

```

**      2650 DOS Version 2.6a
**      Delays for 2MHz CPU
**      I. Binnie 18-11-81

:      DFLT      1

:START    EQU      6800
:RAM      EQU      7000

:TTD      EQU      3      40mS Track-Track Delay
:DUAL     EQU      1      work drive
:RETRYC   EQU      3      no of retries

:FD1771   EQU      0C      1771 port
:FDCMD    EQU      FD1771  command/status
:FDTRK    EQU      FD1771+1 track register
:FDSEC    EQU      FD1771+2 sector register
:FDDAT    EQU      FD1771+3 data register
:FDINT    EQU      10      Control port

:BUSY     EQU      1      Status bits
:CRC      EQU      8      Cyclic redundancy check
:LSD      EQU      4      Lost data
:RNF      EQU      10      Record not found
:HDENG    EQU      20      Head engaged
:WPRT     EQU      40      Write protect
:NREADY   EQU      80      Drive not ready

:COUNT   EQU      2B4      Monitor
:KBD      EQU      285
:MCRLF    EQU      008A      routines
:ULPT     EQU      70F0      Printer COUNT

:LRAM     EQU      <RAM

**      ERROR CODES

**      R      Read error
**      W      Write error
**      F      Full disc
**      S      Seek error
**      N      Not ready
**      P      Protected disc
**      O      Open error
**      D      Device does not exist
**      H      Hex error
```

		ORG	START	
6800	1F683D	:	BCTA, UN	DOS
6803	6893	:	ACON	ENTER
				Normal entry point
6805	6917	:	ACON	OPNR
				Open seq file, for read
6807	6B7F	:	ACON	GETCH
				Get next char
6809	6B96	:	ACON	CHAIN
				Chain to a file
680B	69CF	:	ACON	OPNW
				Open for writing
680D	6AB1	:	ACON	PUTCH
				Put next char
680F	6B7A	:	ACON	CLSR
				Close read stream
6811	6AF1	:	ACON	CLSW
				Put char; close file
6813	6C15	:	ACON	FIND
				Find file
6815	6C0C	:	ACON	FNEXT
				Find next file this spec
6817	6EE2	:	ACON	ID
				General I/O routine
6819	6D65	:	ACON	DRIVE
				Select drive R0
681B	6E67	:	ACON	ITEM
				Move buf ptr to next item
681D	6E29	:	ACON	DOUT
				Print R2, R3 as decimal
681F	6E92	:	ACON	LINE
				Accept a line of chars
6821	6EDC	:	ACON	CRLF
				Put 0D, 0A on any stream
6823	6C92	:	ACON	STDR
				Move to start of dir
6825	6CA2	:	ACON	STD1
				Move to Tk R2, Sc R3
6827	6E12	:	ACON	PNAM
				Print file name just found
6829	6CBB	:	ACON	RDSC
				Read a sector
682B	6CFB	:	ACON	WRSC
				Write a sector
682D	6F10	:	ACON	HXOUT
				Hexout (BOUT from R0)
682F	6F29	:	ACON	GNMB
				Get Number to XGOTO
6831	6F0D	:	ACON	PADR
				Print Address R0, R2
6833	6E5F	:	ACON	DEC
				Is R0 Decimal?
6835	6D86	:	ACON	NFD
				Not found routine
6837	6CEF	:	ACON	WBSDD
				Delay, wait* for busy low
6839	694C	:	ACON	ASGN
				Assign dev to a stream
683B	6CAE	:	ACON	RDXSC
				Read next sector

/* Cold entry point - initialises RAM

```

683D 04D0 :DOS LODI, R0 0D0 Force interrupt
683F D40C : WRT, R0 FD1771
6841 20 : EORZ, R0
6842 07E7 : LODI, R3 CDRV-RAM+1
6844 CF5000 : STRA, R0 RAM, R3- Clear scratch RAM
6847 5B7B : BRNR, R3 #-3
6849 0C0FFB : LODA, R0 DFDRV
684C D410 : WRT, R0 FDINT Force system drive
684E 040B : LODI, R0 B+TTD Restore
6850 D40C : WRT, R0 FD1771

6852 0702 : LODI, R3 2
6854 0F4FFB :DR LODA, R0 DFDRV, R3- Set default drives
6857 CF70D7 : STRA, R0 SYSD, R3
685A 0F6FE7 : LODA, R0 STRTUP, R3 point to coldstart file
685D CF7070 : STRA, R0 BUFF, R3
6860 5B72 : BRNR, R3 DR

6862 0707 : LODI, R3 7
6864 0F4F8C :VB LODA, R0 VDUB, R3- Setup VDU stream 0
6867 CF7000 : STRA, R0 IDNT, R3
686A 5B7B : BRNR, R3 VB
686C 0471 : LODI, R0 LRAM+1
686E CF700F :SBF STRA, R0 HBUF, R3 Setup Buf ptrs
6871 8401 : ADDI, R0 1
6873 8710 : ADDI, R3 10
6875 E770 : COMI, R3 70
6877 9875 : BCFR, EQ SBF
6879 0700 : LODI, R3 0
687B 3B9C :PENTER BSTR, UN *IPRN+1 "DOS"

687D 0510 : LODI, R1 10 Keyboard stream
687F 3F6C15 : BSTA, UN FIND Coldstart file
6882 1A05 : BCTR, N ENTERK not found
6884 3F6917 : BSTA, UN OPNR
6887 1B0A : BCTR, UN ENTER Execute startup

6889 0707 :ENTERK LODI, R3 7
688B 0F4F93 :KB LODA, R0 KBDB, R3- Setup KBD stream 1
688E CF7010 : STRA, R0 IDNT+10, R3
6891 5B7B : BRNR, R3 KB

```

/* Normal entry point - preserve all

```

6893 0402 :ENTER LODI, R0 LCOM
6895 93 : LPSL Set LCOM, clear others
6896 0715 : LODI, R3 PMT-STR
6898 3F6F01 :IPRN BSTA, UN PRCR Prompt
689B 3F6E92 : BSTA, UN LINE Get command line
689E 0C1072 : LODA, R0 BUF First char
68A1 3F6E5F : BSTA, UN DEC Decimal?
68A4 1A6D : BCTR, LT ENTER (& 0, ignore line
68A6 1936 : BCTR, GT FCMD No, look for command file
68A8 3F6F29 : BSTA, UN GNMB Yes, get address,
68AB 1FF0DD : BCTA, UN *XGOTO branch to it

```

68AE	CC10E0	:ERR	STRA, R0	ERR0	Plant error info
68B1	CD10E1	:	STRA, R1	ERR1	
68B4	CE10E2	:	STRA, R2	ERR2	
68B7	CF10E3	:	STRA, R3	ERR3	
68BA	C3	:	STRZ, R3		
68BB	0D7006	:	LODA, R0	DRV, R1	Save drive #
68BE	CC10DF	:	STRA, R0	ERRDR	
68C1	040B	:	LODI, R0	0+TTD	Restore
68C3	D40C	:	WRTE, R0	FD1771	
68C5	3F008A	:	BSTA, UN	MCRLF	On VDU,
68C8	043F	:	LODI, R0	A'?'	
68CA	3F02B4	:IMC	BSTA, UN	CDUT	print error letter
68CD	03	:	LODZ, R3		
68CE	3BFB	:	BSTR, UN	*IMC+1	
68D0	1F6889	:	BCTA, UN	ENTERK	to command level
68D3	0C10D8	:NFF	LODA, R0	WRKD	
68D6	ED7006	:	COMA, R0	DRV, R1	Work drv = current drv?
68D9	9808	:	BCFR, EQ	FCMD1	No, try it
68DB	3F6D86	:	BSTA, UN	NFD	Yes, "Not found"
		*: Fail thru if esc typed			
68DE	0520	:FCMD	LODI, R1	20	Use str 2 to search dir
68E0	0C10D7	:	LODA, R0	SYSD	Try SYS drv first
68E3	CD7006	:FCMD1	STRA, R0	DRV, R1	
68E6	3F6D65	:	BSTA, UN	DRIVE	Select drive
68E9	04FF	:	LODI, R0	-1	
68EB	CD700E	:	STRA, R0	DBPT, R1	pointer to file spec.
68EE	3F6C92	:	BSTA, UN	STDR	St of dir
68F1	040F	:	LODI, R0	0F	
68F3	CD7005	:	STRA, R0	PBPT, R1	point to 1st dir entry
68F6	3F6C26	:FDIR	BSTA, UN	FNX	Get dir entfy
68F9	1A58	:	BCTR, N	NFF	None
68FB	0703	:CCMD.	LODI, R3	3	Check if .CMD
68FD	0F4FF5	:	LODA, R0	CMDS, R3-	
6900	EF70C0	:	COMA, R0	DENT+0A, R3	
6903	9871	:	BCFR, EQ	FDIR	No, try next name
6905	5B76	:	BRNR, R3	CCMD+2	
6907	0730	:EXEC	LODI, R3	EX-STR	"Executing"
6909	3F6F01	:	BSTA, UN	PRCR	
690C	3F6E12	:	BSTA, UN	PNAM	Print name
690F	3F6EDA	:	BSTA, UN	CRLF0	
6912	0520	:	LODI, R1	20	
6914	1F6B9B	:	BCTA, UN	CHAINI	Into CHAIN, to execute it


```

** Open a file for read
** Name is at *BUFF; use stream in R1
** Saves R1-R3; 5 subr levels

```

```

6917 CE10D3 :OPNR STRA,R2 SR2 Save R2,R3
691A CF10D4 : STRA,R3 SR3
691D 3F6C15 :SQF BSTA,UN FIND Look it up
6920 9A05 : BCFR,LT S1
6922 3F6D86 : BSTA,UN NFD "Not found"
6925 1B76 : BCTR,UN SQF Try again
6927 1C6945 :S1 BCTA,EQ RT. Return if non-disk
692A 046B : LODI,R0 <GETCH
692C CD7003 : STRA,R0 CODE,R1 Srvce rtne is GETCH
692F 047F : LODI,R0 GETCH
6931 CD7004 : STRA,R0 CODE+1,R1
** Store st of dir ent at DBYT,DSEC
6934 0D7005 : LODA,R0 PBPT,R1
6937 A417 : SUBI,R0 17 Back up one entry
6939 CD700A : STRA,R0 DBYT,R1
693C 0D7008 : LODA,R0 SEC,R1
693F CD7009 : STRA,R0 DSEC,R1
6942 3F6C98 : BSTA,UN FIRST Get first sector to buffer
6945 0E10D3 :RT LODA,R2 SR2 Restore reas
6948 0F10D4 : LODA,R3 SR3
694B 17 : RETC,UN

```

```

** Assign filespec (e.g. LPT: or NAME.EXT) to a stream
** Spec at *BUFF; stream # in R1
** Only R1 saved; 2 subr levels

```

```

694C 0703 :ASGN LODI,R3 3
694E 0FF070 : LODA,R0 *BUFF,R3 Check for dev
6951 E43A : COMI,R0 A' ':'
6953 1827 : BCTR,EQ DPR Yes
6955 06FF : LODI,R2 -1
6957 0E2FF2 :AS1 LODA,R0 DSKD,R2+ No; default to DSK:
695A CD2FFF : STRA,R0 IDNT-1,R1+
695D FB78 : BDRR,R3 AS1
695F 4570 : ANDI,R1 70 Restore R1
6961 0FF070 :AS2 LODA,R0 *BUFF,R3
6964 3F6E5F : BSTA,UN DEC Drv # pres?
6967 1805 : BCTR,EQ DVN Yes
6969 0C10D8 : LODA,R0 WRKD No, use work drive
696C FB04 : BDRR,R3 AS3
696E A431 :DVN SUBI,R0 A' 1'
6970 0701 : LODI,R3 1 Skip digit,dot
6972 CD7006 :AS3 STRA,R0 DRV,R1
6975 03 : LODZ,R3
6976 CD700E : STRA,R0 DBPT,R1 pointer to file spec.
6979 0401 : LODI,R0 1 CC=GT => was disk
697B 17 : RETC,UN
697C 0704 :DPR LODI,R3 4
697E 0603 : LODI,R2 3
6980 8503 : ADDI,R1 3
6982 0ED070 :DPRL LODA,R0 *BUFF,R2- Copy device name

```

```

6985 CD5000 : STRA, R0 IDNT, R1- to IDeNT of stream
6988 5A78 : BRNR, R2 DPRL
698A 0E03 : LODI, R2 3
698C 0ED070 : AS4 LODA, R0 *BUFF, R2-
698F EE6FF2 : COMA, R0 DSKD, R2 Was it DSK: ?
6992 9804 : BCFR, EQ COD No, look it up
6994 5A76 : BRNR, R2 AS4
6996 1B49 : BCTR, UN AS2 As before

6998 0E00 : COD LODI, R2 0 R2 steps thru device table
699A 07FF : LODI, R3 -1 R3 " " dev name
699C 0E6F7F : CODL LODA, R0 DEVS, R2 Lookup table
699F 1A11 : BCTR, N- ERRD End of table if nes
69A1 EFB070 : COMA, R0 *BUFF, R3+
69A4 9806 : BCFR, EQ NEG Not =, try next dev
69A6 F601 : TMI, R2 1
69A8 1B13 : BCTR, EQ CODI Done if 2 chars match
69AA DA70 : BIRR, R2 CODL
69AC 8E04 : NEG ADDI, R2 4
69AE 4E7C : ANDI, R2 7C Put to 4byte bdry
69B0 1B68 : BCTR, UN COD+2

69B2 0444 : ERRD LODI, R0 A'D' D error - Device does not exist
69B4 0E1070 : LODA, R2 BUFF
69B7 0F1071 : LODA, R3 BUFF+1 Provide pointer to it
69BA 1F6BAE : BCTA, UN ERR

69BD 0E2F7F : CODI LODA, R0 DEVS, R2+ Get srvc routine addr
69C0 CD7003 : STRA, R0 CODE, R1 Store in stream info
69C3 0E2F7F : LODA, R0 DEVS, R2+
69C6 CD7004 : STRA, R0 CODE+1, R1
69C9 04FF : LODI, R0 -1
69CB CD7006 : STRA, R0 DRV, R1 Flag as non disk
69CE 17 : RETC, UN CC=LT => Was not disk

```

```

:* Open a file for writes
:* Can't open >1 file for writes (yet)
:* Name at *BUFF; str # in R1
:* Saves R1-R3; 5 subr levels

```

```

69CF CE10D3 : OPNW STRA, R2 SR2 Save regss
69D2 CF10D4 : STRA, R3 SR3
69D5 3F6C15 : SQ2 BSTA, UN FIND Check if already exists this dis
69D8 1C6945 : BCTA, EQ RT No action if non-disk
69DB 1A18 : BCTR, LT OP2
69DD 073B : LODI, R3 ARE-STR
69DF CD10D5 : STRA, R1 SVR1 Save str #
69E2 3F6F01 : BSTA, UN PRCR "Already exists"
69E5 3F0286 : BSTA, UN KBD
69E8 E41B : COMI, R0 1B Escape?
69EA 9C6893 : BCFA, EQ ENTER No
69ED 3F6EDC : BSTA, UN CRLF Yes, move cursor
69F0 0D10D5 : LODA, R1 SVR1
69F3 1B60 : BCTR, UN SQ2

69F5 046A : OP2 LODI, R0 <PUTCH+3 Service routine is PUTCH+3
69F7 CD7003 : STRA, R0 CODE, R1

```


69FA	04B4	:	LODI, R0	PUTCH+3	
69FC	CD7004	:	STRA, R0	CODE+1, R1	
69FF	3F6C92	:	BSTA, UN	STDR	St of dir
6A02	3F6B7F	:IGT1	BSTA, UN	GETCH	
6A05	CC10D9	:	STRA, R0	ADR	Save start of freelist
6A08	3BF9	:	BSTR, UN	*IGT1+1	(GETCH)
6A0A	CC10DA	:	STRA, R0	ADR+1	
6A0D	07F8	:	LODI, R3	10-18	"Zeroth" entry
6A0F	3F6BDF	:OP3	BSTA, UN	DIRF	Move forward 1 dir entry
6A12	0FF0DB	:	LODA, R0	*TMP, R3	
6A15	1978	:	BCTR, P	OP3	
6A17	CC10E5	:	STRA, R0	SAVE	Remember if end of dir
6A1A	0D7008	:	LODA, R0	SEC, R1	
6A1D	CD7009	:	STRA, R0	DSEC, R1	Start of dir entry
6A20	03	:	LODZ, R3		
6A21	CD700A	:	STRA, R0	DBYT, R1	
6A24	CD7005	:	STRA, R0	PBPT, R1	
6A27	0D700E	:	LODA, R0	DBPT, R1	
6A2A	C3	:	STRZ, R3		Write filename
6A2B	0600	:	LODI, R2	0	
6A2D	E60D	:SDL	COMI, R2	0A+3	
6A2F	9A1D	:	BCFR, LT	SEN	
6A31	0FB070	:	LODA, R0	*BUFF, R3+	
6A34	E42E	:	COMI, R0	A'.'	
6A36	980A	:	BCFR, EQ	OP4	
6A38	E60A	:SC2	COMI, R2	0A	
6A3A	9A71	:	BCFR, LT	SDL	
6A3C	20	:	EORZ, R0		
6A3D	3F6AB1	:IPUTCH	BSTA, UN	PUTCH	Pad name with nulls
6A40	DA76	:	BIRR, R2	SC2	
6A42	E430	:OP4	COMI, R0	30	Separator?
6A44	1A08	:	BCTR, LT	SEN	
6A46	E43D	:	COMI, R0	A'='	= also a separator
6A48	1804	:	BCTR, EQ	SEN	
6A4A	3BF2	:	BSTR, UN	*IPUTCH+1	Ok, put in name or ext
6A4C	DA5F	:	BIRR, R2	SDL	
6A4E	E60D	:SEN	COMI, R2	0A+3	
6A50	1805	:	BCTR, EQ	OP5	
6A52	20	:	EORZ, R0		
6A53	3BE9	:	BSTR, UN	*IPUTCH+1	Pad ext
6A55	DA77	:	BIRR, R2	SEN	
6A57	0C10D9	:OP5	LODA, R0	ADR	
6A5A	3BE2	:	BSTR, UN	*IPUTCH+1	Write start of file
6A5C	0C10DA	:	LODA, R0	ADR+1	
6A5F	3BDD	:	BSTR, UN	*IPUTCH+1	
6A61	0706	:	LODI, R3	6	
6A63	20	:	EORZ, R0		
6A64	3BD8	:	BSTR, UN	*IPUTCH+1	4 pointer, 2 spare
6A66	FB7C	:	BDRR, R3	\$-2	
6A68	0703	:	LODI, R3	3	
6A6A	0C10FC	:	LODA, R0	DATV	Date valid?
6A6D	D80B	:	BIRR, R0	NDAT	No
6A6F	05FF	:	LODI, R2	-1	
6A71	0E30FD	:WDAT	LODA, R0	DATE, R2+	
6A74	3B3B	:	BSTR, UN	PUTCH	Write 3 bytes of date
6A76	FB79	:	BDRR, R3	WDAT	

```

6A78 1B05      :      BCTR, UN CDIR
6A7A 20        :NDAT   EORZ, R0
6A7B 3B34      :      BSTR, UN PUTCH      Else clear them
6A7D FB7B      :      BDRR, R3 $-3
6A7F 0E10E5    :CDIR   LODA, R2 SAVE
6A82 9A0F      :      BCFR, N NONFF      Don't upd end of dir mark
6A84 0D7005    :      LODA, R0 PBPT, R1
6A87 E402      :      COMI, R0 2          New sector?
6A89 9805      :      BCFR, EQ OPG
6A8B 0410      :      LODI, R0 10        Yes, skip gap
6A8D CD7005    :      STRA, R0 PBPT, R1
6A90 02        :OPG    LODZ, R2          End of dir mark
6A91 3B1E      :      BSTR, UN PUTCH
6A93 3F6CFB    :NONFF  BSTA, UN WRSC      Move buffer to floppy
6A96 20        :      EORZ, R0
6A97 CD700D    :      STRA, R0 FTYP, R1    Default type
6A9A CD700B    :      STRA, R0 HLEN, R1
6A9D 0401      :      LODI, R0 1
6A9F CD700C    :      STRA, R0 LLEN, R1      Length=1
6AA2 0E10D9    :      LODA, R2 ADR
6AA5 0F10DA    :      LODA, R3 ADR+1
6AA8 3F6CA2    :      BSTA, UN STD1          Read first sector
6AAB 3F6B7F    :      BSTA, UN GETCH        Inc ptr
6AAE 1F6945    :      BCTA, UN RT          Restore R2R3, return

```

```

:* Put one byte to a file opened for writes
:* Byte is in R0; R1 has str #
:* Saves R0-R3; 3 subr levels if disk

```

```

6AB1 CC10D2    :PUTCH  STRA, R0 SR0
6AB4 CF10D6    :      STRA, R3 SVR3          Save R3
6AB7 0D7005    :      LODA, R0 PBPT, R1      Get buf ptr
6ABA C3        :      STRZ, R3
6ABB 3B2D      :      BSTR, UN STMP
6ABD 0C10D2    :      LODA, R0 SR0          Get char
6AC0 CFF0DB    :      STRA, R0 *TMP, R3     Store it
6AC3 DB1A      :      BIRR, R3 PEND
6AC5 0D700B    :      LODA, R0 HLEN, R1
6AC8 C3        :      STRZ, R3
6AC9 0D700C    :      LODA, R0 LLEN, R1     Increment length
6ACC D802      :      BIRR, R0 $+4
6ACE 8701      :      ADDI, R3 1
6AD0 CD700C    :      STRA, R0 LLEN, R1
6AD3 03        :      LODZ, R3
6AD4 CD700B    :      STRA, R0 HLEN, R1
6AD7 3F6CFB    :      BSTA, UN WRSC        Write current buffer
6ADA 3F6CAE    :      BSTA, UN RDNXSC      Read next to get links
6ADD 0702      :      LODI, R3 2
6ADF 03        :PEND   LODZ, R3
6AE0 CD7005    :      STRA, R0 PBPT, R1
6AE3 0F10D6    :      LODA, R3 SVR3
6AE6 0C10D2    :      LODA, R0 SR0          Restore char
6AE9 17        :      RETC, UN

6AEA 0D700F    :STMP   LODA, R0 HBUF, R1     Get approp buf ptr
6AED CC10DB    :      STRA, R0 TMP          Save
6AF0 17        :      RETC, UN

```


;* Close a file that was opened for writes
 ;* First puts one char from R0
 ;* Only R1 saved; 4 subr levels if disk

6AF1	C2	:CLSW	STRZ, R2	Save char
6AF2	3B76	:	BSTR, UN STMP	
6AF4	0D7005	:	LODA, R0 PBPT, R1	
6AF7	C3	:	STRZ, R3	
6AF8	02	:	LODZ, R2	
6AF9	CFF0DB	:	STRA, R0 *TMP, R3	Store last byte
6AFC	0E90DB	:	LODA, R2 *TMP	Save
6AFF	04FF	:	LODI, R0 -1	End of file mark
6B01	CC90DB	:	STRA, R0 *TMP	
6B04	3F6CFB	:	BSTA, UN WRSC	Write last block
6B07	0D7006	:	LODA, R0 DRV, R1	
6B0A	16	:	RETC, N	Exit if nondisk
6B0B	0D7007	:	LODA, R0 TRK, R1	
6B0E	CC10CE	:	STRA, R0 DENP	Trk, sec of last block
6B11	0D7008	:	LODA, R0 SEC, R1	
6B14	CC10CF	:	STRA, R0 DENP+1	
6B17	CE10D0	:	STRA, R2 DENP+2	Trk, sec of following blk
6B1A	20	:	EDRZ, R0	
6B1B	0CB0DB	:	LODA, R0 *TMP, R0+	
6B1E	CC10D1	:	STRA, R0 DENP+3	
6B21	3F6C92	:	BSTA, UN STDR	
6B24	0C10D0	:	LODA, R0 DENP+2	
6B27	0602	:	LODI, R2 2	
6B29	CEF0DB	:	STRA, R0 *TMP, R2	Update st of freelist
6B2C	0C10D1	:	LODA, R0 DENP+3	
6B2F	CEB0DB	:	STRA, R0 *TMP, R2+	
6B32	0607	:	LODI, R2 7	
6B34	0EF0DB	:	LODA, R0 *TMP, R2	
6B37	AD700C	:	SUBA, R0 LLEN, R1	Upd len of freelist
6B3A	CEF0DB	:	STRA, R0 *TMP, R2	
6B3D	7708	:	PPSL WC	
6B3F	0ED0DB	:	LODA, R0 *TMP, R2-	
6B42	AD700B	:	SUBA, R0 HLEN, R1	
6B45	CEF0DB	:	STRA, R0 *TMP, R2	
6B48	7508	:	CPSL WC	
6B4A	3F6CFB	:IWRSC	BSTA, UN WRSC	Put to disk
6B4D	0D7009	:	LODA, R0 DSEC, R1	
6B50	CD7008	:	STRA, R0 SEC, R1	Get st of file's dir
6B53	0D700A	:	LODA, R0 DBYT, R1	
6B56	840F	:	ADDI, R0 0A+3+2	Skip name, ext, start
6B58	CD7005	:	STRA, R0 PBPT, R1	
6B5B	3F6CBB	:	BSTA, UN RDSC	Get incomplete file dir
6B5E	0C10CE	:	LODA, R0 DENP	
6B61	3F6AB1	:IP1	BSTA, UN PUTCH	End of file
6B64	0C10CF	:	LODA, R0 DENP+1	
6B67	3BF9	:	BSTR, UN *IP1+1	
6B69	0D700B	:	LODA, R0 HLEN, R1	
6B6C	3BF4	:	BSTR, UN *IP1+1	Length
6B6E	0D700C	:	LODA, R0 LLEN, R1	
6B71	3BEF	:	BSTR, UN *IP1+1	
6B73	0D700D	:	LODA, R0 FTYP, R1	File type
6B76	3BEA	:	BSTR, UN *IP1+1	
6B78	3BD1	:	BSTR, UN *IWRSC+1	Write completed dir

;* Close file opened for reads

```
6B7A 20      :CLSR      EORZ,R0
6B7B CD7000  :          STRA,R0 IDNT,R1      Close stream
6B7E 17      :          RETC,UN
```

;* Get one byte from file opened for reads or writes
 :* Returns byte in R0; Strm # must be in R1
 :* Saves R1-R3; 3 subr levels if disk

```
6B7F 3F6AEA  :GETCH     BSTA,UN STMP          Setup buff ptr
6B82 0D7005  :          LODA,R0 PBPT,R1      Pros Buf PTR
6B85 D805    :          BIRR,R0 HERE
6B87 3F6CAE  :          BSTA,UN RDNXSC        Get next sector
6B8A 0402    :          LODI,R0 2          Skip links
6B8C CD7005  :HERE      STRA,R0 PBPT,R1
6B8F 0CF0DB  :          LODA,R0 *TMP,R0      Get char
6B92 17      :          RETC,UN

6B93 3F6D86  :NFC         BSTA,UN NFD
```


** Find file, load & execute it
 ** Note MUST use stream 2; uses 5 subr levels

6B96	3F6C15	:CHAIN	BSTA, UN FIND	Loc file
6B99	1A78	:	BCTR, LT NFC	
** Execute entry from command module				
6B9B	3F6C98	:CHAINI	BSTA, UN FIRST	Get first blk
6B9E	20	:	EDRZ, R0	
6B9F	CC10DA	:	STRA, R0 ADR+1	
6BA2	3B5B	:CHAINL	BSTR, UN GETCH	
6BA4	CC10D9	:	STRA, R0 ADR	Address
6BA7	3B56	:	BSTR, UN GETCH	
6BA9	C3	:	STRZ, R3	
6BAA	3B53	:	BSTR, UN GETCH	Block length
6BAC	1827	:	BCTR, Z CXIT	
6BAE	C2	:	STRZ, R2	
6BAF	0D7005	:	LODA, R0 PBPT, R1	Use registers for speed
6BB2	C1	:	STRZ, R1	
6BB3	D907	:MOVE	BIRR, R1 MOV	
6BB5	0520	:	LODI, R1 20	From str 2,
6BB7	3F6CAE	:	BSTA, UN RDNXSC	set next sector
6BBA	0502	:	LODI, R1 2	Skip links only
6BBC	0DF0DB	:MOV	LODA, R0 *TMP, R1	Move from buffer
6BBF	CFF0D9	:	STRA, R0 *ADR, R3	to RAM
6BC2	DB07	:	BIRR, R3 IADR	
6BC4	20	:	EDRZ, R0	
6BC5	8C30D8	:	ADDA, R0 ADR-1, R0+	Inc RAM ptr
6BC8	CC10D9	:	STRA, R0 ADR	
6BCB	FA66	:IADR	BDRR, R2 MOVE	Repeat [blk len] times
6BCD	01	:	LODZ, R1	
6BCE	0520	:CNXT	LODI, R1 20	
6BD0	CD7005	:	STRA, R0 PBPT, R1	Save for GETCHs
6BD3	1B4D	:	BCTR, UN CHAINL	Start new block
6BD5	04FF	:CXIT	LODI, R0 -1	
6BD7	0D90DB	:	LODA, R1 *TMP	Last section?
6BDA	9A72	:	BCFR, N CNXT	No, load new sector
6BDC	9FF0D9	:	BXA *ADR, R3	Yes, autostart
6BDF	8717	:DIRF	ADDI, R3 17	Advance 1 dir entry
6BE1	DB28	:DIR1	BIRR, R3 DRET	Advance 1 byte
6BE3	0710	:	LODI, R3 10	All start at byte 10
6BE5	1F6CAE	:	BCTA, UN RDNXSC	Load next sector of dir
** Get directory entry				
6BE8	0D7005	:GDEN	LODA, R0 PBPT, R1	
6BEB	C3	:	STRZ, R3	
6BEC	3B73	:	BSTR, UN DIR1	Inc ptr by 1
6BEE	0FF0DB	:GDNL	LODA, R0 *TMP, R3	Look at 1st char of name
6BF1	16	:	RETC, N	Found end of dir (FF)
6BF2	1904	:	BCTR, P GOT	Real entry
6BF4	3B69	:	BSTR, UN DIRF	Skip whole entry if null
6BF6	1B76	:	BCTR, UN GDNL	
6BF8	06FF	:GOT	LODI, R2 -1	
6BFA	A701	:	SUBI, R3 1	

```

6BFC 0FB0DB :GDL LODA,R0 *TMP,R3+ Copy from buffer
6BFF CE30B6 : STRA,R0 DENT,R2+ to DENT
6C02 E617 : COMI,R2 18-1
6C04 9876 : BCFR,EQ GDL
6C06 03 : LODZ,R3
6C07 CD7005 : STRA,R0 PBPT,R1 Save in PBPT
6C0A E0 : COMZ,R0 Flag as found
6C0B 17 :DRET RETC,UN

```

```

:** Find next file satisfying a wildcard spec
:** Only R1 preserved; 4 subr levels

```

```

6C0C 0D7006 :FNEXT LODA,R0 DRV,R1 Not found if
6C0F 16 : RETC,LT non disk
6C10 3F6AEA : BSTA,UN STMP Setup TMP
6C13 1B11 : BCTR,UN FNX

```

```

:** Find first file satisfying a spec at *BUFF
:** Only R1 preserved; 4 subr levels

```

```

6C15 3F694C :FIND BSTA,UN ASGN
6C18 0D7006 : LODA,R0 DRV,R1
6C1B E4FF : COMI,R0 -1 If not disk,
6C1D 14 : RETC,EQ consider found
6C1E 3F6C92 : BSTA,UN STDR
6C21 040F : LODI,R0 0F
6C23 CD7005 : STRA,R0 PBPT,R1 Skip freelist
6C26 3B40 :FNX BSTR,UN GDEN
6C28 16 : RETC,N Not found
:** Compare with *BUFF
6C29 0D700E : LODA,R0 DBPT,R1
6C2C C3 : STRZ,R3 R3 steps thru name at *BUFF
6C2D 0600 : LODI,R2 0 R2 is match count/DENT index
6C2F 0FB070 :FDL LODA,R0 *BUFF,R3+
6C32 E42E : COMI,R0 A'.' End of name?
6C34 1819 : BCTR,EQ F1
6C36 E42A : COMI,R0 A'*'
6C38 1A39 : BCTR,LT FEN1 < * is separator
6C3A 9804 : BCFR,EQ FDL1
6C3C 060A : LODI,R2 0A Match 10 chars if *
6C3E 1B6F : BCTR,UN FDL
6C40 E43F :FDL1 COMI,R0 A'?' Match current char if ?
6C42 1809 : BCTR,EQ FOK
6C44 E42C : COMI,R0 A',' Exit if separator
6C46 182B : BCTR,EQ FEN1
6C48 EE70B6 : COMA,R0 DENT,R2 Compare chars
6C4B 9859 : BCFR,EQ FNX Mismatch, get next entry
6C4D DA60 :FOK BIRR,R2 FDL Inc match count, loop

6C4F E60A :F1 COMI,R2 0A
6C51 1807 : BCTR,EQ FEX 10 chars so unique
6C53 0E70B6 : LODA,R0 DENT,R2
6C56 984E : BCFR,Z FNX Mismatch if not null
6C58 060A : LODI,R2 0A

6C5A 0FB070 :FEX LODA,R0 *BUFF,R3+ Get ext char
6C5D E42A : COMI,R0 A'*'

```



```

6C5F 1A16      :      BCTR,LT FEN
6C61 181E      :      BCTR,EQ FGO
6C63 E43F      :      COMI,R0 A'?'
6C65 180A      :      BCTR,EQ FKO
6C67 E42C      :      COMI,R0 A', '
6C69 180C      :      BCTR,EQ FEN
6C6B EE70B6    :      COMA,R0 DENT,R2
6C6E 9C6C26    :      BCFA,EQ FNX
6C71 DA67      :FKO    BIRR,R2 FEX

6C73 E60A      :FEN1   COMI,R2 0A'      End of dir name ?
6C75 180A      :      BCTR,EQ FGO
6C77 E60D      :FEN    COMI,R2 0A+3      End of dir name.ext ?
6C79 1806      :      BCTR,EQ FGO
6C7B 0E70B6    :      LODA,R0 DENT,R2      No; null in dir name?
6C7E 9C6C26    :      BCFA,Z FNX      No, so not found yet
6C81 0401      :FGO    LODI,R0 1      Flag as found
6C83 17        :      RETC,UN

```

```

        :* Read non-disk device
6C84 0702      :RDND   LODI,R3 2
6C86 7510      :RDNDL  CPSL RS
6C88 3F6EE2    :      BSTA,UN 10      Must preserve R3'
6C8B 7710      :      PPSL RS
6C8D DB77      :      BIRR,R3 RDNDL
6C8F 7510      :RTN    CPSL RS
6C91 17        :      RETC,UN

```

```

        :* Move to start of directory
        :* Saves R1 only; 2 subr levels

```

```

6C92 0600      :STDR   LODI,R2 0
6C94 0701      :      LODI,R3 1
6C96 1B0A      :      BCTR,UN STD1

```

```

        :* Get first sector of found file to buffer

```

```

6C98 070D      :FIRST  LODI,R3 0A+3
6C9A 0F70B6    :      LODA,R0 DENT,R3      Start track
6C9D C2        :      STRZ,R2
6C9E 0F30B6    :      LODA,R0 DENT,R3+      St sec
6CA1 C3        :      STRZ,R3

```

```

        :* Move to Track R2, Sector R3

```

```

6CA2 0401      :STD1   LODI,R0 1
6CA4 CD7005    :      STRA,R0 PBPT,R1      Skip links
6CA7 02        :      LODZ,R2
6CA8 CD7007    :      STRA,R0 TRK,R1
6CAB 03        :      LODZ,R3
6CAC 1B0A      :      BCTR,UN SSEC      Get this sector

```

```

** Read next sector
** TMP must be pointing to buffer links
** Saves R1-R3; 2 subr levels

```

```

6CAE 0C90DB :RDNXSC LODA,R0 *TMP      Trk,Sec of next block
6CB1 CD7007 :      STRA,R0 TRK,R1
6CB4 20      :      EORZ,R0
6CB5 0CB0DB :      LODA,R0 *TMP,R0+
6CB8 CD7008 :SSEC  STRA,R0 SEC,R1

```

```

** Read one sector from disk to buffer
** Saves R1-R3; 2 subr levels if disk, or M+1 subr
** levels if non-disk, where M is levels of service rtne

```

```

6CBB 3F6D57 :RDSC  BSTA,UN INIT
6CBE 1A44    :      BCTR,N RDND      If non disk
6CC0 3F6D9B :      BSTA,UN SK      Seek correct track
6CC3 0603    :      LODI,R2 RETRYC
6CC5 048C    :RDL  LODI,R0 SC      Read sector command
6CC7 D40C    :      WRTE,R0 FDCMD
6CC9 5410    :L4  REDE,R0 FDINT      DRQ?
6CCB 9A7C    :      BCFR,N L4
6CCD 540F    :      REDE,R0 FDDAT      Move from data res
6CCF CFF0DB :      STRA,R0 *TMP,R3      into approp buffer
6CD2 DB75    :      BIRR,R3 L4
6CD4 3B1E    :      BSTR,UN WBSY
6CD6 441C    :      ANDI,R0 RNF+CRC+LSD
6CD8 1C6C8F :      BCTA,Z RTN
6CDB F410    :      TMI,R0 RNF      Record not found?
6CDD 3C6DD7 :      BSTA,EQ RESEEK
6CE0 FA63    :      BDRR,R2 RDL      Retry
6CE2 7510    :      CPSL RS
6CE4 0D7008 :      LODA,R0 SEC,R1
6CE7 C3      :      STRZ,R3      Save offending track
6CE8 560D    :      REDE,R2 FDTRK      and sector
6CEA 0452    :      LODI,R0 A'R'
6CEC 1F68AE :      BCTA,UN ERR      Read error

```

```

6CEF 08FE    :WBSDD LODR,R0 *$      Super delay entry
6CF1 C0      :      NOP      to WBSY
6CF2 1B05    :      BCTR,UN WBSD

```

```

** Wait for drive not busy

```

```

6CF4 540C    :WBSY  REDE,R0 FDCMD
6CF6 F401    :      TMI,R0 BUSY
6CF8 16      :      RETC,NE
6CF9 1B79    :WBSD  BCTR,UN WBSY      Delay entry

```



```

** Write a sector from buffer to disk
** Saves R1-R3; 2 subr levels if disk, or M+1 subr levels
** if non-disk.

```

```

6CFB 3F6D57 :WRSC      BSTA,UN INIT
6CFE 1E6E01 :          BCTA,N  WRND      Non disk

6D01 3F6D9B :          BSTA,UN SK
6D04 0603 :          LODI,R2 RETRYC
6D06 CE10E4 :NWPB      STRA,R2 RTYCNT
6D09 04AC :          LODI,R0 0AC      Write sector command
6D0B D40C :          WRTE,R0 FDCMD
6D0D 5410 :          REDE,R0 FDINT
6D0F 9A7C :          BCFR,N  $-2      wait for DRQ or INTERRUPT
6D11 540C :          REDE,R0 FDCMD
6D13 F440 :          TMI,R0  WPRT      Check wrt prot bit
6D15 183C :          BCTR,EQ WPE
6D17 4498 :          ANDI,R0 NREADY+RNF+CRC
6D19 9821 :          BCFR,Z  WERR
6D1B 0FF0DB :L5          LODA,R0 *TMP,R3      Get from buffer
6D1E 5610 :          REDE,R2 FDINT
6D20 9A7C :          BCFR,N  $-2      wait for DRQ
6D22 D40F :          WRTE,R0 FDDAT      Into data reg
6D24 DB75 :          BIRR,R3 L5
6D26 540C :WWAIT      REDE,R0 FDCMD
6D28 F401 :          TMI,R0  BUSY
6D2A 987A :          BCFR,EQ WWAIT
6D2C 4414 :          ANDI,R0 RNF+LSD
6D2E 980C :          BCFR,Z  WERR      write error
**Check CRC of sector just written
6D30 048C :          LODI,R0 8C      Read sector's command
6D32 D40C :          WRTE,R0 FDCMD
6D34 3F6CEF :          BSTA,UN WBSDD      Wait for 1771 to interrupt out
6D37 4418 :          ANDI,R0 CRC+RNF
6D39 1C6C8F :          BCTA,Z  RTN      OK
6D3C F410 :WERR      TMI,R0  RNF      Record not found?
6D3E 3C6DD7 :          BSTA,EQ RESEEK
6D41 0E10E4 :          LODA,R2 RTYCNT
6D44 FA40 :          BDRR,R2 NWPB      try again
6D46 7510 :          CPSL  RS
6D48 0D7008 :          LODA,R0 SEC,R1
6D4B C3 :          STRZ,R3
6D4C 560D :          REDE,R2 FDTRK      Save offending track
6D4E 0457 :          LODI,R0 A'W'      and sector
6D50 1F68AE :WRR      BCTA,UN ERR      Write error

6D53 0450 :WPE      LODI,R0 A'P'      Write protected
6D55 1879 :          BCTR,EQ WRR

```

```

6D57 01      :INIT      LODZ, R1
6D58 7710    :          PPSL      RS
6D5A C1      :          STRZ, R1      Save stream index
6D5B 0D700F  :          LODA, R0 HBUF, R1
6D5E CC10DB  :          STRA, R0 TMP
6D61 0D7006  :          LODA, R0 DRV, R1
6D64 16      :          RETC, N

```

```

** Select drive given in R0
** Saves R1-R3; 1 subr level

```

```

6D65 CD10D5  :DRIVE     STRA, R1 SVR1      Save
6D68 0D10E6  :          LODA, R1 CDRV
6D6B D410    :          WRTE, R0 FDINT      Select side, drive
6D6D 4403    :          ANDI, R0 3      Isolate drive
6D6F CC10E6  :          STRA, R0 CDRV      Save drive
6D72 E1      :          COMZ, R1      compare with current
6D73 180D    :          BCTR, EQ DREX      Same; exit
6D75 540D    :          REDE, R0 FDTRK      Current track
6D77 CD70B2  :          STRA, R0 TRK1, R1      Save it
6D7A 0D10E6  :          LODA, R1 CDRV      new drive
6D7D 0D70B2  :          LODA, R0 TRK1, R1
6D80 D40D    :          WRTE, R0 FDTRK      to FD1771
6D82 0D10D5  :DREX      LODA, R1 SVR1
6D85 17      :          RETC, UN

```

```

** Print "Not found...", return if escape typed
** Saves R1, R2; 2+M subr levels

```

```

6D86 CD10D5  :NFD      STRA, R1 SVR1      Save str #
6D89 071E    :          LODI, R3 NOTF-STR      "File not found"
6D8B 3F6F01  :          BSTA, UN PRCR
6D8E 3F0286  :          BSTA, UN KBD
6D91 E41B    :          COMI, R0 1B      Escape?
6D93 9C6893  :          BCFA, EQ ENTER      No, goto ENTER
6D96 3F6EDC  :          BSTA, UN CRLF      Echo
6D99 1B67    :          BCTR, UN DREX      Restore R1, return

```



```

** Seek to requested track
6D9B 0714 :SK LODI, R3 D'20'
6D9D 540C : REDE, R0 FDCMD
6D9F 4420 : ANDI, R0 HDENG
6DA1 1807 : BCTR, Z SKHLD head not loaded
6DA3 540D : REDE, R0 FDTRK
6DA5 ED7007 : COMA, R0 TRK, R1 Already on read trk?
6DA8 1813 : BCTR, EQ SRDY Yes, no seek
6DAA 0D7007 :SKHLD LODA, R0 TRK, R1
6DAD 9A04 : BCFR, N SKOK
6DAF 044E : LODI, R0 A'F' F error- disk
6DB1 1B18 : BCTR, UN CERR or dir full
6DB3 D40F :SKOK WRTE, R0 FDDAT Track
6DB5 0418 : LODI, R0 18+TTD Seek with HLD
6DB7 D40C : WRTE, R0 FDCMD
6DB9 5410 :W1 REDE, R0 FDINT
6DBB 9A7C : BCFR, N W1
**Wait for drive ready
**Loop for 100 mSec
6DBD 0655 :SRDY LODI, R2 D'85'
**Inner loop 1mSec
6DBF 540C :SRDY1 REDE, R0 FDCMD wait for drive ready
6DC1 9A0C : BCFR, N SKD
6DC3 F87E : BDRR, R0 $
6DC5 FA78 : BDRR, R2 SRDY1
6DC7 FB61 : BDRR, R3 SKHLD seek to keep head loaded
**Drive not ready after 2 Sec
6DC9 044E : LODI, R0 A'N' Drive Not ready error
6DCB 7510 :CERR CPSL RS
6DCD 1B2F : BCTR, UN IERR
6DCF 0D7008 :SKD LODA, R0 SEC, R1
6DD2 D40E : WRTE, R0 FDSEC Write sector reg
6DD4 0700 : LODI, R3 0
6DD6 17 : RETC, UN
6DD7 0703 :RESEEK LODI, R3 RETRYC
6DD9 040B :REPT LODI, R0 08+TTD
6ddb D40C : WRTE, R0 FDCMD restore
6DDD 5410 :W2 REDE, R0 FDINT
6DDF 9A7C : BCFR, N W2
6DE1 0D7007 : LODA, R0 TRK, R1
6DE4 D40F : WRTE, R0 FDDAT Track
6DE6 041F : LODI, R0 1C+TTD Seek with HLD and Ver
6DE8 D40C : WRTE, R0 FDCMD
6DEA 5410 :W3 REDE, R0 FDINT
6DEC 9A7C : BCFR, N W3
6DEE 540C : REDE, R0 FDCMD
6DF0 F410 : TMI, R0 RNF Seek error?
6DF2 985B : BCFR, EQ SKD
6DF4 FB63 : BDRR, R3 REPT
6DF6 0D7007 : LODA, R0 TRK, R1 Get offending trk #
6DF9 7510 : CPSL RS
6DFB C2 : STRZ, R2
6DFC 0453 : LODI, R0 A'S' Seek error - can't find track
6DFE 1F68AE :IERR BCTA, UN ERR

```

```

        ** Write buffer to non-disk dev
6E01 0702 :WRND LODI, R3 2
6E03 0FF0DB :WRNDL LODA, R0 *TMP, R3
6E05 7510 : CPSL RS
6E08 3F6EE2 : BSTA, UN IO
6E0B 7710 : PPSL RS
6E0D DB74 : BIRR, R3 WRNDL
6E0F 7510 : CPSL RS
6E11 17 : RETC, UN

```

```

        ** Print filename from DENT (Directory ENTry)
        ** Use str in R1 (can write to disk, etc)
        ** Saves R1, R2; M+1 subr levels

```

```

6E12 07FF :PNAM LODI, R3 -1
6E14 0F30B6 : LODA, R0 DENT, R3+
6E17 1802 : BCTR, Z CPN
6E19 3B87 : BSTR, UN *IID+1 (IO)
6E1B E709 :CPN COMI, R3 0A-1
6E1D 9805 : BCFR, EQ $+7
6E1F 042E : LODI, R0 A', '
6E21 3F6EE2 :IID BSTA, UN IO
6E24 E70C : COMI, R3 0A+3-1
6E26 986C : BCFR, EQ PNAM+2
6E28 17 : RETC, UN

```

```

        ** Print R2R3 as decimal to stream R1
        ** R2R3 NOT preserved; M+2 subr levels

```

```

6E29 CD10E5 :DOUT STRA, R1 SAVE Save stream no.
6E2C 0500 : LODI, R1 0 Hundreds
6E2E 770B : PPSL WC+CAR+LCOM
6E30 02 :LP1 LODZ, R2
6E31 9804 : BCFR, Z D1
6E33 E764 : COMI, R3 D'100'
6E35 1A06 : BCTR, LT P1
6E37 A764 :D1 SUBI, R3 D'100' Subt dec 100
6E39 A600 : SUBI, R2 0
6E3B D973 : BIRR, R1 LP1
6E3D 7508 :P1 CPSL WC
6E3F 01 : LODZ, R1
6E40 C2 : STRZ, R2 Save hundreds
6E41 7810 : BSNR, R0 PRDG Print digit unless zero
6E43 0500 : LODI, R1 0
6E45 E70A :LP2 COMI, R3 D'10'
6E47 1A04 : BCTR, LT P2
6E49 A70A : SUBI, R3 D'10'
6E4B D97B : BIRR, R1 LP2
6E4D 01 :P2 LODZ, R1
6E4E 5A0B : BRNR, R2 FORC If hunds, force tens
6E50 7801 : BSNR, R0 PRDG Print tens
6E52 03 :UNS LODZ, R3 Units
6E53 8430 :PRDG ADDI, R0 30 Print digit
6E55 0D10E5 : LODA, R1 SAVE
6E58 1F6EE2 : BCTA, UN IO

```



```

6E5B 3B75 :FORC BSTR, UN PRDG
6E5D 1B73 : BCTR, UN UNS

```

```

:* Return CC=EQ if R0 dec char

```

```

6E5F E430 :DEC COMI, R0 A' 0'
6E61 16 : RETC, LT
6E62 E439 : COMI, R0 A' 9'
6E64 15 : RETC, GT
6E65 E0 : COMZ, R0
6E66 17 : RETC, UN

```



Returns R0 unchanged

```

:* Loc next item in buffer
:* Saves R1, R2; single subr level

```

```

6E67 07FF :ITEM LODI, R3 -1
6E69 0FB070 :ITEML LODA, R0 *BUFF, R3+
6E6C E42C : COMI, R0 A', '
6E6E 1B0F : BCTR, EQ SEPS
6E70 E43D : COMI, R0 A' '='
6E72 9B04 : BCFR, EQ NTEQ
6E74 5B10 : BRNR, R3 IGO ABC=
6E76 1B07 : BCTR, UN SEPS =ABC
6E78 E420 :NTEQ COMI, R0 20
6E7A 196D : BCTR, GT ITEML Skip anything significant
6E7C E40D :IT2 COMI, R0 0D
6E7E 14 : RETC, EQ CC = EQ => no item found
6E7F 0FB070 :SEPS LODA, R0 *BUFF, R3+
6E82 E420 : COMI, R0 20
6E84 9976 : BCFR, GT IT2 Skip trailing seps
6E86 8F1071 :IGO ADDA, R3 BUFF+1 Update BUFF
6E89 CF1071 : STRA, R3 BUFF+1
6E8C 17 : RETC, UN Ignore carry

```

```

:* Accept line of chars, put to BUF
:* Handles various control chars
:* Converts to upper case in buffer
:* No regs saved; 2+M subr levels, where
:* M is levels of COUT (M=3 in CBUG)

```

```

6E8D 0718 :PRT LODI, R3 CTU-STR
6E8F 3F6F03 : BSTA, UN PRNT "AU", CR, LF

6E92 0701 :LINE LODI, R3 1
6E94 0470 : LODI, R0 <BUF Point to normal
6E96 CC1070 : STRA, R0 BUFF line buffer
6E99 0472 : LODI, R0 BUF
6E9B CC1071 : STRA, R0 BUFF+1
6E9E 0510 :LLOP LODI, R1 10
6EA0 3F6EE2 : BSTA, UN ID Input from strm 1
6EA3 E401 : COMI, R0 1 Cntrl A
6EA5 1C0022 : BCTA, EQ 22
6EA8 E403 : COMI, R0 3 Cntrl C / ETX?
6EAA 1C6899 : BCTA, EQ ENTERK
6EAD E412 : COMI, R0 12 Cntrl R
6EAF 1B29 : BCTR, EQ CRLF0

```

```

6EB1 E415      :      COMI, R0 15      Cntrl U
6EB3 1858      :      BCTR, E0 PR1
6EB5 E47F      :      COMI, R0 7F      DELETE
6EB7 1804      :      BCTR, E0 N7
6EB9 E409      :      COMI, R0 08      BS
6EBB 9808      :      BCFR, E0 N8
6EBD A701      :N7      SUBI, R3 1
6EBF 1851      :      BCTR, Z LINE
6EC1 0408      :      LODI, R0 08      BS
6EC3 180D      :      BCTR, UN N9
6EC5 E461      :N8      COMI, R0 A'a'
6EC7 1A02      :      BCTR, LT $+4
6EC9 445F      :      ANDI, R0 5F      Conv to upper case
6ECB CF3070     :      STRA, R0 BUF-2, R3+
6ECE E40D      :      COMI, R0 0D
6ED0 1809      :      BCTR, E0 CRLF0     Exit on buffer full
6ED2 0500      :N9      LODI, R1 0
6ED4 3B0C      :      BSTR, UN IO
6ED6 E740      :      COMI, R3 40
6ED8 9844      :      BCFR, E0 LLOP     Exit on buffer full

6EDA 0500      :CRLF0  LODI, R1 0      Put CRLF on stream 0

      :* Put CR, LF on stream R1

6EDC 040D      :CRLF  LODI, R0 0D
6EDE 3B02      :      BSTR, UN IO
6EE0 040A      :      LODI, R0 0A

      :* Do one byte of I/O on stream R1
      :* Saves R1-R3; same subr levels as service routine

6EE2 CC10D2     :IO      STRA, R0 SR0
6EE5 0D7000     :      LODA, R0 IDNT, R1
6EE8 9905      :      BCFR, Z IOPN
6EEA 044F      :      LODI, R0 A'O'
6EEC 1F68AE     :      BCTA, UN ERR      Stream not open
6EEF 0D7003     :IOPN     LODA, R0 CODE, R1
6EF2 CC10D9     :      STRA, R0 ADR
6EF5 0D7004     :      LODA, R0 CODE+1, R1
6EF8 CC10DA     :      STRA, R0 ADR+1
6EFB 0C10D2     :      LODA, R0 SR0
6EFE 1FF0D9     :      BCTA, UN *ADR      Execute

6F01 3B57      :PRCR     BSTR, UN CRLF0
6F03 0500      :PRNT     LODI, R1 0      Print string on stream 0
6F05 0F2F99     :      LODA, R0 STR-1, R3+
6F08 14        :      RETC, Z
6F09 3B57      :      BSTR, UN IO
6F0B 1B78      :      BCTR, UN PRNT+2

      :* Print R0R2 as hex address
      :* Saves R1, R3; M+2 subr levels

6F0D 3B01      :PADR     BSTR, UN HXOUT
6F0F 02        :      LODZ, R2

```



```

** Print R0 as hex on str R1
** Equiv Mon BOUT
** Saves R1-R3; M+1 subr levels

```

```

6F10 CC10E5 :HXOUT STRA,R0 SAVE
6F13 50 : RRR,R0
6F14 50 : RRR,R0
6F15 50 : RRR,R0
6F16 50 : RRR,R0
6F17 3B03 : BSTR,UN HOUT
6F19 0C10E5 : LODA,R0 SAVE
6F1C 440F :HOUT ANDI,R0 0F
6F1E 8430 : ADDI,R0 A'0'
6F20 E439 : COMI,R0 A'9'
6F22 9902 : BCFR,GT $+4
6F24 8407 : ADDI,R0 7 If A-F, add 7
6F26 1F6EE2 : BCTA,UN 10

```

```

** Get number from buffer to XGOTO
** Equiv Mon GNUM
** Saves R1-R3; 2 subr levels

```

```

6F29 771A :GNMB PPSL LCOM+RS+WC
6F2B 20 : EORZ,R0
6F2C C1 : STRZ,R1
6F2D C2 : STRZ,R2
6F2E 07FF : LODI,R3 -1
6F30 CC10D2 :LNUM STRA,R0 SR0
6F33 0FB070 : LODA,R0 *BUFF,R3+
6F36 E40D : COMI,R0 0D C/r?
6F38 1829 : BCTR,EQ OUTX
6F3A 7701 : PPSL CAR
6F3C A430 : SUBI,R0 A'0' Leaves carry on
6F3E 1A1E : BCTR,N DNUM Skip leading seps
6F40 E409 : COMI,R0 9
6F42 990A : BCFR,GT GNOK 0-9 is OK
6F44 A407 : SUBI,R0 7
6F46 E40A : COMI,R0 0A Else should be A
6F48 1A28 : BCTR,LT HER
6F4A E40F : COMI,R0 0F to F
6F4C 1924 : BCTR,GT HER
6F4E D2 :GNOK RRL,R2 Shift R1R2 left 4 bits
6F4F D1 : RRL,R1
6F50 D2 : RRL,R2
6F51 D1 : RRL,R1
6F52 D2 : RRL,R2
6F53 D1 : RRL,R1
6F54 D2 : RRL,R2
6F55 D1 : RRL,R1
6F56 46F0 : ANDI,R2 0F0
6F58 52 : IORZ,R2 Include new digit
6F59 C2 : STRZ,R2
6F5A 0401 : LODI,R0 1
6F5C 1B52 : BCTR,UN LNUM Set Byte flag and loop for next

6F5E 0C10D2 :DNUM LODA,R0 SR0 Found No yet?
6F61 184D : BCTR,Z LNUM Ignore leading spaces

```

6F63	7508	:OUTX	CPSL	WC	
6F65	3F6E86	:	BSTA, UN	IGD	Update BUFF
6F68	CD10DD	:	STRA, R1	XGOTO	
6F6B	CE10DE	:	STRA, R2	XGOTO+1	
6F6E	01	:	LODZ, R1		Return high order in R0
6F6F	7510	:	CPSL	RS	
6F71	17	:	RETC, UN		

6F72	7518	:HER	CPSL	RS+WC	
6F74	0448	:	LODI, R0	A' H'	Hex error
6F76	1F68AE	:	BCTA, UN	ERR	
6F79	E404	:VDU	COMI, R0	4	Simple VDU dev handler
6F7B	16	:	RETC, LT		
6F7C	1F02B4	:	BCTA, UN	COU	

/* Device table

6F7F	5644	:DEVS	DATA	A' VD'	Vdu
6F81	6F79	:	ACON	VDU	
6F83	4B42	:	DATA	A' KB'	Keyboard
6F85	0286	:	ACON	KBD	
6F87	4C50	:	DATA	A' LP'	Printer (LPT:)
6F89	70F0	:	ACON	ULPT	
6F8B	FF	:	DATA	0FF	

6F8C	564455	:VDUB	DATA	A' VDU'	
6F8F	6F79	:	ACON	VDU	
6F91	00FF	:	DATA	0, 0FF	
6F93	4B4244	:KBDB	DATA	A' KBD'	
6F95	0286	:	ACON	KBD	
6F98	00FF	:	DATA	0, 0FF	

6F9A	09093236	:STR	DATA	9, 9, A' 2650 DOS Vers 2.6a', 0	
6FAF	3A3A00	:PMT	DATA	A' ::', 0	
6FB2	205E550D	:CTU	DATA	A' ^U', 0D, 0A, 0	
6FB8	46696C65	:NOTF	DATA	A' File not found...', 0	
6FCA	45786563	:EX	DATA	A' Executing ', 0	
6FD5	416C7265	:ARE	DATA	A' Already exists...', 0	
6FE7	6FE9	:STRUP	ACON	KOLD	
6FE9	312E5354	:KOLD	DATA	A' 1. ST. SYS', 0D	

6FF2	44534B	:DSKD	DATA	A' DSK'	
6FF5	434D44	:CMDS	DATA	A' CMD'	
6FF8	0001	:DFDRV	DATA	0, DUAL	Default sys, work drives

	:	ORG	RAM	
7000	:IDNT	RES	3	Stream identity, 3 chars
7003	:CODE	RES	2	Addr of service routine
7005	:PBPT	RES	1	Program Buffer Pointer
7006	:DRV	RES	1	Bit 1,0=drv #: bit 2=side
7007	:TRK	RES	1	Track #
7008	:SEC	RES	1	Sector #
7009	:DSEC	RES	1	Directory sector
700A	:DBYT	RES	1	Dir byte
700B	:HLEN	RES	1	Length (for
700C	:LLEN	RES	1	writes)
700D	:FTYP	RES	1	File type (0=ordinary)
700E	:DBPT	RES	1	Device Buffer Pointer
700F	:HBUF	RES	1	High byte of buffer addr

	:	ORG	RAM+70	
7070	:BUFF	RES	2	Line buffer pointer
7072	:BUF	RES	40	Normal line buffer
70B2	:TRK1	RES	4	Track info for dual drive
70B6	:DENT	RES	18	Directory ENTRY copy
70CE	:DENP	RES	4	Scratch
70D2	:SR0	RES	1	Save res
70D3	:SR2	RES	1	
70D4	:SR3	RES	1	
70D5	:SVR1	RES	1	
70D6	:SVR3	RES	1	
70D7	:SYSD	RES	1	Sys drive (0 or 1)
70D8	:WRKD	RES	1	Work drv
70D9	:ADR	RES	2	
70DB	:TMP	RES	2	Disk buffer pointer
70DD	:XGOTO	RES	2	Res of GNUM*
70DF	:ERRDR	RES	1	Spare
70E0	:ERR0	RES	1	Error info
70E1	:ERR1	RES	1	
70E2	:ERR2	RES	1	
70E3	:ERR3	RES	1	
70E4	:RTYCNT	RES	1	retry count
70E5	:SAVE	RES	1	
70E6	:CDRV	RES	1	Current drive

	:	ORG	RAM+0FC	
70FC	:DATV	RES	1	Date valid flag
70FD	:DATE	RES	3	DD, MM, YY in BCD

ADR	70D9	ARE	6FD5	AS1	6957	AS2	6961	AS3	6972
AS4	698C	ASGN	694C	BUF	7072	BUFF	7070	BUSY	0001
CCMD	68FB	CDIR	6A7F	CDRV	70E6	CERR	6DCB	CHAIN	6B96
CHAINI	6B9B	CHAINL	6BA2	CLSR	6B7A	CLSW	6AF1	CMD5	6FF5
CNXT	6BCE	COD	6998	CODE	7003	CODI	69BD	CODL	699C
COUT	02B4	CPN	6E1B	CRC	0008	CRLF	6EDC	CRLF0	SEDA
CTU	6FB2	CXIT	6BD5	D1	6E37	DATE	70FD	DATV	70FC
DBPT	700E	DBYT	700A	DEC	6E5F	DENP	70CE	DENT	70B6
DEVS	6F7F	DFDRV	6FF8	DIR1	6BE1	DIRF	6BDF	DNUM	6F5E
DOS	683D	DOUT	6E29	DPR	697C	DPRL	6982	DR	6954
DRET	6C0B	DREX	6D82	DRIVE	6D65	DRV	7006	DSEC	7009
DSKD	6FF2	DUAL	0001	DVN	696E	ENTER	6893	ENTERK	6889
ERR	68AE	ERR0	70E0	ERR1	70E1	ERR2	70E2	ERR3	70E3
ERRD	69B2	ERRDR	70DF	EX	6FCA	?EXEC	6907	F1	6C4F
FCMD	68DE	FCMD1	68E3	FD1771	000C	FDCMD	000C	FDDAT	000F
FDINT	0010	FDIR	68F6	FDL	6C2F	FDL1	6C40	FDSEC	000E
FDTRK	000D	FEN	6C77	FEN1	6C73	FEX	6C5A	FG0	6C81
FIND	6C15	FIRST	6C98	FKD	6C71	FNEXT	6C0C	FNX	6C26
FOK	6C4D	FORC	6E5B	FTYP	700D	GDEN	6BE8	GDL	6BFC
GDNL	6BEE	GETCH	6B7F	GNMB	6F29	GNOK	6F4E	GOT	6BF8
HBUF	700F	HDENG	0020	HER	6F72	HERE	6B8C	HLEN	700B
HOUT	6F1C	HXOUT	6F10	IADR	6BCB	IDNT	7000	IERR	6DFE
IGO	6E86	IGT1	6A02	IIO	6E21	IMC	68CA	INIT	6D57
IO	6EE2	IOPN	6EEF	IP1	6B61	IPRN	6898	IPUTCH	6A3D
IT2	6E7C	ITEM	6E67	ITEML	6E69	IWRSC	6B4A	KB	688B
KBD	0285	KBDB	6F93	KOLD	6FE9	L4	6CC9	L5	6D1B
LINE	6E92	LLEN	700C	LLOP	6E9E	LNUM	6F30	LP1	6E30
LP2	6E45	LRAM	0070	LSD	0004	MCRLF	008A	MOV	6B8C
MOVE	6BB3	N7	6EBD	N8	6EC5	N9	6ED2	NDAT	6A7A
NEQ	69AC	NFC	6B93	NFD	6D86	NFF	68D3	NONFF	6A93
NOTF	6FB8	NREADY	0080	NTEQ	6E78	NWPA	6D06	OP2	69F5
OP3	6A0F	OP4	6A42	OP5	6A57	OP6	6A90	OPNR	6917
OPNW	69CF	OUTX	6F63	P1	6E3D	P2	6E4D	PADR	6F0D
PBPT	7005	PEND	6ADF	?PENTER	687B	PMT	6FAF	PNAM	6E12
PRCR	6F01	PRDG	6E53	PRNT	6F03	PRT	6E8D	PUTCH	SAB1
RAM	7000	RDL	6CC5	RDND	6C84	RDNDL	6C86	RDNXSC	6CAE
RDSC	6CBB	REPT	6DD9	RESEEK	6DD7	RETRYC	0003	RNF	0010
RT	6945	RTN	6C8F	RTYCN1	70E4	S1	6927	SAVE	70E5
SBF	685E	SC2	6A38	SDL	6A2D	SEC	7008	SEN	6A4E
SEPS	6E7F	SK	6D9B	SKD	6DCF	SKHLD	6DAA	SKOK	6DB3
SQ2	69D5	SQF	691D	SR0	70D2	SR2	70D3	SR3	70D4
SRDY	6DBD	SRDY1	6DBF	SSEC	6CB8	START	6800	STD1	6CA2
STDR	6C92	STMP	6AEA	STR	6F9A	STRUP	6FE7	SVR1	70D5
SVR3	70D6	SYSD	70D7	TMP	70DB	TRK	7007	TRK1	70B2
TTD	0003	ULPT	70F0	UNS	6E52	VB	6864	VDU	6F79
VDUB	6F8C	W1	6DB9	W2	6DDD	W3	6DEA	WBSD	6CF9
WBSDD	6CEF	WBSY	6CF4	WDAT	6A71	WERR	6D3C	WPE	6D53
WPRT	0040	WRKD	70D8	WRND	6E01	WRNDL	6E03	WRR	6D50
WRSC	6CFB	WWAIT	6D26	XGOTO	70DD				

BUSY	0001	DUAL	0001	RETRYC	0003	TTD	0003	LSD	0004
CRC	0008	FD1771	000C	FDCMD	000C	FDTRK	000D	FDSEC	000E
FDDAT	000F	FDINT	0010	RNF	0010	HDENG	0020	WPRT	0040
LRAM	0070	NREADY	0090	MCRLF	009A	KBD	028E	COUT	02B4
START	6800	DOS	683D	DR	6854	VB	6854	SBF	585E
?PENTER	687B	ENTERK	6889	KB	688B	ENTER	6893	1PRN	5898
ERR	68AE	IMC	68CA	NFF	68D3	FCMD	68DE	FCMD1	68E3
FDIR	68FE	CCMD	68FB	?EXEC	6907	OPNR	6917	8QF	691D
S1	6927	RT	6945	ASGN	694C	AS1	6957	AS2	6951
DVN	696E	AS3	6972	DPK	697C	DPRL	6982	AS4	698C
COD	6998	CODL	699C	NEQ	69AC	ERRD	69B2	CODI	69BD
OPNW	69CF	SQ2	69D5	OP2	69F5	IGT1	6A02	OP3	6A0F
SDL	6A2D	SC2	6A38	IPUTCH	6A3D	OP4	6A42	SEN	5A4E
OP5	5A57	WDAT	6A71	NDAT	6A7A	CDIR	6A7F	OP6	6A90
NONFF	6A93	PUTCH	5AB1	PEND	6ADF	STMP	6AEA	CLSW	6AF1
IWRSC	6B4A	IP1	6B61	CLSR	6B7A	GETCH	6B7F	HERE	6B8C
NFC	6B93	CHAIN	6B96	CHAINI	6B9B	CHAINL	6BA2	MOVE	6BB3
MOV	6BBC	IADR	6BCB	CNXT	6BCE	CXIT	6BD5	DIRF	6BDF
DIR1	6BE1	GDEN	6BE8	GDNL	6BEE	GOT	6BF8	GDL	6BFC
DRE1	6C0B	FNEXT	6C0C	FIND	6C15	FMX	6C26	FOL	6C2F
FDL1	6C40	FOK	6C4D	F1	6C4F	FEX	6C5A	FKD	6C71
FEN1	6C73	FEN	6C77	FGD	6C81	RDND	6C84	RNDNL	6C85
RTN	6C8F	STDR	6C92	FIRST	6C98	STD1	6CA2	RDNXSC	6CAE
SBEC	6CB8	RDSC	6CBB	RDL	6CC5	L4	6CC9	WBSDD	6CE7
WBSY	6CF4	WBSD	6CF9	WRSC	6CFB	NWPA	6D06	L5	5D1B
WWAIT	6D25	WERR	6D3C	WRR	6D50	WPE	6D53	INI1	6D57
DRIVE	6D65	DREX	6D82	NFD	6D86	SK	6D9B	SKHLD	5DAA
SKOK	6DB3	W1	6DB9	SRDY	6DBD	SRDY1	6DBF	CERR	6DCB
SKD	6DCF	RESECK	6DD7	REPT	6DD9	W2	6DDD	W3	6DEA
1ERR	6DFE	WRND	6E01	WRNDL	6E03	PNAM	6E12	CPN	5E1B
110	6E21	DDUT	6E29	LP1	6E30	D1	6E37	P1	6E3D
LP2	6E45	P2	6E4D	UNS	6E52	PRDG	6E53	FORC	6E5B
DEC	6E5F	ITEM	6E67	ITEML	6E69	N1EQ	6E78	112	6E7C
SEPS	6E7F	IGD	6E86	PR1	6E8D	LINE	6E92	LLOP	6E9E
N7	6EBD	N8	6EC5	N9	6ED2	CRLF0	6EDA	CRLF	6EDC
10	6EE2	IOPN	6EEF	PRCR	6F01	PRNT	6F03	PADR	6F0D
HXOUT	6F10	HOUT	6F1C	GNMB	6F29	LNUM	6F30	GNOK	6F4E
DNUM	6F5E	OUTX	6F63	HER	6F72	VDU	6F79	DEVS	6F7F
VDUB	6F8C	KBDB	6F93	STR	6F9A	PMT	6FAF	CTU	6F82
NOTF	6FB8	EX	6FCA	ARE	6FD5	STRUP	6FE7	KOLD	6FE9
DSKD	6FF2	CMDS	6FF5	DFDRV	6FF8	JDNT	7000	RAM	7000
CODE	7003	PBPT	7005	DRV	7006	TRK	7007	SEC	7008
DSEC	7009	DBYT	700A	HLEN	700B	LLEN	700C	FLYP	700D
DBPT	700E	HBUF	700F	BUFF	7070	BUF	7072	TRK1	70B2
DENT	70B5	DENP	70CE	SR0	70D2	SR2	70D3	SR3	70D4
SVR1	70D5	SVR3	70D6	SYSD	70D7	WRKD	70D8	ADR	70D9
TMP	70DB	XGOTO	70DD	ERRDR	70DF	ERR0	70E0	ERR1	70E1
ERR2	70E2	ERR3	70E3	RTYCNT	70E4	SAVE	70E5	CDRV	70E6
ULPT	70F0	DATV	70FC	DATE	70FD				