Vol. III / No. 3

STRIDE

GRAPHICS & THE NOD

Hyperception

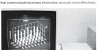
SOFTWARE FOR SAGEBRUSH GRAPHICS ON SAGE AND STRIDE COMPUTERS

HYPERCEPTION PRESENTS THE HYPER-SERIES: SOFTWARE SUPPORT FOR SAGEBRUSH GRAPHICS

In addition to standard SAGEBRUSH command software, Hyperception also offers complete packages which enable immediate and effective use of the outstanding graphics provided by SAGEBRUSH units for SAGE and STRIDE computers:

- **Hyperplot** A business display package, Hyperplot features a truly friendly human interface. Using Hyperplot comes naturally; users typically do not find it necessary to refer to documentation.
  Some of the reasaons why Hyperplot can quickly provide charts and graphs that communicate business information so much more effectively:

  - Menu-driven selections
  - On-line HELP menus
  - Five types of output format (3-D histogram, pie-chart, task-scheduler, multi-variable graph, standard histogram)
  - Data sets created by menu query or text-editor, modified using text-editor
  - Single data set structure, allowing one data set to be presented in any of the five output formats
  - On-line example displays
  - Screen dump to printer with dot-addressable graphics mode



- **Hyperpaint** Turns a SAGEBRUSH display into a graphics sketchpad via the BRUSHRAT option (joystick or mouse). In line with the software design philosophy at Hyperception, Hyperpaint also boasts an exceptional human interface:

  - Icon-driven selections
  - Free-form graphics entry (pencil and varying widths of paint brush)
  - Figure generation (boxes, circles, 'radialoids')
  - Text labelling (horizontal and vertical, varying boldness)
  - Disk storage and retrieval of images
  - Screen dump to printer with dot-addressable graphics mode
  - Constant user feedback provided by screen cursors which change personalities depending on the current function

- **Hypersignal** A powerful signal processing package which takes full advantage of the speed provided by 68000-based machines, Hypersignal offers a few surprises for signal processing professionals:

  - Menu-driven operation in all cases
  - Frequency analysis. Includes 3-D type spectrogram representation with hidden line removal (using the SAGEBRUSH foreground line-drawer for speed), optional cross-hatching, and full 3-axis generation. Also includes FFT generation, Hamming window, and log magnitude approximation (all 68000 asm. lang. coded for speed).
  - Frequency-domain generation of complex signals using BRUSHRAT option (either joystick or mouse) for zero-phase transformation into time waveform.
  - Time analysis. Includes time-waveform display with scaling and zoom functions. Also includes periodic time waveform compression and re-creation via linear predictive coding technique (also 68000 asm. lang. coded).
  - Disk storage and retrieval of images
  - Screen dump to printer with dot-addressable graphics mode

ALL PRICES SUGGESTED RETAIL ONLY: DEALER PRICES MAY VARY

Hyperplot: ________ at $235 ea.
Hyperpaint: ________ at $235 ea. (free with purchase of BRUSHRAT option)
Hypersignal: ________ at $595 ea.
BRUSHRAT: ________ at $225 ea. (includes command set software to access analog channels and external 8-bit digital port)

Method of payment (Texas residents add 6%): ☐ Credit Card ☐ COD ☐ Check or MO

Name: ________ Company: ________
Address: ________ City: ________
Mastercard No.: ________ Interbank No.: ________ Exp. Date: ________
Visa No. ________ Exp. Date: ________

Hyperception • 3636 Lemmon Ave., Suite 400 • Dallas, TX 75219 • (214) 698-5538

Push For Performance

by Verlene Joyce Bonham

Take another look at this month's front cover. Did you notice the small gadget perched on top of the terminal? We call this the NOD.™ It plugs into a standard RS-232C port and is powered with a small external wall-plug supply, like those for calculators. Several small pieces of reflective material come with the NOD. Stick one to the end of a plastic straw or pencil. Put it behind your ear. Turn on the NOD demo, face the screen and nod your head. The cursor will follow the movements of your head. Intriguing, right? It's obviously a potential replacement for light pens, mice, and track balls since it frees both hands for important tasks such as pushing the "Fire!" button on a video game. What else is it good for? We dunno. To quote Einstein, "What good is a new born baby?" We're sure you'll think of something. An easy bet is you'll integrate it with the new graphics development system. (See the article on page 8 of this issue.)

We knew our new graphics would be fast, but it turned out even faster than we had hoped. In fact, certain routines had to be slowed down to make their movement visible! And while we put in a lot of the neat features the MacIntosh has, this is obviously no MacClone. Another trick was to put the standard Stride terminal (the WYSE WY-50) to double duty as a graphics terminal. (And you thought we went through the agony of standardizing on another terminal just to get one that went at 38.4K baud.) Today, the graphics package is a development system only. There's no software for it yet. Again, we're hoping you, our users, will remedy that. At $400 it shouldn't take long.

Note that the graphics package includes the sources to the low-level graphics programs. We continue to follow our "open company" policy even if many do think we're crazy. Yes, it will make it easier for competing companies to rip off the long hard year's work put in by R&D. Some of those routines have some amazing mathematics in them, and it's hard for authors Bill Bonham, Brad Christian and Don Marsh to turn them loose. Note that they are copyrighted. Developers are welcome to incorporate them into applications for the 400 Series — others beware!

Bruce Robertson in R&D has also been very busy. The UNIX release so anxiously awaited for some time is quickly becoming a reality. Those who attended Stride Faire got to see an early preview of System V UNIX. Some of our users have voiced reservations about UNIX. After all, what's a nice company like ours doing in a UNIX environment? One anti-UNIX letter we received pointed out, in an amusing but unprintable way, that most 68000/UNIX companies out there haven't been very profitable despite media bally-hoo and the intensive advertising by AT&T. These companies are writing a new chapter in computer history all right — Chapter 11! Our answer is to point out that most of them based their whole operation on UNIX. Stride Micro, on the other hand, has diversified with many environments. Officially, we now support and market the p-System, CP/M-68K, Idris, RM/COS and UNIX.

Other heavy-duty operating systems are also supported through various software vendors. BOS is a business environment. Mirage is extremely active in the U.K. with support for Pascal, FORTRAN and APL. FourByteFORTH and HyperFORTH provide support for the efficient FORTH language used for process control and mathematics. PDOS has a fast small kernel for real-time applications. MOSYS was developed to support the new Modula-2 language. Volition also supplies a look-alike of the p-System to support their Modula-2 compiler. Hopefully, the native code version of that compiler will soon run under Version IV.2 of SofTech's p-System. TRIPOS is a research environment for systems programmers and the LISP language. Pick is being ported to the Stride. A newcomer, S1, will describe itself soon here in *In Stride*.

That makes 15 operating systems to date, and we may have missed one or two. Thanks for the concern, but a little UNIX won't hurt us.

And another important announcement — 256K bit RAMs are now available on the 400 Series. Would you believe a 420 with 2M bytes of memory? A 440 will hold 8M bytes and a 460 will hold up to 12M bytes. Also new, lower RAM prices are in effect as of February 11, 1985.

The NOD, graphics, UNIX, and expanded RAM memory—quite a lineup of new products to entice you, all designed to live up to our reputation. Our trademark, "Performance By Design" is not just hype. Take a look at the details in this issue's articles and see what we mean. □

In STRIDE

Editor: Verlene Bonham
Advertising: Diana Swaim

***In Stride* is a monthly publication of Stride Micro. Subscriptions are $24 for one year. Purchase of a Stride computer includes a one-year subscription upon receipt by Stride Micro of a fully completed owner's registration card. Back issues are available for $2.00 per copy, as supply lasts. Requests for subscriptions, reprint permission, ad rates, bulk orders, or submission of prospective articles should be sent to the Editor, *In Stride*.**

Postmaster: Change of address notices should be sent to :

Editor, In Stride
Stride Micro
4905 Energy Way
Reno, NV 89502
(702) 322-6868 (8am — 5pm PST)
TWX: 910-395-6073

Stride Micro Eastern Division
15 New England Executive Park
Suite 120
Burlington, MA 01803
(617) 229-6868

Stride Micro Southern Division:
14755 Preston Road
Suite 600
Dallas, TX 75240
(214) 392-7070

Stride Micro, U.K.:
Chesham House
136 Regent St.
London W1R 5FA
Tel:01-437 6900
TLX:267653

Intelligent Robotics and Modula-2

by Lewis J. Pinson

Intelligence is a human characteristic that has many varied definitions and that has different implied meanings depending upon our individual experiences. It is something that is thought to be inherently good and having a lot of it is desirable. However, we find it very difficult to accurately define, very difficult to measure; and the currently experimental methods for enhancing intelligence provide only limited and questionable success.

In spite of our limited understanding of intelligence, we seek to provide artificial intelligence to robotics systems and other machinery. In order to do this we must find ways to describe and quantify intelligent features. These intelligent features are then provided to the robot in the form of computer software. *Figure 1* illustrates one concept of an intelligent robot with its multiple sensors, intelligence and general mobility. Of course practical robots today are quite different from this romanticized prototype, and much more closely resemble a tank married to a vacuum cleaner.

No doubt because machines were created by humans, machine intelligence mimics human intelligence, even to having different degrees of sophistication. Four levels of robot intelligence are commonly defined, ranging from the 'dumb robot' to the 'intelligent robot'.

Level 1:
The Dumb Robot

This level is represented by the totally preprogrammed robot. Every action is part of a preprogrammed sequence of steps. The dumb robot typically performs automatically repeated program sequences and has few if any sensors. Dumb robots are rarely called "robots" and often are simply referred to as "automated" machines.

Level 2:
The Expert System Robot

This robot makes decisions based on a set of preprogrammed hierarchical decision rules and a stored knowledge base. It can accept sensor input as part of the decision process; however, it can only handle situations for which it has been programmed. A non-expert human working with an "Expert System" robot can often provide results of the caliber expected of highly skilled professionals, hence the term "expert." An example of this type of system is one where a doctor or para-medical human expert works with an expert robot system for medical diagnosis.

Level 3:
The Trainable Robot

The trainable robot is essentially a dumb robot that can learn the new actions needed to accomplish a task if a human operator shows it, by example, what to do. Robots with teach pendants fit this category. (A teach pendant is a mechanical device for recording the movements made by a human instructor for later repetition by the robot.) The expert system can be trainable if upon recognizing an unknown condition, it prompts a trainer for additional information.

Level 4:
The Intelligent Robot

The intelligent robot has the ability to learn. It applies the rules of logic or some other method for making new decisions and testing their validity. The intelligent robot has sophisticated and controllable sensors for gathering information. It is controlled by intelligent software and has large memory capacity. The key to the intelligent robot is that it makes *its own new decisions*.

In terms of where we are now and where we want to go, it has been said that today's robot is deaf, dumb, blind and has one arm tied behind its back. Most have single manipulator arms with few if any sensors. Their primary applications are in automating simple industrial tasks. Sensors are being added to the robot arm along with processing and decision heuristics for automating more complex tasks. Simple object recognition is possible; however, the processing is still slow and depends on preprogrammed decision logic.

The goal for future robotics systems is to incorporate more intelligence-like features such as the ability to: accept natural language instructions, understand voice input, have visual discrimination capability, be capable of intelligent motion, be capable of learning, be self-maintaining and have adaptive CAI (Computer Aided Instruction) capabilities. There will be a continued need for not-so-smart robotics systems especially where the motivation is not intelligence but economics, safety, productivity, or quality control.

Robotics systems can be described in terms of three specific functional areas. These are the *sensors* , the *computer* , and the *mechanical* parts. All components of the three areas must co-ordinate well, but the computer is of particular interest in designing an intelligent robot, since that is where the seat of intelligence will be.

". . . today's robot is deaf, dumb, blind and has one arm tied behind its back."

Therefore, of the many diverse disciplines that contribute to the design of a robotic system, one of the most important is 'intelligent software development'. Consider the following two related, but separate, development methods:

#1:Development of Intelligent Software

The methods of artificial intelligence, knowledge representation and knowledge acquisition make possible the development of software that exhibits human-like intelligence capability.

#2:Intelligent Development of Software

The methods of software engineering, modular design, abstraction and generic programming make the development of any software system a more logical, high-level, human intelligence-like activity.

Application of both of these methods produces an intelligent software system that has the desirable features of: ease of development, ease of maintenance, expandable knowledge base, intelligent decision making and human understandability.

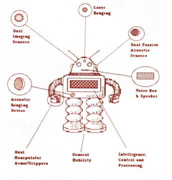
One design approach for the intelligent development of software for an intelligent robotics application is to develop a 'robot abstraction'. This abstraction defines objects that represent the physical components of the robotics system and the actions that represent the functionality of these components. As a minimum, the three major components of a robot (sensors, mechanics and computer) should be described in terms of objects that can be represented as abstract data types. Operations on these objects must then be defined in terms of functional abstractions with the appropriate binding.

Modula-2 is a language well-suited for defining the 'robot abstraction'. Using the object-oriented techniques and the support for abstraction provided by Modula-2, it is now possible to develop the objects (abstract data types) and the actions (processes) that represent an intelligent robotics system.

These objects and actions that represent the robotics system hardware have clearly defined control interfaces. The software interfaces provided by Modula-2 are also clearly defined and provide a logical and synergistic approach for development of the software system. Because Modula-2 supports low-level access to the hardware control interfaces and the speed available through native-code implementations of Modula-2 is excellent, a particularly good, fast link between the hardware and software interfaces can often be accomplished without resorting to assembly code.

Other features of Modula-2 such as extensibility and support for generics make it a language well-suited for control applications.

The finer details of the robot abstraction are currently in the process of being defined. The successful completion of this task will provide useful tools for the programming and control of intelligent robotics systems. The choice of Modula-2 as the language for implementation of the robot abstraction provides an opportunity to develop intelligent software using intelligent methods. □



What Every Good Robot Should Have

Figure 1

Ed. Note: Lewis J. Pinson is an associate professor of computer science at the University of Colorado at Colorado Springs and the president and founder of Computer Intelligence Company. He is co-author of the book "Introduction to Computer Science with Modula-2" scheduled to be published in 1986 and author of "Electro-optics" to be published summer of 1985 both by John Wiley & Sons. Professor Pinson was a speaker at Stride Faire '85.

Customer Service

by Peggy Lakey

CP/M-68K Lost Files

CP/M-68K keeps a copy of the floppy disk directories in memory and uses that to update files. If you swap a floppy you must type a CTRL/C before the new disk is inserted. Otherwise, the system will continue to use the old directory — causing you to lose files. CTRL/C will insure that you are using the correct directory. If in doubt, a CTRL/C will not hurt anything.

1600 Block Floppy Format

Going back in history a bit, when the Sage II was first announced, SageUtil, the precursor of Sage4Util and UTIL, allowed users to mess around with the floppy disk parameters. (UTIL still does, but since then we've added a simple menu of standard selections so most folks don't have to go into the low-level setup parameters.) This flexibility allowed our users to read/write to various formats used by other systems. Well, by bending the rules a bit, the users found out that Lo! and Behold! they could cram 1600 blocks onto a standard 1280 block floppy. The 1600 block scheme was probably first started by some users of NCI's p-System which uses a 1600 block format.

Despite dire warnings and statements from us that we did *not* recommend the 1600 block format, a number of people adopted it. We have to admit that some of us here at Stride played with it a bit, but it was too much of a problem to switch between the 1280 and 1600 formats so no one ever stayed with it.

For a long time it seems that it didn't really cause much of a problem. Stride was shipping Mitsubishi floppy drives then, which seemed to handle the format OK, although Mitsubishis are not spec'd to run it. In case you're wondering what rule is being violated here, its called the GAP 3 parameter and it sets the length (and time) of the space between the end of one track and the start of the header of another track. One way to cram more information on a disk, is to make each track smaller. Obviously, you can't cut the amount of data, so you cut the gap down. NCI's disk format, by the way, also shortens GAP 3.

As more machines were shipped, we found that some Mitsubishi drives had a seating problem, where the diskette was not always properly aligned when inserted. When a better drive than the Mitsubishi came along, Stride switched. The new TEAC drives are noisier than the Mitsubishi, but we have yet to insert a disk and have the system fail to find it. However, the TEAC is not as tolerant as the Mitsubishi for 1600 block operation.

If you are not sure which drive you have, check the front. On a **Mitsubishi** drive, the latch is centered, you press it to open. A round red light is on the bottom left. On a **TEAC** drive, the latch is a handle, no hole shows in front. An arrow shows how to move the handle. A large rectangular red light is on the upper left.

We evaluated a number of drives before going to the TEAC, and none of them did very well at 1600 blocks —and none of their manufacturers spec'd the GAP 3 parameter to allow for it. So our advice is to politely ignore well-meaning folks who give you a little tip on how to get more floppy room by using the 1600 block format.

If you simply *must* read a diskette formatted for 1600 blocks, first check the precomp setting in UTIL. It should be 250 not 125. This may make the difference between being able to read the information or not. Try increasing the Retry rate. (See the Low-level options in UTIL.)Clean the heads. (The last two suggestions apply to any time you're having trouble reading a floppy.)

One good indication of whether the whole idea is feasible or not is to try formatting a diskette at 1600 blocks. If it works, you got lucky with a little bit better drive. There is no guarantee, however, that it will read a 1600 block diskette created on someone else's drive!

If you have been using the 1600 block format, you should change back to the standard (1280 blocks). Stride does not support or recommend the 1600 block format and may remove it from UTIL to discourage vendors from distributing software in that mode.

SVS FORTRAN Random Function

The RAN function is incorrectly documented on page 149 of the manual. The definition should be:

```
REAL FUNCTION RAN(I)
INTEGER*4 I
```

Where if I=0, then the same number that was returned for the previous call is returned. If I < 0, then a new sequence of random numbers is started. If I > 0, then a new random result is returned. □

Outstanding Dealer

Shown busily at work on CLM's CAD system are (left to right) Professor C.L. Miller, Lou Das Coulias (Sr. systems engineer), Jacky McFadden (vice president) and Jim Zinner (Sr. projects engineer).

CLM/Systems, Inc. Marks 30th Year

CLM/SYSTEMS, INC., 3654 Gandy Blvd., Tampa, Florida, 33611 is the creation of Professor C.L. Miller. An international leader and pioneer in the application of computers to engineering. Among his many awards he has received the George Westinghouse Medal. At Massachusetts Institute of Technology he was the founding Director of the Civil Engineering Systems Laboratory and the Urban Systems Laboratory.

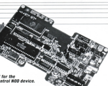
Professor Miller founded CLM/SYSTEMS, INC. in 1968, but the history of his work goes back to 1955. This year marks 30 years of pioneering computer work by the CLM team.

Targeted for the civil-engineering and surveying market, CLM's main product is a turnkey CAD system based on a Stride 440 with an integrated set of software modules: COGO (computerized coordinate geometry), ICES (integrated civil-engineering system) and CLM Graphics (a graphics software and hardware package).

This system has a large area of applications: civil and site engineering; surveying and mapping; transportation engineering; construction and mining; resources management and mapping; utility mapping and engineering; urban planning and public services.

The package also includes complete systems integration, one year of maintenance and upgrades. In addition to all this, a week's training for two people in Florida is included. Other services available are on-site help and custom programming.

Sixteen highly technical professionals at CLM/SYSTEMS, INC., join with Professor Miller to create a highly skilled team and a worthy Stride Micro dealership. □



Infrared LED Technology is used for the wireless cursor control NOD device.

NOD Yes! for Graphics

by Wilbur Harvey

What exactly is a NOD? The front cover of this issue shows a small box perched on the terminal with a little window on the front.

Inside this NOD box is an array of LED's and sensors. When a small piece of reflective material is positioned in front of the box, the light from the LED's bounces off the material and back to the sensors, identifying the location of the reflection. In this way, the NOD can track the position of the material and report changes to the computer. If you stick the reflector to your forehead, for example, you can "nod" your head in various directions and the cursor will follow your head movements.

Potential applications for the NOD are menu selection, arcade games, process control, CAD/CAM, interfaces for the handicapped, motion sensing and many more. Because the NOD reflector can be placed on the head, or eyeglasses, both hands are left free for normal keyboard input. (Our favorite placement, thus far, is a reflector on a plastic straw behind the ear.) This gives it a distinct advantage over a mouse, light pen, or track ball. All of those require the operator to remove a hand from the keyboard and pick up the device. Since the NOD works on reflected light from the LEDs, there are no extra wires, batteries, or grid tablets to use up desk space.

The reflective surface is a special 3M coating composed of almost microscopic glass beads. These create a "retroreflector," a surface which reflects light back in the same direction from which it came. The reflector does not have to be very large, only about 1 square inch, to work. The area commonly used by Stride is shown by the circle on this page.

Even if you were to look directly into the window of the NOD, you would not see any lights. The LEDs used are infrared like those in TV remote controls and are designed so as to not distract your eye from the main viewing screen.

The window itself is a special plastic made to filter out background light, especially the light from the overhead lamps found in most offices. This allows only the LED light signals to reach the sensors inside.

The NOD interface uses standard RS-232C signals and can communicate with any computer system having this type of serial port. The present size of the housing is 4"W x 6"D x 1"H, which fits nicely on top of most terminals. The NOD is powered by a calculator-like wall-plug power supply.

At this time, the NOD is a development tool only and is being marketed as a "Gamma" test product in single-unit quantity. Suggested retail price is $400. It will be a standard stock item as of April 12, 1985 and will be publicly introduced on May 6, 1985. Appropriate patents are pending.

Graphics

One obvious use for the NOD is with the new Stride graphics development package. The 400 Series video board drives a standard (monochrome) Stride terminal (or Wyse WY-50) to provide 784 x 325 resolution with one bit per pixel. In order to work in graphics mode, the terminal must have four shunts installed on its main IC board. You can easily do this in two minutes with only a screwdriver. Note that the video board works *only* with the Wyse WY-50 (or the Stride standard terminal). The video board connects to the terminal at the DB25 Modem port with a special 16 inch cable. It is important not to try to extend the cable beyond 16 inches. Many of the speed parameters of the graphics system depend on keeping the cable short.

The video board itself is a VMEbus compatible card which mounts on standoffs in any Stride 400 Series system. This includes the 420 floppy-based system. For the 420 and 440, the video option includes a new sheet metal backpanel and an extra VME socket. A special jumper cable connects the video board to one of the computer's serial ports.

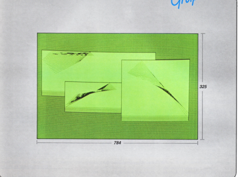
When planning a graphics installation, remember that the terminal/video option and NOD option each use one serial port. On a 420 that means one port for the terminal, one for the NOD with two left for printers and/or modems. If you use a parallel printer, both extra serial ports are available for other uses.

The video board design has 32K bytes of RAM for storing the screen bit-map which may be addressed in the same manner as any other system RAM. Adding the video RAM does not effect the speed in which the main system RAM is accessed. No extra wait states occur during access to the system RAM. Read and write access to the video RAM may involve from 0 to 8 extra wait states (average of 4) depending on when the access occurred. This is because access is done asynchronously. However, this type of access is much faster than that of many other graphics systems which only allow access to their video RAM during horizontal or vertical retrace.

"The speed of the graphics is such that, depending upon application and system capacity, the video option is capable of producing real-time graphic animation."

The actual speed of a system RAM to video RAM transfer is 35 fps (frames per second) while a 68000 register to video RAM transfer achieves 58 fps. (Motion pictures are only 24 fps.) The register transfer is useful for a fast display of a simple pattern or in clearing the screen.

One of the downfalls of cheaper graphics systems, image flicker, is avoided nicely as the speed of writing to the video RAM is faster than a screen refresh cycle. Screen updates look clean and sharp.



Multiple windows can slide across the screen while their contents are being updated. The patterns and the windows are re-drawn dynamically.

The video RAM refresh is done separately from the main CPU board RAM refresh operation. Again, normal memory operation is not impacted by the addition of the graphics system. This insures that programs which do not run graphics will be just as fast on a video station as they are on a system without that option.

The speed of the graphics is such that, depending upon application and system capacity, the video option is capable of producing real-time graphic animation. The video board will operate with CPUs running either a 10 MHz or 12 MHz system clock. The MMU (memory management option) will also work with a video board running on a 10 MHz system.

Another unique feature of our graphics system is the ability to handle both terminal-generated characters and bit-mapped graphics from the video board at the same time. The signal from the video board, which comes through the "MODEM" port and the regular text signal are logically "OR"ed by the terminal. This gives the effect of the text overwriting the graphics display behind it. Note that text will not show up over heavily filled in areas of a graphics display as both of the signals will be "ON" in this case.

To switch between regular "character" mode and the combined "graphics and character" mode, you simply type a command sequence at the terminal. **Shift/Break, Esc~1, Shift/Break** puts you in the combined "graphics and character" mode. **Shift/Break, Esc~0, Shift/Break** puts you back into regular "character" mode.

For now, the software package for the graphics system is a set of low-level utilities intended for use by developers. Source to the utilities is provided so that they can easily be incorporated into the target application. Some of the higher level routines, such as window management, are written in Modula-2 using Volition's native code compiler for Version IV.2 of the p-System. The algorithms for the higher level routines will be provided in case you are working in another language than Modula-2.

The long-awaited release of inexpensive graphics for the Stride machines promises to open a new field of interesting applications. The NOD is an intriguing new idea whose total impact on the future cannot be estimated now, but whose potential is tremendous.

Perhaps the best feature of the entire graphics system is its suggested retail price at $400. As stated above, it operates using the inexpensive Stride/Wyse Wy-50 terminal. Few, if any high-speed graphics systems are available in this price range.

Currently, the NOD is in Gamma test in single-unit quantities as a development tool only. As of April 12, 1985 it will be a standard stock item. Patents are pending. ☐

Ed. Note: Wilbur Harvey is the inventor of the NOD and designer of the graphics board. An early member of the Stride/Sage team, Harvey is a Sage Fellow at Stride Micro.

Memory Management for UNIX System V

by Bruce Robertson

System V UNIX has finally arrived on the Stride 400 Series. The Beta test version was released during Stride Faire on February 8-10. Gamma test release is scheduled for March 15, 1985 with formal release set for May 6, 1985.

UNIX is a popular operating system that has been widely ported to 68000 machines in the past year or two. As there are plenty of UNIX manuals out there and many beginning guides to the system, the main questions you may have are not what UNIX is, but what is unique about UNIX on the 400 Series.

As Stride has a reputation for fast machines, it was important that the UNIX implementation be as speedy as possible. The original implementations on mainframe hardware with multitudes of users have given UNIX a reputation as a slow operating system. We wanted to disprove that — on the 400 Series at least. One of the major efforts made towards speed was the design criteria set for the MMU (Memory Management Unit) option.

Hardware memory management is required for System V UNIX to protect user areas and to arbitrate memory space for each process. UNIX is not available for Sage IV machines, as there is no clean way to retrofit a memory management system for them.

The Stride 400 Series MMU was designed specifically to work with System V UNIX, although it is flexible enough to work well for almost any operating system's needs. It is a factory installed option, and while it is fairly simple to do, we really don't recommend it since you must disassemble the machine to the CPU board level. Once that's done, the 68000 processor must be removed from its socket. Selected shunts are cut on the CPU board. The MMU board is inserted into the CPU socket. The MMU has its own 68000 processor soldered in so the original CPU is left over. The change is invisible to any system not utilizing the features of the MMU.

The 400 Series MMU allocates up to 4 segments (or processes) of 2M bytes each. UNIX only uses two of the segments, one for the kernel in Supervisor Mode and the other to map the currently running user process. Each of the segments allows up to 512 pages of 4K bytes each. Each time the system switches between processes, the user map in the MMU is reloaded.

An exceptional level of performance is provided in that the MMU does not add a wait-state to memory accesses from the CPU board. As the CPU board can now have up to 2M bytes of memory using 256K bit RAMs (an option also announced at Stride Faire '85), most programs will run at maximum speed.

The MMU works best on 10 MHz systems. It can be used with the 12 MHz processor option, however this is largely impractical since at 12 MHz, wait-states must be inserted, effectively slowing the system back down to 10 MHz.

At normal 10 MHz operation, a single wait-state will be added to memory accesses from the Winchester board or RAM board. Because the UNIX kernel is loaded into the bottom of memory, it takes advantage of the fastest memory in the system. While the total amount of utilities in the system is very large, the kernel itself is less than 150K bytes, depending on options.

The BIOS, on the other hand, is stored at the top of memory and may reside in memory accessed with an additional wait-state. This should be kept in mind if BIOS drivers are changed. Most of the drivers used by UNIX are BIOS drivers. However, you can easily install your own drivers in UNIX without changing the BIOS.

This difference in access is important when running benchmarks. Beware! If you time a UNIX program you may get different values depending on what part of memory your program is loaded and, of course, depending on how many users there are on the system. There is no way for the user to force the load location of the program.

How fast is the system? With no other users on the system, the Eratosthenes Sieve, written in C, ran in 1.7 seconds with 32 bit arithmetic. The equivalent time for Idris is 2.3 seconds. This compares with the fastest time on the 400 Series of 1.12 seconds for 68000 assembly code. A time of 1.7 seconds is a very excellent UNIX time.

Each user can start as many processes and take up as much of memory as is available at any given time. Each process, however, is limited to the 2M byte maximum set by the MMU. When loaded, each process is allocated three main areas: the text or code segment, the initialized data (such as constants) and the variable data area. The stack pointer starts at the top of the 2M byte space, with one 4K page initially. The system will automatically assign more memory to the stack in 4K page increments as needed. The data area can be increased on special request by a program. Most UNIX programs assign the size of their data area. For example, the editor EMACS requests more data area room when loading a very large file.

If a process is less than the 2M byte limit, but larger than the actual memory on the system, this results in an error condition indicated by the "no memory error" message.

"With 256K byte RAM chips, a CPU board can have 2M bytes of no wait-state memory."

Disk access is about five times faster than Idris on the 400 Series. A gain of about 10-15% is due to the disk driver using 1K byte blocks instead of 512 byte blocks. UNIX also keeps a hash— coded index of the blocks last used in memory. The blocks themselves are also kept in memory, the amount defaults to 50 but can be set by the user.

Another area which creates an effective speed increase is the scheduler, the UNIX scheduler algorithm being somewhat more efficient than previous versions and look-alikes, such as that in Idris. Operation remains the same as on most UNIX machines. The current time slice for each process is one second. The scheduler is optimized for interactive use and is intended to limit user "hogs" from stealing all of the resources of the system. The system assigns priority automatically. Only the SUPERUSER, of which there can only be one, is allowed to increase the priority of a

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	COMD
3	S	root	0	0	28	0	20	29000	1	261ee	15:35:38	?	21:31	swapper
101	S	root	1	0	0	39	20	3f000	6	12000	15:35:38	?	0:03	/etc/init
3	S	root	0	0	0	0	20	3c000	1	261fa	15:35:38	?	0:09	swapper
101	S	bruce	64	1	0	30	20	bd000	10	20e90	15:40:54	cons	0:09	[csh]
101	S	root	65	1	0	28	20	57000	5	26310	15:40:54	?	0:00	[getty]
101	S	root	43	1	0	26	20	63000	5	136dc	15:40:37	cons	0:00	/usr/lib/errdemon
101	S	lp	60	1	0	26	20	80000	7	1e474	15:40:50	?	0:00	/usr/lib/lpsched
101	S	root	54	1	0	39	20	6e000	5	12000	15:40:45	cons	0:02	/etc/cron
101	S	root	66	1	0	28	20	8a000	5	263b0	15:40:54	h3	0:01	[getty]
101	S	root	67	1	0	28	20	98000	5	26406	15:40:54	h4	0:01	[getty]
101	S	bruce	537	1	0	30	20	d0000	9	21042	15:59:04	h5	0:05	[csh]
101	S	root	69	1	0	28	20	7d000	5	264b2	15:40:55	h6	0:01	[getty]
101	S	root	70	1	0	28	20	8d000	5	26508	15:40:55	h7	0:01	[getty]
101	S	root	71	1	0	28	20	87000	5	2655e	15:40:55	h8	0:01	[getty]
101	S	bruce	563	537	0	30	20	4e000	9	2113a	16:00:34	h5	0:01	vi .profile
101	S	bruce	564	563	0	30	20	b6000	7	21178	16:00:40	h5	0:01	[csh]
101	S	bruce	565	564	0	30	20	71000	4	211b6	16:00:42	h5	0:00	sh
101	S	bruce	566	565	0	30	20	ff000	5	211f4	16:00:46	h5	0:00	[write]
101	S	bruce	569	566	0	30	20	107000	4	21232	16:00:51	h5	0:00	[sh]
101	S	sys	546	1	0	30	20	c8000	4	21270	16:00:04	cons	0:00	[sh]
101	S	sys	551	546	0	39	20	df000	8	12000	16:00:09	cons	0:00	/usr/lib/sa/sadc 1200 3
101	S	bruce	570	569	0	28	20	10d000	4	2645c	16:00:51	h5	0:00	sh
101	S	bruce	575	64	0	30	25	10a000	25	2132a	16:05:49	cons	0:13	make
101	S	bruce	576	575	0	30	25	113000	6	21368	16:06:03	cons	0:01	cc -c -O addng.c
101	S	bruce	583	576	34	20	25	13b000	51	2b680	16:06:33	cons	0:03	as -o addng.o
101	S	root	582	64	0	28	20	14d000	4	262ae	16:06:26	cons	0:02	su
101	R	root	584	582	80	100	20	150000	8		16:06:37	cons	0:04	ps -elf

This display is a "snapshot" of the current processes in a 440 running System V UNIX, with two users logged on and a C compile in the background.

process. However, if you care to be polite, you can reduce the priority of your process by running it with the "nice" command. This reduces the load on the system. A high-priority process will pre-empt any others until it is finished. Then the lower-priority processes resume.

You get a lot for your money in the UNIX release. Four languages are standard: C, FORTRAN 77, SNO (a version of SNOBOL), and BS (a interactive interpretive pseudo-Basic). VI is the standard text editor. EMACS is available separately for $395. SED is a stream editor for very large files. LEX is a lexical analyzer generator. YACC is a parser (Yet Another Compiler Compiler). AWK is a pattern matcher. M4 is a macro preprocessor. NROFF & TROFF are well-known text processors. EQN is an equation preprocessor for NROFF and TROFF. TBL is a table preprocessor for NROFF and TROFF. VOLCOPY, FINC, and FREC are utilities for file backup to floppy or tape. The source for the device drivers is shipped standard.

UNIX is shipped standard on one 45M byte ¼" tape data cartridge. Otherwise, the release takes nearly 20 Stride diskettes in CPIO format, interchangeable with other System V 1280 block floppies. If shipped on floppies, there will be an additional charge.

To run UNIX requires at least a 15M byte hard disk, and a tape drive is recommended to simplify updates and system file management.

The minimum amount of memory for UNIX is 512K bytes for one user. For most applications, another 100K bytes per user will suffice. More should be allowed if a user is running very large applications. With 256K byte RAM chips, a CPU board can have 2M bytes of no wait-state memory which will support 8-10 users fairly well. Performance is generally optimum for 5-6 users, depending on your application. The release will support up to 16 users. Special licensing is needed for more than 16.

How portable are other UNIX programs to the new Stride environment? If the program was not developed on a 68000 it will have to be recompiled before it will run. Fortunately, the C language for UNIX is fairly standard and little difficulty should be experienced recompiling a program. If the original computer did not have a 68000 processor, the program will probably just have to be re-linked.

Networking has always been a major concern of UNIX users as a large number of them deal with extensive networks in university environments. Although the Beta release does not include networking support, it is a natural for the 400 Series with both Omninet and Ethernet hardware available. The first formal release (May 6, 1985) of the product is scheduled to include networking.

System V UNIX sells for $950 on the 400 Series. The MMU sells for $500. For a short, easy-to-read introduction to UNIX and its command structure, try the book by John D. Halamka, Real World UNIX, from Sybex.

The number of UNIX users is large and growing even larger as business applications find a home in this flexible environment. Stride Micro's low-cost, fast machines and the popularity of UNIX make the combination an attractive base for many and varied applications. □

Ed. Note: Bruce Robertson is a software engineer for Stride Micro and has been active in the support and systems work for the Idris product. He is responsible for the majority of the systems work done in porting System V UNIX to the Stride 400 Series.

The January 15th issue of Electronic Products carries the awards story with a full page color feature on Stride Micro's 400 Series Micro computers.

Electronic Products Honors 400 Series

"Innovations, breakthroughs, darn good products — describe them as you will. Electronics designers showed time and again last year that they were capable of turning out one provocative product after another."

With this statement Alfred Rosenblatt, managing editor for Electronic Products, announced the winners of the magazine's Ninth Annual Product of the Year Awards. Stride Micro was proud to see the 400 Series Computer was the only micro to win the award — despite the host of micros announced in 1984 by such major companies as Apple, IBM and AT&T. (Data General's One Lap Computer did make the list.) Twenty-one products were honored, ranging from new op amps to laser printers to flat panel displays.

Just being amongst such distinguished company is a recognition much prized by the hard-working team here at Stride Micro. A small, but dedicated user following has always known Stride has "a darn good product," but this acknowledgement from such a highly influential magazine is great praise indeed.

The award is to be presented by Electronic Products Managing Editor, Alfred Rosenblatt and his associates here in Reno at Stride Micro's corporate office. □

A TRUE FAMILY OF PRODUCTS:

$2900 to $60,000

STANDARD FEATURES:

- 10 MHz 68000 (12 MHz optional)
- 5¼" 640K byte floppy disk drive
- 256K bytes of parity RAM
- 4K bytes of battery backed-up CMOS RAM
- Omninet networking hardware
- VMEbus
- Battery backed-up real-time clock
- 4 to 22 RS-232C serial ports
- Centronics compatible parallel port
- 5' terminal cable
- 90 day limited warranty
- Liaison Operation System (p-System IV.2 w/LAN software)
- Stride configuration utilities
- Stride Multiuser BIOS
- TeleTalker communications software.

FAMILY OPTIONS:

- 10M to 448M bytes hard disk storage
- MMU (memory management)
- FPU (hardware floating point unit)
- Memory expansion to 12M bytes
- A second 640K byte floppy drive
- Streaming ¼" tape drive
- Graphics package
- NOD wireless communications device
- Modem

OPERATING SYSTEMS:

- UNIX System V, CP/M-68K, p-System, Idris and RM/COS

LANGUAGES:

- Pascal, C, Modula-2, LISP, COBOL, FORTRAN 77, Basic, FORTH and APL

The Stride Product Catalog and Stride Software Directory contain additional information about the software and hardware products available for Stride computers. Call (702) 322-6868 to find the Stride dealer nearest you.

Stride, "Performance By Design" and NOD are trademarks of Stride Micro.

Stride Micro, 4905 Energy Way, Reno, NV 89502 (702) 322-6868, TWX 910-395-6073.