Disk Directory

On the screen is a disk directory produced by the CP/M utility STAT, which gives considerably more information than most directory displays

Recs
The number of records in the file can vary; here the records are 128 bytes long.

Bytes
The length of the file in Kbytes

Ext
The extent is an alternative measure of the disk space occupied by the file

Acc
Access: a file can be marked for reading and writing (R/W), or for reading only (R/O)

File Name
The full file name starts with a drive name (A: or B:), followed by the file name (AUTOST, for example), followed by the file extension (.COM, for example), which may give some information about the file's contents

```
Recs  Bytes  Ext Acc Filename
   0     0K    1 R/W B:ACNTLIST.DTA
  11     2K    1 R/W B:ANT
  10     2K    1 R/W B:ANT.BAK
  64     8K    1 R/W B:ASM.COM
  16     2K    1 R/O B:CODELIST.DTA
  16     2K    1 R/W B:AUTOST.COM
   1     1K    1 R/W B:COMPANY.DTA
   2     1K    1 R/W B:CONTROL.DTA
  34     5K    1 R/W B:COPY.COM
  40     5K    1 R/W B:DDT.COM
   7     1K    1 R/W B:DUMP.COM
 250    32K    2 R/W B:INSTALL.COM
   4     1K    1 R/W B:JUNK
  16     2K    1 R/W B:LOAD.COM
   6     1K    1 R/W B:MICROLIN
  40     5K    1 R/W B:ML.COM
  86    11K    1 R/W B:MOVCPM.COM
  58     8K    1 R/W B:PIP.COM
   2     1K    1 R/W B:SCREEN.ASM
   1     1K    1 R/W B:SCREEN.COM
   4     1K    1 R/W B:SCREEN.DOC
  42     6K    1 R/W B:STAT.COM
Bytes Remaining On B: 85K
```

binary, sequential, and random access files. We will look at each of these methods separately.

BINARY FILES

A binary file is simply a copy of a portion of user memory, a good example being a SAVEd program. Imagine the area of RAM available to the user as a simple notepad. If you were keen to preserve vital notes or interesting drawings, you would tear the relevant pages off and keep them where they were handy. Binary files work in the same way. When a SAVE command is given, the DOS stores the FILENAME on disk, marking it in some special way as a binary file and then copies the relevant area of memory byte-by-byte to the disk. The program is stored in linked blocks (with the markers at the end of each block indicating where the next block begins) until the end of the program or data is reached. The last block finishes with an end-of-file marker. Using our analogy we could say that we have named and stored a page from our notepad in a filing cabinet drawer, and added the name of the file to the contents list on the drawer.

On being RETURNed or ENTERed, a BASIC program line is compressed into a tokenised form in which the BASIC keywords are coded by the BASIC interpreter into one-byte numbers. These can be manipulated more easily and decoded back into text for LISTing purposes. As a binary file is a RAM image it can also store ASCII codes and binary data. The facility to save ASCII files is useful for storing the contents of screen memory, for example, so that screen displays can be SAVEd and later LOADed back into the same area of memory easily. In addition, some disk operating systems and BASICs allow a BASIC program to be stored in ASCII form. This enables the untokenised program to be edited as a text file in machines with sophisticated editor programs.

Binary files are very simple to use and manage, but they are limited by two factors. First of all, it is only possible to SAVE the relevant information as one continuous section of data. As a consequence, the information must be retrieved in the same manner and therefore binary files must be LOADed back into memory in their entirety. Secondly, the maximum size of a file is limited by the amount of RAM available to the user.

In the next instalment of the course we will look at sequential files, which allow a file to be as long as is required (within the limits of disk space), and random access files, which use different methods to allow the DOS to store data in such a way that it can be freely retrieved and updated.