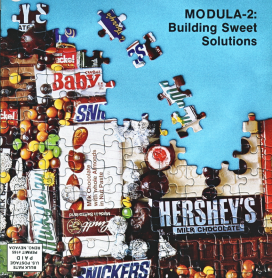


# STRIDE

## MODULA-2: Building Sweet Solutions



Milk Chocolate  
with Whole Almonds  
in Nut Paste

HERSHEY'S  
MILK CHOCOLATE

HERSHEY'S  
MILK CHOCOLATE

SNICKERS

NOVA 1000  
1000  
1000  
1000  
1000



## MOSYS THE MODULA-2 SYSTEM

At last, a native code Modula-2 system for Stride and Sage computers. MOSYS is complete and self-contained, the ideal program development environment. MOSYS includes compiler, screen editor, document processor and symbolic debugger.

Pricing — First copy .....	\$800
2 - 9th copy .....	\$400
10 - 99th copy .....	\$200

Educational site licenses and source licenses also available on attractive terms.

### Availability —

in the U.S. from:

Washington Information Ltd.  
5880 Reservoir Road  
Gonzales, Oregon 97133  
Phone: (503) 928-2582  
Contact: Mr. J. Stander

outside the U.S. from:

Robinson Systems Ltd  
Red Lion House  
St. Mary St., PAINSWICK  
Glos. GL5 8QZ  
Phone: (0452) 812888/812912  
Telex: 52432 DSA G  
Contact: Mr. S. Kirk

## STRIDE/SAGE-IBM PC BUS INTERFACE

■ NOW STRIDE/SAGE OWNERS CAN USE IBM PC EXPANSION BUS CARDS INCLUDING:

- IBM bit map graphics cards
- MICROSOFT mouse
- MICROM EYE video camera

■ ALL IBM CARDS RUN AT FULL BUS SPEED

■ LOW COST

- SAGE-IBM conversion card with software \$350
- STRIDE adapter board \$100
- expansion box for IBM cards \$400

■ HIGH PERFORMANCE SOFTWARE FULLY INTEGRATED WITH UCSD P-SYSTEM INCLUDED

- libraries to drive graphics, mouse, and videocamera from UCSD Pascal
- applications programs for painting, electronic design, printing & more
- SOURCE CODE FOR ALL SOFTWARE

■ FOR MORE INFORMATION CALL OR WRITE:

## SYMMETRIC RESEARCH

15 CENTRAL WALK, SUITE 19, KIRKLAND WA 98033 • 206-828-8980

TRADEMARKS: IBM PC, MICROSOFT, MICROM EYE, STRIDE/SAGE





## Inside in Stride

### Installation Blues

By Verlene Joyce Bohman

There is no better place to dream up ideas than your very own sunny porch with a view of the Sears Mountains and a cold drink in your hand. So, in anticipation of summer, Mr. Bohman (and myself) decided to replace the Page IV air home with a new Stride 440 and spend a little more time around the house. When the code fit a snag, we could recharge at the magazine or lean around on the lawn.

So began the new installation. We know the software would be a cinch—after all, Bill wrote the multuser system. However, quite a few systems. When we reached the house, we entered the large room as "the computer room" and got in recharging and a separate transformer. It had been fine for quite some time and seemed backordered.

We'd added a lot of equipment over the years.

As I had just rearranged a bathroom and made it my "tools room", I decided that moving my terminal there would be easier than finding a home for all the old computer stuff.

The cable for my terminal was going to be the only problem. I had gotten a new Stride terminal (Type 441-50) and I wasn't about to settle for less than 38.4K baud—also if my tool room was around the corner, through the kitchen and down the hall from the computer room.

I remembered that my friend, Jim Skane, at Teonic Engineering sells a kit to convert a cable-up to 4000 feet. It converts RS232-C signals to RS422 signals and consists of three cables and two small power packs, like those for calculators. The next day I telephoned Jim. He was just leaving on vacation so he loaned me the setup he used at Stride Park—with a 400-foot cable—to use until he got back and we could figure out a permanent solution.

That night I dropped the monitor cable (it weighs about 12 lbs) on the floor between my terminal and the computer to give it a test run. I plugged in the computer cable and connected its power pack. Then I plugged in the terminal cable-with its power pack. There are cables from computer to terminal cable. A couple of commands on the keyboard assured me that everything worked, just as advertised. That was the next part.

Two hours later, I had the desk in the back room cleared off enough for the ter-

minal I am using, sewing machine (sometimes I must learn to program one of those) and heater were moved to free up an electrical outlet. The terminal took its proud position on my little table and job was done.

Another hour later, I bought gray monitor cables—single-core shielding at the door frames down the hall and into the book room. It's why I'm wearing these. (Don't have the right color, it's not like the whole area.)

Actually, the distance is only about 80 feet. An RS-232 cable might work—especially if I was willing to go back to 19.2K or 9600 baud. (The EIA standard, which everyone usually ignores, says 30K tops, about 19.2K baud, not go only 90 feet.)

However, the installation question still remains: where am I going to run the new cables? My choices are few: 1) Start a new style of interior decorating and run it down the hallway across Year 12 (I don't do that) into the attic, although the attic ceiling is very low. 2) Run it under the floor and light the black walls (good). 3) Cut holes through the exterior walls, run through the attic and pray for no leaks. 4) Custom build a new facility with proper cabling—a perfect solution but expensive.

I'll probably have to fumble through the last idea (some thinking, why can't there be a good way to run cables now? Modern offices have some features to cope with the problem—drop ceilings, new floors, modular partitions, etc. Conventionalists have it done for a century, too. In recharging we should heed our homes in early COMBUs. But let's face it, most structures built today are just unimproved caves.

I can dream about a house with modular wall panels, that is, the panels pop off and move around as needed and are not just decorations but at this point it is only a dream. The rich can solve the problem by throwing money at it, but the rest of us are stuck with rooms designed in pre-OP.

We need a good installation method for standard buildings. Since the construction industry isn't working on it, the solution has to come from inside the electronic community. Realize there is some work going on to control appliances and heating systems using existing power wiring, but the bandwidth doesn't support a 38.4K baud terminal.

A few other experiments are being placed, although research, nothing you can buy. If you happen to have one, "real" one, drop me a line. But until a breakthrough happens, good luck battling back wires in your attic and baby banglers in the attic.

© 1987 by the author. All rights reserved. This article is published by Stride.



Editor: Verlene Bohman  
Advertising: Kate Harper

in Stride is a monthly publication of Stride News. Subscriptions are \$24 for one year. Purchase of a Stride computer includes a one-year subscription (upon receipt). Stride News is a fully computerized owner's registration card. Each issue are available for \$2.00 per copy, to suppliers. Requests for advertising forms, reprint permission, ad rates, bulk orders, or submission of press articles should be sent to the Editor, in Stride.

Stride: "Promotes the Design and AEC as Instruments of Social Change"

Postmaster: Change of address notices should be sent to:

Editor, in Stride  
Stride News  
6801 Energy Way  
Suite 100  
Boulder, CO 80501  
(303) 440-8888 (Mon—Friday)  
Fax: (303) 440-8875

Stride News Eastern Office  
12 New England Executive Park  
Suite 100  
Burlington, MA 01802  
(617) 279-8888

Stride News Southern Division  
1212 Madison Way  
Suite 400  
Dallas, TX 75240  
(214) 360-0275

Copyright 1987-1989 Stride News  
All rights reserved

# MIRAGE

## FASTEST OPERATING SYSTEM ON THE STRIDE

MIRAGE is a true multi-user, multi-tasking operating system, yet resides in only 45K ram. MIRAGE takes full advantage of the inherent power of the Motorola 68000 by allowing direct access to 16MB memory. Written entirely in 68000 assembler, execution speed is optimized. Features include device independence, built-in screen handling, multi-directory filing system, full interrupt-driven capability, background and immediate printing, and built-in provision for networking. A comprehensive set of re-entrant utilities is supplied, including a very powerful screen editor and 68000 macro assembler.

## MULTI-USER IN ONLY 45K RAM

Distributed  
and  
supported in U.S.A. by:

Internationally By:

**A<sup>P</sup>L ELECTRONICS**

DIVISION OF  
INTEGRATED ELECTRONIC TECHNOLOGIES  
P.O. BOX 176 PENNDDEL, PA 19647  
(215) 752-4180 TELEEX 756618 (BIT UD)

SAHARA SOFTWARE LIMITED  
DIVISION OF MICRO APL LTD  
UNIT 15 TIDEWAY INDUSTRIAL  
ESTATE 87 KURTLEIGH ST.  
LONDON SW8 5SP

I would like more information on MIRAGE and these supported languages:

- APL 68000
- SWEFTE COMPILED BASIC
- SWEFTE FORTRAN 77
- SWEFTE PASCAL
- TRAP RECORD MGT SYSTEM
- RANDOM WORD PROCESSOR

Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City/State/Zip \_\_\_\_\_  
Telephone \_\_\_\_\_  
 dealer  user

I would like a sales rep to contact me



# International

## Launching the 400 Series in the U.K.

The official launch in the United Kingdom of the Sindr 400 Series will be made by TCI Limited on May 10 at the London Press Centre. Plans for the event are being coordinated through Sindr's new support office at 28 Buckingham Gate. The new office will be shared jointly with TCI, Sindr Micro's exclusive distributor in the UK, and will replace the Regent Street office.

"The reorganization of our offices provides a better opportunity for communication with our overseas users," said Richard Kress, who recently assumed the duties and responsibilities of Vice President of International Operations. "This will give us an opportunity to support our targeted distribution, TCI, in a more efficient manner."

Located near Buckingham Palace, the office will be staffed by a Sindr Sales



Sindr's new UK office is located near Victoria, Buckingham Palace in London.

Represent. The office also will serve as the European base for Russ Patterson who was recently named Atlantic Region Sales Manager. In addition to coordinating distribution activities for Sindr distributors, the office will supply product literature, answer telephone and letter inquiries, handle invoice training seminars, conduct demon-

strations, and, of course, take orders for the Sindr 400 Series.

With user capability expanded to 16 users (22 on some operating systems), VMfiles for CRM applications and larger disk and memory capacity, the new series is being well received by European computer users. □

### UNIVERSAL TRUSS PROGRAM

••• 20+ years (less a decade old) proven program

••• 20+ code files, as large as 100,000 lines of code

••• Available in two major versions: The program handles plane, space, and curved trusses. It also handles curved members and curved joints.

••• 20+ file and element capability list

••• 20+ code files, as large as 100,000 lines of code

••• 20+ code files, as large as 100,000 lines of code

••• 20+ code files, as large as 100,000 lines of code

••• 20+ code files, as large as 100,000 lines of code

••• 20+ code files, as large as 100,000 lines of code

### UNIVERSAL DAM & WALL STABILITY ANALYSIS PROGRAM



FIG. 2-1



FIG. 2-2

- Practically all shapes with min. level
- Analysis numerically on:
  - Section cuts at various elevations
  - Earthquake analysis per International's uplift on base
- Complete report to file 8,500 24647

••• Direct Interactive Telephone Dial-up by End Users for FREE! Single Copies & FREE Trial Run

••• Price: Commercial Phone: 303 539-6688 303 539-6643 Tead Rate: 300

Phone: 303 539-6605 303 531-6144 Tead Rate: 1200

••• Protocol: Data 386 B Stop 816 I Parity On

••• Login ID: ABCDF180 ENDF180 GDEF180

••• TURNKEY System of Hardware/Software Available

## COMPREHENSIVE ANALYTIC ENGINEERING, INC.

STRUCTURES - PROGRAMMING - VIBRATION - FOUNDATION - CIVIL

247 Ridgewood Dr. Rocky Hill, Conn. 06067 Tel. 303 563-7386



to Ed Hayden

Bailey Hydraulics, the nation's largest supplier of hydraulic aircraft fittings, had just moved to a new 200,000-square-foot facility in Mansfield, Texas. The company had run successfully until then with manual systems, but paperwork was beginning to be a problem. Tom Bailey, the owner, had been thinking about a computer for some time but had some reservations about changing a winning formula.

Tom's own expression "if it ain't broke, don't fix it." That's sound advice for most companies. On the other hand, Bailey Hydraulics had grown to the point where some of the old systems were working at full capacity, particularly in the accounts receivable department.

Tom decided, with the move to the new location, the time had come to test a new system. They had the space and, if it didn't work, they could always go back to the old manual system. He remembered a software demonstration conducted by a company called i-Concepts that he had attended several months earlier which had impressed him. After a call to Lew Whitton, Vice President of the business systems division of i-Concepts, Tom was convinced that he had made the right decision.

After analyzing Bailey's operations, the engineer at i-Concepts suggested a multi-user system that would allow Tom's staff to perform various tasks simultaneously. It could easily be reconfigured to encompass all areas of the business including inventory control. While Tom had confidence in the advice, he still felt a little like a nervous bridegroom. He was ready to get married but he didn't feel very comfortable about it. He decided that he couldn't get hurt by going with an IBM PC/XT and the accounts receivable software that i-Concepts carried

## BOS Multiuser Accounting Software Is Ideal For Distributors

Fortunately for Tom and Bailey Hydraulics, the operating system and accounts receivable software that i-Concepts carried was BOS software. While they were sure that Tom and his staff would soon surpass the single user PC/XT, the people at i-Concepts felt that it's sometimes better to understand people, particularly when they haven't had a computer before, and then upgrade them when they become computer-wise enough to want more.

As Whitton said, "That's why we picked BOS software two years ago, and why it's still the only one we carry. It works exactly the same on over 75 different computers and upgrades to multiuser systems of very little extra cost. So, when Tom needed he only wanted one station and one application, that's all we sold him, even though he knew he would ultimately need more."

The installation and conversion went so smoothly that by the end of the first month, Donna Bell, Bailey's office manager, began to wonder why they weren't doing payables

on the computer as well. She had become frustrated with the limitations of a single user system. The most her staff used the computer, the more they wanted to use it and everybody seemed to want to use it at the same time. One obvious solution was to buy another XT but this would have required a substantial amount of double-checking to insure everyone was working with the same files. They also considered networking but felt that it would not provide the speed necessary for their present application.

Back Tom went to i-Concepts to see what could be done to add accounts payable software, upgrade his multiuser system and minimize the loss that he was sure he would have to take when his investment's almost new XT. To his delight, he discovered the iBOS/3 operating system he had could run the IBM PC/XT in a four-user system with a single software upgrade and the addition of inexpensive CRT's.



Discussing the installation of the new system are, from the left, Don Matthews, sales; Tom Bailey, owner and president of Bailey Hydraulics; Donna Bell, office manager; and Shelly Cooper, EDI.



Pictured working with their new Sage 8000 system are Betty Mena, left, and Betty Cooper, right, of Bailey Hydraulics.

There were some limitations inherent in the PC/XT, however. While the M800's operating system can run as many as 30 work stations at one time, the IBM XT will only support two serial ports. The addition of two CRT's, at very little extra cost, still looked like a good idea so Tom decided to add modeling as well as accounts payable software and bought an additional printer to take advantage of the powerful 8008 print spooler that came with his system.

**"... the dreaded inventory conversion was much easier than anyone anticipated."**

But he still had reservations about "total automation." His manual inventory system was functioning quite adequately. He wondered how could anyone who didn't know anything about his business create an inventory system that would work better than the one he had?

Another month passed and I-Concepts got another call.

"Here Tom/Bailey, I was just wondering if we do not put our inventory on a computer what would we lose out on right? Oh! And by the way, could we add the telephone sales operation so they would have their own CRT's and I'd like to have one in my office and we also need a terminal on the back desk. In fact, in most of our offices we need them. Maybe it would be better if you

just came out and told me what we needed."

The people at I-Concepts listened to all this with big smiles on their faces and to their credit, not once did anyone say, "I told you so!" They already had the system plans worked up, so they were able to give Tom a very quick response. The new system was a Sage II.

For sometime, the Sage II computer had impressed everyone at I-Concepts. This was the new system they offered Tom. Based on the powerful 80008 processor, it enhances the throughput of the M800's operating system. Six stations were standard on the machine. A 4 port serial expansion card from Integratec Technical Services (Aurora, Ill.) allowed 18 additional stations since the Sage II will support up to four of the cards. This would allow Tom to have up to 22 work stations with much greater capacity and speed than provided by the PC/XT.

Tom gave them the okay and the new system was ordered. When it arrived, once again Tom and his staff were delighted at the ease of conversion. The M800's software worked just as well on the Sage IV hardware as it had in the IBM and the data files were transferred as easily as transferring from one IBM PC to another.

Today eight months after delivery of the first piece of equipment, Bailey Hydraulics is fully automated and the Sage IV system is working like a charm. Donna Cooper can't believe how easily the conversion was done. She reports that the system was fully operational at the end of the first three months and that the dreaded inventory

conversion was much easier than anyone anticipated.

"While it was the hardest part of the whole change-over, it really went well," she said. "Involving all 18,000 inventory items took only four days with two people working full time. Actually counting the stock was the hardest part."

Tom and Betty of the staff were equally pleased. The smooth installation was a credit to Tom Bailey's business acumen and his positive attitude. Tom gave a great deal of credit to his wife Betty, who was very involved in the selection and the conversion process. Betty and her staff appreciated the project with enthusiasm and conviction. They had no one of the operators or computer phone operators during many conversions. All the people at I-Concepts worked closely with the Bailey staff. And in the words of Tom and Betty, "We didn't need much help, but when we did, they were always here right away."

The Sage IV computer and the 8008 software were also factors in the little saga. They felt that what they were supposed to, when they were supposed to, faster than anyone expected them to and at a good company, by least than other comparable systems.

Tom/Bailey said, "I wouldn't have done a project out when I fear the horror stories that some of my business associates have gone through with computer conversions. I feel very fortunate." □

**Dr. Peter D. Clayton is president of I-CONCEPTS, INC., 10000 WILSON ROAD, SUITE 1000, ANN ARBOR, MI 48106. PHONE: 313-963-0000. OFFICE FAX: 313-963-0001. SERVICE FAX: 313-963-0002. TELEX: 251100 I-CONCEPTS, INC., ANN ARBOR, MI 48106. CREDIT: 313-963-0001.**



**Richard F. Shockey**, of the University of Connecticut, is a co-author of "Software Engineering with Modula-2 and Ada."



**Bill Markham**, Sage 3, showed how to use Modula-2 in a research and development tool for integrating hardware and software systems.



**Randy Wirth**, of the Pacific Systems Group at DEC, is a co-author of the book "Software Builders Modula-2 Implementation Notes."



**Wallace Wirth**, of the Federal Institute of Technology, ETH, Switzerland, is a co-author of "Pascal, Modula, and Modula-2."



**Jim Henney**, a PULL software consultant and writer, teaches Modula-2 seminars and is writing real-time applications in Modula-2.



**Tom DeMarco**, of the Atlantic Systems Group, wrote the book "Programming with 'Coding Your System'."

## The Modula-2 Mindset

Featured authors, computer specialists who share a common thought: Modula-2 is the ideal for a programming language. Each has invested considerable effort in promoting Modula-2, some with no immediate financial return: Modula-2 is not supported by any major software group in the mainframe way Ada is supported by the US Department of Defense. "While Modula-2 is now gaining more acceptance as good computers become available, it was pretty much out there at first."

Modula-2 was created in 1987 by Professor Niklaus Wirth, of the Federal Institute of Technology at ETH, Switzerland's equivalent of MIT, who is also the originator of Pascal, named as the successor to Pascal. Modula-2 is different enough that considering it a mere Pascal extension is missing the point. With introduced concepts new and important concepts that are not just "New" to Pascal. These concepts are the key to thinking in the mindset of Modula-2.

### Check Out The Details

For details on the language, various articles and books are available. MODULA-2 is a Modula-2 users group, also formed with the intention of providing feedback with up-to-date information on the status of the standardization efforts currently in progress. Magazines such as the Journal of Pascal, Ada and Modula-2 contribute to the discussion of issues on the language. (Give the editor at the end of this article for addresses.)

The Modula-2 books go more than references to the form of the language. They

offer valuable philosophical insights into the purpose of the more complex constructs. This was needed since some ideas, such as Modules, seemed simple at first but really require a good discussion of the implications involved for effective use. How much importance should be placed on these new constructs? The answer may be found in observing how users relate to their languages.

### Artificial vs Natural Languages

In the 1930s, B. Sapir, the famous linguist, and S.L. Worf, linguist and engineer, proposed the Sapir-Worf hypothesis which contends that the structure of a language determines the consciousness of human thought. Loglan, the artificial "log" language, was created by James Cooke Brown to test this hypothesis. As a research tool it had promise, but as a full language it fell somewhat short. Research on Loglan continues today through the Loglan Institute in Gainesville, Florida. But perhaps the Sapir-Worf hypothesis can be better validated by comparing other more popular artificial languages: FORTRAN, BASIC, Pascal, COBOL, Modula-2, etc.

This may be stretching a point, as technically, computer languages are not natural spoken languages. But evidence suggests that users relate in similar ways to both computer and natural languages. Consider the observation by Richard Driess in his foreword to Ed Joyce's *Modula-2: A Developer's Guide and Support Manual*, 1986.

"From my impressions, computer users and programmers usually regard the selection of a programming language as similar to the selection of a second natural language, such as German, French, or Italian. People usually choose a second natural language based on factors such as how easy the language is to learn, historical areas of use, how widespread a particular language is, where they want to travel. How politically significant the users of the language are, and other reasons quite unrelated to the language's basic structure. Rarely are natural languages discussed and evaluated according to their primary purpose, which is the expression of an idea in the communication of a thought. . . . Selection to people (programmers) face the real issue, which is: Is language A better for the creation of efficient, readable, and debuggable programs than language B?"

Tom DeMarco, of the Atlantic Systems Group, also offers the comment on the selection of the problem: "There is a curious religious phenomenon involved. People's adherence to a language is not based on the nature of the language. It is based on the semantics of thought that cause persons to be Jewish, Irish, a Mission or a Catholic. I had to do with their upbringing, training and values. Originally, perhaps, it came from a set of sensitive rational decisions. But these were made so long ago and they have become so committed to them over time that the issues can't be re-examined anymore. So languages are not something that people tend to learn based on the sheer value of the language."



**Richard E. Shinn**, editor in chief of *Flow*, *Ask and Answer* 2, is co-author of "Module 2: A Software Development Process."



**Tom Wobeser**, of IBM Corporation, works on the design and implementation of graphics in Module 2.



**Lewis A. Phipps**, president of Phipps Intelligence, works on databases and data access in "Introduction to Database Science and Module 2."



**Roger Swenson**, of PDS, is an implementer of the early Visual Pascal and was lead of project Dynamic Module Implementation in Module 2.



**Richard Shinn**, vice president of Intel, is Director of the Intel Software Development and President of the Module 2 Corporation.



**Brian Kirk**, of Robinson Systems, offers *ASCL*, a real-time operating system environment for Module 2.

## Thoughts from Users

The complex task of getting programmers to switch to another language may be tough. They are not really thinking about better programming but reacting with a gut-level defensive posture based on which language they are currently using. Still, there may be hope. After all, programmers are some of the more logical people around. Module 2 lists states in his article in *Byte* (Aug. 84): "It is precisely the ability to think in both object-oriented abstractions from the framework of a compiled programmer." Again, we come back to how we think, not the structure of the language.

### Module 2 Concepts

What are the changes in thinking required to program optimally in Module 2? Randy Bush, who is heavily involved in the British Standards Institute's effort to standardize Module 2, uses a simple analogy to compare languages: "The difference between Module 2 and other languages is like the difference between clay and Legos. With the plastic interlocking parts of the Legos set, I have a constrained environment with which I can build reasonable structures pretty much the way I want. If I put it together in a known, reliable, reproducible and verifiable library. On the other hand, I can also build reasonable small structures with clay in any form I want. However, it is not controlled, not checked. Otherwise what happens when you build something large with clay—a scoop? The A.S.D. Schwartz department store has about an 800-square-foot

display built out of Legos. It could never be built out of clay."

Jon Bundy, consultant and lecturer, has taught many classes in Pascal and Module 2 and offers the following observations: "Most Pascal programmers have never heard of Object-Oriented Design and they learn about Module 2 with Object Oriented Design, the programming problem is first expressed in spoken or written English and then broken down into nouns and verbs. The nouns often represent abstract programming 'objects,' while the verbs represent the abstract transactions which must be performed on these objects. The result is a very high-level abstraction that is ready for translation into a program in Module 2, the translation is fairly direct, with objects becoming data structures and verbs becoming procedures, all related within a Module. In Pascal the data types must be far away (usually from the 'verb') for this to be done easily, although I can be lenient.

"FORTRAN programmers have even more difficulty adjusting to Module 2, since FORTRAN lacks complex data structures built as records and pointers, and type checking is not available to police the use of the data. FORTRAN programmers often explicitly object to the idea of type checking since it requires more rigorous application of programming concepts. The lack of complex data structures makes the introduction of Object Oriented Design methodologies more difficult than with a Pascal program.

"Unlike FORTRAN, the language 'C'

offers rich data structures, although lacks type checking. 'C' programmers too often dislike rigorous type checking, arguing that the security gained is not worth the effort involved.

"BASIC users are especially handicapped because of the lack of parameterized procedures in the original language. This lack prevents them from designing complex programs in a modular fashion, and they are often unaccustomed to many modern programming practices. While some newer BASICs contain suboutine-like features such as parameter passing, type checking and Modules are absent.

"Perhaps none of these features makes a difference when designing a small program, but we have found them to be essential on large projects."

Another key concept that researchers to Module 2 may stumble over is that the application (definition) of the program and its implementation (how it does) are less related but different things. Richard S. Munn, in his book, *MODULE 2: A Software Development Approach*, explains abstraction: "A major approach to process solving abstraction, which involves model building. . . . Typically, abstractions deal with objects and operations on the objects. High-level abstractions are not concerned with implementation details. For example, one may understand the concept of an automobile brake without caring whether it is a drum brake or a disc brake. The abstraction concept of 'braking' is independent of its implementation. Our ability separate the high-level operational from

# Module-2

use to view a system from the implementor's point of view. . . . allows us to understand complex systems."

The author of the meter was expressed to Alfred North Whitehead, 20th-century philosopher (Feb. 15, 1861—Dec. 30, 1947): "Civilization advances to the extent that we can do complicated things without thinking about them."

DeMoss agrees. "The telephone system is a good example of this. If you had to know everything about the complete telephone system in order to make a call—dial tone, carrier, electronic switching, etc., you'd never get the call made. The trick is to avoid a confrontation with complexity. You must encapsulate the low-level processes. People only need to know how to dial, how to use a directory and how to talk. The programmer needs this encapsulation. He should have the same flexibility to check for components as telephone engineers do. They don't have to design each of the details over and over again; they buy them from a catalog."

## It's Not Perfect

A programmer who first looks at Module-2 may express the kind of pain and doubt shown in articles such as one written by David V. McNeil in the December 1984 ACM SIGNUM. McNeil expressed several concerns, many of which were of the "big-pick" type and few of which demonstrated the differences in philosophy between those who see Module-2 and those on the outside.

McNeil claims that Module-2 is "Not a General-Purpose Language. The lack of convenient I/O is only the first evidence that Module-2 is not the general-purpose language that many of its promoters claim it to be. . . . A general-purpose language includes nesting-logic strings and convenient phrase-logic string-procedures. These are not included in Module-2 or its standard Module library, although they can be simulated—much less efficiently—in Module-2. . . . Most commercial extensions to Pascal include BCD, long integers, or alternatives with their digits. Module neither includes these, nor even precisely defines the basic types that it does include. Again, although these types can be simulated in Module, an activity that sits outside the point of a program to be written. . . . Its set constructor is not precisely defined, but is supposed to be small, but several arbitrary sets of characters, for example."

On the other hand, Module includes a lot of features for systems programming, such

as hexadecimal and octal I/O, direct memory addressing, low-level I/O and so on. Considering that we've not been put in it, and what has been put in it, it is plain that Module is a systems-programming language and not a general-purpose language."

Here, we see a dislike for so much of the content of Module, yet so much of the fact that functions that were inherent in other languages are now moved outside of the definition of the language proper into libraries. Perhaps McNeil's fear is that the programmer may be forced to write those libraries himself; this idea within the Module-2 community seems to be quite different. Someone else is going to write these routines and offer the code in a library, saving the application programmer all the trouble. If, however, the library doesn't do what is needed, the programmer has the option not available in other languages of writing it.

A real life case in point is the work being done by the Pacific Systems Group (PSG). For a real-time application they needed an asynchronous version of the standard I/O library. Essentially, they kept the standard library I/O definition, but rewrote the code to be asynchronous. Flawed, perhaps, but did not find that the standard library definition seriously inhibited our implementation. However, the documentation of the definition requires considerable effort in understanding.

Perhaps Module-2 is not an "easy" language if novices and pros alike are confused on the first round with I/O. McNeil also admitted the confusion and his concern that Module-2 is "Not a Teaching Language. The systems-programming orientation of the language, the lack of high-level I/O syntax, and the confusing I/O and output-related evidence that Module-2 is not a teaching language. . . . The very simplest example program should include some I/O, but it must, therefore, also import I/O procedures, from one of the standard Modules. To be honest and precise, this necessitates a discussion of Modules, separate-compilation, definition and implementation pairs, export lists, qualification and import lists. Although a program that says add two integers and displays the sum."

The effective size of the language is also very great. It includes 40 reserved words, 26 operators and delimiters and 26 standard identifiers, for a total vocabulary of 102—without I/O or trigonometric functions. The standard libraries add 130 more identifiers for a total of 230, although some implementations provide yet 200 more! By contrast, Pascal's entire vocabulary numbers 78 operators and identifiers. So

Module-2 is too large, too cumbersome, and necessarily oriented to be a convenient teaching language."

Perhaps there is not better for teaching, although it really depends on the subject being taught. Surely, a good "teaching language" is not one which is convenient for the instructor to teach, but one that deals with how to think about a programming problem and how to solve it. The goal of computer science teaching should not be to have things easy for the teacher of student, but to graduate students who can handle real-life programming jobs.

## A Commercial Language

The heated discussions that occur between programmers of any new language and programmers of other languages usually determine how these larger concerns are a "big-pick" battle over syntax differences. Spiteful discussions can be quite exciting and entertaining. However, it really makes little difference to program a design if a particular command that always has a semi-colon after it. A change in syntax will always involve a short frustrating period of time when programmers must train their fingers to type the new sentences. A more important issue concerns how a new language is seen by industry.

Bill Bohman, Super-Software International, sees Module-2 extremely. He observes, "The advertisement computer's such as the 68000 native code implementation undoubtedly limit limits of Science Computer Systems promises an attractive environment for commercial applications as the code generator is extremely fast and clean." The only word here is **fast** (and Bohman's work is in hardware drivers and graphics).

Security was a concern expressed by almost all of the users. For good security they felt that there are missing requirements—separate compilation as in Fortran, and type checking as in Pascal. Pascal extensions such as UCSD units are awkward, and in fact, not secure. (This is a point difficult to understand without a technical discussion.) Going to separately compiled Modules is a major step toward security, permitting clean structure for good design in both data records and code.

Stan Kim, of Bohman Systems, which markets the MCV's Module-2 operating system, stresses the advantages of Module-2. In the September, 1984 issue of *Microprocessors and Microsystems*, he says, "Computers have evolved at a steady rate over the past 20 years with their components increasing in complexity by a fac-

50 of 10 every five years but with costs continuously decreasing. However, to such progress has been achieved with software ... productivity has stagnated. Finally, because it is 800 labour intensive and little use is made of previously developed software-components ... Modula-2 is the most important feature of the Modula system is that for the first time it opens the way to increased software productivity by introducing the concept of a library of software components packaged in the form of so-called Modules."

IBM is also excited about the change that a large set of commercially available Modules may make to the industry. "First, Modula-2 and Mod share an important concept which is the idea of encapsulation, variously called Modules, units, packages, etc. The next few years will see a major undertaking in the building and sharing of huge component libraries. An example of Telos which markets an Ada environment, proposed a forecast that within 10 years the major Ada decisions which are made today on a program basis will be made on a Module basis. A design team will build the program themselves, then break it up into Modules, some of which they'll buy outside, and some of which they'll write internally. Typically, a small Module will be only 20 lines or so long. They'll buy packages of Modules. This is quite different than the decisions being made now, where the choice is either buy the program outright (say \$5,000 lines of code) or write the entire thing from scratch."

If this, Dattner and Bowles are right, the pricing structure of programs may also change drastically. Obviously, the cost of a small Module can be priced reasonably compared to an entire special program. Shorter development times also means lower company costs.

## Proving it

The superiority of any one language may not, however, remain a pure intellectual speculation. In 1984, the Atlantic Systems Guild sponsored the "Coding War Games" - a tournament in which different fields were given the same problem to solve in a program in the language of their choice. The '84 results were inconclusive as to language superiority but did provide some interesting feedback on programming working conditions. In 1984, only one Modula-2 programmer participated, but the sponsor Hope is enthusiastically positive and experienced Modula-2 programmers to the '86 games. Studies such as this will go further towards proving the worth of a language than any of the abstract system set for

## Conclusion

The inclusion of computer languages is not complete with Modula-2. But many alternative codes and other code ways to reach the process of program solving with tools that will do the job. The search for better may prove painful, but assuming a "Modula-2 Mindset" may also prove productive. □

## Reference

Subscription Department (212) 680-6034  
Journal of Pascal, Ada and Modula-2, John Wiley & Sons, 605 3rd Ave., New York, NY 10158

MODULA Modula 2 Users' Association c/o Pacific Systems Group P.O. Box 9177 Palo Alto, CA 94303

Modula Corporation (801) 226-8938/8940 N. University Ave., Provo, UT 84604 (Also MO for Macintosh & IBM-PC, ETC)

Modula Research Int. (981) 3375-1520 1673 West 800 North, Provo, UT 84604 (Also MO for IBM-PC, ETC, Source: ETC) (the original ETC version.)

LOGICAL Institute Inc. (884) 371-9574 1701 NE 75 St., Gainesville, FL 32607

Maritime Information Ltd. (502) 529-2562 6850 Reservoir Road, Corvallis, OR 97331 (Also Modula (the 802 Operating System for the Slide Page: 686/500) )

Scientific Computer Systems (206) 885-9500 14857 9th, 31st Coast, Redmond, WA 98052 (Also 802 68660 compiler for Slide Page: 686)

## Books

Brown, J.C. 1975 *Logen, A Logical Language*. Institute The Logen Institute.

Greene, R. 1984 *Modula-2 for Pascal Programmers*. New York: Springer-Verlag.

Joyce, E. 1980 *Modula-2: A Beginner's Guide and Shipped Manual*. Reading: Addison-Wesley.

Prich, Lewis 1988 *Introduction to Computer Science with Modula-2*. New York: John Wiley & Sons.

Rain, E. 1981 *Language*. New York: Harcourt Brace.

Reiser, S.H. 1985 *Modula-2 with Self-Structure*. Belmont: Wadsworth.

Wherry, S.L. 1978 *Language, Thought and Reality*. Edited by J.D. Carol. Cambridge: MIT Press.

Winters, R. and R. Shivers. 1984 *Software Engineering with Modula-2 and Ada*. New York: John Wiley & Sons.

Wormer, R.S. and G.A. Ford. 1985 *Modula-2, A Software Approach*. New York: John Wiley & Sons.

Worm, M. 1983 *Programming in Modula-2*. New York: Springer-Verlag.

## Beta Test Offer!

**Modula-2  
68000  
Native Code  
Compiler**

**\$500**

A beta test version of a 68000 native code compiler is now available for (Scientific) Sage computers with minimum disk storage and 1M byte of RAM.

The offer includes documentation and release. (90-90 days from April 15, 1985)

Scientific Computer Systems Corp.  
14857 9th, 31st Coast  
Redmond, WA 98052  
(206) 885-9500



By Ramon C. Jorjuez

Do you ever need a character that doesn't exist on your printer? Perhaps it was a special mathematical symbol or maybe you needed a character for a non-English lan-

## Printer Package Provides Custom Fonts

guage. Unfortunately, many word processors today ignore the needs users have to create and develop special characters or symbols for their work.

As needed special character sets in my own work, I developed P.U.P., a Printer Utility Package for the Datatec 80 and 85 printers at the Bronx College. There are actually two different versions of the program, one for each type of printer.

In order to produce a good document, P.U.P. also had to have text formatter features such as underlining, superscripting and subscripting. It supports five font styles: a fixed Data Processing Quality (DPQ), Correspondence Quality (CQ), Enhanced, Emphasized and Bold. Six print sizes for the Datatec can be used: 10, 12, 14, 15, 18 and 24 characters per inch.

P.U.P. also supports true proportional spacing. This is an important feature used to adjust the spacing between characters

to fill the lines across the page. The right edge is no longer ragged.

Other methods used to do this, such as fitting between equalized characters, do not produce the same clean look that true proportional spacing has. Most of these methods generate "bars" of large white spaces running through the copy, even on very large paragraphs. P.U.P.'s output fills the lines evenly.

The user also has control over the amount of spacing used in the text character mesh. This is handy if you need to expand or expand text to fit a certain sized area.

### Simple to Use

To create a document, first compose a regular text file using any editor under the S-System that creates a ".TEXT" file. The standard editor (SOS), will allow you to call it as usual. (With Word\*, the suffix must be set to ".TEXT" instead of ".Q"). As pieces in the text where the special character is a feature is needed, put an underline, a \*, followed by the printer directive. For example, \* **BOLD** will cause all of the text after it to be printed in BOLD style.

To print out the file, simply type **B**, then **PRINT** then give it the file name.

### Creating a Font

With the font routines, you can define hundreds of special characters if you see design and print them in any size or style including Correspondence Quality. The custom CQ characters are generated in four-pass printing mode. A custom font (see last line of the sample) has already been generated and comes with the package.

The special characters defined by the user in a standard text file are called a "font" (as in typesetting). The fonts are easy to set up, the definition being a simple list of stars and "Q"s. Two definitions are used per character. One is for the Data Processing Quality mode and another for Correspondence Quality mode.

Each character in the special text file is given a name which is used to print it out. For example, **alpha** would define the Greek character alpha.

A level indicator specifies where the character sits on the line. **ASCENDER** indicates it sits squarely on the line, while **DESCENDER** indicates it drops slightly below the line. For example, the lower-case letter **g** is a descender character and the letter **B** is an ascender character.

Features included: underlining, superscripts, subscripts (including descenders), padding, 5 print styles:

data processing quality  
correspondence quality  
enhanced  
emphasized  
bold

6 print sizes:

10 CPI  
12 CPI  
14 CPI  
15 CPI  
18 CPI  
24 CPI

arbitrary line spacing, and proportional spacing (left/right margins). In addition, you can add new characters of your own design to the character set. A complete break font is supplied!

alpha beta gamma delta epsilon zeta eta theta iota



# Utilities

The following is an example of the Correspondence Quality test definition of the Greek letter alpha:

```
alpha
*****
CG Pattern
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

When you type the pattern in on your terminal, make the characters all different colors. The terminal display leaves more room between lines than the pattern will. When you actually print from the characters will be stored and later, it is a good idea to run a small file that will print your test characters to test how they look and change them if need be.

Once you have designed the character, a program called TYPERIT converts it to the FONT.DATA file. Another program called LISTTYPE will convert the FONT.DATA file back to a text file, being able to regenerate the test files very quickly in case you may have lost the original file or not sure which version you are using.

The Greeks permits hold up to 96 special characters at one time. However, with P.L.P., you can have hundreds of characters as the program keeps track of which

ones are loaded in the pattern and will dynamically loads new ones when you need it. No special commands are needed at any time.

## Summary

P.L.P. is intended as a word-processing add-on. It works well for applications such as letter writing, reports and small manuscripts. Because it uses any standard system font file, its advantage is that it will work with a variety of word processors. □

*Dr. Mike Rowland, Author has a B.S. in Engineering Physics from the University of Colorado. He is currently working for Lawrence Berkeley Laboratory. He is interested in optics and laser technology. P.L.P. is distributed by P.L.P. as a courtesy to the users of the P.L.P. system. Contact P.L.P. at the address listed below. Dr. Rowland, 220000*

## FourByteForth

*Address Your Entire Speed!*

### 32 Bit Forth for Stride/Sage Computers

Screen Editor

65000 Assembler

Fast Compilation

125 screens/minute

Debugging Tools

Disassembler/Disassembler

Fast Execution

Byte Slave 18 sec/10 lines

available as standalone system or under CP/M/MS-DOS

### Software Architects

1800 Green  
Berkeley, CA 94702  
(415) 848-2185

The Most Powerful Language  
for the Stride Computer  
is also the Cheapest!

## \$250

plus \$8 shipping  
CA residents add tax



## RELATIONAL CONCURRENT MULTI-USER DATA BASE

- Network Multi-User Support (not yet independent)
- Individual Record Lockout
- Automatic Relational File Access
- Unlimited Number of Fields, Files, and Records
- Full Base Comparison
- PASCAL Program Access to Data
- Application Developer Package
- Creatively Inquiries Routines

For more information contact:  
IOTC Inc.  
P.O. Box 1085  
Lanette, NY 13095  
(607) 724-5888



**IOTC Inc.**

# Customer Service

by Pepper Lacey

## Strike vs. Sage Test

If you're writing a program that must run on both Sage and Strike 486 Series computers and the program needs to know which computer it is, you can use the BIOS configuration control (CMOS) to read CMOS memory. From the p-System, this is a simple LMBREAD:

```
HEXADDR 00000000;
VAL 1400 0000 0000;
CMOS 0
CMOS 1
CMOS 1400000000 0011 0000;
CMOS 1400000000 0011 0000 0000;
CMOS 1400000000 0011 0000 0000;
CMOS 1400000000 0011 0000 0000;
CMOS 1400000000 0011 0000 0000;
CMOS 1400000000 0011 0000 0000;
```

Note that the Sage does not return any information to the string "0" as the BIOS call is not defined for a Sage. Therefore, its important to initialize "0" so that the program is not checking whatever was in memory last.

For CP/M-86K IBM PC/XT and C, use the LMBREAD call in the same way as the LMBREAD call in the example above. Under fortran (LMB, BIOS access) read becomes through installed device drivers, so the type of check is not recommended.

Refer to pages 181 of the September 1984 Owner's manual or page 617 of the more recent 1985 Owner's manual for information on reading this area in assembly code.

## p-System Print Spooler

The single user print spooler in the p-System can be enabled by using the file SPOOLER(COOKIE) in the local partition volume. SYSTEMPASCAL must include the SPOOLER(COOKIE) (you can substitute the LIBRARY CODE program, but it is normally in there.) Use the SETUP(COOKIE) program to turn on the **HAZ SPOOLING** file in SYSTEMSPOOLING. Normally that file is OFF. Then locate SPOOLER and tell it what file you wish to print.

Remember the spooler is CPU intensive as it constantly "looks around" for something to print. It also uses about 2K of your memory space. See page 140 of the p-System manual for more information.

The Laser PRINTSERVER is available for systems where SYSTEMPASCAL has been setup for networking. (The **Setup TERMINAL** program will do this for you.) The network utility **SHARE** might be run on all users who need to access the printer

through the PRINTSERVER. When a user sends a file to the printer it is stored on disk until the printer is free, then it is printed out. A header in front of the file identifies whose output it is. Note that the printer user should have a fairly large hard disk area with which to work. See page 318 of the p-System manual for more information.

## Using PRINTP

If you can't get the **printp** routine to work under IBM C, it is most likely because you don't include **cmh.h**. The following shows how to include it in your program:

```
#include <cmh.h>
#include <stdio.h>
extern
int printp(char *filename);
```

## Life after Coca Cola

We've just found out how to remove Coca Cola from a cassette. (A junior sister left it all over the only copy of a critical source cassette.) We did this once with success but it not guaranteed the method. Blast and clean the cassette is abused until most of the residue seems to be washed away. Do not touch or scrub the cassette surface. (Dissolve residue goes, carefully remove the cassette from the jacket, inspect the surface of the cassette and clean it several times. Let it dry. Take a NEW cassette and carefully remove its jacket. Put the old cassette in the new jacket. Tape the edges together. Be careful when you try it. (Copy immediately and blow it out.)

## Stand-alone Loader

On page 619 of Volume 1 of the September, 1984 printing of Strike Owner's manual (page 640 of the February, 1985 printing), is a discussion of **ARMLOAD**, a program which allows you to load an assembly program as a stand-alone program. The documentation projects to tell you that the load address must be 536 bytes more than the target address. (536 decimal = 21A hex). Thus, a target address of 28450 would require a load address of 2871A.

The difference in the addresses is needed because the actual code in a p-System code file is located after a 512 byte header and a 26 byte code file information record.

If you run the program **COMPRESS** on the file before loading, the code information will be stripped off and the target address will be the same as the load address. □

**How do I transfer files from a foreign operating system to IBM/OS/2?**

Transfer the IBM/OS/2 system as a file first. This is discussed in the "user" file (a documentation file) located in the directory `DP_.../USER`.

**Should the system LED blink when the DISKSERVER for Liaison networking is running?**

Yes, network tasks occur at various intervals.

**What's a "Mail Impairable Error" and why am I getting it when I compile?**

Your program is trying to compile with the SYSTEM.PROCAL and SYSTEM.LIBRARY units. In the PC-X releases of the p-System, these no longer have interface versions for the compiler to use because Softech split the development tools into a different package. Change your program to use the correct units. They can be found on the LITELITE diskette in the LIB25 subdir.

**Why are two routines turned off in the Windows Benchmark published in the January 1988 issue of its 3000?**

These two routines are rarely used, turned on in the benchmark for historical reasons. But, some of the early machines could not do them. Since then, almost every hardware test you will see has been taken with those two routines turned off.

**How do I turn off the "spooler" when I type a key on my 386/486 (WYSY WY-80) terminal?**

Type `SHIFT+ENTER` and the keys will no longer make the "spooler" type it again if you would like it back on.

**How do I read a floppy under IBM/OS/2?**

Do a LABEL command. When it asks for a device give it `0000`. Then it will ask for a volume, give it the volume name. If you are not sure of the volume, type up a `FRT/004`. This will give you a `DIR/004` disk command. To show the files on the volume, do a `LIST` volume.



Steve Grubick



Judy McCullough

**Steve Grubick**, hardware engineer, is responsible for many 386 products, among them the 12 MHz option on the 400 Series. Calling himself a "hardware software fan-dyman," Steve also writes test software.

While at the University of Nevada Reno, he combined a chemical engineering major with computer science courses. Electronics was only a hobby at first. Then he became a senior technician at Thomas Barton Instruments, attached to Barry Nevada and finally joined 386co.

Steve likes to spend his time cruising the Nevada highways on his Honda 750. For a change of scenery, he rides to the wine country of Northern California, the beautiful Napa Valley. Another way Steve likes to travel is by air. He is studying for a pilot's license and eventually an instructor license. In a spare room at home, Grubick conducts his interest in chemical engineering. At times, a fascination for the science of the stars and planets draws him to astronomy.

He likes to spend his time heavily on **Judy McCullough**, executive assistant to Richard Kras, Vice President of Finance, Administration and International Operations. Judy's studies throughout the day are many and varied. She provides administrative assistance to Paul Coleman, our president and aids Carl Hedges, our Controller, with the preparation of financial reports and summaries. The rest of us get a helping hand and a smile when we need it.

Before coming to 386co, McCullough was the personnel coordinator at Sparks Family Hospital, marketing administrator at International Roles and a legal secretary. With this background, she was made responsible for the important job of maintaining sensitive legal documents, handling all of the filing, registration, banking and action implementation herself.

Oh, weekends, Judy and her husband, Duane, relax and get landscaping in the desert. She says she's a good butler's better.

## New Product Releases

### Data Flow Diagrams

**BURLEO** is an interactive program for drawing data flow diagrams. Such diagrams are an important development technique for structured programs. For Software P.O. Box 712, Quarts, CA 91010 (916) 354-6762

### Communications Software

The popular **PARADISE** communications package is now ported to the 386/486 Series computers. New features include **MODEM** and **CRC-16** protocols, support for high-speed connections to mainframes and a revised (order) reference manual. Available for sale in 13 major p-System versions. Amtek Communications, 510 E. Third St., Tempe City, AZ 85284 (602) 947-0004

### Data Base Record Manager

**PRACIT DBRM** provides access to and manipulation of records in a database. Its interface is a library of C programming language functions plus high level tools. MOM Computers P.O. Box 1988, John's Landing 9500 Tal (916) 656-1544 Telex: 45-9297 5A

### High Speed Data Link

This link provides inexpensive reliable data transmission up to 4096 feet at rates to 15,880,000 bps. Custom quotes are available. Tecon Engineering P.O. Box 11386, Reno, NV 89510.

### 286 Emulator

The **MOORHAVE** environment allows real-time emulation of an 286 construction over the full 8M byte addressing range of the 286. It runs the average CP/M-80 program at three speed of a 1 1/2 MHz 286. **MOORHAVE** GmbH, Postfach 716560, D-8000 München 71, West Germany 089 760047

### Pass to C Translator

This Pascal to IBM C translator is fast within p2 and will run under CP/M-80 or p286. It handles nested procedures, intrinsic functions, separately compiled units, and types including long integers and p-System DAT pointers. Available only as licensed source. Maritime Software Associates, 4688 Reservoir Road, Corvallis, OR 97331 (503) 829-2552 or 829-8822

# RM/COS /COBOL

*For the Stride 400 Series*

Designed specifically for the commercial environment, RM/COS provides a true 32-bit performance advantage over general purpose operating systems such as UNIX. RM/COS features include:

- Identical software user interface among UNIX systems.
- Reduced memory requirements — about 10 Kbytes.
- Rich control language.
- Reduced level linking.
- File system streams data directly in use of page frames.
- Interactive user input.
- Extensive multi-tasking commercial operating system.
- Real system time output.
- Dynamic file allocation.
- Write update status.
- Multi-level hierarchical directories.

In use since 1975, now with more than 25,000 installed sites, RM/COBOL provides all the features commonly required for a true small system computing. RM/COBOL is a COSA certified open file implementation of the RM/COBOL 14 COBOL standard.

- Full Level 2 relative and indexed file access methods are supported.
- Multi-Key index files may have 16 Alternate Keys, including Duplicates.
- Record and file level locking supports multiple activity and shared files.
- Compact output code, segmentation and dynamic memory management support the development and execution of very large applications.
- Applications written in RM/COBOL can migrate, with no changes at all to source code or object code, to all the environments that support RM/COBOL.

**STRIDE**



Stride Microsystems  
4800 Energy Valley  
Farms, San Jose, CA  
(415) 352-6600  
Telex: 940-366-6075

Circle 17 on Reader Service  
Circle 18 on Reader Service  
are trademarks of Stride Microsystems.