June 1985

Exchange

Hardware

1 Connecting Personal Computers: The Major Options

5 The IBM PC Network: An Overview

9 The IBM Personal Computer Network (Part 1)

Software

17 TopView

26 Case Conversion in BASIC

27 Clearing Part of the Screen in BASIC

28 What's New in DOS 3.00 and BASIC 3.00

37 XOR and Pixel Color Manipulation

Computing Theory

39 How Compilers Work: An Overview

Getting Started

41 The Mechanics of Assembly Language (Part 2)

Random Data

43 Commonly Used Extensions for PC File Names

44 Troubleshooting Your IBM PC

48 **New Products**

51 **Ask IBM**

52 **Editor's Comments**

Exchange of IBM PC Information

IBM

Exchange of IBM PC Information is a monthly publication of the National Distribution Division, International Business Machines Corporation, Boca Raton, Florida, USA.

Editor	Michael Engelberg
User Group Editor	Bernard Penney
Writer, Design Consultant	Karen Porterfield
Writer	John Warnock
Illustrators	John Alfred Dorn III Narda Lebo Marc Rosenthal
Production	Cohen and Company
User Group Support Manager	Gene Barlow

Exchange of IBM PC Information is distributed at no charge to registered PC user groups. To register with us, please write to:

IBM PC User Group Support
IBM Corporation (2900)
P.O. Box 3022
Boca Raton, FL 33431-0922

To correspond with *Exchange*, please write to:

Editor, *Exchange*
IBM Corporation (2900)
P.O. Box 3022
Boca Raton, FL 33431-0922

POSTMASTER: send address changes to *Exchange of IBM PC Information*, IBM Corporation (2900), P.O. Box 3022, Boca Raton FL 33431-0922.

Connecting Personal Computers: The Major Options

Ken Goosens
Capital PC User Group
Washington, DC

The great advantage of connecting computers is that more than one user can access the same resource at the same time. This can increase productivity, lower costs, and allow new types of applications. But it also makes applications interdependent, increases the risk of disaster, and requires more overhead to manage.

Connecting personal computers (PCs) is therefore not a panacea and may even be undesirable. One of the greatest advantages of PCs is that they provide a dedicated and complete computer system that people can use with little or no support or supervision from data processing professionals. Applications can be quickly developed and put into production by the people who will use them. They can use powerful packages, and no third parties need be involved.

Once PCs are connected and resources shared, however, their use must be coordinated and controlled; security and access becomes more of an issue; and all parties must be consulted to determine who will be affected by new changes. Connecting PCs together can be counterproductive because it lessens their independence and re-institutes third party overhead. It is justified only when the inherent costs and disadvantages are outweighed by cost reductions and increases in productivity.

The three main advantages of connecting PCs are:

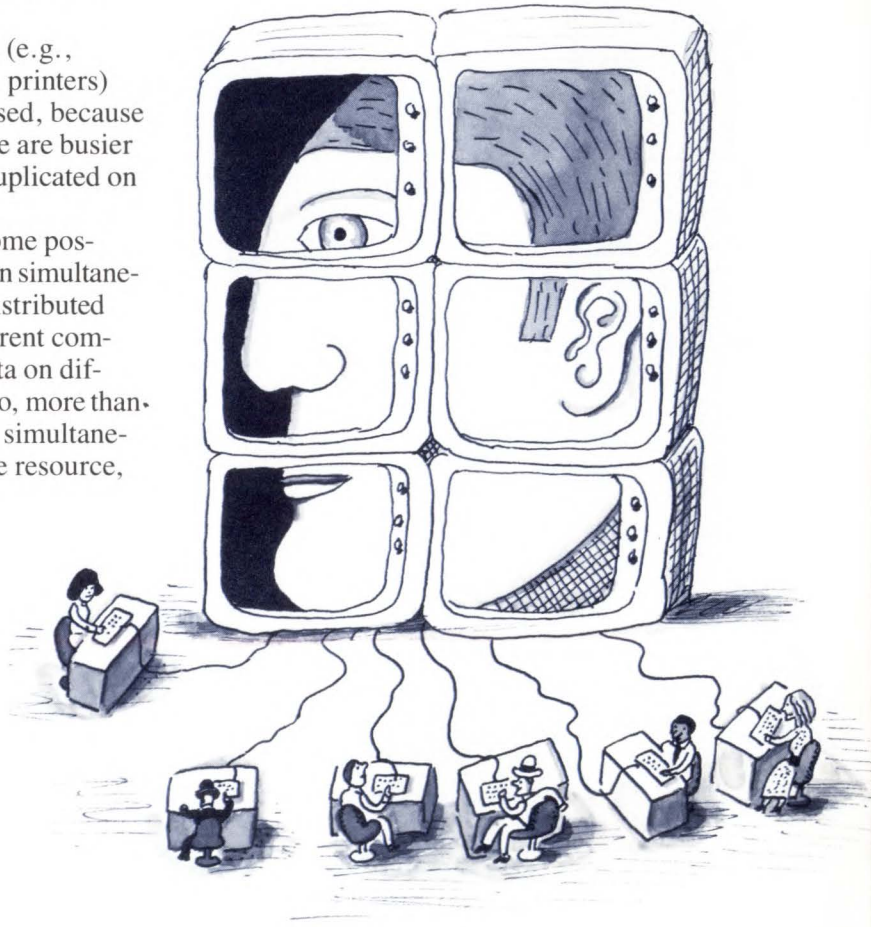
(1) Applications that have to access the same resource, such as a data base, can be run concurrently rather than in sequence.

(2) Fewer peripherals (e.g., modems, plotters, printers) need to be purchased, because those that you have are busier and need not be duplicated on multiple systems.

(3) Applications become possible that depend on simultaneously accessing distributed resources on different computers, such as data on different systems; also, more than one computer can simultaneously use the same resource, such as a single hard disk.

Deciding How to Connect PCs

There is no one best way to connect PCs. PCs might be used by a few secretaries doing word processing, by several service advisers in an automobile repair shop, by 15 real estate agents in an office, or by a diverse group of programmers and analysts in a large business with mainframe computers and 200 people in data processing. The right answer depends on how many PCs are to be connected, how they are to be used, what computing resources are already available, what the organization can afford, what level of internal support can be given, what skills the users have, and how work is currently organized.



To be warranted, connecting the PCs not only has to be affordable but has to increase productivity. This requires that the system mesh with the skills and style of an organization.

The best we can do in this article, without knowing the particular resources and type of work in an organization, is to compare how PCs could be connected. You will have to evaluate the different options against your own requirements.

Multi-User Problems

A multi-user system has problems for which there are no counterparts in a single-user system. These technical problems must be solved if the multi-user system is to work reasonably. Only an operating system that was designed to be multi-user will automatically solve them, which means that good multi-user systems must provide their own special solutions when added to single-user operating systems such as IBM PC DOS.

Every system that allows resources to be shared must deal with the fact that requests can conflict. Some devices, such as printers, plotters and most modems, can service only one user at a time. These resources must be managed so that only one of the many users who have access to them actually use them at any given time. The system must be able to detect that a device is busy and then either deny others use of it or else queue the request, storing it for execution after the device becomes free.

The system must solve simultaneous requests for the same resource, and the processing of service requests must be consistent. One possible problem is that the same space on a hard disk is allocated to two different users. When a request for more space is received, the system typically checks to see what is free. But if a second request comes in before the information is revised, the system may erroneously think that the space is free when it is in fact allocated, so that the files of two users get intertwined and whole segments of data of one or more users get lost.

When two users load a copy of the same file and make different changes, the last one who writes back may destroy the changes made by the other. A multi-user system must have some way to lock data that one user is updating so that others cannot update the same data.

To be warranted, connecting the PCs not only has to be affordable but has to increase productivity.

Splitting the Processor

An unusual way of connecting PCs is first to "split" one PC into several. The PC's central processor rapidly cycles between users, alternatively giving them processing time. Each user has a keyboard and "dumb" terminal attached to the shared computer. Most mainframe and minicomputers work this way; most microcomputers do not. To split a PC, three major technical problems have to be overcome: (1) there has to be a way to attach more than one terminal and keyboard; (2) there must be a program that switches the processor between users and separates the input and output of each; and (3) the processor has to be powerful enough to support more than one user.

The IBM personal computer family was never designed to support additional terminals. Third parties have nevertheless tried to retrofit such a technology, with rather poor results. These systems have been plagued by slow performance, incompatibility with standard IBM programs, unreliability and failure to solve the multi-user problem.

Shared Bus

Splitting a processor among users means that each user has to contend for processing time, which necessarily must degrade the performance of the system for other users. An alternative approach is to give each user a separate dedicated processor, but to attach each computer to a common "bus" which shares its attached devices. Response time is better because the users are not contending for a single processor. A "bus" is like a slot or channel. Each IBM PC contains a private bus to which expansion cards are attached. Peripherals are shared by attaching them to a public bus or by making the computer's private bus sharable.

There are two types of shared bus systems, which differ as to how a processor is provided. One system provides just the common bus for peripherals; an existing PC is connected to it by adding an interface card to the PC's private bus and connecting the two. Only

peripherals with a special controller will work on this bus, but it does support hard disks and tapes. The system locks out a tape drive to others when in use, and the hard disk has limited multi-user protection at the drive level, based on a hardware setting on the interface card. Only about four PCs can be effectively supported on the bus, and the cost of the interface is about as much as the network card.

A second type involves a slave processor on a card. The user provides a terminal with a built-in keyboard that is attached to the processor via a serial port. The advantage is that you do not have to buy a stand-alone computer to expand the system, but only a cheaper expansion card with the processor built into it. One difficulty with this approach, however, is getting a terminal that has the same keys as the IBM keyboard, and one that will support graphics.

The ancestor of the modern shared bus system was a multiplexer, which allowed several PCs to be cabled into it, and which in turn was cabled to a hard disk. Multiplexers, however, gave no multi-user protection.

Network

A network consists of computers wired together plus attached devices that the computers can share. Each computer has some sort of network interface card, and each public device has to have an intelligent server which processes requests to it. Most older and larger networks built the server into the device, which makes the device independent of the computers attached to the network but limits the number of devices that work on the network. Most networks designed for microcomputers put a server processor on the computer interface card, so that devices on the PC have to be shared. This lowers the cost to network PCs, since additional devices do not have to be purchased just to share on the network. But it makes the network dependent on having that PC running on it. Some networks support "diskless" systems which have only a processor, keyboard, and terminal.

Another strategy for joining PCs is to tie independent PCs into another computer as a "workstation."

Networks for personal computers are not a mature technology. All allow a hard disk to be shared, most support a printer, few will share a modem or plotter, and none yet will share a 9-track tape drive. The multi-user problem is not fully solved. Most networks support file and record locking only with specially modified network versions of software, rather than passively locking out all software at the network level. Security to limit access to files is weak on most networks. Finally, no widely accepted standard for networks exists.

If you are seriously considering a network:

- Look for the ability to share many devices that are attached to PCs.
- Make sure you can share all the drives on a hard disk.
- Look for support for your existing equipment rather than special or proprietary equipment.
- Find out whether multiple users can write to different files at the same time on the same drive.
- Ask what happens when two users with write access to the same file both attempt to use it simultaneously.
- Look for a system with a passive lockout system that does not require the user to lock files or single user software to be modified.
- Demand a security system that will distinguish among users as well as groups of files.
- Ask whether there is special networking software that supports electronic mail, print spooling, multi-user calendaring, and especially data bases with record locking.
- Get a list of known software that runs on stand-alone PCs that will not run on the network.
- Check whether there are gateways to other networks, and whether one PC directly wired to a second computer can act as a kind of "cluster controller" and service other computers on the network, allowing them access to the other computer. Some networks will support the Personal Computer AT and 3270 PC as well as the standard IBM PC.

Workstations

Another strategy for joining PCs is to tie otherwise independent PCs into another computer as a "workstation." The PC is used like a dumb terminal in that the other computer runs jobs based on the input of the PC. The other computer is a "host" for the PC, and PCs share through the host.

The host is usually a multi-user, multi-tasking mainframe or minicomputer, and the microcomputer either uses a modem to dial into the host or is directly wired to the host. In some cases the PC must "emulate" a special type of terminal that the host expects, and a special hardware board may or may not be needed. By putting special software on the host and PC, the two systems can exchange data.

The amount of integration between the host and the PC workstation varies enormously. At one extreme, a program running on the PC can transparently access and use data on the host. The very same program can run on both the PC and the host, allowing applications to be developed on one computer and transferred to the other. The exchange of applications makes it easier to train users and support both computers, as well as provide PC applications with the option to run on more powerful computers, or mainframe applications to be offloaded to PCs.

The multi-user problem has been well solved on minicomputers and mainframes, so that joining PCs through them should give the necessary multi-user protection. Consider how easy and inexpensive it is to cable PCs directly to the larger computer and to exchange data. Cabling a PC to an IBM mainframe can cost up to $1000 per machine, whereas other computers can be directly cabled to the PC's serial port with very little trouble or expense. Do not overlook the need to install software to support data exchange.

The workstation approach can be very attractive to organizations that already have heavy investments in existing computer systems. PCs can tie into existing data bases and production systems; corporate data can be shared without bypassing existing security; existing structures can support the PC; and no radically different equipment need be purchased just to join PCs.

Using the PC as a workstation is also an excellent way to join together a large number of PCs, especially when combined with software that runs on both systems.

Comparison of Alternatives

If you have a PC and just want other people to be able to work on it at the same time, the split processor approach is attractive. Simply load up one PC with a hard disk that is big and fast, a modem, a plotter, a letter quality printer and a fast dot matrix printer. The disadvantages are that current split processors usually cannot be expanded beyond four users, and each added user degrades the system's performance.

Don't get a processorless system unless your PC can support more users. The key question is whether the bottleneck in your processing is the processor itself rather than the time it takes to access data on a floppy diskette, or the printer speed, or how fast you can type. Applications that are computationally bound (continuously running the processor) will take as much or more time when run on a processorless system as when run on an independent PC. If you sit waiting for your computer to do its calculations, adding more dumb terminals will only make performance worse. The approach of adding processorless terminal banks on a PC is based on the fact that most of the time the processor sits waiting. If most of your processing is interactive, such as word processing, keying in data, and working with small spreadsheets, and the processor spends most of its time just waiting for the user or peripheral response, your PC can productively handle more than one user.

If performance is an issue or you need to connect more than four PCs, consider the slave processor approach. However, if you find having different keyboards confusing, or need graphics on every terminal, or need to share peripherals on different machines, you need to join full PCs, not just slave processors. Shared bus systems usually have the disadvantage that the cable connecting the PC to the shared system can only be a few feet long, so that the PCs have to be physically near each other.

If you need to join more than five PCs, or full PC systems, or have various groups of PCs that need to work together and share distributed resources, consider selecting a local area network designed to join PCs. But be ready to take on additional data processing functions. Be cautious about trying to join more than 20 PCs, because of limitations on the amount of information the connecting medium can support as well as increased complexity.

If you already have an office automation network, consider adding PCs as another device. If you want a comprehensive office automation system that can support word processing, electronic mail and calendaring, as well as traditional data processing, with shared devices like tape drives, laser printers, copy machines, and hard disks, don't go for networks that just join PCs together. Look for an office automation network or a minicomputer system.

If you have many PCs used for

production systems, consider linking them together as workstations on a more powerful computer and integrating the processing using software that runs on each system. This gives you the option to run the same applications on the more powerful computer. But you had better have some money to spend and a small data processing department to run the system.

If you already have more powerful computer systems and many scattered PCs, explore how the PCs can be used as workstations. But beware saddling the connected PCs with the traditional third party overhead associated with mainframes. Look for common software systems that can run the same system on both and exchange not only data but applications as well. Remember that programs with a mainframe history are usually not nearly as easy to use as most PC software. And don't force users to centralize their data storage unless their applications really need it.

Final Cautions

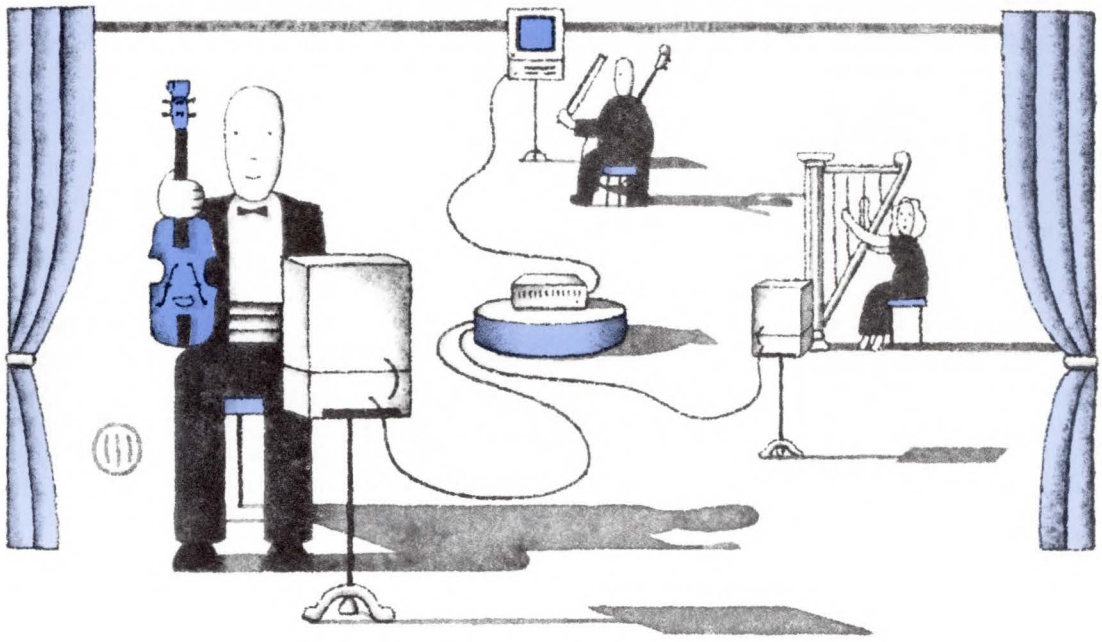
Personal computers were not initially designed to share resources among multiple users, and most of the technologies discussed in this paper are still developing. Conventional wisdom is that pioneers get killed. If that generalization were completely true, no new system would ever get started, but it is still good advice. Expect frustrations, because you won't be able to do everything you want. Don't just take the word of vendors—talk to other people who are using the system, and try it out before implementing. If you are using computer systems to run a business or do your job, don't be first—just be right.

The IBM PC Network: An Overview

Fred Williams
North Texas IBM PC Users Group
Arlington, Texas

The IBM PC Network is a two-megabit Carrier Sense Multiple Access/Collision Detect (CSMA/CD) broadband network. It uses standard 75-ohm coax cable (compatible with cable TV) and standard broadband components.

(The above paragraph is intended to contain the most technical information in this article. Anyone who wants to know the bits, bytes and buzzwords is encouraged to purchase the PC Network Technical Reference manual.)



The network requires all networked PCs to use PC DOS 3.10. Older model PCs require a ROM BIOS upgrade. (The IBM PCjr cannot participate in a PC Network.)

Any PC that functions as a resource server on the network must have a fixed disk, and therefore must be either a PC with Expansion Unit, PC XT, or Personal Computer AT. The network supports a Remote Program Load (RPL) feature that allows diskless PCs to exist as workstations on the network, as long as at least one file server is present on the network.

Number of Nodes and Distances
As announced, the IBM PC Network supports a maximum of 72 nodes within a 1000-foot radius of the Network Translator Unit. (A node is network terminology for an intelligent device, normally a computer, attached to the network). The Network Translator Unit is described in the IBM literature as a low-cost single channel unit that is required for each network.

The IBM PC Network design is based on Sytek broadband technology, and Sytek has indicated there is a very high level of compatibility between the two. IBM Network literature mentions several times that the Network may be further expanded through the use of an OEM broadband network that uses standard broadband components.

Any node may be located up to 200 feet from the IBM Network Translator Unit through the use of standard 75-ohm coax cable. The maximum radius of 1000 feet requires the use of special IBM PC Network Cable Kits to extend the capabilities of the IBM Network Translator Unit. These cabling components also are used to increase the maximum number of attached nodes from 8 to 72. The IBM literature mentions a 256 node maximum for the IBM Network Translator Unit.

Use of OEM components allows the network to be expanded to 1000 nodes connected within a five-kilometer (3 + mile) radius of the OEM network translator unit. Sytek network literature includes case studies of operational 3000 + node networks spread much farther than 5 miles.

Remember in all discussions of the network radius, the measurement is as the crow flies, not necessarily as the coax runs. So the standard 200-foot radius can get eaten up rather quickly. If offices to be networked are not immediately adjacent to the translator unit, extender kits most likely will be necessary.

PC Network Adapter
Not only does it provide the physical link to the PC Network, but more importantly it provides the network intelligence.

The heart of the IBM PC Network is the IBM PC Network Adapter, a special expansion card required to connect a personal computer to the PC Network. The PC Network Adapter requires one long expansion bus slot and must be located in the main System Unit, not an Expansion Unit. (As mentioned earlier, this adapter does not work in the PCjr.) Each PC may have a maximum of two PC Network Adapters installed.

I would venture to say it (PC Network Adapter) has more on-board horsepower than most of the PCs it is designed to plug into.

The IBM PC Network Adapter provides network interface services up through and including the "session layer" as defined in the ISO Reference Model. (An explanation of the ISO Reference Model, a widely accepted international standard for "layered" design of computer-based networks, is beyond the scope of this article). This is accomplished by a rather high level of intelligence provided by the combination of the adapter's hardware and its on-board NETBIOS (NETwork Basic Input/Output System) ROM.

By implementing the network connection from the lower ISO layers up through the session layer, the PC Network Adapter offers a high level of network connectivity, yet does not adversely affect the operating-system-dependent, ISO-defined presentation layer, which is the next layer above the session layer. This network implementation allows for the easiest link between a diverse number of workstation operating systems. Also, the PC's hardware and software are freed of both the overhead and responsibilities of implementing the network connection. This greatly reduces the PC system hardware and software requirements directly related to the network. Finally, this design gives PC-resident applications software an almost transparent level of interface to the PC Network.

The PC Network Adapter has an on-board 6MHz Intel 80188 processor, 6MHz Intel 82586 Communications Controller, 14MHz Sytek Serial Interface Controller, 32K x 8 Network Protocol ROM, 16K x 8 Network RAM, 32K x 8 Node ID PROM, PC Interface Controller, 8K x 8 NETBIOS ROM, and an RF Modem. (You could say it is a full card.) Although it is a dedicated function subsystem, I would venture to say it has more on-board horsepower than most of the PCs it is designed to plug into.

PC Network Translator Unit
As mentioned earlier, each PC Network requires one Network Translator Unit. In broadband network terminology, the IBM PC Network Translator Unit acts as a headend unit, which translates and balances the send and receive channels of the network stations. This translating and balancing is necessary for proper broadband network operation.

One thing worth mentioning about the IBM PC Network Translator Unit is a statement made by IBM in the announcement literature. This statement is as much related to the PC Network Adapter as it is to the Network Translator Unit. I will quote it directly and you, too, can let your mind wander into the blue sky:

> The spectral purity and signal quality of the translator does not support cable amplifiers, multiple channels or video. Note, however, that this is not a statement of the IBM PC Network Adapter which does permit active networks with multiple channels and video.

PC Network Program
Let's move on to the IBM PC Network software implementation. Software is what provides the user-apparent flavor of the actual network. The physical network determines the speed and data integrity of the network implementation, and the network software features establish the network's usability and productivity.

Most operating systems should be able to access the network capabilities in much the same way and as easily as they handle other input/output devices. This will greatly simplify the actual implementation of the network link.
If a programmer can successfully utilize the PC system's local disk storage, the task of using the network link should prove to be no more complicated.

The IBM PC Network appears to be a very appealing product.

To reduce the effort and time required in implementing some of the more generic and appealing features of a newly installed network, IBM has developed the IBM PC Network Program, licensed on a per-PC basis.

The IBM PC Network Program provides three levels of support to the PC user. These include, and I quote: "a full screen menu-oriented operator interface using function keys, help screens and non-technical vocabulary; a DOS-like command line interface for faster operation and batch file processing; and a program interface with low-level sharing control and network status for application developers."

Also provided as a part of the PC Network Program are file/print server drivers that can run in the background on a PC. As mentioned earlier, the server computer must have a hard disk and also must have a minimum of 256KB of memory.

Any computer that wishes to use the services of a network-resident file/print server directly will need to have the PC Network Program "redirector" module. This module redirects all local file I/O and print I/O requests over the network to the server. This makes the remote system's resources appear local to the redirected computer. Any PC wishing to use the redirection feature of the IBM PC Network Program must have a minimum of 128KB of memory and one double-sided diskette drive.

Multiple file/print servers may be active on a network. The file/print servers run in the background on the host PC system, allowing the PC to run user applications simultaneously with the server operation. This means that a server computer may be used as if it were not operating as a network server.

One feature of the IBM-supplied print server is the ability to share up to three IBM-compatible print devices on the network. The print queue may contain up to 100 print files concurrently. The server's operator can examine and modify the print queue and control the operation of the related print devices. Remote users of the print server also may use special functions to examine the status of their print files in the print queue.

The IBM PC Network Program also allows stand-alone single-session versions of the print/file server. This feature permits individual users to share files, but it does not run in the background and does not allow concurrent operation of any other user application.

Additionally, the IBM PC Network Program provides two different modes of message transfer between network users. One mode uses the full-screen operator interface and allows the user to edit, send, receive, save, and recall messages. The other mode allows the receipt of messages directly to the screen, printer or disk file, and permits sending messages from the DOS command line.

DOS 3.10

The major changes to DOS in release 3.10 are for support of the IBM PC Network. New features include multitasking and file sharing support. Multitasking allows your PC to run more than one application at a time. This permits a network resource server to run concurrently with other user applications on the same PC. The file sharing features assist the system designer in ensuring the data integrity of a data file that is being accessed by more than one user application at a time.

Summary of PC Network

As announced, the IBM PC Network appears to be a very appealing product. The broadband technology provides a very flexible network design medium which allows the implementation of networks ranging in sizes from 3 to 1000 nodes. These nodes may physically be located anywhere within an eight-mile circle. The smaller networks are designed to be user-installable. All network components for up to a 72-node network, including cabling, are available preassembled from IBM. Installation amounts to no more than unpacking, locating and cabling the network components together. After some initial testing, the network should be ready to go.

As mentioned by IBM in its network announcement literature, the IBM PC Network may be expanded further through the use of third-party OEM broadband network products. Several third-party PC product suppliers have announced support of the IBM PC Network.

With the ability to expand the backbone network cable system using Sytek components (there are Sytek-based networks installed and running with 3000 + nodes), and with the seemingly endless supply of third-party equipment becoming PC Network-compatible, networking choices are becoming unlimited.

One additional major advantage of the IBM (Sytek) system is its complete lack of dependence on a network "host" system. The network is made up totally of "peer" nodes without dependence on any one node for network control. The way this is done is highly technical, rather ingenious and beyond the scope of this article.

Network Cabling

I was at first skeptical of the pure coax-based network scheme, but for all the wrong reasons. At first, coax is more expensive per foot than twisted pair, and coax installation is more technically demanding than the old twisted pair. However, the percentage of installed base of twisted pairs will decline rapidly in the near future. I say this based on the assumption that an ever-increasing amount of all forms of data—digital, visual and audio — will flow into and out of the average business office. This will require a multitude of baseband carrier circuits, either forever-multiplying, multiple twisted pairs, or (more effectively) a single broadband circuit based on common coax. If you doubt my theory, ask for a tour of your company's current PBX system.

Also, the major cost involved in installing any cabling system is not the cost of the cable per foot. The greatest costs are the salaries of construction personnel and the physical requirements of the installation in order to meet local fire codes. Both of these factors vary from location to location.

Building Your Own PC Network

By using off-the-shelf IBM PC Network components, it will be quite possible to build a three-station network with laser printer (from a third party) output for under $20,000. By using IBM's preassembled cable kits, the system is user-installable. Also, use of the IBM PC Network Program file/print server software eliminates the need for any special programming effort.

The nice thing about the PC Network is that it may be expanded to include 1000 (perhaps more) nodes, plus carry voice and video over many miles of network. It truly is mind-boggling.

The IBM Personal Computer Network

(Part 1)

John Warnock
IBM Corporation

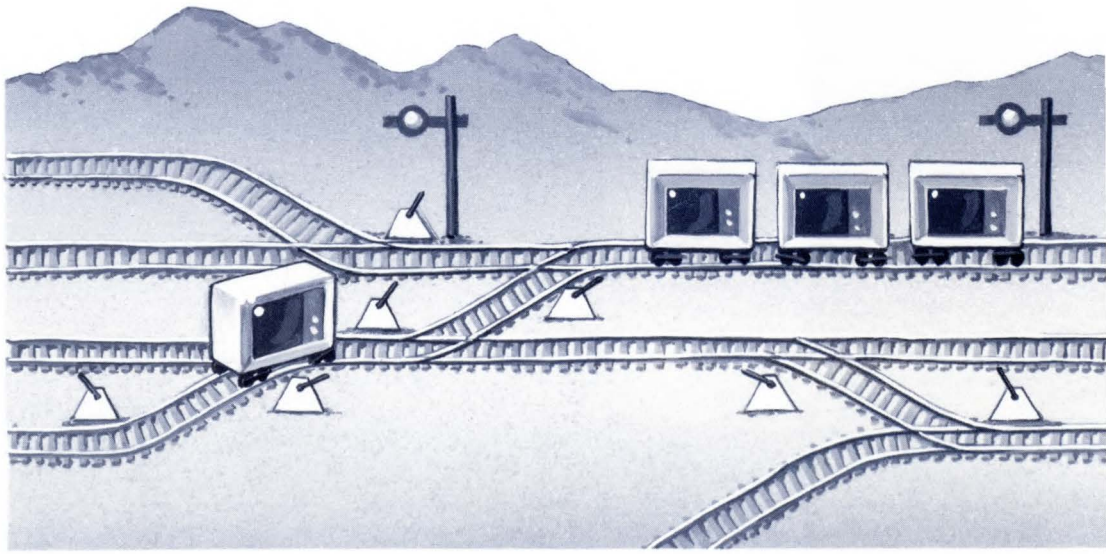
(Editor's note: This is the first part of an article about the IBM Personal Computer Network, providing an overview of the hardware and a description of the network architecture. Part 2, to be published next month, covers the IBM Personal Computer Network Program. This article is adapted from the IBM Personal Computer Seminar Proceedings.*)*

The IBM PC Network is a low-cost, broadband local area network designed for the IBM Personal Computer family. Its advanced technology provides a highly reliable, low-maintenance network using standard cable television (CATV) coaxial cable and connection hardware.

Networking Fundamentals

Broadband networks differ in a number of ways from the more familiar baseband networks. In a baseband network, the data is directly modulated on the cable using all of its bandwidth. In a broadband network, the data is modulated on a carrier frequency that uses only a small portion of the cable bandwidth. This makes other channels on the cable available for other uses, for example audio and video signals.

The IBM PC Network uses frequency modulation (FM). FM is immune to most noise, whereas amplitude modulation (AM), another method of signal transmission, is susceptible to noise. Just as lightning has little effect on an FM radio, noise has little effect on FM data. Once a carrier signal is available to the receiver, noise has very little effect, because the information is encoded into a frequency, not an amplitude.



Each personal computer (node) in the network has a Network Adapter card that handles communications. Each card follows the rules of Carrier Sense Multiple Access/ Collision Detect (CSMA/CD) protocol to avoid chaos and confusion.

In the CSMA/CD protocol, Carrier Sense means that each IBM PC Network Adapter continuously monitors traffic on the data channel. Multiple Access means that any network adapter can begin transmitting when there is a pause on the channel. If only one network adapter begins transmitting during a pause, it gains control of the channel and transmits its message without interruption. If two or more network adapters begin transmitting during the same pause, their signals collide. Collision Detection means that the receivers detect the collision of their signals. Using a random backoff-and-retry algorithm, each adapter will attempt a re-transmission at a later time.

The CSMA/CD protocol can be supported by baseband and broadband networks. The two kinds of networks support the CSMA/CD protocol in two different ways.

On baseband networks, the signal delivered to the receiver is affected by the distance from the transmitter. The greater the distance, the weaker the signal. In order to detect collisions, the receiver must be able to detect a

small signal from a distant transmitter in the presence of a large signal from a close transmitter.

On a broadband network, in order to detect a collision, the difference in signal strength between any two transmitters must be kept small at the receiver. This requirement led to the development of balanced networks. A balanced network is a network in which the signal strength from any transmitter to any receiver is the same.

A balanced network is designed in two parts. One part is designed so that the signal strength from any transmitter to one centrally located receiver is the same. A second part is designed so that the signal strength from a centrally located transmitter to each receiver is the same. The central location is called the head end. The device that moves the data from the centrally located receiver to the centrally located transmitter is called a translator.

Since broadband networks have unused cable bandwidth, the two networks actually can be on the same cable at different frequencies. In this case, and in the case of the IBM PC Network, the translator "translates" and amplifies the signal received at the centrally located receiver (the adapter's transmit frequency) to the frequency of the centrally located transmitter (the adapter's receiver frequency).

In any network there must be some connection between the hardware (wires, signals, transmitters, receivers, frequencies) and the software (programs, messages, information, status). Many networks use a layered approach

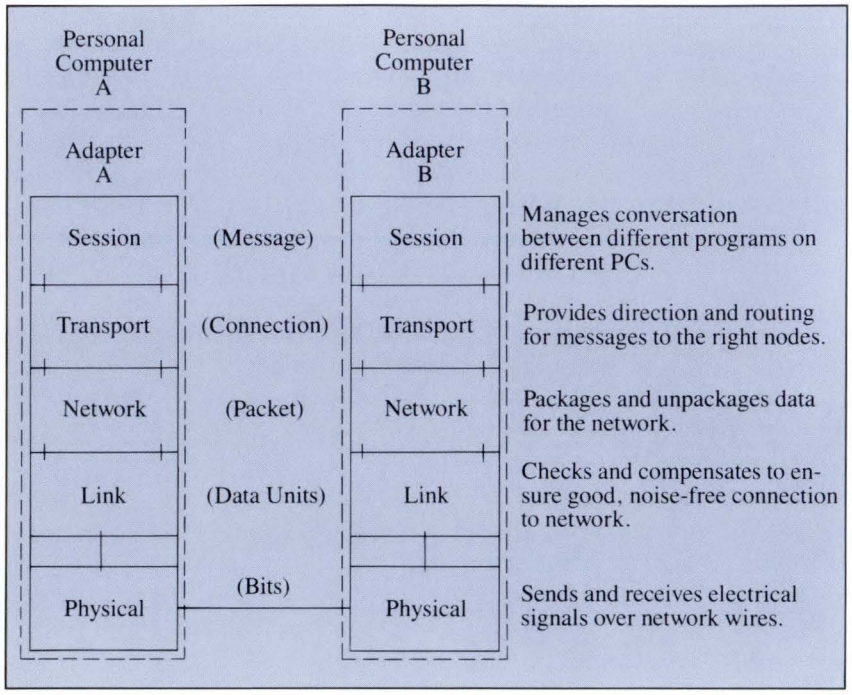


Figure 1. Protocol Layers

in which each layer performs translation between the preceding and following layers as well as error detection. Moving the information between the network (hardware) and the application (software) requires several layers of protocol, as shown in Figure 1.

Each layer has its own specific set of tasks. The physical and link layers are performed by the hardware; they insure a good connection and error-free transmission between the various members of the network.

The network layer prepares and packages data to be sent with error checking characters. It strips incoming messages of these characters and checks for errors as data is received.

The transport and session layers of the network are concerned with finding and maintaining communications with the right partner in the network. The network layer is responsible for data routing. The session layer keeps track of whom your personal computer is talking with, and who spoke last. This is important because you could have several sessions going on at the same time. You could be talking with a host computer while your payroll program is transferring information to the general ledger running on another personal computer.

IBM PC Network Objectives

Several key objectives influenced the overall design of the IBM PC Network.

- The network should be open to the industry, and key interfaces should be published.
- The network should be expandable.
- The network should require no host of any type; a peer-to-peer network is essential.

- Network firmware should conform to industry standards, if possible, and be layered into protocol layers.
- Network function should be executed on the adapter card and should off-load the Personal Computer from low-level network interaction.
- All network function below the adapter interface should be implemented in read-only memory. This allows loading programs remotely and requires no memory or diskettes in the IBM Personal Computer.

A session layer adapter interface was the design choice for several reasons. The presentation service layer is dependent on the operating system and therefore differs for various systems. The session layer also is the best layer for introducing the concept of names instead of addresses.

IBM PC Network Concepts

Some concepts designed into the IBM PC Network are:

- Peer Network. In a peer network, each member is treated equally on a first-come, first-served basis. There is no 'host' concept as in telecommunication operations. There are no required centralized facilities of any type on the network other than the hardware Translator Unit. Peer nodes can be connected with a reliable, point-to-point connection called a virtual circuit.
- Name Service. A name can be given to each member that is physically connected to the network. Names can be general, such as 'John', instead of specialized names or numbers.

If numbers were used, each adapter would have one fixed address on the network. By using names, the IBM Personal Computer Network Program may add up to 16 names for various programs running in the IBM Personal Computer. Each name may have associated sessions or commands which are separate from other names.

An example of the use of names is: the IBM PC Network Program may be loaded into the IBM Personal Computer. Once this is done, devices or files may be shared across the network by name. If a file server adds the name SERVER 1, a number of users, (i.e., John, Mary, Bill, ...) may add their names to their IBM Personal Computers and access the file server. Once the original IBM Personal Computer is turned off, John, Mary and Bill may move to any other IBM Personal Computers on the network and add their names; there is no dependency on physical addressing and no required centralized directory of names.

Names also may be clustered into logical groups on the network. If a number of users add a group name, then a message sent to that group may be received by all. An example might be that a number of people add a name such as DEPTXYZ. Then any messages sent to DEPTXYZ may be received by everyone with that name. Names must be used for both Session Services and Datagram Services.

- Session Services. After the names for each member are specified, two of the members may communicate with each other in a mode called a session. Sessions are similar to a telecommunication-reliable, point-to-point, two-way connection. For the IBM PC Network, a session also can be referred to as a virtual circuit.
- Datagram Service. The IBM PC Network supports messages called datagrams. Datagram services do not provide a reliable point-to-point connection. The datagrams are sent only once, and must be received immediately. The most common type is the broadcast datagram.

The adapter supports all network and protocol functions to assure that messages and data are sent from one computer to another on the network. It also provides the mechanism for returning command status to the IBM Personal Computer following command execution.

In conjunction with the IBM Personal Computer Network Program and DOS 3.10, these hardware components allow connected Personal Computers to share files and printers and to pass programs, data and messages to other Personal Computers in the network. Figure 2 shows how the components are connected in a typical network.

IBM PC Network Hardware Components

The IBM Personal Computer Network has three basic hardware components:

- The IBM PC Network Adapter, which plugs into an expansion slot in the Personal Computer system unit

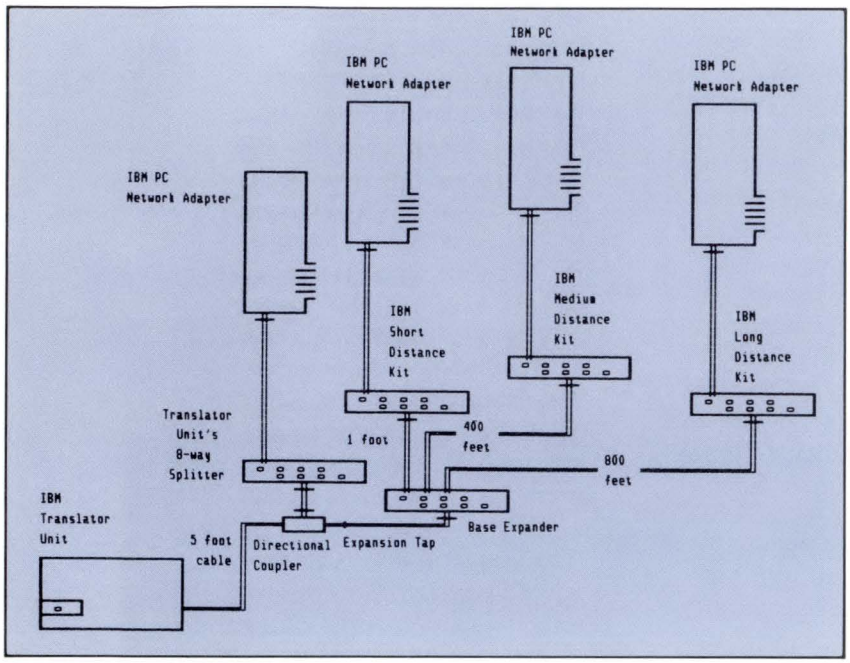


Figure 2. IBM PC Network Components

- The IBM PC Network Translator Unit, which provides the network signal and frequency translation
- The IBM PC Network Cable Kits, which connect the Network Adapters to the Translator Unit

IBM PC Network Translator Unit

The IBM PC Network Translator Unit forms the center of the IBM PC Network. It is a single-channel device that provides broadband frequency translation from the return channel to the forward channel. The Translator Unit is supplied with a separately packaged AC power transformer that plugs into a standard electrical outlet and powers the unit.

The unit also is supplied with a connector assembly for attaching up to eight IBM Personal Computer stations within a radius of 200 feet (cable segments purchased separately). This assembly also has an expansion port for attaching the IBM PC Network Base Expander Kit.

In conjunction with the IBM PC Network Cable Kits, the Translator Unit supports up to 72 stations within a 1000-foot radius. With a custom cable installation (not offered by IBM), the Translator Unit supports up to 256 stations within a 1000-foot radius.

IBM PC Network Adapter

The key to providing the network connection is the IBM PC Network Adapter card, which plugs into one expansion slot of an IBM Personal Computer, IBM Personal Computer XT, IBM Portable Personal Computer or IBM Personal Computer AT. (The PC Network Adapter is not supported on the IBM PCjr or the IBM Personal Computer Expansion Unit.)

Each IBM Personal Computer in the network requires its own PC Network Adapter card. Each card is supplied with a three-meter attachment cable. This cable can be connected directly—or with up to 200 feet of additional cable—to the IBM PC Network Translator Unit or to an IBM PC Network Short, Medium or Long Distance Kit.

With the adapter installed in an IBM Personal Computer, a user has the ability to communicate and share resources with any other node on the network.

The adapter card has its own intelligence. It contains an Intel 80188 microprocessor; an Intel 82586 Local Area Network (LAN) controller; a single frequency RF modem; two VLSI gate arrays; 40KB of Read-Only Memory (ROM) and 16KB of Random Access Memory (RAM). These components allow the adapter to perform all networking functions, including protocol processing through the session layer, and free the IBM Personal Computer from having to perform these functions.

The PC Network Adapter card features:

- Fixed-frequency assignment for midsplit broadband
- Extensive self-test and diagnostic capabilities
- An on-card auxiliary processor
- Session layer protocol support
- A unique 48-bit network identifier
- Direct memory access data transfer
- Station bootstrap across network support

The IBM PC Network Adapter supports the five layers of data transfer protocols shown in Figure 1. Each layer comprises one or more protocol services, and each layer communicates only with the layer immediately above or below it. This structure allows a modular design of the protocols.

The physical layer is implemented using the RF modem on the adapter and the interface logic to the Intel 82586 Communications Controller.

The link layer is implemented primarily in hardware by using the Intel 82586.

The other three layers (network, transport and session) are implemented using the Intel 80188 processor and ROMs on the adapter. Also, the layers provide a reliable virtual connection service, a name support facility and a low-overhead datagram service.

IBM PC Network Cable Kits

The IBM PC Network Cable Kits are designed to extend the functional capabilities of the IBM PC Network Translator Unit. These cable kits can be used to increase the maximum number of attached stations from eight to 72, and to increase the maximum distance of coverage from a radius of 200 feet to a radius of 1000 feet.

The product offering includes a Base Expander, three Distance Kits and four cable segments, which are preassembled from standard CATV components. All products are designed with industry-standard 'F' connectors for ease of installation. The cable kits include all components and cable segments required for end-to-end connection.

The network cabling components are:

- The IBM PC Network Base Expander Kit. This unit is required to expand the IBM PC Network to more than eight stations or more than 200 feet. It is a prerequisite for the attachment of the Short, Medium or Long Distance Kits described below. The Base Expander Kit can support up to eight Short, Medium or Long Distance Kits in any combination.
- The IBM PC Network Short Distance Kit. The Short Distance Kit attaches directly to the Base Expander Kit and allows up to eight stations to connect to the network within a 200-foot radius of the Short Distance Kit.
- The IBM PC Network Medium Distance Kit. The Medium Distance Kit, with two 200-foot cables, attaches to the Base Expander Kit. It allows up to eight stations to connect to the network, within a 200-foot radius from the Medium Distance Kit, for up to a maximum of 600 feet from the Network Translator Unit.
- The IBM PC Network Long Distance Kit. The Long Distance Kit, with four 200-foot cables, attaches to the Base Expander Kit. It allows up to eight stations to connect to the network, within a 200-foot radius from the Long Distance Kit, for up to a maximum of 1000 feet from the Network Translator Unit.

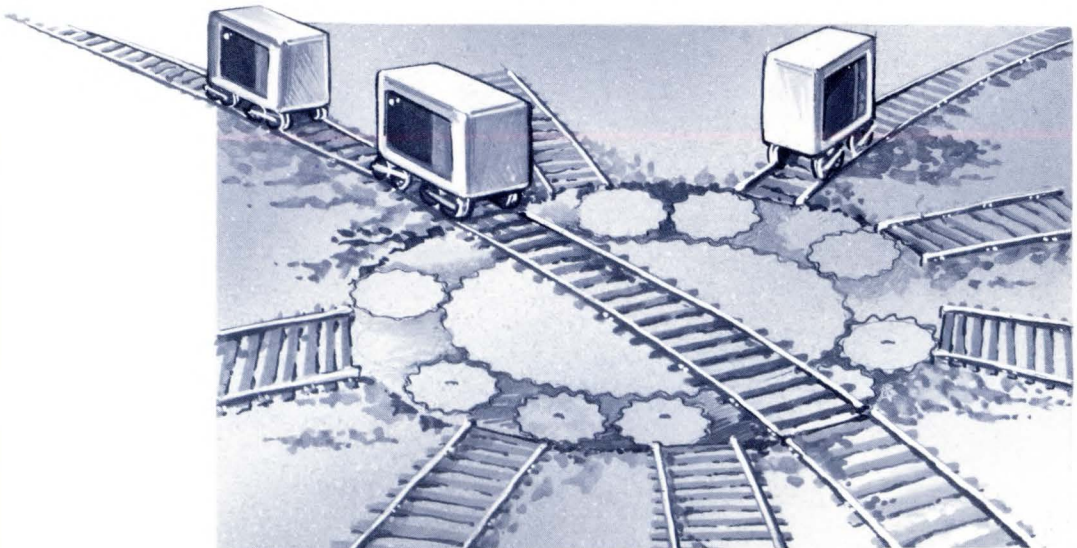
The network cabling segments come in four lengths: 25 feet, 50 feet, 100 feet and 200 feet. These cabling segments can be used in the following ways:

- To attach a Network Adapter to the Network Translator Unit connector assembly at a distance of up to 200 feet.
- To attach the Medium or Long Distance Kit to the Base Expander Kit at the specified distance.
- To attach a Network Adapter to a Short, Medium or Long Distance Kit at a distance of up to 200 feet.

You may also provide or subcontract from CATV vendors your own wiring and frequency translation. On this basis, you may increase the distance and number of personal computers supported. With a commercial translator you may support a network of 1,000 personal computers for a radius of 5,000 meters, and provide video and multiple channel capabilities too.

Network Basic Input/Output System (NETBIOS)

The IBM PC Network Adapter contains the Network Basic Input/Output System (NETBIOS) in its read-only memory. The NETBIOS is a software interface between the adapter and IBM Personal



Computer programs. The NETBIOS places the unique features of a local area network into a standard format.

The NETBIOS functions and interfaces provide the basis for all program control of the network. They are layered into five levels of protocol (described earlier), and are derived from contingency work in international standards. The highest layer of support on the IBM PC Network Adapter is commonly called the "session" layer.

The NETBIOS includes four major groups of functions accessible by interrupt X'5C'. The four groups are:

- Session control, to create a session and interchange information with another user (name) on the network
- Datagram functions, to send and receive one-to-one or broadcast datagrams
- Name control, to define multiple users within a node
- Network adapter status and control

The significance of these high-level interfaces in NETBIOS is:

- They offer an interface for coding network programs conducive to programmer productivity.
- They permit network application programs to be smaller and use less storage.
- They make it possible for multiple operating systems and application programs from any source to use the network with consistent protocol.
- They execute on the adapter card, and thus free the system processor for execution of application code.

An important feature of the NETBIOS is the concept of user-assignable names. There can be multiple names associated with each adapter card, and the names are globally known on the network. This makes it easier to dynamically configure the network under operator or application control. The NETBIOS supports up to 32 concurrent full-duplex sessions on each adapter card.

The NETBIOS on the adapter presents a high-level interface to the programmer, eliminating the need to know network protocol details. This high-level interface improves system performance by off-loading network programs onto the adapter. Also, this off-loading feature saves memory because the network programs are in the adapter's memory and not in the personal computer's memory.

Network security is not built into the NETBIOS. Instead, it is the responsibility of the operating system or application program to make certain that data or devices are secure on the network.

Programming the IBM PC Network Adapter

The following section outlines basic concepts of how to program your adapter using the NETBIOS and its interface to the IBM Personal Computer. Although the network is composed of many layers, the NETBIOS presents a consistent interface to the IBM Personal Computer operating system and should always be used.

Data Transfer

Two basic types of data transfer are supported. Data transfer under session support makes sure that a message is received. If the line to another system drops or is lost, the NETBIOS returns an error code.

Data transferred using datagram support goes directly to the link layer. This type of transfer does not contain the features found in the session layer.

Name Support

You must communicate on the network by using names. Each adapter can hold up to 16 names and one permanent node name. Each has a length of 16 characters, and all 16 characters are always used in a name.

Using the Network

In order to communicate on the network, a few simple steps are required.

1. Add your name to the table of names on the adapter. This is the name that you are known by on the network.
2. Establish a session with another name on the network. This gives you a logical connection with another name. The other name can be in your name table or in a name table of another adapter.
3. Send and receive messages using that session.

Network Control Block Commands

The Network Control Block (NCB) is used to issue commands to the adapter from the IBM Personal Computer memory. The command block is invoked by interrupting the adapter and pointing to the command block. When the command completes, the IBM Personal Computer is interrupted and status is returned. Data transfer from the IBM Personal Computer to the adapter is primarily via direct memory access.

NCB commands control an adapter on the network. The commands are divided into four categories:

- General
- Name support
- Session support
- Datagram support

Within each category, the commands are further divided into wait and no-wait options. The wait option means that when you issue the command, the processor waits until the command is completed before returning to the next instruction. The no-wait option means that the processor returns immediately after issuing the command and is interrupted at the post address when the command is completed.

Name Support Commands

Name support commands allow your IBM Personal Computer to be known by a name on the network. The commands are Add Name, Add Group Name, and Delete Name. A name can be a unique name or a group name on the network. For Add Name, the adapter checks to see if a name is unique, and returns an error if anyone else is using the name you want to add. When using Add Group Name, the same name can be added by many adapters on the network.

The adapter can have up to 16 names in the local name table. A permanent node name is always present; it consists of ten bytes of binary zero followed by the unique adapter unit ID number. This permanent node name also is unique on the network.

Session Support Commands

Session support commands allow you to establish a logical connection (session) on the network, send and receive messages, end sessions, and read session status. More than one command can be outstanding because the connection is in a two-way simultaneous transmission mode.

Sessions are established between any two names on the network. These names can be on your adapter or any other adapter. Names are used to establish sessions, but after the sessions are established, a one-byte number is used to refer to each session. A maximum of 32 sessions is allowed. The same name pair can be used to establish more than one session.

Session support gives you reliable data transfer and receipt of a message. Messages can range from 0 to 65,535 characters in length.

Datagram Support Commands

Datagram support commands allow you to send a message to a name, a group name, or to broadcast a message to everyone. These commands also allow you to receive a datagram message from a name, a group name, or from anyone on the network. Datagram support differs from session support in several ways. The message is never acknowledged by the receiver's adapter, so it is up to the sender and receiver to agree on their own network protocols. Messages can range from 0 to 512 bytes in length.

Datagrams are smaller than session SENDs, and require additional protocol interaction for reliable data transmissions. For reliable transmissions, session SENDs always should be used.

Remote Program Load (RPL) Feature

The NETBIOS on the IBM PC Network Adapter provides the capability to boot a computer from the IBM PC Network. The IBM PC Network NETBIOS redirects the initial diskette read requests to the network if there are no other drives enabled and the RPL jumper on the adapter is removed. The boot request goes to a special IBM name on the network called IBMNETBOOT. The IBMNETBOOT name must be active, and it must handle the boot request from remote IBM Personal Computers. This function does not depend on the operating system and can operate with any system that uses diskette I/O during bootstrap. The main restriction is that the operating system must use Interrupt 13 requests, and not try to use the diskette hardware directly.

Performance Considerations

There are a number of steps that may be taken to optimize the performance of each adapter. The major consideration when using the adapter should be to send data in as large a block as possible. This reduces network overhead and maximizes throughput. Data also should be sent using sessions, since this allows reliable data transfer and maximum performance. Issuing multiple receive requests also is helpful in some environments, since it allows overlap of network activity and local I/O activity on the IBM Personal Computer.

In a single point-to-point session, multiple sends of large data blocks on one adapter and multiple receives on the other end will maximize throughput. Using the reset command to configure the adapter to the smallest number of sessions and outstanding commands also is worthwhile, since the memory on the adapter is limited.

After space is set aside for session entries and command blocks, the remainder is used for buffers. The packet transmitted on the network is equal to or less than the size of the transmission buffer. Therefore, the fewer configured sessions, the larger the packet transmitted on the network.

In order to take advantage of the PC Network Adapter features:

- Use session support, not datagram support
- Send data in large blocks whenever possible
- Issue multiple commands
- Overlap network requests with other Personal Computer I/O
- Configure the adapter to only the number of sessions and commands you need

The IBM PC Network Technical Reference manual contains further information about the meaning of each command. It also contains pseudo code that describes the network action when each command is issued; the NETBIOS assembly listing; and the packet formats on the network. The manual includes two sample programs to help you get started using the adapter.

* * *

This article continues next month with a description of the IBM Personal Computer Network Program.

Let's Hear From You!

As an IBM Personal Computer user, you have viewpoints about IBM's products and services. As a reader of *Exchange*, you have opinions about this news journal, as well as ideas about what you would like it to contain.

Your comments are important to us. We welcome you to send us your comments (please include your name, user group and address). Each month we'll print selected letters.

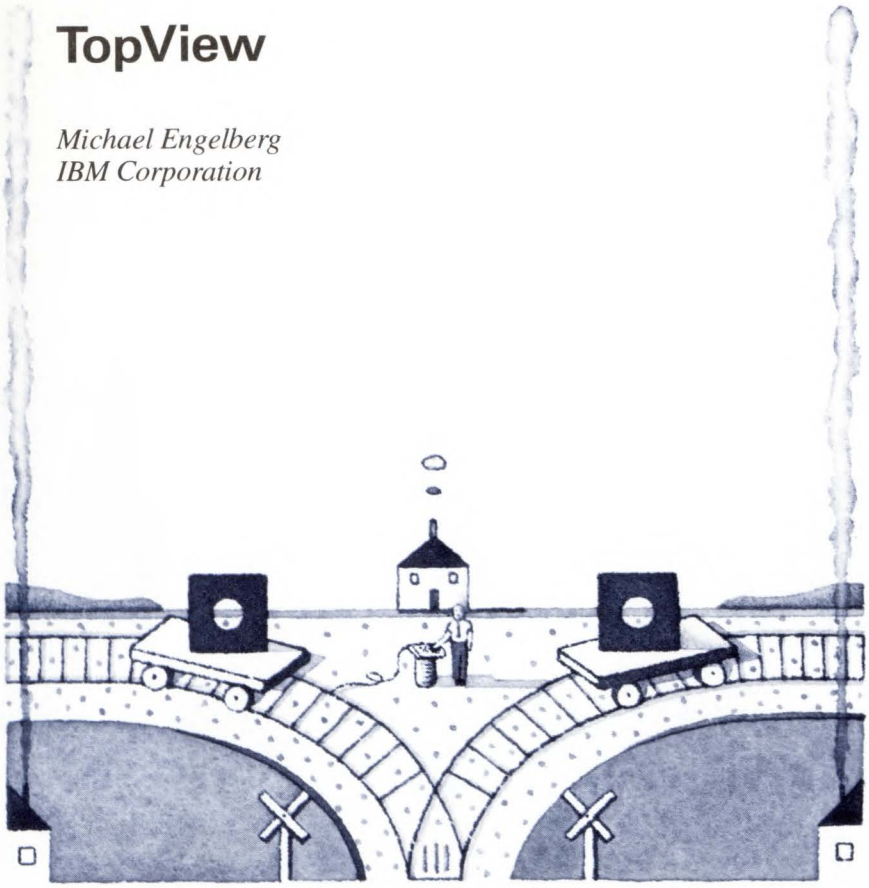
Please send your comments to:

Editor, *Exchange*
IBM Corporation (2900)
P.O. Box 3022
Boca Raton, FL 33431-0922

Let's hear from you!

TopView

Michael Engelberg
IBM Corporation



TopView provides a multitasking and windowing environment for software applications that run on an IBM Personal Computer AT, IBM Portable Personal Computer, IBM Personal Computer XT or IBM Personal Computer. TopView lets you run more than one program at the same time, and view screens from more than one program on your display device. TopView also lets you transfer data between the programs you are running.

In the TopView environment, you can run different applications, or you may choose to run multiple copies of certain programs. TopView lets you switch between applications, so that you can work within one application while others are running.

All of TopView's facilities are available to applications that are specifically designed to run with TopView, i.e., programmed using the TopView applications programming interface. In general, you can take advantage of only a subset of TopView facilities when running existing applications. At a minimum, the facility of switching from one application to another is supported for existing applications.

TopView comes with an on-line tutorial that teaches you how to use it. In addition, a series of on-line Help screens offers additional assistance.

TopView requires DOS 2.00, 2.10, 3.00 or 3.10.

TopView Operating Environment and Programmer's ToolKit

TopView consists of two separately priced products. The first is the TopView operating environment itself. The second is the TopView Programmer's ToolKit, containing the system interfaces and programming tools for developing applications that will take full advantage of TopView features.

This article has three parts. Part 1, which runs this month, is about the TopView operating environment (hereafter called TopView). Part 2 features the TopView Programmer's ToolKit. Part 3 covers writing TopView-compatible applications.

Productivity Enhancements

TopView enhances your productivity by:

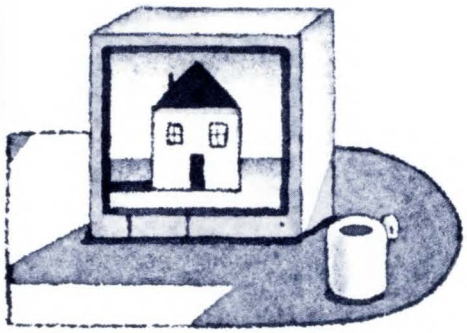
- Letting you control the flow of work you give your Personal Computer
- Making interruptions easier to handle
- Eliminating the need to end one application before starting another
- Producing more total work in a period of time by running overlapped applications

Major Features of TopView

Major features of TopView are:

- *Window-Oriented:* A window is a rectangular area that an application uses on the display screen. A window can occupy the entire screen or a portion of the screen as small as a single character. Therefore the screen can display several application windows at the same time.

Most existing application programs that run in the TopView environment were de-



signed to run as stand-alone programs rather than as part of a multitasking environment. Therefore most existing applications have a single window that covers the entire screen. However, if an application follows certain guidelines for behavior, and it does not write directly to the video buffer, then TopView can perform windowing operations such as moving and sizing (described later).

- *Menu-Driven:* A menu is a list of items, contained in a window, from which you can make selections. TopView uses pop-up menus that provide access to all TopView functions. After a menu pops up on the screen, you can select from the menu with the help of a pointing device such as the keyboard or a mouse. After you make your selection, the pop-up menu disappears.

  Most pop-up menus are under user control—you decide whether to make a menu appear. Some pop-up menus are controlled by TopView, and appear when TopView needs additional information or wants to pass information on to you.

  When a pop-up menu appears, its window occupies only a portion of the screen, temporarily overlaying, but not destroying, the previous contents of that portion of the screen. When a pop-up menu disappears, the previous contents, which had been overlaid by the pop-up window, once again appear.

- *Multitasking:* In the TopView environment, under the control of a single user, several application programs can run at the same time. TopView accomplishes this using a technique known as time slicing. With time slicing, TopView switches rapidly from one program to another so that it seems as if each program is running in the computer by itself.

  To permit multitasking while letting you interact with the computer, TopView uses the concept of a foreground and a background.

  At any given time, only one application program is running in the foreground. The foreground program is the only one that you can interact with. Within the window of the foreground application you can do things such as entering data, selecting items, windowing and editing.

  One or more additional programs may be executing simultaneously in the background. However, you cannot interact with background applications.

  In the TopView environment you can manage as many application programs as will fit in your computer's memory.

  The windows of all applications that are running concurrently are updated as directed by the applications, even when the applications are running in the background.

  You can switch between foreground and background applications by either making selections from a pop-up menu or quickly pressing the keyboard or mouse buttons twice to cycle from one application to the next.

  The window of the foreground application has a double-line border. The windows of all background applications (unless hidden or suspended) have single-line borders. However, when an application uses the full screen, its border does not appear.

  Windows can overlay other windows without destroying the contents of the overlaid windows.

  Note: Some existing software applications cannot run in the background.

- *DOS Services:* Many Disk Operating System (DOS) commands can be executed within TopView by selecting the DOS Services program.

  DOS Services uses multiple pop-up menus from which you can select and perform the more frequently used DOS commands:

Copy, Print, Type, Rename and Erase. With DOS Services you also can list directories or portions of directories, and sort them by name, extension, size and date/time.

Because you can perform the DOS commands supported under DOS Services without exiting TopView, the other programs you have started can continue to run in the background.

DOS Services also provides a way to perform additional DOS commands. By selecting the DOS Services Other option, you can specify DOS commands other than those listed above. Some of the additional commands are called "external" commands, and they require access to the DOS command files.

The complete list of DOS functions that TopView can perform is shown in Table 1. To use the external commands (those marked with an asterisk), you should use the DOS PATH command to specify the location of the DOS command files before you start TopView.

Table 1. DOS Commands That TopView Can Perform

*COMP	ERASE	TIME
COPY	*FIND	*TREE
DATE	MKDIR	TYPE
DIR	PRINT	VER
*DISKCOMP	RENAME	VOL
*DISKCOPY	*SYS	

TopView Terminology

Here is some terminology associated with TopView:

- *Cursor:* The cursor is the blinking underscore character that appears inside the window of the foreground application. The cursor usually marks the location where you enter information.
- *Pointer:* The pointer is the solid square character used by TopView itself. (In graphics mode, discussed later, the pointer is an arrow.) You use the pointer to perform TopView functions such as selecting from a pop-up menu, moving and sizing windows, and marking information to be transferred.
- *Pointing Device:* A pointing device is any device that can be used to control the pointer on the screen. All pointing devices in TopView are interrupt-driven; they require interrupt handlers to tell the system what they are doing. TopView handles pointing device interrupts using programs called pointing device drivers. In addition to handling interrupts, each pointing device driver also translates the unique characteristics of its pointing device into a standard format usable by TopView.

  TopView provides pointing device drivers for the keyboard and for the customer-supplied mice listed under the topic "Mouse" below.

  You may want to use other types of pointing devices, such as a graphics tablet, touch-sensitive display, joystick, track ball, touch pad, or cursor keypad. Perhaps you may want to design your own pointing device. You can use any pointing device as long as a TopView-compatible pointing device driver program exists for the device. You may write a pointing device driver program using the guidelines in the TopView Programmer's ToolKit reference manual.

  When you initially set up your TopView system, you tell TopView which pointing device you are using. You can then use only that pointing device. If you have a mouse, you cannot use the arrow keys on the keyboard to move the pointer; however, you can still use the arrow keys to move the cursor (if one exists), and you can use other keyboard keys to enter data.

  To change from one pointing device to another, you will have to set up your TopView system again.
- *Mouse:* A mouse is a pointing device that controls movement of the pointer. Moving the mouse on a flat surface such as a desk will move the pointer in the same direction on the screen. You can move the mouse, and therefore the pointer, in any direction.

  TopView supports four specific customer-supplied mice:
  - —Microsoft Mouse for IBM Personal Computers™ (Parallel Interface), part number 037-099
  - —Microsoft Mouse for IBM Personal Computers™ (Serial Interface), part number 039-099
  - —PC Mouse by Mouse Systems,™ Inc., part number 900120-214 (serial interface)
  - —Visi On Mouse by VisiCorp,™ Inc., part number 69910-1011 (serial interface)

  IBM has tested the use of these devices with TopView functions as of the date of announcement of TopView. However, IBM does not endorse or recommend one non-IBM product over another, and does not warrant these devices in any way.

The use of a mouse is optional. The functions of the buttons on a mouse also are available on the keyboard by using the Ctrl, Home (button 1), PgUp (button 2), Alt (button 3), and cursor control keys.

TopView Menus

The first menu you see after the welcome window is the Start-a-Program Menu, referred to as the Start Menu in this article.

The Start Menu lists the names of application programs that you have chosen to include in your TopView environment. You control the choices that appear in this menu (except that DOS Services is already listed).

Topview places a solid square in front of all program names that you can select from the Start Menu. (In all Topview menus, a solid square appears in front of the items you can select.)

You can start any program listed in the Start Menu by selecting it with the pointer.

To add the name of a program to the Start Menu, you select the option called "Add a Program to Menu." Also, to delete the name of a program from the Start Menu, you select the option called "Delete a Program from Menu." The Add Menu and the Delete Menu will be discussed later.

The main TopView menu has not yet appeared. You can see the main menu at any time by pressing either the Alt key on the keyboard, buttons 1 and 2 together on a two-button mouse, or button 3 on a three-button mouse.

The main menu, called the TopView Menu, allows you to select any of the TopView functions.

The functions available in the TopView Menu are:

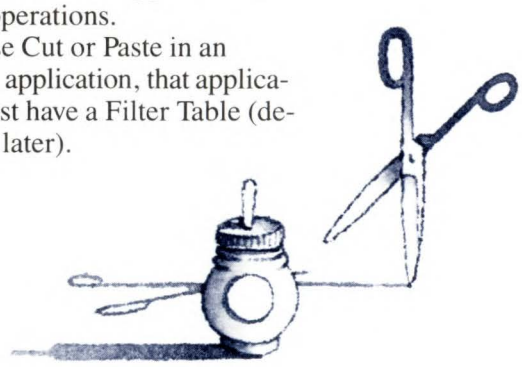
- *Scroll:* You can use the Scroll function to see the portion of the application's data that is outside the window. Within certain existing applications, by invoking Scroll and moving the pointer, you can display any portion of the application's data that has been written to the window.
- *Window:* You can do several things to a window. When you select the Window function, another pop-up menu, called the Window Menu, appears. Inside the Window Menu are these choices:
  - —*Move:* You can move a window to any convenient place on the display screen.
  - —*Size:* You can increase or decrease the size of a window. The normal size window for an application is the full screen, but the window can be reduced to any usable size.
  - —*Zoom:* A window that has been reduced in size can be expanded to its full size by using the Zoom function.
  - —*Unzoom:* After you have zoomed a window, you can restore it to its reduced size by using the Unzoom function.
  - —*Hide:* You can remove a window from the screen by using the Hide function. When you hide a window, you do not stop the processing being done by the application. To bring back the window you have hidden, you must bring up the main TopView Menu and switch to the application in the hidden window. The window will then come back to the screen as the foreground application.

    Note: If the application you are hiding has multiple windows, all windows belonging to that application are hidden.

    You would likely use the Hide function if you have several applications running concurrently and you do not need to see them all on the screen.
- *Scissors:* The Scissors function lets you transfer information between or within application windows. To use Scissors you choose from these functions:
  - —*Copy:* With the Copy function, you can copy information from the foreground window to a storage buffer maintained by TopView, while retaining the information in the foreground application.
  - —*Cut:* The Cut function is similar to Copy, except that the original information in the foreground application is deleted.
  - —*Paste:* After a Copy or Cut has placed some information into the storage buffer, Paste brings that information out of the buffer and into the current foreground application.

The Copy function is available in all non-graphics applications. The Cut function is available in applications that support block delete operations.

To use Cut or Paste in an existing application, that application must have a Filter Table (described later).



- *Help:* You can bring up the TopView Menu and select Help when it is available. TopView and the TopView Programmer's ToolKit have a comprehensive set of help screens. When you select Help, TopView will display the particular help screen that you need. You are now able to bring up additional help screens, or to quit from the help screens and return to your application.

Table 2 lists the Help screens available in TopView. Part 2 of this article will carry a table listing the Help screens available in the TopView Programmer's ToolKit.

Table 2. Help Screens in TopView

Programs	**Alarm Clock**
Start a Program	Set Time
Add a Program to the Start-a-Program Menu (5 Help screens)	Set Date
Delete a Program from the Start-a-Program Menu (2 screens)	Set Alarm
Change Program Information (3 screens)	**Calculator**
DOS Services	Correspondence between calculator and keyboard keys
Copy (2 screens)	Display Window
Print (2 screens)	Memory Window
Type	Tape Window
Rename (2 screens)	Change Sign (+/-) Key
Erase (2 screens)	Clear (CLR) Key
Other (2 screens)	Exchange (EXC) Key
Print from Other (2 screens)	Percent (%) Key
Directory	Recall (RCL) Key
Scroll	Store (STO) Key
Sort By	Sum (SUM) Key

- *Suspend:* The Suspend function temporarily halts the processing of an application, and it removes that application's window from the screen. To resume processing later, you will have to bring up the main TopView Menu once again, and switch to the suspended application, which then becomes the foreground application.
- *Quit:* The Quit function lets you stop running an application. When you quit an application, that application's name is removed from the Switch Menu (described next). To resume that application later, you will have to bring up the Start Menu and select that application.
- *Switch:* The Switch function lets you move from program to program. The Switch Menu contains the list of programs that were started in your current TopView work session. When you select a program from the Switch Menu, it immediately becomes the foreground application.

Note: You do not need to use the Switch Menu to move from program to program. You also can cycle between programs by pressing twice, in rapid succession, the button(s) you use to access the TopView main menu. The order in which you cycle through programs is determined by their order in the Switch Menu.

From the Switch Menu you can determine if a program is running, hidden or suspended. In the menu, a hidden program is preceded by a dash and a suspended program is preceded by an asterisk.

- *Programs:* When you select the Programs function, the Start Menu once again appears. You can now start a program or use the Add a Program to Menu, Delete a Program from Menu, and Change Program Information functions. (These functions will be discussed following the explanation of how TopView manages applications.)
- *Exit:* With the Exit function, you exit TopView and return control to DOS. You can exit only after you have quit all applications that you have been running.

How TopView Manages Applications

To manage the application programs that run in its environment, TopView maintains information about the unique operating characteristics of each application. TopView maintains this program information in its Consolidated Program Information File, Program Information Index File, and application Filter Tables (optional).

- *Consolidated Program Information File:* Many existing IBM Personal Computer software application products can run in the TopView environment. Program information for these supported applications is contained in the TopView Consolidated Program Information File (CPIF). Its file name is TV.PIF.

Each record in the CPIF contains information about one application. This information consists of:

—Program title
—Full pathname of the program's start-up file (either a .COM or .EXE file)
—Program parameters
—Location of data files (default drive and directory)
—Memory requirements (minimum, maximum and system)
—Type of display screen, e.g., 80 x 25 color
—Number of screen pages the application uses
—Window size and offsets
—Shared program information
—Range of swapped software interrupt vectors
—Program behavior characteristics

- *Program Information Index File:* To reduce access to the larger CPIF, TopView uses a Program Information Index (PII) File. Its file name is TV.PII. There is a one-to-one correspondence between records in the CPIF and records in the PII File. For each application the PII File contains the name of the program file and its minimum and system memory requirements.

  TopView uses the minimum and system memory requirements in the PII File to determine whether there is sufficient memory available to start a program. If enough memory is available, TopView places a solid square in front of the program's name in the Start Menu. The solid square tells the user that a program can be selected. The PII File is used in building the Start and Add Menus.

- *Filters:* Application Filter Tables, in conjunction with the TopView Filter Program, let you use the Scissors function to move information within a window or to transfer information from one window (the sending window) to another (the receiving window).

  You must first use either the Copy or Cut option to mark the data to be transferred. TopView then transfers the data from the sending window into an internal buffer. To put the data into the receiving window, you switch to the receiving application, select the Scissors feature, select the Paste option, move the pointer to where you want to place the data, and press button 1 on the pointing device. TopView then writes the information from its internal buffer to the receiving application's keyboard buffer. By creating these keystrokes, TopView enters data into the receiving application just as though you had used the keyboard to enter the data in insert mode.

  If an application was not programmed to handle Scissoring, then in order to handle it the application will need a Filter Table that supports scissoring.

  Included on the TopView diskette are Filter Tables for some existing IBM software products. If you intend to add applications other than these to your TopView environment, and you want your applications to perform functions that require filtering, you will have to supply your own Filter Table for each such application. Application developers should include Filter Tables in their products. (Guidelines for writing application Filter Tables are given in the TopView Programmer's ToolKit reference manual.)

Adding a Program

Let's assume that a Program Information File has been created for an application you wish to run in the TopView environment. Before you can run that application, you must add its name to the Start Menu.

You do this by displaying the main TopView Menu and selecting "Add a Program to Menu." TopView then displays the Add Menu, which contains the names of the existing IBM Personal Computer software products that can run in the TopView environment. (If an application runs with TopView but is not listed in the Add Menu, TopView gives you a way to add that application as well.)

When you select an application that is listed in the Add Menu, the name of that application is added to the Start Menu. For a listed program, TopView already knows the name of the program file itself (e.g., for Personal Editor it is PE.EXE). Also, TopView already has a record for that program in both its Consolidated Program Information File, TV.PIF, and its Program Information Index File, TV.PII. However, TopView has to ask you where it can find the program file. After you respond with a complete path to your program file, TopView places your information into the program information record.

A program not listed in the Add Menu can be added by using the Other option in the Add Menu. When you select Other, these things happen:

- TopView asks you for the location of your program file. After you respond, TopView goes to that location on diskette or fixed disk.
- TopView searches for all Program Information Files that exist at the location you specified. (A Program Information File must have an extension of .PIF.)
- TopView puts together a menu consisting of the names of the programs found in the Program Information Files. You then select which program(s) you want to add to the Start Menu.
- After you select the program(s) you want to add, TopView adds the corresponding Program Information Files as records in the files TV.PIF and TV.PII.

  (This is the process TopView uses to add the programs from the TopView Programmer's ToolKit diskettes. You need only tell TopView the location of the ToolKit programs. TopView then finds six Program Information Files from diskette #1 (or two from diskette #2), builds a menu consisting of their names, and lets you select the program(s) you want to add to your TopView environment.)
- If TopView finds no Program Information File at the location you specify, then TopView builds its own program information by asking you to supply the name of the program file and the minimum amount of memory it requires.

  However, the Program Information File record needs more information than TopView asks you for. The remaining information is default information furnished by TopView itself. You can review and change the contents of the Program Information File record by going to the Start Menu, selecting Change Program Information, and selecting the name of the program whose information you want to change.

Deleting a Program

If you no longer want to have an application program in your TopView environment, you can delete it from the Start Menu.

To do this, bring up the Start Menu, choose the "Delete a Program from Menu" option, then choose the name of the program you want to delete. Before TopView does the deletion, it asks you for confirmation of your request.

If the deleted program was originally listed in the Add Menu, TopView retains the program's information in TV.PIF and TV.PII. However, if the deleted program was not listed in the Add Menu, TopView also deletes the program information in TV.PIF and TV.PII.

Changing Program Information

At any time you can change the information that TopView maintains for your application program. To change program information, bring up the Start Menu, select the Change Program Information option, then select the name of the program whose information you want to change.

You will then see a full screen of information about your program. You can step through the screen, make any changes you wish, and then tell TopView either to save your changes or to ignore them. (You would select the ignore option if you know you have made errors in your new information.)

Some of the more frequent reasons for changing program information are:

- Changing the location where TopView can find your program file. You will change the location if you acquire an additional diskette drive or fixed disk drive, or if you create new subdirectories on your fixed disk that must now be included in the path that points to the program file.
- Changing the specified memory requirements. You may need to increase or decrease the minimum memory requirement for your application.

  If you added a program that was not listed in the Add Menu and you did not know how much memory it needs, TopView gave that program a default minimum memory of 52KB. If this is not enough memory, you will have to increase the minimum.

  On the other hand, if you are running many applications at once, and you are uncertain whether you can start another application, you may wish to reduce (if possible) the minimum memory required by one or more applications.

  Check your application's documentation for the minimum amount of memory required to run the application. If no information is available, an iterative process, during which you try several values, may help you determine the actual minimum memory required. However, an inaccurate minimum memory requirement may inhibit your application's performance.

Graphics Mode

TopView supports graphics applications. Both medium-resolution color (320 x 200 4-color pixels) and higher-resolution black-and-white (640 x 200 2-color pixels) video modes can be used for graphics applications.

Most graphics applications write directly into display adapter memory, so they can run only in the foreground with a full-screen window. You can start several graphics applications at one time, but they will not run as background tasks. Also, graphics applications cannot be windowed; however, a graphics application may be able to be windowed while it is in text mode.

Two Useful Programs

TopView comes with two simple but useful application programs, Alarm Clock and Calculator. These programs were created as samples to illustrate the functions of TopView. You can add them to your Start Menu and start them as you would any other application.

The Alarm Clock program allows you to set the date, time, and up to five alarms. After you set an alarm, the computer beeps when the alarm time occurs.

The Calculator program lets you add, subtract, multiply, divide, and find percentages and square roots of numbers. It also provides a worksheet that displays your calculations, and memory that stores them.

Both of these programs come with a set of Help screens that teach you how to use them.

TopView On-Line Tutorial

The TopView package comes with a tutorial diskette. You should take this on-line tutorial after you see the welcome window and before you use TopView.

The tutorial covers these subjects:

TopView terminology
Getting help when you need it
Adding applications to your TopView environment
Starting your applications
Switching between applications that have been started
Windowing
Scissoring
Ending an application
Using DOS commands

Existing Software that Runs With TopView

Software products supported by TopView are listed in two TopView Application Guides, one for IBM software and one for software available from non-IBM sources. The application guides are available free of charge to registered licensees. They also are distributed to Authorized IBM Personal Computer Dealers and IBM marketing representatives.

TopView Product Contents

The TopView product comes with:

- One program diskette
- One on-line tutorial diskette
- TopView User's Guide
- TopView Quick Reference Card
- TopView Registration Card

TopView Hardware Requirements

Following is a list of the computer hardware necessary to accommodate the TopView environment.

- An IBM Personal Computer AT, IBM Portable Personal Computer, IBM Personal Computer XT or IBM Personal Computer.
- A minimum of 256KB or Random Access Memory (RAM). This minimum memory configuration will allow TopView to run a program that requires up to 80KB for its execution when used outside TopView and with DOS 2.00 or 2.10 (68KB with DOS 3.00 or 3.10).
- Two double-sided diskette drives, or one double-sided diskette drive and a fixed disk.
- An 80-column display and its corresponding adapter card. The IBM Monochrome Display requires the Monochrome Display and Printer Adapter or the Enhanced Graphics Adapter. The IBM Color Display, or a compatible monitor, requires the Color/Graphics Adapter or the Enhanced Graphics Adapter. The IBM Enhanced Color Display requires the Color/Graphics Adapter or Enhanced Graphics Adapter. The IBM Professional Graphics Display requires the Professional Graphics Adapter running in emulator mode.

  Note: TopView supports the Enhanced Graphics Adapter and the Professional Graphics Adapter only when they are configured to run in compatibility mode. The extended graphics modes of both the Enhanced Graphics Adapter and the Professional Graphics Adapter are not supported by TopView.

  When you use the IBM Professional Graphics Adapter, you must configure TopView to use the keyboard or the parallel mouse pointing device. TopView does not support the serial mouse with the Professional Graphics Adapter.

The above list constitutes the basic hardware requirements for TopView. However, the following hardware is recommended for optimal performance and usability:

- A Personal Computer AT or XT that has a fixed disk and one double-sided diskette drive.
- As much as 640KB of memory. Larger memory is necessary in order to run several large programs at the same time; to run programs that have several windows; and to copy or cut a large block of text using the TopView Scissors function.
- An IBM Graphics Printer or other compatible printer, and a Printer Adapter.

* * *

Part 2 of this article, appearing next month, features the TopView Programmer's ToolKit.

Using TopView: a Scenario

Here's an example of how you might use the TopView environment. The following scenario uses two hypothetical applications, a word processor and a spreadsheet, which are assumed to run with TopView and to perform TopView functions.

After you start DOS and TopView, you will see a screen welcoming you to TopView. The screen offers you the choice of taking the on-line tutorial or continuing; we shall continue here.

If this is the first time you are using TopView, you should now add your word processing and spreadsheet applications to the list of applications that you want to have in your TopView environment.

To start your first application, display the Start-a-Program Menu and select the word processor. The application will come up on the display as the foreground application.

In this example, assume that you are using the word processor to write a report, and you decide to enhance your report by adding a table. In your judgment, the best way to create the table is to use the spreadsheet application. You also want to see the word processing window while you are working in the spreadsheet application.

To have both applications visible on the screen simultaneously, you must first display the Window Menu, reduce the current word processing window down to a smaller size, and move the smaller window to a convenient area on the screen.

You can give the word processor a task to perform while it is in the background. Examples of possible tasks might be to print out the report in its current state, or reformat the report.

It is now time to start the spreadsheet application. You do this by bringing up the Start-a-Program Menu and selecting the spreadsheet application.

(If you are not sure how to accomplish all of this, you can bring up the on-line Help facility or refer to the TopView Quick Reference Card.)

After you start the spreadsheet application, it becomes the foreground application, occupying the active window. You can reduce the spreadsheet window to co-exist with the word processing window, or you can let the spreadsheet window temporarily overlay the word processing window. For this example assume you want both windows to co-exist. You will now have to reduce the size of the spreadsheet window.

Because it is the foreground window, the spreadsheet window is bordered by a double line, and the word processing window by a single line. Meanwhile the word processor is running in the background.

If you forgot the name of the file that contains your spreadsheet data, you can find it by starting DOS Services as your foreground application. Now three applications have been started—the word processor, the spreadsheet, and DOS Services. From the DOS Services menu you can select a directory search to find the file name Then you can quit DOS Services if you desire, or you can bring up the Switch Menu, select the spreadsheet application, and create the table.

After the table is created, you can copy it from the spreadsheet application to the word processor application by using the TopView Scissor functions.

You can now quit the spreadsheet application. This will remove the spreadsheet application from memory, take its window off the screen, and take the application's name out of the Switch Menu; however, the application's name will remain in the Start Menu. You have not yet quit the word processing application, so it returns to the foreground.

Your word processing data file now includes a report created by the word processor and a table created by the spreadsheet application. You can use the word processor once again to print the report.

Case Conversion in BASIC

Brad Whitlock
Raleigh Personal Computer Club
Research Triangle Park,
North Carolina

I wrote a quiz program in BASIC which contained a long character string (specifically, a Bible verse). I wanted to maintain the correct capitalization in the data file so that it would display correctly, but I didn't want to require the person taking the quiz to type in the correct capitalization. Therefore, before I could compare the person's entered answer with the correct answer, I had to convert them both to all upper (or lower) case.

Initially I coded this conversion in BASIC, with a loop going through both strings one character at a time, testing for a lower case letter, and replacing it with the corresponding upper case letter. Waiting seven or eight seconds for all this to happen may not seem long, but this is the age of instant results. Such a simple function seemed to be an ideal introduction to 8088 machine language. I took out my 8088 book, and came up with the USR function in Figure 1.

The BASIC program that includes this USR function is shown in Figure 2.

Notice that, because of the way integers are stored in low/high form, each of the byte pairs in the DATA statements is transposed when assembled. For example, the first instruction is entered as the number &H033C (which equals 828), but it is assembled as 3C03.

```
CSEG        SEGMENT BYTE PUBLIC "CODE"
CONVERT     PROC    FAR
            ASSUME  CS: CSEG
            CMP     AL,03          ; is the parameter
                                   ; a string?
            JNE     OUT            ; no, then don't do
                                   ; anything
            MOV     SI,DX          ; copy address of
                                   ; string descriptor
            CMP     BYTE PTR [SI],0 ; is the string null
                                   ; (length 0)?
            JE      OUT            ; yes, then no con-
                                   ; version needed
            MOV     CL,0[SI]       ; put string length
                                   ; in C register
            MOV     CH,0           ; (clear high order
                                   ; half)
            INC     SI             ; bump to address
                                   ; of string
            MOV     SI,0[SI]       ; point SI to the
                                   ; string
NEXT:       CMP     BYTE PTR [SI],061H ; compare against
                                   ; "a"
            JL      SKIPIT         ; don't convert if
                                   ; lower
            CMP     BYTE PTR [SI],07AH ; compare against
                                   ; "z"
            JG      SKIPIT         ; don't convert if
                                   ; higher
            AND     BYTE PTR [SI],0DFH ; convert to upper
                                   ; by turning off x'20'
SKIPIT:     INC     SI             ; look at the next
                                   character until
            LOOP    NEXT           ; all characters are
                                   ; looked at
OUT:        RET                    ; go home
CONVERT     ENDP
CSEG        ENDS
            END
```

Figure 1. USR Function

I have indicated what changes are needed to cause conversion to lower case instead of upper. In the assembly code, this corresponds to checking for the range A-Z and turning on the X'20' bit with an OR instruction if the character is in the range. Finally, note that this is not a well-behaved function. Besides returning the function result of an all upper case string, it also changes the original string to all upper case. If you need to keep a mixed-case copy of the string, make a copy before invoking this function. I suppose it more properly should be a CALLed subroutine, but the USR interface looked much simpler to code around.

While debugging this code, I was reminded of just how unforgiving computers can be. I had heard many times that it is important to use the VARPTR function immediately before you need the value, since variables move around. The code I had was:

```
100 DEF SEG
110 DEF USR0 = VARPTR (S%(0))
120 UPPERCASE$ = USR0(MIXED-
CASE$)
```

```
100 ' Reserve space and set up the subroutine.
110 '
120 DIM S%(17)
130 FOR I=0 TO 17
140  READ S%(I)
150 NEXT I
160 '
170 '
180 ' Make sure we are pointing to BASIC's data space.
190 '
200 DEF SEG
210 '
220 '
230 ' Get a string to convert the case of. Then print the
240 ' original string, the results of the user function call,
250 ' and the original (altered) string after the function call.
260 '
270 INPUT "Give me a string ('quit' to stop)";S$
280 PRINT "Before the case conversion —> ";S$
290 DEF USR0=VARPTR(S%(0))
300 PRINT "Result of the function call —> ";USRO(S$)
310 PRINT "After the case conversion —> ";S$
320 PRINT
330 '
340 '
350 ' See if we were just told to stop. If not, go back for more.
360 '
370 IF S$<>"QUIT" THEN GOTO 270
380 END
390 '
400 '
410 ' Here is the machine language code, waiting to be read into
420 ' an integer array. Note the four values to change to select
430 ' conversion to upper case vs. conversion to lower case.
440 '
450 DATA &H033C, &H1E75, &HF28B, &H3C80, &H7400
460 DATA &H8A17, &HB50C, &H4600, &H348B, &H3C80
470 DATA &H7C61 :' use &H7C41 for lower case, &H7C61 for upper.
480 DATA &H8008
490 DATA &H7A3C :' use &H5A3C for lower case, &H7A3C for upper.
500 DATA &H037F
510 DATA &H2480 :' use &H0C80 for lower case, &H2480 for upper.
520 DATA &H46DF :' use &H4620 for lower case, &H46DF for upper.
530 DATA &HF0E2, &H00CB
```

Figure 2. BASIC Program

This code kept hanging my PC. But, when I defined UPPERCASE$ before statement 110, everything worked well. I had been assuming that the function would be evaluated before the variable was allocated, but such is not the case. It allocated UPPERCASE$ before calling my USR function, which made my function move to a different place than it was when I got the VARPTR address. When BASIC went to execute my subroutine, it found gobbledegook.

The lesson is, don't even think about any new variables between getting the address and getting back from the subroutine.

Clearing Part of the Screen in BASIC

Andy Hopkins
PC Professional Users Group
Wilmington, Deleware

Want to clear a portion of your display screen? Here is a short routine to do this.

```
10 REM Data is machine language routine for
   CLEER%(R1%,C1%,R2%,C2%)
20 DATA &H8B55, &HB8EC, &H0600,
   &H07B7, &H768B, &H8A0C, &H8B2C
30 DATA &HA76, &HC8A, &H768B,
   &H8A08, &H8B34, &H676, &H148A
40 DATA &HCDFE, &HC9FE, &HCEFE,
   &HCAFE, &H10CD, &HCA5D, &H8, &H0
50 RESTORE 2: DIM CLEER%(21): FOR 1%
   =0 TO 21: READ CLEER%(1%): NEXT
```

Note: the word CLEER is used because CLEAR is a reserved word in BASIC.

The four variables R1%, C1%, R2%, C2% define the four corners of the area you wish to clear. The BASIC program that calls this routine should set R1% to the top line, C1% to the left column, R2% to the bottom line and C2% to the right column. Next, insert the following line in your calling program:

```
DEF SEG: CLEER=VARPTR(CLEER%(0)):
    CALL CLEER (R1%,C1%,R2%,C2%)
```

The area outlined by this box will instantly clear in the same way the CLS command clears the entire screen.

What's New in DOS 3.00 and BASIC 3.00

John Warnock
IBM Corporation

DOS 3.00

The IBM Personal Computer DOS 3.00 is a powerful extension of earlier versions of DOS. Its new functions make the Personal Computer more versatile and functional.

DOS version 3.00 features improved protection of files against unauthorized access. It also gives you more control over your PC environment by letting you specify more about attached devices, keyboards, and language support.

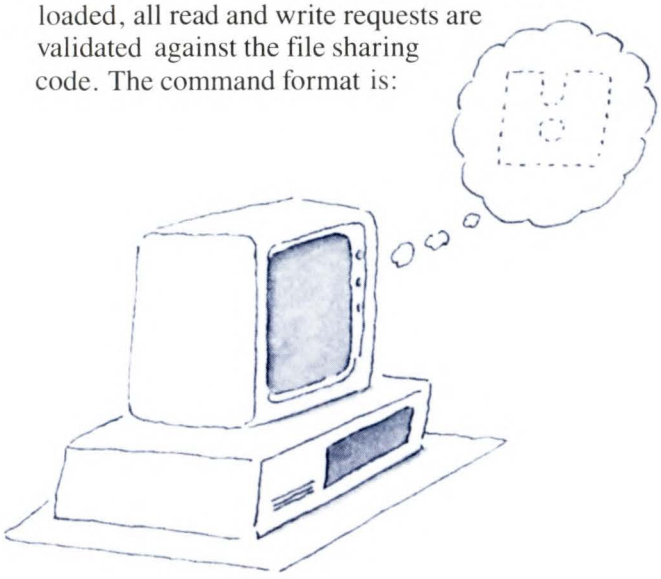
DOS version 3.00 is designed to support the IBM Personal Computer AT, IBM Personal Computer XT and IBM Personal Computer. DOS 3.00 accommodates up to 3MB of memory, treating anything below 640KB as regular memory and anything above 1MB as a virtual disk. DOS 3.00 also supports 1.2MB diskettes and 20MB disks along with the 16-bit channel and the higher speeds of the 80286 processor and 80287 Math Co-processor.

DOS 3.00 and BASIC 3.00 provide all the function and capabilities of earlier versions. DOS 3.00 requires 36K of memory compared to 24K for 2.10 and 2.00 and 12K for DOS 1.00 and 1.10.

DOS 3.00 includes the following enhancements:

- File sharing and protection on a file-by-file basis. File sharing is accomplished with two new DOS commands, SHARE and ATTRIB. Along with revised DOS calls, these commands allow you to set up an environment for sharing files between concurrent applications.

  In addition to loading the file sharing routines, the SHARE command establishes the number of files and sector locks allowed. When SHARE is loaded, all read and write requests are validated against the file sharing code. The command format is:



[d:][path] SHARE [/F:filspace][/L:locks]

where

[d:][path] is the drive and path where the SHARE command file is located.

[/F:filspace] allocates memory in bytes to hold the sharing information on each file. The default is 2048. Each open file requires 11 bytes in addition to the length of the file name.

[/L:locks] describes the number of sector locks you wish to allocate space for. The default is 20. Sector locks prevent two programs from simultaneously trying to change the same information in the same file.

To give your system special features, you can use the CONFIG.SYS file and special commands to define them. One of the things you can determine is the number of file control blocks (FCBs) that are available to the computer. One file may have several FCBs open if several programs are using it.

When you limit the number of FCBs available to DOS, you limit the amount of file activity it can do at any given time. When sharing is active, the number of file control blocks is strictly limited to the number specified in the FCBS statement in the CONFIG.SYS file (see the discussion of the FCBS command later). DOS will automatically attempt to re-use as many FCBs as possible.

An attribute is a characteristic of a file. While DOS automatically sets most file characteristics, you can set the access attribute yourself.

The ATTRIB command lets you set or remove a read-only protection on an individual file, or display its attribute. The command format is:

[d:][path] ATTRIB [+R|-R] [d:][path] filename [.ext]

where

[d:] and [path] are used to specify drive and path of the ATTRIB command file.

+R sets the attribute of the file to read-only.

-R removes the read-only protection.

[d:][path] filename [.ext] tells DOS which file you want to check or set the attribute for.

For example, to set the file SAMPLE.TXT to read-only, you would type:

```
A> attrib +r sample.txt
```

If you then type:

```
A> attrib sample.txt
```

the result is

```
R A:\SAMPLE.TXT
```

The "R" indicates that the file is read-only.

To remove the read-only protection, you would type:

```
A> attrib -r sample.txt
```

If you then type:

```
A> attrib sample.txt
```

the result is

```
A:\SAMPLE.TXT
```

The R is no longer present when read-only protection is removed.

- Background printing enhancements. You can now specify the drive and path for the PRINT command file when initiating the command. You also can specify a device and other additional system parameters when starting the command. The PRINT command now looks like this:

```
[d:][path] PRINT [/D:device][/B:buffsiz]
  [/U:busytick] [/M:maxtick][/S:timeslice]
  [/Q:quesiz][/C] [/T][/P] [[d:][path]
  [filename][.ext]...]
```

where

[d:][path] tells DOS which drive and directory contains the PRINT command file.

[/D:device] allows you to specify the output device, where device is PRN, LPT1, LPT2, LPT3, COM1, COM2, AUX, etc.

[/B:buffsiz] is the internal buffer size. The default is 512.

[/U:busytick] specifies the number of clock ticks that pass before PRINT says the print device is unavailable. Then PRINT gives up its timeslice. The default is 1.

[/M:maxtick] allows you to specify from 1 to 255 clock ticks for PRINT to use each time it prints characters on the print device. The default is 2.

[/S:timeslice] lets you set the timeslice value from 1 to 255. The default is 8.

[/Q:quesiz] determines how many files can be in the print queue. 10 is the default. You may select from 1 to 32.

[/C] allows you to cancel any number of specified print files. The next /P terminates the command.

[/T] terminates all queued print files and stops the file currently being printed.

[/P] lets you print any number of specified print files. The next /C terminates the command.

- Enhanced error recovery and additional error reporting facilities. DOS 3.00 checks for file control block availability, and reports exceptions when file sharing is active.

  DOS also checks to see if a file is in a locked condition, or if a sector could not be found.

  Sharing violations and write violations are also reported when they occur.
- Additional hardware support for 1.2MB diskette drives, non-volatile timer, and 20MB fixed disks. DOS 3.00 supports the new high capacity (1.2MB) diskette drive along with the 20MB fixed disk and the CMOS clock of the Personal Computer AT.

The single-sided, double-sided and high-capacity diskette drives are supported by DOS 3.00. While the single- and double-sided diskettes hold 40 tracks with 8 or 9 sectors per track, the high-capacity diskettes hold 80 tracks at 15 sectors per track.

The FORMAT command has been expanded to allow you to format single-and double-sided diskettes on a high-capacity drive. If no parameters are specified, FORMAT defaults to the maximum capacity of the drive. /1 continues to work for single-sided diskettes in double-sided or high-capacity drives. For formatting double-sided diskettes in a high- capacity drive, you would use the statement:

FORMAT A:/4

where

A: is the drive where the diskette is located.

/4 describes a double-sided diskette.

Note: If you format or write single- or double-sided diskettes using a high capacity drive, you may not be able to read the diskettes on single- or double-sided drives.

DOS version 3.00 also supports a 20MB fixed disk. The major change to provide this support was to increase the size of the entries in the File Allocation Table from 12 to 16 bits for disks with more than 4086 file clusters. DOS counts the number of clusters on the disk in order to determine what size the disk is. You should use the DOS function calls rather than interpret the File Allocation Table yourself.

- Screen dump facility improved to add support for additional display and printer interfaces. The GRAPHICS command now lets you specify the type of printer that the display copy will go to. This allows color print screen capability as well as printouts from graphics displays. The improved GRAPHICS command uses the following format:

[d:][path] GRAPHICS [printer type][/R][/B]

where

[d:][path] specifies the drive and path where the GRAPHICS command file is located.

[printer type] defaults to GRAPHICS if not specified, otherwise it should be one of the following:

COLOR1 for a color printer with a black ribbon

COLOR4 for a color printer with a red, green, blue, and black ribbon

COLOR8 for a color printer with a cyan, magenta, yellow, and black ribbon

COMPACT for the IBM Compact Printer

GRAPHICS for the IBM Graphics Printer

/R reverses the value of white to print black, otherwise black prints as black and white prints as white.

/B tells printer types COLOR4 and COLOR8 to print the background color as well, otherwise it is omitted.

Graphics also has been improved under DOS 3.00 with the GRAFTABL command. This command loads a table of the upper 128 ASCII characters into memory for use by the Color/Graphics Adapter. It is useful for special, foreign and box graphic characters. The command uses this format:

[d:][path] GRAFTABL

where

[d:][path] specifies the drive and path where the GRAFTBL command file is found.

- Linker program enhanced to support up to 1 MB of memory. The Linker (LINK) program can now produce a load module up to 1MB in size. Program segments and groups may be combined into classes as large as 1MB. The practical limit is 640 KB because DOS uses upper memory for display addressing and other features.

- VDISK RAM disk added to allow system memory to be allocated for a virtual disk. If your system has more memory than needed to run a program, you can tell the computer to use the excess memory as if it were an extra disk drive. This is called a virtual disk—it isn't really a disk, but to the system it looks like one.

  You access this feature through the DEVICE configuration statement in your CONFIG.SYS file. You must also have the VDISK.SYS file on your DOS diskette. The file contains the device driver that simulates a diskette drive using computer memory. Because it is in memory, a virtual disk operates much faster than either a disk or diskette.

  You can specify up to 24 virtual disks (less the number of your actual diskettes or disks) by providing a device statement for each virtual disk.

  Each virtual disk uses the amount of memory you specify, from 1K to the available memory of your machine. Each virtual disk also increases the size of DOS by 720 bytes, plus any disk buffers you may have. Virtual disks normally reside in low memory, but can be placed in the extended memory (above the 1MB boundary) of the Personal Computer AT.

  You can install a virtual disk by placing a statement similar to the following in your CONFIG.SYS file:

```
DEVICE=[d:][path] VDISK.SYS
       [bbb][sss][ddd][/E]
```

  where

  [d:][path] is the drive and directory path where the VDISK.SYS file is located.

  [bbb] is the (decimal) disk size in KB. The default is 64 KB. You can specify any value from 1 to the size of your system. VDISK will automatically reduce the size of the virtual disk to allow at least 64 KB of working memory in your system, and defaults to 64 KB if you specify more memory than your system has available.

  [sss] is the sector size—128, 256, or 512 bytes. If not defined, or incorrectly stated, it defaults to 128.

  [ddd] is the number of directory entries, ranging from 2 to 512. If incorrectly stated, or left off, the default is 64. One of the directory entries is used as a volume label for your disk. The rest are used for file names. VDISK will round the number of entries up to fill the nearest logical sector boundary. If your directory entries and File Allocation Table are larger than your virtual disk, then VDISK will automatically begin reducing your directory a sector at a time until it fits on the virtual disk.

  /E tells VDISK to use the extended memory of the Personal Computer AT. The device driver will reside in low memory. The virtual disk and each succeeding virtual disk will start at the 1MB boundary and go higher.

  For example, to set up a 320KB drive with 256-byte sectors and enough entries for 30 files, you could use:

```
DEVICE=VDISK.SYS 320 256 31
```

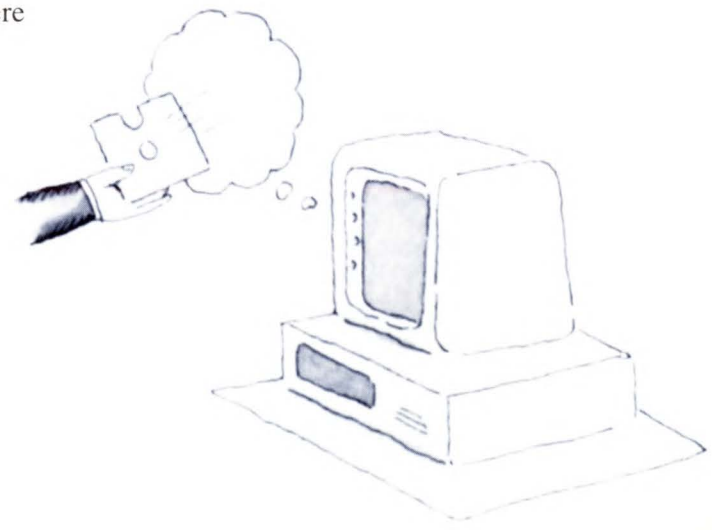
  VDISK uses one of the directory entries as a volume label, and rounds up the number of entries to the next sector boundary.

- Additional configuration statements for CONFIG.SYS file. In addition to the SHARE command and the enhanced DEVICE configuration statement, DOS 3.00 has other commands and configuration statements to expand your system configuration.

  COUNTRY lets you specify the date and time format for the country you desire. It also changes the currency symbol and decimal separator to match. The command format is:

```
COUNTRY = xxx
```

  where



xxx is a three-digit country code. The default, 001, is the United States. 14 other countries are supported. The country command does not translate DOS messages into other languages, however.

The FCBS configuration statement tells DOS how many file control blocks can be open concurrently. (File control blocks are used by some applications to read, write, create and delete files.) This is different from the number of files allowed, because one file may have several file control blocks associated with it when it is shared by more than one program.

The statement format is:

FCBS = m,n

where

m is the total number of files opened by file control blocks. It can be any number from 1 to 255. The default is 4. If a program tries to open more than this number of files while sharing is active, DOS looks for the least recently used file, closes it, and re-uses its FCB.

n is the number of file control blocks protected from automatic reclamation by DOS. The first n files cannot be reused by DOS unless they are closed by their applications.

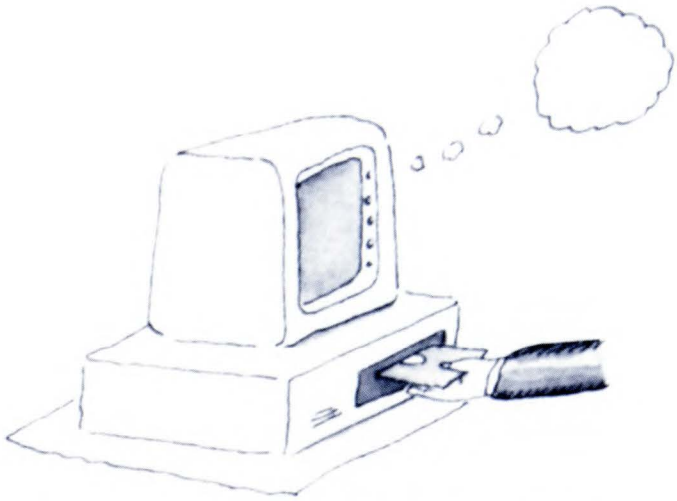
LASTDRIVE sets the maximum number of drives you may access. The format is

LASTDRIVE = x

where

x represents the letter (A-Z) of the last valid drive letter DOS may accept. It may not be less than the highest drive on your system.

- The BACKUP and RESTORE commands have been enhanced. You can now back up a diskette to another diskette, a diskette to a disk, and a disk to another disk, as well as a disk to diskettes. RESTORE supports these same combinations.



- User-selected keyboard type and date/time formats. KEYBxx loads special country support programs which replace the keyboard support resident in ROM BIOS. Each program uses about 2KB of memory. The command format is:

[d:][path] KEYBxx

where

[d:][path] specifies the drive and path where the keyboard support program is located.

xx represents one of five countries:

FR = France
GR = Germany
IT = Italy
SP = Spain
UK = United Kingdom

Only one keyboard program should be loaded at a time. It resides in memory, and can be removed only by rebooting the system. Each new keyboard program takes control from the previous one.

Pressing Ctrl-Alt-F1 returns control to the USA keyboard format resident in DOS. Ctrl-Alt-F2 switches control back to the country support keyboard program.

SELECT, like the COUNTRY and KEYBxx commands, allows you to select the keyboard layout and the date and time format. This command builds new CONFIG.SYS and AUTOEXEC.BAT files that contain the correct KEYBxx and COUNTRY commands.

The format is:

[d:][path] SELECT xxx yy

where

[d:][path] is the drive and path where the SELECT command file is located.

xxx specifies the three-digit country code for the COUNTRY command.

yy is the two-letter keyboard code for the KEYBxx command.

Acceptable values are:

Country	Country Code (xxx)	Keyboard Code (yy)
United States	001	US
France	033	FR
Spain	034	SP
Italy	039	IT
United Kingdom	044	UK
Germany	049	GR

- Greater user control of disk volume labels. LABEL lets you create, change or delete a disk volume label on a formatted disk without reformatting the disk. It allows for up to 11 characters like the /V option of the FORMAT command.

  The LABEL format looks like this:

  [d:][path] LABEL [d:][volume label]

  where

  [d:][path] specifies the drive and path where the LABEL command file is located. If the second [d:] is not specified, the default drive is assumed.

  [volume label] is the 11-character volume label. If you leave it blank, you will be prompted for it. If you leave it blank again, the volume label is deleted. Previously, the volume label could only be changed by the FORMAT command.

- New DOS function calls. To accommodate these new commands, new DOS function calls have been added, old calls have been changed, and file information has been changed.

  Byte 11 of each file's directory entry contains its attribute. If that value is 01H, any attempt to open the file as output (3DH) will result in an error.

  Several function calls are changed or added to take advantage of the new features of DOS 3.00. The list of new and changed function calls is:

Call	Description (* indicates new call)
38H	To get current country or set a new country. Allows you to set month/day/year format, time format, currency separators and date separators.
3DH	Open a file. Now includes sharing and inheritance flags. Inheritance means the file is passed to another program by the calling program.
44H	I/O device control. You can set the count of retries for sharing.
45H *	Duplicate a file handle.
59H *	Get extended error.
5AH *	Create temporary file. Uses extended error codes.
5BH *	Create a new file. Uses extended error codes.
5CH *	Lock/unlock file access. Prevents other programs from accessing a file.
62H *	Get Program Segment Prefix address. Returns segment address of currently executing program.

- BASIC 3.00 interpreter provides better access to DOS and device drivers through new commands and support of new hardware. BASIC now supports the 1.2MB diskette drive and the 20MB fixed disk drive. BASIC also allows you to execute DOS commands and other programs from within a BASIC program.

  Four new reserved words allow you to access unique device drivers specified by your CONFIG.SYS file.

- DOS 3.10 support planned in development of DOS 3.00. DOS 3.00 has been designed with the memory requirements of DOS 3.10 in mind. Therefore, memory size should not be a consideration in upgrading from DOS 3.00 to DOS 3.10.

Besides these enhancements, DOS 3.00 is repackaged to include an Application Setup Guide which aids users in installing additional applications on their system. Like DOS 2.10, DOS 3.00 comes with a User's Guide and a Reference Manual. The Technical Reference Manual is available separately. It contains new information not contained in the DOS 2.10 Technical Reference Manual, such as the new DOS calls, file attributes, and file sharing.

Each DOS 3.00 package comes with:

- A DOS 3.10 Upgrade Kit offer
- A BASIC 3.00 Upgrade Package Order form
- An Important Information sheet

To use DOS 3.00, your system must have one double-sided diskette drive and 96KB of memory. If you have a fixed disk, 128KB of memory is recommended. DOS 3.00 automatically loads from the A or C drive if present, and is a prerequisite for BASIC 3.00 Disk and Advanced versions. DOS also provides the I/O for the Pascal, FORTRAN, COBOL, and BASIC compilers and the assembly language programs they generate.

Compatibility

Many programs can be migrated to DOS 3.00 from earlier versions of DOS. They must not use reserved or unpublished DOS or BIOS addresses. Due to machine performance differences, they should not use loops for timing. They are relocatable and must not use absolute addresses.

BASIC 3.00

The IBM Personal Computer BASIC 3.00 is a more powerful version of earlier forms of the language. You can now exercise direct control over non-standard peripherals through device drivers. BASIC 3.00 gives you more information about hardware problems. It also gives you the ability to call DOS, procedures, and other programs from within BASIC. You can even pass information to those programs and procedures.

Major improvements in BASIC 3.00 include:

- Additional access to device drivers and DOS
- Changes to support Personal Computer AT hardware
- New commands to access DOS and device errors from within BASIC

Programs written under BASIC versions 1.00, 1.10, 2.00 and 2.10 are upward compatible with BASIC version 3.00. However, due to the increased memory requirements of DOS 3.00, you may need additional memory to continue to run some older BASIC programs.

Under BASIC 3.00, these additional commands are available:

- SHELL, which allows you to execute DOS commands and programs from within a BASIC program. When BASIC executes a SHELL command, it loads a second copy of COMMAND.COM into memory. It then executes the commands following the SHELL command. Control returns to the BASIC statement following the SHELL command.

  The procedure or program called by BASIC becomes a "child" of the program that called it. You can use the ENVIRON statement to pass parameters to the child. (See description of ENVIRON statement later.)

  For example, if you wrote a program to create a file called ONE, and then wanted to sort it into a file called TWO, you would type:

```
180 SHELL "SORT < ONE > TWO"
```

With BASIC 3.00, you can now exercise direct control over non-standard peripherals through device drivers.

Your computer would then execute the sort, and automatically return to execute the statement at line 190 when the sort ended.

Another example would be to suspend BASIC, or interrupt your program to check for which version of DOS you were using. You might include the instruction:

```
560 SHELL "COMMAND.COM"
```

or

```
560 SHELL
```

You will see the DOS prompt, and can type any DOS command except BASIC or BASICA. Type:

```
>ver
```

When you are finished using DOS you must type:

```
>exit
```

in order to resume your BASIC program.

You can also run BATch files from your program just as you would from DOS. To do this, you would say:

```
575 SHELL "AUTOEXEC.BAT"
```

Your BATch file could look like this:

```
DATE
TIME
EXIT
```

EXIT must always be the last command in the BATch file in order to return control to your BASIC program.

You also can go to DOS from within a BASIC session by simply typing:

```
SHELL
```

Your DOS session could look like this:

```
>dir
>exit
```

Immediately after the SHELL command, you should also issue a SCREEN statement followed by a CLS statement in order to restore the display.

Caution: Be sure to close any files that might be used by the child process, and reopen those files when the process ends. Otherwise, unexpected results may occur.

- IOCTL statement and IOCTL$ function, which allow BASIC to send control data to a device driver and to read input from the device. Device drivers reside within DOS. They provide the instructions that make a device (such as the printer) work. Drivers for displays, printers, communications lines, disks, and diskettes are included in DOS. You need not be aware of them.

  Sometimes you may want to connect a device to your Personal Computer that DOS does not support, such as a clock card or tape drive. DOS allows you to write your own device driver to control that machine, and to include the driver in DOS when you start your computer. You can even give that driver the name of an existing driver such as LPT1. Your LPT1 will then take precedence over the LPT1 that comes with DOS. In your BASIC program you open it like any other file.

Now you can use the ERDEV and ERDEV$ variables to get more information about hardware problems.

The IOCTL statement lets you send control information to any device driver you may have created. The form that the control information takes depends on the device driver you use. The format is:

```
IOCTL [#]filenum, string[; string] . . .
```

where

[#]filenum is the number of the device driver used in the OPEN statement.

string is the character expression containing the control information. A string cannot be longer than 255 characters. Multiple commands within the character expression can be separated by a semicolon.

The IOCTL$ function will read a data string from a device driver. You can use it to get device status information, or to make sure of configuration information. Its format is:

```
v$ = IOCTL$([#]filenum)
```

where

v$ is the name of the variable you want the information stored in.

[#]filenum is the number of the device driver used in the OPEN statement.

- ERDEV and ERDEV$, new device error variables that allow you to read INT24 error codes. DOS 3.00 provides more information about device errors. BASIC lets you use the ERR and ERL variables to get information about the BASIC errors you encounter, but it won't tell you about DOS error messages or conditions. Now you can use the ERDEV and ERDEV$ variables to get more information about hardware problems.

  ERDEV is a read-only variable. Like ERR, it contains an error number, except it refers to DOS errors instead of BASIC errors. You can access it via any one of the following instructions:

```
120 PRINT "Error: ";ERDEV
```

or

```
120 V = ERDEV: PRINT V
```

  ERDEV is 16 bits long. It holds the DOS interrupt 24 error code in the lower 8 bits, and the attribute word of the device header block in the upper 8. You can use the DOS Technical Reference manual to determine something about the error and the device that caused it.

  Like ERDEV, ERDEV$ is a read-only variable. Unlike ERDEV, ERR, and ERL, it is a string variable and contains the name of the device that caused the error. You may display the device name with any one of the following statements:

```
120 PRINT "Device: ";ERDEV$
```

or

```
120 V$ = ERDEV$: PRINT V$
```

  ERDEV and ERDEV$ are best used within error recovery statements like ERR and ERL. (You must still interpret them via the DOS Technical Reference manual to get the most use out of them.)
- ENVIRON statement and ENVIRON$ function let you modify and retrieve parameters from BASIC's environment table. BASIC 3.00 lets you call DOS, programs and BATch files from within a BASIC program. ENVIRON lets you set the stage before invoking child processes.

  For example, you might be running a program that resides in a directory called PROGRAMS, and want to leave BASIC and list directory of your C disk. You would use the statements:

```
100 ENVIRON "PATH=C:\"
110 SHELL "DIR"
```

  You might also want to remove the path setting completely when you are done. To do this you would say:

```
120 ENVIRON "PATH=;"
```

The semicolon removes PATH from the BASIC environment table and compresses the space.

  Just as ENVIRON will let you modify the environment table, ENVIRON$ will let you examine the values inside the environment table. ENVIRON$ will let you look at the environment table by parameter name or by number. The formats are:

```
v$ = ENVIRON$(parm$)
```

or

```
v$ = ENVIRON$(n)
```

where

parm$ is the name of the parameter as a string variable or character string you want to see

n is the number of the parameter you want to see

Besides these additional commands, the BASIC 3.00 Reference manual has new documentation for interfacing machine language subroutines with the BASIC language interpreter. The book discusses and provides numerous examples of special code to include in assembly language routines for the BLOAD command. There are also several examples of simple short cuts for assembly language routines.

The BASIC 3.00 Reference manual has been reorganized for greater ease of use and clarity. It now consists of a handbook which describes general information on BASIC, a Reference detailing the specific commands used by BASIC, and a Quick Reference card. The BASIC Reference manual is shipped with the Personal Computer AT System Unit. It will be phased in with the other system units. The PCjr has its own reference manual.

XOR (Exclusive Or) and Pixel Color Manipulation

John Schnell
New York Personal Computer, Inc.
New York, New York

One of the most interesting uses of XOR is in manipulating graphic pixels when using the PUT command. Many people who experiment with graphics are mystified by how the XOR option works. Perhaps a demonstration program and some binary tables can clarify the mechanics of this exotic operation.

Remember that any boolean operation works on the bit level, combining ones and zeros according to the rule of the chosen boolean operator. The result of any combination can be looked up in the appropriate logic truth table in your BASIC manual.

Let's look at the boolean table for binary XOR operation (Table 1), which is set up with T/F and 1/0 arguments. Notice when A and B are the same, the result is FALSE or 0. When A and B are different, the result is TRUE.

Table 1

A	B	A XOR B	A	B	A XOR B
F	F	F	0	0	0
F	T	T	0	1	1
T	F	T	1	0	1
T	T	F	1	1	0
(false/true)			(zeros/ones)		

Next, let's look at how color is represented on the medium-resolution screen. A single pixel is represented by two bits. This gives us the four possible color combinations shown in Table 2.

Table 2

Decimal	Binary Bits	Color
0	00	background
1	01	green or cyan
2	10	red or magenta
3	11	yellow or white

Note: We can disregard the fact that there are two palettes, since the screen pixel-bit map is the same for both. Palette 1 colors are the same as palette 0 colors, with blue added.

We want to know what new Pixel color is generated when we combine two old pixel colors using XOR. We break the operation down into bits and XOR each bit. We then recombine the resulting bits to derive the resulting color.

Here is an example. We want to take a red pixel, XOR it with a yellow pixel, and see what resulting color will be. First:

```
red    = binary 10 (decimal 2).
yellow = binary 11 (decimal 3).
```

Next, combine the left side bits according to the XOR table:

```
red       yellow  result
1x XOR    1x   =   0x
```

Next, combine the right side bits according to the XOR table:

```
red       yellow  result
x0 XOR    x1   =   x1
```

Finally, recombine the resulting bits into a two-bit value:

```
0x + x1 = 01
```

What color is 01? Green! So if we XOR red with yellow, the resulting color will always be green. That's how simple it is.

Table 3 shows PUT XOR operations. It lists all combinations of two source colors and the boolean results, which may be used to anticipate XOR-generated colors.

If you want to do animation, it is important to note that the color combinations are totally reciprocal. This makes it possible to animate multi-color figures using GET and PUT with the XOR option, drawing and erasing an image over a complex background in a totally non-destructive manner.

Table 3

(binary)				(decimal)			
A	XOR	B	= C	A	XOR	B	= C
00		00	00	0		0	0
00		01	01	0		1	1
00		10	10	0		2	2
00		11	11	0		3	3
01		00	01	1		0	1
01		01	00	1		1	0
01		10	11	1		2	3
01		11	10	1		3	2
10		00	10	2		0	2
10		01	11	2		1	3
10		10	00	2		2	0
10		11	01	2		3	1
11		00	11	3		0	3
11		01	10	3		1	2
11		10	01	3		2	1
11		11	00	3		3	0

A = original color
B = color on screen which is XORed with A
C = result color from the XOR operation

The program in Figure 1 sets up a simple table of color boxes onto which skulls are systematically drawn in three colors using the DRAW statement and the PUT XOR statement to demonstrate the boolean operation at work.

```
100 'XORDEMO.BAS - John Schnell - Nov 1984
110 'Demonstrates how PUT command changes colors
    in XOR mode
120 DIM BKGND(800):DIM SKULL(100)
130 CLS
140 SCREEN 1 : CLS
150 LINE (16,0)-(31,15),1,BF 'create cyan bkgnd box
160 LINE (32,0)-(47,15),2,BF 'create magenta bkgnd
    box
170 LINE (48,0)-(63,15),3,BF 'create white bkgnd box
180 GET (0,0)-(63,16),BKGND 'capture four bkgnd
    boxes
190 PUT (0,0),BKGND 'erase from corner
200 PUT (100,10),BKGND 'display bkgnd boxes in three
210 PUT (100,34),BKGND 'horizontal rows.
220 PUT (100,58),BKGND
230 '
240 LOCATE 1 ,14 : PRINT "0 1 2 3"; 'labels for four
    bkgnd colors
250 LOCATE 11,14 : PRINT "XOR test"; 'display test
    title
260 CLR = 1 : X = 10 : GOSUB 510 'draw cyan skulls
270 CLR = 2 : X = 34 : GOSUB 510 'draw magenta
    skulls
280 CLR = 3 : X = 58 : GOSUB 510 'draw white skulls
290 PUT (0,0),SKULL  'erase skull up in left corner
300 LOCATE 3,2 : PRINT "1";  'list pixel draw colors
310 LOCATE 6,12 : PRINT "2";
320 LOCATE 9,12 : PRINT "3";
330 LOCATE 20,1 : END
340 '
500 'Subroutine to draw skull and PUT XOR it over four
    backgrounds
510 DRAW "C=CLR;"  'create skull in current color
520 DRAW "BM3,1R9F2D7L2D2G2L5H2U2L2U7E2"
530 DRAW "BM4,5R1F1L3D1R3G1L1" 'skull draw
    instructions
540 DRAW "BM10,5R1F1L3D1R3G1L1"
550 DRAW "BM7,10R1BM+1,+2L3"
560 GET (0,0)-(15,15),SKULL 'get skull into array
570 PUT (100,X),SKULL,XOR 'put skull over bkgnd
580 PUT (116,X),SKULL,XOR 'put skull over cyan
590 PUT (132,X),SKULL,XOR 'put skull over magenta
600 PUT (148,X),SKULL,XOR 'put skull over white
610 RETURN
```

Figure 1. Drawing Skulls in Three Colors

How Compilers Work: An Overview

Lou Morgan
Madison IBM-PC User's Group
Madison, Wisconsin

What are compilers and how do they work? Many computer users ask this question after the programming bug has bitten them. To most people, a compiler appears to be a "black box program" that translates source code written in a high-level language (such as FORTRAN, BASIC, Pascal or C) into a language the computer can understand and execute.

Compilers vs. Interpreters

Computers cannot understand English words and grammar. They also cannot understand the highly structured words and sentences of programming languages. These languages must be translated by a compiler or interpreter before a computer can understand them.

A compiler or interpreter is a program that looks up each "word" of your programming language in a kind of dictionary (or lexicon) and, in a series of steps, translates into machine code.

An interpreter translates one line of source code at a time into machine code, then executes it. Debugging and testing is relatively fast and easy in interpreted code, since the entire program doesn't have to be reprocessed each time a change is made.

The BASIC.COM and BASICA.COM programs are interpreters. Interpreted programs run much slower than compiled programs, because they must be translated every time they are run. Programmers often test and debug their programs using an interpreter and then compile them for production use.

How Compilers Work

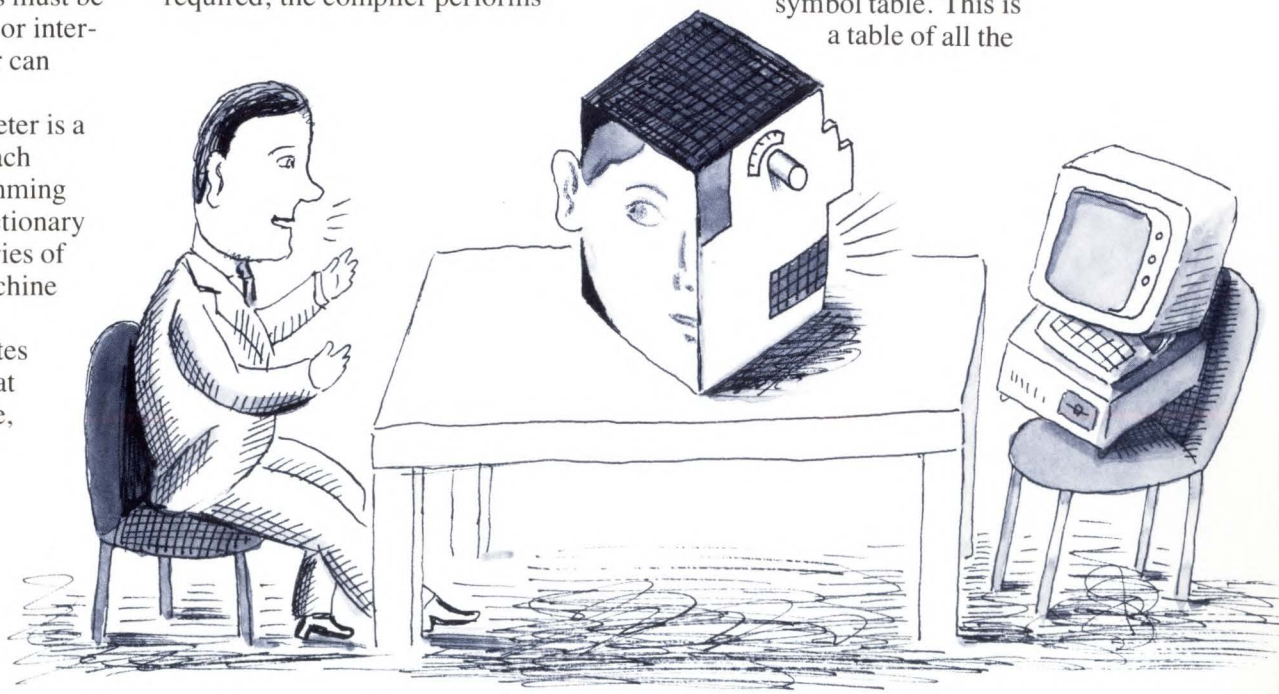
Compilers translate source code into virtual machine language. In the IBM PC, this virtual language is 8088 machine code.

Most compilers convert programs in three steps. Each step is called a pass. A particular compiler may have one program per pass, or may combine two or three steps in a single program. For a very complex language, a step may be so difficult to perform that it is broken up into many smaller steps. Regardless of how many passes or programs are required, the compiler performs only three main functions—lexical analysis, syntax analysis and code generation. During each pass of the compiler, the source code moves closer to becoming virtual machine language (or whatever language the compiler is designed to generate).

Lexical Analysis

In the first pass of a compiler, the source code is passed through a lexical analyzer, which converts the source code to a set of tokens. A token generally a number representing some keyword in the language. A compiler has a unique number for each keyword (e.g., IF, WHILE, END), and each arithmetic or logical operator (e.g. +, −, *, AND, OR). Numbers are represented by a token which indicates that the next thing should be interpreted as a number. The tokens put the programming language into a form that can be checked for proper structure and order.

The other important task of the lexical analyzer is to build a symbol table. This is a table of all the



identifiers (variable names, procedures and constants) used in the program. When an identifier is first recognized by the analyzer, it is inserted into the symbol table, along with information about its type, where it is to be stored, and so forth. This information is used in subsequent passes of the compiler.

Syntax Analysis

After the lexical analyzer translates a program into tokens of keywords, variables, constants, symbols and logical operators, the compiler makes its next pass. To describe what happens during this function, I will briefly explain grammars, parsing and semantic routines.

Like any language, programming languages have a set of rules governing the structure of the program. Each computer language has its own grammar that makes it unique. Some grammars are complex (e.g., PL/I) and others are relatively easy (e.g., Pascal). The programmer must observe all the structural rules of a language to make logical sense to the computer.

Next come the parsing routines, which check to see that the program obeys the language rules. The parser reads in the tokens generated by the lexical analyzer and compares them to the set grammar of the programming language. If the program has followed the rules of the language, it is syntactically correct. However, when the parser encounters an error, it issues a warning or error message and tries to continue. Some parsers try to correct a faulty program; others do not. When the parser reaches the end of the token stream, it will tell the compiler that either the program is grammatically correct and compiling can continue or the program contains too many errors and compiling must be aborted. If the program is grammatically correct, the parser will call for semantic routines.

For gramatically correct programs, the next functions—the semantic routines—perform two tasks: checking to ensure that each series of tokens will be understood by the computer when it is fully translated to machine code, and converting the series of tokens one step closer to machine code. The first task takes a series of tokens, called a production, and checks it to see if it makes sense. For example, a production may be correct as far as the parser is concerned, but the semantic routines check whether the variables have been declared, and are of the right type, etc. If the production makes sense, the semantic routine reduces the production for the next phase of compilation, code generation. Most of the code in the compiler is in the semantic routines, which take up the majority of the compilation time.

To summarize: two major routines—the parsing routine and the semantic routine—comprise syntax analysis. The parser checks for the correct order of the tokens and then calls the semantic routines to check whether the series of tokens (a production) will make sense to the computer. The semantic routine reduces the production another step toward complete translation to machine code.

Code Generation

The code generation process determines how fast the code will run and how large it will be. The first part of code generation involves optimization, and the second involves actual machine code generation.

In the optimization step, the compiler tries to make the intermediate code generated by the semantic routines more efficient. This process can be very slow and may not be able to improve the code much. Because of this, many compilers don't include optimizers, and, if they do, they look only for areas that are easy to optimize.

The machine code generation process takes the intermediate code produced by the optimizer (or semantic routines if the compiler has no optimizer) and generates virtual machine code, which in our case is 8088 machine code. It is this part of the compilation phase that is machine-dependent. Each type of computer has an operating system that processes virtual machine code differently; therefore, the code generator is designed specifically for the computer and operating system that the final code will run on. Based on the computer's operating system, at this point the compiler chooses instructions that will yield the fastest execution and smallest code size.

If the program is free from syntactical errors, code generation should take place with no problem. When the code generator is finished, the code produced will be 8088 machine code, but the format of the code is not yet executable. It is in a format (an .OBJ file in our case) that is ready to go to a linker, which creates an executable *.EXE or *.COM file from the machine code generated by the compiler.

The Mechanics of Assembly Language

(Part 2)

Is Hexadecimal Hexing You?

David Betts
San Francisco PC Users Group
San Francisco, California

(Editor's note: David Betts' initial article, "The Mechanics of Assembly Language: Dead Languages and Machine Language," appeared in the March 1985 diskette issue of Exchange.*)*

In this article, we will consider what "hexadecimal" numbers are and how to do arithmetic with them. First, some definitions are in order.

The word "hexadecimal" is a "put together" word that means a number in base 16. So, what is base 16?

A counting system (such as the decimal system we use every day) is built around a "base" or basic unit. The base for our decimal system is the number 10. Thus, we move around in base 10 numbers by powers of ten (i.e., ten multiplied by itself some number of times). For example, consider:

$$12{,}345_{10}$$

Note the subscript "10" below the 12,345. This indicates the base for this quantity is 10.

This quantity—usually written without the subscript 10—is a universally accepted base 10 shorthand for the expanded expression:

$$(1 \times 10^4) + (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (5 \times 10^0)$$

where $10^4 = 10 \times 10 \times 10 \times 10$; $10^3 = 10 \times 10 \times 10$; etc.

Another important property of the base is that it is also the number of different symbols or numerals needed to express quantities in that base. The symbols we use in the decimal system are the familiar numerals:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

The hexadecimal system works precisely the same way as the decimal system, except the base is now the number 16. This brings up a problem. How do we count to 16, if we only have ten symbols? We need six extra symbols or numerals in BASE 16 to stand for the decimal values 10, 11, 12, 13, 14 and 15. Of course, any arbitrary symbols would do, though for practical reasons they should be familiar to us and easy to produce.

IBM set the precedent long ago with the IBM System/360. This mainframe was the first computer to make use of base 16, also called hexadecimal. At that time, IBM chose to use the letters A through F to stand for the extra numbers, and these have been universally accepted since. Thus, to count to the decimal value 16 in hex, we go 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B ,C, D, E, F, 10, where "10" is a one in the "sixteens" place.

It should now be evident why we need to identify quantities from different bases with a subscript. Clearly:

$$10_{10} = 1 \times 10 \quad \text{and} \quad 10_{16} = 1 \times 16$$

are not the same thing. For this reason, most documentation containing hexadecimal numbers is very specific about what base it refers to.

The usefulness of base 16 becomes clearer if we consider how computers manipulate numbers.

At the machine level are binary, or base 2, quantities. From our previous discussion, it should be evident that base 2 has only two symbols, 0 and 1. These symbols correspond exactly with the way computers work mechanically, i.e., as millions of very fast switches that are either on or off. But this limited repertoire of symbols means that the only way to express all of a computer's quantities and characters is to use lots of 1's and 0's. For example, suppose we have just a four-place binary number:

$$XXXX_2$$

How many different things, like letters of the alphabet or numbers, can these four places represent? By counting up from zero, adding a one each time, we can see all the possible values of a four-place binary number:

0000 = 0
0001 = 1
0010 = 2 (we "carried")
0011 = 3
0100 = 4
.
.
1111 = 15

This makes a total of 16 numbers or things we could represent. Not many. However, this scheme can continue by simply adding binary digits or places. Thus, eight places are sufficient to represent up to 256 different characters, enough for the whole alphabet (both upper and lower case), the numerals, and other symbols. Sixteen places are sufficient to represent any unique location in 65,536 memory locations, and twenty places can uniquely locate any place in the approximately one million memory locations in the IBM PC.

The difficulty is that strings of twenty 1's and 0's get unwieldy to write or to work with for humans. This is where the hexadecimal notation is advantageous. Remember that a four-place binary number allowed us to count up to 16. So does a one-place hexadecimal number. Using hexadecimal numbers, we can shorten our notation by three-fourths, and our twenty-digit binary number can be reduced to five hex digits.

Let's do some simple arithmetic with both binary and hexadecimal numbers. To simplify the discussion, I will refer to hexadecimal numbers as simply HEX, decimal as DEC and binary as BIN.

Converting numbers from one base to another is a must for this kind of work. Several handy tools are available to do this, like the DEC/HEX/BIN conversion feature of SIDEKICK's calculator program (SIDEKICK is a trademark of Borland International), or the special hex calculators for programmers. Those of you without access to these items may find the program in Figure 1 useful. It is a Microsoft BASIC program that generates and prints a table for HEX to DEC conversion.

```
10 'Print a hex to decimal conversion table
20 'Change LPRINTs to PRINTs for screen
   output
30 COUNT = 0 : LPRINT SPC(5);
40 FOR I% = 0 TO 15
50 LPRINT HEX$(I%); SPC(3);
60 NEXT I% : LPRINT : LPRINT
70 FOR I% = 0 TO 15 : LPRINT
   HEX$(I%); SPC(1);
80 FOR J% = 0 TO 15
90 LPRINT USING "####"; COUNT;
100 COUNT = COUNT + 1
110 NEXT J%: LPRINT
120 NEXT I%
```

Figure 1. HEX to DEC Conversion Program

Type in and run the program. This will give you a way to check hex arithmetic as we proceed.

Both BIN and HEX numbers behave according to the same rules we are familiar with in DEC numbers—carrying and borrowing. All we have to do is get used to working with these four problems:

a) HEX add, no carry:

```
  11
+ 13
----
  24
```

b) HEX add with carry:

```
  3E
+ AF
----
  ED
```

c) HEX subtract, no borrow:

```
  24
- 13
----
  11
```

d) HEX subtract with borrow:

```
  ED
- AF
----
  3E
```

Operations in BIN are even easier, though somewhat tedious to write down. Only two symbols are involved, as the following problems illustrate:

e) BIN add, no carry:

```
    1  100 001 101 100
+                10 011
-------------------------
    1  100 001 111 111
```

f) BIN add with carry:

```
    1  000 101 111 111
+                10 110
-------------------------
    1  000 110 010 101
```

g) BIN subtract, no borrow:

```
    1  100 001 111 111
-1  100 001 101 100
-------------------------
                10 011
```

h) BIN subtract, with borrow:

```
    1  000 101 111 111
-1  000 110 010 110
-------------------------
                10 110
```

Inside the IBM PC, all arithmetic is done in binary in a part of the Intel 8086 or 8088 called the Arithmetic Logic Unit (ALU). The 8086 or −8 CPU chips also have the advantage of "hardwired" multiply and divide instructions. This means the arithmetic of multiplication (which a computer carries out as a series of additions) and division (carried out as a series of subtractions) can be done without the need for instructions that tell the ALU how to proceed. This, in turn, allows the 8086 or −8 to do multiplication and division much faster than its predecessors, although still not nearly as fast as the 8087 arithmetic co-processor.

The way a silicon chip adds and subtracts is a little different from the way we do it on paper. There are several important limitations and shortcuts that we need to be familiar with to program effectively. These will be the topic of the third and final part of this article.

Commonly Used Extensions For PC File Names

Mike Hannah
Sandia Computing Newsletter
Albuquerque, New Mexico

The microcomputer is truly a "personal" computer. As such, any file names you wish to create are limited only by the number of characters possible and your own creativity.

It is not necessary to append an extension (e.g., .TXT, .INP) to a file name. However, an application software product may assume that a certain filename extension represents a particular type of file.

Some of you have requested a list of these commonly-used file extension names. The list in Table 1 is by no means exhaustive, but it can be a helpful guide.

Table 1. Commonly-Used File Extension Names

.ASC	- Standard DOS ASCII data file
.ASM	- Assembler source file
.BAK	- Backup copy of another file
.BAS	- BASIC language source file
.BAT	- DOS Batch procedure file
.CFG	- WordMARC configuration file
.CMD	- dBASE II command program file
.CNF	- LOTUS configuration file
.COB	- COBOL language source file
.COD	- Object-listing file of program code
.COM	- DOS binary program file
.DAT	- Data file, usually input to a program
.DBF	- dBASE II database file
.DIF	- Data interchange format file
.DLB	- LOTUS device library
.DOC	- Document or documentation file
.DRV	- LOTUS hardware driver parameter file
.ENG	- WordMARC English language prompt file
.EV	- ExecuVision data file
.EXE	- DOS binary program file
.FMT	- dBASE II format file
.FON	- LOTUS character font definition file
.FOR	- FORTRAN language source file
.FRM	- dBASE II report form definition file
.GLY	- Microsoft WORD glossary file
.HEX	- Hexadecimal machine code file
.HLP	- Listable (printable) file of help information
.LIB	- Library file of binary programs
.LST	- Listable (printable) output file
.MAC	- MACRO assembler source file
.MAP	- Linker memory map file
.MEM	- dBASE II memory save file
.NDX	- dBASE II index file
.OBJ	- Relocatable object file
.OVL	- Binary program overlay file
.OVR	- Binary program overlay file
.PAS	- Pascal language source file
.PAT	- MultiMate printer action table
.PIC	- Graphics encoded picture file
.PRD	- MicroSoft WORD printer definition file
.PRF	- IBM Professional Editor profile file
.PRG	- dBASE II command program file
.PRN	- Printable (listable) output file
.PRO	- ProKey keyboard macro definition file
.REL	- Relocatable object file
.SAT	- MultiMate sheet feeder action table
.SET	- VTERM setup parameters file
.SRC	- WordMARC parameters source file
.STY	- MicroSoft WORD style definition file
.SYS	- DOS system parameter file
.TMP	- Temporary file of any kind
.TOC	- ExecuVision table of contents
.TUT	- LOTUS tutorial data file
.TXT	- Data text file
.UH	- MultiMate dictionary file
.VC	- VisiCalc worksheet file
.WKS	- LOTUS worksheet file
.$$$	- DOS temporary file

Troubleshooting Your IBM PC

Jerry Schneider
Capital PC User Group
Washington, DC

The IBM Personal Computer family comes complete with built-in diagnostic procedures to assist you in identifying many problems that may occur with any of your computer's components. These diagnostic procedures include:

- A Power-On Self-Test (POST) that is performed whenever a PC is powered on.
- General diagnostic testing that you can perform using the Diagnostic Diskette and accompanying Problem Determination Procedures (PDPs) outlined in the Guide to Operations manual for each system.
- Optional advanced diagnostic testing that you can perform using the Advanced Diagnostic Diskette and procedures provided in the Hardware Maintenance and Service manual.

Power-On Self-Test (POST)

Whenever you start up your computer, a series of tests are automatically performed, checking various components in your system. This Power-On Self-Test (POST) provides error or warning messages whenever it encounters a faulty component. The POST provides two types of messages: audio codes and display messages or codes.

Audio codes consist of variations of sounds or beeps that identify the faulty component. If your computer is functioning normally, you will hear one short beep when the system is powered on. If the POST detects a problem; it sounds a different series of audio codes. These audio codes and their corresponding problem areas are shown in Table 1.

On the Portable PC, Personal Computer XT and Personal Computer AT, the POST displays system memory as it is read. The last number displayed (640KB, for example) should be the total amount of memory in your system, including system board memory and any expansion memory.

Table 1.

Audio Code	Problem Area
No beep, continuous beep, or series of short beeps	Power Supply
1 long beep and 1 short beep	System Board
1 long beep and 2 short beeps, or 1 short beep and blank or incorrect display	Monitor adapter card and/or monitor cable
1 short beep and either the red drive LED staying on or Personal Computer BASIC statement	Drive and/or drive adapter card

During the POST, error messages or numeric codes are displayed whenever it detects a problem. In most cases, the error code is a three-or four-digit number. Checking this error code number against the list in Table 2 will identify the malfunctioning component.

General Diagnostics Testing

Each IBM Personal Computer comes with a Guide to Operations (GTO) manual. The GTO includes a diagnostic diskette to assist you in identifying problems your Personal Computer may have.

The diagnostics diskette and corresponding problem determination procedures (PDPs) section of the GTO manual provide step-by-step instructions to test the various parts of your personal computer system, including the system unit with installed options, expansion unit, keyboard, display, and printer. You should perform these tests when you first install your system, at periodic times during the lifetime of the system for precautionary purposes, and whenever errant behavior is detected or suspected in your system.

To run the diagnostics tests, place a backup copy of the diagnostics diskette in your A: drive and power on your system; or, if the system is already on, press the Ctrl-Alt-Del keys simultaneously to reboot the system. The main diagnostics menu will be displayed with options similar to the following:

0 - SYSTEM CHECKOUT
1 - FORMAT DISKETTE
2 - COPY DISKETTE
3 - PREPARE SYSTEM FOR MOVING
4 - SETUP [AT only]
9 - END DIAGNOSTICS

Table 2.

Code	Description
01x	Undetermined problem errors.
02x	Power supply errors.
1xx	System board errors.
101	System board error - Interrupt failure.
102	System board error - Timer failure.
103	System board error - Timer interrupt failure.
104	System board error - Protected mode failure.
105	System board error - Last 8042 command not accepted.
106	System board error - Converting logic test.
107	System board error - Hot NMI test.
108	System board error - Timer bus test.
109	Direct memory access test error.
121	Unexpected hardware interrupts occurred.
131	Cassette wrap test failed.
152	
161	System Options Error-(Run SETUP) [Battery failure].
162	System options not set correctly-(Run SETUP).
163	Time and date not set-(Run SETUP).
164	Memory size error-(Run SETUP).
199	User-indicated configuration not correct.
2xx	Memory (RAM) errors.
201	Memory test failed.
202	Memory address error.
203	Memory address error.
3xx	Keyboard errors.
301	Keyboard did not respond to software reset correctly, or a stuck key failure was detected. If a stuck key was detected, the scan code for the key is displayed in hexadecimal. For example, the error code 49 301 indicates that key 73, the PgUp key, has failed (49 hex = 73 decimal).
302	User-indicated error from the keyboard test, or AT keylock is locked.
303	Keyboard or system unit error.
304	Keyboard or system unit error; CMOS does not match system.
4xx	Monochrome monitor errors.
401	Monochrome memory test, horizontal sync frequency test, or video test failed
408	User-indicated display attributes failure.
416	User-indicated character set failure.
424	User-indicated 80 X 25 mode failure.
432	Parallel port test failed (monochrome adapter).
5xx	Color monitor errors.
501	Color memory test failed, horizontal sync frequency test, or video test failed.
508	User-indicated display attribute failure.
516	User-indicated character set failure.
524	User-indicated 80 X 25 mode failure.
532	User-indicated 40 X 25 mode failure.
540	User-indicated 320 X 200 graphics mode failure.
548	User-indicated 640 X 200 graphics mode failure.
6xx	Diskette drive errors.
601	Diskette power-on diagnostics test failed.
602	Diskette test failed; boot record is not valid.
606	Diskette verify function failed.
607	Write-protected diskette.
608	Bad command diskette status returned.
610	Diskette initialization failed.
611	Timeout - diskette status returned.
612	Bad NEC - diskette status returned.
613	Bad DMA - diskette status returned.
621	Bad seek - diskette status returned.
622	Bad CRC - diskette status returned.
623	Record not found - diskette status returned.
624	Bad address mark - diskette status returned.
625	Bad NEC seek - diskette status returned.
626	Diskette data compare error.
7xx	8087 or 80287 math coprocessor errors.
9xx	Parallel printer adapter errors.
901	Parallel printer adapter test failed.
10xx	Reserved for parallel printer adapter.
11xx	Asynchronous communications adapter errors.
1101	Asynchronous communications adapter test failed.
12xx	Alternate asynchronous communications adapter errors.
1201	Alternate asynchronous communications adapter test failed.
13xx	Game control adapter errors.
1301	Game control adapter test failed.
1302	Joystick test failed.
14xx	Printer errors.
1401	Printer test failed.
1404	Matrix printer failed.
15xx	Synchronous data link control (SDLC) communications adapter errors.
1510	8255 port B failure.
1511	8255 port A failure.
1512	8255 port C failure.

Table 2. (cont.)

1513	8253 timer 1 did not reach terminal count.
1514	8253 timer 1 stuck on.
1515	8253 timer 0 did not reach terminal count.
1516	8253 timer 0 stuck on.
1517	8253 timer 2 did not reach terminal count.
1518	8253 timer 2 stuck on.
1519	8273 port B error.
1520	8273 port A error.
1521	8273 command/read timeout.
1522	Interrupt level 4 failure.
1523	Ring Indicate stuck on.
1524	Receive clock stuck on.
1525	Transmit clock stuck on.
1526	Test indicate stuck on.
1527	Ring indicate not on.
1528	Receive clock not on.
1529	Transmit clock not on.
1530	Test indicate not on.
1531	Data set ready not on.
1532	Carrier detect not on.
1533	Clear to send not on.
1534	Data set ready stuck on.
1536	Clear to send stuck on.
1537	Level 3 interrupt failure.
1538	Receive interrupt results error.
1539	Wrap data miscompare.
1540	DMA channel 1 error.
1541	DMA channel 1 error.
1542	Error in 8273 error checking or status reporting.
1547	Stray interrupt level 4.
1548	Stray interrupt level 3.
1549	Interrupt presentation sequence timeout.
16xx	Display emulation errors (327x, 5520, 525x).
17xx	Fixed disk errors.
1701	Fixed disk POST error.
1702	Fixed disk adapter error.
1703	Fixed disk drive error.
1704	Fixed disk adapter or drive error.
1780	Fixed disk 0 failure.
1781	Fixed disk 1 failure.
1782	Fixed disk controller failure.
1790	Fixed disk 0 error.
1791	Fixed disk 1 error.
18xx	I/O expansion unit errors.
1801	I/O expansion unit POST error.
1810	Enable/Disable failure.
1811	Extender card warp test failed (disabled).
1812	High order address lines failure (disabled).
1813	Wait state failure (disabled).
1814	Enable/Disable could not be set on.
1815	Wait state failure (disabled).
1816	Extender card warp test failed (enabled).
1817	High order address lines failure (enabled).
1818	Disable not functioning.
1819	Wait request switch not set correctly.
1820	Receiver card wrap test failure.
1821	Receiver high order address lines failure.
19xx	3270 PC attachment card errors.
20xx	Binary synchronous communications (BSC) adapter errors.
2010	8255 port A failure.
2011	8255 port B failure.
2012	8255 port C failure.
2013	8253 timer 1 did not reach terminal count.
2014	8253 timer 1 stuck on.
2016	8253 timer 2 did not reach terminal count, or timer 2 stuck on.
2017	8251 Data set ready failed to come on.
2018	8251 Clear to send not sensed.
2019	8251 Data set ready stuck on.
2020	8251 Clear to send stuck on.
2021	8251 hardware reset failed.
2022	8251 software reset failed.
2023	8251 software "error reset" failed.
2024	8251 transmit ready did not come on.
2025	8251 receive ready did not come on.
2026	8251 could not force "overrun" error status.
2027	Interrupt failure - no timer interrupt.
2028	Interrupt failure - transmit, replace card or planar.
2029	Interrupt failure - transmit, replace card.
2030	Interrupt failure - receive, replace card or planar.
2031	Interrupt failure - receive, replace card.
2033	Ring indicate stuck on.
2034	Receive clock stuck on.
2035	Transmit clock stuck on.
2036	Test indicate stuck on.
2037	Ring indicate stuck on.
2038	Receive clock not on.
2039	Transmit clock not on.
2040	Test indicate not on.
2041	Data set ready not on.
2042	Carrier detect not on
2043	Clear to send not on.
2044	Data set ready stuck on.
2045	Carrier detect stuck on.
2046	Clear to send stuck on.
2047	Unexpected transmit interrupt.
2048	Unexpected receive interrupt.
2049	Transmit data did not equal receive data.
2050	8251 detected overrun error.
2151	Lost data set ready during data wrap.
2152	Receive timeout during data wrap.
22xx	Cluster adapter errors.
24xx	Enhanced graphics adapter errors.
29xx	Color matrix printer errors.
2901	
2902	
2904	
33xx	Compact printer errors.

Options 0, 1 and 2 are part of the diagnostics procedures. Option 3, "Prepare System For Moving," is used to "park" or secure the heads on a hard disk so that the system unit can be safely moved without damaging the disk or its contents. Option 4 is used with the AT to identify installed options when you first set up your system:

When running the diagnostics procedures, use options 1 and 2 to check the operation of your floppy diskette drives or to prepare a diskette for use without having to reload DOS. Select option 0, "System Checkout," for general testing. Option 0 displays a list of installed options you may want to verify, and then provides a secondary menu of choices similar to the following:

0 - RUN TESTS ONE TIME
1 - RUN TESTS MULTIPLE TIMES
2 - LOG UTILITIES
3 - END SYSTEM CHECKOUT

Both Option 0 and 1 walk you through a series of tests that check each component in the system. During system testing, a three-or four-digit error code will be generated for each component tested. If no problem is found, the last two digits will be 00, for example 300, 900, or 1700. Table 2 lists many of the error codes that may be generated during diagnostics testing.

If an error is detected when running Option 0, select the Log Utilities Option (2) to list any errors to disk or printer, and then select Option 1, Run Tests Multiple Times. This will determine whether the problem is consistent or intermittent.

While the GTO diagnostics do an excellent job of identifying specific problem areas or problem components, they provide only limited assistance as to how you correct the errors. In fact, the advice most frequently provided is to "Have your system unit [or problem device] serviced."

Advanced Diagnostics Testing

If you are the more adventuresome type, you can purchase the IBM Hardware Maintenance and Service (HMS) manual for your particular unit. While the Guide to Operations manual assists you in identifying a problem component, the HMS manual provides you with information to both isolate and repair any failure of a "Field Replaceable Unit (FRU)"—any part or component that has interchangeable replacement parts that are stocked by IBM or the Original Equipment Manufacturer.

The HMS includes an advanced diagnostics diskette and accompanying Problem Isolation Codes (PICs) instructions to isolate and identify problem components.

To run the Advanced Diagnostics tests, follow the same procedures detailed in the previous section for General Diagnostics testing.

The tests performed by the advanced diagnostics diskette are far more detailed and precise than those of the general diagnostics diskette in the GTO. In addition to identifying the problem component, the advanced diagnostics tests further attempt to identify the specific part of the device that is malfunctioning.

Once you identify a problem, the HMS provides detailed instructions as to how you adjust, remove or replace the affected part. The manual includes comprehensive hardware design information and parts lists, which specify replacement part numbers and internal design specifications.

New Products

IBM Personal Computer XT Models 068 and 078

Two new models of the IBM Personal Computer XT, Model 068 and Model 078, are now available. Both new models use the IBM PC XT planar, power supply, keyboard, frame and cover.

The IBM Personal Computer XT Model 068 has 256KB of memory and one 360KB diskette drive. A field-installable 10MB fixed disk and adapter, a second 360KB diskette drive, and an Asynchronous Communications Adapter are available options for Model 068.

The IBM Personal Computer XT Model 078 comes with 256KB of memory and two 360KB diskette drives. One diskette drive can be replaced with an optional 10MB fixed disk and adapter. The Asynchronous Communications Adapter also is available as an option for Model 078.

Mainframe Communication Assistant

Mainframe Communication Assistant is a full-function communications program that allows you to connect any member of the IBM PC family to an IBM host system.

Through asynchronous communication, you can connect an IBM Personal Computer to an IBM host system as though the PC were a 3101 terminal in either block or character mode, with the ability to upload and download files. By using data stream converters on the host system (such as the VM/370 Pass-through Virtual Machine (PVM) or the Series/1 Yale ASCII IUP), the Personal Computer can appear to the host as an IBM 3277 terminal. Or, by using the IBM 3278/79 Emulation Adapter (available only for the IBM Personal Computer and IBM Personal Computer XT), you can connect your PC directly to a host system as though the PC were a 3278/79 terminal.

Host support programs supplied with IBM Mainframe Communication Assistant provide full transfer capabilities (upload and download) for both VM CMS and MVS TSO environments. This function is further enhanced by the ability to upload "changes only" to previously transmitted text files. Checksum and sequence checking is done to ensure data integrity, and the program automatically retransmits any data in error.
Files can be ASCII, XASCII or binary, allowing you to transfer both data and programs.

Mainframe Communication Assistant's "hot key" support permits DOS applications to run concurrently with file transfer operations. The screen recall function lets you save host session screen images (up to 30), which you can then selectively recall or delete. The split screen mode lets you maintain a live host session at the bottom of the screen while selectively recalling previously saved screen images at the top of the screen.

Mainframe Communication Assistant can run in the TopView environment with full windowing capability.

As a member of the Assistant Series of programs, Mainframe Communication Assistant lets you download mainframe data directly into IBM Filing Assistant files. Any Assistant Series program (Writing, Graphing, Reporting, Planning) can then use the data in creating other documents. Mainframe Communications Assistant also provides the capability of downloading QMF and TIF data to a PC in either VC or DIF format. (DIF is a trademark of Software Arts, Inc.) DIF information can then be read into IBM Graphing Assistant and IBM Planning Assistant to create graphs and spreadsheets without re-entry.

Mainframe Communication Assistant can connect to any full-duplex or half-duplex communication service such as THE SOURCE, Dow Jones, CompuServe, etc. (THE SOURCE, America's Information Utility is a service mark of Telecomputing Corporation, a subsidiary of Reader's Digest Association, Inc. Dow Jones News/Retrieval is a registered trademark of Dow Jones and Company Inc. CompuServe is a service mark of CompuServe, Inc.)

The program supports facilities for user-written exits and extensions for automatic logon and access, or for data extraction from a database facility. It also permits stored commands to operate a communications session with responses to both the host communications and the Personal Computer.

Mainframe Communication Assistant comes with three diskettes (one with IBM PC programs, one with Host programs and installation procedures, and one with sample programs and applications), User's Guide, Quick Reference Card, and Registration Card.

Mainframe Communication Assistant runs on all members of the IBM Personal Computer family. It requires a minimum of 128KB of memory (256KB of memory for TopView or "hot key" support); one double-sided diskette drive; and either an IBM Monochrome Display or IBM Color Display with the appropriate adapter.

IBM Document Retrieval Assistant

IBM Document Retrieval Assistant is a version of IBM's Office Correspondence Retrieval System (OCRS) modified for the IBM Assistant Series.

Document Retrieval Assistant lets you search for and retrieve documents, graphs, reports, spreadsheets and other data by describing the content of files in which the data is stored.

In Document Retrieval Assistant, the abstraction and retrieval processes search existing files for keywords, a much easier and more productive process than having to remember DOS file specifications.

In the abstraction process, you select a data file or files to be summarized. Document Retrieval Assistant then creates an abstract, called a document summary, by scanning the document for keywords (date, sender, recipient, subject, etc). The summary file is then stored for use in the retrieval process, allowing you to store the original document elsewhere on diskette or hardcopy while still being able to "find" the document during a search. Document Retrieval Assistant lets you add keywords to the summary file to supplement those it found in the original document.

The retrieval process features a flexible

query design. To retrieve a document, you describe the content of desired document(s) using either random keywords or complete English sentences. Document Retrieval Assistant automatically scans your description to find key-words to use as the basis for the search. This flexibility eliminates the need to learn a special query language.

Once the description is entered, Document Retrieval Assistant searches the summary data base for files that match keywords in the description. In its retrieval process, Document Retrieval Assistant looks up your keywords in a synonym dictionary. This enables documents to be retrieved even though the keywords did not match exactly. (The synonym dictionary requires Document Retrieval Assistant to be installed on a fixed disk.) Document Retrieval Assistant gives you a list of data files in the order of most probable match.

The default data format for Document Retrieval Assistant is the Writing Assistant data stream. The other data streams supported are:

- Revisable Form Text Document Content Architecture (RFTDCA, also known as level 3 DCA), available in DisplayWrite 1, DisplayWrite 2, DisplayWrite 3, and PCWriter
- EasyWriter
- ASCII, available in Peachtext, HomeWord, Personal Editor, etc.
- Wordstar (Wordstar is a registered trademark of MicroPro International)

IBM Document Retrieval Assistant can run in the TopView environment and on the IBM PC Network.

IBM Document Retrieval Assistant runs on all members of the Personal Computer family. It requires a minimum of 128KB (192KB including DOS) and one 360KB double-sided diskette drive. Two double-sided diskette drives and 256KB of memory for TopView and PC Network are recommended.

IBM Planning Assistant Solutions

IBM Planning Assistant Solutions is a set of predefined spreadsheets and formulas that enhance the IBM Assistant Series. The spread-sheets and formulas can use files created with IBM Filing Assistant, can be graphed with IBM Graphing Assistant and can be merged into documents printed with IBM Writing Assistant.

The predefined spreadsheets and built-in mathematical formulas provide financial planning and accounting information for the home user or office professional. Planning Assistant Solutions derives answers to such questions as "What is my net worth?" and "What will be the payment if I put down this much money?"

IBM Planning Assistant Solutions spreadsheets for the home include:

- Home Budget - Aids in planning a monthly family budget. This spreadsheet allows entry of both budgeted and actual amounts and calculates the monthly and annual variance between the two.
- Cash Flow Statement - Provides a summary of income and expenses for a specific period of time, showing what is available for savings and investment.
- Loan Calculation - Calculates the periodic payment required to repay a mortgage or installment loan in equal installments. The Mortgage Loan View calculates the monthly mortgage payment, including property taxes, hazard insurance, flood insurance and other expenses. The Installment Loan View also calculates the monthly payment for an installment loan.
- Loan Amortization - Calculates the periodic payment required to repay a loan in equal amounts. This spreadsheet separates the principal and interest portions of each payment for a twelve-month period, and keeps year-to-date amounts for both principal and interest.
- Net Worth Statement - Records information concerning assets and liabilities and determines net worth.
- Stock Portfolio Analysis - Aids in managing stock portfolios. As stocks are bought and sold, this spreadsheet calculates gains and losses.
- Real Estate Analysis - Calculates the appraised value of real estate. All purchase information can be recorded for future reference. Prospective properties can be evaluated and compared.
- Income Property Cash Flow - Provides a way of testing and projecting cash flows for a typical income property. This spreadsheet gives special attention to financial cash flow, mortgage debt and taxes.

IBM Planning Assistant Solutions spreadsheets for the small business include:

- Travel Expense Form - Records expenses incurred while traveling.
- Financial Statement Worksheet - Records a chart of accounts and adjusting entries. The Balance Sheet and Income Statement are computed using the Planning Assistant VIEW command.
- Time Accounting (Personal and Business) - Compares hours actually worked with budgeted hours and displays variances.
- Accelerated Cost Recovery System (ACRS) - Calculates the depreciation value for all properties acquired since 1980 that qualify for ACRS.
- Straight Line and Double Declining Balance - Contains formulas for calculating depreciation under both Straight Line and Double Declining Balance methods.
- Sum Of The Years - Calculates depreciation expense based on the Sum Of The Years method.
- Monthly Overhead Analysis - Combines a typical itemized cost statement with an accumulated percentages cost statement. Information from this spreadsheet can be consolidated with the Yearly Overhead Analysis using IBM Planning Assistant functions.
- Yearly Overhead Analysis - Evaluates yearly business costs and profits, the cumulative percent of overhead cost, and cumulative yearly net profits. This spreadsheet is consolidated with the Monthly Overhead Analysis using IBM Planning Assistant functions.
- Forecasting - Calculates the projections of business revenues and expenses without historical data on which to base the analysis. Applications can be made for forecasting short- or long-range product sales and margins. The resulting amounts are compounded annually by the specific rate provided.

IBM Planning Assistant Solutions runs on all members of the Personal Computer family. It requires a minimum of 128KB of memory, DOS 2.00 or later, one double-sided diskette drive, and either an IBM Monochrome Display, IBM Color Display or other 80-column color monitor, with the appropriate adapter.

IBM PC Storyboard

IBM PC Storyboard is a set of four modules that make it easy to create vivid presentations with color, special effects and graphics, eliminating many of the time-consuming and costly tasks associated with other media. You can easily modify pictures and stories created with PC Storyboard, and you can create new presentations using parts of existing stories.

Presentations created with PC Storyboard may be shown directly on an IBM PC Color Display, on a TV/composite monitor, or projected with a video projector. Black/white and color hardcopy, each with an extended range of shades or col-

ors, also are available.

The four modules comprising PC Storyboard are Picture Taker, Picture Maker, Story Editor and Story Teller.

The *Picture Taker* module compiles a set of pictures from many different sources for inclusion in a presentation. Picture Taker co-resides in memory with the program that creates the pictures on the screen. The pictures can be saved for later use with the Picture Maker, Story Editor and Story Teller modules (described below).

Screens created by many PC DOS programs can be saved on disk. Screen formats can be IBM monochrome mode, 40- or 80-column color text mode, medium-resolution graphics mode (320 X 200, four-color) and high-resolution graphics mode (640 X 200, two-color).

The *Picture Maker* module lets you create original pictures by using text and figure creation functions. By taking advantage of the "cut-and-paste" and copy functions, you also can modify and include pictures previously made with Picture Maker or saved with Picture Taker. You can "cut" pieces of previously saved pictures and combine them with text or figures to create new pictures. These new pictures can be saved for later use with Picture Maker, Story Editor, or Story Teller. Saved pictures also are accessible by BASIC programs. PC Storyboard comes with a symbol library containing symbols that you can cut and paste. All pictures created or modified by Picture Maker are in medium-resolution graphics mode (320 X 200, four-color).

You can create text using any of four provided type styles (each in five sizes), or you can create custom type styles and size variations. All text can appear in different colors, with an option to outline the letters in another color. In addition, shadowing in eight directions can be added in still another color. Text can be placed anywhere on the picture being created, and an easy-to-use function lets you italicize any text.

With figure creation commands, you can draw unique figures and shapes and create circles, rectangles and lines. Figures can be "filled" with any one of four colors or with program-provided or user-created patterns. A number of half-tone patterns significantly extends the color possibilities. The fill process lets you try all possible fill options by depressing a key to see the figure filled with the next color/pattern option. Figures can also use the shadow options.

A special Composite mode controls and improves pictures created using Picture Maker for use with a TV/composite monitor. You create pictures using composite patterns that display up to 16 simultaneous colors when shown on a composite/TV monitor. This mode can also be used for pictures intended for output to a Video Cassette Recorder.

The print function lets you specify seven gradations from black to white for printing on an IBM Graphics Printer. You can assign any of seven black/gray/white print patterns to all colors/patterns displayed. For an IBM Color Printer, you can select 16 colors from a possible 91.

The *Story Editor* module organizes pictures into presentation sequence and displays the pictures using varied special effects. Sophisticated dissolve/display techniques make transitions from picture to picture and make partial transitions within pictures. Ten dissolve techniques, combined with varying picture-change speeds, give you more than 50 different effects. You also can add run-time color changes to change colors once or continuously through a broad spectrum.

You can manipulate and display complete pictures or partial pictures in any sequence. The editing process is interactive, letting you make changes while reviewing the story. You can include subroutines, loops, and menus. Labels for presentation-time branching are supported.

The *Story Teller* module presents the stories you create with the Story Editor. It lets you control the time it takes a picture to appear, as well as the pause time between pictures. You can select a time range of 0.1 seconds to 5 minutes on a picture-by-picture basis, or, to give a continuous showing, you can override the set time range by pressing a designated key.

This stand-alone module can be on the same diskette as the story and its pictures. Run-time branching and pauses are available. The presentation can start and stop at any point, branch to a key-entered label, or branch on any key or on specific value entered, depending on the run-time flexibility you have built into the story. Presenter menu options give an even greater flexibility to your presentations.

The program lets the presenter back up in the story and then return to the current place. The show chaining capability allows continuous running, or unlimited length presentations.

IBM PC Storyboard runs on any member of the Personal Computer family, including the PC XT/370 and IBM 3270 PC (Picture Taker does not run with the 3270 PC control program). The amount of memory required to run the various modules of PC Storyboard depends on the version of DOS:

PC Storyboard also requires one double-sided diskette drive; and an IBM Color Display or compatible 80-column color monitor with the appropriate adapter (Story Teller does not require a color monitor).

Learning DOS

Learning DOS is a Private Tutor version 2.00 course that teaches how to use the IBM Personal Computer Disk Operating System (DOS). The course is ideal for reviewing and practicing DOS commands, or for beginning computer users who want to learn the fundamentals of operating the IBM Personal Computer. The program explains the commands found in DOS 2.00, 2.10, 3.00 and 3.10. The ten lessons cover start-up, file and diskette commands, the EDLIN editor, DOS commands, file operations, batch files, directories and fixed disk file capabilities.

Learning DOS includes support for the IBM PCjr and IBM Personal Computer AT. It runs on any member of the Personal Computer family with a minimum of 128KB of memory and requires one double-sided diskette drive; a monitor or TV; DOS 2.00 or later; and the IBM PC Private Tutor version 2.00 diskette.

DOS Technical Reference

The new IBM PC Disk Operating System (DOS) Technical Reference manual contains detailed technical information not included in the DOS manuals. It replaces the existing DOS 2.10 and DOS 3.00 Technical Reference manuals.

Intended for the more experienced DOS users, application developers and system programmers, the new Technical Reference manual significantly improves the information in the DOS 2.10 Technical Reference manual and includes information about the features and capabilities of DOS 3.10, the IBM Personal Computer AT, and the IBM PC Network.

The IBM PC DOS Technical Reference manual includes the Update Information Service, which enables registered manual owners to receive update packages with corrections and timely information.

IBM PC DOS Technical Reference Tradeup

Owners of the DOS 2.10 Technical Reference manual or the DOS 3.00 Technical Reference manual may obtain the new DOS Technical Reference manual at a reduced price. Order forms are available from Authorized IBM Personal Computer Dealers and Authorized IBM Personal Computer Software Dealers. To take advantage of the tradeup, owners should submit a signed order form, a check made payable to IBM, and the front cover of their DOS 2.10 or 3.00 Technical Reference manual (but not the binder) as proof of ownership. This tradeup offer expires December 31, 1985.

Ask IBM

(Editor's note: Exchange *will publish technical questions and answers from time to time. We are able to respond to only a small number of technical questions, and will publish questions and answers of general interest. If you have a technical question of limited interest, or if you need an immediate answer, we encourage you to ask members of your user group or your Authorized IBM Personal Computer Dealer.)*

Q: I am having difficulties saving functions in APL using the full-screen editor. Occasionally, all of the changes I made during an editing session were not saved.

A: A function cannot be saved when the SI (State Indicator) contains an error. If you forget to clear the SI and you enter a number of changes in a function using the full-screen editor, you cannot save the changes and can exit the editor only via the Quit function key.

Q: How can I use the EDITFILE primitive in Logo to create a new file?

A: The EDITFILE primitive cannot automatically create a new file. To create a new file, you should first open the file with the OPEN primitive, then close it using the CLOSE primitive.

Q: When I use the various compiler options, how much extra code does the BASIC Compiler generate?

A: Table 1 shows the amount of code generated for each compiler option.

The choice of option depends on the length of the program, the use of many or few multiple-statement lines, and the needs of the program. A detailed explanation of the options appears in the BASIC Compiler manual.

Table 1.

BASIC Compiler Option	Bytes per Statement	Bytes per Line
/E		4
/X	4	
/V	1	
/W		1

The Special Code Parameters (options /A, /C, /D, /N, /R, /S and /O) are used for the special situations shown in Table 2.

Table 2.

Option	Use	Usual Impact on Size of Code Generated
/A	Generates complete object code listing	None
/C	Allows you to set COM buffer size	None (but affects available memory)
/D	Allows debugging, TRON/TROFF usage, error handling	Increases
/N	Relaxing of line numbering constraints	Decreases
/R	Array storage ordering (row-column major)	
/S	String storage — writes strings >4 bytes to disk during compilation	Decreases code generated during compilation. May increase size of running program
/O	Allows program to run without BASRUN.EXE. License is required for programs marketed using BASRUN.EXE	Increases

Q: DOS How much memory does the DOS PRINT command use?

A: The DOS 2.00/2.10 PRINT command uses 3200 bytes of memory. The DOS 3.00/3.10 PRINT command lets you set the buffer size when you initially execute the PRINT command.

Editor's Comments

A New Look for *Exchange*

During the past year, the IBM PC User Group Support Department has published the monthly diskette newsletter *Exchange of IBM PC Information*, and has sent a copy to every registered IBM Personal Computer user group. The user group made the diskette *Exchange* available to its members, usually in the group's software library.

Each month the diskette *Exchange* contained several articles that we selected from the many user group newsletters sent to us. It also contained information from IBM, such as new product announcements and User Group Support Department news.

We believe *Exchange* has been valuable in several ways. *Exchange* encourages user group members to write good articles for their own group newsletters and for potential publication in *Exchange*. It offers all user group members the opportunity to obtain and use information from their fellow PC users. Finally, it gives IBM a way to increase the level of PC knowledge among users and to enhance user productivity.

Because of these positive factors, we intend not only to continue distributing *Exchange* to user groups, but to make it an even better product.

Beginning with this issue, *Exchange* takes on a new appearance, adds new content, and increases its circulation.

The obvious change is that we have converted *Exchange* to a printed publication in news journal format. Now that it is printed, you can read *Exchange* at your leisure. Program code formerly available on the diskette will now be placed on the User Group Support Department's bulletin board system data base for downloading.

We have significantly enhanced the content of *Exchange* by adding IBM-written articles. Now *Exchange* contains the usual good user-written articles you have come to expect, as well as IBM-written technical articles about IBM PC hardware and software. This combination is intended to make *Exchange* a well-rounded publication offering additional perspectives and increased value.

We are now sending each registered PC user group a sufficient number of copies for all of its regularly active members. As in the past, there is no charge to user groups for *Exchange*.

To accomplish our goals, we have increased our staff. I will edit the newsletter as well as write some of the IBM-written articles. Bernard Penney, editor of the diskette *Exchange*, is now the User Group Editor, responsible for developing the user-written portions of *Exchange*. Karen Porterfield, John Warnock, Steve Mahlum and others will contribute articles as well as provide design and production assistance.

Even though our staff has grown, we still rely heavily on you, our readers, to contribute to the exchange of technical information about the IBM Personal Computer. We encourage you to maintain active involvement in your user group, write good articles for your group's newsletter, and (as a result) see your articles published in *Exchange*.

We hope you enjoy and benefit from the new *Exchange*. We look forward to serving your needs and publishing your PC information for the benefit of all IBM PC user group members.

Michael Engelberg
Editor

"To be warranted, connecting the PCs not only has to be affordable but has to increase productivity. (Page 2)

"Another strategy for joining PCs is to tie independent PCs into another computer as a "workstation." (Page 3)

"I would venture to say it (PC Network Adaptor) has more onboard horsepower than most of the PCs it is designed to plug into. (Page 6)

"The IBM PC Network appears to be a very appealing product. (Page 7)

"The IBM Personal Computer DOS 3.00 is a powerful extension of earlier versions of DOS. Its new functions make the Personal Computer more versatile and functional. (Page 28)

"With BASIC 3.00, now you can exercise direct control over nonstandard peripherals through device drivers. (Page 34)

"Now you can use the ERDEV and ERDEV$ variables to get more information about hardware problems. (Page 35)