

INCLUDING SOFTWARE SHOPPER, YOUR SHAREWARE SOURCE  
HUNDREDS OF PROGRAMS TO TRY BEFORE YOU BUY!

For Tandy and  
MS-DOS Compatibles

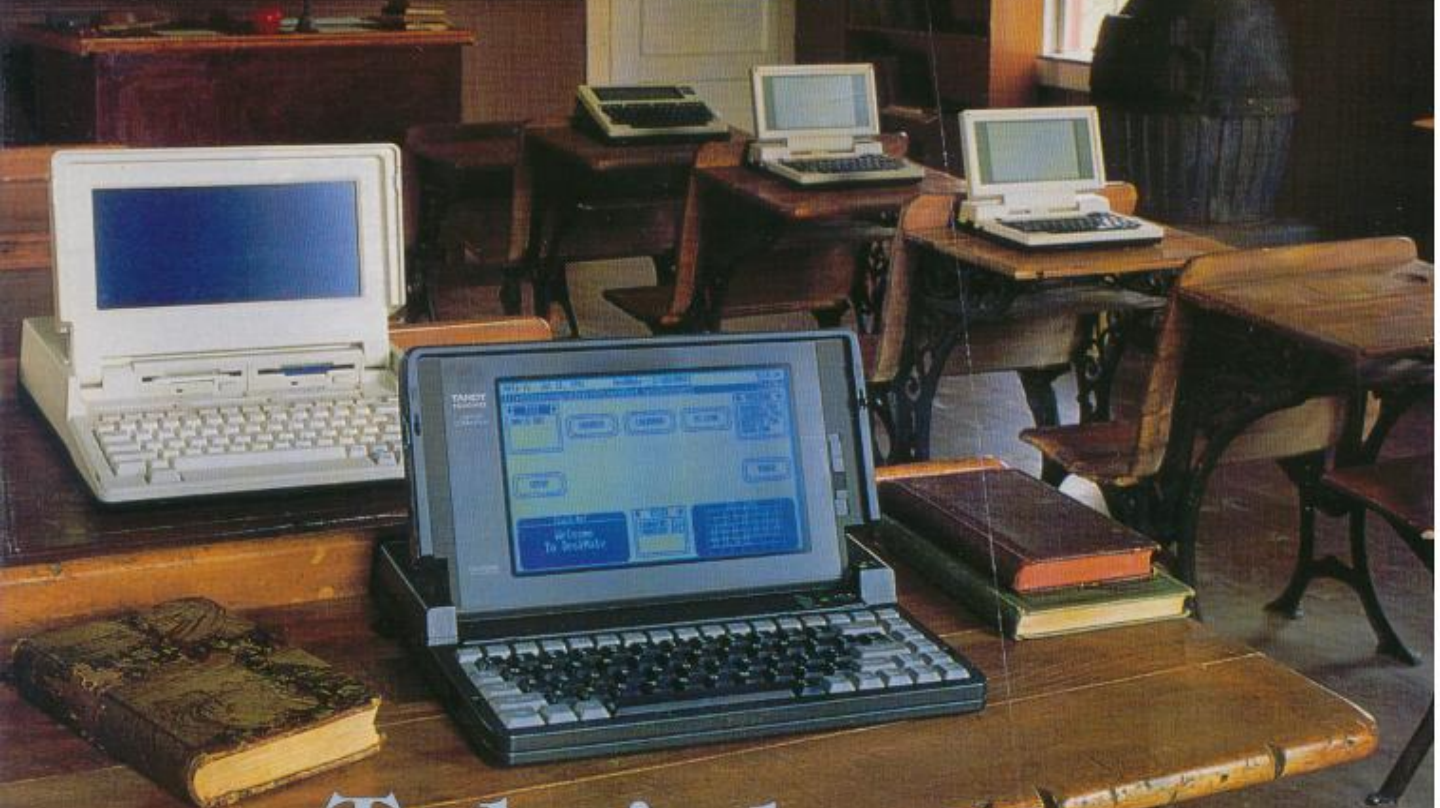
# PCM<sup>®</sup>

The Premier Personal Computer Magazine for Tandy<sup>®</sup> Computer Users

U.S. \$3  
Canada \$3.50

Vol. IX No. 3  
September 1991

# COMPUTERS — IN — EDUCATION



Today's lessons:  
*Laser Math*  
*Test 'Em*





# Digging Deep in the DOS Toolbox

by Gregory Shultz  
Technical Editor

**?** I have a Tandy 1000 TL with two 3 1/2-inch disk drives that runs under DOS 3.3. I have produced an alphanumeric file of 75 to 80 pieces of data using the following DOS command sequence: `SORT < CON > DATA.FIL`. From time to time I would like to be able to enter an additional 10 to 20 entries into this file and have them sorted and integrated into the proper sequence. What DOS command sequence accomplishes this task?

Robert A. Claudy  
Lawrenceburg, Tennessee

**!** The answer to your question can be found by digging deep into the DOS toolbox and pulling out the *redirection*, *piping* and *filter* tools. These three simple yet often overlooked tools can perform powerful operations when combined in the proper manner. I use the word *overlooked* due to the fact that these three tools have been a part of MS-DOS since Version 2.0 but remain underused because of poor coverage in many DOS manuals.

---

Greg Shultz teaches a computer-skills class in Louisville's adult education program and enjoys helping others learn about computing. He can be contacted on Delphi (username GREGSHULTZ) or on PC-Link (username PCM GREG), where he hosts the PCM Q&A forum on the second and fourth Tuesdays of the month.

Once you have a basic understanding of how these tools do their jobs, experimentation is the best way to discover how they can be used to help you perform yours. Remember: You are limited only by your knowledge of the available tools; the more you know, the further you will be able to go.

As your example application has already shown, combining these tools in an assembly-line fashion allows you to perform such complex operations as having the data you type into the computer automatically sorted and stored in a file in one fell swoop. For the benefit of those who may not know about these tools, let's take a brief look at how each one works and then see how they can be combined to create powerful commands to solve problems such as yours.

Before we can begin, you must have a basic understanding of standard input and output. Whenever you type a command into the computer, you are using the standard input device — the keyboard. As you type the command, it is shown on the standard output device — the display screen. These two devices are together called the *console*, and DOS knows this input/output combination as `CON`. Most DOS commands and application programs accept input from and direct output to the standard devices.

Using the redirection symbols `<` and `>`, we can redirect input and output to and from sources other than the standard devices. For example, we can redirect information des-

igned for our display screen to the printer or to a file. Another example is a program that normally gets its input from the keyboard

**T**he redirection, piping and filter tools have been a part of MS-DOS since Version 2.0 but remain underused because of poor coverage in many DOS manuals.

receiving it from an ASCII file via redirection. The two redirection symbols can be combined in a number of ways to perform these sample tasks, and many more are shown in Figure 1.

Very similar in function to the redirection symbols, yet different enough in operation to merit its existence, is the DOS piping



tool. This tool is represented by the | symbol. The difference is that — while redirection can be used in conjunction with files, devices and filters — piping works almost exclusively with DOS filters to provide a shorter and much more efficient method of performing redirection operations that involve filters.

You will find the filters on your DOS disk or subdirectory under the guise of the three files SORT.EXE, FIND.EXE and MORE.COM. To use them, you must make sure they are accessible to DOS (either on the same disk in floppy-disk systems or located via the path on hard-disk systems). Another point to stress is that filters work only with ASCII text.

These filters work as you would imagine. Data is put into the filter via redirection or piping on one end, the filter performs its respective task, and the altered results are output on the other end. We have already seen the SORT filter in action in your application, but piping and filters can be combined in many ways as shown in Figure 2.

On any ASCII file sent to it, the SORT filter performs its sorting operations by analyzing each line in the ASCII file and then sorting it according to the ASCII values of the characters. By default SORT uses the first character on each line as the sort key; by adding switches to the command line, however, you can alter the character used as the sort key and even perform a reverse sort. Examples of the SORT filter in use are shown in Figure 3.

The FIND filter is used to find text strings contained in an ASCII text file. In addition to working only with ASCII files, three other guidelines must be followed for this filter to function. First, FIND is case-sensitive, meaning that it sees DATA and data as two different strings. Second, the text string being searched for must be contained in quotation marks ("DATA"). Third, wild cards cannot be used in naming the file to be searched. Like the SORT filter, FIND's operation can be modified through the use of switches as shown in Figure 4.

The MORE filter works similarly to the DIR/P command; it shows a file, rather than a directory, one page at a time and prompts the user to press a key to continue viewing. While the other filters can perform numerous tasks and be altered with switches, MORE performs only one primary task and works the same way with both piping and redirection. Examples of the MORE filter in use are shown in Figure 5.

Now let's return to your question and see how we can solve the problem of adding and sorting new entries into an existing file employing the tools we have learned. First of all, your initial command sequence can be shortened from SORT < CON > DATA.FIL

to SORT > DATA.FIL. The word CON is actually redundant since input is coming from the default input device — the keyboard. Input is terminated by pressing the F6 function key, which closes and saves the sorted file.

Using a variation of the command sequence shown above allows us to put additional entries into the file. The command sequence SORT >> DATA.FIL takes input from the keyboard, sorts it, and appends it to the existing DATA.FIL file. While the new input will be sorted and appended to the existing file, the new entries will not be

**To speed up the replicating of 720K disks and avoid disk elbow, you can use VDISK.SYS and XCOPY.EXE rather than DISKCOPY.**

integrated into the existing file's sorted order. The command sequence SORT < DATA.FIL > DATA.FIL will take care of that requirement.

Of course you can perform this task in a number of ways, using various combinations of redirection and piping. Creating a batch file that contains the two command sequences shown above simplifies the process of adding and sorting the new entries.

We have only seen one application of these DOS tools, but many combinations are possible. As I've said before, experimentation is the best way to learn how to use these tools. Figure 6 shows a few more applications using them. If you come up with a command sequence using these tools that you think is innovative, send them to me here at PCM and I will put them in my column for all to see.

**Q** I am looking for some tips on using the DOS DISKCOPY command. I have a Tandy 2500 XL with 4 Meg of RAM, a 1.44-Meg 3 1/2-inch disk drive, and a 40-Meg hard disk. Of course my trusty old 1000 sits right next to the 2500, and I use both of them extensively.

My problem stems from the frustration I feel when I must continually replace the source and target disks during a DISKCOPY

operation. I am replicating a single 720K disk 24 times for use as example disks for a computer class I am teaching. Since Drive A of the 2500 is the only 3 1/2-inch drive I have between my two computers, using DISKCOPY A: A: is my only recourse.

I am saving for a CD-ROM and so do not want to add another 3 1/2-inch disk drive to the 2500. Having read in my DOS manual that DISKCOPY uses all available memory to perform its task, I have unloaded everything else possible from memory. Nevertheless, the operation is still time-consuming and involves several disk swaps. Is there a trick I am missing, or am I doomed to suffer disk elbow from removing and inserting disks?

George Helm  
Atlanta

**!** To speed up the replicating of 720K disks and avoid disk elbow, you can use VDISK.SYS and XCOPY.EXE rather than DISKCOPY. Setting up a RAM disk with VDISK and using XCOPY to transfer the files decreases the amount of time involved, since it takes advantage of the RAM disk's speed and the XCOPY command's efficiency. The only disadvantage of using these commands instead of DISKCOPY is if the target disks are unformatted; DISKCOPY formats disks as it copies.

The first step involves adding the line DEVICE = C:\DOS\VDISK.SYS 720 512 / E to your CONFIG.SYS file. (If VDISK.SYS is not in the \DOS subdirectory, include the appropriate path.) This creates a 720K RAM disk with 512-byte sectors in extended memory and assigns it as Drive D. You must reboot your computer for this to take effect.

Next, insert the single source disk into Drive A and type XCOPY A:\\*.\* D: /S/V. Using the /S switch copies all the source disk's subdirectories (as well as the files contained in them) to the target disk. If the source disk also contains empty subdirectories that you want to copy to the target disk, include the /E switch along with the /S switch. The /V switch verifies that the files are copied correctly. Now making copies of your example disk simply involves reversing the drive specifiers in the previous command (XCOPY D:\\*.\* A: /S/V).

You can work around the disadvantage I mentioned by creating a batch file that includes the FORMAT command followed by the XCOPY command. Remember that to format a 720K disk in the 2500's 1.44-Meg drive, you must use the appropriate switches with the FORMAT command. Use FORMAT A: /N:9 for DOS 3.3 and FORMAT A: /F:720 for DOS 4. Use the /S switch as well if you want these to be bootable system disks. □