Personal Systems

Issue 2, 1991

IBM
Disk Operating System Version 4.00
3.5" diskettes
Programming Family
IBM
Operating System/2 Standard Edition Version 1.3
OS/2
3.5" diskettes
Programming Family
AIX Operating System
Using The Operating System
AIX
IBM

IBM Personal Systems Technical Solutions

IBM®

IBM Personal Systems Technical Solutions is published quarterly by the U.S. Marketing and Services Group, International Business Machines Corporation, Roanoke, Texas, U.S.A.

Editor	Libby Boyd
Consulting Editor	Ed Bamberger
Automation Consultant	Andy Frankford
Communications	*Elisa Davis*
Design	*Corporate Graphics*
Illustrator	*Bill Carr*
Manager	*Bill Hawkins*

To correspond with *IBM Personal Systems Technical Solutions*, please write the editor at:

IBM Corporation
Internal Zip 40-A2-04
One East Kirkwood Blvd.
Roanoke, TX 76299-0015

To subscribe to this publication, use an IBM System Library Subscription Service (SLSS) form, available at IBM branches, and specify form number GBOF-1229.

Permission to reprint material from this magazine is granted to publications that are produced for non-commercial use and do not charge a fee. When republishing, include the names and companies of authors, and add the words "Reprinted by permission of *IBM Personal Systems Technical Solutions*." Other publications may reprint only with permission of the editor.

Titles and abstracts, but no other portions, of information in this publication may be copied and distributed by computer-based and other information-service systems.

IBM believes the statements contained herein are accurate as of the date of publication of this document. However, IBM hereby disclaims all warranties as to materials and workmanship, either expressed or implied, including without limitation any implied warranty of merchantability or fitness for a particular purpose. In no event will IBM be liable to you for any damages, including any lost profits, lost savings or other incidental or consequential damage arising out of the use or inability to use any information provided through this service even if IBM has been advised of the possibility of such damages, or for any claim by any other party.

Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

This publication could contain technical inaccuracies or typographical errors. Also, illustrations contained herein may show prototype equipment. Your system configuration may differ slightly.

IBM has tested the programs contained in this publication. However, IBM does not guarantee that the programs contain no errors.

This information is not intended to be a statement of direction or an assertion of future action. IBM expressly reserves the right to change or withdraw current products that may or may not have the same characteristics or codes listed in this publication. Should IBM modify its products in a way that may affect the information contained in this publication, IBM assumes no obligation whatever to inform any user of the modifications.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming or services in your country.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

All specifications are subject to change without notice.

Contents

Hardware

Software

Random Data

Trademarks

IBM, AIX, Micro Channel, Operating System/2, OS/2, Personal Computer AT, AT, Personal System/2, PS/2, PROFS, and RT are registered trademarks of International Business Machines Corp.

AIXwindows, AS/400, Assistant Series, Audio Visual Connection, AVC, BookManager, C/2, COBOL/2, Communications Manager, Database Manager, Independence Series, Personal Computer XT, PC XT, Personal System/1, PhoneCommunicator, Portmaster, Presentation Manager, Screen Reader, SpeechViewer, Systems Application Architecture, SAA, SNA, System/370, VTAM, and 3090 are trademarks of International Business Machines Corp.

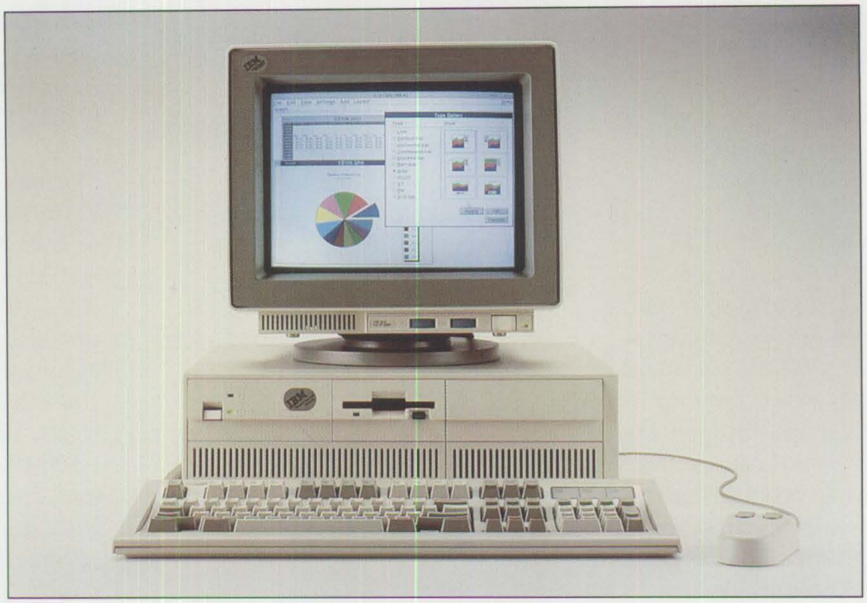
Aox is a registered trademark and MicroMASTER is a trademark of Aox Inc.
Apple and Macintosh are registered trademarks of Apple Computer Inc.
Adobe Type Manager and PostScript are trademarks of Adobe Systems Inc.
AT&T and UNIX are registered trademarks of AT&T.
AutoCAD is a registered trademark of Autodesk Inc.
COMDEX is a registered trademark of The Interface Group Inc.
EASEL is a registered trademark of Interactive Images Inc.
Intel is a registered trademark, and i860, 386, i486, 486, 80286, 80386, 80386SX, 80486 and ActionMedia are trademarks of Intel Corp.
Microsoft and MS-DOS are registered trademarks, and Windows and Word are trademarks of Microsoft Corp.
Novell and NetWare are registered trademarks of Novell Inc.
NextStep is a trademark of NeXT Inc.
OSF and Motif are trademarks of Open Software Foundation.
Xerox and Ethernet are registered trademarks of Xerox Inc.
X Window System is a trademark of Massachusetts Institute of Technology

IBM PS/2 Model 90 XP 486 and Model 95 XP 486

Thomas Bocchino, Louis Capps,
Mark Shipley and Gene Troskey
IBM Corporation
Boca Raton, Florida

Reprinted and adapted from
Innovations, *Issue 4, 1990,*
an IBM internal publication

This article describes the IBM Personal System/2® Model 90 XP 486 desktop system and the Model 95 XP 486 floor-standing system, which both feature the Extended Graphics Array (XGA) video subsystem and Small Computer System Interface (SCSI) storage devices.



PS/2 Model 90 XP 486 Desktop System with Display

The Model 90 Desktop System

The PS/2 Model 90 XP 486 is a family of three expandable desktop systems based on the 32-bit i486™ processor and IBM's Micro Channel® architecture. A unique attribute of the system is its processor upgrade capability provided by a removable processor complex. The 25 MHz system comes with an 80 MB or 160 MB fixed disk, and the 33 MHz model boasts a 320 MB fixed disk.

All models feature the new XGA graphics and 4 MB of 70 nanosecond (ns) standard memory. The system can be expanded to 32 MB of memory on the system board, thus saving an expansion slot that would otherwise be needed for a memory expansion adapter.

The system board has four 32-bit expansion slots for adapter cards and one direct memory access (DMA) parallel port. It offers two DMA serial ports, an advantage over other products that offer only one standard serial port.

The IBM PS/2 Micro Channel SCSI Adapter with Cache is a standard feature. It allows the attachment of up to seven SCSI devices from a single adapter that occupies one of the four card slots. Of the four DASD bays that are provided, one holds 5.25-inch, half-high devices, and the other three are for 3.5-inch devices. Two of the bays are available for expansion.

The Model 90 is intended for users of power applications. It speeds through complex financial analyses and other numeric-intensive applications. It runs power-hungry CAD/CAM and engineering programs that require high-resolution imaging and graphics. And it is great for desktop publishing and professional presentation graphics.

The Model 95 Floor-Standing System

The PS/2 Model 95 XP 486 is a top-of-the-line, floor-standing system for use as a LAN server, communications gateway or bridge, or as a multi-user host. It is also ideal for users of power applications requiring high performance and for users needing high capacity and expandability. Applications include design graphics, multimedia and financial modeling.

The Model 95 is based on the Micro Channel and the 32-bit i486 microprocessor. The 25 MHz system comes with a 160 MB or 320 MB fixed disk drive. The other model runs at 33 MHz and has a 320 MB fixed disk drive.

The PS/2 Micro Channel SCSI Adapter with Cache is a standard feature. There are a total of seven bays for storage devices, which can hold up to five 320 MB high-speed, SCSI fixed disk drives, for a maximum internal capacity of up to 1.6 GB. The bays can also accommodate CD-ROM drives, 5.25-inch diskette drives and tape back-up devices. The new PS/2 External Storage Enclosure for SCSI Devices (3511-003) can be added as an option to hold an additional 2.1 GB of fixed disk storage.

Like the Model 90, the system includes the same upgradeable processor complex and 4 MB of 70 ns standard memory. There are eight 32-bit expansion slots, two of which are used for the standard XGA Display Adapter/A and the SCSI adapter. There is one DMA serial port and one DMA parallel port on the system board. A 320-watt, worldwide power supply provides ample power for options.

Parts Commonality and Options

Considerable effort went into designing common parts for the desktop and floor-standing systems to reduce manufacturing costs and to ensure compatibility. The same two 80486-based processor complexes, one running at 25 MHz and the other at 33 MHz, are used in both systems. The 80486 microprocessor contains an internal 8 KB memory cache and floating-point processor unit. The addition of an optional 256 KB Cache card offers improved performance.

Another unique feature of the Model 90 and 95 is that a 3 MB protected partition on the fixed disk is reserved for system software.

IBM provides 2 MB and 4 MB 70 ns memory expansion kits so that the amount of memory in a system can be increased. The kits must be added in matched pairs because the memory subsystem design uses interleaving to provide zero wait state performance. The design does parity checking and allows single in-line memory modules (SIMMs) of different speeds to be used on the same system board. IBM announced that it intends to provide 8 MB SIMMs, when they become available, as an optional feature for both systems, allowing a total of up to 64 MB of memory on the system board. IBM stated that it intends for OS/2 to support the full memory capacity of the systems.

A 3.5-inch, 1.44 MB diskette drive is standard on both systems. A new 5.25-inch Slim High Diskette Drive option with electrical button eject can be used on either system. A unique selectable boot feature allows for booting from any drive.

The 2.3 GB Full-High SCSI Tape Drive option was announced for the Model 95 and the new 3511 External Storage Enclosure for SCSI Devices. For systems requiring the ability to save/restore, archive and distribute data, this high-performance (245 KB-per-second data transfer rate), high-capacity option provides an attractive solution.

The new 80 MB and 160 MB SCSI Fixed Disk Drive options can be used on Models 60, 65, 80, 90, 95 and 3511. The 80 MB drive has two disks, four heads, 17 millisecond (ms) average seek, and a 1.35 MB-per-second data transfer rate. The 160 MB drive has four disks, eight heads, 16 ms average seek, and a 1.50 MB-per-second data transfer rate. They both have 32 KB buffers and use synchronous SCSI mode operation.

Another unique feature of the Model 90 and 95 is that a 3 MB protected partition on the fixed disk is reserved for system software. When the system is installed, the entire Reference Diskette is copied into this partition, along with the image of the Power-On Self-Test (POST) and the Basic Input/Output System (BIOS). Afterward, at initial microcode load (IML) time, the system loads the BIOS program from the fixed disk into system memory. If the system encounters an error, or if the configuration is changed, the system automatically runs the IML code. This is much easier for the user because it eliminates the need to have the Reference Diskette available.

This IML feature also makes it easier to upgrade BIOS in the future. Instead of BIOS being in erasable programmable read-only memory (EPROMs) on the processor complex, it resides in a file format on the fixed disk. The Reference Diskette with the new image of BIOS is all that is needed for upgrading.

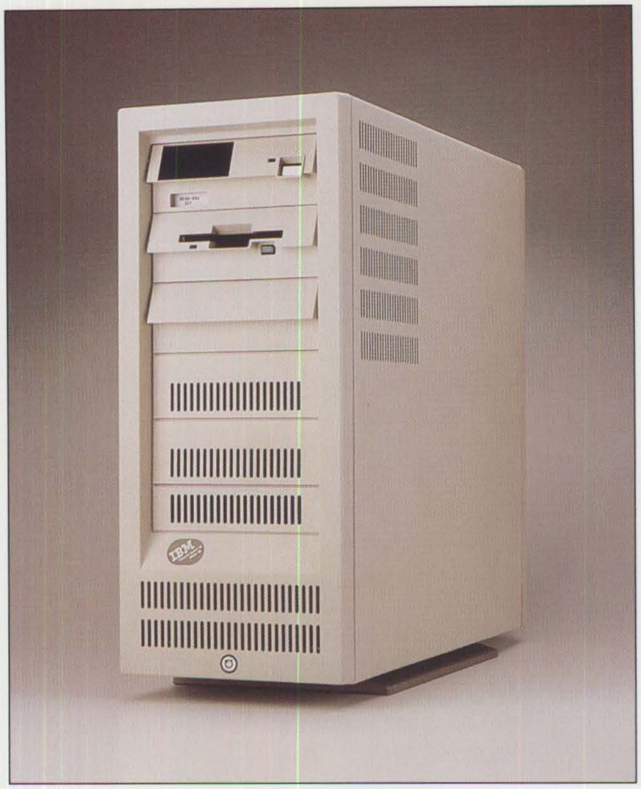
Competitive Strengths

One of the major competitive strengths of Models 90 and 95 is the removable processor complex. This provides investment protection for the owner because it extends the life of the system over time. As processor performance becomes critical to overall system throughput, the system can be upgraded with a new processor complex.

Another major strength comes from XGA as the new video subsystem for the PS/2 family of Micro Channel products. XGA is an evolution of the Video Graphics Array (VGA) offered on the first PS/2 systems. It is a high-performance, 32-bit DMA bus master that is compatible with existing VGA applications and with the 8514/A Adapter Interface. With a 14-inch PS/2 8515 Color Display or a 16-inch PS/2 8514 Color Display, it can display images in 1024 by 768 resolution with 256 colors. It also drives the other PS/2 monochrome and color displays.

XGA is optimized for use by window managers and other graphical user interfaces. It is significantly faster than VGA in a DOS environment, and its software improves performance even more in a windowed environment. The result is a graphics subsystem that more than doubles the number of picture elements while improving response time.

A PS/2 XGA Display Adapter/A is available as an option for Models 55, 65, 70, 80, 90 and 95. A PS/2 Video Memory Expansion Option can be added to the card to upgrade it to 256 colors at 1024 by 768; this maximum memory configuration is standard for some foreign countries. The adapter comes with device drivers for DOS, OS/2 Presentation Manager™ and Microsoft® Windows™. Numerous vendors intend to update their applications to work with XGA.



PS/2 Model 95 XP 486 Floor-Standing System

Models 90 and 95 are the fastest personal systems IBM has ever offered. IBM has made available a performance guide to summarize the results of its extensive performance measurements with various applications, operating systems and configurations. One example from the guide says that the Model 95-0KD runs AutoCAD®, Release 10, in a DOS environment with a performance improvement of up to 808 percent when compared to the reference Model 50-061.

Figure 1 shows the relative performance of DOS applications run on the new and previously announced PS/2 systems, compared to the PS/2 Model 50-061. The bar for each system indicates the range of performance for various applications from the slowest to the fastest. (The Personal System/2 Performance Guide, order number G326-0041, is available through IBM branch offices.)

The 320 MB SCSI fixed disk drive has a media data transfer rate of 2.0 MB per second with a very fast average seek time of 12.5 ms. It has a rotary voice coil actuator, closed loop servo, 1:1 sector interleave, a 64 KB buffer and self-diagnostics. The 32-bit bus master SCSI adapter has a maximum SCSI data transfer rate of 5 MB per second.

Other PS/2 Models

Other PS/2 models use larger-capacity fixed disk drives. The Model 65 SX uses the 320 MB SCSI fixed disk, making it an ideal low-end LAN server for a departmental network. The Model 80 386 20 MHz and 25 MHz versions using the 120 MB fixed disks are being replaced by new models using the 160 MB fixed disk; it also offers a new 80 MB model to replace the original 60 MB model.

On the other end of the scale, a new Model 55 LS was introduced without diskette drives or fixed disk drives for the LAN market, although data storage devices can be easily added as options. It comes preconfigured with an 1133 Token-Ring 16/4 Adapter/A or a 1091 Adapter/A for Ethernet networks. This makes installing the system in the network as simple as plugging it in. The user's files and programs are protected by a pre-installed remote program load feature.

Operating System Support

Operating system support is provided by Disk Operating System (DOS) Version 3.3 and Version 4.0, OS/2 Standard Edition 1.2 and 1.3 and OS/2 Extended Edition 1.2 and 1.3. LAN server support is provided by OS/2 LAN Server 1.3 and popular alternatives from independent software vendors.

OS/2 Version 1.3 requires only a 2 MB memory base, yet runs up to 25 percent faster than earlier versions. Users who may have needed 3 to 7 MB of memory can now run with 2 to 6 MB, saving 1 MB across the range. Users who previously needed more than 8 MB of memory can now save 2 MB. With a new selective install feature, the user can leave out functions that he does not need when installing OS/2 on his fixed disk; this can save up to 4 MB of fixed disk storage.

In addition, OS/2 1.3 features a built-in Adobe Type Manager™ that facilitates the addition of scalable PostScript™ fonts for display and printer support. This provides the ability to display and print text of any size. Another convenient feature for programmers is that the REXX procedures language previously available only in the Extended Edition of OS/2 1.2 is now available in the base version, OS/2 Standard Edition 1.3.

DOS 3.3 and OS/2 1.1 support up to two fixed disks; DOS 4.0 and OS/2 1.2 support up to seven fixed disks and OS/2 1.3 supports up to 24 SCSI storage devices.

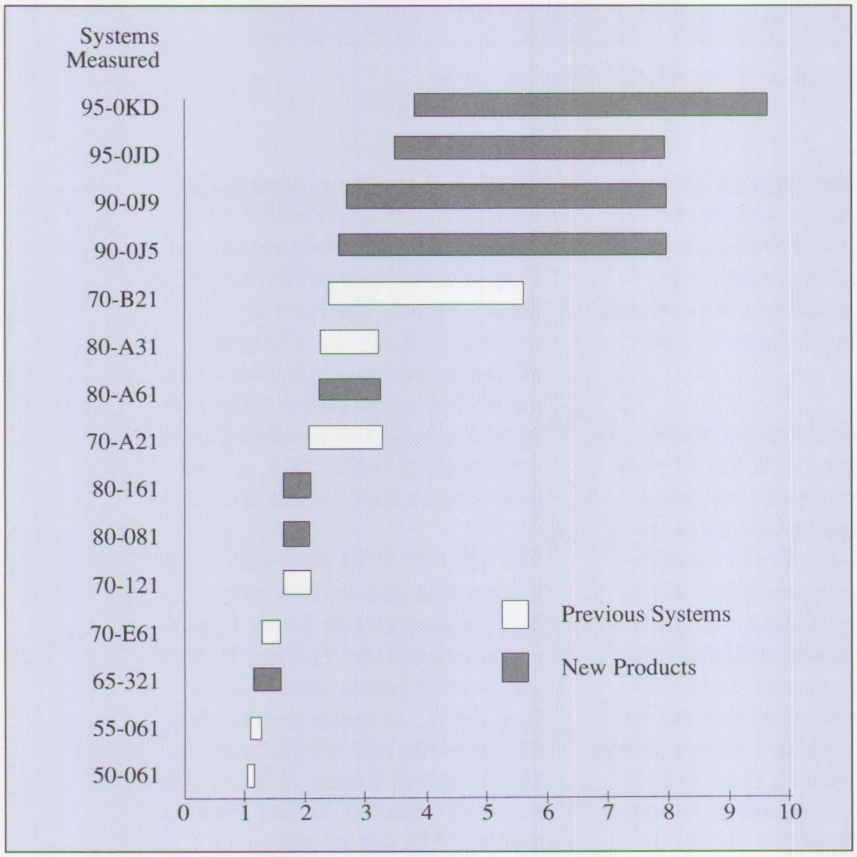


Figure 1. Overall Performance in the DOS-Based Environment

A Novel Approach to System Board Design

The PS/2 Model 90 XP 486 and Model 95 XP 486 were designed to meet customers' requirements for enhanced video graphics, memory and media expansion, processor versatility, and to provide system longevity.

I/O Enhancements: Both system boards, shown in Figures 2 and 3, contain new, enhanced serial and parallel ports with direct memory access (DMA) capability. This feature significantly increases the throughput of the system by allowing the I/O controller to transfer blocks of data, rather than single bytes. This frees the system microprocessor to work on other activities while the I/O information is being sent or received. By using DMA, system interrupts that occur after blocks of data – rather than single bytes – are ready to transfer.

The serial ports also have a new high-speed data capability to complement the DMA feature. The ports can transmit and receive data at 345.6 kilobits per second (kbps) – significantly faster than the 19.2 kbps provided by previous designs.

The Model 90 also has a second (9-pin) serial port because many configurations require a second port for modems or serial plotter devices. Model 95 also has additional Micro Channel slots in which a second serial card can be installed.

Keeping Time: The time-of-day (TOD) clock was implemented using an alternative technology, which consumes much less power, requiring a smaller-capacity battery to retain the data. The design uses almost 100 times less power in standby mode, allowing the use of a "coin-cell" type battery, which is less expensive and easier to replace. The battery life is also much longer, requiring replacement only every five to six years instead of the two to three years required by most existing designs. The size of the non-volatile RAM (NVRAM) has been increased from 2 KB to 8 KB in anticipation of future requirements.

Diskette Enhancements: Several enhancements were made to the floppy disk subsystem. The new Intel® 82077 floppy disk controller provides high reliability and lower cost, while out-performing its predecessors. The controller has DMA capability, with an internal 16-byte first in, first out (FIFO) and internal phase-locked loop. The output buffers are located on-chip, eliminating the cost of external drivers. The DMA feature is software-controlled, allowing the controller to run in interrupt mode if necessary. The internal phase-locked loop is designed to switch automatically between data rates, which allows the use of different devices such as 5.25-inch diskette and tape drives.

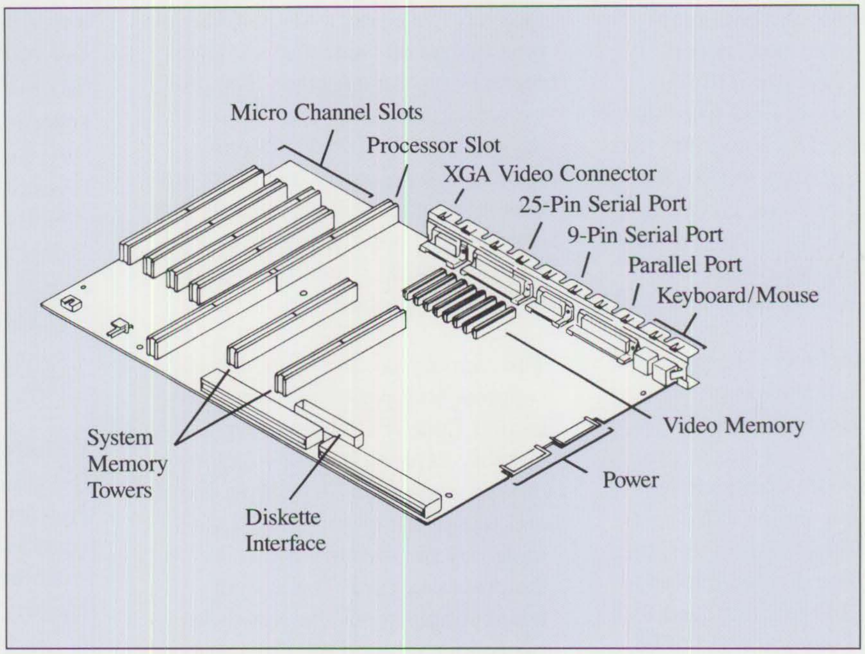


Figure 2. Model 90 XP 486 System Board

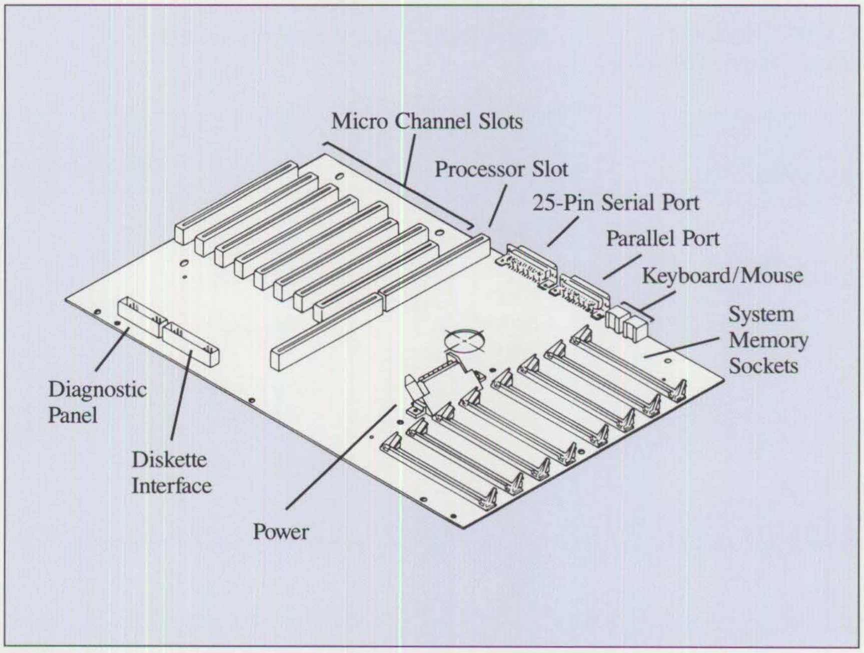


Figure 3. Model 95 XP 486 System Board

Selectable Boot: A "boot-mapping" function has been added, which allows the system to start from any of the installed fixed disks or floppy disks. This important feature allows customers to boot the system from any drive, including the 5.25-inch floppy. The boot sequence is configured by the user during setup, and stored in NVRAM.

Common Interface: The common processor card feature is probably the most differentiating aspect of the design. The intent was to provide a system with upgradeable processing, without adding extra cost to the base product. To do this, the system had to be divided into processor versus I/O function, and then a logical breaking point had to be defined (Figure 4).

The processor interface has been kept simple to allow flexibility in the processor card design. The interface has two parts, a Micro Channel-type bus on one side and a system memory bus on the other. The processor card contains the "brain" of the system: the DMA controller, memory management, the microprocessor and cache. The system board contains I/O ports, interrupt controller, TOD clock, system timers, and the diskette controller.

The memory portion of the interface supports both parity and Error Correction Code (ECC), up to eight memory expansions in a wide array of sizes and speeds. The type, speed and size of the memory are determined by the memory controller on the processor card. The system board simply buses the signals between memory and the processor card. Memory is located on the system board in Model 95, and on "tower" cards in Model 90. The tower cards make the most efficient use of the available system space. The vertical design provides two very important benefits:

- Memory upgrades are easier than on previous systems. The riser cards on the Model 90 are directly accessible with the system cover removed.
- Reliability is also improved because memory runs cooler by being in the direct channel of airflow.

A Platform for Future Systems: The system boards of these new systems provide significant improvements to previous solutions. The versatility of these designs assures that they will be platforms for future systems.

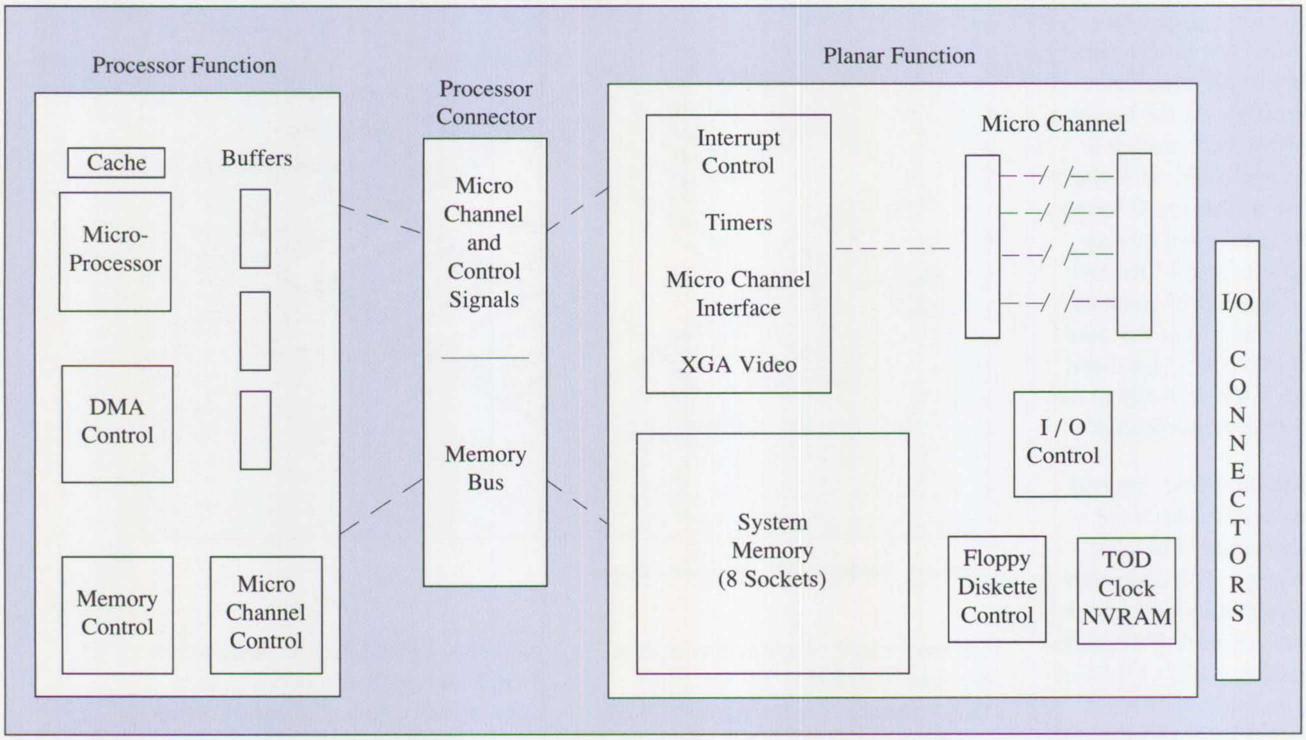


Figure 4. Common Interface for Models 90 and 95

Electrical Characterization of the Models 90 XP 486 and 95 XP 486

To develop the fastest, most reliable PS/2s to date, the development team gave special focus to the area of signal quality. High-speed digital designs must take into account the physics of sending a high-speed digital signal across a printed circuit board.

At sufficient transition speeds, sending a pulse of electricity down a printed circuit board wire is like dropping a stone into a small pond. Ripples are generated, which can bounce back and forth in the line before "dampening" out. This phenomenon is referred to as "transmission-line effects." In a pond, all that happens is a couple of fish are frightened, but signal ringing and undershoot in an electrical circuit can cause system malfunctions and even component failure.

Electronic Eyes: To isolate and eliminate signal quality problems in a PC design, it is first necessary to "see" them. After building the physical prototype, the development team characterized the prototype's electrical parameters, giving special emphasis to timing, ringing, undershoot and overshoot.

The newest "weapon" available to the designers for performing this task is the digital oscilloscope. The digital oscilloscope differs from the older analog scopes in that the incoming waveform is digitally sampled and stored in memory. This allows greater freedom to both trigger on a specific waveform of interest and to manipulate the captured waveform on the screen.

Now That's Fast! Since no single type of oscilloscope can perform every task optimally, two types were used. One has a very wide bandwidth; the other has a high sampling rate. Bandwidth is the frequency at which the instrument attenuates the incoming signal 3 db. The bandwidth of the digital oscilloscopes used to characterize the Model 90 and 95 systems was 1 GHz. This scope, effective for measuring repetitive signals, was used the majority of the time. However, because of its lower sampling rate (10 million samples per second), this scope was not used to capture single events.

To capture single events, the team used what is known as a "single shot" digital oscilloscope, which has a very high sampling rate (1 billion samples per second). This instrument is useful for capturing single events and displaying them accurately, but does not have the wide bandwidth of the repetitive oscilloscope.

A Probing Issue: Another critical aspect of performing high-speed signal measurements on the new systems was the type of oscilloscope probes used. The oscilloscope probes can have a tremendous effect on the waveform that appears on the screen. In order to minimize the effect on the circuit to be measured, the probe capacitance should be kept to the absolute minimum, along with the ground lead inductance. Depending on the type of oscilloscope, either active probes or 100:1 passive probes were used. Both probes had capacitances below 2.5 pf. Additionally, special ground clips were used to keep ground lead inductances to minimal values.

Quality Leadership: Special attention was given to the details of signal quality by the development team. The end result was that the team was able to identify early in the prototype stage which signals needed special attention to improve system reliability. Changes to the final level of printed circuit boards included rerouting certain nets to decrease the signal flight time, rebuffering, or adding termination to the nets. This effort will show up in the form of improved reliability.

Advanced i486 Processor Complexes

The IBM PS/2 Model 90 XP 486 desktop system and the Model 95 XP 486 floor-standing system use new technology and interface techniques to optimize performance of the 80486 microprocessor.

The new systems offer greater levels of flexibility with a "processor complex" upgrade strategy that allows a wide range of processor performance in both the floor-standing and desktop systems.

Standard on the Model 90 and 95 is a high-performance 25 or 33 MHz 80486 processor complex. A 33 MHz 80486 processor complex upgrade option for the 25 MHz systems is also available.

Processor Complex: The 25 and 33 MHz 80486 processor complexes take advantage of the new Processor Complex Interface available on the Model 90 and 95 system platforms to offer a memory subsystem optimized for the 80486.

In addition to Multi-Level Memory Interleave, these processor complexes support a 256 KB second-level cache for improved performance in a multitasking, multiple bus master environment. The processor complexes also improve system usability by supporting automatic single in-line memory module

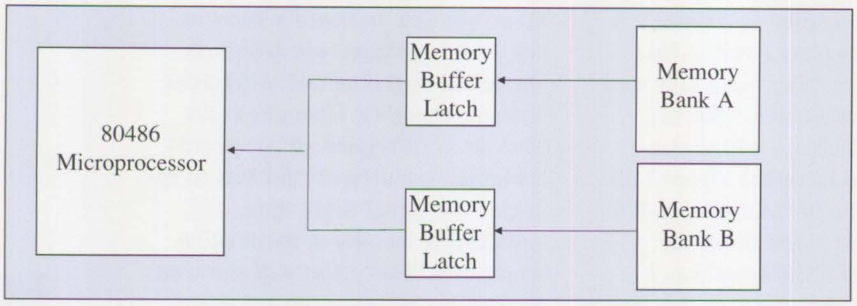


Figure 5. Two-Way Banked Memory Interleave Architecture

(SIMM) configuration, SIMM-specific parity error logging, and the initial microcode load (IML) software strategy.

Common parts used between the 25 and 33 MHz processor complexes and the ability to use the same processor complex in either the Model 90 or 95 system have reduced manufacturing assembly and debug time for four different system configurations.

Multi-level Memory Interleave: The 25 and 33 MHz 80486 processor complexes use features of the new Processor Complex Interface and a custom complex memory controller to implement two types of memory interleaving that optimize processor performance when accessing system board memory in a multiple bus master system.

The first level of memory interleaving is two-way banked memory interleaving, which improves performance by reading two 32-bit banks of memory simultaneously and then feeding the data to the 80486 one 32-bit doubleword after the other (Figure 5).

When a 32-bit doubleword is read from one bank, the other bank is also read, and its results are temporarily stored in a memory buffer latch.

If the 80486 is reading consecutive memory locations, the next 32-bit doubleword will come from the memory buffer latch without the added delay of a memory access.

Since the 80486 processor uses the new burst read technique (not available in the 80386), it typically requires four consecutive doublewords of data to fill its cache and instruction prefetch queues. The combination of the 80486 burst cycle and the two-way banked memory interleaving allows the 80486 to read four doublewords in the time it would take to do four independent reads.

Second-level memory interleave is active only when a bus master "owns" the Micro Channel. This type of interleave allows alternating simultaneous system board memory requests from the current bus master and the processor.

Earlier PS/2 systems locked the processor out completely from system board memory as soon as a bus master gained control of the Micro Channel. The processor could run out of the cache during this time on systems with cache, but as soon as there was a cache miss or a write cycle to system board memory, the processor was halted until the bus master released the Micro Channel to the processor. The 25 and 33 MHz processor complexes can interleave system board memory accesses from the processor with system board memory accesses from the bus master during the time the bus master "owns" the Micro Channel instead of locking out the processor from system board memory. This type of interleaving increases the concurrency between the processor and bus masters.

Figure 6 shows the performance gained using second-level memory interleave. While the CPU performs network searches in memory (CPU cycles), data can be transferred concurrently to a requester on the network with a communications bus master card (bus master cycles).

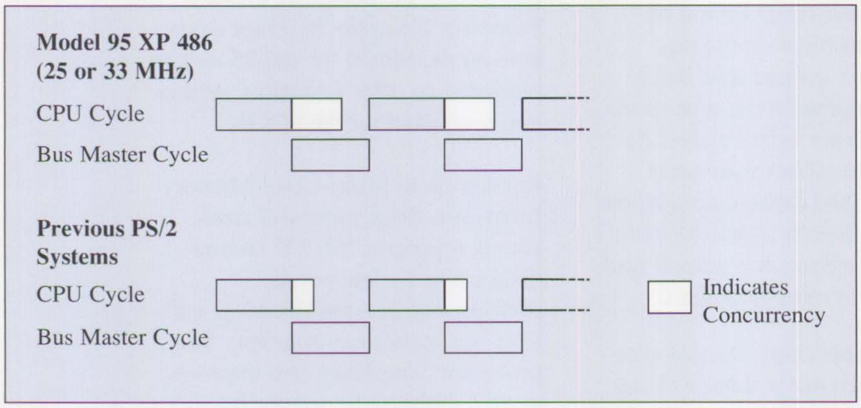


Figure 6. Increased Processor and Bus Master Concurrency in an Environment Where the PS/2 System Is Used as a Server

Second-level Cache Provides Second Try: In addition to the 8 KB cache that is integral to the 80486 processor and standard on the 25 and 33 MHz processor complexes, a 256 KB second-level cache is offered as an option for either processor complex. The optional cache provides a wider area of fast storage for improved performance in multitasking operating system environments.

When the cache option is installed, any read cycles that miss in the 80486 internal cache are forwarded to the optional cache for a second try. If the data is present in the optional cache, the 80486 reads the information in zero wait-state cycles. The second-level cache uses a two-way set associative algorithm and can transfer four 32-bit doublewords to the 80486 (using burst read) in five cycles.

Pick a SIMM, Any SIMM: The 25 and 33 MHz processor complexes support a variety of parity-check SIMM types with speeds ranging from 70 to 85 nanoseconds and sizes from 1 to 8 MB per SIMM. This flexibility, along with the eight SIMM sockets provided in Models 90 and 95, allows the system board to hold from 2 to 64 MB of high-speed, interleaved cached memory. Automatic SIMM configuration shields the user from complex set-up procedures by using a software algorithm to poll the SIMM types and configure the memory controller for optimum performance based on each SIMM's speed and size. The only requirement is that the user install SIMMs in matched pairs (both SIMMs in a pair are the same type) to support the two-way banked memory interleave.

For added performance, the 25 and 33 MHz processor complexes support dynamic memory speed adjustment to change the cycle time for a SIMM access on a cycle-by-cycle basis. So, if fast and slow SIMMs are installed in the same system, the fast SIMMs are accessed with shorter cycles than the slower SIMMs.

Automatic SIMM configuration shields the user from complex set-up procedures.

Improved Parity Checking: Parity-check errors on previous PS/2 systems locked up the system with no indication of which SIMM contained the failing bit. Now, the 25 and 33 MHz processor complexes implement SIMM-specific parity error logging by latching the select line for the SIMM that generated the parity error, allowing the non-maskable interrupt handler to save the SIMM socket number in battery-backed RAM so that when the system is restarted, diagnostics can inform the user of the failing SIMM. Future operating system support of this feature will allow the operating system to determine which tasks can be closed without error before shutting down the system.

BIOS Upgrade Capability: In the past, Basic Input/Output System (BIOS) upgrades were difficult to distribute because a change to the read-only memory (ROM) was required. The new PS/2 systems offer a new IML system software strategy that allows BIOS to be updated with a diskette, and setup and diagnostics to be run directly from the hardfile.

Previous PS/2 systems supported all of Power-On Self-Test (POST) and BIOS in ROM. The IML strategy present in the new systems supports two stages of POST code. Stage 1 of IML is located in ROM on the processor complexes and consists of only Stage 1 POST code. Stage 2 is located on the hardfile and consists of the second part of POST and all of BIOS. After the Stage 1 code is executed, Stage 2 is loaded and overlays Stage 1 in RAM. After the Stage 2 POST code executes, the operating system is started, or if an error occurred during any part of POST, setup is automatically run from a special system partition located on the hardfile, and the error code is explained. The system partition is new to PS/2 systems and allows part of the hardfile to be reserved by the system to hold setup, diagnostics and the Stage 2 IML code that contains BIOS and POST.

Parts Commonality: A continuing goal with PS/2 designs is to assist manufacturing with increased testability and decreased manufacturing overhead and debug time. The new processor complexes support this effort by having many common parts, including the ROM that holds the POST code. Because of this, not only does the same processor complex fit in either the Model 90 or Model 95 system with no changes, but the same second-level cache option will also fit on either the 25 or 33 MHz processor complex. This type of commonality offers manufacturing fewer hardware differences to support four system types and enables the user to upgrade from a 25 MHz to a 33 MHz processor complex without changing any other options in the system.

The new 25 and 33 MHz processor complexes used in the PS/2 Model 90 and Model 95 systems are a strong statement that IBM is creating new levels of performance and flexibility. The multi-interleaved memory interface, optional 256 KB second-level cache and 64 MB memory expansion on the system board show that the new processor complexes are high performers. Upgrade capability from a 25 MHz processor complex to a 33 MHz processor complex and multiple SIMM configurations protect the customer's investment. Improved SIMM isolation at parity error time and automatic boot of setup and diagnostics from the hardfile will increase customer confidence in system reliability. Finally, commonality between the 25 and 33 MHz processor complexes and the optional second-level cache improve manufacturability both at debug and assembly time.

The processor complex strategy is IBM's response to customer requirements for high performance and investment protection in its PS/2 systems.

Advanced Technology Demonstrations

The breadth and depth of the PS/2 system platforms and their tremendous growth potential and flexibility have been shown in a variety of demonstrations by IBM.

One demonstration was a glimpse into the future, using a Model 95 XP 486 based "Super-Server." It was connected to a Model 90 XP 486 and two Model 55 LS systems as clients on an IBM 16-megabit-per-second Token-Ring LAN.

The Model 95 XP 486 was configured with advanced technology options and an advanced version of OS/2 LAN Server 1.2 currently in beta test. It contained an Aox® Micromaster™ 80486 coprocessor and an advanced IBM bus master Token-Ring LAN adapter. Two 32-bit bus master SCSI adapters controlled nine 320 MB SCSI fixed disk drives, housed in the new 3511 DASD enclosure. They also controlled a 600 MB CD-ROM drive and an internal 2.3 GB full-high SCSI tape drive. All of this was protected with a vendor-developed, uninterruptable power supply.

One Model 55 LS was "rippled" (from RIPL, remote initial program load) or started up by transferring the OS/2 operating system over the LAN. The second Model 55 LS was started by transferring the DOS operating system over the LAN. In both cases, the program and data files remained on the server, not on the workstation; this approach leads to a greatly reduced workstation investment for the customer.

In another demonstration, the Model 90 XP 486 and the two Model 55 LAN stations simultaneously presented real-time, full-motion video data transfer at 30 frames per second over the LAN using the ActionMedia® Adapter. This demonstration vividly showed the power and flexibility of IBM's new personal systems, a rapid evolution from the first IBM Personal Computer, announced in August 1981, nine short years ago.

A Glimpse at the Future

Personal systems have come a long way in the last decade, and industry forecasters project equally awesome advances in the next decade. In 10 years, personal systems may be built with microprocessors containing 100 million transistors, executing 2 billion operations a second at clock rates of 250 MHz. Memory chips are likely to have from 256 Mbits to 1 Gbit each.

IBM will undoubtedly apply the same creativity and enthusiasm to these future challenges.

ABOUT THE AUTHORS

The authors are IBM engineers who have worked on the PS/2 Model 90 XP 486 or PS/2 Model 95 XP 486, or both, since the program's inception.

Thomas A. Bocchino is a staff engineer in the System Planar Development I department in the ESD development laboratory's Advanced Systems group.

Louis B. Capps is a senior associate engineer working in the System Development II department in the Boca Raton ESD development laboratory's Advanced Systems group.

Mark Shipley is a staff engineer in the System Planar Development I department in the ESD development laboratory's Advanced Systems group.

Gene T. Troskey is an advisory engineer in the High-End Program Management department of the ESD development laboratory's Advanced Systems group.

Choosing an I/O Bus Architecture

Patsy A. Bowlds, Ph.D.
IBM Corporation
Boca Raton, Florida

This article contains guidelines for choosing a personal computer architecture. It points out some of the issues and transition challenges. The article is based on a portion of a new book titled *Micro Channel Architecture: Revolution in Personal Computing*. Information about the book follows the article.

"Which bus to take?" is a prime topic for discussion in almost every organization and continues to be the subject of numerous articles in the trade press. This decision forms the basis for the purchase of billions of dollars of equipment in the '90s and will affect almost every aspect of information management in the coming decade.

The list of major contenders for small systems includes Micro Channel architecture, the AT® bus, the Extended Industry Standard Architecture (EISA), and Apple's® NuBus. The book on which this article is based contains an in-depth comparison of the technical merits of these bus architectures. What follows here are some useful guidelines for choosing a bus architecture for small systems, with emphasis on some of the nontechnical factors.

Most systems in use today are AT bus systems and industry estimates project that Micro Channel systems will continue to dominate the 32-bit bus environment through 1993. Therefore, in this article, the focus for managing transitions is on the relationships between these two buses.

Why Choose?

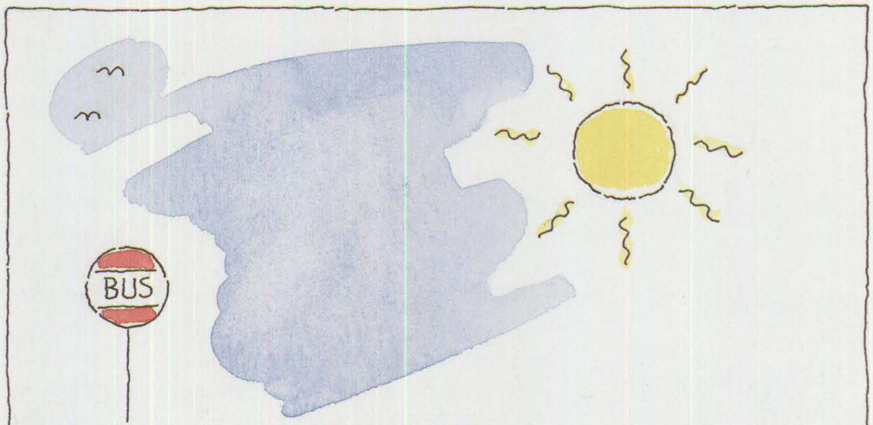
At this point, you may wonder "Why even make a choice?" The answer is driven by a number of factors that relate to how computers are used and technology advances. There are a number of forces in the early 1990s that will drive a change from earlier personal computer buses:

- The 386 and, later, 486 systems will replace 286 and 386SX systems as the entry-level workstation. (You don't have to look very far into the future to see this one!)
- High-frequency (≥25 MHz) 32-bit systems will dominate the mid-range, technical workstation and server environment with increasing presence of RISC processors.
- Multitasking 32-bit operating systems (OS/2, UNIX®) will use and require the features of advanced I/O bus architectures delivering significant performance benefits.
- I/O-intensive applications, such as page printing, multimedia, and image processing, will drive 32-bit I/O requirements at the personal workstation level.
- Migration of function from mainframe systems to "personal computers" (both stand-alone and networked) and advanced workstations will impose stringent reliability, data integrity, and fault-tolerant requirements.

These trends are already present to some degree. It is clear that today's purchase decisions should be made with these directions in mind, if at all possible. Even though the majority of systems used today are based on the AT bus, they simply cannot address these directions adequately. A choice must be made. What is the best choice, and when and how it should be made, are the real questions.

Disposable Computers?

One aspect that may concern you is the rate of obsolescence of equip-



ment. Why not buy the cheapest systems available and "throw them away" in a couple of years? Why plan for the future with today's purchases? You probably already know the answers. Look around and you'll see that all that "obsolete" equipment is still with you. You really don't throw it away, you just live with it. That's why it's so important to purchase equipment that complements your future direction instead of inhibiting it.

Which Bus to Take?

Part of the answer lies in the technical merits of the various choices that have been discussed at length in a number of publications. Most of these publications conclude that Micro Channel architecture has significant technical advantages. These conclusions, however, are somewhat rhetorical. In most organizations, it's not possible to instantly convert to a 32-bit bus standard regardless of its technical merits. Today's environments are a complex mixture of systems with widely varying user requirements. How to effectively use AT and Micro Channel systems in the same organization and manage the migration to Micro Channel architecture in the "real world" interests us all.

These topics will be especially interesting for MIS managers and those responsible for equipment purchases and inventory management.

The Ideal Computer Architecture

What makes an architecture good (or bad, for that matter)? The first requirement for a computer architecture is to define a set of rules that provide hardware and software compatibility for systems based on it. After compatibility, there are a myriad of attributes that are important. One of these attributes is affordability. The best architecture imaginable is of little use if it is not affordable for its intended applications. While it is desirable for a computer architecture to accommodate a broad range of features and functions, cost-effective implementations should be possible. Ideally, the higher cost elements of an architecture should be optional features that can be added without affecting compatibility with other implementations.

The limits of an architecture should be extendable in order to accommodate improved design concepts that may not be invented yet.

Another important attribute of a computer architecture is its data integrity characteristics, or ability to handle information without error. For example, an extra or a missing bit of information could mean the difference between $100,000 or $1,000,000 in a banking transaction – a nontrivial sum! An erroneous instruction can cause an addition to be performed instead of a subtraction. An incorrect address could result in a confidential message sent to the wrong person on a local area network. At the minimum, a computer system should be able to indicate that an error condition has occurred.

Some architectures focus on ease of use as a primary attribute. The software architecture of Apple's Macintosh® focused on a graphical user interface with mouse support to promote ease of use. Another example is found in Micro Channel architecture's automatic configuration feature, which simplifies the setup of systems and the installation of expansion boards.

An architecture should provide for expandability of a computer system. The ability to add performance or functions to a system is extremely important. Some examples of expandability in computer architecture include the number of expansion slots, supported I/O devices, multiprocessor support, and maximum storage capacity.

Another good attribute is extendability. The limits of an architecture should be extendable in order to accommodate improved design concepts that may not be invented yet. An example of this is the use of reserved pins in connector design. Reserving pins for future use allows for the addition of function that is not currently defined.

Flexibility is also an important attribute. An architecture should have a wide range of design and application alternatives. The original PC architecture lacked flexibility because it could be used only with the 4.77 MHz 8088, and its I/O bus could handle only 8-bit transfers. The consequence was that a major redesign was required for the IBM PC AT® with its faster 16-bit 80286 microprocessor. In addition to promoting a broad range of implementation possibilities, flexibility and extendability enhance the longevity of an architecture.

Another factor that enhances the value of an architecture is its acceptance by the industry and by a critical mass of users. One reason for the success of the original IBM Personal Computer was its acceptance by the industry and the standardization that it brought. Industry and user acceptance go hand in hand. The greater the number of potential users, the greater the financial opportunity for developers of add-on hardware and software applications. The availability of such products further increases a system's value.

A good computer architecture should provide a growth path and investment protection for the individual systems based on it. The ability to upgrade system performance or change its function enhances the value of a computer system and prevent its obsolescence.

Performance capabilities of an architecture are also important. This is probably the attribute most affected by the limits of an architecture. Adding higher speed processors, more memory, or advanced operating systems are examples of how an architecture can affect performance.

Reliability, availability, and serviceability (IBM uses the acronym RAS) are also important attributes of computer architectures. Reduction of failure rates and fault-tolerant capabilities are other benefits of a good architecture. Problems should be easy to diagnose and service should be quickly and easily performed.

A perfect architecture incorporating all of these attributes can never be achieved. Les McDermott, one of IBM's lead Micro Channel architects, says "The ideal computer architecture supports an infinite number of functions, has unlimited performance capabilities, and costs absolutely nothing." Trade-offs between cost and function must always be made. For example, an architecture suitable for supercomputers is unsuitable for personal computers. The challenge is to find the right balance.

Affordability	Flexibility
Compatibility	Industry Acceptance
Data Integrity	Investment Protection
Ease of Use	Performance
Expandability	RAS Characteristics
Extendability	

Figure 1. Computer Architecture Characteristics (not prioritized)

An interesting exercise is ranking the characteristics of an ideal architecture. As you will discover, balancing the characteristics of the ideal computer architecture is a nontrivial task. But this is the same kind of exercise that IBM marketing, engineering, programming, product planning, manufacturing, and service engineers faced when they defined Micro Channel architecture. (These characteristics are shown in Figure 1.)

In order to answer the question of which strategic bus architecture to choose, you must first rank the importance of these attributes. There is no universal "right" answer, however. What is important to one company or organization may not be important to others. In addition, priorities of a given company may change over time. Also, these priorities are certainly different for different application environments.

If affordability is at the top of your list, you would probably choose AT bus entry systems for short-term value. For longer-term value, 16-bit Micro Channel systems are the best choice because of their investment protection features. On the basis of overall technical merits, Micro Channel is the winner.

In terms of IBM's priorities, data integrity was at the top of the list in developing Micro Channel architecture. Nothing is more important than data integrity in computer architecture. There was no room for compromise on this point. Compatibility across multiple platforms for years to come was also well up the list.

The best news is that Micro Channel architecture has all the characteristics on the list in both current and future systems. Although Micro Channel expansion boards are not physically compatible with the original PC boards, they are compatible with the software developed for the IBM PC and AT. This compromise, while carrying its own cost, provided a number of benefits related to all the other characteristics of an ideal architecture.

Micro Channel architecture provides a flexible, high-performance information transfer medium that can be implemented at low cost on personal systems. But it is also broad enough to be used by midrange and mainframe computer products. The architecture incorporates a flexible set of parameters that enhances its longevity by permitting a mixture of attachment devices that can operate at a wide variety of speeds. Its structure also minimizes the dependency on any single processor type or logic technology. The flexibility of the specifications also enables individual computer systems to be upgraded and enhanced. It is an open architecture that now has widespread acceptance by the computer industry with more than 1,000 expansion boards available from nearly 400 companies. Micro Channel architecture also provides better performance, reliability, and data integrity – important attributes in today's sophisticated multiuser and multitasking environments.

In terms of IBM's priorities, data integrity was at the top of the list in developing Micro Channel architecture.

The other bus architectures have some of these attributes. There are several good examples of systems based on NuBus, EISA, and AT architecture. But the benefits of Micro Channel architecture are much broader in their scope, affecting many more system implementations, and delivering benefits to a much wider range of applications.

Decisions, Decisions, Decisions

To better understand how business factors influence the choice of bus architecture, let's examine some typical situations:

Scenario 1: The Tick Tock Watch Company (a fictitious company) is under extreme cost constraints. It manufactures windup watches, having recently sold the quartz division to another company. Revenues are flat, and the company is in the process of reducing its number of employees. It has decided to make DOS the strategic operating system for workstations for the next few years. It has a hodgepodge of XT and AT clones from a variety of manufacturers. The company has a small inventory of AT expansion boards and parts.

Recommendations

This company is in trouble. It probably won't be in business in two years because of its obsolete product line unless it can be transformed to a growth business. Obviously, its priorities begin and end with affordability. This is a clear-cut case for purchasing workstations based on the AT bus for several reasons. The existing board and parts inventory can be effectively used. The choice of DOS as the operating system and the absence of LANs are also factors.

IBM has a competitive range of AT-bus systems for this company at an excellent entry price. Vendor selection is an extremely important criterion. For any new system purchases, this company should focus on vendors such as IBM whose products have a high resale value and whose staying power in the business is secure. This will minimize both cost and risk as Tick Tock gets its "works" moving again. If the company goes under in a couple of years, it will be selling all of its data processing assets. Another possibility that should be investigated is leasing equipment instead of purchasing.

Scenario 2: Med Tech, Inc. (a fictitious company) is a small growth company with rapidly increasing revenues whose primary business is designing and manufacturing innovative medical diagnostic equipment. Most of the profits are reinvested for future growth rather than distributed to stockholders. Its workstation strategy is to enable automation in every aspect of the business. Most of the employees are technical people who use their workstations for design and functional simulation of leading-edge equipment. The company has high processing requirements at the workstations. There are plans for numerous LAN servers in order to provide electronic mail services and shared graphical files of complex designs.

Recommendations

This is an easy one. This environment can benefit immediately from standardizing on the Micro Channel bus. Selection of technical workstations and servers from the RISC System/6000™ family for their UNIX applications is definitely recommended for Med Tech. Additional servers and workstations in the PS/2 system family or Micro Channel clones for business and office applications are recommended. This company needs considerable processing power and capacity at all levels, and should standardize on 386SX and higher Micro Channel platforms in order to run 32-bit applications. It's clear that performance and current and future compatibility are near the top of the list of priorities for its strategic bus architecture.

Scenario 3: The two preceding scenarios are, of course, quite atypical. Usually our choices are not so clearly defined. Our final scenario is more representative of organizations today.

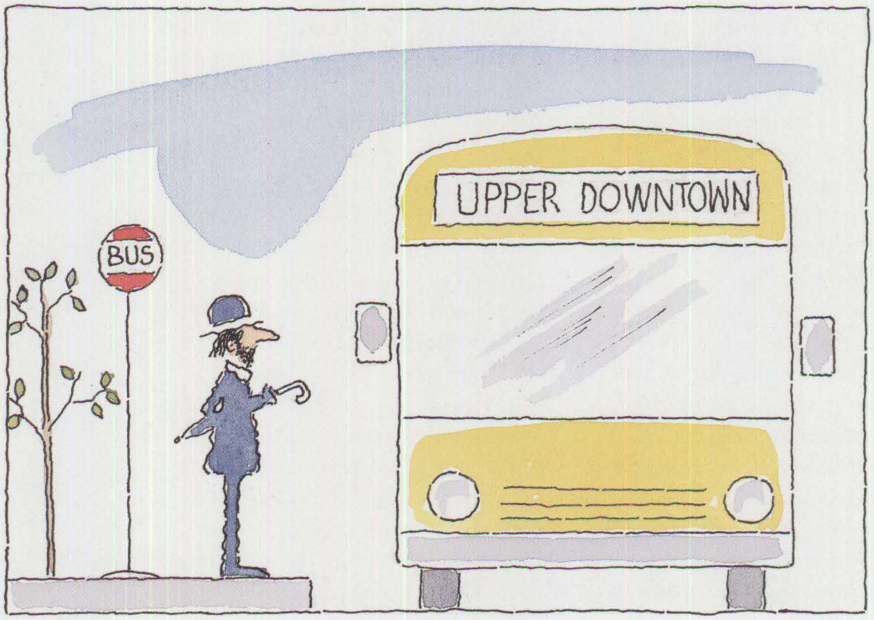
SMI International (a fictitious company) is a medium-sized company whose primary business is life and casualty insurance for United States servicemen and women. The company's installed base of equipment includes a variety of systems based on the PC and AT bus. SMI is planning to implement a LAN strategy at the department level, but will continue to use its IBM mainframes as a central data repository. Over the next few years, SMI recognized that OS/2 has numerous benefits for its environment, but plans to continue using DOS as the primary operating system at its workstations for the near future.

A strategic application for the future includes image processing to alleviate the vast amounts of paper handling and storage. In addition, a fully automated system using expert programs for policy issuance is planned. The automated system will require an easy-to-use graphical interface and will include a marketing presentation.

Recommendations

SMI International is on the verge of implementing some important applications that will contribute enormously to productivity and profitability. This forward-thinking company is poised to gain a significant competitive advantage because of its plans for automation, networks, and image processing. Reduction in marketing and service expenses and paper-handling represent enormous potential savings.

SMI should select Micro Channel architecture for all purchases immediately. Its strategic plans for LANs, OS/2, image processing and multimedia will be best served by Micro Channel systems. An economical choice for the workstation standard is a desktop 386SX system with a SCSI storage subsystem. The 386SX microprocessor can run 32-bit OS/2 when it becomes available. The Micro Channel bus delivers high-performance image processing



through the use of bus masters like IBM's Image Adapter/A. SMI's image strategy would be complemented by the SCSI bus master because almost all optical storage devices feature the SCSI interface. In addition, the systems can be upgraded using bus masters like the Aox MicroMASTER 486.

For the automated policy issuance program, a good choice would be a more powerful Micro Channel system with a multimedia marketing presentation using Micro Channel products. The Audio Visual Connection™ and a touch screen, and an image capture system with a removable SCSI read/write optical storage device are recommended for this application. This combination can be used to capture and store a photograph of the applicant, hand-written information, and the applicant's signature. A policy and wallet card could be issued at the time of application and put in force, pending verification of the applicant's motor vehicle or medical records and credit card information.

Micro Channel personal systems are a good choice for personnel engaged in actuarial research at SMI because of their ability to accommodate compute-intensive and data base "engines" like the i860™ Wizard adapter. Another excellent choice would be a Micro Channel technical workstation like one of the IBM RISC/System 6000 POWERstations.

A logical choice for the departmental LAN servers are 32-bit Micro Channel systems like the IBM PS/2 Model 95 XP 486 systems running OS/2. These I/O-intensive applications in a multitasking environment are a perfect fit for the performance and data integrity features of Micro Channel systems.

For existing inventory of AT-bus systems, there are a number of alternatives, including downgrading systems to less sophisticated applications. What this company should not do, however, is to postpone implementation plans for their strategic applications because of their existing inventory of AT-bus systems. The cost savings and productivity increases for the image processing system and automated policy issuance program will quickly pay for the additional equipment costs.

Bus Strategies

Let's look at three bus strategies and how they might fit in your organization. These are the choices:

1. Continue to invest in AT-bus systems for the foreseeable future.

2. Maintain both AT and Micro Channel bus architectures.

3. Make the transition as rapidly as possible to Micro Channel architecture systems.

The AT-Bus Strategy: The traditional PC/XT/AT bus does not use the full capabilities of 386 and 486 architecture and does not accommodate any 32-bit I/O devices. Under what conditions would you choose this bus? If your only uses now and in the future are DOS personal productivity applications, this bus strategy makes sense. This is especially true if you are willing to trade off a shorter equipment life for a lower initial price.

Stand-alone systems with limited performance requirements and single-tasking applications do not exploit the full potential of Micro Channel architecture. In addition, the inability to upgrade these systems with additional processing power dictates a strategy based on approximately a 2-3 year equipment life. This strategy may suit your requirements quite well in the near term.

In the longer term, however, this 16-bit strategy will necessarily give way to a 32-bit I/O bus, just as the 8-bit PC bus yielded to the 16-bit AT bus when the price/performance ratio became attractive enough.

A Two-Bus Strategy: For many purposes, a two-bus strategy using both AT and Micro Channel systems is a viable approach for the near future. The AT bus is adequate for 286 and 386SX systems for DOS applications in a stand-alone environment or as DOS network requesters with moderate performance requirements. It is also a good choice for laptop systems because of their limited expandability, shorter equipment life, and rapidly changing display technology. If this two-bus strategy is your choice, Micro Channel systems are appropriate selections for servers and for those users who require higher performance than the 386SX systems can deliver, especially for OS/2 or UNIX applications.

This two-bus approach is a good interim strategy, but is not viable in the long term. Processor power will continue to come down the cost curve, and a 32-bit I/O bus will be required to effectively utilize it. It is only a matter of time, however, until this strategy will give way to systems with a 32-bit I/O bus.

The Micro Channel Standard: There are benefits in having a single bus strategy in most environments. Even though few boards are moved from system to system, inventory management and purchasing power are important benefits of a single bus strategy. Another benefit lies in the reduction of support and software requirements. Multiple bus strategies can add unnecessary complexity in setup, service, and software requirements.

One of the advantages of Micro Channel architecture that adds value to managing the transition to the 32-bit environment is its 16-bit system and expansion board implementations. Because 16-bit systems and boards that utilize almost all of the advanced features of this architecture can be built, this brings value to a large number of end users.

A sensible transition plan, therefore, can be based on Micro Channel 386SX systems. This effectively positions you for the 32-bit future without immediately paying the price for the higher function. It also provides a good base upon which to build future applications like image processing and multimedia. Many 16-bit Micro Channel boards offer throughput that approaches that of 32-bit boards in some applications. Many advanced function boards available today are 16-bit implementations.

You should plan on the 386SX microprocessors being around for awhile. Faster chips and integrated functions will prolong their lives. So a near-term investment in 16-bit Micro Channel systems and boards is a good plan.

Micro Channel expansion boards are at the beginning of their useful life rather than near obsolescence. All Micro Channel boards have been designed and built since 1987 using modern electronics and VLSI circuitry. Many boards built for the AT bus were designed and built years ago with older, slower components that may reduce performance in newer systems.

Transition Planning

At this point, you are probably convinced that Micro Channel architecture represents the best choice of microcomputer bus architecture for the future. However, what is a sensible transition plan? Many companies have large inventories of AT boards and AT bus machines. Introducing a different bus architecture into this environment complicates inventory management and end-user support. Standardizing with Micro Channel systems, with a strategy of migration over time, requires some careful planning.

Some appropriate steps are:

1. New purchases should be limited to Micro Channel systems. If this isn't feasible, a good place to start is with users having high-performance requirements and for server applications. Establish Micro Channel systems as the workstation standard as soon as possible, using economical 386SX systems initially.

2. Identify key applications that effectively utilize the value of Micro Channel architecture solutions. Image processing, multimedia, compute-intensive, and server applications are excellent showcases for Micro Channel systems because of their dramatic cost savings and productivity improvements.

3. Develop a used-equipment plan for older systems. The time-honored method is to downgrade these systems to other users. This is still the best plan if it doesn't interfere with implementing key applications in your organization. Other alternatives are trading for new systems, selling the inventory, or donating to nonprofit organizations. Discuss these options with your accounting department.

A New Paradigm

Introducing Micro Channel systems to new users requires a new paradigm for performance. Most people think of performance in terms of raw CPU power: in order to get better performance, you need a new system with a faster CPU. For systems based on Micro Channel architecture with its multimaster capabilities, a paradigm of distributed processing is required: performance benefits are derived from expansion boards. In other words, high-performance, application-specific bus masters can deliver additional power to existing systems.

While this may not be quite as exciting as a new system, it is far more cost-effective and versatile (although it's hard to imagine people bragging about their new adapter!). In years to come, entire computers will reside solely on expansion boards.

Look to Tomorrow

Having a vision of the future is important in today's confusing environment. Knowing where you want to go is the first step to getting there. Managing transition in a changing environment is, perhaps, the most challenging of all tasks. Making the right decisions now and planning for the future are the tasks at hand. One thing is sure: a high-performance multitasking bus architecture is an absolute requirement for the workstations of the '90s.

ABOUT THE AUTHOR

Dr. Patsy Bowlds is an IBM senior engineer and technical consultant in the Entry Systems Division. She is an internationally known lecturer on Micro Channel architecture, having presented to thousands of IBM's customers, technical advisors, and industry groups around the world. She is currently responsible for the technical support of IBM's marketing channels on behalf of the Boca Raton Laboratory. In her 17 years with the company, Pat has held a variety of management positions, including manager of strategic systems planning for the Boca laboratory. She is the author of several technical papers and holds patents and technical disclosures in various fields. She was awarded IBM Outstanding Innovation, Outstanding Technical Achievement, Excellence, and Invention Achievement awards. She received her Ph.D. from the University of Alabama and her B.S. from St. Bernard College.

The book, *Micro Channel Architecture: Revolution in Personal Computing* is a description of the features and benefits of Micro Channel architecture and how it can be used in today's and tomorrow's environments. The book includes a description of Micro Channel systems available from IBM (including the PS/2, RISC System/6000 and Micro Channel/370) and their performance characteristics. Micro Channel implementations from other companies are also described. It describes high-function Micro Channel adapters that address key application areas like image processing, multimedia, and servers. The author assesses other 32-bit bus architectures like EISA and NuBus and offers advice for making choices. Included are a chapter on the basics of computer architecture and design, and an extensive glossary. The foreword is by IBM Vice President Dr. Robert L. Carberry, one of IBM's most respected scientists.

Micro Channel Architecture: Revolution in Personal Computing is published and copyrighted by Van Nostrand Reinhold, 115 Fifth Avenue, New York, NY 10003. To order the book, call 1-800-926-2665. The ISBN number is 0-442-004338.

The Network Is the Message

Chet Heath
IBM Corporation
Boca Raton, Florida

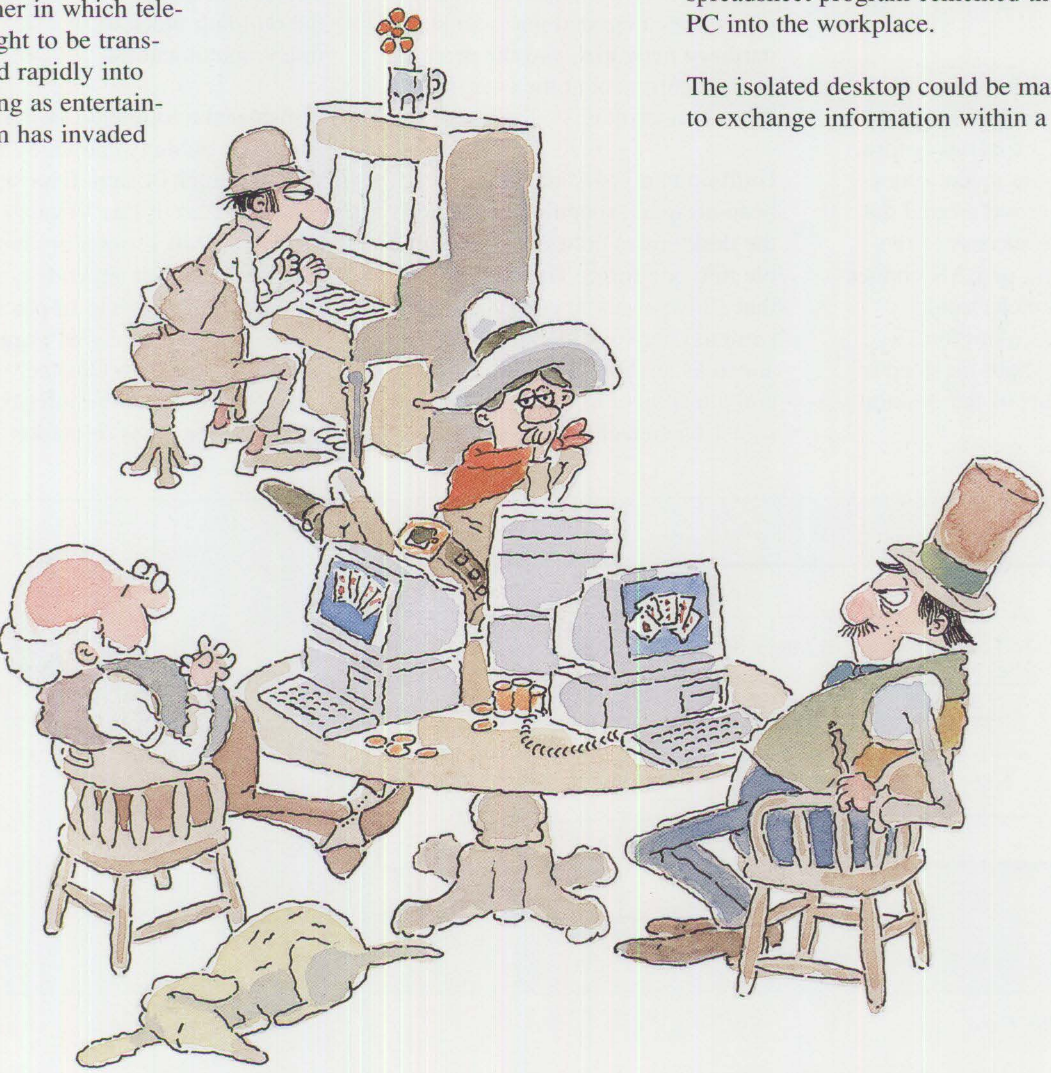
This description of a concept called "workgroup computing" is reprinted from the recently published book *The Micro Channel Architecture Handbook* by Chet Heath and Winn L. Rosch. Information about the book follows the article.

A Busmaster Example

Not unlike the manner in which television allowed thought to be transmitted and converted rapidly into popular understanding as entertainment, a new medium has invaded the workplace – this time to efficiently allow us to coordinate our effort and direction when groups work toward common goals. Not insidiously, but silently, we absorbed the new philosophy by an evolutionary adaptation of our daily work habits. This "Trojan horse" of technology is the workgroup system. Its soldiers are made of software and silicon, and more than ever, they will free men to think rather than work.

When personal computers were first accepted in the workplace, they were perceived and promoted as individual tools (sometimes toys) of productivity. They followed "dumb" terminals that were dependent upon the vitality of distant resources, by providing the individual means required to accomplish routine desktop drudgery fully under the control, and sight, of the user. Beyond that, they did not imply large investment and could be justified by the improved productivity of the individual working alone. Simple individual tasks, such as composition of a letter, filing, or income tax preparation, justified the systems in the workplace and occasionally in the home. Perhaps most of all, the ability to reduce complex, numerically based decisions by way of the spreadsheet program cemented the PC into the workplace.

The isolated desktop could be made to exchange information within a



group or office by physically exchanging the storage medium, the diskette. The inexpensive incorporation of the hard disk, with an initial capacity of 30 diskettes, led to local manipulation of information larger than the capacity of a single diskette. Further, the desire to control information located in the distant systems and to interact with that data led to a need to communicate in real time. The first connection was to the very same distant resources from which the desktop had earlier declared independence. But then, more slowly, it spread *between* the desktops, as workers became more dependent upon the output of their peers than on the distant resource.

The workers needed to network locally, much as medieval knights had in "round tables" centuries before them, or, as in more recent times, workers had convened around the water cooler. The success of the local area network, or LAN connectivity, between the electronic desktop systems is, therefore, a strong statement about the need of the modern knights of the desktop to band together to achieve common goals.

Why, if humans tend to herd together, are not personal systems initially designed as integrated workgroup systems rather than stand-alone boxes? It is because, like many inventions before it, the personal computer had to develop an individual following before it could be integrated into society. Airplanes preceded airports and airlines, electric lights glowed from batteries before power stations were established, and Bell spoke one-on-one with Watson long before the switchboard came along. Of course today air travel, electric light, and the telephone are integrated into vast standardized networks, and the design of the instrument at the user interface assumes that.

Until now, no personal system had been designed specifically to allow the desktops to collectively assemble into one force. The connectivity that allows users to congregate in a common network has so far been added as an "afterthought" peripheral attachment to stand-alone designs. The attachment was either through a bus-attached card, or similar logic on a system board, or outboard of the system entirely, through a peripheral port. It is now time for the personal system to integrate the interconnectivity of the network into its basic design. PS/2 systems with Micro Channel architecture have done exactly that.

When Micro Channel personal systems were first announced, it was stated that much of the new concept was very subtle in nature. One component, peer-to-peer bus master communication, was an element of that subtlety. Ironically, peer-to-peer implies interaction between individuals, like Bell and Watson, yet it is the building block of intercommunication.

Consider the following design concept, first publicly shown by IBM at the Computer Dealers Exposition (COMDEX®) in Las Vegas, in the fall of 1989. It is not a product, but it could be, and it depends heavily on the foundation laid in place within PS/2 systems that implement standardized Micro Channel interfaces. It should not be inferred that IBM has any plans to market such a

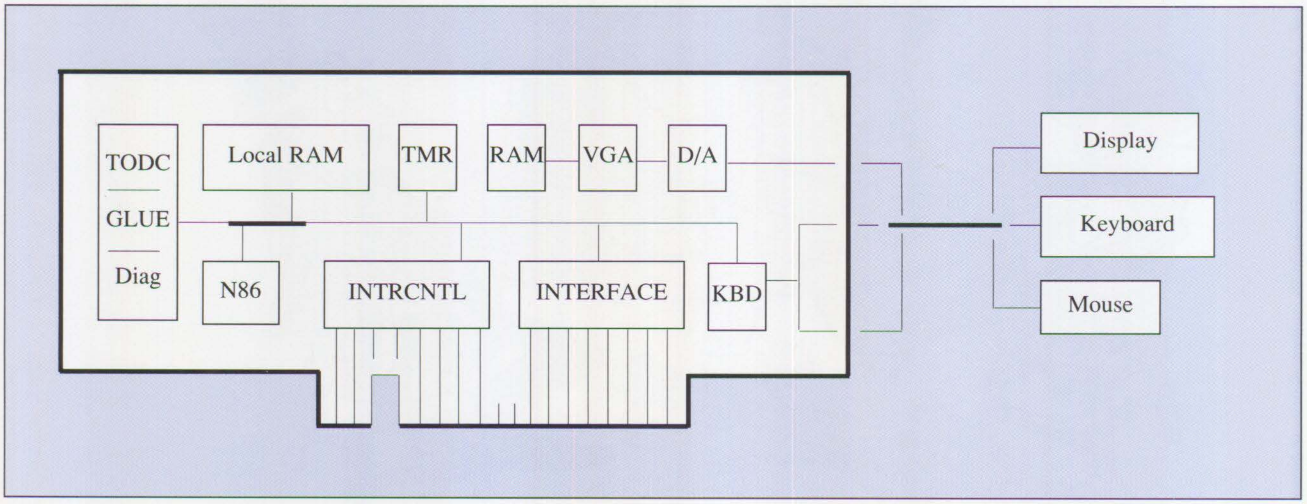


Figure 1. The GUESS Card

system; it is shown here merely as the author's example of the flexibility of Micro Channel architecture.

During the initial definition of the Micro Channel interface, the author had envisioned six conceptual tests of the extendability of the personal system. In one, the basic computer and its associated subsystem support components (such as interrupt controller, time-of-day clock, and system timer) would be assembled along with sufficient memory to contain an operating system and applications on a single card (Figure 1). Also resident on the card is the basic user interface of keyboard, mouse, VGA (or better) display, and compatible diagnostic ports, to make the "system on a card" serviceable.

The basic user interface is available from standard connectors on the rear bracket to the existing VGA, keyboard and pointing devices. It has been shown elsewhere that these interfaces can be propagated hundreds of feet.

As a bus master, this card could form the compatible heart of a system, requiring only the addition of disk, diskette, connectivity and printer interface on a system "Combo" board with power supply and cabinet for completion as in Figure 2.

This, in itself, is an exciting concept, but only begins the possibilities of such a design point. In a sense, the card is a system, less all the previously mentioned elements. To software, it forms a Generic User Environment of a Small System (GUESS).

Now plug a number of such GUESS cards into any PS/2 system with Micro Channel architecture to form a cluster of users in close proximity, as in Figure 3.

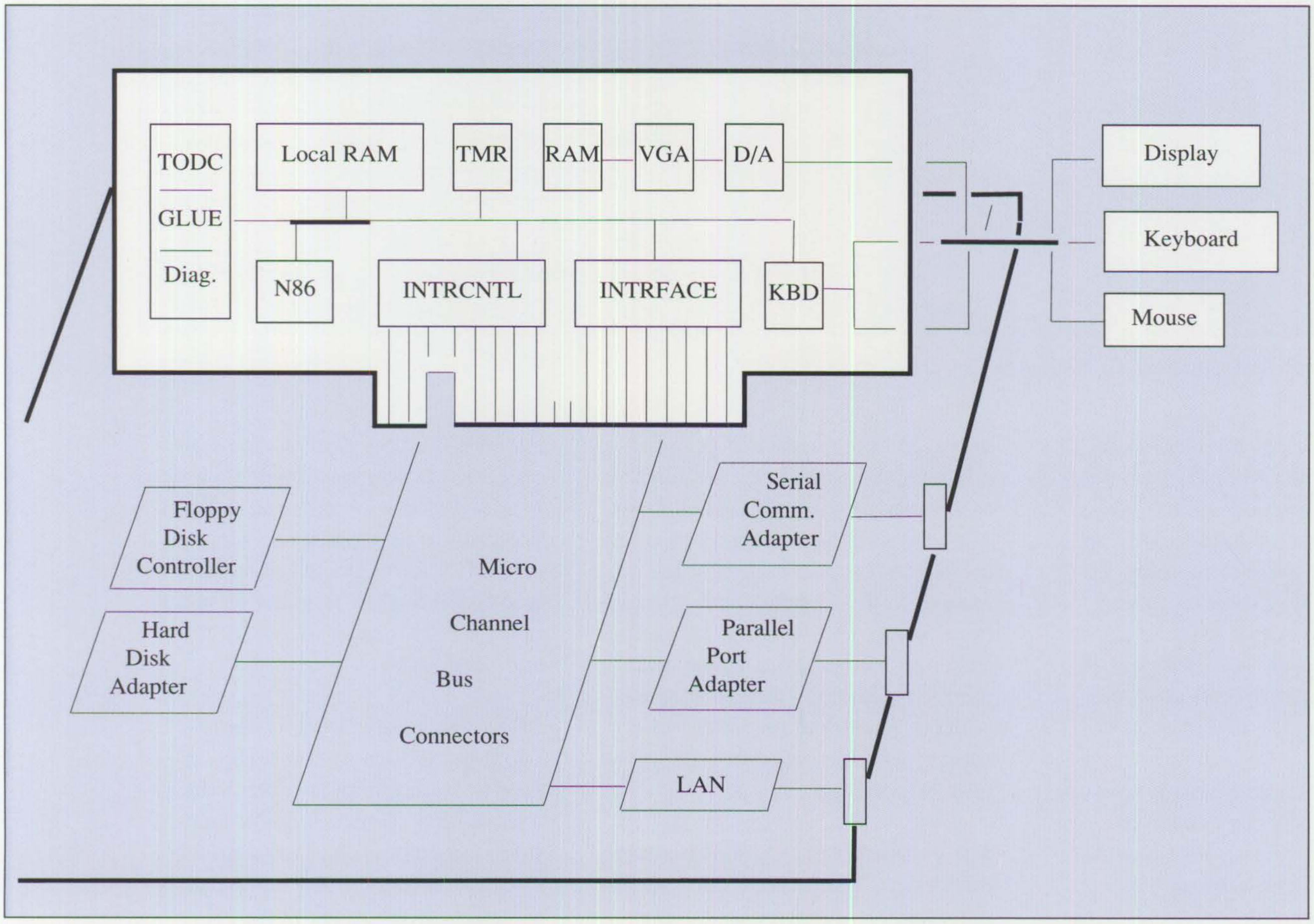


Figure 2. A Small "Entry" System Based on a GUESS Card

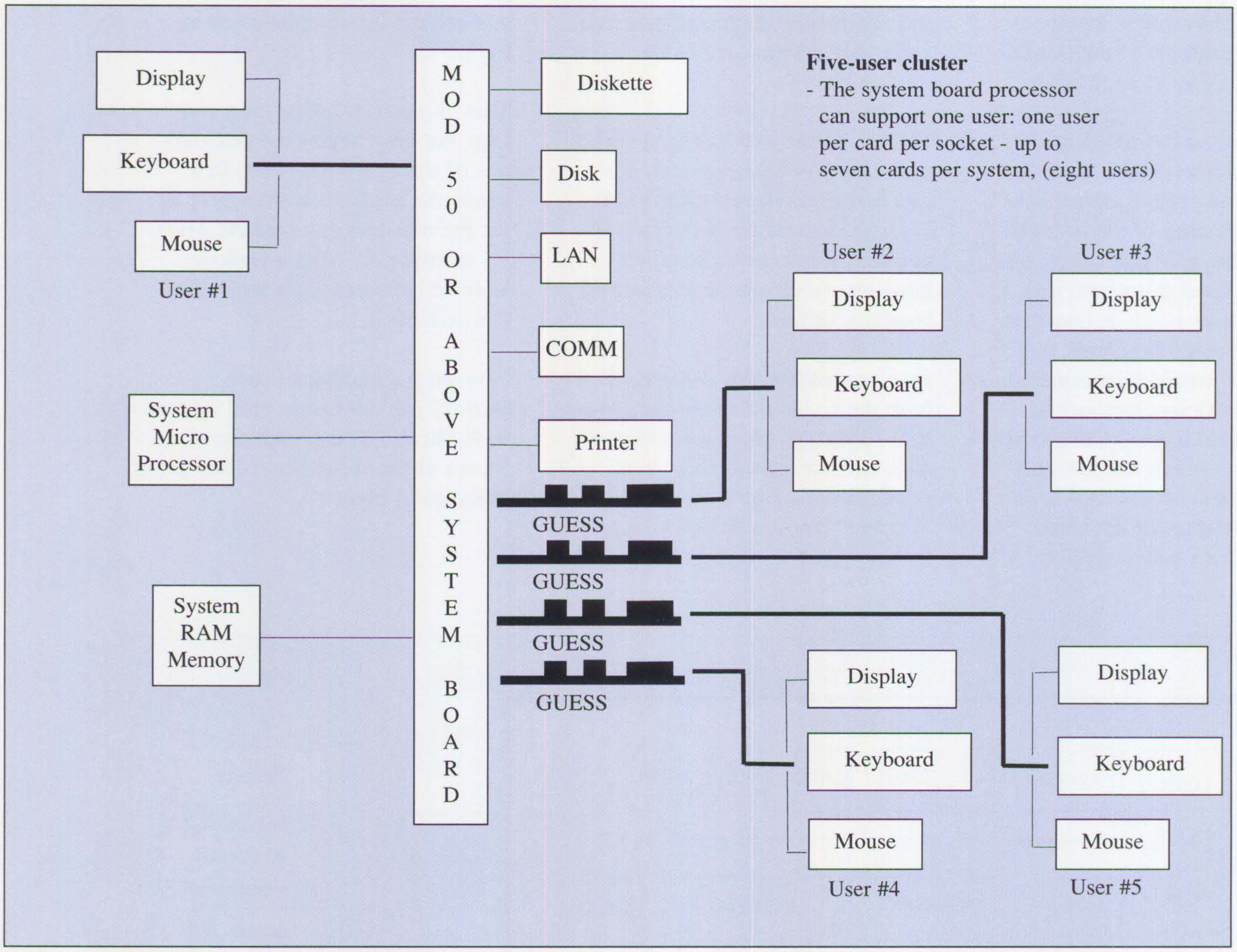


Figure 3. Cluster Host with Four GUESS cards

Because the cards are bus masters, and because the Micro Channel defines public memory on the bus or peripheral, each GUESS card can read from or write to memory located in the system or on other cards, even on other GUESS cards. The last property is called peer-to-peer bus master operation on the Micro Channel.

The concept could be made to work in a limited way for AT-based systems without peer-to-peer bus master communication. In such a system, a common system resource is assigned all data transfer responsibility, and the number of peripherally attached users becomes limited by the performance of that element. The central system resource is often the processor, and it already has enough to do. Dependence on a central resource is “designed-in” obsolescence, as the performance and number of users is limited by the collective load on the central resource, which is fixed in capacity at time of purchase.

The cluster host LAN does not have this defect. The GUESS card need only support its own data transfer requirements.

As each GUESS card is added, the processor power to support its user is added to the system, and the connectivity to communicate is added as well. Micro Channel architecture and GUESS work together to maintain the user’s asset in system unit and peripherals.

In theory, an operating system could be defined to coordinate access to common system elements, such as disk, printer, communication connectivity, and even bridge to external LAN. Such an operating system is a monumental task if taken as an isolated effort. Fortu-

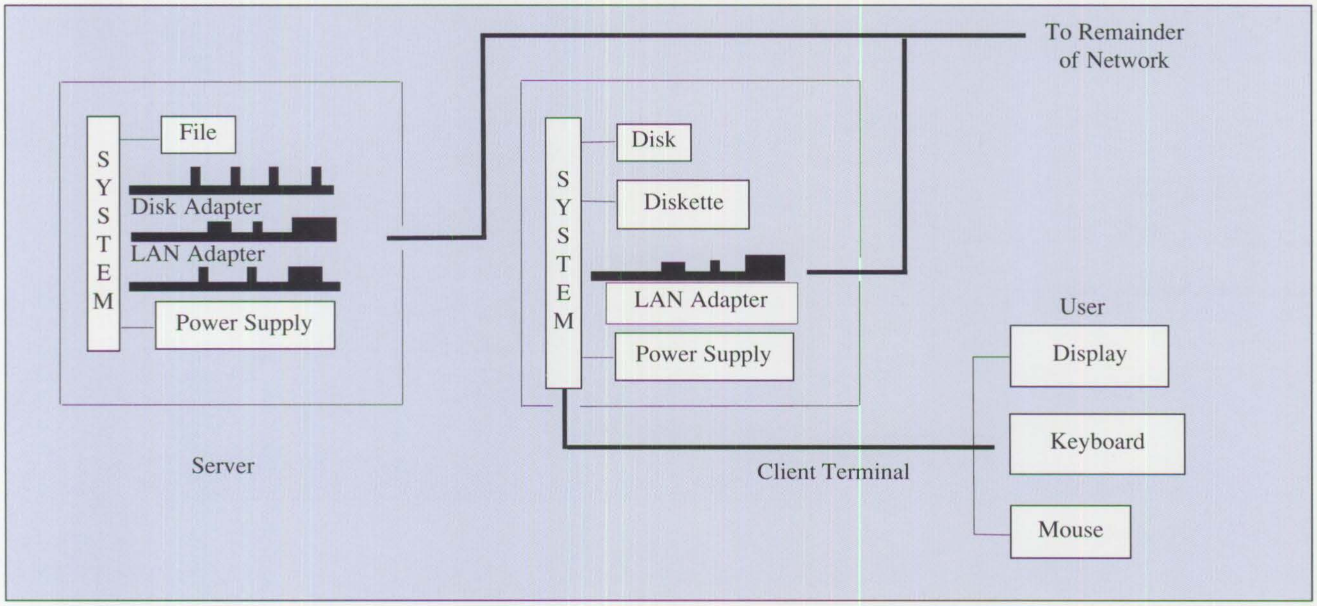


Figure 4. Conventional LAN Topology

nately, there is an existing foundation to build on. Because Micro Channel multimaster arbitration uniquely grants the bus to one owner at a time, and because that owner is "publicly" declared on the arbitration bus, and again because of the public addressing of I/O, the Micro Channel interface can emulate a token ring LAN where the bus grant is the "token" and the bus address is the client or server address.

Further, peer-to-peer communication is a principal advantage of token ring over host connectivity because only one block transfer on the bus is required per message. Peer-to-peer communication and coordination of multiple processors and subsystems is the domain of a control block architecture, such as Micro Channel's Subsystem Control Block (SCB) architecture. Provision for resource allocation, error detection and recovery, and diagnostic isolation of the processors is the domain of Micro Channel Program Option Select (POS), Parity, and Channel Check protocols. ***Net: This concept "needs" PS/2's Micro Channel architecture.***

The cluster host-attached peripherals, disk, diskette, LAN bridge, connectivity, and print adapters are then "served" to the GUESS card's attached users by modifications of standard local-area-network control programs. The cluster host "sees" each GUESS card as a user on a network, and each GUESS card "sees" the cluster host as the server. GUESS cards relate to each other as peer clients in the network. The resulting system is not diskless, because a portion of the system disk can be allocated to each user; however, the user interface is disketteless. The placement of the disk and diskette at a point physically removed from the user has security and environmental advantages and is actually desirable in many applications.

The utilization of the peripherals is no higher than that of a small LAN of eight users or less. The bus traffic may be reduced over a comparable LAN as the cards could use one transfer to communicate peer-to-peer with SCB-designated peripherals. A conventional server system would need at least two, and perhaps four, transfers with system memory and the microprocessor in order to move the same data. The network control program is helped by the SCB architecture by delegating part of its function out to the cards.

The Micro Channel interface has a default burst data transfer rate of 20 million bytes per second. As a local area network serial interface, that is equivalent to 160 megabits per second or 10 times present token ring network speeds. ***Net: Actual performance should exceed that of a comparable small LAN.***

Consider the impact on the user and system owner of such a configuration. In a conventional LAN topology, as in Figure 4, the server and clients each have many duplicated functions. Many of these functions

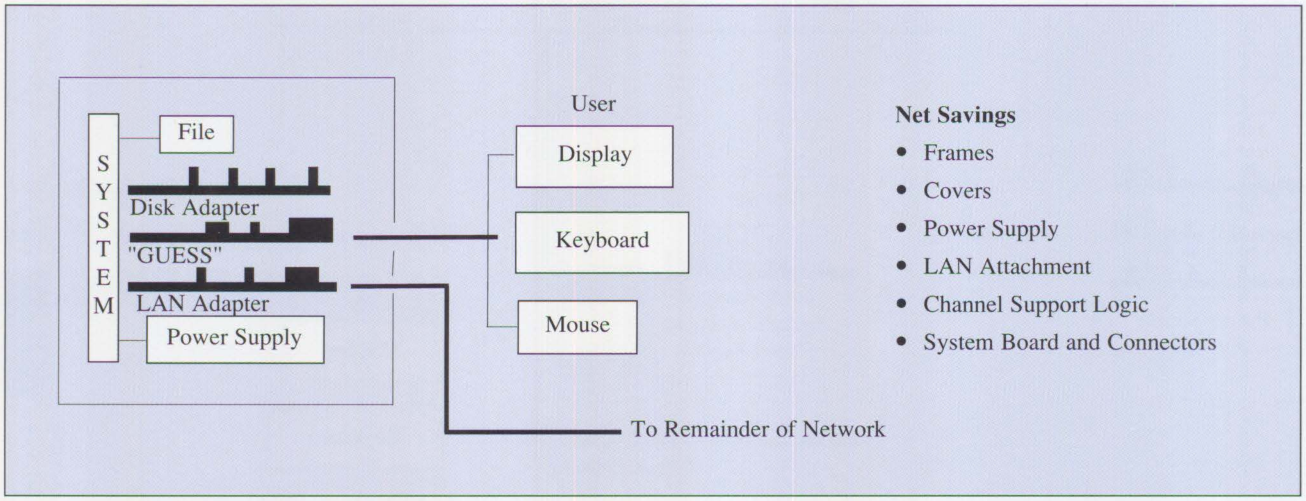


Figure 5. Cluster Host LAN Topology

do not contribute to data movement or, like the diskette or client LAN card, have very low utilization in normal operation.

In a cluster of GUESS cards, each user is supported by a card rather than a system unit. In effect, the card plugs into a server, as in Figure 5.

Because all of the basic I/O functions are provided by the "cluster host" system, the user has the basic function of the desktop system less the cost of frames, covers, local disk and diskette, local area network card, and much of the system board logic required to build a conventional system. These elements can be a majority of the cost in a system. ***Net: The per-user cost of attachment should be less than a comparable conventional LAN design.***

Furthermore, because the GUESS-connected users share the system peripherals, the cost of those peripherals is "amortized" over all the users, and the cost of the system is thereby reduced additionally. Therefore, a centralized SCSI file system can outperform the lower-cost file systems of small desktop systems, yet the cost per user "seat" in the system is less than eight such small disk systems.

An optical "FDDI" LAN, as in Figure 6, may be prohibitively expensive if connected to each client in a

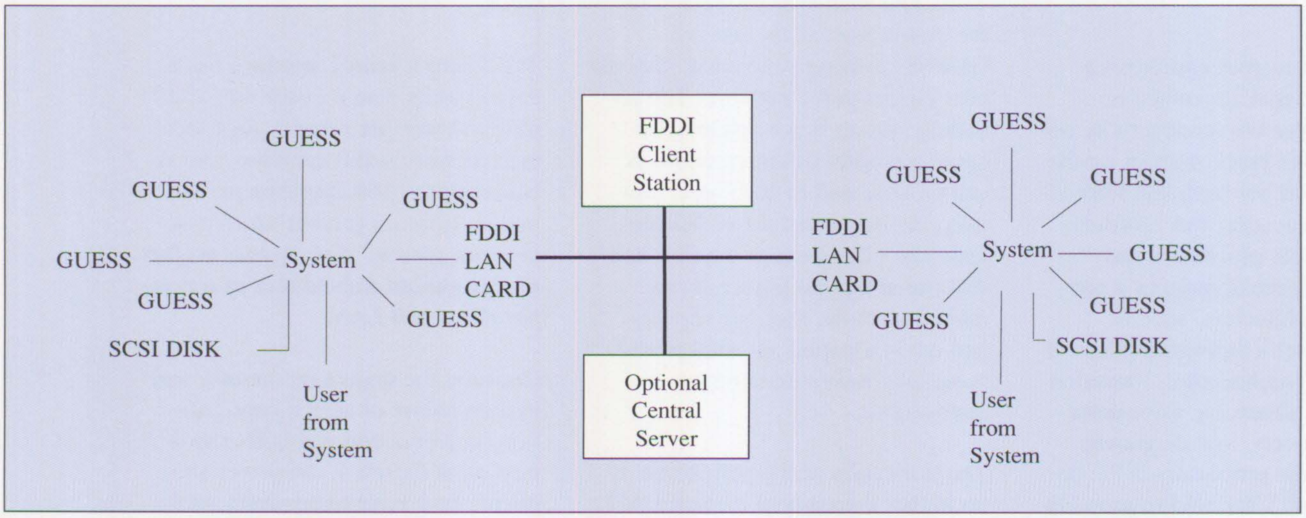


Figure 6. Fourteen-User System with Optical LAN abs Dual SCSI File Systems

conventional LAN topology. It would cost as little as one seventh as much per user if added as part of the cluster host system. A similar scenario exists for the incorporation of a centralized numeric processor, like IBM's i860-based Wizard card in the cluster host.

The concept is flexible. It is believed that the GUESS card can emulate an "X-Window" with a real processor instead of a virtual one. The real processor stores applications in slower, less expensive RAM on the GUESS card, rather than the high-speed RAM of the central AIX system. In the configuration shown at COMDEX, a 7437 small, 370-based Micro Channel product ran VM (SP5). It provided PROFS® mainframe host 3279-like attachment across the Micro Channel via a "virtual" 3274 control unit emulated in software, with virtual TCA and DCA cards as mailbox windows in the system RAM of a cluster host PS/2 Model 80. The interface to users was the familiar "hot key" between host VM applications and DOS on the GUESS card. No external dependency on a mainframe was required for basic operation (however, such connectivity was not designed out, either). ***Net: The per-user cost of entire systems may be further reduced over a conventional LAN or host attachment configurations.***

The client's power supply, LAN attach card, LAN wiring, the multiplexing address units of conventional LANs, and the mechanical disk and diskette systems are major contributors to the failure rate of LAN-connected personal computers of any bus design. The elimination of these elements from the GUESS card improves the reliability of the user interface several fold. The availability of each server to the conventional network is dependent on the server's power supply and file system. The same is true for the cluster host design. An uninterruptible power supply will protect the server in a conventional LAN; with cluster host, it protects the clustered clients as well. Because the clients draw the power of a card rather than that of a system, up to 90 percent of the electric power per client can be conserved. It is therefore economical to provide such protection. ***Net: The reliability of a cluster host-based LAN can, in theory, exceed that of a conventional LAN topology***.

In a conventional LAN, a client can be brought off-line for service without interrupting the availability of the system to other clients. One shortcoming of the cluster host LAN is that the entire cluster must be powered down to service a GUESS card, interrupting a system in continuous operation. Modern component reliability can reduce the probability of such an interruption of availability to a rare occurrence. This loss of availability during service operations is not a concern when service can be scheduled for idle periods. Inoperative stations can be cycled off-line by POS while awaiting service.

The potential exists for the GUESS card to assume the system's duties and control peripheral adapters and memory much as the system would, albeit at degraded performance levels, because the GUESS card is a master and a system on the bus as well. Such operation could conceivably be invoked manually or perhaps with very complex software automatically. ***Net: Availability can be made equivalent to conventional LANs in most scenarios.***

Another consideration with the cluster host LAN is the cost of wiring. Where a facility has been wired for token ring, that wiring can be used to interconnect clusters rather than individual users in the conventional LAN. This provides up to an eight-to-one cost effectiveness for existing wiring with the cluster host LAN. Due to the limitation in cable length from the cluster host, if users are spaced at a distance greater than a few hundred feet from the cluster, it may be advantageous to bridge to token ring for that set of users. Where users are clustered in close proximity, like in many service industries, the cluster host LAN topology is advantageous. One hybrid configuration is achieved when a desktop system is shared between two adjacent workspaces, as in Figure 7.

The system, with diskette and peripheral devices, is at the left hand of one user and the right hand of another. The same hybrid might appear as a two-user transportable system with a GUESS card installed in a P70 system.

Potentially, bridging to conventional LAN has other advantages. Only the transfers that "escape" the cluster appear on the LAN. Local departmental traffic need not occupy or burden the network. Only one token ring address may need to be consumed per cluster, potentially raising the total number of stations on the LAN to 1024, rather than the normal limit of 256. When systems initially load programs to the diskless stations on power-up, the cluster host LAN can transfer a full 16-megabyte system memory load theoretically in a fraction of the time of conventional token ring LAN.

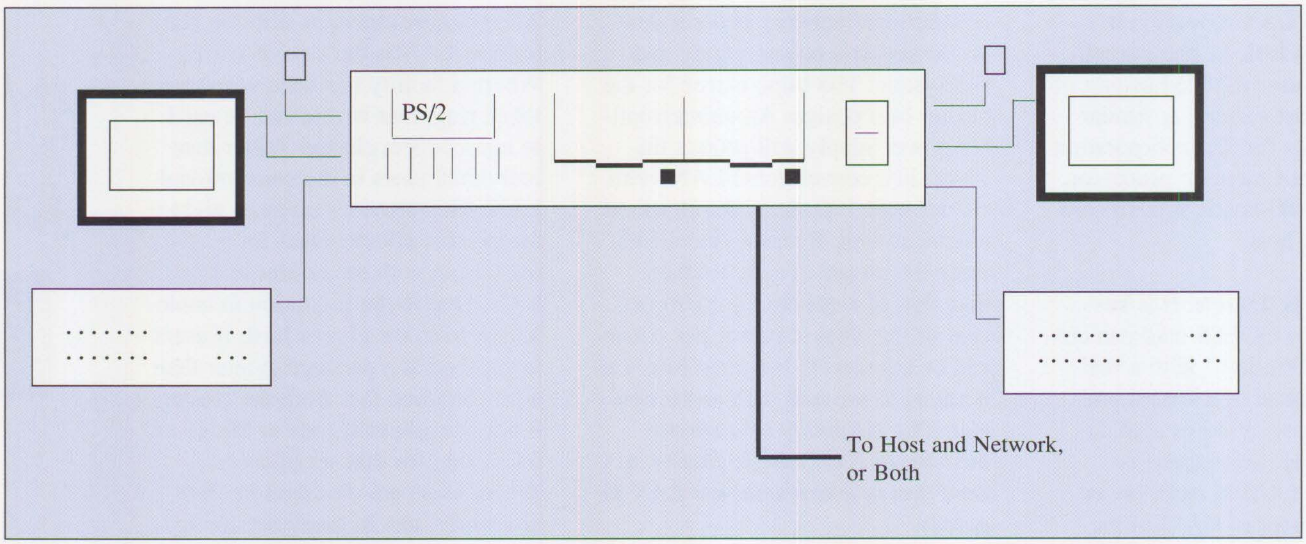


Figure 7. Hybrid Two-User System

The concept is highly dependent upon characteristics of the Micro Channel interface. Fully 85 percent of the system can be purchased today in the form of high-end PS/2 systems, with the advantages of the approach being derived from the dual use of the Micro Channel interface as an ultra-high-speed data transmission path, as well as the peripheral attachment means. Yet, much work lies between concept and reality with this new approach to workgroup systems. New LAN control programs, new hardware packaging, BIOS, diagnostics and Power On Self-Test procedures must be designed. New means of bridging to conventional LANs must be defined. It is even possible that an insurmountable obstacle may be discovered that prevents implementation of the concept altogether. Yet with IBM's open disclosure of the concept to the industry, there will be many who are attracted to the challenge.

Where the conventional LAN is still implemented with stand-alone system designs, the external connectivity adds to both the cost and failure rate of the system. The serial LAN data throughput acts as a limiter to performance of the distributed intelligence.

Alternately, the integration of the workgroup computer designs out the limitations of conventional LANs, and does for the group what the PC did for individuals: it gives them autonomy. The next logical step is the integrated workgroup computer, where "the network is the Micro Channel."

ABOUT THE AUTHOR

Chet Heath is a Senior Member of Technical Staff at IBM's Entry Systems Laboratory in his 21st year with IBM. He holds B.S. and M.S. degrees in electrical engineering. Chet was lead engineer for the definition of Micro Channel architecture and gave the architecture its name. He has published many articles explaining architecture and workgroup computing in technical journals and the trade press. He has received IBM Quality and Author awards. He has earned IBM's eighth-level invention award, Outstanding Technical Achievement, Outstanding Innovation, and Corporate Technical Achievement awards for his work on Micro Channel architecture and other inventions.

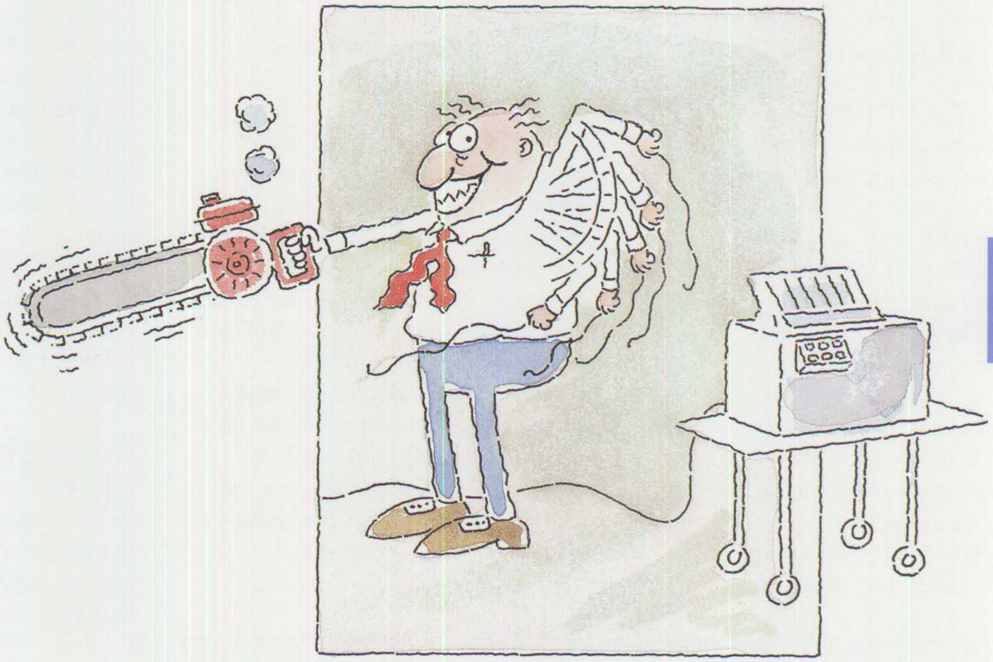
The Micro Channel Architecture Handbook was written to accommodate the needs of both beginning and advanced users. It is a guide to the Micro Channel, from the basic overview to a detailed look at every discrete function found in IBM PS/2 Model 50 through the new RISC System/6000 workstations, and beyond. It includes ideas and insights for system design applications. The introduction was written by Dr. Robert L. Carberry, IBM Vice President.

The Micro Channel Architecture Handbook, written and copyrighted by Chet Heath and Winn L. Rosch, is published by Brady Division of Simon & Schuster, Inc. The ISBN number is 0-13-583493-7. It can be ordered by calling 1-800-624-0023.

Invoking Printer Job Properties

Dave Dill
IBM Corporation
Dallas, Texas

One of the most frequently asked OS/2 questions is "How do I make my printer print a job using landscape orientation?" The truth is, using a printer's specialized features for selected output is quite easy. A single Presentation Manager call added to your application enables you to select any combination of a printer's features for output by invoking the Print Manager's Job Properties. This article describes how to use the PM DevPostDeviceModes call to invoke Job Properties.



When data is sent to the Print Manager, there are two sets of variables, or properties, that determine exactly how the data is printed.

The primary set of variables, called Printer Properties, is established when the printer driver is added to the Print Manager. The actual Printer Properties vary with each printer driver but usually define optional printer hardware, size and type of papers, fonts, supported codepages, print quality, paper orientation, paper tray selection and spooler filter type.

The Printer Properties for a printer driver apply to all printing directed to that driver. A subset of these variables can be changed for each print job allowing for variations in output appearance. These variables are called Job Properties.

An application can support the setting of Job Properties through the use of the DevPostDeviceModes (WDPDM) Device Function call. The use of this call causes a Job Properties dialog box to be displayed, allowing you to select the desired properties for the current print job.

To add Job Property selection to your application's print routine, issue the DevPostDeviceModes (WDPDM) call with the DPDM-POSTJOBPROP option prior to opening the Print Manager.

The DevPostDeviceModes call must be issued twice. First, with a null pointer as parameter two, to obtain the size of the Driver Data structure. This structure can range in size from 50 to 700 bytes depending upon the printer driver. Figure 1 shows how to code the first DevPostDeviceModes call in COBOL. For C programs, see the *Presentation Manager Programming Reference,* Volume 1 (64F0276).

```
CALL "WDPDM" using  hab,
                    NULL,
                    DrvrLength,
                    DrvrName,
                    PrtLength,
                    PrtName,
                    NULL,
                    NULL,
                    DPDM-POSTJOBPROP,
                    StructureSize.
```

Figure 1. Call to Determine Driver Data Structure Size

Upon return, StructureSize contains the length of the required data structure DEVSTRUCT-DRIVERDATA. Following the creation of the DEVSTRUCT-DRIVERDATA structure, of the size returned in StructureSize, reissue the DevPostDeviceModes call (WDPDM) with the DEVSTRUCT-DRIVERDATA pointer as parameter two. (*Note:* COBOL programs must use the WinCreateDataStructure (WUCDS) call to create the necessary data structures.) This causes the Job Properties' dialog for the specified printer driver to appear, allowing the user to select the desired properties. The driver then returns the updated driver data structure to your program. Figure 2 shows how to code the second DevPostDeviceModes call in COBOL.

The content and format of these structures varies with each printer and their formats have not been externalized. However, it is not necessary that your program examine the structure. Simply pass the entire structure, as part of the SplQmOpen (SMOPEN) or DevOpenDC (WDOPEN) call used to open the Print Manager. Figure 3 shows how to code the SplQmOpen call in COBOL. For C programs, see the *Presentation Manager Programming Reference*, Volume 1.

```
CALL "WDPDM" using  hab,
                    DEVSTRUCT-DRIVER,
                    DrvrLength,
                    DrvrName,
                    PrtLength,
                    PrtName,
                    NULL,
                    NULL,
                    DPDM-POSTJOBPROP,
                    StructureSize.
```

Figure 2. Call to Invoke Job Properties Dialog

```
CALL "SMOPEN" using LengthTokenID,
                    TokenID,
                    LengthDataStructure,
                    DevOpenStruct,
                    hSPL.
```

Figure 3. Call to Open Spool File

With the SplQmOpen call, the TokenID is coded as a single character "*" and the LengthTokenID is defined as 1. The LengthDataStructure defines the number of entries specified in the DevOpenStruct. The last parameter, hSPL, is the handle of the open spool entry returned by this call. The DevOpenStruct is defined in the individual language bindings reference manuals or the *Presentation Manager Programming Reference*, Volume 1.

Use of the DevPostDeviceModes call offers users extensive flexibility in how output is printed. Yet it does not require complex application code to support unique printer features.

ABOUT THE AUTHOR

Dave Dill is a senior market support representative in the Personal Systems Technical Support Center, providing technical support to IBMers and customers for OS/2 Standard Edition and Presentation Manager. Dave joined IBM in 1968 and worked in the field as a finance industry systems engineer. He then moved to Finance Industry Development where he helped develop several industry and PC-based applications. For the last five years, he has provided market support for PCs and for the DOS and OS/2 operating systems.

Comparing PC-DOS, OS/2, and AIX PS/2

Craig Chambers
IBM Corporation
Dallas, Texas

This article examines and compares the AIX PS/2, PC-DOS, and OS/2 operating systems. The capabilities, similarities, and differences of these three systems are clarified so you can properly position them in your installation.

Today, many personal computer users are reaching the limits of the PC-DOS operating system and are facing the decision of how to upgrade their systems. Users are trying to decide whether to upgrade to OS/2, UNIX, or AIX. This article compares the PC-DOS, OS/2 Standard Edition, and AIX PS/2 operating systems to help these DOS users make their upgrade decision. Recently released windowing products, such as Microsoft Windows Version 3 and AIXwindows™, may have a significant impact on this decision and are discussed as well.

This is written for DOS users. Readers should be familiar with computers, PC-DOS, how disk drives are organized (tracks and sectors), the differences between ROM and RAM, and similar fundamental concepts. This perspective will be used to explain how OS/2 and AIX compare and contrast with DOS. The various graphical user interfaces of each system are also examined.

AIX is IBM's enhanced version of UNIX. The AIX version discussed in this article is for the PS/2. Comments about AIX will generally apply to UNIX as well.

What is an Operating System?

A computer is a collection of hardware that by itself can do nothing. Before a computer can perform useful work, it must have a program called an operating system installed. This situation is similar to an automobile. Without gasoline, it will not run. The operating system "fuels" the computer and has three basic functions, which are:

- Organizing data into a file system so it can be used by programs
- Loading and executing programs
- Allowing programs to use the devices attached to the system

The faster and more powerful the car, the better the grade of gasoline it needs. The same is true with computers. Small computers can make do with a simple operating system.

As computer hardware gets more powerful, then the operating system must be equally powerful.

Entry-Level Systems

DOS (both PC-DOS and the similar MS-DOS®) is the most widely used computer operating system in the world today. It is installed on more than 40 million personal computers. For simplicity, PC-DOS and MS-DOS will be referred to as simply DOS.

DOS became available in 1981 with the release of the Intel 8088-based IBM Personal Computer (PC). The first DOS version required just 16 KB of RAM and was designed to run with just 64 KB of RAM and a single 160 KB floppy disk drive. Then, a personal computer with 64 KB of RAM was a large system, so the 1 MB 8088/DOS address space was extremely generous. DOS is based conceptually on the older 8-bit CP/M system and provided a similar set of commands, command prompt, and function set.

DOS was originally written for small 8088 microcomputer-based systems. It was designed to run just one program at a time and to be used by one person at a time. Because of this, it does not contain functions for data security, data sharing, performance monitoring, and so forth.

Over time, as the capability of the personal computer hardware increased and hardware prices dropped, the power of the personal computer system increased many times. With this increase in power came many more powerful application programs. In fact, a new type of program, known as the terminate-and-stay-resident (TSR) or pop-up, became available.

These programs get around the DOS limitation of allowing just a single program to be active at one time. By taking control of the hardware timer and keyboard, these programs can perform a certain level of multitasking. In fact, the DOS PRINT command takes advantage of this ability.

By the late 1980s, personal computers based on the newer Intel 80286™, 80386™, and 80486™ microprocessors were equipped with as much as 16 MB of RAM and contained hard disk drives with capacities of several hundred megabytes. The number of TSRs had multiplied to the extent that they were very difficult to manage and used a substantial portion of the DOS 1 MB address space.

Although the system had as much as 16 MB of RAM, DOS could only use 1 MB of it, because it was still designed for the 8086 processor. With the release of Version 4.00, DOS included support for "expanded memory." While this new type of memory simplifies writing for large programs, it does not eliminate the need for application programmers to code elaborate routines to use the extra system memory. Without this support, many applications no longer fit into the 1 MB area after the device drivers and TSRs are loaded.

Another enhancement introduced with DOS 4.00 was support for hard files with capacities greater than 32 MB.

One major advantage of DOS is the large number of programs written for it. As a result, DOS connects to most other systems and the applications generally have very polished user interfaces.

Advanced Systems Become Available

The limitations of DOS are removed by OS/2, the operating system announced in 1987. OS/2 uses the same commands, has the same basic system structure, and is a logical follow-on to DOS. OS/2 is so compatible with DOS that most DOS programs can be run unchanged in one of the OS/2 sessions, called the DOS compatibility box.

DOS is designed to operate on the 8086/8088 processor family that runs in a system processor state known as *real mode*. In real mode, applications have total control of the system and assume that no other programs are executing while they are running. Therefore, these applications can completely take over the system hardware.

Real mode is the only mode supported by the 8086/8088 and compatible system processors. It was not named until Intel released the 80286 processor, which contains a more powerful mode of operation called *protected mode*.

In protected mode, special hardware in the microprocessor separates applications, making them unaware that other programs are active. If a program tries to access another program's memory or hardware devices, the operating system can catch it and terminate the program, thereby preventing a system crash. Protected mode allows access to the full 16 MB of RAM that can be installed in systems based on 80286 (and later) microprocessors.

OS/2 and AIX PS/2 operate in protected mode. OS/2 and AIX programs use the full amount of memory installed in the system.

This eliminates the complexities of dealing with "regular" memory, extended memory, and expanded memory as is necessary in DOS. OS/2 runs only on personal computers built with the Intel 80286, 80386, and 80486 processors. AIX PS/2 requires an 80386 or 80486 processor.

Both OS/2 and AIX PS/2 can run multiple programs concurrently, without their applications using any elaborate programming to share system resources. OS/2 programs also can contain multiple tasks that allow a single application program to do several things concurrently.

Like DOS, OS/2 was designed to be used by a single person on a single personal computer; therefore, OS/2 does not contain any data security or performance monitoring code. Because multiple programs can be active concurrently, data integrity functions are required. OS/2 does ensure that data files are used by only one program at a time. As discussed later, AIX PS/2 is a multiuser system and has much greater built-in security.

OS/2, like DOS, is designed to be run only on IBM personal computers or other compatible systems. AIX PS/2 only runs on the PS/2. There are AIX versions for the RISC System/6000, RT®, and System/370™ that allow an AIX PS/2 application to be moved easily to any of these other systems.

OS/2 comes in two versions: Standard Edition and Extended Edition. Standard Edition is a direct replacement for stand-alone DOS systems, while Extended Edition adds a communications manager and database manager package to the base standard edition system. With these functions integrated into the system, it is easier to develop integrated applications for OS/2 than it is for DOS.

Because OS/2 operates similarly to DOS, little training is needed for users migrating from DOS.

Portable Systems

The IBM AIX System is a derivative of the UNIX system originally developed by Bell Labs in the late 1960s and early 1970s. From the beginning, UNIX was designed to be a multiuser, multitasking system. UNIX is organized similarly to DOS and OS/2. However, because it is a multiuser and multitasking system, it provides security as an inherent system function.

UNIX became popular in part because it was so easy to modify. AT&T® distributed the UNIX source code, allowing users to tailor the system to suit their needs.

Because UNIX was very easy to modify, at least 13 different semicompatible derivative systems appeared on the market, leading to confusion and incompatibilities. In 1988, the Open Software Foundation (OSF™) was formed to solve this problem. OSF uses a significant portion of AIX Version 3 as the basis for a compatible version of UNIX.

AIX maintains the design it inherited from the UNIX system. Its programs are portable across a wide range of IBM systems, including the PS/2, RT, RISC System/6000, and System/370™.

While DOS is used more than any other operating system, UNIX and its derivative systems run on a greater variety of computers than other operating systems.

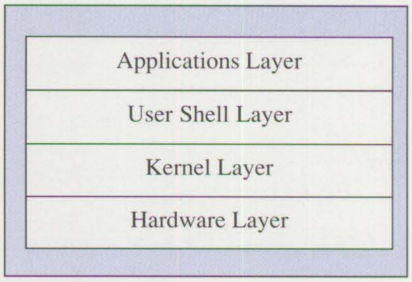


Figure 1. DOS Layer Structure

Because of limited function of DOS, it is relatively easy to learn. While AIX PS/2, like the other UNIX derivative systems, is much more powerful than DOS, it is also probably the most difficult operating system for microcomputer users to learn. This is due to its large number of system commands. OS/2 falls somewhere in between.

Architecture

DOS Architecture: DOS can be thought of as having several layers (Figure 1). The first, or lowest layer, is the physical hardware itself. Built on, and maybe in, the hardware is the BIOS, which is a Read-Only Memory (ROM)-based group of software routines used to control the hardware. When a PC is considered IBM-compatible, that PC has a similar BIOS interface. The BIOS contains low-level control functions, rudimentary device drivers, and other code required to initialize (boot) the system.

The next higher layer is the DOS kernel, which is the heart of the system. This portion provides the communication between application programs and the BIOS/hardware layers. The DOS kernel has the application program interfaces (APIs) that provide hardware independence to the application programs. The kernel is loaded when the system is

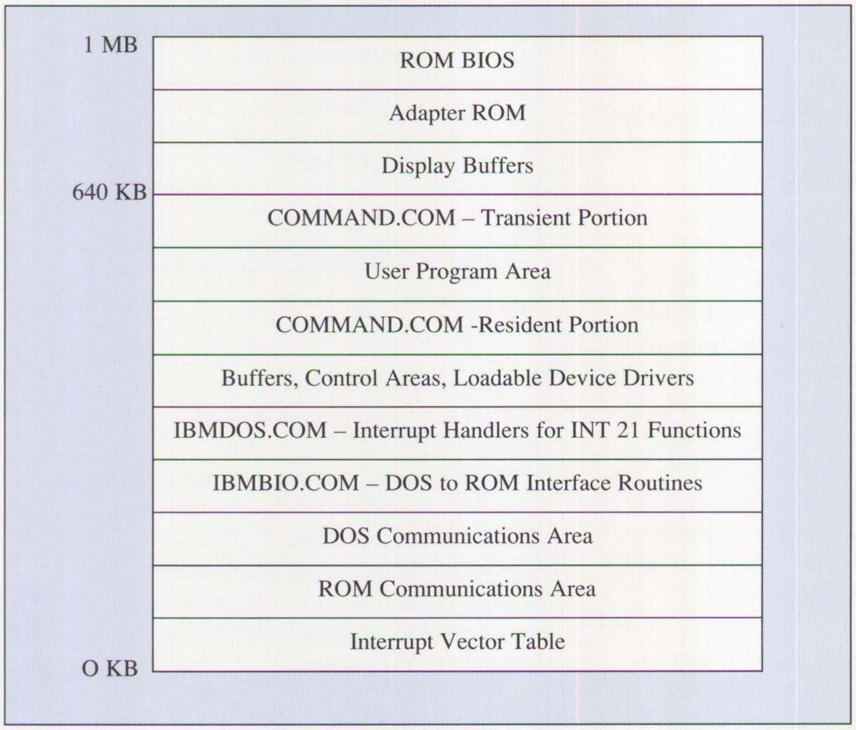

1 MB
640 KB
O KB
ROM BIOS
Adapter ROM
Display Buffers
COMMAND.COM – Transient Portion
User Program Area
COMMAND.COM -Resident Portion
Buffers, Control Areas, Loadable Device Drivers
IBMDOS.COM – Interrupt Handlers for INT 21 Functions
IBMBIO.COM – DOS to ROM Interface Routines
DOS Communications Area
ROM Communications Area
Interrupt Vector Table

Figure 2. Map of DOS RAM

started and always remains in memory.

The highest layer is the command processor or shell that contains the COMMAND.COM program. This program gives users the familiar **C>** prompt.

Many system commands are executed from within COMMAND.COM; however, some commands, known as external commands, are really separate programs. COMMAND.COM loads and executes these programs as needed. All application programs can be thought of as external commands because, like external commands, they are loaded and executed by COMMAND.COM.

COMMAND.COM has two sections: the resident portion, which loads next to the kernel; and the transient portion, which is located at the top of the available system memory (but still below the 640 KB line). The transient portion is reloaded as needed after an application program or external command ends. This makes the maximum amount of memory available for application programs.

The application programs are the highest layer in the system. These programs actually "do the work" for the system's user.

System memory organization, which is shown in Figure 2, is explained in *IBM Disk Operating System Version 4.00 Technical Reference* (15F1420) pages 6-2 and 6-3.

OS/2 Architecture: The OS/2 system looks very much like DOS. This is because it has the same basic layers.

In the memory map shown in Figure 3, the systems are again very similar. The main difference is the use of the memory beyond the 1 MB limit of DOS.

The user interface looked the same in the first release of OS/2. The command structure is the same; the main difference is the name of the programs. For example, the shell program is called COMMAND.COM in both DOS and the DOS compatibility box of OS/2. The shell is named CMD.EXE in OS/2 protected-mode sessions.

Beginning with version 1.1, OS/2 added a Graphical User Interface (GUI) referred to as *Presentation Manager*. This presents a more "user friendly" interface, because it is mouse-driven and provides helps to users when needed.

Another significant way that OS/2 differs from DOS is that it's a virtual storage system. With OS/2, one can start more programs than will physically fit into system memory. When this happens, OS/2 memory is considered to be *overcommitted.* Programs are swapped out of memory until sufficient memory has been freed to accommodate the request that started the swapping process.

Beginning with OS/2 Version 2, demand paging handles memory overcommitment. This requires the support of the Intel 80386 microprocessor; therefore, this system release cannot run on an IBM AT or Intel 80286-based PS/2 systems.

Another characteristic that makes OS/2 unique among these systems is that it's a preemptive, time-critical, multitasking system.

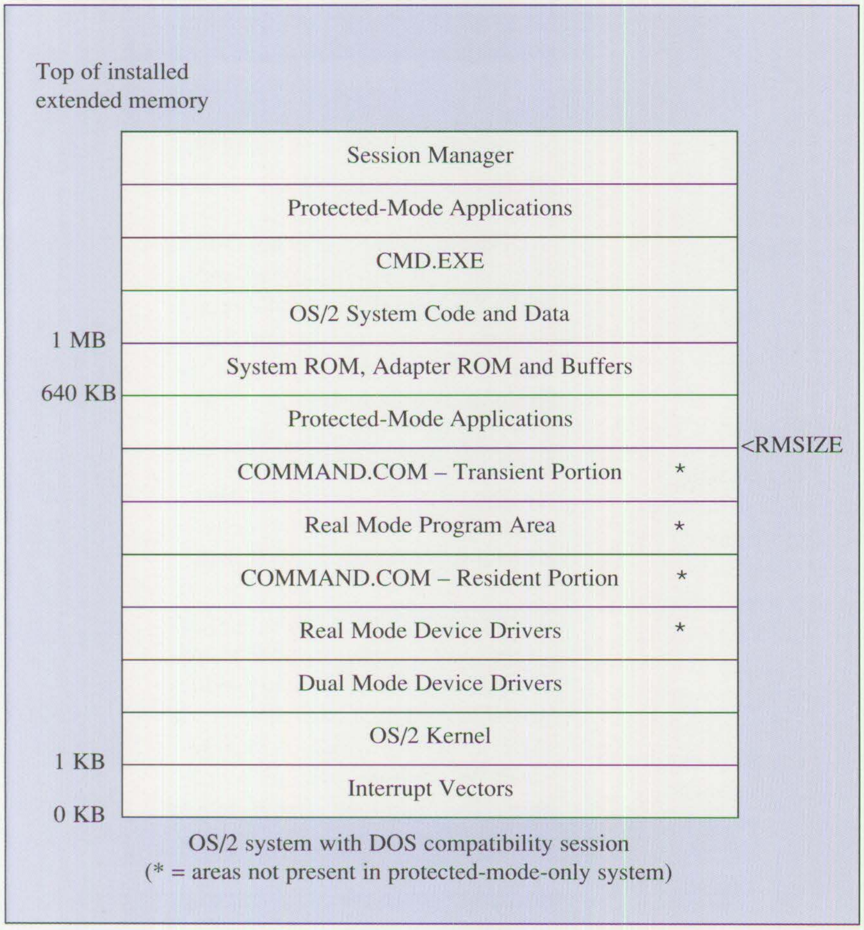

Top of installed extended memory
Session Manager
Protected-Mode Applications
CMD.EXE
OS/2 System Code and Data
1 MB
System ROM, Adapter ROM and Buffers
640 KB
Protected-Mode Applications
<RMSIZE
COMMAND.COM – Transient Portion *
Real Mode Program Area *
COMMAND.COM – Resident Portion *
Real Mode Device Drivers *
Dual Mode Device Drivers
OS/2 Kernel
1 KB
Interrupt Vectors
0 KB
OS/2 system with DOS compatibility session
(* = areas not present in protected-mode-only system)

Figure 3. OS/2 Memory Map

AIX Architecture: UNIX was designed to be used on various types of computers. While UNIX contains some Assembler routines, it is primarily written in a high-level language known as *C* with just a small amount of Assembler code. Therefore, UNIX can be run on any system that has a C compiler. AIX is the version designed to run on IBM systems.

AT&T gave universities the right to use the UNIX system, which fostered many enhancements to the basic system. The most common versions of the system available today are UNIX System V, released by Bell Labs in 1983, and the BSD 4.2 or 4.3 systems that were developed at the University of California at Berkeley.

The user interface is similar to DOS, except it has a $ as the prompt. AIXwindows, a new graphical interface based on the MIT X Window System™, is now available. AIXwindows is similar in concept and appearance to OS/2 Presentation Manager.

AIX has the same basic layered structure as DOS and OS/2. Like OS/2 Version 2, AIX is a virtual storage system. AIX can swap complete processes out of memory and simultaneously do demand paging.

Unlike DOS and OS/2, AIX is designed to run on many different types of computers. The kernel layer in Figure 4 is tailored by IBM for a specific system processor, such as the PS/2, RT, RISC System/6000, or System/370. AIX creates the required device-independent appearance for the higher level user shell and application layers.

AIX PS/2 offers two different user shells, the Bourne shell and the C shell. The Bourne shell, which is the default, was developed by AT&T. The C shell, with syntax closely matching the C language, was developed later. (AIX PS/2 does not include the other widely-used UNIX shell, the Korn shell, which is the default on RISC System/6000 systems.) AIX PS/2 is a merger between version 2 of the AT&T system and the BSD 4.3 function, all in a unique UNIX system.

DOS Merge

Another interesting alternate shell for DOS users is called *DOS Merge*. This shell actually copies the files from the DOS 3.30 distribution disk into the AIX PS/2 file system, making most DOS commands available. After AIX users enter the command, "DOS", they can execute familiar DOS commands like DIR or use their familiar text editing program. They can also run other stan-

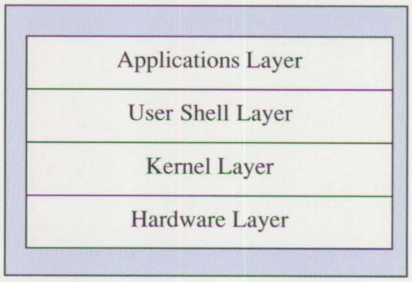


Figure 4. AIX Layer Structure

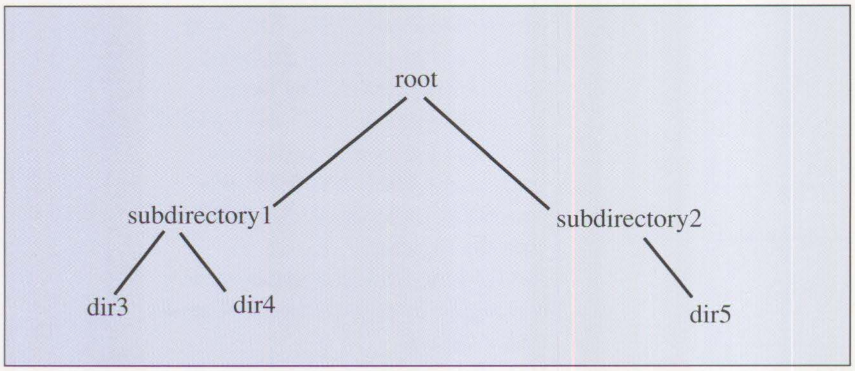


Figure 5. Directory Tree

dard DOS applications from the command line, just as they would in a DOS system.

Because AIX PS/2 filenames can be longer than those allowed by DOS, the AIX PS/2 system maps the longer AIX filenames to meet DOS requirements, allowing DOS programs to access these files.

Another way to use DOS Merge is to enter DOS command names on the AIX command line. This way, DOS and AIX commands can be intermixed without actually getting into the DOS environment. This technique could be used to start a familiar DOS shell program while running AIX programs.

File Systems

Directory Structure: Externally, the DOS and OS/2 file systems look similar to AIX. This is because DOS and OS/2 designs were conceptually based on the UNIX system, as was AIX. In all of these systems, the file structure (Figure 5) looks similar to an upside-down tree, with a single directory called the *root directory* and multiple levels of subdirectories branching out below the root.

When listing files in a directory, two interesting entries will be encountered. They are the dot (**.**) and dot-dot (**..**) entries. One common feature of these systems is using dot (**.**) and dot-dot (**..**) directory names. The dot (**.**) directory is the current one, and the dot-dot (**..**) directory is the parent directory. Therefore, to move up one level in the directory structure, enter **cd ..** on the command line. The only time this is not true is when the root directory is the current one and there is no higher directory level.

File Naming Conventions: In DOS and OS/2, except for files on High Performance File System (HPFS) devices, filenames are limited to 11 characters, which is an eight-character name followed by a "**.**" (period), followed by a three-character extension. This was done in the original release of DOS for compatibility with the CP/M system.

In the AIX PS/2 and OS/2 HPFS file systems, filenames can be up to 255 characters and may or may not contain one or more periods.

In DOS and OS/2, the system determines the use of a file by its file extension. Executable programs have a file extension of either .COM or .EXE. Files with an extension of .BAT in DOS and in the DOS compatibility sessions of OS/2, and .CMD in OS/2 protected-mode sessions, are text files containing scripts of commands executed by the command processor.

AIX does not use a filename to determine its function. Instead, the file's attribute flags in the directory contain this file information. If a file can be executed either as a program or as a shell script, the execute flag in its directory entry is turned on. This gives the user greater flexibility in naming files, but makes it impossible to determine a file's function by its name.

There is one other interesting quirk in the system naming conventions. In DOS and OS/2, directory names are separated by a back slash, \ . In AIX, a forward slash, / is used instead. While both mean the same thing, the / can look strange to experienced DOS users and vice versa.

Wildcards: The AIX system has far greater flexibility when using wildcards in commands with filenames. In both DOS and OS/2, the use of wildcards is limited to the ***** and **?** character to represent any character or characters.

In AIX, patterns and ranges of values also can be used. The ***** and **?** are used as in the DOS and OS/2 systems, but AIX introduces the **[ ]** and **[-]** which are inclusive list operators.

The character in a filename must match the characters in the **[ ]** exactly or be in the range indicated by the **[-]** symbols. For example, **ab[cde]*** would include all files beginning with the letters **ab** and followed by either **c**, **d**, or **e**. This also could be written **ab[c-e]***.

DOS	OS/2	AIX	Description
APPEND	DPATH		Set search path to be used by the system for program data if the data file is not found in the current directory
CHDIR CH	CHDIR CH		Change the current directory, or display current directory if no parameters are added (abbreviated CD)
		cd pwd	Change current directory Display present working directory
CHKDSK	CHKDSK	fsck	Check structure of disk or file system
CLS	CLS	clear	Clear screen
COMP	COMP	cmp	Compare two files
		comm	Compares files, line by line
COPY	COPY	cp	Copy files
DATE	DATE	date	Set system date
DIR	DIR	ls -s	List the files in a directory
		li	List the files in a directory
ERASE DEL	ERASE DEL	rm	Erase (remove) a file
		del	Delete a file with confirmation
FIND	FIND	find grep	Find text in a file
FORMAT	FORMAT		Format a new diskette
		mkfs	Make file system, including superblock, inodes, and datablocks
MKDIR MD	MKDIR MD	mkdir	Make a new subdirectory
PATH	PATH	PATH	Set search path used by system to locate commands
PRINT	PRINT	print	Print a file or get status about print job
PROMPT	PROMPT		Change the command prompt
RENAME	RENAME	mv	Rename (move) a file. (May be abbreviated REN in DOS and OS/2)
REPLACE	REPLACE		Copy files only if an existing file already exists on target device
RD	RD	rd	Remove a directory from a disk. In AIX, **rm -r** will erase all files in the directory and its subdirectories, then will remove the directories.
SET	SET	set	Display an environment parameter
SET	SET	env	Set an environment parameter for execution of one command
SET	SET	set	Clear an environment variable
SORT	SORT	sort	Sort a file
TIME	TIME	date	Set system time
TYPE	TYPE	cat	Display a file on the screen
		bfs	Display a file on the screen
		pg	Display a file, one page at a time
XCOPY	XCOPY		Copy files into respective subdirectories on target disk

Figure 6. Common DOS, OS/2, and AIX PS/2 Commands

OS/2	AIX	Description
DETACH	&	Run a process in the background. (In AIX, the **"&"** is appended to the end of the regular command.)
HELP	man	Display the online manual
MOVE	mv	Move a file to a different subdirectory
PSTAT	ps	Display process status
START	open sh	Start a program as a separate process

Figure 7. OS/2 Commands

The **!** is used to indicate **not**, so **[!a]*** would indicate all files that do *not* start with the letter **a**. This gives the AIX system very powerful file manipulation capability.

System Commands

According to the IBM command reference manuals, DOS Version 4.00 has 66 commands, OS/2 Standard Edition Version 1.2 has 93, OS/2 Extended Edition has 137, and AIX PS/2 Version 2.0 has approximately 300 commands.

Figure 6 compares commonly used DOS, OS/2, and AIX commands. The table starts with the DOS command, followed by the equivalent OS/2 command (because it is a logical extension of DOS), and ends with the equivalent AIX command. Following this is a brief description of each command's function.

OS/2 has more commands than DOS. New OS/2 commands and their AIX equivalents are shown in Figure 7. Figure 8 lists commands unique to AIX (and UNIX).

With DOS, only one command may be entered at a time, whereas with OS/2 and AIX PS/2, multiple commands may be entered. Both OS/2 and AIX PS/2 provide for the conditional execution of a command based on the results of the command preceding it. For example, the link command runs only if the compile was successful.

OS/2 and AIX PS/2 also provide a way to do the opposite and only run a command if the preceding command fails; for example, starting an editor to look at error messages that have been logged if a compile fails.

AIX PS/2 has *command substitution*, which is a unique capability not found in DOS or OS/2. If a command is enclosed in back quotes, for example ‘date‘, the output of that command is substituted in place of the ‘date‘ string in the command. This allows variable data to be used in a system command.

Note: AIX commands are case-sensitive; DOS and OS/2 commands are not.

The commands shown in Figure 8 are unique to AIX. While this list is far from complete, they are examples of the types of commands found in AIX. In some cases, similar commands are available for DOS and OS/2 as separate utility application packages. Each AIX shell also has its own set of commands, which are listed with the shell descriptions in the AIX PS/2 command reference manual.

Environment Variables

The environment is an area in memory that contains strings of text associated with a text identifier. For example, if the command SET NAME=SYSTEM1 is entered at the command prompt, the string NAME=SYSTEM1 would be copied into the environment area. A program could later read this string, and some action could be taken based on the value associated with the variable NAME. For example, the program could use the name SYSTEM1 in a report heading. This value could be used to change the execution flow of a shell script or batch file.

All of these systems use environment variables. In DOS, the only variable established automatically when the system starts is COMSPEC. This variable tells the system where COMMAND.COM is located if it needs to be reloaded after a program has overlaid the transient portion in memory.

OS/2 sets different variables that are dependent on the system version and the options selected at installation time. Typical variables are PATH and DPATH, which tell the system where to look for program and data files. The PROMPT environment variable is used to make it easier to differentiate the protected-mode sessions from the DOS sessions. Other variables are HELP, BOOKSHELF, and KEYS, all of which tell the system that certain functions are needed or where help files are located.

DOS and OS/2 users are just beginning to take advantage of the environment. For example, most compilers now use the environment to locate needed libraries, temporary files, and so forth.

AIX systems make greater use of the environment to control the flow of execution in shell scripts, protect data, and so forth.

Several variables are used with the Bourne shell. HOME defines the user's home directory. This is the current directory when the user logs on to AIX PS/2. PATH is the same as for DOS and OS/2. PS1 is the prompt string displayed on the command line. If a command continues beyond one line, PS2 is the prompt string displayed on the second and succeeding command lines.

The AIX PS/2 system is a multiuser system, which means that both the system and programs must identify the user and the terminal type being used. The following variables are used for this purpose: LOGNAME, LOGTTY, USER, and TERM.

The LIBPATH, MAIL, MAILCHECK, SHELL, and TZ (TIMEZONE) variables are also set at logon time. Other variables may be defined at logon time if they are set up in the user's profile by the system administrator.

The other shells each have their own set of standard variables. For example, one interesting variable included in the C shell is *noclobber*. If this variable is set, files will not be written over, or clobbered, by other files because of redirected output.

There is another significant difference in the way the environment is handled between AIX PS/2 and the other systems. In both DOS and OS/2, all set environment variables are passed to child processes or programs when they are started. However, this is not true in AIX. AIX does not pass all variables to a child process. If a child process needs another variable, a specific export command must be issued for that variable. Only those variables established at logon time are passed.

COMMAND	Description
alias	Allows user to rename commands – C shell only
ar	Archive utility (similar to DOS ARC or LOADRAM)
at	Run a command once at a specified time
awk	Almost a complete database package, "awk" searches files and formats output; "awk" stands for the last names of its authors: Aho, Weinberger, and Kernighan. See the reading list for more information.
cron	Execute a command at scheduled intervals
df	Displays free space on a disk
dircmp	Compare two directories and the contents of all files found in both directories
find	Locate files in the file system
kill	End active processes
ln	Create file links (aliases)
mount	Connect physical disk drive to logical file system
mvdir	Rename (move) a directory
nroff/troff	Text formatting commands that include justification, hyphenation, page numbering, font selection, and so forth
pr	Paginate file before printing with "lp" command
shutdown	This command *must* be used to stop the system before it is powered off. This ensures that all control information for the file system kept in memory is saved on the disk drives.
spell	Check spelling
tail	Display last lines of a file
touch	Set date and time of a file
tr	Translate characters in a file
wc	Count lines, words, and characters in a file

Figure 8. Commands Unique to AIX

Shell Programs, Batch Files, REXX

To compile a program, it is often necessary to execute and re-execute the same command or group of commands. Sometimes it is also necessary to execute a long sequence of commands in a specific order. The purpose of batch files, or shell scripts as they are called in AIX PS/2, is to simplify these tasks.

A batch file is a text file that contains a sequence of commands. When the batch filename is entered as a command, the system reads the file and executes the commands sequentially.

In both DOS and the DOS compatibility session of OS/2, a batch file is a text file with a .BAT extension. In an OS/2 protected-mode session, the batch file has an extension of .CMD. (".CMD" stands for "command," which is another name for this type of file.) The names differ to help users distinguish batch files with DOS commands from command files containing OS/2 protected-mode commands.

The most commonly executed batch files are probably the AUTOEXEC.BAT and STARTUP.CMD files that are executed each time DOS and OS/2, respectively, are booted. The .profile shell script is executed when users log on to AIX.

Frequently, it is not sufficient to simply execute a predefined sequence of commands. It is often necessary to take different actions depending on the results of one of the commands. For example, if the command sequence compiles a file, adds it to a library, and then links the program with other library members, the library and link steps should be skipped if the program had compilation errors.

To do this, program return codes must be tested and branches made in the batch file depending on the test results.

Batch files are commonly used to install programs. This process requires the ability to determine if specific files already exist on the target disk.

The batch file capability that comes with DOS and OS/2 contains these fundamental capabilities. Batch files can also test and set environment variables, test command parameters against text strings, and perform simple looping.

OS/2 Extended Edition Version 1.2 and both Standard and Extended Editions of Version 1.3 include REXX, an enhanced command language. Because REXX is designed to be used by end users, its procedures can be clearly structured and easy to understand.

REXX procedures are often more complex than simple .BAT or .CMD procedures even though they have a .CMD file type. REXX procedures cannot be executed in the DOS compatibility session.

One important capability of REXX is its ability to receive input interactively from the user. For example, to insert a user's name in the command prompt, the PULL command will accept the text string from the user. This is not possible under DOS or in regular OS/2 command procedures.

The AIX PS/2 system has similar power. Like REXX, AIX shell scripts can accept user input, which can then be used to control the script's execution path.

Graphical User Interfaces

Graphical user interfaces, or GUIs, may affect your choice of system. The concept of a graphical user interface was originally developed at the Xerox® Corporation in the 1970s. It was here that the ideas for icons, mouse, and graphical control of computer systems were first developed.

Why Use a GUI? It has been shown that graphical interfaces make new applications easier to learn. With the GUI, all applications look similar. But for those application software users who have considerable experience, the GUI may be more of a hindrance than a help. Competent users will often have developed keyboard macros or other shortcuts for their main applications that do not translate to the GUI interface.

The real advantage of the GUI may not be the graphical interface, but its ability to share data between programs. GUI's cut-and-paste capability enables data to be shared among multiple applications.

For example, when running Microsoft's Excel and Word™ under Windows or Presentation Manager, spreadsheet charts and tables from Excel can automatically be included in a text document written with Word. This way, a significant amount of repetitive typing could be automated. But a GUI would streamline this process.

One major advantage of installing OS/2 Version 2 or Windows Version 3 is their ability to run multiple DOS applications simultaneously in windows with cut-and-paste capability.

In effect, with OS/2 Version 2 and Windows Version 3, users can run their existing DOS programs while taking advantage of the advanced cut-and-paste capabilities of the graphical window environment. This is accomplished without additional cost and with little effort, which could be a significant productivity gain. OS/2 Versions 1.2 and 1.3 can also do this between text-based, protected-mode programs if they are run in a Presentation Manger text window.

The basic difference between the Windows 3 and OS/2 Presentation Manager is the smoothness of the user interface. The Windows program runs on DOS, which does not have the functional capabilities found in OS/2. This gives the OS/2 implementation of similar function a smoother, more responsive feel.

DOS: In 1984, Microsoft released the first version of its Windows program for the DOS operating system. The 8088, with its low system performance, was not fast enough to support the overhead required of the graphical interface, and the 640 KB memory limit did not leave enough room for significant application programs after Windows was loaded.

The 80386 is the first Intel microprocessor in the 808X family that was designed after the PC became available. One of its features was a new mode of operation, called Virtual 8086 or Virtual Machine mode. This mode, which is available on both the 80386 and 80486 microprocessors, allows the system to emulate, in hardware, multiple complete 8086 systems. This is the most common form of DOS-compatible multitasking available now.

Each application sees its own 8086 machine and is not aware that other programs are actually running in the same system. The multitasking program uses the hardware features to trap accesses to shared memory or devices and allocates or schedules them so programs function without interference.

As systems became more powerful, Windows was upgraded, until 1990, when Version 3 was released. Windows Version 3 supports the virtual machine or virtual 8086 capability of the 80386 and 80486 microprocessors as well as supporting protected-mode operation with the 80286, 80386, and 80486 microprocessors. On these systems, performance is acceptable – even with the windowing overhead in the system.

Unlike Windows and Presentation Manager, the X Window System is client/server based.

Between 1985 and 1990, Microsoft also refined the appearance and function of the Windows program. When version 3 was released, it operated smoothly and was enhanced by the availability of numerous third-party programs.

OS/2: OS/2 was jointly developed by Microsoft and IBM, and was announced in 1987. At that time, Presentation Manager was announced as part of OS/2 Version 1.1, which became available in 1989. Microsoft gained considerable experience with its Windows program, and IBM had graphics experience with GDDM in the mainframe environment. OS/2 Presentation Manager now presented a powerful interface with capabilities beyond those in Windows.

Since the release of the Presentation Manager in OS/2 Version 1.1, Microsoft has redesigned Windows. Windows Version 3 looks almost identical to the Presentation Manager in OS/2 Version 1.2.

AIX: There are several graphical interfaces in use in UNIX systems. The first GUI for UNIX systems is the X Window System, which is a network-based graphics windowing system that became available in 1985. X Window is the preferred windowing system for UNIX systems and is used by many large corporations, including IBM.

Unlike Windows and Presentation Manager, the X Window System is client/server-based. This means that applications, or portions thereof, can be run on different machines across a network, not just on the user's local computer. In effect, a person with a network-connected system could run a remote UNIX database program in a window even though the local computer is running DOS.

The OSF developed a standard GUI, called Motif™, that is based on X Windows. It is the most promising UNIX GUI, although it does not have all the functions that should be a part of a GUI. For example, it has no file manager or graphics functions. However, IBM's DESKTOP, which is based on Motif, can be used to provide the file manager functions. Graphics programs can be written using the lower-level X Windows calls from the Motif application. AIXwindows also includes all of these features.

Functional Comparison

	DOS	OS/2	AIX PS/2
Open System	No	No	Yes
Portable to OEM	No	No	No
Portable Applications	Yes	Yes	Yes
Multiuser	No	No	Yes
Multitasking	No	Yes	Yes
Architecture	16 bit	16 bit (1)	32 bit
SAA Compatible	No	Yes	No (2)
File System	Yes (3)	Yes	Yes (4)
Minimum CPU Required	8088	80286 (5)	80386SX
RAM Required	256 K	2 MB	6 MB
Case-Sensitive Commands	No	No	Yes

1 – OS/2 Version 2.0 is 32-bit
2 – AIX PS/2 will be SAA-compatible when POSIX standards will not be violated
3 – DOS 4.00 only
4 – When running DESKTOP
5 – OS/2 Version 2.0 requires 80386SX

Figure 9. Functional Comparison of DOS, OS/2, and AIX PS/2

An interesting alternative is the NextStep™ environment, which IBM offers on AIX. NextStep, which is not a network-based system, has powerful object-oriented tools, much like those found in the OS/2 Presentation Manager.

Summary

This article has presented the capabilities, similarities, differences, and limitations of the DOS, OS/2, and AIX PS/2 operating systems to help you determine which system is appropriate for your requirements. These are summarized in Figure 9.

In summary, DOS is the operating system of choice for users with small systems, or for users who want the easiest system to learn. It has few commands, which are not case-sensitive and generally are simple words. There is a large variety of relatively inexpensive software available for DOS.

OS/2 is an extension of DOS. It requires a more powerful system, but uses many of the same commands and has the same disk structure, making for a relatively easy migration from DOS. It is the logical system for DOS users whose needs have grown more complex.

AIX PS/2 is the most powerful and complex of the three systems. It is designed for multiple users who share the same computer or who need compatibility or connectivity with UNIX systems. AIX PS/2's large number of commands, coupled with their case sensitivity and cryptic names, make it more difficult for new users than either DOS or OS/2.

More to Come

Look for the second part of this article in the next issue of *IBM Personal Systems Technical Solutions.* It will contain a more detailed technical explanation of how these systems operate.

Editor's note: In 1989, IBM and Microsoft announced their intention to deliver a 32-bit version of OS/2. That version is referred in this article as OS/2 Version 2.0. No formal announcement (official product name, content, availability, and so on) has been made by the two companies. Reference to OS/2 Version 2.0 does not indicate a commitment by IBM to introduce a product under that name.

ABOUT THE AUTHOR

W. Craig Chambers is a senior market support representative, supporting RISC System/6000 and AIX in the Engineering/Scientific National Support Center. Previous assignments include lead systems engineer at several large MVS/JES2/JES3 accounts, technical support for communications products such as the 3270 system, 3270-PC, the Workstation Program, and Enhanced Connectivity Facilities, and technical support for OS/2 SE. He has been a presenter for IBM's Technical Coordinator Program television broadcasts and has published several IBM technical bulletins. Craig received B.S.M.E. and M.S.M.E. degrees from Purdue University.

Programming PM Using the COBOL/2 Bindings

Dave Dill
IBM Corporation
Dallas, Texas

Companies have many options when undertaking application development for the Presentation Manager environment. The use of COBOL as the PS/2 development language instead of C is one of these options. This article points out several potential advantages to using COBOL and highlights the restrictions COBOL imposes.

Growing support for OS/2 and the Presentation Manager (PM) graphical interface has created a perplexing problem for many companies. With thousands of COBOL applications, millions of lines of COBOL source code and legions of COBOL programmers, companies are now being told that to use OS/2 effectively, they must retrain their programmers and redesign and rewrite their existing applications.

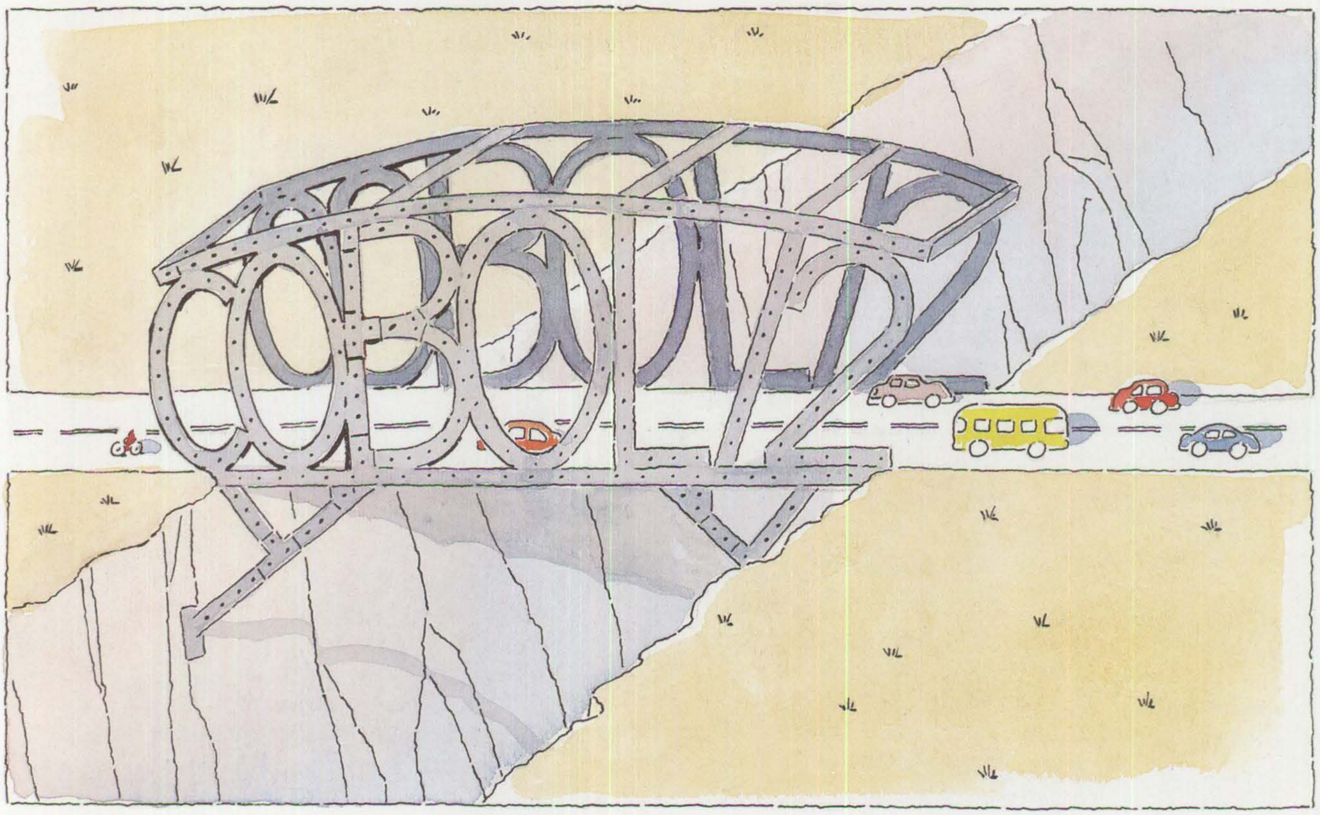
This understandably creates concern. As they wrestle with limited data processing resources, companies cannot always justify rewriting stable, well-performing applications for the sake of a new interface or a new platform.

The perplexing part of this situation is that most people agree that OS/2 is the best platform for future growth and many indicate that OS/2 will eventually be their personal computer operating system of choice. The problem is how to get from here to there at a reasonable cost and within a reasonable time period.

Alternatives to Programming in C

Companies have options when considering the PM as a potential programming environment and they should carefully examine each option. There are alternatives to the extensive retraining and rewriting necessary with a simultaneous conversion to C language and PM.

Contrary to popular myth, C is not the only language that supports the PM environment. COBOL may also be used to produce effective PM applications. While COBOL may not be correct for every application and designers and programmers may still need some C programming experience, for a vast majority of a company's PC applications, COBOL is a real and viable alterna-



tive to writing in C. In some cases, COBOL can actually increase available options and reduce conversion costs by offering the potential of porting existing COBOL applications to the PC platform. Rework would be limited to a new user interface.

The ability to use COBOL as the primary programming language within the PM environment can drastically alter the cost equation for an OS/2 - PM conversion. Companies that cannot justify the move based upon a simultaneous conversion to C and PM should take another look at the costs and time involved with the C language conversion removed from the equation.

Here is where savings might be realized:

Reduced Programmer Retraining: The requirement to simultaneously learn both a new programming language and a new programming environment can place a heavy burden on designers and programmers. While retraining is under way, productivity falls and delivery dates are often extended. The elimination of the new language requirement significantly reduces retraining time.

PM education is significantly shorter when a COBOL programmer is studying COBOL examples and writing COBOL sample programs. The education process is so much faster without the parallel C language education, that COBOL shops intending to convert to C should use COBOL for their initial PM education. The C language training can be deferred until the PM education is complete.

Shorter Development Cycles: Staying with the current programming language could mean staying within the existing development system. That means using existing procedures, methods, coding standards, debugging and testing tools. Keeping all of this in place can have a big impact on development times and product quality.

Unified Programming Staff: The ability to use a common programming staff for program development and maintenance across all platforms increases a company's flexibility. And, it may reduce the size of the overall staff.

There are two distinct types of COBOL compilers for use with PM.

Program Commonality between Platforms: A key justification for the C language is its portability across all platforms. COBOL, too, is portable across platforms. If your company is already using COBOL for the host platform, using COBOL for the PC environment can allow implementation of a very flexible development environment where programs are built using the most economical development process and run on the most economical platform.

The Different Approaches to COBOL Support for PM

There are two distinct types of COBOL compilers for use with PM. While each type of compiler produces approximately the same look and feel, they utilize different programming approaches and philosophies. To make a proper comparison between compilers, it is important to understand their differences and what positive and negative effects each type might have on the development process.

Extended Compilers: This approach uses COBOL compilers that have been modified and extended to accept existing PM calls, support the PM Pascal calling structure, and produce reentrant PM programs. By supporting the existing PM interface, these compilers produce PM programs that have no restrictions in their support of PM. However, because these compilers support C oriented PM calls and the Pascal calling structure, their coding is somewhat more complex than normal COBOL coding. In that case, additional training may be required for a programming staff.

New PM Interface: Rather than modifying the compiler to work with the existing PM interface, this approach changes the PM interface to accommodate standard COBOL compilers. By supporting COBOL's character-based data values and normal calling structure, this approach produces smoother, more standardized code that is very familiar to COBOL programmers. But because this approach uses a standard COBOL compiler, it produces nonreentrant programs that have some PM support restrictions. In spite of the nonreentrant restrictions, however, this interface supports 94% of the 549 common PM programming calls, including 100% of the graphics function calls and 99% of the window function calls. Twelve new PM calls have been added to support PM requirements that standard COBOL compilers cannot process. This new character-based PM interface, called the COBOL/2 Bindings,

```
hwndFrame = WinCreateStdWindow (HWND_DESKTOP, WS_VISIBLE,
                                fFrameFlags, szMyWindowClass, NULL,
                                OL, NULL, ID_RESOURCE, &hwndClient) ;
```

Figure 1. The WinCreateStdWindow call using C language

```
call OS2API '__WinCreateStdWindow'
               using by value HWND-DESKTOP size 4
                     by value WS-Style
                     by reference FrameFlags
                     by reference MyWindowClass
                     by value 0 size 4
                     by value 0 size 4
                     by value 0 size 4
                     by value 1 size 2
                     by reference hwndClient
                     returning hwndFrame.
```

Figure 2. The WinCreateStdWindow call using an extended compiler

```
call "WICTRS" using HWND-DESKTOP,
                    WS-Style,
                    FrameFlags,
                    MyWindowClassSize,
                    MyWindowClass,
                    NULL,
                    NULL,
                    NULL,
                    NULL,
                    ID-RESOURCE,
                    hwndClient,
                    hwndFrame.
```

Figure 3. The WinCreateStdWindow call using the COBOL/2 Bindings

is part of the IBM OS/2 Programming Tools and Information Kit for both versions 1.2 and 1.3.

To show how these interfaces differ, Figures 1, 2 and 3 show the PM call to create a standard window, WinCreateStdWindow. Figure 1 shows the WinCreateStdWindow call written using the C language. Figure 2 shows the same call written using an extended compiler, and Figure 3 shows this call written using the IBM COBOL/2 Bindings. Each example produces the same window.

The Functions of the COBOL/2 Bindings

There are, for each supported PM language, unique requirements that are satisfied by language-specific PM interface code. This code, called the language bindings, is part of the OS/2 Programming Tools and Information Kit. There are bindings for C, Macro Assembler, COBOL and FORTRAN.

The COBOL/2 Bindings include the PM character-based interface supporting COBOL's standard calling convention, new COBOL-oriented calls to replace the existing C oriented PM calls, predefined COBOL structures, constants and messages. Specifically, the COBOL/2 Bindings contain:

- COBOL versions of 527 existing PM function calls
- 12 new COBOL calls
- 158 PM data structures defined in COBOL
- 1500 OS/2 constants defined in COBOL
- 6700 PM messages and constants defined in COBOL, for use in the program's include files

Software Requirements for Using the COBOL/2 Bindings

The list of required software for use with the COBOL/2 Bindings is fairly short. OS/2 Standard or Extended Edition Release 1.2 or higher is required. The COBOL/2

These manuals are available only as part of the Programming Tools and Information Kit:	
Programming Guide	64F0273
Presentation Manager COBOL/2 Bindings Reference	64F0280
Presentation Manager Programming Reference Volume 1	64F0276
Presentation Manager Programming Reference Volume 2	64F0277
Building Programs	64F0274
Control Program Programming Reference	64F0275
These manuals can be ordered through an IBM branch office:	
Programming The Presentation Manager Using The COBOL Bindings	GG22-9463
Systems Application Architecture Common User Access Advanced Interface Design Guide	SC26-4582
Systems Application Architecture Common Programming Interface Dialog Reference	SC26-4356
Systems Application Architecture Common Programming Interface Presentation Reference	SC26-4359

Figure 4. Documentation for Programming the Presentation Manager Using the COBOL Bindings

Bindings are not supported by OS/2 Version 1.0 or 1.1. The COBOL/2 Bindings from the IBM OS/2 Programming Tools and Information Kit Version 1.2 or higher must be installed and available during the compile and link steps. Finally, a COBOL compiler capable of producing OS/2 protect-mode programs must be used. Any compiler specified software is additional.

COBOL PM Programming Documentation

A large amount of documentation is available for PM programming. However, most of it is designed and written for C programmers. While these books contain good information about PM programming, they assume that the reader is familiar with C, so they present only C or Macro Assembler examples. For COBOL programmers, these books are inadequate. There are, however, some books available that can assist COBOL programmers when used for specific information. Most of these books are C oriented, so COBOL programmers must be careful using them.

Figure 4 contains a list of books that a COBOL programmer using the COBOL/2 Bindings will find helpful. These books are in addition to the compiler's documentation and should be used as described.

The *Programming Guide* is a good place to start, especially for COBOL programmers new to PM. This book gives an overview of the PM environment showing the major parts of PM and how programs interact with PM.

Programming the Presentation Manager Using the COBOL Bindings shows how to code the basics of a PM program when using the COBOL/2 Bindings. This book shows how PM calls fit together to form a standard COBOL PM program. All examples are fully explained, and each chapter contains source code for the sample program.

Presentation Manager COBOL/2 Bindings Reference shows, parameter by parameter, how to code each PM call for a COBOL compiler. This is the only COBOL language-specific book included with the OS/2 Programming Tools and Information Kit.

The two *Presentation Manager Programming Reference* volumes contain the only complete description of how each call functions, its associated messages and possible return codes. These books are written for C programmers, so to avoid real confusion, read the text and skip the call definitions and examples. Use the COBOL/2 Bindings book as a coding reference.

PM Functions Not Supported by the COBOL/2 Bindings

Some PM functions are not available to nonreentrant COBOL programs. While the number of unsupported functions is small, it is important to understand each of these functions and its impact on the OS/2 - PM platform to be built.

Sent messages: Nonreentrant COBOL programs may not receive synchronously transferred messages (referred to as sent messages) from PM or other window procedures. Even though PM is a message-based system, this is not as great a restriction as might be assumed. First, nonreentrant programs can receive messages posted directly to the message queue. Many PM messages are posted messages, and nonreentrant programs can process these messages without restrictions.

Second, PM contains a special Language Support Procedure that assists nonreentrant programs in receiving and processing sent messages. Third, while nonreentrant programs may not receive sent messages, they may send these messages. The ability to send these types of messages gives nonreentrant programs the ability to control PM windows.

Specifically, only three PM functions are precluded to nonreentrant programs because of the synchronously transferred message restriction. They are:

- Dynamic Data Exchange – Nonreentrant programs may not exchange data with other PM programs via the DDE support within PM.
- Delayed rendering to the clipboard – Nonreentrant programs may cut, copy to, and paste from the clipboard, but they may not own the clipboard. This precludes the display of the clipboard's contents and delayed rendering of images to the clipboard.
- Owner draw – The ability to add graphics to dialogs, their components, or menus at the time they are displayed is not supported. Graphics may be included on dialogs at the time they are created.

Window Subclassing: Because nonreentrant programs may not receive sent messages, they may not intercept sent messages destined for other window procedures. Window subclassing involves intercepting these messages and thus is not supported.

Advanced Video Function Calls: Commonly called AVIO, this class of calls supporting PM text windows is not available to nonreentrant programs. Nonreentrant programs do support full-screen VIO programs.

The Heap Manager: This facility for managing memory segments and memory objects is not available to nonreentrant PM programs.

System Hooks: The ability to set hooks within PM and receive early notification of events or obtain early access to data is not available to nonreentrant COBOL PM programs.

Help Manager: Use of the Information Presentation Facility to automatically handle help requests is not available to nonreentrant programs. The use of the IPF requires the setting of help hooks but the setting of system hooks is not supported by the COBOL/2 Bindings. All help functions within programs using the COBOL/2 Bindings must be user-written.

The Language Support Procedure for Nonreentrant Programs

The inability to receive sent messages is the most severe restriction on nonreentrant COBOL programs. The sending and receiving of messages is the cornerstone of communications within the PM environment. A failure to fully participate in this communication process would prohibit such a program from running within the PM environment.

To assist nonreentrant programs with sent messages, PM includes special reentrant routines that are automatically invoked to support all windows and dialogs built by a nonreentrant program. These routines are called the Language Support Procedure. These procedures intercept messages sent to the nonreentrant program and, based upon the importance of the message, either posts the message to the program's message queue or processes the message on behalf of the program. The Language Support Procedure gives back to nonreentrant programs the ability to receive and process the important sent messages.

PM Functions Programmed Differently with the COBOL/2 Bindings

Because so much PM function is available to a nonreentrant program, the real comparison between languages should be the area of how the code must be written – specifically, what must be programmed differently when using nonreentrant languages. Here are the six major parts of a PM program that must be coded differently when using the COBOL/2 Bindings.

Window and Dialog Procedures: Procedures, as they are defined under PM, are not supported by nonreentrant programs. Nonreentrant programs do not have multiple entry points, a requirement for PM to automatically locate and execute different procedures for different windows and dialogs. A nonreentrant program must receive all messages through a common entry point and determine which window procedure is to receive the message, based upon the message handle. Then, it performs or branches to the correct program routine. There is no limit on what these routines can do, only that for nonreentrant programs the program must determine which routine gets which message and how that routine is to be executed.

Instance Data: This is data with a common definition that varies for each procedure. For example, if the color of each window were different, the color of a window could become instance data, varying by window procedure. As instance data, PM automatically makes the correct color available to each window procedure through a common definition. Nonreentrant programs must manage their own save areas and determine which data, if any, applies to which procedure.

Notification Messages: These messages are sent from PM to a procedure to give notification that an event has occurred, such as the creation of a window or a dialog. Notification messages are sent messages and are not available to nonreentrant programs. Nonreentrant programs can perform all the functions usually performed when receiving a notification message. These programs simply branch directly to the notification routine after checking the PM call's return code.

Structures and Constants: Some structures and constants cannot be declared using standard COBOL convention. Special PM calls have been added to allow nonreentrant COBOL programs to declare and work with these structures and constants.

Dynamic Storage Allocation: Unlike C programs that can allocate and deallocate memory as needed, nonreentrant COBOL programs must statically declare all PM-related save areas, structures, and constants at compile time.

Special Value of Pointers: Many C implementations of PM routines make use of known pointer values within calculations. Nonreentrant PM programs may not access or make use of the value of pointers.

Meeting the Challenge

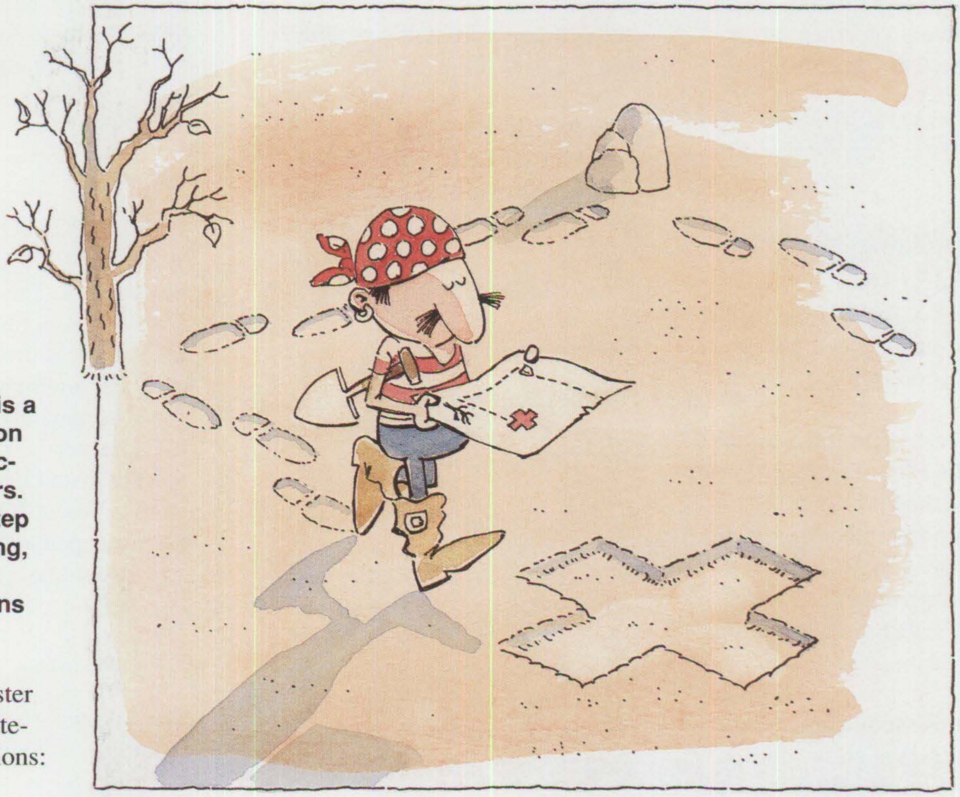
Make no mistake about it, the OS/2 - PM environment is complex, and to achieve all the productivity available from an OS/2 system, PM applications should be complex. But this complexity should not extend the development process itself. Requiring designers and programmers to learn a new programming language will complicate and lengthen the development process. And this process is one already under pressure from normal company demands and the introduction of the OS/2 PM environment.

Just as the C language is not the only language for every PC application, it would be incorrect to claim COBOL as the only PM programming language a company will ever need. A more realistic scenario is a combination of languages, generators and development tools, each able to develop certain types of applications more efficiently and economically. An average OS/2 programming department might use two or three programming languages, a PM generator and several prototyping and debugging tools as a normal mix, with selection of the exact language or tool based upon individual application requirements.

ABOUT THE AUTHOR

Dave Dill is a senior market support representative in the Personal Systems Technical Support Center, providing technical support to IBMers and customers for OS/2 Standard Edition and Presentation Manager. Dave joined IBM in 1968 and worked in the field as a finance industry systems engineer. He then moved to Finance Industry Development where he helped develop several industry and PC-based applications. For the last five years, he has provided market support for PCs and for the DOS and OS/2 operating systems.

Installing and Using the DOS Database Requester



Marty Sirkin
IBM Corporation
Austin, Texas

The DOS Database Requester is a feature of OS/2 Extended Edition 1.2 that allows DOS users to access databases on OS/2 servers. This article contains step-by-step instructions to make configuring, installing, and building DOS Database Requester applications a very simple process.

While the DOS Database Requester supports a full subset of SQL statements, it does have some limitations:

- DOS machines may not be database servers – databases *must* reside on OS/2 Extended Edition 1.2 machines
- Remote precompiling and binding is not supported
  - Precompiling and binding *must* be done on an OS/2 EE workstation
  - Compiling and linking may be done on a DOS machine
- A few Database Manager™ functions are not supported; see the *Database Administrator's Guide* (S01F-0267) for a list of these functions

The DOS Database Requester uses the NETBIOS interface (not APPC or SQLLOO) to communicate with the OS/2 EE server. This is the same protocol that the LAN server and LAN requester uses. OS/2 EE servers may accept requests from both OS/2 EE requesters and DOS Database Requesters concurrently, provided everything is configured properly.

Installation

The following steps are necessary to install the DOS Database Requester on OS/2 EE server workstations and on DOS workstations:

On the OS/2 Extended Edition 1.2 Server:

1. Install the workstation as an OS/2 Extended Edition database server.

2. Include the DOS Database Requester when asked.

This should be done on only one server workstation, because files are copied onto the server's hard disk and later copied onto the DOS workstation.

3. Install NETBIOS support files:

- NETBDD.SYS
- ACSNETB.DLL

by:

- Changing the flag in the Communications Manager profile to load NETBIOS support and then reinstalling, or
- Copying from another workstation, or
- Unpacking them from the installation diskettes using the UNPACK program

The ACSNETB.DLL file should be placed in the \CMLIB\DLL directory, and the NETBDD.SYS file belongs in the \CMLIB directory.

To complete the installation, ensure that the following statement is on one line in the CONFIG.SYS file:

```
DEVICE=C:\CMLIB\NETBDD.SYS
CFG=C:\CMLIB\XXXXXXXX.CFG
```

Notes:

- The statement must follow the LANDD.SYS line.
- XXXXXXXX is the name of the configuration file created with BCS. If it is not present, add it.

4. If a 3270 card is installed in the server, use the PS/2 reference diskette to ensure the DLC and 3270 cards have different interrupt levels. An interrupt level on 3 is fine for the DLC card when both the DLC and the 3270 start have an interrupt level of 2.

5. Set the workstation name to a unique value, which allows the workstation to be located using NETBIOS. You may set this name when installing the system; it may be changed using the System/Reconfigure Database Manager/Workstation Name option of Query Manager.

6. Perform a STARTDBM before using your server to accept requests from DOS workstations. Communications Manager does not need to be started to communicate via NETBIOS for the DOS Database Requester.

On the DOS Database Requester:

The following files can be obtained from the DBDRQLIB subdirectory of any OS/2 EE 1.2 server after answering "yes" to the question "Install DOS DB Requester?"

1. Directly under the root directory on any drive, create a directory called DBDRQLIB. Although the DOS Database Requester can run from a floppy drive, it's preferable to use a hard drive.

2. In the new directory, create a file called DBDRQLIB.CFG. A template of this file is placed on any server that includes DOS Database Requester. The new file must contain the following three statements:

```
SQLNAME = NAME
```

NAME is the DOS workstation name. This statement is used for NETBIOS communications.

```
SQLSIZE = BS/WS
```

BS is the block size used to transmit data. WS is the work size of RAM allocated for the DOS Database Requester. These values can be adjusted, depending on the amount of available RAM, but 4096 and 16834 are good starting values.

```
SQLLIB = DIR
```

DIR is a valid path to the rest of the DOS Database Requester files. There is no restriction against the DIR being a path to the DBDRQLIB directory itself. This is recommended so all DOS Database Requester files reside in the same directory.

3. Place the following three message files (used internally for DOS Database Requester) in the directory listed in the DBDRQLIB.CFG file:

```
SQLLOGON.MSG
```

```
SQLZK001.SRC
```

```
SQLZK001.IDX
```

4. Place the following two files in a directory on the workstation search path. Use these files for logging on and logging off, giving similar function to UPM on OS/2. (*Note*: These files may not be needed because DOS Database Requester has APIs for logging on and logging off.)

```
SQLLOGON.EXE
```

```
SQLLOGOF.EXE
```

5. Install PC LAN Support Program Version 1.1 (or later) on the DOS workstation. This program contains device drivers (.SYS files). The LAN support program is not part of OS/2 EE and may have to be purchased separately (it's included with LAN Server 1.2). Device drivers must be listed in the DOS workstation's CONFIG.SYS file. Due to dependencies among files, they should be listed in the following order:

```
DXMAOMOD.SYS
```

The following arbitrator driver is required:

```
DXMCOMOD.SYS
```

Include the following only if token ring is used for communications:

```
DXMEOMOD.SYS
```

Include Ethernet® in the following if used for communications: (This file is available with PC LAN Support Program Version 1.3 or later.)

```
DXMGOMOD.SYS
```

Include the following only if PC LAN is used for communications:

```
DXMTOMOD.SYS
```

The following NETBIOS driver is required:

```
TIMERINT.SYS
```

This timer interrupt driver is required if the machine has BIOS dated June 1985 or earlier. It's not required for a PS/2.

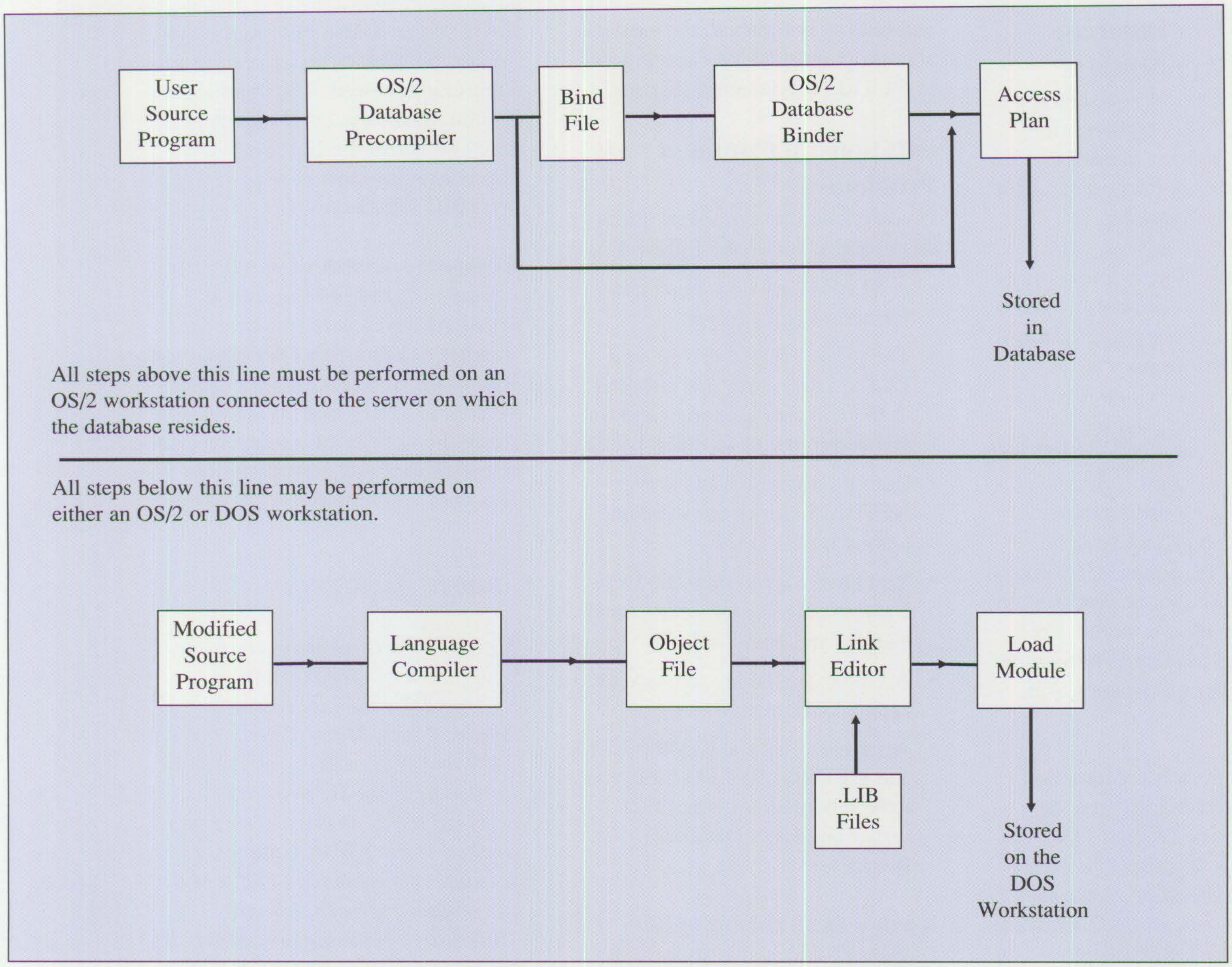

User Source Program
OS/2 Database Precompiler
Bind File
OS/2 Database Binder
Access Plan
Stored in Database
All steps above this line must be performed on an OS/2 workstation connected to the server on which the database resides.
All steps below this line may be performed on either an OS/2 or DOS workstation.
Modified Source Program
Language Compiler
Object File
Link Editor
Load Module
.LIB Files
Stored on the DOS Workstation

Figure 1. How to Build a DOS Database Requester Application

6. Include **cr=y** after the NETBIOS device driver in the CONFIG.SYS file. This forces a hard reset of the DLC card when rebooting the system.

Directory

The DOS Database Requester uses a directory to locate databases on remote servers. Databases must be cataloged before they can be accessed remotely. The DOS Database Requester includes APIs for cataloging and uncataloging databases and also for viewing the database directory.

The database directory for the DOS Database Requester is simpler than the OS/2 Extended Edition database directory structure. All databases must be remote, because DOS workstations are requesters only. Therefore, a system database directory and volume directory are unnecessary. Only the location of the remote database needs to be included in the DOS Database Requester database directory. Each remote database to be accessed is stored as an entry in the directory. Each entry includes the database name and alias, the workstation name of the server on which it is located, and the adapter (0 or 1) of the DLC card. In addition, there are two optional parameters for a comment and code page.

To establish a path from the DOS workstation to the database, the Database Manager accesses NETBIOS and locates the server with the given workstation name. The server then checks the system database directory until it finds the proper volume. The volume directory is checked and locates the correct directory.

Building DOS Database Requester Applications

The DOS Database Requester is a powerful tool. This section briefly describes the process needed to build executable applications for the DOS Database Requester.

Like any application, a source file containing SQL statements needs to be built. It's necessary to precompile and bind the program to the database(s) against which it will run. Precompilation and binding, which must be done on an OS/2 EE workstation, is easiest if done on the server holding the database. A precompile or bind can be done from an OS/2 EE Requester workstation connected to the workstation where the database is stored. Precompiling and binding is necessary only if the program contains SQL statements.

Once the program is precompiled, compilation and linking may be done at either the DOS or OS/2 EE workstation. Real-mode libraries must be used if compilation and linking is done on an OS/2 EE workstation, because OS/2 compilers can use protected-mode versions that do not work on DOS. After compiling and linking is finished, the executable program must be copied back to the DOS workstation (Figure 1).

Solutions to Common Problems

If you encounter a problem during this process, verify the following:

- A STARTDBM has been performed on your server
- The proper device drivers and DLL on the server are installed in the correct directories and referenced in the CONFIG.SYS file
- Both the requester and server NETBIOS names are valid and unique on the LAN
- You have logged onto the DOS Database Requester with a valid user ID and password
- All databases to be accessed are cataloged correctly
- After changing the CONFIG.SYS and DBDRQLIB.CFG files, you have rebooted both the OS/2 server and DOS Database Requester

If only a small number of DOS Database Requester workstations can be connected to the database server, and the LAN requester or LAN server is also running on that server, NETBIOS resources must be carefully balanced. The default configuration for the LAN requester or LAN server in the IBMLAN.INI file consumes most of the NETBIOS resources.

Reduce the number of users, names, commands, and sessions in the IBMLAN.INI file to increase the number of allowable DOS Database Requester connections. After that, be sure to verify that you have not made the number of allowable LAN requester or LAN server sessions less than you require.

ABOUT THE AUTHOR

Marty Sirkin is a senior associate programmer in IBM's Entry Systems Division laboratory in Austin, Texas. Marty joined IBM in 1989 and presently is a developer in Remote Data Services. He received a B.S. in engineering and applied science from California Institute of Technology and a M.S. in computer science from the University of Washington. Marty is on leave from his Ph.D. work at the University of Washington.

OS/2 LAN Server 1.3 Overview

Steven French and Roy Feigel
IBM Corporation
Austin, Texas

This article describes OS/2 LAN Server 1.3, its purpose, structure, and new features. Also described are the changes to the DOS LAN Requester product shipped with it. This new version of IBM's local area network software offers improved performance and usability, yet maintains compatibility.

The OS/2 LAN Server 1.3 was developed because users wanted increased performance, more features, lower memory requirements, and better usability in local area network (LAN) software. The server software and its corresponding DOS and OS/2 requester software, builds on the performance improvements of the previous version, while maintaining compatibility with it.

OS/2 LAN Server 1.3 runs on the new OS/2 Extended Edition 1.3. It benefits from many of the operating system's improvements to the print spooler, loader, and swapper. Other changes include improved usability, locking, and named-pipe performance. The DOS LAN Requester has added support for Ethernet network adapters and Microsoft Windows 3.0.

Some Background

During the last five years, PC LANs have progressed from being an interesting technical curiosity used for specialized applications to an integral part of many businesses. This phenomenal growth is expected to continue, and with it comes the requirement for faster, easier-to-use networks.

Initially, the PC LAN Program (PCLP) allowed small-to-medium-size PC networks to share files and printers. Its performance constraints became apparent; many were because of limitations of the DOS environment for which PCLP was designed. IBM therefore decided to develop with Microsoft an OS/2-based server that would deliver better performance and be easier to use and program than PCLP. The result of this unique joint development effort was OS/2 LAN Server 1.0, and in March 1990, OS/2 LAN Server 1.2.

With OS/2 LAN Server 1.2 came many new features and a completely rewritten file server, logon server, access control system, and DOS LAN Requester. A new common security system, User Profile Management (UPM), that can be shared among products, was added. Usability was improved. An Application Programming Interface (API) was published, enabling software developers to write programs that use network features without requiring users to remember network commands or menu options. Future versions of LAN software will use this common API, allowing for compatibility and interoperability across future releases. Many LAN software products from other companies support the same API.

Customers asked for many changes to LAN Server 1.2, including better performance. As a result, OS/2 LAN Server 1.3 was developed to meet our customers' fast-changing needs.

Structure

As shown in Figure 1, the required software running on an OS/2 LAN Server machine is divided into several parts:

Network services are constantly running network applications.

Network utilities execute for brief periods while users execute com-

What is a Network?

A network is a collection of resources and the machines that share them, and is typically described as having "server" machines and "requester" machines. A shared resource can be a group of files and directories, or it could be a printer or even a serial device (for example, a modem). The processing capacity of a server machine can also be shared by running programs submitted from a requester on a server in the server's memory. The collection of servers and the resources that they share when they are managed as a single unit are referred to by IBM as a "domain."

Users do not need to know where a resource is to access it on a properly configured domain. Domains also contain collections of user definitions and their appropriate access control profiles. Network-connected machines located within one close geographic area are referred to as a local area network, but they may contain more than one domain, and they may be connected through bridges.

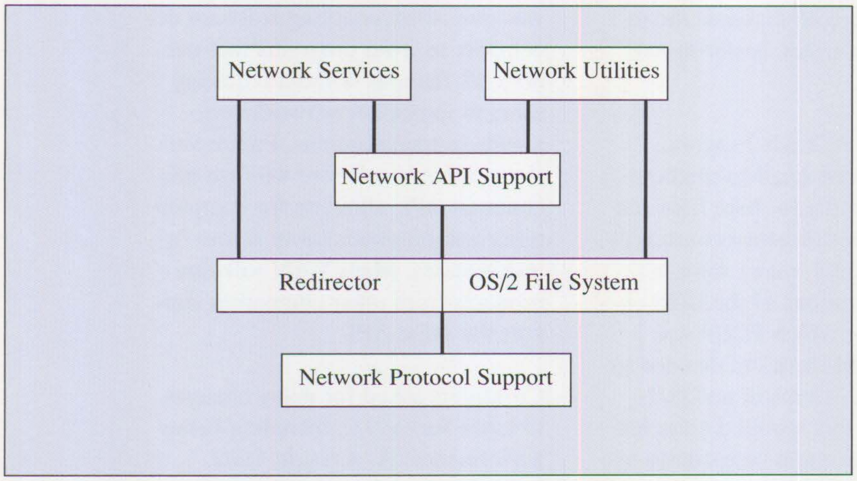


Figure 1. Network Structure

mands either from network menus or from the command line.

Network API support provides the intermediate layer that network applications and utilities use to access network functions. The network API layer directly accesses many of the network data structures such as user and access control information.

The *Redirector* acts as an extension of the *base operating system* that reroutes operating system calls intended for network devices across the network. It uses NETBIOS and its accompanying lower-level *networking protocol support*.

Two examples of the interdependence of the network layers are the "AT" command and the "LOGON" command. Both descriptions are considerably simplified for the purposes of this article.

The AT Command: The first example of cooperation of the layers examines the AT command, which is used for scheduling a command to be run at regular intervals in the future.

When the administrator executes this network utility, a data structure with the names and times for all regularly scheduled commands is updated. To verify that the user has administrator privilege, the AT command must call the network API (NetUserGetInfo( )). The network API must then make calls into the base operating system to retrieve the required information from the user accounts list (NET.ACC file). Once the command has been scheduled, it will be run by a scavenger process in the constantly running "server" service. The server checks when the next AT scheduled command is to be executed, and when the time comes, calls the base operating system to begin a new command processor (CMD.EXE) to run the scheduled command.

If the command involves an operation on a network drive or printer (for example, "DIR g:" where g: is a network connection to a files resource on another server), then the redirector will intercept the command and construct a network command called a server message block (SMB). The SMB is then sent to the NETBIOS device driver by the redirector. The NETBIOS device driver passes the frame to lower-level network protocol support (normally the 802.2 layer), which hands it to the network adapter for transmission across the networking cables to the target server machine.

Notice how all of the layers cooperated in the execution of a single command.

The Logon Command: Another example of the interdependence of the layers is the UPM logon option from the Desktop Manager menus.

The logon menu option, and most programs selectable from the UPM and network menu interfaces, may be thought of as network utilities just like commands entered from the command line. The logon program calls the network API layer to get the default domain name, then calls NetWkstaSetUID2( ) to validate the user ID and password at the domain controller. The NetWkstaSetUID2( ) call updates data structures in the redirector to store the userid and password. These may be used later when reestablishing connections to disconnected resources and for establishing connections to new servers.

The NetWkstaSetUID2( ) call passes to the redirector, which packages the call and sends it to NETBIOS, which sends it to lower-level protocol support for transmission across the network. When the call arrives at the domain controller, the constantly running logon server service (NETLOGON) processes the request and passes it back across the network to the logon program.

Network Services

There are ten network services shipped with the LAN Server 1.2 and 1.3 products (Figure 2.). Because they do not interact with the console (keyboard or screen) and run until explicitly stopped, they differ considerably from the network utilities. The network services are of two types: three requester services, which can run on any OS/2 requester or server machine, and seven server services. Programmers can easily write custom network services of their own by using the OS/2 LAN API.

The largest and most important of the services is the server service, often called the file server. It is actually a collection of more than one process doing many diverse tasks at the same time. Portions of the server code are heavily dependent on the redirector and network API support modules. It uses the most memory of all of the network services and often requires significant tuning for optimal performance. For this reason, it was given much attention during the development of OS/2 LAN Server 1.3. Among the many changes are its print support, memory usage, and improved file locking performance.

With the new spooler contained in OS/2 1.3, redirected printing is significantly faster and network printing features are easier to operate. LAN Server 1.3 can run on machines with less memory than LAN Server 1.2. One of the most significant uses of memory in the file server is for the allocation of "bigbuffs," which are 64 K buffers used for large information transfers across the network. These are dynamically allocated, and do not permanently take up as much memory as with previous versions. These buffers are useful for servers under heavy load servicing large read and write requests. Because more memory is available for LAN Server 1.3, a network administrator can change the IBMLAN.INI configuration file to allow allocation of more bigbuffs on the server. This results in increased performance.

Requester Services	
Net popup	Displays messages on the screen
Messenger	Send messages to other users
Workstation	General requester support
Server Services	
Replicator	Network backup facility
Server	Basic file and print serving
Logon Server (NETLOGON)	Validates user logons
PC DOS RIPL	Support for diskless machines
Alerter	Notifies of important network events
Net Run	Support for remote program execution
DLRINST	Automatic download of DLR to PCLP

Figure 2. Network Services

File locking, one of the most common network operations, has been improved because of changes to server lock queueing strategies.

Some server functions and configuration options have been added. More DosDevIoctl calls are supported across the network and an option to disable automatic dynamic share cleanup has been added. Dynamic share cleanup was necessary on earlier versions that allowed limited numbers of network shares to be active concurrently. Now, many shares can be active at one time, and routine cleanup of shares is not always necessary. Some administrators prefer dynamic shares be automatically cleaned up when not actively in use. Others prefer cleanup of dynamic shares by the network administrator only. Both options are now available.

To speed up many common network operations, such as network administration and network logon, it was necessary to improve performance of some key named-pipe operations. Named pipes are a two-way interprocess communication method used by requesters and additional servers for communication with the logical server and the NETLOGON service. Through changes to the way named pipes are used and to the base operating system support for named pipes, simultaneous named pipe operations are handled better, which allows more requesters to log on simultaneously to OS/2 LAN Server 1.3. It also improves the speed of some of the more time-consuming network administration options when they are run on a busy network. The reliability of NETLOGON service has also improved. On networks where some of the servers have problems with their system clock, LAN Server 1.3 now replicates more efficiently. This is because the system clock is now synchronized with the domain controller at server startup. OS/2 and DOS requesters synchronize their clocks at logon time.

Minor changes have been made to other network services. For example, the PC DOS RIPL (remote boot) service now has increased network security. Periodic improvements will be made to the network services, and new services may be written in the future for special purpose tasks.

Network Utilities

The most visible part of the network, and the part that most obviously distinguishes one network product from one another, network utilities include most commands that a user types in at the OS/2 prompt. Examples are NET USE, LOGON, and AT. Network utilities also include the network full-screen interface and UPM menus. The network utilities are transient, and execute for only brief periods. To maintain compatibility with previous releases and to ease migration, the format of existing LAN Server 1.0 commands was not dramatically altered for LAN Server 1.2 and 1.3. However, many new commands were added.

While the basic structure of the LAN Server 1.3 menus has not changed, the usability of many options was improved, in response to customer requests. Among the most noticeable changes are improvements to simplify managing long lists of data through the menus. Because LAN Server 1.3 domains are expected to be larger than those of previous versions, it was important to improve support for lengthy network administration tasks.

The AT command, which is used to schedule tasks for future execution on a server, was changed to allow those tasks to execute with network administrator privilege. For security reasons, this required that the command be scheduled by a network administrator. Some of the more visible command changes were made to LOGON and LOGOFF, which are now noticeably faster on large, heavily loaded networks.

One of the most important of the network utilities is the network messages and help support. Many messages were added, and help texts were improved to simplify the task of analyzing potential problems.

Redirector

Considered by some an extension of the base operating system, the redirector handles rerouting of operating system requests destined for network devices. When an attempt is made to access a file that is located on a network drive, the redirector intercepts and reformats the request, then sends it to NETBIOS for transmission across the network. An extremely complex and delicate piece of code, the redirector was significantly modified for LAN Server 1.2. This was done to support a new class of server message blocks, as well as to support the new OS/2 high performance file system. LAN Server 1.3 was changed to better support access to PCLP servers. Its behavior is improved when out of resources, and when unexpected network hardware errors are encountered. This portion of the network is largely invisible, and rarely needs to be considered except for a few configuration settings in the IBMLAN.INI file.

Two changes have been made to the redirector heuristic options (WRKHEURISTICs). A user can now choose whether the redirector should ignore requests by an application to flush data to disk immediately. Users may choose this option if running an application that unnecessarily ties up network bandwidth by frequently issuing DosBufReset calls. The other new heuristic allows more flexibility for DOS print buffer timeouts.

While usability considerations for the redirector are few, changes were made to improve the messages that it generates. Messages and helps are now available when unexpected NETBIOS errors are generated.

Network API support

The network API support provides an intermediate layer between the network utilities (and services) and the redirector. It includes support for the many network function calls that applications (including the LAN Server code itself) use. Over 100 API calls in 24 categories are documented and available for use by programmers. Because of critical compatibility considerations, the format of the calls has not changed from LAN Server 1.2 to 1.3. Changes to this layer, except for performance changes, are largely invisible to users.

NETBIOS and Lower-Level Network Protocol Support

Composed of many cooperating layers, the lower level network protocol support includes NETBIOS and the 802.2 layer. For OS/2 machines running on Ethernet (and in the future for those running on other NDIS compatible network adapters) it also includes a protocol manager. It also includes the communication manager configuration features used for NETBIOS and 802.2 support. Discussions of specific changes to these layers is beyond the scope of this article.

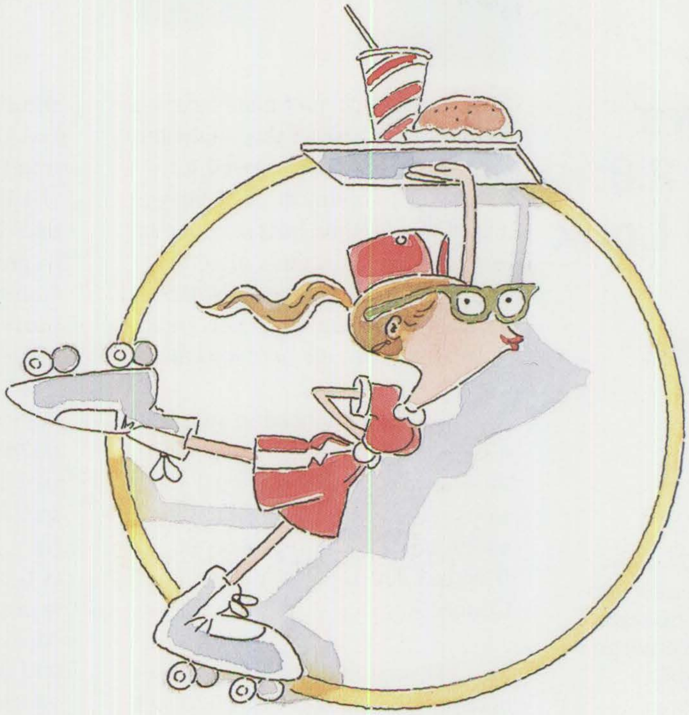
Base Operating System

Changes to the base operating system (OS/2 1.3) are not discussed here, but because some changes significantly affect network performance, they will be described. The base operating system provides crucial features such as the print spooler and manager, presentation manager (PM) and traditional operating system features. The new spooler is significantly faster and easier to manage across the network. Among the most important of the operating system features are memory management and program loading, and both have been improved for network applications running on OS/2 1.3. Applications running on redirector drives no longer swap as heavily when physical memory is overcommitted, and programs load more quickly on redirected drives. A side effect of the improved memory management is that servers and requesters can often run in slightly less physical memory than they did under OS/2 1.2.

DOS LAN Requesters

Because most users of IBM network products operate DOS requesters, improvements to the DOS Requester code are very visible. Structured like the OS/2 LAN products, the DOS LAN Requester is actually a collection of many cooperating layers. Additional DOS command options include a NET STOP command that removes most of DOS LAN Requester from memory when network access is no longer needed. This is important in many memory-constrained DOS systems.

An option to prompt the user for a password has been added to NET LOGON. The API support has been increased so that Microsoft Windows 3.0 can access the network API from protect mode. This allows many network functions, such as using network resources, to be available from within Microsoft Windows. The redirector has been enhanced to improve performance on applications that do many conflicting lock requests.



More memory is now available when starting programs from the application selector.

With LAN Server 1.3, IBM is including DOS LAN Requester as well as DOS LAN Support program version 1.2. Included in this new version of LAN Support program are device drivers for some popular Ethernet network adapters. Helps have also been improved for the DOS LAN Requester.

An Improved LAN Server

With the customer requirements for ever faster and more reliable network products, network software designers are constantly pushed to improve their products. With the release of OS/2 LAN Server 1.3, IBM increased performance, added new features and improved reliability. These changes should help make the LAN Server even more suitable for general-purpose networks, as well as for the customer-designed, custom-programmed networks that are now beginning to appear.

ABOUT THE AUTHORS

Steven French is a senior associate programmer at IBM's Entry Systems Division in Austin, Texas. He joined IBM in 1989 and presently is the technical interface to LAN development. Steve received a B.A. in computer science and an M.S. in electrical and computer engineering from Rice University.

Roy Feigel is an advisory programmer in IBM's Entry Systems Division in Austin, Texas. He is the lead designer for the OS/2 LAN Server 1.2/Extended Edition 1.2 LAN Requester/DOS LAN Requester. In 1982, Roy joined IBM in Austin as a programmer assigned to text applications on the 5520 Administrative System. He received a B.S. in computer science from the University of Southwestern Louisiana.

IBM Windows Connection 2.0: A Graphical Look for Host Applications

Kevin Maier
IBM Corporation
Boca Raton, Florida

IBM Windows Connection is an application that enables users to view host sessions and operate them from the graphical environment created by Microsoft Windows.

Graphical environments for personal systems have become very popular. This is evidenced by the soaring popularity of several graphical user interfaces (GUIs) such as OS/2 Presentation Manager and Microsoft Windows. GUIs are popular because they give users better control of the system, and because they create more productive environments for appropriate applications. Hundreds of applications have been developed for GUIs, and growth is expected to continue.

New applications are being developed to replace virtually all aspects of computer-based interfaces. Also, users are looking for additional graphical applications to run on their systems.

One area in particular is the host mainframe environment typically used within large corporations. For many years, large and bulky dedicated terminals have been replaced with personal computer systems by adding both an adapter card to establish the physical connection and emulation software to drive it. Function and performance have continued to increase in this application area. Within OS/2 Extended Edition, the Communications Manager can place multiple host sessions in windows on the screen, allowing multiple sessions to be operated and viewed. There are, however, some limitations with the user interface.

IBM has seen the need to supply users of Microsoft Windows with this capability of having host sessions within windows and having additional function. As a result, IBM has developed Windows Connection.

The 1.0 version of Windows Connection allowed users of Microsoft Windows 2.X to access their host sessions without leaving the Windows environment. Windows Connection 1.0 permitted only 3270 emulation, so users of 5250 terminals were left out. With the recent announcement of Windows Connection 2.0, 5250 users with Microsoft Windows 3.0 can now have a graphical interface to their host sessions.

Highlights of Windows Connection 2.0

Windows Connection 2.0 is a Microsoft Windows 3.0 application developed by IBM. It allows host sessions to be viewed and operated from the graphical environment created by Windows. It can interact with other Windows applications through Dynamic Data Exchange (DDE) and the Windows clipboard.

Windows Connection 2.0 also enhances the user interface, making many functions, such as file transfer, easy to use. A keyboard remapper is also included.

Windows Connection, however, is *not* a terminal emulator. It requires the availability of an emulator that has a High-Level Language Application Programming Interface (HLLAPI). Windows Connection uses HLLAPI to access the host and to provide host functions. Windows Connection also allows for an improved migration path to OS/2 Extended Edition.

Windows Connection runs on any 80286, 80386, 80386 SX, or 80486-based system having at least one megabyte of memory, a graphics display, a mouse, and an enhanced or AT-style keyboard. Windows Connection requires DOS 3.30 or 4.00, Microsoft Windows Version 3.0, and the appropriate hardware adapter and host communications program.

(*Note:* The amount of memory used with Windows 3.0 varies. In most cases, two to four megabytes of memory gives acceptable performance. The minimum recommended memory yields minimum performance.)

IBM supports the following emulation programs:

- IBM PC 3270 Emulation Program Entry Level version 1.22
- IBM Personal Communications/3270 (with the latest corrective service diskette)
- IBM AS/400 PC Support for Release 3

Using the host communications program, the following communication protocols are supported:

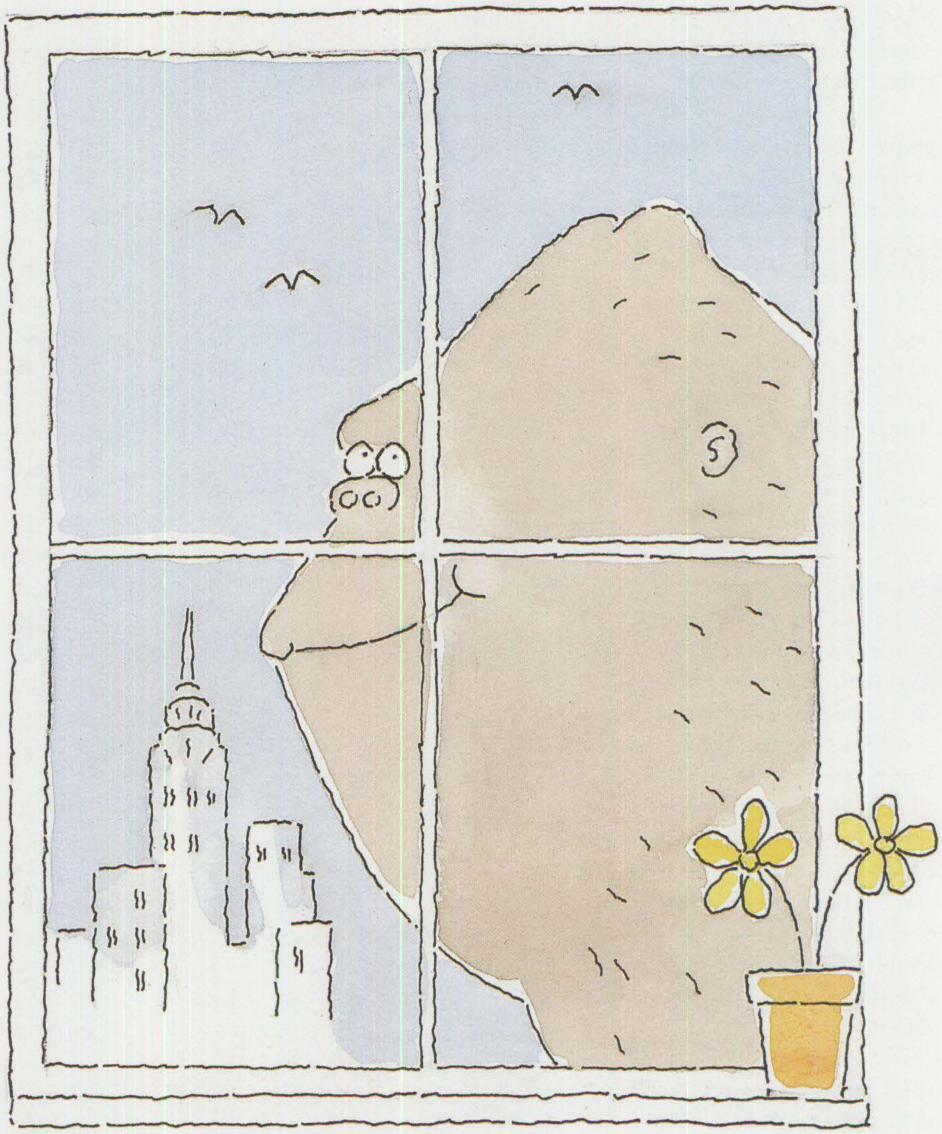
- Synchronous Data Link Control (SDLC)
- Distributed Function Terminal (DFT) mode
- Control Unit Terminal (CUT) mode

- Asynchronous
- IEEE 802.2
  - — IBM Token-Ring
  - — IBM PC Network
  - — Ethernet (requires LAN Support Program 1.2)
- Twin-Axial 5250 (AS/400)

Windows Connection is installed by using the File Manager from Windows and executing the INSTALL.EXE program on the Windows Connection diskette. This allows Windows Connection to add the necessary information to the Windows initialization file (WIN.INI), set up the Windows Connection group, and add the required program items (Windows Connection and the keyboard remapper).

Features of Windows Connection 2.0

Windows Connection has an improved file transfer interface. File transfer is handled via a drop-down menu and a dialog box for filenames and transfer options. Windows Connection also supports multiple file (batch) transfers and can automatically generate the target filename by using templates that can be customized by the user. Both short and long dialog boxes are available, and VM and MVS extended options are supported. This greatly simplifies file transfer for the user. For AS/400™ users, file transfer is done using the virtual drive support within PC Support. These users can employ the system utilities to copy files from one drive path to another.



Windows Connection also allows the user to customize each session independently. The session colors are easily changed within a drop-down menu showing a palette of selectable colors and display attributes. Screen fonts can be changed within a drop-down menu that shows a list of the available typefaces, point sizes, and a sample text string preview. The session title can be modified in the same manner, allowing the user to select a short or long session name, subtitle, session dimensions, session type, and an optional separator character.

Other functional enhancements include *hotspots* and the *pop pad*.

Hotspots allow the mouse to be used for entry of PF keys, numeric topic entry, text string entry, or custom macro execution. Hotspots are defined by the user from a drop-down dialog box showing the available options. The user can enable desired options by placing the mouse pointer on the appropriate option and clicking the left mouse button. This selection is a toggle; clicking the left mouse button again will disable the function.

The pop pad consists of two groups of eight functions each that aid the user in performing common emulator functions. For example, a 3270 keyboard has special keys, such as Clear, PA1, PA2, and so forth. These keys do not exist on a PC-style keyboard. With emulation programs, multiple key sequences, such as Alt-Esc, Ctrl-F9, execute these functions. The pop pad is designed to enhance the user interface for these needed functions.

The pop pad is activated by clicking the right mouse button with the pointer in an active window. The pop pad is then displayed, and the function can be selected. The pop pad has default settings that are usually adequate for most users, but are easily changed by using the keyboard remapper program. When a user points to the desired function button on the pop pad and clicks the left mouse button, the function is performed by Windows Connection, and the pop pad is removed from the window.

Windows Connection also allows copy and paste functions to and from the Windows clipboard. When marking text using the mouse, Windows Connection generates a reverse video preview of the marked area. This allows text information to be transferred easily to and from other Windows applications such as word processors. Because of the single-screen limitation of the host, however, only one full screen of text can be copied to the clipboard at a time. Also, when pasting text to a host session, the user is limited by the definition of fields on the screen – that is, the user cannot paste text into a protected field.

Windows Connection also has a full Dynamic Data Exchange (DDE) interface to Windows. It conforms to the full Windows 3.0 protocol. This allows Windows Connection to function as a DDE server to other Windows applications such as Microsoft Excel and Word for Windows. Full two-way communication and data exchange are allowed. This creates a strong foundation for custom application development.

Windows Connection has a full Dynamic Data Exchange interface to Windows.

Session management and multi-session management viewing are also provided. These allow functions such as Auto-logon, opening and closing of session windows, and linking a macro to a session. The user can also save preferred configurations that enable Windows Connection to open the host sessions as active windows or as icons. A "jump" function is available that permits the user to select between active sessions.

The Key Mapper program allows the keyboard and pop pad definitions to be customized easily. It generates a graphic representation of the keyboard in a window. With a simple point-and-select interface, the user can use the mouse to query the current key assignments and to change them. The pop pad is also shown graphically on the screen, and changes are made in the same manner using the mouse. The Key Mapper also aids the user in macro generation. Specific functions are provided for both 3270 and 5250 emulations, and a help option is available. Changes can be saved easily in a key map file and can be reloaded without closing Windows Connection.

Summary

Windows Connection is a simple application that allows the user to access the host from a graphical environment while enhancing much of the available function. This, coupled with the capability of future application expansion and ease-of-use, makes Windows Connection a must for any Windows user who needs host connectivity.

ABOUT THE AUTHOR

Kevin Maier is a market support representative at IBM's Entry Systems Division Boca Raton Laboratory. Kevin joined IBM in 1977. His experience includes mid-range and personal computers. For the past five years, he has provided technical support for personal computer hardware and operating systems. He is currently in technical support for desktop publishing.

SNA Definitions for 3270 Emulators – Part II

William J. Wen
Houston, Texas

This is the second part of an article begun in our last issue. Here, we will apply the concepts shown in the first part to specific connectivity scenarios. We define six scenarios: a combination of two PC 3270 emulators communicating via the Token Ring Network (TRN) through three types of controller gateways. In these examples, we explicitly point out how the various parameters relate to one another.

In part one, I provided the necessary background information to relate the addressing scheme from the emulators through the controller gateway to VTAM™. With this information, we now will show six sample configurations that result from two 3270 emulators communicating through three different types of TRN controller gateways.

Token Ring Network Controller Gateways

There are three types of TRN controller gateways. One type is the local 3174 controller gateway. 3174 models 01L, 11L, 12L, 21L, and 22L with the TRN 3270 gateway feature can be configured for this gateway. PUs going through the 3174-x1L gateway appear to VTAM as PUs daisy-chained on a channel with the x1L gateway.

The second type is the remote 3174 controller gateway. Ten 3174 models can be configured as remote controller gateways: 3174 models 01R, 11R, 02R, 12R, 21R, 51R, 61R, 52R, 62R, and 90R with the remote TRN 3270 gateway feature. PUs going through a remote 3174 controller gateway appear to VTAM as PUs multidropped connected to a nonswitched SDLC line.

The third type is the 37XX communications controller gateway running the Network Control Program (NCP). Three different communications controllers may be configured for this gateway role: the 3720, 3725, and the 3745 with the Token Ring Interface Coupler (TIC), and running NCP Version 4 Release 2 or higher. PUs going through a NCP gateway appear to VTAM as PUs connected to switched lines.

Some conventions are used in the configuration examples on the following pages (Example #01 through Example #06). Each example has three columns:

- Left column is for definitions from the specific emulator
- Center column is for definitions of the specific controller gateway
- Right column is for VTAM definitions

The description for a field sometimes stretches beyond the bounds of its designated column. For example, with OS23270 LU definitions, the "LU local address (nau address) ...[kk]" definitions are long enough to have stretched into the center column. Whenever possible, the examples used the same field names as in the actual configuration. Some field names are not configurable or are implied; I have enclosed such field names in parentheses. The "Destination Address" field is the exact field name displayed when customizing OS23270. On the other hand, "(SAP ID of OS23270)" is the field name coded into OS23270 and cannot be customized by the operator.

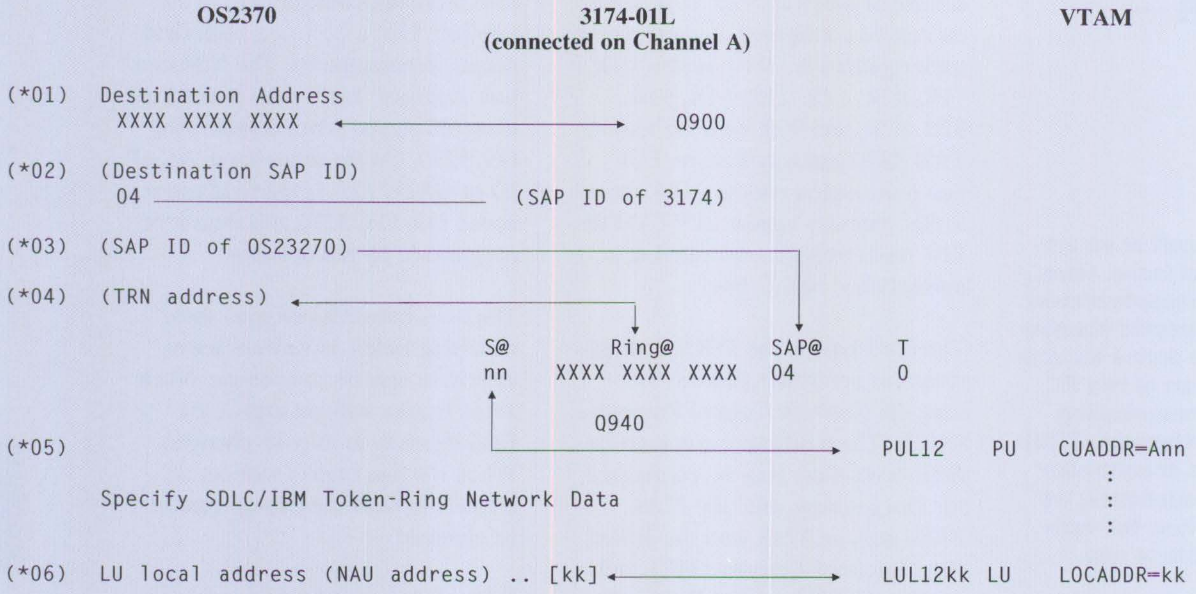
The lines between columns show matching fields. How lines terminate have specific meanings. When a line breaks with an arrow, the field it points to may be changed. When the line breaks without an arrow, the field it points to cannot be changed.

When viewing 3174 customization in TEST mode, you will see the screens in the same sequence as in Figures 1 through 8. I will refer to specific customization questions in the examples. (Part one of this article referred to these figures as Z401 through Z408.)

ABOUT THE AUTHOR

William J. Wen received a bachelor's degree in electrical engineering from the University of Houston – Central Campus. He worked for six cooperative education terms in the IBM National Technical Support Center in Dallas. Bill is employed as a systems engineer, working on system integration and performance analysis of PC systems.

Example #01: OS23270 through a Local 3174 Gateway



```
            OS2370                         3174-01L                              VTAM
                                   (connected on Channel A)

(*01)   Destination Address
         XXXX XXXX XXXX                                 Q900

(*02)   (Destination SAP ID)
         04                          (SAP ID of 3174)

(*03)   (SAP ID of OS23270)

(*04)   (TRN address)

                                   S@       Ring@          SAP@      T
                                   nn    XXXX XXXX XXXX    04        0

                                           Q940
(*05)                                                           PUL12     PU    CUADDR=Ann
                                                                                    :
        Specify SDLC/IBM Token-Ring Network Data                                    :
                                                                                    :

(*06)   LU local address (NAU address) .. [kk]                  LUL12kk  LU    LOCADDR=kk
```

*01 The Destination Address in OS23270 needs to match the TRN address of the 3174, which is in the 3174 customization Question 900 (Figure 5). OS23270 allows the full 12 digits of the destination address to be specified.

*02 The DSAP ID in OS23270 is coded as 04 and may not be customized. This SAP ID matches the 3174 SAP ID. Neither field may be modified.

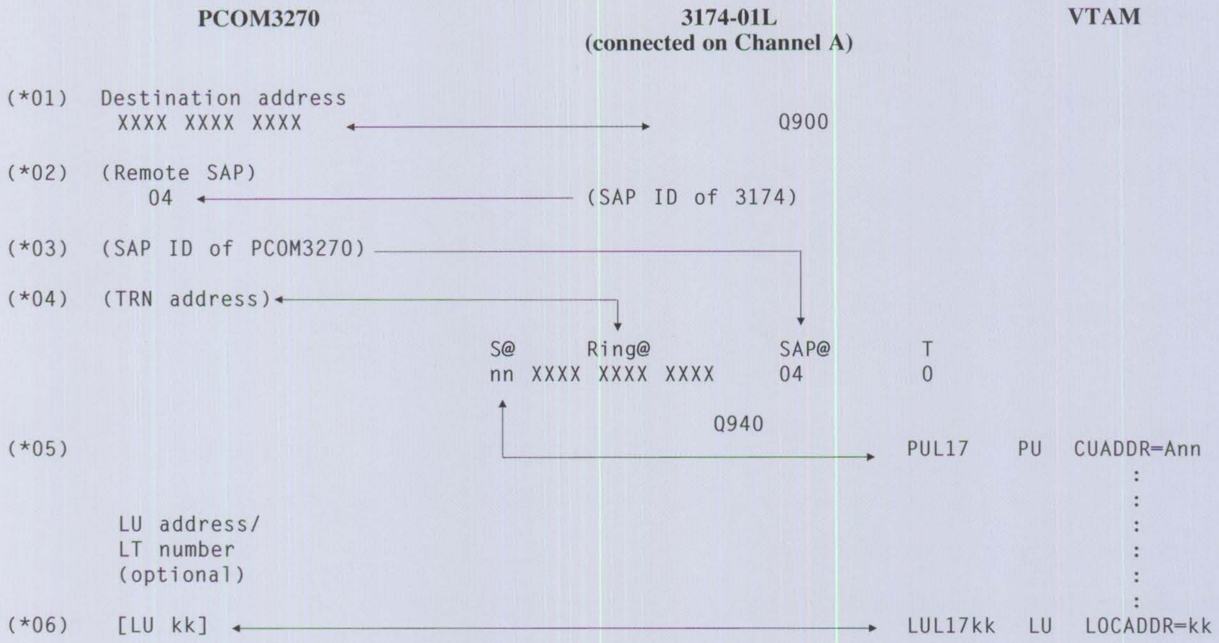
*03 The SAP@ field in the 3174 Question 940 (Figure 6) needs to match the SAP ID of OS23270, which is 04.

*04 The TRN address of the PC running the 3270 emulator needs to be defined in the Ring@ field of 3174 Customization Question 940 (Figure 6). Although it is possible to input the Universal Administered Address (UAA) of the PC TRN card in the Ring@ field of Q940, it is recommended that the PC TRN address be a Locally Administered Address (LAA) instead. Each TRN card has the UAA coded on the card and is unique, as each UAA has to be registered with a central authority. The LAA assigned by the user during installation of the TRN card and needs to be unique in a specific TRN installation. If you specify the UAA of the PC TRN card in the 3174 and later needed to swap out the PC TRN card, your network administrator will need to bring the 3174 offline, recustomize 3174 questions, and re-IML (Initial Machine Load). This set of operations would disrupt the work environment for all other devices accessing the 3174 controller gateway.

*05 Because this is a local 3174 controller gateway, the PUs going through the gateway appear as local devices daisy-chained off a specific channel. Consequently, the TRN address of a downstream PU is associated to the CUADDR= field of a PU definition in VTAM (through the 3174 Q940).

*06 How the LU local addresses are associated to specific sessions on the OS23270 side is different from the DOS 3270 emulators. The DOS 3270 emulators use positional dependence. OS23270 allows the operator to specify what local address is associated to a display session.

Example #02: PCOM3270 through a Local 3174 Gateway



*01 The destination address in PCOM3270 needs to match the TRN address of the 3174, which is in 3174 customization Question 900 (Figure 5). PCOM3270 allows the full 12 digits of the destination address to be specified.

*02 The SAP ID of the controller gateway is specified in the Remote SAP. The value of 04 needs to be specified for this field, as the SNA controller TRN gateways (3174, 3720, 3725, and 3745) all use a 04 SAP ID.

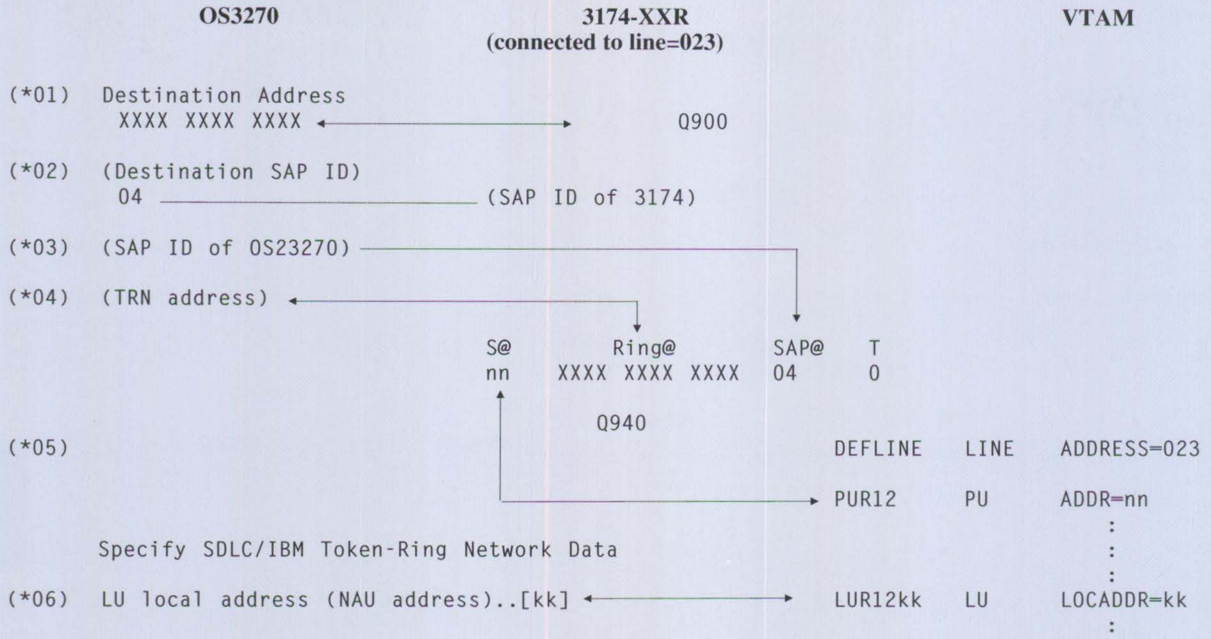
*03 The SAP@ field in the 3174 Question 940 (Figure 6) needs to match the SAP ID of PCOM3270, which is 04.

*04 See item *04 in Example #01.

*05 See item *05 in Example #01.

*06 The LU local addresses are associated differently from specific sessions on the PCOM3270 side to other DOS 3270 emulators. Most DOS 3270 emulators use positional dependence. PCOM3270 allows the operator to specify what specific local address is associated with a display session. Specifying values for the LU address is optional under PCOM3270. If values are not specified, then the default of the first session is LOCADDR=02, the next is LOCADDR=03, and so on.

Example #03: OS23270 through a Remote 3174 Gateway



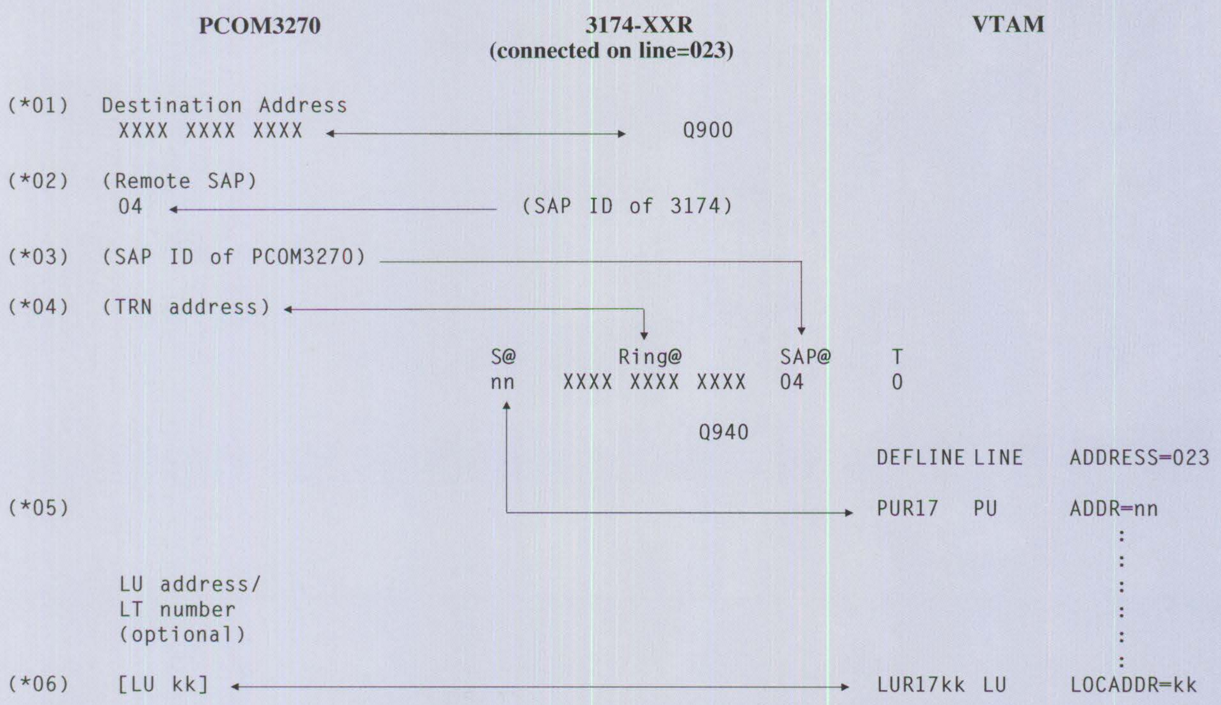
```
                 OS3270                        3174-XXR                         VTAM
                                        (connected to line=023)

(*01)  Destination Address
         XXXX XXXX XXXX <-------------------->            Q900

(*02)  (Destination SAP ID)
         04 ________________________ (SAP ID of 3174)

(*03)  (SAP ID of OS23270) ---------------------------------+
                                                             |
(*04)  (TRN address) <--------------------+                  |
                                          v                  v
                                S@      Ring@          SAP@   T
                                nn   XXXX XXXX XXXX    04     0
                                ^
                                |       Q940
(*05)                           |                        DEFLINE   LINE   ADDRESS=023
                                |
                                +----------------------> PUR12     PU     ADDR=nn
                                                                            :
       Specify SDLC/IBM Token-Ring Network Data                             :
                                                                            :
(*06)  LU local address (NAU address)..[kk] <-----------> LUR12kk   LU     LOCADDR=kk
                                                                            :
```

*01-*04 These matching fields are the same as a local 3174 controller gateway. Refer to items *01 through *04 of Example #01 for information on items *01 through *04 of this configuration, respectively.

*05 Because this is a remote 3174 controller gateway, the downstream PUs going through this gateway appear as remote PUs multidropped off a SDLC nonswitched line. So, the TRN address of a downstream PU is associated with the ADDR= field of a PU definition in VTAM (through 3174 Question 940).

*06 This matching field is associated in the same way as a local 3174 controller gateway. Refer to item *06 of Example #01 for information on this item.

Example #04: PCOM3270 through a Remote 3174 Gateway



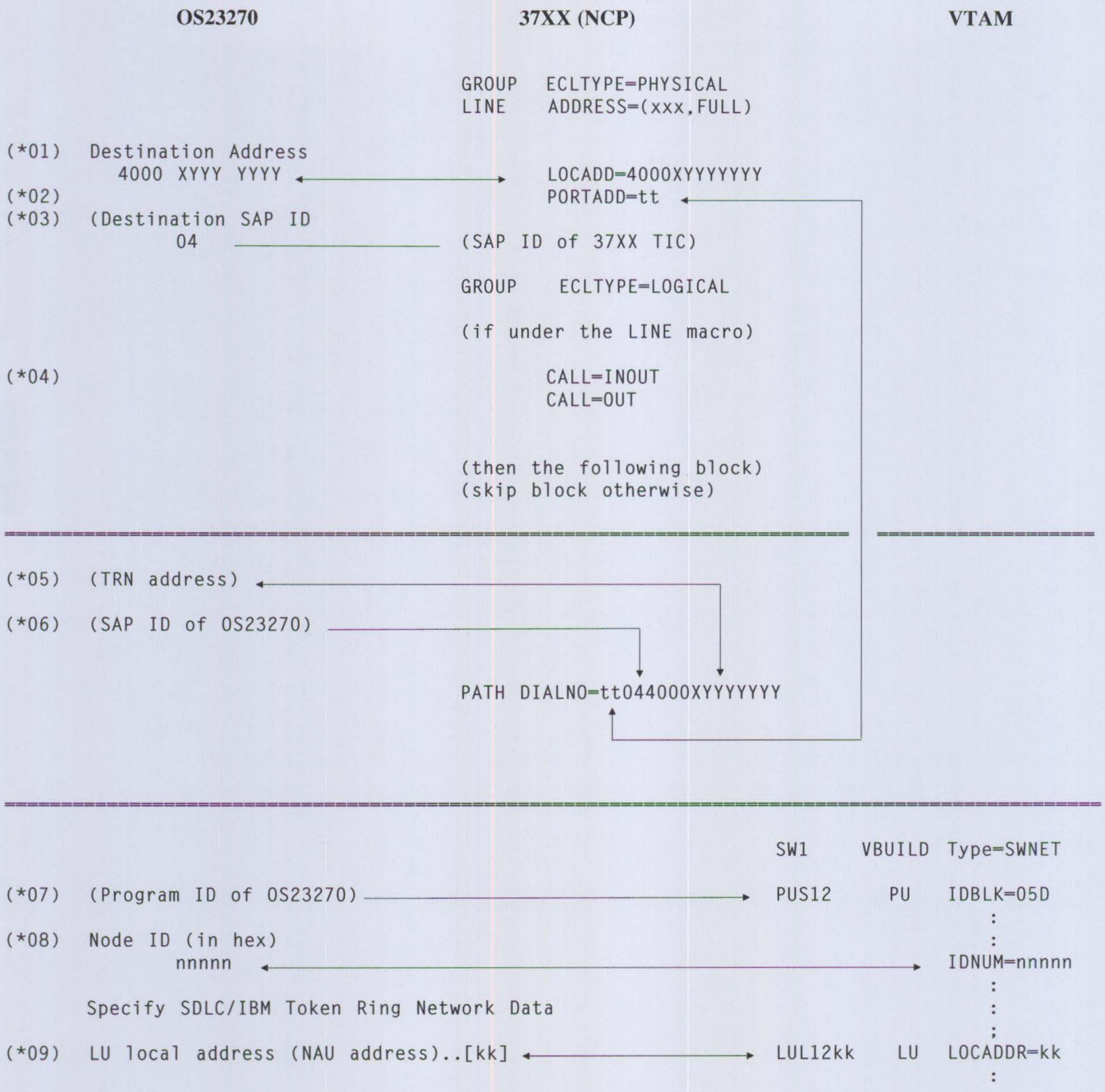
```
                    PCOM3270                    3174-XXR                     VTAM
                                          (connected on line=023)

(*01)   Destination Address
          XXXX XXXX XXXX  <-------------->     Q900

(*02)   (Remote SAP)
          04  <---------------------- (SAP ID of 3174)

(*03)   (SAP ID of PCOM3270) ---------------------+
                                                  |
(*04)   (TRN address) <-------------+             |
                                    v             v
                        S@      Ring@           SAP@      T
                        nn   XXXX XXXX XXXX     04        0
                        ^
                        |               Q940
                        |                               DEFLINE LINE   ADDRESS=023
(*05)                   +-----------------------------> PUR17   PU     ADDR=nn
                                                                          :
                                                                          :
          LU address/                                                     :
          LT number                                                       :
          (optional)                                                      :
                                                                          :
(*06)     [LU kk] <-----------------------------------> LUR17kk LU     LOCADDR=kk
```

*01-*04 These matching fields are the same as with a local 3174 controller gateway. Refer to items *01 through *04 of Example #02 for information on items *01 through *04 of this configuration, respectively.

*05 See item *05 in Example #03.

*06 This matching field is same as a local 3174 controller gateway. Refer to item *06 of Example #02 for information on this item.

Example #05: OS23270 through a 37XX NCP Gateway



```
                OS23270                          37XX (NCP)                          VTAM

                                        GROUP   ECLTYPE=PHYSICAL
                                        LINE    ADDRESS=(xxx,FULL)

(*01)   Destination Address
           4000 XYYY YYYY  <-------------->     LOCADD=4000XYYYYYYY
(*02)                                           PORTADD=tt
(*03)   (Destination SAP ID
                04     ----------------  (SAP ID of 37XX TIC)

                                        GROUP    ECLTYPE=LOGICAL

                                        (if under the LINE macro)

(*04)                                           CALL=INOUT
                                                CALL=OUT


                                        (then the following block)
                                        (skip block otherwise)
===============================================================   ============

(*05)   (TRN address)

(*06)   (SAP ID of OS23270)


                                        PATH DIALNO=tt044000XYYYYYYY


=============================================================================

                                                              SW1     VBUILD  Type=SWNET

(*07)   (Program ID of OS23270) ------------------------>     PUS12     PU    IDBLK=05D
                                                                                  :
(*08)   Node ID (in hex)                                                          :
              nnnnn   <-------------------------------------------------->  IDNUM=nnnnn
                                                                                  :
        Specify SDLC/IBM Token Ring Network Data                                  :
                                                                                  ;
(*09)   LU local address (NAU address)..[kk] <--------------> LUL12kk     LU    LOCADDR=kk
                                                                                  :
```

*01 The destination address in OS23270 needs to match the TRN address of the 37XX TIC, which is defined in the LOCADD= field under the Physical Group for the specific TIC (these definitions are in NCP).

*02 One TIC may have up to four TRN connections; the four TRN connections are distinguished by port addresses. A port address is the PORTADD= field under the Physical Group in NCP. Each port on a TIC has a unique TRN address (LAA), and the same 37XX may have multiple ports connected to the same TRN or to separate TRNs. VTAM does not need to know the TRN address of a specific port, as VTAM communicates through a port address, and NCP translates the port address to a TRN address. To communicate to VTAM, the downstream PU, in turn, needs to know only the 37XX TIC TRN address, not the port address.

*03 The Destination SAP ID in OS23270 needs to match the SAP ID of the 37XX TIC, which is 04. The operator cannot customize the destination SAP ID in OS23270.

*04 There is an option in the LINE macro under a Logical Group in NCP that defines whether VTAM may initiate contact with a downstream PU. Specifying CALL=IN indicates only that the downstream PU can initiate contact; specifying CALL=INOUT indicates that either the downstream PU or VTAM can initiate contact.

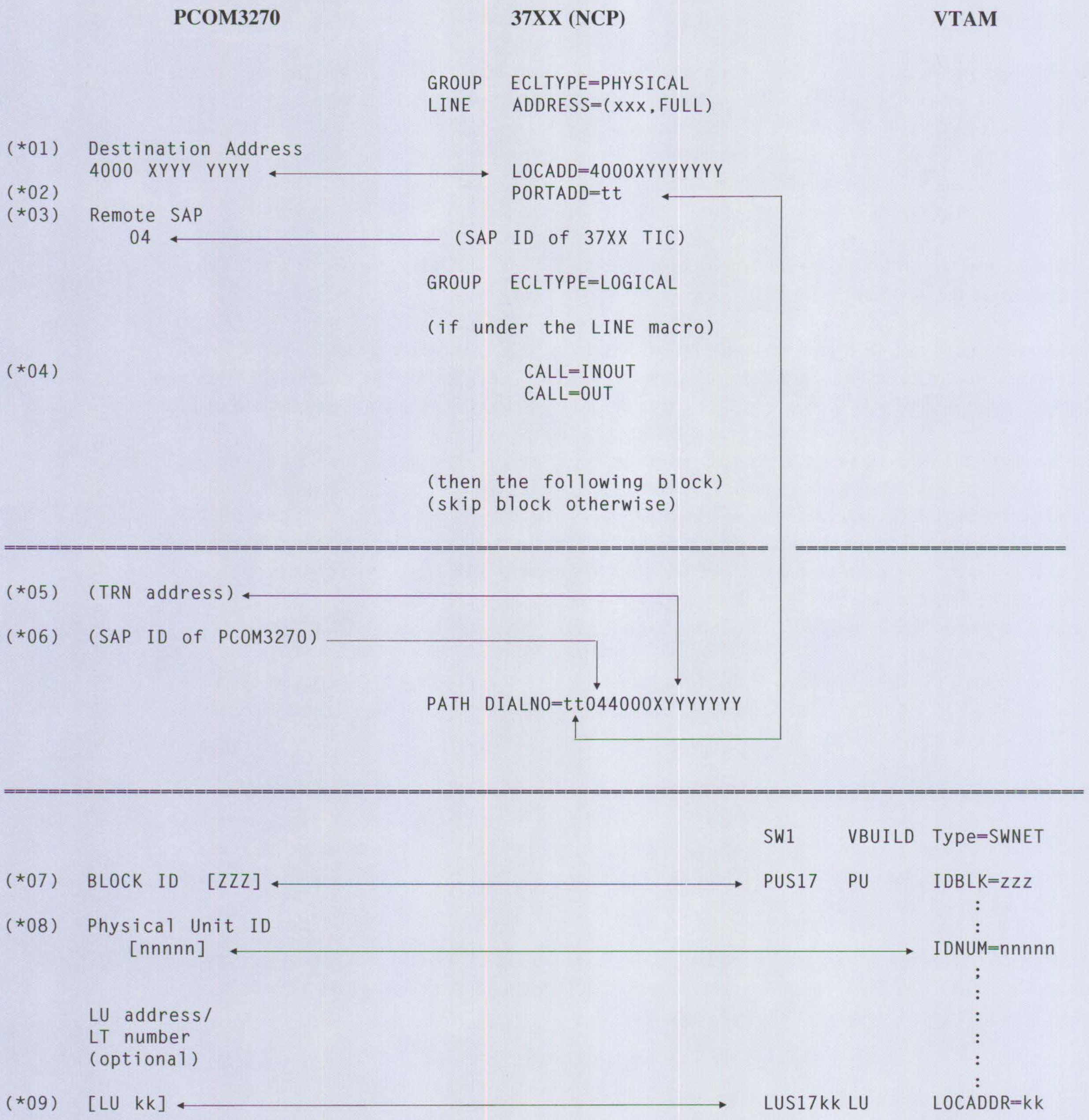
*05-06 These two sets of matching fields apply only when the NCP lines going through the TIC are for host-initiated contact. Keep in mind that downstream PUs on the TRN appear to the host as switched PUs. The DIALNO= field under the PATH macro define a "phone number" that VTAM can use to contact the downstream PU. The DIALNO= field includes information on which TIC address to go through, what destination SAP ID to establish a link with (the destination SAP would be the source SAP ID of the 3270 emulator), and the TRN address of the downstream PU. For a downstream PU-initiated contact, the fields in *05 and *06 do not need to match, as the 37XX dynamically saves the downstream PU's SAP ID and TRN address.

*07 The IDBLK= under a PU definition in a VTAM Switched Major Node needs to match the program ID of OS23270, which is X'05D'. Both the IDBLK= and the IDNUM= fields must match the XID from the downstream PU or VTAM will reject the call.

*08 The IDNUM= under a PU definition in a VTAM Switched Major Node (the same PU definition with the matching IDBLK= field) needs to match the "Node ID (in hex)" field in OS23270. When a downstream PU initiates contact with the host, it sends its program ID together with its PUID (in the form of an XID) to VTAM. This needs to match the IDBLK= and IDNUM= fields, respectively, under a specific PU definition found in a VTAM Switched Major Node.

*09 How LU local addresses are associated to specific sessions on the OS23270 side is different from the DOS 3270 emulators. The DOS 3270 emulators use positional dependence. OS23270 allows the operator to specify what local address is associated to a display session.

Example #06: PCOM3270 through a 37XX NCP Gateway



```
        PCOM3270                          37XX (NCP)                          VTAM

                                   GROUP  ECLTYPE=PHYSICAL
                                   LINE   ADDRESS=(xxx,FULL)

(*01)   Destination Address
        4000 XYYY YYYY  <---------------> LOCADD=4000XYYYYYYY
(*02)                                     PORTADD=tt  <-----------+
(*03)   Remote SAP                                                |
            04  <-------------------- (SAP ID of 37XX TIC)        |
                                                                  |
                                   GROUP  ECLTYPE=LOGICAL         |
                                                                  |
                                   (if under the LINE macro)      |
                                                                  |
(*04)                                       CALL=INOUT            |
                                            CALL=OUT              |
                                                                  |
                                                                  |
                                   (then the following block)     |
                                   (skip block otherwise)         |
=================================================================   =====================

(*05)   (TRN address) <---------------------------------+         |
                                                        |         |
(*06)   (SAP ID of PCOM3270) -----------------+         |         |
                                              v         v         |
                                   PATH DIALNO=tt044000XYYYYYYY   |
                                              ^                   |
                                              +-------------------+

=========================================================================================

                                                              SW1     VBUILD  Type=SWNET

(*07)   BLOCK ID  [ZZZ] <-------------------------------->   PUS17   PU      IDBLK=zzz
                                                                                :
(*08)   Physical Unit ID                                                        :
           [nnnnn]  <-------------------------------------------------->  IDNUM=nnnnn
                                                                                :
                                                                                :
        LU address/                                                             :
        LT number                                                               :
        (optional)                                                              :
                                                                                :
(*09)   [LU kk] <-------------------------------------------->  LUS17kk LU      LOCADDR=kk
```

*01 The Destination Address in PCOM3270 needs to match the TRN address of the 37XX TIC, which is in the LOCADD= field under the Physical Group for the specific TIC (these definitions are in NCP).

*02 See item *02 in Example #05.

*03 The Remote SAP in PCOM3270 needs to match the SAP ID of the 37XX TIC, which is 04.

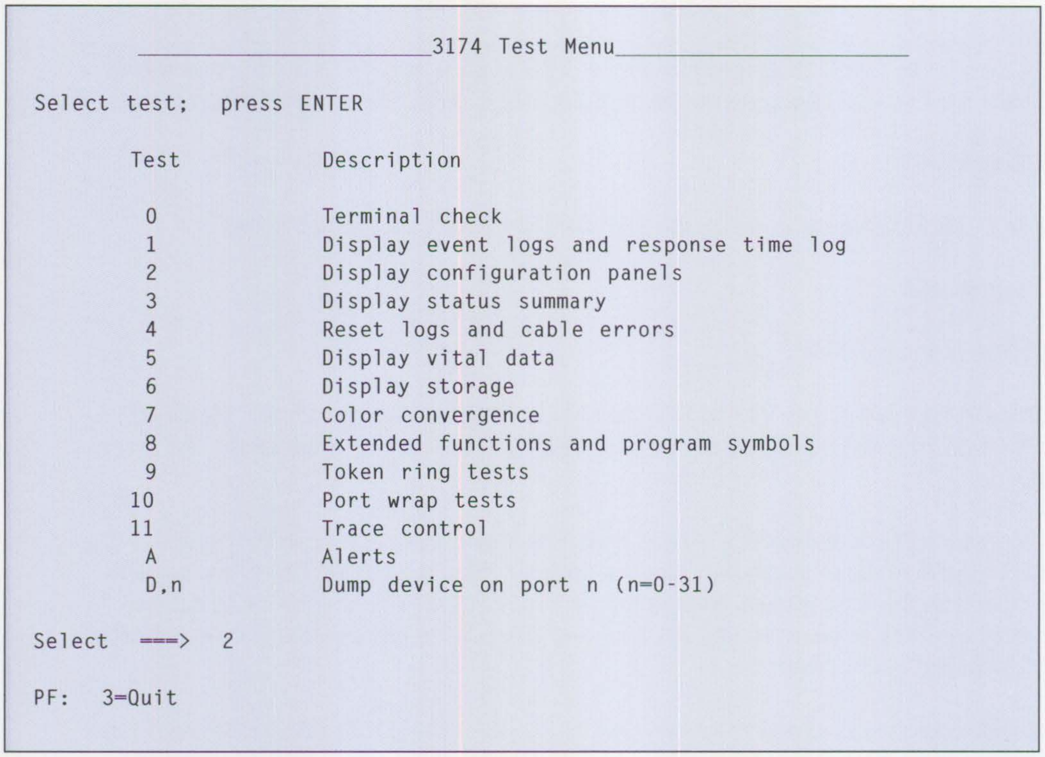
*04 See item *04 in Example #05.

*05-*06 See items *05-*06 in Example #05.

*07 The IDBLK= under a PU definition in a VTAM Switched Major Node needs to match the "Block ID" field in PCOM3270. PCOM3270 is different from any other IBM 3270 emulator because it allows the user to specify the program ID.

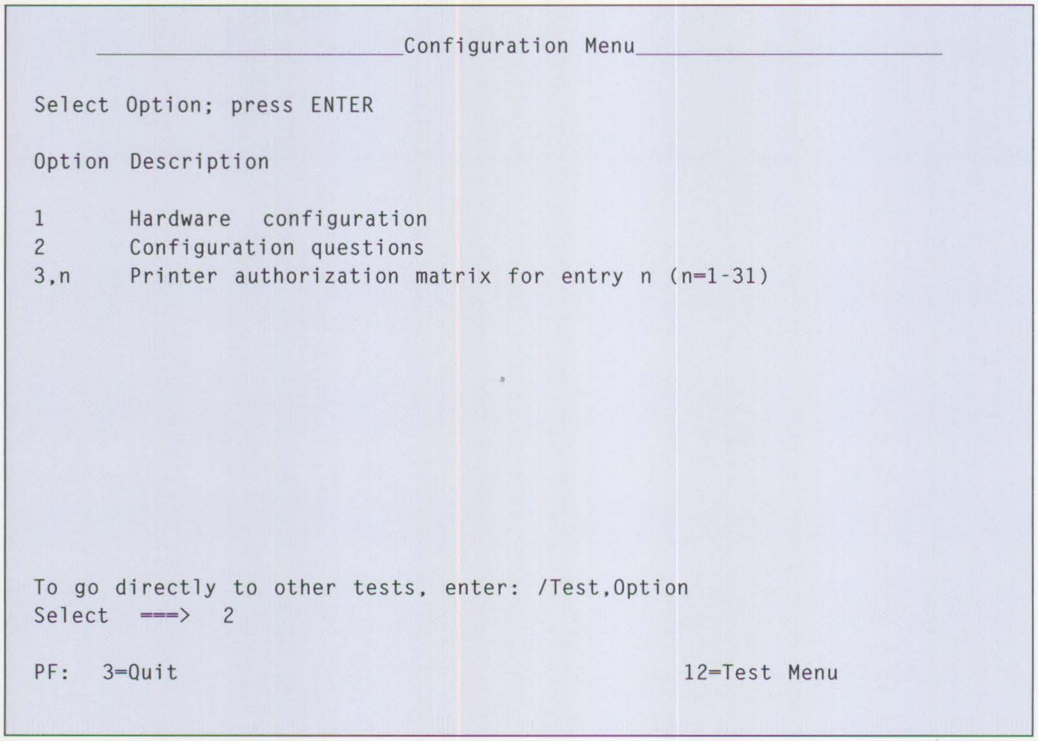
*08 The IDNUM= field under a PU definition in a VTAM Switched Major Node (the same PU definition that has the matching IDBLK= field) needs to match the "Physical Unit ID" field in PCOM3270. When a downstream PU initiates contact with the host, it sends its program ID with its PUID (in the form of an XID) to VTAM, which would need to match respectively the IDBLK= and IDNUM= fields under a specific PU definition found in a VTAM Switched Major Node.

*09 How the LU local addresses are associated to specific sessions on the PCOM3270 side is different from other DOS 3270 emulators use positional dependence. PCOM3270 allows the operator to specify what specific local address is associated display session. Specifying values for the LU address is optional under PCOM3270. If values are omitted, then the default is the first session in LOCADDR=02, the next LOCADDR=03, and so on.



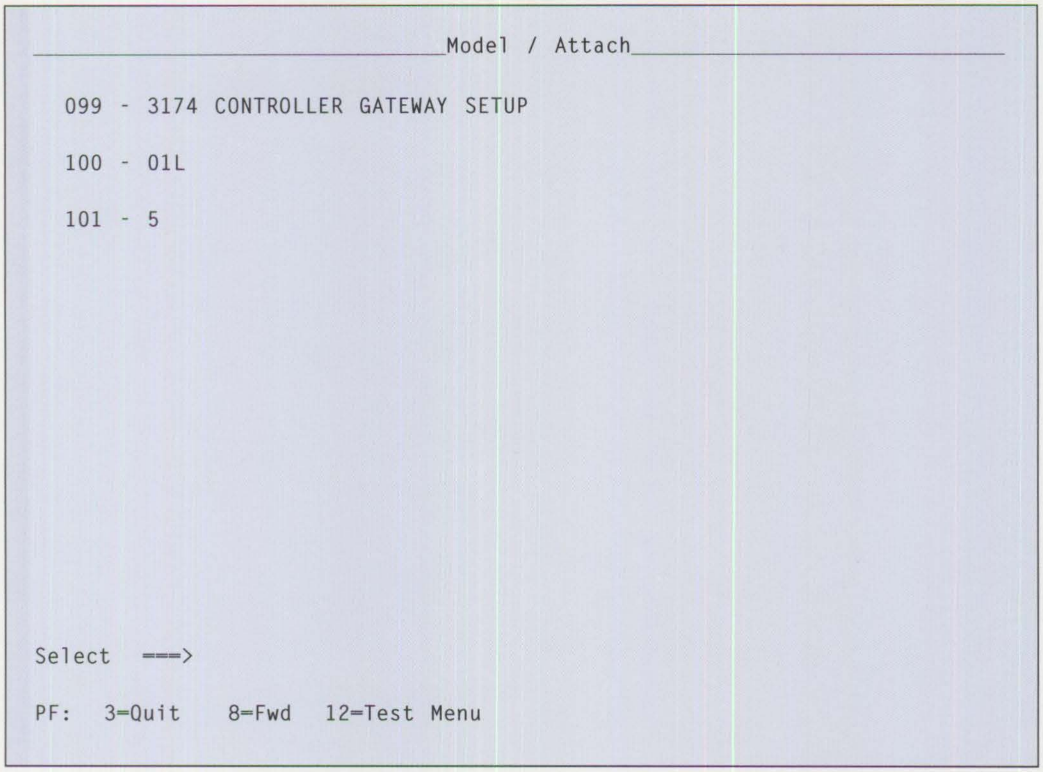
```
                    ____________________3174 Test Menu____________________

  Select test;  press ENTER

         Test           Description

           0            Terminal check
           1            Display event logs and response time log
           2            Display configuration panels
           3            Display status summary
           4            Reset logs and cable errors
           5            Display vital data
           6            Display storage
           7            Color convergence
           8            Extended functions and program symbols
           9            Token ring tests
          10            Port wrap tests
          11            Trace control
           A            Alerts
           D,n          Dump device on port n (n=0-31)

  Select  ===>  2

  PF:  3=Quit
```

Figure 1. First Screen in TEST Mode



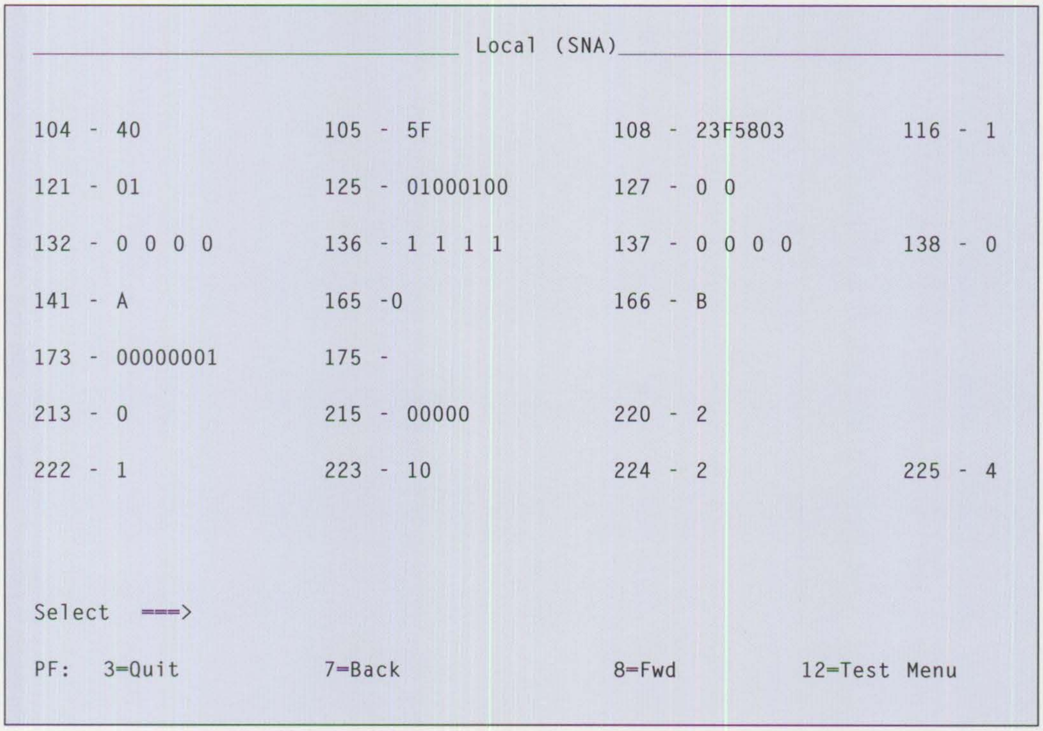
```
                  ____________________Configuration Menu____________________

  Select Option; press ENTER

  Option  Description

  1       Hardware  configuration
  2       Configuration questions
  3,n     Printer authorization matrix for entry n (n=1-31)










  To go directly to other tests, enter: /Test,Option
  Select  ===>  2

  PF:  3=Quit                                       12=Test Menu
```

Figure 2. TEST Mode after Selecting "Display Configuration Panels"



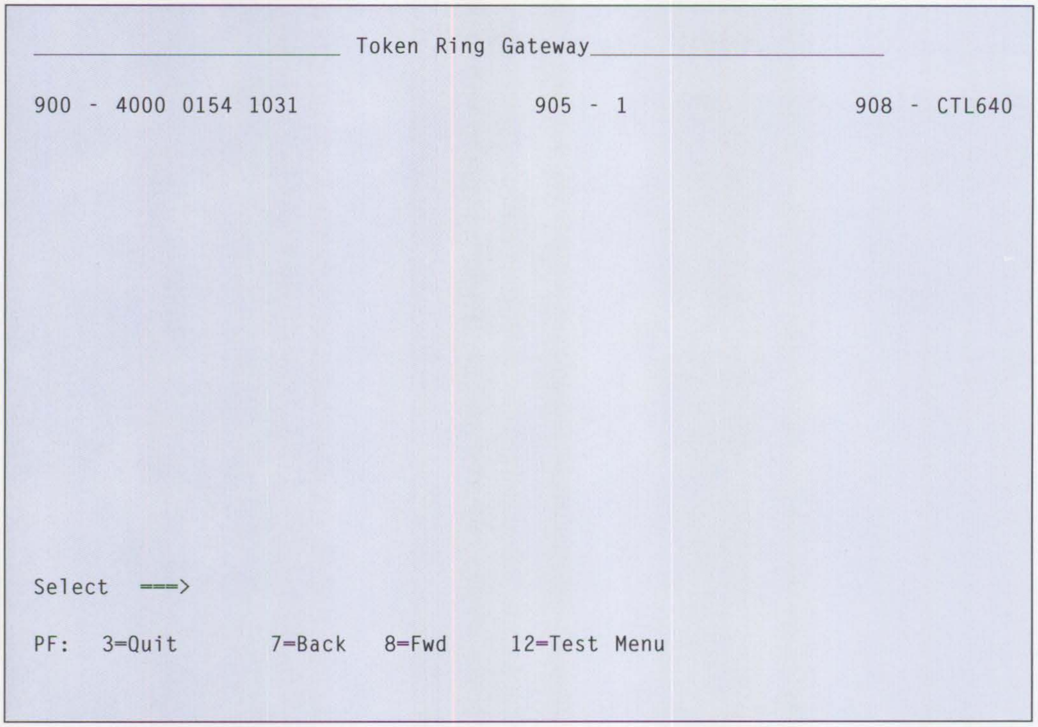
```
_______________________________Model / Attach_____________________________

   099 - 3174 CONTROLLER GATEWAY SETUP

   100 - 01L

   101 - 5









Select   ===>

PF:  3=Quit    8=Fwd   12=Test Menu
```

Figure 3. Controller Attachment Information



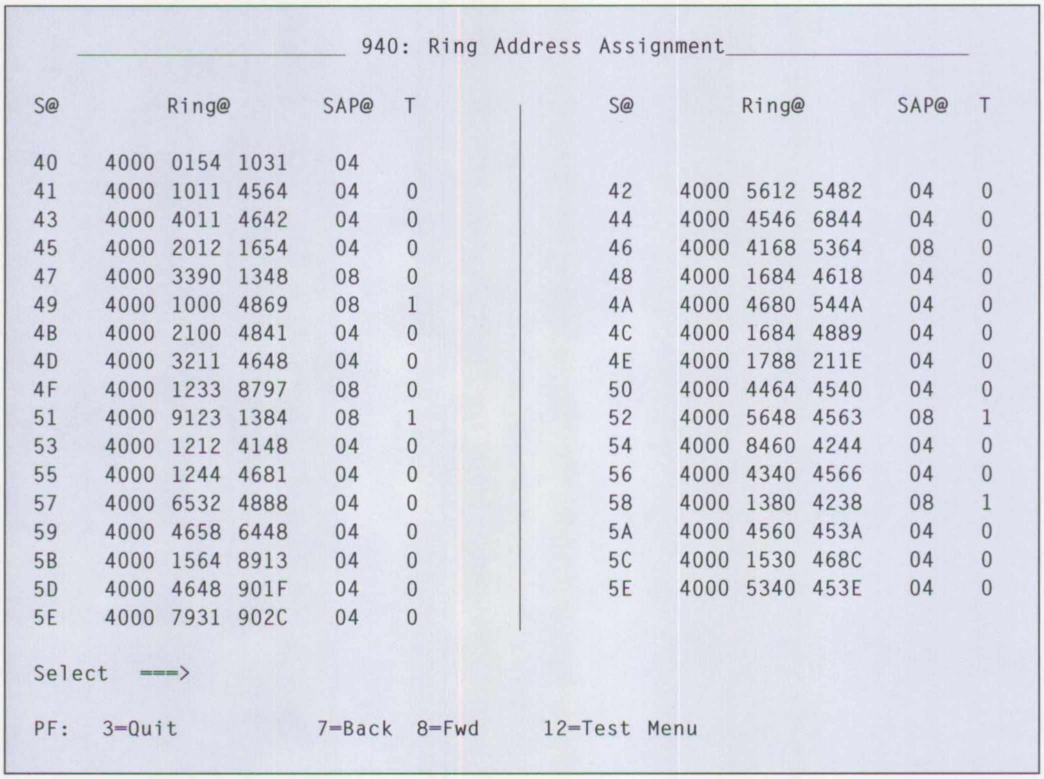
```
________________________________ Local (SNA)______________________________


104 - 40             105 - 5F             108 - 23F5803        116 - 1

121 - 01             125 - 01000100       127 - 0 0

132 - 0 0 0 0        136 - 1 1 1 1        137 - 0 0 0 0        138 - 0

141 - A              165 -0               166 - B

173 - 00000001       175 -

213 - 0              215 - 00000          220 - 2

222 - 1              223 - 10             224 - 2              225 - 4



Select   ===>

PF:  3=Quit          7=Back               8=Fwd         12=Test Menu
```

Figure 4. Local SNA Definitions



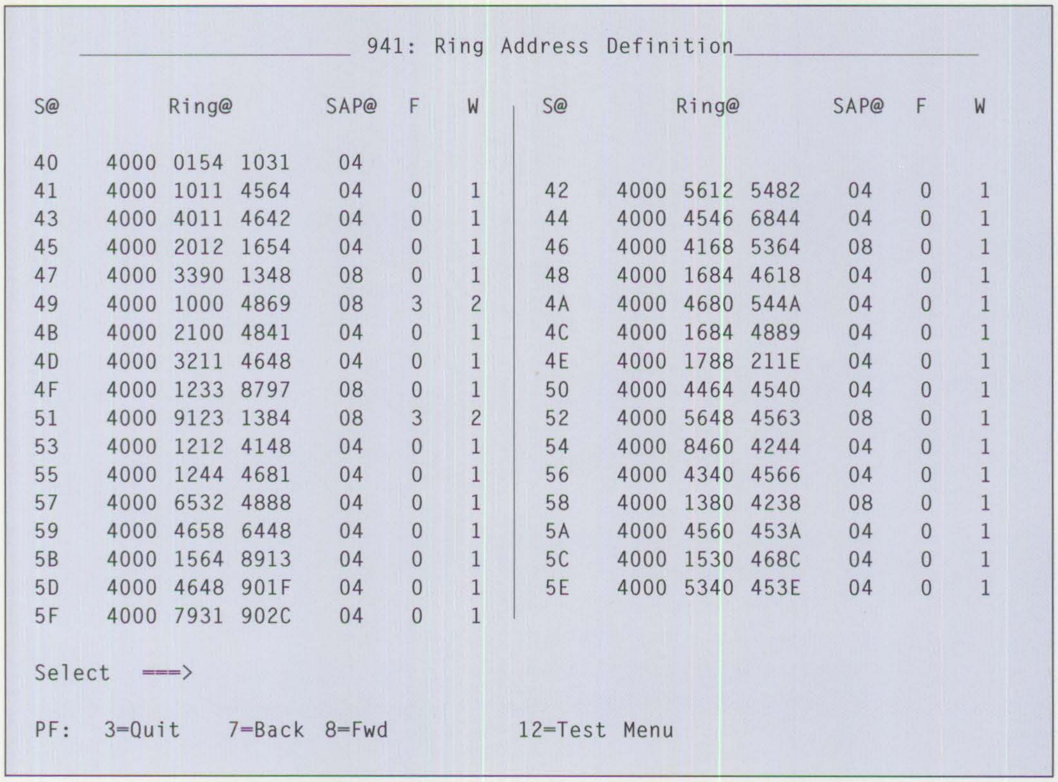
```
              ________________________ Token Ring Gateway_______________________

900 - 4000 0154 1031                    905 - 1                    908 - CTL640













Select   ===>

PF:  3=Quit        7=Back   8=Fwd      12=Test Menu
```

Figure 5. Token Ring Definitions for this Gateway



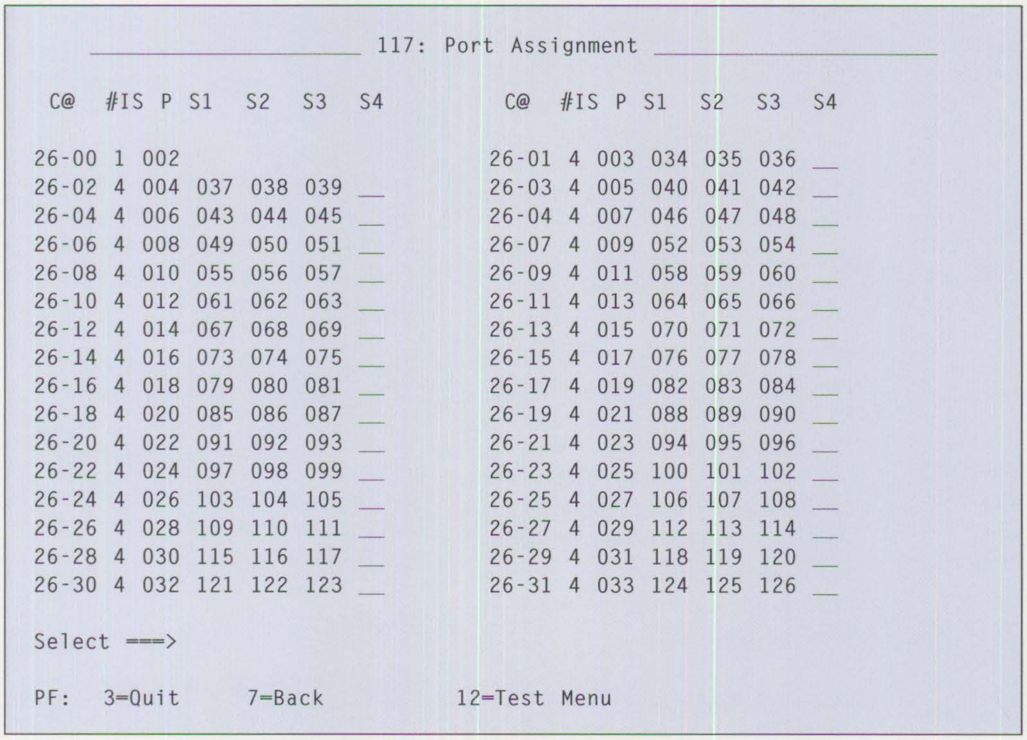
```
     _____________________ 940: Ring Address Assignment___________________

S@         Ring@         SAP@   T            S@         Ring@         SAP@   T

40    4000 0154 1031      04
41    4000 1011 4564      04    0            42    4000 5612 5482      04    0
43    4000 4011 4642      04    0            44    4000 4546 6844      04    0
45    4000 2012 1654      04    0            46    4000 4168 5364      08    0
47    4000 3390 1348      08    0            48    4000 1684 4618      04    0
49    4000 1000 4869      08    1            4A    4000 4680 544A      04    0
4B    4000 2100 4841      04    0            4C    4000 1684 4889      04    0
4D    4000 3211 4648      04    0            4E    4000 1788 211E      04    0
4F    4000 1233 8797      08    0            50    4000 4464 4540      04    0
51    4000 9123 1384      08    1            52    4000 5648 4563      08    1
53    4000 1212 4148      04    0            54    4000 8460 4244      04    0
55    4000 1244 4681      04    0            56    4000 4340 4566      04    0
57    4000 6532 4888      04    0            58    4000 1380 4238      08    1
59    4000 4658 6448      04    0            5A    4000 4560 453A      04    0
5B    4000 1564 8913      04    0            5C    4000 1530 468C      04    0
5D    4000 4648 901F      04    0            5E    4000 5340 453E      04    0
5E    4000 7931 902C      04    0

Select   ===>

PF:  3=Quit           7=Back  8=Fwd      12=Test Menu
```

Figure 6. Definitions of Downstream PUs



```
 ____________________ 941: Ring Address Definition__________________

S@        Ring@          SAP@  F    W  | S@        Ring@          SAP@  F    W

40    4000 0154 1031     04            |
41    4000 1011 4564     04    0    1  | 42    4000 5612 5482     04    0    1
43    4000 4011 4642     04    0    1  | 44    4000 4546 6844     04    0    1
45    4000 2012 1654     04    0    1  | 46    4000 4168 5364     08    0    1
47    4000 3390 1348     08    0    1  | 48    4000 1684 4618     04    0    1
49    4000 1000 4869     08    3    2  | 4A    4000 4680 544A     04    0    1
4B    4000 2100 4841     04    0    1  | 4C    4000 1684 4889     04    0    1
4D    4000 3211 4648     04    0    1  | 4E    4000 1788 211E     04    0    1
4F    4000 1233 8797     08    0    1  | 50    4000 4464 4540     04    0    1
51    4000 9123 1384     08    3    2  | 52    4000 5648 4563     08    0    1
53    4000 1212 4148     04    0    1  | 54    4000 8460 4244     04    0    1
55    4000 1244 4681     04    0    1  | 56    4000 4340 4566     04    0    1
57    4000 6532 4888     04    0    1  | 58    4000 1380 4238     08    0    1
59    4000 4658 6448     04    0    1  | 5A    4000 4560 453A     04    0    1
5B    4000 1564 8913     04    0    1  | 5C    4000 1530 468C     04    0    1
5D    4000 4648 901F     04    0    1  | 5E    4000 5340 453E     04    0    1
5F    4000 7931 902C     04    0    1  |

Select   ===>

PF:  3=Quit     7=Back  8=Fwd          12=Test Menu
```

Figure 7. I-Frame Definitions for Downstream PUs



```
 ____________________ 117: Port Assignment ____________________

 C@    #IS  P  S1   S2   S3   S4          C@    #IS  P  S1   S2   S3   S4

26-00 1 002                             26-01 4 003 034 035 036 __
26-02 4 004 037 038 039 __              26-03 4 005 040 041 042 __
26-04 4 006 043 044 045 __              26-04 4 007 046 047 048 __
26-06 4 008 049 050 051 __              26-07 4 009 052 053 054 __
26-08 4 010 055 056 057 __              26-09 4 011 058 059 060 __
26-10 4 012 061 062 063 __              26-11 4 013 064 065 066 __
26-12 4 014 067 068 069 __              26-13 4 015 070 071 072 __
26-14 4 016 073 074 075 __              26-15 4 017 076 077 078 __
26-16 4 018 079 080 081 __              26-17 4 019 082 083 084 __
26-18 4 020 085 086 087 __              26-19 4 021 088 089 090 __
26-20 4 022 091 092 093 __              26-21 4 023 094 095 096 __
26-22 4 024 097 098 099 __              26-23 4 025 100 101 102 __
26-24 4 026 103 104 105 __              26-25 4 027 106 107 108 __
26-26 4 028 109 110 111 __              26-27 4 029 112 113 114 __
26-28 4 030 115 116 117 __              26-29 4 031 118 119 120 __
26-30 4 032 121 122 123 __              26-31 4 033 124 125 126 __

Select ===>

PF:  3=Quit     7=Back          12=Test Menu
```

Figure 8. LU Local Address Definitions for Coaxial Attachment

Little Solutions

We invite you to share your "little solutions" in this column. Send them to us in care of the editor.

Cursor Speedup

On IBM PS/2 systems, there's an option on the reference diskette to control the speed of the cursor when holding down the arrow keys. Setting the cursor speed to "fast" is ideal for a person experienced with an application because it makes movement through a document faster and reduces the delay between operations.

Simply insert the reference diskette, re-boot your system, and select the "Set Features" option. At the next menu level, select "Keyboard Speed," and then select "Fast." Press ESC until the system tells you to reboot. This change will be saved even after power-off. – *Lincoln Fetcher, Northwest Airlines, St. Paul, Minnesota.*

80386 Memory Expansion Option

There is some confusion about the memory returned upon Power On Self Test (POST) whenever an 80386 Memory Expansion Option is installed on a PS/2. If, for example, you have 8 MB on the system board and 8 MB on the 80386 Memory Expansion Option, why does the POST show only 8 MB instead of 16 MB? The reason is this: PS/2 Micro Channel architecture can initialize adapters in two different ways. Most adapters use the ADF file on the reference diskette or a driver that comes with the adapter. The SETUP program initializes these adapters before POST is run.

The Enhanced 80386 Memory Expansion Option, a more complicated adapter, uses an initialization program that runs after POST. This initialization program stores information in track 0 of the fixed disk. Or, if booting from diskette, a special device driver, DOSMEMDD.SYS, loads the necessary initialization program. Upon power-on, POST checks the hardware and counts the memory it knows about. Because the Enhanced 80386 Memory Option initializes after POST, this memory does not show on the POST memory count. After POST is complete, control is passed to the Enhanced 80386 Memory Expansion Option's initialization program. The adapter is loaded, the remaining memory is then counted, and control is passed to the operating system.

To ensure that the 80386 is installed, an easy test (using DOS 4.01) would be to install a VDISK.SYS of an amount larger than the system board memory – for example, 12 MB. In the CONFIG.SYS, add the following line:

```
DEVICE=C:\DOS\VDISK.SYS 12288
```

If the VDISK.SYS loads successfully, the following should appear after restarting the system:

```
VDISK version 3.40 virtual
   disk x:
Buffer size:12288
```

This verifies that the 80386 Memory Expansion Option is installed and working. – *Alissa Ross, Watauga, Texas*

XGA Adapter/A

The Extended Graphics Array (XGA) Adapter/A is a 32- or 16-bit PS/2 adapter with a Base Video Extension Connector (BVEC). It can be inserted into any 16-bit or 32-bit expansion slot, except for the Auxiliary Video Extension Connector (AVEC) slot in a Micro Channel PS/2 with a 386SX, 386, or 486-based system unit.

The XGA Adapter/A is often installed in an inappropriate slot. In the PS/2 Models 55, 60, 65, 70, and 80, it is tempting to install it into the slot where the 8514/A Adapter must reside. But this slot, known as the AVEC slot, will not accommodate the XGA Adapter/A. – *Jacque Bresnahan, Dallas*

How Much XGA VRAM?

If you need to find out the amount of video RAM that's on your PS/2

Model 90 or Model 95, there's a fast and easy way of doing this without doing a physical inspection of the system board or the XGA Adapter/A.

Use the Model 90 or Model 95 reference diskette, and run Advanced Diagnostics by pressing CTRL A; you will have the option to test the system. At the Device Test Menu, select choice 6 – Advanced Function General Test. At the end of this test, a screen appears that displays the amount of installed video memory. – *Jacque Bresnahan, Dallas*

Callback with FTTERM

The callback security feature of many electronic information systems may be used with IBM's PC/Host File Transfer and Terminal Emulator Program V2.1 (FTTERM).

Callback is the security feature for dialing in and entering a userid and password. The system drops the connection and performs a "callback" to a pre-specified telephone number. This assures that your userid and password cannot be used from any telephone but the one you specified.

From the FTSETUP program, edit the Prefix field under Auto-Dial Definitions adding the number that you want to call. Insert the command definition, enter FTTERM, and dial the number of the callback system. After the system drops the line, you are returned to the Auto-Dial Directory of FTTERM. Press "Enter." You are now waiting for the system to call you back. After receiving the CONNECT message, you're on your way.

This function cannot be automated because there is presently no way to implement it from FTHLLAPI.

Finally, be sure to create an Auto-Dial profile that contains the command **S0=0** and activate this profile at the end of your session. This keeps the modem from answering the line after the session is complete. – *David Randolph, Dallas*

SCSI Performance Tips

These suggestions can improve performance in a DOS single-task environment.

1. Use the utility DISK386.SYS (on the system reference diskette) for the following applications *only*:

- Windows/386 below version 3.0
- QEMM™ below version 5.0
- 386 MAX™ below version 4.08
- 386 MAX Professional™ below version 4.08
- Workstation Program Version 1.12 and prior versions

A better solution to using the DISK386.SYS utility is to update the programs to their latest versions.

If DISK386.SYS utility has been previously installed, and you do not use one of those five applications, then use an editor program to edit your CONFIG.SYS file to remove the statement:

```
DEVICE=DISK386.SYS
```

2. Install the IBM Disk Cache program (hidden file on the system reference diskette). To install the IBM Disk Cache program:

- Ensure that the system has IBM DOS Version 3.30 or 4.01 installed on fixed disk C.
- When the DOS prompt (usually C>) appears, insert the reference diskette into diskette drive A.
- Type A:IBMCACHE and press Enter
- Follow the instructions displayed.
  - Set CACHE LOCATION: (Extended Memory)
  - Set disk CACHE SIZE: (256)KBytes
    For systems with 4 MB RAM or greater, the disk cache size could be (1000)KBytes.
  - Set CACHE PAGE SIZE: (8) Sectors

3. In those cases where applications will perform many short writes (1 byte to 2 Kbytes) to the hard file, applications such as compilers and databases (database indexing), install a disk caching program that is able to do queued writes. Some examples of this type of disk cache program are:

- Super PC Kwik by Multisoft Corp.
- Vcache by Golden Bow Systems
- Flash by Software Masters Inc.

4. Install only one disk cache program on the system at a time. If an application has its own disk cache, remove other installed disk cache programs (such as IBMCACHE and those in performance tip number 3).

5. Where performance in an application that does short writes is needed, consider increasing the number of buffers in the system. The number of buffers in the system is set with a line in the CONFIG.SYS file, such as BUFFERS=30. The performance tradeoff is that each buffer takes available conventional DOS memory space. As the number of buffers is increased, the amount of memory available to run programs is decreased.

6. Always fully populate planar memory to its maximum capacity before installing a memory expansion adapter.

7. There is another utility that performs the same as the DEVICE=DISK386 mentioned in tip number 1. This is the DEVICE=GENS386.SYS utility supplied on the option diskette with the IBM CD-ROM option. For this utility, follow the statements in tip number 1. *–Avery Lyford and George Tracy, IBM, Boca Raton, Florida*

EEINST, REINST, and EECFG

Here's an explanation of the correct use of the EEINST, REINST and EECFG commands in OS/2 EE 1.2.

Use EEINST only when you need to interrupt the installation of OS/2 to format partitions other than C: or to update your CONFIG.SYS file. To use EEINST correctly, do this:

1. During the original installation of OS/2 EE 1.2, you are prompted for diskette #6. (This diskette begins the installation of the Extended Edition portion of OS/2.) At this time, and this time only, you can interrupt the installation process by rebooting your machine. If you have created extended HPFS or FAT partitions and wish to format them and add the IFS=HPFS statement to your CONFIG.SYS file, or perform other system maintenance, reboot your machine now, *without* inserting diskette #6.

2. After you have made all the desired changes to the system, insert diskette #6 into A: and type EEINST to start the installation of OS/2 EE 1.2.

The EE component of OS/2 includes User Profile Management (UPM), Communications Manager (CM), Database Manager (DBM), and LAN Requester. All but one of these components can be installed later with the REINST command. UPM is the part that cannot be installed or reset with REINST. If, after completing the installation, you use the EEINST command, ERROR 01=UNPACK will appear. This indicates you are attempting to install UPM code onto files that already exist, causing damage to existing files. Once this error is received, REINST is no longer available. In order to use REINST again, the Base Operating System and EE portion must be reinstalled. (Formatting the drive is not neccessary.)

Use EEINST only *before* installing files from diskette #6. Once any files have been installed from diskette #6, the REINST command must be used to install EE.

Use REINST when installing portions of EE such as LAN Requester, or removing an unneeded component such as Database Manager or old CM configuration files. REINST is also used after you have used the Advanced Configuration utility of CM to add new profiles, such as 5250 terminal emulation or Remote Data Services (RDS). REINST is used to install the code or new features for the new CM profiles that weren't installed using the Basic Configuration File Services (BCS) and the EECFG command. To use REINST:

1. Insert diskette #6 into A: and type REINST at the OS/2 command prompt.

2. Follow the menus to install or remove configuration files or components.

EECFG is the command to create and install a new Basic Configuration Service file for communications. To use EECFG:

1. Insert diskette #6 into A: and type EECFG.

2. Select the features of CM you need, such as LAN services, Remote Data Services, 3270 or 5250 emulation, and ASCII emulation.

EECFG can be thought of as a combination of Advanced Configuration of CM and REINST, because after the .CFG file is created and installed into the x:\CMLIB\ subdirectory, REINST is used (implicitly) to install the needed code. This has caused some confusion, because from here on, EECFG and REINST are the same. Remember, EECFG is used to create a new .CFG file, and REINST is used to add the new code and related features for an updated .CFG file.

The EECFG and REINST commands are documented in Chapter 2 of *IBM Operating System/2 Extended Edition Version 1.2 Getting Started (84X0841). – Rusty Thomas, Fayetteville, Arkansas*

IBM THINKable

Bill Frank
IBM Corporation
Boca Raton, Florida

THINKable is a multimedia software program for the PS/2 that can help therapists treat people suffering attention and memory loss as a result of injuries or disabilities.

IBM always has been strongly committed to making computer technology available for people with disabilities. State-of-the-art computer technology has been applied to a series of products to help those with disabilities lead independent lives; thus the title applied to this line of products – The Independence Series™. *Screen Reader™* helps the visually impaired to become computer users through voice communications. *SpeechViewer™* is a clinical tool used by speech pathologists and professionals to increase the efficiency of speech therapy. *PhoneCommunicator™* brings a wide range of telephone communication options to the fingertips of people who are hearing- or speech-impaired. These products were designed to assist disabled people to function in the workplace and at home. (The Independence Series was described in *IBM Personal Systems Technical Solutions*, Issue 2, 1990, G325-5006.)

THINKable is the latest addition to the Independence Series of products for people with disabilities. It is a clinical tool that can help people with cognitive impairments develop memory and improve their attention spans.

Imagine trying to go through life unable to remember people, your own phone number, or unable to focus on an issue or task before you. It would be frustrating and upsetting.

Approximately 12 million individuals suffer from memory loss as a result of injury, developmental disabilities, substance abuse, degenerative disease, or neurological disorders. Without the benefit of memory and a reliable attention span, the quality of life for all those people suffers. And the rest of us are denied the full talent, creativity and contributions that those 12 million people can bring to our society.

THINKable is a software program that operates on the Personal System/2 using multimedia technology. The program uses graphics capability to help those people with cognitive impairments redevelop memory and improve attention span through visual association.

THINKable enables psychologists, occupational therapists, speech and language pathologists, and doctors to assist the patient or client in working independently. The software creates a structured environment in which skills can be practiced in four critical focus areas that relate to activities of daily living:

- Visual attention
- Visual discrimination
- Visual memory
- Visual sequential memory

Within each focus area are four levels of cognitive difficulty. Therapists can use sample practice strategies provided with THINKable or tailor practice sessions to meet individual client needs.

Animation, photos, and natural voice are used to prompt users through practice sessions that help improve memory, attention, discrimination, and sequencing. Sample tasks that can be performed using THINKable include responding to specific images that appear on screen, matching pairs of images that are alike or not alike, and recalling the order in which images appear.

For example, voice or text, or both, can prompt the user to touch a target image, like a chair, and on the next screen to identify images that are different, like an ice cream cone or flowers.

Case Management

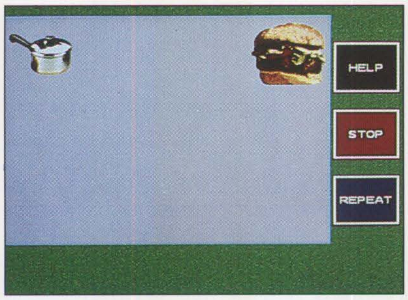
THINKable uses a hierarchical structure consisting of elements, treatment components, session plans, and workbooks that parallel the way treatment professionals currently plan for therapy. It includes a starter set of exercises for immediate use. These exercises can be modified for specific individuals, and specialized treatment plans can be created. Online forms make it easy to design exercises that offer sufficient cognitive difficulty to give patients a feeling of accomplishment. Up to 14 treatment sessions can be planned in advance.

The software also includes automatic data collection, analysis, and reporting. It enables clinicians to easily generate graphic and tabular reports for case managers, doctors, families, and insurers.

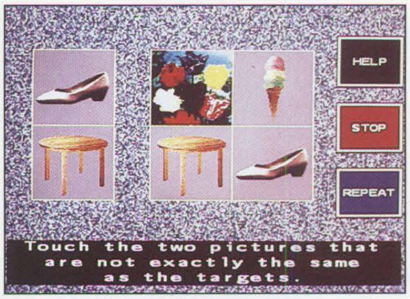
Convenience Kit

THINKable is available in a Convenience Kit that includes:

- Six 3.5-inch program diskettes
- THINKable Getting Started
- THINKable Introduction Videotape
- THINKable Reference
- Audio Capture and Playback Adapter (ACPA)
- Two sets of headphones
- One Y-cord adapter

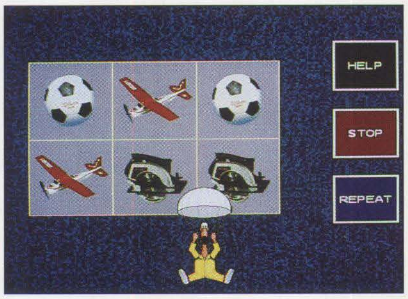


Visual Attention

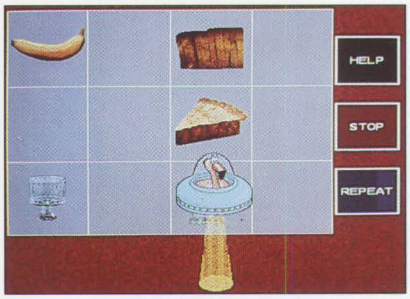


Touch the two pictures that are not exactly the same as the targets.

Visual Discrimination



Visual Memory



Visual Sequential Memory

Supported Systems

THINKable is supported on Personal System/2 models with 80286, 80386, or 80486 processors.

THINKable requires a minimum of 4 MB of free memory, a 30 MB fixed disk, and OS/2 Standard Edition, Version 1.3. Those who plan to use OS/2 Extended Edition must have the memory and fixed-disk size required by that version.

For information about purchasing THINKable, call 1 800 228-0752.

THINKable is marketed by The Psychological Corporation, San Antonio, Texas.

ABOUT THE AUTHOR

Bill Frank is a marketing plans program administrator for Special Needs Systems at IBM's Entry Systems Division Laboratory. Since joining IBM in 1968, Bill has held several management positions, including regional administration manager. He has held headquarters staff planning positions for Field Engineering Division, Americas/Far East, and Entry Systems Division. He is currently the product planner for THINKable. Bill holds a B.S. from Bradley University and an M.B.A. from DePaul University.

IBM Personal Systems Developer

The *IBM Personal Systems Developer* is a quarterly publication written primarily for OS/2 application programmers. It features a variety of technical articles, such as programming tips and techniques, product reviews of new software tools, application development case studies, and interviews with OS/2 industry leaders. Examples of articles from recent issues are:

"Performance Tips and Guidelines"
"Writing OS/2 Device Drivers"
"Corel Draw Migration to OS/2 PM"
"Spotlight on Autodesk"

IBM employees, customers, and software vendors write articles for the *Developer*. The magazine is published as part of IBM's Developer Assistance Program, which offers a variety of support services for companies that are writing OS/2 applications for resale.

Subscriptions can be ordered by calling the publisher at 1-800-READ-OS2. IBM employees can subscribe through Mechanicsburg's Systems Library Subscription Service (SLSS), order number G362-0001.

Articles from the first seven issues of the *IBM Personal Systems Developer* have been published in a 770-page book titled *OS/2 Notebook: The Best of the IBM Personal Systems Developer*. The book can be bought at a local bookstore or by calling Microsoft Press at 1-800-MS-PRESS.

New Products

Hardware

IBM Proprinter 24P

The IBM Proprinter 24P is a low-cost, narrow-carriage, 24-wire, serial dot matrix impact printer for attachment to IBM Personal Computers and non-IBM PC hosts. The Proprinter 24P produces excellent letter-quality printing and bidirectional alignment, in addition to high-resolution graphics printing. It offers adjustable flat-belt tractors that are easy to load and reliable, and a new user-friendly operator panel that can be understood by the novice user. The IBM Proprinter 24P is a good price performer for the business workstation or home use. Additional efficiency is provided by the automatic sheet-feed and pull-tractor options, both of which are currently available on the IBM Personal System/1™ printer.

Letter # 191-881, January 22, 1991

IBM Token-Ring Network 16/4 Busmaster Server Adapter/A

The IBM Token-Ring Network 16/4 Busmaster Server Adapter/A is used to attach an IBM PS/2 with Micro Channel® architecture, configured as a server, to an IBM Token-Ring Network operating at either 16 million or 4 million bits per second. The IBM Token-Ring Network 16/4 Busmaster Server Adapter/A provides high throughput capability while requiring less host-processor involvement than previous IBM Token-Ring adapters.

This reduction in host-processor involvement is enabled through the busmaster capability of the IBM Token-Ring Network 16/4 Busmaster Server Adapter/A, which allows adapter-to-system-memory transfer of data independent of the host processor. The server's processor is available for other tasks.

Highlights:

- Extends life expectancy of server hardware by exploiting PS/2 busmaster capability.
- Reduces requirements for additional server hardware as applications or workstations are added.

Letter # 190-208, December 18, 1990.

Realtime Interface Co-Processor Selectable Interface Board/A and Related Features

The Realtime Interface Co-Processor Selectable Interface Board/A and related features provide additional electrical interface and cabling options to the announced and available Realtime Interface Co-Processor Portmaster™ Adapter/A and Multiport Adapter Model 2. The addition of the interface board provides electrical interfaces for CCITT V.35, CCITT X.21, RS-232-D, and RS-

422-A full-duplex communication capabilities.

Highlights:

- Provides a flexible, high-function communications co-processor when used with the Realtime Interface Co-processor Portmaster Adapter/A or Multiport Adapter Model 2
- Supports single, very high-speed (up to 2.04 Mbps on the Portmaster Adapter/A or 1.54 Mbps on the Multiport Adapter Model 2) full-duplex data rates
- Supports multiple, high speed (up to 64 Kbps) full-duplex data rates

Letter # 190-204, December 4, 1990.

Software

Writing to Write™ Form I

Writing to Write Form I is a unique and revolutionary product which enables second grade students to write what they can think. It is a balanced curriculum where the teacher and the courseware are equal instructional partners. The program design is instructional, so that the students actively participate and learn by doing, as opposed to "drill and practice." Writing to Write Form I encompasses the required stages of the "process writing" approach: prewriting, drafting, editing, revising, publishing and sharing. The spiral curriculum, the combination and coordination of computer execises, the ancillary print activities, and the teacher instruction, including the stages of process writing, makes Writing to Write Form I a courseware product which takes full advantages of technology. It will operate within a Teaching and learning with Computers (TLC) instructional framework.

Highlights:

- Enables the student to understand and become proficient in writing as a process, including pre-writing, drafting, peer review, revision, editing and publishing
- Assists the student in learning to write clearly and effectively
- Nurtures student creativity as the student translates thoughts into writing
- Nurtures student creativity as the student translates thoughts into writing
- Exposes students to traditional grammar through syntax
- Promotes student understanding of the relationship between writing and reading
- Enhances and builds higher order thinking skills
- Promotes the building of vocabulary skills
- Provides a spiral curriculum with an environmental approach that allows the learner to build upon the language knowledge that the student brings to school

Letter # 291-025, January 29, 1991.

Intel OS/2 and AIX i860 Software Development Tools for C and FORTRAN

IBM announced C and FORTRAN compiler support for the PS/2 Wizard Adapter running under both AIX PS/2 and OS/2 operating systems. The new software packages are vendor-logo programs that enable software developers to develop applications that take advantage of the numeric-intensive processing power of the Intel i860 OS/2 microprocessor on the PS/2 Wizard Adapter. The Intel i860 OS/2 and AIX Software Development Tools for FORTRAN are both native compilers that utilize the i860 to compile programs. Each package contains the required tools to enable development under OS/2 or AIX operating systems. The new FORTRAN tools replace the Intel i860 OS/2 Software Development Tools announced by IBM on April 3, 1990.

Highlights:

- User productivity gains are realized through a flexible, desktop solution for a broad range of applications that typically run on workstation or mainframe platforms.
- The customer's investment is protected when an upgrade path migrates from a general-purpose PS/2 system configuration to a technical computing platform.
- The application developer can extend the application base to include both C and FORTRAN applications, running under OS/2 or AIX.

Letter # 290-816, December 18, 1990.

PS/2 ImagePlus – Statement of Direction

IBM intends to provide a PS/2 LAN-based addition to the ImagePlus® family. PS/2 ImagePlus will be a complete end user configurable implementation of ImagePlus in a PS/2 LAN-based environment utilizing OS/2 Extended Edition.

PS/2 ImagePlus includes image storage and retrieval with document, folder, and case management. It will provide a comprehensive solution for small departments and work groups of organizations that can benefit from ImagePlus without having to rely on a host processor connection for image support.

PS/2 ImagePlus is being designed using IBM's SAA™ and CUA standards as implemented in the OS/2 EE Presentation Manager. It is being developed jointly by IBM and Eastman Kodak Company.

Letter # 290-796, December 18, 1990.

Index to Past Issues of *IBM Personal Systems Technical Solutions*

Issue 1, 1991 (G325-5010)
XGA – Raising Video Expectations
Choosing betw. Shielded and Unshielded Wiring for Data Transmission
Compatibility of LAN Servers and Requesters
Running DOS LAN Requester and Novell® Netware® Concurrently
Breaking the 640 KB DOS Memory Barrier
Understanding an OS/2 CONFIG.SYS File
OS/2 EE 1.2 Database Manager Performance
OS/2 EE 1.2 Competitive Performance
An Intelligent Front-End EASEL® Application
Enabling Software for National Language Support
SNA Definitions for 3270 Emulators
Diskette Failures Caused by Contamination

Issue 4, 1990 (G325-5009)
First Look at the New IBM PS/1 Computer
Using the IBM 4019 LaserPrinter Effectively
Micro Channel Interface Chip Sets
Token Ring Bus Master LAN Adapters
Extension of Wiring Rules for 4-Mbit/s Token Ring Using UTP Lobes
SCSI and DISK.386.SYS
Operating System Platforms: A Business Perspective
Minimum OS/2 1.2 DASD Requirements
User Profile Management
Understanding OS/2 1.2 LAN Server Performance
PM: An Object-Oriented Approach
DOS 4.00 SHARE
A "C" Programming Model for DOS Device Drivers
An Electronic Bulletin Board for PC Users

Issue 3, 1990 (G325-5007)
DOS – A Look under the Hood to See How It Spins
Memory Management in a DOS Environment
FASTOPEN – The DOS Performance Enhancer
DOS 4.00 Compatibility Issues
'Out of Environment Space' Errors
A New LAN Requester for DOS Systems
Creating a Dialog Box Dynamically Using WinCreateDlg
An Alternative for the OS/2 START Command
CUA: A Consistent Interface

Issue 2, 1990 (G325-5006)
OS/2 End User Advantages
What's New in OS/2 Standard Edition Version 1.2?
An Application Developer's View of OS/2
Object-Oriented Programming with C and OS/2 PM – Is It Possible?
Design Goals and Implementation of the New HPFS
OS/2 EE 1.2 Database Manager – Remote Data Services
OS/2 EE Database Manager Precompiler API
UNION, INTERSECT, EXCEPT
Writing a Database Manager COBOL/2 Program
Database Manager Programming with Procedures Language 2/REXX
APPC Performance Tips for OS/2 EE
EASEL OS/2 EE PROFS: Host Code Interface
PS/2 RPG II Application Platform and Toolkit
The IBM Independence Series Products

Issue 1, 1990 (G325-5005)
Introduction to Local Area Networks
IEEE 802.3 LAN Considerations
The IBM Token-Ring Network: A New Generation
How to Design and Build a 4-Mbit/s Token-Ring LAN
How to Design and Build a 16-Mbit/s Token-Ring LAN
Making the Cabling Decision
IBM Cabling System Highlights
Communication Strategy for Growth

Issue 4, 1989 (G325-5004)
Requirements for Advanced Bus Architecture
Micro Channel System Configuration Considerations
Features and Benefits
Bus Masters and Applications
Overview of Extended Micro Channel Functions
New Micro Channel Features
SCB – An Architecture for Micro Channel Bus Masters
Design Alternatives with Micro Channel Systems
The IBM PS/2 Micro Channel SCSI Adapters
PS/2 Wizard Adapter
Experience in Bus Master Design
Bus Master Adapters from Independent Option Vendors
Bus Masters and OS/2
Micro Channel Issues in AIX PS/2
Information for Developers
Book Report: "The Winn Rosch Hardware Bible"

“**Automatic SIMM configuration shields the user from complex set-up procedures.** (page 9)

“**In terms of IBM’s priorities, data integrity was at the top of the list in developing Micro Channel architecture.** (page 14)

“**Actual performance should exceed that of a comparable small LAN.** (page 23)

“**A single Presentation Manager call added to your application enables you to select any combination of a printer’s features for output.** (page 27)

“**This perspective will be used to explain how OS/2 and AIX compare and contrast with DOS.** (page 29)

“**There are two distinct types of COBOL compilers for use with PM.** (page 42)

“**OS/2 EE servers may accept requests from both OS/2 EE requesters and DOS Database Requesters concurrently.** (page 47)

“**With the release of OS/2 LAN Server 1.3, IBM increased performance, added new features and improved reliability.** (page 55)

“**Windows Connection has a full Dynamic Data Exchange interface to Windows.** (page 58)

“**THINKable is the latest addition to the Independence Series of products for people with disabilities.** (page 75)