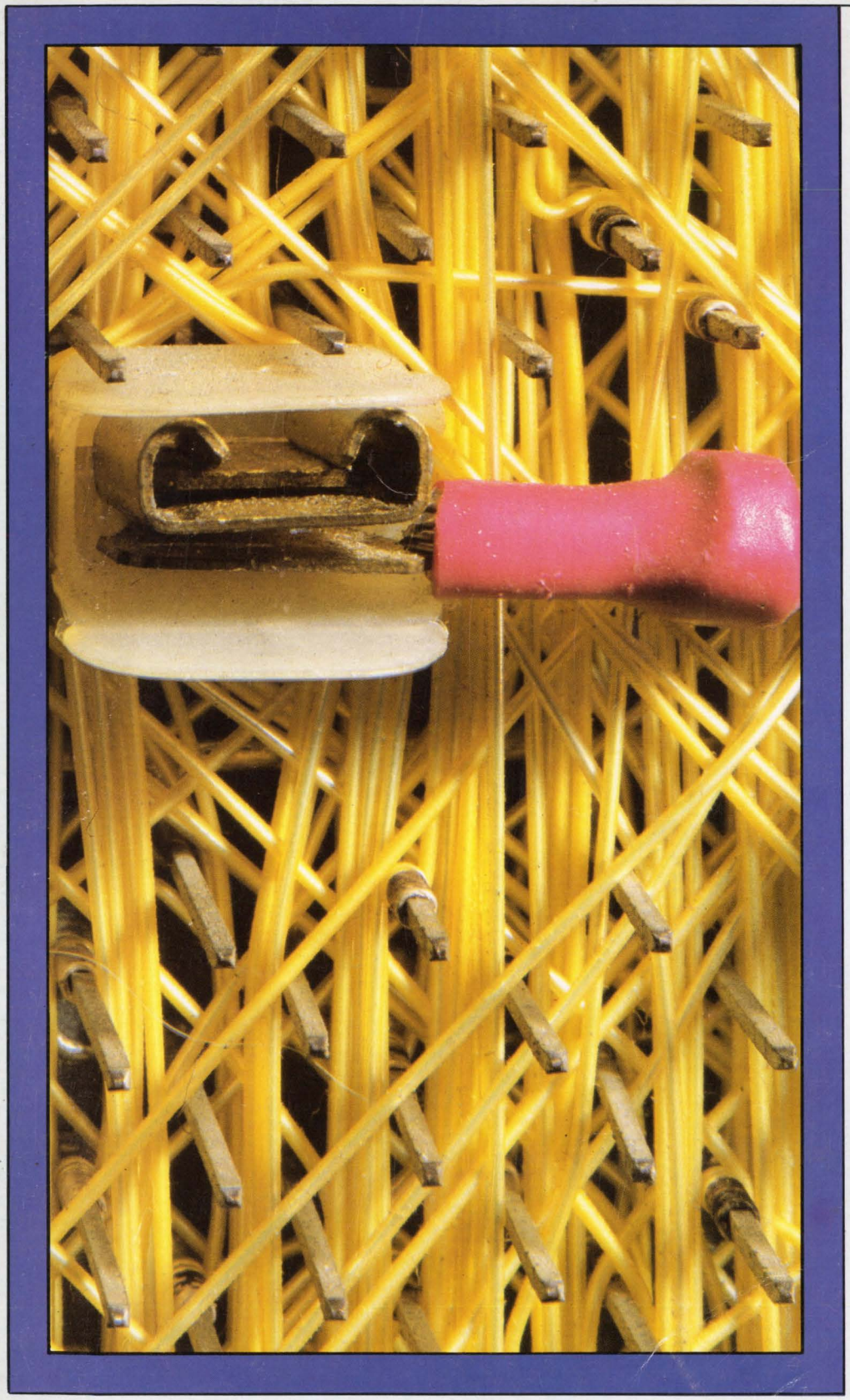
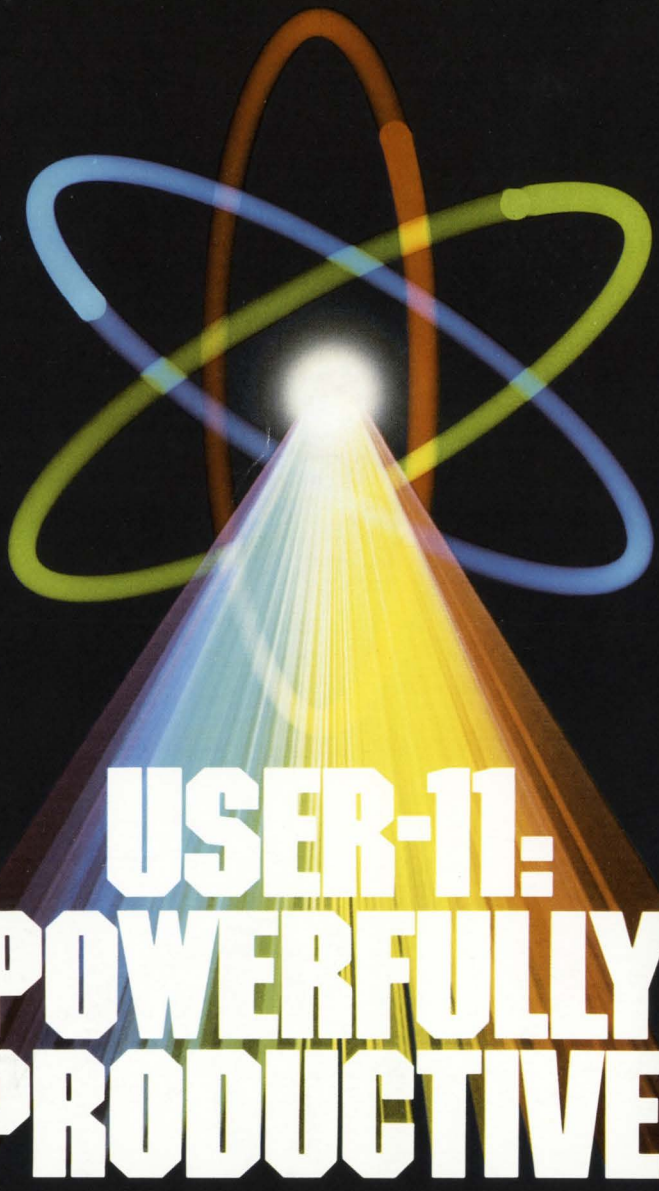
RSTS PROFESSIONAL

Volume 4, Number 5

October 1982
$10.00/issue, $35.00/year



INSIDE:



USER-11: POWERFULLY PRODUCTIVE.

People productivity. It's more important than ever. And a good database system can mean *real* productivity.

USER-11 is a high-performance database system.

It is a fact: Software designed with USER-11 is built more quickly, operates more reliably, and performs better than other software techniques.

USER-11 is unique. It's easy to install. Easy to learn. And easy to apply. Adaptive tools and a standard approach ensure that maintenance is easier than ever.

A key to USER-11's success is its powerful, dictionary-based modules. Software developers simply describe and assemble these modules to create custom business packages—at an unprecedented rate.

Naturally USER-11 is supported with excellent documentation and a variety of training options for beginner to expert. Our commitment is to your complete satisfaction.

Whether you are a software provider or a software user, we guarantee you will be delighted.

Ask us about USER-11 and our family of business software products, or better yet, ask a *productive* USER!

NCCS USER 0110

North County Computer Services, Inc.
2235 Meyers Ave.,
Escondido, California 92025
(714) 745-6006, Telex: 182773

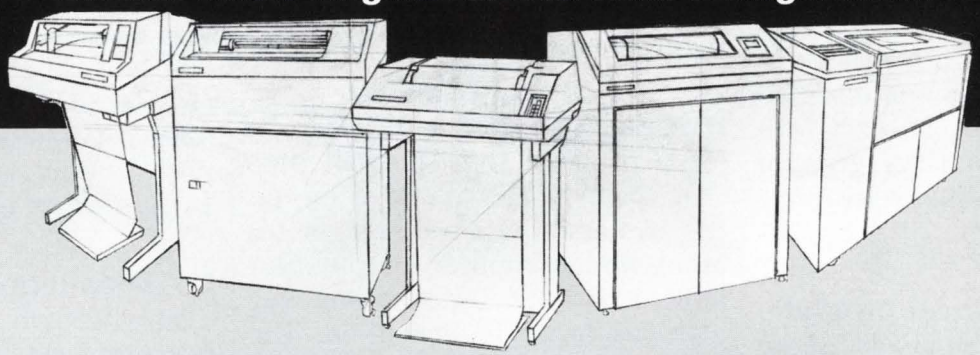
*USER-11 is currently available for DEC computers using the RSTS operating system.

CIRCLE 30 ON READER CARD

Southern Systems has a printer that's right for your computer.

Compatibility guaranteed with all DEC processors!

PDP/8, PDP/11, LSI/11, VAX, DEC 10, DEC 20.
Choose either long-line or short-line configuration.



Get the printer system that's guaranteed... from the leading add-on printer company, Southern Systems. You'll add-on a printer, plus unmatched expertise and top-quality service, when you add-on Southern Systems. Maximum performance/minimum downtime results from technological innovation, expert installation and long-term service, nationwide. Sales offices in all major U.S. cities and in western Europe, United Kingdom and Canada.

Southern Systems

The Printer System Problem-Solvers
2841 Cypress Creek Road, Fort Lauderdale, Florida 33309
(305) 979-1000 • (800) 327-5602 • Telex 522135
In Canada (800) 661-1165 [Alberta: (403) 230-2044]
In Europe 44-3-726-7282

My computer system is a

_____

... and I need a printer system in the speed range of:
__ 33-55 cps (letter quality)
__ 200-300 lpm __ 600-900 lpm __ 1000 plus
__ Parallel
__ Serial (synchronous or asynchronous)
My requirements are __ Immediate
__ 3-6 months __ For information only

Name_____
Title_____
Company_____
Address_____
City_____
State_____ Zip_____
Telephone_____

CIRCLE 2 ON READER CARD

GIVE IMMEDIATE SOFTWARE SUPPORT TO A USER OF ANOTHER SYSTEM

Provo, Ut — Suppose you have a customer that is having trouble running one of your programs on his system. He has a deadline to meet and needs the problem solved immediately (within minutes, not days). What do you do?

Link to his system using CALOUT. With CALOUT you go ahead and hook up your system to his. Do it with any standard phone line and modem. Once you're in there with CALOUT, you can run any program on his systems through your terminal on your system.

Watch the user run his program. Now, the idea is to run CONTRL on his system. With CONTRL you can see everything that is happening on his terminal. If the user does something wrong, you can tell him how to do it right. Or just type the correct entry in at your terminal. It will show up on the user's terminal too.

See program bugs first-hand. With CONTRL you see any program bugs the user encounters. It's much easier to understand a problem when you see it yourself. And since the user knows you see it too, it boosts his confidence that you'll solve the problem quickly.

Transfer the corrected program with CALOUT. Once you've fixed the program, you can get it to your user fast. With CALOUT you can send the program over the phone to his system. No need to worry about file attributes and protection codes. CALOUT takes care of them. Send binary or source files. It makes no difference. And if the phone line is noisy, CALOUT does error checking and correction which guarantees an accurate transfer.

DOC lets user log problems. To log a problem the user just runs DOC. Then he can run programs on the system. DOC records all his input and computer output. But DOC does not interfer in any way with his program execution. The user can even make logs of screen painting programs.

Retrieve the user's log files. Using CALOUT you can rapidly receive DOC log files and program reports from the user's system. Now you have on hand the exact input and computer output for the problem. What cleaner way to clear up complicated problems?

Unique Bootstrap Capability

The first time your system links up to a remote system, CALOUT automatically transfers the necessary sending and receiving modules to the remove system.

No need for operator assistance.

No special programs to load on the user's system by magtape or other convential distribution media.

Link to any system you need to when you need to:

License only the host system where CALOUT resides. With CALOUT on your system you can establish links with other systems. There are no additional licensing costs for each new link.

Call up any remote system and transfer files whenever you need to.

Utilities can be purchased together or separately. All utilities are available for VMS or RSTS/E. For the VMS, DOC is called CATCH-IT, and CONTRL is called DEMO. Also, you can use CALOUT to support systems running RSX-11M, RT-11, and other operating systems.

No installation. None of these utilities require any special hardware or installation. Just load the utility programs on your system and run them. In addition, each utility comes with a complete instruction manual.

Clyde

How to get more information. Call Mary at (801) 224-5306, or write: **Clyde Digital Systems, Inc.** 3707 North Canyon Road/Bldg. 3-E, Provo, UT 84604. Booth 517, Dexpo West.

Contents

Coming . . .

- Some EDT Wishes Granted
- SIMINI — Part 3
- Runtime Systems
- A Login Command File
- A V7.2 Report
- Immediate Mode VT100 Functions
- More Performance Reports
- Stock Quotes Using a RSTS System
- RSTS/E in High Schools
- ATPK.TEC
- More . . .

The RSTS Professional Magazine, October 1, 1982, Vol. 4, No. 5. Published bi-monthly. Single copy price $10.00, $35.00 per year 3rd class; $50 Canada and US 1st class. All other countries, air mail, $60 US. Published by M Systems, Inc., 161 E. Hunting Park Avenue, Philadelphia, Pa. 19124. Application to mail at Second Class postage rates is pending at Philadelphia, Pa. Send all correspondence and change of address to: Box 361, Ft. Washington, Pa. 19034-0361, telephone (215) 542-7008.

From the editors. . .

IF YOU WANT IT YOU HAVE TO ASK FOR IT

Carl B. Marbach

I have recently returned from the Australia DECUS meetings held during July in Melbourne, Victoria. Our RSTS friends 'down under' are some of the most friendly, fine people I have been priviliged to meet during my years of RSTS involvement. Next year's symposium chairman Chris Brett was a terrific host and we were proud to be able to participate with them at their meeting. For the sightseers in the group we hope to have some pictures and a full report in the next issue.

I have now attended DECUS symposia in Canada, Britain, and Australia as well as the U.S. and I thought I would share with you one major difference between the U.S. and other countries which I think we can address. One of the valuable sessions I attended in Melbourne was titled "non-DEC Software Packages for RSTS/E." This session, given by users, gave no prices but did discuss what the package did, how well it did it, how easy it was to use, what kind of documentation was there and a users' overall evaluation of the product. Some of the products discussed included BACKUP packages, WORD PROCESSING packages, SORT utilities, SEARCH and SELECT packages and others. Each was discussed frankly, openly and by more than one person; every one of us in the room learned something about how to make our systems run and work better. Isn't that what we were there for?

The closest the U.S. chapter has come to this type of discussion is with a 'birds of a feather' session where we once discussed a Word Processing system among users of that system. Have you ever tried to schedule a 'birds of a feather session'? Hard to do within the time constraints of a jam-packed five day DECUS symposium. Clearly, the U.S. chapter needs to move more towards serving the user directly with practical knowledge he can take home with him. A RSTS site around the corner from here doesn't send someone to DECUS (well, once they did) and I am sure their reasoning is that it doesn't make economic sense to the company. Most DECUS symposia fees are paid by companies and they have a right to expect something in return. If we don't give it to them then they won't come. Did you know that attendance was DOWN at the Atlanta meeting? I would suggest that one way to increase attendance is to make it more valuable for attendees and their companies.

I have been told that if the SIG (RSTS in this case) wants to do something, they can do it if it doesn't violate any DECUS rules. Well if it doesn't violate them in Australia it doesn't violate them in the U.S. (even if water does go down the drain counterclockwise). Would you like to have a session where users present their experience with non-DEC packages? Here's a deal: Write me a postcard (preferred) or a short note saying so and I'll fight a battle (if necessary) with the DECUS powers to make it happen. Send it to: Carl Marbach, non-DEC packages, Box 361, Fort Washington, PA 19034-0361. If you want it, we'll fight for you. Let there be no mistake: We want a non-DEC package seminar at the St. Louis spring DECUS meetings, I offer to chair the session and have at least ten packages presented by users. And remember: YOU asked for it!

"It was seven years ago today Sergeant Pepper taught the band to play . . . (sic)"

Dave Mallery

Seven years ago today, I was in Maynard, taking the RSTS series. I remember long afternoons in the lab getting that first record-io program to work . . . finally understanding the difference between LET and LSET. I remember being shown a real live 11/70 there in the field service training area.

That was a great time for me. I was fresh from nine years of DP experience, first three years of RCA (remember the 501??(72-43-61)), then the balance in Big Blue, mostly assembly and communications, then a stint running a service bureau with a system-3/10, then back to mainframes with a 135 running DOS/VS and CICS. The transition from IBM to DEC was a radical change. It's hard to imagine two more totally different philosophies within the same industry. They certainly yield wildly different machines. (Isn't it interesting how the 135's and 3/10's are all gone today, but the 11/40's from pre-1975 are still going strong?)

My 70 had to go down last night because of a power glitch. It had been up for fifty days. Other than tape and printer problems, we were down a year ago with a disk problem. I can't remember when the processor went down last . . . probably about four years ago.

All the programs I wrote in the fall of 1975 are still operational, still in BASIC+. They have been through years of pruning and changing, and will probably be still functioning seven years from now on micro-J11 equipped desktop machines. They, basically, don't go down.

I still write code. I write it in BASIC+. I have learned a great deal about structured code in seven years (it didn't exist then) . . . I must say that I write much better code today than in 1975. I hope I can say the same in 1989.

Hey Carl, what about that Sabbatical???

RSTS PROFESSIONAL

Editors
R.D. Mallery
Carl B. Marbach

Assistant Editor/Advertising
Helen B. Marbach

Assistant Editor/Operations
Peg Leiby

Administrative Assistant
Hope Makransky

Editorial Assistant
Linda DiBiasio

Subscription Fulfillment
Kathi B. Campione
Claire Hollister
Steven Barsh
Margie F. Pitrone

United Kingdom Representative
Pauline Noakes
RTZ Computer Services Ltd.
P.O. Box 19, 1 Redcliff Street
Bristol, BS99-7JS
Phone: Bristol 24181

Contributors
John Cato
Steven L. Edwards
Michael C. Greenspon
Edward A. Heinrich
Greg Justice
Michael H. Koplitz
Peg Leiby
Michael Mayfield
Bob Meyer
W. Franklin Mitchell, Jr.
Paul O'Nolan
Phred
David Spencer
Ralph Zwier

Cartoons
Douglas Benoit

Game Verification Team
Bill Marbach, Mary Marbach,
Sydra Mallery, Sam Mallery

Design & Production
Grossman Graphics

Printing & Binding
Schneider Litho Co., Inc.

Cover Photo
David Sheppard
11/40 Backplane, Flea's Eye View

Editorial Information: We will consider for publication all submitted manuscripts and photographs, and welcome your articles, photographs and suggestions. All material will be treated with care, although we cannot be responsible for loss or damage. (Any payment for use of material will be made only upon publication.)

BEFORE you add memory (or anything else) to increase system performance

You should add DOPTER!

DOPTER is an easy to use RSTS/E disk copying program which

INCREASES SYSTEM PERFORMANCE UP TO 50%.

DOPTER performs all of the standard functions necessary to structure a RSTS/E disk volume and automatically does the following:

- Places all files and free space in their optimum positions.
- Produces better optimized MFD/UFD's than REORDR.
- Deletes unused file attributes from source, task, and object library files saving UFD and cache accesses.
- Places and pre-extends the MFD.
- Places the most used files at the front of the UFD's.
- Places the UFD's with the most activity toward the front of the MFD.

For More Information

If you would like more information on how you can increase the performance of your RSTS/E system with DOPTER and a free copy of "RSTS/E DISK OPTIMIZATION IN A MULTI-USER ENVIRONMENT", phone or write SPH today.

RSTS/E is a registered trademark of Digital Equipment Corporation.

SPH

System Performance House, Inc.

5522 Loch More Court · Dublin, Ohio 43017 · 614-265-7788

CIRCLE 108 ON READER CARD

LETTERS to the RSTS Pro...

Send letters to: Letters to the RSTS Pro, P.O. Box 361, Ft. Washington, PA 19034-0361.

Dear Sir or Madam:

The complaint raised by the Systems Manager for the RSTS System at the Baptist Memorial Hospital raises some interesting legal points.

There is an enormous installed base dependent upon RSTS and the PDP-11 System. It has become obvious over the last several years that the VAX family is no real improvement for most of this installed PDP-11/RSTS base, and that perhaps DEC has made another marketing error perhaps even more catastrophic than its ill-fated word processing efforts, WPS-100/102/200/202. If such is the case, market forces will soon demonstrate to DEC the error of their ways as it did in the case of the early DEC word processors.

What is most significant to me as an attorney and RSTS-dependent professional is that perhaps DEC has adopted as a corporate policy the method of planned system obsolescence and lack of software portability characteristic of certain other companies in the field. If this is the case, then we, as DEC users committed to RSTS and the PDP-11 family, have a legitimate complaint and what the law refers to as a cause of action. Any legal action should be directed against DEC management and compel them to recognize their fiduciary obligations and responsibility toward the installed user base for their major systems. As DEC advertised, promoted, marketed, and sold, so they are responsible to the users who committed their business operations to those systems. Only a corporate catastrophe on the order of reorganization under the Bankruptcy Act should limit the responsibility of a Fortune 500 company toward its customer's investment. This is even more the case where the company is primarily a hardware vendor rather than a company promoting software applications. Once before DEC demonstrated similar cavalier disregard for its installed user base when it brought forth the PDP-11 system family. Our firm was one of those so unfortunate to have committed to the PDP-8 on the advice of DEC corporate management who assured us that legal systems applications did not require the investment in a PDP-11 system, and then two years later when the PDP-8 system proved totally inadequate and its promised upgrades were never brought to market, we were left to rebuild our firm's data processing/word processing/data base management system with a used PDP-11/34 because DEC could not deliver 11/44 systems at the time we needed them.

We considered litigation at that time, but felt that we were an isolated case and that DEC, unlike IBM, had a corporate policy of user support and loyalty to its installed data base that would render it essentially immune from legal challenge. At this time, we are no longer so certain of these beliefs and are giving serious thought to raising the question in the Courts by way of a class action on behalf of all those users so unfortunate as to be similarly situated. Such an action would seek a declaratory judgement imposing upon DEC the duty to direct its systems evolution toward the defined needs of its established user base before diverting substantial corporate resources toward speculative ventures of limited utility and questionable market value. As a representative of stockholders in the company as well as a user myself, I beleive that such an action has merit unless a rational explanation is forthcoming from corporate management at DEC.

The value of RSTS and the PDP-11 family is evidenced throughout the Department of Defense, other agencies of government, hospitals such as Baptist Memorial Hospital, and of most concern to me in the arena of litigation in the rather unequal struggle between two and half million essentially destitute Viet Nam combat veterans with a PDP-11/34 operating under RSTS/E against six of the largest multinational conglomerate corporations in the chemical industry and all of their data processing resources in the Agent Orange litigation.

We would be interested in hearing the comments of other PDP-11 family users running RSTS and concerned about the corporate commitment of Digital Equipment Corporation to its installed data base.

Very truly yours,
YANNACONE & YANNACONE, P.C.
Victor John Yannacone, Jr.

—

Thought you might find this interesting. It's about the "NONAME-est" NONAME state I've ever seen. it was produced by executing a 'CHAIN' in BASIC-PLUS to —NL:'.

```
14      NB34    +BASIC  "C(0R)  2(16)N+16N       127.6(+0.0)
14      NB34    +BASIC  "C(0R)  2(16)N+16N       127.6(+0.0)
R
RUN
14      NB34    +BASIC  "C(0R)  2(16)N+16N       127.6(+0.0)
14      NB34    +BASIC  "C(0R)  2(16)N+16N       127.6(+0.0)
14      NB34    +BASIC  "C(0R)  2(16)N+16N       127.6(+0.0)
        03:29 PM        13-Jul-82

Ready
RUN
        03:30 PM        13-Jul-82

Ready
14      NB34    +BASIC  "C(0R)  2(16)N+16N       127.?(+0.1)
LIST
        03:32 PM        13-Jul-82

Ready
```

Questions:

a. Any way to get EDT to simulate TECO's /72 and /B+ switches?

b. Will future releases of CSPCOM be compatible with the patches, etc. mentioned in the article 'Getting The Most Out of CSPCOM' *(RSTS Pro., V.3, #3, Sept. 1981, p.64)* you printed a while back?

Thanks. K. Wallewein, Programmer Analyst
ATCO Ltd., Calgary Alberta

Our Editor Columnist, David Spencer, answers your first question: *Hold onto TECO because EDT will not do that for you automatically, nor does there seem to be a way of writing a macro to do it.*

b. *The use of CSPCOM for non cusps is not supported. We believe that it will continue to be functional in new releases, and if we are lucky enough to have someone send us new (if necessary) patches, we will be able to keep 'Getting the Most Out of CSPCOM'.* ***Ed.***

—

Might I suggest inclusion, every year or so, of an index based on all articles which have appeared in *RSTS PROFESSIONAL* to date.

There is so much valuable information in your magazine that it tends to get a little frustrating when one can recall seeing an article on a particular topic, but cannot remember either the issue or the heading/page under which it appeared.

E.N. Lynskey, Systems Manager
CBL Wellington, Ltd., Wellington, New Zealand

No sooner said than done, see ***RSTS Professional Cumulative Index,*** *by Greg Justice, on page 32 of this issue.*

—

The following macros, included in our standard initialiser file ($TECO.INI, loaded automatically if no user TECO.INI is available) may be helpful additions to the Erskine set *(RSTS Pro., V.4, #3, June 1982, p.84).* nMF moves forward over n words, while nMR reverses over n words. n defaults to 1 in both cases, and a word is a string of letters or digits separated by one or more of the characters loaded into Q-register Z.

As given, MR will leave the cursor at the end of a word — to get it to leave the cursor at the start of a word, include MF after the closing angle bracket.

```
! load forwards macro !
@!UF \ +OUZ QZ"E1UZ' QZ<@:S/!EGZ/ "S OA "R F>
  | F<' |0;' > \
! load reverse macro !
@!UR \ +OUZ QZ"E1UZ' QZ<R @-:S/!EGZ/ "S R -1A
  "C F> | F<' |0;'> \
! define word separators !
@!UZ" +-, ()* \ !! = : < >/[ ]"
! include tab, lf, vt, ff, cr !
9UZ 5<@ QZ :!UZ// %Z>
$$
```

Yours Sincerely, Austin Kinsella
Regional Technical College, Carlow, Ireland

—

Regarding LOGOUT's deletion of .TMP files *(August 1982, V.4, #4, Dear RSTS Man, p.36),* Mr. (Ms?) Man's answer neglected one consideration. LOGOUT will delete "????" + NUM1$(job-.num%)+."TMP". Having chosen the format xxxx**.TMP as a reasonable file name convention for some temporary files manipulated in several of our application packages, we discovered these files were "mysteriously" disappearing. After sufficient finger-pointing and "you deleted them" accusations aimed at our users (guilty until proven innocent), we suspicioned RSTS.

Sure enough, part of a successful LOGOUT includes housekeepping deletion of all files in the current account named ???? (your current job number); two positions; leading zero last.TMP. Useful when you know it.

Bob Dudley, President
Meramec Automated Solutions, St. Louis, Mo.

Note: Since ???? is ambiguous in common usage (ours, not RSTS's), strictly speaking the deletion is %%%%**.TMP in RT11's wild card format; i.e. the job number must (!) occupy positions 5 and 6 of the filename or the file will not be deleted. Wouldn't it be nice if RSTS understood RT11's wildcard * and % conventions?

—

I must respond to Michael Koplitz's article on *The RSTS/E Environment (RSTS Pro., April 1982, V.4, #2, p.74):*

The 31KW limitation is INDEED caused by a sign problem — The monitor maintains two locations: MAXLOW and MAXHI which contain respectively the highest location used by the user in the low-core PLUS one; and the LOWEST location used by the jobs RTS, with a 0 designating the NULL rts. RSTS core management routines and user mapping routines use the comparisons:

```
CMP   MAXHI, MAXLOW
                    CMP Addr, MAXLOW
BHIS . . . .
                    BLO . . . . .
```

to determine the legality of the users low core. Since the allocation unit by the monitor for user memory is 1KW, a 32KW low core, by the rules above would be represented by a 1 in MAXLOW, and a 0 in MAXHI hence causing a sign problem in the comparisons, especially the BLO. NOTE however, that when using libraries or RTS's, the full 32KW virtual address space is available.

Under RSX monitor emulation, no intepretation of the user's high core is used for pseudo-vectors. Instead, the monitor KNOWS about the user using RSX emulation, and traps errors at the

. . . continued on page 53

Picture Your Disk Files

The next time your system is thrashing around, remember this: you could be losing up to 50 percent* of your performance due to fragmented files.

DISKIT provides you with all the tools you need to create and manage well-structured disks. Use DISKIT to race through backup at 10,000 blocks/minute as it optimizes your disks.

Plus, DISKIT includes four other tools which help you improve system security, programmer productivity, and system performance.

For more information on this amazing product, please write or call.

*Based on information supplied by our customers.

Software Techniques Incorporated

Business tools for the work of man.

5242 Katella Avenue
Los Alamitos, CA 90720
United States
Phone: (714) 995-0533

287 London Road
Newbury, Berkshire RG13 2QJ
United Kingdom
Phone: 44 (0) 635-30840

CIRCLE 65 ON READER CARD

ADDLIB

ADD A RESIDENT LIBRARY WITHOUT AN ADDRESS

By Edward A. Heinrich, Real-Time Software, Inc.
420 Lexington Avenue, New York, N.Y. 10017

Introduction: With the release of V7.0 of RSTS, DEC provided us with the feature of resident libraries which can be used for a variety of applications such as interjob communications, shared data areas and/or re-entrant code that can be shared by many different tasks, i.e., RMSRES, BASICS and EDT. The major drawback of resident libraries, at least from a system management point of view, is the fact that they must be added at a specific address, which the user must calculate. Proper system management dictates that the libraries be added either at the beginning or end of user space. I personally prefer the high end of memory since we can make adjustments to XBUF and the amount of data space used for buffers without having to worry about computing new load addresses for all the libraries.

The Problem: At our installation we have several CPU's with different memory configurations. When we want to place the packs from one CPU on a different system, we have to recalculate the address to load the libraries or we wind up with either an '?Illegal byte count for I/O' error message or fragmented memory. In addition, we also have 'foreign' memory on our systems. Whenever a memory problem arises, the old "It's the foreign memory" cry is heard and we have to pull it off the bus. Again a new address must be computed in order to successfully load the libraries.

The Solution: These problems are not unique to our shop. A friend has the same problems at his installation and he suggested a program that calculates the amount of memory on the system and computes a load address for each library. Thinking it was a good idea, and not being able to resist a challenge, I have written a little utility which will add either a resident library or a runtime system at the highest available address in memory. The program checks for any locked out memory and the location of XBUF in an attempt to avoid any problems caused by disabling memory in the event of hardware problems or the placing of XBUF in an area of memory other than that immediately following the monitor and default runtime system. The program is run at system start-up time using an INIT.* command file.

The input for ADDLIB consists of the amount of memory required by the library or runtime, a slash "/", and the name of the library or runtime to load and switches. For a resident library /REM, /1USER, /RW, /NOLOG are valid switches. When a runtime system is requested, ADDLIB reads the default values from the last block of the runtime. The only valid switch for runtimes is /STAY. (What do you want from free software?) The program works to the best of my knowledge, it is currently installed on all our CPU's and at several customer sites. However, Real-Time Software makes absolutely no commitment to support it and takes no responsibility for any errors in it. The program was coded in Basic Plus 2. It can be compiled under CSPCOM if you make the variable names use '.'s instead of '__'s. If you wish to make it run under Basic Plus, I leave it to you to take it down to that level.

The following is an example of an INIT command file used at system startup to run ADDLIB.

```
; DBO: [1,2]LIB.CMD
; Function: RUN ADDLIB To add Resident Libraries
; Edit Date: 26 May 82
;
DETACH
LOGIN KB:[1,2]
FORCE KB: RUN [1,3]ADDLIB
FORCE KB: 8/CSPCOM.LIB/STAY
FORCE KB: 8/BASICS.LIB/STAY
FORCE KB: 1/SRUN.LIB<0>/REM/RW
FORCE KB: 1/[1,3]SPOOL.LIB<0>/REM/RW
FORCE KB:
FORCE KB:
FORCE KB: 21/EDT.LIB/REM
FORCE KB:
FORCE KB: BYE/F
FORCE KB:
ATTACH
```

```
!                                                                       &
!       Project         : In-House Development                          &
!       Program         : ADDLIB.B2S                                    &
!       Author          : Ed Heinrich                                   &
!                         Real-Time Software, Inc.                      &
!                         420 Lexington Avenue                          &
!                         New York, N.Y. 10170                          &
!       Function        : Add Resident Libraries and Run-Time Systems   &
!       Edit Date       : 24-May-82                                     &
!       Edit Level      : V1.00                                         &
!       Inspiration     : John Rechenberg of Avon Books                 &
!                                                                       &
!***********************************************************************&
!                                                                       &
!       Copyright (C) 1982                                              &
!       Real-Time Software, Inc.                                        &
!       New York, N.Y. 10170                                            &
!                                                                       &
!       The information in this software is subject to change without   &
!       notice and should not be construed as a commitment by           &
!       Real-Time Software, Inc.                                        &
!                                                                       &
!       Title to and ownership of this software shall at all times      &
!       remain with Real-Time Software, Inc.                            &
!                                                                       &
!       This software is unreleased and Real-Time Software makes        &
!       no commitment to support it.                                    &
!                                                                       &
!***********************************************************************&
!                                                                       &
!       ADDLIB adds Run-Time System and Resident Libraries at the high  &
!       end of physical memory without the need for the operator to     &
!       compute a location.                                             &
!                                                                       &
!       Input format: Length of memory needed, '/', and                 &
!       RTS or LIB name to add with any switches -                      &
!       /STAY for RTS                                                   &
!       /REM, /1USER, /RW, /NOLOG for LIB                               &
!       <CR> or ^Z to exit                                              &
!                                                                       &
!       Input example: 23/RMSRES.LIB/REM                                &
!       Requires 23K of memory and requests removal from memory         &
!       when no jobs are attached to it.                                &
!                                                                       &
!***********************************************************************&
```

THE HOUSE of VAX

Under one roof, Hamilton provides *all* your VAX needs. Here's how:

RENTAL Latest VAX Systems for 6 to 24 month rentals, with full upgrade flexibility, purchase option and prompt delivery.

TIMESHARING Extensive VMS based library of DEC layered products, application packages and software tools.

RSTS / RSX to VMS Conversion Services featuring fully supported PDP11 operating systems for media and software conversion.

SOFTWARE LIBRARY Including Word Processing, Data Base Management, Business & Engineering Graphics, Accounting, System Management and Accounting, Container Optimization, COBOL Program Generator.

HARDWARE CONFIGURING All DEC systems and peripherals together with Hewlett Packard and Tektronix graphic devices, Dataproducts and Diablo printers, pretested and installed on your site with manufacturers warranty and maintenance available.

VAX, VMS, PDP11, RSTS, RSX, RT11 & CTS are trademarks of the Digital Equipment Corporation

HAMILTON
6 Pearl Court, Allendale, N.J. 07401

Please rush me information on the following:
(Circle and/or fill out items below)

RENTAL 730/750/780 PDP11 _____
TIMESHARING APPLICATION _____
CONVERSION RSTS RSX RT11 CTS
SOFTWARE _____
HARDWARE _____

Name _____
Position _____
Company _____
Address _____
City _____ State _____ Zip _____
Telephone Number _____

HAMILTON

HGL Software Hamilton Rentals

6 Pearl Court, Allendale, N.J. 07401
TOLL FREE 800-631-0298
In New Jersey 201-327-1444

415 Horner Ave., Toronto, M8W4W3
416-251-1166
TOLL FREE
Ont. & Que. 800-268-2106
All Other Prov. 800-268-0317

NEW YORK • DALLAS • MONTREAL • CALGARY
LONDON • PARIS • DUSSELDORF

CIRCLE 10 ON READER CARD

```
120     !*************************************************************** &
        !                                                               &
        !                 M O D I F I C A T I O N   H I S T O R Y       &
        !                                                               &
        !       Ver   Edit Date   Who   Reason                          &
        !       ---   ---------   ---   ------                          &
        !       V1.0  29-Apr-82   eah   Original Release                &
        !                                                               &
        !*************************************************************** &

140     !*************************************************************** &
        !                                                               &
        !       P R O G R A M   V A R I A B L E S   D E F I N E D       &
        !                                                               &
        !       Variable        Used For                                &
        !       --------        --------                                &
        !       E_ND_MEMORY%    End of systems physical memory          &
        !       F_ILE$          Run-Time or Library to Add              &
        !       F_ILE%          Pointer to the last block in .RTS       &
        !       L_EN%           Length of memory needed                 &
        !       Q_%             Bit map                                 &
        !       S_WITCH$        Any switches to apply to the Add        &
        !       SYS_CALL%()     Sys call work array                     &
        !       T_%             Flag for switches present               &
        !       T_P%            Flag for either .LIB or .RTS            &
        !                                                               &
        !*************************************************************** &

800     !*************************************************************** &
        !                                                               &
        !                       W O R K   S P A C E                     &
        !                                                               &
        !*************************************************************** &
          DIM SYS_CALL%(30%)                            ! SYS work array  &
        \ DIM #1%, F_ILE%(112%,255%)                    ! .RTS virtual array &
        \ MAP (FIRQB)   FIRQB$ = 30%                    ! FIRQB mapped    &
        \ MAP (FIRQB)   FILL$  =  8%,                                     &
                        M_EMLST%                                          &

900     !*************************************************************** &
        !                                                               &
        !         C O M P I L E   T I M E   V A R I A B L E S           &
        !                                                               &
        !*************************************************************** &
          .DEFINE .PRG$ = 'ADDLIB  V1.0'                                  &
        \ .DEFINE .FNC$ = "Real-Time Software, Inc.  --  Adds RTS's and LIB's" &
                ! Some handy strings                                      &

1000    !*************************************************************** &
        !                                                               &
        !               M A I N   P R O G R A M   L O G I C             &
        !                                                               &
        !*************************************************************** &
          ON ERROR GOTO 23000                           ! Error traps     &
        \ PRINT #0%                                                       &
        \ PRINT #0%, .PRG$ + HT + RIGHT(SYS(CVT$$(1545%)),3%)             &
        \ PRINT #0%, .FNC$                                                &
                ! Program identification                                 &

2000      FIRQB$ = SYS(CHR$(6%) + CHR$(-3%))                              &
                ! Get UU.TE1                                             &
        \ T_EMP% = M_EMLST%                                               &
                ! Save it in temporary variable                          &

2100      WHILE -1%                                                       &
                \ N_EXT% = PEEK(T_EMP% + 2%)                              &
                \ GOTO 2200 UNLESS N_EXT%                                 &
                \ T_EMP% = N_EXT%                                         &
                \ X_BUF% = N_EXT% IF     (N_EXT% AND 2%)                  &
                \ L_OCK% = N_EXT% IF     (N_EXT% AND 4%)                  &
        \ NEXT                                                            &
                ! Loop a while                                           &
                ! Follow memory list pointers until there aren't any more &
                ! Look for and save XBUF and Locked out memory locations  &
                ! Save previous pointer if more                          &

2200      E_ND_MEMORY%  = PEEK(T_EMP% + 8%) / 32%                         &
        \ B_EG_XBUF%    = PEEK(X_BUF% + 8%) / 32%                         &
        \ E_ND_XBUF%    = B_EG_XBUF%  + (PEEK(X_BUF% + 6%) AND 255%) -1%  &
        \ E_ND_RSTS%    = PEEK(M_EMLST% + 6%) AND 255%                    &
        \ IF    L_OCK%                                                    &
          THEN  B_EG_LOCK%  = PEEK(L_OCK% + 8%) / 32%                     &
              \ E_ND_LOCK%  = B_EG_LOCK%  + (PEEK(L_OCK% + 6%) AND 255%) -1% &
                ! Store the end of physical memory                       &
                ! Calculate the beginning and end of XBUF                &
                ! Calculate end of monitor (Monitor 1st entry in MEMLST) &
                ! Check for locked out memory and compute it if some     &

3000    ! $code for input of data                                         &
          PRINT #0%                                                       &
        \ PRINT #0%, 'Enter Length and Name of LIB or RTS to be Added ';  &
                ! User instructions                                      &

3100      INPUT LINE #0%, T_EMP$                                          &
        \ GOTO 32000 UNLESS LEN(EDIT$(T_EMP$,-1%))                        &
        \ I_%    = INSTR(1%,T_EMP$,'/')                                   &
        \ GOTO 3100 UNLESS I_%                                            &
        \ L_EN%  = VAL%(LEFT(T_EMP$,I_%-1%))                              &
        \ T_EMP$ = EDIT$(RIGHT(T_EMP$,I_%+1%),189%)                       &
        \ CHANGE FSS$(T_EMP$,1%) TO SYS_CALL%                             &
        \ IF    RECOUNT                                                   &
          THEN  F_ILE$    = LEFT(T_EMP$,LEN(T_EMP$)-RECOUNT)              &
              \ S_WITCH$ = RIGHT(T_EMP$,LEN(T_EMP$)-RECOUNT+1)            &
              \ T_%      = -1%                                            &
          ELSE  F_ILE$    = T_EMP$                                        &
                ! Take in a command                                      &
                ! Look for amount of memory we need                      &
                ! Reprompt if we don't get an answer                     &
                ! Store it as integer                                    &
                ! Clean it up a bit                                      &
                ! Exit unless one                                        &
                ! Do a file name scan (.FSS) on it                       &
                ! IF    recount then we may have switches                &
                !       Isolate file name and switches                   &
                !       Set flag for later use                           &
                ! ELSE  Just save the whole mess                         &

3200      T_P% = 16%                                                      &
        \ T_P% = 128% IF    RAD$(SYS_CALL%(11%)+SWAP%(SYS_CALL%(12%))) = 'RTS' &
                ! T_P% is flag for RTS/LIB                               &
                ! Defaults to .LIB                                       &

3300      SYS_CALL%(I_%) = 0% FOR I_% = 11% TO 20%                        &
        \ GOTO   3400 UNLESS T_P% = 128%                                  &
        \ GOSUB 10000                                                     &
        \ GOTO   3600 UNLESS T_%                                          &
        \ Q_% = 0%                                                        &
        \ SYS_CALL%(18%) = SYS_CALL%(18%) OR 128%                         &
                IF    INSTR(1%,S_WITCH$,'/STAY')                          &
        \ GOTO 3600                                                       &
                ! Clean out part of the FIRQB                            &
                ! Skip over unless some switches                         &
                ! Look for /STAY (We take defaults from the RTS)         &

3400      Q_% = 0%                                                        &
        \ Q_% = Q_% OR 2%   IF   INSTR(1%,S_WITCH$,'/1USER')              &
        \ Q_% = Q_% OR 4%   IF   INSTR(1%,S_WITCH$,'/RW')                 &
        \ Q_% = Q_% OR 8%   IF   INSTR(1%,S_WITCH$,'/NOLOG')              &
        \ Q_% = Q_% OR 16%  IF   INSTR(1%,S_WITCH$,'/REM')                &
        \ SYS_CALL%(18%) = SYS_CALL%(18%) OR 128%                         &
                IF    INSTR(1%,S_WITCH$,'/STAY')                          &
                UNLESS    Q_% AND 16%                                     &
                ! Look for Library switches                              &
                ! Check for possible confict on stay switch              &
                ! (/REM takes precedence)                                &

3600    ! compute an address to add at                                    &
          L_OAD% = E_ND_MEMORY% - L_EN%                                   &
        \ E_ND_MEMORY% = L_OAD%                                           &
                ! Subt                                                   &
```

From now on, there's only one choice for financial modeling software: MAPS/HOST MAPS/PRO

(well, maybe two choices)

MAPS/HOST runs on DEC's PDP and VAX computers. MAPS/PRO runs on DEC's new Professional 350 desktop computer.

Both have a common planning language. Both feature large multi-dimensional worksheet capacity, user-friendly dialog, visual data editing, unlimited logic with extensive financial functions, multi-level hierarchical consolidations, "what if" and sensitivity analysis, target-seeking, a built-in financial database subsystem and business graphics.

Call us tollfree at (800) 547-1000 (in California call (415) 856-1100) or return the coupon for further information.

1900 Embarcadero Road
Palo Alto, CA 94303
(415) 856-1100

Regional Offices: New York, Dallas, San Francisco, Los Angeles

Ross Systems 1900 Embarcadero Road, Palo Alto, CA 94303

YES Send me information on MAPS/PRO and MAPS/HOST

NAME

TITLE

COMPANY

ADDRESS

CITY, STATE, ZIP

TELEPHONE

RSTS 1082

CIRCLE 1 ON READER CARD

THE RSTS CRYSTAL BALL — Part 3

By Michael C. Greenspon, Integral Information Systems, Los Angeles, California

Greetings, fellow RSTS users. This month, the Crystal Ball offers monitor enhancements, late breaking news on V7.2, and other items of interest to the RSTS community.

RANDOM

Before I dive into monitor hacking, I have a few random notes for you. The following definition was (anonymously) submitted last month. I thought I'd pass it along to you.

DCL \ dee' cee ehl \ abbr. [cs. DEC's "user interface"] 1: Decrepit Command Language. 2: Command Language of the Damned. 3: DEC's Colossal Lossage.

If you program in TECO, or even if you don't, consider this little gem. I wish I could give credit to the author, but, alas, the version I have is completely uncommented, and I have no idea where it came from. Note that any dollar signs ($) in the following listing are really ESCapes, and uparrows (↑) indicate control characters.

```
+0UN QN"E 20UN '
BUH BUV HK
QN< J BUQ QN*10/3UI
QI< \ +2*10+(QQ*QI)UA
B L K QI*2-1UJ QA/QJUQ
QA-(QQ*QJ)-2\ 10@I// -1%I ↑
QQ/10UT QH+QT+48UW QW-58"E 48UW %V ' QV"N QV↑T '
QWUV QQ-(QT*10)UH >
QV↑T @↑A/
/$$
```

Don't feel bad if it isn't immediately obvious to you what this macro does. It's name is PI.TEC. It takes one argument, the number of digits to calculate (default is 20), and outputs the value of pi on the terminal. The more digits of precision you ask for, the longer it takes to calculate each one. I have run it successfully with an argument of 1000, however, it took over a week to complete on our 70. If anyone knows who the original author is, please drop me a note.

MONITOR ENHANCEMENTS

If DEC saw the copy of RSTS that we are running, they probably wouldn't recognize it. Our monitor is quite non-standard, due to many features I have added. My latest monitor hack is an implementation of system load averages, similar to those maintained by Tenex, TOPS-20, etc.

Load average is a figure which gives an immediate, obvious indication of system loading. Basically, load average is the number of processes (or jobs, in this case) needing CPU time, averaged over a period of time. (Conversely, you can think of load average as the average number of CPUs it would take to give everyone full attention.) Three averages are maintained: 1, 5, and 15 minutes.

The averages are internally computed using pseudo double-precision math, and are normalized to 16 bits for ease of use. The averages are fixed point, and are accessed by PEEKing at cells in monitor memory. I have modified my MACRO-11 SYSTAT program to print the averages (like TOPS-20), and it would be trivial to modify DEC's SYSTAT to do the same. Eventually, I will rewrite the TTSYST code (control/T) and include the one minute load average in it (again, like TOPS-20).

The source for the load average code (LOADAV.MAC) is listed at the end of the article. The load average computation routine is called once a second through the DECnet NSP timeout hook. I used this since it is directly accessible in source form from TBL.MAC; i.e., requires no binary patches. I modified the hook so that DECnet SHOULD still work, however since our sites are running our own kludge-net for the moment, I can't guarantee this. (If you have problems, contact me and I'll investigate.)

Once a second, every second, the load average code scans JOBTBL and counts the number of jobs in a run state. Optionally, it will include jobs in a disk wait, or any other type of wait, so if your system is disk bound, the load average will still "feel right." The code then takes this count, averages it into the old buckets, and normalizes the result to 16 bits. The computation routine is all executed at priority 3, and it is very short, so system impact should be minimal. (For those of you who are still having small buffer problems, the load average code takes up about 3-4 small buffers.)

The normalized result is stored as load average multiplied by 100 decimal. The three load averages are stored in three consecutive words. The address of these words is pointed to by a word I have located in TTDINT, directly preceding TTYHCT (i.e., at TTYHCT-2). Since the TTYHCT address is returned by the get monitor tables call, it is easy to find the load averages, and without hard-coding an address in your programs.

I am currently running the load average code under V7.1, and it should work without problems under V7.0 and later. I have not encountered any complications as a result of this code, but please remember that it is unsupported and will void any DEC software service agreements you have.

Installation of the load average code is quite simple. The files TBL.MAC and TTDINT.MAC (on the RSTS sysgen tape) need to be modified. CPATCH command files to do this are listed in figures 1 and 2. The source file LOADAV.MAC must be assembled and linked with RSTS in the sysgen process. LOADAV.MAC should reside on the current account during sysgen. The CPATCH command file to modify SYSGEN.CTL

. . . continued on page 51

TRIDAKIT

The system that offers Productivity, Accuracy plus More.

Productivity Plus

PRODUCTIVITY

TRIDAKIT is a set of application development modules that automatically generates up to 90% of program code four times faster than conventional programming methods. Programmers can instruct TRIDAKIT to produce on-line applications with simple easy-to-use commands.

Incorporated in TRIDAKIT is a powerful file management system which can open and maintain 255 files simultaneously. TRIDAKIT based applications allow for caching of files, reduced use of the File Processor (FIP), and a reduction in the use of small buffers.

ACCURACY

Included with TRIDAKIT are documentation, programming, operations and project management modules. Application program code errors are practically eliminated.

Detailed documentation is automatically produced as a by-product of the application definition. Changing the documentation changes the program code. With TRIDAKIT, each member of the project team always has the latest, most up-to-date version of the documentation.

PLUS MORE

Security is strictly enforced with TRIDAKIT. Each user of TRIDAKIT must be defined before an application can be accessed.

TRIDAKIT is responsive to special problems that are commonly forgotten in applications such as audit trails, I/O tracing and "point-of-failure" data base recovery. These are automatically included in TRIDAKIT based applications.

INFORMATION

If you would like more information on how you can increase application development productivity with TRIDAKIT, phone or write today.

Tridacor Systems, Inc.
820 FREEWAY DRIVE NORTH • COLUMBUS, OHIO 43229 • (614) 431-0805

RSTS/E 3271 DATA COMMUNICATIONS

By Michael H. Koplitz

RSTS/E 3271 protocol emulator (PE) allows a user program on a RSTS/E system to exchange data with an IBM 370 series computer. The IBM can be running either CICS/VS or IMS/VS operating system. 3271 PE emulates the IBM 3271 device. The RSTS/E system requires an additional software package and some additional hardware.

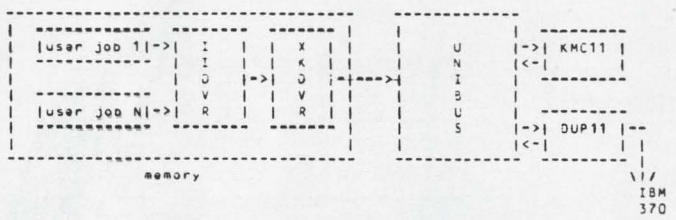
IBM's 3271

The IBM 3271 is a hardware device that controls up to thirty-two printers and display terminals. 3271 is a part of the 3270 Information Display System (IDS). The 3270 IDS subsystem is a fixed function, terminal-oriented system designed for use in an on-line interactive data communications environment. The host IBM computer looks for input data from each of its transmission control unit (TCU) resources. The TCU controls data between the 3271 units attached to it. The IBM utilizes polling . The polling employs a round-robin method to see if each TCU has any messages. Every TCU is polled for information.

RSTS/E does not use polling. RSTS/E uses a vector interrupt scheme. RSTS/E will only respond to a terminal if it informs the CPU that data is waiting. This is a more efficient system. If one user is on the IBM, he must wait for all TCUs to be examined before his entries are processed. On the RSTS/E system the one user would have exclusive use of the system because his would be the only terminal interrupting the CPU.

RSTS/E 3271 EMULATION

RSTS/E 3271 is a combination of hardware and software. The major difference between IBM and RSTS/E 3271 is that the RSTS/E 3271 requires some program interface. The user program is responsible for transmitting and processing data. A KMC11/DUP11 device pair is used to transmit data between the PDP-11 and IBM 370. The KMC11/DUP11 supports up to eight simultaneous communication links.



PDP-11 RSTS/E 3271 COMPONENTS

HARDWARE REQUIREMENTS

The hardware needed for RSTS/E 3271 communications is:

1. DUP11
2. KMC11
3. Modem

DUP11

The DUP11 is a synchronous serial line interface which is capable of "full duplex" communications. The DUP11 translates serial data to parallel data and also translates parallel data to serial data. Data is transmitted to the DUP11 from the UNIBUS in parallel. The communication channel must have serial data transmission capabilities. The DUP11 can transmit data at a maximum speed of 19200 bits per second.

The DUP11 supports DIGITAL's DDCMP and IBM's Bi-SYNC protocols. A double-character-buffer is utilized for receiving and transmitting data. This feature allows for maximized data throughput. The DUP11 can be placed in a multiport network. The DUP11 adds the SYNC characters that IBM requires and also strips these SYNC characters when data is returned from the IBM.

KMC11

The KMC11 is a general purpose microprocessor with UNIBUS compatibility. The KMC11 is used to reduce the load on the CPU and is therefore used to be a data handler rather than a data processor. The functions of the KMC11 are determined by the microprogram contained in its instruction memory. The KMC11 is not programmed to modify its own instruction area, therefore the CPU must load this area for the KMC11.

In 3271 PE communications the KMC11 performs the following functions:

1. Monitors the synchronous communications line via the DUP11 for line errors.
2. Maintains the binary synchronous communication (BSC) line discipline required for interprocessor communications.
3. Translates EBCDIC to ASCII and ASCII to EBCDIC.
4. Calculates the CRC-16 (cyclic redundancy check function 16) for both incoming and outgoing data blocks. It validates CRC-16 on the incoming data's block check characters (BBC).
5. Controls transfers of messages between itself and the PDP-11 memory. The PDP-11 CPU is not involved in this transfer.
6. Handles the conversational BSC polling sequence.
7. Handles the conversational BSC protocol as implemented by the IBM 3271.

MODEMS

Any of the following modems or equivalents is needed at both ends of the communication line.

1. Bell 212 (1200 bits/second)
2. Bell 201C (2400 bits/second)

MAXCIM™

ON LINE INFORMATION...FROM THE PRODUCTION LINE TO THE BOTTOM LINE.

Running a manufacturing plant requires a constant flow of timely information. Not just in terms of nuts and bolts. But in dollars and cents.

And MAXCIM provides precisely that. On-line, real time information... from the production line to the bottom line.

It's the ultimate manufacturing and financial management system. Designed specifically to maximize your plant's productivity. And minimize costs. Operating on-line, in an interactive mode, MAXCIM is a comprehensive system of manufacturing and financial software modules integrated through one common database... a system which provides immediate inquiry capability to manufacturing, finance, engineering and marketing. So management can control the entire manufacturing process. While it's still in process.

Designed to run on *DEC PDP-11 and VAX minicomputers, MAXCIM is user friendly. Data entry and inquiry are menu-driven via standard CRT and teleprinter terminals. So plant personnel don't have to be data processors to process the data they need. On the other hand, you'll like the system's sophistication. Modular flexibility. Multi-level security. Extensive audit trails. And graphics capability.

You'll also appreciate the complete system documentation we provide. And the professional training, installation and support services you receive. All backed by the experience we've gained working with hundreds of companies, large and small.

If you're looking at manufacturing and financial management systems, you can't afford to overlook MAXCIM. No other system is so comprehensive. So flexible. Or so well-supported. Call NCA today. Or write for our brochure.

NCA CORPORATION
Manufacturing/
388 Oakmead Parkway
Sunnyvale, CA 94086
(408) 245-7990

*DEC, PDP-11 and VAX are trademarks of Digital Equipment Corporation.

Baltimore · Boston · Chicago · Dallas · Houston · Los Angeles · Portland · Sunnyvale/Sydney, Australia

3. Bell 208 (4800 bits/second)
4. Bell 209 (9600 bits/second)

SOFTWARE REQUIREMENTS

The 3271 PE package must be purchased from DEC; this is a separate software package. There are two drivers that are added to the monitor. The microprogram for the KMC11 and two utility programs come with the package. The rest is up to the user.

SYSGEN OPTIONS

During SYSGEN time the 3271 PE support must be added to the monitor. The SYSGEN questions are:

KMC11's
00
1
Do you wish IBM protocol support *NO* YES

IIDVR

The II: driver is the input/output interface between the RSTS/E program and the KMC11 program. The II: driver also performs the following functions:

1. Strips buffer control orders from incoming data and adds a minimum set of outgoing data.
2. Maintains control blocks.
3. Maps the RSTS/E job and channel numbers to IBM terminal device addresses.
4. Monitors the amount of data being transferred, prevents too much data from being queued in the PDP-11 for any given terminal.

XKDVR

The XKDVR is the RSTS/E device handler for the KMC11 and DUP11. XKDVR initializes the KMC11 and DUP11 and monitors their activity.

KMCUT.BAS

KMCUT.BAS is a BASIC-PLUS compiled program which directs the IIDVR and XKDVR to initialize the KMC11/DUP11 communication link. It also can terminate the link. KMCUT.BAS also causes the loading of the KMC11 microprogram.

MTO.BAS

MTO.BAS allows the system manager to monitor the communication link. Communication line status can be displayed. Amounts of data going back and forth can be examined. RSTS/E buffer space, used by 3271 can be manipulated.

THE USER INTERFACE

The user interface to 3271 PE can be written in four languages: BASIC-PLUS, BASIC-PLUS-2, COBOL and DIBOL. The least cumbersome way is via BASIC-PLUS or BASIC-PLUS-2. Access to 3271 PE is done by OPENing the KMC11 on a RSTS/E channel. The KMC11 is identified as IIn:, where n can range from 0 to 7. The IIn: is analogous to the lineprinter device LPn:.

To create the data communication link with 3271 PE the program must OPEN the KMC11. The open statement is in the form:

OPEN IIn: AS FILE #X%, RECORDSIZE A%, FILESIZE C%, MODE D%

The RECORDSIZE, FILESIZE and MODE clauses are optional. The IIn: actually associates the channel with the IIDVR. The IIDVR, which is part of the monitor, is sent the data from the job. The IIDVR also transmits the information back to the job. X% is the channel number and must be between one and twelve. The RECORDSIZE clause is used to

RESPOND

SOLVES YOUR RESPONSE TIME MYSTERY

TIME: Late
PLACE: Under the gun
CLUES: The RSTS/E* system's slow. Performance is under par. Users are unhappy. Hardware budget is gone. Conclusive evidence is lacking.

SOLUTION:

RESPOND is Manus' newest RSTS/E tuning tool.
RESPOND provides the information you need to tune your RSTS/E system.
RESPOND tells you where the bottlenecks are.
RESPOND measures response time exclusive of application program inefficiencies.
RESPOND logs statistics and jobs mix.
RESPOND reports tuning statistics in usable format.
RESPOND shows resource utilizations by program, keyboard and user account.
RESPOND ranks programs by resources used.
RESPOND reports long-term trends in summarized form.
RESPOND tells who did what, when and from which keyboard.
RESPOND warns you when free disk or small buffers are critical.
RESPOND logs out inactive terminals.

*RSTS/E is a registered trademark of Digital Equipment Corporation

YES! *My system needs* **RESPOND** *Send information kit.*

Name _____
Company _____
Address _____
_____
Phone _____

MAKING COMPUTERS WORK FOR YOU

MANUS SERVICES CORPORATION
AUTHORIZED digital COMPUTER DISTRIBUTOR

MANUS SERVICES CORPORATION
SEATTLE: 1700 Westlake Avenue North, 98109 • (206) 285-3260
PORTLAND: Kristin Square, Suite 102, 9498 SW Barbur Blvd. 97219 • (503) 245-1261

CIRCLE 39 ON READER CARD

define the buffer length that will be used to store data sent and received. This value becomes the maximum amount of data that can be sent at a given time.

The FILESIZE clause is used to connect the channel to a specific IBM address. Each of the eight terminals assigned to a KMC11 is assigned a specific hexadecimal address. If the user wishes to connect directly to one of the eight addresses, the decimal equivalent of that address is placed in the FILESIZE clause.

The MODE clause is used to establish special properties for the IIDVR. These values are:

Value	Meaning
0%	Non-screen mode. This value is used if the MODE clause is omitted. PUTs are limited to 248 bytes. Do not use this mode if the host IBM is using IMS/DC.
1%	Screen-format mode. The RSTS/E 3271 PE will not add 3270 AID characters to data transmitted to the IBM system. It will not strip these characters when data is returned. If the user program is developed to handle 3277 terminal screen formats then this mode is used. PUTs are limited to 249 bytes.
1000%	Non-screen with specific address mode. This mode indicates that the user wishes to connect to a particular device address. Non-screen format is used.
1001%	Screen format with specific address mode. This mode indicates that the user wishes to connect to a particular address and use screen-format mode.
5000%	Reserved for DIGITAL field service personnel.

Some examples of KMC11 open statements:

```
OPEN "II0:" AS FILE #1%, RECORDSIZE 1920%
OPEN "II1:" AS FILE #2%, RECORDSIZE 248%, FILESIZE 193%, MODE 1001%
```

Sending data to IIDVR to be transmitted to the IBM host is done via the PRINT or PUT statement. The PUT statement must be used in conjunction with the FIELD statement. The PRINT and PUT statements are used in the same way here as they would be if the information were being displayed on a terminal. Remember that the output buffer MUST be built in such a manner that the IBM host will understand the message.

When the data buffer is transmitted to the IIDVR, user buffer space is used to hold the message until the message can be taken by the XKDVR and actually transmitted. The XKDVR then waits for the IBM to acknowledge the receipt of data. The SLEEP statement can be used to temporarily stall the job until data is received from the IBM.

```
SLEEP n% + 32767% + 1%
```

n% is the amount of seconds to wait. If there is data waiting the control is returned immediately to the program. If no data is waiting then the program sleeps. If the time expires then an error message is generated.

The GET statement in combination with the FIELD statement is used to retrieve data from the IBM. The GET statement has the same form as when doing block I/O. An error is generated if no data is waiting in the buffer.

Finally, the CLOSE statement is used to terminate the I/O over the channel. Any pending incoming messages are discarded and system buffers are returned (experience says that not always are incoming messages discarded; inform users to terminate their IBM applications before terminating the RSTS/E software).

```
CLOSE #1%
```

This CLOSE statement will terminate the data link on channel one.

REFERENCES

Digital Equipment Co. Manuals:
RSTS/E 3271 Protocol Emulator User's Guide
System Generation Manual
KMC11 Microprocessor User's Manual

Computer ills ???

We Have The Prescription

Rx When you suffer with the pain of all or part of your computer system down, call us. Tymshare computer maintenance offers prescription maintenance, time and material services, mixed vendor service support, depot level repair and refurbishment programs, and training classes covering systems operation and equipment repair.

We have what it takes to make your installation healthy and a preventive maintenance program to keep it that way.

For more information on how our maintenance programs will best fit your needs, call ....

COMPUTER SYSTEMS and SUPPORT DIVISION

TYMSHARE®

EAST: 10111 George Palmer Hwy. / Bowie, Md 20716 / 301) 459-8363
CENTRAL: 11999 Katy Freeway / Houston, Tx 77079 / (713) 870-0923
WEST: 3300 Capital Avenue / Fremont, Ca 94538 / (415) 794-2538

CIRCLE 36 ON READER CARD

PROTCT.B2S

By Ralph Zwier, Professional Software, Victoria, Australia

PROTCT is a program to toggle the protect bit of a file. To briefly recap the RSTS directory, the UFD of any RSTS account on disk contains a UFD NAME ENTRY for each file in the UFD. One of the bytes in the name entry is the STATUS BYTE which contains status information on the file. PIP prints out some of the status information in a directory command.

One of the bits in the status byte is the protect bit. If this bit is set no DELETE/RENAME is allowed by RSTS no matter how privileged the user. RSTS uses this bit to protect the following files in account [0,1]:

BADB.SYS	on all disks
SATT.SYS	on all disks
INIT.SYS	on bootable disks
MONITR.SIL	on system disks where MONITR is the INSTALLED Monitor SIL file
DEFAULT.RTS	on system disks for the default Runtime System file
SWAP.SYS	on system disks
CRASH.SYS	on system disks
ERR.ERR	on system disks

One of the items on the RSTS wish list has been a UTILITY command UT PROTECT DBO: [8,6]MYFILE.DAT which would set the protect bit on the selected file. PROTCT.B2S will protect or unprotect a file by toggling the protect bit for that file. Its basic use is to protect System Managers or other privileged users from themselves.

We use PROTCT to aid in keeping well-structured disks. We create a protected zero-length file LOCK.LCK at the beginning of each UFD to prevent users from zeroing a contiguous directory.

In theory the program could be used to modify any of the status bits:

I am particularly thinking of a situation I know of where a file was contiguous, and someone used UTILITY to flag it as non-contiguous to try to fix a Protection Violation error. The program error had nothing to do with the contiguity of the file and we wanted to flag the file as contiguous. At the time what we did was copy the file someplace else, delete it and recreate it contiguous. However, this program could have quicky been patched to toggle the contiguous bit.

I have also thought that maybe it's possible to toggle the "Marked for Delete" bit to bring back a file? I am wary of trying it because for an open file there is a memory-resident FCB which will have the MDL bit set: This means that the FCB is inconsistent with the DIRECTORY . . .

```
1         EXTEND
2       ! PROTCT.B2S                                                    &
        ! TOGGLE THE PROTECT BIT FOR A FILE                             &
        ! IN THE USTAT BYTE OF THE UFD                                  &
        !
100       DIM #2%, U%(3583%,7%)                                         &
        \ ON ERROR GO TO 19000                                          &
        !
1000    ! M A I N   P R O G R A M                                       &
          PRINT 'PROTCT - Protect/Unprotect files in a UFD'             &
        !
2000      PRINT 'PPN';                                                  &
        \ LINPUT S.PPN$                                                 &
        \ PPN$=CVT$$(S.PPN$,-1%)                                        &
        \ GO TO 32767 UNLESS LEN(PPN$)                                  &
        \ PPN$ = '(' + PPN$ UNLESS INSTR(1%,PPN$,'(') > 0%              &
        \ PPN$ = PPN$ + ')' UNLESS INSTR(1%,PPN$,')') > 0%              &
        \ INPUT 'Device';DEV$                                           &
        \ DEV$ = 'SY:' UNLESS LEN(DEV$)                                 &
        \ DEV$ = DEV$ + ':'      UNLESS INSTR(1%,DEV$,':') > 0%         &
        \ OPEN DEV$+PPN$+'/MODE:8192' AS FILE #2%                       &
        \ U.CLSZ% = U%(31%,0%)                                          &
        \ I% = FNGET.LINK%(U%(0%,0%),U.CLSZ%)                           &
        !
2500      F.NAME1$=RAD$(U%(I%,1%))                                      &
        \ F.NAME2$=RAD$(U%(I%,2%))                                      &
        \ EXT$=RAD$(U%(I%,3%))                                          &
        \ PRINT 'Blockette ';I%;F.NAME1$+F.NAME2$+'.'+EXT$;             &
        \ USTAT%= 255% AND U%(I%,4%)     ! STATUS WORD                  &
        \ USTAT$=''                                                     &
        \ USTAT$=USTAT$ + 'D' IF USTAT% AND 128%          ! MDL         &
        \ USTAT$=USTAT$ + 'C' IF USTAT% AND 16%           ! Contiguous  &
        \ USTAT$=USTAT$ + 'P' IF USTAT% AND 32%           ! Protected   &
        \ USTAT$=USTAT$ + 'L' IF USTAT% AND 2%            ! Located     &
        \ USTAT$=USTAT$ + 'W' IF USTAT% AND 4%            ! Write access&
        \ PROT% = SWAP%(U%(I%,4%)) AND 255%                             &
        \ PRINT '<';PROT%;'>';' ';USTAT$;TAB(35%);                      &
        \ GOSUB 30500    ! Do we want to toggle bit 5 (32%)             &
        \ I%=U%(I%,0%)                                                  &
        \ I%=FNGET.LINK%(I%,U.CLSZ%)                                    &
        \ GO TO 2500 UNLESS I%=0%                                       &
                !                                                       &
                ! Crawl through the NAME Blockettes displaying the file &
                ! names and the status byte together with the blockette &
                ! number (just for interest).  Jump to routine which asks&
                ! whether we want to toggle bit 5 of the USTAT.  If we answer &
                ! "YES" (sic) bit 5 gets toggled.  It may be possible for &
                ! someone to gry toggling bit 7 to GET BACK A FILE      &
                ! WHICH HAS BEEN MARKED FOR DELETION but I haven't done it &
                !                                                       &
3000      CLOSE #2%                                                     &
        \ GO TO 2000                                                    &
        !
19000   !                                                               &
        ! E R R O R   T R A P S                                         &
        !
19010     E$ = RIGHT(SYS(CHR$(6%)+CHR$(9%)+CHR$(ERR)),3%)               &
        !
19020     IF ERL = 2000 THEN PRINT "Invalid device or PPN"              &
        \ RESUME 2000                                                   &
        !
19030     IF ERL=30500 THEN RESUME 30550
19990     PRINT CHR$(7%);E$;'IN ';'PROTCT at line ';ERL                 &
        \ PRINT 'Aborting...'                                           &
        \ RESUME 32767                                                  &
        !
30000   ! F U N C T I O N   T O   G E T   T H E   L I N K   T O        &
        ! T H E   N A M E   B L O C K E T T E S                         &
        !
          DEF FNGET.LINK%(LINK%,CLUSTERSIZE%)                           &
          =(((LINK% AND 3584%)/512%)*CLUSTERSIZE%                       &
          +(SWAP%(LINK% AND -4096%)/16%))*32%                           &
          +((LINK% AND 496%)/16%)                                       &
        !
30500   ! F U N C T I O N   T O   T O G G L E   B I T   5   O F T H E  &
        ! U S T A T   B Y T E   I N   T H E   U F D                     &
        !
          INPUT NEXT$                                                   &
        \ IF NEXT$ = 'YES'                                              &
                  THEN                                                  &
                  CLOSE #2%                                             &
                \ OPEN DEV$+PPN$ AS FILE #2%,MODE 16384%                &
                \ U%(I%,4%) = U%(I%,4%) XOR 32%                         &
                \ CLOSE #2%                                             &
                \ OPEN DEV$+PPN$ AS FILE #2%, MODE 8192%                &
                !
30550             RETURN                                                &
                        !                                               &
                        ! Check whether we want to PROTECT/UNPROTECT.  If so &
                        ! reopen the UFD in secial mode and do it.  On  &
                        ! completion close the UFD in special mode and re-opens&
                        ! it safely.  We deemed it inadvisable to hold the UFD&
                        ! open in mode 16384 for the whole run, and the  &
                        ! inefficiency of the code is a small price to pay &
                        ! for the added safety.                         &
                        !
32767   END
```

EMULEX ANNOUNCES 34 NEW DISK SUBSYSTEMS FOR DEC USERS.

As a confirmed DEC user you've probably been secretly coveting some of the powerful new drives DEC doesn't offer. Or wanting lower prices and faster delivery than you've been getting.

Well, you can get them from Emulex ready to plug in and run on your DEC LSI-11, PDP-11, or VAX-11. You choose from Control Data Winchester, removable pack and fixed/removable drives up to 675 MBytes.

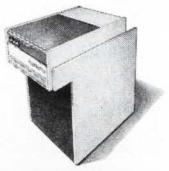
Emulex is known worldwide as *the* DEC controller expert. Over 7500 delivered. So we're a natural to provide you with the widest choice of complete disk subsystems you've ever seen.

Featuring more performance and higher reliability than you've ever dared hope for.

Plus total service support—anything from subsystem to whole system service—up to and including your DEC computer from companies like Control Data, GE, and Tymshare.

Of course, Emulex subsystems can save your company plenty. And we have an attractive rental/purchase option to solve any financing worries. So call us right now. We're the best friend a dedicated DEC user could have.

Call (800) 854-7112 toll-free, outside California. In California, call (714) 557-7580. Emulex Systems Group, 2001 Deere Ave., Santa Ana, CA 92705. TWX 910-595-2521.

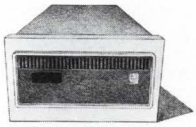


For LSI-11/23 Q-BUS: 80 MB Removable Emulates DEC RM02/03

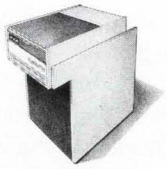
For LSI-11/23 Q-BUS: 64 MB Fixed/Removable Emulates DEC RK06/07

For PDP-11/04-60 UNIBUS: 675 MB Fixed Emulates Two DEC RM05s

For PDP-11/70 CACHE BUS: 80 MB Fixed Emulates DEC RM03

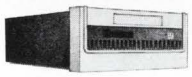


For VAX UNIBUS: 160 MB Fixed Emulates DEC RM03 Expanded



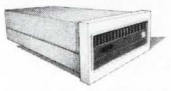
For VAX-11/750 CMI: 80 MB Removable Emulates DEC RM03

For LSI-11/23 Q-BUS: 300 MB Removable Emulates DEC RM05



For LSI-11/23 Q-BUS: 96 MB Fixed/Removable Emulates DEC RK06/07

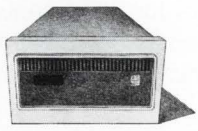
For PDP-11/04-60 UNIBUS: 32 MB Fixed/Removable Emulates DEC RK06/07



For PDP-11/70 CACHE BUS: 160 MB Fixed Emulates Two DEC RM03s

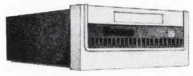
For VAX UNIBUS: 675 MB Fixed Emulates DEC RM05 Expanded

For VAX-11/750 CMI: 300 MB Removable Emulates DEC RM05



For LSI-11/23 Q-BUS: 80 MB Fixed Emulates DEC RM03

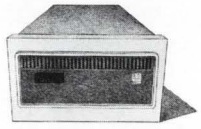
For PDP-11/04-60 UNIBUS: 80 MB Removable Emulates DEC RM02/03



For PDP-11/04-60 UNIBUS: 64 MB Fixed/Removable Emulates DEC RK06/07

For PDP-11/70 CACHE BUS: 675 MB Fixed Emulates Two DEC RM05s

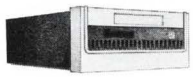
For VAX UNIBUS: 32 MB Fixed/Removable Emulates DEC RK07



For VAX-11/750 CMI: 80 MB Fixed Emulates DEC RM03

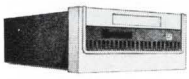
For LSI-11/23 Q-BUS: 160 MB Fixed Emulates Two DEC RM03s

For PDP-11/04-60 UNIBUS: 300 MB Removable Emulates DEC RM05



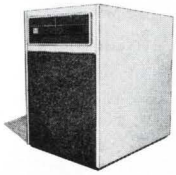
For PDP-11/04-60 UNIBUS: 96 MB Fixed/Removable Emulates DEC RK06/07

For VAX UNIBUS: 80 MB Removable Emulates DEC RM03

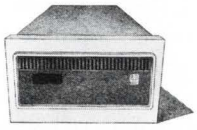


For VAX UNIBUS: 64 MB Fixed/Removable Emulates DEC RK07

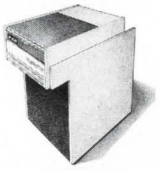
For VAX-11/750 CMI: 160 MB Fixed Emulates DEC RM80 or Two RM03s



For LSI-11/23 Q-BUS: 675 MB Fixed Emulates Two DEC RM05s



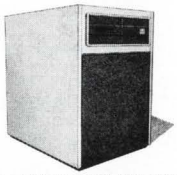
For PDP-11/04-60 UNIBUS: 80 MB Fixed Emulates DEC RM03



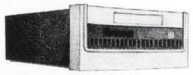
For PDP-11/70 CACHE BUS: 80 MB Removable Emulates DEC RM03

For VAX UNIBUS: 300 MB Removable Emulates DEC RM05

For VAX UNIBUS: 96 MB Fixed/Removable Emulates DEC RK07



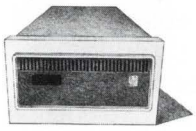
For VAX-11/750 CMI: 675 MB Fixed Emulates Two DEC RM05s



For LSI-11/23 Q-BUS: 32 MB Fixed/Removable Emulates DEC RK06/07

For PDP-11/04-60 UNIBUS: 160 MB Fixed Emulates Two DEC RM03s

For PDP-11/70 CACHE BUS: 300 MB Removable Emulates DEC RM05



For VAX UNIBUS: 80 MB Fixed Emulates DEC RM03

EMULEX

THE GENUINE ALTERNATIVE.

Call (800) 854-7112 toll-free, outside California. In California, call (714) 557-7580.

CIRCLE 58 ON READER CARD

October 1982

David Spencer, Infinity Software Corporation

EDT REVISITED

Since the last two EDT articles, there has been quite a bit of action on the EDT front. In this column I will attempt to provide an update on bugs corrected, commands added, limitations, etc.

1.0 VT52's IGNORED

In the previous two EDT articles, I made a fairly large blunder in announcing that my initializer file was good for everybody. Since I always use a VT100, I forgot about VT52 terminals.

Figure one is a list of the internal key numbering table for VT52's like the one for VT100's in the second article. I include this list in the interest of equal time. I understand that future releases of EDT will not allow defining keys by their internal number, so use this information at your own risk.

It seems a number of people have gotten confused when they tried to use my initializer file on their VT52's. My VT100 initializer file steps on some DEC keys, or defines keys that are not available on the VT52. The solution to this problem is to remove the definitions for the CONT F, GOLD arrow-up, and GOLD arrow-down keys from the initializer.

Among the people that reminded me of my forgetfulness was Larry Ingersoll of Tri-State University in Indiana. Larry also enclosed in his letter a humorous photocopy of a bogus help screen that he picked up at DECUS. I would like to thank whoever is responsible for it.

2.0 EDT RESTRICTIONS

There are two serious restrictions in EDT. The first is a key definition problem. Gordon Shepard of Dickenson College in Pennsylvania called and told me that he had written an initializer to make EDT pretend that it was WORD-11. However, he defined so many keys that he started getting the error message "That key is not definable." Well of course it was definable, so I tried it myself. I created the following ATPK command file to test the problem.

```
EDT NL:/RO,NL:
DEF K 69 AS "QUIT."
.
. (total of 200 identical lines)
.
DEF K 69 AS "QUIT."
QUIT
```

This file invokes EDT "naked"; i.e., without input and without an initializer file. On about the 195th redefinition of key 69 (CONT Y), I began receiving the "That key is not definable" error message. I SPR'd the problem, and was told that there is no garbage collection in EDT. In other words, as keys are defined the text gets appended to a working area until that space is exhausted. I was promised that the next release of EDT will have garbage collection, but that it might be a while before we will be seeing it.

The second problem is related to the first. The space for additional buffers is also the same space used by key definitions. The more keys you define, the fewer buffers you can create. This means very complex initializer files may severely limit or eliminate the ability to define additional buffers. In fact, some quite complex initializer files are never able to get off the ground.

If I may get on a soapbox for a moment, this is a very serious problem and should be remedied as soon as possible. This problem prevents EDT from operating as documented and makes its use very limited. I can't hold the developers to blame, all early versions of the code we write have bugs or oversights. Instead blame must be given to Digital product management for holding back updated releases of EDT. I'm sure that the EDT group has a great number of improvements to EDT. Either they aren't receiving the funding to get new versions of EDT out, or management is holding EDT back. In either case, we, the users, are big losers.

(By the way, some food for thought. This problem does not seem to exist on VAX.)

3.0 MARK INCOMPATIBILITY WITH VTEDIT

It has been mentioned to me that the mark for the TECO VTEDIT keypad editor is a "~ ~ / \ ~ ~" plus a carriage return. The mark I use in my initializer file is a "~ ~ / \ ~ ~" only. Ideally, these marks should be identical. However, since it would be a problem to cleanly search for the mark with the carriage return, I have decided to allow this slight difference. If it bothers you badly enough, it is possible but sloppy, to solve the problem by imbedding a carriage return; but the listing becomes hard to follow.

4.0 INITIALIZER BUGS CORRECTED

There were a few bugs in the initializer file I presented in the first article. These problems were discovered via some very thorough examination by the folks at DCA in Minneapolis. In the June 1982 issue a letter appeared of their findings. I'd like to thank them for their work and interest in EDT. I have since corrected the initializer file where needed.

5.0 NEW COMMANDS ADDED TO INITIALIZER

GOLD CONT B—Move to beginning of sentence.

This command makes use of the BSEN nokeypad command and moves the cursor to the beginning of the current sentence. This is very useful for editing text files, and even programs if your sentence delimiters are set right.

GOLD CONT F—Move to end of sentence.

This command uses the ESEN command. This works like GOLD CONT B but in a forward direction to the end of the current sentence.

GOLD CONT I—Move cursor eight characters.

I find this command useful as an "in between" from single character moves and moving big lumps at a time. This command works in either direction, and makes a line scan a lot easier.

GOLD CONT T—Make current line top line of screen.

This command uses the nokeypad command TOP. What this command does is make the current line the top line on the screen. This is useful when you are interested in what's coming, rather than what has preceded.

GOLD DELETE—Clear an entered buffer name.

This one has become useful with the buffer limitations. Type GOLD and the delete key and EDT asks for a buffer name. Enter a buffer name and it is cleared of all text so it may be used again for whatever. It even works on PASTE and MAIN!

6.0 STOCK COMMAND MODIFIED

In the letter from the people at DCA, they suggested that the HELP key (PF2) be disabled. I couldn't agree more. Most users learn the keypad very rapidly and don't need to see the keypad diagram. And accidentally striking the key becomes annoying. Of course, those who would still like to see the HELP screen can still get help by striking the GOLD key and then PF2.

7.0 MY INITIALIZER PHILOSOPHY TOWARD LANGUAGES

Lately I have seen some neat language specific initializer files floating around. One of the better ones was presented by Paul O'Nolan in the June issue. I have purposely avoided including keys for languages because the need for these keys is specific to each installation. This column is a general forum and I want to reach as many people as possible. I don't wish to crowd and confuse the file with things that not all people can use. (And in addition, as I mentioned before, there is currently a price to be paid in pool space for each key defined.)

By all means I encourage everybody to create his own key definitions for editing the languages he uses. If you have a good set of definitions, please send a copy to me and I would love to do an article on just various keys for languages. EDT is designed to help in doing whatever you're trying to do, regardless of the language that you are using.

8.0 EDT HELP FILE FORMAT

The EDT "LB:EDTHEL.HLP" file is set up in a very simple format, and therefore is easy to modify. The following example demonstrates how the HELP levels are defined and used:

```
1 COMPUTER
A computer is a big electrified rock, which is
designed to pass voltages through various other
devices (other rocks) and make alleged sense.
2 DEVICES
A device is a special type of rock attached to the
main rock (computer). This rock provides the "data"
that the big rock uses.
3 DISKS
A disk is a spinning piece of rock that is
magneticly charged to retain somewhat meaningful
information to be garbled by the big rock.
4 TERMINALS
A terminal is a rock that glows in the dark and
displays the somewhat meaningful information
stored on the spinning rock (disk) and fed to the
big rock (computer).
```

And so forth. . .

If this were your help file, you could get into line mode and type "HELP COMPUTER" and receive the text dealing with a computer being a big rock. It would then say: "Additional information available: DEVICES". If you typed "HELP COMPUTER DEVICES" then you would get the message about devices and "Additional information available: DISKS TERMINALS". And to see the TERMINALS help message type "HELP COMPUTER DEVICES TERMINALS".

The structure of the EDT help file is a level number, level name, and help text until a line starts with a number, space, and level name. In this format, you cannot get help for subtopic DEVICES without prefixing it with the help topic COMPUTER.

The help processor is set up to accept the first matching string of the help topic. If the help file contained help for COMPUTER, CIVIL-WAR, and CINDY, then typing "HELP C" would match on the help for COMPUTER, and typing "HELP CI" would match on CIVIL-WAR. To get help for CINDY, you must type enough text to uniquely match CINDY. Thus "HELP COMPUTER DEVICES TERMINALS" can be reduced to "H C D T" if you wish.

In taking a look at the "LB:EDTHEL.HLP" file, you will notice that the help is just a big text file layed out in this fashion. The keypad mode help is an interesting case because EDT looks for "KEYPAD VT100" if you are a VT100 terminal, and "KEYPAD VT52" if you are anything else. EDT then translates the keys you type into their internal number and uses that to locate help for that key. For example, invoke the keypad help and type the keypad "1" key. EDT will internally generate a search for the text "KEYPAD VT100 01" (if you happened to be set as a VT100) and display the help text for that key.

In line mode you can type "HELP KEYPAD VT100" and actually get the help screen printed for you on the terminal, with one important difference. You also get all the subtopics (i.e., keypad numbers) listed as subtopics for additional information. Armed with this knowledge you can (as I have below) add additional help for keys you have defined or changed.

In addition, you can actually redesign the help screen itself. It is quite okay to use EDT on editing the EDTHEL.HLP file (although I recommend working on a copy until you are happy with the results). Wherever EDT prints the <ESC>, it is actually an ASCII 155. So if you are going to add any escape sequences, be sure to insert a 155 escape or EDT will recognize it as a line terminator and add a carriage return on writing the file. This creates some rather strange looking help screens.

9.0 HELP FILES AVAILABLE ON MAGTAPE

I have written a HELP.BAS format help file for all the EDT commands, stock DEC and my additional ones. I have also added about four blocks of help to the "LB:EDTHEL.HLP" file to provide help for my additional commands. These files and the latest version of my initializer file are available at cost of media plus shipping and handling for $25 from my company. Undoubtably, I am breaking some DEC rule by distributing a largely DEC file. However, it is at my cost, and for a good cause, so perhaps they'll look the other way in this case.

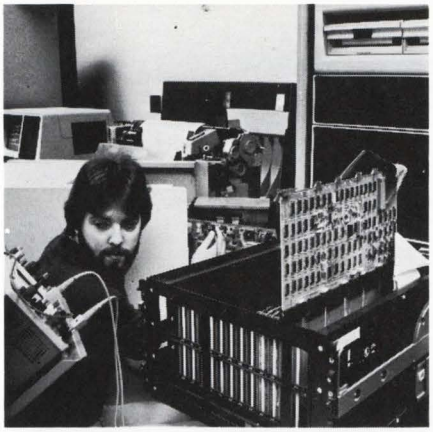
Default EDT VT52 Key Assignments

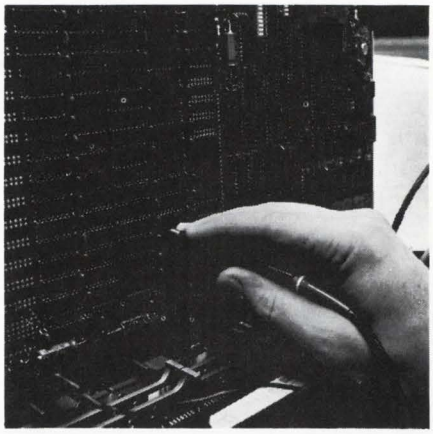
The following list shows the internal EDT key number, stock key number editing definition, and keystroke. The numbering is bound to change and/or become unavailable in future releases of EDT. Use this information at your own risk.

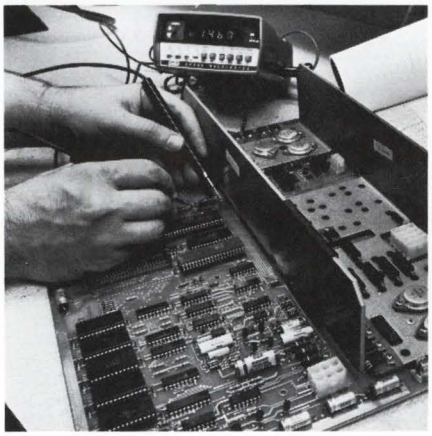
EDT Key	Definition	Keystroke
65535	D-C.	DELETE
0	L.	0(#)
1	W.	1(#)
2	EL.	2(#)
3	CUTSR.	3(#)
4	ADV.	4(#)
5	BACK.	5(#)
6	D+C.	6(#)
7	PAGETOP.	7(#)
8	"".	8(#)
9	DEW.	9(#)
10	HELP.	PF2(#+)
11	D+NL.	PF3(#+)
12	-V.	ARROW-UP(+)
13	+V.	ARROW-DOWN(+)
14	+C.	ARROW-RIGHT(+)
15	-C.	ARROW-LEFT(+)
16	SEL.	.(#+)
17	No definition	(*)
18	No definition	(*)
19	No definition	(*)
20	No definition	GOLD(*)
21	.	ENTER(+)
22	(^M-C).	GOLD 0(#)
23	CHGCSR.	GOLD 1(#)
24	D+EL.	GOLD 2(#)
25	PASTE.	GOLD 3(#)
26	ER.	GOLD 4(#)
27	BR.	GOLD 5(#)
28	UNDC.	GOLD 6(#)
29	EXT ?'Command: '.	GOLD 7(#)
30	^@?'Search for: 'i^@.	GOLD 8(#)
31	UNDW.	GOLD 9(#)
32	HELP.	GOLD PF2(#+)
33	UNDL.	GOLD PF3(#+)
34	CUTSR=DELETE PASTE.	GOLD ARROW-UP(+)
35	(16L).	GOLD ARROW-DOWN(+)
36	ASC.	GOLD ARROW-RIGHT(+)
37	APPENDSR.	GOLD ARROW-LEFT(+)
38	RESET	GOLD .(#+)
39	No definition	(*)
40	No definition	(*)
41	No definition	(*)
42	No definition	GOLD GOLD(*)
43	(CUTSR=DELETE PASTEKS"").	GOLD ENTER(+)
44	No definition	CONTROL @(*)
45	TC.	CONTROL A
46	No definition	CONTROL B
47	No definition	CONTROL C
48	TD.	CONTROL D
49	TI.	CONTROL E
50	FILLSR.	CONTROL F
51	No definition	CONTROL G
52	BL.	CONTROL H
53	TAB.	CONTROL I
54	DBW.	CONTROL J
55	DEFK.	CONTROL K
56	^L.	CONTROL L
57	^M.	CONTROL M
58	No definition	CONTROL N
59	No definition	CONTROL O
60	No definition	CONTROL P
61	No definition	CONTROL Q
62	REF.	CONTROL R
63	No definition	CONTROL S
64	TADJSR.	CONTROL T
65	DBL.	CONTROL U
66	No definition	CONTROL V
67	REF.	CONTROL W
68	No definition	CONTROL X
69	No definition	CONTROL Y
70	EX.	CONTROL Z
71	No definition	GOLD CONTROL @(*)
72	No definition	GOLD CONTROL A
73	No definition	GOLD CONTROL B
74	No definition	GOLD CONTROL C
75	No definition	GOLD CONTROL D
76	No definition	GOLD CONTROL E
77	No definition	GOLD CONTROL F
78	No definition	GOLD CONTROL G
79	No definition	GOLD CONTROL H
80	No definition	GOLD CONTROL I
81	No definition	GOLD CONTROL J
82	No definition	GOLD CONTROL K
83	No definition	GOLD CONTROL L
84	No definition	GOLD CONTROL M
85	No definition	GOLD CONTROL N
86	No definition	GOLD CONTROL O
87	No definition	GOLD CONTROL P
88	No definition	GOLD CONTROL Q
89	No definition	GOLD CONTROL R
90	No definition	GOLD CONTROL S
91	No definition	GOLD CONTROL T
92	No definition	GOLD CONTROL U
93	No definition	GOLD CONTROL V
94	No definition	GOLD CONTROL W
95	No definition	GOLD CONTROL X
96	No definition	GOLD CONTROL Y
97	No definition	GOLD CONTROL Z
98	No definition	GOLD CONTROL [(*)
99	No definition	GOLD CONTROL \(+)
100	No definition	GOLD CONTROL](+)
101	No definition	GOLD CONTROL ^(+)
102	No definition	GOLD CONTROL _(+)
103	No definition	GOLD SPACE
104	No definition	GOLD !
105	No definition	GOLD "
106	No definition	GOLD #

The Single Source for Digital Products Service

DEC Module Repair, Exchange or Replacement







Photos show how we're fully equipped to repair and test DEC,* PDP-11* and TERMINAL MODULES. All modules we repair are 100% inspected.

Visit us at booth 630

DEXPO™ WEST 82
The 2nd National DEC-Compatible Industry Exposition
Anaheim Convention Center
Dec. 7-9, 1982

DIGITAL PRODUCTS REPAIR CENTER

939 Eastwind Drive
Westerville, Ohio 43081
614/890-0939

Expert, Fast and Reliable Work

Serving the computer systems market since 1973, we are now troubleshooting, repairing and testing hundreds of modules weekly.

We can turn around a critical repair job in just 24 hours! Ten-day service is normal.

The modules we return to you are guaranteed to meet or exceed the manufacturer's specifications. And we prove it by submitting an inspection and test report.

For emergency service, 7 days a week, 24 hours a day, call

614/890-0939

*DEC and PDP are registered trademarks of Digital Equipment Corporation.

H-1580

CIRCLE 140 ON READER CARD

RELIANCE ELECTRIC

107	No definition	GOLD $
108	No definition	GOLD %
109	No definition	GOLD &
110	No definition	GOLD '
111	No definition	GOLD (
112	No definition	GOLD)
113	No definition	GOLD *
114	No definition	GOLD +
115	No definition	GOLD ,
116	No definition	GOLD -
117	No definition	GOLD .
118	No definition	GOLD /
119	No definition	GOLD 0(*)
120	No definition	GOLD 1(*)
121	No definition	GOLD 2(*)
122	No definition	GOLD 3(*)
123	No definition	GOLD 4(*)
124	No definition	GOLD 5(*)
125	No definition	GOLD 6(*)
126	No definition	GOLD 7(*)
127	No definition	GOLD 8(*)
128	No definition	GOLD 9(*)
129	No definition	GOLD :
130	No definition	GOLD ;
131	No definition	GOLD <
132	No definition	GOLD =
133	No definition	GOLD >
134	No definition	GOLD ?
135	No definition	GOLD @
136	TC.	GOLD A
137	No definition	GOLD B
138	No definition	GOLD C
139	TD.	GOLD D
140	TI.	GOLD E
141	No definition	GOLD F
142	No definition	GOLD G
143	No definition	GOLD H
144	No definition	GOLD I
145	No definition	GOLD J
146	No definition	GOLD K
147	No definition	GOLD L
148	No definition	GOLD M
149	No definition	GOLD N
150	No definition	GOLD O
151	No definition	GOLD P
152	No definition	GOLD Q
153	No definition	GOLD R
154	No definition	GOLD S
155	TADJSR.	GOLD T
156	DBL.	GOLD U
157	No definition	GOLD V
158	REF.	GOLD W
159	No definition	GOLD X
160	No definition	GOLD Y
161	EX.	GOLD Z
162	No definition	GOLD [
163	No definition	GOLD \
164	No definition	GOLD]
165	No definition	GOLD ^
166	No definition	GOLD _
167	No definition	GOLD `
168	No definition	GOLD a(*)
169	No definition	GOLD b(*)
170	No definition	GOLD c(*)
171	No definition	GOLD d(*)
172	No definition	GOLD e(*)
173	No definition	GOLD f(*)
174	No definition	GOLD g(*)
175	No definition	GOLD h(*)
176	No definition	GOLD i(*)
177	No definition	GOLD j(*)
178	No definition	GOLD k(*)
179	No definition	GOLD l(*)
180	No definition	GOLD m(*)
181	No definition	GOLD n(*)
182	No definition	GOLD o(*)
183	No definition	GOLD p(*)
184	No definition	GOLD q(*)
185	No definition	GOLD r(*)
186	No definition	GOLD s(*)
187	No definition	GOLD t(*)
188	No definition	GOLD u(*)
189	No definition	GOLD v(*)
190	No definition	GOLD w(*)
191	No definition	GOLD x(*)
192	No definition	GOLD y(*)
193	No definition	GOLD z(*)
194	No definition	GOLD {
195	No definition	GOLD |
196	No definition	GOLD }
197	No definition	GOLD ~
198	No definition	GOLD DELETE
199	No definition	[unknown]
200	No definition	[unknown]

This is a keypad key.

* This key can be defined by using the internal EDT number, but cannot be used by either case conversion constraints or key sequence interpretations.

+ This key is a valid for editing, but can be defined using the internal EDT key number only.

[4,9] EDTINI.EDT

```
!       Standard Initializer File       V01.3   25-Jun-82
!+
!       Word Delimiter Macros
!-
DEF M DELIMITERS_PROGRAMMING
F=DELIMITERS_PROGRAMMING
I
DEF K GOLD CONT D AS "EXT DELIMITERS_WORD_PROCESSING."
^Z
C; ISE EN WO '^Z 9ASC 10ASC 11ASC 12ASC 13ASC 27ASC I ()[],-+*/='^Z EX
^Z
!
DEF M DELIMITERS_WORD_PROCESSING
F=DELIMITERS_WORD_PROCESSING
I
DEF K GOLD CONT D AS "EXT DELIMITERS_PROGRAMMING."
^Z
C; ISE EN WO '^Z 9ASC 10ASC 11ASC 12ASC 13ASC 27ASC I ,'^Z EX
^Z
!+
!       Screen Width Macros
!-
DEF M WIDTH_132
I=WIDTH_132
DEF K GOLD CONT W AS "EXT WIDTH_80."
SE SC 132
SE WR 131
^Z
!
DEF M WIDTH_80
I=WIDTH_80
DEF K GOLD CONT W AS "EXT WIDTH_132."
SE SC 80
SE WR 79
^Z
!+
!       Key Definitions
!-
DEF K 10     AS "."
DEF K CONT B AS "-W."
DEF K CONT F AS "+W."
DEF K CONT G AS "PASTE=?'Put buffer: '."
DEF K CONT P AS "PAR."
DEF K CONT X AS "CUTSR=?'Cut buffer: '."
!
DEF K GOLD DEL    AS "EXT CL ?'Clear what buffer: '."
```

Run CP/M On Your DEC With The Bridge. Run inexpensive CP/M® programs on your DEC® minicomputer with The Bridge® virtual microcomputer system. Now, every terminal attached to your minicomputer can run CP/M programs directly, as if they were dedicated microcomputers. Finally — the gap between the worlds of mini and micro disappears.

The Virtual Microcomputer System. The Bridge is a system of hardware and software for your DEC LSI-11,® PDP-11,® or VAX® minicomputer. The hardware, called the z-Board,® executes the program at speeds greater than those of dedicated micros. Together they constitute a breakthrough — unlocking a vast library of cost-efficient software for your minicomputer.

The z-Board. The z-Board is a hardware accelerator for The Bridge featuring four z-80's,® 256K bytes of memory, and a bit slice state machine. A key component of the system, the z-Board provides four users (per board) with an enhanced microcomputer environment. The z-Board is available for the Q-bus® and UNIBUS.®

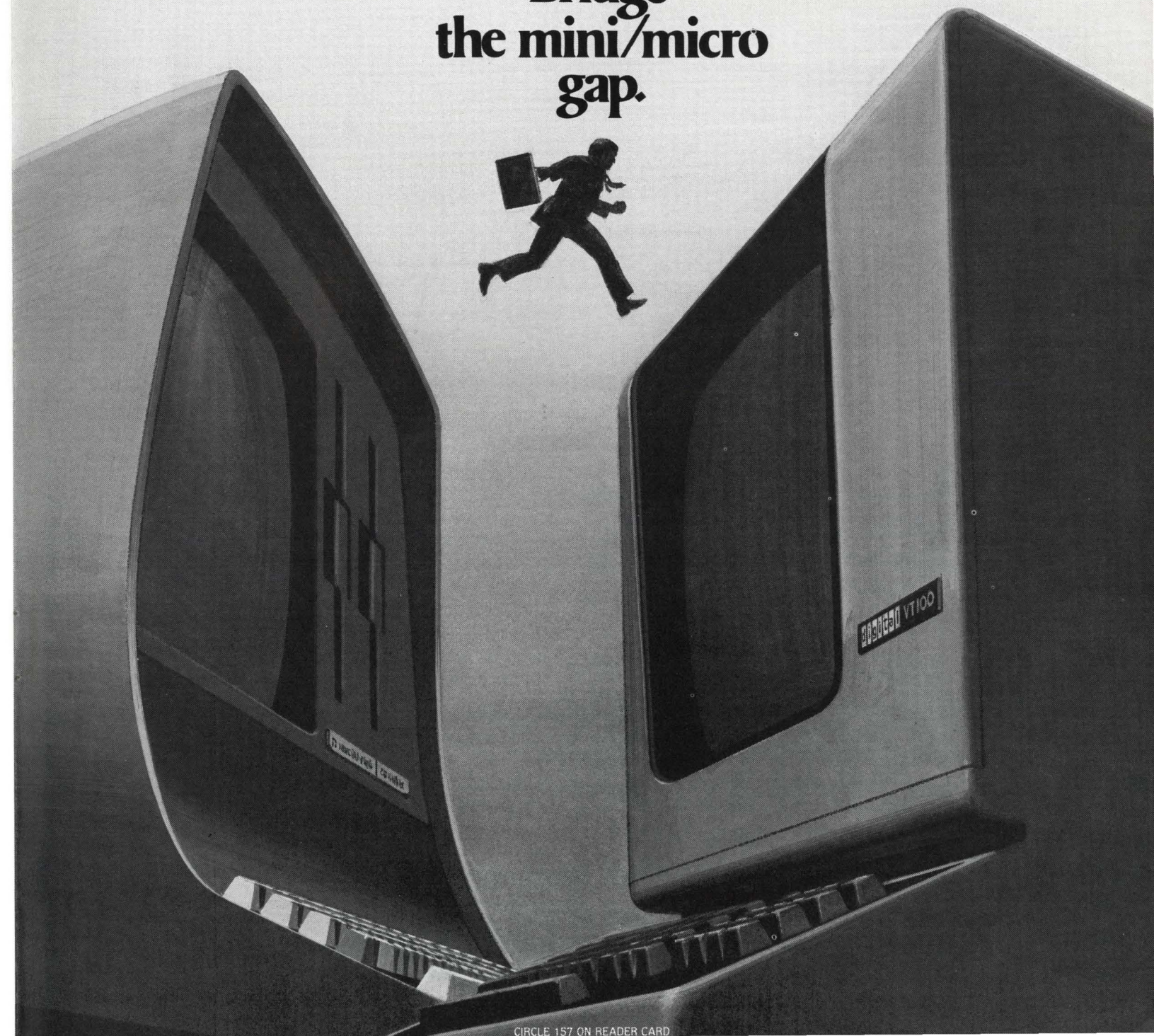
System Integration. One of the key benefits of The Bridge is that all microcomputer operations are centralized in your mini. It's no longer necessary to maintain various micros at different work stations — all with different disk formats. The Bridge gives you one format, available to each operator, at every terminal. Every virtual micro can share files with every other virtual micro in your system.

virtual microsystems

2150 Shattuck, Berkeley, CA 94704

Bridge Your Gap. Make the most of your DEC mini with The Bridge virtual microcomputer system. For more information, call Deryk Van Brunt today at (415) 841-9594.

Bridge the mini/micro gap.



CIRCLE 157 ON READER CARD

```
DEF K GOLD 12     AS "(-22V)."
DEF K GOLD 13     AS "(+22V)."
DEF K GOLD CONT B AS "BSEN."
DEF K GOLD CONT D AS "EXT DELIMITERS_WORD_PROCESSING."
DEF K GOLD CONT G AS "CUTSR=DELETE PASTE=?'Replace with buffer: '."
DEF K GOLD CONT F AS "ESEN."
DEF K GOLD CONT H AS "(-C D-C +C UNDC)."
DEF K GOLD CONT I AS "(8C)."
DEF K GOLD CONT T AS "TOP."
DEF K GOLD CONT W AS "EXT WIDTH_132."
DEF K GOLD CONT X AS "EXT CO SELECT TO=?'Copy to buffer: ' ; F L."
DEF K GOLD CONT Z AS "EXT EX."
!
DEF K GOLD . AS "I~~/\~~^z -6C."
DEF K GOLD / AS "S%~~/\~~%%."
DEF K GOLD B AS "EXT F=?'Buffer: '.."
DEF K GOLD C AS "(C SEL W CHGCSR)."
DEF K GOLD F AS "(SEL PAR FILLSR)."
DEF K GOLD I AS "EXT INC ?'Input file: ' =?' Buffer: '."
DEF K GOLD L AS "EXT F L."
DEF K GOLD M AS "EXT F=MAIN.."
DEF K GOLD O AS "EXT WR ?'Output file: ' =?' Buffer: '."
DEF K GOLD Q AS "EXT QUIT/SAVE."
DEF K GOLD S AS "EXT SH BU."
!+
!       Set Terminal Characteristics
!-
SE WR 79
SE TR
SE K
SE M C
DELIMITERS_PROGRAMMING
!+
!       Set Buffer to MAIN, and Start
!-
F=MAIN
```

[illegible]	Disp Time	Cause Bug	Go Lunch
Something	Cut	Guess	Dele User
Turn Page	Jam Keybd	Pull Plug	Throw Up
Add Junk	Go Slower	Give Up	[illegible]
[illegible]		Boardwalk	[illegible]

Notes:

1. All keys change function on Monday
2. [illegible] key is broken
3. Left arrow (→) moves cursor up
4. CTRL-Z induces sleep
5. [illegible] doesn't
6. Don't expect to [illegible]
7. [illegible] scrolls
8. Single quote (") means apostrophe
9. [illegible] garbles screen
10. Shifted numerics cause cancer
11. You are late for a meeting
12. The meeting has been cancelled
13. Type LOGOUT to enter keypad mode
14. Any key locks keyboard
15. [illegible] not implemented

[illegible]

$ _

You can write for the tape (please enclose a check) or ask questions or whatever via my company:

David Spencer
Infinity Software Corporation
2210 Wilshire Blvd., Suite 801
Santa Monica, Calif. 90403

Until next time, have a good edit. ♥

SCOPE MODE RUBOUTS IN INIT

By Michael Mayfield
Northwest Digital Software

An undocumented feature of the FILL option is very useful for systems that use a video display for their system console. This feature enables scope mode rubouts while using INIT.

To enable scope mode rubouts, reboot the system and use the following command:

Option: FILL SCOPE ♥

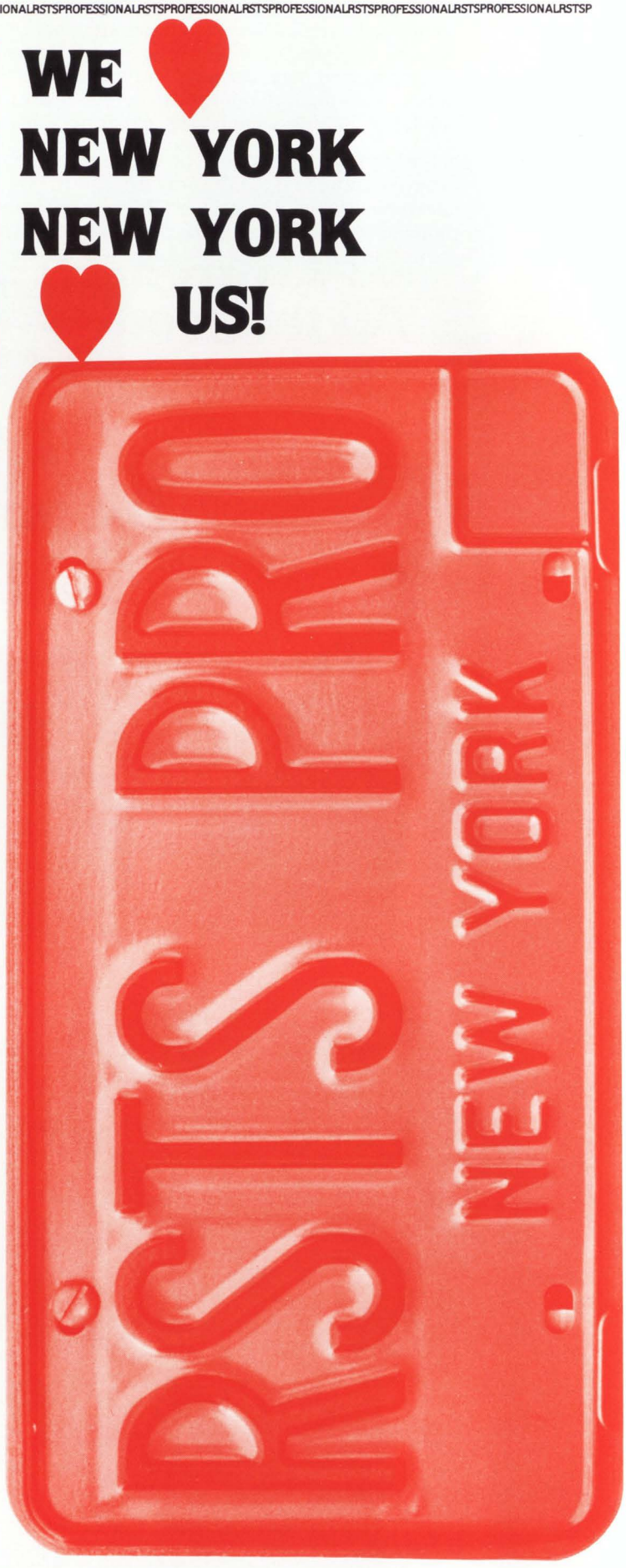


Photo submitted by Richard Rein, RMS Data Services, NY. The license plate belongs to Russell Dietz.

VT 131 from MTI. All the goodies at a lower price.

Digital's new VT131 features built-in advanced video, printer port and block mode.

Now you can get all the features of DEC's VT100 plus the options you've always wanted: advanced video, printer port, U.S./European half and full duplex communications and modem controls, and local edit/block mode transmission.

And all of these features are hundreds of dollars less than it would cost to buy them option by option for a VT100. Although the new VT 131 is totally self-contained and has no functional upgrade options, MTI can retro-fit your terminal with graphics from Digital Engineering to make it Tektronix 4010 and 4027 compatible.

MTI is one of the few Authorized Digital Terminals Distributors, so we can give you the best of two worlds; terminals, based on advanced technology from the industry leader, and the expertise and service from applications specialists.

Whether you buy, rent or lease our equipment, you'll find MTI is the one source for all the terminals, peripherals, systems, applications expertise and service you'll ever need. At good prices. Call us today.

New York: 516/621-6200, 212/767-0677, 518/449-5959
Outside N.Y.S.: 800/645-6530
New Jersey: 201/227-5552
Ohio: 216/464-6688

mti systems corp.

AUTHORIZED digital TERMINALS DISTRIBUTOR

Applications Specialists & Distributors, New York, New Jersey and Ohio.
DEC, Intel, Lear Siegler, Texas Instruments, Dataproducts, Diablo, 3Com, Hazeltine, Racal-Vadic, Digital Engineering, MICOM, Cipher and Elgar.

CIRCLE 107 ON READER CARD

MULTIPLE VERSIONS OF RSTS ON ONE DISK

By Michael Mayfield, Northwest Digital Software

Here it is Saturday. The wife and kids are at the beach but you're stuck in the computer room installing a new release of RSTS. You did the sysgen last week but you couldn't install it because RSTS doesn't allow you to have more than one release on the same disk.

Your only choice is to come in when no one else is using the system and install V7.1 on a new disk and then copy everything from your V7.0 system pack to the new disk. If you could only install V7.1 directly onto your existing system disk, you wouldn't have to wait for all that tape to spin.

Even worse, what if you find a problem in V7.1 and have to switch back to V7.0 until it is fixed. More tape spinning while you copy everything from the V7.1 disk back to the V7.0 disk.

If you were only able to have more than one release of RSTS on the system disk at once, you could switch between versions without having to copy all the files back and forth each time. If a problem occurred with the new release you could simply switch back to the previous one and continue execution.

Well, go ahead and go to the beach. This article details a method which allows any number of releases to be contained on the same disk. You can change releases any time you want. Any of the releases can be specified as the default to use for future startups.

The reason that RSTS does not allow more than one release on the same disk pack has to do with the way the bootstrap works. The bootstrap consists of a 512 byte program located at the first block of the disk. The bootstrap hardware loads this bootstrap program into memory and then executes it.

The job of the bootstrap program is to load INIT.SYS into memory and pass control to it. It does this using a list of device cluster numbers which identify the blocks that contain INIT.SYS. In order to boot a different copy of INIT it would be necessary to change the list of device cluster numbers to identify the blocks used by the new copy of INIT.

As it happens, two undocumented features of the monitor allow you to do just that. The LOAD option allows you to load a new version of INIT while at the OPTION level. The WRITEBOOT option updates the bootstrap to identify the blocks used by the version of INIT that is currently loaded.

The following procedure is used to install a new version of INIT on a disk. Please note that use of this procedure is not documented by Digital and, therefore, they have no responsibility to support it:

1. Copy the current version of [0,1]INIT.SYS into a file in [0,1] with a name other than INIT and a filetype of SAV.
2. Copy INIT.SYS for the new release onto the disk in account [0,1] with a name other than INIT and a filetype of SAV.
3. Bootstrap the disk.
4. Patch both new copies of INIT using the patch shown in the example below.
5. Use the LOAD option to load the desired version of INIT.
6. Use the WRITEBOOT option if the currently loaded version of INIT is to be used as the standard INIT during future bootstraps.

The following example shows the sequence used in installing RSTS V7.1 on a disk that already contains RSTS V7.0. INIT70.SAV is the copy of INIT for V7.0. INIT71.SAV is the copy of INIT for V7.1:

```
RUN $PIP.SAV
*[0,1]INIT70.SAV=[0,1]INIT.SYS
*[0,1]INIT71.SAV=MT0:[0,1]INIT.SYS
*^Z

Reboot the system.

Option: PATCH
File to patch? INIT70.SAV
Base address? FQINIT
Offset address? 10
 Base   Offset  Old     New
??????  000010  035171  ? %INI  ;First 3 characters of filename
??????  000012  076400  ? %T70  ;Last 3 characters of filename
??????  000014  075273  ? %SAV  ;Filetype is .SAV
??????  000016  000000  ? ^C    ;Patch complete

Option: PATCH
File to patch? INIT71.SAV
Base address? FQINIT
Offset address? 10
 Base   Offset  Old     New
??????  000010  035171  ? %INI  ;First 3 characters of filename
??????  000012  076400  ? %T70  ;Last 3 characters of filename
??????  000014  075273  ? %SAV  ;Filetype is .SAV
??????  000016  000000  ? ^C    ;Patch complete
```

To load INIT for V7.0:

Option: LOAD INIT70

To load INIT for V7.1:

Option: LOAD INIT71

To make the currently loaded version of INIT the default for future boootstraps:

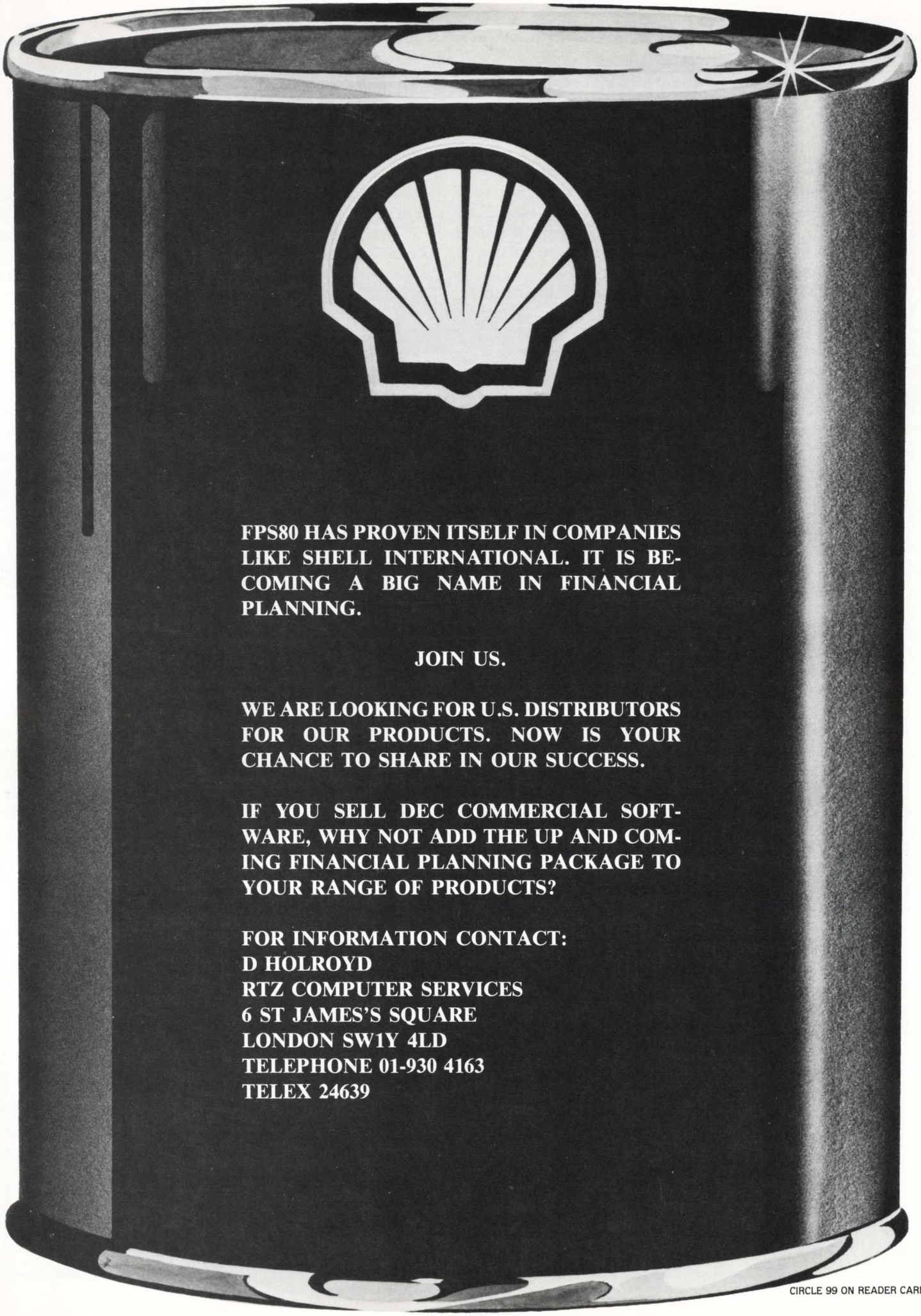
Option: WRITEBOOT

If the WRITEBOOT option is not used, the normal version of INIT will be used the next time the system is bootstrapped.

I wish to thank Software Techniques for their invaluable assistance in developing the procedure described in this article.

RSTS PROFESSIONAL CUMULATIVE INDEX

By Greg Justice, Texas Distributors, Inc., Dallas, TX 75234



FPS80 HAS PROVEN ITSELF IN COMPANIES LIKE SHELL INTERNATIONAL. IT IS BECOMING A BIG NAME IN FINANCIAL PLANNING.

JOIN US.

WE ARE LOOKING FOR U.S. DISTRIBUTORS FOR OUR PRODUCTS. NOW IS YOUR CHANCE TO SHARE IN OUR SUCCESS.

IF YOU SELL DEC COMMERCIAL SOFTWARE, WHY NOT ADD THE UP AND COMING FINANCIAL PLANNING PACKAGE TO YOUR RANGE OF PRODUCTS?

FOR INFORMATION CONTACT:
D HOLROYD
RTZ COMPUTER SERVICES
6 ST JAMES'S SQUARE
LONDON SW1Y 4LD
TELEPHONE 01-930 4163
TELEX 24639

SOME WORD PROCESSING SECURITY TECHNIQUES

By Peg Leiby

I was recently retained as a consultant by a company to assist in training and design for a large word processing application. They have dedicated a PDP 11/44 primarily to word processing with as few other data processing applications as possible.

The company receives numerous telephone or written inquiries on a daily basis. The goal is to reply promptly with a professional personalized letter which contains highly technical information. In the previous 'typing pool' operation, clerks using IBM mag-card typewriters chose from hundreds of standard form letters. Most responses were able to fit into one of these letters although it was often necessary to combine information from more than one. Considerable control and effort were required in this manual system to ensure both timely and accurate information.

WORD-11 was chosen in part because of its extensive List Processing capabilities. List Processing's basic function:

Merges a LIST document (records containing field names with variable information) with a

FORM document (a standard format with field names for variable placement) to produce an

OUTPUT document of the finished product.

A SELECTION SPECIFICATION document allows specific records in the LIST to be merged with a FORM document based on the variable information using conditional keywords, boolean operators, wildcard expressions, etc.

The SELECTION SPECIFICATION can also be set-up to access more than one FORM document during the merge process, essential since the standard letters vary each day.

List Processing also can sort records in a list by any field value. Data processing type reports can be generated by creating a FORM document in the format desired with headings, breaks, footings etc.

I expect that most of the information I give is applicable to DEC's DECWORD as well as to WORD-11. I'm not trying to teach WORD-11 here but rather to supply enough background information to make the problem and its solution intelligible.

The WORD-11 Files

There are three main types of files used by WORD-11 in an active account:

INDEX.WPS	The WORD-11 index of documents
WPS.TSK	Contains among other things stored settings and user defined sequences (referred to as UDK's; stored keystroke sequences)
WPSnnn.WPS	The actual WORD-11 documents (where nnn is the document number).

Because there are so many, the standard letters are divided across accounts by their basic technical category. While this improves directory look-up time, the protection codes of the documents have to be considered for cross-account access when combinations of documents are required.

The default protection code given to the above mentioned files is <60>. This allows read and write access only to the owner's account. There are two commands in the Document and Transfer Utility which allow a user to manipulate the protection codes:

PI for the index file (INDEX.WPS) and

PR for each individual document (WPSnnn.WPS).

Using the same project number, e.g. [200,1], [200,2], [200,3], [200,4], the codes are changed to <56> to allow read access across programmer number boundaries. ('PIP' was used to change all the documents in each account to <56> after the letters were converted.) Once a document and index have allowed this access, the CO command is used to perform the copy with a subsequent GOLD-GET (inserting one document into another) to combine text into one document.

What follows is a description of some of the WORD-11 features that we use in the system along with ways of securing them.

Library Document

This document contains many of the technical terms, standard paragraphs of the existing letters, the list processing record format, and other repetitive information.

The Library Document allows frequently used text to be easily recalled and inserted into other documents. What make this document unique are the identifiers used before the text; for example << beginning paragraph >> could be an identifier with a standard beginning paragraph to follow.

It is necessary to inform each individual WORD-11 account of the document number and location to access as its library document. This is a one time notification if the same document is to be used all the time and is handled in the Editor's Menu accessed via any WORD-11 document.

A Library Document may be shared by different accounts if it has been properly protected. A shared Library Document is easy to secure by placing it in a separate account since the password of that account is controlled.

User Defined Sequences (UDK's)

The user defined key feature (UDK), a method of storing pre-defined keystrokes, is used extensively to automate the entire list processing procedure as well as to sort lists, generate reports, etc.

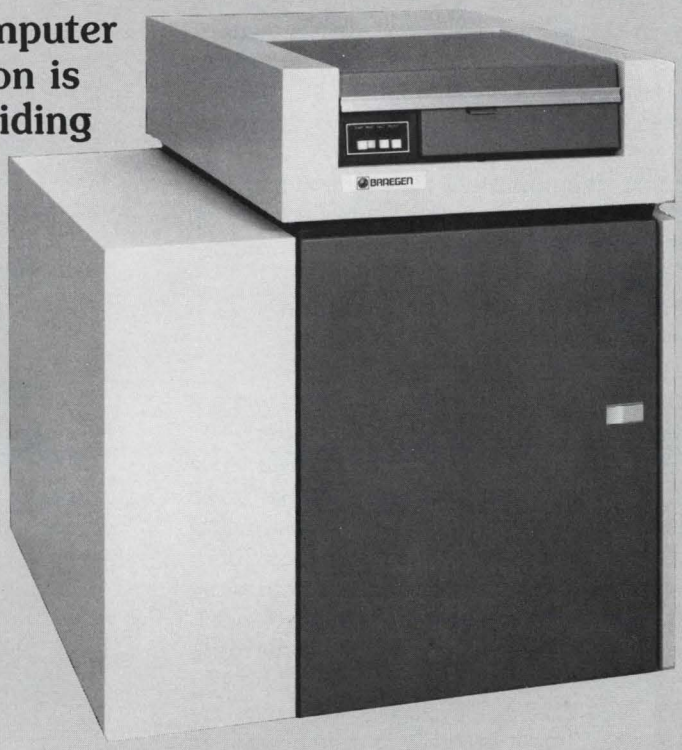
The UDK's were created in the same account as the Library Document and the WPS.TSK file containing them was then 'PIPed' to the accounts where needed. The specific documents in each account, such as the LIST, FORM, SPECIFICATIONS, OUTPUTS, etc. all needed the same WORD-11 document numbers to allow the UDK's to work.

Security can be enhanced by the periodic replacement of all WPS.TSK files from a secure source.

BRAEGEN MPD HAS THE DISK STORAGE SUBSYSTEM YOU NEED

INTRODUCING THE DSL-SERIES OF DISK STORAGE SUBSYSTEMS

Braegen's Minicomputer Peripherals Division is committed to providing the systems and peripherals DEC-users need.



BENEFITS

- Compatible with *DEC *PDP-11 and *VAX Massbus systems.
- Provides *RM03/RM05 Emulation with enhancements.
- Supports mixed drive sizes.
- Supports DEC Dual Port Diagnostics.
- Supports Overlap Seeks.
- Fast delivery.
- Local service and quantity discounts available.

For more information on the DSL-Series of mass storage subsystems or any of our full line of DEC-compatible computer systems, disk and tape subsystems, printers, memories, multiplexers and more, please call or write:

Marketing Dept:

BRAEGEN MPD

3320 East La Palma Avenue
Anaheim, California 92806
Telephone (714) 520-9200

"Be sure to ask about our field service force in over 50 cities nationwide."

*Trademarks of Digital Equipment Corporation

CIRCLE 95 ON READER CARD

Standard Letters

The technical content is such that changes to the standard letters can and do occur on a regular basis. Control of these documents is of utmost importance as the information is used by the recipients as 'truth'. While human error is the primary concern, maliciousness is not ruled out as a possibility.

Placing these sensitive documents in other accounts where the password could be controlled will not work because WORD-11 requires all documents merged during List Processing to be in the same account.

It initially appeared that a customized program would be required to maintain control of these sensitive documents. A file would be created to store the document numbers and the date of the most recently approved change. A look-up in the RSTS directory would be made to compare the actual date a document was changed (per the directory) to the date of an approved change (in the file). Maintaining this file of sensitive documents would require manual up-keep.

'Write Protecting' the document files by adding '2' to the protection code wouldn't work because while preventing editing or deleting from occurring, list processing wouldn't work either. The error message

?'n' is in use or protected against modification

(where 'n' is the document number)

occurs while attempting the merge process.

Unauthorized deletion of a letter will show up when a UDK trying to access the document halts the process.

The solution for securing the letters still seemed to be the program and file mentioned above. It occurred to me that if there were something unique about these documents perhaps the RSTS directory could be used instead of a separate file. . . oh yes, the <56> protection code to allow the copying across accounts. A program would still be needed; however, now a much simpler version could be written. The date would be checked on only those files with a protection code of <56>. The date of last access is changed by WORD-11 whenever a document is edited or even printed. It is not changed when accessed by LIST processing.

A listing of the program follows. The program is executed by typing 'F Secure' at the Main Menu. The program scans each directory using a PPN list kept in an ASCII parameter file. When it encounters a WORD-11 document with a protection code of <56> and a date-written greater than the date of the last run (stored and updated in the parameter file) it creates a list processing compatible record in an output file. There is no need for a maintenance program for the parameter file as any editor can be used to add or change PPN's.

A subsequent chain to the WPSDAT program converts the ASCII text to a WORD-11 document. Finally, WPSDAT returns the user to the WORD-11 Main Menu, giving the user the impression that the terminal has always been controlled by WORD-11. Then a UDK is invoked to sort the list in document number order and execute list processing, printing the desired report.

It should be noted that this is not a fool-proof security system since the users could change protection codes if they knew how. This possibility is minimized since the clerks are kept in the WORD-11 environment. In addition, since they are unaware of the protection code process in the first place, it is unlikely that the PR option would ever be abused. The program used with managerial control does provide reasonable protection.

```
1       EXTEND
2       !                                                               &
        !       SECURE.BAS      Peg Leiby 4/82                          &
        !                                                               &
        !
90        ON ERROR GOTO 19000                                           &
        !
200       F.PAR$ = 'SY:SECURE.DAT'              ! PARAMETER FILE        &
        \ F.LST$ = 'SY:SECURE.LST'              ! LISTING FILE          &
        !                                                               &
        \ F.PAR% = 1%                                                   &
        \ F.LST% = 2%                                                   &
        !
800       GOSUB 6000                            ! OPEN PARAMETER FILE   &
        \ GOSUB 6200                            ! ALLOW DATE CHANGE     &
        !                                                               &
        \ GOSUB 7000                            ! OPEN LISTING FILE     &
        !
1000    !       MAIN LOOP                                               &
        !                                                               &
          FOR PPN% = 1% TO PAR.TOT%             ! LOOP THRU PPN'S       &
        !                                                               &
        \ V$    = SYS(CHR$(6%)                  ! STRING SCAN FOR DIR TO CHECK &
                + CHR$(-10%)                                            &
                + PAR.PPN$(PPN%))                                       &
        !                                                               &
        \ PPN$ = MID(V$,5%,2%)                  ! PROPER PPN FRMT       &
        \ DSK$ = MID(V$,23%,4%)                 !        DEVICE         &
        \ UNT$ = MID(V$,25%,1%)                 !        UNIT           &
        !
1030      PPN.IDX% = 0%                         ! SET INDEX LOOK-UP TO 1ST &
        !
1050      GOSUB 4000                            ! DIRECTORY SYS CALL LOOK-UP &
        \ GOTO 1200     IF      E%              ! EOF (DIRECTORY)       &
        !                                                               &
        \ GOSUB 4200                            ! VALIDATE              &
        !
1100      GOSUB 7200    IF      LST.FLG%        ! DESCREPANCY           &
        \ GOTO 1050                                                     &
        !
1200      E% = 0%                                                       &
        \ NEXT PPN%                                                     &
        !
1400    !       TERMINATION ROUTINE                                     &
        !                                                               &
          GOSUB 6400                            ! STORE TODAY'S DATE    &
        \ GOSUB 7600                            ! CONVERT LIST TO W11   &
        !
1490      GOTO 32767                                                    &
        !                                                               &

4000    !       SYS CALL ROUTINE                                        &
        !                                                               &
          V$    = SYS(CHR$(6%)                  ! SYS CALL TO FIP       &
                + CHR$(15%)                     ! DIRECTORY LOOK ON IDX CODE &
                + CHR$(PPN.IDX%)                ! INDEX OF FILE TO READ &
                + CHR$(SWAP%(PPN.IDX%))                                 &
                + PPN$                                                  &
                + STRING$(16%,0%)               ! UNUSED                &
                + DSK$                          ! DEVICE TO READ        &
                + UNT$)                         ! UNIT TO READ          &
        !
4050      WORD1% = SWAP%(CVT$%(MID(V$,7%,2%)))  ! RSTS FILE NAME        &
        \ WORD2% = SWAP%(CVT$%(MID(V$,9%,2%)))                          &
        \ WORD3% = SWAP%(CVT$%(MID(V$,11%,2%)))                         &
        \ FIL.NAM$      = RAD$(WORD1%)                                  &
                        + RAD$(WORD2%)                                  &
                        + "."                                           &
                        + RAD$(WORD3%)                                  &
        !
4060      FIL.PRT% = ASCII(MID(V$,15%,2%))      ! PROTECTION CODE       &
        !
4080      FIL.DAT% = SWAP%(CVT$%(MID(V$,17%,2%)))                       &
                                                ! LAST ACCESS DATE       &
        !                                                               &
        \ PPN.IDX% = PPN.IDX% + 1%              ! FOR THE NEXT LOOK-UP  &
        !
4090      RETURN                                                        &
        !
4200    !       VALIDATION ROUTINE                                      &
        !                                                               &
          LST.FLG% = 0%                         ! INITIALIZE            &
        \ GOTO 4290     UNLESS  FIL.PRT% = 56%  ! THE ONLY ONES TO CHECK.. &
        !                                                               &
        \ GOTO 4290     UNLESS  LEN(FIL.NAM$) = 10%                     &
                        AND     LEFT(FIL.NAM$,3%) = 'WPS'               &
        !
4250      IF    FIL.DAT% > PAR.DAT%                                     &
          THEN  LST.FLG% = -1%                  ! FOUND DESCREPANCY.. FLAG IT &
        !
4290      RETURN                                                        &
        !
6000    !       OPEN PARAMETER FILE                                     &
        !                                                               &
          OPEN F.PAR$ AS FILE #F.PAR%                                   &
        !
6050      INPUT LINE #F.PAR%, A$                                        &
        \ PAR.TOT% = VAL(CVT$$(A$,4%))                                  &
        !
6060      FOR PPN% = 1% TO PAR.TOT%                                     &
        !                                                               &
        \ INPUT LINE #F.PAR%, A$                                        &
        \ PAR.PPN$(PPN%) = CVT$$(A$,4%)                                 &
        !                                                               &
        \ NEXT PPN%                                                     &
        !
6070      INPUT LINE #F.PAR%, A$                                        &
        \ PAR.DAT% = VAL(CVT$$(A$,4%))                                  &
        !
6090      RETURN                                                        &
        !
6200    !       ALLOW DATE CHANGE                                       &
        !                                                               &
```

```
        PRINT 'Last date checked '; DATE$(PAR.DAT%)                    &
      \ PRINT CHR$(7%)                                                 &
      \ INPUT 'Enter new date (MMDDYY) or <cr> '; X$                   &
      \ X = VAL(X$)                              ! ERR -> 6230         &
        !                                                              &
      \ PAR.DAT% = FND.AY%(X) UNLESS X$ = ''                           &
      \ GOTO 6250                                                      &
        !
6230    PRINT 'Invalid date'                                           &
      \ GOTO 6200                                                      &
        !
6250    PRINT 'Checking the following accounts:  '                     &
      \ PRINT USING '##  \        \' I%, PAR.PPN$(I%)                  &
        !                                                              &
              FOR I% = 1% TO PAR.TOT%                                  &
        !
6290    RETURN                                                         &
        !
6400    !       RE-WRITE PARAMTER FILE WITH TODAY'S DATE               &
        !                                                              &
        KILL F.PAR$                                                    &
      \ OPEN F.PAR$ AS FILE #F.PAR%                                    &
        !
6450    PRINT #F.PAR%, PAR.TOT%                                        &
      \ PRINT #F.PAR%  PAR.PPN$(PPN%)                                  &
        !                                                              &
                    FOR PPN% = 1% TO PAR.TOT%                          &
        !                                                              &
      \ PRINT #F.PAR% USING '#####', FNJ%                              &
        !                                                              &
      \ CLOSE #F.PAR%                                                  &
        !
6490    RETURN                                                         &
        !
7000    !       OPEN LISTING FILE                                      &
        !                                                              &
        OPEN F.LST$ FOR OUTPUT AS FILE #F.LST%                         &
        !
7090    RETURN                                                         &
        !
7200    !       PRINT WORD-11 TEXT FILE                                &
        !
7250    PRINT #F.LST%                   '<ppn>';        PAR.PPN$(PPN%)  &
      \ PRINT #F.LST%                   '<doc>';        MID(FIL.NAM$,4%,3%) &
      \ PRINT #F.LST%                   '<date>';       DATE$(FIL.DAT%) &
      \ PRINT #F.LST%                   '<last date>';  DATE$(PAR.DAT%) &
      \ PRINT #F.LST%                   '<>'                            &
      \ PRINT #F.LST%                                                   &
      \ PRINT #F.LST%                                                   &
        !
7290    RETURN                                                         &
        !
7600    !       LISTING CLOSE & WORD-11 CHAIN                          &
        !                                                              &
        CLOSE #F.LST%                                                  &
        !
7650    COR.COM$        = F.LST$ + CHR$(13%)                           &
                        + '(DM1) LIST: P: Security' + CHR$(13%)        &
                        + 'NO' + CHR$(13%)                             &
                        + 'WPSMEN.BAC[11,0]' + CHR$(13%)               &
                        + CVT%$(0%)                                    &
        !
7660    Z$ = SYS(CHR$(8%) + COR.COM$)                                  &
        !
7680    CHAIN '[11,0]WPSDAT' 30000%                                    &
        !
7690    RETURN                                                         &
        !
19000   E% = ERR                                                       &
        !
19050   RESUME 4090   IF      ERR = 5%                                 &
                      AND     ERL = 4000%                              &
        !                                                              &
      \ RESUME 6230   IF      ERL = 6200                               &
        !
19999   PRINT 'RSTS Error #: '; E%; 'at line number '; ERL             &
      \ STOP                                                           &
        !
20700   !       FLOATING MMDDYY TO JULIAN%                             &
        !                                                              &
        DEF FND.AY%(D)                                                 &
        !
20702   IF      D < 010170.                                            &
        THEN    D7%=-1%                                                &
      \         GOTO 20708              ! EARLIER THAN ALLOWED         &
        !
20703   D8%,D1=D/10000.                                                &
      \ D=D-(INT(D1)*10000.)                                           &
      \ D2,D9%=D/100.                                                  &
      \ D0%=D-(D9%*100%)                ! DECOMPOSE MMDDYY TO D8%,D9%,D0% &
        !
20706   D7% = (D8%-1%)*30% - (D8%>1%) + (D8%>2%)                       &
      \ D0=D0%                                                         &
      \ D7%   = D7%                                                    &
              + ((D8%>2%) AND ((D0/4%) <> INT (D0/4%))) - (D8%>3%)     &
              - (D8%>5%) - (D8%>7%) - (D8%>8%) - (D8%>10%)             &
              + ((D0% - 70%) * 10% ** 3%) + D9%                        &
        !
20708   FND.AY% = D7%                                                  &
      \ FNEND                                                          &
        !
20712   !       CHANGE TODAY'S DATE TO JULIAN%                         &
        !                                                              &
        DEF FNJ%                                                       &
      \ TO.DAY$ = DATE$(0%)                     ! TODAY IN DEC FORMAT  &
        !
20716   MMDDYY$ = MID(TO.DAY$,4%,2%)                                   &
                + RIGHT(TO.DAY$,7%)                                    &
                + LEFT(TO.DAY$,2%)                                     &
        !                                                              &
        !
20718   FNJ% = FND.AY%(VAL(MMDDYY$))                  !TODAYS JULIAN DATE &
        !
20720   FNEND                                                          &
        !
32767   END
```

TIPS & TECHNIQUES

A Column For The Advanced RSTS/E User

Steven Edwards, Software Techniques, Inc.

Optimizing BASIC-PLUS-2 Programs Through Profiling

Profiling is a technique that can help you tune a program by detecting which lines (sections, subprograms) are executed most frequently. After all, why spend your time optimizing a line of code that is executed only a few times. First, a few caveats:

1. although significant improvements in execution time can be realized if you know where to look, no amount of optimizing can save a poor algorithm.
2. all lines are not created equal — some lines take considerably more computer resources than others. This technique cannot differentiate multi-statement lines.

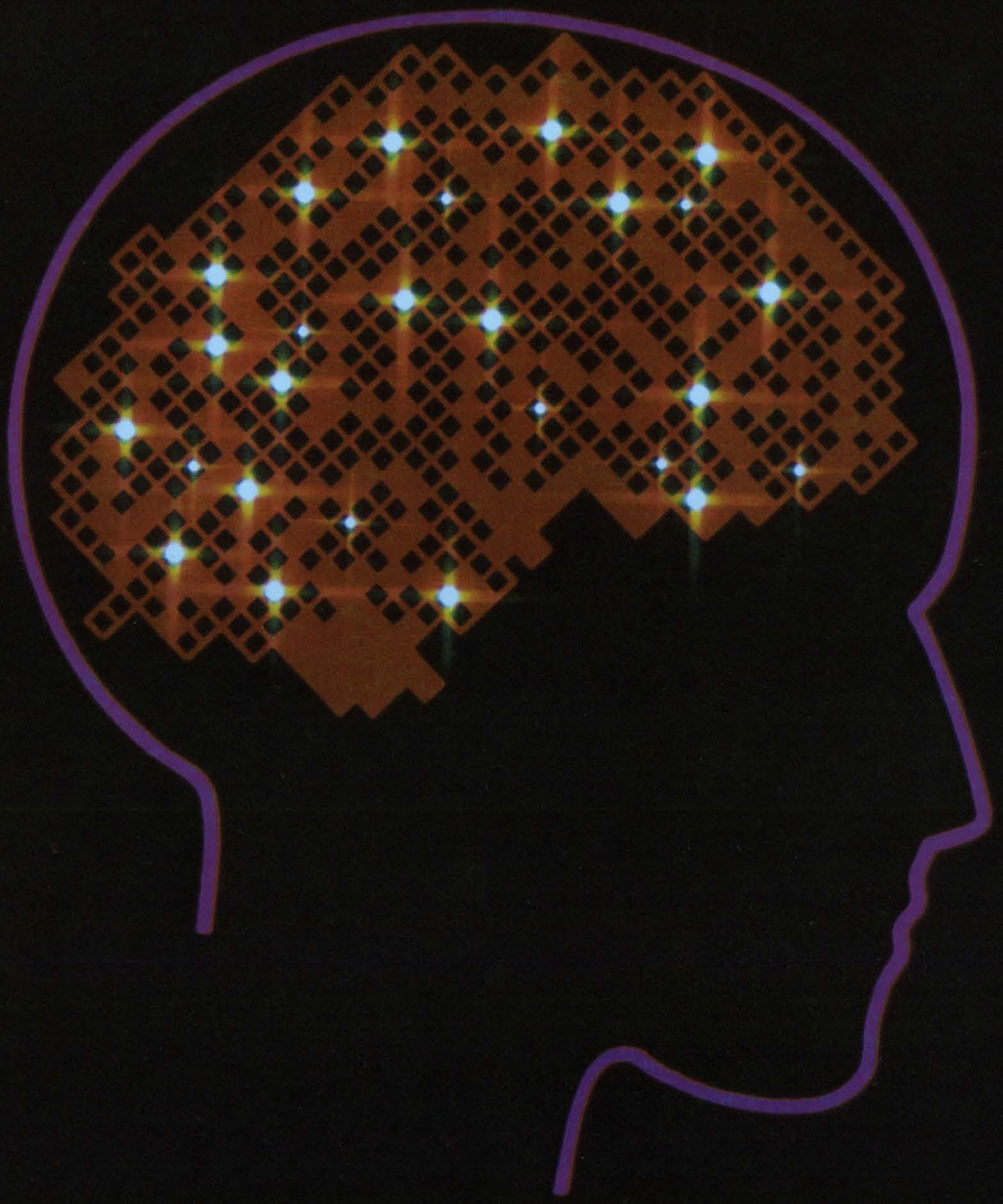
This technique was implemented by writing a short piece of MACRO-11 code named BP2PRO. BP2PRO replaces the BASIC-PLUS-2 LIN$ thread routine and prints the current module name and line number to the user's terminal. This data can be captured and sorted into meaningful information.

Let's take a 'real-life' example. Several years ago, there was a program on one of the RSTS SIG tapes (authored by Richard Case) named TRACE. TRACE 'traces' the directory blocks read to locate a file. In effect, it shows you how many directory blocks FIP will have to read to open a file. The version of TRACE that we use in-house is loosely based on Richard's program. We'll return to the original program (TRANSlated to BASIC-PLUS-2) for our example.

Here is the procedure to profile a BASIC-PLUS-2 program:

1. Assemble the BP2PRO module, using COMMON (from the RSTS distribution tape) and PRE (from the BASIC-PLUS-2 distribution tape).
2. Compile the program and link with the BP2PRO module.
3. Run the program via ATPK to collect the profiling data.
4. Sort the log file.
5. Count the number of times each line was executed. I use the UNIQ program from the DECUS "C" language tape to count the number of times each line was executed. If you don't have this program it is a minor task to write a BASIC program to do the same function.
6. Sort the log file from running UNIQ to produce a file listing the lines by frequency of occurrence.

CHANGE YOUR MIND



Software Techniques Incorporated, one of the world's leading minicomputer consulting groups, announces an exciting educational experience that will change **your** mind about planning, developing, and maintaining software systems.

"Advanced Business Software Development" is an 8-hour seminar held the weekend before DECUS at our new conference center (just minutes away from the Disneyland Hotel).

Now, you can learn the latest techniques in:

- Systems Analysis
- Functional Specification
- Tool-oriented Solutions
- Information Flows
- Advanced Development Techniques
- BASIC-PLUS-2
- The RSTS/E Environment
- Data Structures
- System Controls

Your $585 registration fee covers everything: seminar, course materials, lunch, and comfortable transportation from the Disneyland Hotel. Hours are 9 AM to 6 PM on December 4, 1982.

Call for your reservation. Space is limited.

Software Techniques Incorporated

5242 Katella Avenue, Los Alamitos, CA 90720 (213) 594-9405 (714) 995-0533

PROFIL.SRT

```
  1  TRACE 00090
       ...
 76  TRACE 00590
266  TRACE 00700
266  TRACE 00710
266  TRACE 00730
266  TRACE 00740
344  TRACE 01010
344  TRACE 01020
344  TRACE 01030
344  TRACE 01040
344  TRACE 01050
344  TRACE 01060
344  TRACE 01070
344  TRACE 01080
344  TRACE 01120
344  TRACE 01160
344  TRACE 01170
344  TRACE 01180
```

Using this procedure on TRACE we learn that 92 different lines were executed a total of 5642 times. We can see from the portion of PROFIL.SRT above, that 16 of the lines (about 15%) account for about 90% of the lines executed. Clearly these lines are where we should look to optimize this program.

Upon examining the source to the program, we see that these lines comprise 2 sections of code. The first section (figure 1) determines if the requested file has been found, and the other section (figure 2) unpacks the directory link from the buffer.

Let's see what we can learn from these 2 sections of code (listed in order of contribution to reducing execution time).

1. BASIC-PLUS-2 allows the programmer the means to access word oriented data structures in I/O buffers (via MAP statements) much more efficiently than BASIC-PLUS.

   Using a MAP statement to define the I/O buffer as a large word oriented array allows us to re-write line 1160 to be more efficient.

   NOTE

   Line 1160 contains a "FOR" modifier. BP2PRO counts all 8 executions of statement 1160 as 1 line. Thus statement 1160 is really executed 8 times as frequently as shown in PROFIL.SRT.
2. Most of the lines involve a reference to the array W%( ). However, the subscript is almost always a constant! Array address calculation is a fairly expensive operation compared to a simple variable reference (11 instructions v. 2 instructions).

   We'll re-write the references to W%( ) to reference separate simple integers instead.
3. Lines 1010-1050 determine the address of the desired blockette within the UFD, using a flag word (T6%) to indicate if the sign bit is on in the link word.

   We can simplify lines 1010-1050 a little bit so that the flag word is not needed.
4. BASIC-PLUS-2 (unlike some other languages) guarantees that all expressions in an IF statement are executed, even if the first expression is sufficient to determine the final outcome.

   The IF statement on line 730 determines if the current file is the desired file by comparing the 3 RADIX-50 encoded words that make up the file name and the file type. Obviously, if the first word of the file name does not match (which is the most frequent condition), we know the current file cannot be the desired file.
5. BASIC-PLUS-2 evaluates a number of expressions at compile time.

   The expression "NOT 15%" on line 740 is not one of them. We'll save a few instructions by evaluating it ourselves.

The 'optimized' sections of code are presented in figures 3 & 4.

Summary

We have limited our effort to 2 short sections of code. Profiling the program indicated where our effort could best be applied—and where it could not. The result of this exercise is that the 'optimized' version of the program takes less than ⅓ of the CPU time than the original version did.

```
700     X = FNB(W%(1%))
!       GET FIRST NAME BLOCK

710     GOTO 730
                UNLESS   L%
!       SKIP THE FILE PRINT OUT UNLESS WANTED

        ...

730     IF      W%(2%) = F1%
        AND     W%(3%) = F2%
        AND     W%(4%) = E%
        THEN    GOTO 760
!       IS THIS THE FILE?

740     IF      (W%(1%) AND (NOT 15%)) <> 0%
        THEN    GOTO 700
!       GO TRY AGAIN UNLESS THE FIRST WORD IS ZERO (AFTER THE FLAG
!       BITS HAVE BEEN TRIMMED).
```

FIGURE 1

```
1010    IF      I% < 0%
        THEN    T6% = -1
\               I% = I% AND 32767%
!       CHECK FOR NEGATIVE BLOCKS

1020    T1% = (I% / 4096%) AND 15%
!       SECTOR IN CLUSTER

1030    IF      T6% = -1
        THEN    T1% = T1% + 8%
\               T6% = 0%
!       TAKE CARE OF NEGATIVES

1040    T2% = (I% / 512%) AND 7%
!       CLUSTER MAP POINTER

1050    T3% = I% AND 496%
!       BLOCKETTE - DROP LOW ORDER 4 BITS

1060    T4% = D%(T2% + 1%) + INT(T1% / C%)
!       CLUSTER NUMBER IN BLOCK

1070    T5% = T1% - C% * INT(T1% / C%)
!       SECTOR POINTER IN BLOCK

1080    IF      T4% = S2%
        THEN    GOTO 1120
!       IS BLOCK IN CORE?

        ...

1120    IF      T5% = S3%
        THEN    GOTO 1160
!       CHECK IF SECTOR IS MAPPED

        ...

1160    W%(T9%) = SWAP%(CVT$%(RIGHT(A$, T3% - 1% + (T9% * 2%))))
                FOR T9% = 1% TO 8%
!       MAP THE BLOCKKETTE

1170    B9% = T3%
!       SAVE BLOCKETTE NUMBER

1180    FNEND
```

FIGURE 2

```
700     X% = FNB%(W.1%)
!       GET FIRST NAME BLOCK

710     GOTO 730
                UNLESS  L%
!       SKIP THE FILE PRINT OUT UNLESS WANTED

        ...

730     GOTO 740
                IF      W.2% <> F1%
\       GOTO 740
                IF      W.3% <> F2%
\       GOTO 740
                IF      W.4% <> E%
\       GOTO 760
!       IS THIS THE FILE?

740     IF      (W.1% AND -16%) <> 0%
        THEN    GOTO 700
!       GO TRY AGAIN UNLESS THE FIRST WORD IS ZERO (AFTER THE FLAG
!       BITS HAVE BEEN TRIMMED).
```

FIGURE 3

```
140     MAP     (BUF)
                 BUF%(4095%)
!       MAP THE LARGEST DEVICE CLUSTER AS A WORD ORIENTED ARRAY.

1010    T2% = (SWAP%(I%) AND 254%) / 2%
\       T1% = T2% / 8%
\       T2% = T2% AND 7%
\       T3% = I% AND 496%
!       CLUSTER.
!       SECTOR.
!       CLUSTER. (FOR REAL.)
!       BLOCKETTE.

1060    T4% = D%(T2% + 1%) + INT(T1% / C%)
!       CLUSTER NUMBER IN BLOCK

1070    T5% = T1% - C% * INT(T1% / C%)
!       SECTOR POINTER IN BLOCK

1080    IF      T4% = S2%
        THEN    GOTO 1120
!       IS BLOCK IN CORE?

        ...

1120    IF      T5% = S3%
        THEN    GOTO 1160
!       CHECK IF SECTOR IS MAPPED

        ...

1160    TEMP0% = (T5% * 256) + ((T3% + 1%) / 2%)
\       W.1% = BUF%(TEMP0%)
\       W.2% = BUF%(TEMP0% + 1%)
\       W.3% = BUF%(TEMP0% + 2%)
\       W.4% = BUF%(TEMP0% + 3%)
\       W.5% = BUF%(TEMP0% + 4%)
\       W.6% = BUF%(TEMP0% + 5%)
\       W.7% = BUF%(TEMP0% + 6%)
\       W.8% = BUF%(TEMP0% + 7%)
!       MAP THE BLOCKKETTE

1170    B9% = T3%
!       SAVE BLOCKETTE NUMBER

1180    FNEND
```

FIGURE 4

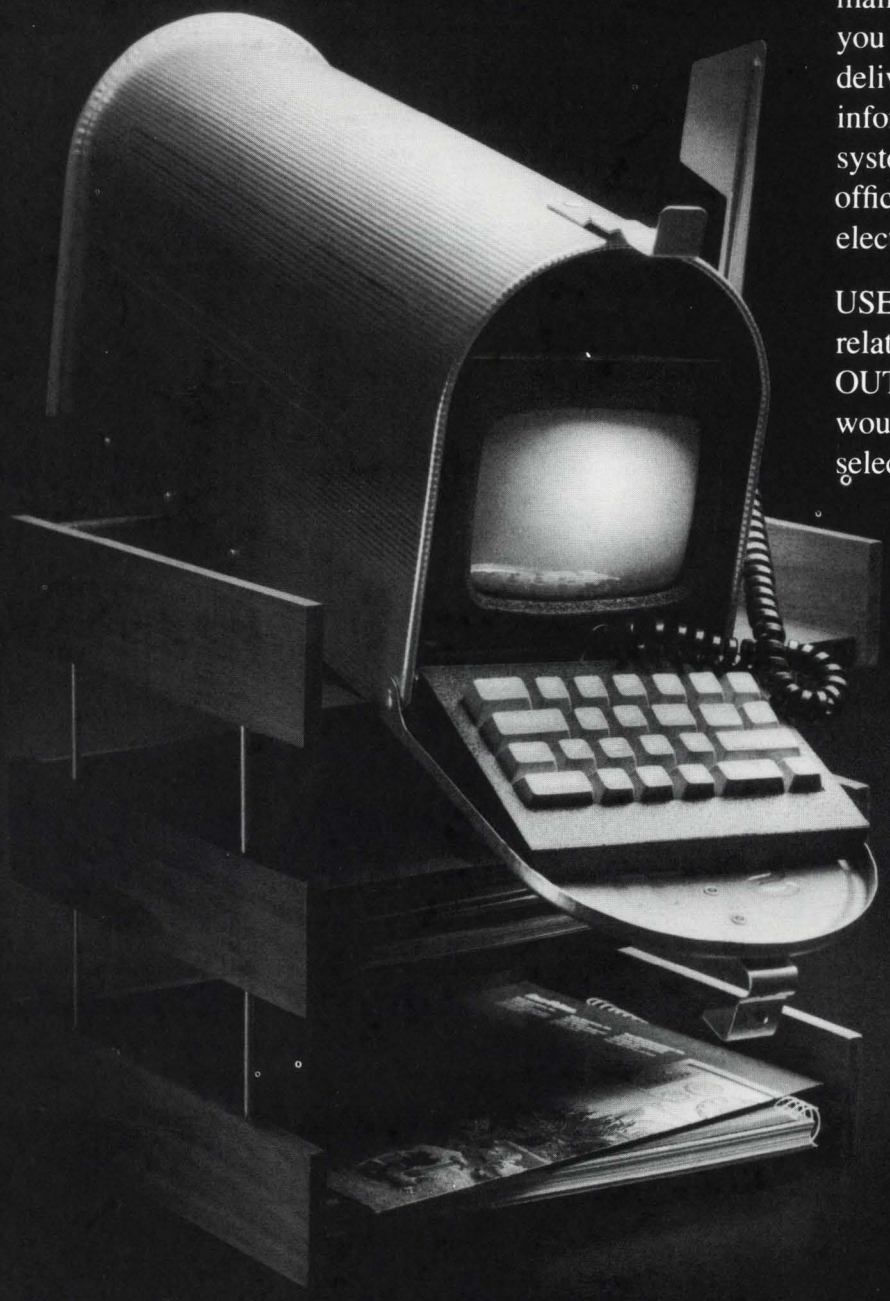
PROFIL.CMD

```
!
!       COMPILE THE BASIC-PLUS-2 SOURCE
!
BP2
OLD TRACE
COM/OBJ
!
!       ASSEMBLE THE PROFILING CODE
!
MAC BP2PRO = PRE, COMMON, BP2PRO
!
!       LINK THE BASIC-PLUS-2 OBJECT
!
TKB TRACE = TRACE, BP2PRO, LB:BP2COM/LB
!
!       RUN THE BASIC-PLUS-2 PROGRAM VIA ATPK TO GET THE PROFILE LOG
!
@ A.TMP = KB:
$DISABLE LOG
RUN TRACE
(enter a non-existant name in a large account)

^C
$ENABLE LOG
!
!       SORT THE PROFILE LOG
!
SRT B.TMP/FO:S = A.TMP/FO:S:80/KE:CN1.11
!
!       COUNT THE IDENTICAL LINES
!
@ C.TMP = KB:
$DISABLE LOG
UNIQ -C B.TMP
^C
$ENABLE LOG
!
!       SORT THE LINE COUNT LOG
!
SRT PROFIL.SRT/FO:S = C.TMP/FO:S:80/KE:CN1.16
!
!       CLEAN UP AFTER OURSELVES
!
UNSAVE A.TMP
UNSAVE B.TMP
UNSAVE C.TMP
!
!       QUE THE SORTED, COUNTED PROFILE FILE
!
QUE PROFIL.SRT
```

BP2PRO.MAC

```
        .Enabl  LC

Title   BP2PRO,<BASIC-PLUS-2 PROFILER>,01,07-AUG-82,<SLE>

;       Package:        In-House

;       Written by:     Steven L. Edwards

;       Date:           07-AUG-82

;       Copyright (c) 1982
;       Software Techniques
;       Los Alamitos, California 90720

;       This  software  is  un-released  and  Software  Techniques has no
;       commitment  to  support  it at this time, unless stated elsewhere
;       in writing.

;
;       Modification History
;
;       Ver/Edit        Date            Reason
;       --------        ----            ------

;       V7.0-01         07-AUG-82       Initial conception.
;                                       COPIED FROM DEB.
;                                       (SLE)

;
;       General Description
;

;               This  routine  replaces  the  Basic-Plus  2 LIN$ routine.
;       This  routine  allows you to do generate some profiling info.

;       Assembly instructions:

;               MAC BP2PRO = PRE, COMMON, BP2PRO
;                              ^      ^
;                              !      !
;                              !      ------- is on the SYSGEN tapes
;                              -------------- is on the BP2 tapes

;       Linking instructions:

;               Edit the ODL, CMD, or TKB command line to include a
;       reference to the object module BP2PRO.  Note: do not link to BASICS
;       because  references  to  resident  libraries  are resolved before
;       object modules.  Ignore the TKB error message:
;
;               %TKB -- *DIAG*-MODULE $ERROR MULTIPLY DEFINES SYMBOL LIN$
;
;               Since  our  module  is referenced first, the task-builder
;       will ignore the Plus-2 routine.

;
;       Global Symbols
;

        .Globl  Lin$                    ; Entry points.
        .Globl  $Cbdsg                  ; Number conversion routine.
        .Globl  $Cdtb                   ; Number conversion routine.
        .Globl  $Otsv                   ; OTS pointer.

;
;       Variable Description and Initialization
;

        .PSECT  RWDATA, RW, D, LCL, REL, CON

PROTXT: .ASCII  <15><12>/??????00000/<15><12>
PROLEN  =       .-PROTXT

        .Even;  in the quietest moments...

;
;       Mainline Program
;

        .PSECT  ROCODE, RO, I, LCL, REL, CON

LIN$:
        PUSH    R0                      ; Save R0.
        PUSH    R1                      ; Save R1.
        PUSH    R2                      ; Save R2.

        GETOTS  R0                      ; Get OTS pointer.
        ADD     #NMPTR,R0               ; Point to module name pointer.
        MOV     (R0),R2                 ; Point to module name.
        MOV     #PROTXT+2,R0            ; Point to name area.
        MOV     (R2)+,(R0)+             ; "NO
        MOV     (R2)+,(R0)+             ;   NA
        MOV     (R2)+,(R0)+             ;    ME"

        MOV     R4,@$OTSV               ; Store the pointer to the line
        MOV     (R4)+,R1                ; Save the current line number.
        MOV     SP,R2                   ; No zero suppression.
        CALLX   $CBDSG                  ; Convert Binary to Decimal, SiGned.

        MOV     #XRB,R0                 ; Address of Xrb.
        MOV     #PROLEN,(R0)            ; Length of text.
        MOV     (R0)+,(R0)+             ;    "     "   "
        MOV     #PROTXT,(R0)+           ; Address of text.
        CLR     (R0)+
        CLR     (R0)+
        CLR     (R0)+
        CLR     (R0)+
        .WRITE                          ; Print the text.

        POP     R2                      ; Restore R2.
        POP     R1                      ; Restore R1.
        POP     R0                      ; Restore R0.

        JMP     @(R4)+                  ; Exit. (Back to Plus 2)

        .END
```

ELECTRONIC MAIL. PRACTICALLY SPEAKING.



Sooner or later you will be using electronic mail. It just makes good sense. When you do, you will want a system that is complete—a delivery system, a scheduling system, and an information manager. Your electronic mail system will become an essential part of your office environment. USER-MAIL is such an electronic mail system*

USER-MAIL's power is easy to control. It relates to the way you work. Electronic IN, OUT, and HOLD baskets are just what you would expect. You can scan your IN basket, selecting only those message subjects you wish to read. Or, you can place a message into your HOLD basket for a number of days to have it automatically reappear in your IN basket on the appointed day. You can even have USER-MAIL recall specific messages by providing your own selection criteria. Replying, forwarding, and sending to groups are as easy as can be. And these are just a few of the features in store for you.

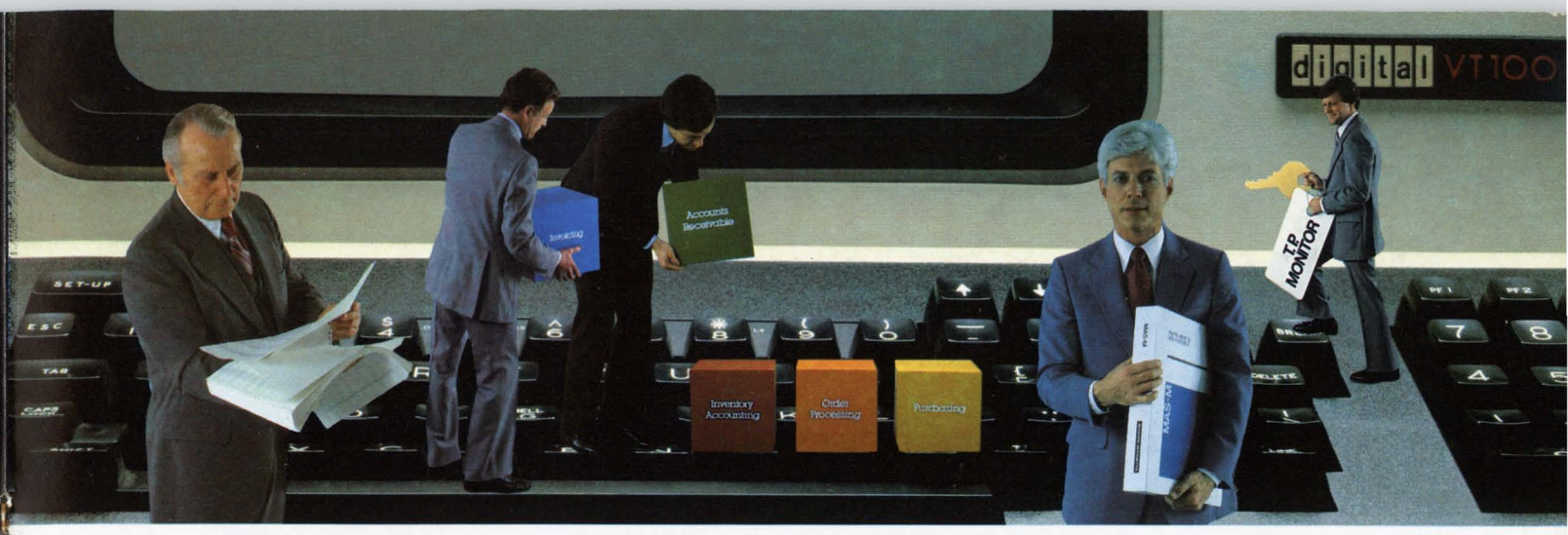
You owe yourself a closer look. Write for a brochure or give us a call direct.

NCCS USER 0110

North County Computer Services, Inc.
2235 Meyers Ave.,
Escondido, California 92025
(714) 745-6006, Telex: 182773

*USER-MAIL is currently available on DEC computers using the RSTS operating system.
RSTS is a registered trademark of Digital Equipment Corporation.
USER-MAIL is a trademark of Logic eXtension Resources.

CIRCLE 76 ON READER CARD



You can get more from your RSTS system with MAS-M.

MAS-M is the application software system from Martin Marietta Data Systems that can help you do more with your DEC hardware. That's because MAS-M is the on-line software system that gives you much more than you'd expect from packaged software.

More Flexibility.

MAS-M's modular design lets you choose from 10 different application systems:

- Accounts Receivable
- Accounts Payable
- General Ledger
- Order Processing
- Invoicing
- Inventory Control
- Inventory Accounting
- Bill of Materials
- Material Requirements Planning
- Purchasing

You can implement just the modules you need to satisfy your demands. And no matter which combination you choose, the MAS-M system is always fully integrated.

MAS-M's flexible design also makes it easy to install, and simple for your users to operate. And, since MAS-M is written in BASIC-PLUS-2, and based on the RMS-11 data management system, the software is fully compatible with your current RSTS/E operating system and DEC software.

More Control.

You can count on MAS-M for more comprehensive data accuracy and security, too.

MAS-M's powerful transaction processing MONITOR gives you maximum control over your data—from start to finish. User passwords and menu selections are checked against user security profiles. Data entry validation is also standardized in the MAS-M MONITOR, so any invalid data can be corrected *before* it reaches your application program.

More Productivity.

MONITOR is also an important tool in developing new applications. You can use MONITOR to create input screens and validation rules on-line. And, MONITOR can help you improve programmer productivity by providing a standard framework for input of code that minimizes the difficulties of user interface and terminal characteristics.

More Support.

You can count on Martin Marietta Data Systems for system development and implementation, comprehensive training, and clear, concise documentation. We can also provide an extensive Maintenance Service to support your MAS-M system.

What it all adds up to is a packaged software system that can give you everything you need to get your jobs done. And more. Write or phone us today, and we'll tell you more about how the MAS-M software system can work for you.

MAS-M

The Software System That Can Help You Do More.

MARTIN MARIETTA DATA SYSTEMS

Martin Marietta Data Systems
Marketing Services, R/H
6303 Ivy Lane, Greenbelt, MD 20770
(301) 982-6500

MARTIN MARIETTA

CIRCLE 111 ON READER CARD

DEXPO West 82.
The Only Show As Important As Your DEC* Computer.

DEXPO West 82.
100% "Compatible" With Your DEC System.

There's a world of compatible products that can make your DEC computer do *more*. Do it faster. Do it cheaper. Or just do it *better*. And you'll discover that world at DEXPO West 82. A world of 250 vendors demonstrating all the latest DEC-compatibles. More DEC-compatibles than at any other computer show in the world.

*DEC and DECUS are registered trademarks of Digital Equipment Corporation.

DEXPO West 82.
The Only Show You Really Need.

DEXPO West 82 makes it easy to find the hardware, software, services and supplies you're looking for. Thousands of alternatives, enhancements and special solutions will be on exhibit—*many for the first time anywhere*. You'll even find a wide range of compatibles for DEC's new personal computer and CAD/CAM introductions. Everyone who owns, manages or uses a DEC computer belongs at DEXPO West 82...top management, financial management, DP management and senior staff.

DEXPO West 82.
Free to All DECUS* Conference Registrants.

DEXPO West 82 is only five minutes from the Anaheim DECUS Conference. It's easy to make the Show part of your schedule. Free shuttle busses will take you from the conference to the Show and back again. And your DECUS conference badge entitles you to enter the Show without paying a registration fee.

Group Registration Discounts Available
Bring Your Boss for FREE!
Mail Coupon for Complete Details

DEXPO™ West 82

The Second National DEC-Compatible Industry Exposition

Anaheim Convention Center
Anaheim, California
December 7-9, 1982

SEND FOR COMPLETE DETAILS ON THE WORLD'S MOST COMPLETE DEC-COMPATIBLE SHOW

☐ Tell me how I can save time and money on DEXPO West 82 registration, air fare, hotel accommodations and car rentals.

☐ Send _____ extra copies for my associates.
(amount)

NAME _____

TITLE _____

COMPANY _____

ADDRESS _____

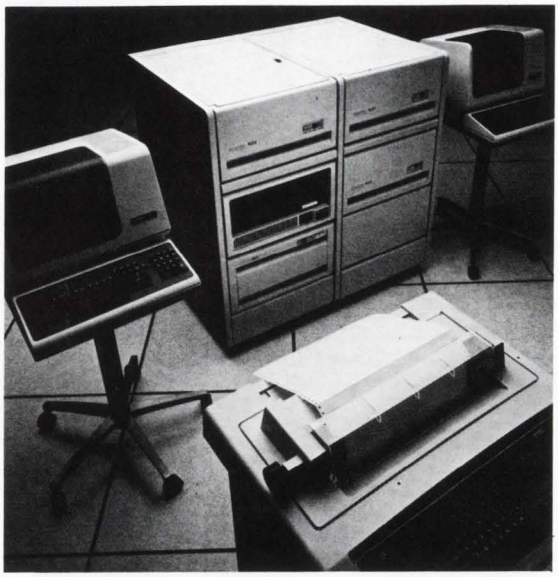
CITY _____ STATE _____ ZIP _____

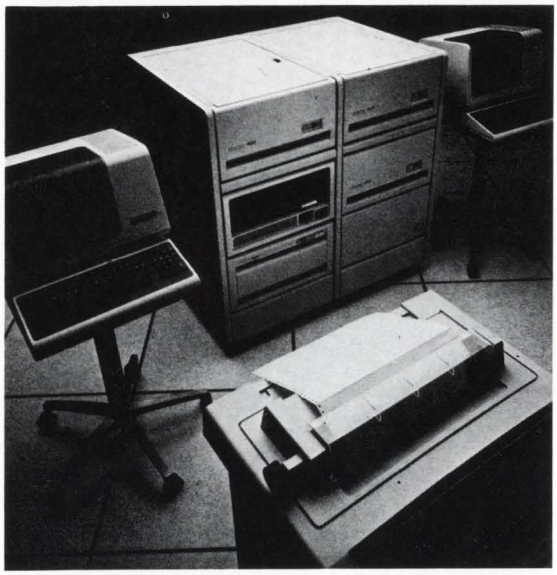
Mail to: Expoconsul International, Inc.
19 Yeger Road, Cranbury, NJ 08512
Tel: 609-799-1661

The VAX-SCENE

Number 10 (RSTS PROFESSIONAL, Vol. 4, No. 5) October 1982

INSIDE:





VAX PERFORMANCE NOTES

By Carl B. Marbach

The current offering of VAX computers consists of the dual processor 11/782, the single processor 11/780, mid-range 11/750 and the smallest VAX the 11/730. Knowing which one of these is right for your application is a hard job.

There is one major premise which all performance data must be measured against:

All application systems are different.

Thus, it is impossible to accurately predict how your system will run on any given processor, with any given options. Before you despair, it is possible to find out the general performance of these machines and make an educated guess about whether your particular application will run. The RSTS newsletter recently carried an article on performance that made some DEC people mad, mostly because it was possible for the reader to become confused about what the data really said. The best way to get performance data is to call your local DEC salesperson and ask him to come out and discuss performance with you, tell him to bring his 'sales guide' and to make sure it has the chapter that includes performance. What I am trying to tell you here is what kind of information you can get, and how to get it; not what the information is.

Inside DEC there are groups designated to do performance measurement and reporting. They have very sophisticated tools (computers and programs) to help them do this measurement. They also have some of the expertise necessary to do this job. These engineers load up various machines with several different 'workloads' to simulate different user application environments. Some of these workloads are an insurance company simulation, a specialized application package (COBOL programs), college registration system simulation and an order entry system.

Each of these simulations is run and data gathered on certain 'metrics'. The metrics measured include average response time, transaction throughput, transactions per hour, mean service time (time needed to complete on whole transaction, user productivity and computation time). This is not meant to be the whole list, but is representative of the type of measurement that is made.

Results of these tests are informative ONLY if the reader takes the time to reflect on the reasons for the differences between two configurations. The number of batch jobs per hour run with 12 interactive terminals by an

11/730 is not improved by increasing from 2MB to 3MB. Why? An 11/750 can support about 50% more interactive terminals while running the same number of batch jobs per hour with a similar increase from 2MB to 3MB of main memory. Why? The 11/730 has run out of CPU power at that level of batch jobs and interactive terminals, adding memory won't stretch the CPU, but the 11/750 has more CPU power to go and adding the memory frees the CPU power to be used by the application instead of managing the (too small) 2MB. What this tells you is not that the 11/730 is a 12 terminal machine and the 11/750 is a 24 terminal machine, but rather that the 11/750 can benefit from more memory and the 11/730 can't. It was interesting to note that on one of these simulations the 11/730 peaked (2 second response) at 20 terminals and 2 MB, the 11/750 at 44 terminals and 6MB, and the 11/780 at 56 terminals and 6MB. Just to show you how foolish a comparison like this can be, during a different simulation an 11/780 NEVER had less than a 2 second response time even with 4MB and 8 terminals! We know that the task must have been more complicated in this case, and that the 2 second response time stayed the same until there were more than 32 active users.

Another interesting parameter to watch in these 'metrics' is when the flat line of a graph begins its sharp upward swing; that is, when response time goes way up as you add the 35th terminal. One of the very nice things about RSTS is the way it seems to degrade slowly rather than 'falling off the end of the table' when it reaches some limit. Up to a point this is true, but beyond that limit — bad news. Small buffers used to be a problem where large systems would run very slowly when the small buffer count dropped below 50-60; memory can be a problem when swapping starts eating up CPU and disk transfer capacity and this hurts more on the slower CPUs. I have seen systems that overload their disk transfer capability and go slowly when several disk intensive jobs are running; even though lots of CPU is still available. Even when systems slow down, or response time increases this can still mean that tuning (software) or some hardware enhancement can significantly increase the machine's capacity. With all this in mind, for one workload the 11/730 started to slow down sharply at 16 users, the 11/750 at 28 users and the 11/780 at 36 users. For comparison between machines these results may be valid, but an 11/780 is more than a 36 user machine! In fact, for comparison the 11/780 is a little more than twice an 11/730 and about 133% of an 11/750 under these conditions. In yet another simulation the 11/750 (6MB) began slowing rapidly at 56 users and the 11/780 (6MB) at almost 80 users, giving the 11/780 about 140% of the 11/750's performance. Getting a flavor for relative power? More can be gotten directly from your DEC salesperson!

There is much more to be said about performance, and lots more VAX information. Next we will talk about single user performance, standard benchmarks like 'whetstones', single instruction timings and some comparisons with the 11/44 and others.

. . . continued on page 84

$FAO

The Formatted ASCII Output Directive (or, Print-Using in Macro . . .)

By Bob "MACRO MAN" Meyer

The $FAO monitor call (excuse me, I mean 'System Service'; Oh, how I wish they'd stop changing terms. . .) turns out to be quite a handy tool. Basically, the call provides a means of getting numeric data to the outside world (which under RSTS requires various numeric conversion subroutines), formats the data just about any way you like, allows you to intermix text with the numbers (i.e. 'Balance: $1950.50'), AND does output in octal,decimal or hex!

The heart of the $FAO directive is the 'control string' parameter. This .ASCID string (.ASCID is like .ASCII, but the assembler creates a STRING DESCRIPTOR before the string for you) is interperted by VMS when the directive is executed and gives you strict control on how the output will look.

If you'll turn to the sample program, in the TEXT psect, locate the label CTRL:; this is the control string for our demo. What your looking at is as follows: The '!13<' specifies a field with a length of 13; this field is terminated a few characters later with the symbols '!>'. The string 'Decimal:' will be output first; the '!UW' indicates that an UNSIGNED WORD will go here; this word will be parameter #1 in the FAO call. The number will be padded to the left with spaces so that the entire string (including 'Decimal:') will have a total length of 13. (Note that the !UW and other such commands must be upper case.)

Moving down the control string, you'll find the command '!OW' (OCTAL WORD), which will output the second parameter (P2) in octal, and '!XW' (HEX WORD) will display the parameter P3 in hex. The '!/' adds a CRLF to the output.

Some other FAO commands are:

!%D	insert today's date in the output string
!%T	insert the current time
!%S	insert the letter 'S' if the next parameter is > 1
!ZW	like !UW (unsigned word), but zero fill
!AS	insert an .ASCID string into the output string

Lengths for items can be specified by inserting a length count between the '!' and the command:

!5ZW	insert a decimal number, and zero fill to 5 places

Repeat counts can also be used:

!3(6ZW)	insert 3 decimal numbers, each 6 characters long

The special symbol '#' tells FAO to use the next parameter on the list for the next required value; for example, the command:

!#(4UW)

interprets the first parameter to be a count of the number of items to follow; each item will be taken as an unsigned word with a length of four.

To use the $FAO call, you must provide the following info: (see label 10$: in the demo program) a control string (as described above); a buffer where the system can return the formatted string ('FAODESC' points to the descriptor which points to the actual FAO buffer); a word in which the system can put the length of the formatted string (FAOLEN); and finally, a number or list of numbers that you'd like formatted in the output string. (A second call is also provided, $FAOL, which acts like $FAO but accepts the address of a LIST of arguments as the P1 parameter, for applications where several pieces of data are to be output.)

Once the $FAO call has been executed, we actually print the contents of the FAO buffer on the terminal. From there I just bump the counter NUM, and loop till I'm convinced that it works.

This demo program should work if keyed in as is. To assemble & link:

```
$ MAC DEMO
$ LIN DEMO
$ R DEMO
```

The results should look something like:

```
Decimal:  0     Octal:  000000     Hex:  0000
Decimal:  1     Octal:  000001     Hex:  0001
Decimal:  2     Octal:  000002     Hex:  0002
Decimal:  3     Octal:  000003     Hex:  0003
Decimal:  4     Octal:  000004     Hex:  0004
                    .
                    .
                    .
Decimal:  10    Octal:  000012     Hex:  000A
Decimal:  11    Octal:  000013     Hex:  000B
Decimal:  12    Octal:  000014     Hex:  000C
```

(except for the 3 dots. . .)

Have fun!

```
.title  fao     formatted ascii output example
.ident  /demo/

.psect  impure
num:    .word   0                       ;number to f
faodesc:                                ;the fao des
        .long   80                      ;length of fao buffer
        .long   faobuf                  ;address of the fao buffer
faobuf: .blkb   80.                     ;the actual buffer
faolen  .word   0                       ;holds length of string returned by fao
ttdesc: .long   80.                     ;terminal descriptor
        .long   ttbuf                   ;address of tt buffer
ttbuf:  .blkb   80.                     ;actual tt buffer
ttlen   =.-ttbuf                        ;length of tt buffer
tt:     .ascid  /TT/                    ;name of this terminal for $assign
ttchan: .word   0                       ;holds channel returned by $assign

.psect  text
ctrl:   .ascid  '!13<Decimal:  !UW !> Octal: !OW Hex: !XW !/'
                                        ;the fao control string

.psect  code
fao:    .word   0                       ;register save mask
        $assign_s-                      ;assign the tt:
                devnam=tt,-
                chan=ttchan
        blbs    r0,10$                  ;continue if assign worked
        ret                             ;else return to VMS with error in R0

10$:    $fao_s-                         ;format an output string
                ctrstr=ctrl,-           ;address of fao control string
                outlen=faolen,-         ;where to put length of formatted string
                outbuf=faodesc,-        ;address of fao descriptor
                p1=num,p2=num,p3=num    ;parameters to be output

        $qiow_s-                        ;print the contents of the fao
                chan=ttchan,-           ;buffer on the tt:
                func=#io$_writevblk,-
                p1=faobuf,p2=faolen

        incw    num                     ;bump the counter
        cmpw    num,#25.                ;have we seen enough?
        blss    10$                     ;not yet; loop

        ret                             ;yes; just exit to VMS

        .end    fao
```

THE RSTS CRYSTAL BALL — Part 3

. . . continued from page 12

is shown in figure 3. A sample BASIC+ program which prints the load averages is listed at the end of the article, after LOADAV.MAC.

If you encounter any problems with the installation or the code, please contact me. If you are not in the mood to key all of the files, send a check for $20.00 to IISI (attn:MCG) and we'll send you a tape with all of these goodies, plus the ONLRES source from last month. Please specify 800 or 1600bpi.

RSTS V7.2 NOTES

Unfortunately, as of this writing, we have not received our V7.2 distribution, so I don't have much information on internal changes. I do know that DEC didn't keep its promise about the new spooling package. Apparently, V7.2 will still run the old, crufty V6A spooling package that we all know and love(?).

I do know of two internal changes in V7.2. The first is a fix for the UU.TRM (set terminal characteristics) call. There are no problems with setting terminal speeds in V7.2.

Secondly, I heard that INIT.SYS was changed to automatically shuffle the memory allocation table, if any entry has changed in size. What this means is if you use ONLRES to change your monitor size, or change other memory allocation values in the monitor SIL, INIT will fix the memory allocation table for you when you reboot. If your monitor got smaller, INIT will push XBUF and the default RTS down in memory so there is no wasted space. Alter-

```
*AG/,MHB/-13C/13-JUL-82,MHB/G/SJM/I#/MCG#<cr>
*AH/<tab>DEFORG<tab>QSTCTL<tab><tab>;QUEUE STATISTICS/V<cr>
<tab>DEFORG<tab>QSTCTL<tab><tab>;QUEUE STATISTICS TABLE<cr>
*AV<cr>
<cr>
*AV<cr>
; ***** TBL IS THE START OF READ-ONLY MEMORY *****<cr>
*I<cr>
<tab>DEFORG<tab>LODCTL<cr>
<cr>
LOADAV::<tab><tab><tab>; Load average buckets, pointed to by LOADPT<cr>
<tab><tab><tab><tab>;  at TTYHCT - 2<cr>
LOAD1M::.BLKW0<tab><tab><tab>; One minute load average<cr>
LOAD5M::.BLKW0<tab><tab><tab>; Five minute load average<cr>
LOADFM::.BLKW0<tab><tab><tab>; Fifteen minute load average<cr>
LOD1ML::.BLKW0<tab>2<tab><tab>; Internal 32 bit one minute load
 average<cr>
LOD5ML::.BLKW0<tab>2<tab><tab>;   "     "   "  five  "    "    "<cr>
LODFML::.BLKW0<tab>2<tab><tab>;   "     "   "  fifteen "   "    "<cr>
<cr>
<esc>*V<cr>
; ***** TBL IS THE START OF READ-ONLY MEMORY *****<cr>
*H/NSPTMO/V<cr>
; NSPTMO - NSP TIMEOUTS.<cr>
*AV<cr>
;<cr>
*AV<cr>
; THIS SUBROUTINE IS CALLED ONCE A SECOND, EVERY SECOND...<cr>
*AV<cr>
;-<cr>
*AV<cr>
<cr>
*AV<cr>
<tab>DEFORG<tab>NSPTMO<cr>
*I<cr>
<tab>DEFORG<tab>LOADAV<cr>
NSPTMO::CALL<tab>NTMO<tab><tab>; Call the real NSP timeout checker<cr>
<tab>SPLC<tab>3<tab><tab>; Back to level 3 just in case<cr>
<tab>JMPX<tab>COMPLD<tab><tab>; Now off to compute the load average<cr>
<cr>
<esc>*V<cr>
<tab>DEFORG<tab>NSPTMO<cr>
*G/DEF/-3DV<cr>
<tab>ORG<tab>NSPTMO<cr>
*AV<cr>
<cr>
*AV<cr>
NSPTMO:<tab>RETURN<tab><tab><tab>;NOTHING UNLESS OVERLAID<cr>
*G/SP/-2DV<cr>
NTMO:<tab>RETURN<tab><tab><tab>;NOTHING UNLESS OVERLAID<cr>
*EX<cr>
```

FIGURE 1. CPATCH Command File for TBL.MAC

```
*AG/,MHB/-13C/13-JUL-82,MHB/G/ABC/I#/MCG#<cr>
*H/TTYHCT::.WORD<tab>0<tab><tab>;# OF UNHANGS PERFORMED/V<cr>
TTYHCT::.WORD<tab>0<tab><tab>;# OF UNHANGS PERFORMED<cr>
*0AI<cr>
LOADPT::.WORD<tab>LOADAV<tab><tab>; Pointer to load average buckets<cr>
GLOBAL<tab><LOADAV><cr>
<cr>
<esc>*0AV<cr>
TTYHCT::.WORD<tab>0<tab><tab>;# OF UNHANGS PERFORMED<cr>
*EX<cr>
```

FIGURE 2. CPATCH Command File for TTDINT.MAC

```
*G/TBL,TBL/V<cr>
TBL,TBL/C=IN:COMMON,KERNEL,DK:CONFIG,IN:CHECK,TBL<cr>
*AI<cr>
$R MACRO.SAV<cr>
LOADAV,LOADAV/C=IN:COMMON,DK:LOADAV<cr>
<esc>*G/TTDVR/V<cr>
TTDINT/C<cr>
*AI<cr>
LOADAV/C<cr>
<esc>*EX<cr>
*G/TBL,TBL/V<cr>
TBL,TBL/C=IN:COMMON,KERNEL,DK:CONFIG,IN:CHECK,TBL<cr>
*AI<cr>
$R MACRO.SAV<cr>
LOADAV,LOADAV/C=IN:COMMON,DK:LOADAV<cr>
<esc>*G/TTDINT/V<cr>
TTDINT/C<cr>
*AI<cr>
LOADAV/C<cr>
<esc>*EX<cr>
```

FIGURE 3. CPATCH Command File for SYSGEN.CTL

nately, if the monitor grew in size, it will push up the other entries to make room.

Finally, a note about named directories. I don't know if named directory support has been removed from V7.2 (I'd have to guess it hasn't). Although the code more or less works, there are several dangerous bugs in it. Occasionally, the code will blast the first block of disks with named directories (i.e. the root of the MFD). This means a potentially nasty cleanup job. I don't have a fix for this bug, nor have I even had the time to look and see what causes it. I would advise against using named directories without such a fix.

CONCLUSION

Several people have requested an ONLSET program; i.e., something that will perform some or all of the functions of INIT's SET command on-line. Also, I have had some time to play with FMS, and have managed to design forms using VT100 graphics, double height characters, and other goodies that DEC insists FMS won't deal with. If there is any user interest in either of these topics, please let me know. Perhaps I will make them the subject of future articles, DECUS sessions, etc.

I hope you have enjoyed this installment of the RSTS Crystal Ball. I will continue to try to present information which is interesting and useful. In the future, I may be able to answer any technical or semi-technical questions that you send to me. In any case, I would enjoy hearing from you, so if you have any questions, gripes, or suggestions, call or write me at the address below. Until next time, hack hack!

Michael C. Greenspon, C/O Integral Information Systems, 9832 Vicar Street, Suite 100, Los Angeles, California 90034, (213) 558-0732

The JDB's connected to the IOB. The IOB's connected to the WCB . . . .
See the RSTS Internals Manual.

```
;?(.ENABL       LC
TITLE   LOADAV,<System load average computation>,11,13-Jul-82,MCG


;
;       Copyright (C) 1982 by
;       Integral Information Systems, Inc., Los Angeles, California
;
;       Title to and ownership of this software shall at all times re-
;       main with Integral Information Systems.
;
;       The information in this software is subject to change  without
;       notice and should not be construed as a commitment by Integral
;       Information Systems.
;
;       This software is unreleased and Integral  Information  System.ENABL     LC
TITLE   LOADAV,<System load average computation>,11,13-Jul-82,MCG


;
;       Copyright (C) 1982 by
;       Integral Information Systems, Inc., Los Angeles, California
;
;       Title to and ownership of this software shall at all times re-
;       main with Integral Information Systems.
;
;       The information in this software is subject to change  without
;       notice and should not be construed as a commitment by Integral
;       Information Systems.
;
;       This software is unreleased and Integral  Information  Systems
;       has  no  obligation  to support it at this time, unless stated
;       elsewhere in writing.
;


;+
;
; COMPLD - Compute the load average
;
; Call: Called once a second through NSPTMO hook
;
;-

.ENABL  LSB

        DEFORG  COMPLD

        CLR     R2              ; Initially no running jobs
        CLR     R1              ; Start with job "0"
10$:    TST     (R1)+           ; Go to next job * 2
        CMP     R1,#JOBMX2      ; Done all we need to do?
        BHI     30$             ; Yes, go do computations
        MOV     JOBTBL(R1),R0   ; Get job data block pointer
        BEQ     10$             ; No job, try next one
        MOVB    JDPRI(R0),R0    ; Get the job's priority
        CMP     R0,#-121.       ; Is the job suspended?
        BLE     10$             ; Yes, go to next one
        MOV     JBWAIT(R1),R0   ; Get the job's waits
        BIT     R0,JBSTAT(R1)   ; Is the job runnable?
        BNE     20$             ; Yes, bump counter
        BIT     #JS.SY,R0       ; Waiting on disks, etc?
..LDWA  ==      .-2     ;**PATCH** Waits to consider runnable

        BEQ     10$             ; No, try next job
20$:    INC     R2              ; Count one more job runnable
        BR      10$             ; And loop

30$:    MUL     #10000.,R2      ; Make it jobs * 10000.
        CALL    100$,R5,<LOD1ML,60.,LOAD1M> ; Compute load averages
        CALL    100$,R5,<LOD5ML,60.*5.,LOAD5M>
        CALL    100$,R5,<LODFML,60.*15.,LOADFM>
        RETURN                  ; That's all

GLOBAL  <JOBMX2,JOBTBL,JBWAIT,JBSTAT,LOAD1M,LOAD5M,LOADFM>
GLOBAL  <LOD1ML,LOD5ML,LODFML>


; Do the load average computation:
;
;       R5 -> Old load average bucket, number of samples in average,
;               normallized load average bucket (16 bits)
;    R2/R3 =  Current number of jobs in a run state * 10000.
;

100$:   MOV     (R5)+,R4        ; Pick up the address of the load average
        MOV     (R4),R1         ; Get the average
        MOV     2(R4),R0        ;  all 32 bits
        DIV     (R5),R0         ; Get 1/nth of load average
        SUB     R0,(R4)         ; Subtract this nth from the average
        SBC     2(R4)           ;  handle carry if any
        CLR     R0              ; Clear for divide of remainder
        DIV     #100.,R0        ; A neat hack to get averages down to zero
        SUB     R0,(R4)         ; Subtract out this smaaaaall number
        SBC     2(R4)           ;  handle carry

        MOV     R2,R0           ; Copy so as not to clobber these
        MOV     R3,R1
        DIV     (R5)+,R0        ; Get 1/nth of jobs in a run state * 10000.
        ADD     R0,(R4)         ; Add it into the load average, forming the new
        ADC     2(R4)           ;  load average * 10000.  Handle carry if any
        MOV     (R4),R1         ; Pick up the load average now
        MOV     2(R4),R0        ; All 32 bits of it
        DIV     #100.,R0        ; Make it into load average * 100. (16 bits,
        MOV     R0,@(R5)+       ;  no round-off error) and save it, fix return
                                ;  address (isn't the PDP-11 instruction set
                                ;  great for a 16 bit processor?)
        RETURN  R5              ; That's it, go home

.DSABL  LSB
.END
10      ! Simple BASIC+ program to display the system load averages. &
        ! See text of article for more information on implementation, etc. &
 &
        DIM M%(30%) &
\       CHANGE SYS(CHR$(6%) + CHR$(-12%)) TO M% ! Get monitor tables part II &
\       LOADPT% = M%(11%) + SWAP%(M%(12%)) - 2% ! Get TTYHCT address - 2 as &
                                                !  pointer to load averages &
\       LOADAV% = PEEK(LOADPT%)                 ! Now get address of averages &
\       L1MIN = FNREAL(PEEK(LOADAV%))           ! Get one minute average &
\       L5MIN = FNREAL(PEEK(LOADAV% + 2%))      ! Get five minute average &
\       L15MIN = FNREAL(PEEK(LOADAV% + 4%))     ! Get fifteen minute average &
\       PRINT USING "Load averages: 1 min: ##.##, 5 min: ##.##, 15 min: ##.##", &
                L1MIN/100., L5MIN/100., L15MIN/100. &
                                                ! Now print out the averages &
\       S$ = SYS(CHR$(9%))                      !  and get out &
 &
\       DEF* FNREAL(INTEGER%)                   ! Convert integer to real &
\       FOO = INTEGER% &
\       FOO = FOO + 65536. IF FOO < 0.          ! Fix BASIC's stupidity about &
\       FNREAL = FOO                            !  unsigned integers &
\       FNEND &
\       END
```

LETTERS to the RSTS Pro . . .

. . . continued from page 6

lower levels. Hence this is not a problem for the 32 KW limit.

I must defend the 'developers' on this point, and my authority stems from the fact that I have been a developer, albeit a short time, and before that, a Region Support person with DEC here in SPR (South Pacific Region). My RSTS experience stems from the V4 days (Do you remember the 28KW RSTS system??).

Thanks again for your great efforts, and a fine mag. Look forward to your DEC Pro.

Regards from down under, Harry Starr
Gold Coast Computer Services
Surfers Paradise, Qld., Australia

—

In an attempt to use the memory exerciser submitted by R.A. Smith of Digital and published in the June/July issue*("Basic Memory Exercising Programs", RSTS Pro. June 1982, V.4, #3, p.8)*, I have found several major bugs which will prevent the program from performing its intended function.

The basic concept of this program is good and if implemented correctly, it will work. However, if this program found any defective memory locations it was due to random chance.

The memory exerciser program is intended to write to, and read from, the resident library. Thus by moving the library around physical memory the program will be able to access specific physical memory locations, and pin-point the actual address at which there is a memory problem.

It is unfortunate that the program as was published did not perform the desired function, even though it would give all appearances of doing so. Shown below are the reasons that this program will not work properly. I have also outlined the changes necessary to have the program access specific physical memory locations.

PROBLEM: The memory exerciser never accesses the resident library MEMCOM. As a result the memory in the common block MEMDAT is never mapped to MEMCOM and the memory that is exercised is wherever the program happens to be loaded in memory.

SOLUTION: Force the program to access the resident library. To access a resident library the program must be attached to the library, create a window into the library, and map that window. The means for accessing a resident library can be done in either of two ways: 1. Use the task builder resident library directives RESLIB or RESCOM. When using these directives the task builder includes in the task image all of the necessary code to access the resident library. All of this is totally transparent to the programmer. For the memory exerciser program this directive would be:

RESCOM = SY : [1,800MEMCOM/RW

This directive "tells" the task builder to set things up so that the program will automatically attach to the resident library MEMCOM in account [1,80], and allow read/write access to the library. The directive also causes the task builder to map the common area MEMDAT to the resident library. See the *Task Builder User's Guide* for more information on accessing resident libraries thru the task builder directives.

2. Use MACRO-11 subroutines to issue the .PLAS directives to attach to a resident library, create a window, map that window and detach from the resident library. This method requires MACRO-11 subroutines because the .PLAS directives are not available from Basic-Plus 2. See the *RSTS/E System Directives Manual* for more information on using the .PLAS directives to access resident libraries.

PROBLEMS: When a program is attached to a resident library that library cannot be unloaded and loaded at a new memory location. This problem occurs when using the task builder directive RESCOM to attach to the resident library. When the problem attempts to unload the resident library, RSTS will return error number 3, "?Account or device in use".

SOLUTION: The .PLAS monitor directives must be used to allow access to the resident library, and to allow the resident library to be unloaded and re-loaded at different memory locations. To accomplish this three .PLAS directives must be issued.

- Attach to a Resident Library
- Create and Map a Window
- Detach from the Resident Library

The first two directives must be issued AFTER the resident library is loaded at the correct address and BEFORE an attempt is made to exercise that memory. The third directive must be issued AFTER the memory has been exercised and BEFORE an attempt is made to unload the resident library.

Attach to a Resident Library — requires (1) the library name, in Rad50; and, (2) the access mode which will be read/write. This directive returns (1) an internal library ID; and, (2) the size of the library in 32-word blocks.

Create and Map a Window — requires (1) the library ID (returned from the attach); (2) the APR to be used for the window; (3) window size in 32-word blocks; (4) offset from start of library (zero in this case); (5) length of window in 32-word blocks; and, (6) accesss flag for the window which will allow write access and map the window. This directive returns (1) the window ID; (2) starting virtual address of the window; (3) the mapped length of the window; (4) and a status flag.

Detach from a Resident Library — requires the library ID (returned from the attach). This directive returns a status flag.

Along with the above mentioned .PLAS directives, the task builder must be "told" where the virtual address of the resident library will be so the common area, MEMDAT, can be located at that address. This address is dependent on the APR to be used in mapping the resident library window. There are 8 APR's available to map the program in memory, numbered 0-7. These APR's map memory in 4K-word sections. The RTS associated with the program will utilize the highest APR's, and the program will utilize the lowest APR's. Thus, the resident library will need to be mapped with an APR somewhere in the middle. The best APR to use is the first one available below the RTS. With the RSX disappearing RTS this APR would be APR7 with a virtual address of 160000(8). With a 4K RTS such as RSX or BP2COM this would be APR67 with a virtual address of 140000(8). Refer to the *PDP-11 Processor Handbook* for more information on APRs and the virtual addresses of each.

To tell the task builder where to locate the common area MEMDAT the following directive is used:

VSECT = MEMDAT:160000

With this directive, the task builder will place the PSECT MEMDAT at the virtual address of 160000(8). This virtual address will be mapped by APR7 and hence the residence library.

When using the .PLAS directives and the task builder VSECT directive as outlined above, the memory exerciser program will work correctly.

Richard W. Hill, Software Specialist
Software Techniques, Inc., Los Alamitos, CA

—

Thank you for taking the time to publish my articles in the June edition of *RSTS Professional (V.4, #3, pp. 51 and 76).*

Rather than constantly sending you enhancements and corrections to our 'JUMP' program, would it be possible for you to mention that any individual interested in the new release contact me personally. The program would be written in PIP FORMAT at 1600 BPI on a tape provided by the interested party.

Thank you for your assistance in this matter.

Patrick Holmay
Director, Computation Laboratory,
Collegeville, Minnesota

—

I really appreciated your pre-symposium seminar here a few weeks ago. You said you were looking forward to some contributions from down-under, so I present this modest little goodie from my software tool kit. Use it with care!

Ralph Zwier, Professional Software
Victoria, Australia

Thank you, Ralph!
[Readers see, PROTCT.B2S, p.20 this issue.]

—

Your correspondent SPIDL *("Dear RSTS Man" August 1982, V.4, #4, P.36)* may find some help in a letter of mine you published in September 1981, *(V.3, #3, p.36).* If he connects a signal from the lineprinter, which is set when the lineprinter is on, through to pins 5 and 8 of the DH11 then you can get the spooler to detect the lineprinter being turned off and spooling will be inhibited until the machine is turned on next day.

George May
Software Sciences Limited
Farnborough, Hampshire

—

Thanx for the excellent magazine; it proves to be an indispensable resource. I have a suggestion for software vendors. Seeing as most of the ones we deal with either subscribe to and/or contribute to the magazine, I am presenting it here:

We have purchased a number of software products advertized in the 'Pro', and implemented some of the monitor enhancements and utilities from the articles. As a result, we have encountered a problem that is more likely to occur as more non-DEC products are put into use: conflicting control-characters. I would like to suggest to those offering software which uses control characters, that provisions be made for users to select what character should be used at installation time. This should result in quite a few additional benefits to both vendors and purchasers, along the lines of reduced maintenance, returned products, additional distribution, etc.

I. cseh
Alcan Canada Products Ltd.
Kingston, Ontario Canada

Good idea. Vendors!

—

Only as I was building my house this summer did I find out what was TECO TESTED . . . A sheet of 4 x ¾" COX Plywood!

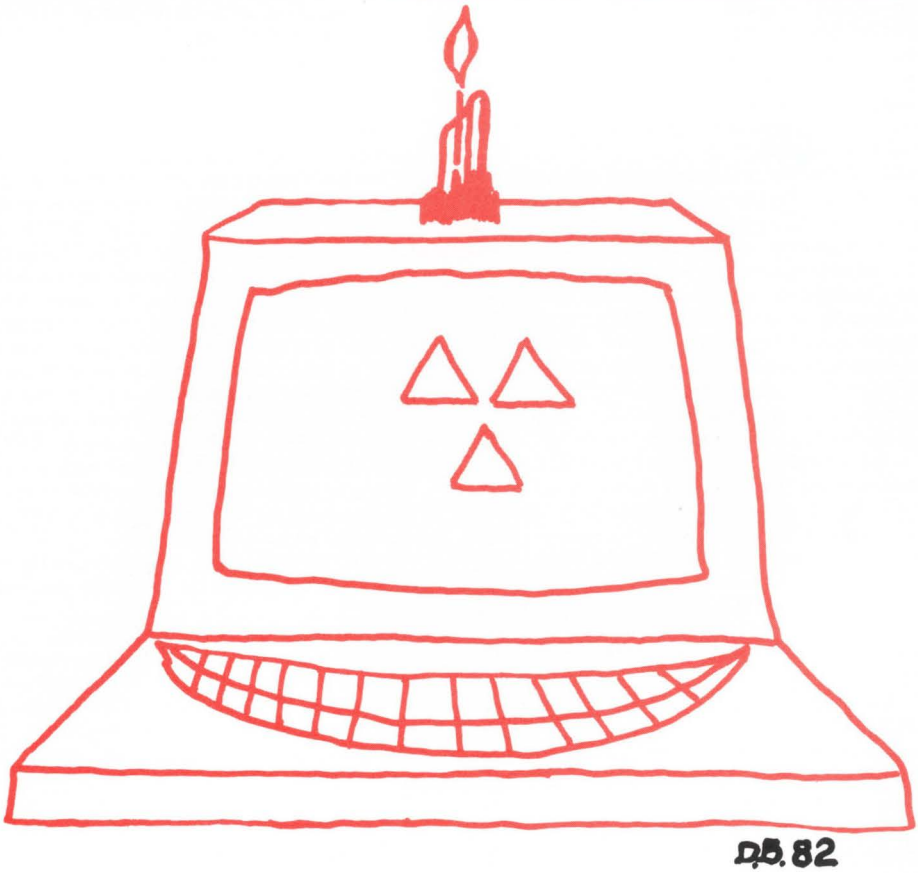
What a stumper . . . Hope I'm not too late to win the booby prize or something.

Jim Perkins, Sr. Analyst
Underhill, Vermont

Jim, the TECO test ended in our last issue (August 1982, V.4, #4). However, it's never too late for ***Honorable Mention****, and so for the record — you're right.*

LETTERS to the RSTS Pro . . .

. . . is your column! Send us your comments, suggestions, photos, or notes of interest to the RSTS community. We'd enjoy hearing from you.



SIMINI

A SIMULATION MINI COMPUTER

By John Cato

In the last issue of **RSTS Professional**, I presented the first part of the SIMINI user guide — the index, introduction and specification. In this issue I present the second part — how to use SIMINI.

Following on from the user guide, as such, is included a few example programs as demonstrated within the complementary simulation MINSIM.

The next issue will contain Part 3, the source listings of SIMINI and MINSIM.

If you would like a complete copy of the SIMINI user guide, please send a self addressed sticky label and £3.75, or $7.50US, to cover photocopying and postage to me: John Cato, R.T.Z. Computer Services Ltd., P.O. Box 19, 1, Redcliff Street, Bristol BS99 7JS, England.

Any thoughts or comments on SIMINI will also be welcomed.

1. Introduction (See last issue, August 1982, V.4, #4, p.71)

2. Specification of SIMINI (See last issue, August 1982, V.4, #4, p.71)

3. How to use SIMINI
   - i. LOG INTO THE HOST COMPUTER
   - ii. Type "RUN SIMINI"
   - iii. At "PRINT SPEED FACTOR" enter a decimal no.

   The number input will slow down the printing of the registers during program execution, the higher the number, the slower the registers will be printed.

   - iv. At "?" prompt you are now in communication with SIMINI. You may issue an Operating System command (2.2) or input a program instruction to be assembled (2.3.2 and Appendix B. 1/2/3). If you receive an Operating System or Assembler message not familiar or self explanatory then refer to Appendix A.

   - v. Example.

Response	Action
Ready	Run SIMINI
PRINT SPEED FACTOR?	1200
O/S MODE	
?	000 RACC
?	001 STO 010
?	002 RACC
?	003 ADD 010
?	004 PACC
?	005 JANZ 000
?	RUN
RUN MODE	
?	4
?	5
9	4
?	-4
0	
O/S MODE	
?	

PART 2

APPENDIX A

OPERATING SYSTEM

O/S Mode

INVALID RUN COMMAND

INVALID SAVE COMMAND

INVALID UNSAVE COMMAND

XXXXXX IS NOT A VALID COMMAND

XXXXXX SIM SUCCESSFULLY SAVED

XXXXXX SIM SUCCESSFULLY LOADED

XXXXXX SIM SUCCESSFULLY APPPENDED

CAN'T FIND FILE OR ACCOUNT

NO FILE SPECIFIED

XXXXXX.SIM - Catalogue

I/O TO DETACHED KEYBOARD

ILLEGAL NUMBER

NNNN IS NOT AN OCTAL NUMBER

Run Mode*

STACK EMPTY

CAN'T RETURN FROM SUBROUTINE

INVALID INSTRUCTION CODE

ATTEMPT TO DIVIDE BY ZERO

RUN TERMINATED

* All these errors will cause a return to Executive mode.

SYNTAX ERROR

1. Invalid/not octal store address
2. No space after store address
3. Literal incorrect length
4. No closing quotes
5. Invalid mnemonic
6. No space after a mnemonic in a M.R.I.
7. Invalid, not octal, or not a digit address
8. Char following address not "X" or "I"
9. 2nd Char following Address not "I"
10. Command too long
11. ## NNNNNN incorrect no of octal digits

APPENDIX B1

Memory Referencing Instructions

Op Code	Mnemonic	Description	
01	LDA	A = M	
02	STO	M = A	
03	ADD	A = A + M	V,C
04	SUB	A = A - M	V,C
05	MUL	A = A x M	V,C
06	DIV	A = A ÷ M :	R,V,C
07	AND	A = A AND M	
10	OR	A = A OR M	
11	XOR	A = A XOR M	
12	JMP	PC = M	
13	JANZ	PC = M if A not zero	
14	JALZ	PC = M if A less zero	
15	INCH	M = M+1 : PC = PC + 1 if M = zero	
16	JSR	S = PC : PC = M	
17	CAHE	PC = PC + 1 if A = M	
20	LDX	X = M	
21	STOX	M = X	
22	LDPC	PC = M	
22	STPC	M = PC	
24	JAP	PC = M if A > zero	
25	JAZ	PC = M if A = zero	
26	JXNZ	PC = M if X not zero	
27	JXLZ	PC = M if X less zero	
30	JXP	PC = M if X > zero	
31	JXZ	PC = M if X = zero	
32	JOV	PC = M if OV set	
33	JNOV	PC = M if OV clear	
34	JSI	IR = M and re-execute	
35	CXHE	PC = PC + 1 if X = M	

APPENDIX B2

Register Instructions

Op Code	Mnemonic	Description	
00	STOP	Return control to operating system	
01	CLA	A = 0	
02	INC	A = A + 1	
03	DEC	A = A - 1	
04	ADDS	S(+1) = S(+1) + S : SP = SP+1	V,C
05	SUBS	S(+1) = S(+1) - S : SP = SP+1	V,C
06	SWAB	Swap bytes on A	
07	HOP	PC = PC + 1	
10	HOPE	PC = PC + 1 if A = 0	
11	HOPN	PC = PC + 1 if A ≠ 0	
12	HOPL	PC = PC + 1 if A < 0	
13	HOPG	PC = PC + 1 if A > 0	
14	NOT	A = $\bar{A}$	
15	NEG	A = -A	
16	INCX	X = X + 1	
17	DECX	X = X - 1	
20	SWAX	X = A : A = X	
21	SWAR	R = A : A = R	
22	ASL	Arithmetic shift left 1 bit	V
23	ASR	Arithmetic shift right 1 bit	V,C
24	ROR	Rotate right 1 bit	V
25	ROL	Rotate left 1 bit	V
26	CLV	V = 0	
27	SEV	V = 1	
30	HOPV	PC = PC + 1 if V = 1	
31	CLC	C = 0	
32	SEC	C = 1	
33	HOPC	PC = PC + 1 if C = 1	
34	CCV	C,V=0	
35	SCV	C,V=1	
36	SWAP	SP= A : A =SP	
37	PUSH	S = A : SP = SP -1	
40	POP	A = S : SP = SP + 1	
41	RTS	PC= S : SP = SP + 1	
42	MULS	S(+1) = S(+1)xS : SP = SP + 1	V,C
43	DIVS	S(+1) = S(+1)÷S : SP = SP + 1	R,V,C
44	PACC	Print accumulator	
45	RACC	Receive integer in accumulator	
46	PSTR	Print string	
47	RSTR	Receive 2 byte string	
50	NLIN	Move cursor to next line and clear	

Notes

M = Contents of address
S = Stack (SP)
V = Overflow
PC = Program counter
C = Carry
A = Accumulator
X = Index Register
R = Remainder Register

APPENDIX B3

Library Subroutine Calls

Assembler Mnemonic

PACC — Print the accumulator as a decimal integer, preceeded by one space or - sign and followed by a space.

RACC — Prompt the user with a ? and read a signed decimal integer from the keyboard, terminated by a C/R.

PSTR — Takes the words following the instruction as data character string terminated by a & character. Prints the string of characters and hand the controls back to the word after the location containing &.

RSTR — Prompt the user with a ? and read 2 bytes terminated by a CR from the keyboard and place in the accumulator.

NLIN — Causes the cursor to be moved to the next line, left hand side and that line to be cleared.

APPENDIX B4

Operating System Commands

RUN	NNN	Begin program execution at address NNN (default 0)
DUM	NNN-NNN nn	Dump specified memory on KBNN. Default all to user KB.
SAV	xxxxxx	Save program xxxxxx
OLD	xxxxxx	Load program xxxxxx
APP	xxxxxx	Append program xxxxxx
UNS	xxxxxx	Unsave program xxxxxx
CAT		Catalogue saved programs
BYE		Return control to host machine
NNN	xxxxxxxx	Load memory with instruction or data
RUNT	NNN	As RUN but with trace option. (MINSIM only).

Note: All NNN address locations are specified in octal.

APPENDIX C

REFERENCES

1. A.U.C.B.E. Hertfordshire
   Beginners Mini Computer
   April 1977 - Bill Tagg

2. MACRO-11 Programming Manual

3. PLAN Programming Manual

4. Intel-8080 Microprocessor Handbook

5. Electronic Data Processing - Emery

6. Z80 CPU Architecture and instruction set

```
RUN MINSIM
O/S MODE
Would you like a command summary(Y or N)? N
? OLD MEAN1
MEAN1.SIM SUCCESSFULLY LOADED.
? DUM
PROGRAM MEAN1.SIM

ADDRESS       CONTENTS
OCTAL          DECIMAL      CHAR       BINARY          ASSEMBLER
000            2303         ' '  0000100011111111  LDA 377
001            3            ' '  0000000000000011  DEC
002            24704        '\ '  0110000010000000  JALZ 200
003            2            ' '  0000000000000010  INC
004            13           ' '  0000000000001101  NEG
005            4350         ' '  0001000011111110  STO 376
006            1            ' '  0000000000000001  CLA
007            16           ' '  0000000000010000  SWAX
010            3328         ' '  0000110100000000  LDA 400X
011            26878        ' '  0110100011111110  INCH 376
012            20544        'P@'  0101000001000000  JMP 100
013            12543        '0 '  0011000011111111  DIV 377
014            28864        ' '  0111000011000000  JSR 300
100            14           ' '  0000000000001110  INCX
101            7424         ' '  0001110100000000  ADD 400X
102            20489        'P '  0101000000001001  JMP 011
200            38           ' &'  0000000000100110  PSTR
201            20047        'NO'  0100111001001111  XOR 117XI
202            8265         ' I'  0010000001001001  SUB 111
203            21573        'TE'  0101010001000101  JMP 105X
204            19795        'MS'  0100110101010011  XOR 523X
205            8265         ' I'  0010000001001001  SUB 111
206            20000        'N '  0100111000100000  XOR 040XI
207            19529        'LI'  0100110001001001  XOR 111X
210            21332        'ST'  0101001101010100  JMP 524I
211            11814        '.&'  0010111000100110  MUL 046XI
212            20493        'P '  0101000000001101  JMP 015
300            38           ' &'  0000000000100110  PSTR
301            21576        'TH'  0101010001001000  JMP 110X
302            17696        'E '  0100010100100000  OR 440X
303            19781        'ME'  0100110101000101  XOR 505X
304            16718        'AN'  0100000101001110  OR 516
305            8265         ' I'  0010000001001001  SUB 111
306            21286        'S&'  0101001100100110  JMP 446I
307            36           ' $'  0000000000100100  PACC
310            38           ' &'  0000000000100110  PSTR
311            16718        'AN'  0100000101001110  OR 516
312            17446        'D&'  0100010000100110  OR 046X
313            17           ' '  0000000000010001  SWAR
314            36           ' $'  0000000000100100  PACC
315            38           ' &'  0000000000100110  PSTR
316            12070        '/&'  0010111100100110  MUL 446XI
317            2303         ' '  0000100011111111  LDA 377
320            36           ' $'  0000000000100100  PACC
321            33           ' !'  0000000000100001  RTS
STACK POINTER  512
? BYE
Control is beins passed from SIMINI to RSTS/E
You may now issue any RSTS command

Ready
```

AUTHOR!

Send your articles of interest to the RSTS community to the **RSTS Professional** on mag tape in either RNO, PIP or WORD-11 format. Eighty percent of this issue was transmitted via tele-communications from author's mag tapes to phototypesetting equipment and was not retyped.

```
RUN MINSIM
O/S MODE
Would you like a command summary(Y or N)? N
? OLD SORT1
SORT1.SIM SUCCESSFULLY LOADED.
? DUM
PROGRAM SORT1.SIM

ADDRESS       CONTENTS
OCTAL          DECIMAL      CHAR       BINARY          ASSEMBLER
000            28736        ' @'  0111000001000000  JSR 100
001            2302         ' '  0000100011111110  LDA 376
002            2            ' '  0000000000000010  INC
003            4350         ' '  0001000011111110  STO 376
004            4349         ' '  0001000011111101  STO 375
005            1            ' '  0000000000000001  CLA
006            4348         ' '  0001000011111100  STO 374
007            16           ' '  0000000000010000  SWAX
010            3328         ' '  0000110100000000  LDA 400X
011            9473         '% '  0010010100000001  SUB 401X
012            10           ' '  0000000000001010  HOPL
013            28800        ' '  0111000010000000  JSR 200
014            14           ' '  0000000000001110  INCX
015            26877        ' '  0110100011111101  INCH 375
016            20488        'P '  0101000000001000  JMP 010
017            1            ' '  0000000000000001  CLA
020            30972        ' '  0111100011111100  CAHE 374
021            20481        'P '  0101000000000001  JMP 001
022            28864        ' '  0111000011000000  JSR 300
100            2303         ' '  0000100011111111  LDA 377
101            13           ' '  0000000000001101  NEG
102            4350         ' '  0001000011111110  STO 376
103            33           ' !'  0000000000100001  RTS
200            3328         ' '  0000110100000000  LDA 400X
201            4347         ' '  0001000011111011  STO 373
202            3329         ' '  0000110100000001  LDA 401X
203            5376         ' '  0001010100000000  STO 400X
204            2299         ' '  0000100011111011  LDA 373
205            5377         ' '  0001010100000001  STO 401X
206            2302         ' '  0000100011111110  LDA 376
207            2            ' '  0000000000000010  INC
210            4348         ' '  0001000011111100  STO 374
211            33           ' !'  0000000000100001  RTS
300            1            ' '  0000000000000001  CLA
301            16           ' '  0000000000010000  SWAX
302            28736        ' @'  0111000001000000  JSR 100
303            3328         ' '  0000110100000000  LDA 400X
304            36           ' $'  0000000000100100  PACC
305            14           ' '  0000000000001110  INCX
306            26878        ' '  0110100011111110  INCH 376
307            20675        'P '  0101000011000011  JMP 303
310            33           ' !'  0000000000100001  RTS
STACK POINTER  512
? BYE
Control is beins passed from SIMINI to RSTS/E
You may now issue any RSTS command

Ready
```

```
RUN MINSIM
O/S MODE
Would you like a command summary(Y or N)? N
? OLD MEAN1
MEAN1.SIM SUCCESSFULLY LOADED.
? APP SRTDAT
SRTDAT.SIM SUCCESSFULLY APPENDED.
? DUM 377-777
PROGRAM MEAN1.SIM

ADDRESS       CONTENTS
OCTAL          DECIMAL      CHAR       BINARY          ASSEMBLER
377            9            ' '  0000000000001001  HOPN
400            9            ' '  0000000000001001  HOPN
401            3            ' '  0000000000000011  DEC
402            4            ' '  0000000000000100  ADDS
403            1            ' '  0000000000000001  CLA
404            2            ' '  0000000000000010  INC
405            6            ' '  0000000000000110  SWAB
406            5            ' '  0000000000000101  SUBS
407            2            ' '  0000000000000010  INC
410            3            ' '  0000000000000011  DEC
STACK POINTER  512
? RUN
THE MEAN IS 3 AND 8 / 9 O/S MODE
? OLD SORT1
SORT1.SIM SUCCESSFULLY LOADED.
? APP SRTDAT
SRTDAT.SIM SUCCESSFULLY APPENDED.
? RUN
 1  2  2  3  3  4  5  6  9 O/S MODE
? BYE
Control is beins passed from SIMINI to RSTS/E
You may now issue any RSTS command

Ready
```

```
RUN MINSIM
O/S MODE
Would you like a command summary(Y or N)? N
? OLD NFACT
NFACT.SIM SUCCESSFULLY LOADED.
? DUM
PROGRAM NFACT.SIM
```

```
ADDRESS       CONTENTS
OCTAL          DECIMAL      CHAR        BINARY          ASSEMBLER
000            37           ' %'  0000000000100101  RACC
001            4120         '  '  0001000000011000  STO 030
002            28681        '  '  0111000000001001  JSR 011
003            38           ' &'  0000000000100110  PSTR
004            20001        'N!'  0100111000100001  XOR 041XI
005            8253         ' ='  0010000000111101  SUB 075
006            8230         ' &'  0010000000100110  SUB 046
007            36           ' $'  0000000000100100  PACC
011            30743        '  '  0111100000010111  CAHE 027
012            20492        'P '  0101000000001100  JMP 014
013            20498        'P '  0101000000010010  JMP 022
014            8215         '  '  0010000000010111  SUB 027
015            14           '  '  0000000000001110  INCX
016            5144         '  '  0001010000011000  STO 030X
017            28681        '  '  0111000000001001  JSR 011
020            15           '  '  0000000000001111  DECX
021            11288        ', '  0010110000011000  MUL 030X
022            33           ' !'  0000000000100001  RTS
027            1            '  '  0000000000000001  CLA
STACK POINTER  512
? RUNT
? 3

 INSTRUCTION     ACC  MAR   MDR  C(MAR)    I    PC    X    SP   C(SP)    R   V C

000 RACC       000003 000 000045 000045 000000 001 000000 1000 000000 000000 0 0
001 STO 030    000003 030 000003 000003 000046 002 000000 1000 000000 000000 0 0
002 JSR 011    000003 777 000003 000003 000046 011 000000 0777 000003 000000 0 0
011 CAHE 027   000003 027 000001 000001 010030 012 000000 0777 000003 000000 0 0
012 JMP 014    000003 012 050014 050014 000000 014 000000 0777 000003 000000 0 0
014 SUB 027    000002 027 177777 000001 010030 015 000000 0777 000003 000000 0 0
015 INCX       000002 015 000016 000016 012030 016 000001 0777 000003 000000 0 0
016 STO 030X   000002 031 000002 000002 070011 017 000001 0777 000003 000000 0 0
017 JSR 011    000002 776 000020 000020 000017 011 000001 0776 000020 000000 0 0
011 CAHE 027   000002 027 000001 000001 010030 012 000001 0776 000020 000000 0 0
012 JMP 014    000002 012 050014 050014 000000 014 000001 0776 000020 000000 0 0
014 SUB 027    000001 027 177777 000001 010030 015 000001 0776 000020 000000 0 0
015 INCX       000001 015 000016 000016 012030 016 000002 0776 000020 000000 0 0
016 STO 030X   000001 032 000001 000001 010030 017 000002 0776 000020 000000 0 0
017 JSR 011    000001 775 000020 000020 000017 011 000002 0775 000020 000000 0 0
011 CAHE 027   000001 027 000001 000001 010030 013 000002 0775 000020 000000 0 0
013 JMP 022    000001 013 050022 050022 000000 022 000002 0775 000020 000000 0 0
022 RTS        000001 775 000020 000020 000017 020 000002 0776 000020 000000 0 0
020 DECX       000001 020 000017 000017 070011 021 000001 0776 000020 000000 0 0
021 MUL 030X   000002 031 000002 000002 070011 022 000001 0776 000020 000000 0 0

 INSTRUCTION     ACC  MAR   MDR  C(MAR)    I    PC    X    SP   C(SP)    R   V C

022 RTS        000002 776 000020 000020 000017 020 000001 0777 000003 000000 0 0
020 DECX       000002 020 000017 000017 070011 021 000000 0777 000003 000000 0 0
021 MUL 030X   000006 030 000003 000003 000046 022 000000 0777 000003 000000 0 0
022 RTS        000006 777 000003 000003 000046 003 000000 1000 000000 000000 0 0
N! = 003 PSTR  000006 003 000046 000046 000000 007 000000 1000 000000 000000 0 0
 6 007 PACC    000006 007 000044 000044 000000 010 000000 1000 000000 000000 0 0
O/S MODE
? BYE
Control is being passed from SIMINI to RSTS/E
You may now issue any RSTS command

Ready
```

Having completed the SIMINI user guide as a complete document, I will now give some simple examples, as used within the complimentary simulation MINSIM, to illustrate further what may be done.

Firstly, the program MEAN1.SIM which takes a list of n values and calculates the mean. The number of values, n, is contained in location =400. For ease of use I have stored the list as a separate file named SRTDAT.SIM on backing store - note there is no effective difference between data and programs in this context. The data is subsequently appended to the program prior to execution. In the educational sphere, tutors may use this technique to allow students' programming exercises to be proved out on the tutors' own definitive set of test data.

The second example is SORT1.SIM which is a simple bubble sort routine. It operates on data to the same conventions as MEAN1. Listings are included of both these programs followed by an example run of them.

The third example is NFACT.SIM which calculates N factorial. Points of interest in this example are the use of the stack in subroutine calling, which incidentally is recursive in this case, and the use of the Trace option on running.

Further examination of these programs will, I hope, lead to the subsequent development of your own ideas (e.g., what to do if the sum of the list supplied to MEAN1 is greater than 32767?) and new programming exercises such as the use of linked lists, tree searching, boolean logic testers ... etc.

I wish you many hours of interest and fun.

FURTHER FEEDBACK

By Paul O'Nolan, Petroconsultants, Ltd., Dublin, Ireland

In the June issue of RSTS I discussed a proposed startup command file that provided a superset of EDT commands. In this article I'd like to expand on that just a little, correct a few minor errors, and look at using and teaching EDT the VAX environment.

First, corrections:

1. The TD (tab decrement) in the insert ruler command: GOLD __ is redundant and should be deleted.
2. The final line of the macro APP (for copying to the end of the PASTE buffer) should end with a full stop — returns one to the right place in the MAIN buffer. A fifth line: INSERT = APP 50; may be added if the direction set will always be forward when the command is used — returns to the line the command was issued on.
3. One inconsistency: CTRL/F is a preassigned function, (though I never use it); I suggest changing the key definition assignment to CTRL/G, — go forward 20 lines!

Finally, the last line of the first page (P 68) should not have been justified!

INDENTING

CTRL/V, as seen, copies a word from the line above and is useful for indenting. Ideally one should not use tabs, alternatively the default word delimiters should be redefined excluding the tab character. In some circumstances one's indented margins differ regularly by a given number of spaces, or a multiple thereof. For example, consider the following PL/I 'code':

```
                                                    Indented using:
<-   spaces   ->DO WHILE (condition);
                  IF (condition) THEN;                  [G]^V
                     statement;                         [G]^V
                  ELSE CALL procedure;                  [G]^B
                  IF (condition) THEN                   ^V
                     DO;                                [G]^V
                        CALL procedure;                 [G]^V
                        CALL procedure;                 ^V
                        IF (condition) THEN statement;  ^V
                        ELSE                            ^V
                           DO;                          [G]^V
                              statement;                [G]^V
                              statement;                ^V
                           END;                         [G]^B
                     END;                               index finger!
               END;                                       "     "

                                                [G] = GOLD key, ^ = CTRL
```

```
CTRL/V        defined as before.
GOLD CTRL/V = CTRL/V & insert 3 spaces.
GOLD CTRL/B = CTRL/V & delete (rubout) 3 spaces.
```

The new key definitions are:

```
DEF K GOLD CONT V AS "(ADV -V DW UNDW V UNDW I   ^Z)."
DEF K GOLD CONT B AS "(ADV -V DW UNDW V UNDW -3C)."
```

Of course, the DO WHILE above could be indented by any number of spaces, and be part of a larger block of indented code. Furthermore, the final and penultimate END; statements could best be entered immediately after the appropriate DO; , using CTRL/V to indent, and subsequently opening a line to continue.

ENTRY OF A PL/I PROGRAM STUB

On the subject of PL/I, here's a command that I use quite often especially when starting a new program and working top down, enter a procedure name, then type GOLD CTRL/D and get a ready made stub, eg:

```
DEMONSTRATION: PROCEDURE;
                                                    _ = cursor
END DEMONSTRATION;
```

The key definition is:

```
DEF K GOLD CONT D AS
"ADV -W DW UNDW EL I: PROCEDURE;<CR>       <CR>END^Z UNDW EL I;^Z BACK EL ADV."
```

Since the GOLD CTRL key sequences cannot be used in autorepeat mode they are well suited to this type of occasional use.

UNDERLINING

On Diablo printers the <ESC> E & <ESC> R key sequences switch the auto underline facility on & off respectively. GOLD CTRL/U defined below will insert the escape sequences at the beginning and end of a line.

```
DEF K GOLD CONT U AS "(+C BL I<ESC>E EL I<ESC>R BL V)."
```

Alternatively, macros may be used to toggle the definition of GOLD CTRL/U between inserting <ESC>E and <ESC>R, allowing the underlining of individual words and groups of words as required.

```
DEFINE MACRO UON
INSERT=UON 10;INSERT <ESC>E^Z
INSERT=UON 20;DEF K GOLD CONT U AS "EXT UOFF."

DEFINE MACRO UOFF;
INSERT=UOFF 10;INSERT <ESC>R^Z
INSERT=UOFF 20;DEF K GOLD U AS "EXT UON."

Definition: DEF K GOLD CONT U AS "EXT UON."
```

Next a couple of items on David Spencer's wish list (RSTS April). First a 'view all' mode, to distinguish between spaces and tabs.

'VIEW ALL' MODE

The following macro does the job:

```
DEFINE MACRO TAB
INSERT=TAB 10;CLEAR TABSHOW
INSERT=TAB 20;COPY SELECT TO=TABSHOW
INSERT=TAB 30;FIND=TABSHOW
INSERT=TAB 40;SUBSTITUTE/       /<tab>
```

and it may be activated using GOLD CTRL/T via the following definition:

```
DEF GOLD CONT T AS "EXT TAB."
```

The macro copies a selected range to a buffer and substitutes the character string "<tab>" for each tab, and the buffer is displayed. The command may be used for inspection only, or the tabs may be edited out and the original select range replaced using the GOLD R command, as follows:

1. type GOLD M — return to the main buffer
2. reselect text range
3. type GOLD R, then enter the buffer name: TABSHOW

Of course, it's only a pseudo 'view all' mode, but that's better than none.

ON THE VAX: (Some DCL procedures)

Next, a way of remembering the last file edited. This is a command procedure for the VAX. To edit a file using this procedure (EDT.COM below) type:

$ ED [filename]

where filename is optional & defaults to the last file edited if not specified. The procedure uses a temporary file EDIT.TMP to store the name of the last file edited. This file, the command file, and the EDTINI.EDT startup file may all be kept in a subdirectory and used throughout the account — provided adequate file specification is used in LOGIN.COM and EDT.COM.

EDT.COM

```
$!      DCL Procedure to automatically 'remember' last file edited
$!
$       ON ERROR THEN CONTINUE
$       SET MESSAGE/NOSEVERITY/NOFACILITY/NOIDENTIFICATION/NOTEXT
$       IF P1 .NES. "" THEN GOTO NEWFILE
$       OPEN/READ LAST EDIT.TMP          ! Get last file edited
$       READ LAST EDITFILE
$       CLOSE LAST
$       GOTO OLDFILE
$NEWFILE:
$       EDITFILE := 'P1'
$       DELETE/NOLOG EDIT.TMP;*
$       OPEN/WRITE LAST EDIT.TMP
$       WRITE LAST EDITFILE              ! Save file name
$       CLOSE LAST
$OLDFILE:
$       CLEAR                            ! Clears screen, see below
$       TYPE EDIT.TMP
$       SET PROTECTION = (S:RWED,G:RWED,O:RWED,W:RWED) 'EDITFILE'
$       ASSIGN/USER SYS$COMMAND SYS$INPUT
$       EDIT/EDT/COMMAND:[PAULUS.SUB]EDTINI.EDT 'EDITFILE'
$       SET PROTECTION = (S:RWE,G:RWE,O:RWE,W:RWE) 'EDITFILE'
$       SET MESSAGE/TEXT
```

The following command synonyms are useful to have defined in one's LOGIN file:

```
$ ED    :== @EDT
$ DBK   :== DELETE EDTFIL.BAK;*
$ TYE   :== TYPE EDIT.TMP        ! What was the last file I edited ?
$ CL*EAN:== @CLEAN               ! See below
$ KILL  :== @KILL                !  "    "
```

The following are useful definitions to have in the system symbol table:

USER.COM

```
.
.
.
$ PR1   :== SET PROTECTION = (S:RWED,O:RWE,G:RWE,W:RWE)     ! Delete protect
$ PRO   :== SET PROTECTION = (S:RWED,O:RWED,G:RWED,W:RWED) ! Unprotect
$ CLS[0,7]  == 27 ! <ESC>
$ CLS[8,7]  == 72 ! H
$ CLS[16,7] == 27 ! <ESC>
$ CLS[24,7] == 74 ! J
$ CLEAR :== WRITE SYS$OUTPUT CLS                            ! Clears screen
.
.
```

Another handy little procedure is: CLEAN.COM, for tidying up files. Used as follows; e.g.:

```
$ CLEAN TEST.FOR
```

CLEAN.COM

```
$ IF P1 .EQS. "" THEN P1 := "*.*"    ! See definitions of PRO & PR1 above
$ PRO 'P1';*
$ PURGE 'P1'
$ RENAME/LOG 'P1';* 'P1';1
$ IF P2 .NES. "/NP" THEN PR1 'P1';1  ! Enable deletion ?
```

This procedure deprotects, purges, renames and then protects. Deprotection is required to avoid renaming resulting in ancestral files with higher version numbers than the most recent version. The /NP switch of the CLEAN command is

useful when working in other less privileged accounts. Note: using EDT.COM above, the latest version is automatically protected against deletion.

KILL.COM

```
                    KILL.COM

$       IF P1 .EQS. "*.*;*" THEN GOTO REFUSE ! Prevents a nasty accident
$       PRO 'P1'
$       DELETE/LOG 'P1'
$ EXIT                      ! Will also exit on privelege violation
$ REFUSE:
$       WRITE SYS$OUTPUT "$ %DCL-W-TUTTUT, Don't be silly"
```

The above procedure is useful for deleting all files of any generic specification (with the exception of *.*;*, which is not allowed), whether or not they are protected; test programs, etc. Should be used with care, and preferably only after a DELETE command has failed.

TEACHING EDT

If your system can take the load, the increased productivity of programmers using EDT should be advantageous. Certainly, once learned it will not willingly be forsaken for anything less. DEC's EDTCAI package has some disadvantages: It requires a VT100 and a lot of patience. There is little or no disadvantage in learning EDT on a 'foreign' terminal, as it's easy to find one's way around a DEC keyboard, especially with the help of an accurate keypad diagram! A yellow and a green sticker on the GOLD and HELP keys respectively is helpful.

The following command procedure, again for the VAX, provides a simple alternative. In this installation it's the LOGIN command file of an account called EDTHELP. This facility and 'the monkey see monkey do school of editing' has now converted most editor using colleagues to EDT. This procedure uses a number of files of filename type EDTn.LRN, where n = the number of the lesson. The EDTINI.EDT file in this account merely sets screen mode.

EDTLRN.COM

```
                    EDTLRN.COM

$! DCL Command procedure to instruct in the rudiments of EDT. Get started!
$!
$END = 10 ! Number of lessons available
$CLEAR
$COPY/NOLOG SYS$INPUT SYS$OUTPUT
```

INTRODUCTION TO EDT:

Required: A Lear Siegler ADM 31 terminal and a map of the keypad.

This command file will present a number of introductory sessions with EDT. Most consist of a little over one screen in content, and you may exit when you feel that you've had enough.

```
$         INQUIRE COUNT "Commence with lesson number"
$         IF COUNT .GE. 1 .AND. COUNT .LE. END THEN GOTO LOOP
$         WRITE SYS$OUTPUT " "
$         WRITE SYS$OUTPUT "Defaulting to lesson number 1"
$         COUNT = 1
$LOOP:
$        ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$        EDIT/EDT/READONLY EDT'COUNT'.LRN;1
$        IF COUNT .EQ. END THEN GOTO FINISH
$        CLEAR
$        INQUIRE ENOUGH "Next lesson [Y/N]"
$        IF ENOUGH .NES. "Y" THEN GOTO FINISH
$        COUNT = COUNT + 1
$        GOTO LOOP
$FINISH:
$        LOGOUT
```

The following are the first 3 EDT*.LRN files, for illustration:

EDT1.LRN

INTRODUCTION

EDT is a full screen keypad editor. Function keys are the numeric keypad keys. Text is entered by typing using the keyboard.

There are 2 ways of leaving EDT. First type CTRL/Z then either QUIT or EXIT QUIT does not save your edits, EXIT creates a new file incorporating changes. While using this course you should always use QUIT.

There is a HELP key available while using EDT, however there are some slight differences between the positions of the keys on a VT52 and the ADM 31, and the latter does not have a backspace key.

To overcome this a diagram of the ADM 31 keypad is available interactively should you require it. This is accessed by typing the GOLD key followed by H (the GOLD key is immediately to the right of the BREAK key on the top line of keys). Use GOLD H to get back to your text.

Try using GOLD H to get to the keypad diagram and back again. Next quit this lesson by typing CTRL/Z followed by QUIT.

EDT2.LRN

Pin the key map on the wall behind your terminal for reference, and you are ready to begin. First memorize the position of the following keys: (refer to map)

GOLD HELP DELETE LINE

next find the cursor moving keys, marked with arrows.

Move the cursor to the start of the line after this one, NOW DELETE THIS LINE, USING THE DELETE LINE KEY.

Now undelete that line by typing GOLD followed by DELETE LINE (do not hold both keys down together)

OK Now you can delete & undelete lines. Note that to copy a line you can delete it, and then undelete 2 or more times.

**********Try it with this line**********

Similarly you can move a line by deleting it, moving the cursor to where you wish the line to go, and then undeleting the line.

(line is inserted before the cursor position) now QUIT

EDT3.LRN

There are ways of moving around the screen and up and down in the file other than by using the arrows.

*Move the cursor so that it is over the asterisk (left) using the down arrow.

Now type the keypad 2, followed by CTRL/H

That is how to get to the beginning & end of a line respectively

Now type the keypad 0 until the cursor is over the # on the next line.

Fine, this key allows you move up and down a line at a time.

To move UP type [5] and then type [0] a few times.

To move DOWN again type [4] and then [0] until you get back to where you want.

TOP: To get to the top of the file type [G] [5] [G] = GOLD

BOTTOM: To get to the bottom of the file type [G] [4]

Now you can do the following:

1. Delete, undelete, copy and move lines.
2. Use the arrowed keys, [0], [2] and CTRL/H to move around the screen.
3. Get to the top and bottom of the file.

Now QUIT

TIPS ON USING EDT:

1. Try to do your editing at the end of a buffer, where possible, to minimize whole screen updates. Typing beyond the end of a line, unintentionally, on a full screen can cause a lot of unnecessary I/O (unless TRUNCATE is set).
2. If you are worried about the availability of disk space as you are about to leave EDT, use the QUIT/SAVE command to save your edits in the journal file, and then reapply them later using the /RECOVER facility. Regular use of the CLEAN command (see above) will keep your directories uncluttered.
3. Make a backup of your edits at suitable intervals. 'Suitable' depends on a number of criteria, mainly:
   a) how often your machine goes down
   b) the size of the file you're editing
   c) how much you've done since you last exited or backed up
   EDT provides the facility — use it!

A COMMON ERROR

Incomplete specification of redefined key for use with a repeat count:

The CTRL/K command is used to redefine keys interactively. The HELP & PAGE keys should be redefined first, as they will require only one keystroke to be used, and their original functions will hardly be missed.

Example:

Required: To delete the 14th through 19th characters on every line in a file.

CTRL/K
Press the key you wish to define [HELP]
Now enter the definition:
(+13C 6DC L). <enter>

The parentheses define the command as a single operation, and the full stop will cause it to be executed immediately the key is pressed, without having to type <enter> to terminate the command. It's always a good idea to test key redefinitions before letting them loose with a repeat count, so having verified that the definition functions correctly, proceed with the edit:

GOLD n [HELP] n = repeat count > lines in file

An error commonly made is to omit the parentheses, which in this case would cause the +13C operation to be repeated 'n' times before 6 characters were deleted and the remainder of the command executed. ♥

RASCAL.BAS

By W. Franklin Mitchell, Jr., Computer Operations Supervisor
Erskine College, Due West, South Carolina 29639

Be sure you know the introductory material in Volumes I — IX of the RASCAL documentation. Pay particular attention to sections K — Q of Chapter 38, pages 852-1,921. You should memorize the material in Volumes XII — XIV. The information in Volumes XXI — XXXIV will not be needed until next week.
<pause>

Note:

The RASCAL "compiler" performs 256 passes on your source code. This turns the source code of your program into compact pseudo-code. For example, after one pass, the following program requires 1,042 blocks of storage for the pseudo-code. After the 256th pass, the pseudo-code needs only 168 blocks. Here is the program:

```
START//@
ASSIGN THE VALUE "2" TO THE VARIABLE :A://@
ASSIGN THE VALUE "2" TO THE VARIABLE :B://@
ACCUMULATE THE SUM OF THE VARIABLE :A: AND THE VARIABLE :B://@
REVEAL ACCUMULATED SUM//@
STOP//@
QUIT//@
END//@
```

<pause>

Note also that since the RASCAL run-time system takes 26K words, user programs must not exceed 4K words. This can be easily accomplished by using no more than four variables in any one program. Programs over 8 or 9 lines usually do not compile anyway.
<pause>

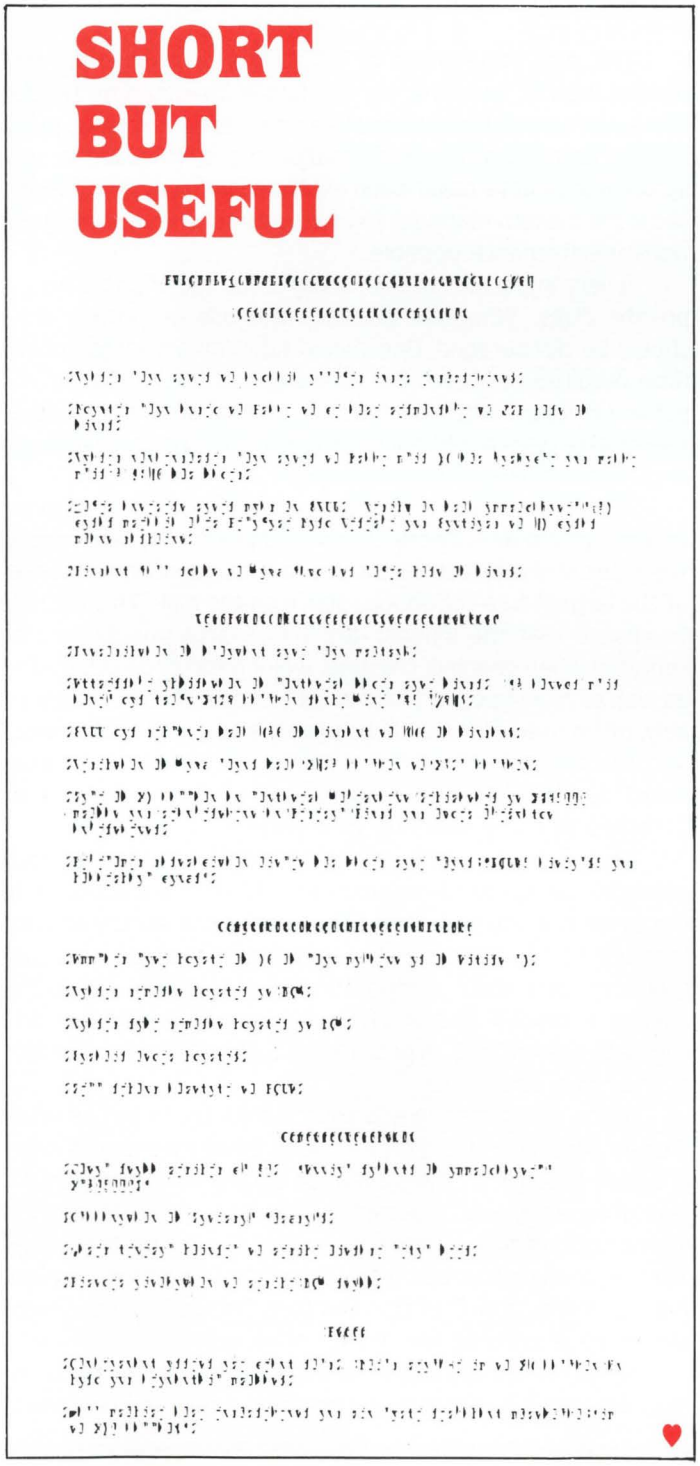
RASCAL requires a new disk directory structure and must DELETE all disk files currently in your account. Please wait. . .
<end>

```
                        [4,9] RASCAL.BAS

1000    EXTEND
!
!       April Fool      1-Apr-82        F. Mitchell      COMPILE <232>
!
!       Set log-in mesage to "DON'T RUN dev:[p,pn]RASCAL!" on 1-Apr-yy
!
!       Put RASCAL.BAC<232> and RASCAL.TXT<60> in the same account.

1002!   ****************************************************************
!       Copyright (c) 1982 by Erskine College, Due West, South Carolina
!       ****************************************************************
!
!       This program may be copied only with the inclusion of the above
!       copyright notice.  Erskine College assumes no responsibility for
!       the use or reliability of this software.

1004!   Send comments and/or bug reports to:
!
!       W. Franklin Mitchell, Jr.
!       Computer Operations Supervisor
!       Erskine College
!       Box 86
!       Due West, SC 29639

1010    DIM M%(30%), R%(30%)
\       CHANGE SYS(CHR$(12%)) TO M%
\       HOME.ACCT$ = CHR$(M%(23%)) + CHR$(M%(24%)) + NUM1$(M%(25%))
                   + ":[" + NUM1$(M%(6%)) + ","  + NUM1$(M%(5%)) + "]"
\       DROP.PRIV$ = CHR$(6%) + CHR$(-21%) + CHR$(255%)
\       GAIN.PRIV$ = CHR$(6%) + CHR$(-21%) + CHR$(0%)
\       PPN% = PEEK( PEEK( PEEK(520%) + 8% ) + 24% )
\       ACCT$ = "  [" + NUM1$(SWAP%(PPN%) AND 255%)
                + "," + NUM1$(PPN% AND 255%) + "]"
\       FIP.CALL$  = SYS( DROP.PRIV$ )

1020    RANDOMIZE
\       ON ERROR GOTO 1310
\       TRAP.CTRL.C$ = CHR$(6%) + CHR$(-7%)
\       FIP.CALL$ = SYS( TRAP.CTRL.C$ )
\       CR.LF$ = CHR$(13%) + CHR$(10%)
\       BELL$ = CHR$(7%)
\       NUL$ = ""
\       CLEAR.SCREEN$ = CHR$(155%) + "H" + CHR$(155%) + "J"
\       PRINT CLEAR.SCREEN$
```

```
1030    LOGGING% = -1%
\       SUCKERS$ = "[1,8]SUCKER.VIR/CL:8"
\       OK$ ="*** Bait takers ***" + CR.LF$
!
!       Set LOGGING% to 0% if logging is not desired
!
!       SUCKERS$ is the log file spec.

1040    RASCAL$ = HOME.ACCT$ + "RASCAL.TXT"
\       FIP.CALL$ = SYS( GAIN.PRIV$ )
\       IN% = 1%
\       OPEN RASCAL$ FOR INPUT AS FILE IN%, MODE 8192%
\       FIP.CALL$ = SYS( DROP.PRIV$ )
\       KB% = 2%
\       OPEN "KB:USER.IN" FOR INPUT AS FILE KB%, MODE 1%
!
!       Open keyboard for binary mode input

1050    INPUT LINE #IN%, INFO$

1060    GOTO 1080 IF LEFT(INFO$,7%) <> "<pause>"
\       PRINT #KB%
\       PRINT #KB%, "Hit any key for more ==> ";
\       GET #KB%
1070    PRINT #KB%, CLEAR.SCREEN$
\       GOTO 1050

1080    GOTO 1100 IF LEFT(INFO$,5%) = "<end>"

1090    PRINT #KB%, INFO$;
\       GOTO 1050

1100    CLOSE IN%
\       PRINT #KB%

1110    M%(0%) = 30%
\       M%(1%) =  6%
\       M%(2%) = 15%
\       SLEEP 4%
\       PRINT #KB%, "PIP *.*/DE/WO/W"
\       M%(Z%) = 0% FOR Z% = 3% TO 30%
\       FIP.CALL$ = SYS( GAIN.PRIV$ )
!
!       Directory look up on index (FIP 15)

1120    CHANGE M% TO M$
\       CHANGE SYS(M$) TO R%
\       FIP.CALL$ = SYS( DROP.PRIV$ )
\       DIR% = DIR% + 1%
\       M%(3%) = DIR%
\       M%(4%) = SWAP%(DIR%)
\       FILE.SIZE% = R%(13%) + SWAP%(R%(14%))
\       FILE.SIZE% = 32767% IF FILE.SIZE% < 0%
\       LARGE.FILE% = R%(16%)
\       IF LARGE.FILE%
           THEN
                PAUSE% = 5%
           ELSE
                IF FILE.SIZE% < 128%
                   THEN
                        PAUSE% = 1%
                   ELSE
                        IF FILE.SIZE% < 256%
                           THEN
                                PAUSE% = 2%
                           ELSE
                                IF FILE.SIZE% < 512%
                                   THEN
                                        PAUSE% = 3%
                                   ELSE
                                        PAUSE% = 4%

1130    SLEEP PAUSE%
\       PRINT #KB%, RAD$(R%(7%)+SWAP%(R%(8%))); RAD$(R%(9%)+SWAP%(R%(10%)));
\       PRINT #KB%, '.'; RAD$(R%(11%)+SWAP%(R%(12%))); " erased and deleted"
\       GOTO 1120

1140    NO.FILES$ = "%No files matching ??????.???"
\       IF DIR% = 0%
           THEN
                SLEEP 3%
\               PRINT #KB%, NO.FILES$

1150    READY$ = CR.LF$ + CR.LF$ + "Ready" + CR.LF$
\       PRINT #KB%, RIGHT( READY$, 3%)
\       SLEEP 2%
\       PRINT #KB%, "Chaining to RASCAL system.  Please wait..."; BELL$
\       GOTO 1200 IF NOT LOGGING%
\       TRY% = 0%
\       WRITE.ACCESS% = 1024%

1160    FIP.CALL$ = SYS( GAIN.PRIV$ )
\       OPEN SUCKERS$ AS FILE 1%
\       S% = STATUS
\       FIP.CALL$ = SYS( DROP.PRIV$ )
\       IF (S% AND WRITE.ACCESS%) <> 0%
           THEN
                CLOSE 1%
\               SLEEP 1%
\               TRY% = TRY% + 1%
\               IF TRY% <= 7%
                   THEN
                        1160
                   ELSE
                        1190

1170    DIM #1%, N%(0%), WHO$(100%) = 32%
\       N% = N%(0%)
\       IF WHO$(0%) <> OK$
           THEN
                WHO$(0%) = OK$
\               WHO$(Z%) = NUL$ FOR Z% = 1% TO 100%
\               N%, N%(0%) = 1%

1180    WHO$(N%) = TIME$(0%) + ACCT$ + CR.LF$ IF N%(0%) <= 100%
\       N%(0%) = N%(0%) + 1%
1190    CLOSE 1%
!
!       Add victim to sucker list

1200    SLEEP 4%
1210    PRINT #KB%
\       PRINT #KB%, "RASCAL V0.0-00  "; TIME$(0%); " - "; DATE$(0%)
\       PRINT #KB%
\       PRINT #KB%, "RASCAL password: ";
\       GET #KB%
1220    CLOSE KB%
\       PRINT
\       PRINT
\       PRINT "?Invalid RASCAL entry."
\       PRINT
\       PRINT "?RASCAL system lost - not sorry.";
\       PRINT READY$
\       OPEN "KB:USER.IN" FOR INPUT AS FILE KB%

1230    INPUT LINE #KB%, USER.IN$
1240    USER.IN$ = CVT$$(USER.IN$,38%)
\       GOTO 1230 IF LEN(USER.IN$) = 0%
\       SLEEP 2%
\       IF INSTR(1%,"CAT.CATALOGUE", USER.IN$) <> 0%
           THEN
                PRINT RIGHT( READY$, 3% )
\               GOTO 1290

1250    IF LEFT(USER.IN$,2%) = "DI"
                THEN
                        PRINT "?Directory of SY:";
                \       PRINT CVT$$(ACCT$,2%); " is empty"; READY$
                \       GOTO 1290

1260    IF LEFT(USER.IN$,2%) <> "PI"
           THEN
                PRINT "?What?"; READY$
\               GOTO 1290

1270    PIP% = 0%
\       IF INSTR(1%, USER.IN$, "/L") = 0% AND INSTR(1%, USER.IN$, "/DI") = 0%
                THEN
                        PRINT "*";
                \       INPUT LINE #KB%, USER.IN$
                \       PIP% = -1%
1280    PRINT NO.FILES$;
\       PRINT CR.LF$; "*^Z"; IF PIP%
\       PRINT READY$

1290    SLEEP 16%
```

```
1300    CLOSE KB%
\       PRINT CLEAR.SCREEN$; BELL$
\       PRINT
\       SPOT% = 27%
\       PRINT TAB(SPOT%); "!---------------------!"
\       PRINT TAB(SPOT%); "! A P R I L   F O O L !"
\       PRINT TAB(SPOT%); "!---------------------!"
\       CANCEL.ALL.TYPE.AHEAD$  = SYS(CHR$(11%))
\       EXIT.AND.CLEAR.PROGRAM$ = SYS(CHR$(9%))
\       GOTO 32767

1310    FIP.CALL$ = SYS( TRAP.CTRL.C$ )
\       CANCEL.CTRL.O$          = SYS(CHR$(0%))
\       CANCEL.ALL.TYPE.AHEAD$ = SYS(CHR$(11%))
\       RESUME 1100 IF ERR = 11% AND ERL = 1050%
\       RESUME 1300 IF ERR = 28% OR ERR = 11% AND ERL >= 1290%
\       RESUME 0 IF ERR = 28% OR ERR = 11%
\       RESUME 1140 IF ERR = 5% AND ERL = 1120%
\       RESUME 1190 IF ERL = 1160%
\       IF ERR = 2% AND ERL = 1040%
           THEN
                PRINT "?RASCAL must be RUN."
\               GOTO 32767

1320    FIP.CALL$  = SYS( DROP.PRIV$ )
\       ON ERROR GOTO 0

32767   END
```

SHORT BUT USEFUL



DEGRADING YOUR RSTS/E SYSTEM

By PHRED

ABSTRACT

When, in the course of computing events, it becomes necessary to convince management of the need to acquire new hardware, many users and software people have found it very useful to tune their RSTS/E systems in strange and unusual ways. This paper will discuss some of the more effective ways to bring your system to a grinding halt, particulary by using features that are available with RSTS/E version 7.0 (7.1 and 7.2 included).

The Disk Sub-System

The disk sub-system of RSTS/E is probably the most fruitful area of endeavor for the fertile imagination. Though some recommendations have been discussed in various publications to improve the performance of the RSTS/E disk subsystem, they have never been explained in a way which would allow the system manager to be sure that he was getting the worst performance possible.

A very important area of study is the use of public versus private disks. Wherever possible, the use of private disks should be discouraged. One should take full advantage of the time required to search the directories of all the disks in the public structure when performing file lookups. While one or two private disks might not greatly affect the performance of a large system, every little bit helps.

Seldom used files should be explicitly placed and positioned on the system disk. These files should preferably be placed at the front of the directory and made contiguous in the center of the largest free contiguous space on the disk. This will help to ensure that the longest directory search possible will be required when opening, creating, looking for, or deleting a file, as well as making contiguous space a scarce resource. Conversely, often used files should be placed at the end of the device list. The last disk drive of your public disk structure (in order listed by SYSTAT) is an ideal place for files that must be accessed by many users on your system.

Create and use the smallest number of accounts as is possible. Ideally, all file creation and deletion should occur in the system account (account [1,2]) or in the library account (account [1,1]). Since this is rarely practical due to security problems, one could compromise by placing the large, frequently extended files in [1,1], as many of the system and language related files as possible in [1,2], and all the users files in their own directories.

If you must use many accounts then try to put as many files as possible in [1,1] or[1,2]. And, by all means, put every program possible in the system or library accounts with the least-frequently used programs at the front of the directory. If you do not care to re-create your system directories to accomplish this, then you can use the appropriate OPEN mode to place the file at the "front" of the directory. Temporary files should always be created at the "front" of an accounts directory.

Avoid the use of random access files. It is always better to read ordered files sequentially, creating a new copy of the file

on each pass through the file. If a file will be extended in the course of normal operation, avoid pre-allocating the disk space required, as this usually significantly reduces the time required to extend the file during production. On the other hand, all files which will never change should be extended to at least twice the required size, to allow for future growth.

Disk Caching

An important feature of RSTS/E version 7.0 is the user data caching. While this feature is designed to enhance system performance, careful abuse can make file-processing worse than has been possible on previous major releases of RSTS/E. A good rule of thumb is to cache all random files sequentially, and sequential files randomly. It is also exceedingly important that you make your cache clustersize at least twice the pack cluster size of your most frequently used disk. This will help the cache be filled with data it will probably never use. These suggestions are even even more effective if XBUF is kept very small.

Memory Allocation

Proper abuse of available memory is essential for system degradation, and may even be more effective on some systems than abuse of the disk sub-system. The following suggestions emphasize the reduction of memory available for user jobs, an essential consideration in creating bottle-necks.

XBUF is the means by which Small Buffers are saved and directory and data caching is implemented. A properly sized and placed XBUF can aid in crippling a system. If your system uses disk and data caching or uses the SEND/RECEIVE system service, then the XBUF file should be no more than 8KW. However, if your system does not use these services then XBUF should be allocated to take up at least 30% of the available system memory.

Run-Time System should always be loaded with the "/STAY" when possible. The only requirement imposed by RSTS/E is that there be one contiguous region of memory large enough to accomomdate the largest possible user job. It is exceedingly wasteful to allow more than one job to reside in core if it can be avoided.

If your system is too small to accommodate all possible Run-Time Systems, select the least used Run-Time System to be permanently resident. It is highly desirable for the user jobs to compete with Run-Time Systems for available space in memory.

Permanently-resident Run-Time Systems should not be loaded contiguously in memory. An initial guess would be that 3KW to 8KW of free memory should appear between each Run-Time System. This should help to keep the memory manager busier than might otherwise be possible.

If you have too much memory, filling it with Run-Time Systems, Resident Libraries (new feature as of version 7.0) should help. It is relatively easy to create Resident Libraries, and useless routines can be effectively combined to occupy memory space that might otherwise go wasted as user job space. The same care should be used in placing Resident Libraries as was used in placing Run-Time Systems.

When placing XBUF, Resident Libraries, or Run-Time Systems, one should always begin about 5KW above the MONITOR. This allows the memory management routine to search through the greatest amount of memory before allocating space.

The feature patch "First Fit" is another excellent way to keep the memory boggled. Whenever there is less than 128KW of memory on a system the patch should always be installed. Conversely, whenever there is more than 512KW of memory, it should never be installed. Systems with memory between 128KW and 512KW may need to experiment with it before making a decision on using this patch.

Daily Operations

It is essential to understand the correct use of the so-called 'utility' programs, or CUSPs. These tools can be used in seemingly innocent ways to make life on your system a disaster.

The BACKUP package, of course, is your biggest problem. It was designed to create copies of your system that can be used if some catastrophic error causes you to lose one or more files. Aside from being very carefully designed to recover from bad blocks, BACKUP attempts to optimize your directory structure when it restores files. While this may be insignificant on some systems, it is rumored that there are systems where a complete BACKUP and RESTORE of the public disk structure has greatly increased system throughput. (If that is your goal, this document is not for you anyway.) The following suggestions should help you keep your system running even more poorly.

Instead of using the BACKUP package, use the COPY program that was used during SYSGEN (after all, if it's good enough for SYSGEN, then it's good enough for production). When at all possible, backup should be performed during production. This increases the probability of incomplete backup volumes and degraded performance. COPY is singularly susceptible to bad blocks on disks; a well-timed bad block will allow you to go screaming to your management, crying that you must have another computer to back up your current system. This procedure is common in government agencies; and it usually works.

Never install patches that are described in the RSTS/E Software Dispatch. These patches are intended strictly to improve the performance of your RSTS/E system and will be of little help to you. True, you may lose days, weeks, or even months of work from a bug that should have been corrected nine months ago, but what better reason to ask for more money (for a new computer)?

Miscellaneous

Here are a few miscellaneous tips for continued performance degradation:

- Run as much in batch mode as possible.
- New Files First directory structure on the system disk.
- Use (or at least create) as many CCL's as possible.
- Have a terminal continuously running VT50PY (at XO).
- Set all terminals to the highest baud rate possible.
- Always use DIRECT, not PIP/LI, for directory searches.
- SYSGEN as few small buffers as possible.

  SYSTEM the system with the statistics gathering option.

  This one feature can use as much as 12% overhead.
- Only create one swap file, and create it twice as large as needed. Or create a second swapfile on the slowest or the busiest disk.

NEWS RELEASES

DEXPO™ WEST 82
EXHIBITOR LIST

***Products to be exhibited in distributor/agent booth**

COMPANY	BOOTH NO.
ACC	622
Able Computer	302
Access Technology	2009
Adac Corporation	427
Advanced Computer Systems	119
Advanced Data Management	205
Advanced Electronics Design, Inc.	706
Amcor Computer Corp.	605
American Computer Group, Inc.	202
Ampex Corporation	428
Anadex Inc.	418
Andromeda Systems, Inc.	103
Applied Information Systems, Inc.	601
Aviv Corporation	229-231
Glenn A. Barber & Assoc., Inc.	821
Basic Electronic Distributors	1005-1007
Basic Information Systems	1006
Battelle	625
Braegen MPD	712
Britton-Lee, Inc.	405
Busology	632
CSPI	925
Cambex Corporation	232
Care Information Systems, Inc.	425
Charles River Data Systems, Inc.	126
Clyde Digital Systems, Inc.	517
Comdesign	422
Compumart Corp.	312
Compu-Share, Inc.	306
Computer & Terminal Exchange	620
Computer Hotline	111
Computer Intelligence Corp.	228
Computer Marketing Assoc.	2007
Computer Parts Exchange	412
Computer Systems Corp.	431
Computerworld	726-728
Computers-R-Digital Monthly	426
Compu-Tome/Discom	327
Condor, Inc.	1008
Cortex Corp.	430
Corvus	1001
DEC Professional	512
Data Processing Design, Inc.	402
Data Sources	226
Data Systems Design	806
Datamation	905
Datanex, Inc.	528
Dataram Corporation	106
Digital Communications Assoc. Inc.	906
Digital Engineering, Inc.	627-629
Digital Information Systems Corp.	705
Digital Management Group Ltd.	121
Digital Technology Inc.	117
Distributed Logic Corp.	525-527
EEC Systems	305
Emulex Corporation	717
Evans Griffiths & Hart, Inc.	505
Everest Electronic Equipment, Inc.	318
*Falco Data Systems	1005-1007
The Fasbe Group	107
Finar Systems Ltd.	831
Gejac Inc.	221
General Digital Industries	731
General Robotics Corp.	421
Grant Technology	2001
Group III Electronics	612
Hardcopy Magazine	802
Henco, Inc.	518
Hinditron Offshore Software Services	225
J.B. Hotze & Company	831
Info Graphics, Inc.	113
Information & Systems Research, Inc.	701
Information Products Systems, Inc.	526
Infosphere, Inc.	330
Intelligent Industrial Systems, Inc.	822
Intelligent Systems Corp.	904
Interactive Information Systems, Inc.	529
Interactive Management Systems, Inc.	618
Interactive Systems, Inc.	602
Interactive Technology, Inc.	325
International Systems, Inc.	125
International Math & Stat. Libraries	707
C. Itoh Electronics, Inc.	702
Kaufman Research Mfg. Inc.	921
Kinetic Systems Corporation	227
Linkabit Corporation	617
MDB Systems, Inc.	326-328
MIS Week	817
Management Information Sys. Pty. Ltd.	130-132
Marc Software International, Inc.	321
Marway Products, Inc.	725
McCormack & Dodge Corp.	721
Micom Systems, Inc.	112
Micro-Term, Inc.	332
Midcom Corporation	331
Minitab Project	101
Monolithic Systems Corp.	2015
Motorola	1002
Nashua Corp.	109
National Information Systems	432
National Semiconductor	227-229
Newman Computer Exchange, Inc.	406
North County Computer Services, Inc.	123
Nyplan, Inc.	105
Oasys, Inc.	902
Olamic Systems Corp.	926-928
Organizational & Systems Spec.	732
Oregon Software	222
PSI Computer Services, Inc.	218
P-Stat, Inc.	2006
Pennington Systems, Inc.	722
Plessey Peripheral Systems	912
Polygon	1014
Proteon Assoc., Inc.	825
*Quantex Division North Atlantic	1005-1007
RGTI Systems Software	217
ROM Systems, Inc.	532
Ranyan, Inc.	530
Relational Software Inc.	417
Relational Technology Inc.	317
Reliance Electric Company	630
Ross Systems, Inc.	631
S & H Computer Systems, Inc.	201
SPSS Inc.	115
Saturn Systems, Inc.	502
Signal Technology, Inc.	118
C. D. Smith & Assoc., Inc.	329
Softool Corporation	230
Software AG	322
Software House	206
Software International	604
Software Results Corp.	626-628
Synemation Div. of Synemed	606
System Industries, Inc.	818
Tektronix, Inc.	522
Televideo	1012
Teltone	2012
The 11 Connexion	128
Tymshare/Computer Systems & Support	812
Tymshare/Dynasty	812
U. S. Design Corp.	102
Viking Software Services	531
Virtual Microsystems, Inc.	718
Webster Electronics, Inc.	805
Western Data	127
Western Peripherals	122
Western Resource Technology, Inc.	429
Why Systems, Inc.	521
Wyle Electronic Mktg. Group	922
Xerox Computer Services	506
Xylogics, Inc.	212
Xyplex, Inc.	826
Za-Tro Corp.	621
Zia Corp.	608

"BRING YOUR BOSS FOR FREE" ADMISSIONS POLICY ANNOUNCED FOR DEXPO/WEST 82; MORE THAN 200 BOOTHS ALREADY TAKEN FOR 2ND DEC-COMPATIBLE EXPOSITION

Anaheim, CA — A new "Bring Your Boss for Free" admissions policy makes it possible for data processing managers, top corporate executives and senior technical staff members at companies using Digital Equipment Corp. computers to register for DEXPO/West 82 at reduced prices ... when they come in pairs. DEXPO/West 82, the Second National DEC-Compatible Industry Exposition, will be held at the Anaheim Convention Center, Anaheim, CA, December 7-9, 1982.

The announcement was made today by Larry Hollander, president of Expoconsul International, Inc., organizer of the DEXPO Shows. "We hope to encourage greater communication among all levels of management at DEC-using companies," Hollander explained. "Because DEXPO features all the latest in DEC-compatibles, and *only* DEC-compatibles, the Show gives DEC-users an excellent opportunity to explore maximum potential of their computer systems, — and then make fully informed decisions." Visitors from the same company will be required to register in pairs and present business cards as documentation in order to take advantage of the program. Both registrants will then pay only half the usual at-door registration fee — that is, two for the price of one.

DEXPO/West 82 will be the largest DEC-compatible show ever — twice the size of the first DEXPO held last May. The first Show featured 125 vendors occupying 150 booths. DEXPO/West will have 250 vendors in 300 booths. For additional information, contact Larry Hollander, president, Expoconsul Internatioal, Inc., Cranbury, NJ 08512; Call (609) 799-1661.

TALKS ON DEC-COMPATIBLE SOFTWARE DOMINATE SCHEDULE OF PRODUCT FORUMS TO BE PRESENTED AT DEXPO WEST 82

Anaheim CA — Reflecting the rapid pace of technical advances in DEC—compatible software, discussions of these advances dominate the Product Forums scheduled for DEXPO WEST 82. While the wide-ranging program affords ample coverage of hardware topics as well, talks on software productivity, data analysis programs and relational data base management systems typify the software orientation of the Product Forums as a whole. DEXPO WEST 82 — The Second National DEC-Compatible Industry Exposition — will be held at the Anaheim Convention Center, Anaheim, CA, DECEMBER 7-9, 1982.

"By attending a Product Forum, in the space of only twenty minutes, executives who own, manage or use DEC computers will discover exactly what the latest DEC-compatible technology has to offer," said Larry Hollander. Hollander is president of Expoconsul International, Inc., organizers of the DEXPO Shows: DEXPO WEST 82; DEXPO EAST 83; and DEXPO EUROPE 83. "And since the presentations will be made by exhibiting DEC-compatible vendors," he continued, "visitors will have the opportunity to follow up on sessions of greatest interest to them by contacting the speakers right in their exhibit booths."

Among the exhibits, visitors will find 250 vendors demonstrating thousands of DEC-compatibles, including hardware, software and related services and supplies. The Product Forums, which complement what will be the world's largest exhibition of DEC-compatible products and services ever held, are free to all Show registrants. Approximately 60 Product Forum presentations will be made during the three-day event.

For additional information, contact Larry Hollander, president, Expoconsul International, Inc., 19 Yeger Road, Cranbury, NJ 08512; call (609) 799-1661.

DEC-COMPATIBLES AT DEXPO WEST 82 TO INCLUDE FULL RANGE OF HARDWARE, SOFTWARE & SERVICES: MANY ON EXHIBIT FOR FIRST TIME

Anaheim CA — In preparation for what will be the world's largest exhibition of DEC-compatible products and services ever held, approximately 250 vendors are readying their latest offerings for exhibit at DEXPO WEST 82 — The Second National DEC-Compatible Industry Exposition. And an informal survey of exhibitor plans reveals that visitors will find the entire range of DEC-compatible hardware, software and related services and supplies well represented at the December 7-9 event, set for the Anaheim Convention Center, Anaheim, CA.

"Literally thousands of DEC-compatibles will be demonstrated," announced Larry Hollander, president of Expoconsul International, Inc., "including hundreds that have never been exhibited before." As a notable example, Hollander cited the totally new category of DEC-compatibles spawned by the introduction of DEC's personal computers. Expoconsul International, Inc., is the organizer of DEXPO WEST 82 as well as the other DEXPO Shows — DEXPO EAST 83 and DEXPO EUROPE 83.

"The fact is, most owners, managers and users of DEC computers are not aware of all the technically advanced DEC-compatible resources on the market," Hollander explained. "Now for the first time, they'll have access to the technologies that promise to make their DEC systems even better." DEXPO WEST 82 is expected to attract 8,000 visitors from every management level — top management, DP management, senior technical staff, administrative management and other members of the decision-making team.

Additional information is available by contacting Larry Hollander, president, Expoconsul International, Inc., 19 Yeger Road, Cranbury, NJ 08512; call (609) 799-1661.

"MRP PLUS" TO BE INTRODUCED AT DEXPO/WEST 82

Belmont MA — Interactive Management Systems, Belmont, MA, will introduce "MRP PLUS": An Innovative Approach to MRP II" at the Second Annual DEC-Compatible Exposition, Anaheim Convention Center, December 7-9, 1982.

The product of the total integration of IMS' twelve accounting and manufacturing software packages, "MRP PLUS" is designed to give the executive and data processing professional the tools he or she needs for successful Manufacturing Resource Planning (MRP II).

MRP II, as defined by Oliver Wight in *The Executive's Guide to Successful MRP II* (Oliver Wight Limited Publications/Prentice-Hall, 1982) is, "a company game plan for manufacturing, marketing, engineering, and finance." The objective of the plan is to ensure that all divisions of a company are working as a team for the betterment of overall corporate productivity and growth.

"MRP PLUS" not only meets the objectives of MRP II, but helps the executive to break through specific corporate bottlenecks and to achieve critical goals such as reduced inventory and obsolescence, improved customer service and accountability, and decreased purchasing costs and overtime.

The twelve IMS software packages integrated within "MRP PLUS" are: Accounts Payable, Accounts Receivable, General Ledger, Financial Planning, Inventory Management and Control, Bill of Materials, Shop Floor Control, Materials Requirement Planning, Order Processing and Sales Analysis, Payroll, Fixed Assets, and Purchasing.

IMS' native mode software is written in VAX-11 BASIC, utilizing VAX-11 RMS, making it compatible with DEC's Datatrieve query language.

For more information on "MRP PLUS", write or call Mike Carabetta, IMS V.P. Marketing, 375 Concord Avenue, Belmont, Massachusetts 02178-3094, (617) 489-3550.

COMPUTER SYSTEMS CORPORATION ANNOUNCES VERSION 1.1 OF CALC-11

Indianapolis IN — CALC-11, Version 1.1 of the electronic spreadsheet system, is now available for the PDP-11 and VAX computers. Computer Systems Corporation will be demonstrating CALC-11 at the Second National DEC-Compatible Exposition, Anaheim Convention Center, December 7-9, 1982.

The CALC-11 product is one in a series of products from the Software Products Division of Computer Systems Corp. CALC-11 has been designed to operate on all of the popular PDP-11 operating systems (RT-11, RT-11 with TSX plus, RSX-11, RSTS/E) as well as the VMS operating system for the VAX computer. CALC-11 will be available for the Unix operating system in the near future.

CALC-11 incorporates some exciting features not generally available on previous spreadsheet systems. In addition to providing many of the standard features, CALC-11 allows for data access to non CALC-11 files, file encryption, three dimensional spreadsheet (paging), multiple nested function capability, very flexible spreadsheet adjustment and various data formating options.

Distribution and media is supplied with each purchase of CALC-11. The user manual incorporates both the tutorial section and the user reference section. The first time user of CALC-11 need only follow the tutorials in order to become proficient with this utility. The installation section of CALC-11 is very easy and requires less than a page of instruction.

CALC-11 licenses are priced according to which particular operating system is required. CALC-11 pricing starts at $2000.00. Educational and volume discounts are available. For additional information, contact: Computer Systems Corporation, 5540 Rock Hampton CT., Indianapolis, IN 46268, Attn: David Tortora, Phone (317) 872-7200, Telex 27-6243.

NYPLAN 3.0 — AN IMPORTANT NEW RELEASE OF THE NYPLAN FINANCIAL MODELING SYSTEM TO BE EXHIBITED FOR THE FIRST TIME AT DEXPO WEST 1982

Kirkland WA — Sophisticated business graphics which can be output to a wide variety of plotting devices, a matrix window with split screen, logic window with editor, visible calculation, help command, and goal seeking capabilities are a few of the exciting new features potential users will be able to try out at the 2nd National DEC Compatible Exposition, Anaheim Convention Center, Dec. 7-9, 1982. The NYPLAN system is presently available on RT-11, TSX+, RSX, IAS, RSTS, VAX/VMS, and TOPS 10/20 operating systems. Information available from NYPLAN INC., 135 Lake St., So. Kirkland, WA 98033, (206) 822-6074.

XYLOGICS TO DISPLAY COMPLETE LINE AT DEXPO WEST

Burlington MA — XYLOGICS will have their complete line of Peripheral Processors on display. These controllers are emulators interfacing disk and tape subsystems to Digital Equipment Corp.'s Q-Bus and Unibus; Data General's I/O Bus; and Intel/National's Multibus computer systems.

Software Saver Systems will be displayed in XYLOGICS "XL" series of packaged products. Utilizing Digital Equipment Corp.'s LSI-11/23 microprocessor and software products, the "XL" series, is a complete desk top packaged system that contains 22 bit addressing, up to 4MB of memory with RSX-11M, Q-Bus backplane, 34MB Winchester disk, 17MB tape cartridge, and 8 prewired RS232 ports.

XYLOGICS new Storage Cells which provide new data storage solutions for PDP-11, LSI-11, VAX, Nova/Eclipse, and Multibus users will also be exhibited. The XSC100 disk and streamer tape unit provides up to 320MB of on-line storage capacity for PDP-11 and VAX users. The XSC200 series provides up to 140MB of storage capacity in a 5¼" × 19" cabinet for Q-Bus, Unibus, Multibus, and DG I/O Bus systems.

LOW COST, POWER-LOSS DATA RETENTION FOR LSI-11 AT ADAC DEXPO/WEST BOOTH 427

Woburn, MA — ADAC's Model 1816CMOS battery-backed RAM memory boards offer up to 30-day retention of 8 or 16 K word units of system data in the event of power outage or shutdown. Available in 168-hour and 30 day models for either 8K or 16K words, these half-quad boards plug directly into a single slot of the LIS-11 backplane.

On-board automatic switching and recharging circuitry keep the self-contained nickel cadmium batteries charged to peak efficiency. System data will be maintained at temperatures as high as 40°C for 30 days or longer. Full operating temperature range is 0° to 55°C.

Model 1816CMOS features write protect dip switches that are accessible while the memory board is positioned in the backplane.

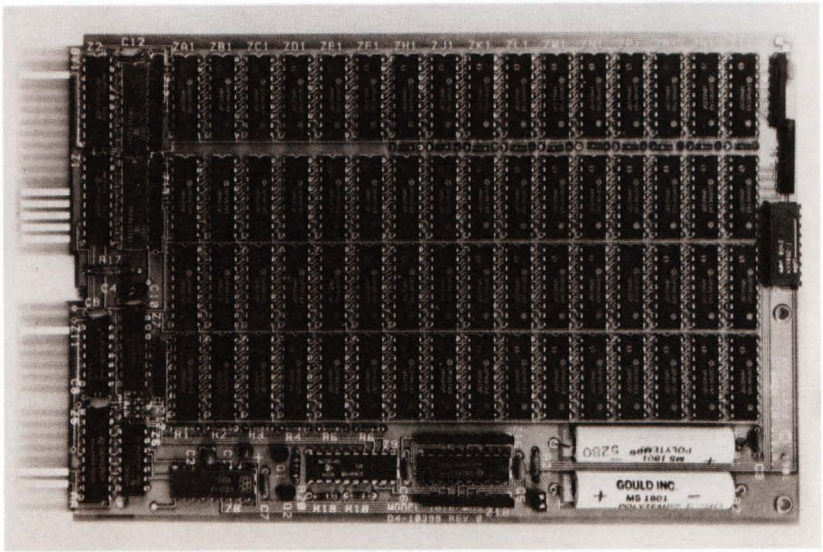
Pricing for the 1816CMOS series is highly competitive with extremely low "per word" storage cost. Unit prices are:

Model 1816CMOS-8S (8K, 168 hour retention)	$695.00
-8L (8K, 30 day retention)	$795.00
-16S (16K, 168 hour retention)	$1095.00
-16L (16K, 30 day retention)	$1295.00

For complete details and for special quantity pricing information, contact: Mr. Ben Minsk, ADAC Corp., 70 Tower Office Park, Woburn, MA 01801; Phone: (617) 935-6668; Telex: 949329.

DISC'S DBL AT DEXPO

Sacramento, CA — DBL is a business-oriented programming language which provides users of PDP-11 and VAX computers with portability, increased programmer productivity, and efficiency of operation. DBL is a much extended superset of DEC's DIBOL-11 but can accept virtually all exisiting DIBOL-11 sources unmodified. The extensions in DBL include structured programming facilities, source library "include" capabilities, self-contained ISAM file structures, auto/load shared runtime support, and intermodule "global" data storage features. The DBL Symbolic Debugger permits immediate debugging of DBL programs without special advance preparations. The DBL ISAM facility is tree-structured and self-reorganizing. Reorganizing includes reclamation of unused index and data entries thus



ADAC - Model 1816CMOS

greatly minimizing the need for frequent file reorganization. Portability of DBL programs is of paramount concern. Currently spanning all of DEC's major operating systems, DBL is now being developed for non-DEC environments so as to provide the widest possible choice of hardware and systems for the DBL users and distributors.

This product will be on exhibit for the first time by DISC at DEXPO WEST 82.

"TWICE-THE-LIFE", 8" DUAL-SIDES, DUAL-DENSITY FLOPPY DISKS FOR DEC'S Q-BUS*

Berkeley, CA — The FD4 Two-Megabyte Floppy Disk System from Synemed, Inc., offers twice the MTBF, twice the media life of other systems and a happy solution to the threat of critical data losses!

Based on NEC's reliable Soft-Touch** drive, this competitively priced dual-sided, dual density disk system provides Q-Bus* users with a superlong MTBF of 15,000 hours (compared to an average of 8,000 for other drives) — and the FD4 is the *only* system with air-cushion protection to extend media life. A unique air dampener at the base of the head-loading mechanism eliminates the pinching, or biting of disk surfaces that leads to the excessive wear and information dropout common to other double-sided floppy disk systems. Now, even in those applications where data loss is most critical, media wear and it's associated costs can be cut dramatically without jeopardizing data integrity.

Pluggable in half-quad slot and completely transparent to DEC RX02 software, the FD4 runs DEC's ZRXDBO diagnostic and looks like a double size RX02 with 1962 formatted blocks per disk.

Please direct inquiries to:
DEXPO/WEST 82 - BOOTH 606),
SYNEMED, INC., 1215 Fourth Street, Berkeley, CA 94710, Attn: Kevin Miller.
*Trademark Digital Equipment Corp.
**Trademark NEC Information Systems, Inc.

NIS, INC. AT BOOTH 432 WILL INTRODUCE PASCAL-PLUS AND PRES

Cupertino, CA — National Information Systems will be showing ACCENT R and VUE and introducing PASCAL-PLUS and PRES at the Second National DEC-Compatible Exhibition, DEXPO/WEST

ACCENT R is a high productivity development tool allowing computerized information management applications to be developed in a fraction of the normal time. ACCENT R's combination of a relational DBMS with a non-procedural query language and a 4th generation high-level structured programming language makes immediate and direct access to information a reality for strategic planning, decision support, and information management on DECsystem-10 and -20 computers.

VUE is an interactive project management system that provides a convenient tool to plan and manage complex projects. VUE performs critical paths analysis for projects with up to 3,000 activities. VUE performs very well in environments where multi on- and/or off-site users need program access. The menu-driven user interface allows easy data entry, modification, and report selection. VUE runs on DECsystem-10/20, HP3000, VAX, PDP-11, and Perkin Elmer. Reporting is done on CRT's, printers, or optional graphics plotters.

PASCAL-PLUS is a production oriented compiler for DEC-10 and DEC-20 users, developed for the serious Pascal implementor. PASCAL-PLUS extends standard features, removes many of the arbitrary restrictions appearing in standard PASCAL versions, and utilizes an advanced optimization design. New features, normally occurring in languages such as ADA, FORTRAN, and ALGOL, have been added to increase programming power and flexibility. The PASCAL-PLUS compiler produces code that is 30-40 percent more compact than most standard compiler versions, with a corresponding increase in processing speed.

PRES is an interactive personnel resource management system. PRES provides record keeping and record tracking for all employees, including such features as job history, benefits administration, compensation, and salary

history. Special government requirements, such as affirmative action reports, are met, as are regular corporate needs. PRES offers a powerful reporting function. This menu-driven system allows easy data entry, modification, and report selection.

Dave Bray (PRES), Joe Rizzo (VUE), and John Enyedy (ACCENT R, PASCAL-PLUS) will be representing National Information Systems at Booth 432, waiting to demonstrate the power of these products.

National Information Systems Inc., is headquartered at 20370 Town Center Lane, Cupertino, Calif. 95014.

A FIRST FOR THE U.S. SOFTWARE INDUSTRY (at DEXPO/WEST 82)— HINDITRON OFFSHORE SOFTWARE SERVICES

. . . is the first time exhibition of a service that offers cost effective options to the software industry that will help you get more from your computer for less.

HINDITRON's software development centre in Bombay, India, is geared to meet your specific requirements, using a wide range of computer and microprocessor systems.

HINDITRON offers software services for developing Application Software, System Software, Process Control Software and Software Conversions on turnkey and contract basis.

A multidisciplinary team of over one hundred software professionals has provided such services satisfactorily to DEC and DEC OEMs in the Asian, African and Middle-East regions.

Now for the first time these services are offered to the U.S. software industry.

For more information on HINDITRON OFFSHORE SOFTWARE SERVICES contact Harish Mehta, Director, HINDITRON COMPUTER SYSTEMS & CONSULTANTS PRIVATE LIMITED, Eros Bldg., 5th Floor, 42, M. K. Road, Bombay 400 020, India, Tel. No. 22-09-20.

RSTS/E NETWORKING SOFTWARE TO BE DEMONSTRATED AT DEXPO/WEST

Toronto, Canada — Digital Management Group Ltd. will demonstrate DMG/NET — networking software for RSTS/E users — at DEXPO/WEST.

DMG, a DEC-oriented software and consulting company, developed DMG/NET as an effective and economical solution to today's complex networking problems. DMG/NET provides a networking growth path as the user's needs grow, since network carriers from direct cable links through leased lines and auto-dial units to X.25 networks are supported in any combination. Thus a network that starts out as a simple 2 system link can grow to a multi system, multi-carrier network as and when required.

"DMG/NET evolved out of a project for a client," says John Dightam, President of DMG. "Their need was for a communication network that would provide low cost interactive access from any of their head office terminals to a wide variety of DEC and non-DEC computers throughout Canada and the United States. They also needed low cost file transfer capability between their seven RSTS/E sites across Canada. We were asked to provide all this, with the mandatory specification that no modifications be made to either the standard RSTS/E system or their PDP-11 hardware, as supported by DEC."

DMG/NET provides reliable, trouble-free networking and takes advantage of the cost savings offered by X.25 networks, especially over long distances. DMG/NET interfaces with Tymnet, Telenet and overseas X.25 networks. Some users have found that they have cut their communication costs by as much as 90%.

DMG/NET provides two-way file transfer and interactive dialogue capabilities between a local RSTS/E host and other RSTS/E systems. Interactive communication with IBM and other non RSTS/E systems is also possible, and file data can be transferred from non-RSTS to RSTS by use of the "log" option contained in DMG/NET. All this is accomplished using short, English language commands, which can be easily understood by non-computer personnel.

The "alternate path" capability of DMG/NET allows all available alternatives for network connection to be explored, including all outgoing ports, remote access numbers and different carriers, if these facilities are available on the computer. If a connection cannot be completed because all lines are in use, or because of a fault in the network or communications unit, DMB/NET will describe the situation in an easily understood message displayed on the user's screen. When a network problem does occur, DMG/NET diagnostic routines help pinpoint the problem quickly. Comprehensive usage statistics recorded by the sytem facilitate management of day-to-day traffic on the network.

Through table-driven software, DMG/NET provides total interface and network flexibility. For example, as new micro-computers become available and are interfaced with the RSTS/E host, DMG/NET can handle the dialogue requirement between the two central processors.

DMG/NET is available on either a perpetual license or a rental basis — both with comprehensive support plans. "We are really committed to support," says DMG President Dightam, "and we have designed DMG/NET so that it lends itself to remote diagnosis. For example, from our Toronto office, we can reconfigure, diagnose faults, and load new versions into customer computers anywhere in the world — just as if they were in the next office."

DMG/NET will be exhibited by DMG in **Booth No. 125**, at DEXPO/WEST '82.

For more information on DMG/NET, contact Ken Allsopp, Digital Management Group Ltd., 4800 Yonge Street, Willowdale, Ontario, Canada M2N 6G5, Telephone: (416) 225-7788.

RSTS/E, PDP-11 and DEC are registered trademarks of Digital Equipment Corporation. DMG/NET is a registered trademark of Digital Management Group Ltd.

BRITTON-LEE INTRODUCES RELATIONAL DBMS FOR DEC VAX/VMS

Los Gatos CA — Britton-Lee, Inc. announces the introduction of the SYSTEM 300 and SYSTEM 600 relational database management systems for DEC computer systems. These products are for use with the VAX-11/750 and VAX-11/780 computers that use the VMS operating system. Additionally, versions of the SYSTEM 300/600 are available for use with PDP-11 and VAX-11 computers running the UNIX operating system.

The heart of the SYSTEM 300/600 is the Britton-Lee Intelligent Database Machine (IDM). The IDM combines relational database management software with hardware designed specifically to perform database management at high speeds. Located between computer and disks, the IDM offloads the entire database management function from the computer.

The low-priced SYSTEM 300 is ideally suited for medium demand VAX applications. The SYSTEM 600 is a higher performance product, making it an ideal match for high demand applications with large databases and many users. The SYSTEM 300 can control up to four SMD compatible disk drives for databases of up to 2.7 billion bytes. The SYSTEM 600 controls up to sixteen disk drives for a maximum database size of 10.8 billion bytes. The SYSTEM 300/600 attaches to the VAX computer via a high speed IEEE-488 parallel interface.

The SYSTEM 300/600 includes an IDM (mounted in a 40 inch cabinet), hardware to interface directly to the VAX UNIBUS adapter, and Support Software that runs under the VMS operating system. Also included is one year of software maintenance, which includes software updates and phone-in consulting. Additional software maintenance and hardware service contracts are available. Installation is available separately.

Also included with the SYSTEM 300/600 is the Britton-Lee IDL Query

Language. This user-friendly query language allows unskilled users to easily access and modify SYSTEM 300/600 databases. Included for application programming, is a runtime subroutine library that allows VAX-1 FORTRAN and COBOL programs, and "C" programs to access SYSTEM 300/60 databases. Optional VMS precompilers allow IDL commands to be embedded in VAX-11 FORTRAN and COBOL programs, and "C" programs. Database administration utilities, which allow for backup, crash recovery and bulk loading of data, are included in the standard product.

A unique feature of the SYSTEM 300/600 is that it can be connected to several computers simultaneously. This allows multiple VAX computers to use it as a centralized database resource. The optional SYSTEM 300/600 Interface Package provides this feature.

The Britton-Lee SYSTEM 300 and SYSTEM 600 can be ordered beginning June 1, 1982. The SYSTEM 300 is priced starting at $62,900 and the SYSTEM 600 starts at $85,300. The SYSTEM 300/600 Interface Package that allows the System 300/600 to be connected to additional VAX computers is available for $15,800. Pricing for the UNIX versions of the SYSTEM 300/600 is the same.

A public demonstration of the SYSTEM 300/600 will be held at DEXPO West.

For more information: Phil Amend, Vice-President, Marketing, Britton-Lee, Inc. Kathy Shadley, Marketing Administrator, Britton-Lee, Inc. 408/378-7000.

Intelligent Database Machine, IDM, and IDL are registered trademarks of Britton-Lee, Inc.

DEC, VAX, VMS, PDP and UNIBUS are registered trademarks of Digital Equipment Corporation.

CAMBEX AT DEXPO IN DECEMBER

Waltham MA — Cambex Corporation will be announcing new memory products for the DEC VAX-11/780* and VAX-11/750* series of processors at "The Second National DEC-Compatible Exposition (DEXPO)".

Cambex has been a supplier of OEM memories for over a decade and will exhibit its UNIBUS and Q-Bus compatible memories along with the new VAX products. Additionally, there will be an add-on version of their semiconductor RK05* emulator disk.

All of the products have gained wide acceptance in varied applications areas and across the PDP-11* and LSI-11* families. Many users have boasted of significant increases in performance, since the products are utilizing 64K dynamic RAMS and the highest quality components and manufacturing facilities available.

The Cambex Booth is #232. Any inquiries can be made to Mini/Micro Marketing Department (617) 890-6000.
*Registered Trademarks of Digital Equipment Corporation.

COMPLETE DATA ANALYSIS SYSTEM FOR VAX ANNOUNCED

Nashville TN — S&H Computer Systems, Inc. has introduced INDAS™ (Integrated Data Analysis System), the first complete data analysis system for Digital Equipment Corporation VAX/VMS systems.

What Is INDAS

INDAS™ (Integrated Data Analysis System) is a complete data analysis system for Digital Equipment Corporation VAX/VMS systems. In one integrated system INDAS™ provides (1) databases facilities (including the ability to sort, merge, update and join databases); (2) a complete programming language; (3) a matrix manipulation language; (4) powerful statistical analysis procedures including the General Linear Model.

Although a number of products are available for the VAX that provide programming languages or database access or statistical analysis, INDAS™ provides all of these features in a single integrated system. Using INDAS™, the data analyst performs the complete analysis from the raw data to the final printed results within the INDAS™ system.

The database, programming language and plotting and charting facilities make INDAS™ a flexible system well suited to a broad range of applications in science and business.

INDAS™ Database

INDAS™ features relational database capabilities. The MERGE statement implements the relational join operator, linking data from several data sets by a common key. The UP-DATE statement applies transaction records to a master file. The IF, KEEP and DROP statements implement subsetting by records and variables. Taken together, these statements allow users to maintain and extract data with a minimum of programming.

INDAS™ Programming Language

INDAS™ provides a complete programming language that includes structured programming facilities, subprograms, numeric and character data types, matrices, as well as statements for selecting, merging, dating and joining data sets. Because the language compiler is part of the INDAS™ system, it is not necessary to compile and link programs separately.

A powerful set of intrinsic functions is part of the programming language. In addition to the usual mathematical and transcendental functions, INDAS™ provides functions to perform operations such as computing probability functions, character string manipulation, and state name and zip code conversion.

Operations on character strings include the ability to extract or alter substrings, determination of the length of a string, concatenation of two or more strings, location of one string within another and removal of selected characters. INDAS™ also provides character-string matrices.

INDAS™1 Matrix Manipulation Language

In addition to a conventional programming language, INDAS™ provides a complete matrix manipulation language. This facility is similar in power to APL but does not require a special terminal. The matrix manipulation language allows matrix variables, constants and functions. The allocation of memory space for matrices is completely dynamic and automatic. Arithmetic expressions involving matrices can be written as simply and directly as those involving scalars.

For example, if X is a matrix variable, the following statement computes the inverse of the matrix resulting from the product of the transpose of X with X:

Y = INV (X'* X)

Matrix operators available in the language include: addition, subtraction, matrix multiplication, scalar multiplication, transpose, horizontal concatenation, vertical concatenation, comparison and summation over selected rows and columns. Generalized subscripts allow access to single matrix elements or sections of the matrix. Intrinsic matrix functions include: inverse, generalized inverse (Moore-Penrose), determinant, singular value decomposition, solution of linear equations, sweep operator, transcendental functions and probability funtions.

INDAS™ General Linear Model

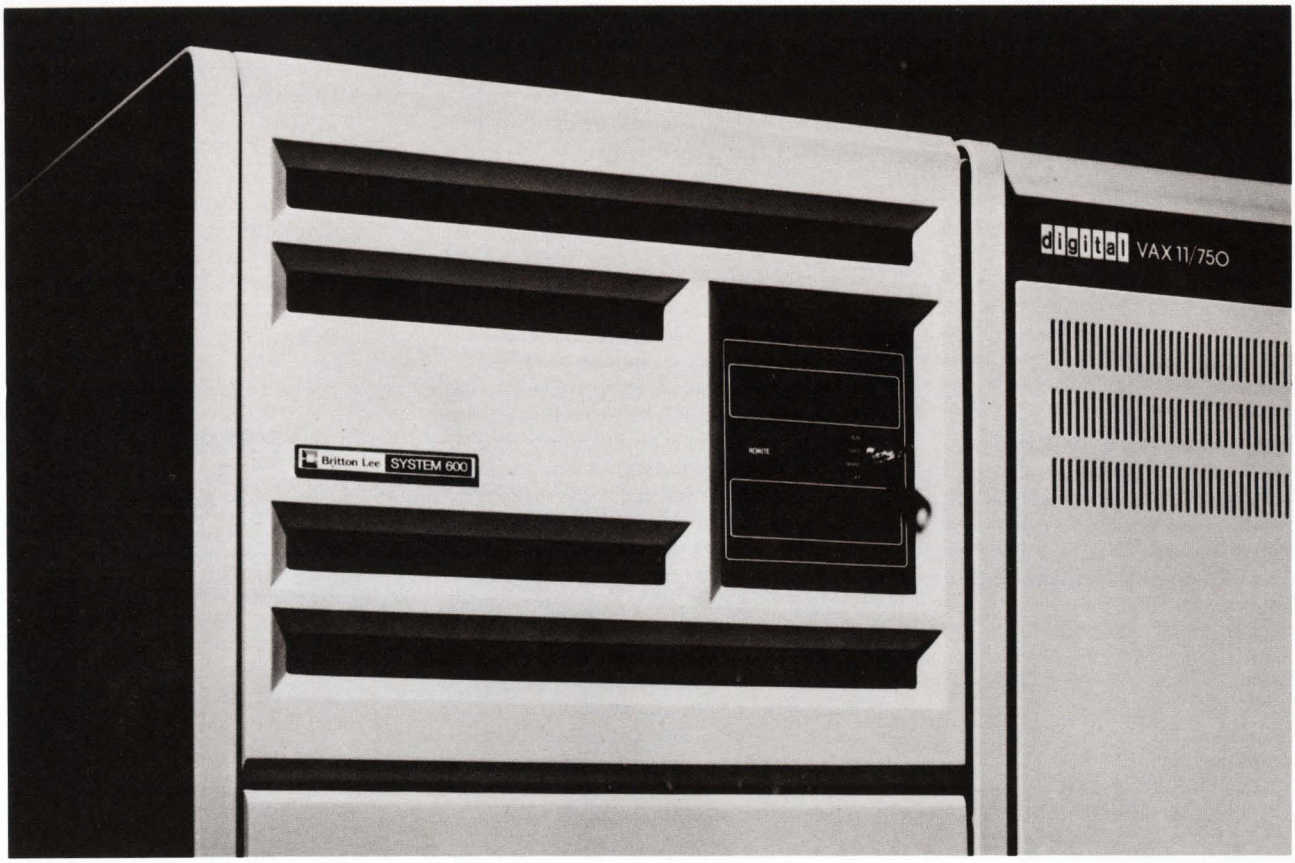
The General Linear Model procedure analyzes data within the framework of an arbitrary linear statistical model. This procedure is exceptionally comprehensive, encompassing in a unified setting:

- Analysis of Variance (ANOVA)
- Regression
- Analysis of Covariance (COANOVA)
- Multivariate Analysis of Variance (MANOVA)

The syntax is simple yet powerful, paralleling the notation used to describe linear models. A particular strength of the General Linear Model is its ability to accept unbalanced data, which allows analysis of the type of data frequently encountered in applications but which cannot be analyzed with traditional methods.

INDAS™ Statistical Procedures

INDAS™ has a comprehensive set of procedures for statistical analysis. This



THE NEW BRITTON-LEE SYSTEM 300 AND SYSTEM 600 (SYSTEM 600 shown) BRING HIGH—PERFORMANCE RELATIONAL DATABASE MANAGEMENT TO DEC VAX COMPUTER USERS.

includes descriptive and univariate statistical analysis; analysis of variance for balanced and unbalanced designs; non-parametric analysis of variance; t-tests; probit analysis; linear models for catagorical data; multiple, stepwise and non-linear regression; autoregression; correlation (Pearson, Spearman and Kendall), canonical correlation; discriminant analysis; cluster analysis; and factor analysis. Graphics facilities include procedures for generating curve and contour plots, bar charts, pie charts, star charts and block charts.

INDAS™ is Extensible

INDAS™ users can write procedures to be incorporated the the INDAS™ system and called just as the standard INDAS™ procedures. This allows a flexible and convenient way to develop specialized analyses. All of the database and programming features of INDAS™ can provide the input to a user-written procedure.

INDAS™ Is Another Quality Product From S&H

S&H Computer Systems, Inc. has been developing high quality system software for Digital Equipment Corporation computers since 1976. In addition to INDAS™, S&H has developed the popular TSX-Plus™ operating system, the high-performance COBOL-Plus™ compiler and the versatile RTSORT™ disk sort program. S&H products are in use at over 3,000 sites in more than 25 countries. S&H plans to deliver the first INDAS™ systems early in 1983.

Please address domestic sales inquiries to Gary Manookian, and all other inquiries to Richard Dohrmann, Vice President, at S&H Computer Systems, Inc. 1027 17th Avenue S., Nashville, Tennessee 37212, (615) 327-3670, Telex 786577 S&H NAS.

ADVANCED SOFTWARE PRODUCTS announces QSORT Version 3.0

Lathrup Village, MI — Advanced Software Products has developed a sort utility, QSORT, that uses the operating system's efficiency — not the programmer's — to sort large quantities of data and merge unsorted files. QSORT Version 3.1 with expanded file sorting capacity, will be released for the first time at DEXPO/WEST 82. Meanwhile, QSORT, Version 3.0, to be released October 1st, 1982, now lets programmers get on with the business of programming systems instead of programming sorts.

Performance

QSORT is a high performance sort utility developed for the PDP-11 computers utilizing CTS-500/RSTS-E operating systems. Written completely in PDP-11 MACRO enables QSORT to take full advantage of the powerful features of the operating system. Programmers no longer have to build, translate and compile a sort file; QSORT allows the programmer to describe the sort with a SINGLE command line.

Efficiency

QSORT has been developed and installed over a two year period in institutions where large volumes of data are sorted for daily and monthly reporting. To enhance speed performance, QSORT was developed as a run-time system utility-only the data swaps in and out, not the program. The program will only swap in-never out. This is due to the fact that the code is fixed and need not be saved. QSORT is re-entrant; only one copy of the program is ever loaded into memory no matter how many users are concurrently utilizing QSORT.

Generally, QSORT requires 1.5 times the input file for work space. This space can be allocated to any random access device supported by the system it is running on.

Enhanced

In addition to high performance and efficiency, the three major features of the newest version of QSORT (Version 3.0) gives QSORT file sorting capacity that supports 2^{16} blocks of data with any number of records. QSORT is also capable of merging two *unsorted* files into a single sorted file. And, QSORT can handle fixed length, no-span, unformatted ASCII files as typically used by BASIC-PLUS programmers.

Features

QSORT is flexible and may be called by any language which supports "core common". Upon successful completion, QSORT can be chained to execute another program.

Enhanced speed performance, full RSX-11 compatability, and auto attribute recognition for RMS-11 sequential files round out the additional enhancements of QSORT Version 3.0.

An optional SORTG/GSORT processor allows QSORT to be substituted for most DIBOL or MCBA sorts without any user program modifications.

Availability

QSORT Version 3.0 is available for $1,200.00. Marketed exclusively by ADVANCED SOFTWARE PRODUCTS, further information, including a 7-day trial DEMO, may be obtained by calling or writing: Advanced Software Products, 28690 Southfield Rd., Suite 291, Lathrup Village, MI 48076 (313) 569-5570.

SOFTOOL² CHANGE AND CONFIGURATION CONTROL (CCC™)

Goleta CA — SOFTOOL CORPORATION is pleased to announce the release of its change and configuration control (CCC™) environment, a new product to be exhibited for the first time at DEXPO™/WEST 82.

For a long time, software managers and programmers alike have been waiting for a software tool that would allow them to automatically keep track of all software and documentation changes and versions. CCC™ fulfills this need in a most impressive manner. CCC™ can reconstruct software and documents upon demand!

CCC™ is a comprehensive system for the management of software changes and configuration control. It is an interactive tool with an easy-to-use interface. It supports, in addition to automatic program reconstruction, impact of change reports, access control, archiving, compression, encryption, automatic backup and recovery, as well as other powerful features. It includes an editor, and provides interactive tutorials that facilitate training. CCC™ can handle programs written in any programming language.

CCC™ is part of SOFTOOL², which is an integrated set of tools for software management, development, and maintenance.

CCC™ is available, off-the-shelf, for the DEC VAX computer.

A permanent license for CCC™, including one year maintenance, costs $30,000. Various lease plans are also available.

SOFTOOL² products can be seen at **Booth 230** at the Dexpo™/West 82 Convention.

ZIA AT BOOTH 604

Morris Plains, NJ — Zia Corporation, a vendor of DEC compatible utility software, announced today that the following products will be exhibited in Booth 604 at DEXPO 82.

Zia Back/Rest . . . a full incremental backup and restore utility for users of RT-11, TSX and TSX Plus. Operates on all supported random access devices and handles multi-volume backups automatically.

Zia RIFLE . . . designed to facilitate the transfer of files between two PDP-11 computers, hard wired or through dial-up lines, using the RT-11 operating system. Files transferred may be of any type, including source, data, object, or binary.

Zia Virtual Terminal . . . Permits RT-11, TSX or TSX plus users to establish a connection between his local computer and nearly any remote time-sharing system. The user then may interact with the remote computer using his local terminal or transmit files in either direction. A simple control language offers many options including transmission speed control and automatic call placement and termination.

For further information please contact Glenn Burnett at (201) 328-7177.

COMPATIBLE HARDWARE & SOFTWARE FOR DEC PERSONAL COMPUTERS AMONG NEW PRODUCTS TO BE FEATURED AT DEXPO WEST 82, ANAHEIM, DEC. 7-9

Anaheim CA — Although dozens of vendors are planning major DEC-compatible product introductions as part of their exhibits in DEXPO WEST 82, it appears that the introductions that will generate the most excitement and industry-wide interest will be the products designed to be used with Digital Equipment Corporation's personal computers.

"Right now, the biggest customers for DEC's personal computers are the people already using larger DEC machines, and they're eager to discover the full range of hardware and software available for the personal computers," stated Larry Hollander, president of Expoconsul International, Inc. "Since every single one of the 8,000 or so executives attending DEXPO West 82 will come from companies now using DEC systems, the Show will be the focal point of their personal computer buying decisions.

To accommodate this interest in the personal computer compatibles, exhibitors offering these products will be identified by a special sign. "We are taking every step to help the DEC-compatible industry communicate and prosper," Hollander added. DEXPO WEST 82 — The Second National DEC-Compatible Industry Exposition — will be held at the Anaheim Convention Center, Anaheim, CA, December 7-9, 1982.

The Show was recently expanded to make room for 250 exhibiting companies that will occupy 300 exhibit booths. Twice the size of DEXPO 82 held last May, DEXPO WEST 82 will be the world's largest, most comprehensive exhibition of DEC-compatibles in history.

For additional information, contact Larry Hollander, president, Expoconsul International, Inc., 19 Yeger Road, Cranbury, NJ 08512; call (609) 799-1661.

PRODUCT NOTES & UPDATES

SOUTHERN SYSTEMS ANNOUNCES PLAN TO BEGIN OEM SALES OF INTERFACES

Fort Lauderdale, FL — Southern Systems, Inc. (SSI), add-on printer company, has announced the start of a new OEM sales program for SSI-developed and manufactured computer-printer interfaces.

The OEM program of interface sales will be an addition to the company's servicing of end-users with complete line printer systems using the newest technologies in speed ranges of 200 to 1600 lines per minute.

In the interface sales program, OEMs, system houses, distributors and other volume users will receive quantity purchase prices that range from $200 to $2,000 depending on the type of computer interfacing required.

Southern Systems designs and manufacturers line printer interfaces and controllers for all Digital Equipment Corporation computers as well as for Data General, Interdata, Texas Instruments, Hewlett-Packard, IBM and Burroughs. SSI also makes a wide range of serial communication interfaces, both synchronous and asynchronous, including those compatible with IBM protocols 2780/3780, Burroughs, Univac and NTR.

The company is located at 2841 Cypress Creek Road, Fort Lauderdale, FL 33309, (305) 979-1000.

WHY SYSTEMS RELEASES DECISION SUPPORT TOOL FOR DEC COMPUTERS

Redmond, WA — A new decision support system has been released by WHY Systems Inc. The DIGICALC™ package was designed for Digital Equipment Corporation computers, and is currently available on VAX using the VAX/VMS operating system and on PDP-11 systems running under RSTS/E or CTS-500.

DIGICALC is an interactive, CRT-oriented planning and forecasting tool used by accountants, engineers, managers, executives, scientists and

others who want to automate their number processing needs. Both simple and complex numeric calculations can be quickly and easily performed, giving the user more time to work on other matters. "Increased productivity is the key," said Wayne Yarnall, President of WHY Systems. "DIGICALC is designed to give key personnel the answers to complex problems quickly, so that they can make important decisions with the best possible data at their fingertips." Yarnall added, "An investment in DIGICALC gives your top people additional leverage, and it has a payback period that can be measured in days or weeks, not months or years."

The most common uses of DIGICALC are for budgeting and financial modeling. An accounting worksheet is displayed on the video screen. A simple model might consist of budget descriptions down the left-hand column, headings for twelve months across the top, with totals for each month, and cross totals for each budget category. Data can be entered directly onto the screen by the user, or it can come from an external source such as an existing General Ledger system on the same computer.

In a modeling session, any of the factors that make up the model can be changed, and the entire model can be immediately viewed from the new prospective. This "What if" capability is especially helpful to the business manager. The manager can say "What if sales drop 3.8 percent, and inflation increases 1.2 percent per month for the next six months." The new scenario will be displayed, and the results are available to continue the modeling process. Worksheets can be prepared for individual departments, and can be consolidated into divisional and organization-wide models. After all the interactive modeling is complete, DIGICALC will print reports that can be used throughout the organization.

The product can be used effectively in any environment with the proper operating system. DIGICALC is completely self-contained, and no extra-cost system hardware or software options are required in order to use it. Facilities for transfering data into and out of DIGICALC for use with other computer applications are provided.

WHY Systems, Inc. specializes in the development of software for DEC computer systems. Its offices are located at 16902 Redmond Way, Redmond, Wash. 98052; tel. (206) 881-2331.

DIGITAL PRODUCTS REPAIR CENTER BROCHURE FROM RELIANCE ELECTRIC

Cleveland, OH — A new four-page illustrated brochure describing how the Reliance Electric Digital Products Repair Center provides fast, reliable repair, exchange or replacement of PC Boards and related equipment is now available.

The file-size brochure points out that the Digital Products Repair Center is equipped to solve every possible kind of PC Board problem through the use of advanced state-of-the art equipment and instrument calibration procedures. PC Boards can be diagnosed and tested under simulated operating conditions, including heat and humidity environmental simulations, plus no-load and full-load testing.

Explained in the brochure, containing photos of troubleshooting and testing procedures, is that the Center's capabilities include repair of PC Boards from Reliance® and Toledo® equipment plus AutoMate® Programmable Controllers, computer and computer interface systems, and UDAC™ Programmable Process Controllers. Repair service is also furnished for computer and control equipment made by Digital Equipment Corp., Data General and GRI. Noted is that 10-day service is normal but emergency repairs can be made in 24 hours.

Also described is the Reliance Electric parts exchange program for selected AutoMate Programmable Controller components which gives users immediate repair service to minimize equipment downtime, and the AutoMate CRT Programmer Rental System for reducing programming costs.

To obtain a copy of the new Digital Produces Repair Center Brochure — H-2661-2 — write to Reliance Electric Company, 24701 Euclid Avenue, Cleveland, Ohio 44117.

DIGITAL EQUIPMENT TO MARKET INTERNATIONALLY RAXCO RABBIT SOFTWARE

Atlanta, GA — RAXCO Inc. announces an international marketing agreement with Digital Equipment. Under the terms of the agreement, Digital Equipment Australia Pty Limited will market and support RAXCOs "Digital tested" RABBIT Software on an exclusive basis throughout Australia and New Zealand. Negotiations to provide similar services throughout Europe are expected to be concluded shortly. RAXCO will continue to directly provide RABBIT Software throughout North America.

In the July 1982 issue of "digital news" Digital is quoted as follows: "Digital is pleased to announce a new program designed to make it easier for customers to obtain some of the Software written by third parties in Australia, New Zealand and Overseas.

"This worldwide program was established in response to the many requests made by customers to Digital, especially in countries such as Australia and New Zealand. Customers had expressed a concern at DECUS Meetings and in their Survey responses to Software Services, that it was expensive and time consuming to obtain Software Applications packages written by Software Houses and by major customers in Europe and the U.S.A. It was not easy for them to evaluate packages remotely or to negotiate an agreement for their purchase. They were also concerned about local support for these packages.

"In addition, many customers have developed software which they believe to be of a high enough quality to be marketed. They asked Digital to use its extensive marketing capabilities to make these Applications Packages available to Digital's customer base worldwide.

"The Digital Application Software Library has been setup by Digital to provide customers with access to packages developed locally and overseas by software houses and by customers."

RABBIT Software is available for DEC computers running VMS, RSTS/E and soon-to-be-announced RSXX-11m plus.

RABBIT-1 Software provides system resource accounting, auditing and billing reports.

RABBIT-2 Software provides graphic system performance analysis through an interactive command language.

RABBIT-3 Software is a job accounting and monitor program for RSTS/E systems.

RABBIT-4 Software is a computer security system for users and data files under RSTS/E.

RABBIT-5 Software is a high speed file backup and restore system for VMS.

Over 500 RABBIT Systems have been installed worldwide in the past two years, including England, Ireland, Germany, France, Australia, South Africa, Mexico, Canada and the U.S.A.

For more information contact: RAXCO Inc., Suite 200, 6520 Powers Ferry Road, Atlanta, Georgia 30339; U.S.A., Telephone: (404) 955-2553; TWX 810-766-2256; Telex 54-2659.

NEW PRODUCTS

CABLESHARE ANNOUNCES ADVANCED MARKET RESEARCH TOOL

London, Ontario — Cableshare, Inc. announces Surveyor, a unique approach to automated market research. Stand-alone units with both Telidon & Videodisc prompt consumers to answer questions by touching a pressure sensitive screen. This approach is faster, more accurate and can handle a greater volume of data cheaper than other interviewing techniques.

The full video with sound option will allow pretesting of T.V. commercials and new products. The Telidon images will match conventional questionnaires but

with full colour and computer-provided questions. Researchers can branch to different questions based on the consumer's response. Product discount coupons or vouchers can be automatically printed at the end of an interview as a gift for co-operation.

Cableshare Inc. is a public company which develops computer information systems, software products and Telidon business systems and markets them in Canada, the United States and abroad.

INNOVATIVE ELECTRONICS OFFERS IBM 3270 BSC CONVERSION

Miami, FL — Innovative Electronics, Inc. has announced the MC-80/600-01 communications processor. Emulating an IBM 3274-51C communications controller running configuration support level A, this device converts an inexpensive DEC VT-100 compatible terminal into a full function IBM 3277-1, 3277-2, 3278-1, 3278-2 terminal communicating with the IBM host using the EBCIDIC binary synchronous communications (BSC) protocol.

Economical asynchronous ASCII terminals such as the DEC VT-100 can be used in both local and dial-up applications. Full screen mapping is performed by the MC-80/600-01: data displayed on the asynchronous ASCII terminal will be the same as an IBM 3277/3278 display station, with virtual screen sizes of 480, 960, and 1920 characters. All screen formatting capabilities are supported. These include protected or unprotected alphanumeric, numeric, and non-display fields, character highlighting, and auto-tabbing. All keyboard functions are supported including vertical and horizontal cursor movement, erase, insert, delete 24 program function (PF) and program access (PA) keys. Status information is displayed in **English** at the bottom of the display.

Maintainabiity of the unit is dramatically increased by the extensive on and off line diagnostics. Diagnostic indications, provided through a seven segment display, include indications of the device being polled or selected and transmission errors. Off-line diagnostics are executed when the system is powered on, as well as when initiated by the operator. The CPU, ROM, RAM, and data paths are always tested. Operator initiated tests include the exercise of attached peripheral devices or "canned" messages to and from attached processors. A third level of testing is provided through test connectors, which connect outputs to inputs, completely testing every interface signal. These diagnostics enable the clerical operator to easily isolate failures to the commuincations processor, mode, or terminal device allowing the system vendor to reduce the cost to maintain the system through initial telephone assistance. This stand alone self powered micro-communications processor provides up to 16K of ROM, 16K of RAM, and two serial communications ports. The price for the basic unit is $1745.00. A model sharing port is optional. Delivery is from stock to 30 days. An additional port, allowing two independent terminal devices to be supported is $250.00 additional and will be available this fall.

NEW MDB MODEM CONTROL BOARD DOES ITS OWN TROUBLESHOOTING

Orange, CA — A new asynchronous serial interface with model control for LSI-11 based systems will be introduced by MDB Systems, Inc., the world's largest independent manufacturer of interface products. The interface is completely compatible with the DEC CLV11-E and, at the same time, offers a number of significant features which are not present on the DEC board. Notable among these is a troubleshooting capability.

The single line RS-232-C interface is designated the MLSI-DLV11-ED, and it is appropriate for a wide range of modem oriented applications. Among its unique features are device addressing, and interrupt vectors which are switch selectable for operating convenience. Data rates from 50 to 19.2K baud are both switch and program selectable. To assure complete functionability in contemporary systems, the MDB board has four level interrupt capability which is jumper selectable.

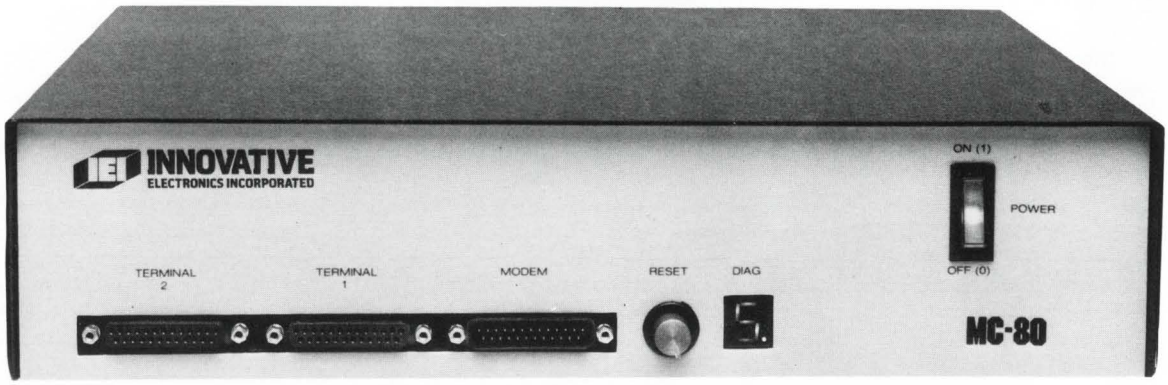
VAX USERS GET A REMOTE, INTERACTIVE TRAINING TOOL

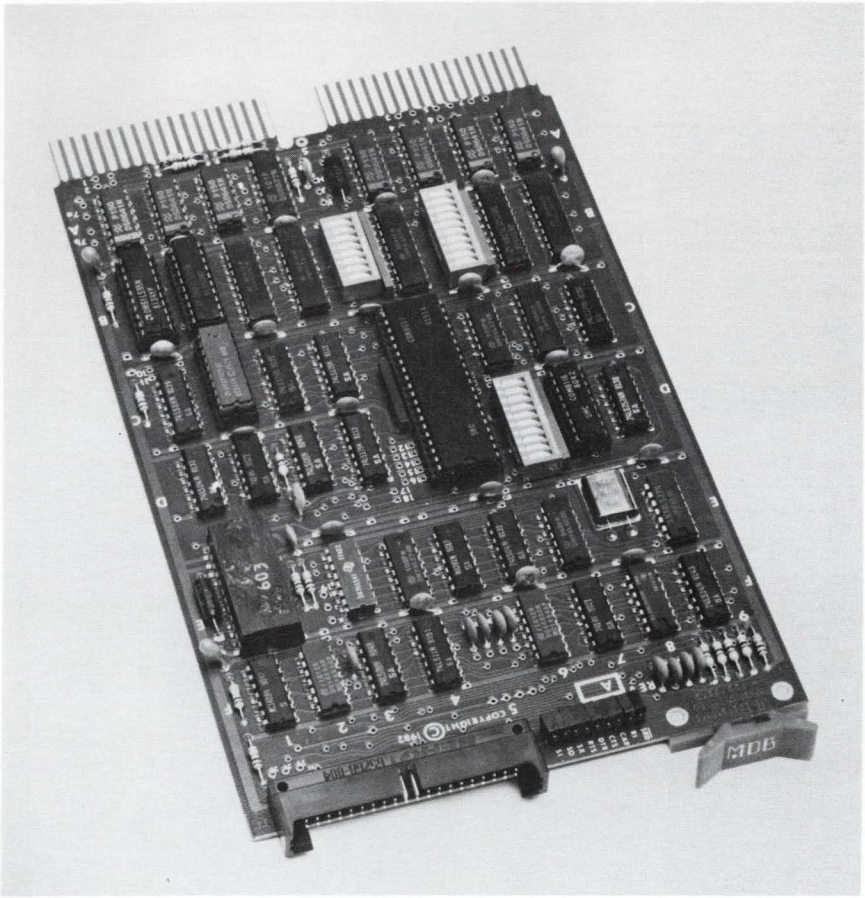
Provo, UT — Clyde Digital Systems, Inc., a principal supplier of application software, announces DEMO, a stand alone utility package for interactive user training and trouble-call support under VAX/VMS. The product also provides a framework for remote software demonstration. VMS software vendors can now demonstrate software and support customers without expensive traveling. DEMO is also invaluable to any VAX site which supports remote users. The DEMO product is similar in function to the popular CONTRL program currently running under the RSTS/E operating system.

This new product runs entirely in VAX native mode and gives users system support never before available under VMS. DEMO allows an authorized support manager to watch any terminal on the system, remote or local. This includes seeing all user keystrokes and computer response. This is invaluable in training and consulting situations where the user needs advice or assistance with computer interaction. The program even allows the support manager to insert input from his terminal as though he were the user. This input as well as the computer's response is echoed to the user's terminal.

DEMO captures in a log file the events of the entire session. All input whether from the support manager's terminal, the user's terminal, or the computer, is identified. This provides the user valuable documentation for future referenece.

Product information may be obtained from: Mary at: (801) 224-5306, 3707 N. Canyon Road, Suite 3-E, Provo, UT 84604.





NEW ASYNCHRONOUS SERIAL INTERFACE MODULE WITH MODEM CONTROL FOR LSI-11 AND LSI-11/23 BASED COMPUTERS. MLSI-DLV11-ED. (FROM MDB)

EMULEX INTRODUCES STATCON SERIES 21 MULTIPLEXER THAT ADDS PORT CONCENTRATION FOR REMOTE DEVICES

Santa Ana, CA — Emulex Corporation today introduced a unique communications product line that for the first time combines both standard host computer multiplexing and remote statistical port concentration capabilities in a single interface module.

The new STATCON SERIES 21 Multiplexer/Port Concentrator takes advantage of the company's CS21 series host multiplexers and adds special microprogramming to provide the demultiplexing capabilities required to funnel remotely concentrated data into a single computer RS-232 port. Up to 32 local and/or remote lines may be connected to any DEC PDP-11 or VAX-11 computer system using just a single backplane slot and without any user modification of standard software. Each remote link plugs into one of the 16 local RS-232 ports and the remaining number of RS-232 ports may be allocated to local asynchronous line applications.

"The object in using any statistical data concentrator is to reduce telephone line costs by sharing one line among several remote terminals," explained William Dollar, Emulex product manager for communications. "In the past, this was accomplished by having a statistical concentrator at each end of the communications link; but this meant that at the CPU end there were as many local ports and cables between the CPU and the local concentrator as there were remote terminal interfaces. While the number of phone lines was reduced, the user paid for a full local concentrator and host multiplexer capabilities.

"Users would obviously prefer to interface the composite communication line directly into the CPU through a single high speed data channel, thus using only one port," he continued. "This previously presented a problem because a complex customer software package had to be developed to handle all aspects of multiplexing, demultiplexing, error handling, line initialization and synchronization, and remote concentrator configuration. Very few users could justify this kind of effort, even in view of the hardware cost savings.

"The new CS21/MX host multiplexer and its companion CM22/EX local statistical port concentrator have been programmed by Emulex to provide all of these demultiplexing functions in a manner which is completely transparent to the DEC system's existing operating software," he said. "Now the network user can reduce total system hardware and also free up available RS-232 ports for other uses without having to make a substantial software development effort.

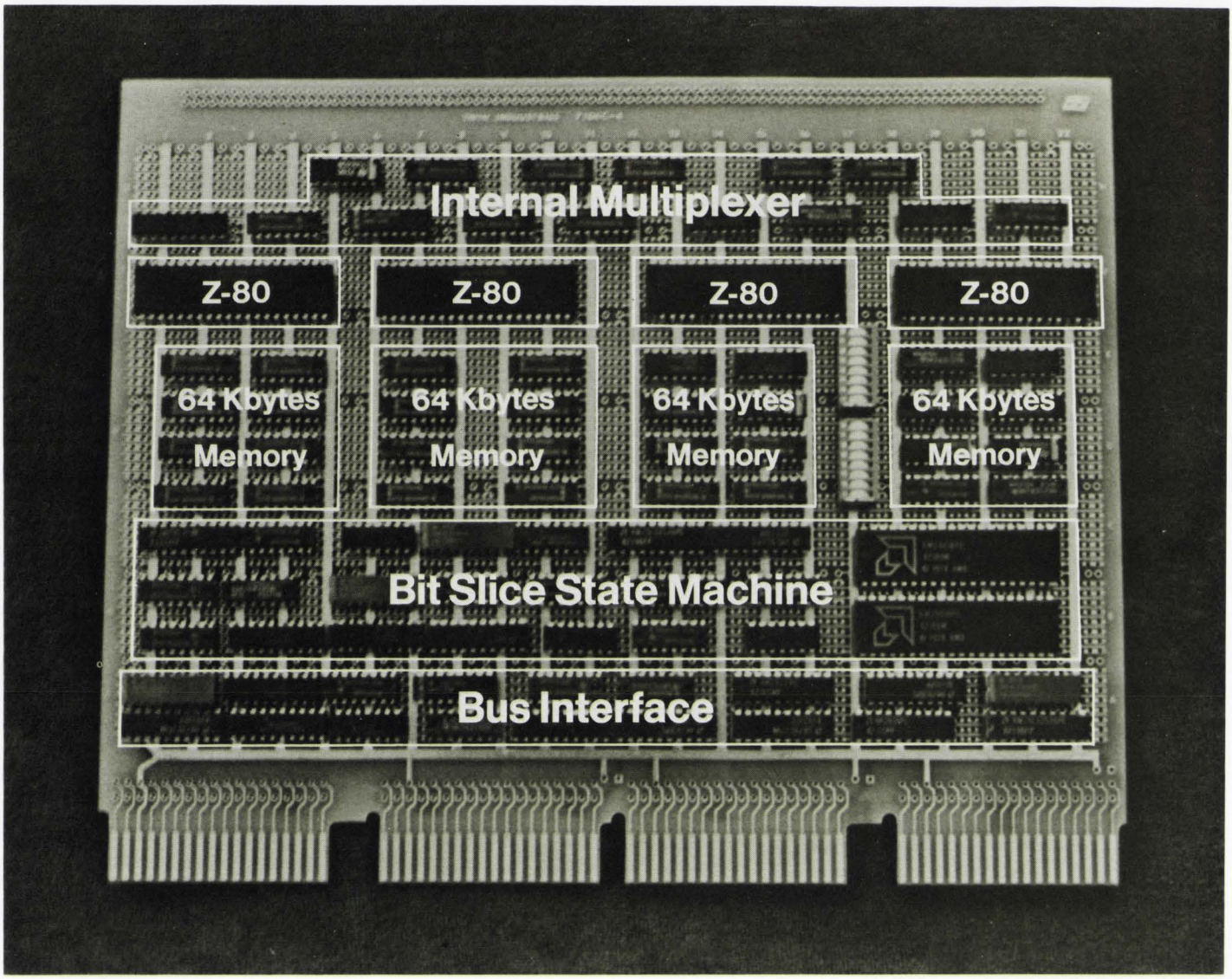
"In addition, the CM22/EX is protocol compatible with the standard Micro800/2 and the Micro8000/2 remote data concentrators made by Micom Systems, Inc. Existing or potential users of these Micom products may therefore avail themselves of this capability without any remote concentrator modification," Dollar said.

The CSM21/MX is offered with either DH11 or DZ11 emulations for use with PDP-11 or VAX-11 computer systems made by DEC. Both are transparent under all PDP-11 Operating Systems and the DZ11 version is transparent under VAX/VMS. Emulex also supports the DH11 emulation under VAX/VMS with its own software to give users a higher performance alternative than the DZ11 version.

"The CS21/MX computer interface plays an important role in the task of port concentration since it acts as the interpreter between the host computer and the local port concentrator," Dollar added. "On the port concentrator side, it is responsible for multiplexing and demultiplexing of data, for modem control, for handling data rate changes, for monitoring link error status, for providing a local/remote diagnostic loopback, and for control of data flow. On the CPU side, it makes the local and remote mix of lines appear to be two functional DH11 asynchronous line multiplexers; or, in the case of the DZ11 emulation, it appears as four separate DZ11 asynchronous line multiplexers. This means the operating system does not have to be patched or otherwise modified for concentration applications."

A single remote link configuration consists of the CS21/MX controller, a 16-port RS-232 Distribution Panel, and a CM22/EX Port Concentrator. An additional CM22/EX is added for each remote link; each CM22/EX plugs into one of the local RS-232 ports. Unused RS-232 ports may be used for local asynchronous lines, provided the total number of local plus remote channels does not exceed 32. The CM22/EX Port Concentrator can be ordered with or without integral modems. Standard Micom Micro800/2 and/or 8000/2 statistical concentrators are used at the remote end of each link.

The user defines through the communication I/O driver whether modem control will be a function of the remote channels or not. If it is not defined, the CM22/EX will not pass modem information to or from the remote



concentrator. If it is defined, the CS21/MX will inform the local port concentrator and the modem signal states for each remote channel on the link will be passed through the CM22/EX transparently. These passed signals then serve as the status or action elements to and from the communication I/O driver.

Individual channel data rate changes are passed through the CM22/EX to the remote data concentrator. Standard DH11 and DZ11 data rates are supported in the corresponding emulation. Link status is sent by the CM22/EX Port Concentrator to the CS21/MX for error reporting to the I/O driver. Data flow is generally controlled by the CM22/EX, which also monitors buffer status of the remote concentrator. Each channel may control data flow, however, through the normal XON/XOFF function. During extremely busy CPU periods, the CS21/MX can also control the incoming data flow from the CM22/EX to the host.

"The CM22/EX Port Concentrator also serves several important functions in management of the communication link and the remote concentrator," Dollar noted. "It is responsible for all error control of data and transfers only error-free data blocks to the host computer. It also handles data buffering requirements, initialization and synchronization of the communication link, and configuration of the remote terminal concentrator to establish data rates and code levels for each channel."

A minimum STATCON SERIES 21 configuration includes a single CS21/MX computer interface PC board, a CP21 16-port distribution panel, and a CM22/EX statistical port concentrator without model. List price for this capability is $6,300. Additional port concentrators may be added at a list price of $2,200, without modem capability. Emulex also provides the remote Micom-compatible statistical concentrators with 4-16 channels, with or without integral modems, at standard established prices.

Like all other Emulex products, these new units may be combined in the company's mix-and-match OEM volume purchase plan to qualify for pricing discounts. Emulex will market and support the STATCON SERIES 21 products both directly and through certain of its authorized dealers.

CP/M* MICRO CO-PROCESSORS FOR THE DEC* UNIBUS AND Q-BUS

Berkeley, CA — Virtual Microsystems Inc. announced today the release of the **z-Board** micro co-processor for the UNIBUS and Q-Bus. The **z-Board** works with VMI's product **THE BRIDGE** Virtual Microcomputer System, to provide a complete virtual CP/M environment on a VAX, PDP-11, or LSI-11 minicomputer.

The UNIBUS version of the z-Board features 4 z-80 micro-processors and a full 256 kbytes of memory, along with a bit slice state machine which handles the bus interface. The Q-Bus version is similar, but may be populated with 1 to 4

z-80's. Each of the z-80's operates independently, providing each BRIDGE user with full microcomputer functionality at any of the attached terminals.

A BRIDGE user begins a session by running THE BRIDGE as a task under VMS, RSX, RSTS, RT-11, or UNIX. THE BRIDGE first tests to see if a z-80 is available on the z-Board. If one is available, the user is "attached" to it. From that point on, the user feels as though his terminal is a fast CP/M microcomputer using hard disk. He can then run any of the programs in the vast CP/M software library. If a slot on the z-Board is not available (any number of z-Boards can be added to a system), the user is '"attached" to THE BRIDGE simulator, which performs all calculations in software. THE BRIDGE handles all I/O, disk storage, printing, and communications. The system also makes use of "virtual floppy disks," files on the hard disk which are like a CP/M floppy disk.

THE BRIDGE Virtual Microcomputer System, complete with z-Board, helps the information manager to manage the proliferation of nicrocomputers. A virtual microcomputer is available to any person on the system at anytime, yet the files are not distributed among various incompatible devices. A virtual microcomputer system can help reduce the costs of hardware significantly, and provide a much more integrated data processing environment. THE BRIDGE and z-Board combination are extraordinarily cost-effective; each slot on the z-Board costs around $1000 — far less than a "low-end" personal computer; yet THE BRIDGE provides hard disk, access to the fast system printers, and is integrated into an already existing, professionally managed system.

THE BRIDGE and z-Board are available *now* for the full range of DEC computers under all the major DEC operating systems. THE BRIDGE costs between $1000-$3500 depending on the size of the minicomputer; each UNIBUS z-Board costs $4000 — Q-Bus versions vary between $1250 and $3500 depending on the number of installed microcomputers.

THE BRIDGE and **z-Board** are trademarks of Virtual Microsystems Inc.

CP/M is a trademark of Digital Research.

DEC, RSX, VMS, RT-11, RSTS, and VAX are trademarks of Digital Equipment Corporation.

———————

Occasionally we are requested to print news that may be of interest to the RSTS community. We are happy to offer this feature to our readers. We reserve the right to print only as time and space permit. We cannot return photos or manuscripts. Send news releases to: RSTS News Release, P.O. Box 361, Ft. Washington, PA 19034-0361.

HELP NEEDED

The RSTS PRO with Carl and Dave is trying to put together a **working** PDP-11/20 **running** RSTS V4A.

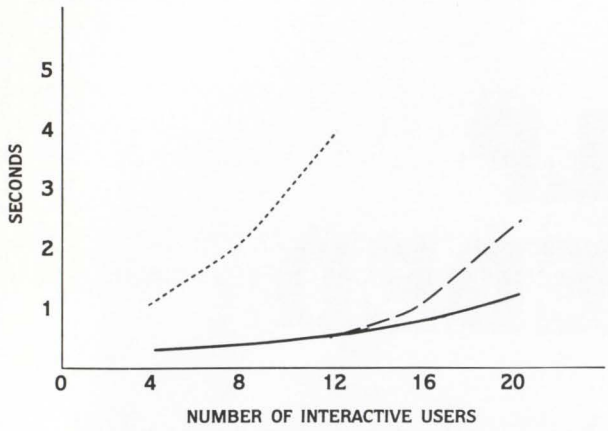
This was the first PDP-11 to run RSTS and we think it would make a nice addition to the DEC Computer Museum (if they accept it).

We need **documentation** for Version 4, as well as any SYSGEN-related information. If you have any or know where some is, please contact us at 215-542-7008. Thank you.

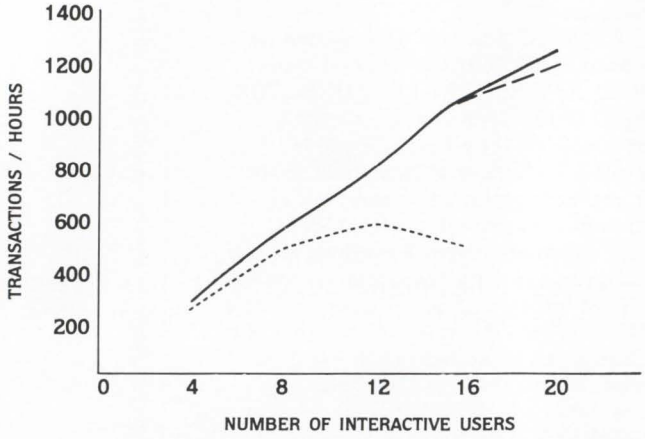
Carl & Dave

VAX PERFORMANCE NOTES

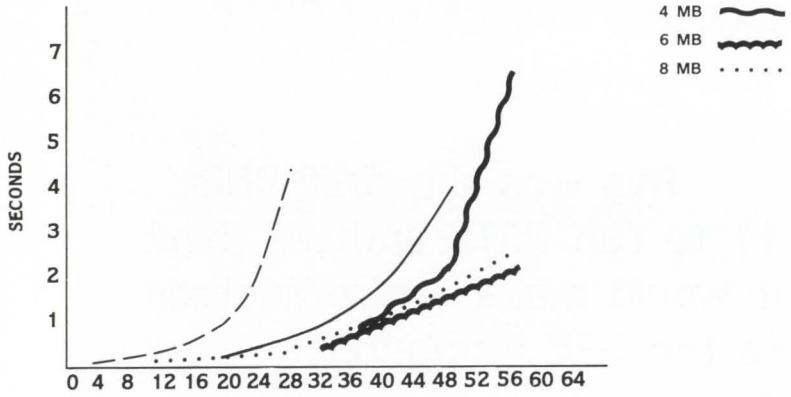
. . . continued from page 49



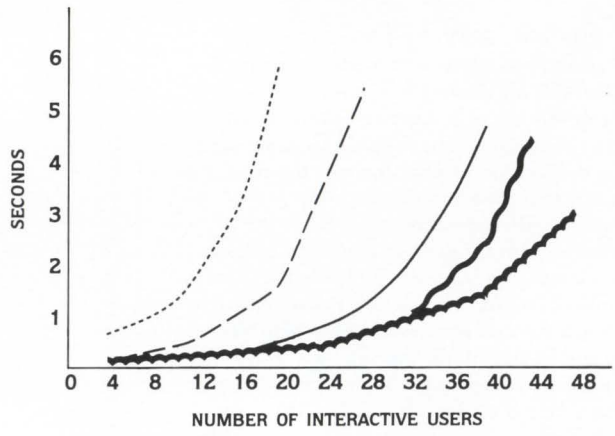
THIS WORKLOAD DEMONSTRATES THAT FOR THE 11/730 1 MB IS TOO LITTLE MEMORY. 2 MB IS ADEQUATE UP TO 16 USERS AND THAT THE CPU HAS THE POWER TO SUPPORT 3 MB.



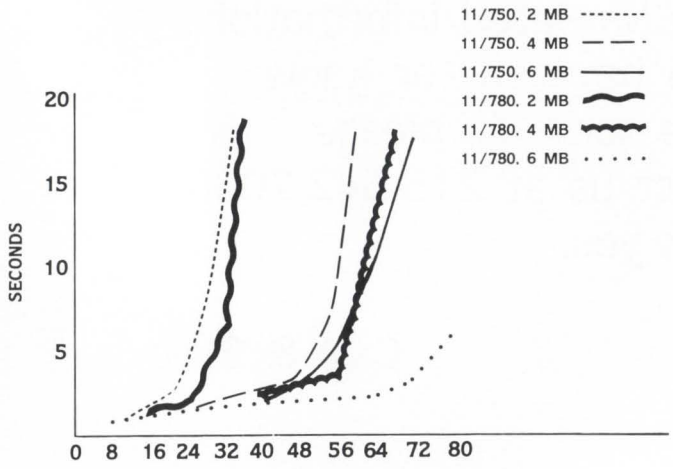
THIS SUPPORTS OUR CONTENTION THAT 2-3 MB IS A BETTER MEMORY SIZE FOR THE 11/730.



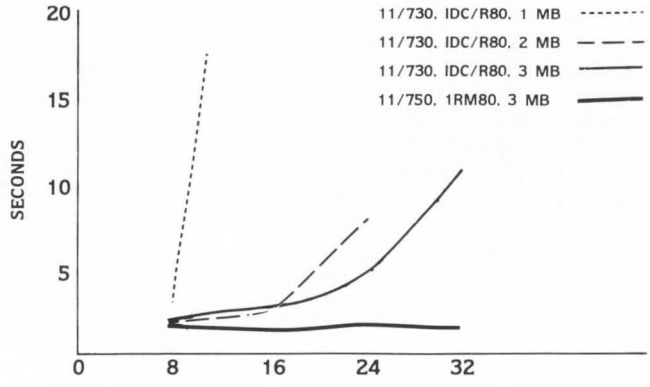
THE 11/780 CONTINUES TO SUPPORT MORE USERS UP TO 6 MB. BUT 8 MB DOESN'T SEEM TO HELP AT ALL.



ADDING MEMORY TO THE 11/750 CONTINUES TO INCREASE PERFORMANCE UP TO 6 MB.



THIS SHOWS HOW THE SYSTEMS "FALL OFF THE TABLE". THE 11/750 PEAKS AT 56 USERS AND 6 MB WHILE THE 11/780 BEGINS TO DEGRADE AT 80 USERS AND 6 MB.



THE 11/730 WHICH DIES WITH 8 USERS AND 1 MB, BEGINS TO DEGRADE AT 24 USERS AND 3 MB. NOTE THAT THE 11/750 WITH 3 MB SHOWS NO SIGNS OF DEGREDATION AT 32 USERS.

RSTS/E INTERNALS MANUAL

The RSTS community has been clamoring for years for a book that details the inner workings of RSTS/E. Well, clamor no more. Michael Mayfield of Northwest Digital Software, and M Systems, the publisher of The RSTS Professional and The DEC Professional Magazines, have teamed up to produce the RSTS/E Monitor Internals Manual.

This manual describes the internal workings and data structures of the RSTS/E monitor. It also notes differences in the internal structures between version 7.1 and earlier versions of the monitor. Future updates will include changes for new versions of the monitor.

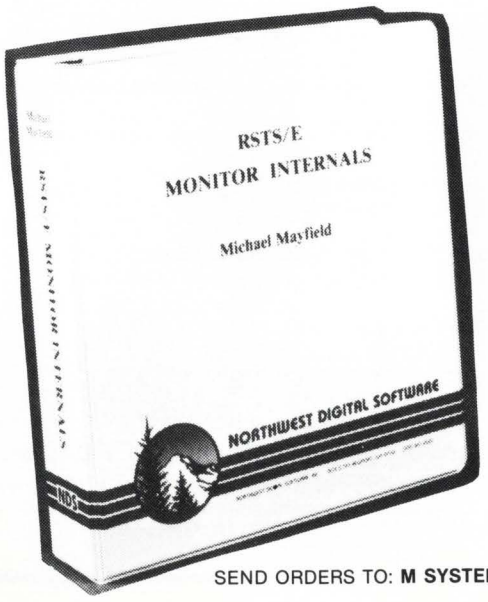
Information is available for all levels of users:

- Gain a basic understanding of the workings of the monitor for optimizing system performance.
- Information on disk structures allows recovery of data from corrupted disk packs.
- Special uses of runtime systems and resident libraries allow complex applications to be developed without degrading system performance.
- Write your own custom device drivers for that "foreign" device you need to add but thought you couldn't.

CONTENTS:

Chapter 1 describes the structures used by the monitor that are resident on disk. These include the directory structure, disk allocation tables, Save Image Library (SIL) formats, bootstrap formats and bad block mapping.

Chapter 2 describes the tables used within the monitor to control system resources and provide program services. These tables provide job, memory, file and device control, as well as program services such as interjob communication.



Chapter 3 contains information on writing and installing a custom device driver. It describes the entry points and information the driver must provide to the monitor as well as the subroutines and macros the monitor provides for the driver.

Chapter 4 contains information that enhances information already provided by Digital on writing custom resident libraries and runtime systems. It concentrates mainly on non-standard uses of resident libraries and runtime systems to increase system performance and functionality.

Appendix A provides six quick reference foldout charts:

- The directory structure.
- The monitor tables.
- Fixed memory locations and common data structures.
- Monitor subroutines.
- Device driver entry points.
- Device driver macros.

Appendix B provides examples of the peek sequences required to access most of the monitor tables. It also contains an example program that uses many of the monitor tables to display a job and open files status.

Appendix C provides an example device driver.

Appendix D provides an example runtime system that doubles as a menu system for restricting specified users to a menu of options.

$95.00

SEND ORDERS TO: **M SYSTEMS, INC., BOX 361, FORT WASHINGTON, PA 19034-0361**

A COUPLE OF CUSP ENHANCEMENTS

By Michael H. Koplitz

There are still many RSTS/E installations that use the Digital spoolers, this installation being no exception to that rule. It has always been a pain to use the QUE.BAS program because when a Control-C is typed during the listing of a spooler, the program would go to "Ready" status. QUE.BAS should be like PIP which goes back to the prompt when a Control-C is entered during a listing of an account or file. The first enhancement shown here is to QUE.BAS. This enhancement will allow the user to type a Control-C during the listing of any of the spoolers while in QUE.BAS and control will return to the "#" prompt. If two Control-Cs are entered, QUE.BAS performs the operations as if an "E" command had been entered. If a Control-C is entered to the command line input the "E" command is executed.

None of the original lines of code in QUE.BAS were edited. All the changes given are new lines of code. They can be entered by any editor.

Code changes to QUE.BAS

```
1005        X$ = SYS(CHR$(6%)+CHR$(-7%))
            IF (E0% AND 16%) = 0%
              !SET CONTROL-C TRAP IF USER IS LOGGED IN

10605       CONTROL.CEE.TRAP% = 0%
              !SET THE CONTROL-C TRAP COUNTER TO ZERO

19077       IF ERR = 28% THEN CONTROL.CEE.TRAP% = CONTROL.CEE.TRAP%
                + 1%
\           CO$ = "E" IF CONTROL.CEE.TRAP% = 2%
                OR ERL = 10600
\           RESUME 10610 IF CONTROL.CEE.TRAP% = 2%
                OR ERL © 10600
\           X$ = SYS(CHR$(6%)+CHR$(-7%))
\           RESUME 1050
\             !TAKES CARE OF CONTROL-C TRAP ERROR.
```

The next enhancement is to MONEY.BAS. MONEY.BAS is a good utility to monitor the activity of the system. The only problem with MONEY.BAS is that the report does not indicate when the data was reset. Therefore it is impossible to know what time frame the MONEY report is referring to. The following enhancement creates a file named "$MONEY.DAT" which contains the date and time when the data was reset. To keep some integrity with this date, a user can not ask to reset data on selected accounts. The date in $MONEY.DAT therefore refers to the whole system.

The code to make this enhancement is given below. Modified lines of code have an asterisk in front of the line number. These changes can be made with any editor.

Code changes to MONEY.BAS

```
  110       ON ERROR GOTO 10000
\           OPEN "$MONEY.DAT" AS FILE #1%
\           INPUT #11%,LAST.DATE$
\           CLOSE #11%
              !OPEN THE MONEY FILE AND GET LAST DATE OF RESET

* 1110      PRINT #1%," on ";DATE$(0%);" at ";TIME$(0%);
\           I% = 1%
\           PRINT #1%," with Data being RESET"; IF R9%
\           PRINT #1%
\           PRINT #1%,"Data last reset on ";LAST.DATE$
\           GOSUB 8100
              !PRINT REST OF HEADER

  1115      GOTO 1120 UNLESS R9%
\           OPEN "$MONEY.DAT" OF ROUTPUT AS FILE #11%
\           PRINT #11%,DATE$(0%);" ";TIME$(0%)
\           CLOSE #11%
              !RESET THE DATA LAST RESET DATE BECAUSE DATA IS RESET.

* 9000      PRINT "Selected accounts can not be reset"
                IF R9%
\           R9% = 0%
              !CHECK RESET CONDITION

  9005      ON ERROR GOTO 9100
\           INPUT "Account";M%(8%),M%(7%)
\           GOTO 9005 IF M%(8%) < 0% OR M%(8%) > 254%
                OR M%(7%) < 0% OR M%(7%) > 254%
                OR (M%(8%) = 0% AND M%(7%) < > 1%)
              !SELECTIVE ACCOUNT LISTING — ENDS WITH CTRL/Z.
              !CHECK GENERAL RANGE
              !ALLOW ONLY [0,1] IF PROJECT NUMBER IS ZERO.

* 9020      PRINT #1%,"Data last reset on ";LAST.DATE$
\           GOSUB 8100
\           GOSUB 2000
\           GOTO 9005

* 9060      PRINT "Cannot find that Account"
\           GOTO 9005

  9115      GOTO 32767

  10000     LAST.DATE$ = "??????"
\           RESUME 140
              !UNKNOWN LAST DATE
```

Until this new version of MONEY.BAS is run with the data being reset the date on which the data was reset will show as ????????. ♥

CLASSIFIED

Send Classified Ads to: RSTS Classified, P.O. Box 361, Ft. Washington, PA 19034-0361. $1.00 per word, first 12 words free with one year's subscription. [Be sure to include a phone number or address in your message.]

Basic Plus Programmer, Small college EDP, (507) 663-4282, Carl Henry.

FOR SALE. 11/40 w/256k, new 800/1600 magtape, DZ11-A, RPO3, LA36, LA120, VT100/AVO, RSTS, Word-11, SAVER. (503) 635-8781.

Copier for sale. 1½ years old—hardly used. Gestetner 2010 w/some supplies. Maintenance contract kept equipment like new. $1000. (215) 537-0782.

RSTS CHEAP!
$350/PORT, $20/MB
on **WALL STREET, NYC**
Local error-free dial-in
or leased line termination
and in **PHILADELPHIA**
DEC 11/70, 24 hrs, 7 days
Secure Backups
B+2, DIBOL, COBOL81
DATATRIEVE, WORD-11, EDT
Secured by **LOCK-11**
NATIONWIDE DATA DIALOG
DAVE MALLERY **215-364-2800**

RSTS RESCUE SQUAD
We salvage all kinds of disasters:
- unreadable disks
- ruined UFDs and MFDs repaired
- immediate response
- telephone DIAL-UP
- on-site
- software tools
- custom recovery
- 90% success to date
- more than 1 GB rescued to date

Brought to you by
On Track Systems, Inc.
and a well known (and read)
RSTS expert.
CALL 24 HOURS
215—542-7133

LOOKING FOR DEVELOPMENT TIME?
NO KILOCORE TICK CHARGES
NO CPU CHARGES
RSTS/E TIME
$7 PER HOUR CONNECT TIME
BASIC PLUS 2
COBOL
BASIC PLUS, PASCAL, "C" } WITH CROSS COMPILER SUPPORT
WORD-II WORD PROCESSING
WAFE, TECO, EDT } PROGRAM EDITING
BUDGET BYTES™ 212-944-9230
by **Omnicomputer™ INC.**
1430 Broadway, New York, N.Y. 10018

DEC BEST VALUES
PRE-OWNED DEC EQUIPMENT
BUYING AND SELLING
SYSTEMS • CPU's • PERIPHERALS • TERMINALS
OPTIONS • MEMORY • COMPATIBLES
CALL DICK BAKER (305) 979-2844
dataware incorporated
Carico Center
2845 NW 62nd Street
Ft. Lauderdale, Florida 33309
Telephone (305) 979-2844

GROSSMAN GRAPHICS
215 537-0782
5041 frankford phila., pa. 19124

Still $200! MORE FEATURES!
$200. RSTS/E* WORD PROCESSOR CBEDIT.BAS
Basic-Plus* program with VDT input, window edit and document save. Add, locate, global change, replace, delete, block move and file merge, etc. Crash and operator error recovery. Supports DEC, Hazletine and Mime standard VDT's. Others easy to add.
Fully formatted output (margins, justify, center, underscore, super-subscript, headers, page numbers, etc.), to typewriter, line printer or disk. Bidirectional driver for Diablo-Xerox 16-17.
User's manual and source code included.
9-Track $200. RKO5 $260. ppd
T.F. Hudgins & Associates, Inc.
P.O. Box 10946, Houston, Texas 77018
Woods Martin 713/682-3651
*TM Digital Equipment Corporation

BACK ISSUE OFFER
ALL 13 BACK ISSUES OF THE *RSTS PROFESSIONAL*
$90.00
Send check to:
THE RSTS PROFESSIONAL
P.O. BOX 361
FT. WASHINGTON, PA 19034-0361
— Payment Must Accompany Order —

The *FAMOUS* RSTS PROFESSIONAL TEE-SHIRT is now for sale!
Send size desired and $6.95 for each shirt to:
RSTS TEE-SHIRT
P.O. Box 361
Ft. Washington, PA 19034-0361
Shirts available in adults sizes only:
Small - Medium - Large - X-Large

Your old video terminals needed. George Fox College (503) 538-8383 Ext. 330.

Pacific Northwest Commercial Time Sharing. Compu-Time, Inc., Portland, Oregon (503) 246-7766.

Superior computer systems seeks subcontractors for the NY metropolitan area. Send resume to 100 North Village Avenue, Rockville Center, NY 11570.

Distributor of DEC hardware, software, maintenance and consulting. American Business Computers (713) 265-2573.

Multibus Development System. Terminal, printer, 2-8" drives, many extras. Jon Coleman (415) 930-9932.

For Sale: PDP8E, RK05, DEC TAPE. Carson-Newman College (615) 475-9061.

Australian VAX Expertise: PRAXA Phone (03) 233-5318; (03) 878-8139; (03) 877-6443 Total systems implemented.

RSTS RSX-11 RT11 Consulting. Quality cost less in the end (404) 261-4634.

DIBS-11, DEC introduced it... VERTEX made it work... CALL (614) 846-DIBS.

For Sale: PDP-11/34A, 128KW, RK07'S, RX01'S, DZ11, 6 VT52'S, RSTS/E and many other licenses. Ron Klausewitz (304) 473-8008.

PLAN PLUS
The Financial Planning & Reporting System
PLAN PLUS LTD
PO BOX 327 LONDON W4 4QD
TELEX 8813271 GECOMS G
Telephone 01-834 7334 ex 7935

LOOK at the "tear-out" cards in this issue.

There are subscription cards for you or a friend.

There's a FREE gift for you.

Bring in new subscribers and collect rewards. See "BOUNTY HUNTERS" card.

There's a READERS INQUIRY card for your convenience.

CHANGES??????

Are you changing addresses? Please let us know so you won't miss one issue of the *RSTS Professional*.

FORMER

Name _____

Address _____

City/State _____

Zip _____

PRESENT/Near Future

Name _____

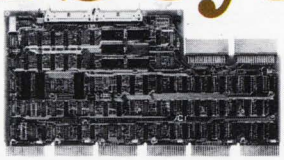
Address _____

City/State _____

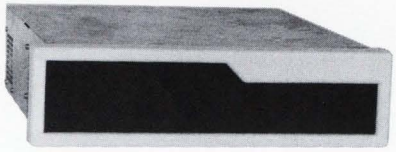
Zip _____

LIST OF ADVERTISERS

Don't wait for that $100,000 RSTS system replacement.



"ENABLE" what you already have instead.



Right now. For under $10,000.

See ABLE products at IRUS '82, Boston Park Plaza, Booths 63, 64, 65 & 66 and at DEXPO/WEST '82, Anaheim Convention Center, Booth 402.

That's the smart thing to do. We know because we've already proved it with ENABLE™, our backplane-resident device for expanding any 18-bit, UNIBUS, RSTS/E system to 4M bytes, and with MEGABOX™, an add-on equivalent for systems without adequate space in the host machine. ENABLE and MEGABOX installations have been working throughout the world for over a year. Users consistently report system throughput to be doubled or better in every case.

If you want to get mileage like that out of your present investment, there is no other way. Only ENABLE or MEGABOX can give you those extra years of outstanding performance without complete system replacement. Our package consists of the ENABLE control, unbundled software and an optional cache with 8K bytes of high-speed memory. DEC OS, maintenance software and diagnostics remain fully applicable; ENABLE or MEGABOX become active only when the ENABLE software patch is installed and recognized.

Once again, ABLE has produced a first-place winner in the continuing DECATHON competition for leadership in the wide world of DEC compatibility. If you think this first is good for RSTS users, just ask us what Mini-MAGNUM™ will do for RSX11M users. That product will be our eighteenth first.

ABLE COMPUTER

CORPORATE HEADQUARTERS
ABLE COMPUTER
1732 Reynolds Avenue
Irvine, CA 92714. (714) 979-7030

NATIONAL OFFICES
Burlington, MA (617) 272-1330
Dallas, TX (214) 387-2855
Daly City, CA (415) 755-6040

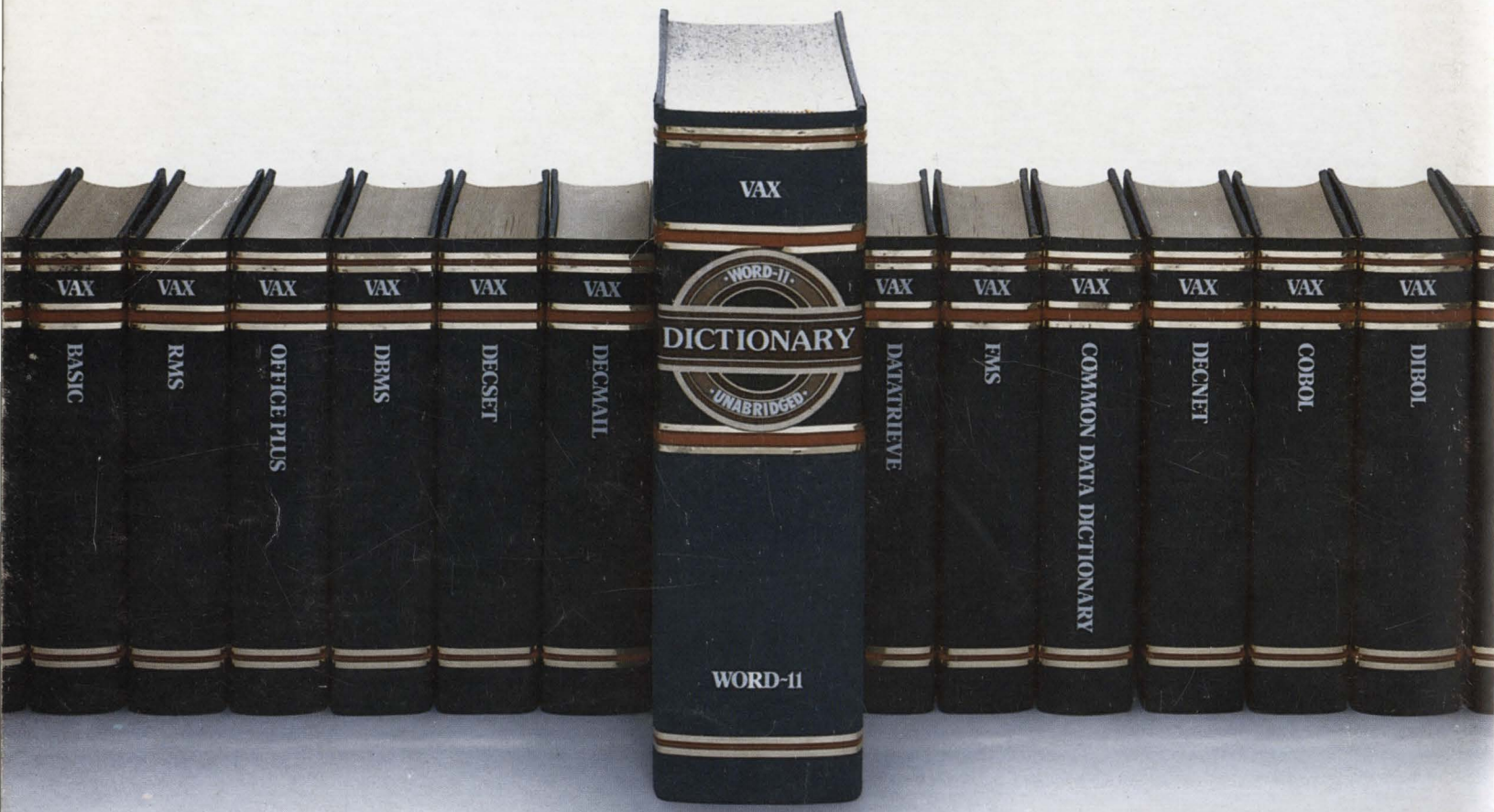
INTERNATIONAL OFFICES
Canada (Toronto) (416) 270-8086
England (Newbury) (0635) 32125
W. Germany (Munich) 089/463080

When we come in first, you always come out ahead.

DEC, UNIBUS, RSX and RSTS are trademarks of Digital Equipment Corporation.

CIRCLE 56 ON READER CARD

WORD-11.
The Word Processing System Vital For VAX.



The Unabridged Addition.

No matter how many additions you've made to your VAX,™ it won't be complete without WORD-11.™ It's the sophisticated word processing system designed to help you use all the other information you've stored. With features like list processing, built-in dictionaries for spelling error detection, automatic table of contents and footnoting that make report writing a snap.

WORD-11 runs concurrently with data processing on multiple terminals. It's easy to use. And it's been up and running in hundreds of installations all over the world for years.

WORD-11 is easy to install and operate. It's cost effective. And it's supported by a dedicated team of experienced specialists. It could be the most important addition you make to your VAX installation. For details, Call, (714) 993-4160.

WORD-11 is also available for Digital's RSX-11M,™ RSX-11M PLUS™ and RSTS/E™ operating systems.

Data Processing Design, Inc.
AUTHORIZED digital COMPUTER DISTRIBUTOR

CORPORATE OFFICE
181 W. Orangethorpe, Suite F
Placentia, CA 92670
714-993-4160 Telex 182-278

N Y OFFICE
420 Lexington Avenue, Suite 633
New York, NY 10170
212-687-0104

WASHINGTON D C OFFICE
4520 East-West Highway, Suite 550
Bethesda, MD 20814
301-657-4098

CIRCLE 59 ON READER CARD