Patricia Seybold's

VOL. 4, NO. 9
ISSN: 0887-3054
SEPTEMBER 1989

INSIDE

EDITORIAL

Sanctified Standards ....Page 2
Should we be forced to overlook superb technology simply because it isn't already part of an accepted standard? Of course not.

NEWS ANALYSIS

Uniplex outshines the rest of the show at UniForum in Boston by introducing an X-Window version of Uniplex II Plus and new DOS and A/UX versions of its products • Nixdorf provides image processing powers to its Targon Office architecture with DCPA Image • Relational Technology ports its Ingres RDBMS to Intel's 80486 architecture • Hummingbird Technologies introduces HCL-eXceed: an implementation of the X-Window Release 11.3 Server which runs on AT or more powerful PCs ....Page 15

UNIX IN THE OFFICE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

The Double Standard

DOS under Unix Moves to the Next Generation

By David S. Marshak

WE ARE ENTERING a major period of transition, one in which we will be moving from a single dominating operating system on the desktop—DOS—to a multitude of desktop systems—DOS, Macintosh, OS/2, and Unix. As these last two more powerful multitasking systems vie for leadership in market share and in innovative *(continued on page 3)*

• E D I T O R I A L •

Don't Worry, Be Happy

By Judith S. Hurwitz

I WAS SPEAKING with an MIS director the other day who is cautiously thinking that he might like to implement a Unix-based system in his department. His overriding concern was that there seemed to be too much confusion over which version of Unix will prevail. My answer to him was, "Don't worry, be happy."

Why should a wary, conservative MIS director be happy about the rivalry within the Unix market? It has something to do with the importance of innovation and the inevitability of change. If Unix is to become a widely used and accepted commercial operating system, it needs to evolve. It needs to incorporate all of the commercial characteristics that MIS organizations take for granted in IBM and Digital proprietary operating systems—things like security and system administration. Unix must continue to evolve from its roots as a 20-year-old operating system into a modern operating system. In a world where there is one clear Unix winner, what incentive would there be for innovation? The competition between the Open Software Foundation (OSF) and Unix International (UI) for the hearts and minds of computer application developers and users ensures innovation. For example, if OSF adds a sophisticated new file system to its operating system, UI has to do the same. If UI adds multiprocessing to System V, OSF will do the same for OSF/1.

The same argument can be made for the user interface "wars." If the industry converges around Motif as the one and only standard user interface, we may be stuck with mediocrity. Motif has some good foundation technology, but lots more is needed in the user interface arena. There are important and innovative aspects of UI's Open Look that meet the needs of users. And then there is NextStep. This object-oriented interface builder offers capabilities and tools not available in either Motif or Open Look. Should we be forced to overlook superb technology just because it isn't already part of the sanctioned standard?

The picture we are painting of diversity and competition is not neat and clean, and this will make many traditional MIS managers uncomfortable. In the traditional mainframe environment, nothing is left to chance. MIS management knows years in advance what to expect from future versions of operating systems. It plans out precisely what it will implement at which points in its organizations. Nothing is a gamble. Nothing is unexpected.

But the world is changing. As the demand for easy access to vital corporate data begins to accelerate, we are moving beyond the comfortable traditional MIS environment. MIS will have to continue to change its approach to the processing of information to match users' needs and expectations. Object-oriented applications and management of those objects are the new frontier that these managers will be forced to explore. Without innovation, we will be locked into the static information systems of today. Therefore, when a company like IBM announces that it will provide NextStep in addition to Motif, it is a rational action. When Steven Jobs offers Mach as the operating system for his new workstations, he is reacting to the need to build on today's achievements, move ahead to the next generation, and never stop seeking higher function.

Both MIS managers and software developers should applaud competition and innovation. It is a good sign for our industry when vendors try to create something better. The only way that the computer industry will regain its lost momentum is by stimulating the imaginations of users about what is possible. Users will begin to buy more powerful systems when they see innovation that can be translated into better management of information and tasks.

So the next time a vendor announces something new and different, the user community should react with excitement and hope. Don't worry, be happy. ◐

Patricia Seybold's UNIX in the Office (ISSN 0887-3054) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices. POSTMASTER: Send address changes to *Patricia Seybold's UNIX in the Office*, 148 State Street, Suite 612, Boston, MA 02109.

•DOS UNDER UNIX•

(continued from page 1) applications, users will still want access to the DOS environment. As they migrate, users cannot afford to be cut off from the popular DOS programs which have not yet been ported to or rewritten for new platforms, not to mention their favorite, homegrown DOS applications.

Parallel with this transition is a movement in the way access is provided. Emulation technology is quickly evolving from the traditional hardware coprocessor, to use of the Virtual 8086 mode on the Intel 386, to the newest generation of software coprocessors based on high-performance workstations.

In this article, we will examine the current methods of providing DOS access, their strengths and weaknesses, and where they are going. We will then take a peek into the future, examining the role that emulation technology will take in the next-generation distributed network computing environment.

Approaches to DOS Emulation

DOS emulation technologies fall into three specific categories:

- Hardware coprocessors
- PC emulation using the Intel 286/386 chips
- Software coprocessors

Each of the technologies attempts to accomplish the same essential task: to create an environment that looks and behaves exactly like a PC running DOS. The look and behavior of the environment must be indistinguishable both from the user and from the DOS application.

As we shall see, there are a number of elements that emulation technologies must address in simulating a DOS PC. These include the Intel 8086 processor and its instruction set, the other supporting chips that make up a PC, the BIOS (Basic Input Output System) that allows communication with peripherals, and DOS and its file system.

These can be handled in hardware, software, or a combination of the two. As we examine each of the technologies, we will see that each level conceptually, as well as historically, follows the preceding technology, as each level operates at a higher level of abstraction, i.e., at a level further removed from the hardware.

Hardware Coprocessors

Hardware coprocessing is nothing new. In the pre-PC era, one could purchase a Z-80 coprocessor for an Apple II in order to run "industry standard" CP/M. And CP/M on a Z-80 coprocessor was a popular option for the early IBM PC. With the dominance of DOS and the PC, Intel-based coprocessors were developed for host- and workstation-based non-Intel systems, as well as some kludged products such as the infamous Wang PC.

Current hardware coprocessing solutions, such as that offered by Phoenix Technologies (Norwood, Massachusetts), include all the facilities required to run the PC environment concurrently with or as a task under the host operating system (see Illustration 1). The Phoenix coprocessor also integrates the DOS file system with the host's operating system. Under Unix, for example, all PC software and data structures are translated by Phoenix's emulation software into Unix file formats, allowing file-sharing among users of the different systems.

The hardware coprocessor recreates the PC environment mainly in the hardware itself, with both the CPU and the associated chips found on the board.

The advantage of hardware coprocessing is that all CPU and associated chip operations run at full processor speed. The downside of hardware coprocessing includes the cost—two CPUs instead of one—and the inflexibility of having to install a specific piece of hardware.

The hardware coprocessor solution is still available, though it tends to be relegated to niche situations and little new work is being done in the field. The current hardware coprocessors (for example, those shipped by Apollo), based on 286- or 386-level chips, are specifically aimed at ultrahigh-performance needs.

The future of the hardware coprocessing emulation option is dim, as the technology is already on the decline and will become less important as software simulation becomes more advanced. (However, other types of hardware coprocessing, particularly those based on an intelligent bus technology such as the BusMaster technology of IBM's MicroChannel Architecture or MCA, hold great promise for high-performance workstations. An example here might be a PS/2 with an i486 processor with Intel's i860 RISC chip as a graphics coprocessor.)

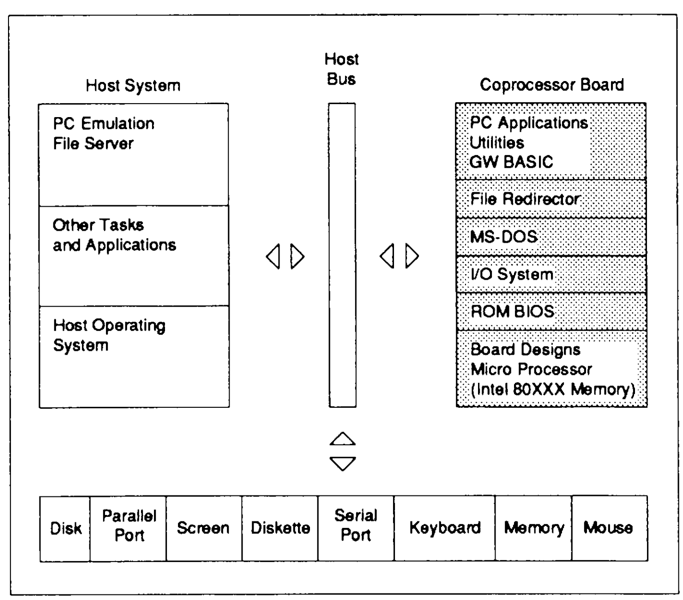


Illustration 1. Phoenix Coprocessor Board

PC Emulation: Building on Intel's Built-In Emulation Capabilities

Conceptually, the next step up from a separate hardware processor is having the system processor itself do the emulation. This was made possible with Intel's introduction of the 80286, and made feasible in terms of performance with the 80386. (To avoid confusion, we will refer to the 286/386/486... line as the 386, unless a distinction is necessary.)

The 386 32-bit processor can run a 32-bit multitasking operating system such as Unix. The resulting machine can be used either as a host for a small multiuser system or as a powerful single-user workstation. In a very short time, it has become the most popular platform for delivering Unix systems, having captured, according to IDC, 50 percent of the Unix market last year.

With any emulation technology, be it hardware, virtual, or software, the emulation of the CPU instruction set is only the beginning.

The 386 also has one capability not found in any non-Intel processor: It can run one or more Virtual 8086 modes while running other applications in its own Protected mode. And this can all be under the control of the native operating system. In other words, the instructions of a DOS application can be executed directly on the hardware, without any translation. This is done (if it is the only task running) at the full processor speed. And, since the native operating system is itself multitasking, the DOS application can run as one of many concurrent Unix tasks.

This points out one of the advantages that Unix has vis-à-vis OS/2 in providing DOS emulation. Since OS/2 runs in the Protected mode of the 286 instruction set rather than the 386, it can only use the 8086 Real mode to run DOS software. Since the Real mode does not have the Virtual mode's ability to isolate each environment, the 286 (and any operating system built upon it) is limited to a single DOS task at a time—the compatibility box. Unix, on the other hand, makes full use of the 386's Protected mode, permitting multiple Virtual 8086s simultaneously, each with its own virtual environment. The next generation of OS/2 (OS/3?, OS/386?) should also have this ability.

This is not to say that running DOS under a 386 version of Unix is automatic or even easy. With any emulation technology, be it hardware, virtual, or software, the emulation of the CPU instruction set is only the beginning. The resulting PC environment must include (or appear to include) a complete subset of PC hardware, including all I/O subsystems, as well as the system software.

Hardware systems that must be addressed, in addition to the microprocessor, include:

- Video controller
- Keyboard
- Interrupt controller
- Hard disk file system
- Mouse
- Serial and parallel ports controller
- RT/CMOS
- Floppy diskette controller
- Memory
- DMA controller

In terms of system software, the PC environment must contain implementations of MS-DOS and the ROM BIOS (Read-Only Memory Basic Input Output System). The ROM BIOS is what determines a system's ability to support various PC-based peripherals, hardware components, value-added features, and software applications that write directly to the hardware. In order to accomplish all of this, the 386 machine requires an emulation facility, whose basic role is to make the DOS OS and application think that they are running on a dedicated machine.

In today's Unix 386 market, virtually all systems come with or have as options one of two emulation facilities: VP/ix, codeveloped by Phoenix Technologies and Interactive (Santa Monica, California), or Merge 386 from Locus Computing Corporation (Inglewood, California).

VP/ix AND MERGE 386. Although a great debate is currently raging on which of these products does a better, more flexible, or easier-to-use job of providing DOS access under Unix, the two resemble each other much more than they differ. The products share the same approach, basically provide the same capabilities, and, in those areas where one may have a current advantage (such as Merge 386's X-Window support or VP/ix's SunRiver workstation support), the other is, as we shall see, moving rapidly to catch up.

Both VP/ix and Merge 386 provide DOS environments that look, feel, and act like any AT. Both can run all 8088/86 applications and all applications that run in the Real mode of the 286. The only applications that cannot be run are those that address the 286's Virtual mode: Desqview, Windows 386, and OS/2 (Windows 2.0 runs just fine, thank you). In order to run these, as well as to create an emulated OS/2 environment, this technology will have to wait for Intel to provide 286 or (even better) 386 Protected mode emulation in the 386/486 line sometime in the future. We hear that Intel is working on this, but we suspect that it is not high on the company's agenda. Thus, we may not see this capability in the 586 or even the 686.

Most important of all, both VP/ix and Merge 386 address the four key issues of this type of technology: emulation, access, integration, and performance.

DOS Emulation. VP/ix and Merge 386 both run as tasks within the Unix environment. The user must log in and run DOS as he or she would run any other application (this, of course, can be part of a script or the user's startup profile). When the user types the command to begin DOS, it is as if a PC had just been issued a warm boot. The local config.sys and autoexec.bat files are read, and the user is presented with a C:> prompt (Z:> with VP/ix). The user then has access to what appears to be a complete PC, including, if so configured, expanded and extended memory.

The user can then run any DOS application from either a directory on the hard drive or from a floppy, if the system is configured to allow floppy access. (Note that this is still a Unix system, and therefore requires a power user or an administrator to do the optimal configurations.)

DOS users will see very little difference in using the DOS application under VP/ix or Merge 386, with performance and look-and-feel quite acceptable.

While this application is running, the user can hot-key to another login prompt (or an already active session), and run another DOS session or a Unix application. This can be done for as many active Unix sessions as the host will support.

Running multiple DOS sessions will impact performance in the same way that running any other Unix tasks simultaneously does. Two tasks will cut performance to about half, three to about one-third, etc. (This is a difficult concept for some DOS users.) In our experience, performance can suffer significantly after three or more simultaneous DOS sessions are launched, particularly in some of the emulated areas such as keyboard or mouse activity.

DOS Access. The emulation discussion in the previous section refers specifically to a 386 Unix workstation, where DOS is being run from the console (i.e., the workstation's own keyboard and monitor). This is basically a single-user environment where the user has access to both Unix and DOS sessions.

When we move to multiuser access, things change quite a bit. Running a DOS session from a terminal is far more removed from the traditional PC DOS environment than running DOS from the console. This has a lot to do with the nature of the terminal used, as well as the performance degradation one experiences with asynchronous communication.

Both VP/ix and Merge 386 provide multiuser DOS access on the 386 platform, and, in fact, one of the strong markets for them is the small workgroup with a 386 and a number of terminals "running" DOS. The types of devices supported range from simple ASCII terminals to high-performance specialty units linked by fiber optic technology, such as those produced by SunRiver.

The standard ASCII terminal solution specifically addresses the low-cost scenario where a number of terminal users can share a single CPU and, frequently, a single copy of the application. While cost is on the side of this solution, its implementation has two specific drawbacks. The first is that the DOS application will run in a character-only mode, or the IBM MDA (Monochrome Display Adapter) mode. This could come as quite a shock to users who have become comfortable with CGA (color graphics adapter) or EGA (extended graphics adapter), as well as preventing access to those applications (MS Windows, etc.) that require graphics.

The second problem is that standard terminals do not map well to PCs, generally supporting a 24-line display—as opposed to the PC's 25 lines—and not providing AT-style keyboards, which necessitates relearning or reassigning keys as well as some octave-plus reaches. A middle ground is provided by some terminals, known as PC Scancode terminals, that have been specifically developed for use with DOS applications. These support the 25 lines and the AT keyboard, but do not support any additional graphics capabilities.

On the high end, there are some terminals, most notably from SunRiver, which permit multiple users to access graphics-based DOS applications on the same host. Priced at around $2,000, these terminals (a standard PC keyboard and EGA or VGA monitor) connect via fiber optic cable to a board that sits in the host machine. This board provides both the fiber-based connectivity and the video emulation, making multiuser performance acceptable. Currently, the SunRiver technology is supported by VP/ix only (providing Phoenix one area of differentiation), although Locus has announced that similar support will be available in the next release of Merge 386 (Release 2.0), due for initial shipment next month.

X-Window Access. One more form of access will become increasingly important as DOS implementations under Unix move from small, multiuser configurations to integrated networked environments. This is the ability for an X-Window workstation (or X terminal) to open one or more DOS windows within the X environment. In a networked environment, the X clients (remember that, in X terminology, the client is the application often running on the host, and the server is the user's display station) could run any machine on the network

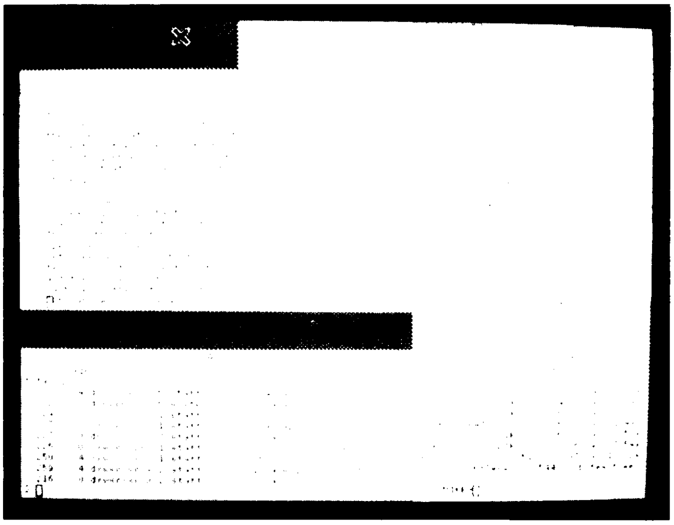


Photo 1. An X-Window display on an IBM PS/2 Model 80 running AIX and Merge 386. The top two windows are running a DOS word processor and a DOS spreadsheet. The bottom two windows contain the respective directory listing commands in Unix and DOS for the same subdirectory. At the extreme upper right is the X-Window control menu.

(see Photo 1). Thus, a single 386 machine (or more, if usage warrants it) could be a DOS server for multiple Unix or X terminal users.

Here is one important point of differentiation between Merge 386 and VP/ix: Merge provides X-Window client support, while VP/ix currently does not. This means that, when the Merge 386 window is opened, it can display full monochrome or CGA graphics; with VP/ix, the window can only look like an ASCII terminal. Phoenix plans to introduce X client support within VP/ix by the end of the year

We will further examine the relationship between DOS emulation technology and X-Window as we discuss the future role of DOS emulation in the distributed network computing model.

DOS/Unix Integration. In today's market, it is not sufficient to simply provide DOS emulation under Unix. Users of the different operating systems need a way to easily move from one environment to the other, as well as needing a way to share files and resources across operating systems, particularly in a networked environment.

With current Intel-based emulation, as well as with the software coprocessing technology, high priority is placed on integrating the emulated DOS environment with the host system. This integration occurs on a number of levels: file systems, command and application access, shared peripherals, and application-to-application communications.

File system integration involves three areas: physical access, file naming, and file translation. Physical access to the hard disk is provided to both the DOS and Unix environments by mapping the DOS system on top of the existing Unix system. Thus, DOS users have access to the existing Unix directory structure (given the proper permissions), while directories and subdirectories created within DOS (using the DOS MkDir command) are actually created as Unix subdirectories, with both systems having access to the files within them.

Translating filenames between systems can create problems. Since DOS filenames are limited to eight characters with three character extensions, longer Unix filenames must be modified to be viewed from DOS. Each system handles this slightly differently, but both make it clear to the user that the modification has taken place. File-naming, from the point of view of the Unix system, is not a problem, as any DOS filename can be given the same Unix filename.

VP/ix and Merge 386 both permit users to run DOS commands from the Unix environment, and Unix commands from the DOS prompt (though, in the case of Merge 386, these can only be noninteractive applications). Thus, a DOS user can run either DIR or LS and see essentially the same results, albeit in the format of the system from which the command comes. Where identical Unix and DOS commands exist, and where they do not take the identical actions, the application from the system where the user is currently residing will take precedence, unless the user specifically designates that it comes from the other system. This designation varies between the products. Both products provide utilities to translate Unix files to DOS and vice versa.

One of the most powerful features of these systems is their ability to allow applications from the different environments to "talk to each other." This is done via the Unix Pipes mechanism. The results of one application—a sort or a query, for example—can be piped as input for another application. Under VP/ix and Merge 386, the DOS and Unix applications can be piped to each other interchangeably and transparently.

Performance = Video Performance. Performance is the bugaboo of emulation technology. Users are not willing to run emulated DOS applications if performance does not at least approach that of the common PC platforms. This means that the original XT-like speeds must now be upped to near-AT speeds.

For the virtual 8086 emulations, the processor speed is not a question. As noted above, a single DOS application will appear to run at the native processor's speed. Rather, the great-

Phoenix and the Ashes

ONE OF THE companies most prominent in the DOS emulation arena, Phoenix Technologies (Norwood, Massachusetts), is currently undergoing a difficult time. Best known for its BIOS, which emulates the IBM PC and enabled the whole "clone" industry, Phoenix has recently suffered through a management change, a losing quarter, significant personnel layoffs, and a suit brought against it by some of its stockholders.

Speculation about the cause of these lean times has abounded both publicly and privately. The reasons given have ranged from a falloff in the PC clone business to the company's commitment to IBM's MicroChannel architecture. And, as we have noted, the public relations momentum in DOS emulation technologies has seemingly swung to others, most notably with SCO's Open Desktop selection.

In our discussions with Phoenix, company spokespersons have told us that the past quarter's poor showing is directly attributable to delays on some large contracts for providing Unix and RISC architectures to computer manufacturers. The company is, in fact, currently rethinking its role in this area of the Unix market, and may withdraw from it.

Phoenix's recent woes do little to change our evaluation of the company's technology and products. We do fear, however, that Phoenix's aggressive moves into the next generation of emulation technologies may be slowed or stymied by a lack of resources.

That, if it comes to pass, would be a shame, for the company has much more to offer than PC clone parts.

Video Performance And X-Window

IN ORDER TO understand why achieving good video performance via X-Window is so difficult, particularly for EGA (extended graphics adapter) and VGA (Video Graphic Array), we must take a closer look at how video emulation is actually accomplished.

CGA (color graphics adapter) and Hercules emulation is accomplished by creating two "shadow" video memories in RAM. Each of these buffers is an image of what appears on the screen. The first is IBM compatible, and it appears to DOS as video memory. DOS applications can write directly to it, eliminating the necessity for the emulation software to intercept every DOS instruction and determine whether it affects the screen.

Periodically, the emulation software compares this shadow to the other shadow, which is essentially the previous version of what was on the screen. When changes are detected, the emulation software first writes the changes to the second shadow screen, then translates this to the host system-compatible video, and, finally, uses the host system's windowing system (which itself may not be that speedy) to output it to the screen.

Since EGA and VGA emulations require much more information to be moved, they will naturally run slower. But this is only part of the story. The shadow memory technique employed for CGA and Hercules that allows DOS to write directly to a screen image will not work for EGA and VGA. This is because EGA and VGA are three-dimensional, not like the two-dimensional CGA and Hercules implementations, which can be mapped to a specific place in RAM.

EGA and VGA are planar, operating on four distinct planes in addition to the screen map. This allows for very dense pixel representations and a whole lot of colors, but makes emulation very difficult. It forces the emulator to revert to trapping each instruction to examine whether the particular instruction affects the screen, and, generally, to implement an instruction-by-instruction translation and updating of the windowing system. Currently, with the Intel-based emulation products, this process is just too slow to make it usable. It is unlikely that adequate performance will be possible without some customized hardware assistance at the display-station level.

There is more hope of supporting EGA and VGA within X-Window with the software coprocessors. First, the very high speeds of some of these processors can alleviate some of the video performance issues, with adequate performance being obtained from Insignia Solutions' SoftPC, for example, running on Hewlett-Packard precision architecture and 68040. Second, the optimizing techniques being employed by the current generation of software emulators should go a long way toward "batching" the video instructions, reducing the problems of the instruction-by-instruction inspection, translation, and implementation.

---

est problem lies in emulating the video output of the PC. This emulation takes both a portion of the processor's power and a great amount of memory. As the emulated resolution increases, performance decreases.

Both VP/ix and Merge 386 currently support monochrome, Hercules, CGA, EGA, and VGA (for Merge 386, VGA is with the next release). On the console, performance on graphics up to CGA is fine, with acceptable EGA and VGA performance now being reached.

However, the video performance issues are compounded when running the emulation software under X-Window. The additional amount of information to be moved, coupled with the additional instructions required by EGA and VGA emulations, make adequate X-Window performance above CGA very difficult (see "Video Performance and X-Window," above).

We are just beginning to see some attempts to use high-end processors (RISC and 68040) and software coprocessing emulators (see "Software Coprocessors: The Next Generation" below) to support EGA and VGA in X-Window, but this is not currently possible (in a usability sense) on the Intel platform. Rather, the companies who do their emulation on the Intel platform are working on forms of hardware assistance at the display station itself. Currently, the research is into customized chips which would both emulate EGA or VGA and do the translation-to-host format.

VP/ix VS. MERGE 386. The choice between VP/ix and Merge 386 is not really left up to the end user. Both products are OEM-ed through hardware and systems vendors, and the user gets whichever product runs on the system he or she purchases. For example, VP/ix is included with Sun's 386i, while Merge 386 is available on the IBM PS/2 Model 80 running AIX.

This is not to deny the real battle between the products. Each is trying to secure its position with its own OEMs while hoping to woo away some of the customers of the other. Thus far, the tide seems to favor Locus, with the recent announcement of Merge 386 as the standard DOS emulation technology for SCO's Open Desktop.

This announcement hits Phoenix on two fronts: First, Open Desktop promises to be an exciting and potentially very lucrative offering for those companies involved. Second, SCO was one of the largest, if not the largest, resellers of VP/ix, having included it in its Xenix offerings as soon as VP/ix was first

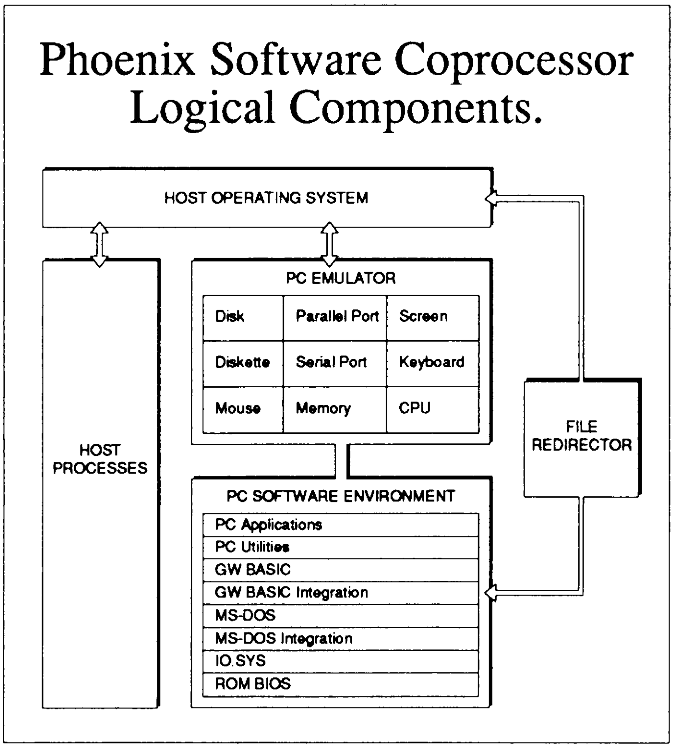


Illustration 2.

available. According to John Harker, SCO manager of Networking Systems Product Marketing, the decision to go with Merge 386 for Open Desktop was not related to the relative merits of the two companies or products. Rather, it was made because SCO had already settled on Locus's PC XSight Server (which allows DOS PCs to act as X terminals) as the DOS X-Window implementation for Open Desktop.

At the same time that it seems to be on the defensive with its VP/ix product, Phoenix has turned more attention to its software coprocessing technology, with which the company believes it can reach a broader market.

Software Coprocessors: The Next Generation

As more and more powerful desktop workstations become the norm, they will tend to be based on two types of processors: 386/486 and RISC (Reduced Instruction Set Computers). While the Virtual 8086-based emulation solution works fine for the 386 platform, the need for a software-only DOS emulation, a need which has already existed on the Motorola 68020/30 platform, will greatly increase.

Because of this need, and specifically because of the industry's current focus on RISC, much attention is being given to bringing out a new generation of software-based emulation products.

The great advantage of software coprocessing is its flexibility. It is not bound to a single processor architecture. Once running on one architecture, it can be ported, with the ease (or difficulty) of any other application, to other architectures.

The great disadvantage is speed, and the reason for this is quite simple. In the hardware coprocessor or virtual 8086 emulation, the instruction set runs directly on the chip. With a software coprocessor, in its simplest form, each instruction has to be read, its native counterpart located, and this instruction (or set of instructions) is then executed. In real life, this is more complex, and performance can typically take a 5 to 1 or even 10 to 1 hit.

Software coprocessors can have an advantage in video performance, particularly EGA and VGA. This is due to their ability to optimize the DOS video calls (see "Video Performance and X-Window," page 7).

Other than the CPU instruction set performance issue, software coprocessing offers the same abilities—emulation, access, and integration—that are found in the Intel-based solution. With the two current products—SoftPC from Insignia Solutions and the Software Co-processor from Phoenix Technologies—the 80286 in Real mode is emulated (see Illustration 2). There is no reason why the 286 or 386 Protected mode could not be emulated, thus permitting OS/2 emulation under Unix. For both of these companies, the issue is lack of market demand rather than the technical challenge.

The development of software coprocessing technology can be looked at as having three generations, the first being a straight instruction-by-instruction translation, the second bringing more power and certain optimization techniques, and the third bringing an architecture that assures portability of both the system to be emulated and the system on which the emulation is to be run.

FIRST GENERATION: INTERPRETATION. The first level of emulation is termed "static CPU" by the British company Insignia Solutions, one of the pioneers in software emulation. It is essentially an exercise in interpretation, translating the instruction set from one processor to another, though performance can be somewhat enhanced by techniques such as caching and code buffering. This generation can, on 68020-class machines, emulate DOS at approximately XT speeds.

The first viable DOS software coprocessor of this class was introduced by Insignia Solutions for the Motorola 68000 family, first the SoftPC for the Sun3 in 1987 and then for the Macintosh in early 1988. These coprocessors are sold basically as a retail, end-user, or reseller add-on to the native OS. Insignia also produced some "first generation" products for the OEM market.

The first-generation products are limited both by their instruction-by-instruction approach and by the relatively low power and functionality of the processors and systems (i.e., lack of virtual memory support) on which they run.

SECOND GENERATION: MORE POWER, MORE SPEED, MORE ARCHITECTURE. The second generation of software-based emulation products is marked by two distinct trends: (1)

the use of faster, more powerful processors, and (2) a re-architecting of the concepts of emulation technology.

In one sense, the second generation is simply a function of the availability of faster and faster chips. When one has access to a 10 to 20 MIPS machine, a 10 to 1 performance loss does not look all that slow, when compared to the PC's native environment. With today's fast processors, AT-like speeds are consistently being reached.

Parallel to the increase in processing power, the second generation has seen the development of new intelligent simulation techniques aimed at speeding up and enhancing the emulation process. The enhanced emulator, called "Dynamic CPU" by Insignia, is able to optimize the code by "learning" what the most frequently used instructions are, providing a more direct route to the instructions, and caching the most heavily used routines in fast main memory. (For a look at how this is done, see "Anatomy of a Software Simulator," page 10.)

The goal of the second-generation architecture is to move as far as possible from an interpretive to a compiled process. This is done by implementing some expert system-based learning techniques. The implementation is layered (see "Anatomy of a Software Simulator"), so that each level uses information gleaned from the level below it in order to eliminate a stage in the instruction processing. Learning is done each time a section of the code is executed. The more frequently a routine is executed, the more it can be optimized. This is particularly nice, since it is precisely the most frequently executed routines that one would like to see run faster.

With the Phoenix system described in "Anatomy of a Software Simulator," learning, once it has occurred, is saved for the next execution of that particular application. Insignia's product uses a similar simulation learning technique. However, the learning is not saved from one session to another, though this feature, according to Insignia, could be easily added to the product if an OEM requested it.

The second-generation architectures do have some limitations. First, since the expert systems rely on optimizing code that is repeated exactly, they cannot deal at an optimized level with code that modifies itself. Second, this architecture is particularly effective with sections of code that are repeated frequently. Thus, it tends to benchmark very well, as most benchmarks involve multiple iterations of processes, though it does not necessarily give an equivalent performance boost to the everyday user. One of the goals of future generations of simulation technology is to be able to optimize all, not just recurring, instructions. This is an area where a lot of work is currently taking place.

CURRENT SOFTWARE COPROCESSORS. The two companies OEM-ing software coprocessors, Insignia and Phoenix, have both taken similar approaches in implementing second-generation products. Both products provide the full set of emulation, access, file integration, and integration with the native windowing system. As with the Intel-based emulation, the battle between them is being fought over OEMs, rather than for the end users.

THIRD GENERATION: PORTABLE SOFTWARE, PORTABLE EMULATION ARCHITECTURE. We are beginning to see the appearance of some of the third generation's features, and portability is one of its hallmarks. Specifically, Phoenix is implementing a fourth level of its simulation model (see "Anatomy of a Software Simulator"). This level is noninteractive. Rather, it acts as a compiler that creates the most optimized version of an application and compiles it to run directly as a host processor task.

This capability will be useful in two areas. First, it will allow end users to create optimized software versions for themselves. Second, the technology has potential as a quasi-porting tool for software developers. This would enable developers to sell optimized versions of their existing software to run on the new platforms, rather than writing or recompiling them, with the advantages of lower cost and quicker time to market. But it would also have the disadvantage of not being able to make the most of the capabilities of the new platform. In other words, a highly optimized emulated DOS version of 1-2-3 running under Unix is still a DOS version.

Phoenix has taken portability one step further. The company has architected its simulation technology so that it is portable both as to the target and the source platforms (see Illustration 3). The process is built around Phoenix's Translation Engine Assembly (TEA), which is a compiler that takes the 286 instruction set and generates the code for the target platform. Additional target modules can be written without changing the TEA, providing portability of the 286 instruction set across platforms.

The architecture is modular, with the source instruction set and the targets not having to know about each other. Therefore, other instruction sets could be substituted for the 286 instruction set without having to modify either the target modules or the TEA itself. For Phoenix, this architecture not only has more flexibility than a traditional coding and porting routine, but it also has produced better performing simulators on each platform. Phoenix plans to offer the Translation Engine Assembly and its architecture to standards bodies for certification as a standard way to provide portability of simulations.

(continued on page 12)

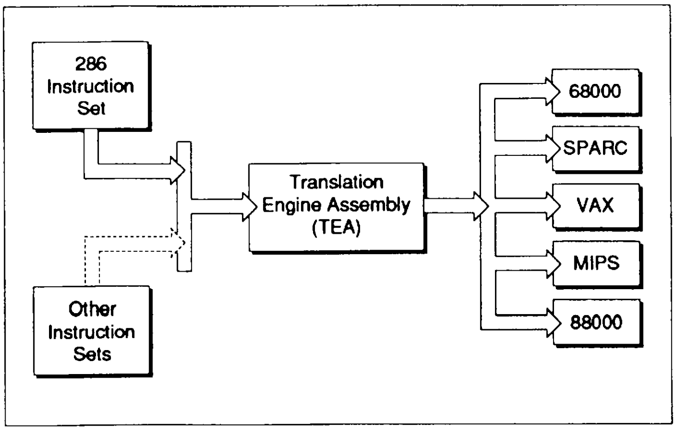


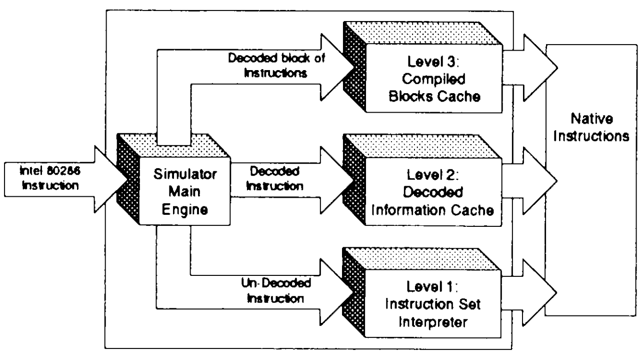
Illustration 3. Phoenix portable emulation architecture.

Anatomy of a Software Simulator

IN ORDER TO understand how new intelligent simulation techniques that enable learning by the enhanced emulator can actually reduce instructions and increase performance, we can look at the implementation of the Phoenix Software Co-Processor simulation technology.

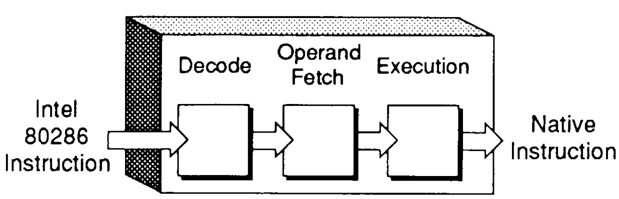
The Phoenix simulator supports three levels of instruction processing:

- Level 1: Instruction Set Interpreter
- Level 2: Decoded Information Cache
- Level 3: Compiled Blocks Cache



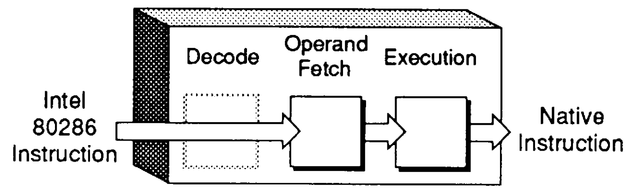
Instruction-processing levels of the Phoenix Simulator.

At Level 1, the Instruction Set Interpreter looks at the program counter, takes the instruction, decodes it, goes to a table to find the correct host instruction(s), then executes the instruction(s). Next, it goes back to the program counter, moves to the next instruction, and so on.



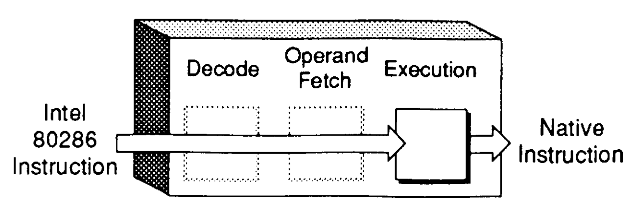
Level 1: Instruction Set Interpreter.

At Level 2, the results of the Level 1 processing are cached in a code buffer for later use. The number of the program counter is retained so the instruction can be accessed again without decoding. Before interpreting any PC instruction, the simulator checks whether or not the decoded information exists in the cache. If it does, the simulator can immediately execute the instructions. If not, the simulator reverts to Level 1 interpretation. At Level 2, the most frequently used instructions can be cached in the fastest memory.



Level 2: Decoded Information Cache.

At Level 3, the simulator is able to deal with blocks of instructions rather than one instruction at a time. An internal mechanism converts blocks of frequently used instructions into the blocks of host instructions that most efficiently implement them. These instruction blocks are then cached in a buffer for future use. The simulator dynamically determines if a particular set of instructions is suitable for the Compiled Blocks Cache by incrementing a counter each time a particular block of instructions is decoded. When the instructions have been executed a pre-specified number of times, the simulator automatically invokes a built-in optimizing compiler to create and cache the corresponding set of host instructions. Thus, when the beginning of the set is encountered, the whole routine can be executed directly, without any decoding.



Level 3: Compiled Blocks Cache.

This environment contains safeguards and fallbacks. If the program is self-modifying, i.e. the instruction that the program counter was pointed at has changed, the simulator reverts to the next lower level where it is able to accurately execute the instructions. It then relearns until it can get to Level 3 again.

XDOS: Converting DOS Programs to Unix

WHILE THIS ARTICLE focuses on the ability of Unix systems to provide access to DOS applications through emulation technologies, there is a more direct approach to providing popular PC products to Unix users. This, of course, is to actually port the product to the Unix environment, an approach that can run at the full processor speed and take advantage of full Unix functionality.

Major PC software vendors have committed to porting or rewriting their software for Unix, and a number of key PC applications—WordPerfect and Microsoft Word, for example—are now available on many Unix platforms. However, in general, moving to Unix has been a slow and difficult process for most PC software developers, many of whom are familiar only with the limited functionality allowed by DOS and find the capabilities provided by a multitasking, 32-bit operating system somewhat overwhelming. This, the lack of resources of some of the developers to put forth a strong Unix development effort, and the confusion over which Unix versions and which Unix platforms to develop first, has severely slowed down any mass movement of DOS applications to Unix.

THE XDOS CONVERSION TOOL. Hunter Systems (Mountain View, California) is now delivering XDOS, a tool that converts executable DOS applications into native executable Unix applications for whatever Unix-based system users want to run them on.

An XDOS conversion has two stages: porting and installation. Porting is done only once for each DOS application and is performed by Hunter Systems or another software developer. Installation of a DOS application on a target system running XDOS is performed only once by the end user.

XDOS converts 8086 instructions into the native instructions for 68020/30, 80386, and RISC processors, converts calls to DOS into calls to Unix, and converts single-user programs into reentrant, multiuser programs. The result is executable code that can then be used by single or multiple Unix users.

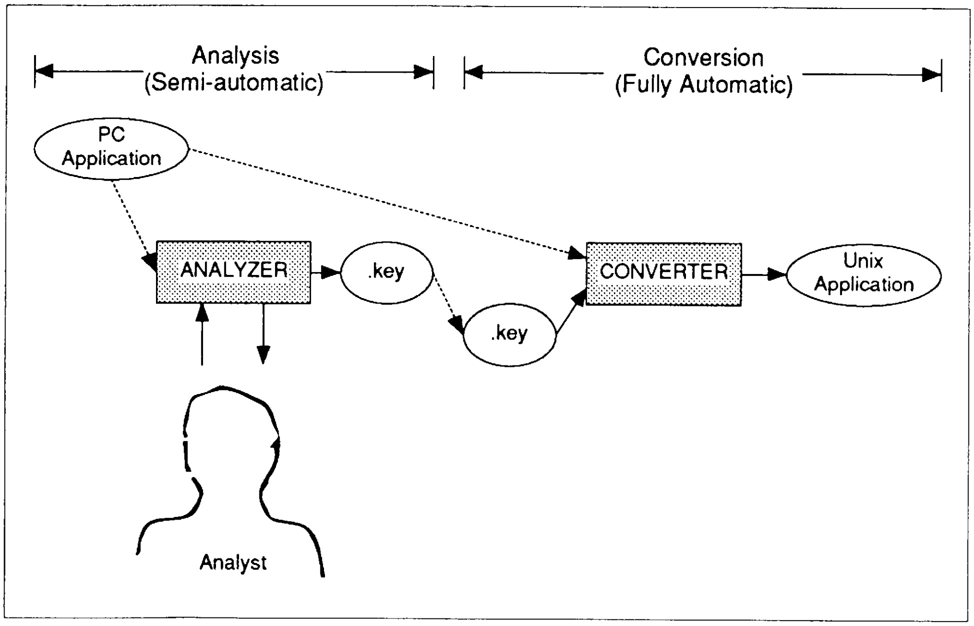
XDOS conversion is built on a technology developed by Hunter called Binary Compilation. Binary Compilation is a combination of an optimized analysis and conversion compiler and a human "expert" who deals with such anomalies as self-modifying code and complex computed jumps. The human expert's analysis combined with compiler-optimized analysis goes into a Key file for the application, and is then usable for any conversion of that specific application. (See illustration below.)

Because a Key file is required for each application to be converted, XDOS's usefulness for end users is currently limited to those DOS applications for which Hunter provides Key files. These include:

- Brief
- dBase III+
- Lotus 1-2-3
- Microsoft Word
- MultiMate Advantage II
- RBASE
- WordPerfect
- WordStar Professional

Hunter also offers a developer's version, which enables software or system developers to do their own analysis and Key files.

XDOS is available for the ARIX 800 Series and System 90, Honeywell-Bull XPS-100, Motorola Delta Series, NCR Tower, Sony NEWS, Sun3 and 386i, SCO Xenix for 386s, and Unisys 5000.



XDOS Binary Compiler System.

(continued from page 9)

This points out the potential for emulations other than DOS. As much of the Unix world moves away from 68000-based software to RISC (for example, 88000), not all of the current software can or should be ported. Emulation of the 68000 chip family will become a useful bridge between systems. Thus, the user who has upgraded to an 88000 could open an emulated 68000 window. Other possibilities include VAX emulation for Digital's RISC machines, OS/2 under Unix (here, the current lack of market demand is an inhibiting factor), or Macintosh emulation (which may be subject to some legal issues).

The Future Role of Emulation Technology

Today, DOS emulation technology, in all three of its forms, serves one basic role: to give DOS access to users of non-DOS systems. For the most part, this is occurring in small multiuser and workstation environments.

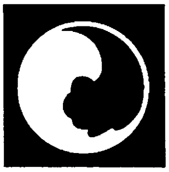
Tomorrow, while these environments will still exist, the emphasis will be on providing DOS emulation in a distributed network computing (DNC) environment. The combination of the client-server networking model and the X-Window distributed display capabilities will enable any user from an X display station on the network, be it a powerful workstation or an X terminal, to open a window and launch a DOS application. The DOS window will be available just as any window would be to run a Unix or VMS application. And those users who need EGA and VGA for more than occasional use will have hardware assistance on their workstations to provide performance equal to the standard standalone DOS PC.

While X-Window and DNC enable the access to processes throughout the network, these processes have to run somewhere. For DOS, they will have to run as emulated tasks on a non-DOS machine, since DOS, as a single-tasking operating system, cannot function as a server for anyone else. Thus, the roles of both Intel- and non-Intel-based machines will become DOS servers, generally running a native Unix with a DOS emulation technology as one of its tasks, or, in some cases, as its primary task. The choice between the Intel and non-Intel architectures will depend somewhat on how the capabilities of each develop—particularly the processor vs. video performance trade-off. However, both will play an essential part in the mixed environment network.

As we move further into the future and away from DOS, the situation will again change. There will no longer be a single system that must be emulated on all platforms such as DOS is today. Rather, there will be a multitude of operating systems and processor architectures, and the role of emulation technology, particularly software coprocessing, will be to provide both a migration path and a real-time bridge between them.

Patricia Seybold's Office Computing Group

Special Research Reports



OSF And UNIX International *IMPLEMENTING THE OPEN SYSTEMS AGENDA*

Available for $495

By Judith S. Hurwitz

The inception of the Open Software Foundation (OSF) and Unix International (UI) has changed the dynamics of the open systems industry for the future. These organizations are being influenced by the standards-setting work of X/Open.

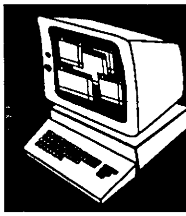
This report will look at the origins, developments, and structure of these organizations as well as their plans for the future development of their environments. It will also examine the implications of each organization and the impact they will have on Unix and the standards movement as we move into the 1990s.

Among the highlights of this in-depth report are:

- Applications Binary Interfaces: How OSF and UI will implement the ABIs and the implications for shrink-wrapped applications.
- User Interface: A close look at OSF's Motif and AT&T's Open Look
- Operating System Kernels: Design, implementation, and technologies
- Communications and Networking Foundations: How each group is building its networking underpinnings
- The Development Environment: Tools that will emerge for developing applications in these environments.
- Administration, Backup, and Security: How these vital areas are being addressed
- X/Open and the Common Applications Environment: The impact that X/Open's portability guides will have on OSF and UI

Call 1-800-826-2424 to order or to receive more information on this report (in Mass., call 617-742-5200). Or fax your order or request to 617-742-1028.

Digital's Office Strategy *AN OPEN ARCHITECTURE*



Available for $495

By Judith R. Davis

Digital Equipment Corporation has finally recognized the importance of the desktop in winning the battle for the network and network services. The company's newly articulated office strategy stresses the user's ability to select a non-Digital desktop device—a DOS-based PC, a Macintosh, or a Unix workstation—and still fit comfortably into the Digital networking environment.

This special report details Digital's approach to an open architecture and the importance of both architectural underpinnings and third-party support. It also covers the components of the strategy, including the convergence of VMS and Ultrix, CDA (Digital's compound document architecture), DECwindows, decision support, electronic publishing, image, voice, and document management.

We also assess the features and functionality of Digital's recent product announcements and how well they support the overall strategy. These products include the new hardware and workstation platforms, DECwrite and DECdecision—the company's first DECwindows and CDA-compliant software—and the VAXimage family of imaging products.

Last, but perhaps most important, we compare Digital's strategy, architecture, and products to those of its primary competitor, IBM.

Call 1-800-826-2424 to order or to receive more information on this report (in Mass., call 617-742-5200). Or fax your order or request to 617-742-1028.

NEWS
PRODUCTS • TRENDS • ISSUES • ANALYSIS
ANALYSIS

•INSIDE•

Impressive Introductions from Uniplex at UniForum. **Page 15**

Nixdorf Adds Image Processing to its Targon Office Architecture. **Page 16**

Relational Technology Ports Ingres to the 486 Environment. **Page 17**

Hummingbird Technologies Turns PCs into X Servers. **Page 17**

•UNIPLEX•

Expanding Horizons

UniForum in Boston was basically a bust for those Unix groupies looking for new and exciting applications. There simply weren't any to be found. The few applications that were on display—Q-Office, Alis, and Interleaf, among others—didn't have anything new to show. The vendors (or OEMs) simply trotted out old versions of the software.

The only major introductions at the show were made by Uniplex (Irving, Texas). The company, which is the acknowledged leader in the Unix office system market, has been getting a lot of grief from folks like us for not keeping up with the state-of-the-art technologies. President Jeff Waxman has oft been quoted proclaiming Uniplex II Plus as "state-of-the-market, not state-of-the-art." But analysts were concerned that the company might not be ready when yesterday's art state became today's market demands.

We no longer have to worry. Uniplex has obviously put a great deal of thought and development effort into the new technologies, and the results are impressive.

GOING THE X ROUTE. Uniplex II Plus is designed for character-based terminal users. Performance and consistency are major priorities. Therefore, moving to one of the new graphical user interfaces (GUIs) has been problematic for the company. The solution introduced at UniForum, and available to ship, is a native X-Window version of the product, Uniplex Windows, which maintains the character-based interface yet takes advantage of the windowing environment, iconization of files, and mouse usage. Uniplex Windows does not sport pull-down menus—you still have the full-screen main menu and soft key interface. But you can manipulate these soft keys and menus with the mouse. And, surprisingly, the combination of interface styles does not seem incongruous. (See Photo 1.)

By choosing this implementation, Uniplex maintains consistency with terminal users. And performance has been maintained. The demos we saw were on a Sun360, and they were pretty snappy.

Uniplex Windows supports any X-11.3 Windows-based implementation. At this release, the product uses the Athena Toolkit from MIT to provide the GUI. The Uniplex design uses a single binary module based on the X Desktop interface technology from IXI Limited, which can be modified to support specific graphical interfaces, such as Motif.

Uniplex is committed to writing to Motif as well as any other GUI when "there is market demand." (We keep hearing that!) But the company is firm about maintaining the character-based interface for terminal users. Users will always have the option of using either the GUI or the character mode.

The X-Window interface is an add-on to Uniplex II Plus that will not make existing systems obsolete. According to Product Manager Simon Q. Walden, "It is important to note that systems integrators, commercial resellers, and end users...will be able to integrate other solutions with Uniplex Windows, just as solutions can be added to the character-based Uniplex interface."

BECOMING A PC PLAYER. Not only has Uniplex brought its product to the new world of GUIs, but the company also has finally made the PC a full player in the Uniplex game. Uniplex DOS 6.1 ($795) is much more than a terminal emulation product. The Uniplex word processor, spreadsheet, screen builder (customization tool), and card index feature all reside on a 286 or above PC. The PC would typically be connected via NFS (Network File System) to a Unix host running Uniplex, but this isn't necessary. All these modules work independently of a Unix

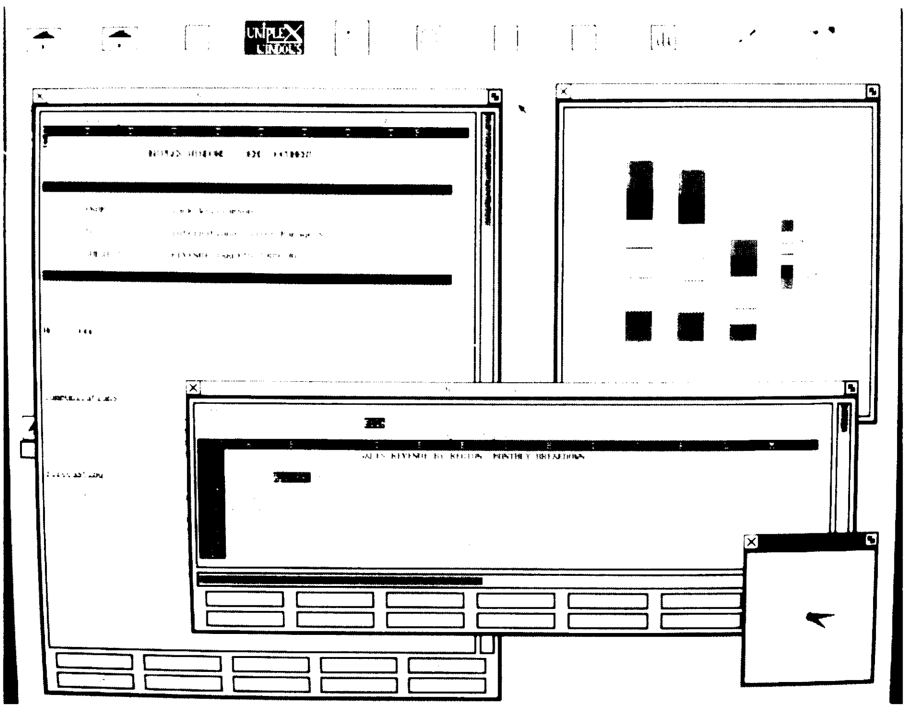


Photo 1. Uniplex Windows. Notice that files and tools are iconized (across the top) in typical X fashion. And the windows each boast scrollbars and icons for window manipulation. But the bottom of the screen is pure Uniplex, displaying the active soft key commands.

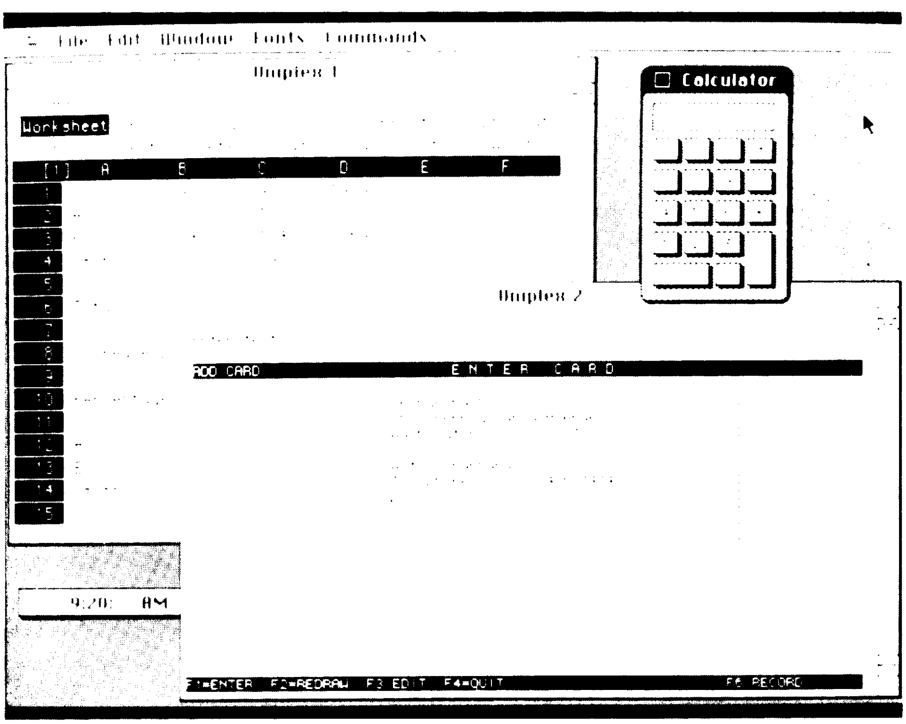


Photo 2. The Mac A/UX interface combines Macintosh's pull-down menus with Uniplex's soft keys.

host, yet the file formats are completely compatible with host files. In most implementations, however, the PC will be hooked up to the host, and PC users will have full access to all Uniplex II Plus modules (E-mail, calendaring, etc.).

The neat thing about the DOS product is the transparency of the connection to the host. The user simply picks, say, E-mail off the PC menu. The system flashes a subtle message saying that it is accessing the host. The performance is pretty impressive. Again, the system was running on a Sun360, not a particularly high-performance system, yet access time was a matter of a few seconds. If the user doesn't specifically look at the prompt line at the bottom of the screen, he or she would never know that the system wasn't working locally.

RUNNING ON THE MAC. Also announced at UniForum was the newest A/UX version of Uniplex, 6.01. The Mac version does include standard Mac pull-down menus, but the familiar Uniplex soft keys are still active (see Photo 2). Pricing for the A/UX version of Uniplex begins at $1,495 for a single user and ranges up to $9,245 for a 32-user license. —*R. Marshak*

•NIXDORF•

Image Processing

We've been impressed with Nixdorf's strategy ever since the company announced its Targon Office architecture last year. Targon's approach is to anchor next-generation technology (complete with distributed computing, object orientation, and procedural automation) with an integrated, standard environment. Targon targets three platforms: DOS, OS/2, and Unix. Support for communications standards such as SNA, OSI, ISDN, and TCP/IP is also provided. Furthermore, Nixdorf isn't arrogantly assuming that you'll only want to play with Targon products, and thus it provides access into existing applications. Not a bad picture, eh?

Well, add to it DCPA Image, a document image processing system. It's a dedicated hardware and software system—designed specifically for image processing. To that end, it features a capacity for storing large images (up to 500,000 bytes per uncompressed 8 1/2-by-11 inch paper), a 200 dpi (dots per inch) video display, and high-speed OCR (optical character recognition). DCPA Image documents on the system can be displayed, modified, annotated, enlarged, reduced, printed, faxed, and mailed electronically to other systems. The DCPA application appears as a window that is superimposed on top of the image being processed.

The system also handles document indexing and supports shared libraries. Nixdorf's distributed image network architecture lets you retrieve an image transparently from any image storage device, be it local or remote.

DCPA ARCHITECTURE. The system has two main components: the image workstation and the host system running DCPA. The host unloads document image processing tasks onto the workstation.

An image workstation connects to the host via a standard RS-232 serial port. In distributed environments, multiple image processors are interconnected on an Ethernet image LAN, over which multiple image peripherals (e.g., scanners, faxes, and printers) and storage devices may be shared.

VITAL STATISTICS. DCPA Image was designed for Targon 35, 31, or 386 host systems running DCPA with special purpose image workstations. The system will run you anywhere from $20,000 for a single-image workstation to $500,000 for a complete-image network. Mass storage options include an internally mounted 50MB Winchester drive and a 900MB optical WORM (write once, read many) drive.

CONCLUSION. An investment in image processing is a sign of a forward-thinking company. The impact of image is just beginning to be realized, and it's precisely this kind of technology that has made Nixdorf catch our eye. (And, it seems, it will continue to do so; we've penciled in Nixdorf as a focus for a future feature article.) ◐

—L. Brown

•RELATIONAL TECHNOLOGY•

Porting to the 486

Relational Technology recently announced plans to port Ingres, its relational database management system (RDBMS) to Intel's 486 architecture. The announcement is basically good PR for Relational Technology. Ingres is the first major Unix RDBMS to be promoted as a 486-based product. (Ever since Intel announced the 486 at Comdex last April, vendors have been falling over each other to be the first to announce future products based on the 486 chip. So Relational Technology can take bragging rights here.) But we expect the other vendors to follow suit. The real good news for Ingres users is performance.

THE 486 CHIP. The 486 environment will offer Ingres some significant characteristics:

- Compatibility with 386 machines and applications.
- Multiprocessing support.
- Some RISC-design techniques (to execute frequently used instructions).
- Speed. According to Intel, the speed of 486 environments equals and sometimes surpasses many RISC implementations, including SPARC, Motorola 88000, and MIPS R3000.

Relational Technology estimates that Ingres will run three to five times faster in a 486-based environment than it does on a standard 386 PC computer.

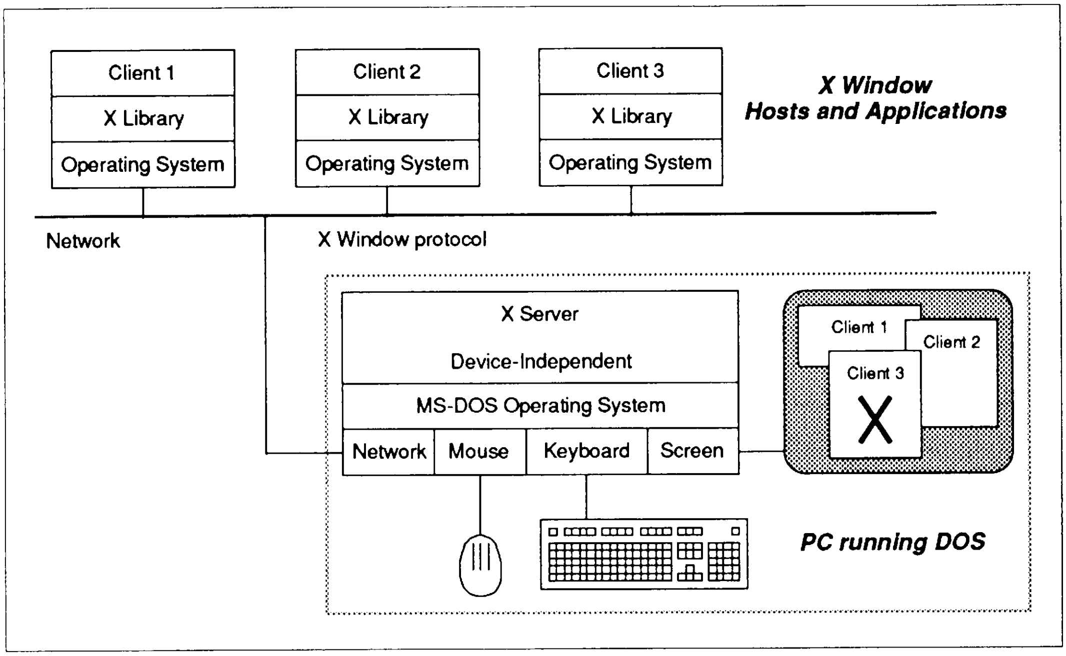
JOINING FORCES. Relational Technology will work with Corollary Incorporated (Irvine, California) to port Ingres to multiprocessor 386 and 486 Unix environments. The two companies have a joint marketing and engineering agreement that will take advantage of Corollary's 386/smp (symmetrical multiprocessing) operating system. Incidentally, 386/smp is a multiprocessing version of Unix based on Xenix System V and SCO's Unix System V/386 for computers using 386 and 486 processors.

SCO and Open Desktop. Considering that Corollary's operating system is partially based on SCO technology and that Ingres is currently being bundled with Open Desktop, the technical developments between the Corollary and Relational Technology become a little more intriguing. With the performance enhancements that the 486 architecture extends, Open Desktop becomes even more promising as a robust desktop computing option. ◐ *—L. Brown*

•X-WINDOW•

The Hummingbird Flies

As X-Window gains validity as an applications architecture, the race is on for capturing the low-end desktop side of the X application. Currently, the division seems to be between those who would have us use dedicated X terminals—Acer, Visual Technologies, Advanced Computing Devices, etc.—and those who would have us use our current desktop device as an X display station—Digital, Apple, Locus (with PC Xsight), etc.



DOS X server in the X-Window architecture.

Both solutions are based on the X-Window architecture, which distributes the application between a host (known as the client, in X terminology) and the user's display station (known as the X server). The display station's only role (other than providing network connectivity) is to "serve" the application by building the screen that the user sees and allowing the user and the application to manipulate it via the windowing paradigm.

The X terminal does just that and only that. The PC or Mac, on the other hand, runs the X server software as the primary task under its operating system, in effect emulating the X terminal. The X terminal has the advantage of lower cost because it doesn't need such PC extras as disk drives and operating systems, while the PC-based X server offers at least the ability to have files transferred to and from the PC disk drives. With multitasking or, at least, context switching, the PC or Mac can run applications in addition to the X server, allowing some degree of integration (generally of the copy-and-paste variety) between the environments.

The key advantage of the PC is that there are already millions of them out there, many looking for their role in the new distributed architecture. It seems predestined that this role will be as an X display station.

THE HUMMINGBIRD FLIES IN THOUGH THE WINDOW. Hummingbird Technologies (Markham, Ontario), a small engineering and consulting company heretofore specializing in voice/data integration and micro-to-mainframe products, has entered this debate squarely on the side of using existing PCs as X display stations. Its initial product in this area, the HCL-eXceed, is an implementation of the X-Window Release 11.3 server, which runs on AT or higher class PCs.

Available immediately to end users ($495) and OEMs, HCL-eXceed enables the PC to emulate a graphics workstation operating in a networked environment. The PC as the X server can simultaneously present up to 16 applications (clients) running on the same or different host computers connected via TCP/IP. The presentation windows can be manipulated via the mouse, and information can be moved between applications via the X-Window Copy-and-Paste facility.

The HCL-eXceed server contains an embedded TELNET application that provides initial connectivity to the host. Using this facility, the user is able to log in as an ASCII terminal. Once logged in, the user may launch an X-Window application, and HCL-eXceed will automatically switch into server mode when the first X-Window connection is detected. The TELNET application also enables file transfer between the PCs disk drives and any of the hosts on the network.

Requirements. HCL-eXceed is a software-only product that will run on any AT or better DOS machine which has:

- 640KB of RAM.
- EGA or VGA graphics (super EGA and VGA are supported).
- EGA or VGA monitor.
- Hard disk.
- LIM memory (optional).
- Ethernet network interface card. HCL-eXceed currently supports PC TCP/IP products from Exelan, FTP Software, and the Wollongong Group.

We expect MS Windows and OS/2 PM versions to follow, as well as other network (i.e., DECnet) support in future versions.

Features. HCL-eXceed is a feature-rich implementation of an X server, particularly when compared with some of the low-end X terminals. For example, HCL-eXceed supports EGA and VGA adapters, allowing up to 16 colors per color map. Client applications may choose which 16 colors are to be displayed at any given time, with the user free to select default foreground and background colors from a compiled database of colors and their associated RGB values.

Another powerful feature of the HCL-eXceed server is its support for X-Window Release 11.3 font creation and naming and alias schemes. HCL-eXceed maintains its own font database on the PC. Logical font names may be constructed automatically from named properties or from a preprocessed font property. The server supports wildcards as well as aliases for font open and search requests.

A combination Make/Font-Compiler compiles source-file fonts into customized HCL-eXceed run time fonts. The utility can selectively build fonts by analyzing the respective dates of the source and run-time versions of the fonts. The compiler supports wildcard specifications and can make an entire directory of fonts in a single run.

X DISPLAYS. We can expect to see more implementations of low-end X display servers, both from the X terminal and the PC camps. Eventually, the distinction will tend to blur, as X terminals become more like PCs by, for example, adding disk drives.

—*D. Marshak*

Topics covered in Patricia Seybold's Computer Industry Reports in 1988/1989:

Back issues are available, call (617) 742-5200 for more information.

Office Computing Report

1988—Volume 11

#	Date	Title
11	Nov.	Upper CASE Tools—Getting Users to Help Design Systems
12	Dec.	Executive Information Systems

1989—Volume 12

#	Date	Title
1	Jan.	Digital's Office Strategy—Going After the Desktop
2	Feb.	Compound Documents Take the Stage
3	Mar.	Wang's Freestyle
4	Apr.	Focus on Training Technologies—Office Systems Training from DG, Digital, HP, IBM, and Wang
5	May	The LAN Office Emerges—WordPerfect Enters OA Race
6	June	Objects at Floodtide—Object Orientation Permeates New Development
7	July	DECwrite—Advanced Applications under DECwindows, Part I
8	Aug.	DECdecision—Advanced Applications under DECwindows, Part II

UNIX in the Office

1988—Volume 3

#	Date	Title
11	Nov.	Data General's Unix—A New Beginnning
12	Dec.	Document Processing with Interleaf *Second Feature:* Apollo and Sun Service and Support

1989—Volume 4

#	Date	Title
1	Jan.	Unisys Ofis Ensemble—In Search of Cohesion
2	Feb.	A Tale of Two Operating Systems—System V.4 and OSF1
3	Mar.	FrameMaker 2.0—The Art and Science of Document Processing
4	Apr.	CMU's Andrew Project—A Prototype for Distributed, Object-Oriented Computing
5	May	Sybase—The Next Generation
6	June	Thoroughly Modern Unix—The Shape of New Operating Systems
7	July	Surveying the Users—Taking the Pulse of Commercial Unix
8	Aug.	Graphical User Interfaces—A Developer's Quandary

Network Monitor

1988—Volume 3

#	Date	Title
11	Nov.	Data General's PC Networking Strategy
12	Dec.	3+Open—3Com Makes Its Move

1989—Volume 4

#	Date	Title
1	Jan.	APPN—Peer-to-Peer Networking Arrives at IBM
2	Feb.	Bridges & Routers—Dividing to Conquer
3	Mar.	BBN Builds Big Networks—Designing Private Packet-Switched Networks
4	Apr.	Northern Telecom DNS—Trying to Build Intelligent Networks
5	May	NetView—IBM's Enterprise-Wide Manager
6	June	Apple's Networking Strategy—Bringing the World to the Desktop
7	July	Distributed Object Management—Networks Made Easy... And More
8	Aug.	Proteon—Platforms for the '90s