

DOS Command Processor#

Super Command Processor

by: Bryan Schappel

If the typed line does not contain a command, then the filename entered will be loaded as a binary file.

```
0100      .OPT NO LIST
0110 ;-----
0120 ;
0130 ;Super Command Processor
0140 ;
0150 ;by: Bryan Schappel
0160 ;
0170 ;If the typed line does not
0180 ;contain a command, then the
0190 ;filename entered will be loaded
0200 ;as a binary file.
0210 ;
0220 ;-----
0230 ;
0240 ;*** Commands ***
0250 ;
0260 ;ERA [fname] = delete file
0270 ;REN [f1,f2] = rename f1 to f2
0280 ;PRO [fname] = lock file
0290 ;UNP [fname] = unlock file
0300 ;DIR [Dn:] = directory of D#:
0310 ;RUN [adr] = run at address
0320 ;COPY      = copy a File
0330 ;CAR      = enter Cart
0340 ;TYPE      = read BATCH file
0350 ;KILL      = KILL BBK CP
0360 ;
0370 NUMCOM = 10      ;# OF COMMANDS
0380 IOCB5 = $0390   ;Adr of IOCB5
0390 MYBUF = $0500   ;I/O Buffer
0400 ZIOCB = $20     ;Zpage IOCB
0410 ;
0420 ;Zero Page Variables
0430 ;
0440      *=   $D6
0450 SL      1      ;Load adr lo
0460 SH      1      ;Load adr hi
0470 STL     1      ;start adr lo
0480 STH     1      ;start adr hi
0490 BLL     1      ;length lo
0500 BLH     1      ;length hi
0510 BAL     1      ;buffer byte 1
0520 BAH     1      ;buffer byte 2
0530 XSAV    1      ;save loc
0540 LNPOS   1      ;CP line pos
0550 LNSAV   2      ;COPY length
0560 COUNT   1      ;counter
0570 ;
0580 ;OS Equates
0590 ;
```

```

0600 CIOV = $E456 ;CIO Vector
0610 ICCOM = $0342 ;CIO Command
0620 ICBAL = $0344 ;CIO buffer lo
0630 ICBAH = $0345 ;CIO buffer hi
0640 ICBLL = $0348 ;CIO length lo
0650 ICBLH = $0349 ;CIO length hi
0660 AUX1 = $034A ;CIO aux 1
0670 AUX2 = $034B ;CIO aux 2
0680 RUNAD = $02E0 ;run address
0690 EOL = $9B ;end of line
0700 FR0 = $D4 ;FP number
0710 TRAMSZ = $06 ;cart in?
0720 BOOT? = $09 ;boot flag
0730 DOSVEC = $0A ;DOS Vector
0740 DOSINI = $0C ;init vec
0750 WARMST = $08 ;warm st flag
0760 MEMLO = $02E7 ;lo mem pntr
0770 INITAD = $02E2 ;init addr
0780 HATABS = $031A ;Handler Table
0790 SHFLOK = $02BE ;Caps toggle
0800 IFP = $D9AA ;Int to FP
0810 FASC = $D8E6 ;FP to ASC
0820 INBUFF = $F3 ;FP pntr
0830 EDITRV = $E400 ;E: Handler Tab
0840 ;
0850 ;BUMP Macro
0860 ;
0870 .MACRO BUMP
0880 INC %1
0890 BNE @BUMP
0900 INC %1+1
0910 @BUMP
0920 .ENDM
0930 ;
0940 ;PRINT Macro
0950 ;
0960 .MACRO PRINT
0970 LDA # <%1
0980 LDY # >%1
0990 JSR EPRINT
1000 .ENDM
1010 ;
1020 ORIGIN = $1D7C
1030 *= ORIGIN
1040 ;
1050 ;This routine attempts to
1060 ;Binary Load a file.
1070 ;If the file is not Binary an
1080 ;Error #181 is given.
1090 ;
1100 ;ENTER:
1110 ;MYBUF=Filename to load minus
1120 ; a .COM extension.
1130 ; Load is on IOCB #1
1140 ;
1150 LOADIT LDY #0
1160 STY COUNT
1170 STY RUNAD
1180 STY RUNAD+1

```

```

1190     JSR FINDFILE
1200     LDY #0
1210     JSR FINDARG
1220     LDA # <COM
1230     LDY # >COM
1240     JSR ADD_EXT
1250     JSR CLOSE1
1260 ;
1270     JSR SET_OPN
1280     JSR SET_DEV
1290     JSR SET_4
1300     BPL LOAD_MAIN
1310 GO_ERR JMP IOERROR
1320 NO_BIN LDY #181
1330     BNE GO_ERR
1340 ;
1350 ;Main load Loop
1360 ;
1370 LOAD_MAIN JSR READ2
1380     LDX BAL
1390     INX
1400     BNE NO_BIN
1410     LDX BAH
1420     INX
1430     BNE NO_BIN
1440     JSR SET_DEST
1450 ;
1460 GET_FIL LDA # <EPRDN
1470     STA INITAD
1480     LDA # >EPRDN
1490     STA INITAD+1
1500     JSR READ2
1510     LDX BAL
1520     INX
1530     BNE NOT_FF
1540     LDX BAH
1550     INX
1560     BNE NOT_FF
1570     JSR READ2
1580 NOT_FF LDA BAL
1590     STA STL
1600     LDA BAH
1610     STA STH
1620     LDX COUNT
1630     BNE PASS2
1640     INC COUNT
1650     STA ADDRESS+1
1660     LDA STL
1670     STA ADDRESS
1680 PASS2 JSR READ2
1690     LDA BAL
1700     SEC
1710     SBC STL
1720     STA BLL
1730     LDA BAH
1740     SBC STH
1750     STA BLH
1760     BUMP  BLL
1770     JSR GET_DATA

```

```

1780      BMI JSTART
1790      JSR JINIT
1800      JMP GET_FIL
1810 ;
1820 JINIT JMP (INITAD)
1830 JSTART JSR CLOSE1
1840      LDA RUNAD
1850      TAX
1860      ORA RUNAD+1
1870      BEQ LOAD_GO
1880      STX ADDRESS
1890      LDA RUNAD+1
1900      STA ADDRESS+1
1910 LOAD_GO JMP (ADDRESS)
1920 ;
1930 ;Get DATA from file
1940 ;
1950 GET_DATA LDA STL
1960      STA SL
1970      LDA STH
1980      STA SH
1990 ;
2000 GET_REC LDX #$10
2010      LDA #7
2020 CALL_CIO STA ICCOM,X
2030      LDA SL
2040      STA ICBAL,X
2050      LDA SH
2060      STA ICBAH,X
2070      LDA BLL
2080      STA ICBLL,X
2090      LDA BLH
2100      STA ICBLH,X
2110      JMP CIOV
2120 ;
2130 ;Read 2 bytes from file
2140 ;
2150 READ2 LDA # <BAL
2160      STA SL
2170      LDA #0
2180      STA SH
2190      STA BLH
2200      LDA #2
2210      STA BLL
2220      BNE GET_REC
2230 ;
2240 ;Set AUX1 to 4
2250 ;
2260 SET_4 LDA #4
2270      STA AUX1,X
2280      JMP CIOV
2290 ;
2300 ;E: Print Routine
2310 ;
2320 ;ENTER:
2330 ;A=LSB of string
2340 ;Y=MSB of string
2350 ;
2360 ;EXIT:

```

```

2370 ;A=zero
2380 ;
2390 EPRINT STA FR0
2400     STY FR0+1
2410 EPL LDY #0
2420     LDA (FR0),Y
2430     BEQ EPRDN
2440     JSR EPUT
2450     INC FR0
2460     BNE EPL
2470     INC FR0+1
2480     BNE EPL
2490 EPRDN RTS
2500 ;
2510 ;E: Put Byte Routine
2520 ;
2530 ;ENTER:
2540 ;A=character to print
2550 ;
2560 EPUT TAY
2570     LDA EDITRV+7
2580     PHA
2590     LDA EDITRV+6
2600     PHA
2610     TYA
2620     RTS
2630 ;
2640 ;E: Input Routine
2650 ;
2660 EINPUT LDX #$00
2670     LDA #$40
2680     STA SHFLOK
2690 ;
2700 ;Get a 64 byte line in MYBUF
2710 ;
2720 ;ENTER:
2730 ;X=IOCB Index ($10,$20,...)
2740 ;
2750 INP_MYB LDA #0
2760     STA SL
2770     STA BLH
2780     LDA #64
2790     STA BLL
2800     LDA #5
2810     STA SH
2820     JMP CALL_CIO
2830 ;
2840 ;Get a Hex # from MYBUF
2850 ;
2860 ;ENTER:
2870 ;Y=offset into MYBUF where
2880 ; the Hex # starts
2890 ;
2900 ;EXIT:
2910 ;FR0,FR0+1 = binary number
2920 ;
2930 ;
2940 GRAB_HEX LDA #0
2950     STA COUNT

```

```

2960     STA FR0
2970     STA FR0+1
2980 G4LOOP LDA MYBUF,Y
2990     CMP #EOL
3000     BEQ HEX_OUT
3010     CMP #20
3020     BNE TESTIT
3030 HEX_OUT LDA COUNT
3040     BEQ HEX_BAD
3050 HEX_GOOD CLC
3060     RTS
3070 HEX_BAD SEC
3080     RTS
3090 ;
3100 TESTIT LDX #0F
3110 G4SCAN CMP HEXDIG,X
3120     BEQ GOTG4D
3130     DEX
3140     BPL G4SCAN
3150     BMI HEX_BAD
3160 ;
3170 GOTG4D ASL FR0
3180     ROL FR0+1
3190     ASL FR0
3200     ROL FR0+1
3210     ASL FR0
3220     ROL FR0+1
3230     ASL FR0
3240     ROL FR0+1
3250     TXA
3260     ORA FR0
3270     STA FR0
3280     INC COUNT
3290     INY
3300     LDA COUNT
3310     CMP #5
3320     BCS HEX_BAD
3330     BCC G4LOOP
3340 ;
3350 ;Close IOCB #1
3360 ;
3370 CLOSE1 LDX #10
3380     BNE CLOSEIT
3390 ;
3400 ;Close IOCB #2
3410 ;
3420 CLOSE2 LDX #20
3430 ;
3440 ;Close Any IOCB
3450 ;
3460 ;ENTER:
3470 ;X=IOCB Index ($10,$20,...)
3480 ;
3490 ;EXIT:
3500 ;X=IOCB Index
3510 ;
3520 CLOSEIT TXA
3530     PHA
3540     LDA #0C

```

```

3550     STA ICCOM,X
3560     JSR CIOV
3570     PLA
3580     TAX
3590     RTS
3600 ;
3610 ;CP's PROMPT + INPUT Routine
3620 ;
3630 INPUT PRINT PROMPT
3640     JSR EINPUT
3650     BPL INPUTLV
3660     JSR IOERROR
3670     JMP INPUT
3680 INPUTLV LDA MYBUF
3690     CMP #EOL
3700     BEQ INPUT
3710     RTS
3720 ;
3730 ;#####
3740 ;
3750 ;Start of Commands
3760 ;
3770 ;#####
3780 ;
3790 ;Enter Cart
3800 ;
3810 GOCART LDA TRAMSZ
3820     BNE TRY_CART
3830     LDA # <NOCART
3840     LDY # >NOCART
3850     JMP EPRINT
3860 TRY_CART JMP ($BFFA)
3870 ;
3880 ;Kill CP
3890 ;
3900 KILL JSR UNHOOK
3910     STA WARMST
3920     JSR GOCART
3930     JMP (DOSVEC)
3940 ;
3950 ;Un-Hook OS Patches
3960 ;
3970 UNHOOK LDA #$FF
3980     STA DOSVEC
3990 CP_HI LDA #$FF
4000     STA DOSVEC+1
4010     LDA RESET+1
4020     STA DOSINI
4030     LDA RESET+2
4040     STA DOSINI+1
4050     LDY INDEX
4060     LDA # <EDITRV
4070     STA HATABS,Y
4080     LDA # >EDITRV
4090     STA HATABS+1,Y
4100 OLD_ML LDA #$FF
4110     STA MEMLO
4120 OLD_MH LDA #$FF
4130     STA MEMLO+1

```

```
4140     LDA #0
4150     STA SETFLAG
4160     RTS
4170 ;
4180 ;Erase a File
4190 ;
4200 ERASE LDA #33
4210     BNE DOXIO
4220 ;
4230 ;Lock a File
4240 ;
4250 PROTECT LDA #35
4260     BNE DOXIO
4270 ;
4280 ;Unlock a File
4290 ;
4300 UNPROTECT LDA #36
4310     BNE DOXIO
4320 ;
4330 ;Rename a File
4340 ;
4350 RENAME JSR FINDFILE
4360     LDY LNPOS
4370     JSR FINDARG
4380     LDX XSAV
4390     LDA #' ,
4400     STA FNAME,X
4410 REN_LP LDA MYBUF,Y
4420     STA FNAME+1,X
4430     INX
4440     INY
4450     CMP #EOL
4460     BNE REN_LP
4470     LDA #32
4480     PHA
4490     BNE XIO_ENT
4500 ;
4510 ;Perform an XIO
4520 ;
4530 ;ENTER:
4540 ;A=XIO Number
4550 ;Y=Offset to Filename in MYBUF
4560 ;
4570 DOXIO PHA
4580     JSR FINDFILE
4590 XIO_ENT JSR CLOSE1
4600     PLA
4610     STA ICCOM,X
4620     JSR SET_DEV
4630     STA AUX1,X
4640     JSR CIOV
4650     BMI IOERROR
4660     RTS
4670 ;
4680 ;Set ICBAL/H to DEVICE
4690 ;
4700 ;ENTER:
4710 ;X=IOCB Index
4720 ;
```



```

4730 SET_DEV LDA # <DEVICE
4740     STA ICBAL,X
4750     LDA # >DEVICE
4760     STA ICBAH,X
4770     LDA #0
4780     STA AUX2,X
4790     RTS
4800 ;
4810 ;Handle I/O error
4820 ;
4830 ;ENTER:
4840 ;Y=I/O Error #
4850 ;
4860 ;EXIT:
4870 ;Channel #1 is closed
4880 ;
4890 IOERROR STY FR0
4900 IO_ERR LDA #0
4910     STA FR0+1
4920     JSR CLOSE1
4930     JSR IFP
4940     JSR FASC
4950     LDY #$FF
4960 IOERR INY
4970     LDA (INBUFF),Y
4980     STA ERBUF,Y
4990     BPL IOERR
5000     AND #$7F
5010     STA ERBUF,Y
5020     LDA #EOL
5030     STA ERBUF+1,Y
5040     LDA #0
5050     STA ERBUF+2,Y
5060     LDA # <ERTXT
5070     LDY # >ERTXT
5080     JMP EPRINT
5090 ;
5100 ;This will grab a filename from
5110 ;MYBUF and put it in the
5120 ;device buffer.
5130 ;
5140 ;ENTER:
5150 ;Y=Offset to Filename in MYBUF
5160 ;
5170 FINDFILE LDA DEFDEV
5180     STA DEVICE
5190     LDA DEFDEV+1
5200     STA DEVICE+1
5210 ;
5220 FIND1 LDX #0
5230     LDA MYBUF+1,Y
5240     CMP #' ':
5250     BEQ X2
5260     LDA MYBUF+2,Y
5270     CMP #' ':
5280     BNE FIND2
5290     LDA MYBUF+1,Y
5300     STA DEVICE+1
5310     LDA MYBUF,Y

```

```

5320     STA DEVICE
5330     INY
5340     INY
5350     INY
5360     BNE FIND2
5370 ;
5380 X2  LDA MYBUF,Y
5390     STA DEVICE
5400     INY
5410     INY
5420 ;
5430 FIND2 LDA MYBUF,Y
5440     STA FNAME,X
5450     CMP #EOL
5460     BEQ FIND3
5470     CMP #\$20
5480     BEQ FIND3
5490     INY
5500     INX
5510     CPX #13
5520     BNE FIND2
5530 FIND3 LDA #EOL
5540     STA FNAME,X
5550     LDA #0
5560     STA FNAME+1,X
5570     STX XSAV
5580     RTS
5590 ;
5600 ;This handles the 'RUN' command
5610 ;
5620 MEMRUN LDA MYBUF,Y
5630     CMP #EOL
5640     BEQ RUN_IT
5650 ;
5660 PULL_IT JSR GRAB_HEX
5670     BCS BAD_RUN
5680     LDA FR0
5690     STA ADDRESS
5700     LDA FR0+1
5710     STA ADDRESS+1
5720 RUN_IT JMP (ADDRESS)
5730 BAD_RUN LDY #180
5740 MEM_ERR JMP IOERROR
5750 ;
5760 ;Get disk Directory
5770 ;
5780 DIRECT LDA #6
5790     BNE DIR_GO
5800 ;
5810 ;Get a TYPE of a File
5820 ;
5830 TYPE LDA #4
5840 DIR_GO STA DIR_COM+1
5850     JSR FINDFILE
5860     LDA DEVICE+3
5870     CMP #EOL
5880     BNE OPEN_DIR
5890     LDX #3
5900 CP_DIR LDA DIRNAM,X

```

```

5910     STA DEVICE+3,X
5920     DEX
5930     BPL CP_DIR
5940 ;
5950 OPEN_DIR JSR CLOSE1
5960     JSR SET_OPN
5970     JSR SET_DEV
5980 DIR_COM LDA #6
5990     STA AUX1,X
6000     JSR CIOV
6010     BMI MEM_ERR
6020 ;
6030 DIROK LDX #$10
6040     JSR INP_MYB
6050     BMI DIRDONE
6060     LDY #0
6070 DIRCK1 LDA MYBUF,Y
6080     CMP #EOL
6090     BEQ DIRCK2
6100     INY
6110     BNE DIRCK1
6120 DIRCK2 LDA #0
6130     STA MYBUF+1,Y
6140     PRINT MYBUF
6150     JMP DIROK
6160 DIRDONE JMP CLOSE1
6170 ;
6180 ;Set Destroy Flag
6190 ;
6200 SET_DEST LDA #0
6210     STA WARMST
6220     RTS
6230 ;
6240 ;Add File Extension
6250 ;
6260 ;ENTER:
6270 ;A=LSB of extension
6280 ;Y=MSB of extension
6290 ;
6300 ADD_EXT STA FR0
6310     STY FR0+1
6320     LDX #$FF
6330     LDY #4
6340 ;
6350 EXT_SCAN INX
6360     LDA FNAME,X
6370     CMP #'.'
6380     BEQ EXT_RTS
6390     CMP #EOL
6400     BNE EXT_SCAN
6410 EXT_ADD LDA (FR0),Y
6420     STA FNAME,X
6430     INX
6440     DEY
6450     BPL EXT_ADD
6460 EXT_RTS RTS
6470 ;
6480 ;This is where RESET comes
6490 ;

```

```

6500 RESET JSR $FFFF
6510     JSR UNHOOK
6520 ;
6530 HOOKUP LDX #$00
6540     JSR CLOSEIT
6550     JSR SET_OPN
6560     LDA # <EDEV
6570     STA ICBAL
6580     LDA # >EDEV
6590     STA ICBAH
6600     STX AUX2
6610     LDA #12
6620     STA AUX1
6630     JSR CIOV
6640 ;
6650 HOOK1 CLD
6660     LDY EDITRV+4
6670     LDX EDITRV+5
6680     INY
6690     BNE NO_UPX
6700     INX
6710 NO_UPX STY EGET+1
6720     STX EGET+2
6730     LDY #0
6740 ;
6750 FIND_E LDA HATABS,Y
6760     CMP #'E
6770     BEQ GOT_E
6780     INY
6790     INY
6800     INY
6810     CPY #33
6820     BCC FIND_E
6830 ;
6840 GOT_E INY
6850     STY INDEX
6860     LDA # <NEW_E.HAN
6870     STA HATABS,Y
6880     LDA # >NEW_E.HAN
6890     STA HATABS+1,Y
6900 ;
6910     LDX #$50
6920     JSR CLOSEIT
6930     LDY #$0F
6940 COPY_E LDA EDITRV,Y
6950     STA NEW_E.HAN,Y
6960     LDA IOCB5,Y
6970     STA BAT_IOCB,Y
6980     DEY
6990     BPL COPY_E
7000 ;
7010     LDA # <[BAT_GET-1]
7020     STA NEW_E.HAN+4
7030     LDA # >[BAT_GET-1]
7040     STA NEW_E.HAN+5
7050 ;
7060 SKIP_BAT LDA SETFLAG
7070     BNE INIT_RTS
7080     INC SETFLAG

```

```

7090     LDA DOSINI
7100     STA RESET+1
7110     LDA DOSINI+1
7120     STA RESET+2
7130     LDA # <RESET
7140     STA DOSINI
7150     LDA # >RESET
7160     STA DOSINI+1
7170 ;
7180     LDA DOSVEC
7190     STA UNHOOK+1
7200     LDA DOSVEC+1
7210     STA CP_HI+1
7220     LDA # <GO_CP
7230     STA DOSVEC
7240     LDA # >GO_CP
7250     STA DOSVEC+1
7260 ;
7270     LDA MEMLO
7280     STA OLD_ML+1
7290     LDA MEMLO+1
7300     STA OLD_MH+1
7310     LDA # <PROGEND
7320     STA MEMLO
7330     LDA # >PROGEND
7340     STA MEMLO+1
7350 INIT_RTS RTS
7360 ;
7370 ;COMMAND PROCESSOR ENTRY
7380 ;
7390 GO_CP JSR HOOK1
7400     LDA #$FF
7410     STA WARMST
7420 ;
7430 MAIN CLD
7440     JSR INPUT ;get line
7450     LDA MYBUF
7460     CMP #'*      ;Batch?
7470     BNE NO_BAT
7480     JMP BATCH
7490 ;
7500 NO_BAT CMP #' ;
7510     BEQ MAIN
7520     LDY #0
7530     JSR FINDFILE
7540     LDA DEVICE+3
7550     CMP #EOL
7560     BNE NO_REM
7570     LDA DEVICE
7580     STA DEFDEV
7590     LDA DEVICE+1
7600     STA DEFDEV+1
7610     JMP MAIN
7620 ;
7630 NO_REM LDA # <COMTAB
7640     STA FR0
7650     LDA # >COMTAB
7660     STA FR0+1
7670     LDX #0

```

```

7680 COMLP LDY #0
7690 COMCK LDA MYBUF,Y
7700     CMP (FR0),Y
7710     BNE TRYNEXT
7720     INY
7730     CPY #3
7740     BNE COMCK
7750 ;
7760     LDA COMADRL,X
7770     STA COMJSR+1
7780     LDA COMADRH,X
7790     STA COMJSR+2
7800     JSR FINDARG
7810 COMJSR JSR $FFFF
7820     JMP MAIN
7830 ;
7840 TRYNEXT LDA FR0
7850     CLC
7860     ADC #3
7870     STA FR0
7880     BCC TRY_NH
7890     INC FR0+1
7900 TRY_NH INX
7910     CPX #NUMCOM
7920     BNE COMLP
7930     JSR LOADIT
7940     JMP MAIN
7950 ;
7960 ;Find an Argument on Line
7970 ;
7980 ;ENTER:
7990 ;Y=Offset to start search
8000 ;
8010 ;EXIT:
8020 ;Y=1st char of argument
8030 ;
8040 FINDARG LDA MYBUF,Y
8050     CMP #EOL
8060     BEQ NEXTARG
8070     CMP #EOL
8080     BEQ NEXTARG
8090     INY
8100     BNE FINDARG
8110 NEXTARG LDA MYBUF,Y
8120     CMP #EOL
8130     BNE FOUNDARG
8140     INY
8150     BNE NEXTARG
8160 FOUNDARG STY LNPOS
8170     RTS
8180 ;
8190 ;Handle a BATCH File
8200 ;
8210 BATCH LDY #0
8220     LDA #EOL
8230     STA MYBUF
8240     JSR FINDARG
8250     JSR FINDFILE
8260     LDA # <BAT

```

```

8270      LDY # >BAT
8280      JSR ADD_EXT
8290 BAT_GO LDA #12
8300      JSR BAT_CIO
8310 ;
8320      LDA # <DEVICE
8330      STA ICBALB
8340      LDA # >DEVICE
8350      STA ICBAHB
8360      LDA #4
8370      STA AUX1B
8380      LDA #3
8390      JSR BAT_CIO
8400      BPL BAT_MAIN
8410      LDA #12
8420      STY FR0
8430      JSR BAT_CIO
8440      JSR IO_ERR
8450 BAT_MAIN JMP MAIN
8460 BAT_9B LDA #EOL
8470      BNE BAT_XIT
8480 ;
8490 BAT_GET LDA BAT_IOCB
8500      BPL BAT_PROC
8510 BAT_NORM LDX #$00
8520      LDY #1
8530 EGET JMP $FFFF
8540 ;
8550 BAT_PROC LDA #0
8560      STA ICBLLB
8570      STA ICBLHB
8580      LDA #7
8590      JSR BAT_CIO
8600      BMI BAT_NORM
8610      LDA BAT_ZIO+15
8620      PHA
8630      JSR EPUT
8640      PLA
8650 BAT_XIT LDX #$00
8660      LDY #1
8670      RTS
8680 ;
8690 ;Perform BATCH CIO
8700 ;
8710 BAT_CIO STA ICCOMB
8720      JSR BAT_SWAP
8730      LDX #$50
8740      JSR CIOV
8750      STA BAT_ZIO+15
8760      BPL BAT_SWAP
8770      TYA
8780      PHA
8790      LDX #$50
8800      JSR CLOSEIT
8810      PLA
8820      TAY
8830 ;
8840 ;Swap IOCB Blocks
8850 ;

```

```

8860 BAT_SWAP TYA
8870     PHA
8880     LDY #\$0F
8890 BAT_SLP LDA ZIOCB,Y
8900     PHA
8910     LDA BAT_ZIO,Y
8920     STA ZIOCB,Y
8930     PLA
8940     STA BAT_ZIO,Y
8950     LDA IOCB5,Y
8960     PHA
8970     LDA BAT_IOCB,Y
8980     STA IOCB5,Y
8990     PLA
9000     STA BAT_IOCB,Y
9010     DEY
9020     BPL BAT_SLP
9030     PLA
9040     TAY
9050     RTS
9060 ;
9070 ;Set COPY IOCB
9080 ;
9090 ;ENTER:
9100 ;X=IOCB Index
9110 ;A=CIO Command
9120 ;
9130 ;EXIT:
9140 ;CIO set for I/O of 12K block
9150 ;at \$3000
9160 ;
9170 SET_CPY STA ICCOM,X
9180     LDA #0
9190     STA ICBLI,X
9200     STA ICBAL,X
9210     LDA #\$30
9220     STA ICBLH,X
9230     STA ICBAH,X
9240     RTS
9250 ;
9260 ;Set ICCOM for Open
9270 ;
9280 ;ENTER:
9290 ;X=IOCB Index
9300 ;
9310 SET_OPN LDA #3
9320     STA ICCOM,X
9330     RTS
9340 ;
9350 ;Copy I/O Error
9360 ;
9370 ;ENTER:
9380 ;Y=CIO Error #
9390 ;
9400 ;EXIT:
9410 ;Channels #1 and #2 are closed
9420 ;
9430 CPY_IOR JSR IOERROR
9440     JMP CLOSE2

```



```

9450 ;
9460 ;Handle COPY Verb
9470 ;
9480 COPY JSR FINDFILE
9490     JSR CLOSE1
9500     JSR SET_DEV
9510     JSR SET_OPN
9520     JSR SET_4
9530     BMI CPY_IOR
9540         PRINT DEVICE
9550     LDY LNPOS
9560     JSR FINDARG
9570     JSR FINDFILE
9580     JSR CLOSE2
9590     JSR SET_DEV
9600     JSR SET_OPN
9610     LDA #8
9620     STA AUX1,X
9630     JSR CIOV
9640     BMI CPY_IOR
9650 ;
9660     JSR SET_DEST
9670 COPY.2 LDX #$10
9680     LDA #7
9690     JSR SET_CPY
9700     JSR CIOV
9710     BMI EOF
9720     LDA #1
9730     STA COUNT
9740     BNE SAVLEN
9750 ;
9760 EOF LDA #0
9770     STA COUNT
9780 ;
9790 SAVLEN LDA ICBLH,X
9800     STA LENSAB
9810     LDA ICBLH,X
9820     STA LENSAB+1
9830     LDX #$20
9840     LDA #11
9850     JSR SET_CPY
9860     LDA LENSAB
9870     STA ICBLH,X
9880     LDA LENSAB+1
9890     STA ICBLH,X
9900     JSR CIOV
9910     LDA COUNT
9920     BNE COPY.2
9930     JSR CLOSE1
9940     JMP CLOSE2
9950 ;
9960 ;Program Text/Buffers
9970 ;
9980 NOCART .BYTE "No Cart",EOL,0
9990 PROMPT .BYTE EOL
010000 DEFDEV .BYTE "D1:",0
010010 DIRNAM .BYTE ".*",EOL
010020 EDEV .BYTE "E:",EOL
010030 ERTXT .BYTE "Error- "

```

```
010040 ERBUF .BYTE " ",EOL,0
010050 HEXDIG .BYTE "0123456789"
010060 .BYTE "ABCDEF"
010070 COM .BYTE EOL,"MOC."
010080 BAT .BYTE EOL,"TAB."
010090 ;
010100 ;Command Tables
010110 ;
010120 COMTAB .BYTE "ERAPROUNPREN"
010130 .BYTE "DIRCOPRUN"
010140 .BYTE "CARTYPKIL"
010150 ;
010160 COMADRL .BYTE <ERASE
010170 .BYTE <PROTECT
010180 .BYTE <UNPROTECT
010190 .BYTE <RENAME
010200 .BYTE <DIRECT
010210 .BYTE <COPY
010220 .BYTE <MEMRUN
010230 .BYTE <GOCART
010240 .BYTE <TYPE
010250 .BYTE <KILL
010260 ;
010270 COMADRH .BYTE >ERASE
010280 .BYTE >PROTECT
010290 .BYTE >UNPROTECT
010300 .BYTE >RENAME
010310 .BYTE >DIRECT
010320 .BYTE >COPY
010330 .BYTE >MEMRUN
010340 .BYTE >GOCART
010350 .BYTE >TYPE
010360 .BYTE >KILL
010370 ;
010380 ADDRESS .WORD EPRDN ;RUN Addr
010390 DEVICE .BYTE "D1:"
010400 ;
010410 ;Buffers
010420 ;
010430 FNAME .BYTE "AUTORUN.BAT"
010440 .BYTE EOL,0
010450 17
010460 INDEX 1 ;HATABS Index
010470 SETFLAG 1 ;set up?
010480 ;
010490 ;Wedge/Handler Space
010500 ;
010510 ;For Batch Processing
010520 ;
010530 BAT_ZIO 16 ;Zpage IOCB
010540 NEW_E.HAN 16 ;E: Handler
010550 BAT_IOCB 16 ;IOCB #5 Copy
010560 ICCOMB = BAT_IOCB+2
010570 ICBALB = BAT_IOCB+4
010580 ICBAHB = BAT_IOCB+5
010590 ICBLLB = BAT_IOCB+8
010600 ICBLHB = BAT_IOCB+9
010610 AUX1B = BAT_IOCB+10
010620 ;
```

```
010630 PROGEND = *
010640 T_LENGTH = PROGEND-ORIGIN
010650 ;
010660 ;Entry Point to Start CP
010670 ;
010680 ENTRY LDX #0
010690 STX SETFLAG
010700 STX WARMST
010710 INX
010720 STX BOOT?
010730 JSR HOOKUP
010740 PRINT CREDITS
010750 JSR CLOSE2
010760 JSR SET_OPN
010770 JSR SET_DEV
010780 JSR SET_4
010790 PHP
010800 JSR CLOSE2
010810 PLP
010820 BPL HAV_BAT
010830 LDA TRAMSZ
010840 BNE G_CART
010850 JMP (DOSVEC)
010860 G_CART JMP ($BFFA)
010870 HAV_BAT PRINT PROMPT
010880 PRINT FNAME
010890 PLA
010900 PLA
010910 JMP BAT_GO
010920 ;
010930 CREDITS .BYTE $7D,EOL
010940 .BYTE "BBK CP - (C) 1987 "
010950 .BYTE "ANALOG Computing",EOL
010960 .BYTE "by: Bryan Schappel"
010970 .BYTE EOL,0
010980 ;
010990 ;Set Run Address
011000 ;
011010 *= RUNAD
011020 .WORD ENTRY
011030 .OPT LIST
011040 .END
```