNE C.NKDN          ; IF NOT ^S IGNORE IT
3759 2AAF 20 OF 10 C.WCQ: JSR KEYDWN          ; LOOK FOR NEXT KEY
3760 2AB2 C9 03 CMP #'C&$3F          ; ^C ?
3761 2AB4 FO 17 BEQ CABORT
3762 2AB6 C9 11 CMP #'Q&$3F
3763 2AB8 DO F5 BNE C.WCQ          ; C. WAIT FOR CONTROL-Q
3764          ; C. NO KEY DOWN
3765 2ABA 68 C.NKDN: PLA          ; RESTORE ACC
3766 2ABB C9 0A CMP #$0A          ; IGNORE <LF>S
3767 2ABD FO 0C BEQ C.OEX
3768 2ABF C9 0D CMP #$0D
3769 2AC1 DO 05 BNE C.OUT1
3770 2AC3 20 E0 2B JSR CRLF          ; NEW LINE <CR> & <LF>
3771 2AC6 18 CLC
3772 2AC7 60 RTS
3773
3774 2AC8 20 95 2B C.OUT1: JSR CTYOUT
3775 2ACB 18 C.OEX: CLC
3776 2ACC 60 RTS
3777
3778 2ACD 68 CABORT: PLA          ; RESTORE BUT DISCARD
3779 2ACE 20 E0 2B JSR CRLF          ; NEW LINE
3780 2AD1 A9 5E LDA #'^          ; PRINT ^C
3781 2AD3 20 95 2B JSR CTYOUT
3782 2AD6 A9 43 LDA #'C
3783 2AD8 20 95 2B JSR CTYOUT
3784 2ADB 4C 03 10 JMP JWARM          ; RE-START PROGRAM
3785

```

```

3787          .SBTTL P      Device Drivers
3788
3789          ; P          Printer
3790
3791 2ADE EA          P.OPN: NOP          ; ALLOW FOR JSR
3792 2ADF EA          NOP
3793 2AEO EA          NOP
3794 2AE1 20 FF 2A    JSR          P.CRLF
3795 2AE4 18          CLC
3796 2AE5 60          RTS
3797
3798 2AE6 EA          P.CLO: NOP          ; ALLOW FOR JSR
3799 2AE7 EA          NOP
3800 2AE8 EA          NOP
3801 2AE9 18          CLC
3802 2AEA 60          RTS
3803
3804 2AEB 38          P.INP: SEC          ; SET EOF CONDITION
3805 2AEC 60          RTS
3806
3807 2AED          P.OUT:          ; USER DEFINABLE PRINTER DRIVERS
3808          ; DEFAULT IS PARRALLEL PRINTER
3809 2AED C9 OA          CMP          #$0A          ; LF ?
3810 2AEF FO 07          BEQ          P.OEX          ; IGNORE 1fs
3811 2AF1 C9 OD          CMP          #$0D          ; ? CR
3812 2AF3 DO 05          BNE          P.OCHR          ; PRINT IF NOT
3813 2AF5 20 FF 2A    JSR          P.CRLF
3814 2AF8 18          P.OEX: CLC
3815 2AF9 60          RTS
3816
3817 2AFA 20 12 10    P.OCHR: JSR          PRINTR          ; SEND LF TO PRINTER
3818 2AFD 18          CLC
3819 2AFE 60          RTS
3820
3821
3822 2AFF 20 12 10    P.CRLF: JSR          PRINTR          ; SEND CR TO PRINTER
3823 2B02 EA          NOP          ; IF DELAY IS NEEDED
3824 2B03 EA          NOP
3825 2B04 EA          NOP
3826 2B05 A9 OA          LDA          #$0A          ; LF AFTER CR
3827 2B07 20 12 10    JSR          PRINTR
3828 2BOA 60          RTS
3829

```

```

3831          .SBTTL M          Device Drivers
3832
3833          ; M          Memory
3834
3835 2B0B AD B9 35 M.OPN: LDA BEGMEM+0          ; ADDRESS OF BOTTOM OF MEMORY
3836 2B0E 85 00          STA MEMPTR+0
3837 2B10 AD BA 35          LDA BEGMEM+1
3838 2B13 85 01          STA MEMPTR+1
3839
3840 2B15 AD B9 35          LDA BEGMEM+0          ; CHECK IF ENDMEM AFTER BEGMEM
3841 2B18 CD BB 35          CMP ENDMEM+0
3842 2B1B AD BA 35          LDA BEGMEM+1
3843 2B1E ED BC 35          SBC ENDMEM+1
3844 2B21 B0 01          BCS M.OERR          ; SETS CARRY IF NOT OKAY
3845 2B23 60          RTS          ; OTHER WISE, CARRY CLEAR
3846
3847 2B24 A9 1E          M.OERR: LDA #MEMBER          ; MEMORY BUFFER ERROR
3848 2B26 20 82 2F          JSR ERROR
3849
3850 2B29 A0 00          M.CLO: LDY #0          ; INIT INDEX
3851 2B2B A9 1A          LDA #'Z&$3F
3852 2B2D 91 00          STA (MEMPTR),Y
3853 2B2F 18          CLC          ; OKAY FLAG
3854 2B30 60          RTS
3855
3856 2B31 A0 00          M.INP: LDY #0          ; INIT INDEX
3857 2B33 B1 00          LDA (MEMPTR),Y          ; GET DATA
3858 2B35 C9 1A          CMP #'Z&$3F          ; ^Z ($1A) EOF MARKER
3859 2B37 FO 05          BEQ M.IEX
3860 2B39 20 5F 2B          JSR M.INC
3861 2B3C 18          CLC
3862 2B3D 60          RTS
3863
3864 2B3E 38          M.IEX: SEC          ; SET EOF REACHED FLAG
3865 2B3F 60          RTS
3866
3867
3868 2B40 A0 00          M.OUT: LDY #0          ; OUTPUT TO MEMORY
3869 2B42 91 00          STA (MEMPTR),Y          ; INIT INDEX
3870 2B44 20 5F 2B          JSR M.INC
3871 2B47 A5 01          LDA MEMPTR+1
3872 2B49 CD BC 35          CMP ENDMEM+1          ; CHECK HIGH BYTE
3873 2B4C 90 07          BLT M.OCC          ; OKAY IF LESS THAN
3874 2B4E A5 00          LDA MEMPTR+0
3875 2B50 CD BB 35          CMP ENDMEM+0
3876 2B53 B0 02          BGE M.OFUL          ; OKAY IF LESS THAN
3877 2B55 18          M.OCC: CLC          ; SET OKAY FLAG
3878 2B56 60          RTS
3879
3880
3881 2B57 20 29 2B          M.OFUL: JSR M.CLO          ; MEMORY FULL ON OUTPUT
3882 2B5A A9 04          LDA #MEMFUL          ; CLOSE THE FILE ANYWAY
3883 2B5C 20 82 2F          JSR ERROR          ; MEMORY FULL ERROR
3884
3885 2B5F E6 00          M.INC: INC MEMPTR
    
```

3886	2B61	DO	O2	BNE	M.INCR
3887	2B63	E6	O1	INC	MEMPTR+1
3888	2B65	60		M.INCR: RTS	
3889					
3890					

```
3892 .SBTTL U Device Drivers
3893
3894
3895 2B66 4C 72 2B U.OPN: JMP U..ER ; USER OPEN FILE
3896 2B69 4C 72 2B U.INP: JMP U..ER ; USER READ FILE
3897 2B6C 4C 72 2B U.OUT: JMP U..ER ; USER WRITE FILE
3898 2B6F 4C 72 2B U.CLO: JMP U..ER ; USER CLOSE FILE
3899
3900
3901 2B72 A9 08 U..ER: LDA #USRERR ; USER ERROR
3902 2B74 20 82 2F JSR ERROR
3903
3904
3905
```

```

3907          .SBTTL  System I/O
3908
3909
3910          ; INPUT CHAR WITH ECHO IF NEEDED
3911 2B77  8E  DO  36  CTYIN:  STX   CTY.IX  ; SAVE X
3912 2B7A  8C  D1  36          STY   CTY.IY  ; SAVE Y
3913 2B7D  20  O6  10          JSR   SYSIN  ; GET THE CHAR
3914 2B80  8D  CF  36          STA   CTY.IA  ; SAVE IT
3915 2B83  AE  16  10          LDX   ECHO    ; CHECK ECHO FLAG
3916 2B86  FO  O3          BEQ   CNECHO
3917 2B88  20  OC  10          JSR   SYSOUT  ; PRINT IF NEEDED
3918 2B8B  AC  D1  36  CNECHO: LDY   CTY.IY  ; RESTORE Y
3919 2B8E  AE  DO  36          LDX   CTY.IX  ; RESTORE X
3920 2B91  AD  CF  36          LDA   CTY.IA  ; RESTORE ACC
3921 2B94  60          RTS          ; RTNS WITH (Z N) SET ACCORDINGLY
3922
3923          ; OUTPUT CHAR WITHOUT DESTROYING REGS
3924 2B95  8D  D2  36  CTYOUT: STA   CTY.OA  ; SAVE ACC
3925 2B98  8E  D3  36          STX   CTY.OX  ; SAVE X
3926 2B9B  8C  D4  36          STY   CTY.OY  ; SAVE Y
3927 2B9E  4C  OC  10          JMP   SYSOUT  ; PRINT THE CHAR
3928 2BA1  AC  D4  36          LDY   CTY.OY  ; RESTORE Y
3929 2BA4  AE  D3  36          LDX   CTY.OX  ; RESTORE X
3930 2BA7  AD  D2  36          LDA   CTY.OA  ; RESTORE ACC
3931 2BAA  60          RTS          ; RTNS WITH (Z N) SET ACCORDINGLY
3932
3933 2BAB  A2  32          PROMPT: LDX   #PRMPT^  ; FALL THROUGH TO PRINT STRING
3934 2BAD  AO  5D          LDY   #PRMPT&$FF
3935
3936          ; CAN ONLY PRINT STRINGS < 255 CHARS
3937 2BAF  86  O3          PRTSTR: STX   PRTPTR+1  ; SAVE ADDRESS IN PRINT POINTER
3938 2BB1  84  O2          STY   PRTPTR+0
3939 2BB3  20  E0  2B  PRTENT: JSR   CRLF    ; NEW LINE      (ALTERNAT ENTRY POINT)
3940 2BB6  AO  O0          LDY   #0        ; INIT INDEX POINTER
3941 2BB8  B1  O2          PRTST1: LDA   (PRTPTR),Y  ; GET CHAR
3942 2BBA  FO  O6          BEQ   PRTEND    ; EXIT ON NULL <$OO>
3943 2BBC  20  95  2B          JSR   CTYOUT
3944 2BBF  C8          INY          ; POINT TO NEXT CHAR
3945 2BC0  DO  F6          BNE   PRTST1   ; WILL ALWAYS BRANCH
3946 2BC2  60          PRTEND: RTS
3947
3948
3949          ; PRINT ACC AS 2 HEX CHARS
3950 2BC3  48          PRTBYT: PHA          ; SAVE BYTE TWICE
3951 2BC4  48          PHA
3952 2BC5  4A          LSR   A        ; GET HIGH NIBBLE
3953 2BC6  4A          LSR   A
3954 2BC7  4A          LSR   A
3955 2BC8  4A          LSR   A
3956 2BC9  20  D2  2B          JSR   HEXASC   ; PRINT 1st NIBBLE
3957 2BCC  68          PLA          ; RESTORE BYTE
3958 2BCD  20  D2  2B          JSR   HEXASC   ; PRINT 2nd NIBBLE
3959 2BD0  68          PLA          ; RESTORE BYTE
3960 2BD1  60          RTS          ; RTS  X = X  Y = Y  A = A
3961

```



```

4006
4007 2C1C 29 7F          PRVCHR: AND    #$7F          ; MASK OFF HIGH BIT
4008 2C1E C9 7F          CMP    #$7F
4009 2C20 F0 04          BEQ    PRVRUB        ; RUBOUT INVALID CHAR
4010 2C22 C9 20          CMP    #$20          ; <space>
4011 2C24 B0 02          BGE    PVCHR1
4012 2C26 A9 20          PRVRUB: LDA    #$20          ; <space>
4013 2C28 4C 95 2B      PVCHR1: JMP    CTYOUT        ; PRINT THE CHAR
4014
4015                      ; PRINT CHAR or PERIOD
4016 2C2B 29 7F          CHRPER: AND    #$7F          ; MASK OFF PARITY
4017 2C2D C9 7F          CMP    #$7F
4018 2C2F F0 04          BEQ    CHRRUB
4019 2C31 C9 20          CMP    #$20          ; <space>
4020 2C33 B0 02          BGE    CHRVAL
4021 2C35 A9 2E          CHRRUB: LDA    #'.
4022 2C37 4C 95 2B      CHRVAL: JMP    CTYOUT
4023
4024 2C3A 20 09 10        SPACE4: JSR    CTYSPA        ; PRINT FOUR SPACES
4025 2C3D 20 09 10        JSR    CTYSPA
4026 2C40 20 09 10        SPACE2: JSR    CTYSPA        ; PRINT TWO SPACES
4027 2C43 20 09 10        JSR    CTYSPA        ; RTS X = X Y = Y A = <space>
4028 2C46 60
4029
4030
    
```

```

4032          .SBTTL  Disk Drivers
4033
4034          ; by   Keith Sproul
4035
4036 2C47          CHKDRV:          ; CHECK DRIVE STATS
4037 2C47 8D 33 36          STA      TMPDRV          ; SAVE DRIVE #
4038 2C4A 20 41 2E          JSR      SDRVST          ; SENSE DRIVE STATUS
4039 2C4D A2 32          LDX      #DRVMSG^          ; HIGH BYTE
4040 2C4F A0 65          LDY      #DRVMSG&$FF          ; LOW BYTE
4041 2C51 20 AF 2B          JSR      PRTSTR
4042 2C54 AD 33 36          LDA      TMPDRV          ; RESTORE DRIVE #
4043 2C57 09 30          ORA      #$30          ; MAKE INTO ASCII NUMBER
4044 2C59 20 95 2B          JSR      CTYOUT
4045 2C5C 20 09 10          JSR      CTYSPA          ; PRINT A SPACE
4046 2C5F AD 00 36          LDA      DSKSTS+0          ; GET ST-3
4047 2C62 8D 4B 36          STA      ACC          ; SAVE IT IN ACC FOR DOING SHIFTING
4048 2C65 20 C3 2B          JSR      PRTBYT          ; PRINT IT
4049 2C68 20 40 2C          JSR      SPACE2          ; PRINT 2 SPACES
4050 2C6B A2 46          LDX      #'F          ; INDICATE DRIVE FALUT
4051 2C6D 20 A0 2C          JSR      CHKSUB
4052 2C70 A2 57          LDX      #'W          ; INDICATE DRIVE WRITE PROTECTED
4053 2C72 20 A0 2C          JSR      CHKSUB
4054 2C75 A2 52          LDX      #'R          ; INDICATE DRIVE READY
4055 2C77 20 A0 2C          JSR      CHKSUB
4056 2C7A A2 30          LDX      #'O          ; INDICATE DRIVE @ TRACK ZERO
4057 2C7C 20 A0 2C          JSR      CHKSUB
4058 2C7F A2 32          LDX      #'2          ; INDICATE IF DOUBLE SIDED
4059 2C81 20 A0 2C          JSR      CHKSUB
4060 2C84 20 40 2C          JSR      SPACE2          ; PRINT 2 MORE SPACES
4061 2C87 AE 33 36          LDX      TMPDRV          ; GET DRIVE #
4062 2C8A BD C4 35          LDA      DRIVES,X
4063 2C8D OA          ASL      A          ; * 2
4064 2C8E OA          ASL      A          ; * 4 (3 CHARS & A ZERO)
4065 2C8F AA          TAX          ; USE AS INDEX
4066 2C90 A0 03          LDY      #3
4067 2C92 BD FB 1A          CHKDLP: LDA      DSKTAB,X
4068 2C95 20 95 2B          JSR      CTYOUT
4069 2C98 E8          INX
4070 2C99 88          DEY
4071 2C9A DO F6          BNE      CHKDLP
4072 2C9C AD 00 36          LDA      DSKSTS+0          ; RETURN WITH DISK STATUS IN ACC
4073 2C9F 60          RTS
4074          ; CHECK DRIVE SUBROUTINE
4075 2CA0 A9 20          CHKSUB: LDA      #$20          ; <space>
4076 2CA2 OE 4B 36          ASL      ACC
4077 2CA5 90 01          BCC      CHKS1
4078 2CA7 8A          TXA          ; IF CONDITION NOT MET THEN PRINT <space>
4079 2CA8 20 95 2B          CHKS1: JSR      CTYOUT          ; OTHERWISE, PUT CHAR INTO ACC FROM %X
4080 2CAB 20 09 10          JSR      CTYSPA          ; <space>
4081 2CAE 60          RTS
4082

```

```

4084          .SBTTL  Disk Drivers - Read Sector Routine
4085
4086          ; ENTER WITH:
4087          ;   xxSCTK =      SECTOR/TRK FLAG
4088          ;   xxPAGE =      DMA PAGE
4089          ;   xxxDEN =      DENSITY 1 = SINGLE
4090          ;   ;           2 = DOUBLE
4091          ;   xxxDRV =      DRIVE #
4092          ;   xxxTRK =      TRACK
4093          ;   xxxSCT =      SECTOR
4094
4095          ; FLOPPY DISK READ SECTOR
4096          ; THIS ROUTINE READS ONE SECTOR FROM DISK INTO MEMORY.
4097          ; ***CAUTION***
4098          ; CURRENT CYLINDER NUMBER IS TAKEN FROM SEEKCM+3, IF IT DOES
4099          ; NOT MATCH THE ACTUAL HEAD POSITION, A READ ERROR WILL RESULT
4100          ; *****
4101          ; RETURNS WITH CARRY CLEAR IF NO READ ERROR.
4102          ; THE STATUS AREA IS UPDATED AFTER A READ, 7 BYTES ARE STORED.
4103
4104          ; NOTE:  USES DMA MODE BUT WAITS FOR OPERATION TO COMPLETE BEFORE
4105          ; RETURNING.
4106          ; A READ DATA COMMAND IS USED AND THE SK BIT IS ZERO.  THIS
4107          ; MEANS THAT DELETED SECTORS WILL BE READ AND BIT 6 OF STATUS 2
4108          ; WILL BE SET ON RETURN.
4109
4110          ; READ SECTOR ROUTINE.
4111 2CAF AD 13 36 RDSCTR: LDA INPDRV ; GET READ DRIVE #
4112 2CB2 AE 15 36 LDX INPTRK ; GET TRACK #
4113 2CB5 20 88 2E JSR SEEK ; SEEK TRACK X
4114 2CB8 AD 1B 36 LDA INPAGE ; GET DMA PAGE #
4115 2CBB 20 29 2F JSR DMASET
4116 2CBE 8D 49 36 STA DMATMP ; USED IF A RE-TRY IS DONE
4117 2CC1 AD 19 36 LDA INBYSC ; GET SCTR/TRK FLAG
4118 2CC4 29 03 AND #$03 ; JUST IN CASE
4119 2CC6 8D 74 2F STA READCM+6 ; PUT INTO 'N' FIELD
4120 2CC9 AD 13 36 LDA INPDRV ; GET CURRENT DRIVE #
4121 2CCC 29 07 AND #$07 ; JUST IN CASE
4122 2CCE 8D 70 2F STA READCM+2 ; PUT DRIVE/SIDE IN SECOND BYTE OF COMMAND
4123 2CD1 4A LSR A ; ISOLATE HEAD # AND PUT INTO H FIELD
4124 2CD2 4A LSR A
4125 2CD3 8D 72 2F STA READCM+4 ; H
4126 2CD6 AD 16 36 LDA INPST ; GET SECTOR NUMBER
4127 2CD9 8D 73 2F STA READCM+5 ; SET AS SECTOR TO READ
4128 2CDC 8D 75 2F STA READCM+7 ; SET ALSO AS LAST SECTOR TO READ
4129 2CDF AD 1A 36 LDA INPDEN ; GET DENSITY FLAG
4130 2CE2 20 29 2D JSR READ ; READ IT
4131 2CE5 B0 01 BCS RD.ERR
4132 2CE7 60 RTS
4133
4134 2CE8 A9 0D RD.ERR: LDA #RDSERR ; READ SECTOR ERROR
4135 2CEA 20 82 2F JSR ERROR
4136

```



```

4176
4177           ; ALTERNATE ENTRY POINT           (SET-UP REQUIRED)
4178
4179           ; SBR USED BY RDTEMP & RDSCTR
4180 2D29 C9 O1 READ:  CMP #1           ; SINGLE DENSITY?
4181 2D2B FO O9          BEQ RSDEN
4182 2D2D C9 O2          CMP #2           ; DOUBLE DENSITY?
4183 2D2F FO O9          BEQ RDDDEN
4184 2D31 A9 OA          LDA #DENERR      ; IF NEITHER THEN INTERNAL DENSITY ERROR!
4185 2D33 20 82 2F      JSR ERROR
4186
4187 2D36 A9 O6          RSDEN: LDA #$06      ; READ S-DEN CMD
4188 2D38 DO O2          BNE RDXDEN      ; WILL ALWAYS BRANCH
4189 2D3A A9 46          RDDDEN: LDA #$46      ; READ D-DEN CMD
4190 2D3C 8D 6F 2F      RDXDEN: STA READCM+1 ; PUT INTO CMD
4191 2D3F A9 O5          LDA #5           ; INIT # OF RETRYS TO FOUR
4192 2D41 8D 48 36      STA RETRYS
4193           ; RE-ENTER HERE ON RETRYS
4194 2D44 AD 6A 2F      RDRTRY: LDA SEEKCM+3 ; ESTABLISH CYLINDER NUMBER
4195 2D47 8D 71 2F      STA READCM+3 ; C
4196 2D4A A9 O1          LDA #$01      ; DMA DISK => MEM
4197 2D4C 8D E8 DF      STA FDCHWC ; SET UP DMA DIRECTION
4198 2D4F A2 10          LDX #READCM-DSKCMD ; RELATIVE ADDRESS OF READ COMMAND
4199 2D51 20 BF 2E      JSR COMAND ; SEND COMMAND TO DISK CONTROLLER
4200 2D54 AD E8 DF      READWT: LDA FDCIRQ ; WAIT UNTIL INTERRUPT REQUEST FROM FDC
4201 2D57 30 FB          BMI READWT
4202 2D59 20 E4 2E      JSR RESULT ; SET UP TO READ STATUS INTO STATUS AREA
4203 2D5C 20 O7 2F      JSR RERROR ; TEST IF ANY ERRORS
4204 2D5F BO O1          BCS READER ; RTS C = 0 IF OKAY
4205 2D61 60
4206           RTS
4207 2D62 AD 49 36      READER: LDA DMATMP ; RE-SET THE DMA PAGE REGISTER
4208 2D65 8D EA DF      STA FDGDMA
4209 2D68 CE 48 36      DEC RETRYS ; BUMP UP COUNTER
4210 2D6B DO D7          BNE RDRTRY
4211 2D6D 38             SEC ; SET ERROR CONDITION
4212 2D6E 60             RTS
4213
4214           ; BRANCHED TO FROM ABOVE
4215
4216

```

```

4218          .SBTTL  Disk Drivers - Write Sector Routine
4219
4220          ;      FLOPPY DISK WRITE SECTOR
4221          ;      THIS ROUTINE WRITES ONE SECTOR FROM MEMORY TO DISK.
4222          ;      ENTER WITH MEMORY PAGE NUMBER TO WRITE FROM IN Y (MUST BE
4223          ;      IN DISK CONTROLLER RAM), SECTOR NUMBER TO WRITE TO IN X
4224          ;      WRITES ON CURRENT TRACK), AND DISK DRIVE NUMBER AND DISKETTE
4225          ;      SIDE IN A.
4226          ;      ***CAUTION***
4227          ;      CURRENT CYLINDER NUMBER IS TAKEN FROM SEEKCM+3, IF IT DOES
4228          ;      NOT MATCH THE ACTUAL HEAD POSITION, A WRITE ERROR WILL RESULT
4229          ;      *****
4230          ;      RETURNS WITH CARRY CLEAR IF NO WRITE ERROR.
4231          ;      THE STATUS AREA IS UPDATED AFTER A WRITE, 7 BYTES ARE STORED.
4232
4233          ;      NOTE:  USES DMA MODE BUT WAITS FOR OPERATION TO COMPLETE BEFORE
4234          ;      RETURNING.
4235
4236
4237 2D6F  AD  23  36  WRSCTR: LDA    OUTDRV      ; GET WRITE DRIVE #
4238 2D72  AE  25  36      LDX    OUTTRK     ; GET TRACK #
4239 2D75  20  88  2E      JSR    SEEK       ; SEEK TRACK X
4240 2D78  AD  2B  36      LDA    OUPAGE     ; GET DMA PAGE #
4241 2D7B  20  29  2F      JSR    DMASET
4242 2D7E  AD  29  36      LDA    OUBYSC     ; GET SCTR/TRK FLAG
4243 2D81  29  03          AND    #$03       ; JUST IN CASE
4244 2D83  8D  7E  2F      STA    WRITCM+6   ; PUT INTO 'N' FIELD
4245 2D86  AD  23  36      LDA    OUTDRV     ; GET CURRENT DRIVE #
4246 2D89  29  07          AND    #$07       ; JUST IN CASE
4247 2D8B  8D  7A  2F      STA    WRITCM+2   ; PUT DRIVE/SIDE IN SECOND BYTE OF COMMAND
4248 2D8E  4A          LSR    A          ; ISOLATE HEAD # AND PUT INTO H FIELD
4249 2D8F  4A          LSR    A
4250 2D90  8D  7C  2F      STA    WRITCM+4   ; H
4251 2D93  AD  26  36      LDA    OUTSCT     ; GET SECTOR NUMBER
4252 2D96  8D  7D  2F      STA    WRITCM+5   ; SET AS SECTOR TO WRIT
4253 2D99  8D  7F  2F      STA    WRITCM+7   ; SET AS LAST SECTOR TO WRITE ALSO
4254 2D9C  AD  2A  36      LDA    OUTDEN     ; GET DENSITY FLAG
4255 2D9F  20  E6  2D      JSR    WRITE      ; WRITE IT
4256 2DA2  BO  01          BCS    WR.ERR     ; ERROR!
4257 2DA4  60          RTS
4258
4259
4260 2DA5  A9  0E          WR.ERR: LDA    #WRSERR ; WRITE SECTOR ERROR
4261 2DA7  20  82  2F      JSR    ERROR
4262
4263

```

```

4265
4266 ; WRITE SECTOR ROUTINE.
4267 ; USING THE TEMP VARIABLES
4268 ; USED IN OPEN ROUTINES
4269 ; Can also be set up to WRITE
4270 ; a TRACK or part of a TRACK.
4271 2DAA 20 B0 2D WRTEMP: JSR WRRTRN ; WRITE THE SECTOR
4272 2DAD B0 F6 BCS WR.ERR ; BRANCH IF ERROR
4273 2DAF 60 RTS
4274
4275
4276
4277 ; WRITE & RETURN WITH ERROR COND
4278 2DB0 AD 33 36 WRRTRN: LDA TMPDRV ; GET WRITE DRIVE #
4279 2DB3 AE 35 36 LDX TMPTRK ; GET TRACK #
4280 2DB6 20 88 2E JSR SEEK ; SEEK TRACK X
4281 2DB9 AD 3B 36 LDA TMPAGE ; GET DMA PAGE #
4282 2DBC 20 29 2F JSR DMASET
4283 2DBF 8D 49 36 STA DMATMP ; USED IF A RE-TRY IS DONE
4284 2DC2 AD 39 36 LDA TMBYSC ; GET SCTR/TRK FLAG
4285 2DC5 29 03 AND #$03 ; JUST IN CASE
4286 2DC7 8D 7E 2F STA WRITCM+6 ; PUT INTO 'N' FIELD
4287 2DCA AD 33 36 LDA TMPDRV ; GET CURRENT DRIVE #
4288 2DCD 29 07 AND #$07 ; JUST IN CASE
4289 2DCF 8D 7A 2F STA WRITCM+2 ; PUT DRIVE/SIDE IN SECOND BYTE OF COMMAND
4290 2DD2 4A LSR A ; ISOLATE HEAD # AND PUT INTO H FIELD
4291 2DD3 4A LSR A
4292 2DD4 8D 7C 2F STA WRITCM+4 ; H
4293 2DD7 AD 36 36 LDA TMPSTC ; GET SECTOR NUMBER
4294 2DDA 8D 7D 2F STA WRITCM+5 ; SET AS SECTOR TO READ
4295 2DDD AD 38 36 LDA TMLSCT ; GET TEMP LAST SECTOR
4296 2DE0 8D 7F 2F STA WRITCM+7 ; SET AS LAST SECTOR TO READ
4297 2DE3 AD 3A 36 LDA TMPDEN ; GET DENSITY FLAG
4298 ; FALL THROUGH TO WRITE
4299

```



```

4301
4302           ; ALTERNATE ENTRY POINT           (SET-UP REQUIRED)
4303
4304           ; SBR USED BY WRTEMP & WRSCTR
4305 2DE6 C9 01 WRITE: CMP #1 ; SINGLE DENSITY?
4306 2DE8 FO 09 BEQ WRSDEN
4307 2DEA C9 02 CMP #2 ; DOUBLE DENSITY?
4308 2DEC FO 09 BEQ WRDDEN
4309 2DEE A9 0A LDA #DENERR ; IF NEITHER THEN INTERNAL DENSITY ERROR!
4310 2DFO 20 82 2F JSR ERROR
4311
4312 2DF3 A9 05 WRSDEN: LDA #$05 ; WRITE S-DEN CMD
4313 2DF5 DO 02 BNE WRXDEN ; WILL ALWAYS BRANCH
4314 2DF7 A9 45 WRDDEN: LDA #$45 ; WRUTE D-DEN CMD
4315 2DF9 8D 79 2F WRXDEN: STA WRITCM+1 ; PUT INTO CMD
4316 2DFC A9 03 LDA #3 ; ALLOW FOR TWO RETRY
4317 2DFE 8D 48 36 STA RETRYS
4318           ; RE-ENTER HERE ON RETRYS
4319 2E01 AD 6A 2F WRRTRY: LDA SEEKCM+3 ; ESTABLISH CYLINDER NUMBER
4320 2E04 8D 7B 2F STA WRITCM+3 ; C
4321 2E07 A9 00 LDA #$00 ; DMA MEM => DISK
4322 2E09 8D E8 DF STA FDCHWC ; SET UP DMA DIRECTION & ALLOW SYSRAM WRITE
4323 2E0C A2 1A LDX #WRITCM-DSKCMD ; RELATIVE ADDRESS OF WRITE COMMAND
4324 2E0E 20 BF 2E JSR COMAND ; SEND COMMAND TO DISK CONTROLLER
4325 2E11 AD E8 DF WRITWT: LDA FDCIRQ ; WAIT UNTIL INTERRUPT REQUEST FROM FDC
4326 2E14 30 FB BMI WRITWT
4327 2E16 20 E4 2E JSR RESULT ; SET UP TO WRIT STATUS INTO STATUS AREA
4328 2E19 20 07 2F JSR RERROR ; RTNS WITH C = 0 IF OKAY
4329 2E1C B0 01 BCS WRITER
4330 2E1E 60 RTS
4331
4332 2E1F AD 49 36 WRITER: LDA DMATMP ; RE-SET THE DMA PAGE REGISTER
4333 2E22 8D EA DF STA FDCDMA
4334 2E25 CE 48 36 DEC RETRYS
4335 2E28 DO D7 BNE WRRTRY
4336 2E2A 38 SEC ; SET ERROR CONDITION
4337 2E2B 60 RTS
4338
4339
4340
4341
4342 .SBTTL Disk Drivers - Subroutines
4343
4344 ; .READ DISKSBR.A
4345 .MCALL DISKSR ; DISK SUB-ROUTINES
4346 2E2C DISKSR
(1)
(1) ; SUBROUTINES THAT DON'T CHANGE.
(1) ; THIS INCLUDES THE FOLLOWING SUBROUTINES:
(1)
(1) ; DSKINI INIT & RESET IF NEEDED THE FDC
(1) ; SDRVST SENSE DRIVE STATUS
(1) ; RECAL RECALIBRATE DRIVE IN A
(1) ; SEEK SEEK DRIVE A TO TRACK X
(1) ; SPECIFY SET DISK PARAMETERS

```

```

(1) ; SINTST SENSE INTERRUPT STATUS
(1) ; COMAND COMMAND PHASE OF FDC
(1) ; RESULT RESULT PHASE OF FDC
(1) ; RERROR SENSE IF ANY ERRORS IN RESULT PHASE
(1) ; DMASET SET DMA REG ACORDING TO PAGE # IN ACC
(1)
(1) 2E2C ; DSKINI: ; INIT THE FDC & SYSTEM
(1) 2E2C AD EE DF LDA FDCMSR ; CHECK MAIN STATUS REGISTRE
(1) 2E2F 10 FB BPL DSKINI ; WAIT UNTIL REQUEST FOR MASTER GOES TRUE
(1) 2E31 29 10 AND #$10 ; CHECK DIO
(1) 2E33 FO 03 BEQ .+5 ; IF NOT WANTING TO TALK
(1) 2E35 20 E4 2E JSR RESULT ; ELSE LISTEN
(1) 2E38 20 4C 2E JSR SPECIFY
(1) 2E3B A9 01 LDA #$01
(1) 2E3D 8D E8 DF STA FDCHWC ; TURN WRITE PROTECT OFF
(1) 2E40 60 RTS ; & DATA DIR = READ FROM DISK
(1)
(1) ; SENSE DRIVE STATUS
(1) 2E41 8D 6D 2F SDRVST: STA SDSTCM+2 ; PUT DRIVE # IN SENSE DRV STAT CMD 2ND BYTE
(1) 2E44 A2 OD LDX #SDSTCM-DSKCMD ; SENSE DRIVE STATUS CMD OFFSET
(1) 2E46 20 BF 2E JSR COMAND
(1) 2E49 4C E4 2E JMP RESULT ; GET RESULT
(1) ; & RETURN
(1)
(1)
(1) ; .PAGE
(1)
(1) ; FLOPPY DISK RESET AND SPECIFY. THIS ROUTINE FIRST RESETS THE
(1) ; FLOPPY DISK CONTROLLER AND THEN ESTABLISHES THE SEEK RATE, HEAD
(1) ; LOAD TIME, HEAD UNLOAD TIME, AND DMA OPTION BY USING A SPECIFY
(1) ; COMMAND. SPECIFY DATA MUST BE PREFORMATTED IN MEMORY AT SPECCM.
(1)
(1)
(1) 2E4C AD E8 DF SPECIFY: LDA FDCIRQ ; TEST IF AN INTERRUPT IS PENDING
(1) 2E4F 30 03 BMI SPECF1 ; JUMP AHEAD IF NOT
(1) 2E51 20 59 2E JSR SINTST ; EXECUTE SENSE INTERRUPT STATUS COMMAND
(1)
(1) 2E54 A2 00 SPECF1: LDX #SPECCM-DSKCMD ; SET ADDRESS OF SPECIFY COMMAND
(1) 2E56 4C BF 2E JMP COMAND ; EXECUTE THE SPECIFY COMMAND
(1) ; & RETURN
(1)
(1)
(1) 2E59 SINTST:
(1) 2E59 A2 04 LDX #SNSICM-DSKCMD ; OFF SET FOR SENSE INTERRUPT STATUS CMD
(1) 2E5B 20 BF 2E JSR COMAND ; EXECUTE THE CMD
(1) 2E5E 4C E4 2E JMP RESULT ; GET THE RESULTS
(1) ; & RETURN
(1)
(1)
(1) ; .PAGE
(1)
(1) ; FLOPPY DISK RECALIBRATE
(1) ; THIS ROUTINE SEEKS THE DRIVE NUMBER LOADED INTO A TO TRACK ZERO
(1) ; REGARDLESS OF WHERE IT CURRENTLY IS.
(1)
    
```

```

(1) ; THE STATUS BYTES AT DSKSTS ARE MODIFIED AS A RESULT OF THE
(1) ; RECALIBRATE OPERATION.
(1) ;
(1) ; RECALIBRATION IS PERFORMED BY ISSUEING A RECALIBRATE COMMAND,
(1) ; WAITING FOR INTERRUPT REQUEST TO GO TRUE, AND PERFORMING A
(1) ; SENSE INTERRUPT STATUS COMMAND.
(1)
(1) 2E61 29 03 RECAL: AND #$03 ; PUT DRIVE NUMBER INTO SECOND BYTE OF
(1) 2E63 8D 66 2F STA RECLCM+2 ; RECALIBRATE COMMAND
(1) 2E66 A2 06 LDX #RECLCM-DSKCMD ; EXECUTE THE RECALIBRATE COMMAND
(1) 2E68 20 BF 2E JSR COMAND
(1) 2E6B AD E8 DF RECAL1: LDA FDCIRQ ; READ DISK CONTROLLER INTERRUPT REQUEST
(1) ; (FDC IRQ JUMPER REMOVED OR IRQ DISABLED)
(1) 2E6E 30 FB BMI RECAL1 ; WAIT UNTIL THE FDC REQUESTS AN INTERRUPT
(1) 2E70 20 59 2E JSR SINTST ; EXECUTE SENSE INTERRUPT STATUS COMMAND
(1)
(1) 2E73 AD 01 36 LDA DSKSTS+1 ; LOOK AT PRESENT CYLINDER NUMBER
(1) 2E76 DO 0B BNE RECALE ; ERROR IF NON-ZERO
(1) 2E78 AD 00 36 LDA DSKSTS+0 ; LOOK AT ST-0 STATUS REGISTER
(1) 2E7B 29 F8 AND #$F8 ; DELETE DON'T CARE BITS
(1) 2E7D C9 20 CMP #$20 ; REMAINING BITS MUST BE $20
(1) 2E7F DO 02 BNE RECALE ; GO TO ERROR RETURN
(1) 2E81 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
(1) 2E82 60 RTS
(1)
(1) 2E83 A9 06 RECALE: LDA #RCLERR ; SET RECALIBRATE ERROR
(1) 2E85 20 82 2F JSR ERROR
(1)
(1) ;.PAGE
(1)
(1) ; FLOPPY DISK SEEK
(1) ; THIS ROUTINE SEEKS THE DRIVE (A) TO TRACK (X)
(1) ; THE STATUS BYTES AT DSKSTS ARE MODIFIED AS A RESULT OF THE
(1) ; SEEK OPERATION.
(1) ; SEEK IS PERFORMED BY ISSUEING A SEEK COMMAND,
(1) ; WAITING FOR INTERRUPT REQUEST TO GO TRUE, AND PERFORMING A
(1) ; SENSE INTERRUPT STATUS COMMAND.
(1)
(1) ; A = DRIVE
(1) ; X = TRACK
(1)
(1) 2E88 29 03 SEEK: AND #$03 ; FORMAT DRIVE NUMBER INTO SECOND BYTE OF
(1) 2E8A 8D 69 2F STA SEEKCM+2 ; SEEK COMMAND AND SELECT HEAD 0 2ND BYTE
(1) 2E8D 8E 6A 2F STX SEEKCM+3 ; STORE NEW CYLINDER NUMBER INTO 3RD BYTE
(1) 2E90 8A TXA ; SAVE NEW CYLINDER NUMBER ON THE STACK
(1) 2E91 48 PHA
(1) 2E92 A2 09 LDX #SEEKCM-DSKCMD ; EXECUTE THE SEEK COMMAND
(1) 2E94 20 BF 2E JSR COMAND
(1)
(1) 2E97 AD E8 DF SEEK1: LDA FDCIRQ ; READ DISK CONTROLLER INTERRUPT REQUEST
(1) 2E9A 30 FB BMI SEEK1 ; WAIT UNTIL THE FDC REQUESTS AN INTERRUPT
(1) 2E9C 20 59 2E JSR SINTST ; EXECUTE SENSE INTERRUPT STATUS COMMAND
(1)
(1) 2E9F 68 PLA ; COMPARE PRESENT CYLINDER NUMBER WITH
(1) 2EAO CD 01 36 CMP DSKSTS+1 ; DESIRED CYLINDER NUMBER

```

```

(1) 2EA3 DO 09 BNE SEEKER ; ERROR IF NOT THE SAME
(1) 2EA5 AD 00 36 LDA DSKSTS+0 ; LOOK AT ST-0 STATUS REGISTER
(1) 2EA8 29 F8 AND #$F8 ; DELETE DON'T CARE BITS
(1) 2EAA C9 20 CMP #$20 ; REMAINING BITS MUST BE $20
(1) 2EAC FO OF BEQ SEEKOK ; GO TO ERROR RETURN
(1)
(1) 2EAE AD 00 36 SEEKER: LDA DSKSTS+0 ; GET ST-0
(1) 2EB1 29 03 AND #$03 ; CHECK IF CURRENT DRIVE AT FAULT
(1) 2EB3 CD 69 2F CMP SEEKCM+2
(1) 2EB6 DO 05 BNE SEEKOK ; IF NOT, THEN ASSUME OKAY
(1) ; ELSE FALL THROUGH
(1) 2EB8 A9 05 LDA #SEKERR ; DISK DRIVE SEEK ERROR
(1) 2EBA 20 82 2F JSR ERROR
(1)
(1) 2EBD 18 SEEKOK: CLC ; CLEAR CARRY FOR NORMAL RETURN
(1) 2EBE 60 RTS
(1)
(1) ; .PAGE
(1) ; SEND COMMAND TO FLOPPY DISK CONTROLLER
(1) ; ENTER WITH RELATIVE ADDRESS OF COMMAND BYTES IN X (RELATIVE TO
(1) ; DSKCMD) AND NUMBER OF BYTES IN COMMAND IN Y.
(1) ; ROUTINE SENDS THE BYTES TO THE FLOPPY DISK CONTROLLER AND
(1) ; RETURNS WITH THE CARRY FLAG OFF.
(1) ; IF AN ERROR IS DETECTED, THE CARRY FLAG IS ON.
(1) ; IF THE CONTROLLER IS BUSY, THE ERROR RETURN IS TAKEN
(1)
(1) 2EBF COMAND:
(1) 2EBF AD EE DF LDA FDCMSR ; LOOK AT MAIN STATUS REGISTER
(1) 2EC2 29 10 AND #$10 ; LOOK AT ALL OF THE BUSY BITS
(1) 2EC4 DO 19 BNE CMDPHE ; ERROR IF EXECUTING PREVIOUS
(1) ; COMMAND (EXCEPT SEEK)
(1) 2EC6 BC 5E 2F LDY DSKCMD,X ; GET NUMBER OF BYTES IN CMD <KS>
(1) 2EC9 E8 INX ; POINT TO NEXT ENTRY IN CMD TABLE <KS>
(1) 2ECA AD EE DF CMDPH1: LDA FDCMSR ; LOOK AT MAIN STATUS REGISTER
(1) 2ECD 10 FB BPL CMDPH1 ; WAIT UNTIL REQUEST FOR MASTER GOES TRUE
(1) 2ECF 29 40 AND #$40 ; TEST DATA DIRECTION BIT
(1) 2ED1 DO 0C BNE CMDPHE ; ERROR IF FDC WANTS TO TALK
(1) 2ED3 BD 5E 2F LDA DSKCMD,X ; GET A COMMAND BYTE
(1) 2ED6 8D EF DF STA FDCDR ; STORE IT IN THE DISK CONTROLLER
(1) 2ED9 E8 INX ; POINT S TO NEXT COMMAND BYTE
(1) 2EDA 88 DEY ; DECREMENT COMMAND BYTE COUNT
(1) 2EDB DO ED BNE CMDPH1 ; GO TRANSFER NEXT BYTE IF NOT DONE
(1) 2EDD 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
(1) 2EDE 60 RTS
(1)
(1) 2EDF A9 OF CMDPHE: LDA #FCMDER ; FDC COMMAND ERROR
(1) 2EE1 20 82 2F JSR ERROR
(1)
(1)
(1) ; .PAGE
(1) ; RECEIVE STATUS FROM FLOPPY DISK CONTROLLER
(1) ; STATUS BYTES ARE STORED SEQUENTIALLY IN MEMORY STARTING AT
(1) ; DSKSTS, THE NUMBER READ IS DETERMINED BY THE FDC BUSY STATUS
(1)

```

```

(1) ; ROUTINE READS THE STATUS BYTES IN THE RESULT PHASE AND RETURNS
(1) ; WITH THE CARRY FLAG OFF.
(1)
(1)
(1) 2EE4 A2 00 RESULT: LDX #0 ; INIT INDEX POINTER
(1) 2EE6 AD EE DF RESULT1: LDA FDCMSR ; LOOK AT MAIN STATUS REGISTER
(1) 2EE9 10 FB BPL RESULT1 ; WAIT UNTIL REQUEST FOR MASTER GOES TRUE
(1) 2EEB 29 40 AND #$40 ; TEST DATA DIRECTION BIT (DIO)
(1) 2EED FO 13 BEQ RSLPHE ; ERROR IF FDC WANTS TO LISTEN
(1) 2EEF AD EF DF LDA FDCDR ; GET A STATUS BYTE FROM THE DATA REGISTER
(1) 2EF2 9D 00 36 STA DSKSTS,X ; PUT IF INTO MEMORY
(1) 2EF5 E8 INX ; POINT X TO NEXT STATUS BYTE
(1) 2EF6 EA NOP ; DAMN SLOW CONTROLLER CHIP! 12US RESPONSE
(1) 2EF7 EA NOP ; TIME FROM READ TO VALID BUSY STATUS
(1) 2EF8 EA NOP
(1) 2EF9 A9 10 LDA #$10 ; LOOK AT BUSY BIT IN MAIN STATUS REGISTER
(1) 2EFB 2D EE DF AND FDCMSR
(1) 2EFE DO E6 BNE RESULT1 ; GO FOR ANOTHER STATUS BYTE IF STILL BUSY
(1) 2F00 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
(1) 2F01 60 RTS
(1)
(1) 2F02 A9 10 RSLPHE: LDA #FRSLER ; FDC RESULT ERROR
(1) 2F04 20 82 2F JSR ERROR
(1)
(1) ; .PAGE
(1)
(1) 2F07 AD 00 36 RERROR: LDA DSKSTS+0 ; CHECK STATUS REGISTER 0
(1) 2FOA 29 D8 AND #$D8 ; MASK OUT NON-ERROR BITS
(1) 2FOC FO 09 BEQ RERR1 ; JUMP AHEAD IF NO OBVIOUS ERROR
(1) 2FOE C9 40 CMP #$40 ; TEST IF ABNORMAL TERMINATION ERROR
(1) 2F10 DO 15 BNE R..ERR ; TRUE ERROR IF NOT
(1) 2F12 AD 01 36 LDA DSKSTS+1 ; IF ABNORMAL, TEST IF END OF CYLINDER
(1) ; <KS> CHECK FOR $04
(1) 2F15 10 10 BPL R..ERR ; TRUE ERROR IF NOT, OK IF SO
(1) 2F17 AD 01 36 RERR1: LDA DSKSTS+1 ; CHECK STATUS REGISTER 1
(1) 2F1A 29 35 AND #$35 ; MASK OUT NON-ERROR BITS
(1) 2F1C DO 09 BNE R..ERR ; GO TO ERROR IF ANY OF REMAINDER SET
(1) 2F1E AD 02 36 LDA DSKSTS+2 ; CHECK STATUS REGISTER 2
(1) 2F21 29 33 AND #$33 ; MASK OUT NON-ERROR BITS
(1) 2F23 DO 02 BNE R..ERR ; GO TO ERROR IF ANY OF REMAINDER SET
(1) 2F25 18 CLC ; CLEAR CARRY FOR NORMAL RETURN
(1) 2F26 60 RTS
(1)
(1) 2F27 38 R..ERR: SEC
(1) 2F28 60 RTS
(1)
(1) ; .PAGE
(1) ;
(1) ; SUBROUTINE TO ACCEPT THE PAGE ADDRESS OF A MEMORY BUFFER AND
(1) ; SET THE DMA ADDRESS REGISTER WITH THE APPROPRIATE VALUE. THE
(1) ; BUFFER ADDRESS MUST BE IN THE DISK CONTROLLER RAM.
(1) ; RETURN WITH CARRY CLEAR IF ADDRESS IS OK, SET IF ADDRESS IS
(1) ; INVALID.
(1) ; FOR THIS ROUTINE TO WORK, THE ORIGIN OF THE USER RAM MUST BE
(1) ; EQUATED TO USRRAM AND THE ORIGIN OF THE SYSTEM RAM MUST BE
(1) ; EQUATED TO SYSRAM.
(1) ;

```

```

(1) ; ENTER WITH BUFFER PAGE ADDRESS IN A, EXIT WITH DMA ADDRESS IN A
(1) ; AND THE DMA ADDRESS REGISTER.
(1)
(1) 2F29 C9 80 DMASET: CMP #USRRAM/256 ; COMPARE WITH BEGINNING OF USER RAM
(1) 2F2B 90 OE BCC DMAST1 ; JMP IF LESS THAN USER RAM
(1) 2F2D C9 A0 CMP #USRRAM/256+32 ; COMPARE WITH END OF USER RAAM
(1) 2F2F B0 OA BCS DMAST1 ; JUMP IF NOT IN USER RAM
(1) 2F31 38 SEC ; IF IN USER RAM, COMPUTE RELATIVE PAGE
(1) 2F32 E9 80 SBC #USRRAM/256 ; ADDRESS IN USER RAM
(1)
(1) 2F34 48 PHA ; SAVE RESULT
(1) 2F35 A9 00 LDA #USRRAM/32& $FF ; TEST IF USRRAM IS ON AN ODD 4K BOUNDARY
(1) 2F37 30 12 BMI DMAST2 ; JUMP ON ODD BOUNDARY
(1) 2F39 10 16 BPL DMAST3 ; JUMP ON EVEN BOUNDARY
(1) 2F3B C9 C0 DMAST1: CMP #SYSRAM/256 ; COMPARE WITH BEGINNING OF SYSTEM RAM
(1) 2F3D 90 1A BCC DMASTE ; ERROR IF NOT IN DISK CONTROLLER RAM
(1) 2F3F C9 DF CMP #SYSRAM/256+31 ; COMPARE WITH END OF USABLE SYSTEM RAM
(1) 2F41 B0 16 BCS DMASTE ; ERROR IF NOT IN DISK CONTROLLER RAM
(1) 2F43 38 SEC ; IF IN SYSTEM RAM, COMPUTE RELATIVE PAGE
(1) 2F44 E9 A0 SBC #SYSRAM/256-32 ; ADDRESS IN SYSTEM RAM
(1) 2F46 48 PHA ; SAVE RESULT
(1) 2F47 A9 00 LDA #SYSRAM/32& $FF ; TEST IF SYSRAM IS ON AN ODD 4K BOUNDARY
(1) 2F49 10 06 BPL DMAST3 ; JUMP ON EVEN BOUNDARY
(1) 2F4B 68 DMAST2: PLA ; RETRIEVE RELATIVE ADDRESS
(1) 2F4C 49 10 EOR #$10 ; FLIP BIT 6 (AFTER SHIFT) FOR ODD BOUNDARY
(1) 2F4E 4C 52 2F JMP DMAST4
(1)
(1) 2F51 68 DMAST3: PLA ; RETRIEVE RELATIVE ADDRESS
(1) 2F52 0A DMAST4: ASL A ; MULTIPLY RELATIVE PAGE ADDRESS BY 4
(1) 2F53 0A ASL A
(1) 2F54 8D EA DF STA FDCDMA ; AND PUT RESULT IN DMA ADDRESS REGISTER
(1) 2F57 18 CLC
(1) 2F58 60 RTS ; NORMAL RETURN
(1) ; RTNS WITH DMA CODE IN ACC
(1)
(1) 2F59 A9 16 DMASTE: LDA #DMAERR ; DMA PAGE ERROR
(1) 2F5B 20 82 2F JSR ERROR
(1)
(1)

```

```

4349          .SBTTL  Disk Drivers - Command Tables
4350
4351
4352          ;      LIST OF DISK COMMANDS, STARTS IN PROGRAM AREA BUT MUST BE IN
4353          ;      RAM SO THAT CERTAIN BYTES OF THE COMMANDS CAN BE CHANGED
4354
4355          DSKCMD:          ; START OF PREFORMATTED DISK COMMANDS
4356          2F5E 03        SPECCM: .BYTE 3          ; #          ; THREE BYTES IN SPECIFY COMMAND
4357          2F5F 03        .BYTE $03        ; CMD        ; SPECIFY COMMAND
4358          2F60 AF        .BYTE $AF        ; SRT        ; SEEK SPEED=6ms HEAD UNLOAD TIME=240ms
4359          2F61 24        .BYTE $24        ; HLT        ; HEAD LOAD TIME=40ms DMA MODE
4360
4361          2F62 01        SNSICM: .BYTE 1          ; #          ; 1 BYTE IN SENSE INT. CMD
4362          2F63 08        .BYTE $08        ; CMD        ; SENSE INTERRUPT STATUS COMMAND
4363
4364          2F64 02        RECLCM: .BYTE 2          ; #          ; 2 BYTES IN RECALIBRATE CMD
4365          2F65 07        .BYTE $07        ; CMD        ; RECALIBRATE COMMAND
4366          2F66 00        .BYTE 00         ; DRV        ; DRIVE NUMBER IN BITS 0-1
4367
4368          2F67 03        SEEKCM: .BYTE 3          ; #          ; 3 BYTES IN SEEK CMD
4369          2F68 0F        .BYTE $0F        ; CMD        ; SEEK COMMAND
4370          2F69 00        .BYTE 00         ; DRV        ; DRIVE NUMBER IN BITS 0-1, SIDE NUMBER B2
4371          2F6A 00        .BYTE 00         ; NCN        ; NEW CYLINDER NUMBER
4372
4373          2F6B 02        SDSTCM: .BYTE 2          ; #          ; 2 BYTES IN SENSE DRIVE STATUS CMD
4374          2F6C 04        .BYTE $04        ; CMD        ; SENSE DISK STATUS COMMAND
4375          2F6D 00        .BYTE 00         ; DRV        ; DRIVE NUMBER IN BITS 0-1, SIDE NUMBER B2
4376
4377          2F6E 09        READCM: .BYTE 9          ; #          ; 9 BYTES IN READ DATA CMD
4378          2F6F 46        .BYTE $46        ; CMD        ; READ DATA, MFM, READ DEL DATA
4379          2F70 00        .BYTE 00         ; DRV        ; HEAD ZERO, DRIVE NUMBER IN BITS 0-1
4380          2F71 00        .BYTE 00         ; C          ; NEEDS PRESENT CYLINDER NUMBER
4381          2F72 00        .BYTE 00         ; H          ; NEEDS HEAD NUMBER
4382          2F73 00        .BYTE 00         ; R          ; NEEDS SECTOR NUMBER TO READ
4383          2F74 01        .BYTE 01         ; N          ; CODE FOR 256 BYTES PER SECTOR
4384          2F75 00        .BYTE 00         ; EOT        ; LAST SECTOR TO READ
4385          2F76 0E        .BYTE $0E        ; GPL        ; GAP LENGTH FOR 26 SECTORS, 256 BYTES/SECT
4386          2F77 FF        .BYTE $FF        ; DTL        ; DATA LENGTH = $FF SINCE N IS NON-ZERO
4387
4388          2F78 09        WRITCM: .BYTE 9          ; #          ; 9 BYTES IN WRITE DATA CMD
4389          2F79 45        .BYTE $45        ; CMD        ; WRITE DATA, MFM
4390          2F7A 00        .BYTE 00         ; DRV        ; HEAD ZERO, DRIVE NUMBER IN BITS 0-1
4391          2F7B 00        .BYTE 00         ; C          ; NEEDS PRESENT CYLINDER NUMBER
4392          2F7C 00        .BYTE 00         ; H          ; NEED HEAD NUMBER
4393          2F7D 00        .BYTE 00         ; R          ; NEEDS SECTOR NUMBER TO WRITE
4394          2F7E 01        .BYTE 01         ; N          ; CODE FOR 256 BYTES PER SECTOR
4395          2F7F 00        .BYTE 00         ; EOT        ; LAST SECTOR TO WRITE
4396          2F80 0E        .BYTE $0E        ; GPL        ; GAP LENGTH FOR 26 SECTORS, 256 BYTES/SECT
4397          2F81 FF        .BYTE $FF        ; DTL        ; DATA LENGTH = $FF SINCE N IS NON-ZERO
4398
4399

```

```

4401          .SBTTL  Error Routine
4402
4403          ;          .READ  ERROR.A
4404          .MCALL  ERROR
4405  2F82          ERROR
(1)
(1)
(1)  2F82  8D  46  36  ERROR:  STA  ERRNO          ; SAVE ERROR NUMBER
(1)  2F85  20  E0  2B          JSR  CRLF          ; NEW LINE
(1)  2F88  A9  3F          LDA  #'?
(1)  2F8A  20  95  2B          JSR  CTYOUT
(1)  2F8D  20  09  10          JSR  CTYSPA
(1)  2F90  AD  46  36          LDA  ERRNO          ; GET ERROR NUMBER
(1)  2F93  20  C3  2B          JSR  PRTBYT
(1)  2F96  A0  02          LDY  #2          ; PRINT THE LAST TWO ADDRESSES ON THE STACK
(1)  2F98  20  40  2C  ERPADR: JSR  SPACE2        ; ERROR PRINT ADDR
(1)  2F9B  68          PLA          ; GET LOW BYTE OFF OF STACK
(1)  2F9C  AA          TAX          ; SAVE IN X REG
(1)  2F9D  68          PLA          ; GET HIGH BYTE OFF OF STACK
(1)  2F9E  20  C3  2B          JSR  PRTBYT        ; PRINT HIGH BYTE
(1)  2FA1  8A          TXA          ; GET LOW BYTE BACK FROM X REG
(1)  2FA2  20  C3  2B          JSR  PRTBYT        ; PRINT LOW BYTE
(1)  2FA5  88          DEY
(1)  2FA6  D0  F0          BNE  ERPADR        ; DO TWO TIMES
(1)  2FA8  20  E0  2B          JSR  CRLF
(1)
(1)  2FAB  A9  E4          LDA  #ERRORS&$FF  ; INIT TO BEGINING OF ERROR TABLE
(1)  2FAD  85  02          STA  PRTPTR+0     ; LOW BYTE
(1)  2FAF  A9  2F          LDA  #ERRORS^    ; HIGH BYTE
(1)  2FB1  85  03          STA  PRTPTR+1
(1)  2FB3  A0  00          LDY  #0          ; INIT INDEX POINTER
(1)  2FB5  B1  02  ERRRLP: LDA  (PRTPTR),Y
(1)  2FB7  F0  12          BEQ  ERR.ER      ; IF AT END OF TABLE THEN INVALID ERROR #
(1)  2FB9  CD  46  36          CMP  ERRNO      ; IS IT THE RIGHT ERROR # ?
(1)  2FBC  F0  0D          BEQ  ERRPRT     ; IF SO GO PRINT IT
(1)  2FBE  20  DD  2F  ERRLPO: JSR  INCPTR    ; IF NOT, SKIP THIS ERROR MSG
(1)  2FC1  B1  02          LDA  (PRTPTR),Y
(1)  2FC3  D0  F9          BNE  ERRLPO
(1)  2FC5  20  DD  2F          JSR  INCPTR     ; SKIP OVER THE ZERO
(1)  2FC8  4C  B5  2F          JMP  ERRLP
(1)
(1)  2FCB          ERR.ER:          ; IF NO MATCH, THEN FALL THROUGHM
(1)
(1)
(1)  2FCB  20  DD  2F  ERRPRT: JSR  INCPTR    ; SKIP OVER ERROR #
(1)  2FCE  A0  00          LDY  #0
(1)  2FDO  B1  02  ERPRLP: LDA  (PRTPTR),Y
(1)  2FD2  F0  06          BEQ  EREXIT
(1)  2FD4  20  95  2B          JSR  CTYOUT
(1)  2FD7  C8          INY
(1)  2FD8  D0  F6          BNE  ERPRLP     ; WILL ALWAYS BRANCH
(1)  2FDA  4C  03  10  EREXIT: JMP  JWARM      ; GO BACK TO WARM START
(1)
(1)
(1)  2FDD  E6  02          INCPTR: INC  PRTPTR+0
    
```



```

(1) 2FDF D0 O2          BNE    INCEXT
(1) 2FE1 E6 O3          INC    PRTPTR+1
(1) 2FE3 60            INCEXT: RTS
(1)
(1)
4406
4407      0001          CMDERR =    $01          ; Command Error
4408      0002          DSKERR =    $02          ; Disk Type Error
4409      0003          DRVERR =    $03          ; Invalid Drive # Error
4410      0004          MEMFUL =    $04          ; Memory Full
4411      0005          SEKERR =    $05          ; Seek Error
4412      0006          RCLERR =    $06          ; Recalibrate Error
4413      0008          USRERR =    $08          ; User Device Error
4414      000A          DENERR =    $0A          ; Invalid Density Error
4415      000B          NOTIMP =    $0B          ; Not Implimented Yet
4416      000C          FIOERR =    $0C          ; File I/O Error
4417      000D          RDSERR =    $0D          ; Read Sector Error
4418      000E          WRSERR =    $0E          ; Write Sector Error
4419      000F          FCMDER =    $0F          ; FDC Comand Error
4420      0010          FRSLER =    $10          ; FDC Result Error
4421      0011          OPNERR =    $11          ; Open Error, File Not Found
4422      0012          CLOERR =    $12          ; Close Error, File Not Closed
4423      0013          INVFNM =    $13          ; Invalid File Name
4424      0014          NUMERR =    $14          ; Invalid Char in Number
4425      0015          NOPARM =    $15          ; Missing Needed Parameter
4426      0016          DMAERR =    $16          ; DMA Page Error
4427      0017          DEVERR =    $17          ; Invalid Device
4428      0018          DEFERR =    $18          ; Define Error
4429      0019          UNDFDT =    $19          ; Undefined Drive
4430      001B          DBTERR =    $1B          ; Disk is not CODOS
4431      001C          INVNUM =    $1C          ; Invalid Number
4432      001D          INVPRM =    $1D          ; Invalid Parameter
4433      001E          MEMBER =    $1E          ; Memory Buffer Error
4434
4435      00FF          ERRERR =    $FF          ; (Internal) Invalid Error #
4436

```

4438

4439 2FE4
 4440 2FE4 01 43 6F
 4441 2FF3 02 49 6E
 4442 3006 03 49 6E
 4443 3017 04 4D 65
 4444 3024 05 53 65
 4445 3030 06 52 65
 4446 3043 08 55 73
 4447 305D 0A 49 6E
 4448 3074 0B 4E 6F
 4449 3089 0C 46 69
 4450 3099 0D 52 65
 4451 30AC 0E 57 72
 4452 30C0 0F 46 44
 4453 30D3 10 46 44
 4454 30E5 11 4F 70
 4455 3101 12 43 6C
 4456 311F 13 49 6E
 4457 3132 14 49 6E
 4458 314A 15 4D 69
 4459 3166 16 44 4D
 4460 3176 17 49 6E
 4461 3186 18 44 65
 4462 3194 19 44 72
 4463 31A7 1B 44 69
 4464 31C5 1C 49 6E
 4465 31D5 1D 49 6E
 4466 31E8 1E 4D 65
 4467 31FD 00
 4468 31FE FF 28 49
 4469
 4470

ERRORS:

.ASCIZ <CMDERR>'Command Error'
 .ASCIZ <DSKERR>'Invalid Disk Type'
 .ASCIZ <DRVERR>'Invalid Drive #'
 .ASCIZ <MEMFUL>'Memory Full'
 .ASCIZ <SEKERR>'Seek Error'
 .ASCIZ <RCLERR>'Recalibrate Error'
 .ASCIZ <USRERR>'User Device is Undefined'
 .ASCIZ <DENERR>'Invalid Density Error'
 .ASCIZ <NOTIMP>'Not Implimented Yet'
 .ASCIZ <FIOERR>'File I/O Error'
 .ASCIZ <RDSERR>'Read Sector Error'
 .ASCIZ <WRSERR>'Write Sector Error'
 .ASCIZ <FCMDER>'FDC Command Error'
 .ASCIZ <FRSLER>'FDC Result Error'
 .ASCIZ <OPNERR>'Open Error, File Not Found'
 .ASCIZ <CLOERR>'Close Error, File Not Closed'
 .ASCIZ <INVFNM>'Invalid File Name'
 .ASCIZ <NUMERR>'Invalid Char in Number'
 .ASCIZ <NOPARM>'Missing Required Parameter'
 .ASCIZ <DMAERR>'DMA Page Error'
 .ASCIZ <DEVERR>'Invalid Device'
 .ASCIZ <DEFERR>'Define Error'
 .ASCIZ <UNDFDT>'Drive not Defined'
 .ASCIZ <DBTERR>'Disk in Drive 0 is not CODOS'
 .ASCIZ <INVNUM>'Invalid Number'
 .ASCIZ <INVPRM>'Invalid Parameter'
 .ASCIZ <MEMBER>'Memory Buffer Error'
 .BYTE 0 ; END of ERROR LIST
 .ASCIZ <ERRERR>'(Internal) Invalid Error # ???'

```

4472          .SBTTL Text Strings
4473
4474
4475 321E 4D 54 55 TTLMS1: .ASCIZ 'MTU DISKEX 1.0'
4476 322E 44 69 73 TTLMS2: .ASCIZ 'Disk Exchange Program'
4477 3244 43 6F 70 TTLMS3: .ASCIZ 'Copyright 1981 by M.T.U.'
4478
4479 325D 44 49 53 PRMPT: .ASCIZ 'DISKEX>' ; DISK EXCHANGE PGM PROMPT
4480 3265 44 72 69 DRVMSG: .ASCIZ 'Drive #'
4481 326D 44 65 66 DI.MSG: .ASCIZ 'Default Input Drive = '
4482 3285 44 65 66 DO.MSG: .ASCIZ 'Default Output Drive = '
4483 329D 42 45 47 BEGMSG: .ASCIZ 'BEGGINING of internal file buffer $'
4484 32C3 45 4E 44 ENDMMSG: .ASCIZ 'END address of internal file buffer $'
4485 32E9 20 20 20 PSZMSG: .ASCIZ ' SIZE = ' ; PRINT SIZE MESSAGE
4486 32F4 20 46 69 FSZMSG: .ASCIZ ' File Size = $'
4487 3303 43 6F 6E CONFIRM: .ASCIZ 'Confirm (Y/N) ?'
4488
4489          ; DIRECTORY HEADERS
4490
4491 3313 20 56 53 CDHDR: .ASCIZ ' VSN Filename BAT' ; CODOS
4492 332A 20 46 4C FLXHDR: .ASCIZ ' FLEX ST END SIZE DATE' ; FLEX
4493 3351 20 43 50 CPMHDR: .ASCIZ ' CP/M E SZ BAT ENTRYS' ; CP/M
4494 3371 20 49 42 IBMHDR: .ASCIZ ' IBM' ; IBM
4495 3376 20 52 53 RSXHDR: .ASCIZ ' RSX-11' ; RSX-11
4496 337E 20 48 2E HDEHDR: .ASCIZ ' H.D.E.' ; HDE DOS
4497 3386 20 54 52 TRSHDR: .ASCIZ ' TRS-DOS DATE ATTR' ; TRS-80 DOS
4498 33A5 20 55 2E PS1HDR: .ASCIZ ' U.C.S.D. PASCAL SINGLE DENSITY' ; UCSD PASCAL
4499 33C5 20 55 2E PS2HDR: .ASCIZ ' U.C.S.D. PASCAL DOULBE DENSITY' ; UCSD PASCAL
4500 33E5 20 55 73 USRHDR: .ASCIZ ' User Header' ; USER
4501 33F2 20 43 48 CMXHDR: .ASCIZ ' CHROMATICS PW START LENGTH CREATED ACCESSED STATUS'
4502          ;
4503          ; 12345678.123 ** 12345678 12345678 12345678 12345678 1234
4503 342D ERRHDR:          ; ERROR
4504 342D 00 00          .WORD 0 ; END OF LIST
4505
4506
4507          ; EBCDIC <=> ASCII TABLES & ROUTINES
4508
4977
  
```

```
4979          .SBTTL  Variables & Buffers
4980
4981          ; VARIABLES THAT MUST HAVE A VALUE ON STARTUP
4982
4983 35B9 00 39          BEGMEM: .ADDR  END          ; ADDRESS OF START OF MEMORY
4984 35BB FF 7F          ENMEMEM: .ADDR  USRRAM-1      ; ADDRESS OF END OF AVAILABLE MEMORY
4985
4986
4987
4988 35BD 4F 50 45      CDSTXT: .ASCII  'OPEN O'<$OD>  ; CODOS 'OPEN DRIVE' COMMAND
4989
4990
4991
4992 35C4          DRIVES:          ; LABEL USED FOR INDEXING
4993          ; THESE ARE THE CURRENT DISK
4994          ; TYPE OF THE DRIVE
4995 35C4 00          DRIVE0: .BYTE  0
4996 35C5 00          DRIVE1: .BYTE  0
4997 35C6 00          DRIVE2: .BYTE  0
4998 35C7 00          DRIVE3: .BYTE  0
4999
5000 35C8 01          DRIVEI: .BYTE  1          ; DEFAULT INPUT DRIVE
5001 35C9 00          DRIVEO: .BYTE  0          ; DEFAULT OUTPUT DRIVE
5002
5003          ; ***** END OF MEMORY THAT MUST BE SAVED *****
5004
5005 35CA 35CB        DRVTMP:  .=.+1          ; USED IN SET-DRIVE ONLY
5006
```

```

5008          .SBTTL Non-Zero Page Storage
5009
5010          3600          . =          <<.^>+1)*256          ; EVEN PAGE #
5011
5012 3600 3610          DSKSTS: . = .+16          ; STORAGE for NEC-765 STATUS BYTES
5013
5014          ; INPUT DEVICE VARIABLES
5015 3610 3612          INPJMP: . = .+2          ; INPUT INDIRECT JUMP
5016 3612 3613          INPDEV: . = .+1          ; INPUT DEVICE #
5017 3613 3614          INPDRV: . = .+1          ; INPUT DRIVE #
5018 3614 3615          INPPTR: . = .+1          ; INPUT DEVICE SECTOR POINTER
5019 3615 3616          INPTRK: . = .+1          ; INPUT DEVICE TRACK #
5020 3616 3617          INPSCT: . = .+1          ; INPUT DEVICE SECTOR #
5021 3617 3618          INLTRK: . = .+1          ; LAST TRACK IN FILE (IF USED)
5022 3618 3619          INLSCT: . = .+1          ; LAST SECTOR IN FILE (IF USED)
5023 3619 361A          INBYSC: . = .+1          ; BYTES/SECTOR of INPUT DRIVE
5024 361A 361B          INPDEN: . = .+1          ; INPUT DENSITY FLAG
5025 361B 361C          INPAGE: . = .+1          ; INPUT DMA PAGE
5026          3620          . = .+4
5027
5028          ; OUTPUT DEVICE VARIABLES
5029 3620 3622          OUTJMP: . = .+2          ; OUTPUT INDIRECT JUMP
5030 3622 3623          OUTDEV: . = .+1          ; OUTPUT DEVICE #
5031 3623 3624          OUTDRV: . = .+1          ; OUTPUT DRIVE #
5032 3624 3625          OUTPTR: . = .+1          ; OUTPUT DEVICE SECTOR POINTER
5033 3625 3626          OUTTRK: . = .+1          ; OUTPUT DEVICE TRACK #
5034 3626 3627          OUTSCT: . = .+1          ; OUTPUT DEVICE SECTOR #
5035 3627 3628          OULTRK: . = .+1          ; LAST TRACK IN FILE (IF USED)
5036 3628 3629          OULSCT: . = .+1          ; LAST SECTOR IN FILE (IF USED)
5037 3629 362A          OUBYSC: . = .+1          ; BYTES/SECTOR of OUTPUT DRIVE
5038 362A 362B          OUTDEN: . = .+1          ; OUTPUT DENSITY FLAG
5039 362B 362C          OUPAGE: . = .+1          ; OUTPUT DMA PAGE
5040          3630          . = .+4
5041
5042          ; TEMPORARY LOCATIONS USED BY OPEN FILE ROUTINES
5043 3630 3632          TMPJMP: . = .+2          ; TEMP INDIRECT JUMP
5044 3632 3633          TMPDEV: . = .+1          ; TEMP DEVICE #
5045 3633 3634          TMPDRV: . = .+1          ; TEMP DRIVE #
5046 3634 3635          TMPPTR: . = .+1          ; TEMP DEVICE SECTOR POINTER
5047 3635 3636          TMPTRK: . = .+1          ; TEMP DEVICE TRACK #
5048 3636 3637          TMPSC: . = .+1          ; TEMP DEVICE SECTOR #
5049 3637 3638          TMLTRK: . = .+1          ; LAST TRACK TO READ
5050 3638 3639          TMLSCT: . = .+1          ; LAST SECTOR TO READ
5051 3639 363A          TMBYSC: . = .+1          ; BYTES/SECTOR of TEMP DRIVE
5052 363A 363B          TMPDEN: . = .+1          ; TEMP DENSITY FLAG
5053 363B 363C          TMPAGE: . = .+1          ; TEMP DMA PAGE
5054 363C 363D          TMPIOF: . = .+1          ; TEMP INPUT or OUTPUT FLAG (TEMP ONLY)
5055          3640          . = .+3
5056

```

```

5058
5059 3640 3642      CMDJMP:  .=.+2      ; INDIRECT COMMAND JUMP
5060 3642 3644      NUMBER:  .=.+2      ; 16 BIT NUMBER
5061 3644 3645      BUFPTR:  .=.+1      ; INPUT BUFFER POINTER
5062 3645 3646      ATMLN:   .=.+1      ; LENGTH of ATOM
5063
5064 3646 3647      ERRNO:   .=.+1      ; LAST ERROR NUMBER
5065 3647 3648      NORTRY:  .=.+1      ; NUMBER of RETRYS SINCE LAST INITIALIZE
5066 3648 3649      RETRYS:  .=.+1      ; TEMP RETRY COUNTER
5067 3649 364A      DMATMP:  .=.+1      ; DMA TEMP SAVE (USED ONLY WHEN RE-TRYed)
5068 364A 364B      DELIMT:  .=.+1      ; TEMP SAVE of CURRENT DELIMITER
5069
5070                                     ; SHORT TERM REGISTER SAVE
5071 364B 364C      ACC:      .=.+1      ; ACC
5072 364C 364E      WORD:    .=.+2      ; TEMP SAVE WORD
5073
5074                                     ; SEPERATE SYSTEM VARIABLES
5075
5076 364E 364F      CDSTMP:  .=.+1      ; TEMP FOR CDS DIR
5077
5078 364F 3650      FLXTFL:  .=.+1      ; FLEX TAB FLAG
5079 3650 3651      FLXTCT:  .=.+1      ; FLEX TAB COUNTER
5080
5081 3651 3652      CPMIRC:  .=.+1      ; CP/M INPUT RECORD COUNTER
5082 3652 3653      CPMEXI:  .=.+1      ; CP/M EXTENT NUMBER INPUT
5083 3653 3654      CPMBPI:  .=.+1      ; CP/M BLK TABLE POINTER
5084 3654 3655      CPMNSI:  .=.+1      ; CP/M # OF SECTORS
5085 3655 3665      CPMBTI:  .=.+16     ; CP/M BLOCK TABLE INPUT
5086
5087 3665 3666      CPMEXT:  .=.+1      ; CP/M TEMP SECTOR NUMBER
5088 3666 3667      CPMTFL:  .=.+1      ; CP/M CHAR COUNT (FOR EXPANDING TABS)
5089 3667 3668      CPMTCT:  .=.+1      ; CP/M TAB COUNTER
5090 3668 3669      CPMTBT:  .=.+1      ; CP/M TAB COUNTER ALSO
5091
5092 3669 366A      CPMORC:  .=.+1      ; CP/M OUTPUT RECORD COUNTER
5093 366A 366B      CPMEXO:  .=.+1      ; CP/M EXTENT NUMBER OUTPUT
5094 366B 366C      CPMBPO:  .=.+1      ; CP/M BLK TABLE POINTER
5095 366C 366D      CPMNSO:  .=.+1      ; CP/M # OF SECTORS
5096 366D 367D      CPMBTO:  .=.+16     ; CP/M BLOCK TABLE OUTPUT
5097
5098 367D 368D      CPMBUF:  .=.+16     ; CP/M NAME BUFFER
5099 368D 369D      CPMBFI:  .=.+16     ; CP/M NAME BUFFER INPUT
5100 369D 36AD      CPMBFO:  .=.+16     ; CP/M NAME BUFFER OUTPUT
5101 36AD 36AE      CPMOTC:  .=.+1      ; CP/M OUTPUT TEMP CHAR SAVE
5102
5103 36AE 36AF      IBMDSN:  .=.+1      ; IBM DIRECTORY SECTOR NUMBER
5104
5105                                     ; CHROMATICS TEMPS
5106 36AF 36B3      CMXSOF:  .=.+4      ; CHROMATICS START-OF-FILE
5107 36B3 36B7      CMXL0F:  .=.+4      ; CHROMATICS LENGTH-OF-FILE
5108 36B7 36BB      CMXIL:   .=.+4      ; CHROMATICS LENGTH-OF-FILE INPUT
5109 36BB 36BF      CMXOL:   .=.+4      ; CHROMATICS LENGTH-OF-FILE OUTPUT
5110 36BF 36C1      CMXTMP:  .=.+2      ; TEMP USED FOR DIRECTORY STATUS
5111

```

```

5113
5114           ; SUB-PROGRAM VARS
5115
5116 36C1 36C5      FSZCTR:  .=.+4           ; FILE SIZE COUNTER
5117 36C5 36C6      FRMDRV:  .=.+1          ; FROM DRIVE
5118 36C6 36C7      TODRV:   .=.+1          ; TO DRIVE
5119 36C7 36C8      SUBDEN:  .=.+1          ; SUB-PGM DENSITY
5120 36C8 36C9      SUBYSC:  .=.+1          ; SUB-PGM BYTES/SECTOR
5121 36C9 36CA      SBFRST:  .=.+1          ; FIRST SECTOR
5122 36CA 36CB      SBLAST:  .=.+1          ; LAST SECTOR
5123 36CB 36CC      SUBFLG:  .=.+1          ; READ/WRITE FLAG (R/W)
5124
5125 36CC 36CD      SUBDRV:   .=.+1          ; SUB-PGM DRIVE #
5126
5127 36CD 36CE      SUBTRK:  .=.+1          ; SUB-PGM DEVICE TRACK #
5128 36CE 36CF      SUBSCT:  .=.+1          ; SUB-PGM DEVICE SECTOR #
5129
5130
5131
5132
5133
5134 36CF 36D0      CTY.IA:   .=.+1          ; CTY INPUT SAVE ACC
5135 36D0 36D1      CTY.IX:   .=.+1          ; CTY INPUT SAVE X
5136 36D1 36D2      CTY.IY:   .=.+1          ; CTY INPUT SAVE Y
5137
5138 36D2 36D3      CTY.OA:   .=.+1          ; CTY OUTPUT SAVE ACC
5139 36D3 36D4      CTY.OX:   .=.+1          ; CTY OUTPUT SAVE X
5140 36D4 36D5      CTY.OY:   .=.+1          ; CTY OUTPUT SAVE Y
5141
5142           ; BUFFERS
5143
5144           3700           . = <<. ^>+1>*256 ; EVEN PAGE #
5145
5146 3700 3800      INBUFF:   .=.+256        ; INPUT COMMAND BUFFER
5147 3800 3880      ATMBUF:   .=.+128        ; ATOM BUFFER
5148
5149
5150           3900           . = <<. ^>+1>*256 ; EVEN PAGE #
5151           3900      END     = .
5152           2900      SIZE    = END-START
5153           000A      K       = SIZE/1024
5154
5155           1000           .END   START
    
```


P.OCHR	2AFA	3812	3817#										
P.OEX	2AF8	3810	3814#										
P.OPN	2ADE	595	1450	3791#									
P.OUT	2AED	599	1450	3807#									
QDELIM	19F3	1241#											
QDLMEQ	1A05	1241#											
RCLERR=	0006	4346	4412#	4445									
RDDDEN	2D3A	4183	4189#										
RDEXIT	14CE	651	657#										
RDLP	14BF	650#	653										
RDRTRN	2CF3	1024	4144	4153#									
RDRTRY	2D44	4194#	4210										
RDSCTR	2CAF	1567	1577	1580	1735	1744	1981	2028	2137	2418	2507	2807	2921
		3017	3197	3388	3566	3586	3698	3706	4111#				
RSDEN	2D36	4181	4187#										
RDSERR=	000D	4134	4417#	4450									
RDTEMP	2CED	853	899	1823	1945	1969	2057	2086	2219	2846	2954	3464	4144#
RDXDEN	2D3C	4188	4190#										
RD.ERR	2CE8	4131	4134#	4145									
READ	2D29	4130	4180#										
READCM	2F6E	4119*	4122*	4125*	4127*	4128*	4161*	4164*	4167*	4169*	4171*	4190*	4195*
		4198	4377#										
READER	2D62	4204	4207#										
READF	14B7	309	646#										
READWT	2D54	4200#	4201										
RECAL	2E61	479	501	1365	4346#								
RECALE	2E83	4346#											
RECAL1	2E6B	4346#											
RECLCM	2F64	4346*	4364#										
RERROR	2F07	4203	4328	4346#									
RERR1	2F17	4346#											
RESET	135E	293	479#										
RESULT	2EE4	479	4202	4327	4346#								
RETRYS	3648	4192*	4209*	4317*	4334*	5066#							
RRCLLP	136E	479#											
RSCTR	164F	847	853#										
RSLDAT	123C	349#											
RSLEQU	124A	349#											
RSLPHE	2F02	4346#											
RSLSTX	120D	349#											
RSTDNR	1383	479#											
RSTRCL	136C	479#											
RESULT1	2EE6	4346#											
RSXCLO	26FE	1426	3115#										
RSXDBS=	0100	1431	3103#										
RSXDEF	26FC	1423	3107#										
RSXDIR	26FC	1424	3108#										
RSXHDR	3376	1430	4495#										
RSXINP	26FE	1427	3113#										
RSXOPN	26FE	1425	3112#										
RSXOUT	26FE	1428	3114#										
RSXSCT=	011A	1432	3104#										
RUBOUT	1970	1239#											
R..ERR	2F27	4346#											
SBFRST	36C9	781*	876	933*	940	5121#							

SBLAST	36CA	784*	873	936*	942	5122#								
SCTTAB	1C29	1432#												
SDRIVE	1328	295	447#											
SDRVER	1347	451	463#											
SDRVIN	134C	456	466#											
SDRVOU	1355	458	470#											
SDRVST	2E41	479	4038	4346#										
SDSTCM	2F6B	4346*	4373#											
SECTOK	160A	818	820	824#										
SECTOR	15E5	317	809#											
SFEK	2E88	4113	4155	4239	4280	4346#								
SEEKCM	2F67	4194	4319	4346*	4368#									
SEEKER	2EAE	4346#												
SEEKOK	2EBD	4346#												
SEEK1	2E97	4346#												
SEKERR=	0005	4346	4411#	4444										
SINTST	2E59	276	4346#											
SIZE =	2900	5152#	5153											
SMEMER	127A	352	361	369#										
SNSICM	2F62	4346	4361#											
SPACE2	2C40	349	398	431	1595	1614	1768	1774	2169	3231	3253	4026#	4049	
		4060	4405											
SPACE4	2C3A	417	905	3215	4024#									
SPECCM	2F5E	4346	4356#											
SPECFY	2E4C	4346#												
SPECF1	2E54	4346#												
START	1000	188#	344	5152	5155									
STATOK	11EB	339	344#											
STATUS	11DE	283	337#											
SUBDEN	36C7	774*	792	810	885	5119#								
SUBDRV	36CC	829*	887	5125#										
SUBFLG	36CB	816*	845	5123#										
SUBINI	1AE4	1364#												
SUBSCT	36CE	842*	871*	872	877*	893	5128#							
SUBTRK	36CD	835*	878*	879	891	5127#								
SUBYSC	36C8	778*	812	889	5120#									
SVCENA=	00EE	176#	998*	1001*	1006*	1555*	1558*	1638*	1664*	1684*				
SYSIN	1006	190#	3913											
SYSOUT	100C	193#	3917	3927										
SYSRAM=	C000	134#	174	190	193	194	1029	4346						
TMBYSC	3639	499*	813*	890*	1023*	1059*	1107*	2528*	2641*	2947*	4159	4284	5051#	
TMLSCT	3638	841*	895*	943*	1019*	1819*	1936*	2053*	2082*	2212*	2215*	2840*	2842*	
		3463*	3478*	4170	4295	5050#								
TMLTRK	3637	5049#												
TMPAGE	363B	844*	897*	946*	1014*	1138*	1367*	1944*	1968*	2056*	2085*	2953*	4156	
		4281	5053#											
TMPDEN	363A	497*	811*	886*	1021*	1063*	1111*	2530*	2643*	2945*	4172	4297	5052#	
TMPDEV	3632	1048*	1096*	5044#										
TMPDRV	3633	486*	488	500	828*	888*	949*	952*	1016*	1045*	1093*	1375*	1551	
		1649	1655	2526*	2639*	2943*	4037*	4042	4061	4153	4162	4278	4287	
		5045#												
TMPIOF	363C	1136*	1644	1955	2365	2524*	2637*	2876	3546	5054#				
TMPJMP	3630	493*	495*	1162*	1164*	1171*	1173*	1215*	1217*	1318	5043#			
TMPPTR	3634	3544*	3554	3574	5046#									
TMPSCT	3636	840*	894*	908	941*	1018*	1818*	1935*	1950*	2052*	2081*	2211*	2214*	

CMDPRS	275#	276									
DEVICE	1436#	1448	1449	1450	1451	1452					
DISKSR	4345#	4346									
DSKEQU	171#	174									
ERROR	4404#	4405									
FDCRST	478#	479									
GETNUM	1237#	1243									
GOEXIT	976#	977									
GTATOM	1237#	1241									
INSTRG	1237#	1239									
JEQ	1834	2236	2249	2262							
JNE	1905	2224	2228	2240	2253	2266	2279	2292	2389	3264	
MACDEF	171#										
PRESLT	348#	349									
SVC	176#	999	1556	1650	1656	1662	1672	1677	1682		
SVCBYT	176#										
SVCDEF	171#	176									
TABLE	1404#	1423	1424	1425	1426	1427	1428	1430	1431	1432	

ADC	758	991	994	1335	1589	2134	2144	2693	2696	2700	2703	2718	3532	3535
	3538	3966	3967											
AND	338	406	479	773	777	1028	1197	1239	1243	1288	1374	2929	2936	3045
	3235	3680	3681	3682	3962	4007	4016	4118	4121	4160	4163	4243	4246	4285
	4288	4346												
ASL	490	536	1049	1097	1159	1243	2685	2687	2689	2930	2931	2932	2933	3421
	4063	4064	4076	4346										
BCC	583	600	653	672	691	705	715	731	760	1042	1090	2672	4077	4346
BCS	276	352	361	449	483	503	518	581	598	618	651	670	689	770
	776	780	783	825	831	837	926	929	932	935	977	984	1025	1068
	1116	1166	1175	1221	1998	2006	2539	2652	3423	3844	4131	4145	4204	4256
	4272	4329	4346											
BEQ	276	339	379	408	456	458	479	534	794	796	818	820	847	849
	874	987	1239	1241	1243	1291	1330	1349	1583	1592	1601	1646	1749	1757
	1821	1841	1849	1856	1863	1870	1877	1884	1905	1940	1957	1987	2000	2015
	2027	2039	2066	2074	2141	2224	2228	2240	2253	2266	2275	2279	2288	2292
	2301	2314	2324	2366	2389	2467	2498	2550	2615	2811	2859	2877	2987	3002
	3014	3047	3049	3054	3069	3081	3264	3390	3395	3467	3470	3473	3476	3547
	3602	3608	3614	3620	3626	3632	3638	3641	3684	3689	3746	3756	3761	3767
	3810	3859	3916	3942	3977	3988	3993	4009	4018	4181	4183	4306	4308	4346
	4405													
BGE	451	485	520	532	772	827	833	839	1158	1243	2218	2496	2613	2845
	3003	3015	3070	3082	3196	3697	3876	4011	4020					
BLT	748	755	875	881	957	1239	1243	1621	1636	2177	2182	2404	2440	2489
	2606	2823	2828	2855	2998	3001	3007	3013	3065	3068	3074	3080	3399	3428
	3432	3436	3481	3873										
BMI	276	1041	1057	1061	1089	1105	1109	1239	2147	2479	2485	2591	2602	3242
	3754	3965	3986	4201	4326	4346								
BNE	276	349	381	620	622	624	626	861	1030	1071	1187	1194	1239	1241
	1243	1254	1258	1262	1272	1303	1337	1355	1586	1591	1604	1612	1617	1645
	1740	1754	1756	1761	1767	1785	1793	1828	1834	1836	1843	1851	1858	1865
	1872	1879	1902	1919	1924	1929	1956	1989	1993	2002	2010	2070	2089	2154
	2159	2175	2233	2236	2246	2249	2259	2262	2272	2285	2298	2305	2311	2318
	2340	2345	2350	2355	2367	2386	2469	2552	2560	2565	2571	2574	2852	2858
	2878	3033	3092	3201	3214	3236	3238	3244	3262	3548	3597	3604	3610	3616
	3622	3628	3634	3640	3649	3653	3657	3758	3763	3769	3812	3886	3945	3990
	3995	4071	4188	4210	4313	4335	4346	4405						
BPL	479	1657	1750	2146	2194	2424	2459	2536	2649	2714	3035	3529	3560	3580
	4346													
BRK	1241	1605	1606											
CLC	756	990	1073	1083	1126	1202	1224	1239	1241	1243	1334	1559	1588	1651
	1658	1678	1685	1726	1970	1982	2019	2032	2042	2062	2092	2124	2133	2143
	2426	2461	2475	2580	2662	2673	2691	2698	2716	2799	2900	2923	2990	3022
	3055	3095	3108	3131	3182	3290	3308	3354	3381	3437	3530	3567	3587	3658
	3662	3690	3700	3708	3714	3731	3741	3747	3771	3775	3795	3801	3814	3818
	3853	3861	3877	3964	4346									
CLD	217	276	1338	1357										
CMP	276	378	450	455	457	484	519	531	728	771	793	795	817	819
	826	832	838	846	848	860	873	880	956	986	1029	1070	1157	1186
	1193	1239	1241	1243	1253	1257	1261	1290	1329	1348	1600	1620	1635	1827
	1833	1835	1840	1842	1848	1850	1855	1857	1862	1864	1869	1871	1876	1878
	1883	1918	1923	1928	1999	2001	2181	2217	2227	2232	2235	2239	2245	2248
	2252	2258	2261	2265	2271	2274	2278	2284	2287	2291	2297	2300	2304	2310
	2313	2317	2323	2339	2344	2349	2354	2488	2549	2551	2559	2605	2810	2827
	2844	2851	2857	2997	3000	3006	3012	3032	3046	3048	3064	3067	3073	3079

	3195	3435	3480	3596	3601	3603	3607	3609	3613	3615	3619	3621	3625	3627
	3631	3633	3637	3639	3648	3652	3656	3683	3696	3745	3755	3757	3760	3762
	3766	3768	3809	3811	3841	3858	3872	3875	3963	3987	3989	3992	3994	4008
	4010	4017	4019	4180	4182	4305	4307	4346	4405					
CPX	1755	1784	2176	2495	2612	2822	3431							
CPY	747	754	2403	2439	2854	3398	3427							
DEC	1336	1988	2008	2468	2484	2564	2573	2601	2834	4209	4334			
DEX	479	2715	3034	3091										
DEY	349	1239	1243	1603	1611	1739	1753	1760	1766	1901	2153	2158	2174	2193
	2385	2423	2458	2535	2648	3213	3261	3559	3579	4070	4346	4405		
EOR	4346													
INC	380	382	619	621	623	625	761	871	878	954	1618	2014	2038	2179
	2214	2215	2478	2486	2493	2510	2522	2541	2590	2603	2610	2626	2635	2654
	2825	2841	2842	2916	2917	2989	2995	3010	3053	3062	3077	3193	3433	3477
	3478	3527	3542	3688	3694	3703	3885	3887	4405					
INX	276	349	409	1241	1243	1255	1259	1263	1267	1268	1269	1270	1292	1293
	1294	1295	1296	1297	1298	1299	1300	1301	1352	1502	1507	1519	1584	1602
	1609	1752	1770	1775	1776	1829	1837	1844	1852	1859	1866	1873	1880	1888
	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1900	1903	1908	1909
	1910	1911	1912	1913	1914	1920	1925	2069	2150	2165	2166	2167	2225	2229
	2242	2255	2268	2281	2294	2307	2320	2329	2330	2331	2332	2333	2334	2335
	2341	2346	2352	2358	2359	2360	2363	2370	2372	2373	2374	2375	2376	2377
	2378	2379	2380	2381	2382	2384	2387	2401	2437	2707	2821	3037	3206	3207
	3208	3209	3210	3260	3404	3416	3430	3599	3605	3611	3617	3623	3629	3635
	3646	3650	3654	4069	4346									
INY	276	746	753	1239	1241	1830	1838	1845	1853	1860	1867	1874	1881	1915
	1921	1926	2230	2243	2256	2269	2282	2295	2308	2321	2337	2342	2347	2402
	2438	2853	3397	3426	3944	4405								
JMP	188	189	190	193	194	199	344	349	357	366	383	435	468	472
	479	504	524	557	587	604	640	658	678	697	724	732	785	854
	863	910	958	977	1002	1007	1032	1143	1239	1316	1317	1318	1319	1327
	1834	1905	1906	2224	2228	2236	2240	2249	2253	2262	2266	2279	2292	2389
	2390	2557	2938	2967	3203	3264	3265	3268	3332	3335	3336	3337	3338	3339
	3784	3895	3896	3897	3898	3927	3968	4003	4013	4022	4346	4405		
JSR	223	224	227	228	231	232	234	236	238	240	241	244	247	250
	253	254	257	259	261	264	266	268	269	276	349	351	360	370
	394	397	398	400	411	412	414	416	417	419	420	423	424	426
	427	430	431	434	438	440	442	448	453	461	464	479	482	487
	501	502	507	510	513	517	529	533	552	553	556	560	563	577
	578	580	582	585	594	595	597	599	602	611	617	628	631	633
	635	637	639	646	648	650	652	655	657	666	667	669	671	673
	675	677	684	685	688	690	692	694	696	704	714	716	717	718
	719	720	721	722	723	726	734	736	738	740	741	744	745	749
	752	769	775	779	782	788	798	804	809	814	822	824	830	836
	851	853	858	859	862	866	884	899	900	902	904	905	907	909
	912	914	924	925	928	931	934	950	953	961	977	983	996	1024
	1035	1040	1046	1067	1072	1077	1088	1094	1115	1120	1132	1165	1169	1174
	1179	1185	1195	1213	1220	1239	1241	1243	1250	1275	1305	1326	1364	1365
	1372	1501	1506	1510	1511	1512	1513	1514	1518	1524	1567	1569	1571	1577
	1580	1594	1595	1599	1608	1610	1614	1615	1622	1624	1625	1627	1633	1666
	1669	1735	1738	1744	1746	1759	1763	1765	1768	1769	1771	1772	1773	1774
	1777	1779	1780	1782	1783	1796	1797	1798	1813	1823	1886	1945	1969	1981
	1997	2005	2028	2046	2057	2075	2086	2091	2135	2137	2149	2152	2155	2157
	2160	2163	2164	2168	2169	2172	2173	2189	2201	2219	2327	2411	2418	2447
	2480	2504	2507	2511	2538	2542	2592	2599	2621	2628	2651	2656	2669	2671

	2719	2807	2809	2812	2817	2818	2833	2846	2850	2870	2873	2884	2887	2905
	2908	2921	2922	2954	2957	2962	2966	2970	2988	3017	3018	3050	3087	3197
	3205	3212	3215	3218	3220	3222	3223	3226	3228	3230	3231	3240	3246	3249
	3252	3253	3255	3257	3267	3343	3388	3393	3396	3400	3401	3402	3403	3405
	3406	3407	3408	3409	3410	3411	3412	3425	3458	3464	3466	3469	3472	3475
	3566	3586	3698	3706	3740	3744	3753	3759	3770	3774	3779	3781	3783	3794
	3813	3817	3822	3827	3848	3860	3870	3881	3883	3902	3913	3917	3939	3943
	3956	3958	3975	3979	3984	3985	3991	3998	4000	4002	4024	4025	4026	4027
	4038	4041	4044	4045	4048	4049	4051	4053	4055	4057	4059	4060	4068	4079
	4080	4113	4115	4130	4135	4144	4155	4157	4185	4199	4202	4203	4239	4241
	4255	4261	4271	4280	4282	4310	4324	4327	4328	4346	4405			
LDA	191	219	233	235	237	239	245	251	258	260	265	267	276	337
	343	349	353	355	362	364	369	372	374	377	386	389	399	401
	405	413	415	418	425	437	454	460	463	466	470	479	492	494
	496	498	500	506	509	512	522	538	540	543	545	548	550	554
	559	562	584	601	612	632	634	636	638	647	654	665	674	693
	706	708	710	712	727	729	735	737	739	743	751	757	787	792
	797	803	810	812	815	821	843	845	850	865	872	876	879	885
	887	889	891	893	896	901	903	906	908	913	923	938	940	942
	945	948	951	955	960	981	985	993	997	1000	1005	1013	1015	1020
	1022	1027	1034	1043	1051	1053	1056	1060	1064	1066	1069	1079	1081	1091
	1099	1101	1104	1108	1112	1114	1122	1124	1131	1137	1139	1141	1156	1161
	1163	1170	1172	1178	1183	1189	1196	1214	1216	1239	1241	1243	1252	1256
	1260	1264	1271	1274	1280	1304	1333	1366	1500	1505	1517	1523	1547	1549
	1551	1554	1557	1563	1565	1568	1570	1572	1575	1578	1582	1585	1598	1607
	1619	1623	1626	1634	1637	1639	1641	1644	1649	1655	1663	1665	1668	1683
	1731	1733	1742	1748	1778	1781	1788	1790	1806	1809	1811	1815	1817	1826
	1832	1839	1847	1854	1861	1868	1875	1882	1885	1917	1922	1927	1931	1934
	1939	1943	1946	1949	1953	1955	1967	1979	1986	1990	1992	2003	2009	2017
	2022	2024	2031	2047	2050	2055	2067	2072	2076	2079	2084	2127	2129	2132
	2140	2161	2180	2191	2195	2197	2199	2208	2210	2216	2223	2226	2231	2234
	2238	2244	2247	2251	2257	2260	2264	2270	2273	2277	2283	2286	2290	2296
	2299	2303	2309	2312	2316	2322	2326	2338	2343	2348	2353	2361	2365	2399
	2405	2407	2410	2414	2416	2421	2435	2441	2443	2446	2450	2452	2456	2466
	2470	2472	2474	2487	2491	2497	2499	2503	2508	2523	2525	2527	2529	2533
	2548	2554	2561	2563	2566	2568	2570	2576	2604	2608	2614	2616	2620	2624
	2627	2636	2638	2640	2642	2646	2668	2670	2683	2692	2695	2702	2708	2711
	2717	2726	2803	2805	2808	2816	2826	2836	2838	2843	2849	2856	2867	2876
	2898	2919	2928	2935	2942	2944	2946	2948	2950	2952	2956	2961	2986	2996
	2999	3005	3008	3011	3021	3031	3038	3063	3066	3072	3075	3078	3085	3089
	3093	3189	3191	3194	3200	3217	3219	3221	3225	3227	3229	3233	3237	3239
	3243	3245	3248	3250	3254	3256	3342	3384	3386	3389	3394	3414	3417	3420
	3424	3434	3459	3461	3479	3486	3488	3490	3492	3494	3496	3498	3500	3504
	3507	3510	3514	3518	3521	3524	3528	3531	3534	3537	3540	3543	3546	3550
	3552	3554	3557	3562	3564	3570	3572	3574	3577	3582	3584	3595	3600	3606
	3612	3618	3624	3630	3636	3642	3647	3651	3655	3668	3671	3674	3677	3687
	3695	3704	3724	3780	3782	3826	3835	3837	3840	3842	3847	3851	3857	3871
	3874	3882	3901	3920	3930	3941	3974	3976	3978	3999	4001	4012	4021	4042
	4046	4062	4067	4072	4075	4111	4114	4117	4120	4126	4129	4134	4153	4156
	4159	4162	4168	4170	4172	4184	4187	4189	4191	4194	4196	4200	4207	4237
	4240	4242	4245	4251	4254	4260	4278	4281	4284	4287	4293	4295	4297	4309
	4312	4314	4316	4319	4321	4325	4332	4346	4405	4519	4524			
LDX	215	221	225	229	242	248	255	262	276	349	392	395	421	428
	432	479	488	629	856	1241	1243	1251	1289	1351	1581	1648	1654	1661
	1671	1676	1681	1736	1745	1824	2016	2029	2040	2058	2065	2087	2138	2221

	2409	2445	2494	2547	2611	2660	2705	2813	2869	2872	2883	2886	2904	2907
	2985	3019	3030	3051	3088	3198	3391	3594	3686	3915	3919	3929	3933	4039
	4050	4052	4054	4056	4058	4061	4112	4154	4198	4238	4279	4323	4346	
LDY	222	226	230	243	249	256	263	276	349	376	393	396	422	429
	433	630	742	750	857	1239	1241	1243	1302	1597	1737	1747	1751	1764
	1825	1899	2151	2156	2171	2190	2222	2364	2383	2420	2455	2502	2532	2619
	2645	2848	3202	3211	3259	3392	3419	3465	3468	3471	3474	3556	3576	3850
	3856	3868	3918	3928	3934	3940	4040	4066	4346	4405				
LSR	402	403	2972	2973	2974	2975	3952	3953	3954	3955	4123	4124	4165	4166
	4248	4249	4290	4291										
NOP	192	340	341	342	479	2894	2895	2896	3791	3792	3793	3798	3799	3800
	3823	3824	3825	4346										
ORA	246	252	349	407	1190	1239	1552	1792	1820	2026	2073	2162	2937	2976
	2979	3251	4043											
PHA	276	479	982	1184	1219	1239	1933	1938	1948	1951	1954	2045	2357	2362
	2553	2575	2815	2871	2874	2971	3234	3693	3752	3950	3951	4346		
PHP	1373													
PLA	479	988	1004	1188	1199	1205	1222	1227	1239	1941	1942	1961	1963	1965
	1973	1975	1977	2060	2395	2397	2431	2433	2556	2578	2655	2819	2889	2891
	2911	2913	2978	3241	3699	3707	3765	3778	3957	3959	4346	4405		
PLP	1376													
ROL	1243	2686	2688	2690	3422									
RTS	276	349	443	762	800	882	915	964	967	1074	1084	1117	1127	1167
	1176	1191	1203	1208	1225	1229	1239	1241	1243	1265	1281	1306	1339	1359
	1368	1377	1503	1508	1515	1520	1525	1560	1628	1652	1659	1673	1679	1686
	1727	1794	1799	1804	1971	1983	2020	2033	2036	2043	2063	2093	2125	2183
	2202	2206	2427	2462	2476	2481	2512	2517	2543	2581	2593	2629	2632	2657
	2663	2674	2721	2727	2800	2829	2863	2901	2924	2981	2991	3023	3026	3040
	3056	3060	3096	3109	3117	3132	3142	3183	3187	3278	3291	3298	3309	3316
	3355	3362	3382	3438	3483	3568	3588	3643	3659	3663	3691	3701	3709	3712
	3716	3726	3732	3742	3748	3750	3772	3776	3796	3802	3805	3815	3819	3828
	3845	3854	3862	3865	3878	3888	3921	3931	3946	3960	3980	3996	4028	4073
	4081	4132	4146	4205	4212	4257	4273	4330	4337	4346	4405	4520	4525	
SBC	387	390	730	1243	1354	2709	2712	3505	3508	3511	3519	3522	3525	3669
	3672	3675	3678	3843	4346									
SEC	385	1207	1228	1239	1241	1243	1353	1803	2035	2205	2516	2631	2706	2862
	3025	3059	3116	3141	3277	3297	3315	3361	3482	3503	3517	3667	3711	3715
	3725	3749	3804	3864	4211	4336	4346							
SED	1332	1350												
SEI	218													
STA	220	276	354	356	363	365	373	375	388	391	452	467	471	486
	489	493	495	497	499	523	530	535	539	541	544	546	549	551
	555	613	614	615	616	707	709	711	713	759	774	778	781	784
	811	813	816	828	829	834	835	840	841	842	844	877	886	888
	890	892	894	895	897	927	930	933	936	939	941	943	946	949
	952	989	992	995	998	1001	1006	1014	1016	1017	1018	1019	1021	1023
	1044	1045	1047	1048	1052	1054	1058	1059	1062	1063	1065	1076	1080	1082
	1092	1093	1095	1096	1100	1102	1106	1107	1110	1111	1113	1119	1123	1125
	1136	1138	1140	1142	1162	1164	1171	1173	1200	1206	1215	1217	1239	1241
	1243	1331	1367	1375	1548	1550	1553	1555	1564	1566	1566	1573	1576	1579
	1638	1640	1642	1664	1684	1732	1734	1743	1789	1791	1807	1808	1810	1812
	1816	1818	1819	1932	1935	1936	1944	1947	1950	1962	1964	1966	1968	1974
	1976	1978	1980	1991	2004	2007	2023	2025	2041	2048	2049	2051	2052	2053
	2056	2061	2068	2077	2080	2080	2081	2082	2085	2088	2128	2130	2136	2192
	2196	2198	2200	2209	2211	2212	2396	2398	2400	2406	2408	2415	2417	2422

	2432	2434	2436	2442	2444	2451	2453	2457	2471	2473	2492	2509	2524	2526
	2528	2530	2534	2555	2562	2567	2569	2577	2598	2609	2625	2637	2639	2641
	2643	2647	2661	2682	2684	2694	2697	2701	2704	2710	2713	2804	2806	2837
	2839	2840	2868	2885	2888	2890	2892	2899	2906	2909	2912	2914	2920	2934
	2943	2945	2947	2949	2951	2953	3009	3039	3052	3076	3086	3090	3094	3190
	3192	3385	3387	3415	3418	3460	3462	3463	3487	3489	3491	3493	3495	3497
	3499	3501	3506	3509	3512	3515	3516	3520	3523	3526	3533	3536	3539	3541
	3544	3551	3553	3555	3558	3563	3565	3571	3573	3575	3578	3583	3585	3670
	3673	3676	3679	3705	3836	3838	3852	3869	3914	3924	4037	4047	4116	4119
	4122	4125	4127	4128	4158	4161	4164	4167	4169	4171	4190	4192	4195	4197
	4208	4244	4247	4250	4252	4253	4283	4286	4289	4292	4294	4296	4315	4317
	4320	4322	4333	4346	4405									
STX	1241	2030	2059	2412	2448	2505	2622	2958	2963	3020	3911	3925	3937	4346
STY	1241	2413	2449	2506	2623	2959	2964	3912	3926	3938				
TAX	404	479	491	521	537	1050	1098	1160	1198	1223	1279	1590	2145	2501
	2618	2820	2977	4065	4405	4518	4523							
TAY	1239	2720	2725	2980										
TXA	349	410	439	479	1201	1218	1356	1587	1616	2142	2699	2814	3263	4078
	4346	4405												
TXS	216	276												
TYA	441	1239												
.ADDR	283	285	287	289	291	293	295	297	299	301	303	305	307	309
	311	313	315	317	319	321	323	325	327	329	331	332	1402	1423
	1424	1425	1426	1427	1428	1430	1431	1432	1448	1449	1450	1451	1452	4983
	4984													
.ASCII	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400
	1401	3440	4988											
.ASCIZ	210	282	284	286	288	290	292	294	296	298	300	302	304	306
	308	310	312	314	316	318	320	322	324	326	328	330	4440	4441
	4442	4443	4444	4445	4446	4447	4448	4449	4450	4451	4452	4453	4454	4455
	4456	4457	4458	4459	4460	4461	4462	4463	4464	4465	4466	4468	4475	4476
	4477	4479	4480	4481	4482	4483	4484	4485	4486	4487	4491	4492	4493	4494
	4495	4496	4497	4498	4499	4500	4501							
.BLKB	176													
.BLKW	176													
.BYTE	201	203	205	209	211	999	1448	1449	1450	1451	1452	1556	1650	1656
	1662	1672	1677	1682	2731	2732	2733	2734	2735	2736	2737	2738	2739	2740
	4356	4357	4358	4359	4361	4362	4364	4365	4366	4368	4369	4370	4371	4373
	4374	4375	4377	4378	4379	4380	4381	4382	4383	4384	4385	4386	4388	4389
	4390	4391	4392	4393	4394	4395	4396	4397	4467	4536	4537	4538	4539	4540
	4541	4542	4543	4545	4546	4547	4548	4549	4550	4551	4552	4554	4555	4556
	4557	4558	4559	4560	4561	4563	4564	4565	4566	4567	4568	4569	4570	4572
	4573	4574	4575	4576	4577	4578	4579	4581	4582	4583	4584	4585	4586	4587
	4588	4590	4591	4592	4593	4594	4595	4596	4597	4599	4600	4601	4602	4603
	4604	4605	4606	4608	4609	4610	4611	4612	4613	4614	4615	4617	4618	4619
	4620	4621	4622	4623	4624	4626	4627	4628	4629	4630	4631	4632	4633	4635
	4636	4637	4638	4639	4640	4641	4642	4644	4645	4646	4647	4648	4649	4650
	4651	4653	4654	4655	4656	4657	4658	4659	4660	4662	4663	4664	4665	4666
	4667	4668	4669	4671	4672	4673	4674	4675	4676	4677	4678	4680	4681	4682
	4683	4684	4685	4686	4687	4689	4690	4691	4692	4693	4694	4695	4696	4698
	4699	4700	4701	4702	4703	4704	4705	4707	4708	4709	4710	4711	4712	4713
	4714	4716	4717	4718	4719	4720	4721	4722	4723	4725	4726	4727	4728	4729
	4730	4731	4732	4734	4735	4736	4737	4738	4739	4740	4741	4743	4744	4745
	4746	4747	4748	4749	4750	4752	4753	4754	4755	4756	4757	4758	4759	4761
	4762	4763	4764	4765	4766	4767	4768	4770	4771	4772	4773	4774	4775	4776

	4777	4779	4780	4781	4782	4783	4784	4785	4786	4788	4789	4790	4791	4792
	4793	4794	4795	4797	4798	4799	4800	4801	4802	4803	4804	4806	4807	4808
	4809	4810	4811	4812	4813	4815	4816	4817	4818	4819	4820	4821	4822	4831
	4832	4833	4834	4835	4836	4837	4838	4840	4841	4842	4843	4844	4845	4846
	4847	4849	4850	4851	4852	4853	4854	4855	4856	4858	4859	4860	4861	4862
	4863	4864	4865	4867	4868	4869	4870	4871	4872	4873	4874	4876	4877	4878
	4879	4880	4881	4882	4883	4885	4886	4887	4888	4889	4890	4891	4892	4894
	4895	4896	4897	4898	4899	4900	4901	4903	4904	4905	4906	4907	4908	4909
	4910	4912	4913	4914	4915	4916	4917	4918	4919	4921	4922	4923	4924	4925
	4926	4927	4928	4930	4931	4932	4933	4934	4935	4936	4937	4939	4940	4941
	4942	4943	4944	4945	4946	4948	4949	4950	4951	4952	4953	4954	4955	4957
	4958	4959	4960	4961	4962	4963	4964	4966	4967	4968	4969	4970	4971	4972
	4973	4995	4996	4997	4998	5000	5001							
.END	5155													
.ENDC	137	144	151	158										
.IFEQ	132	139	146	153										
.IFT	133	140	147	154										
.LIST	125	172	4976											
.MACRO	1404	1436												
.MCALL	171	176	275	348	478	976	1237	4345	4404					
.NLIST	178	4509												
.PAGE	169	212	527	807	1010	1086	1129	1181	1248	1286	1324	1434	1630	1729
	1801	2185	2393	2429	2464	2514	2545	2586	2666	2796	2831	2881	2983	3028
	3043	3185	3456	3591	3665	4005	4137	4175	4264	4300	4437	5057	5112	
.SBTTL	1	49	114	122	182	272	280	335	475	574	643	681	701	767
	918	973	1038	1153	1235	1362	1383	1456	1495	1528	1692	2096	2744	3099
	3120	3146	3281	3322	3373	3720	3736	3787	3831	3892	3907	4032	4084	4218
	4342	4349	4401	4472	4513	4532	4826	4979	5008					
.TITLE	2													
.WORD	1453	4504												

Errors detected: 0

*,DSKFTP=DSKFTP
 Run-time: 10 11 2 Seconds
 Core used: 18K

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

* * * L P T S P L R u n L o g * * *

20:31:07 LPDAT LPTSPL version 104(16650) Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit
20:31:07 LPDAT Job DSKFTP sequence #1423 on Printer 0 [LOCAL] at 11-Sep-82 20:31:07
20:31:12 LPMSG Starting File PS:<KSPROUL>DSKFTP.LST.1
20:41:44 LPMSG Finished File PS:<KSPROUL>DSKFTP.LST.1
20:41:44 LPEND Summary: 154 Pages of Output
20:41:44 LPEND 69 Disk Pages Read
20:41:44 LPEND 86.130 Seconds CPU Time Used

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job DSKFTP Req #908 for KSPROUL Date 11-Sep-82 20:41:44 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

Faint, illegible text covering the majority of the page, likely bleed-through from the reverse side of the document.

