

NOTE channelnumber,integer variable,integer variable

POINT channelnumber,integerexpression,integerexpression

Consider the following program segment. The NOTE command on line 10 stores the sector number of the file opened on channel 1 in the variable, SECTOR%. The current position in the sector is stored in SECTORPOS%:

```
10 NOTE 1%,SECTOR%,SECTORPOS%  
.  
.  
.  
90 POINT 1%,SECTOR%,SECTORPOS%
```

Suppose several disk operations have occurred between lines 10 and 90. When line 90 is executed, the disk operating system returns to the file position it was at when line 10 was executed.

```
10 OPEN "O",1%,"TEST"  
20 FOR T%=1% TO 5%  
30     IF T%=3% THEN NOTE 1%,SECTOR%,SECTORPOS%  
40     T1%=T%+5%:PUT 1%,T1%  
50 NEXT T%  
60 CLOSE 1%  
100 OPEN "R",1%,"TEST"  
110 POINT 1%,SECTOR%,SECTORPOS%  
120 GET 1%,TEST%  
130 PRINT TEST%
```

The above program creates the file named "TEST" on disk 1, and stores in it the numbers 6, 7, 8, 9, and 10. SECTOR% and SECTORPOS% will contain the information on where the third number of the file is located. Line 100 opens the file in the random mode so that we can get just the third number. The POINT command positions us at the start of the third number, and the following command gets the number and stores it in TEST%. Line 130 prints '8', the third number of the file.

With POINT and NOTE you can set up a file, and GET and PUT to it without reading or writing the whole file. The trick is to use the NOTE command for, say, every sixteenth file element, and to store the sector and sector position data in an array. Then, when you want to get to an element, you use the data in the array with a POINT command to position as close as possible to the element. Finally, you use GET commands to reach the particular data element. If you have a big file and need to access records quickly, this is one way to do it. You can even store the array in a file, and when you want to work with the main file, you first read the array from the disk. See reference manual section on NOTE for an example.

#### GET and PUT (alternate form)

There is an alternate form of GET and PUT which allows you to specify the number of bytes to be transferred. For example, GET 1%,A%(0%),50% will read 50 bytes from the file into the array A%. Since an integer is two bytes long this will read values for A%(0%) through A%(24%). This is much faster than getting the integers one at a time. PUT 1%,A(10%),20% will put A%(10%) through A%(19%) into the file. On a GET you need to be careful that the array is large enough to hold all the bytes. For instance, if A%