

SECTION VI

PROGRAM LISTING

SECTION VI

PROGRAM LISTING

This section provides a complete source listing of the DOS program.

PAGE 003 DOC .SA:0

00630 00112
 00640 00113
 00650 00114
 00660 00115
 00670 00116
 00680 00117
 00690 00118
 00700 00119
 00710 00120
 00720 00121
 00730 00122
 00740 00123
 00750 00124
 00760 00125
 00770 00126
 00780 00127
 00790 00128
 00800 00129
 00810 00130
 00820 00131
 00830 00132
 00840 00133
 00850 00134
 00860 00135
 00870 00136
 00880 00137
 00890 00138
 00900 00139
 00910 00140
 00920 00141
 00930 00142
 00940 00143
 00950 00144
 00960 00145
 00970 00146
 00980 00147
 00990 00148
 01000 00149
 01010 00150
 01020 00151
 01030 00152
 01040 00153
 01050 00154
 01060 00155
 01070 00156
 01080 00157
 01090 00158
 01100 00159
 01110 00160
 01120 00161
 01130 00162
 01140 00163
 01150 00164
 01160 00165
 01170 00166
 01180 00167
 01190 00168
 01200 00169

 * I N S T R U C T I O N S F O R U S E

*
 *

* ERROR NUMBERS AND THEIR MEANING

* (THE EQUATES ARE USED SO THAT ERRORS CAN BE RESEARCHED USING XREF LIST)

* DEFINITIONS START WITH BASIC LINE NUMBER 256 IN DOS

0000	A	ERR0	EQU	0	256	NO ERRORS
0001	A	ERR1	EQU	1	257	I/O ERROR - DRIVE NOT READY
0002	A	ERR2	EQU	2	258	I/O ERROR - WRITE PROTECTED
0003	A	ERR3	EQU	3	259	I/O ERROR - WRITE FAULT
0004	A	ERR4	EQU	4	260	I/O ERROR - SEEK ERROR OR RECORD NOT FOUND
0005	A	ERR5	EQU	5	261	I/O ERROR - CRC ERROR
0006	A	ERR6	EQU	6	262	I/O ERROR - LOST DATA
0007	A	ERR7	EQU	7	263	I/O ERROR - UNDEFINED BIT 1
0008	A	ERR8	EQU	8	264	I/O ERROR - UNDEFINED BIT 0
0009	A	ERR9	EQU	9	265	REGISTER ARGUMENT INVALID
000A	A	ERR10	EQU	10	266	FILE'S DIRECTORY ENTRY NOT FOUND
000B	A	ERR11	EQU	11	267	DIRECTORY IS FULL
000C	A	ERR12	EQU	12	268	FILE WAS CREATED BY "OPEN" FUNCTION
000D	A	ERR13	EQU	13	269	FILE NOT CLOSED AFTER CHANGES
000E	A	ERR14	EQU	14	270	ATTEMPTING TO ACCESS AN UNOPENED FILE
000F	A	ERR15	EQU	15	271	ATTEMPT TO READ - READ PROTECTED
0010	A	ERR16	EQU	16	272	RBA OVERFLOW (EXCEEDS 3 BYTES - 16,777,216)
0011	A	ERR17	EQU	17	273	ACCESS BEYOND EOF - EXTENSION NOT ALLOWED
0012	A	ERR18	EQU	18	274	FAT REWRITE ERROR
0013	A	ERR19	EQU	19	275	ATTEMPT TO CLOSE UNOPENED FILE
0014	A	ERR20	EQU	20	276	CAN'T ACCESS RANDOMLY - REC SIZE IS ZERO!
0015	A	ERR21	EQU	21	277	ATTEMPT TO WRITE - WRITE PROTECTED
0016	A	ERR22	EQU	22	278	CAN'T EXTEND FILE - DISK CAPACITY EXCEEDED
0017	A	ERR23	EQU	23	279	ERROR WHILE LOADING OVERLAY - FUNCTION NOT PERFORMED
0018	A	ERR24	EQU	24	280	INSUFFICIENT PRINT SPACE ALLOCATED
0019	A	ERR25	EQU	25	281	I/O ERROR DURING BASIC LINE READ
001A	A	ERR26	EQU	26	282	PROGRAM'S LOAD ADDRESS IS TOO LOW
001B	A	ERR27	EQU	27	283	FIRST BYTE OF PROGRAM FILE NOT EQUAL TO ZERO
001C	A	ERR28	EQU	28	284	SPACE FOR BUFFERED KBD NOT BIG ENOUGH
001D	A	ERR29	EQU	29	285	NOT ENOUGH MEMORY
001E	A	ERR30	EQU	30	286	OUTPUT FILE ALREADY EXISTS
001F	A	ERR31	EQU	31	287	WRONG DISKETTE

*

* DISK DATA CONTROL BLOCK (DCB) FORMAT

PAGE 004 DOC .SA:0

DOS - INSTRUCTIONS

```

01210 00170
01220 00171
01230 00172
01240 00173
01250 00174
01260 00175
01270 00176
01280 00177
01290 00178
01300 00179
01310 00180
01320 00181
01330 00182
01340 00183
01350 00184
01360 00185
01370 00186
01380 00187
01390 00188
01400 00189
01410 00190
01420 00191
01430 00192
01440 00193
01450 00194
01460 00195
01470 00196
01480 00197
01490 00198
01500 00199
01510 00200
01520 00201
01530 00202
01540 00203
01550 00204
01560 00205
01570 00206
01580 00207
01590 00208
01600 00209
01610 00210
01620 00211
01630 00212
01640 00213
01650 00214
01660 00215
01670 00216
01680 00217
01690 00218
01700 00219
01710 00220
01720 00221
01730 00222
01740 00223
01750 00224
01760 00225
01770 00226
01780 00227

```

```

*****
*
* BYTES CONTENTS
* -----
* THESE ITEMS ARE A COPY OF DISK DIRECTORY ENTRY
* 0-7  FILENAME
* 8-10  FILE EXTENSION
* 11  FILE TYPE
*      (0=BASIC PGM,1=BASIC DATA,2=MACHINE LANG. PGM,3=TEXT ED. SOURCE)
* 12  ASCII FLAG (0=BINARY, FF = ASCII FILE)
* 13  NUMBER OF FIRST CLUSTER IN FILE
* 14-15 NUMBER OF BYTES IN USE IN LAST SECTOR OF FILE
* THESE ITEMS WERE ADDED, USING LAST 16 BYTES OF DIRECTORY ENTRY
* 16  CURRENT FILE STATUS
*      BIT 0 ON ALLOWS READS
*      BIT 1 ON ALLOWS WRITES
*      BIT 2 ON ALLOWS FILE CREATE IF NON-EXISTANT
*      BIT 3 ON ALLOWS FILE EXTENSION BEYOND EOF ON ACCESS ATTEMPTS
*      BIT 4 ON MEANS WORK FILE - DELETE FILE WHEN CLOSED
*      BIT 5 ON PREVENTS REWRITE OF FAT EVERY TIME A SECTOR IS ADDED TO
*              THE FILE. (MINOR POWER FAILURE INCONSISTANCY COULD RESULT)
*      BIT 6 ON MEANS I/O BUFFER IS SHARED. EACH LOGICAL I/O REQUIRES
*              A PHYSICAL I/O
*      BIT 7 RESERVED FOR FUTURE OPTION(LIKE RELEASE SPACE WHEN FILE SHORTENED)
*      (ALL BITS OFF = FILE CLOSED)
* 17-18 LOGICAL RECORD SIZE (AS OF LAST TIME FILE WAS CLOSED)
*      ZERO MEANS VARIABLE LENGTH WITH RECORDS TERMINATED BY THE
*              DELIMITER STORED BELOW.
*      $FFFF MEANS VARIABLE LENGTH WITH FIRST TWO BYTES OF RECORD
*              CONTAINING SIZE OF THE REST OF THE RECORD.
*      ALL OTHER VALUES MEAN FIXED LENGTH OF SPECIFIED SIZE.
* 19  VARIABLE LENGTH RECORD TERMINATOR
* 20-31 AT PRESENT, UNUSED PART OF DIRECTORY ENTRY - USE WITH CAUTION.
*
* THESE ITEMS ARE USED FOR PHYSICAL I/O PARAMETERS
* 32  LAST I/O OPCODE
* 33  LAST I/O DRIVE
* 34  LAST I/O TRACK
* 35  LAST I/O SECTOR
* 36-37 LAST I/O BUFFER POINTER
* 38  LAST I/O RESULT CODE
*
* THESE ITEMS ARE FOR LOGICAL USE
* 39-40 LOGICAL RECORD BUFFER (CAN BE SAME AS DCBBUF IF DCBRSZ=256)
* 41-42 LAST I/O PHYSICAL RECORD NUMBER (BEFORE XLATE INTO SECTOR WITHIN
*      GRANULE). THIS IS THE RECORD CURRENTLY IN THE BUFFER.
* 43-45 CURRENT RELATIVE BYTE ADDRESS (RBA) OF FILE DATA POINTER
* 46-47 CURRENT LOGICAL RECORD NUMBER
* 48  MODIFIED DATA TAG - SET NON-ZERO WHEN BUFFER CONTENTS CHANGED
*
* EQUATES FOLLOW FOR MEANINGFUL SOURCE CODE WHEN ACCESSING DCB
* IE: STD DCBLRN,U SAVE NEW LOGICAL RECORD NUMBER
*      (BETTER THAN STD 46,U )
0000  A DCBFNM EQU 0      FILE NAME
0008  A DCBFEX EQU 8     FILE NAME EXTENSION
000B  A DCBFTY EQU 11    FILE TYPE
000C  A DCBASC EQU 12    ASCII CODE
000D  A DCBFCL EQU 13    FIRST CLUSTER NUMBER

```

```

PAGE 005 DOC .SA:0 DOS - INSTRUCTIONS

01790 00228 000E A DCBNLS EQU 14 NUMBER OF BYTES USED IN LAST SECTOR
01800 00229 0010 A DCBCFS EQU 16 CURRENT FILE STATUS
01810 00230 0011 A DCBRSZ EQU 17 RECORD SIZE
01820 00231 0013 A DCBTRM EQU 19 VAR LEN RECORD TERMINATOR
01830 00232 0014 A DCBMRB EQU 20 MAX RBA
01840 00233 0017 A DCBUSR EQU 23 USER AREA
01850 00234 0020 A DCBOPC EQU 32 OPERATION CODE
01860 00235 0021 A DCBDRV EQU 33 DRIVE
01870 00236 0022 A DCBTRK EQU 34 TRACK
01880 00237 0023 A DCBSEC EQU 35 SECTOR
01890 00238 0024 A DCBBUF EQU 36 I/O BUFFER ADDRESS
01900 00239 0026 A DCBOK EQU 38 I/O RESULT CODE
01910 00240 0027 A DCBLRB EQU 39 LOGICAL RECORD BUFFER ADDRESS
01920 00241 0029 A DCBPRN EQU 41 PHYSICAL RECORD NUMBER IN BUFFER
01930 00242 002B A DCBRBA EQU 43 CURRENT RELATIVE BYTE ADDRESS
01940 00243 002E A DCBLRN EQU 46 CURRENT LOGICAL RECORD NUMBER
01950 00244 0030 A DCBMDT EQU 48 MODIFIED DATA TAG
01960 00245 0031 A DCBSZ EQU DCBMDT+1 SIZE OF DCB (CURRENTLY 50 BYTES)
01970 00246 *
01980 00247 *****
01990 00248 * EQUATES TO SUPPORT ROUTINES IN ROM OPERATING SYSTEM
02000 00249 *****
02010 00250 A000 A POLCAT EQU $A000
02020 00251 A012 A ROLTAB EQU $152 KBD ROLLOVER TABLE
02030 00252 A00A A JOYIN EQU $A00A
02040 00253 A006 A BLKIN EQU $A006
02050 00254 A004 A CSRDN EQU $A004
02060 00255 A00C A WRTLDR EQU $A00C
02070 00256 A008 A BLKOUT EQU $A008
02080 00257 007C A BLKTYP EQU $7C
02090 00258 007D A BLKLEN EQU $7D
02100 00259 007E A CBUFAD EQU $7E
02110 00260 010C A IRG EQU $10C
02120 00261 015A A PQTS EQU $15A JOYSTICK POT VALUES
02130 00262 011A A ALPHLK EQU $11A KBD RTN'S ALPHA LOCK SWITCH
02140 00263 *
02150 00264 *****
02160 00265 * EQUATES TO XREF USE OF PIA'S
02170 00266 *****
02180 00267 FF21 A U4ACR EQU $FF21 CONTROL REG
02190 00268 FF20 A U4ADR EQU $FF20 DATA REG
02200 00269 FF20 A U4ADD EQU $FF20 DATA DIRECTION REG
02210 00270 FF23 A U4BCR EQU $FF23
02220 00271 FF22 A U4BDR EQU $FF22
02230 00272 FF22 A U4BDD EQU $FF22
02240 00273 FF01 A UBACR EQU $FF01
02250 00274 FF00 A UBADR EQU $FF00
02260 00275 FF00 A UBADD EQU $FF00
02270 00276 FF03 A UBBCR EQU $FF03
02280 00277 FF02 A UBBDR EQU $FF02
02290 00278 FF02 A UBBDD EQU $FF02
02300 00279 *
02310 00280 * MISC ADDITIONAL EQUATES
02320 00281 0035 A ENABLE EQU %00110101
02330 00282 0034 A DSABLE EQU %00110100
02340 00283 * COLOR VALUES
02350 00284 0000 A BUFF EQU %00000000
02360 00285 0055 A CYAN EQU %01010101

```

```

PAGE 006 DOC .SA:0 DOS - INSTRUCTIONS

02370 00286 00AA A MGNTA EQU %10101010
02380 00287 00FF A ORANGE EQU %11111111
02390 00288 0000 A GREEN EQU %00000000
02400 00289 0055 A YELLOW EQU %01010101
02410 00290 00AA A BLUE EQU %10101010
02420 00291 00FF A RED EQU %11111111
02430 00292 * CODES RETURNED BY POLCAT FOR FUNCTION KEYS
02440 00293 005E A UP EQU $5E UP ARROW
02450 00294 000A A DOWN EQU $0A DOWN ARROW
02460 00295 0009 A RIGHT EQU $09 RIGHT ARROW
02470 00296 0008 A LEFT EQU $08 LEFT ARROW
02480 00297 005F A SUP EQU $5F SHIFT UP ARROW
02490 00298 005B A SDOWN EQU $5B SHIFT DOWN ARROW
02500 00299 005D A SRIGHT EQU $5D SHIFT RIGHT ARROW
02510 00300 0015 A SLEFT EQU $15 SHIFT LEFT ARROW
02520 00301 0003 A BREAK EQU $03 BREAK KEY
02530 00302 000C A CLEAR EQU $0C CLEAR KEY
02540 00303 005C A SCLEAR EQU $5C SHIFTED CLEAR
02550 00304 000D A ENTER EQU $0D ENTER KEY
02560 00305 0040 A AT EQU $40 "a" KEY
02570 00306 0013 A SAT EQU $13 SHIFTED "a" KEY
02580 00307 *
02590 00308 *****
02600 00309 * DOS MACRO AND LOGICAL EQUATES
02610 00310 *****
02620 00311 DOS MACR CALL A DOS FUNCTION
00312 2630 LDA #\1 OPTION
00313 2640 JSR [\0] INDIRECT FUNCTION ADDR
00314 2650 ENDM
02660 00315 *
02670 00316 * EQUATES USED WITH DOS MACRO
02680 00317 *
02690 00318 * THE FOLLOWING USED WITH "OPEN"
02700 00319 0600 A OPEN EQU $600 OPEN FUNCTION
02710 00320 0004 A CREATE EQU 4 ALLOWS FILE CREATION ON OPEN IF NOT FOUND
02720 00321 0008 A EXTEND EQU 8 ALLOWS EXTENSION OF FILE TO POINT OF ACCESS
02730 00322 0001 A INPUT EQU 1 USED TO SIGNIFY THAT READS ARE ALLOWED
02740 00323 0001 A IN EQU 1 SHORTER FORM OF ABOVE
02750 00324 0002 A OUT EQU 2 ALLOWS WRITES
02760 00325 000E A OUTPUT EQU CREATE+EXTEND+OUT USUAL COMBINATION FOR OUTPUT FILES
02770 00326 0010 A WORK EQU 16 CAUSES FILE TO BE KILLED WHEN CLOSED (WORK FILE)
02780 00327 0020 A FAST EQU 32 MINIMIZES FAT REWRITES
02790 00328 0040 A SHARE EQU 64 USED WHEN 2 OR MORE FILES SHARE THE SAME I/O BUFFER
02800 00329 * EXAMPLES:
02810 00330 * DOS OPEN INPUT TO READ AN EXISTING FILE
02820 00331 * DOS OPEN OUTPUT TO CREATE & EXTEND AN OUTPUT FILE
02830 00332 * DOS OPEN IN+OUT TO UPDATE AN EXISTING FILE (NO EXTENSIONS)
02840 00333 * DOS OPEN INPUT+OUTPUT+WORK TO CREATE, EXTEND, READ & WRITE AND KILL
02850 00334 * WHEN CLOSED (A WORK FILE)
02860 00335 * "SHARE" CAN BE ADDED TO ANY OF THE ABOVE EXAMPLES IF 2 OR MORE FILES
02870 00336 * WILL BE USING THE SAME I/O BUFFER AT THE SAME TIME. THIS OPTION CAUSES
02880 00337 * A PHYSICAL I/O TO REFRESH THE BUFFER WITH EVERY LOGICAL I/O OPERATION.
02890 00338 * WITHOUT THIS OPTION, SEVERAL LOGICAL READS OR WRITES TO OR FROM THE
02900 00339 * SAME PHYSICAL SECTOR CAN BE DONE WITH A SINGLE PHYSICAL I/O. "SHARE"
02910 00340 * INCREASES THE AMOUNT OF ACTUAL I/O ACTIVITY, BUT ALLOWS USE OF MANY
02920 00341 * FILES AT THE SAME TIME WITH MUCH LESS MEMORY REQUIREMENTS FOR BUFFERS.
02930 00342 *
02940 00343 * USED WITH "CLOSE" FUNCTION

```

```

PAGE 007 DOC .SA:0 DOS - INSTRUCTIONS

02950 00344 0602 A CLOSE EQU $602 CLOSE A FILE OPTIONS NOT USED
02960 00345 0000 A IT EQU 0
02970 00346 * EXAMPLE:
02980 00347 * DOS CLOSE,IT TO CLOSE A FILE
02990 00348 *
03000 00349 * USED WITH "READ" AND "WRITE" FUNCTIONS
03010 00350 0604 A READ EQU $604 READ A RECORD
03020 00351 0606 A WRITE EQU $606 WRITE A RECORD
03030 00352 0001 A RBA EQU 1 TO READ USING REL BYTE ADDR
03040 00353 0000 A RECORD EQU 0
03050 00354 0000 A REC EQU 0
03060 00355 0002 A UPDATE EQU 2 TO PREVENT ADVANCING REC NBR OR RBA AFTER A READ
03070 00356 0008 A NOW EQU 8 1 = ENSURE I/O BUFFER IS WRITTEN TO DISK AFTER LOGICAL WRITE
03080 00357 * EXAMPLES:
03090 00358 * DOS READ,RECORD TO RANDOMLY READ BY RECORD NUMBER
03100 00359 * (FIXED LENGTH RECS ONLY)
03110 00360 * (USE THIS FOR NORMAL SEQUENTIAL READ OF FIXED LENGTH)
03120 00361 * DOS READ,RBA TO READ THE RECORD POINTED AT BY RBA
03130 00362 * (REQUIRED IF USING VARIABLE LENGTH RECORDS)
03140 00363 * DOS READ,UPDATE TO READ BY REC NBR WITHOUT ADVANCING REC NBR
03150 00364 * DOS READ,RBA+UPDATE TO READ THE RECORD POINTED AT BY RBA & NOT CHANGE RBA
03160 00365 * DOS WRITE,REC WRITE VIA RECORD NUMBER (FIXED LENGTH ONLY)
03170 00366 * DOS WRITE,RBA WRITE FIXED OR VARIABLE RECORD
03180 00367 * DOS WRITE,UPDATE UNLIKELY OPTION - WRITES RECORD BUT DOES NOT CHANGE
03190 00368 * RBA OR REC NUMBER. COULD BE REWRITTEN AGAIN.
03200 00369 * DOS WRITE,RBA+NOW SAME AS: DOS WRITE,RBA FOLLOWED BY DOS RELSE,IT
03210 00370 *
03220 00371 0608 A RELSE EQU $608 USE TO RELEASE I/O BUFFER WITHOUT CLOSING FILE
03230 00372 * IF CONTENTS OF BUFFER HAVE BEEN CHANGED, IT IS REWRITTEN. THEN DCBPN
03240 00373 * IS SET TO $FFFF TO ENSURE A PHYSICAL I/O BEFORE THE NEXT LOGICAL I/O.
03250 00374 * USE THIS FUNCTION WHEN USER IS CONTROLLING A SHARED BUFFER.
03260 00375 * EXAMPLE:
03270 00376 * DOS RELSE,IT
03280 00377 *
03290 00378 * USED WITH OVERLAYABLE FUNCTIONS
03300 00379 060A A DO EQU $60A USE TO LOAD IF NECESSARY, THEN EXECUTE AN OVERLAY
03310 00380 060C A GO EQU $60C USE TO XFER CONTROL FROM ONE OVERLAY TO ANOTHER IN SAME AREA
03320 00381 060E A LOAD EQU $60E USE TO LOAD A SYSTEM OVERLAY - IT IS LOADED AT THE
03330 00382 * EXAMPLE:
03340 00383 * DOS DO,MAP
03350 00384 *
03360 00385 * THE FOLLOWING USED WITH "LOAD" AND "DO" FUNCTIONS
03370 00386 0001 A INIT EQU 1 INITIALIZATION OF DOS
03380 00387 * EXAMPLE:
03390 00388 * DOS DO,INIT EXIT PROGRAM & RE-INITIALIZE DOS
03400 00389 * NOTE: STACK AND OLYLOC SHOULD BE RESET BEFORE USING THIS OVERLAY
03410 00390 *
03420 00391 000E A MENU EQU 14 DISPLAY DOS MAIN MENU
03430 00392 * EXAMPLE:
03440 00393 * LDS #STACK
03450 00394 * LDD #OVRLAY WHERE OVERLAY AREA SHOULD START
03460 00395 * STD >OLYLOC
03470 00396 * DOS DO,MENU
03480 00397 *
03490 00398 000A A MAP EQU 10 DISPLAY BASIC LINES
03500 00399 * EXAMPLE:
03510 00400 * LDD #280 FIRST LINE NUMBER TO BE DISPLAYED
03520 00401 * LDY #283 LAST LINE TO BE DISPLAYED

```

```

PAGE 008 DOC .SA:0 DOS - INSTRUCTIONS
03530 00402 * LDU <CURSOR STARTING DISPLAY ADDRESS
03540 00403 * (IF STARTING ADDR IS ZERO, SCREEN WILL BE CLEARED FIRST AND ROUTINE
03550 00404 * WILL EXIT WITH U->FIRST CHAR AFTER FIRST LEFT BRACKET ON SCREEN)
03560 00405 * PSHS D,Y,U (PARAMETERS ARE PASSED IN THE STACK)
03570 00406 * DOS DO,BASMSG
03580 00407 * PULS D,Y,U NORMALIZE STACK
03590 00408 * BNE ERROR BRANCH ON ANY FAILURE IF DESIRED
03600 00409 *
03610 00410 0002 A RUNIP EQU 2 KEYIN A NAME AND RUN PGM
03620 00411 * EXAMPLE:
03630 00412 * DOS DO,RUNIP
03640 00413 *
03650 00414 0005 A CPYFLE EQU 5 GET INFO FROM USER & COPY A FILE
03660 00415 * EXAMPLE:
03670 00416 * DOS DO,CPYFLE (IF "GO" USED, DOS MENU FOLLOWS COPY FUNCTION)
03680 00417 *
03690 00418 000B A FIELDI EQU 11 INPUT A MAPPED FIELD
03700 00419 * EXAMPLE:
03710 00420 * LDY DEST WHERE THE DATA GOES IN MEMORY
03720 00421 * LDU FLDADR POINT TO FIELD ON SCREEN
03730 00422 * DOS DO,FIELDI INPUTS THE FIELD
03740 00423 * B IS RETURNED CONTAINING LAST KEYSTROKE ENTERED
03750 00424 *
03760 00425 000C A EXEC EQU 12 GIVEN USRDCB CONTENTS, LOAD ROOT & EXECUTE PROGRAM
03770 00426 * EXAMPLE:
03780 00427 * (WHATEVER LOGIC TO PUT NAME IN DCB AT "USRDCB")
03790 00428 * DOS GO,EXEC JUMP TO LOAD & EXECUTE OVERLAY
03800 00429 *
03810 00430 000D A REALTM EQU 13 CLOCK DISPLAY OVERLAY (SEE SKEL FOR EXAMPLE OF USE)
03820 00431 *
03830 00432 000F A BUFPRN EQU 15 BUFFERED PRINT OVERLAY
03840 00433 * EXAMPLE:
03850 00434 * LDU #SIZE (TOTAL MEMORY TO BE USED (ROUTINE + BUFFER)
03860 00435 * (ROUTINE IS ABOUT 320 BYTES)
03870 00436 * DOS DO,BUFPRN (SETS IT UP - OVERLAY & BUFFER PROTECTED FROM
03880 00437 * BEING OVERLAYED).
03890 00438 * FROM THIS POINT ON, CHARACTERS PRINTED BY CALLING "PRNT" WILL GO
03900 00439 * THROUGH BUFFERED I/O. TO WRAP UP AT E0J, DO THIS:
03910 00440 * CLRA
03920 00441 * JSR [PRNT] REQUEST TO END BUFFERING.
03930 00442 * THIS WILL CAUSE "PRNT" TO WAIT UNTIL THE BUFFER IS EMPTIED (PRINTER
03940 00443 * HAS CAUGHT UP), AND THEN OVERLAY AND BUFFER AREA ARE RELEASED.
03950 00444 *
03960 00445 0011 A COPY EQU 17 COPY A FILE
03970 00446 * GIVEN:
03980 00447 * U->SOURCE FILE DCB (NOT OPENED)
03990 00448 * Y->DEST FILE DCB (NOT OPENED)
04000 00449 * B (BIT 0) - OFF IF NO DISKETTE SWAPPING, ON FOR DISKETTE SWAPPING
04010 00450 * RETURNED A=ERROR NUMBER
04020 00451 *
04030 00452 * SIMILAR FUNCTIONS FOR USING USER OVERLAYS
04040 00453 0610 A DUSR EQU $610 LOAD IF NECESSARY & EXECUTE USER OVERLAY
04050 00454 0612 A GUSR EQU $612 JUMP TO A DIFFERENT OVERLAY
04060 00455 0614 A LODUSR EQU $614 LOAD USER OVERLAY
04070 00456 * USER SHOULD PROVIDE EQUATES FOR HIS OVERLAYS HERE
04080 00457 *
04090 00458 0616 A ERROR EQU $616 JSR HERE FOR DISPLAY OF ERR MSG
04100 00459 *

```

```

PAGE 009 DOC .SA:0 DOS - INSTRUCTIONS
04110 00460 061B A TIME EQU $61B TURN ON/OFF TIME ROUTINE
04120 00461 0001 A ON EQU 1
04130 00462 0000 A OFF EQU 0
04140 00463 * EXAMPLE:
04150 00464 * LDU #TMERTN LOAD ADDR OF ROUTINE
04160 00465 * DOS TIME,ON GO ACTIVATE THIS ROUTINE
04170 00466 *
04180 00467 061A A PRNT EQU $61A PRINT A CHARACTER ON PRINTER
04190 00468 * THIS IS CHANGED BY CALLING BUFFERED PRINTER OVERLAY TO POINT
04200 00469 * AT BUFFERED IO ROUTINE
04210 00470 *
04220 00471 061C A KEYIN EQU $61C POLL KEYBOARD FOR INPUT CHARACTER
04230 00472 * THIS IS CHANGED BY CALLING BUFFERED KEYBOARD OVERLAY TO POINT
04240 00473 * AT BUFFERED IO ROUTINE
04250 00474 *
04260 00475 061E A BASIC EQU $61E JMP HERE TO RETURN TO BASIC
04270 00476 *
04280 00477 *****
04290 00478 * O T H E R U S E F U L M A C R O S F O L L O W
04300 00479 *****
04310 00480 ENABLI MACR ENABLE INTERRUPTS
04320 00481 4320 ANDCC #%11101111
04330 00482 4330 ENDM
04340 00483 *
04350 00484 DSABLI MACR DISABLE INTERRUPTS
04360 00485 4360 ORCC #%01010000
04370 00486 4370 ENDM
04380 00487 *
04390 00488 NEGD MACR NEGATE D
04400 00489 4400 COMA
04410 00490 4410 COMB
04420 00491 4420 ADDD #1
04430 00492 4430 ENDM
04440 00493 *
04450 00494 LSRD MACR LOGICAL SHIFT RIGHT D
04460 00495 4460 LSRA
04470 00496 4470 RORB
04480 00497 4480 ENDM
04490 00498 *
04500 00499 LSLD MACR LOGICAL SHIFT LEFT D
04510 00500 4510 LSLB
04520 00501 4520 ROLA
04530 00502 4530 ENDM
04540 00503 *
04550 00504 CLRD MACR CLEAR D
04560 00505 4560 CLRA
04570 00506 4570 CLRB
04580 00507 4580 ENDM
04590 00508 *
04600 00509 INCD MACR ADD 1 TO D
04610 00510 4610 ADDD #1
04620 00511 4620 ENDM
04630 00512 *
04640 00513 *****
04650 00514 * S Y S T E M R A M - D O S
04660 00515 *****
04670 00516 * ADDITIONAL WS USING EXTENDED ADDRESSING
04680 00517A 015E ORG $15E

```

```

PAGE 010 DOC .SA:0 DOS - INSTRUCTIONS
04690 00518A 0600          ORG      $600
04700 00519          * AREA WHERE USER ACCESSABLE VECTORS & VARIABLES STORED
04710 00520A 0600      0020  A VECTOR RMB 2*16  2 BYTES PER VECTOR
04720 00521          * OPEN   OPEN A DISK FILE
04730 00522          * CLOSE  CLOSE A DISK FILE
04740 00523          * READ   READ FROM A DISK FILE
04750 00524          * WRITE  WRITE TO A DISK FILE
04760 00525          * RELSE  RELEASE I/O BUFFER (ALLOW USE FOR ANOTHER FILE)
04770 00526          * DO     LOAD & EXECUTE A SYSTEM OVERLAY
04780 00527          * GO     LOAD ON TOP OF CURRENT OVERLAY & JUMP TO SYSTEM OVERLAY
04790 00528          * LOAD   LOAD SYSTEM OVERLAY
04800 00529          * DOUSR  LOAD & EXECUTE USER OVERLAY
04810 00530          * GOUSR  LOAD ON TOP OF CURRENT OVERLAY & JUMP TO USER OVERLAY
04820 00531          * LODUSR  LOAD USER OVERLAY
04830 00532          * ERROR  DISPLAY ERROR NUMBER IN "A"
04840 00533          * TIME  TURN ON/OFF TIME INTERVAL ROUTINE
04850 00534          * PRNT  PRINT A CHARACTER ON PRINTER
04860 00535          * KEYIN  INPUT NEXT KEYSTROKE FROM KEYBOARD
04870 00536          * BASIC  RETURN TO BASIC CONTROL
04880 00537A 0620      0002  A CLOCK  RMB  2      COUNT OF 60THS OF A SECOND
04890 00538A 0622      0001  A RETRYS RMB  1      NUMBER OF I/O RETRYS INITIALLY SET TO 5
04900 00539A 0623      0002  A RATE  RMB  2      TIME CONSTANT THAT CONTROLS PRINTER TRANSMISSION SPEED
04910 00540A 0625      0002  A OLYLOC RMB  2      ADDRESS WHERE CURRENT OVERLAY WAS LOADED
04920 00541A 0627      0002  A USRBSE RMB  2      BASE OF USER'S ROOT + 1. POINTS TO ENTRY ZERO OF OVERLAY'S RBA*
04930 00542A 0629      0002  A HOOK1  RMB  2      JUST BEFORE CHECKING FOR AUTO EXECUTE
04940 00543A 062B      0002  A HOOK2  RMB  2      JUST BEFORE BRANCHING TO USER PROGRAM
04950 00544A 062D      0002  A HOOK3  RMB  2
04960 00545A 062F      0002  A HOOK4  RMB  2
04970 00546A 0631      0002  A HOOK5  RMB  2
04980 00547A 0633      0002  A RETURN RMB  2      CONTAINS TWO RTS CODES - ALL HOOKS RETURN THRU HERE
04990 00548A 0635      0031  A DOSDCB RMB  DCBSZ  DCB USED TO READ SYSTEM OVERLAYS
05000 00549A 0666      0031  A MSGDCB RMB  DCBSZ  DCB USED TO READ "MAPS" AND MESSAGES
05010 00550A 0697      0031  A USRDCB RMB  DCBSZ  DCB USED TO READ USER'S PROGRAM & OVERLAYS
05020 00551A 06C8      0100  A SYSBUF RMB  256    BUFFER FOR SYSTEM USE(DIRECTORY + FAT READS & WRITES)
05030 00552          A FATSZ  EQU  69    FILE ALLOCATION TABLE (FAT) SIZE
05040 00553A 07C8      0045  A FAT0  RMB  FATSZ  SAVE AREA FOR DRIVE 0 FAT TABLE
05050 00554A 080D      0045  A FAT1  RMB  FATSZ  SAME FOR DRIVE 1
05060 00555A 0852      0045  A FAT2  RMB  FATSZ
05070 00556A 0897      0045  A FAT3  RMB  FATSZ
05080 00557          A FATS  EQU  FAT0
05090 00558A 08DC      0002  A MAXMEM RMB  2      ADDR OF HIGHEST USEABLE MEMORY
05100 00559A 08DE      0001  A DRIVES RMB  1      MAX NBR OF DRIVES TO SEARCH ON GLOBAL OPEN
05110 00560          OPT    L
05120 00561A 08DF      0001  A ENDWSE RMB  1      END OF EXTENDED WS
05130 00562          OPT    NOL
05140 00563          *****
05150 00564          * D O S      S T A R T S   H E R E
05160 00565          *****
05170 00566A 0989          ORG      ORGIN  SEE 1ST MODULE FOR VALUE ASSIGNED
00010 00567          OPT    L
00020 00568          TTL   DOS - I/O ROUTINES
00030 00569          OPT    NOL
00040 00570          *****
00050 00571          *       O P E N   D I S K   F I L E
00060 00572          *
00070 00573          * GIVEN:
00080 00574          * A=DESIRED FILE STATUS
00090 00575          * U->DCB

```

```

00100 00576      * DCBDRV,U = DRIVE TO BE CHECKED ($FF=CHECK ALL DRIVES)
00110 00577      *
00120 00578      * BEFORE CALLING "OPEN", DCB SHOULD CONTAIN: FILENAME, EXTENSION,
00130 00579      * I/O BUFFER ADDRESS, NAME AND EXTENSION ONLY ARE COMPARED
00140 00580      * TO DIRECTORY ENTRIES TO FIND MATCH. TYPE AND ASCII FLAG ARE USED ONLY
00150 00581      * WHEN CREATING FILE (OTHERWISE THEY ARE OVERLAYED BY EXISTING VALUES).
00160 00582      * ALL I/O NEEDED TO OPEN FILE USES THE 256 BYTE AREA POINTED TO BY
00170 00583      * LAST I/O ADDRESS AS A BUFFER.
00180 00584      *
00190 00585      * OPEN WORKS EXACTLY THE SAME FOR INPUT OR OUTPUT! ACTION IS CONTROLLED
00200 00586      * BY FILE STATUS SUPPLIED IN "A" (SEE DCBCFS IN DCB DESCRIPTION).
00210 00587      *
00220 00588      * OPENING A NON-EXISTANT FILE - IF CREATION IS ALLOWED, FIRST 32 BYTES OF
00230 00589      * DCB ARE PLACED IN DIRECTORY EXCEPT THAT DCBFCL IS SET TO $FF, DCBNLS
00240 00590      * IS SET TO ZERO AND DCBCFS IS SET TO PROVIDED STATUS.
00250 00591      *
00260 00592      * OPENING AN EXISTING FILE - THE 32 BYTE DIRECTORY ENTRY OVERLAYS THE
00270 00593      * FIRST 32 BYTES OF THE DCB EXCEPT FOR DCBCFS WHICH IS SET TO THE PROVIDED
00280 00594      * VALUE.
00290 00595      *
00300 00596      * WHEN FILE IS OPENED, DCBPRN IS SET TO $FFFF (AN INVALID VALUE), DCBRBA
00310 00597      * IS SET TO ZERO, AND DCBLRN IS SET TO ZERO. AT ANYTIME BEFORE OR AFTER
00320 00598      * CALLING OPEN, DCBLRB CAN BE SET OR CHANGED.
00330 00599      *
00340 00600      * FILE TYPE AND ASCII FLAG CAN BE CHANGED AFTER OPEN TO CAUSE THEM TO BE
00350 00601      * CHANGED WHEN FILE IS CLOSED.
00360 00602      *
00370 00603      *
00380 00604A 0989 E6 CB 21 A DOPEN LDB DCBDRV,U
00390 00605A 098C 34 16 A PSHS D,X
00400 00606A 098E C1 FF A CMPB #$FF REQUEST FOR SCAN OF ALL DRIVES
00410 00607A 0990 27 0A 099C BEQ D00 IF YES
00420 00608A 0992 C1 04 A CMPB #4 VALID DRIVE REQUESTED?
00430 00609A 0994 25 07 099D BCS D01 IF YES
00440 00610A 0996 B6 09 A LDA #ERR9 PARAMETER ERROR
00450 00611A 0998 A7 E4 A DOERR STA ,S
00460 00612A 099A 35 '96 A PULS D,X,PC RETURN WITH ERROR CONDITION
00470 00613A 099C 5F D00 CLRB START WITH DRIVE ZERO
00480 00614A 099D E7 CB 21 A D01 STB DCBDRV,U
00490 00615A 09A0 4F CLRA SAY LOOK FOR MATCH
00500 00616A 09A1 17 02D2 0C76 LBSR CHKDIR CHECK DIRECTORY ON THIS DRIVE FOR MATCH
00510 00617A 09A4 27 50 09F6 BEQ D05 IF MATCH FOUND
00520 00618A 09A6 2B 0B 09B0 BMI D03 IF NO I/O ERRORS - JUST DIDNT FIND IT
00530 00619      * IT WAS SOME KIND OF I/O ERROR
00540 00620A 09A8 B1 01 A CMPA #1 DRIVE NOT READY?
00550 00621A 09AA 26 EC 0998 BNE DOERR IF NO
00560 00622A 09AC 6D 61 A TST 1,S REQUEST FOR SPECIFIC DRIVE?
00570 00623A 09AE 2A E8 099B BPL DOERR IF YES, THEN THIS IS AN ERROR
00580 00624A 09B0 6D 61 A D03 TST 1,S REQUEST FOR SPECIFIC DRIVE?
00590 00625A 09B2 2A 09 09BD BPL D04 IF YES, I DIDNT FIND HIS FILE
00600 00626A 09B4 E6 CB 21 A LDB DCBDRV,U LAST DRIVE CHECKED
00610 00627A 09B7 5C INCB
00620 00628A 09B8 F1 0BDE A CMPB DRIVES ANOTHER VALID DRIVE TO CHECK?
00630 00629A 09BB 25 E0 099D BCS D01 IF YES
00640 00630      * MATCH NOT FOUND - IS IT OK TO CREATE?
00650 00631A 09BD A6 E4 A D04 LDA ,S (DESIRED STATUS)
00660 00632A 09BF B5 04 A BITA #CREATE CREATE BIT ON?
00670 00633A 09C1 26 04 09C7 BNE D04A IF YES

```

```

PAGE 012 IO .SA:0 DOS - I/O ROUTINES
00680 00634A 09C3 86 0A A LDA #ERR10 FILE DIRECTORY ENTRY NOT FOUND
00690 00635A 09C5 20 D1 0998 BRA DOERR
00700 00636A 09C7 6D 61 A D04A TST 1,S ANY DRIVE SPECIFIED?
00710 00637A 09C9 2A 03 09CE BPL D04B IF SPECIFIC
00720 00638A 09CB 6F C8 21 A CLR DCBDRV,U CREATE ON DRIVE ZERO
00730 00639A 09CE 86 FF A D04B LDA #FF SAY LOOK FOR OPEN SLOT
00740 00640A 09D0 17 02A3 0C76 LBSR CHKDIR SCAN THE DIRECTORY
00750 00641A 09D3 27 06 09DB BEQ D04C IF SLOT FOUND
00760 00642A 09D5 2A C1 0998 BPL DOERR IF SOME KIND OF I/O ERROR
00770 00643A 09D7 86 0B A LDA #ERR11 DIRECTORY IS FULL
00780 00644A 09D9 20 BD 0998 DOERRL BRA DOERR
00790 00645A 09DB A6 E4 A D04C LDA ,S DESIRED STATUS
00800 00646A 09DD A7 61 A STA 1,S SAVE IT
00810 00647A 09DF 86 0C A LDA #ERR12 SAY DIRECTORY WAS CREATED
00820 00648A 09E1 A7 E4 A STA ,S
00830 00649A 09E3 86 FF A LDA #FF
00840 00650A 09E5 A7 4D A STA DCBFCL,U SET NUMBER OF 1ST CLUSTER
00850 00651A 09E7 CLRDC
00860 00652A 09E9 ED 4E A STD DCBNLS,U CLEAR BYTES IN LAST SECTOR
00870 00653A 09EB ED C8 14 A STD DCBMRB,U CLEAR MAX RBA
00880 00654A 09EE A7 C8 16 A STA DCBMRB+2,U
00890 00655A 09F1 17 0263 0C57 LBSR DCBDIR XFER DATA TO DIRECTORY
00900 00656A 09F4 20 1B 0A0E BRA D06 GO CONTINUE PROCESSING
00910 00657 * DIRECTORY ENTRY FOUND
00920 00658A 09F6 A6 E4 A D05 LDA ,S DESIRED STATUS
00930 00659A 09F8 A7 61 A STA 1,S SAVE IT
00940 00660A 09FA 6F E4 A CLR ,S
00950 00661A 09FC A6 8B 10 A LDA DCBCFS,X CHK PREVIOUS FILE STATUS
00960 00662A 09FF 27 0D 0A0E BEQ D06 IF IT WAS CLOSED
00970 00663A 0A01 84 0E A ANDA #CREATE+EXTEND+OUT IF LAST OPENED TO MODIFICATION?
00980 00664A 0A03 27 09 0A0E BEQ D06 IF NO
00990 00665A 0A05 6D 8B 10 A TST DCBCFS,X CHK PREVIOUS FILE STATUS
01000 00666A 0A08 27 04 0A0E BEQ D06 IF IT WAS CLOSED
01010 00667A 0A0A 86 0D A LDA #ERR13 SAY IT WASNT PREVIOUSLY CLOSED
01020 00668A 0A0C A7 E4 A STA ,S
01030 00669 * XFER DIRECTORY ENTRY TO DCB
01040 00670A 0A0E A6 61 A D06 LDA 1,S DESIRED STATUS
01050 00671A 0A10 A7 8B 10 A STA DCBCFS,X PUT IN DIRECTORY ENTRY
01060 00672A 0A13 17 0249 0C5F LBSR DIRDCB XFER DIRECTORY ENTRY TO DCB
01070 00673A 0A16 A6 C8 10 A LDA DCBCFS,U
01080 00674A 0A19 84 0E A ANDA #CREATE+EXTEND+OUT WRITES ALLOWED?
01090 00675A 0A1B 27 05 0A22 BEQ D06A IF NO
01100 00676A 0A1D 17 031F 0D3F LBSR SYSWRT REWRITE DIRECTORY RECORD
01110 00677A 0A20 26 B7 09D9 BNE DOERRL IF I/O ERROR
01120 00678A 0A22 86 02 A D06A LDA #2
01130 00679A 0A24 A7 C8 23 A STA DCBSEC,U
01140 00680A 0A27 17 02FD 0D27 LBSR SYSRED READ FAT RECORD
01150 00681A 0A2A 26 AD 09D9 BNE DOERRL
01160 00682A 0A2C 17 021C 0C4B LBSR ADRFAT POINT "X" AT FAT TABLE IN MEMORY
01170 00683A 0A2F 34 40 A PSHS U
01180 00684A 0A31 CE 06C8 A LDU #SYSBUF POINT TO BUFFER
01190 00685A 0A34 C6 45 A LDB #69 BYTES TO MOVE
01200 00686A 0A36 17 022E 0C67 LBSR XFRUX MOVE THEM
01210 00687A 0A39 35 40 A PULS U
01220 00688 * DO OPEN RESETTNG
01230 00689A 0A3B CC FFFF A LDD #FFFF
01240 00690A 0A3E ED C8 29 A STD DCBPRN,U
01250 00691A 0A41 CLRDC

```

```

PAGE 013 IO      .SA:0      DOS - I/O ROUTINES

01260 00692A 0A43 ED   C8 2B  A      STD   DCBRBA,U
01270 00693A 0A46 A7   C8 2D  A      STA   DCBRBA+2,U
01280 00694A 0A49 ED   C8 2E  A      STD   DCBLRN,U
01290 00695A 0A4C 6F   C8 30  A      CLR   DCBMDT,U
01300 00696A 0A4F 16   0080 0ADE      LBRA  DC5
01310 00697
01320 00698
01330 00699
01340 00700
01350 00701
01360 00702
01370 00703
01380 00704
01390 00705
01400 00706
01410 00707
01420 00708
01430 00709A 0A52 4F   DCLOSE CLRA      (RESULT CODE)
01440 00710A 0A53 34   16      A      PSHS  D,X
01450 00711A 0A55 4F   CLRA
01460 00712A 0A56 17   021D 0C76      LBSR  CHKDIR  CHECK DIRECTORY FOR A MATCH
01470 00713A 0A59 27   07  0A62      BEQ   DC1     IF MATCH FOUND
01480 00714A 0A5B 2A   02  0A5F      BPL  DCERR   IF I/O ERR
01490 00715A 0A5D 86   0A      A      LDA  #ERR10  DIRECTORY ENTRY NOT FOUND
01500 00716A 0A5F 16   FF36 0998 DCERR LBRA  DOERR
01510 00717A 0A62 A6   C8 10  A DC1  LDA  DCBCFS,U IS FILE OPEN?
01520 00718A 0A65 26   04  0A6B      BNE  DC2
01530 00719A 0A67 86   13      A      LDA  #ERR19  CLOSING UNOPENED FILE
01540 00720A 0A69 20   F4  0A5F      BRA  DCERR
01550 00721A 0A6B EC   C8 22  A DC2  LDD  DCBTRK,U
01560 00722A 0A6E 34   06      A      PSHS  D      SAVE LOC OF DIR ENT
01570 00723A 0A70 17   02D0 0D43      LBSR  REWRITE REWRITE BUFFER IF IT HAD BEEN MODIFIED
01580 00724A 0A73 35   06      A      PULS  D
01590 00725A 0A75 26   E8  0A5F      BNE  DCERR   IF I/O ERROR OCCURRED IN THE PROCESS
01600 00726A 0A77 ED   C8 22  A      STD  DCBTRK,U RESTORE LOC OF DIR ENT
01610 00727A 0A7A A6   C8 10  A      LDA  DCBCFS,U
01620 00728A 0A7D 34   C2      A      PSHS  A      SAVE FOR DIRECTORY RE-WRITE DECISION
01630 00729A 0A7F 6F   C8 10  A      CLR  DCBCFS,U CLEAR CUR FILE STATUS IN DCB
01640 00730A 0A82 84   10      A      ANDA #WORK    WORK FILE TO BE DELETED?
01650 00731A 0A84 27   18  0A9E      BEQ  DC4     IF NO GO REWRITE DIRECTORY & FAT TABLE
01660 00732A 0A86 6F   C4      A      CLR  ,U     MARK DIRECTORY ENTRY AS RE-USEABLE
01670 00733A 0A88 34   10      A      PSHS  X     SAVE ADDR OF DIRECTORY ENTRY
01680 00734
01690 00735A 0A8A 17   01BE 0C4B      * MARK FAT TABLE ENTRIES AS AVAILABLE
01700 00736A 0A8D A6   4D      A      LBSR  ADRFAT POINT "X" AT FAT TABLE IN MEM
01710 00737A 0A8F 2B   0B  0A9C      LDA  DCBFL,U GET FIRST CLUSTER NUMBER
01720 00738A 0A91 E6   86      A DC3  BMI  DC3A   IF NO CLUSTERS IN USE
01730 00739A 0A93 6F   86      A      LDB  A,X    GET NUMBER OF NEXT CLUSTER
01740 00740A 0A95 6A   86      A      CLR  A,X    CLEAR CLUSTER ENTRY
01750 00741A 0A97 1F   98      A      DEC  A,X    SET TO $FF
01760 00742A 0A99 4D   TSTA
01770 00743A 0A9A 2A   F5  0A91      BPL  DC3     IF MORE TO GO
01780 00744A 0A9C 35   10      A DC3A  PULS  X     ADDR OF DIR ENTRY
01790 00745A 0A9E 17   01B6 0C57 DC4  LBSR  DCBDIR XFER TO DIRECTORY
01800 00746A 0AA1 35   02      A      PULS  A     PRE-CLOSE CFS
01810 00747A 0AA3 84   0E      A      ANDA #CREATE+EXTEND+OUT WRITES ALLOWED?
01820 00748A 0AA5 27   15  0ABC      BEQ  DC4B
01830 00749
* SET DCBNLS TO REFLECT DCBMRB (MAX RBA)

```

PAGE 014 IO .SA:0 DOS - I/O ROUTINES

```

01840 00750A 0AA7 4F          CLRA
01850 00751A 0AAB E6      8B 16  A      LDB   DCBMRB+2,X
01860 00752A 0AAB 26      08  0AB5  BNE   DC4A
01870 00753A 0AAD EC      8B 14  A      LDD   DCBMRB,X IS IT A NULL FILE
01880 00754A 0AB0 27      03  0AB5  BEQ   DC4A IF YES
01890 00755A 0AB2 CC      0100  A      LDD   #$100
01900 00756A 0AB5 ED      0E          STD   DCBNLS,X
01910 00757A 0AB7 17      0285 0D3F  LBSR  SYSWRT RE-WRITE DIRECTORY RECORD
01920 00758A 0ABA 26      A3  0A5F  BNE   DCERR IF I/O ERROR
01930 00759A 0ABC 17      018C 0C4B  DC4B  LBSR  ADRFAT
01940 00760A 0ABF 34      40          PSHS  U SAVE DCB ADDR
01950 00761A 0AC1 CE      06C8  A      LDU   #SYSBUF POINT TO SYSTEM'S BUFFER
01960 00762A 0AC4 C6      45          LDB   #69
01970 00763A 0AC6 17      01A6 0C6F  LBSR  XFRXU XFER INTO BUFFER
01980 00764A 0AC9 30      41          LEAX  1,U
01990 00765A 0ACB C6      BA          LDB   #256-69-1
02000 00766A 0ACD 17      0197 0C67  LBSR  XFRUX CLEAR REST OF BUFFER TO %FF
02010 00767A 0ADD 35      40          PULS  U RESTORE DCB ADDR
02020 00768A 0AD2 86      02          LDA   #2
02030 00769A 0AD4 A7      CB 23  A      STA   DCBSEC,U
02040 00770A 0AD7 17      0265 0D3F  LBSR  SYSWRT WRITE IT
02050 00771A 0ADA 27      02  0ADE  BEQ   DC5
02060 00772A 0ADC A7      E4          STA   ,S IF I/O ERROR
02070 00773A 0ADE 6D      E4          TST   ,S SET COND CODES
02080 00774A 0AE0 35      96          PULS  D,X,PC
02090 00775          *
02100 00776          *****
02110 00777          * READ A LOGICAL DISK RECORD
02120 00778          *
02130 00779          * GIVEN: U -> DCB (THAT HAS ALREADY BEEN OPENED!)
02140 00780          * A = FUNCTION DESIRED CODED AS FOLLOWS:
02150 00781          * BIT 0 ON TO READ VIA RBA
02160 00782          * OFF TO READ VIA LRN
02170 00783          * BIT 1 ON TO READ WITHOUT CHANGING POINTER
02180 00784          * OFF TO EXIT AFTER POINTING AT NEXT (PREVIOUS) RECORD
02190 00785          * BIT 2 ON TO READ BACKWARDS
02200 00786          * OFF TO READ FORWARD
02210 00787          * EXAMPLE: A=ZERO TO READ THE CURRENT LOGICAL RECORD AND THEN ADVANCE
02220 00788          * THE LOGICAL RECORD NUMBER BY 1. A = 2 TO "READ FOR UPDATE" A LOGICAL
02230 00789          * RECORD. A = 1+4 (5) TO READ STARTING WITH THE RBA'TH BYTE OF DATA
02240 00790          * IN THE FILE, FOR DCBRSZ BYTES. THEN SET RBA TO POINT DCBRSZ BYTES
02250 00791          * AHEAD OF THE FIRST BYTE READ.
02260 00792          *
02270 00793          * NOTE: LOGICAL RECORD SIZE, RECORD STORAGE ADDRESS AND I/O BUFFER
02280 00794          * ADDRESS ARE USED. IF LOGICAL RECORD SIZE IS 256, RECORD STORAGE
02290 00795          * AND I/O BUFFER MAY BE THE SAME ADDRESS. IF DCBRSZ IS ZERO, READS WILL
02300 00796          * TRANSFER BYTES FROM THE FILE TO [DCBREC] UNTIL A CHARACTER MATCHING
02310 00797          * DCBTRM IS TRANSFERRED.
02320 00798          *****
02330 00799A 0AE2 34      32          A DREAD PSHS  A,X,Y
02340 00800A 0AE4 CC      010F  A      LDD   #$0100+ERR15
02350 00801A 0AE7 17      0091 0B7B  LBSR  RDWR DO SETUP COMMON TO READ AND WRITE
02360 00802          *
02370 00803          * LOOP TO XFER BYTES TO RECORD AREA
02380 00804          * (X->BUFFER, Y->RECORD AREA)
02390 00805A 0AEA E6      CB 2D  A DR5  LDB   DCBRBA+2,U DISPLACEMENT IN CURRENT SECTOR
02400 00806A 0AED 4F          CLRA
02410 00807A 0AEE A6      8B          LDA   D,X GET A BYTE

```

```

PAGE 015 IO .SA:0 DOS - I/O ROUTINES
02420 00808A 0AF0 A7 A0 A STA ,Y+ STORE IN RECORD AREA
02430 00809A 0AF2 6C C8 2D A INC DCBRBA+2,U ADVANCE POINTER IN BUFFER
02440 00810A 0AF5 26 1E 0B15 BNE DR5B IF IN SAME SECTOR
02450 00811A 0AF7 17 0249 0D43 LBSR REWRITE ENSURE PREVIOUSLY MODIFIED DATA GETS WRITTEN
02460 00812A 0AFA 26 0E 0B0A BNE DR5AA IF WRITE ERR
02470 00813A 0AFC EC C8 2B A LDD DCBRBA,U
02480 00814A 0AFF C3 0001 A ADDD #1 POINT TO NEXT SECTOR
02490 00815A 0B02 ED C8 2B A STD DCBRBA,U
02500 00816A 0B05 17 0291 0D99 LBSR CALSEC RECALCULATE TRACK & SECTOR
02510 00817A 0B08 27 06 0B10 BEQ DR5A IF OK
02520 00818A 0B0A 32 67 A DR5AA LEAS 7,S SCRAP STUFF IN STACK
02530 00819A 0B0C A7 E4 A STA ,S
02540 00820A 0B0E 35 B2 A PULS A,X,Y,PC
02550 00821A 0B10 17 01D4 0CE7 DR5A LBSR DSKRED
02560 00822A 0B13 26 F5 0B0A BNE DR5AA IF I/O ERROR
02570 00823A 0B15 EC E4 A DR5B LDD ,S GET COUNT DOWN VALUE
02580 00824A 0B17 27 09 0B22 BEQ DR5C IF VARIABLE LENGTH STRING
02590 00825A 0B19 83 0001 A SUBD #1
02600 00826A 0B1C ED E4 A STD ,S
02610 00827A 0B1E 26 CA 0AEA BNE DR5 GO GET ANOTHER CHR
02620 00828A 0B20 20 07 0B29 BRA RDWRX GO DO CLEAN-UP COMMON TO READ AND WRITE
02630 00829A 0B22 A6 C8 13 A DR5C LDA DCBTRM,U STRING DELIMITER
02640 00830A 0B25 A1 3F A CMPA -1,Y WAS LAST CHR STORED A DELIMITER?
02650 00831A 0B27 26 C1 0AEA BNE DR5 IF NO, KEEP GOING
02660 00832 *
02670 00833 *****
02680 00834 * CLEAN UP COMMON TO READ AND WRITE
02690 00835 *****
02700 00836 * RECORD HAS BEEN READ - CLEAN UP
02710 00837A 0B29 35 06 A RDWRX PULS D
02720 00838A 0B2E A6 C8 10 A LDA DCBCFS,U FILE STATUS
02730 00839A 0B2E 85 40 A BITA #SHARE OPTION SET?
02740 00840A 0B30 27 08 0B3A BEQ DR6A IF NO
02750 00841A 0B32 17 020E 0D43 DR6AA LBSR REWRITE FREE UP BUFFER
02760 00842A 0B35 CC FFFF A LDD #FFFF MARK INVALID SECTOR IN BUFFER
02770 00843A 0B38 20 09 0B43 BRA DR6B
02780 00844A 0B3A A6 65 A DR6A LDA 5,S R/W OPTION
02790 00845A 0B3C 85 08 A BITA #NOW REWRITE NOW?
02800 00846A 0B3E 26 F2 0B32 BNE DR6AA IF YES
02810 00847A 0B40 EC C8 2B A LDD DCBRBA,U LAST SECTOR ACCESSED
02820 00848A 0B43 ED C8 29 A DR6B STD DCBPRN,U MARK WHICH SECTOR IS NOW IN BUFFER
02830 00849 * CHECK FOR NEW DCBMRB
02840 00850A 0B46 EC C8 2B A LDD DCBRBA,U
02850 00851A 0B49 10A3 C8 14 A CMPD DCBMRB,U
02860 00852A 0B4D 25 16 0B65 BCS DR6D IF IN A LOWER SECTOR
02870 00853A 0B4F 26 08 0B59 BNE DR6C IF A HIGHER SECTOR
02880 00854A 0B51 A6 C8 2D A LDA DCBRBA+2,U
02890 00855A 0B54 A1 C8 16 A CMPA DCBMRB+2,U
02900 00856A 0B57 25 0C 0B65 BCS DR6D IF A LOWER BYTE
02910 00857A 0B59 EC C8 2B A DR6C LDD DCBRBA,U
02920 00858A 0B5C ED C8 14 A STD DCBMRB,U
02930 00859A 0B5F A6 C8 2D A LDA DCBRBA+2,U
02940 00860A 0B62 A7 C8 16 A STA DCBMRB+2,U
02950 00861A 0B65 A6 65 A DR6D LDA 5,S READ/WRITE OPTION
02960 00862A 0B67 84 02 A ANDA #UPDATE SHOULD RBA & LRN BE RESET TO STARTING VALUE?
02970 00863A 0B69 35 32 A PULS A,X,Y
02980 00864A 0B6B 27 0A 0B77 BEQ DR6E IF NO
02990 00865 * RESTORE ORIGINAL POINTERS

```

```

PAGE 016 IO .SA:0 DOS - I/O ROUTINES

03000 00866A 0B6D A7 C8 2D A STA DCBRBA+2,U
03010 00867A 0B70 AF C8 2B A STX DCBRBA,U
03020 00868A 0B73 10AF C8 2E A STY DCBLRN,U
03030 00869A 0B77 6F E4 A DR6E CLR ,S
03040 00870A 0B79 35 B2 A PULS A,X,Y,PC
03050 00871 *
03060 00872 *****
03070 00873 * SETUP FOR READ OR WRITE
03080 00874 * GIVEN: A=1 FOR READ, 2 FOR WRITE
03090 00875 * B=ERR NBR FOR POSSIBLE USE
03100 00876 *****
03110 00877A 0B7B 34 06 A RDWR PSHS D SAVE IN CASE NEEDED
03120 00878 * IS FILE OPEN?
03130 00879A 0B7D A6 C8 10 A LDA DCBCFS,U
03140 00880A 0B80 26 08 0B8A BNE RDWR1 IF YES
03150 00881A 0B82 86 0E A LDA #ERR14 IF NOT OPEN
03160 00882A 0B84 32 64 A RDWRER LEAS 4,S (DIDN'T NEED IT AND RET ADDR)
03170 00883A 0B86 A7 E4 A STA ,S
03180 00884A 0B88 35 B2 A PULS A,X,Y,PC
03190 00885 * IS THIS TYPE OF OPERATION ALLOWED (READ OR WRITE)?
03200 00886A 0B8A A5 E4 A RDWR1 BITA ,S (1 FOR READ, 2 FOR WRITE)
03210 00887A 0B8C 26 04 0B92 BNE RDWR2 IF YES
03220 00888A 0B8E A6 61 A LDA 1,S (ERROR NUMBER PROVIDED)
03230 00889A 0B90 20 F2 0B84 BRA RDWRER
03240 00890 *
03250 00891 * CHECK FOR STARTING RBA
03260 00892 * IS I/O BY RBA OR LOGICAL RECORD?
03270 00893A 0B92 A6 64 A RDWR2 LDA 4,S OPTION PROVIDED
03280 00894A 0B94 84 01 A ANDA #RBA
03290 00895A 0B96 26 0E 0BA6 BNE RDWR4 IF READ VIA RBA, USE RBA'S CURRENT CONTENTS
03300 00896 * READ BY RECORD NUMBER
03310 00897A 0B98 EC C8 11 A LDD DCBRSZ,U FIXED OR VARIABLE LENGTH RECORDS?
03320 00898A 0B9B 26 04 0BA1 BNE RDWR3 IF FIXED LENGTH
03330 00899A 0B9D 86 14 A LDA #ERR20 CANT CALCULATE - RSZ = ZERO
03340 00900A 0B9F 20 E3 0B84 BRA RDWRER
03350 00901A 0BA1 17 01B3 0D57 RDWR3 LBSR CALRBA CALCULATE RECORD'S STARTING RBA
03360 00902A 0BA4 26 DE 0B84 BNE RDWRER IF OVERFLOW OCCURRED
03370 00903 *
03380 00904 OPT L
03390 00905 * MAKE SURE STARTING RECORD IS IN BUFFER
03400 00906A 0BA6 EC C8 2B A RDWR4 LDD DCBRBA,U (RELATIVE RECORD NEEDED)
03410 00907A 0BA9 10A3 C8 29 A CMPD DCBPRN,U IS NEEDED RECORD IN BUFFER?
03420 00908A 0BAD 27 11 0BC0 BEQ RDWR4A IF YES
03430 00909A 0BAF 17 0191 0D43 LBSR REWRTE REWRITE BUFFER IF IT HAS BEEN MODIFIED
03440 00910A 0BB2 26 D0 0B84 BNE RDWRER IF I/O ERROR IN THE PROCESS
03450 00911A 0BB4 17 01E2 0D99 LBSR CALSEC CALCULATE TRACK & SECTOR
03460 00912A 0BB7 26 C8 0B84 BNE RDWRER IF TRYING TO GO BEYOND EOF
03470 00913A 0BB9 17 012B 0CE7 LBSR DSKRED READ THE SECTOR
03480 00914A 0BBC 26 C6 0B84 BNE RDWRER IF I/O ERR
03490 00915A 0BBE 20 05 0BC5 BRA RDWR5
03500 00916A 0BC0 17 01DC 0D9F RDWR4A LBSR CSENT CHECK FOR EOF
03510 00917A 0BC3 26 BF 0B84 BNE RDWRER IF TRYING TO GO PAST EOF
03520 00918 *
03530 00919 * CORRECT STARTING SECTOR IS IN BUFFER
03540 00920 * GET SET TO XFER RECORD
03550 00921A 0BC5 35 26 A RDWR5 PULS D,Y (D=1/0, ERR NBR, Y = RETURN ADDR)
03560 00922 OPT NOL
03570 00923A 0BC7 AE C8 2E A LDX DCBLRN,U

```

```

PAGE 017 IO      .SA:0      DOS - I/O ROUTINES

03580 00924A 0BCA 34 10      A      PSHS  X      SAVE IN CASE POINTERS DON'T ADVANCE
03590 00925A 0BCC 30 01      A      LEAX  1,X     POINT TO NEXT RECORD
03600 00926A 0BCE AF 08 2E    A      STX   DCBLRN,U
03610 00927A 0BD1 AE 08 2B    A      LDX   DCBRBA,U
03620 00928A 0BD4 A6 08 2D    A      LDA   DCBRBA+2,U
03630 00929A 0BD7 34 12      A      PSHS  A,X     SAVE IN CASE POINTERS DON'T ADVANCE
03640 00930A 0BD9 EC 08 11    A      LDD   DCBRSZ,U GET RECORD LENGTH
03650 00931A 0BDC 34 06      A      PSHS  D      SAVE AS COUNT DOWN VALUE FOR LOOP
03660 00932A 0BDE 34 20      A      PSHS  Y      SAVE RET ADDR
03670 00933A 0BE0 AE 08 24    A      LDX   DCBBUF,U ADDR OF BUFFER
03680 00934A 0BE3 10AE 08 27  A      LDY   DCBLRB,U ADDR OF LOGICAL RECORD BUFFER
03690 00935A 0BE7 39      A      RTS      RETURN TO READ OR WRITE LOOP
03700 00936      *
03710 00937      *****
03720 00938      * WRITE A LOGICAL DISK RECORD
03730 00939      *
03740 00940      * GIVEN: U -> DCB (THAT HAS ALREADY BEEN OPENED!)
03750 00941      * A = FUNCTION DESIRED CODED AS FOLLOWS:
03760 00942      * BIT 0 ON TO WRITE VIA RBA
03770 00943      * OFF TO WRITE VIA LRN
03780 00944      * BIT 1 ON TO WRITE WITHOUT CHANGING POINTER
03790 00945      * OFF TO EXIT AFTER POINTING AT NEXT (PREVIOUS) RECORD
03800 00946      * BIT 2 ON TO WRITE BACKWARDS
03810 00947      * OFF TO WRITE FORWARD
03820 00948      * BIT 3 ON TO RELEASE BUFFER AFTER WRITE
03830 00949      * OFF TO WAIT UNTIL PHYSICAL I/O IS NECESSARY
03840 00950      * NOTE: FUNCTION IS NEARLY THE SAME AS DREAD - SEE NOTES UNDER DREAD.
03850 00951      *****
03860 00952A 0BE8 34 32      A      DWRITE PSHS  A,X,Y
03870 00953A 0BEA CC 0215    A      LDD   #0200+ERR21
03880 00954A 0BED 8D 0C 0B7B  A      BSR   RDWR     DO SETUP COMMON TO READ AND WRITE
03890 00955      *
03900 00956      * LOOP TO XFER BYTES FROM RECORD AREA
03910 00957      * (X->BUFFER; Y->RECORD AREA)
03920 00958A 0BEF E6 08 2D    A      DW5   LDB   DCBRBA+2,U DISPLACEMENT IN CURRENT SECTOR
03930 00959A 0BF2 4F      CLRA
03940 00960A 0BF3 AE 08 24    A      LDX   DCBBUF,U ADDR OF BUFFER
03950 00961A 0BF6 30 08      A      LEAX  D,X     DETERMINE ADDR IN BUFFER
03960 00962A 0BF8 A6 A0      A      LDA   ,Y+     GET BYTE FROM RECORD AREA
03970 00963A 0BFA A7 B4      A      STA   ,X     STORE IN BUFFER
03980 00964A 0BFC 6C 08 2D    A      INC   DCBRBA+2,U ADVANCE POINTER IN BUFFER
03990 00965A 0BFF 26 23 0C24  A      BNE   DW5B    IF IN SAME SECTOR
04000 00966A 0C01 17 00E6 0CEA  A      LBSR  DSKWRT  REWRITE SECTOR
04010 00967A 0C04 26 0E 0C14  A      BNE   DW5AA   IF I/O ERROR
04020 00968A 0C06 EC 08 2B    A      LDD   DCBRBA,U
04030 00969A 0C09 C3 0001    A      ADDD  #1     POINT TO NEXT SECTOR
04040 00970A 0C0C ED 08 2B    A      STD   DCBRBA,U
04050 00971A 0C0F 17 0187 0D99  A      LBSR  CALSEC  RECALCULATE TRACK & SECTOR
04060 00972A 0C12 27 06 0C1A  A      BEQ   DW5A    IF OK
04070 00973A 0C14 32 67      A      DW5AA LEAS  7,S   SCRAP STUFF IN STACK
04080 00974A 0C16 A7 E4      A      STA   ,S
04090 00975A 0C18 35 B2      A      PULS  A,X,Y,PC
04100 00976A 0C1A 17 00CA 0CE7  DW5A  A      LBSR  DSKRED
04110 00977A 0C1D 26 F5 0C14  A      BNE   DW5AA   IF I/O ERROR
04120 00978A 0C1F 86 01      A      LDA   #1
04130 00979A 0C21 A7 08 30    A      STA   DCBMDT,U MARK NEW REC AS MODIFIED
04140 00980A 0C24 EC E4      A      DW5B LDD   ,S     GET COUNT DOWN VALUE
04150 00981A 0C26 27 09 0C31  A      BEQ   DW5C    IF VARIABLE LENGTH STRING

```

```

PAGE 018 IO .SA:0 DOS - I/O ROUTINES

04160 00982A 0C2B 83 0001 A SUBD #1
04170 00983A 0C2B ED E4 A STD ,S
04180 00984A 0C2D 26 C0 0BEF BNE DW5 GO GET ANOTHER CHR
04190 00985A 0C2F 20 07 0C38 BRA DW6
04200 00986A 0C31 A6 CB 13 A DW5C LDA DCBTRM,U STRING DELIMITER
04210 00987A 0C34 A1 3F A CMPA -1,Y WAS LAST CHR STORED A DELIMITER?
04220 00988A 0C36 26 B7 0BEF BNE DW5 IF NO, KEEP GOING
04230 00989 *
04240 00990 * RECORD HAS BEEN WRITTEN - CLEAN UP
04250 00991A 0C38 86 01 A DW6 LDA #1
04260 00992A 0C3A A7 CB 30 A STA DCBMDT,U ENSURE THIS SECTOR GETS REWRITTEN (EVETUALLY)
04270 00993A 0C3D 7E 0B29 A JMP RDWRX CLEAN UP SAME AS FOR READ
04280 00994 *
04290 00995 *****
04300 00996 * RELEASE THE I/O BUFFER
04310 00997 * (USED WHEN USER WANTS TO CONTROL SHARED BUFFER)
04320 00998 * GIVEN: U->DCB
04330 00999 *****
04340 01000A 0C40 17 0100 0D43 DRELSE LBSR REWRTE REWRITE BUFFER CONTENTS IF NECESSARY
04350 01001A 0C43 CC FFFF A LDD #FFFF
04360 01002A 0C46 ED CB 29 A STD DCBPRN,U FORCE READ NEXT TIME
04370 01003A 0C49 4F CLRA
04380 01004A 0C4A 39 RTS
00010 01005 OPT L
00020 01006 TTL DOS - SUPPORTING SUBROUTINES
00030 01007 OPT NOL
00040 01008 *****
00050 01009 * POINT "X" AT FAT TABLE IN MEMORY
00060 01010 * GIVEN: U-> DCB CONTAINING DCBDRV
00070 01011 * RETURNED:X
00080 01012 *****
00090 01013A 0C4B 8E 07CB A ADRFAT LDX #FATS FAT TABLE STORE AREA
00100 01014A 0C4E A6 CB 21 A LDA DCBDRV,U DRIVE CONTAINING FILE
00110 01015A 0C51 C6 45 A LDB #69 NUMBER OF BYTES SAVED
00120 01016A 0C53 3D MUL
00130 01017A 0C54 30 8B A LEAX D,X POINT TO CORRECT AREA
00140 01018A 0C56 39 RTS
00150 01019 *
00160 01020 *****
00170 01021 * X F E R B Y T E S R O U T I N E S
00180 01022 *****
00190 01023 * XFER 32 BYTES FROM DCB (AT ,U) TO DIRECTORY (AT ,X)
00200 01024A 0C57 34 56 A DCBDIR PSHS D,X,U
00210 01025A 0C59 C6 20 A LDB #32 BYTES TO XFER
00220 01026A 0C5B 8D 0A 0C67 BSR XFRUX
00230 01027A 0C5D 35 D6 A PULS D,X,U,PC
00240 01028 * XFER 32 BYTES FROM DIRECTORY AT ,X TO DCB AT ,U
00250 01029A 0C5F 34 56 A DIRDCB PSHS D,X,U
00260 01030A 0C61 C6 20 A LDB #32
00270 01031A 0C63 8D 0A 0C6F BSR XFRUX
00280 01032A 0C65 35 D6 A PULS D,X,U,PC
00290 01033 * TRANSFER B BYTES FROM ,U TO ,X
00300 01034A 0C67 A6 C0 A XFRUX LDA ,U+
00310 01035A 0C69 A7 80 A STA ,X+
00320 01036A 0C6B 5A DECB
00330 01037A 0C6C 26 F9 0C67 BNE XFRUX
00340 01038A 0C6E 39 RTS
00350 01039 * TRANSFER B BYTES FROM ,X TO ,U

```

```

PAGE 019 RTN      .SA:0          DOS - SUPPORTING SUBROUTINES

00360 01040A 0C6F 1E   13      A XFRXU  EXG   X,U
00370 01041A 0C71 8D   F4      0C67  BSR   XFRUX
00380 01042A 0C73 1E   13      A       EXG   X,U
00390 01043A 0C75 39                RTS
00400 01044                *
00410 01045                *****
00420 01046                * CHECK DIRECTORY ON THIS DRIVE
00430 01047                * GIVEN: A=ZERO IF LOOKING FOR A MATCH
00440 01048                *       A NOT ZERO IF LOOKING FOR AVAILABLE SLOT
00450 01049                *       U -> DCB
00460 01050                * RETURNED: A=ZERO IF REQUEST SUCCESSFUL
00470 01051                *       A=FF IF NO MATCH FOUND
00480 01052                *       A=1-8 IF I/O ERROR
00490 01053                *       IF SUCCESSFUL, X-> DIRECTORY ENTRY IN BUFFER
00500 01054                *****
00510 01055A 0C76 34   06      A CHKDIR PSHS  D       SAVE OPTION
00520 01056A 0C78 CC  1103     A       LDA   #1103
00530 01057A 0C7B A7   CB 22     A       STA   DCBTRK,U SET TO READ DIRECTORY TRACK
00540 01058A 0C7E E7   CB 23     A       STB   DCBSEC,U SET TO READ FIRST DIRECTORY ENTRIES
00550 01059                * RETRY ONLY IF DRIVE IS READY!
00560 01060A 0C81 B6   0622     A CD1   LDA   >RETRYS
00570 01061A 0C84 34   02      A       PSHS  A
00580 01062A 0C86 86   02      A       LDA   #2          ** CHANGED IN VER 6 **
00590 01063A 0C88 B7   0622     A       STA   >RETRYS
00600 01064A 0C8B 17   0099 0D27  LBSR  SYSRED  DO PHYSICAL READ
00610 01065A 0C8E 35   04      A       PULS  B       GET ORIG NBR OF RETRYS
00620 01066A 0C90 27   12      0CA4     A       BEQ   CD2       IF I/O OK
00630 01067A 0C92 F7   0622     A       STB   >RETRYS
00640 01068A 0C95 81   01      A       CMPA  #ERR1   DRIVE NOT READY?
00650 01069A 0C97 26   04      0C9D     A       BNE   CD1A    IF I SHOULD TRY SOME MORE
00660 01070A 0C99 A7   E4      A CD1E    STA   +S
00670 01071A 0C9B 35   B6      A       PULS  D,PC
00680 01072A 0C9D 17   0087 0D27  CD1A  LBSR  SYSRED  GO TRY SOME MORE
00690 01073A 0CA0 26   F7      0C99     A       BNE   CD1E    IF STILL ERROR
00700 01074A 0CA2 20   03      0CA7     A       BRA   CD2A
00710 01075                * CHECK THE DIRECTORY ENTRIES IN THIS RECORD
00720 01076A 0CA4 F7   0622     A CD2   STB   >RETRYS
00730 01077A 0CA7 86   08      A CD2A   LDA   #8       NUMBER OF DIRECTORY ENTRIES PER REC
00740 01078A 0CA9 A7   61      A       STA   1,S
00750 01079A 0CAB 8E   06C8     A       LDX   #SYSBUF  POINT AT SYSTEM BUFFER
00760 01080A 0CAE 6D   E4      A CD3    TST   +S       OPTION?
00770 01081A 0CB0 27   0A      0CBC     A       BEQ   CD5       IF LOOKING FOR A MATCH
00780 01082A 0CB2 A6   84      A       LDA   +X       LOOK AT 1ST BYTE
00790 01083A 0CB4 27   02      0CB8     A       BEQ   CD4       IF I FOUND RE-USABLE SPACE
00800 01084A 0CB6 2A   18      0CD0     A       BPL   CD7       IF NOT USEABLE
00810 01085A 0CB8 6F   E4      A CD4    CLR   +S
00820 01086A 0CBA 35   86      A       PULS  D,PC    RETURN SUCCESSFULLY
00830 01087                * COMPARE LOOP
00840 01088A 0CBC A6   84      A CD5    LDA   +X       LOOK AT 1ST BYTE OF DIRECTORY ENTRY
00850 01089A 0CBE 27   10      0CD0     A       BEQ   CD7       IF DELETED ENTRY
00860 01090A 0CC0 2B   1F      0CE1     A       BMI   CDB      IF END OF DIRECTORY ENTRIES
00870 01091A 0CC2 5F                CLRB  CHARACTER POSITION COUNTER
00880 01092A 0CC3 A6   B5      A CD6    LDA   B+X     CHR IN DIRECTORY FILE NAME
00890 01093A 0CC5 A1   C5      A       CMPA  B,U     CHR IN DCB FILE NAME
00900 01094A 0CC7 26   07      0CD0     A       BNE   CD7       IF NOT A MATCH
00910 01095A 0CC9 5C                INCB
00920 01096A 0CCA C1   0B      A       CMPB  #11     MORE CHARACTERS TO COMPARE?
00930 01097A 0CCC 25   F5      0CC3     A       BCS   CD6       IF YES

```

```

PAGE 020 RTN .SA:0 DOS - SUPPORTING SUBROUTINES
00940 01098 * MATCH FOUND
00950 01099A 0CCE 20 EB 0CB8 BRA CD4
00960 01100 *
00970 01101A 0CD0 30 8B 20 A CD7 LEAX 32,X POINT TO NEXT DIRECTORY ENTRY
00980 01102A 0CD3 6A 61 A DEC 1,S MORE ENTRIES TO LOOK AT IN THIS REC?
00990 01103A 0CD5 26 D7 0CAE BNE CD3 IF YES
01000 01104A 0CD7 6C C8 23 A INC DCBSEC,U
01010 01105A 0CDA A6 C8 23 A LDA DCBSEC,U
01020 01106A 0CDD 81 0C A CMPA #12 MORE DIRECTORY RECORDS TO READ?
01030 01107A 0CDF 25 A0 0CB1 BCS CD1 IF YES
01040 01108 *
01050 01109A 0CE1 86 FF A CD8 LDA #FF
01060 01110A 0CE3 A7 E4 A STA ,S
01070 01111A 0CE5 35 86 A PULS D,PC
01080 01112 *
01090 01113 *****
01100 01114 * PHYSICAL DISK READ
01110 01115 * GIVEN: U->DCB
01120 01116 * FUNCTION: READ INTO DCBBUF
01130 01117 * (NOTE:DSKCON RETRYS ON ERROR 5 TIMES)
01140 01118 * RETURNED:DCBOK = RESULT CODE (ALSO IN A)
01150 01119 *****
01160 01120A 0CE7 86 02 A DSKRED LDA #2 READ SECTOR OP CODE
01170 01121A 0CE9 8C A FCB #8C SKIP OVER NEXT INSTR
01180 01122 *
01190 01123 *****
01200 01124 * PHYSICAL DISK WRITE
01210 01125 * ESSENTIALLY SAME AS ABOVE
01220 01126 *****
01230 01127A 0CEA 86 03 A DSKWRT LDA #3 WRITE OP CODE
01240 01128A 0CEC A7 C8 20 A DSKIO STA DCBOPC,U
01250 01129A 0CEF 6F C8 30 A CLR DCBMDT,U
01260 01130 * FALL THRU
01270 01131 *
01280 01132 *****
01290 01133 * CALL DSKCON
01300 01134 * GIVEN:PARAMETERS IN DCB
01310 01135 * FUNCTION: XFER PARAMS TO [C006]
01320 01136 * CALL DSKCON
01330 01137 * MOVE RESULT CODE TO DCB
01340 01138 * LEAVE RESULT CODE IN A
01350 01139 *****
01360 01140A 0CF2 34 14 A XFRIOP PSHS B,X
01370 01141A 0CF4 BE C006 A LDX >#C006
01380 01142A 0CF7 EC C8 20 A LDD DCBOPC,U
01390 01143A 0CFA ED 81 A STD ,X++
01400 01144A 0CFC EC C8 22 A LDD DCBTRK,U
01410 01145A 0CFF ED 81 A STD ,X++
01420 01146A 0D01 EC C8 24 A LDD DCBBUF,U
01430 01147A 0D04 ED 81 A STD ,X++
01440 01148A 0D06 34 0B A XIOENT PSHS DP
01450 01149A 0D08 4F CLRA
01460 01150A 0D09 1F 8B A TFR A,DP
01470 01151A 0D0B 8D 10 0D1D BSR D0IO DO I/O
01480 01152A 0D0D 35 08 A PULS DP
01490 01153A 0D0F 4F CLRA
01500 01154A 0D10 E6 84 A LDB ,X GET RESULT CODE
01510 01155A 0D12 27 07 0D1B BEQ XIOX IF NO ERROR, EXIT

```

```

PAGE 021 RTN      .SA:0      DOS - SUPPORTING SUBROUTINES

01520 01156      * GENERATE ERROR NUMBER BASED ON WHICH BIT IS ON
01530 01157A 0D14 58      XIOA  LSLB      IS THIS BIT SET?
01540 01158A 0D15 25      03  0D1A      BCS  XIOB      IF YES
01550 01159A 0D17 4C      INCA
01560 01160A 0D18 20      FA  0D14      BRA  XIOA
01570 01161A 0D1A 4C      XIOB  INCA
01580 01162A 0D1B 35      94      A  XIOX  PULS  B,X,PC
01590 01163      0024      A  ZZ  EQU  ERR1+ERR2+ERR3+ERR4+ERR5+ERR6+ERR7+ERR8
01600 01164      * THE ABOVE LINE SIMPLY PUTS ERR1-B ON THE XREF MAP
01610 01165A 0D1D 34      76      A  DOI0  PSHS  D,X,Y,U
01620 01166A 0D1F B6      0622      A      LDA  >RETRY5
01630 01167A 0D22 BE      C004      A      LDX  >$C004
01640 01168A 0D25 6E      04      A      JMP  4,X
01650 01169      *
01660 01170      *****
01670 01171      * PHYSICAL DISK READ - SYSTEM FUNCTIONS
01680 01172      * SAME AS DSKRED EXCEPT SYSTEM'S BUFFER USED
01690 01173      *****
01700 01174A 0D27 86      02      A  SYSRED LDA  #2
01710 01175A 0D29 34      14      A  SYSIO  PSHS  B,X
01720 01176A 0D2B E6      CB 21      A      LDE  DCBDRV,U
01730 01177A 0D2E BE      C006      A      LDX  >$C006
01740 01178A 0D31 ED      B1      A      STD  ,X++
01750 01179A 0D33 EC      CB 22      A      LDD  DCBTRK,U
01760 01180A 0D36 ED      B1      A      STD  ,X++      TRACK & SECTOR
01770 01181A 0D38 CC      06CB      A      LDD  #SYSBUF
01780 01182A 0D3B ED      B1      A      STD  ,X++
01790 01183A 0D3D 20      C7  0D06      BRA  XIOENT      FINISH UP LIKE USER IO
01800 01184      *
01810 01185      *****
01820 01186      * PHYSICAL DISK WRITE - SYSTEM FUNCTIONS
01830 01187      *****
01840 01188A 0D3F 86      03      A  SYSWRT LDA  #3
01850 01189A 0D41 20      E6  0D29      BRA  SYSIO
01860 01190      *
01870 01191      *****
01880 01192      * IF DATA IN BUFFER HAS BEEN MODIFIED (DCBMDT NOT = 0) CHECK
01890 01193      * TO SEE IF WRITES ARE ALLOWED. IF NO, DO NOT SET ERROR - JUST EXIT.
01900 01194      * IF YES, REWRITE BLOCK IN BUFFER (EXIT WITH ERROR IN A IF WRITE NO GOOD.)
01910 01195      *
01920 01196      * GIVEN: U->DCB CONTAINING DCBPRN = PHYSICAL REC NUMBER THAT IS IN BUFFER.
01930 01197      *****
01940 01198A 0D43 6D      CB 30      A  REWRITE TST  DCBMDT,U DATA IN BUFFER MODIFIED?
01950 01199A 0D46 26      02  0D4A      BNE  RW1      IF YES
01960 01200A 0D48 4F      RWX  CLRA
01970 01201A 0D49 39      RWXX RTS
01980 01202A 0D4A A6      CB 10      A  RW1  LDA  DCB.CFS,U
01990 01203A 0D4D 84      02      A      ANDA #OUT      ARE WRITES ALLOWED?
02000 01204A 0D4F 27      F7  0D48      BEQ  RWX      IF NO, EXIT WITH NO ERROR
02010 01205A 0D51 8D      4C  0D9F      BSR  CSENT      RE-ESTABLISH TRK & SEC FROM PRN
02020 01206A 0D53 26      F4  0D49      BNE  RWXX      IF NG, EXIT WITH ERROR
02030 01207A 0D55 20      93  0CEA      BRA  DSKWRT      GO DO REWRITE & RETURN TO CALLER
02040 01208      *
02050 01209      *****
02060 01210      * CALCULATE RELATIVE BYTE ADDRESS FROM LOGICAL RECORD NUMBER
02070 01211      * (DCBRBA = DCBRSZ * DCBLRN)
02080 01212      *****
02090 01213A 0D57 34      70      A  CALRBA PSHS  X,Y,U

```

```

PAGE 022 RTN      .SA:0      DOS - SUPPORTING SUBROUTINES

02100 01214A 0D59 30  C8 2B  A      LEAX  DCBRBA,U
02110 01215A 0D5C 31  C8 11  A      LEAY  DCBRSZ,U
02120 01216A 0D5F 33  C8 2E  A      LEAU  DCBLRN,U
02130 01217A 0D62 6F  84      A      CLR   ,X
02140 01218A 0D64 6F  01      A      CLR   1,X
02150 01219A 0D66 6F  02      A      CLR   2,X
02160 01220A 0D68 A6  21      A      LDA   1,Y
02170 01221A 0D6A E6  41      A      LDB   1,U
02180 01222A 0D6C 3D      MUL
02190 01223A 0D6D ED  01      A      STD   1,X
02200 01224A 0D6F A6  21      A      LDA   1,Y
02210 01225A 0D71 E6  C4      A      LDB   ,U
02220 01226A 0D73 3D      MUL
02230 01227A 0D74 E3  84      A      ADDD  ,X
02240 01228A 0D76 25  1D 0D95 BCS   CRBAER  IF CARRY
02250 01229A 0D78 ED  84      A      STD   ,X
02260 01230A 0D7A A6  A4      A      LDA   ,Y
02270 01231A 0D7C E6  41      A      LDB   1,U
02280 01232A 0D7E 3D      MUL
02290 01233A 0D7F E3  84      A      ADDD  ,X
02300 01234A 0D81 25  12 0D95 BCS   CRBAER
02310 01235A 0D83 ED  84      A      STD   ,X
02320 01236A 0D85 A6  A4      A      LDA   ,Y
02330 01237A 0D87 E6  C4      A      LDB   ,U
02340 01238A 0D89 3D      MUL
02350 01239A 0D8A EB  84      A      ADDB  ,X
02360 01240A 0D8C 25  07 0D95 BCS   CRBAER
02370 01241A 0D8E E7  84      A      STB   ,X
02380 01242A 0D90 4D      TSTA
02390 01243A 0D91 26  02 0D95 BNE   CRBAER
02400 01244A 0D93 35  F0      A      PULS  X,Y,U,PC
02410 01245A 0D95 B6  10      A      CRBAER LDA  #ERR16
02420 01246A 0D97 35  F0      A      PULS  X,Y,U,PC
02430 01247      *
02440 01248      *****
02450 01249      * CALCULATE TRACK & SECTOR
02460 01250      *
02470 01251      * GIVEN: DCBPRN = RELATIVE RECORD NUMBER
02480 01252      * FUNCTION: FOLLOW CLUSTER CHAIN UNTIL PROPER CLUSTER FOUND
02490 01253      * RESULT: DCBTRK & DCBSEC IF RECORD IN RANGE
02500 01254      * THEY POINT TO LAST SECTOR IF NOT IN RANGE.
02510 01255      * A = ZERO IF SUCCESSFUL
02520 01256      * NON ZERO IF NOT
02530 01257      *****
02540 01258A 0D99 EC  C8 2B  A      CALSEC LDD  DCBRBA,U DESIRED REC NUMBER
02550 01259A 0D9C ED  C8 29  A      LDD  DCBPRN,U SAVE AS THE REC IN THE BUFFER
02560 01260A 0D9F A6  4D      A      CSENT LDA  DCBFCL,U
02570 01261A 0DA1 34  12      A      PSHS  A,X
02580 01262A 0DA3 BE  07C8  A      LDX  #FATS
02590 01263A 0DA6 A6  C8 21  A      LDA  DCBDRV,U
02600 01264A 0DA9 C6  45      A      LDB  #FATSZ
02610 01265A 0DAB 3D      MUL
02620 01266A 0DAC 30  8B      A      LEAX  D,X POINT TO PROPER FAT TABLE
02630 01267A 0DAE EC  C8 29  A      LDD  DCBPRN,U REC NUMBER DESIRED
02640 01268A 0DB1 6D  E4      A      TST  ,S
02650 01269A 0DB3 2B  15 0DCA BMI  CS3 IF AT END OF CLUSTERS (NULL FILE)
02660 01270A 0DB5 B3  0009  A      CS1  SUBD  #9
02670 01271A 0DB8 25  1B 0DD5 BCS  CS4 IF IN THIS CLUSTER

```

```

PAGE 023 RTN      .SA:0      DOS - SUPPORTING SUBROUTINES

02680 01272A 0DBA 34 06      A      PSHS  D
02690 01273A 0DBC A6 62      A      LDA  2,S
02700 01274A 0DBE A6 86      A      LDA  A,X      GET NEXT CLUSTER POINTER
02710 01275A 0DC0 2B 06      0DC0   BMI  CS2      IF AT END OF CLUSTERS
02720 01276A 0DC2 A7 62      A      STA  2,S
02730 01277A 0DC4 35 06      A      PULS  D
02740 01278A 0DC6 20 ED      0DB5   BRA  CS1
02750 01279A 0DC8 35 06      A CS2   PULS  D
02760 01280      *
02770 01281      * REC IS BEYOND END OF CURRENT CLUSTERS
02780 01282      * AM I ALLOWED TO ADD ANOTHER CLUSTER?
02790 01283A 0DCA A6 CB 10  A CS3   LDA  DCBCFS,U
02800 01284A 0DCD 84 08      A      ANDA  #EXTEND  AM I ALLOWED?
02810 01285A 0DCF 26 66      0E37   BNE  CS6      IF YES, GO TRY IT
02820 01286A 0DD1 86 11      A CS3A  LDA  #ERR17  EXTENSION NOT ALLOWED
02830 01287A 0DD3 20 46      0E1B   BRA  CSERR
02840 01288      *
02850 01289      * RECORD IS IN THIS CLUSTER
02860 01290A 0DD5 CB 0A      A CS4   ADDB  #10    (RESULT IS 1-9)
02870 01291      * IS THE SECTOR NUMBER IN B IN USE IN THIS CLUSTER YET?
02880 01292A 0DD7 A6 E4      A      LDA  ,S      (CLUSTER NUMBER)
02890 01293A 0DD9 6D 86      A      TST  A,X      IS THIS CLUSTER THE LAST IN THE FILE?
02900 01294A 0DDB 2A 44      0E21   BPL  CS5      IF NO
02910 01295A 0DDD 34 06      A      PSHS  D      CLUSTER NUMBER/SECTOR NUMBER
02920 01296      * IS THIS RECORD BEYOND CURRENT LAST SECTOR USED?
02930 01297A 0DDF E6 86      A      LDB  A,X
02940 01298A 0DE1 C4 3F      A      ANDB #63    CURRENT LAST SECTOR USED
02950 01299A 0DE3 E1 61      A      CMPB 1,S    THIS ONE
02960 01300A 0DE5 24 08 0DF2  BCC  CS4A   IF THIS IS LESS OR EQUAL TO CURRENT END
02970 01301A 0DE7 E6 C8 10  A      LDB  DCBCFS,U GET FILE STATUS
02980 01302A 0DEA C4 08      A      ANDB #EXTEND FILE EXTENSIONS ALLOWED?
02990 01303A 0DEC 26 18 0E06  BNE  CS4B   IF YES
03000 01304A 0DEE 35 06      A CS4AE  PULS  D
03010 01305A 0DF0 20 DF 0DD1  BRA  CS3A   EXTENSION NOT ALLOWED
03020 01306A 0DF2 26 28 0E1F  CS4A  BNE  CS4C   IF NOT IN LAST SECTOR
03030 01307A 0DF4 A6 CB 10  A      LDA  DCBCFS,U
03040 01308A 0DF7 84 08      A      ANDA  #EXTEND ALLOWED?
03050 01309A 0DF9 26 24 0E1F  BNE  CS4C   IF ITS OK
03060 01310      * IS REC BEYOND LAST BYTE?
03070 01311A 0DFB E6 CB 2D  A      LDB  DCBRBA+2,U
03080 01312A 0DFE 4F      A      CLRA
03090 01313A 0DFE 10A3 4E  A      CMPD  DCBNLS,U
03100 01314A 0E02 25 1B 0E1F  BCS  CS4C   IF OK
03110 01315A 0E04 20 E8 0DEE  BRA  CS4AE  IF NG
03120 01316      * EXTEND LAST SECTOR IN THIS CLUSTER
03130 01317A 0E06 E6 61      A CS4B  LDB  1,S    SECTOR NUMBER
03140 01318A 0E08 CA C0      A      ORB  ##C0
03150 01319A 0E0A E7 86      A      STB  A,X      PUT IN FAT TABLE
03160 01320      * FAT HAS CHANGED - CAN I BYPASS UPDATE THIS TIME?
03170 01321A 0E0C A6 CB 10  A      LDA  DCBCFS,U
03180 01322A 0E0F 84 20      A      ANDA  #FAST
03190 01323A 0E11 26 0C 0E1F  BNE  CS4C   IF YES
03200 01324A 0E13 8D 69 0E7E  BSR  WRWFAT  RE-WRITE FAT TABLE TO REFLECT CHANGE
03210 01325A 0E15 27 08 0E1F  BEQ  CS4C   IF I/O WAS OK
03220 01326A 0E17 35 06      A      PULS  D
03230 01327A 0E19 86 12      A      LDA  #ERR1B  FAT RW ERR
03240 01328A 0E1B A7 E4      A CSERR  STA  ,S
03250 01329A 0E1D 35 92      A      PULS  A,X,PC

```

```

PAGE 024 RTN      .SA:0          DOS - SUPPORTING SUBROUTINES

03260 01330A 0E1F 35 06      A CS4C PULS D          CONTINUE - IT IS NOW WITHIN RANGE OF FILE
03270 01331      * RECORD IS IN RANGE OF FILE - XLATE CLUSTER INTO TRACK & SECTOR
03280 01332A 0E21 A6 E4      A CS5  LDA ,S          CLUSTER NUMBER
03290 01333A 0E23 44          LSRA          IS THIS AN ODD CLUSTER?
03300 01334A 0E24 24 02      0E2B BCC CS5A        IF NO
03310 01335A 0E26 0B 09      A      ADDB #9          IF YES, USE SECTORS 10-18
03320 01336A 0E28 E7 C8 23  A CS5A STB DCBSEC,U
03330 01337A 0E2B 81 11      A      CMPA #17        IS CLUSTER BELOW DIRECTORY?
03340 01338A 0E2D 25 01      0E30 BCS CS5B        IF YES
03350 01339A 0E2F 4C          INCA        IF NOT GO ONE TRACK FARTHER
03360 01340A 0E30 A7 C8 22  A CS5B STA DCBTRK,U
03370 01341A 0E33 6F E4      A      CLR ,S
03380 01342A 0E35 35 92      A      PULS A,X,PC
03390 01343      *
03400 01344      * TRY TO ADD ANOTHER CLUSTER TO THE FILE
03410 01345      * NEXT CLUSTER USED WILL BE THE CLOSEST ONE TO THE LAST ONE USED BY
03420 01346      * THIS FILE. IF FIRST EVER FOR THIS FILE, IT WILL BE CLOSEST TO MIDDLE.
03430 01347A 0E37 E6 E4      A CS6  LDB ,S          LAST CLUSTER NUMBER USED
03440 01348A 0E39 2A 02      0E3D BPL CS6A        IF NOT VERY FIRST ASSIGNED TO FILE
03450 01349A 0E3B C6 22      A      LDB #34          START SEARCH AT CLUSTER 34
03460 01350A 0E3D 4F          CS6A CLRA        STARTING DISPLACEMENT
03470 01351A 0E3E 34 06      A      PSHS D
03480 01352      * LOOP TO LOOK FOR AN AVAILABLE CLUSTER
03490 01353A 0E40 A6 61      A CS7  LDA 1,S        LAST CLUSTER OF FILE
03500 01354A 0E42 AB E4      A      ADDA ,S          ADD DISPLACEMENT
03510 01355A 0E44 81 44      A      CMPA #68        IN RANGE OF TABLE?
03520 01356A 0E46 24 06      0E4E BCC CS7A        IF NO
03530 01357A 0E48 E6 86      A      LDB A,X          GET FAT TABLE BYTE
03540 01358A 0E4A C1 FF      A      CMPB ##FF      IS IT AVAILABLE
03550 01359A 0E4C 27 18      0E69 BEQ CS8          IF YES
03560 01360A 0E4E A6 61      A CS7A LDA 1,S
03570 01361A 0E50 A0 E4      A      SUBA ,S          LOOK THE OTHER WAY
03580 01362A 0E52 25 06      0E5A BCS CS7B        IF NOT IN RANGE OF THE TABLE
03590 01363A 0E54 E6 86      A      LDB A,X          GET FAT TABLE BYTE
03600 01364A 0E56 C1 FF      A      CMPB ##FF      AVAILABLE?
03610 01365A 0E58 27 0F      0E69 BEQ CS8          IF YES
03620 01366A 0E5A A6 E4      A CS7B LDA ,S
03630 01367A 0E5C 4C          INCA
03640 01368A 0E5D A7 E4      A      STA ,S
03650 01369A 0E5F 81 44      A      CMPA #68        HAVE I TRIED ALL POSSIBILITIES?
03660 01370A 0E61 25 DD      0E40 BCS CS7          IF NOT YET
03670 01371A 0E63 35 06      A      PULS D          NORMALIZE STACK
03680 01372A 0E65 86 16      A      LDA #ERR22      DISK FULL
03690 01373A 0E67 20 B2      0E1B BRA CSERR
03700 01374A 0E69 E6 62      A CS8  LDB 2,S        ORIGINAL ENDING CLUSTER
03710 01375A 0E6B 2A 04      0E71 BPL CS8A
03720 01376A 0E6D A7 4D      A      STA DCBFCL,U THIS IS FIRST CLUSTER
03730 01377A 0E6F 20 02      0E73 BRA CS8B
03740 01378A 0E71 A7 85      A CS8A STA B,X          ADD TO CHAIN
03750 01379A 0E73 C6 C0      A CS8B LDB ##C0      SAY NONE OF THESE SECTORS USED
03760 01380A 0E75 E7 86      A      STB A,X
03770 01381A 0E77 35 06      A      PULS D
03780 01382A 0E79 35 12      A      PULS A,X          NORMALIZE STACK
03790 01383A 0E7B 7E 0D9F  A      JMP CSENT        GO TRY AGAIN FROM THE TOP!
03800 01384      *
03810 01385      *****
03820 01386      * REWRITE FAT TABLE ON DIRECTORY TRACK
03830 01387      *

```

```

PAGE 025 RTN      .SA:0      DOS - SUPPORTING SUBROUTINES

03840 01388
03850 01389
03860 01390
03870 01391A 0E7E 34 10 A WRTFAT PSHS X
03880 01392A 0E80 BE C006 A LDX >#C006
03890 01393A 0E83 86 03 A LDA #3 WRITE
03900 01394A 0E85 A7 80 A STA ,X+
03910 01395A 0E87 A6 C8 21 A LDA DCBDRV,U
03920 01396A 0E8A A7 80 A STA ,X+
03930 01397A 0E8C CC 1102 A LDD ##1102 TRACK 17, SECTOR 2
03940 01398A 0E8F ED 81 A STD ,X++
03950 01399A 0E91 EC E4 A LDD ,S ADDR OF FAT TABLE
03960 01400A 0E93 ED 81 A STD ,X++
03970 01401A 0E95 34 08 A PSHS DP
03980 01402A 0E97 4F CLRA
03990 01403A 0E98 1F 8B A TFR A,DP
04000 01404A 0E9A AD 9F C004 A JSR [#C004] DO IO
04010 01405A 0E9E 35 08 A PULS DP
04020 01406A 0EA0 A6 84 A LDA ,X RESULT
04030 01407A 0EA2 35 90 A PULS X,PC
00010 01408 TTL DOS - PAGING & OVERLAYS
*
00020 01409
00030 01410
00040 01411
00050 01412
00060 01413
00070 01414
00080 01415
*****
00090 01416A 0EA4 7E 0ECF A DOS JMP DOS1 JUMP OVER DISPLACEMENTS TO OVERLAYS
00100 01417A 0EA7 0329 A FDB B1-DOS
00110 01418A 0EA9 035C A FDB B2-DOS
00120 01419A 0EAB 037E A FDB B3-DOS
00130 01420A 0EAD 03AE A FDB B4-DOS
00140 01421A 0EAF 0440 A FDB B5-DOS
00150 01422A 0EB1 0529 A FDB B6-DOS
00160 01423A 0EB3 0607 A FDB B7-DOS
00170 01424A 0EB5 0610 A FDB B8-DOS
00180 01425A 0EB7 0619 A FDB B9-DOS
00190 01426A 0EB9 0622 A FDB B10-DOS
00200 01427A 0EBB 06EA A FDB B11-DOS
00210 01428A 0EBD 0762 A FDB B12-DOS
00220 01429A 0EBF 07D9 A FDB B13-DOS
00230 01430A 0EC1 088A A FDB B14-DOS
00240 01431A 0EC3 08B1 A FDB B15-DOS
00250 01432A 0EC5 0984 A FDB B16-DOS
00260 01433A 0EC7 0A49 A FDB B17-DOS
00270 01434A 0EC9 0C0B A FDB B18-DOS
00280 01435A 0ECB 0CA1 A FDB B19-DOS
00290 01436A 0ECD 0D2A A FDB B20-DOS
* MINIMUM INITIALIZATION FOLLOWS
00310 01438A 0ECF DOS1 DOS DO,INIT GO INITIALIZE (MENU ETC)
00320 01439A 0ED5 4D TSTA
00330 01440A 0ED6 27 04 0EDC BEQ DOS2
00340 01441A 0ED8 AD 9F 0616 A JSR [ERROR]
00350 01442A 0EDC 7E 0FF6 A DOS2 JMP OBASIC
00360 01443A 0EDF CC 10A2 A DOS3 LDD #OVRLAY
00370 01444A 0EE2 FD 0625 A STD >OLYLOC
00380 01445A 0EE5 39 RTS

```

```

PAGE 026 ML      .SA:0      DOS - PAGING & OVERLAYS

00390 01446      *
00400 01447      *****
00410 01448      * 8 BIT PRINTER DRIVER
00420 01449      * GIVEN: A=CHR TO BE SENT TO PRINTER
00430 01450      * RETURNED:A IN TACT
00440 01451      *          CC = Z CONDITION IF SENT OK
00450 01452      *          CC = NON-Z IF PRINTER NOT READY - TRY AGAIN
00460 01453      *****
00470 01454      * IS PRINTER READY?
00480 01455A 0EE6 34 16 A DPRNT PSHS D,X
00490 01456A 0EE8 B6 FF22 A LDA >U4BDR
00500 01457A 0EEB 44 LSRA
00510 01458A 0EEC 24 04 0EF2 BCC DP1 IF READY
00520 01459A 0EEE 86 01 A LDA #1 SET NON-Z CONDITION
00530 01460A 0EF0 35 96 A PULS D,X,PC
00540 01461A 0EF2 34 01 A DP1 PSHS CC SAVE INTERRUPT STATUS
00550 01462A 0EF4 DSABLI NO INTERRUPTS DURING HARD LOOP TIMING
00560 01463A 0EF6 A6 61 A LDA 1,S CHR TO SEND
00570 01464A 0EF8 5F CLR B
00580 01465A 0EF9 8D 1A 0F15 BSR LPSND SEND START BIT
00590 01466A 0EFB C6 08 A LDB #8 BITS TO SEND
00600 01467A 0EFD 34 04 A PSHS B LOOP COUNTER
00610 01468A 0EFF 5F DP2 CLR B
00620 01469A 0F00 44 LSRA
00630 01470A 0F01 59 ROL B
00640 01471A 0F02 59 ROL B
00650 01472A 0F03 8D 10 0F15 BSR LPSND SEND THE BIT
00660 01473A 0F05 6A E4 A DEC ,S
00670 01474A 0F07 26 F6 0EFF BNE DP2 GO BACK FOR NEXT BIT
00680 01475A 0F09 35 04 A PULS B
00690 01476      * INITIATE STOP BIT (IT CONTINUES UNTIL PRINTER SAYS "READY")
00700 01477A 0F0B C6 02 A LDB #2
00710 01478A 0F0D F7 FF20 A STB >U4ADR
00720 01479A 0F10 35 01 A PULS CC RESTORE INTERRUPT STATUS
00730 01480A 0F12 4F CLR A SET ZERO CONDITION CODES
00740 01481A 0F13 35 96 A PULS D,X,PC
00750 01482A 0F15 F7 FF20 A LPSND STB >U4ADR LATCH BIT TO OUTPUT
00760 01483A 0F18 BE 0623 A LDX >RATE TIME CONSTANT FOR TRANSMISSION
00770 01484A 0F1B 30 1F A LPDLP LEAX -1,X
00780 01485A 0F1D 26 FC 0F1B BNE LPDLP
00790 01486A 0F1F 39 RTS
00800 01487      *
00810 01488      *****
00820 01489      * TURN ON OR OFF A TIME DRIVEN ROUTINE
00830 01490      * GIVEN: U-> START OF ROUTINE THAT FOLLOWS SPECS
00840 01491      *****
00850 01492A 0F20 34 46 A DTMEON PSHS D,U
00860 01493A 0F22 4D TSTA REQ FOR ON OR OFF?
00870 01494A 0F23 27 0C 0F31 BEQ DTMEOF IF OFF
00880 01495A 0F25 FC 010D A LDD >IR0+1
00890 01496A 0F28 ED 41 A STD 1,U
00900 01497A 0F2A 33 43 A LEAU 3,U
00910 01498A 0F2C FF 010D A STU >IR0+1
00920 01499A 0F2F 35 C6 A PULS D,U,PC
00930 01500      *
00940 01501      *****
00950 01502      * TURN OFF A TIME DRIVEN ROUTINE
00960 01503      * GIVEN: U -> START OF ROUTINE

```

```

PAGE 027 ML .SA:0 DOS - PAGING & OVERLAYS

00970 01504 *****
00980 01505A 0F31 33 43 A DTME0F LEAU 3,U ADDR STORED IN CHAIN
00990 01506A 0F33 34 50 A PSHS X,U
01000 01507A 0F35 CE 010D A LDU #IR0+1
01010 01508A 0F38 AE C4 A DT0 LDX ,U LOOK AT ADDR OF NEXT ROUTINE
01020 01509A 0F3A 8C 0F58 A CMPX #STDIME IS IT END OF CHAIN?
01030 01510A 0F3D 27 0A 0F49 BEQ DT02 IF YES, GET OUT
01040 01511A 0F3F AC 62 A CMPX 2,S IS IT THE ONE SOUGHT?
01050 01512A 0F41 27 0A 0F4D BEQ DT03 IF YES
01060 01513A 0F43 1F 03 A TFR D,U
01070 01514A 0F45 33 5E A LEAU -2,U
01080 01515A 0F47 20 EF 0F38 BRA DT0
01090 01516A 0F49 35 50 A DT02 PULS X,U
01100 01517A 0F4B 35 C6 A PULS D,U,PC
01110 01518 * X POINTING AT DESIRED ROUTINE
01120 01519 * U POINTING AT WHERE THAT POINTER CAME FROM
01130 01520A 0F4D AE 1E A DT03 LDX -2,X GET ADDR THAT DESIRED ROUTINE POINTS TO
01140 01521A 0F4F AF C4 A STX ,U UNLINK HIS ROUTINE
01150 01522A 0F51 35 50 A PULS X,U
01160 01523A 0F53 35 C6 A PULS D,U,PC
01170 01524 *
01180 01525 *****
01190 01526 * STANDARD TIME ROUTINE - LINKED IN BY INITIAL START UP ROUTINE
01200 01527 *****
01210 01528A 0F55 7E 0000 A STMX JMP >0
01220 01529A 0F58 FC 0620 A STDIME LDD >CLOCK
01230 01530A 0F5B A INCD
01240 01531A 0F5E FD 0620 A STD >CLOCK
01250 01532A 0F61 4F CLRA
01260 01533A 0F62 1F 8B A TFR A,DP ENSURE ROM ROUTINE USES PAGE ZERO
01270 01534A 0F64 20 EF 0F55 BRA STMX
01280 01535 *
01290 01536 *****
01300 01537 * CALL A SYSTEM OVERLAY (OR USER OVERLAY)
01310 01538 * GIVEN: OVERLAY NUMBER IN "A"
01320 01539 * OVERLAY IS LOADED IF NOT PRESENT IN MEMORY
01330 01540 * NOTE: X IS NOT PRESERVED - USED FOR OVERLAY BASE ADDRESS
01340 01541 *****
01350 01542A 0F66 8A 80 A USR0LY ORA #80
01360 01543A 0F68 BE 0625 A SYS0LY LDX >OLYLOC POINT AT CURRENT OVERLAY LOAD AREA
01370 01544A 0F6B A1 1F A CMPA -1,X IS THE DESIRED OVERLAY ALREADY THERE?
01380 01545A 0F6D 27 11 0F80 BEQ SYS03 IF YES
01390 01546A 0F6F 34 04 A PSHS B
01400 01547A 0F71 4D TSTA SYSTEM OR USER?
01410 01548A 0F72 28 04 0F78 BMI SYS01 IF USER
01420 01549A 0F74 8D 5C 0FD2 BSR SYSLOD LOAD THE OVERLAY
01430 01550A 0F76 20 06 0F7E BRA SYS02
01440 01551A 0F78 8D 45 0FBF SYS01 BSR USRL0D LOAD THE OVERLAY
01450 01552A 0F7A 27 02 0F7E BEQ SYS02 IF OK
01460 01553A 0F7C 35 84 A PULS B,PC IF LOAD ERROR
01470 01554A 0F7E 35 04 A SYS02 PULS B
01480 01555A 0F80 8E 0F9B A SYS03 LDX #SYS04 WHERE TO GO ON THE WAY BACK FROM THE OVERLAY
01490 01556A 0F83 34 10 A PSHS X
01500 01557A 0F85 BE 0625 A LDX >OLYLOC OVERLAY LOAD AREA
01510 01558A 0F88 30 02 A LEAX 2,X ENTRY POINT WITHIN OVERLAY
01520 01559A 0F8A 34 10 A PSHS X
01530 01560A 0F8C 30 1E A LEAX -2,X PROVIDE USER WITH HIS BASE ADDRESS
01540 01561A 0F8E 34 14 A PSHS B,X

```

```

PAGE 02B ML      .SA:0      DOS - PAGING & OVERLAYS

01550 01562A 0F90 EC 84      A      LDD      ,X      GET SIZE OF OVERLAY
01560 01563A 0F92 30 8B      A      LEAX     D,X      POINT TO END OF OVERLAY
01570 01564A 0F94 30 03      A      LEAX     3,X      POINT TO BASE OF NEXT OVERLAY AREA
01580 01565A 0F96 BF 0625    A      STX      >0LYLOC
01590 01566A 0F99 35 94      A      PULS     B,X,PC   BASE ADDR OF OVERLAY
01600 01567
01610 01568
*
* ON THE WAY BACK, ADJUST 0LYLOC
01620 01569A 0F9B 34 17      A SYS04  PSHS     CC,D,X
01630 01570A 0F9D BE 0625    A      LDX      >0LYLOC
01640 01571A 0FA0 30 1D      A      LEAX     -3,X
01650 01572A 0FA2 EC 84      A      LDD      ,X      GET SIZE OF THIS OVERLAY
01660 01573A 0FA4
01670 01574A 0FA9 30 8B      A      LEAX     D,X      POINT AT BEGINNING OF OVERLAY I AM EXITING
01680 01575A 0FAB BF 0625    A      STX      >0LYLOC   SAVE IT
01690 01576A 0FAE 35 97      A      PULS     CC,D,X,PC
01700 01577
01710 01578
01720 01579
01730 01580
01740 01581
01750 01582
*****
* RETURN FROM ONE OVERLAY & XFER CONTROL TO ANOTHER
* GIVEN: STACK NORMALIZED AS IF READY TO RTS FROM AN OVERLAY
* A = DESIRED OVERLAY NUMBER
*****
01760 01583A 0FB0 8A 80      A DUSRGO  ORA      ##80
01770 01584A 0FB2 34 06      A DGO     PSHS     D      SAVE D
01780 01585A 0FB4 EC 64      A      LDD      4,S      (RET ADDR TO SYS04)
01790 01586A 0FB6 ED 62      A      STD      2,S
01800 01587A 0FB8 CC 0F68    A      LDD      #SYSOLY CAUSE "RETURN" TO SYSOLY AFTER "UNDOING"
01810 01588A 0FBB ED 64      A      STD      4,S
01820 01589A 0FBD 35 86      A      PULS     D,PC     RETURNS TO SYS04
01830 01590
*
01840 01591
*****
* LOAD A SYSTEM OVERLAY (OR USER OVERLAY)
* GIVEN: A = OVERLAY NUMBER
*****
01870 01594
01880 01595A 0FBF 8A 80      A USRL0D  ORA      ##80
01890 01596A 0FC1 34 60      A      PSHS     Y,U
01900 01597A 0FC3 10BE 0627    A      LDY      >USRBSE
01910 01598A 0FC7 CE 0697    A      LDU      #USRDCB
01920 01599A 0FCA 8D 7A 1046  A      BSR      PAGEIN   LOAD THE OVERLAY
01930 01600A 0FCC 27 02 0FD0  A      BEQ      SLDX     IF LOADED OK
01940 01601A 0FCE 86 17      A      LDA      #ERR23
01950 01602A 0FD0 35 E0      A SLDX    PULS     Y,U,PC
01960 01603A 0FD2 34 60      A SYSLOD  PSHS     Y,U
01970 01604A 0FD4 10BE 0EA5  A      LDY      #DOS+1   LOC OF OVERLAY'S RBA TABLE IN MEMORY
01980 01605A 0FDB CE 0635    A      LDU      #DOSDCB  POINT AT SYSTEM'S DCB
01990 01606A 0FDB 8D 69 1046  A      BSR      PAGEIN   LOAD THE OVERLAY
02000 01607A 0FDD 26 02 0FE1  A      BNE      ABORT   IF SYSTEM FAILURE
02010 01608A 0FDF 35 E0      A      PULS     Y,U,PC
02020 01609
*
02030 01610
*****
* FATAL ERROR OCCURRED IN DOS - CAN'T PROCEED
*****
02050 01612
02060 01613A 0FE1 8E 0400    A ABORT   LDX      ##400   VID
02070 01614A 0FE4 CE 100D    A      LDU      #ABTM5G
02080 01615A 0FE7 C6 10      A      LDB      #16
02090 01616A 0FE9 17 FC7B 0C67  A      LBSR     XFRUX
02100 01617A 0FEC CE 0400    A      LDU      ##400
02110 01618A 0FEF C6 F0      A      LDB      #256-16
02120 01619A 0FF1 17 FC73 0C67  A      LBSR     XFRUX

```

```

PAGE 029 ML      .SA:0      DOS - PAGING & OVERLAYS

02130 01620A 0FF4 8D 3F 1035      BSR DERR      WAIT FOR A KEYSTROKE
02140 01621A 0FF6 4D              0BASIC TSTA
02150 01622A 0FF7 27 0D 1006      BE0 0BAS1
02160 01623A 0FF9 10CE 0400      A LDS #STACK
02170 01624A 0FFD BD 0EDF A JSR DOS3      RESET STACK & OLYLOC
02180 01625A 1000              DOS DO,MENU
02190 01626A 1006 7F 0071 A 0BAS1 CLR >#71
02200 01627A 1009 6E 9F FFFE A JMP [ $FFFE ]
02210 01628A 100D 53 A ABTMSG FCC /SYSTEM/
02220 01629A 1013 60 A FCB $60
02230 01630A 1014 46 A FCC /FAILURE/
02240 01631A 101B 6060 A FDB $6060
02250 01632      *
02260 01633      *****
02270 01634      * USER ABORT ROUTINE
02280 01635      * GIVEN: ERROR NUMBER IN A
02290 01636      *****
02300 01637A 101D 1F 89 A DERROR TFR A,B
02310 01638A 101F 86 01 A LDA #1      (ADD 256 TO IT)
02320 01639A 1021 34 06 A PSHS D      SAVE FOR LATER
02330 01640A 1023 CC 0032 A LDD #50     START OF INSTRUCTIONS
02340 01641A 1026 10BE 0045 A LDY #69     END OF INSTRUCTIONS
02350 01642A 102A CE 0000 A LDU #0      CLEAR SCREEN FIRST
02360 01643A 102D 8D 0D 103C BSR DOMAP     GIVE INSTRUCTIONS
02370 01644A 102F 35 06 A PULS D
02380 01645A 1031 1F 02 A TFR D,Y
02390 01646A 1033 8D 07 103C BSR DOMAP     DISPLAY ERROR
02400 01647A 1035      DERR SYSTEM POLCAT WAIT FOR ANY KEYSTROKE
02410 01648A 1039 27 FA 1035      BE0 DERR
02420 01649A 103B 39      RTS
02430 01650      *
02440 01651      *****
02450 01652      * DO MAP DISPLAY FUNCTION
02460 01653      *****
02470 01654A 103C 34 66 A DOMAP PSHS D,Y,U
02480 01655A 103E      DOS DO,MAP
02490 01656A 1044 35 E6 A PULS D,Y,U,PC
02500 01657      *
02510 01658      *****
02520 01659      * L O A D O V E R L A Y R O U T I N E
02530 01660      *
02540 01661      * GIVEN: A=OVERLAY NUMBER
02550 01662      * U-> PROGRAM DCB
02560 01663      * Y-> TABLE CONTAINING RBA'S OF OVERLAYS
02570 01664      * THE FILE MUST HAVE PREVIOUSLY BEEN OPENED!
02580 01665      *****
02590 01666A 1046 BE 0625 A PAGEIN LDX >OLYLOC
02600 01667A 1049 A7 1F A STA -1,X
02610 01668A 104B 1F 89 A TFR A,B
02620 01669A 104D C4 7F A ANDB #$7F
02630 01670A 104F 58      LSLB 2 BYTES PER VECTOR
02640 01671A 1050 4F      CLRA
02650 01672A 1051 EC AB A LDD D,Y      GET RBA OF START OF OVERLAY
02660 01673A 1053 C3 0005 A ADDD #5      ADJUST TO RBA WITHIN DISK FILE
02670 01674A 1056 6F C8 2B A CLR DCBRBA,U
02680 01675A 1059 ED C8 2C A STD DCBRBA+1,U
02690 01676A 105C CC 0002 A LDD #2      LENGTH OF A SIZE FIELD
02700 01677A 105F ED C8 11 A STD DCBRSZ,U SET TO READ 2 BYTES

```

```

PAGE 030 ML      .SA:0      DOS - PAGING & OVERLAYS

02710 01678A 1062 AF C8 27 A      STX  DCBLRB,U
02720 01679A 1065 CC FFFF A      LDD  #FFFF
02730 01680A 1068 ED C8 29 A      STD  DCBPRN,U FORCE INITIAL PHYSICAL READ
02740 01681A 106B 8D 18 1085 BSR  PIRD
02750 01682A 106D 30 02 A      LEAX 2,X
02760 01683A 106F EC D8 27 A      LDD  [DCBLRB,U] LENGTH OF ROUTINE (INCLUDING SIZE WORD)
02770 01684A 1072 AF C8 27 A      STX  DCBLRB,U WHERE REST OF OVERLAY GOES
02780 01685A 1075 30 8B A      LEAX D,X POINT TO END OF OVERLAY + 2
02790 01686A 1077 ED 1E A      STD  -2,X SAVE HIS SIZE AT END
02800 01687A 1079 6F 84 A      CLR  ,X SAY NO VALID OVERLAYS FOLLOW
02810 01688A 107B 83 0002 A      SUBD #2 SIZE OF THE REST
02820 01689A 107E ED C8 11 A      STD  DCBRSZ,U SAVE AS RECORD SIZE
02830 01690A 1081 8D 02 1085 BSR  PIRD
02840 01691A 1083 4F CLRA
02850 01692A 1084 39 RTS
02860 01693A 1085 PIRD DOS  READ,RBA
02870 01694A 108B 27 08 1095 BEQ  PIERX
02880 01695A 108D 32 62 A      LEAS 2,S BYPASS RET ADDR
02890 01696A 108F BE 0625 A      PIERR LDX  >OLYLOC
02900 01697A 1092 6F 84 A      CLR  ,X SAY THIS OVERLAY DOSN'T EXIST IN MEMORY
02910 01698A 1094 4D TSTA SET COND CODES
02920 01699A 1095 39 PIERX RTS
02930 01700 *
02940 01701 *****
02950 01702 * MINIMUM LOGIC TO LOAD & PASS CONTROL TO USER PROGRAM
02960 01703 * JUMP HERE FROM OVERLAY 12
02970 01704 *****
02980 01705A 1096 B12A DOS  READ,RBA READ IN THE ROOT SEGMENT
02990 01706A 109C AE C8 27 A      LDX  DCBLRB,U BASE OF PROGRAM
03000 01707A 109F 1F 15 A      TFR  X,PC JUMP TO ROOT
03010 01708 *
03020 01709A 10A1 00 A      FCB  0 PLACE WHERE NUMBER OF 1ST OVERLAY LOADED GOES
03030 01710 *
03040 01711 *****
03050 01712 * O V E R L A Y   S E C T I O N   F O L L O W S
03060 01713 * ALL SECTIONS THAT FOLLOW ARE RELOCATABLE.
03070 01714 * (THE FIRST OVERLAY IS LOADED AT THIS ADDRESS)
03080 01715 *****
03090 01716 *
03100 01717 * THE FOLLOWING ROUTINE SIMPLY SHIFTS PART OF DOS DOWN TO $989. IT
03110 01718 * IS LOADED AFTER THE END OF THE REST OF THE PGM SO AS TO PREVENT
03120 01719 * CONFLICTS WITH BASIC.
03130 01720 * IT IS CLOBERBERED WHEN FIRST OVERLAY IS LOADED!
03140 01721A 10A2 8E 18D0 A      OVRLAY LDX  #LASTPG
03150 01722A 10A5 CE 0989 A      LDU  #$989
03160 01723A 10A8 10BE 051B A      LDY  #DOS-ORGIN AMOUNT OF PGM TO XFER
03170 01724A 10AC A6 80 A      OVLPLDA  ,X+
03180 01725A 10AE A7 C0 A      STA  ,U+
03190 01726A 10B0 31 3F A      LEAY -1,Y
03200 01727A 10B2 26 FB 10AC BNE  OVLPL
03210 01728 * INITIALIZE VECTORS AT $600
03220 01729A 10B4 CE 0600 A      LDU  #$600
03230 01730A 10B7 8E 1104 A      LDX  #VECINI
03240 01731A 10BA C6 C8 A      LDB  #ENDVEC-VECINI
03250 01732A 10BC BD 0C6F A      JSR  XFRXU MOVE IT TO $600
03260 01733 * FROM THIS POINT ON, VECTORS AT $600 MAY BE USED
03270 01734A 10BF 10CE 0400 A      LDS  #STACK
03280 01735A 10C3 BE 010D A      LDX  >IR0+1 VECTOR TO DISK ROM TIME ROUTINE

```

```

PAGE 031 ML      .SA:0      DOS - PAGING & OVERLAYS

03290 01736A 10C6 30 05 A LEAX 5,X BYPASS CHECK FOR WHICH INTERRUPT IT IS
03300 01737A 10C8 BF 010D A STX >IR0+1 STORE REVISED ENTRY POINT
03310 01738A 10CB CE 0F55 A LDU #STMX
03320 01739A 10CE DOS TIME,ON
03330 01740A 10D4 FC A000 A LDD >POLCAT ADDR OF ROM KBD SCAN ROUTINE
03340 01741A 10D7 FD 061C A STD >KEYIN SAVE IN KEYIN VECTOR
03350 01742 * DETERMINE MEMORY SIZE
03360 01743A 10DA 8E 7FFF A LDX ##7FFF END OF 32K
03370 01744A 10DD A6 84 A LDA ,X
03380 01745A 10DF 43 COMA
03390 01746A 10E0 A7 84 A STA ,X
03400 01747A 10E2 A1 84 A CMPA ,X
03410 01748A 10E4 27 03 10E9 BEQ OVLP1 IF 32K MACHINE
03420 01749A 10E6 8E 3FFF A LDX ##3FFF FOR 16K
03430 01750A 10E9 BF 08DC A OVLP1 STX >MAXMEM
03440 01751A 10EC 86 04 A LDA #4 MAX NUMBER OF DRIVES
03450 01752A 10EE B7 08DE A STA DRIVES
03460 01753A 10F1 CE 0635 A LDU #DOSDCB
03470 01754A 10F4 10CE A040 A LDS #STACK
03480 01755A 10F8 BD 0EDF A JSR DOS3
03490 01756A 10FB DOS OPEN, INPUT READ ONLY
03500 01757A 1101 7E 0EA4 A JMP DOS
03510 01758 *
03520 01759A 1104 0989 A VECINI FDB DOPEN POINTER TO OPEN FUNCTION
03530 01760A 1106 0A52 A FDB DCLOSE
03540 01761A 1108 0AE2 A FDB DREAD
03550 01762A 110A 08EB A FDB DWRITE
03560 01763A 110C 0C40 A FDB DRELSE RELEASE I/O BUFFER
03570 01764A 110E 0F68 A FDB SYSOLY CALL SYSTEM OVERLAY
03580 01765A 1110 0FB2 A FDB DGO JUMP BETWEEN SYSTEM OVERLAYS
03590 01766A 1112 0FD2 A FDB SYSL0D LOAD A SYSTEM OVERLAY
03600 01767A 1114 0F66 A FDB USR0LY CALL USER OVERLAY
03610 01768A 1116 0FB0 A FDB DUSRGO JUMP BETWEEN USER OVERLAYS
03620 01769A 1118 0FBF A FDB USRL0D LOAD A USER OVERLAY
03630 01770A 111A 101D A FDB DERROR USER FATAL ERROR EXIT
03640 01771A 111C 0F20 A FDB DTMEON TIME ROUTINE ON/OFF
03650 01772A 111E 0EE6 A FDB DPRNT 8 BIT PRINTER DRIVER
03660 01773A 1120 0000 A FDB 0 SLOT FOR KEYIN
03670 01774A 1122 0FF6 A FDB 0BASIC RETURN TO BASIC
03680 01775A 1124 0000 A FDB 0 INITIAL CLOCK VALUE
03690 01776A 1126 05 A FCB 5 INITIAL RETRY COUNT
03700 01777A 1127 00AE A FDB #AE PRINTER TIME CONSTANT
03710 01778A 1129 10A2 A FDB OVRLAY LOAD ADDRESS FOR NEXT OVERLAY
03720 01779A 112B 0000 A FDB 0 BASE ADDR OF USER PGM + 1
03730 01780A 112D 0633 A FDB RETURN HOOK1
03740 01781A 112F 0633 A FDB RETURN HOOK2
03750 01782A 1131 0633 A FDB RETURN HOOK3
03760 01783A 1133 0633 A FDB RETURN HOOK4
03770 01784A 1135 0633 A FDB RETURN HOOK5
03780 01785A 1137 3939 A FDB #3939 RETURN CODE FOR HOOKS
03790 01786 * INIT COPY OF DOSDCB
03800 01787A 1139 44 A FCB /DOS BIN/
03810 01788A 1144 00 A FCB 0,0,0,0,0,0,0,0
03820 01789A 114D 00 A FCB 0,0,0,0,0,0,0,0,0,0,0,0
03830 01790A 1159 00 A FCB 0,#FF,0,0
03840 01791A 115D 06CB A FDB SYSBUF
03850 01792A 115F 00 A FCB 0
03860 01793A 1160 06CB A FDB SYSBUF

```

```

PAGE 032 ML      .SA:0      DOS - PAGING & OVERLAYS

03870 01794A 1162      00      A      FCB      0:0:0:0:0:0:0:0
03880 01795      * INIT COPY OF MSGDCB
03890 01796A 116A      44      A      FCC      /DOS      BAS/
03900 01797A 1175      00      A      FCB      0:0:0:0:0:0:0:0
03910 01798A 117E      00      A      FCB      0:0:0:0:0:0:0:0:0:0
03920 01799A 118A      00      A      FCB      0:$FF:0:0
03930 01800A 118E      06CB      A      FDB      SYSBUF
03940 01801A 1190      00      A      FCB      0
03950 01802A 1191      0000      A      FDB      0      (SET WHEN USED)
03960 01803A 1193      00      A      FCB      0:0:0:0:0:0:0:0
03970 01804      * INIT COPY OF USER PGM DCB
03980 01805A 119B      20      A      FCC      /      BIN/
03990 01806A 11A6      00      A      FCB      0:0:0:0:0:0:0:0
04000 01807A 11AF      00      A      FCB      0:0:0:0:0:0:0:0:0:0
04010 01808A 11B8      00      A      FCB      0:$FF:0:0
04020 01809A 11BF      06CB      A      FDB      SYSBUF
04030 01810A 11C1      00      A      FCB      0
04040 01811A 11C2      0000      A      FDB      00
04050 01812A 11C4      00      A      FCB      0:0:0:0:0:0:0:0
04060 01813      *
04070 01814A 11CC      00      A      ENDVEC FCB      0      END OF PRESET DATA
04080 01815      *
00010 01816      *****
00020 01817      * INITIAL START UP - CHECK FOR AUTO EXECUTE
00030 01818      *
00040 01819      *****
00050 01820A 11CD      0033      A B1      FDB      B2-B1      SIZE OF OVERLAY
00060 01821      * CHECK FOR AUTO PROGRAM EXECUTION
00070 01822A 11CF CC      0001      A      LDD      #$1
00080 01823A 11D2 10BE      0001      A      LDY      #$1
00090 01824A 11D6 CE      0000      A      LDU      #$0
00100 01825A 11D9 BD      103C      A      JSR      DOMAP      CLR SCREEN & IF AUTO EXISTS, DISPLAY IT
00110 01826A 11DC BE      0697      A      LDX      #USRDCB
00120 01827A 11DF C6      08      A      LDB      #8      NAME LENGTH
00130 01828A 11E1 A6      C4      A      LDA      ,U      GET 1ST CHR FROM SCREEN
00140 01829A 11E3 81      60      A      CMPA     #$60      IS IT A BLANK?
00150 01830A 11E5 27      13      11FA      BEQ      MENU0     IF YES, NO AUTO FUNCTION
00160 01831A 11E7 A6      C0      A      A STRT1 LDA      ,U+
00170 01832A 11E9 81      60      A      CMPA     #$60
00180 01833A 11EB 25      02      11EF      BCS      STRT2
00190 01834A 11ED 88      40      A      EORA     #$40
00200 01835A 11EF A7      80      A      A STRT2 STA      ,X+
00210 01836A 11F1 5A      *
00220 01837A 11F2 26      F3      11E7      BNE      STRT1
00230 01838A 11F4      *
00240 01839A 11FA      MENU0     DOS      GO,EXEC      GO LOAD & EXECUTE PROGRAM
00250 01840      *
00260 01841      *****
00270 01842      * MAIN MENU SELECTION 2 - EXECUTE A PROGRAM
00280 01843      *****
00290 01844A 1200      0022      A B2      FDB      B3-B2      SIZE OF OVERLAY
00300 01845A 1202 CC      0200      A      LDD      #512      STARTING LINE NUMBER
00310 01846A 1205 10BE      0225      A      LDY      #549      ENDING NUMBER
00320 01847A 1209 CE      0000      A      LDU      #0
00330 01848A 120C BD      103C      A      JSR      DOMAP      DISPLAY SCREEN FORMAT & GET ADDR OF INPUT FIELD
00340 01849      * NOTE U -> FIRST INPUT FIELD ON SCREEN
00350 01850A 120F 8E      0697      A      LDX      #USRDCB      POINT AT DCB
00360 01851A 1212 34      50      A      PSHS     X,U      ADDR OF VID AREA & DEST AREA

```

```

PAGE 033 OLY .SA:0 DOS - PAGING & OVERLAYS
00370 01852A 1214 DOS D0:FIELDI INPUT A FIELD
00380 01853A 121A 35 50 A PULS X,U
00390 01854A 121C DOS GO:EXEC GO EXECUTE IT
00400 01855 *
00410 01856 *****
00420 01857 * MAIN MENU SELECTION 3 - TURN ON CLOCK DISPLAY
00430 01858 *****
00440 01859A 1222 0030 A B3 FDB B4-B3 SIZE OF OVERLAY
00450 01860A 1224 FC 08DC A LDD MAXMEM
00460 01861A 1227 83 00B6 A SUBD #B14-B13+5 ALLOW ROOM FOR CLOCK ROUTINE
00470 01862A 122A FD 08DC A STD MAXMEM
00480 01863A 122D 10BE 0625 A LDY OLYLOC
00490 01864A 1231 34 20 A PSHS Y
00500 01865A 1233 FD 0625 A STD OLYLOC
00510 01866A 1236 CE 0418 A LDU ##400+32-8 DISPLAY AT TOP RIGHT CORNER
00520 01867A 1239 5F CLRBR
00530 01868A 123A 10BE 0007 A LDY #7
00540 01869A 123E DOS D0:REALTM TURN ON DISPLAY
00550 01870A 1244 32 64 A LEAS 4,S NORMALIZE STACK
00560 01871A 1246 35 20 A PULS Y
00570 01872A 1248 10BF 0625 A STY OLYLOC
00580 01873A 124C DOS GO:MENU
00590 01874 *
00600 01875 *****
00610 01876 * MAIN MENU SELECTION 4 - DISPLAY FREE SPACE MAP
00620 01877 *****
00630 01878A 1252 0092 A B4 FDB B5-B4 SIZE
00640 01879A 1254 CC 0046 A LDD #70 START OF SCREEN FORMAT
00650 01880A 1257 10BE 0063 A LDY #99 END OF FORMAT
00660 01881A 125B CE 0000 A LDU #0
00670 01882A 125E 8D 103C A JSR DOMAP DISPLAY FORMAT
00680 01883A 1261 4F CLRRA
00690 01884A 1262 34 02 A PSHS A (DRIVE COUNTER)
00700 01885A 1264 CE 0400 A LDU ##400 VID BUFFER
00710 01886A 1267 8D 6C 12D5 BSR FRES1 FIND STARTING DISPLAY POSN
00720 01887 * LOOP ONCE PER DRIVE
00730 01888A 1269 1183 05FF A FRE1 CMPU ##5FF MORE DISPLAY ROOM?
00740 01889A 126D 24 3D 12AC BCC FREX IF NO
00750 01890A 126F 34 40 A PSHS U SAVE NEXT DISPLAY ADDRESS
00760 01891A 1271 BE C006 A LDX >#C006 POINT AT PARAMETERS
00770 01892A 1274 86 02 A LDA #2 (READ)
00780 01893A 1276 E6 62 A LDB 2,S (DRIVE)
00790 01894A 1278 ED 81 A STD ,X++
00800 01895A 127A CC 1102 A LDD ##1102 (TRK 17, SEC 2)
00810 01896A 127D ED 81 A STD ,X++
00820 01897A 127F CC 06CB A LDD #SYSBUF
00830 01898A 1282 ED 81 A STD ,X++
00840 01899A 1284 86 02 A LDA #2 (ONLY 2 RETRYS)
00850 01900A 1286 B7 0622 A STA >RETRYS
00860 01901A 1289 BD 0D1D A JSR D0I0
00870 01902A 128C 86 05 A LDA #5 (RESTORE TO 5)
00880 01903A 128E B7 0622 A STA >RETRYS
00890 01904A 1291 35 40 A PULS U (DISPLAY LOC)
00900 01905A 1293 10BE 06CB A LDY #SYSBUF
00910 01906A 1297 C6 44 A LDB #68 LOOP COUNT
00920 01907A 1299 A6 84 A LDA ,X RESULT
00930 01908A 129B 27 1B 12B8 BEQ FRES IF OK
00940 01909A 129D 86 58 A FRE3 LDA ##58 (X)

```

```

PAGE 034 OLY .SA:0 DOS - PAGING & OVERLAYS

00950 01910A 129F 8D 32 12D3 BSR FRESET
00960 01911A 12A1 5A DECB
00970 01912A 12A2 26 F9 129D BNE FRE3
00980 01913A 12A4 6C E4 A FRE4 INC ,S DRIVE COUNT
00990 01914A 12A6 A6 E4 A LDA ,S
01000 01915A 12A8 81 04 A CMPA #4 MORE DRIVES TO GO?
01010 01916A 12AA 25 8D 1269 BCS FRE1 IF YES
01020 01917A 12AC 8D 1035 A FREX JSR DERR WAIT FOR A KEYSTROKE
01030 01918A 12AF 35 02 A PULS A
01040 01919A 12B1 DOS GO,MENU
01050 01920A 12B7 39 RTS
01060 01921 * DISPLAY FOR THIS DRIVE
01070 01922A 12B8 A6 A0 A FRE5 LDA ,Y+
01080 01923A 12BA 2B 04 12C0 BMI FRE6 IF PART OR ALL AVAILABLE
01090 01924A 12BC 86 58 A LDA ##58 (X)
01100 01925A 12BE 20 0C 12CC BRA FRE8
01110 01926A 12C0 81 FF A FRE6 CMPA ##FF ALL AVAILABLE?
01120 01927A 12C2 26 04 12C8 BNE FRE7 IF PART USED
01130 01928A 12C4 86 6E A LDA ##6E (PERIOD)
01140 01929A 12C6 20 04 12CC BRA FRE8
01150 01930A 12C8 84 0F A FRE7 ANDA ##F
01160 01931A 12CA 8A 70 A ORA ##70
01170 01932A 12CC 8D 05 12D3 FRE6 BSR FRESET
01180 01933A 12CE 5A DECB
01190 01934A 12CF 26 E7 12B8 BNE FRE5
01200 01935A 12D1 20 D1 12A4 BRA FRE4 GO BACK FOR NEXT DRIVE
01210 01936 *
01220 01937 * STORE CHR ON SCREEN & FIND NEXT DISPLAY POSN
01230 01938A 12D3 A7 5F A FRESET STA -1,U
01240 01939A 12D5 A6 C0 A FRES1 LDA ,U+
01250 01940A 12D7 81 6E A CMPA ##6E PERIOD?
01260 01941A 12D9 27 08 12E3 BEQ FRESX
01270 01942A 12DB 1183 0600 A CMPU ##600 END OF SCREEN?
01280 01943A 12DF 26 F4 12D5 BNE FRES1 IF NO
01290 01944A 12E1 33 5F A LEAU -1,U
01300 01945A 12E3 39 FRESX RTS
01310 01946 *
01320 01947 *****
01330 01948 * MAIN MENU SELECTION 5 - COPY FILES
01340 01949 *****
01350 01950A 12E4 00E9 A B5 FDB B6-B5 SIZE OF OVERLAY
01360 01951A 12E6 34 10 A PSHS X
01370 01952A 12E8 20 62 134C BRA B5A
01380 01953A 12EA 0031 A B5DCB1 RMB DCBSZ
01390 01954A 131B 0031 A B5DCB2 RMB DCBSZ
01400 01955A 134C CC 0226 A B5A LDD #550 START OF FORMAT
01410 01956A 134F 10BE 0257 A LDY #599 END OF FORMAT
01420 01957A 1353 CE 0000 A LDU #0 CLEAR SCREEN FIRST
01430 01958A 1356 8D 103C A JSR D0MAP DISPLAY SCREEN
01440 01959A 1359 C6 07 A LDB #7
01450 01960A 135B DOS D0,INPTS
01460 01961A 1361 C1 03 A CMPB #BREAK
01470 01962A 1363 27 5F 13C4 BEQ B5X
01480 01963 *
01490 01964 * ENTER PUSHED SET UP DCBS
01500 01965A 1365 8E 0400 A B5J LDX ##400
01510 01966A 1368 EE E4 A LDU ,S BASE ADDR
01520 01967A 136A 33 46 A LEAU B5DCB1-B5,U POINT AT SOURCE DCB

```

```

PAGE 035  OLY      .SA:0      DOS - PAGING & OVERLAYS

01530 01968A 136C 8D 07 1375      BSR B5K SET UP SOURCE DCB
01540 01969A 136E 33 C8 31 A LEAU DCBSZ,U POINT AT DEST DCB
01550 01970A 1371 8D 02 1375      BSR B5K
01560 01971A 1373 20 2F 13A4     BRA B5L
01570 01972      * SETUP A DCB
01580 01973A 1375 8D 18 138F B5K BSR B5TAB
01590 01974A 1377 C6 08 A LDB #8
01600 01975A 1379 31 C4 A LEAY ,U
01610 01976A 137B 8D 19 1396     BSR B5MOV
01620 01977A 137D 8D 10 138F     BSR B5TAB
01630 01978A 137F 31 48 A LEAY DCBFEX,U
01640 01979A 1381 C6 03 A LDB #3
01650 01980A 1383 8D 11 1396     BSR B5MOV MOVE EXTENTION
01660 01981A 1385 8D 08 138F     BSR B5TAB
01670 01982A 1387 A6 B4 A LDA ,X
01680 01983A 1389 80 70 A SUBA ##70 (ZERO)
01690 01984A 138B A7 C8 21 A STA DCBDRV,U
01700 01985A 138E 39 RTS
01710 01986A 138F A6 B0 A B5TAB LDA ,X+
01720 01987A 1391 81 5B A CMPA ##5B
01730 01988A 1393 26 FA 138F     BNE B5TAB
01740 01989A 1395 39 RTS
01750 01990A 1396 A6 B0 A B5MOV LDA ,X+
01760 01991A 1398 81 60 A CMPA ##60
01770 01992A 139A 25 02 139E     BCS B5MOV1
01780 01993A 139C 80 40 A SUBA ##40
01790 01994A 139E A7 A0 A B5MOV1 STA ,Y+
01800 01995A 13A0 5A DEC B
01810 01996A 13A1 26 F3 1396     BNE B5MOV
01820 01997A 13A3 39 RTS
01830 01998      *
01840 01999A 13A4 8D E9 138F B5L BSR B5TAB TO Y/N
01850 02000A 13A6 E6 84 A LDB ,X
01860 02001A 13AB C1 59 A CMPB ##59 Y
01870 02002A 13AA 27 04 13B0     BEQ B5M
01880 02003A 13AC C1 4E A CMPB ##4E N
01890 02004A 13AE 26 14 13C4     BNE B5X
01900 02005A 13B0 AE E4 A B5M LDX ,S BASE
01910 02006A 13B2 33 06 A LEAU B5DCB1-B5,X
01920 02007A 13B4 31 B8 37 A LEAY B5DCB2-B5,X
01930 02008A 13B7 DOS DO,COPY
01940 02009A 13BD 4D TSTA
01950 02010A 13BE 27 04 13C4     BEQ B5X
01960 02011A 13C0 AD 9F 0616 A JSR [ERROR]
01970 02012      *
01980 02013A 13C4 35 10 A B5X PULS X
01990 02014A 13C6 DOS GO,MENU
02000 02015A 13C0 39 RTS
02010 02016      *
02020 02017 *****
02030 02018 * DISPLAY SELECTED DIRECTORY LIST
02040 02019 *****
02050 02020A 13CD 00DE A B6 FDB B7-B6
02060 02021A 13CF 30 89 000A A LEAX B6ARG-B6,X
02070 02022A 13D3 34 10 A PSHS X
02080 02023A 13D5 20 2D 1404     BRA B6A
02090 02024A 13D7 00 A B6ARG FCB 0,0
02100 02025A 13D9 20 A FCC /

```

```

PAGE 036  OLY      .SA:0          DOS - PAGING & OVERLAYS

02110 02026A 13E4      20      A      FCC      /
02120 02027A 1404 CC 0258  A B6A  LDD      #600
02130 02028A 1407 10BE 0289  A      LDY      #649
02140 02029A 140B CE 0000  A      LDU      #0
02150 02030A 140E BD 103C  A      JSR      DOMAP  DISPLAY INPUT SCREEN
02160 02031
02170 02032A 1411 C6 03      A      LDB      #3      NUMBER OF FIELDS
02180 02033A 1413      DOS      DO,INPTS GET INPUTS
02190 02034
02200 02035A 1419 EE E4      A      LDU      ,S
02210 02036A 141B 33 42      A      LEAU     2,U      POINT TO NAME
02220 02037A 141D 8E 0400  A      LDX      ##400
02230 02038A 1420 8D 74 1496  BSR      B6TAB
02240 02039A 1422 C6 08      A      LDB      #8
02250 02040A 1424 8D 77 149D  BSR      B6MOV
02260 02041A 1426 8D 6E 1496  BSR      B6TAB
02270 02042A 1428 C6 03      A      LDB      #3
02280 02043A 142A 8D 71 149D  BSR      B6MOV
02290 02044A 142C 8D 68 1496  BSR      B6TAB
02300 02045A 142E A6 80      A      LDA      ,X+
02310 02046A 1430 84 03      A      ANDA    #3
02320 02047A 1432 EE E4      A      LDU      ,S
02330 02048A 1434 A7 C4      A      STA      ,U
02340 02049A 1436 6F 41      A      CLR      1,U
02350 02050
02360 02051A 1438 A6 80      A B6D  LDA      ,X+
02370 02052A 143A 81 6E      A      CMPA    ##6E
02380 02053A 143C 27 15 1453  BEQ     B6E
02390 02054A 143E 91 6F      A      CMPA    #6F /
02400 02055A 1440 27 11 1453  BEQ     B6E
02410 02056A 1442 8C 0600  A      CMPX    ##600
02420 02057A 1445 25 F1 1438  BCS     B6D
02430 02058A 1447 BD 1035  A B6D1 JSR      DERR  WAIT FOR A KEYSTROKE
02440 02059A 144A 35 40      A      PULS    U
02450 02060A 144C      DOS      GO,MENU
02460 02061A 1452 39      RTS
02470 02062A 1453 EE E4      A B6E  LDU      ,S
02480 02063A 1455 34 50      A      PSHS   X,U
02490 02064A 1457      DOS      DO,SCNDR
02500 02065A 145D 35 50      A      PULS   X,U
02510 02066A 145F A6 41      A      LDA      1,U      ENTRY FOUND?
02520 02067A 1461 2B E4 1447  BMI     B6D1  IF NO
02530 02068A 1463 33 4D      A      LEAU    13,U    POINT AT NAME FOUND
02540 02069A 1465 30 1F      A      LEAX   -1,X
02550 02070A 1467 C6 08      A      LDB     #8      MAX NAME LENGTH
02560 02071
02570 02072A 1469 A6 84      A B6F  LDA      ,X
02580 02073A 146B 81 6E      A      CMPA    ##6E
02590 02074A 146D 26 09 1478  BNE     B6G
02600 02075A 146F A6 C0      A      LDA      ,U+
02610 02076A 1471 8A 40      A      ORA     ##40
02620 02077A 1473 A7 80      A      STA     ,X+
02630 02078A 1475 5A      DECB
02640 02079A 1476 26 F1 1469  BNE     B6F
02650 02080
02660 02081A 1478 EE E4      A B6G  LDU      ,S
02670 02082A 147A 33 C8 15  A      LEAU    21,U    POINT AT EXT
02680 02083A 147D A6 80      A      LDA     ,X+

```

```

PAGE 037  OLY      .SA:0      DOS - PAGING & OVERLAYS

02690 02084A 147F B1 6F      A      CMPA  ##6F      /
02700 02085A 1481 26 B5      1438   BNE  B6D      GO GET NEXT ONE
02710 02086A 1483 C6 03      A      LDB  #3
02720 02087A 1485 A6 84      A B6H   LDA  +X
02730 02088A 1487 B1 6E      A      CMPA  ##6E      .
02740 02089A 1489 26 AD      1438   BNE  B6D
02750 02090A 148B A6 C0      A      LDA  +U+
02760 02091A 148D 8A 40      A      ORA  ##40
02770 02092A 148F A7 80      A      STA  +X+
02780 02093A 1491 5A      A      DECB
02790 02094A 1492 26 F1      1485   BNE  B6H
02800 02095A 1494 20 A2      1438   BRA  B6D
02810 02096A 1496 A6 80      A B6TAB LDA  +X+
02820 02097A 1498 B1 5B      A      CMPA  ##5B
02830 02098A 149A 26 FA      1496   BNE  B6TAB
02840 02099A 149C 39      A      RTS
02850 02100A 149D A6 80      A B6MOV LDA  +X+
02860 02101A 149F B1 60      A      CMPA  ##60
02870 02102A 14A1 25 02      14A5   BCS  B6MOV1
02880 02103A 14A3 B0 40      A      SUBA  ##40
02890 02104A 14A5 A7 C0      A B6MOV1 STA  +U+
02900 02105A 14A7 5A      A      DECB
02910 02106A 14A8 26 F3      149D   BNE  B6MOV
02920 02107A 14AA 39      A      RTS
02930 02108      *
02940 02109      *****
02950 02110      * FILL FOR ROUTINES NOT YET WRITTEN
02960 02111      *****
02970 02112      * (OTHER MAIN MENU FUNCTIONS)
02980 02113A 14AB 0009   A B7    FDB  B6-B7    SIZE OF OVERLAY
02990 02114A 14AD      DOS  GO,MENU
03000 02115A 14B3 39      RTS
03010 02116      *
03020 02117A 14B4 0009   A B8    FDB  B7-B8    SIZE OF OVERLAY
03030 02118A 14B6      DOS  GO,MENU
03040 02119A 14BC 39      RTS
03050 02120      *
03060 02121A 14BD 0009   A B9    FDB  B8-B9    SIZE OF OVERLAY
03070 02122A 14BF      DOS  GO,MENU
03080 02123A 14C5 39      RTS
03090 02124      *
03100 02125      *****
03110 02126      * GET SCREEN LINES OUT OF BASIC FILE & DISPLAY
03120 02127      *
03130 02128      * GIVEN IN THE STACK(PUSHED BEFORE CALLING:
03140 02129      * (1,S = RET ADDR TO UNDO)
03150 02130      * (2,S = RET ADDR TO CALLER)
03160 02131      * 4,S STARTING LINE NUMBER DESIRED
03170 02132      * 6,S ENDING LINE NUMBER DESIRED
03180 02133      * 8,S INITIAL DISPLAY LOC
03190 02134      *****
03200 02135A 14C6 00CB   A B10   FDB  B11-B10  OVERLAY SIZE
03210 02136      14C6   A MAPBSE EQU  B10    (ONLY THIS LINE & ONE ABOVE MUST CHG TO USE DIF OVRLAY NBR)
03220 02137A 14CB 20 03 14CD  A      BRA  MAP1    BYPASS LOCALS
03230 02138A 14CA 00      A MAPOSW FCB  0      FILE OPEN SW - 0 WHEN OVERLAY 1ST LOADED, 1 FROM THEN ON
03240 02139A 14CB 0000   A MAPLN FDB  0      LAST LINE NUMBER READ
03250 02140      *
03260 02141A 14CD CE 0666   A MAP1  LDU  #MSGDCB POINT AT DCB

```

```

PAGE 038 OLY      .SA:0      DOS - PAGING & OVERLAYS

03270 02142A 14D0 10BE 0625  A      LDY  >OLYL0C (POINTS BEYOND THIS OVERLAY (WHERE NEXT OVRLAY WOULD GO)
03280 02143A 14D4 10AF C8 27  A      STY  DCBLRB,U USE AS LOGICAL RECORD BUFFER
03290 02144A 14D8 6D 04      A      TST  MAPOSW-MAPBSE,X FILE OPENED?
03300 02145A 14DA 26 16 14F2  A      BNE  MAP3      IF YES
03310 02146      * IF FIRST TIME CALLED, OPEN DISK FILE
03320 02147A 14DC      DOS  OPEN,INPUT OPEN DISK FILE
03330 02148A 14E2 86 01      A      LDA  #1
03340 02149A 14E4 A7 04      A      STA  MAPOSW-MAPBSE,X SAY FILE IS OPEN
03350 02150      * RESET TO BEGINNING OF FILE
03360 02151A 14E6      MAP2 CLR D
03370 02152A 14E8 ED 05      A      STD  MAPLN-MAPBSE,X RESET LAST LINE READ
03380 02153A 14EA ED C8 2B  A      STD  DCBRBA,U
03390 02154A 14ED 86 03      A      LDA  #3 (START READING AT RBA 00 00 03)
03400 02155A 14EF A7 C8 2D  A      STA  DCBRBA+2,U
03410 02156      * CHECK TO SEE IF FILE NEEDS TO BE RESET
03420 02157      * (REQUEST MUST BE > LAST LINE READ)
03430 02158A 14F2 CC FFFF  A      MAP3 LDD  #FFFF
03440 02159A 14F5 ED C8 29  A      STD  DCBPRN,U TO FORCE RE-READ INTO BUFFER
03450 02160A 14F8 EC 05      A      LDD  MAPLN-MAPBSE,X LAST LINE READ
03460 02161A 14FA 10A3 64      A      CMPD 4,S 1ST LINE TO BE DISPLAYED
03470 02162A 14FD 24 E7 14E6  A      BCC  MAP2 GO START OVER AT BOF
03480 02163      * CHECK DISPLAY LOC OPTION
03490 02164A 14FF EC 68      A      LDD  B,S STARTING DISPLAY LOC
03500 02165A 1501 26 12 1515  A      BNE  MAP5 IF ADDRESS GIVEN
03510 02166      * CLEAR THE SCREEN
03520 02167A 1503 CE 0400  A      LDU  #400
03530 02168A 1506 EF 68      A      STU  B,S START DISPLAY AT TOP OF SCREEN
03540 02169A 1508 CC 6060  A      LDD  #6060 BLANKS
03550 02170A 150B 10BE 0100  A      LDY  #256
03560 02171A 150F ED C1      A      MAP4 STD  ,U++
03570 02172A 1511 31 3F      A      LEAY -1,Y
03580 02173A 1513 26 FA 150F  A      BNE  MAP4
03590 02174      *
03600 02175      * READ/DISPLAY LOOP
03610 02176      * READ A LINE
03620 02177A 1515 CE 0666  A      MAP5 LDU  #MSGDCB POINT AT DCB
03630 02178A 1518 CC 0004  A      LDD  #4 LENGTH OF LINE NBR & MEM ADDR
03640 02179A 151B ED C8 11  A      STD  DCBRSZ,U SET TO READ 4 BYTE RECORD
03650 02180A 151E      DOS  READ,RBA
03660 02181A 1524 26 65 158B  A      BNE  MAPERR IF I/O ERROR
03670 02182A 1526 10BE 0625  A      LDY  >OLYL0C (LOGICAL REC BUFFER)
03680 02183A 152A EC A4      A      LDD  ,Y GET "MEMORY ADDRESS"
03690 02184A 152C 27 45 1573  A      BEQ  MAP10 IF AT EOF
03700 02185A 152E EC 22      A      LDD  2,Y GET LINE NUMBER
03710 02186A 1530 ED 05      A      STD  MAPLN-MAPBSE,X SAVE FOR FUTURE REFERENCE
03720 02187A 1532 34 06      A      PSHS D
03730 02188A 1534 6F C8 12  A      CLR  DCBRSZ+1,U SET FOR VARIABLE LENGTH RECORDS
03740 02189A 1537      DOS  READ,RBA READ A STRING
03750 02190A 153D 35 06      A      PULS D
03760 02191A 153F 26 4A 158B  A      BNE  MAPERR IF I/O ERROR
03770 02192A 1541 10A3 64      A      CMPD 4,S IS AT LEAST AS FAR AS STARTING LINE NUMBER?
03780 02193A 1544 25 CF 1515  A      BCS  MAP5 NOT FAR ENOUGH, GO READ ANOTHER
03790 02194A 1546 10A3 66      A      CMPD 6,S IS IT BEYOND LAST ONE?
03800 02195A 1549 27 02 154D  A      BEQ  MAP6 IF THIS IS THE LAST ONE
03810 02196A 154B 24 26 1573  A      BCC  MAP10 IF AT END OF RANGE
03820 02197      * LINE FOUND - XFER IT TO SCREEN
03830 02198A 154D 34 30      A      MAP6 PSHS X,Y
03840 02199A 154F BE 0625  A      LDX  >OLYL0C

```

```

PAGE 039 OLY .SA:0 DOS - PAGING & OVERLAYS

03850 02200A 1552 30 01 A LEAX 1,X (SKIP THE "REM" CODE)
03860 02201A 1554 10AE 6C A LDY 8+4,S DESTINATION ADDRESS
03870 02202 * MOVE CHARACTER LOOP
03880 02203A 1557 A6 80 A MAP7 LDA ,X+ GET A CHARACTER
03890 02204A 1559 A1 C8 13 A CMPA DCBTRM,U IS IT THE TERMINATOR BYTE?
03900 02205A 155C 27 0A 1568 BEQ MAP9 IF YES
03910 02206A 155E 81 40 A CMPA #$40 IS IT SPL CHR?
03920 02207A 1560 24 02 1564 BCC MAPB IF NO
03930 02208A 1562 8A 40 A ORA #$40
03940 02209A 1564 A7 A0 A MAPB STA ,Y+
03950 02210A 1566 20 EF 1557 BRA MAP7
03960 02211A 1568 AE 6C A MAP9 LDX 8+4,S
03970 02212A 156A 30 8B 20 A LEAX 32,X
03980 02213A 156D AF 6C A STX 8+4,S
03990 02214A 156F 35 30 A PULS X,Y
04000 02215A 1571 20 A2 1515 BRA MAP5 GO GET NEXT LINE
04010 02216 * FIND START OF INPUT FIELD
04020 02217A 1573 CE 0400 A MAP10 LDU #$400
04030 02218A 1576 10BE 0200 A LDY #512 MAX CHR TO TEST
04040 02219A 157A 86 5B A LDA #$5B (LEFT BRACKET ON SCREEN)
04050 02220A 157C A1 C0 A MAP11 CMPA ,U+
04060 02221A 157E 27 07 1587 BEQ MAP12
04070 02222A 1580 31 3F A LEAY -1,Y
04080 02223A 1582 26 FB 157C BNE MAP11
04090 02224A 1584 CE 0400 A LDU #$400 IF NO FIELD FOUND
04100 02225A 1587 EF 68 A MAP12 STU 8,S
04110 02226A 1589 4F CLRA
04120 02227A 158A 39 RTS
04130 02228A 158B 86 19 A MAPERR LDA #ERR25
04140 02229A 158D 39 RTS
04150 02230 *
04160 02231 *****
04170 02232 * INPUT A FIELD FROM THE KEYBOARD (ECHO ON THE SCREEN)
04180 02233 *
04190 02234 * GIVEN: (,S = RET TO UNDO)
04200 02235 * (2,S = RET TO CALLER)
04210 02236 * 4,S = ADDR OF INPUT FIELD IN WS
04220 02237 * 6,S = ADDR OF INPUT FIELD ON SCREEN
04230 02238 *****
04240 02239A 158E 0140 A B11 FDB B12-B10 SIZE OF OVERLAY
04250 02240A 1590 EE 66 A LDU 6,S
04260 02241A 1592 10AE 64 A LDY 4,S
04270 02242A 1595 1183 0400 A CMPU #$400 NO FIELD DEFINED?
04280 02243A 1599 27 10 15AB BEQ FLDI2 IF NO FIELD MARKERS
04290 02244 * MOVE ORIG CONTENTS TO SCREEN
04300 02245A 159B A6 C4 A FLDI1 LDA ,U LOOK AT DESTINATION POSITION
04310 02246A 159D 81 5B A CMPA #$5B LEFT BRACKET?
04320 02247A 159F 27 0A 15AB BEQ FLDI2 IF YES
04330 02248A 15A1 81 5D A CMPA #$5D RIGHT BRACKET?
04340 02249A 15A3 27 06 15AB BEQ FLDI2 IF YES
04350 02250A 15A5 A6 A0 A LDA ,Y+
04360 02251A 15A7 A7 C0 A STA ,U+
04370 02252A 15A9 20 F0 159B BRA FLDI1
04380 02253A 15AB BD 1035 A FLDI2 JSR DERR WAIT FOR A KEYSTROKE
04390 02254A 15AE 1F 89 A TFR A,B
04400 02255A 15B0 EE 66 A LDU 6,S
04410 02256A 15B2 10AE 64 A LDY 4,S
04420 02257A 15B5 1183 0400 A CMPU #$400 NO FIELD MARKERS?

```

```

PAGE 040  OLY      .SA:0      DOS - PAGING & OVERLAYS

04430 02258A 15B9 27 4A 1605      BE@  FLDIXX  IF NO FIELD MARKERS, EXIT WITH KEY IN A & B
04440 02259A 15B8 81 20          A      CMPA  ##20  WAS IT LOW CONTROL KEY?
04450 02260A 15B8 25 2A 15E9      BCS  FLDIX  IF YES
04460 02261A 15BF 81 5B          A      CMPA  ##5B  SPL CHR/NUMBERS/UPPER CASE?
04470 02262A 15C1 25 04 15C7      BCS  FLDI4  IF YES
04480 02263A 15C3 81 60          A      CMPA  ##60  HIGH CONTROL CODES?
04490 02264A 15C5 25 22 15E9      BCS  FLDIX  IF YES
04500 02265          * FALL THRU WITH LOWER CASE
04510 02266A 15C7 A6 C4          A FLDI4  LDA    ,U
04520 02267A 15C9 81 5B          A      CMPA  ##5B  IS CURSOR OVER START OF FIELD?
04530 02268A 15CB 27 1C 15E9      BE@  FLDIX  IF YES
04540 02269A 15CD 81 5D          A      CMPA  ##5D  OVER END OF FIELD?
04550 02270A 15CF 27 18 15E9      BE@  FLDIX  IF YES
04560 02271A 15D1 1F 98          A      TFR   B,A
04570 02272A 15D3 A7 A0          A      STA  ,Y+  SAVE CHR IN INPUT AREA
04580 02273A 15D5 81 40          A      CMPA  ##40  SPL CHR?
04590 02274A 15D7 24 02 15DB      BCC  FLDI5  IF YES
04600 02275A 15D9 8A 40          A      ORA   ##40
04610 02276A 15DB A7 C0          A FLDI5  STA  ,U+
04620 02277A 15DD 10AF 64        A FLDI5A STY  4,S
04630 02278A 15E0 EF 66          A      STU   6,S
04640 02279A 15E2 5F          CLRB
04650 02280A 15E3 A6 C4          A      LDA  ,U
04660 02281A 15E5 81 5D          A      CMPA  ##5D  FIELD OVERFLOW?
04670 02282A 15E7 26 C2 15AB      BNE  FLDI2
04680 02283          *
04690 02284          * EXIT WITH LAST KEY PUSHED IN B (ZERO IF FIELD OVERFLOW)
04700 02285A 15E9 81 08          A FLDIX  CMPA  #LEFT (LEFT ARROW?)
04710 02286A 15EB 26 18 1605      BNE  FLDIXX
04720 02287A 15ED 86 20          A      LDA  ##20
04730 02288A 15EF A7 A4          A      STA  ,Y
04740 02289A 15F1 A7 C4          A      STA  ,U
04750 02290A 15F3 A6 5F          A      LDA  -1,U
04760 02291A 15F5 81 5B          A      CMPA  ##5B  IN FIRST POSN NOW?
04770 02292A 15F7 27 04 15FD      BE@  FLDIX1  IF YES
04780 02293A 15F9 31 3F          A      LEAY -1,Y
04790 02294A 15FB 33 5F          A      LEAU -1,U
04800 02295A 15FD 86 20          A FLDIX1 LDA  ##20
04810 02296A 15FF A7 A4          A      STA  ,Y
04820 02297A 1601 A7 C4          A      STA  ,U
04830 02298A 1603 20 D8 15DD      BRA  FLDI5A
04840 02299A 1605 39          FLDIXX RTS
04850 02300          *
04860 02301          *****
04870 02302          * ACTUALLY LOAD AND EXECUTE PROGRAM
04880 02303          * GIVEN: DCB FOR THE PROGRAM FILE STORED
04890 02304          *      IN USRDCB
04900 02305          *****
04910 02306A 1606      0077  A B12  FDB  B13-B12  SIZE OF OVERLAY
04920 02307A 1608 34 10          A      PSHS  X      SAVE MY BASE (LOWEST LOAD ADDRESS ALLOWED)
04930 02308          * STEP 1 OPEN THE PROGRAM FILE - DOES IT EXIST?
04940 02309A 160A CE 0697  A      LDU   #USRDCB
04950 02310A 160D 86 FF          A      LDA  ##FF
04960 02311A 160F A7 C8 21  A      STA  DCBDRV,U SEARCH ALL DRIVES
04970 02312A 1612          DOS  OPEN,INPUT
04980 02313A 1618 27 10 162A      BE@  EX1  IF OK
04990 02314A 161A 81 0D          A      CMPA  #ERR13  NOT PREV CLOSED IS OK
05000 02315A 161C 27 0C 162A      BE@  EX1

```

```

PAGE 041 OLY .SA:0 DOS - PAGING & OVERLAYS

05010 02316A 161E AD 9F 0616 A EXERR JSR [ERROR]
05020 02317A 1622 35 10 A PULS X
05030 02318A 1624 DOS GO,MENU
05040 02319 *
05050 02320 * READ FILE PREFIX DATA (LOAD ADDR, RBA OF 1ST OVERLAY, ETC)
05060 02321A 162A BE 0625 A EX1 LDX >OLYL0C POINT BEYOND ME
05070 02322A 162D AF C8 27 A STX DCBLRB,U USE AS LOGICAL REC BUFFER
05080 02323A 1630 CC 000A A LDD #10 READ 1ST 10 BYTES OF PROGRAM FILE
05090 02324A 1633 ED C8 11 A STD DCBRSZ,U
05100 02325A 1636 DOS READ,RBA
05110 02326A 163C 26 E0 161E BNE EXERR
05120 02327A 163E 6D 84 A TST ,X IS 1ST BYTE ZERO?
05130 02328A 1640 27 04 1646 BEQ EX2 IF YES, OK
05140 02329A 1642 86 1B A LDA #ERR27 WRONG TYPE FILE
05150 02330A 1644 20 D8 161E BRA EXERR
05160 02331A 1646 EC 03 A EX2 LDD 3,X (LOAD ADDRESS)
05170 02332A 1648 27 0C 1656 BEQ EX3A IF BASED AT ZERO, ASSUME RELOCATABLE
05180 02333A 164A 10A3 E4 A CMPD ,S HE MUST LOAD ABOVE THIS POINT
05190 02334A 164D 24 04 1653 BCC EX3 IF HE IS OK
05200 02335A 164F 86 1A A LDA #ERR26 LOAD ADDR IS TOO LOW
05210 02336A 1651 20 CB 161E BRA EXERR
05220 02337 * LOAD ADDRESS IS HIGH ENOUGH
05230 02338A 1653 ED C8 27 A EX3 STD DCBLRB,U SET THIS AS LOGICAL RECORD BUFFER
05240 02339A 1656 EC C8 27 A EX3A LDD DCBLRB,U
05250 02340A 1659 INCD
05260 02341A 165C FD 0627 A STD >USRBASE
05270 02342A 165F EC 08 A LDD 8,X (SHOULD BE RBA OF 1ST OVERLAY)
05280 02343A 1661 ED C8 11 A STD DCBRSZ,U THAT IS ALSO HOW BIG ROOT SECTION IS
05290 02344A 1664 E3 C8 27 A ADDD DCBLRB,U RESULT IS WHERE END OF ROOT WILL BE IN MEMORY
05300 02345A 1667 C3 0003 A ADDD #3
05310 02346A 166A FD 0625 A STD >OLYL0C SET THIS AS BASE OF FUTURE OVERLAYS
05320 02347A 166D 1F 02 A TFR D,Y
05330 02348A 166F 86 FF A LDA #FF INVALIDATE WHICH OVERLAY IS IN OVERLAY AREA
05340 02349A 1671 A7 3F A STA -1,Y
05350 02350A 1673 86 05 A LDA #5
05360 02351A 1675 A7 C8 2D A STA DCBRBA+2,U START READING WITH 6TH BYTE
05370 02352A 1678 35 10 A PULS X
05380 02353A 167A 7E 1096 A JMP B12A GO LOAD ROOT & XFER CONTROL TO IT
05390 02354 *
00010 02355 *****
00020 02356 * RELOCATABLE REAL-TIME CLOCK ROUTINE
00030 02357 *
00040 02358 * DESIGNED TO BE LOADED BY MAINLINE OF USER'S PROGRAM, SAVING ITS
00050 02359 * LOAD ADDRESS. THEN ACCESSED THRU THE SAVED VECTOR TO PERFORM
00060 02360 * FUNCTIONS.
00070 02361 *
00080 02362 * GIVEN: B=0 - INITIAL CALL, LINK SELF INTO TIME INTERRUPT AND PROTECT
00090 02363 * MYSELF FROM BEING OVERLAYED
00100 02364 * B=FF - UNLINK AND RELEASE OVERLAY SPACE
00110 02365 * B=1 - GET TIME
00120 02366 * B=2 - SET TIME
00130 02367 * WITH GET & SET TIME, Y CONTAINS SECONDS AND 60THS OF SECONDS
00140 02368 * U CONTAINS HOURS AND MINUTES
00150 02369 * WITH INITIAL CALL, U -> DISPLAY ADDRESS (0=NO DISPLAY DESIRED)
00160 02370 * Y = 1 FOR HOURS, 2 FOR MINUTES, 4 FOR SECONDS
00170 02371 * OR ANY COMBINATION (ADDED TOGETHER)
00180 02372 *****
00190 02373A 167D 00B1 A B13 FDB B14-B13 OVERLAY SIZE

```

```

PAGE 042 OLY2 .SA:0 DOS - PAGING & OVERLAYS

00200 02374 167D A CLK EQU B13 (TO ALLOW CHANGING TO DIFFERENT OVERLAY DURING DEVELOPMENT)
00210 02375A 167F 20 07 1688 BRA CLK1 JUMP OVER LOCALS
00220 02376A 1681 00 A HRS FCB 0 HOURS (COUNTS TO 255)
00230 02377A 1682 00 A MIN FCB 0 MINUTES (ALL VALUES SET TO ZERO WHEN LOADED)
00240 02378A 1683 00 A SEC FCB 0 SECONDS
00250 02379A 1684 00 A CNT FCB 0
00260 02380A 1685 0000 A TMELOC FDB 0 TIME DISPLAY LOC
00270 02381A 1687 00 A TMEOPT FCB 0 HR,MIN,SEC OPTION
00280 02382A 1688 5D CLK1 TSTB WHICH OPTION?
00290 02383A 1689 27 16 16A1 BEQ CLKG0
00300 02384A 168B 30 1E A LEAX -2,X
00310 02385A 168D 5D TSTB
00320 02386A 168E 2B 2B 168B BMI CLKSTP
00330 02387A 1690 5A DECB
00340 02388A 1691 27 07 169A BEQ CLKGET
00350 02389A 1693 EF 04 A CLKSET STU HRS-CLK,X
00360 02390A 1695 10AF 06 A STY SEC-CLK,X
00370 02391A 1698 4F CLRA
00380 02392A 1699 39 RTS
00390 02393A 169A EE 04 A CLKGET LDU HRS-CLK,X
00400 02394A 169C 10AE 06 A LDY SEC-CLK,X
00410 02395A 169F 4F CLRA
00420 02396A 16A0 39 RTS
00430 02397 *
00440 02398A 16A1 EF 08 A CLKG0 STU TMELOC-CLK,X SAVE DISPLAY ADDRESS
00450 02399A 16A3 1F 20 A TFR Y,D
00460 02400A 16A5 E7 0A A STB TMEOPT-CLK,X SAVE DISPLAY OPTION
00470 02401A 16A7 33 89 004E A LEAU CLKTME-CLK,X POINT AT INTERVAL ROUTINE
00480 02402A 16AB AF 44 A STX 4,U SET LDX COMMAND TO LOAD CURRENT X VALUE
00490 02403A 16AD DOS TIME,ON PLUG IN THE CLOCK
00500 02404A 16B3 EC 62 A LDD 2,S RET ADDR TO CALLER
00510 02405A 16B5 34 06 A PSHS D PUT IN TOP OF STACK TO BYPASS NORMAL EXIT OF OVERLAY
00520 02406A 16B7 4F CLRA
00530 02407A 16B8 30 02 A LEAX 2,X TELL USER WHERE TO ENTER ME
00540 02408A 16BA 39 RTS RETURN TO CALLER
00550 02409 *
00560 02410A 16BB 33 89 004E A CLKSTP LEAU CLKTME-CLK,X POINT AT INTERVAL ROUTINE
00570 02411A 16BF DOS TIME,OFF PULL THE PLUG
00580 02412A 16C5 35 06 A PULS D RET ADDR TO CALLER
00590 02413A 16C7 ED 62 A STD 2,S SET TO RET TO HIM AFTER EXITING FROM OVERLAY
00600 02414A 16C9 4F CLRA
00610 02415A 16CA 39 RTS
00620 02416 *
00630 02417A 16CB 7E 0000 A CLKTME JMP >0
00640 02418A 16CE 8E 0000 A LDX #0 THIS INSTR MODIFIED BY ABOVE ROUTINE
00650 02419A 16D1 EC 06 A LDD SEC-CLK,X LOAD SEC & 60THS
00660 02420A 16D3 5C INCB
00670 02421A 16D4 ED 06 A STD SEC-CLK,X
00680 02422A 16D6 C1 38 A CMPB #56 FULL SECOND?
00690 02423A 16D8 25 F1 16CB BCS CLKTME IF NO, EXIT
00700 02424A 16DA 5F CLRB
00710 02425A 16DB 8B 01 A ADDA #1
00720 02426A 16DD 19 DAA
00730 02427A 16DE ED 06 A STD SEC-CLK,X
00740 02428A 16E0 81 60 A CMPA ##60 FULL MINUTE?
00750 02429A 16E2 25 1A 16FE BCS CLKDSP IF NO
00760 02430A 16E4 4F CLRA
00770 02431A 16E5 A7 06 A STA SEC-CLK,X

```

```

PAGE 043 OLY2 .SA:0 DOS - PAGING & OVERLAYS
00780 02432A 16E7 EC 04 A LDD HRS-CLK,X
00790 02433A 16E9 CB 01 A ADDB #1
00800 02434A 16EB 1E B9 A EXG A,B
00810 02435A 16ED 19 DAA
00820 02436A 16EE 1E B9 A EXG A,B
00830 02437A 16F0 E7 05 A STB MIN-CLK,X
00840 02438A 16F2 C1 60 A CMPB ##60 FULL HOUR?
00850 02439A 16F4 25 08 16FE BCS CLKDSP IF NO
00860 02440A 16F6 5F CLRB
00870 02441A 16F7 8B 01 A ADDA #1
00880 02442A 16F9 19 DAA
00890 02443A 16FA ED 04 A STD HRS-CLK,X
00900 02444A 16FC 20 CD 16CB BRA CLKTME
00910 02445 * DISPLAY RESULTS IF NECESSARY
00920 02446A 16FE EE 08 A CLKDSP LDU TMELOC-CLK,X DISPLAY LOC
00930 02447A 1700 27 C9 16CB BEQ CLKTME EXIT
00940 02448A 1702 E6 0A A LDB TMEOPT-CLK,X DISPLAY OPTION
00950 02449A 1704 54 LSRB
00960 02450A 1705 24 04 170B BCC CLK2 IF NO
00970 02451A 1707 A6 04 A LDA HRS-CLK,X
00980 02452A 1709 8D 10 171B BSR CLKEDT
00990 02453A 170B 54 CLK2 LSRB MINUTES DESIRED?
01000 02454A 170C 24 04 1712 BCC CLK3 IF NO
01010 02455A 170E A6 05 A LDA MIN-CLK,X
01020 02456A 1710 8D 09 171B BSR CLKEDT
01030 02457A 1712 54 CLK3 LSRB SECONDS DESIRED?
01040 02458A 1713 24 B6 16CB BCC CLKTME IF NO
01050 02459A 1715 A6 06 A LDA SEC-CLK,X
01060 02460A 1717 8D 02 171B BSR CLKEDT
01070 02461A 1719 20 B0 16CB BRA CLKTME
01080 02462 * EDIT THE BCD NUMBER IN A - DISPLAY AT U
01090 02463A 171B 34 02 A CLKEDT PSHS A
01100 02464A 171D 44 LSRB
01110 02465A 171E 44 LSRA
01120 02466A 171F 44 LSRA
01130 02467A 1720 44 LSRA
01140 02468A 1721 8B 30 A ADDA ##30
01150 02469A 1723 A7 C0 A STA ,U+
01160 02470A 1725 35 02 A PULS A
01170 02471A 1727 84 0F A ANDA ##0F
01180 02472A 1729 8B 30 A ADDA ##30
01190 02473A 172B A7 C1 A STA ,U++
01200 02474A 172D 39 RTS
01210 02475 *
01220 02476 *****
01230 02477 * DOS MAIN MENU DISPLAY
01240 02478 *****
01250 02479A 172E 0027 A B14 FDB B15-B14 SIZE OF OVERLAY
01260 02480 * DISPLAY DOS MENU SCREEN
01270 02481A 1730 CC 0064 A LDD #100 STARTING LINE NUMBER
01280 02482A 1733 08E 00C7 A LDY #199 END OF RANGE
01290 02483A 1737 CE 0000 A LDU #0 SAY CLEAR SCREEN FIRST
01300 02484A 173A BD 103C A JSR DOMAP DISPLAY MENU MAP
01310 02485A 173D MENU1 SYSTEM POLCAT
01320 02486A 1741 27 FA 173D BEQ MENU1
01330 02487A 1743 80 31 A SUBA ##31 LESS THAN 1?
01340 02488A 1745 27 0B 1752 BEQ MENU2 IF 1 ENTERED (RET TO BASIC)
01350 02489A 1747 25 F4 173D BCS MENU1 IF YES

```

```

PAGE 044 OLY2 .SA:0 DOS - PAGING & OVERLAYS
01360 02490A 1749 81 06 A CMPA #6 NUMBER OF MENU SELECTIONS THAT HAVE BEEN WRITTEN
01370 02491A 174B 24 F0 173D BCC MENU1 IF NOT IN RANGE
01380 02492A 174D 4C INCA TO GET OVERLAY NUMBER OF SERVICE ROUTINE
01390 02493A 174E AD 9F 060C A JSR [G0] PAGE IT IN & GO TO IT
01400 02494A 1752 7E 0FF6 A MENU2 JMP OBASIC
01410 02495 *
01420 02496 *****
01430 02497 * BUFFERED PRINT I/O OVERLAY
01440 02498 *
01450 02499 * TO ACTIVATE:
01460 02500 * LDU #SIZE (TOTAL MEMORY TO USE FOR THIS PURPOSE)
01470 02501 * DOS DO,BUFPR
01480 02502 *
01490 02503 * TO USE:
01500 02504 * LDA CHARACTER TO PRINT
01510 02505 * AGAIN CLR B (SAYS "I AM NOT SHUTTING DOWN")
01520 02506 * JSR [PRNT]
01530 02507 * BNE AGAIN IF BUFFER WAS FULL, TRY AGAIN (OR GO DISPLAY MSG)
01540 02508 *
01550 02509 * TO TERMINATE:
01560 02510 * LDB #1 (ANY NON-ZERO SAYS SHUT DOWN)
01570 02511 * JSR [PRNT]
01580 02512 *
01590 02513A 1755 00D3 A B15 FDB B16-B15 SIZE OF OVERLAY
01600 02514 1755 A BP EQU B15 (FOR USE IN RELATIVE ADDRESSING)
01610 02515A 1757 20 0C 1765 BRA BP1 JUMP OVER LOCALS
01620 02516A 1759 0000 A PRIBUF FDB 0 POINTER TO PRINT BUFFER
01630 02517A 175B 0000 A BUFSZ FDB 0 SIZE OF PRINT BUFFER
01640 02518A 175D 0000 A BUF CNT FDB 0 NUMBER OF CHRS IN BUFFER
01650 02519A 175F 0000 A SNDCHR FDB 0 POINTER INTO BUFFER FOR CHR BEING SENT
01660 02520A 1761 0000 A STRCHR FDB 0 POINTER INTO BUFFER FOR CHR BEING STORED
01670 02521A 1763 0000 A PRNTSV FDB 0 SAVE AREA FOR VECTOR TO ORIG PRNT ROUTINE
01680 02522 *
01690 02523 * SEE IF ENOUGH ROOM PROVIDED
01700 02524A 1765 1F 30 A BP1 TFR U,D PUT SPACE ALLOWED IN D
01710 02525A 1767 83 00D8 A SUBD #BPSZ+5 AMOUNT NOT AVAILABLE FOR BUFFER
01720 02526A 176A 24 03 176F BCC BP1A IF ROOM FOR AT LEAST 1 BYTE BUFFER
01730 02527A 176C 86 18 A LDA #ERR24 BUFFER NOT BIG ENOUGH
01740 02528A 176E 39 RTS
01750 02529 * SET UP FOR BUFFERED PRINTING
01760 02530A 176F C3 0001 A BP1A ADDD #1 (ACTUAL SIZE OF BUFFER)
01770 02531A 1772 ED 06 A STD BUFSZ-BP,X SAVE BUFFER SZ
01780 02532A 1774 31 89 00D6 A LEAY BPSZ+3,X POINT AT BASE OF BUFFER
01790 02533A 1778 10AF 04 A STY PRIBUF-BP,X SAVE IT
01800 02534A 177B 31 AB A LEAY D,Y POINT BEYOND END OF BUFFER
01810 02535A 177D 6F A0 A CLR +Y+ SAY NO VALID OVERLAY FOLLOWS
01820 02536A 177F 10BF 0625 A STY >OLYLOC THIS IS WHERE NEXT OVERLAY GOES
01830 02537A 1783 AF 89 00AB A STX BPTME+4-BP,X MODIFY LDX COMMAND
01840 02538A 1787 AF 89 0055 A STX BPOUT+3-BP,X (SO IT KNOWS WHERE LOCAL WS IS)
01850 02539A 178B 33 89 00A7 A LEAU BPTME-BP,X POINT AT TIME ROUTINE
01860 02540A 178F DOS TIME,ON PLUG IT IN
01870 02541A 1795 FE 061A A LDU >PRNT GET ADDR OF ORIGINAL PRINT ROUTINE
01880 02542A 1798 EF 0E A STU PRNTSV-BP,X SAVE IT
01890 02543A 179A 33 89 0052 A LEAU BPOUT-BP,X POINT AT ENTRY FOR BUFFERED PRINT
01900 02544A 179E FF 061A A STU >PRNT
01910 02545A 17A1 EE 62 A LDU 2+S RET ADR TO USER
01920 02546A 17A3 34 40 A PSHS U (BYPASS NORMAL RETURN THRU UN-DO)
01930 02547A 17A5 4F CLRA SAY DONE OK

```

```

PAGE 045 OLY2 .SA:0      DOS - PAGING & OVERLAYS

01940 02548A 17A6 39      RTS
01950 02549              *
01960 02550              * SEND A CHARACTER TO THE PRINTER VIA BUFFERED I/O
01970 02551A 17A7 34 52  A BPOUT PSHS  A,X,U
01980 02552A 17A9 8E 0000 A      LDX  #0      (THIS INSTR MODIFIED BY SETUP LOGIC)
01990 02553A 17AC 5D      TSTB      REQUEST TO SHUT DOWN?
02000 02554A 17AD 26 30 17DF BNE      BPO3
02010 02555A 17AF EC 08  A BPO1  LDD  BUFcnt-BP,X
02020 02556A 17B1 10A3 06  A      CMPD  BUFSZ-BP,X ROOM FOR MORE?
02030 02557A 17B4 25 06 17BC BCS  BPO1A  IF ROOM
02040 02558A 17B6 86 01  A      LDA   #1      SET NON-Z COND
02050 02559A 17B8 35 52  A      PULS  A,X,U
02060 02560A 17BA 24 F3 17AF BCC  BPO1  IF NO ROOM
02070 02561A 17BC      BPO1A DSABLI
02080 02562A 17BE EE 04  A      LDU   PRtBUF-BP,X
02090 02563A 17C0 EC 0C  A      LDD  STRCHR-BP,X DISPLACEMENT IN BUFFER
02100 02564A 17C2 33 CB  A      LEAU  D,U      POINT AT NEXT STORE POSITION
02110 02565A 17C4 C3 0001 A      ADDD  #1
02120 02566A 17C7 10A3 06  A      CMPD  BUFSZ-BP,X WRAP AROUND?
02130 02567A 17CA 25 02 17CE BCS  BPO2  IF NO
02140 02568A 17CC      CLRd
02150 02569A 17CE ED 0C  A BPO2  STD  STRCHR-BP,X
02160 02570A 17D0 A6 E4  A      LDA   ,5      (CHR TO BE PRINTED)
02170 02571A 17D2 A7 C4  A      STA   ,U
02180 02572A 17D4 EC 08  A      LDD  BUFcnt-BP,X
02190 02573A 17D6 C3 0001 A      ADDD  #1
02200 02574A 17D9 ED 08  A      STD  BUFcnt-BP,X
02210 02575A 17DB      ENABLI
02220 02576A 17DD 35 D2  A      PULS  A,X,U,PC
02230 02577      * WAIT
02240 02578A 17DF EC 08  A BPO3  LDD  BUFcnt-BP,X EMPTY YET?
02250 02579A 17E1 26 FC 17DF BNE  BPO3  IF NO WAIT
02260 02580A 17E3 33 89 00A7 A      LEAU  BPTME-BP,X POINT AT TIME ROUTINE
02270 02581A 17E7      DOS   TIME,OFF UN PLUG IT
02280 02582A 17ED EC 0E  A      LDD  PRntSV-BP,X GET ADDR OF ORIG DRIVER
02290 02583A 17EF FD 061A A      STD  >PRnt  RESTORE IT
02300 02584A 17F2 EC 04  A      LDD  PRtBUF-BP,X WHERE NEXT OVERLAY SHOULD HAVE GONE
02310 02585A 17F4 FD 0625 A      STD  >OLYLc
02320 02586A 17F7 35 40  A      PULS  U      (RET ADDR)
02330 02587A 17F9 ED 62  A      STD  2,S    I'M SET TO RETURN VIA UN-DO)
02340 02588A 17FB 39      RTS
02350 02589              *
02360 02590              * TIME INTERVAL DRIVEN PRINT LOGIC
02370 02591A 17FC 7E 0000 A BPTME JMP  >0      (TO NEXT TIME ROUTINE)
02380 02592A 17FF 8E 0000 A      LDX  #0      (INSTRUCTION MODIFIED BY ABOVE LOGIC)
02390 02593              * IS THERE DATA IN THE BUFFER TO BE SENT TO PRINTER?
02400 02594A 1802 EC 08  A      LDD  BUFcnt-BP,X
02410 02595A 1804 27 F6 17FC BEQ  BPTME  IF NO, EXIT
02420 02596              * TRY TO SEND IT (PRINTER MIGHT NOT BE READY)
02430 02597A 1806 EE 04  A      LDU   PRtBUF-BP,X POINT AT BUFFER
02440 02598A 1808 EC 0A  A      LDD  SNdCHR-BP,X DISPLACEMENT WITHIN BUFFER
02450 02599A 180A A6 CB  A      LDA   D,U    GET CHR OUT OF BUFFER
02460 02600A 180C AD 98 0E A      JSR  [PRntSV-BP,X]
02470 02601A 180F 26 EB 17FC BNE  BPTME  IF PRINTER WAS NOT READY
02480 02602              * ADVANCE BUFFER POINTER
02490 02603A 1811 EC 0A  A      LDD  SNdCHR-BP,X
02500 02604A 1813 C3 0001 A      ADDD  #1
02510 02605A 1816 10A3 06  A      CMPD  BUFSZ-BP,X IS POINTER WRAPPING AROUND END OF BUFFER?

```

```

PAGE 046 OLY2 .SA:0          DOS - PAGING & OVERLAYS

02520 02606A 1819 25 02 181D      BCS  BPT1  IF NO
02530 02607A 181B              CLR D
02540 02608A 181D ED 0A      A BPT1  STD  SNDCHR-BP,X SAVE POINTER TO NEXT CHR
02550 02609              * ADJUST BUFFER COUNT
02560 02610A 181F EC 08      A      LDD  BUF CNT-BP,X
02570 02611A 1821 83 0001    A      SUBD  #1
02580 02612A 1824 ED 08      A      STD  BUF CNT-BP,X
02590 02613A 1826 20 D4 17FC  A      BRA  BPTME  EXIT (ONLY SEND ONE CHR PER INTERRUPT!)
02600 02614              *
02610 02615              *****
02620 02616              * BUFFERED KEYBOARD INPUT OVERLAY
02630 02617              *****
02640 02618A 1828          00C5  A B16  FDB  B17-B16
02650 02619          00D3  A BPSZ  EQU  B16-B15 (FOR PREVIOUS ROUTINE'S USE)
02660 02620          1828  A BK   EQU  B16
02670 02621A 182A 20 0C 183B  A      BRA  BK1   JUMP OVER LOCALS
02680 02622A 182C          0000  A KEYBUF FDB  0      ADDR OF KEYBOARD BUFFER
02690 02623A 182E          0000  A KEYSZ  FDB  0      SIZE OF KBD BUFFER
02700 02624A 1830          0000  A KEYCNT FDB  0      NUMBER OF KEYSTROKES IN BUFFER
02710 02625A 1832          0000  A SNDKEY FDB  0      DISPLACEMENT TO NEXT KEY TO GIVE USER
02720 02626A 1834          0000  A STRKEY FDB  0      DISPLACEMENT FOR STORING NEXT KEYSTROKE
02730 02627A 1836          0000  A KEYSV  FDB  0      SAVE AREA FOR ADDR OF ORIGINAL KBD ROUTINE
02740 02628              *
02750 02629              * SET UP FOR BUFFERED KBD
02760 02630A 1838 1F 30      A BK1  TFR  U,D    PUT SPACE ALLOWED IN D
02770 02631A 183A 83 00CA    A      SUBD  #BKSZ+5 AMOUNT NOT AVAILABLE FOR BUFFER
02780 02632A 183D 24 03 1842  A      BCC  BK1A  IF ROOM FOR AT LEAST 1 BYTE BUFFER
02790 02633A 183F 86 1C      A      LDA  #ERR2B  BUFFER NOT BIG ENOUGH
02800 02634A 1841 39          RTS
02810 02635A 1842 C3 0001    A BK1A  ADDD  #1    (ACTUAL SIZE OF BUFFER)
02820 02636A 1845 ED 06      A      STD  KEYSZ-BK,X SAVE BUF SZ
02830 02637A 1847 31 89 00CB  A      LEAY BKSZ+3,X POINT AT BASE OF BUFFER
02840 02638A 1848 10AF 04      A      STY  KEYBUF-BK,X
02850 02639A 184E 31 AB      A      LEAY  D,Y
02860 02640A 1850 6F A0      A      CLR  ,Y+    SAV NO VALID OVERLAY FOLLOWS
02870 02641A 1852 10BF 0625  A      STY  >OLYLOC  NEXT OVERLAY GOES HERE
02880 02642A 1856 AF 89 0082  A      STX  BKTME+4-BK,X MODIFY LDX INSTR
02890 02643A 185A AF 89 0055  A      STX  BKGIVE+3-BK,X DITTO
02900 02644A 185E 33 89 007E  A      LEAU BKTME-BK,X
02910 02645A 1862          DOS  TIME,ON  PLUG IN TIME RTN
02920 02646A 1868 FE 061C    A      LDU  >KEYIN
02930 02647A 1868 EF 0E      A      STU  KEYSV-BK,X
02940 02648A 186D 33 89 0052  A      LEAU BKGIVE-BK,X
02950 02649A 1871 FF 061C    A      STU  >KEYIN
02960 02650A 1874 EE 62      A      LDU  2,S
02970 02651A 1876 34 40      A      PSHS  U
02980 02652A 1878 4F          CLR A
02990 02653A 1879 39          RTS
03000 02654              *
03010 02655              * POLL FOR A CHARACTER TO GIVE USER
03020 02656A 187A 34 54      A BKGIVE PSHS  B,X,U
03030 02657A 187C 8E 0000    A      LDX  #0    (THIS INSTRUCTION MODIFIED BY SETUP)
03040 02658A 187F EC 08      A      LDD  KEYCNT-BK,X COUNT OF BUFFERED CHRS
03050 02659A 1881 26 02 1885  A      BNE  BKG1  IF ONE TO SEND
03060 02660A 1883 35 D4      A      PULS  B,X,U,PC IF NONE, EXIT WITH A=ZERO
03070 02661A 1885 EE 04      A BKG1  LDU  KEYBUF-BK,X ADDR OF BUFFER
03080 02662A 1887 EC 0A      A      LDD  SNDKEY-BK,X DISPLACEMENT
03090 02663A 1889 J3 CB      A      LEAU  D,U    POINT AT CHARACTER

```

```

PAGE 047 OLY2 .SA:0          DOS - PAGING & OVERLAYS

03100 02664                  * ADJUST POINTER TO NEXT POSITION
03110 02665A 188B C3 0001    A      ADDD  #1
03120 02666A 188E 10A3 06    A      CMPD  KEYSZ-BK,X WRAP AROUND?
03130 02667A 1891 25 F2 1885  BCS  BKG1  IF NO
03140 02668A 1893          CLRDR
03150 02669A 1895 ED 0A    A BKG2  STD  SNDKEY-BK,X
03160 02670A 1897 A6 C4    A      LDA   ,U
03170 02671A 1899 34 02    A      PSHS  A
03180 02672A 189B EC 08    A      LDD  KEYCNT-BK,X
03190 02673A 189D 83 0001  A      SUBD  #1
03200 02674A 18A0 ED 08    A      STD  KEYCNT-BK,X
03210 02675A 18A2 6D E4    A      TST  ,S
03220 02676A 18A4 35 D6    A      PULS  D,X,U,PC
03230 02677          *
03240 02678          * TIME INTERVAL KEYBOARD SCAN ROUTINE
03250 02679A 18A6 7E 0000  A BKTME JMP  >0      TO NEXT TIME ROUTINE
03260 02680A 18A9 8E 0000  A      LDX  #0      (MODIFIED BY SETUP)
03270 02681A 18AC EC 08    A BKTMEA LDD  KEYCNT-BK,X
03280 02682A 18AE 10A3 06  A      CMPD  KEYSZ-BK,X IS BUFFER FULL?
03290 02683A 18B1 25 13 18C6 BCS  BKT1  IF NO
03300 02684          * BUFFER IS FULL - GO BEEP
03310 02685A 18B3 8E FF22  A      LDX  #U4BDR
03320 02686A 18B6 A6 94    A      LDA   ,X
03330 02687A 18B8 88 02    A      EORA  #2      COMPLIMENT SOUND BIT
03340 02688A 18BA A7 84    A      STA   ,X
03350 02689A 18BC 108E 0020 A      LDY  ##20     PULSE WIDTH
03360 02690A 18C0 31 3F    A BKT0   LEAY  -1,Y
03370 02691A 18C2 26 FC 18C0 BNE  BKT0
03380 02692A 18C4 20 E0 18A6 BRA  BKTME  EXIT
03390 02693A 18C6 AD 98 0E  A BKT1   JSR  [KEYSV-BK,X] GO POLL KEYBOARD
03400 02694A 18C9 4D          TSTA
03410 02695A 18CA 27 DA 18A6 BEQ  BKTME  IF NO NEW KEYSTROKES, EXIT
03420 02696A 18CC 34 02    A      PSHS  A      SAVE KEY
03430 02697A 18CE EE 04    A      LDU  KEYBUF-BK,X
03440 02698A 18D0 EC 0C    A      LDD  STRKEY-BK,X DISPLACEMENT TO SAVE LOC
03450 02699A 18D2 33 CB    A      LEAU  D,U      POINT AT SAVE LOC
03460 02700A 18D4 C3 0001  A      ADDD  #1      POINT TO NEXT SAVE LOC
03470 02701A 18D7 10A3 06  A      CMPD  KEYSZ-BK,X WRAP AROUND?
03480 02702A 18DA 25 02 18DE BCS  BKT2  IF NO
03490 02703A 18DC          CLRDR
03500 02704A 18DE ED 0C    A BKT2  STD  STRKEY-BK,X
03510 02705A 18E0 EC 08    A      LDD  KEYCNT-BK,X
03520 02706A 18E2 C3 0001  A      ADDD  #1
03530 02707A 18E5 ED 08    A      STD  KEYCNT-BK,X
03540 02708A 18E7 35 02    A      PULS  A
03550 02709A 18E9 A7 C4    A      STA   ,U
03560 02710A 18EB 20 BF 18AC BRA  BKTMEA  GO CHECK FOR ANOTHER KEY DOWN
03570 02711          *
03580 02712          *****
03590 02713          * COPY FILE OVERLAY
03600 02714          * GIVEN: B (BIT 0) = ZERO IF NO DISK SWAPPING, 1 IF SWAPPING
03610 02715          *      U-> SOURCE FILE DCB (UNOPENED)
03620 02716          *      Y-> DEST FILE DCB (UNOPENED)
03630 02717          * USES MEMORY FROM "OLYLOC" TO "MAXMEM"
03640 02718          * USES LAST LINE ON SCREEN FOR PROMPTS IF SWAPPING DISKETTES
03650 02719          *****
03660 02720A 18ED          01BF  A B17  FDB  B18-B17
03670 02721A 18EF 34 76    A      PSHS  D,X,Y,U

```

```

PAGE 048 OLY2 .SA:0 DOS - PAGING & OVERLAYS
03680 02722A 18F1 32 7A A LEAS -6,S
03690 02723A 18F3 20 60 1955 BRA B17A
03700 02724 * ,S COUNT OF SECTORS IN MEMORY
03710 02725 * 1,S EOF SW
03720 02726 * 2,S=NEXT INPUT PRN
03730 02727 * 4,S=NEXT OUTPUT PRN
03740 02728 * 6,S=PGS AVAIL
03750 02729 * 7,S=SWAP SW
03760 02730 * 8,S=BASE
03770 02731 * 10,S=DEST DCB ADDR
03780 02732 * 12,S=SOURCE DCB ADDR
03790 02733 * 14,S=RET ADDR
03800 02734A 18F5 4C A B17M1 FCC /LOAD SOURCE DISKETTE /
03810 02735A 1915 4C A B17M2 FCC /LOAD DESTINATION DISKETTE /
03820 02736A 1935 4C A B17M3 FCC /LOAD S Y S T E M DISKETTE /
03830 02737 *
03840 02738 * SETUP STACK
03850 02739A 1955 C4 01 A B17A ANDB #1 SET TO 1 OR 0
03860 02740A 1957 A6 C8 21 A LDA DCBDRV,U
03870 02741A 195A A1 A8 21 A CMPA DCBDRV,Y SAME DRIVE?
03880 02742A 195D 27 01 1960 BEQ B17B IF YES
03890 02743A 195F 5F CLR B CLR B
03900 02744A 1960 E7 67 A B17B STB 7,S
03910 02745A 1962 4F CLR A CLR A
03920 02746A 1963 5F CLR B CLR B
03930 02747A 1964 ED E4 A STD ,S
03940 02748A 1966 ED 62 A STD 2,S STARTING INPUT PRN
03950 02749A 1968 ED 64 A STD 4,S STARTING OUTPUT PRN
03960 02750A 196A FC 0BDC A LDD >MAXMEM
03970 02751A 196D B3 0625 A SUBD >OLYLOC HOW MUCH MEM TO WORK WITH
03980 02752A 1970 25 03 1975 BCS B17B1 IF NOT ENOUGH
03990 02753A 1972 4D TSTA
04000 02754A 1973 26 51 19C6 BNE B17C IF AT LEAST 1 PAGE
04010 02755A 1975 86 1D A B17B1 LDA #ERR29 NOT ENOUGH MEM
04020 02756 *
04030 02757 * COMMON EXIT
04040 02758A 1977 A7 66 A B17X STA 6,S
04050 02759A 1979 6D 67 A TST 7,S
04060 02760A 197B 27 10 198D BEQ B17XIT
04070 02761 * RECOVER SYSTEM DISKETTE
04080 02762A 197D AE 68 A LDX 8,S
04090 02763A 197F 30 88 48 A LEAX B17M3-B17,X
04100 02764A 1982 8D 0D 1991 BSR B17WTE
04110 02765A 1984 CE 0666 A LDU #MSGDCB
04120 02766A 1987 DOS OPEN,INPUT TO RE-LOAD FAT TABLE
04130 02767A 198D 32 66 A B17XIT LEAS 6,S
04140 02768A 198F 35 F6 A PULS D,X,Y,U,PC
04150 02769 *
04160 02770 * DISPLAY FLASHING MSG & WAIT FOR DISKETTE SWAP
04170 02771A 1991 10BE 05E0 A B17WTE LDY ##400+512-32 (LAST LINE)
04180 02772A 1995 C6 20 A LDB #32
04190 02773A 1997 A6 80 A B17WT1 LDA ,X+
04200 02774A 1999 A7 A0 A STA ,Y+
04210 02775A 199B 5A DECB
04220 02776A 199C 26 F9 1997 BNE B17WT1
04230 02777A 199E 7F 0621 A CLR >CLOCK+1
04240 02778A 19A1 B17WT2 SYSTEM POLCAT WAIT FOR KEYSTROKE
04250 02779A 19A5 81 0D A CMPA #0D

```

```

PAGE 049 OLY2 .SA:0 DOS - PAGING & OVERLAYS
04260 02780A 19A7 27 1C 19C5 BEQ B17WTX
04270 02781A 19A9 8E 05E0 A LDX ##400+512-32
04280 02782A 19AC B6 0621 A LDA >CLOCK+1
04290 02783A 19AF 84 20 A ANDA ##20
04300 02784A 19B1 48 LSLA
04310 02785A 19B2 34 02 A PSHS A
04320 02786A 19B4 C6 20 A LDB #32
04330 02787A 19B6 A6 84 A B17WT3 LDA ,X
04340 02788A 19B8 84 BF A ANDA #%10111111
04350 02789A 19BA AA E4 A ORA ,S
04360 02790A 19BC A7 80 A STA ,X+
04370 02791A 19BE 5A DECB
04380 02792A 19BF 26 F5 19B6 BNE B17WT3
04390 02793A 19C1 35 02 A PULS A
04400 02794A 19C3 20 DC 19A1 BRA B17WT2
04410 02795A 19C5 39 B17WTX RTS
04420 02796 *
04430 02797A 19C6 A7 66 A B17C STA 6,S PAGES AVAILABLE
04440 02798 *
04450 02799 * LOOP TO COPY FILE
04460 02800A 19C8 EE 6C A B17D LDU 12,S SOURCE
04470 02801A 19CA FC 0625 A LDD >OLYLOC
04480 02802A 19CD ED C8 24 A STD DCBBUF,U
04490 02803A 19D0 6D 67 A TST 7,S SWAPPING?
04500 02804A 19D2 27 06 19DA BEQ B17D0 IF NO
04510 02805 * WAIT FOR SOURCE DISKETTE
04520 02806A 19D4 AE 68 A LDX 8,S
04530 02807A 19D6 30 08 A LEAX B17M1-B17,X
04540 02808A 19D8 8D B7 1991 BSR B17WTE
04550 02809A 19DA B17D0 DOS OPEN,INPUT
04560 02810A 19E0 26 95 1977 BNE B17X IF NOT FOUND
04570 02811A 19E2 EC 62 A LDD 2,S
04580 02812A 19E4 ED C8 29 A STD DCBPRN,U SET STARTING SECTOR NUMBER
04590 02813A 19E7 26 12 19FB BNE B17E IF NOT FIRST TIME
04600 02814 * FIRST TIME - SAVE DIRECTORY DATA IN OUTPUT DCB
04610 02815A 19E9 10AE 6A A LDY 10,S
04620 02816A 19EC 33 48 A LEAU 11,U
04630 02817A 19EE 31 2B A LEAY 11,Y EXCEPT FOR NAME
04640 02818A 19F0 C6 15 A LDB #32-11
04650 02819A 19F2 A6 C0 A B17D1 LDA ,U+
04660 02820A 19F4 A7 A0 A STA ,Y+
04670 02821A 19F6 5A DECB
04680 02822A 19F7 26 F9 19F2 BNE B17D1
04690 02823A 19F9 EE 6C A LDU 12,S SOURCE
04700 02824A 19FB 6F E4 A B17E CLR ,S SECTORS IN MEMORY
04710 02825 *
04720 02826 * LOAD LOOP
04730 02827A 19FD BD 0D9F A B17F JSR CSENT XLATE PRN INTO TRACK & SECTOR
04740 02828A 1A00 26 1E 1A20 BNE B17F1 IF OUT OF RANGE
04750 02829A 1A02 BD 0CE7 A JSR DSKRED DO PHYSICAL I/O
04760 02830A 1A05 26 16 1A1D BNE B17XX IF I/O ERR
04770 02831A 1A07 EC C8 29 A LDD DCBPRN,U
04780 02832A 1A0A C3 0001 A ADDD #1
04790 02833A 1A0D ED C8 29 A STD DCBPRN,U
04800 02834A 1A10 6C C8 24 A INC DCBBUF,U
04810 02835A 1A13 6C E4 A INC ,S COUNT SECTORS READ
04820 02836A 1A15 E6 E4 A LDB ,S
04830 02837A 1A17 E1 66 A CMPB 6,S IS BUFFER FULL

```

```

PAGE 050 OLY2 .SA:0 DOS - PAGING & OVERLAYS

04840 02838A 1A19 26 E2 19FD BNE B17F IF NO
04850 02839A 1A18 20 05 1A22 BRA B17G GO WRITE IT
04860 02840A 1A1D 16 FF57 1977 B17XX LBRA B17X THIS STMT USED AS AN UP-LINK
04870 02841 *
04880 02842 * INPUT AT END - SET EOF SW
04890 02843A 1A20 6C 61 A B17F1 INC 1,S
04900 02844 *
04910 02845 * CLOSE INPUT
04920 02846A 1A22 EC CB 29 A B17G LDD DCBPRN,U
04930 02847A 1A25 ED 62 A STD 2,S SAVE FOR NEXT BATCH
04940 02848A 1A27 * DOS CLOSE,IT
04950 02849A 1A2D A6 E4 A LDA ,S ANY SECTORS READ?
04960 02850A 1A2F 27 EC 1A1D BEQ B17XX IF NO, I'M DONE
04970 02851 *
04980 02852 * OPEN OUTPUT
04990 02853A 1A31 6D 67 A TST 7,S SWAPPING?
05000 02854A 1A33 27 08 1A3D BEQ B17H IF NO
05010 02855A 1A35 AE 68 A LDX 8,S
05020 02856A 1A37 30 88 28 A LEAX B17M2-B17,X
05030 02857A 1A3A 17 FF54 1991 LBSR B17WTE WAIT FOR DESTINATION DISKETTE
05040 02858A 1A3D EE 6A A B17H LDU 10,S OUTPUT FILE DCB
05050 02859A 1A3F FC 0625 A LDD >OLYL0C START OF BUFFER
05060 02860A 1A42 ED CB 24 A STD DCBBUF,U
05070 02861A 1A45 DOS OPEN,OUTPUT+FAST
05080 02862A 1A48 27 06 1A53 BEQ B17H1 IF FILE EXISTS
05090 02863A 1A4D 81 0C A CMPA #12
05100 02864A 1A4F 27 10 1A61 BEQ B17H2 IF CREATED
05110 02865A 1A51 20 CA 1A1D BRA B17XX IF OTHER ERROR
05120 02866 *
05130 02867 * FILE EXISTS
05140 02868A 1A53 EC 64 A B17H1 LDD 4,S
05150 02869A 1A55 26 12 1A69 BNE B17I IF NOT FIRST TIME, ITS OK
05160 02870A 1A57 DOS CLOSE,IT
05170 02871A 1A5D 86 1E A LDA #ERR30
05180 02872A 1A5F 20 BC 1A1D BRA B17XX
05190 02873 *
05200 02874 * FILE CREATED
05210 02875A 1A61 EC 64 A B17H2 LDD 4,S
05220 02876A 1A63 27 04 1A69 BEQ B17I IF FIRST TIME, OK
05230 02877A 1A65 86 1F A LDA #ERR31 MISC ERR
05240 02878A 1A67 20 B4 1A1D BRA B17XX
05250 02879 *
05260 02880A 1A69 ED CB 29 A B17I STD DCBPRN,U
05270 02881 *
05280 02882 * WRITE LOOP
05290 02883A 1A6C BD 0D9F A B17J JSR CSENT XLATE PRN INTO TRACK & SECTOR
05300 02884A 1A6F 26 AC 1A1D BNE B17XX
05310 02885A 1A71 BD 0CEA A JSR DSKWRT WRITE SECTOR
05320 02886A 1A74 1026 FEFF 1977 LBNE B17X
05330 02887A 1A78 EC CB 29 A LDD DCBPRN,U
05340 02888A 1A7B C3 0001 A ADDD #1
05350 02889A 1A7E ED CB 29 A STD DCBPRN,U
05360 02890A 1A81 6C CB 24 A INC DCBBUF,U
05370 02891A 1A84 6A E4 A DEC ,S COUNT DOWN SECTORS WRITTEN
05380 02892A 1A86 26 E4 1A6C BNE B17J
05390 02893 *
05400 02894 * CLOSE OUTPUT
05410 02895A 1A88 EC CB 29 A LDD DCBPRN,U

```

```

PAGE 051 OLY2 .SA:0 DOS - PAGING & OVERLAYS
05420 02896A 1A8B ED 64 A STD 4,S SAVE FOR NEXT BATCH
05430 02897A 1A8D 83 0001 A SUBD #1
05440 02898A 1A90 ED CB 14 A STD DCBMRB,U
05450 02899A 1A93 AE 6C A LDX 12,S SOURCE DCB
05460 02900A 1A95 EC 0E A LDD DCBNLS,X
05470 02901A 1A97 ED 4E A STD DCBNLS,U
05480 02902A 1A99 E7 CB 16 A STB DCBMRB+2,U
05490 02903A 1A9C DOS CLOSE,IT
05500 02904A 1AA2 4F CLRA
05510 02905A 1AA3 6D 61 A TST 1,S AT EOF?
05520 02906A 1AA5 1026 FF74 1A1D LBNE B17XX I'M DONE
05530 02907A 1AA9 16 FF1C 19CB LBRA B17D GO COPY ANOTHER BATCH OF SECTORS
05540 02908 *
05550 02909 *****
05560 02910 * GET MULTIPLE USER INPUTS
05570 02911 * GIVEN B=NUMBER OF INPUTS
05580 02912 *****
05590 02913A 1AAC 0099 A B18 FDB B19-B18
05600 02914 0012 A INPTS EQU 18
05610 02915A 1AAE 86 01 A LDA #1
05620 02916A 1AB0 34 06 A PSHS D
05630 02917A 1AB2 CE 0400 A B18B LDU ##400
05640 02918A 1AB5 E6 E4 A LDB ,S
05650 02919A 1AB7 A6 C0 A B18C LDA ,U+
05660 02920A 1AB9 B1 5B A CMPA ##5B [
05670 02921A 1ABB 27 09 1AC6 BEQ B18D
05680 02922A 1ABD 1183 0600 A CMPU ##600
05690 02923A 1AC1 25 F4 1AB7 BCS B18C
05700 02924A 1AC3 CE 0401 A LDU ##401
05710 02925A 1AC6 5A B18D DECB
05720 02926A 1AC7 26 EE 1AB7 BNE B18C
05730 02927 * INPUT A FIELD
05740 02928A 1AC9 7F 0621 A CLR CLOCK+1
05750 02929A 1ACC 30 5F A LEAX -1,U
05760 02930A 1ACE F6 0621 A B18E LDB CLOCK+1
05770 02931A 1AD1 C4 10 A ANDB #16
05780 02932A 1AD3 27 04 1AD9 BEQ B18E1
05790 02933A 1AD5 86 5B A LDA ##5B
05800 02934A 1AD7 20 02 1ADB BRA B18E2
05810 02935A 1AD9 86 1B A B18E1 LDA ##1B
05820 02936A 1ADB A7 84 A B18E2 STA ,X
05830 02937A 1ADD 34 50 A PSHS X,U
05840 02938A 1ADF SYSTEM POLCAT
05850 02939A 1AE3 35 50 A PULS X,U
05860 02940A 1AE5 4D TSTA
05870 02941A 1AE6 27 E6 1ACE BEQ B18E
05880 02942A 1AE8 B1 03 A CMPA #BREAK
05890 02943A 1AEA 27 50 1B3C BEQ B18X
05900 02944A 1AEC B1 0A A CMPA #DOWN
05910 02945A 1AEE 27 1A 1B0A BEQ B18F
05920 02946A 1AF0 B1 5E A CMPA #UP
05930 02947A 1AF2 27 21 1B15 BEQ B18G
05940 02948A 1AF4 B1 0D A CMPA #ENTER
05950 02949A 1AF6 27 44 1B3C BEQ B18X
05960 02950A 1AF8 B1 08 A CMPA #LEFT
05970 02951A 1AFA 27 34 1B30 BEQ B18I
05980 02952A 1AFC B1 20 A CMPA ##20
05990 02953A 1AFE 25 CE 1ACE BCS B18E

```

```

PAGE 052  OLY2  .SA:0  DOS - PAGING & OVERLAYS

06000 02954A 1800 81 5B A CMPA ##5B
06010 02955A 1802 25 20 1B24 BCS B18H
06020 02956A 1804 81 60 A CMPA ##60
06030 02957A 1806 25 1C 1B24 BCS B18H
06040 02958A 1808 20 C4 1ACE BRA B18E
06050 02959 * DOWN
06060 02960A 180A A6 E4 A B18F LDA +S
06070 02961A 180C A1 61 A CMPA 1+S
06080 02962A 180E 24 0E 1B1E BCC B18G1 IF AT END ALREADY
06090 02963A 1810 4C INCA
06100 02964A 1811 A7 E4 A STA +S
06110 02965A 1813 20 09 1B1E BRA B18G1
06120 02966 * UP
06130 02967A 1815 A6 E4 A B18G LDA +S
06140 02968A 1817 81 01 A CMPA #1
06150 02969A 1819 27 03 1B1E BEQ B18G1
06160 02970A 181B 4A DECA
06170 02971A 181C A7 E4 A STA +S
06180 02972A 181E 86 5B A B18G1 LDA ##5B
06190 02973A 1820 A7 84 A STA +X
06200 02974A 1822 20 8E 1AB2 BRA B18B
06210 02975 * TEXT CHR
06220 02976A 1824 8A 40 A B18H ORA ##40
06230 02977A 1826 A7 C0 A STA +U+
06240 02978A 1828 A6 C4 A LDA +U
06250 02979A 182A 81 5D A CMPA ##5D ]
06260 02980A 182C 27 DC 1B0A BEQ B18F
06270 02981A 182E 20 9E 1ACE BRA B18E
06280 02982 * BACK ARROW
06290 02983A 1830 A6 C2 A B18I LDA +,-U
06300 02984A 1832 84 BF A ANDA ##BF
06310 02985A 1834 81 1B A CMPA ##1B
06320 02986A 1836 26 96 1ACE BNE B18E
06330 02987A 1838 A6 C0 A LDA +U+
06340 02988A 183A 20 F4 1B30 BRA B18I
06350 02989 * BREAK OR ENTER
06360 02990A 183C 1F 89 A B18X TFR A+B
06370 02991A 183E 32 62 A LEAS 2+S
06380 02992A 1840 86 5B A LDA ##5B
06390 02993A 1842 A7 84 A STA +X
06400 02994A 1844 39 RTS
06410 02995 *
06420 02996 *****
06430 02997 * SCAN FOR SELECTED DIRECTORY ENTRY
06440 02998 *****
06450 02999A 1845 0089 A B19 FDB B20-B19
06460 03000 0013 A SCNDIR EQU 19
06470 03001A 1847 BE C006 A LDX $C006 PARAMETER AREA
06480 03002A 184A 86 02 A LDA #2 READ
06490 03003A 184C A7 80 A STA +X+
06500 03004A 184E A6 C4 A LDA +U DRIVE
06510 03005A 1850 A7 80 A STA +X+
06520 03006A 1852 CC 1103 A LDD ##1103 TRACK & SECTOR
06530 03007A 1855 ED 80 A STD +X+ LEAVE X -> SECTOR
06540 03008A 1857 108E 06CB A LDY #SYBUF
06550 03009A 185B 10AF 01 A STY 1,X
06560 03010A 185E A6 41 A LDA 1,U STARTING OCCURANCE
06570 03011A 1860 81 48 A B19A CMPA #72 ANY MORE ON THIS DRIVE?

```

```

PAGE 053 0LY2 .SA:0 DOS - PAGING & OVERLAYS
06580 03012A 1B62 24 65 1BC9 BCC B19NO
06590 03013A 1B64 80 08 A B19E SUBA #8
06600 03014A 1B66 25 04 1B6C BCS B19C IF IN THIS SECTOR
06610 03015A 1B68 6C 84 A INC ,X
06620 03016A 1B6A 20 F8 1B64 BRA B19B
06630 03017A 1B6C 88 08 A B19C ADDA #8
06640 03018A 1B6E 27 07 1B77 BEQ B19D
06650 03019A 1B70 C6 20 A LDB #32
06660 03020A 1B72 3D MUL DISPLACEMENT IN THIS SECTOR
06670 03021A 1B73 31 AB A LEAY D,Y OFFSET TO 1ST ENT TO SCAN
06680 03022A 1B75 20 0C 1B83 BRA B19D1
06690 03023A 1B77 34 70 A B19D PSHS X,Y,U
06700 03024A 1B79 AD 9F C004 A JSR [#C004]
06710 03025A 1B7D 35 70 A PULS X,Y,U
06720 03026A 1B7F A6 03 A LDA 3,X RESULT
06730 03027A 1B81 26 46 1BC9 BNE B19NO IF I/O ERR
06740 03028 * COMPARE AGAINST ARGUMENT
06750 03029 * REGISTERS:X->SECTOR NBR
06760 03030 * Y->ENTRY IN BUFFER
06770 03031 * U->SEARCH ARGUMENT
06780 03032A 1B83 34 60 A B19D1 PSHS Y,U
06790 03033A 1B85 C6 08 A LDB #11 BYTES TO COMPARE
06800 03034A 1B87 33 42 A LEAU 2,U TO START OF ARGUMENT
06810 03035A 1B89 A6 A4 A LDA ,Y
06820 03036A 1B8B 27 0C 1B99 BEQ B19E1 IF EMPTY ENTRY
06830 03037A 1B8D 2B 0A 1B99 BMI B19E1 IF END OF DIRECTORY
06840 03038A 1B8F A6 C0 A B19E LDA ,U+
06850 03039A 1B91 81 2A A CMPA #'* WILDCARD?
06860 03040A 1B93 27 1F 1B84 BEQ B19F
06870 03041A 1B95 A1 A0 A CMPA ,Y+
06880 03042A 1B97 27 1D 1B86 BEQ B19G
06890 03043 * NO MATCH
06900 03044A 1B99 35 60 A B19E1 PULS Y,U
06910 03045A 1B9B 6C 41 A INC 1,U
06920 03046A 1B9D A6 41 A LDA 1,U
06930 03047A 1B9F 81 48 A CMPA #72 ANY MORE?
06940 03048A 1BA1 24 26 1BC9 BCC B19NO
06950 03049A 1BA3 31 AB 20 A LEAY 32,Y POINT AT NEXT ENTRY
06960 03050A 1BA6 108C 07C8 A CMPY #SYSBUF+256
06970 03051A 1BA8 25 D7 1B83 BCS B19D1
06980 03052 * READ NEXT SECTOR
06990 03053A 1BAC 108E 06C8 A LDY #SYSBUF
07000 03054A 1BB0 6C 84 A INC ,X
07010 03055A 1BB2 20 C3 1B77 BRA B19D
07020 03056A 1BB4 A6 A0 A B19F LDA ,Y+ BYPASS SOURCE CHR
07030 03057A 1BB6 5A B19G DECB B19E
07040 03058A 1BB7 26 D6 1B8F BNE B19E
07050 03059 * MATCH FOUND
07060 03060A 1BB9 35 60 A PULS Y,U
07070 03061A 1BBB 6C 41 A INC 1,U SEARCH CONTINUES WITH NEXT ENTRY
07080 03062A 1BBD 33 4D A LEAU 2+11,U
07090 03063A 1BBF C6 20 A LDB #32
07100 03064A 1BC1 A6 A0 A B19H LDA ,Y+
07110 03065A 1BC3 A7 C0 A STA ,U+
07120 03066A 1BC5 5A DECB
07130 03067A 1BC6 26 F9 1BC1 BNE B19H
07140 03068A 1BC8 39 RTS
07150 03069A 1BC9 86 FF A B19NO LDA #FF

```

```

PAGE 054 OLY2 .SA:0          DOS - PAGING & OVERLAYS

07160 03070A 1BCB A7 41      A      STA      1+U      SAY NO MORE
07170 03071A 1BCD 39          A      RTS
07180 03072                *
07190 03073A 1BCE 0001      A B20  RMB  1
07200 03074A 1BCF 0001      A B21  RMB  1
07210 03075                *
07220 03076                OPT  L
07230 03077A 1BD0 0001      A LASTPG RMB  1      END OF OVERLAYS
07240 03078 1BD0 0001      A B22  EQU  LASTPG
07250 03079 00C5 0001      A BKSZ  EQU  B17-B16
07260 03080 0718 0001      A PGMSZ EQU  OVRLAY-ORGIN-1
07270 03081 1246 0001      A TOTSZ EQU  LASTPG-ORGIN-1
07280 03082 0EA4 0001      A START EQU  DOS      START OF DISK FILE
07290 03083 20E8 0001      A END   EQU  LASTPG+DOS-ORGIN END OF DISK FILE
07300 03084 10A2 0001      A ENTRY EQU  OVRLAY  INITIAL ENTRY POINT INTO PROGRAM
07310 03085 10A5 0001      A LOWUSR EQU  OVRLAY+3 LOWEST POINT WHERE USER PGM CAN LOAD
07320 03086 0BB9 0001      A FIXIT EQU  $1E00-LASTPG+ORGIN POINT THAT BASIC CLOBBERS
07330 03087                TTL  DOS - CROSS REFERENCE
07340 03088                END

TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

```

Index

&	35
\$	35
@	35
[43
]	43
/AO (Absolute origin)	25
/IM (Assemble into memory)	25
/IM Switch	27
/LP (Assembler listing)	25
/MO (Manual origin)	25, 29
/NL (No listing)	25
/NO (No object code in memory)	25
/NS (No symbol table)	25
/SR (Single record)	25
/SR "switch"	7
/SS (Short screen listing)	25
/WE (Wait on assembly errors)	25
/WS (With symbols)	25
6809	41
6809 Mnemonics, Reference	109
6809 Registers	41
— A —	
Absolute Origin Assembly	28
Addressing-Mode Characters	10
Addressing Modes	42
Direct Addressing	45
Extended Addressing	43
Immediate Addressing	43
Indexed Addressing	43
Indirect Addressing	43
Inherent Addressing	43
Relative Addressing	44
Alphanumeric Character Codes	106
Arithmetic Operators	36
Addition (+)	36
Subtraction (-)	36
Multiplication (*)	36
Division (.DIV.)	36
Modulus (.MOD.)	36
Positive (+)	36
Negative (-)	36
ASCII Codes	105
Alphanumeric Character	106
Color	105
Graphic Character	105
Video Control	105
ASCII Mode	18
Assembler Commands	25
Assembler Commands and Switches, Reference	75
Assembler Pseudo Ops, Reference	85
Assembling	25
Assembling for DOS	30
Assembling for Stand-Alone ZBUG	30
Assembly Display Listing	26
Assembly Listing, Changing	49
COND	49
ENDC	49
INCLUDE	50
OPT	49
PAGE	49
TITLE	49
— B —	
Backups	3
BASIC Command	23
Breakpoints	32
Buffers	61
Byte Mode	17
— C —	
Changing Memory	18
CHROUT	58
Clock Display	16
Closing a Disk File	62
Color Codes	105
Command	42
Complex Operations	37
COND	49
Controlling Assembly Origin	47
END	47
ORG	47
Copy Command	22
<i>Cstartline, range, increment</i>	22
Copy Files	15
<i>Cstartline, range, increment</i>	22
— D —	
Data Control Block	61
Data Control Block (DCB), Reference	91
Defining Symbols	47
EQU	48
SET	48
Delete Command	22
<i>Drange</i>	22
Direct Access	65
Direct Addressing	45
Directory	15
Disk Allocation Map	15
Disk Assembly	30
Assembling for DOS	30
Assembling Stand-Alone ZBUG	30
Display Modes	31
Half-Symbolic Mode	32

Operands	36
Operators	36
Arithmetic	36
Logical	37
Relational	37
Complex Operations	37
Operators	10, 36
Arithmetic	36
Logical	37
Relational	37
OPT	49
ORG	47
Origination Offset Assembly	28
Output Mode	35

— P —

PAGE	49
PAGLEN	27
PAGWID	27
POLCAT	57
<i>Prange</i>	21
Print Command	21
<i>Prange</i>	21
Printer Commands	21
<i>Hrange</i>	21
<i>Trange</i>	21
Processor	9
Registers	9
Opcode	9
Program Editor Commands	21
Copy Command	22
<i>Cstartline, range,</i> <i>increment</i>	22
Delete Command	22
<i>Drange</i>	22
Edit Command	21
<i>Eline</i>	21
Insert Command	22
<i>Istartline, increment</i>	22
Load Command	23
LD <i>filespec</i>	23
LDA <i>filespec</i>	23
Print Command	21
<i>Prange</i>	21
Printer Commands	21
<i>Hrange</i>	21
<i>Trange</i>	21
Renumber Command	22
<i>Nstartline, increment.</i>	22
Replace Command	22
<i>Rstartline, increment.</i>	22
Write Command	23
WD <i>filespec</i>	23
ZBUG Command	22

Pseudo Ops.	10, 47
Pseudo Ops, Reference	85

— R —

Read/Write Option	66
Reading a Disk File	65
Read to a File Sample Program	67
Registers	9
6809	41
Registers, Examining	33
Relational Operators	37
Equal to (.EQU.)	37
Not Equal to (.NEQ.)	37
Relative Addressing	44
Renumber Command	22
<i>Nstartline, increment</i>	22
Replace Command	22
<i>Rstartline, increment</i>	22
Right bracket (])	6
RMB	48
ROM Routines	10, 57
CHROUT	58
POLCAT	57
ROM Routines, Reference	89
Routines	
DOS	10
ROM	10
<i>Rstartline, increment</i>	22

— S —

Sample Program	5, 11
Sample Programs	125
Saving Memory from ZBUG	34
SET	48
Sequential Access	65
Single Stepping	33
Switches	
/AO	25
/IM	25
/LP	25, 27
/MO	25
/NL	25, 27
/NO	25
/NS	25, 27
/SR	7, 25
/SS	25
/WE	25, 27
/WS	25
Symbolic Mode	32
Symbols	10
Examine Memory	32

— T —

TITLE	49
<i>Trange</i>	21
Transferring Memory Blocks	33

— V —

Video Control Codes	105
---------------------------	-----

— W —

WD <i>filespec</i>	23
Word Mode	18

Write Command	23
WD <i>filespec</i>	23
Write to a File Sample Program	67
Writing a Disk File	65

— Z —

ZBUG Calculator	35
ZBUG Command	22
ZBUG Commands	17, 31
ZBUG Commands Reference	77

RADIO SHACK, A DIVISION OF TANDY CORPORATION

**U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5**

TANDY CORPORATION

AUSTRALIA

**91 KURRAJONG ROAD
MOUNT DRUITT, N.S.W. 2770**

BELGIUM

**PARC INDUSTRIEL DE NANINNE
5140 NANINNE**

U. K.

**BILSTON ROAD WEDNESBURY
WEST MIDLANDS WS10 7JN**