

D O S T E C H N I C A L M A N U A L

Chapter 6 Disk Operating System

20/2/80

This is a 4K byte program, resident in read only memory. It provides support for other programs, e.g. high level languages, by dealing with peripheral devices and allowing data to be filed on 5.25" mini-floppy disks. The standard devices which it uses are Acorn's teletext visual display interface, a parallel ASCII keyboard and a Centronics parallel printer interface, contained on Acorn's Versatile Interface Board.

Enter the DOS by pressing the delete (DEL) key on the keyboard (and hold it, if your keyboard is not Acorn's) and the reset (BREAK) key. The system will display on the screen:-

```
Acorn Dos
```

```
* _
```

The * is the DOS prompt, indicating that it is waiting for the user to type in a command. To the right of this is the flashing cursor at which characters will appear. After a character has been typed on the keyboard, it will be displayed on the screen and the cursor will move one space to the right. Incorrect characters may be corrected by typing a delete key, in which case the character to the left of the cursor is erased and the cursor moved back one space. An entire line may be thrown away by typing control x (hold the CTRL key down and type x or X), in which case a new line will be started on the screen. When the line of text is, you think, in a satisfactory form for the DOS, typing RETURN will present it to the DOS for approval. The command will be executed (if possible) and the * prompt will usually reappear, prompting for more commands. If the DOS detects an error in the command, a complaining message will appear, e.g. :-

```
Syntax ?
```

indicates that the DOS recognises the command, but does not approve of the way it is written.

The DOS directly recognises the following commands, all of which can be abbreviated by entering enough characters to distinguish it from other commands, followed by a full stop. Spaces may be used between items (and one is usually obligatory) and leading zeroes in numbers are not required. The commands are listed together with their shortest possible abbreviations.

CAT X .

This command causes the catalogue of drive X (or the current drive, if X is not present) to appear on screen. A typical catalogue will look something like this:-

```
*CAT0
Basic disk v1 drive 0 qual s opt 0
:      £BASIC          £LISP
s:      ZOMBY
```

The title of the disk is Basic disk v1, we are currently using drive 0 and qualifier s and the disk option is 0 (no auto start features). Two files have been saved in qualifier 'space', both of which have been locked to prevent careless deletion. One file has been saved in qualifier s and this has been left unlocked. The catalogue is sorted

by qualifier and file name when it is output. The character X can be omitted or it must be 0 or 1. If not, then

Drive ?
will appear.

DIR X D.

This command causes the catalogue of the drive specified as in CAT to be loaded into memory at hex 2000. The command is often used to wait until completion of the previous operation.

DRIVE x DR.

This command sets the current drive to x, where x can be either 0 or 1 or omitted completely (for compatibility with CAT and DIR). If x is neither of these the error

Drive ?
will appear, if x is multi-character a
Syntax ?
will appear. Drive 0 is set on reset.

SETY

This command sets the current qualifier to Y, where Y can be any character. All following file access will use only the Y portion of the catalogue. Qualifier space is set on reset.

USEY

This command allows the following file operation to use the Y portion of the catalogue. After the file operation is complete, the previous qualifier will be made current again. If an error occurs in the file operation, the qualifier does not immediately revert so that the job can be repeated. To force reversion after an error, use a MON or NOMON command. Using two successive USE commands will result in the loss of the original qualifier.

Definition <s>

The symbol <s> will stand for a string of characters. If the required string does not contain any spaces and does not begin with a " quote, it may be typed directly. If not, it must be enclosed in " quotes, with " quotes in the required string typed as "".

Examples

Required string	<s> form
FRED	FRED or "FRED"
"FRED	""FRED"
"FRED"	""FRED""
"	""
hello	hello or "hello"
a b c	"a b c"
	" "

A valid <s> form can be surrounded by spaces, and has an even number of " quotes in it. When <s> is used as a filename, the string must be less than 8 characters in length, otherwise a

Name ?

message is produced.

TITLE <s> T.

This command sets the title of the disk in the current drive to the first 13 characters in <s>, filling with spaces if there are fewer than 13. It is often useful to include Form Feed (ctrl L) at the start of a title, so that catalogues appear at the top of the screen. If the entire disk is protected, a

Disk prot
message is produced.

OPTION X O.

This command sets the option of the disk in the current drive to the number X. If the entire disk is protected, a

Disk prot
message is produced. The option enables automatic use of the file BOOT in qualifier space on drive 0 when the system is reset. The automatic start may be totally defeated by pressing DELETE while the system is reset, and is enabled by pressing space. The possible modes are

option 0 : do not do anything
option 1 : load the file BOOT
option 2 : run the file BOOT
option 3 : exec the file BOOT

In option 0, the system will not mind if BOOT is not present, in the other modes, a

File ?
message will be produced on reset if BOOT does not exist.

MON M.

This command turns on a message system which writes out a file's information at every file access.

NOMON N.

This command disables messages.

LOAD <s> XXXX L.

This command loads the file <s> on the disk in the current drive from the current qualifier into memory starting at address XXXX. The address XXXX may be omitted, when the file's own address is used. If the file is not found a

File ?
message is produced.

Examples

LOAD file name is space
LOAD FRED
LOAD "FRED"
LOAD FRED 1000
LOAD "FRED" 1000

RUN <s>1 <s>2 R.

This command loads the file <s>1 on the disk in the current drive from the current qualifier into memory at the address for the file. The <s>2 is turned back into the original string form and stored in memory from 0140 upwards, terminated by a carriage return.

Examples

```
RUN FRED
RUN "FRED"
RUN FRED jim
RUN "FRED" jim
RUN "FRED" "jim 1"
DETS 2847 2647
SAVE <s> XXXX YYYY ZZZZ
```

This command saves the block of memory between XXXX (start address) and YYYY (end address plus 1) to the file <s> in the current qualifier of the directory. If entire disk is protected a

Disk prot
is produced, and if <s> is locked a

File prot
message is produced. If <s> exists and is not locked, it is deleted. Starting at the extreme outside of the disk (track 0), a gap large enough to contain the block is searched for; if it cannot be found a

Disk full
message is produced, if there are already 31 files in the catalogue a

Full
message is produced. The ZZZZ address is the execution address which defaults to XXXX if not supplied.

DELETE <s> DE.

This command deletes the file <s> in the current qualifier from the current disk's catalogue. If entire disk is protected a

Disk prot
message is produced, if the file is not found a

File ?
message is produced, if the file is protected a

File prot
message is produced.

GO XXXX

This command causes the machine code subroutine at XXXX to be entered. If XXXX is not given, the last known execution address is used. Warning : the execution address is destroyed by CAT and INFO does not set the execution address.

LOCK <s> LO.

This command locks the file <s> in the current qualifier on the current disk. If entire disk is protected a

Disk prot
message is produced, if the file is not found a

File ?
message is produced.

UNLOCK <s> U.

This command unlocks the file <s> in the current qualifier on the current disk. If entire disk is protected a

Disk prot
message is produced, if the file is not found a

File ?
message is produced.

INFO <s> I.

This command produces information about the file <s> in the current qualifier on the current disk. If the file is not found a

File ?

message is produced. The information is in the following form:-

current	file	load	execution	length	start
qualifier :	lock	FILNAME	address	address	in bytes sector

For example, the information on the files on the example catalogue could be

:	fBASIC	C000	C2B2	01000	002
:	fLISP	2800	2800	02000	012
s:	ZOMBY	3000	C2B2	00312	032

EXEC <s> E.

This command reads the bytes from the file <s> in the current qualifier on the current disk as if they came from the keyboard. If the file is not found a

File ?

message is produced. The file is automatically closed after all the bytes in it have been read. EXEC uses calls to OSFIND OSSHUT and OSBGET.

If the command is not one of the above, then it is treated as a RUN command file name in qualifier space of the disk in drive 0. This is when the <s>2 string is the most useful; assuming the existence of EDIT, EDIT "fred" is a valid command. If the command can not be found a

Command ?

is produced. With the example catalogue, valid commands are BASIC or LISP, which will be loaded and executed.

Error messages

Disk error 08 (clock error)

During a disk read operation a clock bit was missing.

Disk error 0A (late DMA)

During a disk transfer the processor did not respond fast enough. Most likely due to a faulty disk interface card.

Disk error 0C (ID field CRC error)

The cyclic redundancy check derived from the data read back differed from that stored on the disk.

Disk error 0E (data CRC error)

The cyclic redundancy check derived from the data read back differed from that stored on the disk during a disk read.

Disk error 10 (drive not ready)

During a transfer the disk stopped rotating. Generally a badly inserted disk.

Disk error 14 (track zero not found)

Controller failed to find track zero. Generally an unformatted disk.

Disk error 13 (sector not found)

Controller failed to find required sector. Either a corrupted or an unformatted disk.

FORMATTING

A format program is supplied which has the wrong execution address to prevent it from being RUN. The format file should be loaded and the disk removed !!! The disk requiring formatting should be inserted and checked with CAP. Type GO 2E00. The program prompts you with a message, and you must then type the characters Y, E and S, with no mistakes, before the formatter will operate. The program initialises the entire disk and clears the catalogue, then verifies the entire disk. If an error occurs during verification, the formatter should be tried a few more times. A protected disk will produce a

Disk prot.
message.

Technical Description

The DOS supports two single-density single-sided, (or one single-density double-sided) minifloppy drives. It uses an Intel 8271 floppy-disk controller device. Both drives must be of the same type. Each disk provides on-line storage of at least 87 kbytes; the maximum possible storage is 348 kbytes, using two TEAC FD50C drives. A maximum of 31 catalogue entries is allowed per disk surface. A file consists of a contiguous number of sectors on the disk, each sector being 256 bytes long, and there are 10 sectors per track. A soft sectored disk is used incorporating a sector stagger so that files may be read in the minimum number of disk revolutions and track steps. Retries are automatically initiated when reading a sector, and only after the tenth error will a disk error be generated. When the disk is started the catalogue is copied into memory, and this catalogue is assumed valid while the disk is still 'ready'; thus, reading successive files requires as little head movement as possible. Changes made to the catalogue cause it to be written out to the disk.

Transfer between the floppy disk controller and processor is synchronised with the NMI (non-maskable interrupt), which will occur every 64 microseconds for data transfer between memory and a disk. The data transfer interrupt uses between 48 microseconds and 56 microseconds of processor time, the average for a read being 48 and $5/256$ of a microsecond. The average for a write is 51 and $5/256$ microseconds. To achieve this speed a volatile execution block has been used in page zero, so page zero should not be loaded from disk. When a file is loaded, all sectors are loaded from a disk complete. The disk rotates six times per second, and the sector stagger is +3, so that the time to load a file of 2 tracks (5k) is about 1/2 second. The floppy disk controller automatically unloads the head after 10 revolutions have occurred with no further access requests from the processor.

Sequential Files

The DOS entry points involved with sequential files are:-

OSFIND	prepare file for sequential access
OSSHUT	close file, release buffer, tidy up
OSRDAR	read parameters of some open file
OSSTAR	update parameters of some open file
OSBGET	read byte from file
OSBPUT	write byte to file

At any one time there may be several sequential files active, in this DOS up to five. These active files will each have a one byte internal name which will be referred to as a "file handle". Handles are allocated by OSFIND, cancelled by OSSHUT and passed as arguments to all other routines. Proper file handles are all non zero: use of zero as a handle causes some of the routines to perform special functions. An open file has various status information associated with it, including:-

The sequential pointer P (called PTR by BASIC)	
The file extent	E (called EXT by BASIC)
The file region	R

The file is viewed as a row of bytes labelled 0, 1, 2, 3, The sequential pointer holds the number of the next byte to be read or updated. As OSBGET and OSBPUT access bytes of the file, they increment P, which is a three byte value. The file extent E is another three byte value which holds the number of characters stored in the file. E=0 indicates an empty file and when E=P an attempt to go further onwards will return an end of file marker and subsequently cause a 'BRK'. The region R is used when putput is sent to a file. When a new file is created a region of disk is set aside for it. The new file will have an extent of 0 and R will show the size of the disk block allocated. As bytes are written to the file E is incremented and when E=R the file is full and no further bytes may be added. R is always a multiple of the disk sector size (256 bytes) and cannot subsequently be changed (files cannot be extended). When a file is SHUT any unused sectors are released. It will always be the case that

$0 \leq P \leq E \leq R$
and R is a multiple of 256

OSRDAR and OSSTAR provide a means for interrogating P, E and R, and updating P. The ability to change P gives the user random access and update capability for sequential files. If P is set beyond the extent of a file using OSSTAR the space in the file from it's old length to it's new will be filled with hex FF bytes.

OSBPUT writes bytes to a file. On entry A holds the byte to be written and Y holds the handle. If P=R the file is full, and OSBPUT closes the file and executes a 'BRK'. Otherwise byte P of the file is updated and then, if P=E E and P are incremented else just P is incremented. In the normal case when bytes are being added to the end of the file P=E.

OSBGET reads bytes from a file. On entry Y will hold the handle. If P=E there are no more bytes in the file so OSBGET sets the carry flag and returns hex FF, a second attempt to read at end of file causes a 'BRK'. Otherwise OSBGET puts byte P of the file into A, increments P and returns with the carry flag clear.

Both OSBPUT and OSBGET behave specially if used with a file handle of zero. OSBPUT writes to the output stream using OSASCI and OSBGET reads from the input stream using OSECHO. Note that it is legal to use calls to both OSBGET and OSBPUT for a single file, but that excessive use of OSSTAR to update P may cause a lot of disk transfers.

OSSHUT closes the file whose handle is in Y. This involves writing out any buffers that contain data that has been changed, and updating the main disk catalogue to show the length of the file. A zero handle in Y will cause all sequential files to be closed.

OSFIND opens files for input or output. To call OSFIND, it is necessary to provide a block of store containing the file name (terminated by hex 0D, carriage return). Two bytes in page zero point to this block, and the X register contains the address of the pointer. If the carry flag is set the file named must already exist and E and R will be set to its actual size. If the file is not present on disk then OSFIND will return 0, this gives the user a way to detect whether a file exists or not.

If the carry flag is clear and a file with the given name already exists, the old file will be used, but with E set to zero initially. The result of this will be that the data in the old file cannot be accessed, but the region of the new file will be the same as the old. If the old file was protected, OSFIND will fail. If no old file existed a new file is created with E set to zero and R given the default value of 4096 bytes. If there is not enough room on disk, then a 'BRK' is taken. If the user needs to control the size allowed for files (for instance requiring more than the default size), then the files should be pre-allocated by using SAVE so that OSFIND does not create them. Note that file names in OSFIND are modified by the current drive and qualifier.

The region of store from \$2200 to \$27FF is used for sequential file buffers and control blocks. To reduce chance of disk corruption, the software maintains checksums on this memory, causing a 'BRK' if a check fails. In this event the safest thing to do is start from hardware reset, but in most cases it should be safe to shut files first. An unrecommended, but possibly good action would be to set byte \$00C0 to 0, which would cause files to be forgotten without changing the disk.