

One user analyzes if Apparat's latest evolution is worth the money.

NEWDOS/80

*Ken Jackman
West Chester State College
West Chester, PA 19380*

NEWDOS/80 retains all of the functions of NEWDOS+, so users who have become accustomed to the latter can certainly use the new system. Some of the programs have been revised, and several new programs or functions have been added. However, the major changes lie in new sophisticated file formats and methods. One other aspect of the program appears to be for the sophisticated programmer: the price! One hundred and fifty dollars does require some justification.

Briefly, for those not familiar with NEWDOS or NEWDOS+, these systems cleaned up a lot of the bugs in Tandy's DOS. In addition, they provided utility programs that worked! You could junk Radio Shack's Tape-Disk, because LMOFFSET did

the same job, but did it right! A directory check program was included which could warn the unwary TRSDOS user of impending crashes due to directory overwrites—the first really good preventive diagnostic routine I've seen. A disassembler and editor/assembler were included for the Assembly language programmer, and for the Level I user, two programs permitted the use of Level I terms in Disk Basic, and storage on the disk. Superzap, the Basic language disk editor was really the first disk editor. Finally, Basic was enhanced with its own renumbering utility and a utility to list references to line numbers and to variables used in the program.

DOS Commands

NEWDOS/80 includes all of the functions of NEWDOS and NEWDOS+ though some of them have been changed somewhat. Let me list the additional functions in the DOS:

- Break permits enabling or disabling the Break key from the keyboard.

- Chain, like Randy Cook's VTOS command, permits building a file of commands which can be executed by calling the file from DOS.

- HIMEM sets memory size from DOS.

- MDBORT, MDCOPY, MDRET are commands which can be executed from Minidos.

MDBORT returns the user from Minidos to DOS.

MDCOPY permits a restricted file copy function.

MDRET returns the user from minidos to the main program. Program execution picks up where it left off.

- PDRIVE lets the user specify the characteristics of each disk drive. This command permits the programmer to mix 50 and 77 track drives and to mix five-inch disks with eight-inch disks. It even permits the use of the OMIKRON interface with eight-inch drives. Plans to pro-

vide for utilizing the LOBO interface have evidently been dropped. Although provision is made for several different drive types, only a few have actually been implemented. Space is reserved for the additional entries.

- System doesn't load machine language tapes! It does a mini SYSGEN, similar to those done on much larger systems. Passwords may be enabled or disabled; the system can be set to run only; and the screen-print option Debug, and entry to the Minidos can be enabled or disabled. This function can tell DOS if lowercase mods have been installed, enable or disable the Clear key, and perform a host of other useful but not so mind-boggling functions.

- Purge, as in VTOS, permits a review of the entire disk contents, file by file, during which any of the files may be killed.

Enhanced Commands

How about enhancements to

existing commands? Many of the commands which were improved from TRSDOS get additional improvement in NEW-DOS/80.

Careful use of the System command can enable or disable many of the following commands.

Minidos

First, we should look at the Minidos. It can be called by pressing D, F and G simultaneously. In Minidos you can execute all of the DOS commands except Append, Chain, Copy, Format, and the commands, System and PDrive. Only DOS commands can be executed. For instance, you cannot load a new program from Minidos.

Minidos has the advantage of permitting the operator to execute most DOS commands in the middle of a program, and then return to the program in progress. In other words, variables, string space, and so forth are not cleared when a program is interrupted. Minidos lets the operator decide whether to return to DOS (MDBORT) or to resume execution of the program (MDRET).

The Copy command was fairly complex in NEWDOS+. It permitted copying files from alien systems like TRSDOS. The more elaborate Copy commands in NEWDOS/80 permit copying by file, using the entire memory as a copy buffer, specifying the number of tracks to be used, and bypassing the Format function. It also offers several other options which are useful in recovering lost files.

Format has some of the same options as Copy. One can still specify the number of tracks, whether to overwrite old data, and which dates, names, and passwords to use. Since no file copying is done, the parameters controlling file copying do not make sense in the Format command.

Debug can be started in the usual ways. It can also be started by pressing the numbers —1, 2, and 3 simultaneously.

List now permits line numbers to be specified. This has always been available in Basic, but now it is available in the

DOS command as well.

Enhancements to Basic are a bit more nebulous. The file development system receives special treatment in an appendix in the NEWDOS/80 manual. An operator can no longer delete a line by simply typing the line number followed by a carriage return. Deletion of a line requires the use of the Delete command. (Delete, Edit, List, and Auto can still be initiated using only the first letter.) Scrolling commands appear to be unchanged from NEWDOS+. Period, down-arrow, up-arrow, semicolon, slash and comma still serve scrolling/editing functions. The Delete commands have been increased. In addition to the ability to delete a line, it is possible to move a line to a new location (DI) or to make a copy of the line at a new location (DU).

Of course, renumbering and referencing capabilities are retained. REF can be used to list all references to a given number or variable. The listing can be shown on the video or listed on a line printer. The renumbering function permits renumbering lines of Basic within the program and changing the starting line number and the increment for line numbers. Careful use of this capability permits moving blocks of program from one place to another.

CMD Command

The CMD command has been modified. NEWDOS has always permitted the execution of DOS commands through the CMD command. In NEWDOS/80, the return is to Basic after a CMD call unless the call was one of the one-letter calls in TRSDOS, or 'S=---' or 'F=---' is specified. In the case of the 'S' command, return is to DOS. The 'F' command can clear the indices of For...Next loops, either selectively or all at once. Returns from GOSUBs can also be cleared. This variation on the CMD command is clearly one for advanced programmers, enabling them to get out of loops and complex portions of the program without leaving confusing Nexts or Returns on the stacks.

A Sophisticated System

How does the operating system look so far? The computer novice (someone who can use TRSDOS) could use the system. Some of the enhancements probably would not get used. There are enough "bells and whistles" in the system that even an advanced programmer will not use all of them regularly. Some of the commands look like they are imitations of VTOS. The Chain command is a good example of an aspect of VTOS that is worth imitating. Other desirable aspects are missing. The ability to redirect output from one device to another, or to a file would be handy. Autorepeat on the keyboard shouldn't be too hard to implement either. I suppose that a creative systems programmer (or some hobby buff with time on his hands) could use Superzap to install facilities for these functions. Maybe Apparat will provide zaps for them. At any rate, they are desirable functions not included in this DOS.

The Sysgen functions (System and PDrive) are a major step toward a truly sophisticated operating system. The absence of a sophisticated operating system is one of the reasons why the home micros are still regarded as impractical toys by many computer professionals. In short, at this point, the DOS is a major improvement on the older systems, but there is still room for growth.

Utilities

Any major DOS provides a set of utilities for manipulating files, cleaning them up, testing the system, etc. What does NEWDOS/80 have?

NEWDOS+ and NEWDOS/80 both have the DirCheck program that I mentioned earlier. DirCheck analyzes problems in locked out granules, granules which are allocated to more than one file, errors in file entry tables and GAT tables. DirCheck should be run after any and every intensive use of a disk. If there are reasons to suspect a disk has been zapped magnetically or electrically, the

diagnostic should be run. This program does not repair the damage—it simply lets you know where it is!

The new version of SUPERZAP has more commands, and is written in machine language so it runs faster. It can read or write to any unprotected sector of the disk. Any program with that kind of power is also potentially dangerous, so the novice should practice with disks he can afford to clobber before undertaking major modifications.

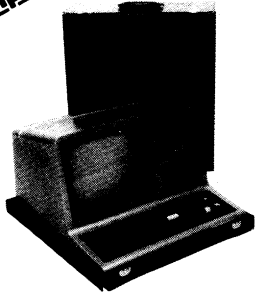
Other utilities present in both versions, as indicated before, are LMOFFSET to permit loading of machine language tapes to disk; the debugger, of course; the disassembler which permits disassembly of either main memory or a file on disk; and the Editor/Assembler. It is worth noting that older versions of NEWDOS required the user to have a copy of TRSDOS and Radio Shack's Editor/Assembler package. This is no longer the case for the DOS but remains the case for the Editor/Assembler.

Other utilities on both systems permit running Level I programs and loading or saving them on disk. Level I data files are not accessible from these programs.

Extras

In addition to the utilities and files on NEWDOS+, NEWDOS/80 has a couple of extras. ASPOOL, according to Apparat, was added at the last minute "as a free program to NEWDOS/80 owners." The program was written by H. S. Gentry, according to Apparat, but no references are given to where (or if) it was published. This spooler is not a very elegant system, but it does work. It will let the keyboard take precedence over the printer in user-defined intervals, provided the keyboard is active. That is, if the keyboard is inactive for a user defined period of time, the spooler will print a line. As long as the keyboard is not active, the spooler will continue printing. As nearly as I can tell, the main program does not run while the spooler is printing. Rather, the spooler "times out" after each

**Computer
Case
Company**



**COMP
CASE**

• RS204

Attaché style cases for carrying and protecting a complete computer set-up. Constructed of the highest quality luggage material with saddle stitching. Will accommodate equipment in a fully operational configuration along with manuals, working papers and disks. Never a need to remove equipment from case. Simply remove lid, connect power and operate. Lid can be replaced and locked for security and protection without disconnecting cables. Fully tested.

• RS201	TRS-80 Model I, Expansion Unit & Drives ..	\$109
• RS202	TRS-80 Monitor or TV set	84
• RS204	TRS-80 Model III	129
• RS205	Radio Shack Color Computer	89
• P401	Paper Tiger 440/445/460	99
• P402	Line Printer II/IV	89
• P403	Epson MX70 or MX80	89
• P404	Epson MX100	99
• CC90	Matching Attaché Case	75

computer case company ✓ 199

5650 INDIAN MOUND CT. COLUMBUS, OHIO 43213 (614) 868-9464



keyboard input, and during the time-out, the main program can run.

If the keyboard is continually active, the spooler is effectively locked out and does not print. If the spooler uses the interrupt (an option), it operates whenever the keyboard is not active, i.e. between key strokes. If it starts to print, key entry must wait until the end of the line. The spooler does permit a number of options, including serial or parallel printers, a way to inject an interrupt routine for the spooler, and a circular buffer. I used it in writing this article, but that certainly does not constitute a thorough test. Perhaps it works better than it appears.

The other new utility program is the lowercase conversion program. Written by Tom Price, this is another program without references. It is a machine-code program that relocates itself into high memory, and permits shift 0 to set or reset lowercase. It provides the software to accommodate lowercase hardware mods.

Other programs included are tutorial in nature. SAMPLE01 is a mixed tutorial/program on the new filetypes. It is designed to be listed as well as run. The REM statements provide an explanation of what is going on while the program is running. It is designed to be listed on a line printer, though I suppose the screen would work in a pinch. It is a novel approach, but it didn't work all that well for me.

The second program in this set is set up the same way, except that the program is run after you have studied the REM statements. Chaintst explains the chaining command and demonstrates its capabilities. The demo is impressive, though the explanation in the REM statements is not all that clear. The technique is slightly more complex than that used in VTOS. This version can specify section IDs and execution for only one section of the chain file. It is possible for one chain file to call another, or for one section of the file to call another section, so that very complex chaining sequences are possible.

File Types

What are these new file types that threaten to revolutionize microcomputing? Apparat has added five new file types! The old sequential and random files still work, so NEWDOS/80 can use seven different file types! All of these new file types have variable record lengths. There are basically two new types: marked-item files with three subtypes, and fixed-item files with two subtypes. The major difference between them is that the marked-item files use the first bytes of each item to describe it while fixed-item files have no such marker. Also, marked-item files of variable record lengths use the first byte of the record to mark the start of the record.

All of the marked-file types enable you to keep track of where you are in the file and in the record. This is done by requiring extra variables in the OPEN, PUT or GET statements. Unfortunately, they have had to use the same symbols used for definition of variable types (!, #, \$, and %). This adds something to the confusion. However, using the LOC command, it is possible to determine exactly where you are within a file. Use of the file location symbols in PUT or GET statements permit reprocessing the same record or same byte within a record, as well as moving to another record (or another byte) at will. Building an index file using the file location capabilities provides for true "indexed sequential" access, as well as other indexed file methods.

The fixed-item type files are probably the easiest to describe, although the manual describes them as the hardest to use. One type is apparently the same as the random files in TRSDOS. It has a fixed record length, defined in the OPEN statement, and division into subrecords is essentially the programmer's problem. The other has no record segmentation and is treated as a string of items. Again, the programmer must keep track of subdivisions or item lengths.

In addition to the documen-

**GO
PIGGYBACK!**



CENTRONICS

**ADD lowercase with our
PLUG-IN piggyback board!**

9WX7H Dot Matrix \$135

5WX7H Dot Matrix \$ 95

TWO complete character sets on board:

96 character ASCII **PLUS** choice of
128 character APL, TRS-80/H-19 Graphics or
Scientific. (Customer defined: add \$50/set).

Most printers convertible: specify logic board #



SYDNEY SEZ:
Shop Comparatively!

DSE ✓368 **Digital Systems Engineering**

12503 King's Lake Drive, Reston VA 22091 (703) 620-2994

ALSO AVAILABLE: New Centronics and Integral Data Systems printers at 15-20% below list; also, used printers of several makes. **Call for information!**

Mastercard/VISA, Check, MO, PO

All products warranted 90 days

tation in the manual proper, Apparat has included a large appendix to the manual that provides an expanded illustration of the file types and how they work. This appendix is probably the clearest part of the manual as far as file manipulation is concerned. Apparat says it was written by one of their Denver customers, but the author's name is not mentioned. Whoever it was certainly deserves credit. The appendix requires building some files and examining them with Superzap. It would be easier to follow had they included the output from Superzap in the appendix. Altogether, the manual, the appendix, and the SAMPLE01 program provide a reasonable introduction to the file types, but I wish they had included a brief discussion of the purposes for which each type could be used.

Documentation

Apparat assumes that you have the TRSDOS manual and other appropriate documentation (i.e., the Editor/Assembler manual and software). The style of writing is not quite so terse as it was in the NEWDOS+ manual, although it still assumes considerable sophistication on the part of the user. It definitely is not on the same level as the Level I manual! The introductory section describes the process of duplicating the disk, but warns against trying to use the system before studying the manual! It is advice that should be taken seriously! The values used in the SYSTEM and PDRIVE commands can set up ways to overwrite the system disk!

The section on the DOS commands includes all of the library commands, rather than just the new or modified commands, as was the case in the old manual. Most of the manual is what you would expect it to be: discussion of special features, enhancements, and minimum system configuration. They have included a glossary and a section on how to apply patches (zaps) to files on the disk. A significant part of setting up and running involves

application of the patches, which are listed and explained. They include patches to some non-NEWDOS files, such as Scripsit and APL/80, for those users who have them. Most of the patches on my disk had already been installed before I got it, but you still have to check it the first time. The patching process takes about half an hour the first time you do it.

Basic

NEWDOS/80 is a programmer's DOS, as opposed to a simple file-handler like Tandy's DOS. For the Basic programmer, the REF function and renumbering capabilities are nearly indispensable. The ability to shuffle lines about or to duplicate them (DI and DU commands) is a significant improvement in the programmer's toolkit. These functions are, I think, the most obviously useful functions to a Basic programmer. There are other, less obvious, functions that are also extremely valuable. For instance, I can best analyze a program when I can get my hands on it. The JKL function lists the video screen so I can get at it. If I'm in the process of debugging a program, and I need to know whether my files in the program match the files on the disk, I press DFG to get into MINIDOS, and list the Directory. If the files don't match, I can Rename the one in the Directory (not a good practice if you've already opened the file in your Basic program). MDRET will return me to the point at which I left off debugging. The more I use the system, the more I appreciate the MINIDOS.

Another benefit is the CMD "F=---" command. Consider the following program segment:

```

100  GOSUB 500
      -----
500  FOR I = 1 TO 100
      -----
550  IF I = 50 GO TO 600
560  NEXT I
570  RETURN
      -----
600  'PROGRAM RESUMES HERE

```

This demonstrates the CMD

"F=---" function. When the program resumes at 600, the first NEXT I statement encountered should send it back to line 500. If not, some error message will be given because there is still the equivalent of 50 I's on the Basic index stack. CMD "F=POPS" cleans up the index stack. The next RETURN statement will send the program back to 100 (or if there is another GOSUB, it will treat them as nested subroutines). CMD "F=POPR" clears both the index stack and the subroutine stack.

While CMD "F=---" should not be necessary in a well written program, it can be very useful during the debugging phase of program development. Attaching the name of the indexing variable clears the index for that variable only, so that in debugging nested loops, you can selectively clear index stacks.

The new file types are an obvious change from the usual Basic file system. They appear to be most useful in business applications such as data base management. They certainly make understanding the file easier when using Superzap. MU files are especially useful during program development, because the pointers tell you the type of data (character, integer, single or double precision) and the length of the data. It does take some getting used to, but it's worth it.

Assembly Language Programming

NEWDOS/80 includes a modified version of Tandy's Editor/Assembler. Only Apparat's modifications are documented, but an assembly language programmer will find all he needs in NEWDOS/80. In addition to the Editor/Assembler, there is the debugger, LMOFFSET for relocating code, and the disassembler. Superzap is, of course, the valuable disk editor for both the assembly and Basic languages. Further, most of the DOS entry points are included in a separate section of the manual, so that for functions like reading and writing to

the disk, the programmer has a guide to follow. Overall, everything required for intelligent programming in assembly language is present on the disk.

Conclusion

What are the pros and cons of the system? Is it really for the more sophisticated user? Who can use it and for what? First, the system is a major improvement on existing operating systems. It is a big step closer to a "real" operating system. It can be used by anyone who uses Radio Shack's TRSDOS, but it is probably too expensive to justify use in a "load and go" situation, where the user buys programs, loads and runs them. Anyone who does their own programming, particularly where they are writing programs for manipulating files, will find this DOS useful. The "serious" hobbyist who is interested in increasing his knowledge and mastery of computer science will find this system useful. There is still room for improvement. It is not a full fledged DOS—yet! For the sophisticated user, there are some desirable commands not implemented. The interrupt handling still leaves a great deal to be desired, and certain file handling techniques have yet to be implemented. (I'm not saying what they are! Look at the OS for a DEC or other large minicomputer system to find them!)

The small business user could use this system to great advantage. I think it will greatly reduce run-time for programs that do a lot of file accessing. It does require some reprogramming to implement the new file types.

Is it worth the cost? If you feel that your investment in your computer, interface and disk drive has been worthwhile, then the price of NEWDOS/80 is reasonable because of the additional capabilities it offers. The only user to whom I would not recommend this system is one who uses the computer only for game playing. He should spend his bucks on Scott Adams' next creation! ■