in STRIDE

December 1985 – Vol I, Issue 3

CONTENTS:

1985 – A Stride Forward

New Books

BIOS & UTIL Versions

OS INFO WORDS

UNIX System Accounting

Linking SVS To C

Modula–2 & C Seminar

Square Root Revisited

New Aladin Database

Porting Via UNIX C

tech notes

1986

Resolutions

— 1985 —
A Stride Forward

by Verlene Joyce Bonham

As we approach year end in this headlong, fast–paced, world of computers, it's probably a good time to reflect on past events. Looking back, it's amazing to realize that Stride Micro survived while so many other companies in our business did not. Did we do something right?

In truth, we did not completely escape the slump. The year started strong on the positive introduction of the Stride 400 Series the previous Fall. Being named "Product of the Year" by *Electronic Products* magazine certainly didn't hurt. In fact, Stride was able to secure a major financial investment during this Spring blossoming. By summer, we began to feel the effects of the general downturn that hit the rest of the computer industry. It wasn't until the leaves began to fall that we started seeing an upturn.

At this time last year, we were concentrating on the p–System market. Thus the realization that Softech MicroSystems was in trouble had us concerned. When they put their products up for sale in mid–year, we were already seeing our p–System sales diminish with developers looking toward other alternatives. Fortunately, things had a happy ending.

We accelerated our UNIX development projects in time for autumn. Our overnight popularity as a powerful UNIX system for both developers and business applications has been a welcome bonus. Additionally, the winter brought new owners for the p–System.

Pecan Software is already off to a great start. They're busily at work fixing bugs, adding features and even spending a bundle on national magazine advertising (a "Turbo Who?" campaign to sell UCSD Pascal against Turbo Pascal). The resulting new energy being pumped into the p–System market is a good sign of better times.

Apart from the product, the year still featured many transitions. In 1984 the International market was our bright spot. But, offshore, the beginning of 1985 was chaotic. We had new distributors, legal entanglements, changing software demands, and a U.S. dollar at record levels. When Rick Kriss took over as vice president of International Operations, things were a mess. But soon he and marketing manager, Keeli Patterson, were on the road to the UK, Germany, India, Japan, etc. spreading good cheer. Now, heading into 1986, the international group is again fueling our optimism. We've obtained a powerful manufacturing partner in the United Kingdom and are once again seeing Strides placed in new exciting global markets.

There were also some rocky times here at the Reno headquarters. The high–energy methods that propelled Stride from a basement to corporate headquarters and regional offices in just four years were showing signs of fatigue. Early in the year, our controller, Curt Hedges, working with accounting supervisor Jonelle Lewis and her crew, did a major reorganization of accounting and financial resources. But there remained a need for additional administrative strength. After a year–long search, founder Rod Coleman convinced Ed Chapin to join the Stride team. Taking over on December 16th, Ed is now completing his first month as president of Stride Micro.

Unlike many of the professional managers that have surfaced in the industry lately, Ed brings a strong technical background to the company, coupled with impressive administrative credits. He served as CEO at Life Touch/Kinderfoto, was a director of Department of Defense programs at the University of California, and a director of tactical and electronic warfare systems at Martin Marietta. In a unique combination, Ed has B.S. degrees in both Chemical and Electrical Engineering as well as an MBA from the University of California.

Ed reports to the Board of Directors under the direction of Rod Coleman as Chairman. But the move frees Rod to concentrate his energies on high–tech research and product development. It's a formula that produced our first breakthrough 68000 micro (Sage II) back in late 1981. We've no doubt that the magic will soon repeat. Along with his fellow magicians in Research and Development, we can already feel the excitement building. Wilbur Harvey, Bill Bonham, Bruce Robertson, Bob Needham, Don Marsh and newcomer Bill Rainey, have already been keeping the pipeline full of new products: the NOD, graphics hardware and software, bigger disks, QIC tape drives, 256K DRAMS, UNIX improvements, p–System IV.21 and more.

Those efforts would have never been seen without the excellent manufacturing team headed by Al Bennett and his managers, Charlie DeGroff and Alice Nottier. The production crew did their usual fine job. Time and again, they really pulled through, despite impossible requests from customers anxious to have their new toys shipped immediately. Those orders are sure to grow as a result of an expanded sales team assembled by sales vice president Rhine Meyering. Just this month we were able to announce new offices in Los Angeles and Chicago.

Keeping up with those new toys from R&D put a burden on our Tech Support group, but they also came through with flying colors. Tom Mitchell and Chris Dunlap deserve a hand for learning UNIX in record time. Cindy Hink streamlined their paperwork and improved their phone habits, while Kelly Weston provided the necessary manuals and documentation. Sheri Coleman gets the credit for organizing the whole operation as well as putting out fires.

Speaking of fires, veteran "firefighter" Lonnie Cline specialized in helping international customers this year and deserves an extra "Well Done!"

Providing technical support to our users has always been a high priority with Stride. We feel that our best product feature has always been the people who stand behind it. In fact, Stride's success may be be attributed to the fact that we are willing to talk about and resolve any questions and/or problems of any users anywhere in the world.

One of our most important support vehicles, our magazine, did undergo a change this year. *In Stride* was always popular, but we responded to those who wanted more technical detail by introducing Tech Notes. Both publications now exist, the *Tech Notes* for user "how to" and *In Stride* for marketing news. (Forgive the personal pat on the back and thanks to all who called to tell me how much they liked the change).

Marketing, under vice president Buddy Frank, has taken on a non–traditional flavor at Stride. It always seemed that Silicon Valley marketing was confined to the stern–faced guys in gray suits with unlimited expense accounts. Those of you who have talked to Buddy have found him to be a warm and personable guy with some exciting new ideas about how to promote a growing computer company ("..be honest"). The secret is that Buddy is a closet computer hacker. With a BSEE, as well as a PR degree, he actually understands more about microcomputers than their "ergonomic appeal." Working with the PR team, marketing saw that Stride was included in articles in major magazines such as *Electronic Products, BYTE, Computer Systems News* and many others. Work has started work on expanded promotion and advertising for '86. We're proud that this year in two different respectable surveys of multiuser computer name recognition, Stride Micro landed in the top ten!

Overall, 1985 has been a year of change. We feel like the football coach who suffered through a rebuilding year, but finished with a winning record and hopes of a championship next season. With a strong Stride Faire already scheduled for the end of February, 1986 could be the winning season for Stride and our friends.

Stride Faire '86 begins on Friday, February 28 at the Nugget Hotel and Casino. Their hotel and convention facilities are brand new, but the Nugget is also where the Reno locals have been going to eat for the last 20 years. Hmmm, my magazine deadline approaches, so those who want to continue this conversation on Stride's great '86 plans will have to meet me at the Faire and we'll share a Pan Roast at the Nugget's Oyster Bar! Until then best wishes through the Holidays and the coming New Year!□

512K Memory Board For SAGE II, $389

Expansion board doubles memory for Sage II. Every board is shipped 100% tested with 2 year guarantee. Installs in 5 minutes.

For purchase and information, contact:

S.E. Microsystems
La Jolla, California
(619) 456-2895

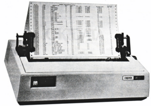
220V Okidata 93 Dot Matrix Printers On Sale For $250

Sorry, the Qume Terminals offered in the October *Tech Notes* are all gone. But a few other items of interest remain in Stride's obsolete inventory and are offered now on a first—come—first— served basis. One which may be of interest to our international users might be the 220V Okidata 93 Printers.

The 93 offers bidirectional printing at 160 characters per second and Correspondence Quality at 40 characters per second. You can use two types of paper feed: friction feed for single sheets and an easily installed tractor feed unit for sprocket paper of different widths. The 93 model has the wide carriage and can print up to 233 characters across at 17cpi.

Printer features are the full 96 ASCII character set with upper and lower characters, superscripts and subscripts, continuous underlining, bold print mode, three font sizes, plus double width characters. You can create your own characters and down—load them into the printer.

These printers have all of the features of a standard Okidata Microline 93 but use 220V power. They are unused, i.e. new! The catch is that it is difficult to modify them for 110V operation —— no kit is available. Shipping is not included in the price. ▯



New Books

Modula–2:
A Seafarer's Manual and Shipyard Guide
by Edward J. Joyce

"Hoist up the anchor! Prepare to cast off on an enlightening and entertaining voyage over the seas of Modula–2." So reads the release from Addison–Wesley for this hearty guide to Modula–2. Modula–2 is a new language developed by Niklaus Wirth, the author of Pascal. *Seafarer's* compares Modula–2 to a sleek sailboat designed for elegance as well as practicality.

Using nautical metaphors, author Joyce explains what to do –– as well as what **not** to do. A practical primer is included which summarizes the differences between Pascal and Modula–2. Programming novices and admirals alike will find smooth sailing.

Skipper Edward J. Joyce is a past editor of *Microcomputer* magazine. His experience includes marketing, software design and engineering. His published works include over 75 articles on topics varying from bats to backpacking and, of course, computers.

The book, ISBN 0–201–11587, is available for $16.96 from Addison–Wesley, General Publishing Group, Reading Massachusetts 01867. Tel: (617) 944–3700 Ext 2614, Lynne Bolduc.

Modula–2:
Programming with Data Structures
by Billy K. Walker

This long–awaited book by Billy K. Walker is finally in your editor's hands. It is a complete student–oriented textbook with strong **useful** programming examples in large print which you can actually read as you type.

It is gratifying to pick up a book for students and see the "Getting Started" section emphasize the early design stages of a program. Later chapters deal with building a user stack, binary trees, priority queues, a Quicksort algorithm, a Heapsort algorithm –– plus more. The *creme de la creme* is chapter 8.4 on processes. A good random number generator is given on page 417.

Walker believes that the best way to learn a programming language is to *read* correct programs first, then write as many programs as possible. *Programming with Data Structures* proves his point.

You can order it by number, ISBN 0–534–05917–1, from Wadsworth Publishing Company, Box CS–01–86, Ten Davis Drive, Belmont, CA 95002.

Advanced UCSD Pascal Programming Techniques
by Barry Demchak
& Eliakim Willner

Even if you have every book on Pascal ever written, you should get this masterpiece by Willner and Demchak. The UCSD Pascal language, or p–System Pascal, has changed a bit since its first conception –– many tricks and hints are offered here for the first time in print. The groundwork of the book began with Richard Gleaves, author of *Modula–2 For Pascal Programmers* and a compatriot of Demchak's.

Demchak was a member of the UCSD team which implemented Pascal under the direction of Professor Ken Bowles at the University of California, San Diego. (The Pascal language itself was designed by Niklaus Wirth). Demchak's knowledge of p–system internals shows in the detailed descriptions of the intrinsics, process control and memory management. During the 2 1/2 year period it took to complete the book, Demchak said he most enjoyed writing section 6–15 on coding style and optimization.

He insisted that this review warn the reader of the "one typo" in the book –– on page 185 the statement **uless:= ord(a) < ord(b0** should be **uless:=odd(a) < odd(b)**. That's dedication!

Eliakim Willner has designed and coded many applications in UCSD Pascal. His extensive knowledge of real–world programming is passed on to the reader in clear–cut examples. Every section contains numerous NOTES, WARNINGS, and HOWEVERS to give the programmer help in understanding what the instructions and examples *really* do.

Willner says, "While Barry's background is in systems and mine is in applications, we both love to muddle around in the other's territory." For example, Demchak did the section on efficient programming while Willner did the work on internal system data structures. Their strength in both fields may why the topics in the book are so *completely* explained.

Willner will be doing more systems work as the technical partner of the new company, Pecan Software Systems, which now owns and markets the UCSD Pascal compiler and the p–System operating system.

Although the title says "Advanced Programming", Willner and Demchak's book offers the novice the best description of UCSD Pascal's unique constructs and the prime source of the magical, essential lore of the system.

It is available for $18.95 at local bookstores or order ISBN 0–13–011610–6 from Prentice–Hall, Englewood Cliffs, N.J. 07632. In Canada, write to Carl Henderson, Prentice–Hall Canada, 1870 Birchmount Road, Scarborough, Ontario, Canada M1P2J7. □

Ed.Note: The text preparation of Demchak and Willner's book and much of the program testing were done on Stride machines, first a Sage II and then a 440. Billy Walker also did much of the work for his book on a Sage II. It's good to see all that hard work finally in print!

Identifying Current BIOS And UTIL Version Numbers

With the release of the IV.21 p–System, Stride also updated the single and multiuser BIOS. The release has been through the Beta, prerelease and first formal release stages during the last few months. An update was also released November 27th.

All files during the various stages share the same release numbers. The dates of the files indicate order of the release.

To find out if you have the latest and greatest BIOS and UTIL files, look at the master files on your release disk with the filer and compare the sizes and dates shown here. Follow instructions to read the version.

If you would like current files, order DK0100 for $25. (Your dealer should be getting a copy soon and can update you also.) This diskette is kept current with all Stride files that have been changed since the last major release. (It does not update you to p–System IV.21, the BIOS and UTIL files are independent of the p–System version).

SYSTEM.BIOS
Version 4.5
50 blocks long
9–Oct–85

To determine the BIOS version, run UTIL, using option B to read the file. The message should be "BIOS Version 4.5 read successfully. Size including buffers 23876 bytes."

Note that when you change the configuration with UTIL the SYSTEM.BIOS file will get a new date. Check only the date on your master diskette; the one on your working system will probably be different!

UTIL.CODE
Version 4.4
87 blocks long
23–Sep–85

To determine the version, run UTIL. It will immediately display "UTIL Version 4.4 Initializing......"

MU.UTIL.CODE
Version 4.2
96 blocks long
2–Jul–85

To determine the version, run MU.UTIL. The first display is "Multi User Configuration Utility – Version 4.2 Initializing......"

MU.CONFIG
Version 4.2
22 blocks long
30–Aug–85

Depending on options, MU.CONFIG may be only 19 blocks long after you install your system. Compare the above with the master file for MU.CONFIG located on your UTILITY diskette.

To determine the version, run MU.UTIL. After typing in the configuration file name, it will display "Configuration Version 4.2 read successfully."

MU.BIOS
Version 6.4
74 blocks long
27–Nov–85

If you have a program called "STRIDE," run it. It will display the version number. The routine used is published on page 4 of the October *Tech Notes* . There is no other way to see the MU.BIOS version. ☐

p–System Stop/start Key

The p–System defines a terminal **stop/start key.** The key is a toggle, defined in SYSTEM.MISCINFO. When typed, the computer will stop sending characters to be displayed on the terminal. The stop/start key must be typed again for the display to be resumed. This is useful when you dump a large file to the screen and your terminal displays too fast for you to read. Use the key to start and stop the display for easy viewing.

On the Sage II & IV, the stop/start key is CTRL/S.

On the Stride 400 Series computers the key is normally undefined. Instead, the 400 Series BIOS uses the XON/XOFF characters **CTRL/S to stop** and **CTRL/Q to start** the display. The start/stop key can be used in addition to these, if the start/stop key is not CTRL/S or CTRL/Q. However, there is really no need for both methods and the XON/XOFF protocol is more common.☐

Operating System Information Words

By Tom Mitchell

The Stride BIOS keeps a table of words which may be used by various operating systems. This table is called the **OPERATING SYSTEM INFO WORDS.** The words may be examined or changed with the Stride utilities **UTIL** or **MU.UTIL.**p–System version IV.21 uses Word 0 and Word 3. The manual documentation is for version IV.20. The release notes for IV.21 need clarification but do give the suggested sizes for these areas.

****

Assignments are for
p–System IV.21 only!

Word 0

Word 0 is used to define the size of the data space. The value must be given in HEX bytes. If the value is zero, a default of 64K is assumed. Stride does not currently support any other data space assignment besides 64K.

Word 1 and Word 2

Word 1 and Word 2 are not used in version IV.21 of the p–System. Previously, in version IV.2, Word 1 was used to set the size, in bytes, of the p–System Code Pool. Word 2 was used to define the size in bytes of the Liaison Socket Pool.

The Liaison Socket Pool is now set in SYSTEM.MISCINFO. The value assigned there is in HEX words. To change SYSTEM.MISCINFO, use the system utility SETUP.CODE.

Word 3

Word 3 controls the overall space for the p–System. The value must be given in HEX K bytes where 1K=1024 bytes. Thus, Word 3 minus the sum of the Socket Pool and data space is equal to the code pool size.

To calculate the Code Pool (in bytes):

```
CodePool =   (Word3*1024)
            -(SocketPool*2)
            -(DataSpace)
```

Once the system has been rebooted, the user can see the Code Pool size by typing **S** for **SET** at the p–System command line.

If Word3 is equal to zero, all available memory is taken by the p–System, leaving no room for ScenicSoft's Modula–2 environment. Note: Many of Stride's graphics demos use this Modula–2 space.□

Mosys

THE MODULA-2 SYSTEM

MOSYS is the first system to provide an integrated software support environment for developing MODULA-2 programs

MOSYS is a *native code* system: complete, self-contained, and well-documented to provide the ideal programming environment.

MOSYS includes compiler, screen editor, document processor and symbolic debugger.

MOSYS PRICING:

New Release 101a
Now only $595

For MOSYS outside U.S. contact:
Robinson Systems Ltd.
Red Lion House
St. Mary St. PAINSWICK Glos GL6 6QR

TWO SPECIAL PRODUCTS FOR THE STRIDE

distributed by

Maritime Infosystems Ltd.
6660 Reservoir Road
Corvallis, Oregon 97333

(503) 929-2552

H.I.T.S.

HALF INCH TAPE SUBSYSTEM

Half inch reel-to-reel tapes, long a mainstay of the mainframe world, are now available to Stride micros.

HITS forges a needed link between Stride and mainframe and mini-computers, providing access to millions of reels of stored data which are not available in other media.

Contact us for a brochure or see the article in September of InStride Tech Notes.

HITS PRICING:

With Kennedy 9600, an 800/1600 BPI 45 IPS (stop/start) tape drive. Introductory price until 12/31/85 $8950 (reg. $9795).

For HITS outside U.S. contact:
Stride Micro International Division

UNIX Accounting

The UNIX operating system includes a package called *Process Accounting.* This package is very useful on systems with many users. It keeps track of the logins, time spent (for billing purposes), the usage of programs and many other events. Periodically, it generates reports which can be read by the system administrator or printed out automatically.

Accounting Printouts

Stride distributes UNIX already setup to the Process Accounting and print out a monthly report. This report may be a problem if the system printer is not robust enough to print 10 or 20 pages unattended. To turn off the automatic printout, edit the file **cron.monthly** (you must do this from superuser). It should look like this:

```
#! /bin/sh
/usr/bin/lp -c -tmon.acct
    /usr/adm/acct/sum/rprt*
su adm -c /usr/lib/acct/monacct
```

Comment out the last two lines by inserting a "#" at the beginning of the lines.

```
#! /bin/sh
#/usr/bin/lp -c -tmon.acct
   /usr/adm/acct/sum/rprt*
#su adm -c /usr/lib/acct/monacct
```

Turning Off "Acct"

On a small system with less than six users, you may decide that you do not need the accounting at all. It can use up a fair amount of disk space. If you have dialup lines, it is dangerous to turn off accounting as you will need to know who is accessing your system from outside.

If you decide to do this, the following command turns it off immediately. However, if you reboot, it will start up again.

```
/usr/lib/acct/turnacct  off
```

You may edit the **/etc/rc** file to remove this process from the list of processes automatically started by the system. Find the following in the file:

```
# Set up Process Accounting
   rm -f /usr/adm/acct/nite/lock*
   /bin/su adm -c /usr/lib/acct/startup
   echo "Process accounting started"
```

Leave the first instruction alone. Comment out the next two lines so that it looks like this:

```
# Set up Process Accounting
    rm -f /usr/adm/acct/nite/lock*
#   /bin/su adm -c /usr/lib/acct/startup
#   echo "Process accounting started"
```

Save the changes. Now go to the accounting directory and remove all files. One easy way to do this is:

```
cd /usr/adm/acct
rm -ri [fns]*
```

You will be prompted to remove each file. Type just a **Y** at each name. Process Accounting is now shut off.

Check These Files

Even without Process Accounting, some log files are kept and will continue to grow:

```
sulog     - successful "su" commands
shutlog   - system shutdowns
/etc/wtmp- logins and logouts
```

As system administrator, you should inspect the log files (you can read them with any editor) every week or so. The **/etc/wtmp** is a data file which you can access via **last.** It will continue to grow slowly with each login. Periodically take a look at it and then clear it with:

```
cp /dev/null /etc/wtmp
```

This replaces it with a short empty file. Don't try to remove it; it will simply be recreated.

More information is available on Process Accounting in Vol. II page 4–1 of the UNIX manual. □

??? ADS ???

Yes, *Tech Notes* is taking ads. As *In Stride* comes out just four times a year and vendor products come out every other week, our developers need a place to show off and maybe make a few sales. After explaining for the 99th time, that *Tech Notes* was intended to be a user support forum and getting told, "Don't users want to know about new software and hardware?" your editor decided that a few ads wouldn't hurt the spirit of the magazine.

The ads have always been a service offered to Stride's vendors at printer costs. The ad format for *Tech Notes* will be black and white only, camera ready copy. No dot matrix listings please — they will not print dark enough. (Ask about typesetting charges). Only these sizes are offered: Full Page – $350, Half Page – $150, Third Page – $135. Deadline for copy is the first of every month. (P.S. Ad space for the February *In Stride* is full.) □

Linking SVS Pascal And SVS FORTRAN to UNIX C Routines

The SVS Pascal and FORTRAN Compilers can be linked to C routines in UNIX. The manual describes a "wrapper" routine which is a cumbersome method of using an assembly code routine to translate the stack between the two compilers. Using a "wrapper" requires a great deal of knowledge about how the different compilers use the stack.

A much easier way to do simple calls exists. In the following examples a C program is linked to a Pascal program and a FORTRAN program. These examples use the built–in **CEXTERNAL** call in the SVS compilers.

The C/Pascal Program

This is the C program, the name can be lower case. Note that the *order of the arguments is reversed from that of the calling SVS program.* This program adds two numbers for an integer result.

```
iadd(b,a)

    int a, b;
{
    return (a + b);
}
```

The C program will be called by this Pascal program. Note that the values are long integers to keep data size the same.

```
PROGRAM TestP;

VAR I,J,K:LONGINT;

FUNCTION add(I,J:LONGINT);
        LONGINT:CEXTERNAL;

BEGIN
I:=2;
J:=8;
K:=add(I,J);
WRITELN(I:7,J:7,L:7);
END.
```

If you need to pass the *address* of the variable, rather than its value, then the Pascal program FUNCTION statement must be changed with the arguments defined as VAR arguments:

```
PROGRAM TestP;
VAR I,J,K:LONGINT;

FUNCTION add(VAR I,J:LONGINT);
        LONGINT;CEXTERNAL;
```

If this is done, then the C program also must change to reference the addresses. Refer to the listing of the C program for the FORTRAN example.

The following set of commands compiles the C program, then the Pascal program is compiled. The **–e** option is all important here. The output of the Pascal compiler is run through the code generator and linked. The UNIX system linker **cc** ties the two together. To execute, run file **a.out.**

```
cc -c TestC.c
/usr/bin/pascal  -e TestP.pas
/usr/bin/code  TestP.i
/usr/bin/ulinker  -l Test1.o
  TestP.obj /usr/lib/paslib.obj
cc Test1.o TestC.o
a.out
```

Extra care must be used with variable/function names. Consider the read in C versus the read in Pascal.

The C/FORTRAN Program

The name in the C program below can be lower case. Note that the *order of the arguments is reversed from that of the calling SVS program.* The C program adds two numbers for an integer result.

Since FORTRAN passes the address of the values instead of the values themselves, the C routine has been changed to reference the addresses also.

```
iadd(b,a)
    int *a, *b;
{
    return (*a + *b);
}
```

The FORTRAN program shows no special way to call the C routine, but it will be done anyway.

```
        INTEGER I,J,K
        I=2
        J=6
100     K=IADD(I,J)
        WRITE(*,900) I,J,K
900     FORMAT(3I7)
        END
```

The following set of commands compiles the C program, then the FORTRAN program is compiled. The **–e** option is again important here. The output of the FORTRAN compiler is run through the code generator and linked. The UNIX system linker **cc** ties the two together. To execute, run file **a.out.**

```
cc -c TestC.c
/usr/bin/fortran  -e TestF.for
/usr/bin/code  TestF.i

/usr/bin/ulinker  -l Test2.o
  TestF.obj
      /usr/lib/ftnlib.obj
      /usr/lib/paslib.obj

cc Test2.o TestC.o
a.out
```

More complex calls are obviously possible, but require care and some knowledge of how arguments are stored and passed in the system. Passing data types generally requires a lot of planning.☐

Seminar For C And MODULA–2

A one–day seminar is being offered in conjunction with Stride Faire '86 by Dr. Richard Wiener and Dr. Claude Wiatrowski. It is a great opportunity to get acquainted with Modula–2.

For Programmers

No matter what your language of choice, you as a programmer could benefit from this seminar in many ways. Pascal programmers will learn the strengths of C and Modula–2. C programmers will learn the benefits of Modula–2. Modula–2 programmers will learn the benefits of C. The strengths and weaknesses of the two popular languages will be shown quickly and painlessly. A carefully crafted set of examples will demonstrate advanced programming in C and Modula–2.

For Software Managers

If you manage a software group, the seminar will provide realistic programming examples to help evaluate C and Modula–2 from a software engineering perspective. You will get a better feel for the strengths and weaknesses of C and Modula–2.

The Content

A graduated set of programming examples will be given in both C and Modula–2. These examples will illustrate the strengths and weaknesses, as well as the similarities and differences, of Modula–2 and C. Each carefully chosen example will be shown in both languages, and will demonstrate important software engineering principles, from high–level concerns such as data abstraction and information hiding, to low–level activities such as hardware–dependent code.

The Benefits

Learning a new language is a major undertaking. Often the benefits and drawbacks of that language are not apparent until the language is learned and some realistic programs written. Without getting mired in the details of each language, the course helps you evaluate two languages in a realistic setting. It will help you decide if learning one of these languages will benefit you. As an added benefit, you will see a number of concise examples of software engineering principles which you may apply to programming in any language.

For a software manager, the decision to use, or not use, a particular language has important economic consequences. This course provides additional data that you can use in making that important decision.

You Should Attend If:

- *You have been thinking about learning one of these useful languages.*
- *You are wondering if you should be thinking about learning one of these languages.*
- *You are responsible for the selection of languages for your firm.*
- *You are a student or beginner who would like to see some well–thought–out examples.*

The Instructors

Dr. Richard Wiener is a professor of computer science at the University of Colorado at Colorado Springs. He is the co–author of five textbooks on Modula–2 and software engineering, as well as a new book, *Modula–2 Software Components,* due to be published in 1986. He received the Outstanding Teacher of the Year award at the University of Colorado at Colorado Springs in 1983. Dr. Wiener is currently the editor–in–chief of the *Journal of Pascal, Ada, and Modula–2.*

Dr. Claude Wiatrowski is the president of Mountain Automation Corporation, Woodland Park, Colorado. He has been a computer hardware and software product designer for more than 16 years. He currently designs instrumentation, automation, control and communications software products. He is the co–author of a computer textbook and received the Outstanding Teacher of the Year award at the University of Colorado at Colorado Springs in 1978. Dr. Wiatrowski is VLSI editor for *Simulation.*

Seminar Details

Place: Stride Faire,
Reno, Nevada
Date: March 3, 1986
(Monday following
Stride Faire)
Time: 9 a.m. to 5 p.m.
Fee: $200 per person

The fee includes the full–day course and a complete set of course notes. Send payment with your registration information, (name, address, phone and language of choice) to the address below by February 14, 1986.

Dr. Richard Wiener
P.O. Box 953
Woodland Park
Colorado 80863

Full payment will be refunded if your cancellation is received by February 14, 1986. No refund will be made for cancellations after that date. Should the course itself be cancelled, full refunds will be made to all participants.

Courtesy of Stride Micro, attendees receive up to 70% off on air fair and an off–season rate on hotel rooms. For more information on the "Faire" rates, call **Stride Faire (702) 322–6868.**□

Qume QVT102 vs Qume QVT102A

Last week someone asked Stride, "What's the difference between Qume QVT102s and QVT102As. We checked with Qume and found that there are very few differences between the newer A model and the original. The QVT102 *can* be upgraded to most QVT102A functions.

A Qume engineer explained the differences:

- *The eight function keys on a QVT102A are user programmable, they generate fixed codes on the QVT102.*
- *The QVT102A has a bi–directional auxillary port. (The upgrade will not make the QVT102 port bi–directional).*
- *The QVT102A uses a combination of keys, CTRL/SHIFT/SETUP, to go into setup mode. (Hold all keys down at the same time). The QVT102 uses just SETUP. The change makes it less likely for you to hit the single SETUP key accidently and change the mode.*
- *The QVT102A has a new command to control the status line.*

To tell which model you have, just try the SETUP option. The upgrade is to install a new U8 EPROM. The QVT102A manual is also available but has few differences from the QVT102 manual. Order from:

Pacific Business Data Systems
6938 Sierra Ct.
Dublin, CA 94568
TEL: (415) 828-9410

#PN 84813-130 $45 EPROM
#PN 35003A $20 Manual

The update is in stock; delivery on the manual may take 30 days. ☐

Square Root Revisited

A lot of Stride users are interested in bit picking in assembly code. The following letter shows another way to do square roots.

Dear Editor,

Professor Gallup's article on the square root function in the September *Tech Notes* was very informative. Following is a description of a similar routine that I've developed which also may be of interest to your readers. This routine takes a 32–bit integer argument and calculates its square root, rounded to the nearest integer. The discussion below outlines the basic algorithm employed in the routine, using Pascal notation.

A decimal number can be rounded to the nearest integer by adding 1/2 and truncating the decimal part. Thus, the square root routine should accept an integer argument X and calculate Y:

TRUNC (SQRT(X)+1/2)

Y is the greatest integer for which

Y–1/2 <= SQRT (X)

or equivalently (squaring both sides),

Y*(Y–1)+1/4 <= X (for X>0).

Since X and Y are both integers, this latter condition is equivalent to Y*(Y–1)<X. So the routine should produce the greatest integer Y for which

Y*(Y–1) < X

The exception in the above equation is the case where X=0, for which Y:=0.

The result of the square root routine, Y, can then be determined by the following computer algorithm.

Begin by setting Y equal to zero. Then, starting with the high order bits and working downward, test to see whether Y*(Y–1) becomes greater than or equal to X when one of Y's bits is changed to 1. If so, leave the bit cleared (so that Y*(Y–1) remains less than X. If not, set the bit to 1. The algorithm can be shown in Pascal–like syntax as follows:

```
Y:= 0;
For i:= 15 downto 0 do
  if (Y+2^i)*(Y+2^i-1)  < X then
    Y:= Y+2^i
```

The loop starts at i:= 15 because the rounded square root of a 32–bit integer will always be less than 2^{16}, assuming that bit 31 of X is used as a sign bit. If X is an unsigned integer in the range $0...2^{32}-1$, the loop should begin at i:= 16 because a few numbers in this range will produce the result Y= 2^{16}.

The algorithm described above needs some rephrasing to optimize machine efficiency (in particular, to avoid use of multiplication operations). The accompanying annotated 68000 assembly listing is a fairly well–optimized implementation which uses 17 machine instructions (excluding range error handling) and executes in about 120 microseconds at 8 MHz.

...The listing is shown on next page...

Sincerely,
Kenneth Johnson

Ed.Note: Questions about the square root routine should be sent to the author at 1215 Brewster Drive, El Cerrito, CA 94530. Tel: (415) 237–3120. Kenneth Johnson is the author of a printer package which generates custom fonts on Okidata printers. See the May 1985 In Stride, page 12.

```
                 ;SQUARE ROOT ROUTINE
SQRT32           ;Usage:
                 ;  Push 32-bit (signed) argument onto stack.
                 ;  JSR to SQRT 32.
                 ;  32-bit square root result is returned on stack.
                 ;Registers used: D0,D1,D2,D3.
;-----------------------------------------------------------
                 ;Synopsis of algorithm:
                 ;
                 ;  X = 32-bit (signed) argument;
                 ;  compute Y[0] = TRUNC(SQRT(X)+1/2)
                 ;          Y is a 32-bit result
                 ;  Y[16] :=0;
                 ;  For i:= 15 downto 0 do
                 ;  If (Y[i+1]+2**i)*(Y[i+1]+2**i-1) < X
                 ;     Then
                 ;       Y[i]:= Y[i+1]+2**i
                 ;     Else
                 ;       Y[i]:= Y[i+1].
                 ;Initialize registers:
                 ;Y[16]:= 0.
                 ;i:= 15.
;-----------------------------------------------------------
        MOVEQ    #31.,D0          ;D0:= 2*i+1 = 31.
        MOVE.L   4(SP),D1         ;D1:= X-Y[i+1]*(Y[i+1]-1) = X
        BMI      ERROR            ;Branch to ERROR if X < 0.
;
;hand code the equivalent of  MOVE.L  #3FFF8000,D2
        .WORD    0010010000111100T,3FFFH,8000H
                                  ;D2:= 2**i*(2*Y[i+1]+2**i-1) = 2**15*(2**15-1).
        MOVEQ    #1,D3            ;D3 = shift count for LSR instr.
        BRA.S    $1               ;
$0:                               ;register status:
                                  ;  D0 = 2*i-1.
                                  ;  D1:= X-Y[i]*(Y[i]-1).
                                  ;  D2 = 2**i*(2*Y[i]+2**i-1).
;
;Note: Bit 2*i-1 of D2 is 1; so the BCLR D0,D2 instr. below
;represents the operation D2:= D2-2**(2*i-1)

        BCLR     D0,D2            ;D2:= 2**i*(2*Y[i]+2**i-1)-2**(2*i-1)
                                  ;   = 2**i*(2*Y[i]+2**(i-1)-1).
        LSR.L    D3,D2            ;D2:= D2/2 = 2**(i-1)*(2*Y[i]+2**(i-1)-1).
$1                                ;Assign i:= i-1.
                                  ;register status:
                                  ;  D0 = 2*i+1.
                                  ;  D1 = X-Y[i+1]*(Y[i+1]-1.
                                  ;  D2 = 2**i*(2*Y[i+1]+2**i-1).
;Note:  bit 2*i+1 of D2 is 0; so the BSET D0,D2 instr. below
;represents the operation D2:= D2+2**(2*i+1)

       CMP.L  D1,D2
       BPL    $2                  ;If X <= (Y[i+1]+2**i)*(Y[i+1]+2**i-1), then
                                  ;branch to $2 with the following implicit
                                  ;assignments:
                                  ;        Y[i]:= Y[i+1].
                                  ;        D1 = X-Y[i]*(Y[i]-1).
                                  ;        D2 = 2**i*(2*Y[i]+2**i-1).
                                  ;X > (Y[i+1]+2**i)*(Y[i+1]+2**i-1):
                                  ;Assign Y[i]:= Y[i+1]+2**i.
       SUB.L  D2,D1               ;D1:= X-Y[i+1]*(Y[i+1]-1)
                                  ;   -2**i*(2*Y[i+1]+2**i-1) = X-Y[i]*(Y[i]-1).
       BSET   D0,D2               ;D2:=2**i*(2*Y[i+1]+2**i-1)+2**(2*i+1)
                                  ;   =2**i*(2*Y[i]+2**i-1).
 $2:   SUBQ.L #2,D0               ;D0:=2*i-1.
       BCC.S  $0                  ;Loop back to $0 if i > 0.
                                  ;Continue when i =0.
                                  ;D2 = 2**i*(2*Y[i]+2**i-1) = 2*Y[0].
       LSR.L  D3,D2               ;D2:= D2/2 =Y[0] = result.
       MOVE.L D2,4(SP)            ;Substitute result for argument.
       RTS                        ;Return.

ERROR                          ;Argument is negative.
```

New ALADIN Multiuser Database Released By ADI Germany

ADI of West Germany has released ADIMENS, a multiuser relational database with integrated word processing for the Stride 400 Series. ADIMENS is compatible with files created by ADI's earlier single–user product ALADIN.

Features

- *Menu Techniques with function keys are used instead of a query language.*
- *The same user interface is used in all modules with 25th line F–keys.*
- *Integrated user help screens exist throughout the program.*
- *Terminals can be configured independently for each user.*
- *A maximum of 15 users can be on–line to any database.*
- *Variable length records allow for efficient storage.*
- *Individual records can be accessed in less than one second.*
- *The report generator is interactive.*
- *Statistical analysis of all data types is available.*
- *Routine tasks can be setup as batch processed jobs.*
- *Password control is implemented on FUNCTIONS, FILES and FIELDS.*
- *Files and fields can be added without re–entering data or rebuilding the database.*
- *Screen layouts can be altered at any time, without affecting the data.*

Specifications

When choosing a database, it is critical to know the limits of the program so that the buyer is confident his data will not outgrow the program within the life of its use. The table at the bottom right shows the maximum number and size of the important ADIMENS parameters.

The Word Processor

The word processor shows BOLD, UNDERLINE and other functions on the screen. This requires a terminal with invisible attributes.

The Stride/Wyse terminal has embedded attributes. The screen will not show the special functions, but will print them correctly. Terminals that do display these functions are Qume, ADM, LINK and TATUNG.

A unique hyphenation feature helps with paragraph formatting.

Applications

A database is one of the most useful of all computer tools. It can be used to register all manner of items and people. User mailing lists are the most common example. Library administration, job progress monitoring, medical patient records, stock control, sales order processing, statistical analysis, personnel records and state and county property records are just of few of the tasks accomplished with a database.

With the ADI Program Generator, custom applications can be built easily.

Distribution

ADIMENS is distributed by Strategic Information Systems, which also distributes the p–System for Stride in the UK.

```
Strategic Information Systems
6-7 Benjamin St.
Albion House
London EC1N
TEL: (01) 608-0818
TX:  (851) 22206 SAS G
Contact: John Rolfs

ADIMENS ( 400 Series )  $1495
ADI PROG Interface      $ 895
```

ADI of West Germany is the parent of ADI America which distributed the original single–user version on the Stride. The two companies now sell separate products. Tech Notes will inform you when ADIMENS has a USA distributor.

REBATE!

A rebate of 25 percent is offered for returned copies of ALADIN 2.6 disk, manual and a copy of the purchase invoice. Rebate offer is good until May 30, 1986. Dealer terms are available on request.☐

```
                  Max Number of:

     Logical Files per Database...........    31
     Screens per Database.................    63
     Records in one logical file.......... 32767
     Bytes in a record....................  4000
     Data Fields per Database.............  1024
     Key (index) fields...................   192
     Simultaneous data fields.............   150
       (Sorts, searches, calculations)
     Data Fields in one report............   500
     Bytes per one Physical data file.....16 Meg
     Bytes per one Physical Key File......16 Meg
```

Porting Via The UNIX C Compiler

By Bruce Robertson

Most C programs will port over to Stride's UNIX with very little work. However, Stride has found a couple of "gotchas" in the Motorola compiler. If you know about these, they are fairly easy to work around. We hope these short examples will help you.

The following double precision pointer bug causes the compiler to fail with an error message.

```
main()
}
   double *p1, *p2

   *p1++ = *p2++;
{
```

This next example is related to the first. It also causes the compiler to fail with an error message. The range of **a** is not important.

```
main()
}
   double a[2][2];

   a[0][0] = a[1][1]=1.0;
{
```

The following is also a double precision pointer problem. The code generated by the compiler is incorrect. No error message appears.

```
main()
}
   double *p1, *p2

   *--p1 = *--p2;
{
```

The next example also compiles to incorrect code with no error message. There are very precise conditions involved: 1) The structure must be between four and eight bytes in length. 2) The pointer must be in a register.

```
struct foo
}
   int x;
   int y;
{;

struct foo a, b;

main()
}
   register struct foo *p;

   a = b = *p;
{
```

An obscure problem exists in the order of Boolean expression evaluations only outside of an **if** statement:

```
foo() && bar();
```

The above evaluates boolean function **foo()** first. The problem is that even if **foo()** is false, **bar()** is still called. The fix is to use an **if** to call **foo()** and then call **bar()**.

One way to scan your files quickly for these constructs is to use **egrep**. For example, the following is all one line that checks for constructs of the type **a=b=c;**.

```
egrep "[a-zA-Z]*[\t]*=
   [\t]*[a-zA-Z]*\t]*=
   [\t]* *.*"
```

You can embellish this further by adding _ and . to the [a–zA–Z] statements to find references to assembler variables. A –> will help with structure, etc. By creating a simple alias for the **egrep** lines you can create a small "suite" of checks to insure you have converted where necessary. □

Room To Brag

Your Tech Note Editor is always ready to listen to you brag about your products. After all, they show off Stride's machines! Just drop a "release" note or juicy tidbit in the mail. (See address on the back). Articles are also welcome; ask for the guidelines.

People & Products

You can run Z80 CP/M programs under Stride's CP/M–68K system using the CPM–Z80 EMULATOR. This product previously was sold by MICROWAVE and is now being distributed by SoftDesign. **Herbert Thiess** of SoftDesign is now shipping a new "improved" version of the emulator and hopes to release versions for OS–9/68K, AMIGA–DOS, UNIX V and PDOS "real soon now". Contact SoftDesign, Lichtenfelser Str. 29, 8000 Munchen 60 Tel: 089/878955

The **Sukows** at EduSoft have been busy. They now have a Prolog, Modula–2 with EDIS, a Modula–2 editor, EDU–Mail, an electronic mail package and a LISP subset all running under the p–System. For more information contact EduSoft S.A., Rue de Gares 14, 1211 Geneva 1, Switzerland Tel:(41) 22–98–4674

Jeff Bower has integrated his Stride/Sage business graphics package with several dot matrix printers. He loves the HP laser jet best, with the Okidata as the "work horse" and the Epson a close second. The graphics package now works with Stride's 400 Series hardware. Contact Hyperception, 3636 Lemmon Ave, Dallas, TX 75219 TEL:(214) 698–5538□

BULK RATE
U.S. POSTAGE
PAID
PERMIT #165
RENO, NEVADA

Editor: Verlene Joyce Bonham

In Stride Tech Notes is a publication of Stride Micro, issued eight times yearly. Subscriptions are $12 for one year and include the *In Stride* parent magazine which is published quarterly, for a total of 12 issues per year.

Tech Notes back issues are also available for $0.50 and *In Stride* back issues for $2.00 as supply lasts.

Purchase of a Stride computer includes a one-year subscription upon receipt by Stride Micro, Reno, of a fully completed owner's registration card.

Requests for subscriptions, reprint permission, ad rates, bulk orders or submission of prospective articles should be sent to the *In Stride* Editor at Stride Micro's Reno address.

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

STRIDE
MICRO

Postmaster: Change of address notices should be sent to:

In Stride **Editor**
Stride Micro
P.O. Box 30016
Reno, NV 90520 –0016
(702) 322–6868 (8a.m.–5 p.m. PST)
TWX: 910–395–6073

Stride Micro Eastern Division
112–116 Washington Street
Marblehead, MA 01945
(617) 639–0750

Stride Micro Southern Division
13760 Noel Road
Suite 300
Dallas, TX 75240
(214) 392–7070

©Copyright 1983–1985 Stride Micro
All Rights Reserved Worldwide