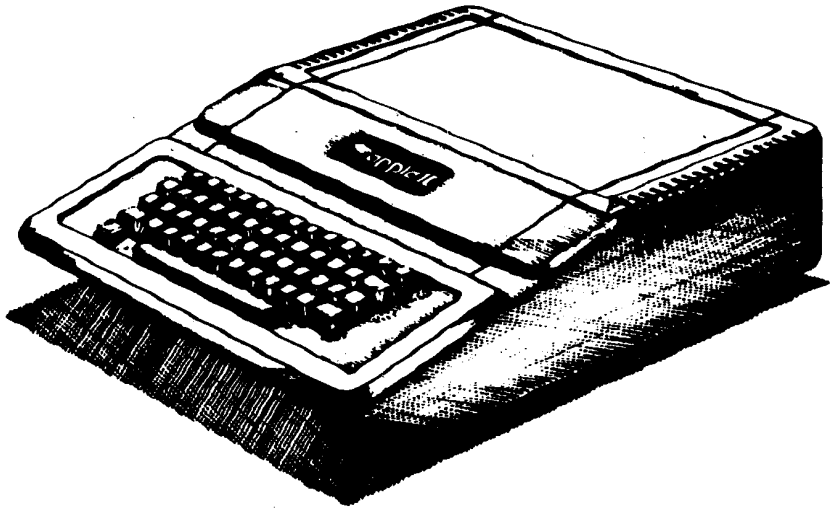
Apple 2 Computer Technical Information



Apple][Computer Family Information

ProDOS Info:
Review BYTE Feb 1984

Document # 300

Ex Libris David T. Craig

"DTCA2DOC-300-00.PICT" 137 KB 2001-04-03 dpi: 300h x 300v pix: 2034h x 2850v

Software Review

ProDOS

by Rob Moore

Although Apple Computer's DOS 3.3 operating system has more software available for it than any other personal computer operating system, it does have some minor problems. Originally designed as an operating system for small floppy disks, DOS 3.3 doesn't easily support 8-inch floppy disks or the increasingly popular Winchester-technology hard-disk drives. Typically, vendors of larger disk drives have resorted to supplying DOS patches or special modified versions of DOS 3.3. Due to its heritage, DOS 3.3 also doesn't support very large files, or interrupts, which are necessary for local networking and other more advanced applications.

To remove these limitations, Apple is releasing ProDOS, a totally new operating system for Apple II and IIe computers. According to Bill Schjelderup, product marketing manager at Apple Computer Inc., ProDOS is "designed to overcome the limitations inherent in DOS 3.3 and provide a significantly improved base for applications software development." Although ProDOS doesn't make DOS 3.3 obsolete, Apple expects that most new applications will be written under ProDOS because of the numerous advantages it offers.

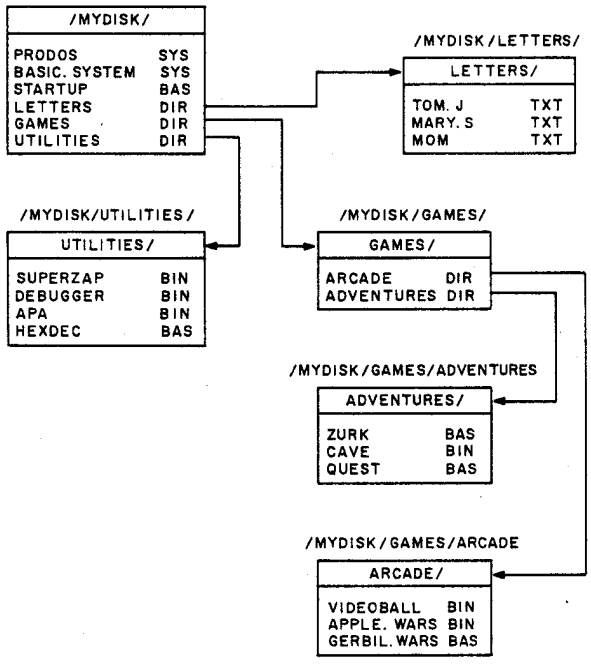


Figure 1: *A typical ProDOS directory structure. The pathname, shown above each directory listing, describes the path followed to reach a particular directory from the volume name on down. If the volume specified is mounted anywhere in the system, ProDOS will find it regardless of which drive it's in.*

New Features

ProDOS provides a variety of new features and greatly enhanced performance over DOS 3.3. Some of the new ProDOS features include

- support for Apple's Profile 5-megabyte Winchester hard-disk drive on an Apple II or IIe
- automatic time and date stamping through built-in drivers for the Thunderclock clock/calendar card (available from Thunderware Inc., 44 Hermosa Ave., Oakland, CA 94618, (415) 652-1737) or through user-installed drivers for other clock/calendar cards
- Unix-like nested directory structures and file types that are compatible with the Apple III SOS operating system so that disks can be interchanged between the two machines
- file sizes that can range from 1 byte to 16 megabytes, and the ability to randomly access any type of file
- up to 256 different types of files, including a number of reserved user-defined types
- support for up to four interrupting devices through user-installed interrupt handlers
- a uniform machine-language interface that lets assembly-language programs easily access and use all the ProDOS features
- use of Apple's 64K-byte extended 80-column text card as a high-speed pseudodisk for ultrafast file accesses

For the average user who programs primarily in BASIC, ProDOS adds a number of new commands and increases the flexibility of many existing DOS 3.3-type commands. In fact, the DOS command structure is *extensible*—ProDOS includes provisions for additional user-added commands to provide special features or allow customization for a particular application.

ProDOS is also much friendlier to outside peripheral vendors than DOS 3.3. ProDOS will automatically recognize and use other disk devices of all sizes, providing they are designed to follow the interface-protocol guidelines described in the *ProDOS Technical Reference Manual*.

For software developers, it will now be possible to

create one disk that will load on either the Apple II or Apple III. The boot loader, common to both ProDOS and SOS, recognizes the type of machine it boots on and attempts to load SOS into an Apple III or ProDOS into an Apple II or IIe. All you have to do is include both operating systems and the necessary files on the same disk.

Volumes, Directories, and Pathnames

A number of differences are found between ProDOS and existing operating systems like Apple's DOS 3.3 or Digital Research's CP/M. Where DOS 3.3 or CP/M specify disks by their *physical* drive location (e.g., S6,D1 for DOS 3.3 or B: for CP/M), ProDOS uses volume names to specify disks rather than the drive locations. Under DOS 3.3 or CP/M it's easy, for example, to delete the wrong file or initialize the wrong disk by inadvertently inserting the disk in the wrong drive. With ProDOS, errors like these don't occur. If ProDOS can't find the specified volume in a particular drive, it searches through all the drives attached to the system until the volume is found or the last drive is searched.

To help keep disk storage organized, ProDOS uses a Unix-like system of nested directories known as a hierarchical directory structure. In addition to having a main directory on each disk (or volume), ProDOS also allows subdirectory files within the main directory. Each subdirectory can hold files of any type, including further subdirectories. This nested directory structure makes it easy to keep large amounts of disk storage organized. (Figure 1 shows an example of a typical ProDOS user's disk directory structure.)

To specify which directory is accessed at any given time, ProDOS uses pathnames. A pathname describes the path to follow through the various levels of directories until you reach the directory where your program resides. For example, if you wanted to run a program called Videowars in a subdirectory called Games, which was in the main directory of a volume called Mydisk, you could type

```
RUN /Mydisk/Games/Videowars
```

To avoid having to retype the entire pathname every time you access a particular directory, ProDOS lets you set up a pathname prefix, which specifies a default directory. If you enter a pathname without a leading slash, it is automatically appended to the path stored in the pathname prefix, and the result is used as the actual pathname. Using the previous example, we could set the pathname prefix to /Mydisk/Games/ and then simply type RUN Videowars.

ProDOS's nested directories and pathname facility add some unique capabilities to your software. Programs can now keep their own directories and file types, avoiding some of the confusion encountered when all the programs on a disk are lumped into a single large directory. On a large hard disk, you can easily partition the disk into logical areas for various types of programs and data

At a Glance

Name
ProDOS

Type
Disk operating system software

Manufacturer
Apple Computer Inc.
20525 Mariani Ave.
Cupertino, CA 95014
(408) 996-1010

Hardware Required
Apple II or IIe computer with a minimum of 64K bytes of memory for operation with Applesoft BASIC; supports 128K-byte memory configuration with Apple's extended 80-column text card

Description
Single-user, single-task operating system with significant performance improvements over Apple's DOS 3.3; includes hierarchical directory structures, numerous predefined and user-definable file types, directory time and date stamping, support for up to four interrupts, file sizes up to 16 megabytes, and block-oriented disk I/O; ProDOS disks are compatible with Apple III SOS disks to allow easy file sharing between the two machines

Special Features
A 64K-byte column adapter card in the Apple IIe auxiliary slot can be used as a pseudodisk to provide significantly faster file accesses; ProDOS provides support for Apple's Profile 5-megabyte hard disk with the Apple IIe interface card

Software
Software utilities available for ProDOS include Filer, a program to initialize and copy whole disks or individual files; Convert, used to transfer files to or from DOS 3.3 disks; Exerciser, to access ProDOS machine-language calls and functions; BASIC.SYSTEM, a DOS command interpreter and user interface for Applesoft BASIC; and Help, which adds a series of help screens to the BASIC user interface; a BASIC Programming Examples disk can be used along with the ProDOS tutorial material in the documentation

Documentation
ProDOS User's Manual provides a tutorial on the Filer and Convert programs; **ProDOS Technical Reference Manual** describes internal organization, machine-language calls, organization of BASIC.SYSTEM, and how to interface additional device drivers and interrupt routines to the ProDOS system; **BASIC Programming with ProDOS** provides tutorial material and describes how to use ProDOS features and files from programs written in Applesoft BASIC

ProDOS Packages
ProDOS User's Manual with /Utilities disk including Filer and Convert
BASIC Programming with ProDOS with BASIC/Examples disk
ProDOS Technical Reference Manual with the Exerciser program on disk
ProDOS Tool Kit package includes 6502 macro assembler, text editor, and debugger with **6502 Assembler/ProDOS Tools Manual**
(The above packages were tentative at the time this article was written.)

Price and Availability
The price was not available at the time of writing; it is due to be released in early 1984 and shipped with new Disk II systems after that

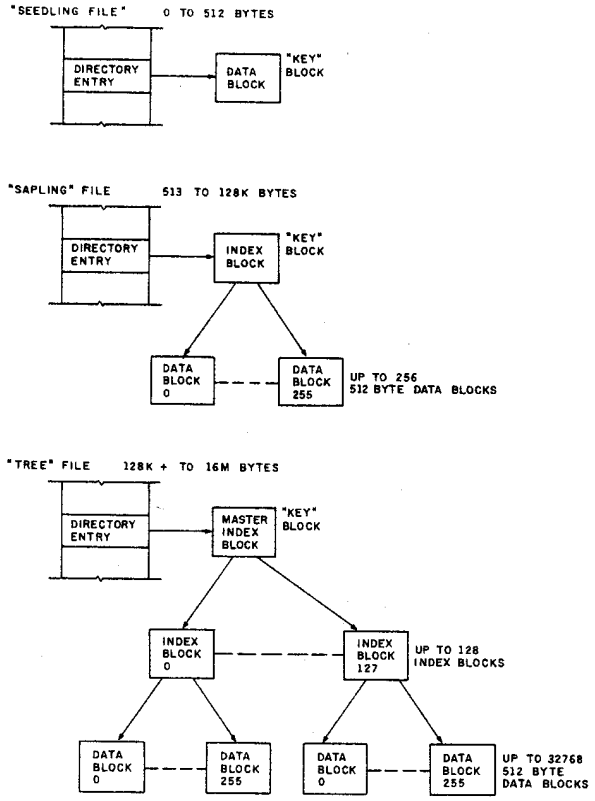


Figure 2: *ProDOS file-storage methods. ProDOS uses a growing "tree" structure to provide efficient storage of files that can range from 1 byte to 16 megabytes. Files smaller than 513 bytes are stored in a single data block and are referred to as "seedling" files. "Sapling" files have an index block that holds block numbers of up to 256 data blocks, thus providing storage for up to 64K bytes. The largest files, stored as "tree" files, use a master index block to designate up to 128 index blocks, each of which can point to 256 data blocks.*

As files grow or shrink during system operation, the file-storage method is automatically changed as the file size crosses the boundaries between "seedling," "sapling," and "tree" sizes.

files without having to resort to such tricks as making the hard disk look like a number of separate, fixed-size floppy disks.

Blocks and Files

Rather than dealing with physical disk tracks and sectors like DOS 3.3 does, ProDOS reads and writes 512-byte blocks. The conversion from disk sectors to blocks is handled by each disk's driver routines. For example, if ProDOS were reading a block from a disk that stored data in 128-byte sectors, the disk driver would read four sectors and supply the data to ProDOS as a single 512-byte block. Since ProDOS simply supplies a block number to the disk driver, it is completely independent of the physical disk-sector sizes or the number of sectors per disk track.

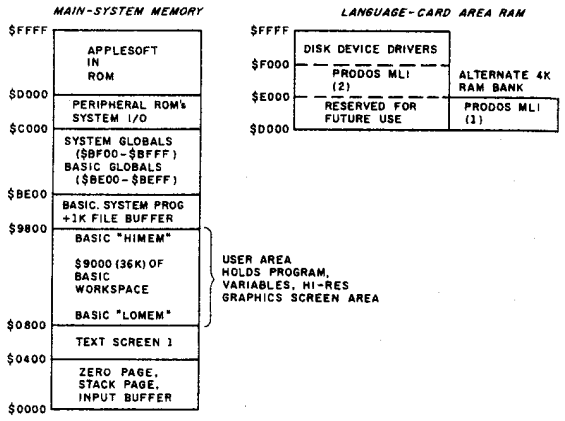


Figure 3: *A memory map of a 64K-byte Apple II or IIe with ProDOS and SYSTEM.BASIC installed. ProDOS is stored primarily in the language-card area of memory, and BASIC.SYSTEM resides in the area formerly occupied by DOS 3.3. Coincidentally, a ProDOS system provides about the same amount of free BASIC user space as a DOS 3.3 system.*

To efficiently accommodate files that can range from 1 byte to 16 megabytes, ProDOS stores files in three different ways (shown in figure 2). Depending on a file's size, it will be stored as either a "seedling" file, a "sapling" file, or a "tree" file. Files with sizes between 1 byte and 512 bytes are stored as seedling files—the data is stored in a single disk block. A sapling file can range in size from 513 bytes to 64K bytes. ProDOS creates an index block that holds the block numbers of up to 256 data blocks, each of which can store 512 bytes of data. Files larger than 64K bytes are stored as tree files that can consist of up to 32,768 data blocks or 16 megabytes of data. To store a tree file, ProDOS uses a master index block that holds the block numbers of up to 128 index blocks, each of which can specify 256 data blocks. As files grow or shrink during disk operations, ProDOS changes the storage method automatically—the entire process is completely transparent to the user.

Inside ProDOS

From the BASIC user's point of view, many ProDOS commands are virtually identical to existing DOS 3.3 commands. However, ProDOS is a completely new design internally. It is normally located in the language-card area of memory and prevents the use of Apple's Integer BASIC that is loaded there, under DOS 3.3. (Figure 3 shows a memory map of a 64K-byte Apple II or IIe with ProDOS installed.) When used with assembly-language programs, ProDOS uses only the memory above hexadecimal BF00, leaving 45.75K bytes of free memory space in a 64K-byte machine.

Also unlike DOS 3.3, ProDOS doesn't include any user interface. All commands are passed to ProDOS through a set of 24 machine-language calls to the ProDOS MLI (machine-language interface), which are similar to the

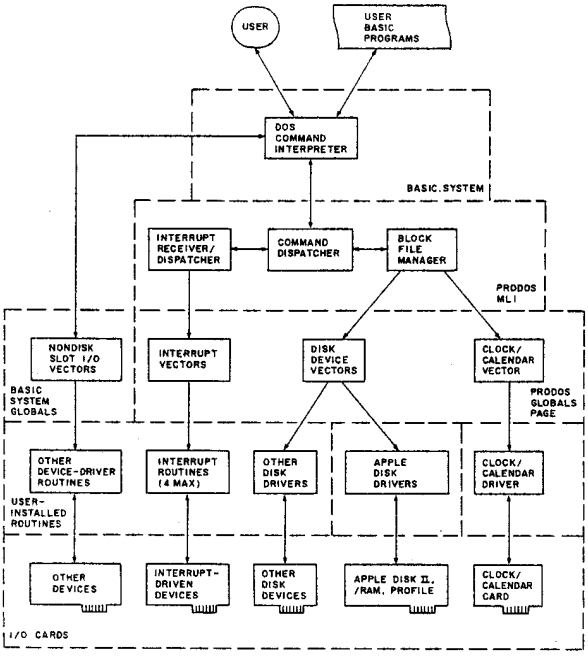


Figure 4: *A detailed block diagram of an Apple II or IIe with ProDOS and BASIC.SYSTEM installed. Arrows indicate the flow of control through the system. ProDOS includes device-driver routines for Apple's Disk II floppy-disk and Profile hard-disk drives. Also included are drivers to handle the /RAM pseudodisk and a Thunderclock clock/calendar card. Any other disk or clock/calendar devices require additional routines installed. Other interface cards with on-board ROM drivers will be recognized and operate properly, providing that they follow Apple's various interface guidelines.*

set of calls used by the Apple III's SOS operating system. No provision for user-entered commands exists within ProDOS itself. Instead, ProDOS is designed to work with various .SYSTEM programs, which translate user-entered commands into the appropriate ProDOS calls. (Figure 4 shows a block diagram of a typical BASIC user's ProDOS system.)

To maintain a consistent DOS interface while allowing Apple to alter the internal code in the future, ProDOS uses a 256-byte page of memory as a ProDOS globals page. The globals page is used to hold the addresses of all disk-driver routines, interrupt handlers, the clock-calendar routine (if installed), and the call address of ProDOS itself. It also holds a memory bit map that is used to indicate which pages of memory are currently used so ProDOS won't overwrite them during a load. The globals page is the only part of ProDOS that Apple guarantees will be consistent when future versions are released. No matter where ProDOS itself is loaded (depending on memory size), the globals page will always be in the same place and contain the same data. To allow programs to deal with future versions of ProDOS, the globals page also includes data to indicate the version of ProDOS currently installed in the machine and data to indicate the type of machine it's installed on. Currently, the system recognizes the Apple II series of computers or Apple IIIs running in Apple II emulation mode. However, room is available for future expansion.

Curiously, ProDOS won't work on most of the Apple II work-alikes. On a Franklin, for example, it hangs up during the boot process and coincidentally leaves Apple's copyright message on the screen. This is unfortunate for the owners of these machines, but you can't really expect the people at Apple to spend much time getting new software to work on unauthorized copies of its hardware.

BASIC.SYSTEM

Most users will talk to ProDOS through the .SYSTEM program called BASIC.SYSTEM, which provides an extended set of DOS 3.3-like commands to Applesoft BASIC. When BASIC.SYSTEM is loaded, it resides just beneath ProDOS in memory. Coincidentally, it leaves about the same amount of free memory as DOS 3.3.

Most BASIC.SYSTEM commands are compatible with the equivalent DOS 3.3 commands to allow existing BASIC programs to run with minimum alterations. A few seldom-used commands have been removed, however, and a number of new commands have been added and existing commands extended. You can now, for example, use OPEN, READ, WRITE, and CLOSE to access any type of file (including directories), and you can use BLOAD or BSAVE on any part of any type of file.

One of the most interesting new commands is simply a dash, a "run anything if possible" command. By typing " – filename", you can run a BASIC program, run a binary program, execute a text file of commands, or load and run a new .SYSTEM program.

To help you deal with ProDOS's nested directories, BASIC.SYSTEM provides a PREFIX command. You can use it to either set the pathname prefix to specify a default directory or to read back the current pathname prefix onto the screen or into a program variable.

BASIC.SYSTEM also provides I/O (input/output) commands that are much more flexible than those found in DOS 3.3. Instead of just using PR# or IN# to specify an I/O slot for input or output, BASIC.SYSTEM provides extensions to these commands that allow you to specify a particular address in memory to call for input or output or even to assign new addresses to given I/O slots. For instance, if your printer-interface card is in slot 1 and you have a special printer-driver routine loaded into hexadecimal address 300, you could type PR#1,A$300. From then on, any output normally sent to slot 1 would be sent to your routine at hexadecimal 300 instead. Another way to accomplish the same result would be to type PR# A$300 when you want to turn the printer on and PR#0 when you want to turn it off.

Adding Commands to BASIC.SYSTEM

In addition to providing a variety of flexible DOS commands, the BASIC.SYSTEM command structure is also extensible—you can add your own commands for special applications. By changing a pointer location in

Listing 1: *A listing of the two benchmark programs used to perform the disk access-time tests for the 500-record file. The programs used for the Apple III and the IBM PC were similar. A 500-record file is created by the program in listing 1a and is read by the program in listing 1b (see "The Apple III and Its New Profile," September 1982 BYTE, page 92).*

(1a)

```
20 D$ =  CHR$ (4): REM  CNTRL-D
80 NR = 500
100  PRINT D$;"OPEN TEST"
110  PRINT D$;"READ TEST"
140  FOR I = 1 TO NR
160  INPUT B$
180  NEXT I
200  PRINT D$;"CLOSE TEST"
220  PRINT "DONE"
```

(1b)

```
20 D$ =  CHR$ (4): REM  CNTRL-D
40 A$ = "12345678123456781234567811234567 8"
60 B$ = A$ + A$ + A$ + A$
80 NR = 500
100  PRINT D$;"OPEN TEST"
110  PRINT D$;"WRITE TEST"
140  FOR I = 1 TO NR
160  PRINT B$
180  NEXT I
200  PRINT D$;"CLOSE TEST"
220  PRINT "DONE"
```

	Apple IIe ProDOS	Apple IIe DOS 3.3	Apple III SOS	IBM PC PC-DOS
Write 500 records	36	175	37	32
Read 500 records	35	221	33	23
Write 32K-byte file	19	44	—	—
Read 32K-byte file	5.5	31	—	—

Table 1: *A comparison of the relative floppy-disk access times of ProDOS and DOS 3.3 with times indicated in seconds. Times for the IBM PC and the Apple III are included for reference purposes. The 500-record file consisted of sequential 128-character records terminated with carriage returns, giving a total file size of just less than 64K bytes. (Benchmark programs used are shown in listing 1.) ProDOS appears to operate effectively six or seven times faster than DOS 3.3.*

the BASIC.SYSTEM globals page, you can specify the address of an external command routine, which will be called if BASIC.SYSTEM gets an unrecognized command.

Apple's Help and APA (Applesoft Programmer's Assistant) programs are two interesting examples of added BASIC.SYSTEM commands. Running the Help program on the /Examples disk adds a special HELP command to the system. You can then type HELP followed by any DOS command, and a screen of information describing that particular command will be quickly loaded from the Helpscreens file on the disk and displayed.

When you run APA, it adds a series of program editing and debugging commands. Some of the commands added by APA let you renumber your program lines, merge two BASIC programs, use automatic line numbering, hold a program in a special memory area, compress a program to remove all REM statements, and obtain an XREF (cross-reference) listing of variables versus line numbers.

Performance

According to Apple, DOS 3.3 transfers data to or from an Apple Disk II at about 1K bytes/sec while ProDOS transfers data at 8K bytes/sec. These figures are the raw transfer rates and do not include overhead time to access directories, open file buffers, or pass the data to Applesoft BASIC.

To evaluate the actual effective speed improvement, I used the same disk-access speed benchmarks used to evaluate the Apple III (see "The Apple III and Its New Profile," September 1982 BYTE, page 92). The evaluation included additional tests to write and read a 32K-byte file of binary data, simulating a large program store or load. (The benchmark programs used are shown in listing 1, and the test results are summarized in table 1.)

The performance increase with ProDOS was impressive. ProDOS ran five to six times as fast as DOS 3.3 when accessing text-file records or reading the 32K-byte binary file and about twice as fast when first storing the binary file. In fact, the times for ProDOS were almost the same as the times recorded for the Apple III running SOS, the progenitor of ProDOS.

It wasn't possible to use the same benchmark programs to measure the access times to the /RAM pseudodisk because /RAM provides 62K bytes of space while the benchmark program writes a 63.75K-byte file. However, the times to save and load the 32K-byte binary file were roughly 1.2 and 0.4 seconds respectively.

BYTE did not receive an Apple IIe Profile hard-disk drive for evaluation. However, based on the test results with the Apple III Profile, you should expect an additional improvement in overall disk speed of a factor of three to four.

Utility Software

Along with ProDOS itself and BASIC.SYSTEM, Apple provides three utility programs: Filer, to manipulate disk files and volumes; Convert, to transfer files between ProDOS and DOS 3.3 format disks; and Exerciser, to access the ProDOS MLI commands.

Filer is used primarily to copy files or whole disks or to initialize new ProDOS disks, but it also provides options that let you delete or rename files, rename volumes, alter file write-protection, and list ProDOS directories. To help the novice user, Filer provides a series of built-in tutor screens that explain the various Filer commands and options. Filer is also useful when you're moving files from one directory to another on the same disk. You could, for example, create a new subdirectory called Games and then use Filer to copy all your game programs from the main directory to Games. If the name

of each game program ended with the same suffix, the Filer wild-card character (=) could be used to accomplish the copy with a single Filer command.

Convert can be used to copy files in either direction between ProDOS and DOS 3.3. Since ProDOS uses different filenaming conventions than DOS 3.3 (filenames can be 15 characters long and cannot contain spaces), Convert modifies DOS 3.3 filenames by truncating them to 15 characters and changing any spaces to periods.

Both Filer and Convert appear to be well written and user-friendly. The prompting is clear and unambiguous—it will be difficult to make an error using either program.

Exerciser is primarily a tool for system program developers. It can be used to manually execute each of the ProDOS MLI calls and to modify areas of memory. Exerciser should be used with caution. It will be very easy for an unwary user to damage the data on a disk because Exerciser lets you read or write data onto any disk block. However, it should be invaluable for debugging systems or for patching a damaged disk.

Documentation

Although the manuals I received for review were early draft copies, the ProDOS documentation looks excellent. All three manuals are well written, containing numerous examples and special cautionary messages when describing areas where problems might occur.

The *ProDOS User's Manual* explains how to use the Filer and Convert utilities. Written at a level suitable for rank beginners, it takes you step by step through each Filer and Convert function, explaining any new term encountered along the way. I found the *ProDOS User's Manual* to be the least needed of the manuals supplied because both Filer and Convert are menu-driven and virtually bulletproof. You simply progress from menu to menu, selecting the options you want—it's almost impossible to make any serious mistakes.

BASIC Programming with ProDOS will probably be the most read of the ProDOS manuals. Even though much of the material included will already be familiar to DOS 3.3 users, it explains all the new ProDOS commands and options and includes descriptions of ProDOS directories, pathnames, and file types. For the beginner, *BASIC Programming with ProDOS* includes all the information necessary to learn to use ProDOS effectively. A large percentage of the manual is spent explaining text files. It includes chapters that describe how text files are created and accessed, how to use random-access files, and how to create files of commands as Exec files. The appendixes include descriptions of the differences between DOS 3.3 and ProDOS when using Applesoft BASIC, a summary of ProDOS commands and features, descriptions of all error messages, and a complete glossary of all terms used.

The *ProDOS Technical Reference Manual* provides a complete description of ProDOS's inner workings. It will be invaluable to anyone who wants to write a system program, install his or her own device drivers, or deal with

ProDOS from assembly language. It includes chapters that describe files and pathnames, calls to the ProDOS MLI, writing system programs, and adding routines to ProDOS. Appendixes provide descriptions of the file-storage methods and directory formats, disk organization, the relationship between ProDOS and SOS, and how to use the Exerciser program.

ProDOS Packages

At the time this was written, Apple had tentative plans to market four different ProDOS packages. For assembly-language programmers, there will be a ProDOS Tool Kit with a new version of Apple's EDASM editor/assembler, a debugger program, and the *6502 Assembler/ProDOS Tools Manual*. The new version of EDASM includes macros and supports the various ProDOS file types and pathnames. Apple will also market a package that consists of the *ProDOS Technical Reference Manual* and the Exerciser program—useful if you want to write additional device drivers or .SYSTEM programs that interface directly to the ProDOS MLI. A BASIC users package will include *BASIC Programming with ProDOS* along with the BASIC /Examples disk. A utilities package will combine the *ProDOS User's Manual* with a /Utilities disk holding Filer and Convert.

Pricing hadn't been set when this article was written. Most previous Apple software packages of this type have cost less than $100.

Conclusions

ProDOS provides a significantly improved operating system for Apple II and IIe computers. With disk accesses about six times faster than DOS 3.3, files up to 16 megabytes long, and support for Apple's Profile 5-megabyte hard-disk drive, ProDOS provides an environment that will make it easier for applications developers to write the increasingly sophisticated software required by the business community. Using disk and file formats that are compatible with the Apple III's SOS operating system, ProDOS allows development of software that will boot up and run properly on either machine—loading ProDOS on Apple IIs or IIes and loading SOS on Apple IIIs.

For the more casual user, ProDOS may initially seem more complex to learn than DOS 3.3 due to its nested directory structure and increased command options. However, it will be easier to keep your programs organized under ProDOS, and the volume names and pathnames will prevent errors that could occur under DOS 3.3 if the wrong slot # or drive were specified. You should note that ProDOS does not make DOS 3.3 obsolete. There is no need to convert existing software to ProDOS unless you need the increased performance or new features.

For hardware developers, ProDOS provides a uniform protocol that lets you interface virtually any type of disk-like storage device without resorting to the now common DOS patches. If Apple's interface guidelines are followed, foreign disks will be recognized and used properly without requiring any alterations to the operating system. (This will also make life a little easier for users.)

On the minus side, neither Apple's UCSD Pascal nor Apple CP/M is compatible with ProDOS, and this may prolong some existing problems. Current suppliers of hard disks for the Apple II generally provide software that lets you partition your disk into areas for DOS 3.3, Pascal, and CP/M, because the three operating systems are not compatible. On the Apple III, Pascal resides on the SOS operating system and shares the nested directory/pathname facilities to manage disk storage. I hope to see a new Pascal for the Apple II with the same features. In the interim, Apple will supply software that will let you partition your Apple IIe Profile hard disk into separate areas for ProDOS and Apple's Pascal 1.1 operating system.

To summarize, Apple's new ProDOS represents a significant improvement over the existing DOS 3.3 operating system and includes features that are unavailable with most other personal computer operating systems. Although it may not be worth the effort to convert your existing DOS 3.3-based software to ProDOS, its capabilities give it some strong advantages for development of new programs.■

Rob Moore (Warner Hill Rd., RFD #5, Derry, NH 03038) is a design engineering manager with an interest in FORTH, graphics, and computer music.