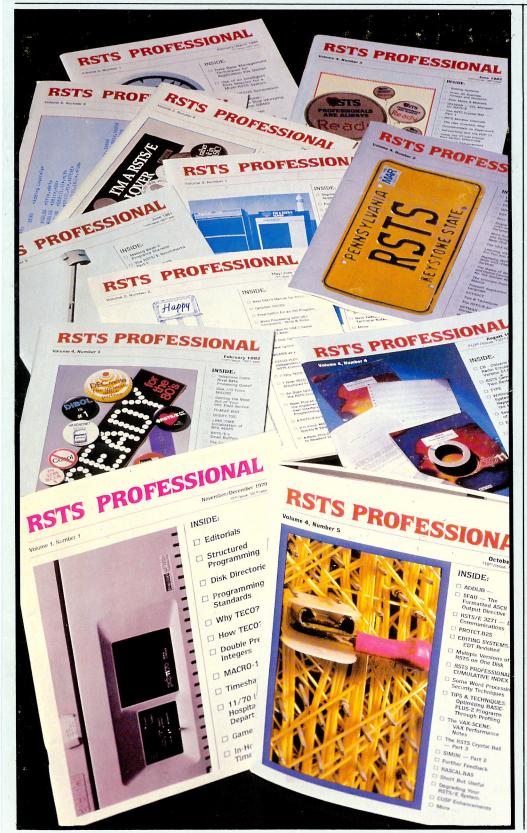
RSTS PROFESSIONAL

Volume 4, Number 6

Third Anniversary Issue

December 1982

\$1000/issue, \$3500/year



INSIDE:

- ☐ How Do You Read RSTS/E Monitor Tables
- ☐ USRDSK Disk Usage Summary Report for RSTS/E
- ☐ EDITING SYSTEMS:

 Some EDT Wishes

 Granted
- ☐ FORMS
- ☐ More on Using the VT100 Printer Port
- ☐ ATPK.TEC
- □ WRTOUT.BAS
- ☐ PROFIL
- ☐ The VAX-SCENE:

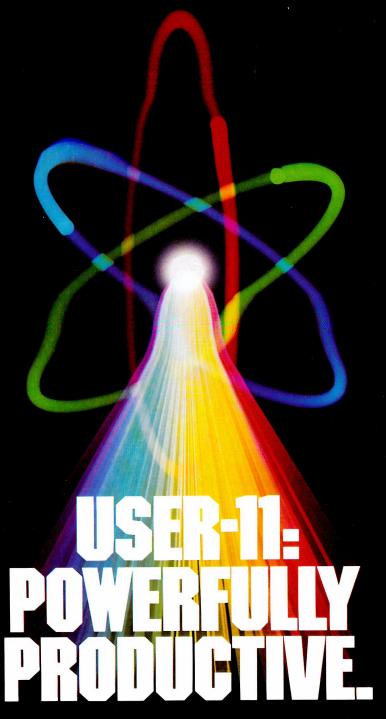
 Reading RMS

 Files From

 VAX MACRO
- ☐ TIPS & TECHNIQUES:

 RSTS Job

 Context
- ☐ HOW TECO
- ☐ SIMINI Part 3
- ☐ A User Written Keyboard Monitor
- ☐ CACHE FLOW
- □ A Security Checklist
- □ DECUS Australia
- ☐ More . . .



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,

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!



North County Computer Services, Inc. 2235 Meyers Ave., Escandida, California 92025

What's the most efficient way to distribute financial models from DEC* to desks?

MAPS/Host & MAPS/Pro



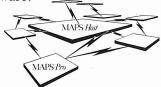
* DEC, VAX and PDP are registered trademarks of Digital Equipment Corporation. MAPS, MAPS/Host and MAPS/Pro are registered trademarks of Ross Systems Incorporated.

OK, ROSS	I'd like to know more about how MAPS can offer me true distributed financial planning. Please send more information on □ MAPS/Host □ MAPS/Pro □ I/We have access to a VAX or PDP-11						
	☐ Have a Ross representative call on me						
Name							
Title							
Company							
Address	<u> </u>						
City	State Zip						
Phone	<u>() </u>						

MAPS™ financial modeling software lets you offer your decision makers a single, comprehensive solution to their complex financial processing needs.

MAPS/Pro software is designed to run on DEC's powerful new Professional 350 desktop computer—to give you the full benefit of the 350's P/OS multi-tasking operating system, 5MB Winchester disk, bitmapped graphics, special function keys and application menus. Fullycompatible MAPS/Host software runs on PDP-11 and VAX computers.

Working independently at desktop 350s, MAPS lets you and your users take advantage of state-of-the-art microprocessing hardware and software.



With 350s linked together, MAPS' common planning language and centrally-administered data base allow users to exchange data and financial models directly. Plus users can upload applications to a PDP-11 or VAX when additional computing power is needed—making *truly* distributed financial planning possible.

Designed for total financial decision support, MAPS features unlimited logic, professional report formats, sophisticated consolidation capabilities and advanced data calculations. Plus instant access to a common data base and model library that maximizes user efficiency. With MAPS there's no limit to the size or complexity of the system you can develop. Yet MAPS' online Help, Business English, and visual data editing make MAPS easy to use. And, fully-documented, it's easy to learn. What's more, Ross "Hot-line" support is as close as the phone if a problem does occur.

For more information on MAPS software, just return our coupon. Or call Ross Systems toll free at (800) 547-1000 (in California, call (415) 856-1100).



1860 Embarcadero Road Palo Alto, CA 94303 Regional offices in San Francisco, New York, Dallas, Los Angeles.

CIRCLE 1 ON READER CARD

SOFTWARE

Provo, UT — Passive security deters tamperers. Familiar methods involve passwords, multi-level access privileges, terminal locks, account dedicated terminals, and other access restrictions.

Need for Passive Security. Passive security provides a necessary deterent to unauthorized access. Operating systems like RSTS/E and VMS come with a degree of passive security already buit in. Additional hardware and software can increase the level of passive security.

Not Enough. Passive security by itself is not enough. A determined individual can break into even the most protected computer systems. Once a user has gained access to the system, passive security weakens. It can't see what the user is doing at the terminal. After gaining access to the system, a covert user works in almost total privacy.

The Active Security Approach. A bank locks its money in a vault and protects its tellers with bullet proof glass. Despite this, it still provides for the occasional undetered thief. Cameras visually record activities in the bank. Here a covert user does not work in privacy. Illegal activity is recorded as it occurs. This is active security.

A Software Camera. Just as a bank needs a video monitor, so does a computer. But the type of camera needed is different. The computer needs a software camera which records all activity occuring at a user's terminal.

PASSIVE -VSACTIVE

CONTRL — Interactive Software Camera. With CONTRL you can look at what is happening on any terminal on your system. You see at your terminal exactly what a remote or local user is typing and exactly what the computer is answering back to the user. With CONTRL watching, a covert user no longer works in private.

Secured Record. While watching with CONTRL, you can request that CONTRL record all of the user's activities in a log file. This log represents a legal history of what the user did at the terminal.

Invisible Eye. The user is unaware that CONTRL is watching. CONTRL does not interfere in any way with programs the user runs.

MONITR — Automatic Software Camera. For fully automated active security you may use MONITR. Like CONTRL it watches and records all activity on a user's terminal. However, MONITR runs without your intervention.

Programmed Monitoring. You tell MONITR which terminals and users to monitor on a periodic basis. MONITR then runs detached and does as instructed. Later you can review MONITR's recorded log for

suspicious terminal activity. Becaus of MONITR's flexability, you ca program it to give as tight a level c active security as desired.

Simple Installation. CONTRL an MONITR run under any RSTS/l system. They require no changes to the operating system or to the terminal handler. Just drop the softwar cameras on your system and the work immediately. In fact they don't even require film. Complete documentation is included with the distribution kit.

Clyde Digital Systems, Inc. is principle supplier of system utilitie and application programs for th PDP-11 and VAX.



How To Get More Information.
Call Lisa at (801) 224-5306, or write Clyde Digital Systems, Inc., 370 North Canyon Road/Bldg. 3-E Provo, UT 84604.

Contents

HOW DO YOU READ RSTS/E MONITOR TABLES? Michael H. Koplitz PEEK, PEEK, PEEK. Here are the definitive peek sequences you always wanted.	8
USRDSK — Disk Usage Summary Report for RSTS/E Philip Hunt Remember DSKLST, well here's one better.	16
David Spencer David keeps making EDT better and better. Happy holidays from our Editor Editor.	22
Jeff Corbett While all IBM operators are skilled at forms alignment, we can use the help.	26
MORE ON USING THE VT100 PRINTER PORT Terry Ridgers How to get more from your computer ports: having a VT100 and a printer share one. The idea just keeps getting better.	30
ATPK.TEC Steve Roy Want to have command substitution with ATPK? TECO can do it!	32
WRTOUT.BAS Ray Tomlinson I did this years ago, but someone always has a better way.	34
PROFIL Joe Blaho We have talked about "profiling" in earlier issues. Here is an implementation in BP2.	38
THE VAX-SCENE READING RMS FILES FROM VAX MACRO Bob "MACRO MAN" Meyer Reading RMS files directly can be fun. It can also be done in Basic Plus, just ask Enterprise Technology. Macro Man uses MACRO of course.	48
TIPS & TECHNIQUES — RSTS Job Context Wef Fleischman Internals can also be fun, 'specially if it's from the experts at Software Techniques. Wef explains what a job actually looks like inside RSTS.	50
HOW TECO Andrew Harrison Is CATALOG finally obsolete? How you can do it in TECO.	56
SIMINI — A Simulation Mini Computer - PART 3	
A USER WRITTEN KEYBOARD MONITOR Ken Harris Would you like users NEVER to have ↑c work? Your own keyboard monitor and default run-time system could do it.	59
CACHE FLOW Kevin Paul Herbert Want to know how the RSTS Cache works, just ask Kevin. Someone did so Kevin found out all about it and now he tells us. I'm glad Kevin is directed towards the forces of good and RSTS.	63
A SECURITY CHECKLIST Dave Mallery DECUS AUSTRALIA	74 75
Carl B. Marbach VERSION 7.2 OF RSTS/E NOW AVAILABLE	80
LATE BREAKING NEWS	81

Coming . .

- Run-Time
 Systems Explained
 (again)
- Number Systems for the PDP-11
- START.BAS
- DIBOL and MACRO, Together!
- QUEDEV.B2S
- V7.2 Reports
- DEXPO and DECUS Revisited
- New Security System Features
- How to Backup a RSTS System
- Floppy Disks Explained
- A Query Language in BASIC PLUS and Record I/O
- More . . .

From the Editors	4
Letters to the RSTS Pro	6
Dear RSTS Man	36
News Releases	
Classifieds	86
List of Advertisers	

The RSTS Professional Magazine. December 1, 1982, Vol. 4, No. 6. Published bi-monthly. Single copy price \$10°0, \$35°0 per year; \$50 Canada and 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. POSTMASTER: send all correspondence and change of address to: RSTS Professional, Box 361, Ft. Washington, Pa. 19034-0361, telephone (215) 542-7008. Copyright © 1982 by M Systems, Inc. No part of this publication may be reproduced in any form without written permission from the publisher.

From the editors. . .

They aren't listening!

Carl B. Marbach

A major computer company (not DEC) ran a series of commercials some time ago telling the world that they were good 'listeners'. This did not happen by accident, they reported, but was a result of in service training classes and seminars on the art of communicating which hinged on 'listening'. In fact, they were teaching the people of their company to 'listen'.

Dealing with people is an art mastered by only a few, and a key to that skill is hearing what the other person has to say. It is frustrating to tell someone something and see that he hasn't 'heard' what you said. Companies can be like that too. Did you ever tell the boss something he didn't hear? Remember, sometimes he doesn't want to hear so don't push too hard. Mass transit is an industry in this country that doesn't hear. Watch: Raise fares and ridership drops which results in loss of revenue; save money by not cleaning the busses and ridership drops which results in loss of revenue; cut schedules and ridership drops which results in loss of revenues. Require exact change, have unfriendly operators, forget air conditioning and heating, leave the burnt out light bulbs burnt out and watch the riders move to their cars and the already clogged roads. Mass transit isn't listening. Try the clean trolleys and busses in Australia where the drivers make change and tell you how to get somewhere and you'll see that it can work. Can't go that far? How about Toronto Canada? Easy, clean, friendly and it works!

How about something closer to our profession, DEC. RSTS is a friendly, efficient, economical and well used operating system. A good (maybe the best) business system. Why are we being short changed? A recent issue of INSIGHT, a DEC produced brochure/magazine, had an article on a new system for every operating system EXCEPT RSTS. Surely an oversight, Dr. Freud. Ethernet, DECNET, Personal Professionals are all announced for everything BUT RSTS. The new FALCON PDP 11 on a chip will run everything, but RSTS was left out. Don't you know we're here? Can't you 'hear' us.

Recently a group of computer dealers got together to form the 'DEC Dealers Association' (more in the January DEC PRO), and one topic of discussion was operating systems and how to sell them with machines. What do you think was one of the top business systems they were concerned with? RSTS. If you were selling a

small business package of A/R, A/P, general ledger, some inventory, order processing, word processing and other utilities with about 4 to 8 terminals, wouldn't RSTS be the way to do it. Maybe on a MICRO-11, an 11/23, 11/23+, 11/24, 11/34 or an 11/40. Bigger? An 11/44 or 11/70. Software? How about 10 years of user written packages and utilities, so many that you have to pick and choose. Sorts, Selects, data bases, backups, office products, editors, screen generators, file processors and too many more to mention. Languages? BASIC, FORTRAN, COBOL, RPG, PASCAL, DIBOL, MACRO and several "higher level" ones. Quite a range of systems; now let me say it loud and clear so they can 'hear' me:

FOR A SMALL BUSINESS SYSTEM, RSTS IS THE BEST OPERATING SYSTEM IN THE WORLD.

Dave Mallery

Westbound on Continental to Silicon Valley—reading about 256 Kbit chips available by 1/83 (in Electronics, which I read every week cover to cover). Reading about 1 Mbit chips hot on their heels. Ain't that neat, 12 chips for 1 MB with ecc. In about 2 vears, maybe 3, that would fit across this page and 11/2 inches high. That's 2 MB + an 11/70 plus ethernet, disc controller and who knows what on a single hex hi board. But why bother with 'hex high', that's long gone. Who needs a backplane when the whole thing fits on a board. Actually, the unibus now runs across the board, and the various chip carriers plug into it. Boy, Gordon, did you ever have a good idea! What's really neat is that all my RSTS based software will still run on this new imaginary wonder.

Accounts Payable still pays if it runs on one board or a whole box full of 11/70! When will the accounting cycle receive the same treatment as the nand gate? It's good to keep that in view as we gaze on the latest wonders of electron beam lithography—journal entries still have to balance, as do vouchers. Invoices (unpaid) still age. Word processing still processes words! Beethoven's 9th fits into a tiny cassette in my Walkman, but it still needs the genius of full symphony orchestra, a chorus and a conductor to realize it, before it can be packaged. There is still room in the world for editorials such as these, inspired by the gin in first class. Thank God!



Editors R.D. Mallery Carl B. Marbach

Assistant Editor/ Advertising Helen B. Marbach

Administrative Editorial Assistant Assistant Linda DiBiasio

Hope Makransky Subscription Fulfillment

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

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

Mid-Atlantic Representative Ed Shaud P.O. Box 187

Radnor, PA 19087 215/688-7233 Contributors

Joe Blaho John Cato Jeff Corbett Wef Fleischman Ken Harris Andrew Harrison Kevin Paul Herbert

Cartoons Douglas Benoit Frank Baginski

Cover Photo David Sheppard N.Y. & New England Representative Jack Garland P.O. Box 314 S.H.S. Duxbury, MA 02332 617/934-6464

Assistant Editor/

Operations

Peg Leiby

Philip Hunt Michael H. Koplitz Bob Meyer Terry Ridgers Steve Roy David Spencer Raymond D. Tomlinson

Design & Production Grossman Graphics Printing & Binding Schneider Litho

Co., Inc.

ALL PROGRAMS PUBLISHED IN THE RSTS PROFESSIONAL ARE WARRANTEED TO PERFORM NO USEFUL FUNCTION. THEY ARE GUARANTEED TO CONTAIN BUGS. THEY ARE DESIGNED TO GET YOU THINKING. THEY ARE INTENDED TO EDUCATE AND ENTERTAIN. THEY ARE PUBLISHED ON THE PREMISE THAT IT IS BETTER TO SPREAD PEOPLE'S BEST EFFORTS AROUND EVEN IF THERE IS AN OCCASIONAL PROBLEM, IF YOU USE THEM, MAKE THEM YOUR OWN. AND YOU WILL NOT GO WRONG.

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.)

*This magazine is not sponsored or approved by or connected in any way with Digital Equipment Corporation. "RSTS" and "DEC" are registered trademarks of Digital Equipment Corporation. Digital Equipment Corporation is the owner of the trademarks "RSTS" and "DEC" and is the source of all "DEC" products --.

Material presented in this publication in no way reflects the specifications or policies of Digital Equipment Corporation. All materials presented are believed accurate, but we cannot assume responsibility for their accuracy or application.

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.

System Performance House, Inc.



LETTERS to the RSTS Pro...

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

RMS temporary files opened under BASIC +2 take on a default system name in Version 7.0 of RSTS; usually of the form: TEMPnn.TMP. Basic +2 under RMS will ignore the account and file name that you specify in your OPEN statement. Problems can arise when you run a program that opens a temporary file on a specific DISK PACK drive from a different account; say a privileged program run from a non-privileged account. If that non-privileged account was not created on that specific DISK PACK drive, then you will receive a system error: ?Disk PACK IS PRIVATE.

Roy Gorka, Systems & Computer Services, DuPont Canada Inc., Mississauga, Ontario, has come across a possible RSTS Version 7.0 BASIC +2 RMS System Bug. It appears that if two files are opened on the same channel (even if the first one is closed before the second OPEN statement is executed), then BASIC +2 under RMS will expect the second file to be located on the same device as the first file, if there is no explicit device specified for the second file. It does not default to the system disk.

Terry Ridgers, Analyst/Programmer
DuPont Canada Inc.
St. Clair River Works, Corunna, ONT.

Since I started reading (and enjoying) the RSTS Professional about two years ago, I've seen various articles about program preprocessors (BLINK, Dec, 1981), program standards, maintenance philosophies, etc.

As expected, everybody has their own methods, and does their own things, in their own little world.

If in fact there are some 10,000 RSTS sites in the world, it would seem to me that consolidation of these 9,999 various methods of program maintenance and development would be a worthwhile effort.

A BASIC-Plus-2 future, combined with an ever-increasing maintenance load, is forcing us to review the BASIC-Plus pre-processor that we currently use.

I'd like to issue a friendly challenge to your readers to send me their ideas on preprocessors. I will summarize the input I receive and report the results in a future issue of RSTS Professional. For discussion, some features we have or are considering are:

- non-standard SCALE factors
- conditional assembling
- compile-time variables
- labels vs. line numbers
- generation of .ODL files
- nesting of command files
- source module libraries
- batch compilation
- ability to process ANY language text or command file
 - compilation instructions internal to a

program

Please forward your notes, suggestions, examples to:

Garry D. Robbins, Technical Services
Analyst

Labatt Brewing Company, 150 Simcoe St. London, Ontario N6A 4M3 Phone (519) 673-5186

P.S. Keep up the excellent work!

Don't forget that BASIC V2 will soon be upon us with some of these features already built in (labels for instance). Maybe the preprocessor needs to work for BASIC, BASIC V2, BASIC-Plus-2 and VAX-BASIC.

The RSTS Pro proved invaluable last week when after a disk failure we were left with a dirty pack with a user data error in the UFD for "1,3". With the aid of Vol. 2, No. 4 we reset the "mounted" bit, set the pointer to the 1st name entry in the UFD to zero, and were then able to proceed with recovering the disk. Wonderful.

There are some important things going on at DEC that the commercial SIG ought to be pursuing. Did you know, for example, that the CTS500 release of the DIBS Dibol accounting application software will use DMS, not RMS? I think that the whole point of the SIG should be to prevent idiocies such as this on DEC's part. Oh well...

Looking forward to hearing from you.

Rob Lamb, System Manager

LA Computer Services Ltd, London

I wanted to compliment you on a fine magazine. The days crawl from issue to issue and I am looking forward to the time "RSTS Pro" becomes a monthly magazine.

Have there been any articles on how to interpet the ERRDIS listing?

It appears to be a useful diagnostic tool, but other than the date and heading I can't seem to interpet anything intelligible. (And DEC doesn't seem to be interested in its results either.)

I have enclosed a little sillyness that I put together for a laugh. I hope you enjoy it. Keep up the good work.

Dennis DeMaria Data Processing Manager Ark-Les Corp., Watertown, Mass.

Thank you, Dennis. Readers can find your "sillyness," Chéz RSTS on page 24.

Integrated Services, Inc., writes thousands of checks and other documents from our PDP computers each month which require the writing of dollars and other numbers in English, with both check and new program volume increasing monthly. Consequently we needed a routine which would convert a number into a string of words without using much program space since we are currently using BASIC PLUS.

I accomplished this with the attached function routine which will create such a

string with a value up to .1E 13-.01, which, so far, exceeds our needs. It would be a simple matter to increase the maximum amount by increments of 1000 times the current limit. The routine uses a minimum of string space overhead and minimum of code by looping back into itself after finding and parsing the largest "set" within the number (i.e., millions) passed to it.

My intentions are to remove the function (FN) part of the routine when it is converted to a CALLed routine in BASIC PLUS II, then placed in a general usage library available to all programs on the system. Having also written FORTRAN and COBOL, I feel this code should be easily converted to one of these languages as a COPY for COBOL and subroutine for FORTRAN.

Raymond D. Tomlinson Systems Programmer Integrated Services, Inc., Aurora, CO. See Raymond's article "WRTOUT.BAS" on page 34 of this issue.

I am writing on behalf of the DECUS Australia Chapter Board and members of the RSTS SIG to express our thanks and appreciation for your services which helped make our 1982 Symposium the success it was.

Both during and since the Symposium I have been gathering comments from attendees to include in my report to the Chairman of the Symposium Planning Committee. All comments associated with the pre-symposium seminar you conducted have been most favourable and it looks as though it is going to be a very hard act to follow in future symposia.

I would like also to take this opportunity to express my appreciation for the very active role you played in the main part of our Symposium in the days that followed your seminar

We hope you enjoyed your stay in Australia as much as we enjoyed your being here and that you have pleasant memories of your visit.

Yours faithfully, Chris Brett DECUS Australia

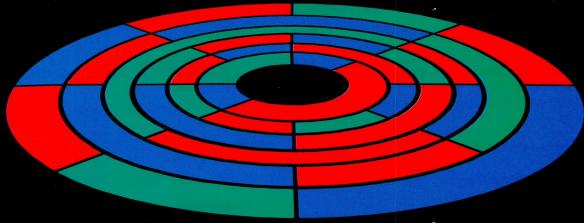
I used to love TECO. I have followed the many articles in the RSTS PROFESSIONAL on TECO with great interest. It is nice to know that someone still appreciates the only true 'DEC Standard Editor'. But, there are always two sides to any story, and I hope you will forgive me for bringing up some anti-TECO points.

I am enclosing, unedited, a copy of a letter I wrote to the editor of a DECUS SIG newsletter a few months ago. I don't think it was published. Perhaps you might want to publish it, if for no other reason than to show that the RSTS PROFESSIONAL dares to print the opinions of those whose beliefs run contrary to those of the majority.

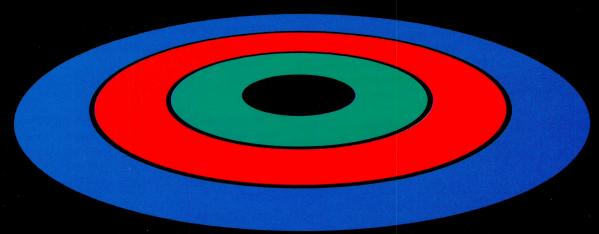
Michael E. Mazzoni, President Process Control Systems, Inc. Brookfield, WI

... continued on page 40

WHAT YOU DON'T KNOW ABOUT YOUR DISKS IS COSTING YOU MONEY



If your disk looks like this, you're wasting system performance.



If your disk looks like this, you're using DISKIT.

When the job you're running requires reading the "red" file, it naturally happens faster on a well-ordered disk. Disks become "fragmented" as you use your computer. The system slows down. And that costs you money.

Now, you can restructure your disks and get back that lost performance (up to 50%) without spending a dime on new hardware. DISKIT is the original software system that makes this possible.

But don't confuse DISKIT with other system utilities, DISKIT is a complete "software tool kit" that optimizes your RSTS/E system.

DISKIT is:

- DSU The utility which restructures the information on your disk, making data fast and easy to access.
- DIR The incredible directory tool that finds files at the rate of 400 per second.
- RDR Reorders disk directories 30 times faster than ever before possible.
- OPEN Displays complete job statistics and file activity so you can see what your system is doing.
- DUS The set of CALLable subroutines which pre-extend file directories, reducing fragmentation.

In today's tight economy, it's more important than ever to get the most out of your hardware investment. Call or write today and start getting your money's worth from your computer.

Software Techniques

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

HOW DO YOU READ RSTS/E MONITOR TABLES?

By Michael H. Koplitz

There have been several articles written that deal with the RSTS/E monitor tables, but none of these articles explains how to write the algorithms necessary to read the tables. BASIC-PLUS can be used to read the monitor tables instead of MACRO-11. This article will give the algorithms necessary to read any table in the job data structure for any job. It is assumed that the reader has an understanding about the job data structures (all of the 1981 issues of the RSTS Pro contain articles about the tables) and of BASIC-PLUS.

One SYS call will be used which gives the pointers to the job table. The job table contains pointers to the individual JDBs, the first job data structure (figure A gives the job data structures).

<code>JOBTBL% = SWAP%(CVT\$%(MID(SYS(CHR\$(6%)+CHR\$(-3%)),11%,2%)))</code> Every memory location can now be accessed by the PEEK function. The variable JOBTBL% points to the first location in the job table. The first entry of the job table is for job zero, the null job, this entry is zero. The final entry of the table is -1.

When utilizing SYS calls the reader should recall that the job number is always returned as JOB.NUMBER*2. The reason for this is that the entries in the job table increase by words.

Therefore the JDB% starting address for any job is:

JDB% = PEEK(JOBTBL% + (JOB.NUMBER% *2%))

If JDB% is zero then the job does not exist on the system.

The entries in the job table can be gathered by the following algorithm:

```
JDIOB% = PEEK(JDB%)

JDFLG6% = PEEK(JDB8+2%)

JDIOST% = PEEK(JDB8+4%) AND 255%

JDPOST% = SWAP%(PEEK(JDB8+4%)) AND 255%

JDWORK% = PEEK(JDB8+6%)

JDJDB2% = PEEK(JDB8+6%)

JDJLG2% = PEEK(JDB8+10%) AND 255%

JDFLG2% = PEEK(JDB8+10%) AND 255%

JDFLG8% = PEEK(JDB8+12%)

JDRESQ% = PEEK(JDB8+12%)

JDRIZE% = PEEK(JDB8+12%)

JDSIZE% = PEEK(JDB8+12%)

JDSIZE% = PEEK(JDB8+22%) AND 255%

JDPRIS = PEEK(JDB8+22%) AND 255%

JDRRST% = SWAP%(PEEK(JDB8+28%)) AND 255%

JDBRST% = SWAP%(PEEK(JDB8+28%)) AND 255%
```

The priority of the job, JDPRI% can be negative. In twos complement the negative number will display as a positive number because it is only a byte and the seventh bit becomes the sign bit. Therefore the following statement is performed if the value is negative:

```
JDPRI% = JDPRI% - 256%
IF JDPRI% > 128%
```

This statement moves the sign bit from bit 7 to bit 15, resulting in a negative number.

The words of the secondary block job, JDB2 can now be read as follows:

```
JDB2% = JDJSB2%

J2TICK% = PEEK(JDB2%)

J2CPU% = PEEK(JDB2%+2%)

J2CPU% = PEEK(JDB2%+4%)

J2KCT% = PEEK(JDB2%+6%)

J2DEV% = PEEK(JDB2%+6%)

PROGRAM.NAMES = RADS(PEEK(JDB2%+12%))

+ RADS(PEEK(JDB2%+24%))

PROJECT.NUMBER% = PEEK(JDB2%+24%) AND 255%
```

A pattern can be seen on how to gather the information from the data structures. Only offsets are needed to read any information. If the data is in the high order byte the word must be "SWAP%()" first, then "AND 255%" on the swapped word will give the bits. Data in the low order byte can be obtained by the word "AND 255%".

When obtaining the current account in a program the function

```
A% = PEEK(PEEK(PEEK(520%)+8%)+24%)
PROJ% = SWAP%(A%) AND 255%
PROG% = A% AND 255%
```

is used. An analysis of this function shows where this data actually comes from. PEEK(520%), memory location 520 decimal, 1010 octal, contains the starting address of the JDB block for the current running job. Therefore JDB% is equal to PEEK(520%), the starting address of the JDB structure.

The next location would be PEEK(JDB% + 8%), substituting JDB% for PEEK(520%). Note that this location is the pointer to the job's JDB2 structure. Therefore JDB2% is equal to PEEK(JDB% + 8%) and when substituted back into the original function the result is:

```
A% = PEEK (JDB2%+24%)
```

Offset 24 in the JDB2% is the PPN. A minor change to this function would allow the gathering of any account number for any job.

```
A% = PEEK(PEEK(PEEK(JOBTBL%+JOB.NUMBER%*2%)+8%)+24%)
PROJ% = SWAP%(A%) AND 255%
PROG% = A% AND 255%
```

The program running for a job would be:

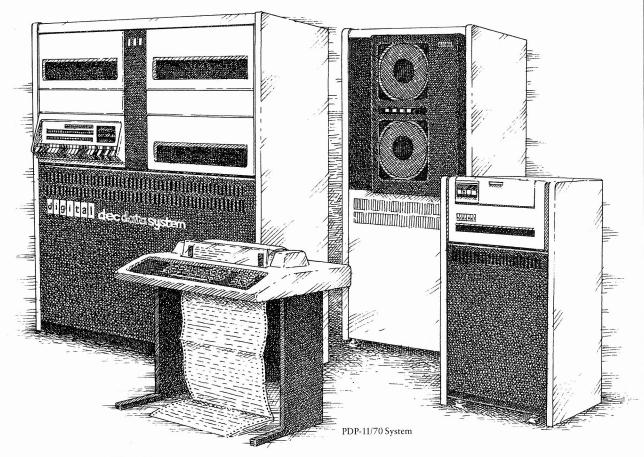
```
PROGRAM.NAMES = RAD$(PEEK(PEEK(JOBTBL$
+ (JOB.NUMBER$*2$))+8$)+12$))
+ RAD$(PEEK(PEEK(PEEK(JOBTBL$
+ (JOB.NUMBER$*2$))+8$)+14$))
```

The RTS being used is:

```
RTS.NAME$ = RAD$(PEEK(PEEK(JOBTBL\{\} + (JOB.NUMBER\{\}^2\{\})+12\{\})+2\{\})\ + RAD$(PEEK(PEEK(PEEK(JOBTBL\{\}^2\{\})+12\{\})+2\{\})\ + (JOB.NUMBER\{\}^2\{\})+12\{\})+4\{\})
```

If the job is a receiver then the ID is:

State of the Art. Again.



Most computers that have been used are just that. Used. But Digital thinks you deserve better than that. So, before we resell any system, we completely refurbish it at the factory.

We actually modernize it to our latest engineering standards. Even the warranty we give it is state of the art. The same 90-day warranty that goes on our new equipment.

That's the difference between our systems and anybody else's. Everybody's were fresh from the factory once. But ours are fresh from the factory twice.

Call your local Digital sales office or 1-800-258-1728.

In New Hampshire, (603) 884-7240. Digital Equipment Corporation, Traditional Products, 16 Hampshire Drive, Hudson, NH 03051.



Now that the methods to read the RSTS/E monitor tables are known, a very descriptive "SYSTAT" can be developed. The following program gives the algorithms to read the rest of the job data structures. The program is oriented to a VT-100 CRT. If a different CRT is used, changes must be made to the FNC\$ function; this is the cursor control.

trol.		
JDB	JDB2	RTS (both)
Ipointer to IOB !	!unposted clock ticks !	!pointer to next RTS!
Iprimary job status !	ICPU time (LSB) !	Iruntime name in !
!mask ! IOSTS !		1 RADIX-50 1
!pointer to WRK !	!kilo-core-ticks(LSB) !	default extension
!pointer to JDB2	Idevice time !	!memory control !
Inew size ! job status!		1 1
Ipointer to RTS		! sub-block !
Iresidency quantum !	1 !	1
!memory control !	!default RTS pointer !	į į
!	!RIB pointer	!starting block RTS !
! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	1	
! Sub-block !	!large data pointer !	!FIP unit ! R.DATA !
	!large data count !	!block number (LSB) !
1 !	Iproject ! programmer !	!offset ! R.FILE !
IL3Q bits to set on !	IDCN of UFD block !	Ires count!use count!
[Runburst[priority]	IWDB pointer !	!min size !max size !
!swap sl.!max size !	!^T mini-sys CPU time !	IRTS character. !
RIB	WDB	IOB
!link to next RIB !	!pointer to next WDB !	IDDB ptr for ch. 0 !
!receiver ID !	!pointer to library 5 !	IDDB, WCB, SCB ch. 1 I
! ! 6 bytes of !	!pointer to library 4 !	IDDB, WCB, SCB ch. 2 !
I ASCII I	<pre>!pointer to library 3 !</pre>	IDDB, WCB, SCB ch. 3 1
lobj type !job # * 2 !	!pointer to library 2 !	IDDB, WCB, SCB ch. 4 !
!unused laccess flag!	!pointer to library 1 !	IDDB, WCB, SCB ch. 5 1
!buffer maximum !	! address window 1 !	IDDB, WCB, SCB ch. 6 !
!pend cnt.Imess max. !	1	IDDB, WCB, SCB ch. 7 I
Ipointer to first PMB !	į	IDDB, WCB, SCB ch. 8 !
!pointer to last PMB !	i	IDDB, WCB, SCB ch. 9 !
llink cnt.!max link !		
		IDDB, WCB, SCB ch. 10 !
Iptr. first link block!	! address window 2 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!	IDDB, WCB, SCB, ch. 11 !
!ptr. last link block !		!DDB,WCB,SCB ch. 12 !
! reserved for !!		IDDB, WCB, SCB ch. 13 !
! network use !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!	! ! !	IDDB, WCB, SCB ch. 14 I
II	! !	!DDB,WCB,SCB ch. 15 !
WCB	FCB	DDB
!status !disk index !	!link to next FCB !	!type !drive index!
!flag bits!job # * 2 !	!link to UFD name ent.!	!unit # !owner * 2 !
	!project !programmer !	!ownership time start!
!next block LSB !		!ownership count !
!pointer to F\$CLUS !		!device flag bits !
!retrival entry number!	!file extension RAD-50!	!small buff empty ptr!
!ptr. to next WCB !	Iprot code !status !	!small buff fill ptr !
	!RR access !N/U access!	!small buff count !
! !	!FBB of first !	!width+l !horiz pos.!
!Retrieval entry !	! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
! ! !! !!	IFBB of name entry !	
1 1	1 1	
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!	I !	
!	!size(MSB) !FIP unit !	
	Ifile size (LSB) !	
	!file cluster size !	
1 !	!pointer to 1st WCB !	
		- The state of the state of

FIGURE A. Some RSTS/E Job Structures

```
!*
!* THIS PROGRAM WAS WRITTEN BY M H KOPLITZ, 23-FEB-82
!*
!* THIS PROGRAM WILL DISPLAY THE JOB DATA STRUCTURES !
              THIS PROGRAM WILL DISPLAY THE JOB DATA STRUCTURES FOR ANY JOB
  010
                EXTEND
                PRINT "Job interval"; INPUT LINE A$
  040
                AS = CVTSS(AS.4%)
                GOTO 050 IF A$
               GOTO 050 IF AS = ""

%% = INSTR(1&,AS,",")

J1% = VAL(AS) IF Y% = 0%

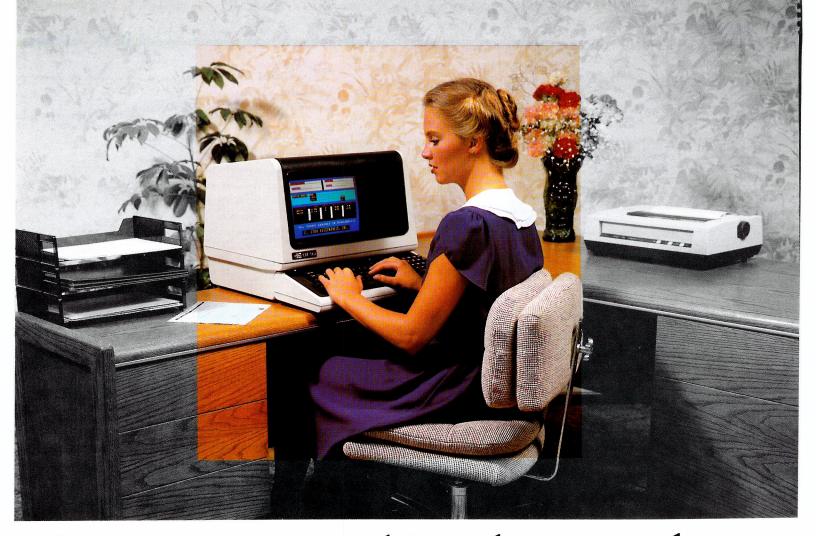
J1% = VAL(LEFT(AS,Y%-1%)) IF Y% > 0%

J2% = VAL(RIGHT(AS,Y%+1%)) IF Y% > 0%

J2% = J1% IF Y% = 0%
                JOBTBL% = SWAP%(CVT$%(MID(SYS(CHR$(6%)
+CHR$(-3%)),11%,2%)))
  050
                                                                                                         !GET START_ADDRESS (
                JOB.NUMBER% = JOB.NUMBER% + 1%
JOB.NUMBER% = 1%
               IF PEEK (JOBTBL%
                GOTO 090 IF JOB.NUMBER% < J1%
GOTO 090 IF JOB.NUMBER% > J2%
                                                                                                         !GET NEXT JOB NUMBEF
!LOOK AT, WHEN ZERO
!NO JOB, -1 MEANS EN
!OF TABLE SO RESET
               PRINT FNC$(0,0);FNC$(0,5);FNC$(1,25);
    "Job";JOB.NUMBER*;
    FNC$(0,7);FNC$(0,10);"Data";
    FNC$(0,9);FNC$(0,8);" Structures";
PRINT FNC$(2,15);FNC$(0,7);FNC$(0,10);"JDB";
    FNC$(0,9);FNC$(0,8);
JDBPTR* = PEEK(JOBTBL* + (JOB.NUMBER**2*))
 100
  1
                                                                                                         !PRINT HEADERS, GET !POINTER TO JDB
                PRINT FNC$(3,1);FNC$(0,7);FNC$(0,25);
FNC$(0,10);"1";
STRING$(34%,113%);"k"
PRINT "x"
 105
                PRINT "X"
FOR X% = 4% TO 13%
PRINT "m"; STRING$(34%,113%); "j"; FNC$(0,8)
FOR X% = 4% TO 13%
PRINT FNC$(X%,36%); FNC$(0,7); FNC$(0,10);
                                             FNC$(0,25); "x"; FNC$(0,9);
FNC$(0,8)
 1
               NEXT X%
                                                                                                         IDRAW THE PRETTY BOX
               PRINT FNC$(4,8); "Primary stats: ";
JFPOST% = PEEK(JDBPTR% + 2%)
PRINT ".RSX";
IF JFPOST% AND 2%**2%
PRINT "."C";
IF JFPOST% AND 2%**3%
PRINT ".2"C";
IF JFPOST% AND 2%**4%
PRINT ".PR";
IF JFPOST% AND 2%**4%
PRINT ".PR";
 110
                IF JFPOST% AND 2%**10% PRINT ".TP";
                PRINT ".TP";
IF JFPOST% AND 2**11%
PRINT ".NL";
IF JFPOST% AND 2%**12%
               PRINT ".LK"
                              IF JFPOST% AND 2%**14%
                                                                                                         IPRINT PRIMARY STATS
               PRINT FNC$(5,11); "Error code: ";
JDIOST% = PEEK(JDBPTR% + 4%) AND 255%
PRINT JDIOST%
 120
                                                                                                         !PRINT ERROR CODE TO
                                                                                                         ! RETURNED AFTER ! MONITOR CALL
               PRINT FNC$(6,9); "Posting mask: ";
JDPOST% = SWAP%(PEEK(JDBPTR% + 4%)) AND 255%
PRINT JDPOST%
 130
                                                                                                         !PRINT FIRQB POST MA
              PRINT FNC$(7,9);"Second stats: ";
JDFLG2% = PEEK(JDBPTR% + 10%) AND 255%
PRINT ".MP";
IF JDFLG2% AND 2%**1%
 140
              PRINT "NP";
IF JDFLG2* AND 2***18
PRINT "LO";
IF JDFLG2* AND 2***2*
PRINT "LO";
IF JDFLG2* AND 2***6*
PRINT "KJ";
IF JDFLG2* AND 2***78
                                                                                                         ISECONDARY STATS
              PRINT FNC$(8,9); "Swap in size: ";

JDSIZN% = SWAP%(PEEK(JDBPTR% + 10%)) AND 255%
150
               PRINT JDSIZN%
                                                                                                        PRINT THE SWAP SIZE
              PRINT FNC%(9,4); "Residency quantum: ";
JDRESQ% = PEEK(JDBPTR% + 14%)
PRINT JDRESQ%
160
                                                                                                        PRINT THE RESIDENCY QUANTUM
              PRINT FNC$(10,10); "Memory size: ";
JDSIZE% = PEEK(JDBPTR% + 22%) AND 255%
PRINT JDSIZE%
                                                                                                        !PRINT THE MEMORY SIZ
              180
                                                                                                        !PRINT THE PRIORITY
190
               PRINT FNC$(12,13); FNC$(0,7); FNC$(0,10);
              "Runburst: ";

JDBRST% = SWAP%(PEEK(JDBPTR% + 28%)) AND 255%
PRINT JDBRST%;FNC%(0,9);FNC%(0,8);
                                                                                                        !PRINT THE RUNBURST
```



Some interesting things happen when you sit down at our CIT-161 Terminal.

Like any office worker, video terminal operators are bombarded by a tremendous spectrum of visual and audio stimulae. Add to that their own thought distractions and you begin to see why a single-color display is at a severe disadvantage competing for user attention.

Our CIT-161 Color Alphanumeric Terminal, on the other hand, makes use of some very interesting color phenomena. Which makes a much more efficient terminal...and makes users a lot more productive.

Studies have shown, for instance, that color can improve user response time by 50% — which translates into faster throughput and decreased time to job completion. Data is made more interesting and easy to follow with the use of color, resulting in less error in interpretation.

Color also has broad attention-getting power, affording the added ability to prioritize data by different hues and



create more significant and long-lasting impressions. Not to mention the subliminal conditioned responses we all have to color. (Red, for example, can be used to flag a problem instantly.)

And color dramatically accelerates the learning process. Meaning that training time can be substantially reduced so operators reach full productivity far more quickly. How will our CIT-161 fit into your system? Perfectly! It's an ANSI-compatible direct replacement for the DEC® VT-100 Series with a full range of advanced video features: 132/80-character column display, single-width or double-width/double-height characters, the unique Window Erase feature, split screen, and more.

Sound interesting? Contact C. Itoh Electronics, Inc. today for more information: 5301 Beethoven Street, Los Angeles, CA 90066, (213) 306-6700. Or call ACRO Corp., exclusive representatives: Irvine, CA (714) 557-5118; Houston, TX (713) 777-1640; Cherry Hill, NJ (609) 667-4114; Chicago, IL (312) 992-2346.

C. ITOH ELECTRONICS, INC.

One World of Quality

```
JDB2% = PEEK(JDBPTR% + 8%)
PRINT FNC$(2,55);FNC$(0,7);FNC$(0,10);
"JDB2";FNC$(0,9);FNC$(0,8);
                                                                                                                                                                                                            PRINT "1US ";
IF CHAR% AND 2%**9%
  200
                                                                                                                                                                                                                                                                                                                        ISOME RTS CHARACTERS
                                                                                                                                IGET POINTER TO JDB2
                                                                                                                                                                                          400
                                                                                                                                                                                                            RIB% = PEEK(JDB2%+18%)
                     FOR X% = 4% TO 13%

PRINT FNCS(X%,39%);FNCS(0,7);FNCS(0,10);

FNCS(0,25);"x";FNCS(0,9);

FNCS(0,8);
                                                                                                                                                                                                            RIDS - FEER(UDE/8+108)

GOTO 500 IF RIDS = 08

PRINT FNC$(15%,55%);FNC$(0,7);FNC$(0,10);

"RIB";FNC$(0,9);FNC$(0,8);
  205
                                                                                                                                                                                                                                                                                                                        IRECEIVER BLOCK EXIST
                                                                                                                                                                                                            FOR X% = 17% TO 21%
                     NEXT X%
                                                                                                                                                                                          405
                     PRINT FNC$(3,39);FNC$(0,7);FNC$(0,25);
FNC$(0,10);"1";
STRING$(39%,113%);"k";FNC$(0,8);
                                                                                                                                                                                                                             PRINT FNC$(X%,39%);FNC$(0,7);FNC$(0,10);
FNC$(0,25);"x";FNC$(0,9);
FNC$(0,8);
                    STRING$\(\)37$\),\(\)37$\),\(\)57$\)
FOR X8 = 4% TO 13%
PRINT FNC$\((\)8(\)%\(\)79%\);\(\)FNC$\((\)9(\)7);\(\)FNC$\((\)9(\)7)\);
FNC$\((\)9(\)7\)%\(\)7\);\(\)FNC$\((\)9(\)7\)\);\(\)FNC$\((\)9(\)7\)\),\(\)7\)\(\)7\)\(\)7\)\(\)8\)
                                                                                                                                                                                                            NEAT: X8
FRINT FNCS(16,39);FNCS(0,7);FNCS(0,25);
FNCS(0,10);*l";
STRINGS(398,113%);*k*;FNCS(0,8);
FOR X% = 17% TO 21%
PRINT FNCS(X%,79%);FNCS(0,7);FNCS(0,10);
FNCS(0,25);*x";FNCS(0,9);
FNCS(0,25);*x";FNCS(0,9);
                     NEXT X%
PRINT FNC$(14,39);FNC$(0,7);FNC$(0,25);FNC$(0,10);
"m";STRING$(39%,113%);"j";FNC$(0,9);
                                                        FNC$ (0,8);
                                                                                                                                IDRAW THE PRETTY BOX
                                                                                                                                                                                                                                               FNC$(0,8)
                                                                                                                                                                                                           PRINT FNC$(21,39);FNC$(0,7);FNC$(0,25);FNC$(0,10);
"m";STRING$(39%,113%);"j";FNC$(0,9);
FNC$(0,8);
[D]
                    PRINT FNC$(4,45); "Unposted clock ticks: ";
J2TICK% = PEEK(JDB2%)
PRINT J2TICK%
  210
                                                                                                                                HINPOSTED CLOCK TICKS
                                                                                                                                                                                                                                                                                                                       IDRAW THE PRETTY BOX
                                                                                                                                                                                                          PRINT FNC$(17%,54%);FNC$(0,7);FNC$(0,10);

"Receiver ID: ";

PRINT CHR$(PEEK(RIB% + 2%) AND 255%);
PRINT CHR$(SWAP*(PEEK(RIB% + 2%)) AND 255%);
PRINT CHR$(SWAP*(PEEK(RIB% + 4%) AND 255%);
PRINT CHR$(SWAP*(PEEK(RIB% + 4%)) AND 255%);
PRINT CHR$(SWAP*(PEEK(RIB% + 6%)) AND 255%);
                    PRINT FNC$(5,57);FNC$(0,7);FNC$(0,10);
    "CPU time: ";

J2CPU = PEEK(JDB28+28)
    + 32768*(SWAP8(PEEK(JDB28+108))
  220
                                                                                                                                                                                         410
                                                       AND 255%)
                    PRINT J2CPU; FNC$(0,9); FNC$(0,8);
                                                                                                                                 IPRINT CPU MSB AND LSB
  230
                    PRINT FNC$(6,53); FNC$(0,7); FNC$(0,10);
                    "Connect time: ";
J2CON% = PEEK(JDB2% + 4%)
PRINT J2CON%;FNC$(0,9);FNC$(0,8);
                                                                                                                                                                                                                                                                                                                       IRECETVER ID
                                                                                                                                                                                         420
                                                                                                                                                                                                           PRINT FNC$(18%,54%); "Message Max: ";
S.MMAX% = PEEK(RIB%+14%) AND 255%
PRINT S.MMAX%
                                                                                                                                ICONNECT TIME
                    PRINT FNC$(7,50); "Kilo-core-ticks: ";

J2KCT = PEEK(JDB2%+6%)

+ 32768*(PEEK(JDB2%+10%) AND 255%)
  240
                                                                                                                                                                                                                                                                                                                       !MESSAGE MAXIMUM
                                                                                                                                                                                                           PRINT FNC$(19%,44%);"Pending message count: ";
S.MCNT% = SWAP%(PEEK(RIB%+16%)) AND 255%
PRINT S.MCNT%
                                                                                                                                                                                         430
                    PRINT J2KCT
                                                                                                                                IKCT MSB AND LSB
                                                                                                                                                                                                                                                                                                                       !PENDING MESSAGE COUN
                    PRINT FNC$(8,54); "Device time: ";
J2DEV% = PEEK(JDB2%+8%)
PRINT J2DEV%
  250
                                                                                                                                                                                         450
                                                                                                                                                                                                           GOTO 700
                                                                                                                                IPRINT DEVICE TIME
                                                                                                                                                                                                          PRINT FNC$(15,55);FNC$(0,7);FNC$(0,10);

"RTS";FNC$(0,9);FNC$(0,8);" (default)";

RTS.DEFAULT% = PEEK(JDB2%+16%) | ILOCATE DEFAULT RTS
                                                                                                                                                                                         500
  260
                    PRINT FNC$(9,53);FNC$(0,7);FNC$(0,10);
                   PRINT FNCS(9,3); FNCS(0,7); FNCS(0,10.
"Program name: ";
J2NAME1% = PEEK(JDB2%+12%)
J2NAME2% = PEEK(JDB2%+14%)
PRINT RADS(J2NAME1%) *RADS(J2NAME2%);
FNCS(0,9); FNCS(0,8);
                                                                                                                                                                                         \
                                                                                                                                                                                                         PRINT FNC$(16,39);FNC$(0,7);FNC$(0,25);
FNC$(0,10);"l";
STRING$(39%,113%);"k";FNC$(0,8)
PRINT FNC$(16%+X%,39%);FNC$(0,7);FNC$(0,25);
"x";FNC$(0,8);
FOR X% = 1% TO 7%
PRINT FNC$(24,39);FNC$(0,7);FNC$(0,25);
"m";STRING$(39%,113%);"j";FNC$(0,8);
FOR X% = 1% TO 7%
PRINT FNC$(24,49);FNC$(0,7);FNC$(0,7);
FOR X% = 1% TO 7%
PRINT FNC$(3%+16%,79%);FNC$(0,7);
FNC$(0,25);"x";FNC$(0,8);
NEXT X%
                                                                                                                                                                                        505
                                                                                                              IPROGRAM NAME
                    PRINT FNC$(10,40); "Byt's transfered (buffer): "; J2PCNT% = PEEK(JDB2%+22%)
PRINT J2PCNT%
 270
                                                                                                                               !BUFFER BYTE COUNT
                   PRINT FNC$(11,62);FNC$(0,7);FNC$(0,10);
    "PPN: ";

J2PPN8 = PEEK(JDB2*+24*)

PROJ* = SWAP*(J2PPN*) AND 255*

PROG* = J2PPN* AND 255*

PRINT "[";NUM15(PROJ*);",";NUM15(PROG*);"]";

FNC$(0,9);FNC$(0,8)
 280
                                                                                                                                                                                                         NEXT X% PRINT FNC$(0,7); FNC$(0,9); FNC$(0,8);
                                                                                                                                                                                                                                                                                                                      IDRAW THE PRETTY BOX
                                                                                                                                                                                                         PRINT FNC$(17,53);FNC$(0,7);FNC$(0,10);
"RTS Name: ";
R.NAME1% = PEEK(RTS.DEFAULT% + 2%)
R.NAME2% = PEEK(RTS.DEFAULT% + 4%)
PRINT RAD$(R.NAME1%);RAD$(R.NAME2%);
FNC$(0,9);FNC$(0,8)
                                                                                                                                                                                        510
                                                                                                                               IPRINT PPN
                   PRINT FNC$(12,54);"^T CPU time: ";
J2CPUI% = PEEK(JDB2%+30%)
PRINT J2CPUI%
 290
                                                                                                                               I^T CPU MINI STAT
                                                                                                                                                                                                                                                                                                                       IPRINT DEFAULT RTS
                   PRINT FNC$(15,15);FNC$(0,7);FNC$(0,10);

"RTS";FNC$(0,9);FNC$(0,8);" (current)";
RTS.CURRENT% = PEEK(JDBPTR%+12%)
 300
                                                                                                                                                                                                         PRINT FNC$(18,50); "Default EXT: ";
R.DEXT% = PEEK(RTS.DEFAULT% + 6%)
PRINT RAD$(R.DEXT%)
                                                                                                                                                                                        520
                                                                                                                              ILOCATE CURRENT RTS
                  PRINT FNCS(16,1);FNCS(0,7);FNCS(0,25);
FNCS(0,10);"1";
STRINGS(348,1138);"k";FNCS(0,8)
PRINT FNCS(168+X8,18);FNCS(0,7);FNCS(0,25);
"x";FNCS(0,8);
FOR X8 = 18 TO 78
PRINT FNCS(24,1);FNCS(0,7);FNCS(0,25);
"m";STRINGS(348,113%);"j";FNCS(0,8);
FOR X8 = 18 TO 78
PRINT FNCS(X4,113%);"j";FNCS(0,8);
FOR X8 = 18 TO 78
PRINT FNCS(X8+16%,36%);FNCS(0,7);
FNCS(0,25);"x";FNCS(0,8);
NEXT X8
 305
                                                                                                                                                                                                                                                                                                                      IPRINT DEFAULT .EXT
                                                                                                                                                                                                         PRINT FNC$(19,46);"FIP unit number: ";
FIP.UNIT% = SWAP%(PEEK(RTS.DEFAULT% + 20%))
AND 255%
                                                                                                                                                                                        530
                                                                                                                                                                                        1
                                                                                                                                                                                                          PRINT FIP.UNIT%
                                                                                                                                                                                                                                                                                                                      !PRINT THE FIP UNIT
                                                                                                                                                                                                         PRINT FNC$(20,51); "User count: ";
R.CNT% = PEEK(RTS.DEFAULT% + 26%) AND 255%
PRINT R.CNT%
                                                                                                                                                                                        540
                                                                                                                                                                                                                                                                                                                      IDEFAULT RTS USER
                                                                                                                                                                                                                                                                                                                      1 COUNT
                   NEXT X%
PRINT FNC$(0,7); FNC$(0,9); FNC$(0,8);
                                                                                                                                                                                                         PRINT FNC$(21,45); "Minimum job size: ";
MIN.JOB% = SWAP%(PEEK(RTS.DEFAULT%+28%))
AND 255%
                                                                                                                                                                                       550
                                                                                                                               IDRAW THE PRETTY BOX
                  PRINT FNC$(17,13);FNC$(0,7);FNC$(0,10);
"RTS Name: ";
R.NAME18 = PEEK (RTS.CURRENT% + 2%)
R.NAME2% = PEEK (RTS.CURRENT% + 4%)
PRINT RAD$(R.NAME1%);RAD$(R.NAME2%);
FNC$(0,9);FNC$(0,8)
310
                                                                                                                                                                                                         PRINT MIN.JOB%
                                                                                                                                                                                                                                                                                                                      !MINIMUM JOB SIZE
                                                                                                                                                                                                         PRINT FNC$(22,45); "Maximum job size: ";
MAX.JOB% = PEEK(RTS.DEFAULT%+28%) AND 255%
PRINT MAX.JOB%
                                                                                                                                                                                       560
                                                                                                                                                                                                                                                                                                                      IMAXIMUM JOB SIZE
                                                                                                                               IPRINT CURRENT RTS
                                                                                                                                                                                       570
                                                                                                                                                                                                          PRINT FNC$(23,46); "Characteristics: ";
                                                                                                                                                                                                         PRINT FNC3(25,46); "Characterist
CHAR% = PEEK(RTS.DEFAULT%+30%)
PRINT "KBM";
IF CHAR% AND 2%**8%
PRINT "IUS ";
IF CHAR% AND 2%**9%
                  PRINT FNC$(18,10); "Default EXT: ";
R.DEXT% = PEEK(RTS.CURRENT% + 6%)
PRINT RAD$(R.DEXT%)
 320
                                                                                                                              IPRINT DEFAULT .EXT
                                                                                                                                                                                       1
                  PRINT FNC$(19,6);"FIP unit number: ";
FIP.UNIT% = SWAP%(PEEK(RTS.CURRENT% + 20%))
AND 255%
PRINT FIP.UNIT%
330
                                                                                                                                                                                                                                                                                                                      ISOME RTS CHARACTERS
                                                                                                                                                                                       700
                                                                                                                                                                                                         PRINT FNC$ (0.0):
                                                                                                                                                                                                        GOTO 1500 IF PEEK(JDB2% + 28%) = 0%
SLEEP 3%
PRINT FNC$(0,0);FNC$(0,5);FNC$(1,20);
                                                                                                                              IPRINT THE FIP UNIT
                  PRINT FNC$(20,11); "User count: ";
340
                  R.CNT% = PEER
PRINT R.CNT%
                                      PEEK(RTS.CURRENT% + 26%) AND 255%
                                                                                                                                                                                                                           "Job";JOB.NUMBER%;FNC$(0,7);
FNC$(0,10);"Resident";FNC$(0,9);
FNC$(0,8);
                                                                                                                              ICURRENT RTS USER
                                                                                                                              ! COUNT
                                                                                                                                                                                                        "Library Structures";
WDB% = PEEK(JDB2% + 28%)
COUNTER% = 0%
WDB% - 12%
                  PRINT FNC$(21,5); "Minimum job size: "; MIN.JOB% = SWAP%(PEEK(RTS.CURRENT%+28%))
350
                  AND 255%
PRINT MIN.JOB%
                                                                                                                                                                                                                                                                                                                     PREPARE TO SHOW THE RESIDENT LIBRARIES
                                                                                                                              IMINIMUM JOB SIZE
                                                                                                                                                                                                        PRINT FNC$(3,1);"Lib";FNC$(3,5);"PPN";
FNC$(3,20);"Name";FNC$(3,30);
"Users";
                  PRINT FNC$(22,5); "Maximum job size: ";
MAX.JOB% = PEEK(RTS.CURRENT%+28%) AND 255%
PRINT MAX.JOB%
360
                                                                                                                                                                                      710
                                                                                                                              IMAXIMUM JOB SIZE
                                                                                                                                                                                                        PRINT FNC$(23,6); "Characteristics: ";
CHAR% = PEEK(RTS.CURRENT%+30%)
PRINT "KBM";
IF CHAR% AND 2%**8%
370
                                                                                                                                                                                                                                                                                                                    !HEADINGS FOR LIBS
```

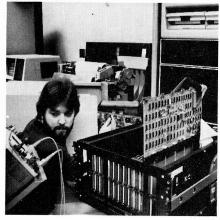
The Single Source for Digital Products Service

DEC Module Repair, Exchange or Replacement

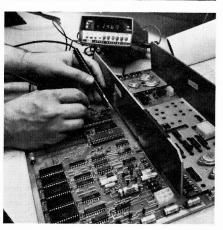
Photos show how we're fully equipped to repair and test DEC,* PDP-11* and TERMINAL MODULES. All modules we repair are 100% inspected.

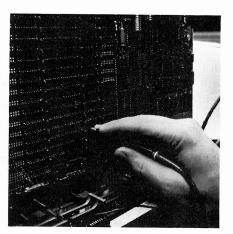
DIGITAL PRODUCTS REPAIR CENTER

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! Tenday 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

 $^\star \text{DEC}$ and PDP are registered trademarks of Digital Equipment Corporation.



RSTSPROFESSIONALRSTSPROFES

```
FOR X% = 2% TO 10% STEP 2%

GOTO 760 IF PEEK(WDB%+X%) = 0%

Z% = Z% + 1%

PRINT FNCS(Z%+4%,1%);

PRINT USING "###",6% - Z%;
                                                                                                                                                                                                                                                                                                                                                                   FILE.LSB = FILE.LSB + 65536
720
                                                                                                                                                                                                                                                                                                                                                                   IF FILE.LSB < 0%
PRINT FNC$(Y1%,64%);
PRINT USING "######",FILE.MSB+FILE.LSB
                                                                                                                                                                                                                                                                                                                                                                   GOTO 1590
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      !FINISH PRINTING FILE
                                                                                                                                                                                                            IPRINT LIB NUMBER
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      ! FCB AND WCB.
                                                         LIB% = PEEK(WDB%+X%)
LIB% = LIB% - 1%
IF LIB% AND 1%
R.NAME1% = PEEK(LIB%+2%)
R.NAME2% = PEEK(LIB%+4%)
730
                                                                                                                                                                                                                                                                                                          1550
                                                                                                                                                                                                                                                                                                                                                                    PRINT USING
                                                                                                                                                                                                                                                                                                                                                                                                 (SWAP% (PEEK (WCB.DDB%+2%))
                                                                                                                                                                                                                                                                                                                                                                    AND 255%);
PRINT FNC$(Y1%,10%);
LEFT(TIME$(PEEK(WCB.DDB%+4%))
                                                          PRINT FNC$(2%+4%,20%);
RAD$(R.NAME1%);RAD$(R.NAME2%); !RESIDENT LIBRARY NAME
                                                                                                                                                                                                                                                                                                                                                                   ,5%);
PRINT FNC$(Y1%,34%); "Device";
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      !PRINT DDB INFORMATIC
740
                                                          L.PPN% = PEEK(LIB%+6%)
L.PROJ% = SWAP%(L.PPN%) AND 255%
L.PROG% = L.PPN% AND 255%
PRINT "[";L.PROJ%;",";L.PROG%;"]";
                                                                                                                                                                                                                                                                                                                                     NEXT X%
PRINT FNC$(0,0);
SLEEP 3%
                                                                                                                                                                                                            IPPN NUMBER
                                                          PRINT FNC$(Z%+4%,30%);
750
                                                                                                                                                                                                                                                                                                        1600
                                                                                                                                                                                                                                                                                                                                     GOTO 090
                                                          R.CNT% = PEEK(LIB%+26%) AND 255%
PRINT USING "###", R.CNT%
                                                                                                                                                                                                                                                                                                       20000 DEF FNC$(L9,C9)
1CURSOR CONTROL FUNCTION
VT-100
                                                                                                                                                                                                           HISER COUNT
760
                             IOB% = PEEK(JDBPTR%)
1500
                          - cursor HOME
- cursor UP
                                                                                                                                                                                                                                                                                                                                                                                           cursor DOWN
cursor RIGHT
                                                                                                                                                                                                                                                                                                                                                                                   - cursor LEFT
- clear EOP
                                                                                                                                                                                                                                                                                                                                                                0,8 - ANSI OFF
0,10 - BOLD ON
0,11 - UNDERSCORE ON
0,12 - BLINK ON
0,13 - REVERSE VIDEO ON
0,14 - DOUBLE HEIGHT TOP 1/2
0,15 - DOUBLE HEIGHT TOP 1/2
0,15 - DOUBLE HEIGHT BOTTOM 1/2
0,16 - SINGLE WIDTH HEIGHT
0,17 - DOUBLE WITH SINGLE HEIGHT
0,18 - LED OFF
0,19 - LED 1 ON
0,20 - LED 2 ON
0,21 - LED 3 ON
0,21 - LED 3 ON
0,22 - LED 4 ON
0,23 - 132 COLUMNS
0,24 - 80 COLUMNS
0,25 - SPECIAL GRAPHICS OF
CHER TARMENT
                                                                                                                                                                                                                                                                                                                                                                   0.9
                                                                                                                                                                                                                                                                                                                                                                                            CHARACTER ATTRIBUTES OFF
                           FOR X% = 0% TO 30% STEP 2%
GOTO 1590 IF PEEK(IOB%+X%) = 0%
WCB.DDB% = PEEK(IOB%+X%)
Y% = X%/2%
Y1% = Y18 + 1%
PRINT FNCS(Y1%,1%);
PRINT USING "##",Y%;
1510
                                                          PRINT USING ** ,18;
PRINT FNCS (Y18,48);
PRINT USING "##",
GOTO 1550 IF PEEK(WCB.DDB%) AND 255%);
                                                                                                                                                                                                            !START LOOP TO SEE
! FILES, GOTO 1550 IF
! A DDB
                                                                                                                                                                                                                                                                                                    Other arguments ,if valid, are assumed to be cursor addresses for direct addressing
                                                          PRINT FNC$(Y1%,22%);
PRINT USING "####",
(PEEK(WCB.DDB%+4%) AND 255%);
1520
                                                                                                                                                                                                            IFIRST PART OF WCB
                                                          PRINT FNC$ (Y1%, 27%);
                                                        MSB.BLOCK%

= SWAP%(SWAP%(PEEK
(WCB.DDB%)+4%) AND 255%)
LSB.BLOCK% = PEEK(WCB.DDB%+6%)
BLOCKI = LSB.BLOCK%
BLOCKI = LSB.BLOCK%
IF BLOCKI < 0%
LSB.BLOCK = LSB.BLOCK
LSB.BLOCK = LSB.BLOCK + 65536
IF LSB.BLOCK < 0%
BLOCKI = LSB.BLOCK + 05536
IF LSB.BLOCK < 0%
BLOCKI = LSB.BLOCK + 70%
BLOCKI = LSB.BLOCK + 10%
BLOCKI = LSB.BLOCK + 10%
BLOCKI = LSB.BLOCK + 10%
BLOCKI = BLOCKI +
1530
                                                                                                                                                                                                           IBLOCK NUMBER
                                                        FCB% = PEEK(WCB.DDB%+8%) - 28%
PRINT FNC$(Y1%,34%);"[";
PRINT USING "###",
(SWAP%(FCB%+4%)) AND 255%);
PRINT ",";
                                                          PRINT USING "###"
                                                          PRINT "]";RADS(PEEK(FCB%+6%));
RADS(PEEK(FCB%+6%));
""."
                                                                                      RADS (PEEK (FCB%+10%));
                                                         RADS(PEEK(FCB%+10%));
PRINT FNCS(Y1%,54%);
PRINT USING "##",
GWAP%(PEEK(FCB%+14%)) AND 255%);
PRINT FNCS(Y1%,60%);PEEK(FCB%+28%);
PRINT FNCS(Y1%,60%);PEEK(FCB%+24%))
AND 255%;
FILE.MSB = SWAP%(SWAP%(PEEK(FCB%+24%))
AND 255%;
FILE.MSB = FILE.MSB + 65536
IF FILE.MSB < 0%
FILE.LSB = PEEK(FCB%+26%)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     !ERROR
```

Word Processing* VAX/VMS, RSTS/E,

*Word-11 by Data Processing Design, Inc. 181 W. Orangethorp Avenue Placentia, CA 92670

RSX-11M

On Track Systems Provides:

- Sales
- Service
- Installation
- Demonstrations
- Training
- Consulting

At your convenience!

At your office!

On Track Systems, Inc.

P.O. Box 245 Ambler, PA 19002-0245 (215) 542-7133

MAXCIN 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. tication. Modular flexibility. Multi-level security.

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.

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

processors to process the data they need. On

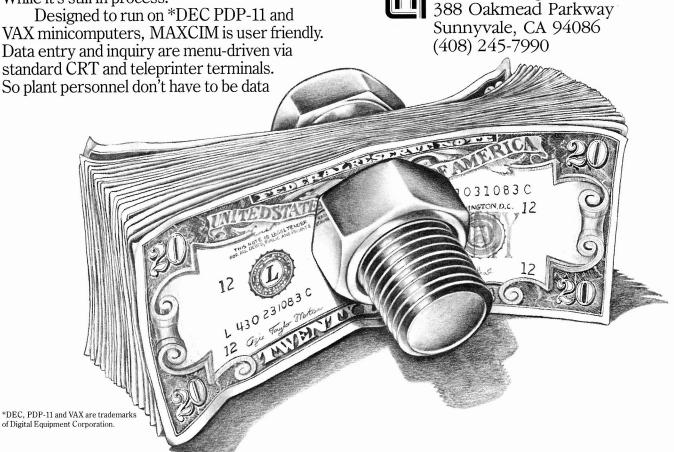
the other hand, vou'll like the system's sophis-

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/R312



Baltimore • Boston • Chicago • Dallas • Houston • Los Angeles • New York • Portland • Sunnyvale USA / Sydney, Australia CIRCLE 52 ON READER CARD

USRDSK Disk Usage Summary Report for RSTS/E

By Philip Hunt, O.L.F.B.P., 6400 E. Broad Street, Columbus, Ohio 43213

USRDSK is a very valuable program designed for the system manager to see, at a glance, usage on selected or all disk devices by ascending PPN. All accounts or accounts with disk storage usage over a certain limit are reported.

Program entry

USRDSK is 'RUN'. A CCL entry point does exist, but processing is the same whether entered by CCL or 'RUN'.

After initialization, the user is requested for an output filename for the USRDSK report, the default is 'USRDSK.RPT'.

The program will then scan the device tables and system-wide logical tables and report the disks found and the number of system-wide logicals in use. The program does not have to be modified for ANY changes in disk structure, it looks up the structure when it executes. System-wide logicals are displayed on the report so the user may get a better understanding of why a certain account has a large disk usage.

The user is then requested for a search block limit for each disk in the system. The default is all usage, but any number may be entered; if it is, usage BELOW that number will not be reported. If a -1 is entered for this limit, the disk will be SKIPPED in the report.

Note: This program MUST be compiled under BP2 as it uses an RMS-INDEXED file to store account information during lookup. It is a '.TMP' file that is deleted.

Installation

BP2

OLD USRDSK COM/OBJ BUILD/IND TKB @USRDSK

Sample Run

RUN [1,3]USRDSK
USRDSK V01.03 RSTS V7.1-11 C OLFBP 11/70
Output report to <USRDSK.RPT> ?

Disks found:

DB0: (System)
DB1: (Private)
DB2: (Private)
DB3: (Private)

There are 53 system-wide logicals

Search block limit for DB0: < All, -1 = Skip > ? 2500 Search block limit for DB1: < All, -1 = Skip > ? 5000 Search block limit for DB2: < All, -1 = Skip > ? 5000 Search block limit for DB3: < All, -1 = Skip > ? 5000

Beginning Account lookup

Processing device — DB0: Processing device — DB1: Processing device — DB2: Processing device — DB3:

Ready

Questions, or comments:

If there are any questions or comments, you may write, or call me at the below address. If you would like a copy of this program on TAPE, please send a tape and \$15.00 to the below address, specify format and density for your tape. Philip Hunt, O.L.F.B.P., 6400 E. Broad Street, Columbus, OH 43213

			ective Disk Ha STS V7.2-04 A	OLFBP 11/70	t		
PP#	DBO: (System) >02500 Used/Quota	DB1: (Private) >05000 Used/Quota	DB2: (Private) >05000 Used/Quota	DB3: (Private) >05000 Used/Quota	Comments		
[000,001]	13264/0			2			
[001,002]	32016/0						
[001,003]	8280/0						
[001,006]	3312/0						
[001,010]	7120/0	•					
[001,011]	14160/0				SY:=LB		
[001,018]	4296/0	-		-			
[001,021]	2880/0	-	-	-			
[001,035]	3648/0			•			
[001,040]			-	11472/0			
[001,050]	5264/0			-			
[001,060]		11568/0	•	-			
[001,062]	7800/0	10584/0					
[001,102]	7800/0 3672/0		-	-	DBO:=DTR		
[001,108]	4072/0						
[001,110]	2904/0			-	DBO: =C		
[001,116]	2856/0						
[001,135]	7120/0						
[002,022]	3096/0						
[002,032]	3192/0	_					
[002,038]	4376/0	5688/0					
[002,042]	3272/0	-					
[002,078]	3128/0	-					
[002,080]	•			5064/0			
[002,092]	6824/0	-	-				
[002,098]		8856/0	6656/0	-			
[002,102]		-		42352/0	DB3:*PM		
[002,103]	-	-		>=65K/0	DB3:=PD		
[002,107]		-	>=65K/0		DB2:=CD		
[002,108]			-	8032/0	DB3: = CW		
[002,110]	43208/0	21248/0	-	•	DB1: *BD		
[002,112]	43200/0		•		DBO:=PH		
[002,113]		18432/0	-	60160/0	DB3:=CH		
[002,115]		1043270	14432/0		DB1:=PI		
[002,117]	8192/0		1443270		DB2:=SC DB0:=FM		
[002,119]	21,72,70	18400/0			DB3:=CB2	DB1:=CB1	DB2:=CB
[003,022]	3088/0				DB3:=CB2	DBITTEBI	DB2:=CB
[003,023]	2768/0						
[003,028]		5264/0 -					
[003,032]	3088/0						
[003,038]		7592/0					
[003,125]			5992/0		DB0:=CLHO	DB3:=CLM1	DB2:=CLM2
[004,058]	-	6032/0				2031-02111	DDLIGGERE
[004,068]	3888/0	-	-	-			
[004,091]		6200/0	-	-			
[004,092]	3824/0	-	-	-			
[004,093]		5344/0	-				
[004,098]	7080/0	5280/0	26632/0	-			
[004,121]		-	13824/0	-	DB2:=PMF	DB1:=PP	



2 on error goto 19000 &

YOU'RE NO SOFT TOUCH WHEN IT COMES TO SOFTWARE.



You're smart enough to buy only what you need—and we've unbundled all of MCBA™ manufacturing software so you can afford everything you want!

- GUARANTEED PERFORMANCE
 ONE-YEAR UPDATE SERVICE
 THOUSANDS OF INSTALLATIONS NATIONWIDE

In addition, INSTALLATION, TRAINING, CUSTOMIZATION, EXCLUSIVE SOFTLINE™ TELEPHONE DIAGNOSTIC SUPPORT, and other services are available for a fee. Buy only as much additional support as you need.



CALIFORNIA SYSTEMS ASSOCIATES

2845 Mesa Verde Drive East, Suite Four, Costa Mesa, California 92626 For demonstration call: (714) 546-9716

RSTS™ Versions Available Now!

California Systems Associates	Unbun	dled Prices	Are A Steal!
ACCOUNTING PACKAGES:		LIST PRICE	OUR PRICE
MCBA General Ledger	v6-1	\$2,000	\$1,350
MCBA Accounts Receivable	v6-1	2,000	1,350
MCBA Accounts Payable	v6-1	2,000	1,350
MCBA Payroll	v6-1	2,000	1,350
MCBA Fixed Assets and Depreciation	v3-0	2,000	1,350
DISTRIBUTION PACKAGES:			
MCBA Customer Order Processing	v1-1	3,000	1,850
MCBA Bill of Materials	v1-1	3,000	1,850
MCBA Inventory Management	v1-1	×3,000	1,850
MCBA Purchase Order/Receiving	v1-1	3,000	1,850
MANUFACTURING PACKAGES:			
MCBA Job Costing	v1-1	2,500	1,650
MCBA Base Materials		0.500	4.050
Requirement Plan	v1-1	2,500	1,650
MCBA Shop Floor Control	v1-1	4,000	2,750
PROFESSIONAL PACKAGE:		0.000	4 000
MCBA Professional Time & Billing		2,000	1,600
MCBA Accountant's Client Write-up		2,000	1,600
WORDWRIGHT wordprocessing		795	750
DATAMANAGE/REPORTWRIGHT		1,600	1,250

MCBA is a registered trademark of Mini-Computer Busines RSTS, DIBOL, DEC, PDP-11 are registered trademarks of Digital Equip

CIRCLE 163 ON READER CARD

RSTSPROFESSIONALRSTSPROFESSIONA

```
DIM M$(30) &
! MONITOR TABLES - PART I &
! M$(3) NULINE &
. M4(1) MAXCHT &
                                                                                                                                                                                                                                                                                                                                        DSK$=DSK$+1$ &
GOTO 32000 IF DSK$>.MAX.DSK$ &
DSK.TYP$(DSK$) = "P" &
DSK$(DSK$) = DEV$ &
                                                                                                                                                                                                                                                                                          ELSE
                            M$(4)
M$(5)
M$(7)
M$(9)
M$(11)
                                                                     DEVCNT & DEVPTR & MEMLST & JOBTBL &
                                                                                                                                                                                                                                                                                                                                        DSK.UNT$(DSK$) = UNIT$ &
                                                                                                                                                                                                                                                                                                                UNTCNT%=UNTCNT%+2% & SATCTL%=SATCTL%+2% & SATCTM%=SATCTM%+2% &
                                                                                                                                                                                                                                            1100
                            M$(11)
M$(13)
M$(15)
M$(17)
M$(19)
M$(21)
                                                                      JBSTAT
                                                                     JBWAIT
UNTCLU
UNTCNT
                                                                                                                                                                                                                                                                                          NEXT UNITS &
                                                                                                                                                                                                                                                                     NEXT DISK.TYP$
                                                                     SATCTL &
                                                                                                                                                                                                                                                                   PRINT "Disks found:" \ PRINT &
                                                                                                                                                                                                                                            1900
                            M$(23)
M$(25)
                                                                     JSBTBL & SATCTM &
                                                                                                                                                                                                                                            2000
                                                                                                                                                                                                                                                                   FOR X$=1$ TO DSK$ &
                                                                                                                                                                                                                                                                                          PRINT DSK$ (X$)+NUM1$(DSK.UNT$(X$))+":";TAB(8$); &
PRINT "(System)" IF DSK.TYP$(X$)="S" &
PRINT "(Private)" IF DSK.TYP$(X$)="P" &
PRINT "(Unknown)" IF DSK.TYP$(X$)<"S" and dsk.typ$(x$)<"P"
                       DIM MO$(30) &
! MONITOR TABLES - PART II &
                            MO$(3)
MO$(5)
                                                                     FREES &
                                                                     DEVNAM &
                                                                      CSRTBL &
                                                                                                                                                                                                                                                                   NEXT XS &
                       ! MO$(7)
! MO$(9)
! MO$(11)
! MO$(13)
! MO$(15)
! MO$(17)
! MO$(19)
! MO$(21)
! MO$(23)
! MO$(25)
                                                                                                                                                                                                                                                                   print "There are";LGCNT$;" system-wide logicals" &
                                                                      DEVOKE
TTYHCT
JOBCNT
                                                                     RTSLST &
                                                                                                                                                                                                                                                                   FOR X$=1$ TO DSK$ & dev$=dsk$(x$)+num1$(dsk.unt$(x$))+":" & PRINT "Search block limit for ";DEV$;" <All, -1=Skip> "; & INPUT BLMT(X$) &
                                                                                                                                                                                                                                            2100
                                                                      ERRCTI. A
                                                                     SNDLST
LOGNAM
DEVSYN
                                                                                                                                                                                                                                                                   DSK.SKIP%(X%) = 0% IF BLMT(X%)<> -1% & NEXT X% &
                       DIM READ.ACCT$(30$) & I USED TO HOLD ACCOUNTING DATA &
920
                                                                                                                                                                                                                                           2990
                                                                                                                                                                                                                                                                   open temp.file$ as file #1% &
                                                                                                                                                                                                                                                                   close #1% & kill temp.file$ &
                       1 &
999
                                                                                                                                                                                                                                           3000
                                                                                                                                                                                                                                                                    ILETS DO IT
                                                                                                                                                                                                                                                                  PRINT &
PRINT "Beginning Account lookup" &
                                                                        CODING
                                              MAIN
                                                                                                                  AREA&
                                                                                                                                                                                                                                                                  PRINT &
open temp.file$ as file #1$, &
organization indexed fixed, &
access modify, allow none, &
primary key ppn$, &
filesize 200$, &
map temp, temporary &
map (temp) ppn$=6$, blocks(16$), quota(16$) &
                       ID.STG$="01.03" &
! SET UP VERSION/EDIT NUMBER. &
1000
                      1010
                                                                                                                                                                                                                                                                   COMPRESS THE DISK TABLE IF NEEDED
                                                                                                                                                                                                                                           3005
                                                                                                                                                                                                                                                                ICOMPRESS THE DISK TABLE IF NEEDIDONTS = 05 &
FOR X$=1$ TO DSK$ &
GOTO 3007 IF DSK. SKIP$(X$) &
DCNT$=DCNT$+1$ &
SELECTIVE$ = "Y" &
GOTO 3007 IF DCNT$ = X$ &
DSK$(DCNT$) = DGK$(X$) &
BLHT(DCNT$) = BLHT(X$) &
DSK. UNT$(DCNT$) = DSK. UNT$(X$) &
DSK. TYP$(DCNT$) = DSK. UNT$(X$) &
DSK. SKIP$(DCNT$) = DSK. TYP$(X$) &
DSK. SKIP$(DCNT$) = 05 &
DSK. SKIP$(DCNT$) = 05 &
                                              ERT$(0%) &
1 PRINT HEADER &
                       LINPUT "Output report to <USRDSK.RPT> ";RPT.FILE$ & rpt.file$="USRDSK.RPT" IF RPT.FILE$="" &
 1015
                       OPEN RPT.FILE$ FOR OUTPUT AS FILE 2$ & GOTO 1020 &
 1016
                       !ERROR & PRINT "?Open error on report file" & goto 1015 &
1017
                                                                                                                                                                                                                                           3007
                                                                                                                                                                                                                                                                 NEXT X$ & DSK$ = DCNT$ &
                      CHANGE SYS(CHR$(6$)+CHR$(-3$)) TO M$ & \ M$(I$)-M$(I$)+SWAP$(M$(I$+1$)) FOR I$=5$ TO 25$ STEP 2$ & \ CHANGE SYS(CHR$(6$)+CHR$(-12$)) TO MO$ & \ MO$(I$)-SWAP$(M$(I$+1$)) FOR I$=5$ TO 25$ STEP 2$ & \ CHANGE SYS(CHR$(6$)+CHR$(-12$)) TO SY$ & \ CHANGE SYS(CHR$(6$)+CHR$(26)) TO SY$ & \ JOBNO$ = SY$(1$)/2$ & \ JOBNO$ = NUM1$(JOBNO$) & \ JOBNO$ = NUM1$(JOBNO$) & \ JOBNO$ = NUM1$(JOBNO$) & \ TEMP.FILE$ = "TEMP*+JOBNO$*-".TMP" & \ TEMP.FILE$ = "TEMP*+JOBNO$*-".TMP" & \ I GET MONITOR TABLES , SETUP TEMP FILE NAME &
                                                                                                                                                                                                                                                                 for x$=1$ to dsk$ & dev$=dsk$(x$)+num1$(dsk.unt$(x$))+":" & print "Processing device - ";dev$ &
                                                                                                                                                                                                                                           3010
1030
                                                                                                                                                                                                                                                                                       3020
 1039
                                              PUBLIC STRUCTURE&
                                                                                                                                                                                                                                                                                         writ% = 0% &
                                                                                                                                                                                                                                          3030
                                                                                                                                                                                                                                                                                        get #1%, key#0% eq ppn$ & goto 3050 &
                                                                                                                                                                                                                                                                                        Irec not found here & blocks(dd$)=-1$ for dd$=1$ to 16$ & quota(dd$)=-1$ for dd$=1$ to 16$ & writ$ = -1$ &
                                                                                                                                                                                                                                          3040
 1040
                             UNTCNT$ = M$(19$) &
                        \ SATCTL$ = M$(21$) & 
\ SATCTM$ = M$(25$) &
                       FOR DISK.TYP$=0$ TO M0$(9$)-2$ STEP 2$ & DEV$=CVT$$(SWAP$(PEEK(M0$(5$)+DISK.TYP$))) & FOR UNIT$=0$ TO PEEK(M$(5$)+DISK.TYP$) & J$=PEEK(UNTCNT$) &
 1050
                                                                                                                                                                                                                                                                                       3050
                                                                    J$=PEEK(UNTONT$) &

IF J$<0$ OR (J$ AND 20480$)<>0$ THEN &
1060 &

DSK$=DSK$+15$ &

GOTO 32000 IF DSK$>.MAX.DSK$ &

DSK.TYP$(DSK$) = "S" &

DSK$(DSK$) = DEV$ &
                                                                                                                                                                                                                                                                                       next y$ & print "?Too many accounts on ";DEV$;" - ABORTING" & goto 32767 &
                                                                                                                                                                                                                                         3060
                                              ELSE
                                                                                                                                                                                                                                         3100
                                                                                                                                                                                                                                                                next x%
                                                                                            DSK.UNT%(DSK%) = UNIT% &
                                                                                                                                                                                                                                         3110
                                                                                                                                                                                                                                                                !table is now loaded, lets print it & lent \$=0 % &
                                                                    UNTCNTS=UNTCNT$+2$ &
SATCTL$=SATCTL$+2$ &
SATCTM$=SATCTM$+2$ &
1060
                                                                                                                                                                                                                                                                gosub 5000 &
find #1%, key#0% ge "000000" &
                            NEXT UNITS & NEXT DISK.TYP$ &
                                                                                                                                                                                                                                         3120
                                                                                                                                                                                                                                                               å
1079
                                             PRIVATE PACKS&
                                                                                                                                                                                                                                         3130
                                                                                                                                                                                                                                                                for dd%=1% to dsk% &
                                                                                                                                                                                                                                                                                      =1% to dsk% & GOTO 3131 IF DSK.SKIP$(X%) & blocks = fnspc$(blocks(dd$),5%) & quota$ = num1$(quota(dd$)) & dquota$ = quota$ + squota$ + s
                      UNTCNT$=M$(19$) & \ SATCTL$=M$(21$) &
1080
                    3131
                                                                                                                                                                                                                                                               next dd% &
                                                                                                                                                                                                                                                               print #2%, " "; & for xx%=1% to lgent% &
```

PDP-11,VAX-11, LSI-11 users...

don't buy a file transfer software package unless you know the answers to these important questions...

Hardware Requirements
Does the package require any special communications hardware?

A No. XOREN IPL-11 uses a link between two standard asynchronous terminal ports.

Local and Remote Links
Can the package link computers on either the same site or different sites?

Yes. For a local link a direct line may be used; for a remote link, XOREN IPL-11 uses a telephone line and modems or acoustic couplers.

Error Checking
Does the package include full
communications error checking?

A Yes—and XOREN IPL-11 recovers from errors by re-transmitting *only* the blocks affected.

Operating Systems Supported What operating systems are supported by the package?

Versions of XOREN IPL-11 are available for most DEC operating systems, including RSX-11M, RSX-11M PLUS, RT-11, CTS-300, RSTS/E, CTS-500, VAX/VMS, P/OS, TSX PLUS.

Linking Different Operating Systems
Is it possible to link different operating
systems and transfer files between them?

A Yes, all versions of XOREN IPL-11 are compatible, so that, for example, files can be transferred from a RSTS/E system to an RSX-11M system.

Software Licence Requirements
Is the package written in a high-level language, for which the user must have a licence?

A No, all versions of XOREN IPL-11 are written in assembly language, thereby maximising processing efficiency.

Types of File Handled
Can the package transfer binary files
(object and task files) as well as source
files?

A Vec

Program Size
How much memory does the package

Memory requirements for XOREN IPL-11 are approximately 15K words for RT-11, CTS-300 and TSX PLUS versions, 18K for RSTS/E and CTS-500 versions, and 21K for RSX-11M, RSX-11M PLUS, VAX/VMS versions. (For RSX-11M and RSX-11M PLUS a 15K 'compact' version is also available).

Control of Transfers

Can file transfers in either direction be controlled and monitored from either computer?

f A Yes.

Multi-Channel Interface Cards
Can the package use for the link a port on a DZ11 or DH11 interface card, as an alternative to a single port device, such as a

A Yes: Except for the RT-11, CTS-300 and TSX PLUS versions, XOREN IPL-11 may use any port supported by the operating system's standard terminal driver. (Under TSX PLUS, only a DL11 or equivalent may be used. Under RT-11/CTS-300, a DZ11 may be used instead, but other users will be affected).

Transparency to other users
When the package operates under multiuser operating systems, can other jobs
continue running unaffected?

A ye

Automatic Modes of Operation
In addition to normal interactive operation, can the package operate in batch mode or under the control of an indirect command file?

Yes, all versions of XOREN IPL-11 have a facility of this kind.

Remote Activation
Can a link be established without an operator at the remote site?

Yes, provided that the remote computer is running a multi-user operating system.

File attributesAre file attributes handled correctly?

A Yes, XOREN IPL-11 transfers file attributes with the file wherever applicable.

File FormatsWhen files are transferred between operating systems which use different file formats, are facilities provided for the necessary format conversions?

Yes. From this point of view, the supported operating systems fall into two groups. The FILES-11 group are RSX-11M, RSX-11M PLUS, and VAX/VMS, while RT-11, CTS-300, TSX PLUS, RSTS/E and CTS-500 form the RT-11 group. Format conversion is only necessary when files are transferred between these two groups. It may be performed using a standard DEC utility such as FLX, or using XOREN FLC-11, a special-purpose utility which has the advantage that it can handle RT-11 files on FILES-11 volumes.

File Transfer Rates

How long do file transfers take?

This depends on such factors as the CPU model, operating system, and line speed. Typical transfer rates for XOREN IPL-11 (based on 512-byte blocks) are 40 to 60 blocks per minute at a line speed of 9600 baud (but 20 to 25 for RSTS/E and TSX PLUS), and 10 to 12 blocks per minute at 1200 baud.

CPU Use
How much CPU time does the package

Tests on a PDP-11/34 show that XOREN IPL-11 uses less than 2% of CPU time at a line speed of 1200 baud, and less than 30% at 9600 baud.

Future Support
Is provision made for the supply of future enhancements, and new releases of the package to run under new versions of operating systems?

f A Yes.

XOREN IPL-11 is a low-cost, easy to implement package for transferring files between two DEC PDP-11, LSI-11 or VAX-11 computers even when they have different operating systems.

A licensed copy of the package run at each computer enables files to be transmitted over direct or telephone links using standard asynchronous terminal ports.

XOREN IPL-11

XOREN IPL-11, the simple solution to **DEC** file transfers.

UK Office

Xoren Computing Limited 28 Maddox Street London W1R 9PF Telephone (01) 629 5932

USA Distributors

Softpak Associates 634½ Venice Boulevard Marina Del Rey California 90291 Telephone (213) 822 1830 **USA** Office

Xoren Computing Suite 800 50 Staniford Street Boston MA 02114 Telephone (617) 720 1096

Midcom Corporation 1940 N Tustin, Suite 117 Orange California 92665 Telephone (714) 998 6041



CIRCLE 100 ON READER CARD

* DEC, PDP-11, LSI-11, VAX-11, DL-11, DZ-11, RT-11, DH-11, RSX-11M, RSX-11M PLUS, CTS-300, RSTS/E, CTS-500 and VAX/VMS are trademarks of Digital Equipment Corporation. TSX PLUS is a trademark of S&H Computer Systems.

Dreaming of

Electronic Mail

Product Name: Dreams Version 5.0

Since its first sale in 1979 Dreams has grown in capability and user acceptance. It is now in use on over 40 RSTS/E systems around the country.

Special Features:

- DECnet compatibility—message transmission to distant nodes. Includes message queuing to unavailable nodes.
- Invoke your favorite style of editing (EDT, DECword, WORD-11, TECO, etc.) with a smooth transition to and from the editor.
- · Flexible method for accessing and maintaining multiple mail files
- · Subjects for mail files as well as individual messages.

Retract unread messages.

Recover your last deleted message.
Specify times as well as dates in relative or absolute form to control message appearance or expiration or to narrow selection criteria.

 Full compatibility with Batch. This opens up a world of possibilities for keeping abreast of unattended operations and for implementation of a repetitive reminder system based on day of the week or other longer intervals.

 System manager may assign defaults for accounts, projects, and the entire system including the assignment of certain privileges.

The Dreams package consists of over 40,000 lines of source code in more than 70 modules plus significant documentation both as documents and as on-line help. CSPCOM or BASIC-Plus-2 builds these sources into only 5 Dreams tasks: TELL, MAIL, WHO, SMASH, and MAN-AGE (plus POSTMN for the DECnet version). Computers with sufficient memory may use the customized resident library and resident run-time system.

A VAX version will be available later.

Typical Electronic Mail Features are also included in Dreams:

Send to names, nicknames, or groups.

- Scan, reply, forward, or store for later appearance.
- New, old, priority, or suppressed messages for each mail file.
- · Automatic routing of messages.
- Many other convenient features.

Ordering information:

Available on 9 track 800 or 1600 BPI tape. Multiple CPU discount schedule:

First License Dreams/5 DECnet modules Second and Third License Fourth and Fifth License Sixth and up	\$3000.00 \$3000.00 40% Discount 50% Discount 70% Discount
	70% Discount Additional 50% off the total

Maintenance and new releases: Annual fee of 12% of current list price after the first year.

For more information contact:

Tom Burtnett DCXX Software Services Dickinson College Computer Center Carlisle, PA 17013 717-245-1513

RSTS/E, VAX, DECnet, and DECword are trademarks of Digital Equipment Corporation. WORD-11 is a trademark of Data Processing Design, Inc.

```
3135
           print #2% &
lent% = lent%+1% &
            if lcnt$ > 50$ then gosub 5000 &
3140
            goto 3120 &
           lend of temp file, lets finish up &
print #2%, chr*(12%); &
close #1%, #2% &
goto 32767 &
4000
           5000
            leop routine &
           PRINT #2$, ">";LM$; &

NEXT T$ &

print #2$ &

for t$=1$ to dsk$ &

print #2$, tab(14$+((t$-1$)*14$));" Used/Quota"; &

NEXT T$ &

DRINT T$ &
            PRINT #2%, " Comments"; & print #2%, CR+LF &
5100
           7000
7005
                        D$ = D$ + 1 0$
7030
            return &
 19000
            1 &
                                           HANDLING &
            resume 3040 if erl=3030 and err=155$ & resume 3100 if erl=3020 and err=5$ & resume 32767 if erl=1015 and err=11 & resume 32767 if erl=2100 & resume 1017 if erl=1016 & resume 3000 if erl=2990 and err=5$ & resume 4000 if erl=3120 and (err=11$ or err=155$) &
 19001
            print "?usrdsk - fatal error #";err;" (";ERT$(err);") at ";ERL &
 19900
             goto 32767 &
            def fnpad$(arg$,lg$) &
  pdl$ = num1$(arg$) &
  pdl$ = len(pdl$) &
  pdl$ = len(pdl$) &
  pdl$ = left("000",lg$-pdl$)+pdl$ &
  fnpad$ = pdl$ \ fnend &
 20000
            def fnspc$(arg,lg$) &
pd1$ = num1$(arg) &
pd1$ = len(pd1$) &
pd1$ = left(" ",lg$-pd1$)+pd1$ &
             fnspc$ = pdl$ \ fnend &
            def fnspe$(arg,lg$) &
pdl$ = numi$(arg) &
pdl$ = len(pdl$) &
pdl$ = pdl$+left(" ",lg$-pdl$) &
fnspe$ = pdl$ \ fnend &
             PRINT "?Too many disks - aborting" \ GOTO 32767
 32767
```

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:



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

EDITING SYSTEMS

David Spencer, Infinity Software Corporation

Some EDT Wishes Granted

Last issue I noted that there is a rather limiting problem with the EDT implementation for PDP-11s. This problem is an outgrowth of the difference between VAX (where EDT grew up), a machine with nearly unlimited address space, and the PDP-11 with a limit of 64K bytes of addressing space.

Not one to let a problem just sit, I invested part of my Labor Day weekend and looked into the problem.

1.0 THE PROBLEM

I disassembled the modules that request additional memory for storing new key definitions and buffer names. As noted before, it is precisely true that there is a fixed block of memory that simply gets appended to. When EDT runs out, that's it. (Apparently VAX must just keep asking for memory and expanding dynamically.)

2.0 THE SOLUTION

I don't have the perfect solution, which would have been to write a garbage collect routine on the work block. That's a little out of the scope of what I can handle. What I have done, though, is locate the work block and make it larger.

The following object module can be entered by hand (it's short, really) and used to create an EDT task with more work space. Notice the comment where I note the old values for the work space and use the new . . .

```
.TITLE
          IODATA
                       , < Disassembled EDT data block > ,07,DEC/DJS,05-Sep-82
.IDENT
          /02.01A/
          IODATA
.PSECT
IO.FNF
          = = 5
IO.BUF::.BLKB
                       400
IO.STS::.WORD
                       0
IO.RFM::.WORD
                       0
SY AUD ..
SY.COM::.WORD
                       1020
        .BLKB
SY.PRI::.WORD
                       6
       .BLKB
                       1020
SY.ALT::.WORD
                       10
                       1020
        .BLKB
IO.RHB::.BLKB
                       1
                       0
IO.WF.::.WORD
IO.LPR::.BYTE
                       0
IO.PRT::.BYTE
                       0
                       0
WF.SIZ::.WORD
WF.IFI::.WORD
                       0
IONAME::.WORD
                       0
IONLEN::.WORD
                       0
                       40
JOUNAM::.BLKB
                       /BUG CHECK / ; Those are three spaces . . .
S.BUGC::.ASCII
                                         ; Stock value is 1750
S.HEAP::.BLKW
                       4230
S.HPOI::.WORD
                       4230
                                         ; Stock value is 1750
                       0
S.WORK::.WORD
WC.TIM::.WORD
                       0
WC.BUK::.BLKB
                       20
WC.REF::.BLKB
                       20
WC.DIR::.BLKB
                       10
```

WC.CAC::.BLKB 10000
G.MCR ::.WORD 24577
.BLKB 120
G.LEN ::.WORD 0

.END

In my version of the IODATA module, the work block (called S.HEAP) has been increased from 1000 (base ten) words to 2200 (base ten) words. This increases the size of all versions of EDT, flat, overlaid, or shared, by 1K word. In other words, flat EDT becomes 28K, overlaid 20K, and shared still has a 21K resident library, but the task increases to 8K.

Please note that I have not solved the problem of running out of work space; I have just made it harder for the problem to happen.

To create a new copy of EDT using IODATA.MAC, use the following general steps:

- Create IODATA.MAC from the above code with your favorite editor.
- 2. Assemble IODATA with \$MAC.TSK like this: RUN \$MAC MAC>IODATA = IODATA MAC> †Z
- 3. Verify the module was entered correctly with a checksum from PAT.TSK.

RUN \$PAT

PAT > NL: = IODATA/CS:056067,IODATA

PAT>†Z

If the checksum doesn't match then go back and review your entry of the IODATA module.

- 4. Get out your RSTS distribution and restore EDT????.??? to the current account. On the 7.1 distribution it is on the second of the three tape set of CUSPs.
- 5. Replace the IODATA module in the EDTLIB library: $_{\mbox{\scriptsize RUN}}$ \$LBR

LBR > EDTLIB/RP = IODATA MODULE "IODATA" REPLACED

LBR>1Z

6. There are three choices of command files for building EDT. They are:

EDTBLD for a flat-out task

EDTOVB for an overlaid task

EDTSHR for a task with a resident library

Task build with the configuration type you desire (my favorite is shared, used as the example here):
RUN STKB

TKB>@EDTSHR

 Add the symbol table to allow symbolic patching (and to create the resident library if selected).
 RUN \$MAKSIL

MAKSIL V7.1-11 RSTS/E V7.1-11 Timesharing

Resident Library name? EDT

Task-built Resident Library input file < EDT.TSK>? EDT.TSK

Include symbol table (Yes/No) < Yes>? YES Symbol table input file < EDT.STB>? EDT.STB Task Image SIL output file < EDT.SIL>? EDT.SIL EDT built in 8 K-words, 460 symbols in the directory

EDT.SIL renamed to EDT.SIL < 104 >

RSTSPROFESSIONALRSTSPROFESSIONA

8. Copy the task to the system account with a protection of "<104>". If the shared version was selected, copy the resident library to the account required and name it with a protection code of "<42>" and add it at the right address with UTILITY.

Before making the final transfers of the new copy of EDT, save the old one for a while until you are absolutely sure that the IODATA module was entered correctly.

If the previous instructions were confusing or too general for you, please don't attempt to perform them. Get somebody who can. I am not trying to insult anybody's intelligence. It's just that performing the installation wrong can screw things up for a while and is hard to live down afterwards.

3.0 ANOTHER WISH GRANTED

A number of issues ago I listed my EDT wishes. One of the things I wanted was the journal file created on the current account. Currently EDT creates the journal file on the disk and account of the input file to be edited. The following patch file, called "JOUPAT.MAC", always creates the journal file on the current account on the public structure, otherwise known as "SY:".

This patch is for the EDT released with RSTS V7.1, and beyond. It will not work for the version 7.0 EDT release, which is different. If you are still running RSTS 7.0, there is no problem with installing and running the RSTS 7.1 release of EDT.

```
,<Patch for journal file>,07,DEC/DJS,05-Sep-82
.TITLE
         IOMOD
         /02.02B/
.IDENT
.GLOBL
         BL$MOV
         SY.AUD
.GLOBL
         IOCOD
PSECT
SSSS. = .
. = $$$$. + 000210
                      TSTJOU
         CALL
. = $$$$. + 000566
                      TSTJOU
         CALL
. = $$$$. + 003242
                       J.COPY
         CALL
= $$$$, + 003376
                      (R2),406
TSTJOU: MOVB
         CMPB
                       @#406,SY.AUD
                      ITS.OK
         BNE
                       @#432,#"NL
FO.CLR: CMP
       BE<sub>0</sub>
                       ITS.OK
                       @#410
       CLR
                       @#432
       CLR
       CLR
                       @#434
ITS.OK: RETURN
J.COPY: CALL
                       FQ.CLR
                       BL$MOV
       JMP
.END
```

To install this patch, perform these general procedures:

- 1. Create JOUPAT.MAC from the listing above with your favorite editor.
- 2. Extract the IOMOD module to be patched from the EDT object library.

 RUN \$LBR

RSTS ARCADE

Infinity Software has arcade quality games for VT52 and VT100 terminals.

BLKADE Up to a total of eight players try to fence in their opponents in this game of skill and strategy.

PACKER Collect the radio-active dust from the maze before the four robots catch you. (VT100 only.)

INVADE Protect the planet with your laser base from the alien invaders from space. (VT100 only.)

SUBS Locate and sink your opponent's submarine before he sinks you.

STRWRS You have five minutes to destroy the Death Star before it destroys your moon base.

WIGGIT Drop water crystals in front of the Martian Wiggits before they clone and capture you! (VT100 only.)

Games are \$39.95 each and come complete with User Guides. Distribution is made on 800 or 1600 BPI magtape (other media arranged at higher cost). California residents please add 6½ percent.

GIVE A GAME FOR CHRISTMAS!

For ordering information, please write or call:

INFINITY SOFTWARE CORPORATION

2210 Wilshire Blvd, Suite 801 Santa Monica, California 90403 (213) 820-2702

CIRCLE 87 ON READER CARD

LBR > IOMOD = EDTLIB/EX:IOMOD LBR > \uparrow Z

 Next, assemble the patch file. RUN \$MAC.TSK MAC>JOUPAT=JOUPAT MAC>1Z

4. Then merge the two modules to perform the patch. RUN \$PAT.TSK

PAT>NEWMOD=IOMOD/CS:001507,JOUPAT/CS:023007 PAT>1Z

If you got a checksum error, stop here. Either there was a patch to the IOMOD module previously (unlikely unless you did it yourself), or the JOUPAT source was entered incorrectly. Go back and review your typing and try again until there are no errors.

5. If all is well, replace the corrected module back into the EDT object library.

PIONEER COMMAND

A COMPREHENSIVE MANAGEMENT SYSTEM FOR MANUFACTURERS AND DISTRIBUTORS

Comprising:

- ★ Sales Order Processing
- ★ Accounts Receivable
- ★ Purchase Order Processing
- ★ Accounts Payable
- ★ General Ledger
- ★ Inventory Control
- ★ Bill of Material Processor
- ★ Material Requirements Planning
- ★ Production Control
- —Works Order Processing & Job Costing
- -Routings
- -Work Centre Loading

Designed for Timeshare and Turnkey installations on PDP 11 RSTS Computers

Based on the NCCS USER 11 Data Management Systems

Installations in
UK, USA, Australia and New Zealand
Local Dealers wanted Worldwide



4 Albion Place Northampton NN1 lUD - U.K. Tel:Northampton (0604) 39096

CIRCLE 149 ON READER CARD

RUN \$LBR LBR>EDTLIB/RP=NEWMOD MODULE "IOMOD" LBR>†Z

6. Finally task-build and run MAKSIL to create the new copy of EDT.

RUN \$TKB

TKB>@EDTSHR

RUN \$MAKSIL

MAKSIL V7.1-11 RSTS/E V7.1-11 Timesharing

Resident Library name? EDT

Task-built Resident Library input file < EDT.TSK>? EDT.TSK

Include symbol table (Yes/No) < Yes>? YES

Symbol table input file < EDT.STB>? EDT.STB

Task Image SIL output file < EDT.SIL>? EDT.SIL

EDT built in 8 K-words, 460 symbols in the directory

EDT.SIL renamed to EDT.SIL < 104 >

And your new copy of EDT is ready to be installed. As in the case of the previous patch with IODATA, make a copy of the good EDT that you have in case something was entered wrong.

If you don't think you can handle entering either or both of the listed modules, you can send \$25 to my attention at the following address. You will receive a magtape (please tell me 800 or 1600 BPI) with the sources to the IODATA and JOUPAT files, and the EDT help files and initializer from the previous issue. INFINITY SOFTWARE CORPORATION, EDT Patch Tape Offer, 2210 Wilshire Blvd, Suite 801, Santa Monica, CA 90403.

Until next issue, have a good edit.

CHEZ RSTS

BILL OF FARE

Your Hosts - The P. De Pea Family

APPETIZERS

OCTAL Soup BASIC-PLUS Chowder GRIPE.TXT on the half-shell Saute'd SET LOGINS Hot Nested Loops DATE\$ Nuts Alphanumeric Soup

ENTREE'S

Filet of BACKUP Broiled in Binary butter, topped with sprig of SPLRUN

Roast RMS in SHUTUP Sauce with CHR\$(9) caps and FIP fries

Deep Sea \$SORT fish stuffed with QUMRUN and choice of ERRLOG, ERRDIS or ERRDAT

RANDOMIZE plate with CHAIN statements and selected NULL STRINGS

All entree's include SYSTAT salad and choice of Memory

FROM THE GRILLE

Grilled TTY.CMD
ON ERROR Plate
I/O Triple Decker
BOT on Bulkie
OPEN FILE sandwich
Formatted ASCII ON Rye
Toasted GO SUB

Grill selections include silicon chips and character strings

DESSERTS

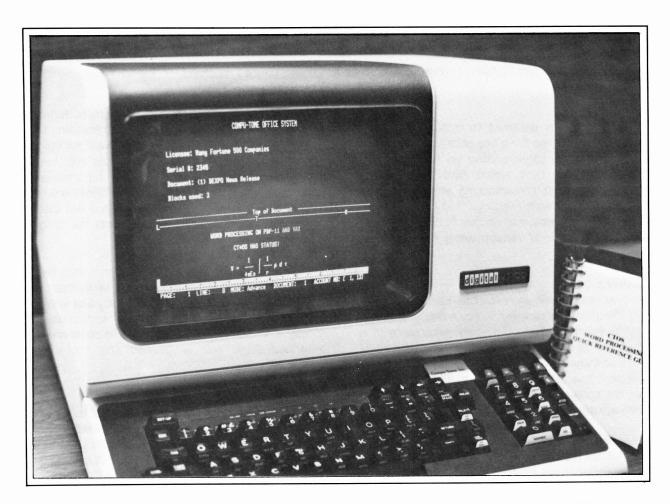
BAD BLOCK Sundae Compiled Crepe Password Puff w/PRINT USING sauce

BEVERAGES

Diet Control C MATRIX Malt Syntax Soda TAB Floating Point Frappe

DECUS Card accepted
All selections are VERSION 7.0 fresh

Dennis DeMaria, Watertown, Mass.



Total Office Automation

for DEC VMS, RSTS/E, RSX-11M, 11M+, TSX+, RT11

CT*OS Word Processing

- Full editing capability global search/replace, cut/ paste, headers, footers, auto numbering, boiler-plate libraries, document assembly, auto bold, underline, uppercase. Movement by word, paragraph, sentence, page, etc. Status line indicates page, line, column position in addition to the mode, document number, and account.
- Simultaneous print and edit variety of serial letter quality and draft printers. Full-page CRT compatibility.
- Full production word processing with multiple terminals and multiple printers. CPU efficient and easy to learn. Interfaces with DP files.
- Adaptable to your own application with user-defined keys and list processing.

Exectronic Mail™

- 60,000 + users 30kb for first user, each additional user 10kb, one-keystroke instructions.
- Management communications system directory of users, log of communications, acknowledge, and nudge capability, and more. Transfer any file on the system.
- Create and read your message with a full-featured word processor.

Electronic Spreadsheet

 Budgets and cash flow projections — create with the spreadsheet package, edit in CT*OS, and send with Exectronic Mail™ to the head office. Or print to the printer or to a disk file for inclusion in other reports. Or send to the typesetter.

Spelling Correction (more than a dictionary)

 Global correction directly on the word processing document, while user dictionary is updated with special terminology — 80,000 word dictionary.

East Coast	Midwest	Pacific
Dennis A. Wills Compu-Tome, Inc. (213) 796-9371	Paul Harris Gina Harris Mountain W. Software (213) 796-9374	Diane Sullivan Ted Dresser Discom- (805) 499-6256 (213) 796-9375

COMPU-TOME, INC.

234 East Colorado Boulevard Pasadena, CA 91101

FORMS

By Jeff Corbett, Programmer/Operator, Oconee County, Walhalla, SC 29691

1.0 Introduction

FORMS is a program designed to make it easier to maintain the proper settings for a line printer spooling job. When the time came to begin setting up forms for the various types of paper that would find itself in the lineprinter, I realized that remembering all of the form names, page lengths and width settings that I would be using could be quite a headache, and neither PLEASE, QUE nor SPOOL had any intention of remembering for me. Thus FORMS was born.

2.0 FORMS User's Guide

FORMS first major task upon start up consists of looking through the message receiver table, making sure the receiver OPSER exists, and storing every occurrence of a "LP?SPL" receiver. If only one is found it is assumed to be the only possible target for form changing and FORMS proceeds to the next operation; otherwise it displays a menu listing of all the spoolers it found and asks for the user's choice. The choice may be specified by number or by receiver ID.

Once the correct spooler has been decided upon, FORMS will read FORMS.DAT and display a menu listing all of the form definitions that it has found. It will then ask for the user's choice. The desired form may be indicated either by the form number or the form name. Typing "O" (zero) will cause the spooler to print the current form and switches, or will align the current form in conjunction with "/A" (see next paragraph). The length of the form name given is all that is checked against the table of legal forms, therefore typing "NO" will match the first form in the table beginning with "NO" (probably NORMAL).

The only switch that FORMS currently has available is the "/A" switch. The "/A" option causes the "/ALIGN" switch to be appended to the form selected, if not already present. This causes the spooler to print a test alignment pattern and to wait to be sent confirmation before resuming operations.

After the form has been selected, FORMS will send OPSER the following command line:

INT < spooler > /FOR < formname > / < switches > and then exit, unless the "/ALIGN" switch is included in the < switches > (see next paragraph). Since OPSER has a say in who sends to the spooler, if the user is not one of the OPSER valid operators all this work will be for naught. However, if the user is on OPSER's table, the request will be forwarded to the spooler which will then process the form definition and send the Operator Services Console the confirmation of the form change. FORMS exits with the "Exit with No Prompt Message" SYS call because the process isn't quite done until the OSC receives the confirmation, and it is assumed that FORMS will be run at the OSC to receive the messages from the spooler.

If the "/ALIGN" switch was in the form definition or requested by the user, FORMS will print "Standing By..." and go into an input wait. The user should wait for the spooler to go through the forms alignment process and then type the appropriate response (either "RETRY" for another check or "GO" to tell the spooler to go ahead). FORMS will repeat this step until a "GO" is typed, after which it will exit.

The FORMS CCL command accepts the following format:

FORMS/<form>[/A]/<spooler>

Note: The <form> and <spooler> may be identified by either name or by number here as in their respective menus. The <spooler> is not necessary and will be ignored if only one spooler is present. The "/A" switch was explained earlier. If a form or spooler is found to be incorrect, the appropriate menu will be printed and the user will be prompted.

3.0 Installation Notes

Before FORMS can be used, it must be compiled, protected <232>, and the data file "FORMS.DAT" must be placed in the same account. This data file should contain all of the forms used at your installation. A sample data file is included at the end of this article. The CCL definition:

CCL FORM-S=[PPN]FORMS.BAC;PRIV 30000 is supported by FORMS. The $<\!232\!>$ protection code and the PRIV option are only necessary if there are non-privileged OPSER "operators" that will be using FORMS, otherwise a $<124\!>$ will suffice.

FORMS may be compiled by CSPCOM if desired, but comment line 120 must be removed before compiling or CSPCOM will generate Err O, and quit compiling. FORMS was developed under RSTS V7.0-07, and minor changes may be required under newer releases of RSTS. I hope you will find this program of use and that it will keep you in top form.

```
EXTEND
                                         FORMSA
              å
                                                                                    1 &
              PROGRAM:
                                          FORMS
                                                                                    1 &
                                          1 . 3
51
              VERSION:
61
                   DATE:
               AUTHOR:
71
                                          Computer Programmer/Operator
Oconee County, South Carolina
                                                                                                  1 6
                                          Copyright (C) 1982
              This software has been furnished free of charge, and may be copied only with the inclusion of the Author's name and this copyright notice. No title to or ownership of this software is hereby transmitted. Neither the author nor the County of Oconee assumes any responsibility for the use or the reliability of this software. The author welcomes any comments and bug reports mailed to J. Corbett, County Mail Room. Oconee County, Walhalla, South Carolina, 26691.
               Room, Oconee County, Walhalla, South Carolina 29691.
20!
                           MODIFICATION HISTORY&
              VER/ED
                                         EDIT DATE
                                                                      Build Valid Spooler ID table &
                                          23-Sep-82
211
                 1.2
                 1.3
```

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.®

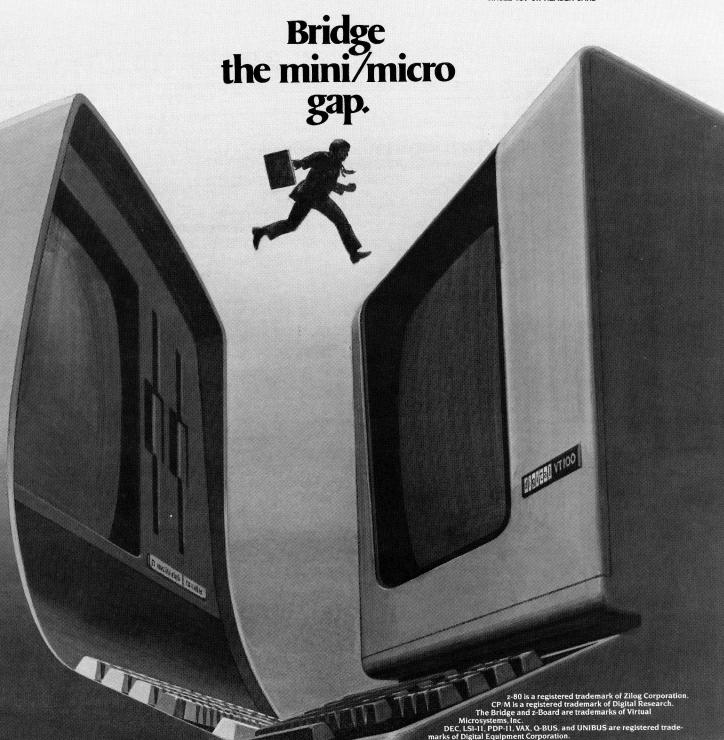
System Integration. One of the key benefits of The Bridge is that all microcomputer operations are centralized in your mini. It's no longer necessary to maintain various micros at different work stations — all with different disk formats. The Bridge gives you one format, available to each operator, at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

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.

CIRCLE 157 ON READER CARD



2150 Shattuck, Berkeley, CA 94704



```
PRINT I$; TAB(5$); LOGNAM$(I$); CR.LF$ FOR I$ = 1$ TO SPOOLERS$ & Print list of spoolers found &
                                                                                                                                                                                                                                                                                                 1160
                                                  GENERAL DESCRIPTION&
                                                                                                                                                                                                                                                                                                                         INPUT 'Change forms on which Spooler <1>'; SPOOLER$

I$ = VAL(SPOOLER$) &

I$ = 1$ IF I$ < 1$ &

I$ = 5 SPOOLER$

THEN &

1260 &

ELSE &

SPOOLER$ = 'Value too big : ' + SPOOLER$ &

4 GOTO 1240 &
                         FORMS is designed to allow the person responsible for maintaining & the proper forms in the line printers to change the type and the & parameters in use by a spooler without having to remember all the & specifics of form width and length, etc. &
1101
                                                                                                                                                                                                                                                                                                 1180
                                                                                                                                                                                                                                                                                                 1200
                         All of the forms used by an installation should be loaded & into the data file $FORMS.DAT using the format: &
1201
                                   <formname>[/<switch1>[/<switch2>[etc...]]] ! <description> &
                                                                                                                                                                                                                                                                                                                            come here on error from VAL(SPOOLER$), i.e. LOGNAM was specified SPOOLER$ = CVT$$(SPOOLER$, 38$) & GOTO 1260 IF SPOOLER$ = LOGNAM$(I$) FOR I$ = 1$ TO SPOOLERS$ & Check LOGNAM against list of spoolers &
                          &
Ex: CHECK /DFLENGTH:22/LENGTH:22/WIDTH:80/HEADINGS:0 ! Pay Checks &
                                                                                                                                                                                                                                                                                                  1220
                         & Note: All switches that could change should be specified, and the & exclamation point must be included. &
                                                                                                                                                                                                                                                                                                                          PRINT 'ino such Spooler - '; SPOOLER$; CLEAR.EOL$ & PRINT UP.CUR$; UP.CUR$; CLEAR.EOL$; & SPOOLER$ = '' & GOTO 1140 IF CCL$ & GOTO 1180 & FOOLER$ = '' & GOTO 1180 & FOOLER$ = '' & GOTO 1180 & FOOLER$ = '' & GOTO 1180 & F
                         \frac{\delta}{\delta} The user then has only to type the number or name of the needed \frac{\delta}{\delta} form at the appopriate time. Form "0" may be specified to request \delta of the spooler the form it is currently using. \delta
                                                                                                                                                                                                                                                                                                 1240
                          FORMS may be entered by "RUN $FORM" or by CCL entry. & The format for the CCL command is: &
1301
                                                FORMS / (formname or number) [/A]/(reciever ID or number) &
                                                                                                                                                                                                                                                                                                                            SPOOLER$ = LOGNAM$(I$) & Remember which spooler for the send to OPSER &
                                                                                                                                                                                                                                                                                                 1260
                          \frac{\delta}{\delta} . The reciever ID need not be specified if only one spooler is in \delta operation. \delta
                                                                                                                                                                                                                                                                                                                           PREFIX$ = CHR$(128\$+64\$) + 'INT ' & OPSER command prefix to requests &
                                                                                                                                                                                                                                                                                                  1280
                          The switch '/A' may be used with the form selection to & check the alignment of the paper. It causes the 'ALIGN switch to & be sent to the spooler, and waits to send 'GO' or 'RETRY' back to & the spooler after the alignment pass is printed. &
                                                                                                                                                                                                                                                                                                                            & READ IN FORMS &
                          This program and data file may be moved to any desired \delta account without affecting program operation with the limitation \delta that the program and the data file reside in the same account. \delta
                                                                                                                                                                                                                                                                                                                           REGN.PRIV$ = SYS(CHR$(6$)+CHR$(-21$)+CHR$(0$)) & OPEN HOME$ + 'FORMS.DAT' FOR INPUT AS FILE 1$, MODE 8192$ & LOSE.PRIV$ = SYS(CHR$(6$)+CHR$(-21$)) &
1501
                                                                                                                                                                                                                                                                                                 1420
                          \delta . If non-priviledged operators are to use this program, it \delta must be protected <232>, and the CCL specification should include \delta the PRIV option. \delta
                                                                                                                                                                                                                                                                                                                            INPUT LINE #1$, V$ &
FORMS$ = FORMS$ + 1$ &
FORM$(FORMS$) = CVT$$(V$,132$) &
                                                                                                                                                                                                                                                                                                  1440
                                                                                                                                                                                                                                                                                                                            GOTO 1440 &
                                                                                                                                                                                                                                                                                                                            Come here on FORMS.DAT end of file &
900
                                                DIMENSION STATEMENT&
                                                                                                                                                                                                                                                                                                                           CLOSE 15 &
                                                                                                                                                                                                                                                                                                  1460
                          DIM M$(30), FORM$(10), LOGNAM$(10) &
                          1 &
1 Change FORM$() dimension to reflect the maximum number of forms &
1 Change LOGNAM$() dimension to reflect the maximum number of spoolers &
                                                                                                                                                                                                                                                                                                  17001
                                                                                                                                                                                                                                                                                                                            FORM SELECTION&
                                                                                                                                                                                                                                                                                                                           GOTO 1760 IF CCL$ AND LEN(FORM$) &
PRINT CLEAR.SCREEN$ &
PRINT 'FORMS '; VER.ED$; TAB(13$); SYSTEM$; &
PRINT ' Forms Menu Spooler: '; SPOOLER$ &
1000
                                                 START&
                        ON ERROR GOTO 19000 &

V$ = SYS(CHR$(6$).CHR$(-7$)) &

DROP.PRIV$ = SYS(CHR$(6$).CHR$(-21$)+CHR$(-1$)) &

ES$ = CHR$(155$) &

CLEAR.SCREEN$ = ES$ + 'R' &

CLEAR.EGL$ = ES$ + 'R' &

CR.LF$ = CHR$(13$) + CHR$(10$) &

SYACIMG$ = '. &

SYSTEM$ = CVT$$(RIGHT(SYS(CHR$(6$)+CHR$(9$)+CHR$(0$)),3$),4$) &

VER.ED$ = "V1.3" &

NOTE: VT52 type screen commands forced (in CLEAR.SCREEN$) and used &
                                                                                                                                                                                                                                                                                                                           PRINT ' FORMS Menu Spooler: '; SPOOLER' & PRINT & SPACING$ = CR.LP$ IF FORMS$ < 11$ & SPACING$ = CR.LP$ IF FORMS$ < 11$ & PRINT FORM$; TAB(S$); FORM$(FORM$); SPACING$ FOR FORM$ = 1$ TO FORMS$ & PRINT UNLESS LEN(SPACING$) & Jump down to parse if CCL entry & Else, clear screen, print header & Set up a blank line between form descriptions if there's room. & Display form options (more than 21 will overfill the screen) &
 1020
                                                                                                                                                                                                                                                                                                                            INPUT 'FORM <1>'; FORM$ &
FORM$ = CYT$$(FORM$,385) &
I$ = INSTR(15,FORM$,'/A') &
IF I$ &
THEN &
ALIGN$ = -15 &
FORM$ = LEFT(FORM$,I$-15) &
                                                                                                                                                                                                                                                                                                    1740
                         CHANGE SYS(CHR$(12$)) TO M$ & HOME$ = '_' + CHR$(M$(23$)) + CHR$(M$(29$)) + NUM1$(M$(25$)) + ':' & HOME$ = '_' + CHR$(M$(23$)) + CHR$(M$(26$) AND 15 & HOME$ = HOME$ + '[' + NUH1$(M$(5$)) + ']' + NUH1$(M$(5$)) + ']' & THEN & HOME$ + HOME$ + HOME$ + HOME$ + '[' + NUH1$(M$(5$)) + ']' & THEN & HOME$ + '[' + NUH1$(M$(5$)) + ']' & HOME$ + '[' + NUH1$(M$(5$))
1030
                                                                                                                                                                                                                                                                                                                            FORM$ = '1' UNLESS LEN(FORM$) &
FORM$ = VAL(FORM$) &
IF FORM$ <= FORMS$ &
THEN &
1820 &
                                                                                                                                                                                                                                                                                                    1760
                                                    &
PRINT '?Please "RUN FORMS"' &
                                                    oron 32767 & Build Device name, Account number of last opened file & Build Device name, Account number of last opened file & If it was opened on channel 15 then it was us loading & Else, ask user to run us and exit &
                                                                                                                                                                                                                                                                                                                                     ELSE & FORM$ = 'Value too big : ' + FORM$ & GOTO 1800 &
                        I Else, ask user to run us and exit &

SNDLST$ = SWAP$(CVT$$(MID(SYS(CHR$(6$)+CHR$(-12$)),19$,2$))) &
SPOOLBERS, OPSER.THERE$ = 0$ &
REON.PRIV$ = SYS(CHR$(6$)+CHR$(-21$)+CHR$( 0$)) &
PTB$ = PEEK(SNDLST$) &
IF PTB$ = 0$ &
THEN &
PRINT &
PRINT &
PRINT *(No message recievers-OPSER not running &
GOTO 32767 &
Get pointer to Reciever's list from 'Get Monitor Tables, II' &
Initialize spoolers count, and OPSER running flag &
Get pointer to first reciever &
Quit now if there aren't any recievers &
 1040
                                                                                                                                                                                                                                                                                                                            I$ = LEN(FORM$) &
GOTO 1820 IF LEPT(FORM$(FORM$),I$) = FORM$ FOR FORM$ = 1$ TO FORMS$ &
Try for a match &
NOTE: We're only trying to match up to the length of the form &
request, i.e. A request of 'N' will match the first form &
starting with 'N'. &
                                                                                                                                                                                                                                                                                                    1780
                                                                                                                                                                                                                                                                                                                            1
                                                                                                                                                                                                                                                                                                                             PRINT CLEAR.SOLS &

I$ = INSTR(1$, FORM$(FORM$), 'I') &

I$ = LEM(FORM$(FORM$)) - 1$ UNLESS I$ &

REQUEST$ = ":FOR" + LEFT( FORM$(FORM$), I$-2$) &

REQUEST$ = REQUEST$ + '/ALIGN' &

REQUEST$ = REQUEST$ + '/ALIGN' &

ALIGN$ = (INSTR(1$, REQUEST$, '/ALIGN') <> 0$) &

ALIGN$ = (INSTR(1$, REQUEST$, '/ALIGN') <> 0$) &

IF ALIGN$ &

THEN &

1840 &

ELSE &

ELSE &
                                                                                                                                                                                                                                                                                                     1820
                                                                                                                                                                                                                                                                                                                              1840 &
ELSE &
    V$ = SYS(CHR$(5$)) &
    COTO 32767 &
    Drop down a line, and clear any complaints while we're at it &
    Set up fully the request line, removing description &
    Tack on the /ALIGN switch if requested and not in form def. &
    Send it to OPSER &
    If aligning form, hang around to send 'GO' or 'RETRY', else &
    Exit, but don't print "Ready" because we're waiting for &
    spooler to respond &
                            IF SPOOLERS$ = 0$ &
   1100
                                    THEN & PRINT & PRINT & PRINT "7No print spoolers on line" & GOTO 32767 &
                                                                                                                                                                                                                                                                                                                               PRINT 'Standing by...' &
                            IF SPOOLERS$ = 15 &
                                                                                                                                                                                                                                                                                                      1840
                                                                                                                                                                                                                                                                                                                              PRINT 4
OPEN '_KB:FORMS.CHD' FOR INPUT AS FILE 15
INPUT #15, MSG$ 4
MSG$ = CVT$$(MSG$,125) &
MSG$ = CVT$$(MSG$,125) &
ERTRY$ = (LEFT(MSG$,25)) = 'RETRY') 4
IF RETRY$ = 0 $ 4

THEN $ 4

PRINT 15Please type "GO" or "RETRY" 4

GOTO 1860 &
   1120
                                     THEN & SPOOLERS$ & GOTO 1260 & If there's only one spooler, there's no choice &
                                                                                                                                                                                                                                                                                                       1860
                            GOTO 1200 IF CCL$ AND LEN(SPOOLER$) &
PRINT CLEAR.SCREEN$ &
PRINT 'FORMS '; VER.ED$; TAB(13$); SYSTEM$; &
PRINT 'Spooler Menu' &
PRINT &
                             PRINT &
Jump down to parse on CCL entry &
Else, clear screen and print header &
```

RSTSPROFESSIONALRSTSPROFESSI

```
CLOSE 1$ & REQUEST$ = ':' + MSG$ & GOSUB 10000 & GOTO 1840 IF RETRY$ & GOTO 32767 &
9999
                                    SEND MESSAGE TO OPSER&
                 LOGNAME$ = 'OPSER ' & MSG$ = PREFIX$ + SPOOLER$ + REQUEST$ + CR.LF$ & SEND$ = CHR$(6$) + CHR$(2$) + CHR$(-1$) + CHR$(0$) & LOGNAME$ + STRING$(10$,0$) & The address for this message & Put the whole message together & Define the sending string &
                  IS = LEN(MSG$) &
IF IS < 204
10020
                          THEN &

V$ = SYS(SEND$ + CHR$(I$) + MSG$) &
                   ELSE & V$ = SYS(SEND$ + CHR$(255$) + LEFT(MSG$,19$)) & MSG$ = RIGHT(MSG$,20$) & GOTO 10020 & GET the length of the message & If length is less than 20, then send its length and it to OPSER & Else send CHR$(255) + the first 19 characters and try this over & again with the remains of the string &
 10040 RETURN &
                  1 &
18999
                                    ERROR HANDLER&
                    1 &
                  V$ = SYS(CHR$(6$)+CHR$(-7$)) &

IF ERR = 26$ &

THEN &

LOSE.PRIV$ = SYS(CHR$(6$)+CHR$(-21$)) &

RESUME 32760 &
 19000
                  IF ERR = 52$ &
THEN &
RESUME 1220 IF ERL = 1200$ &
RESUME 1780 IF ERL = 1760$ &
RESUME 19999 &
Name instead of number specified &
                  IF ERR = 11$ &
 19040
                          THEN & RESUME 1460 IF ERL = 1440$ & RESUME 32760 &
                  IF ERR = 55 AND ERL = 14205 &
THEN &
PRINT '?Needed file "'; HOME$; 'FORMS.DAT" is missing.' &
PRINT 'Please create this file and try again.' &
RESUME 32767 &
 19060
```

```
RESUME 19999
PRINT '7Unexpected FORMS error - '; &
PRINT cvt$4(RIGHT(SYS(CHR$(6$)+CHR$(9$)+CHR$(ERR)), %$), %$); &
PRINT ' at line'; ERL &
GOTO 32760 &
19998
19999
30000
                 1 &
                                  CCL ENTRY&
                1 &

V$ = CVT$$(SYS(CHR$(75)), 85+165+325+128$) &

IF LEFT(V$,55) = 'FORMS' &

THEN &

ON ERROR GOTO 19000 &

V$ = RIGHT(V$,75) &

GOTO 1020 UNLESS LER(V$) &

CCL$ = -1$ &

ALIGN$ = INSTR(2$, V$, '/A') &

V$ = LEFT(V$,ALIGN$-15) + RIGHT(V$,ALIGN$+2$) IF ALIGN$ &

I$ = LEFT(V$,IS-15) IF I$ &

V$ = LEFT(V$,IS-15) IF I$ &

FORM$ = V$ &

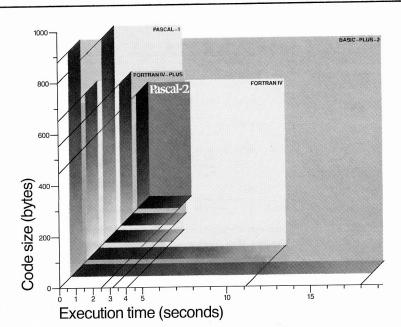
GOTO 1020 &

Start looking for '/A' switch after 1st position &

because form name might start with 'A' &
30020
30040 PRINT '?Illegal CCL entry' &
32760 CLOSE 15
32767 END
       NORMAL/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:1 ! 1%x11(6) gbar, w/header WIDE /DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:0 ! 1%x11(6), no header NARROW/DFLENGTH:66/LENGTH:66/WIDTH:680/HEADINGS:0 ! 8x11(6), no header HEADED/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:0 ! 8x11(6), w/header CHECK /DFLENGTH:62/LENGTH:22/WIDTH:080/HEADINGS:0/ALIGN ! (6) Pay Checks
 Ready
  RUN SFORMS
 CClear Screen>
FORMS V1.2 RSTS V7.0-07 Oconee County Spooler Menu
                                                                                                                           <Default taken>
 Change forms on which Spooler <1>?
  <Clear Screen>
FORMS V1.2 RSTS V7.0-07 Oconee County Forms Menu Spooler: LPOSPL
   1 NORMAL/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:1 | 14x11(6) gbar, w/header
          WIDE /DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:0 ! 14x11(6), no header
```

Pascal-2

PDP-11 RSX, RSTS/E, RT-11 and TSX-Plus



Quicksort benchmark executed on a PDP-11/45.

The Pascal-2 system consists of an optimizing compiler, a highlevel debugger and other utilities.

Write us for benchmark details.

Oregon Software

The Pioneer in Performance Pascal

2340 S.W. Canyon Road Portland, Oregon 97201 (503) 226-7760 TWX: 910-464-4779

FORTRAN IV-PLUS, BASIC-PLUS-2, PDP, VAX, RSX, RSTS/E and RT-11 are trademarks of Digital Equipment Corporation. TSX-Plus is a trademark of S&H Computer Systems.

```
NARROW/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:0 ! 8x11(6), no header
      HEADED/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:1 | 8x11(6), w/header
      CHECK /DFLENGTH:22/LENGTH:22/WIDTH:100/HEADINGS:0/ALIGN ! (6) Pay Checks
                                                                     (Specified by number)
MESSAGE 20: 06-Oct-82 12:17 PM JOB:4 DET SPLIDL[1,41]
Forms change requested -
Form: NARROW /Length:66 /Width:80 /Headings:0 /Align:No /DFLength:66 /Page:Yes
                                                                      (CCL entry)
FORMS
(Clear Screen)
FORMS V1.2
                RSTS V7.0-07 Oconee County Spooler Menu
 1 LPOSPL
 2 LP1SPL
Change forms on which Spooler <1>? LP1SPL
(Clear Screen)
FORMS V1.2 RSTS V7.0-07 Oconee County Forms Menu Spooler: LP1SPL
FORMS V1.2
      NORMAL/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:1 ! 14x11(6) gbar, w/header
      WIDE /DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:0 ! 14x11(6), no header
      NARROW/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:0 ! 8x11(6), no header
      HEADED/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:1 | 8x11(6), w/header
     CHECK /DFLENGTH:22/LENGTH:22/WIDTH:100/HEADINGS:0/ALIGN I (6) Pay Checks
                                                                      (Align the form)
Form (1)2 NORMAL/A
                                                 ! (Waiting to respond to spooler)
Standing by ...
MESSAGE 21: 06-0ct-82 12:18 PM JOB:8 DET SPLIDL[1,8]
Form: NORMAL /Length:66 /Width:132 /Headings:1 /Align:Yea /DFLength:66
/Page:No
REQUEST 22: 06-Oct-82 12:19 PM JOB:8 DET SPLRUN[1,8] LPISPL If forms are properly aligned, enter 'GO' Otherwise, enter 'RETRY': RETRY
                                                                     <Alignment not right>
Standing by ...
REQUEST # 22 DELETED
                   23 : 06-Oct-82 12:19 PM JOB:8 DET SPLRUN[1,8] LPISPL If forms are properly aligned, enter 'GO' r 'KETRY'.
                                                                     <Right this time>
Ready
REQUEST # 23 DELETED
PLE/INT LPISPL: GFFLINE
COMMAND SENT TO 'OPSER
                                                           (What happens with one spooler?)
Ready
JOB #8 'LPISPL' TAKEN OFFLINE
MESSAGE 24 : C6-Oct-82 12:19 PM JOB:2 DET LP1SPL (8) REQUESTED OFF-LINE -- TAKEN OFF-LINE
                                                                        QUMRUN[1,41]
FORMS
   Clne Spooler - No Spooler Henu>
RMS V1.2 RSTS V7.0-07 Oconee County Forms Menu Spooler: LPOSPL
 1 NORMAL/DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:1 | 14x11(6) gbar, w/header
 WIDE /DFLENGTH:66/LENGTH:66/WIDTH:132/HEADINGS:0 ! 14x11(6), no header
 3 NARROW/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:0 ! 8x11(6), no header
    HEADED/DFLENGTH:66/LENGTH:66/WIDTH:080/HEADINGS:1 | 8x11(6), w/header
 5 CHECK /DFLENGTH:22/LENGTH:22/WIDTH:100/HEADINGS:0/ALIGN ! (6) Pay Checks
                                                           ("O"-Report on current form>
          27: 06-0ct-82 12:34 PM JOB:4 DET SPLIDL[1,41]
Forms change requested -
HECK /Length:22 /Width:100 /Headings:0 /Align:No /DFLength:22 /Page:182
MESSAGE
FORMS/CHECK
                                                            </ALIGN in form definition>
Standing by ...
MESSAGE 25: 06-Oct-82 12:20 PM JOB:4 DET SPLIDL[1,41]
Forms change requested -
Form: CHECK /Length:22 /Width:100 /Headings:0 /Align:Yes /DFLength:22 /Page:Yes
REQUEST 26: 06-Oct-82 12:20 PM JOB:4 DET SPLRUN[1,41] LPOSPL If forms are properly aligned, enter 'GO' Otherwise, enter 'RETRY': XX
$Please type "GO" or "RETRY"
Ready
```

REQUEST # 26 DELETED

MORE ON USING THE VT100 PRINTER PORT

By Terry Ridgers, Analyst/Programmer, Dupont, Canada, Inc.,

St. Clair River Works, Corunna, Ontario

Here is an updated modification to the PP CCL mentioned in the RSTS PROFESSIONAL June, 1982 ("Using the VT100 Printer Port Option Effectively," by Meramec Automated Solutions, Inc.). The new version is written in BASIC+2 and will handle any sequential file. It displays error messages together with a help screen if the file does not exist or is not sequential.

In the past we have used PIP to display our list files to the printer port. We have found that with the PP CCL, that our files print on the average of about 22% faster because of the single buffering to the printer port only. It is also unnecessary to switch to 132 columns with PP.

The CCL to set up PPRINT is: CCL PP-PRINT.TSK; 3,0000.

```
*** PPRINT.B2S ***
                 MERAMEC AUTOMATED SOLUTIONS, INC.
                 PROGRAM CONTROLLER FOR VT100 PRINTER PORT OPTION (VT1XX-AC)
                 ENTERED THROUGH CCL PF[/FF] FLN, AT LINE 30000, TO HAVE A FILE PRINTED THROUGH THE PRINTER PORT (/FF WILL FORM FEED PRINTER
                 PRIOR TO PRINTING FILE).
OR, PP/ON AND PP/OFF TO TOGGLE THE PRINTER PORT ON AND OFF
                 MODIFICATION LOG
                 27-JUL-82
                                            - WRITTEN IN BASIC + 2
                                            - DISPLAYS HELP SCREEN /HELP
- PRINTS ANY SEQUENTIAL FILE
- BY TERRY RIDGERS D U P O N T CAN. INC. CORUNNA ONT.
          GOTO 32767
20
1000
         ! MAINLINE PROCESSING
          ON ERROR GOTO 19000
                 1 STANDARD ERROR TRAP
        \ OPEN FLN$ FOR INPUT AS FILE 1%,
ORGANIZATION SEQUENTIAL STREAM
                                                           ! TRY AND OPEN AS ASCII FILE
                                                           I OR RMS SEQUENTIAL STREAM FILE&
I GO PRINT THE FILE &
        \ GOTO 1030
       OPEN FLN$ FOR INPUT AS FILE 1$,
ORGANIZATION SEQUENTIAL FIXED
\ GOTO 1030
                                                           1 TRY AND OPEN AS RMS SEQUENTIAL&
                                                           ! FIXED FILE
! GO PRINT THE FILE
         OPEN FLN$ FOR INPUT AS FILE 1$, ! TRY AND OPEN AS ORGANIZATION SEQUENTIAL VARIABLE ! VARIABLE FILE
                                                           1 TRY AND OPEN AS RMS SEQUENTIAL&
          PRINT IF CCPOS( 0% )
       PRINT " Printing " + FLN$ +"( ^C TO STOP )..." + FNPP.ON$;

1 TURN ON THE PRINTER PORT
        \ T.1% = CTRLC
! ENABLE CTRL/C TRAP
         UNTIL AN.ERRORS
1040
                    \ NEXT
               ! THIS LOOP IS TERMINATED ONLY BY ERROR TRAP TO 19000
                   FNPP.ON$, FUNCTION TO TURN ON PRINTER PORT, AND EXCUTE SYS CALL TO SET PRINTER CHARACTERISTICS
```



THIS ISSUE OF THE RSTS PROFESSIONAL MARKS OUR THIRD ANNIVERSARY.

```
DEF* FNPP.ON$

DUMMY$ = SYS( PP.SET$ )

FNPP.ON$ = PP.ON$
                                                   FNPP.OFF$, FUNCTION TO TURN OFF PRINTER PORT, AND EXCUTE SYS CALL TO RESTORE TERMINAL CHARACTERISTICS
  15100
                               DEF* FNPP.OFF$
                        DUMHY = SYS( VT100.SET$ )
PRINT PP.ON$
FNPP.OFF$ = PP.OFF$ +
                                                                                                                                                          ! TURN ON IF NOT ON ORIGINALLY
                                                                          CHR$(155$) + "[?21" ! RETURN TO VT52 MODE
                         \ FNEND
                                                    FNPP.HELP$, FUNCTION TO SHOW HELP SCREEN
  16000
                             PRINT PP/OPRINT PP/OPPINT 
                                                                         PP/ON TURN PRINTER CONTROLLER MODE ON"
PP/OFF TURN PRINTER CONTROLLER MODE OFF"
PP[/FF] FLN PRINT NAMED FILE THROUGH PRINT PORT,"
                                                                         THEN TURN MODE OFF"
FORM FEED THE PRINTER 1ST IF /FF"
PP/HELP PRINTS THIS HELP SCREEN"
                                 PRINT
                         \ FNEND
  19000 T.15 = RCTRLC IF ERR = 285
1 REENABLE CTRL/C TRAP AS SOON AS POSSIBLE
                     ! REENABLE CTRL/C TRAP AS
RESUME 1010
IF ERL = 1000
RESUME 1020
IF ERL = 1010
PRINT FNPP.OFF$;
I TURN THE PRINTER PORT OFF
PRINT ERT$(ERR)
IF ERR <> 115 AND ERR <> 285
                                                                                                                                                       ! TRY TO OPEN AS SEQUENTIAL FIXED
                                                                                                                                                       ! TRY TO OPEN AS SEQUENTIAL VARIABLE &
                                                                                                                                                      I DISPLAY ERROR
                       PRINT "?Print line > 132 col."

IF ERR = 157%

T.1% = FNPP.HELP%
                                                                                                                                                  ! FILE RECORD INPUT > INPUT BUF SIZE &
                                                                                                                                                      ! DISPLAY HELP SCREEN
                                   IF ERR <> 11% AND ERR <> 28%
 19010
                        CLOSE #1%
                                                              ! ASSUME WE ARE DONE, AND CLOSE-UP SHOP
                         \ GOTO 32767
                                                    ! THAT'S IT FOLKS
                                                    CCL ENTRY POINT:
  30000
                                                                                 PP/ON TURN PRINTER CONTROLLER MODE ON PP/OFF TURN PRINTER CONTROLLER MODE OFF PP[/FF] FLN PRINT NAMED FILE THROUGH PRINT PORT,
                                                                                 THEN TURN MODE OFF
FORM FEED THE PRINTER 1ST IF /FF
PP/HELP PRINTS THIS HELP SCREEN
                      | CCL$ = SYS( CHR$( 7$ ) )
| GRAB THE CCL COMMAND
| ESC.CHR$ = CHR$( 155$ ) |
| USE ASCII 27+128 AS ESCAPE CHARACTER
| DUMMY$ = SYS( CHR$( 6$ ) + CHR$( 16$ ) + CHR$( 0$ ) + CHR$( 255$ ) + STRING$( 28$,0$ ) )
| VT100.SET$ = CHR$( 6$ ) + CHR$( 16$ ) + SEG$( DUMMY$,3$,LEN(DUMMY$) )
| RETRIEVE THE CURRENT TERMINAL CHARACTERISTICS WHICH
| HAVE BEEN SET BY USER
| PP.SET$ = SEG$( VT100.SET$,1$,1$ ) + CHR$( 133$ ) + MID( VT100.SET$,6$,1$)
                                                                                                                                                + MID( VT100.SET$,6$,1$) & + CHR$( 255$) + SEG$( VT100.SET$,8$,LEN(VT100.SET$))&
                     + SEG$( VT100.SET$,8$,LEN(VT100.SET$)

! SAME ATTRIBUTES AS TERMINAL, EXCEPT
! LINE WIDTH OF 132, AND FORMS CONTROL

! PP.ON$ = ESC.CHR$ + "<" + ESC.CHR$ + "[51"

+ CHR$( 13$) + CHR$( 10$)

! STRINGS TO TOGGLE THE PRINTER PORT ON/OFF

! PP.ON$ BEGINS WITH ESCAPE SEQUENCE TO SET VT100 TO ANSI MODE

! AND ENDS WITH A NEW LINE TO FLUSH THE PRINTER BUFFER

! PP.ON$ + CHR$( 12$) ! IF POS(CCL$, "/FF", 1$)

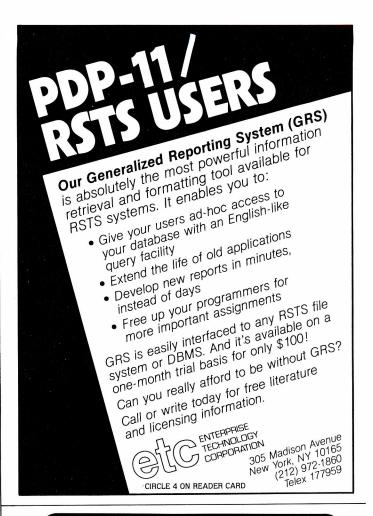
! ADD A FORM FEED FOR PRINTER IF REQUESTED

! IF POS(CCL$, "/ON", 1$) THEN

PRINT FNPP.ON$;

ELSE
                                                          IF POS(CCL$, "/OFF", 1%) THEN
PRINT FNPP.OFF$;
                                                                                   IF POS(CCL$, "/HELP", 1$) THEN
T.1$ = FNPP.HELP$
1 DISPLAY HELP SCREEN
ELSE IF POS(CCL$, "FFF", 1$) THEN
FLN$ = SEG$( CCL$, POS(CCL$, "/FF", 1$) + 3$
                                                                                                                                                                                                                                     LEN(CCL$) )
                                                                                                          GOTO 1000
                                                                                     ELSE FLN$ = SEG$( CCL$, POS(CCL$, " ", 1$) + 1$,
                                                                                                                                                                                                                                     LEN(CCL.$) )
                                                                                                \ GOTO 1000
                                                                                                                                     I IF NOT A TOGGLER COMMAND, THEN GO
                                                                                                                                     ! PROCESS THE FILE TO PRINT
32767 END
```

WE THANK YOU FOR YOUR SUPPORT AND WE AGAIN PROMISE TO KEEP THE RSTS PROFESSIONAL PROFESSIONAL.



TERMINALS FROM TRANSNET

PURCHASE PLAN • 12-24 MONTH FULL OWNERSHIP PLAN • 36 MONTH LEASE PLAN

1					
1	DESCRIPTION	PURCHASE	12 MOS.	ER MONTH	36 MOS.
	LA34 DECwriter IV Forms Ctrl		\$105	\$ 58	\$ 40
	LA100 Letter Printer RO	1,995	190	106	72
	LA120 DECwriter III KSR	2,295	220	122	83
	LA120 DECwriter III RO	2.095	200	112	75
*DEC	LA12A Portable DECwriter	2.950	280	155	106
* DEC	VT100 CRT DECscope	1,695	162	90	61
	VT101 CRT DECscope	1,195	115	67	43
	VT125 CRT Graphics	3,295	315	185	119
	VT131 CRT DECscope	1,745	167	93	63
	VT132 CRT DECscope	1,995	190	106	72
	VT18XAC Personal Computer Option	2,395	230	128	86
	T1745 Portable Terminal	1.595	153	85	58
	T1765 Bubble Memory Terminal .	2,595	249	138	93
TEXAS	TI940 CRT	1,795	173	96	65
INSTRUMENTS	TI785 Portable KSR, 120 CPS	1,795	173	96	65
INSTRUMENTS	TI787 Portable KSR, 120 CPS	2,195	211	117	80
	TI810 RO Printer	1,695	162	90	61
	TI820 KSR Printer	2,195	211	117	80
	ADM3A CRT Terminal	595	57	34	22
LEAR SIEGLER	ADM5 CRT Terminal	645	62	36	24
	ADM32 CRT Terminal	1,165	112	65	42
				1000	1000
CITOII	CIT-101 CRT	1,525	147	82	55
C-ITOH	CIT-161 Color CRT	2,675	257	143	97
procession and the second	CIT-427 Color Graphic CRT	3,095	297	165	112
	910 CRT Terminal	650	62	36	24
TELEVIDEO	925 CRT Terminal	850	82	46	31
	950 CRT Terminal	1,075	103	57	39
	Letter Quality, 7715 RO	2,695	259	144	98
NEC SPINWRITER	Letter Quality, 7725 KSR	3,195	307	171	115
GENERAL ELECTRIC	2030 KSR Printer 30 CPS	1,195	115	67	43
		2,195	211	117	80
EPSON	MX-80 F/T Printer	745	71	42	27
	MX-100 Printer	895	86	48	32
TIMEDI EV	E0400 4 Channel Stat Mux	1,525	147	82	55
TIMEPLEX	E0800 8 Channel Stat Mux	2.050	197	110	74
*DEC is the trademark of	Digital Equipment Corporation	,			

FULL OWNERSHIP AFTER 12 OR 24 MONTHS • 10% PURCHASE OPTION AFTER 36 MONTHS

MICROCOMPUTERS APPLE • COMMODORE • HP87 • DEC

ACCESSORIES AND PERIPHERAL EQUIPMENT
ACOUSTIC COUPLERS • MODEMS • THERMAL PAPER • RIBBONS • INTERFACE MODULES • FLOPPY DISK UNITS



TRANSNET CORPORATION

1945 ROUTE 22 • UNION, N.J. 07083 • (201) 688-7800

TWX 710-985-5485 800-526-4965 OUTSIDE N.J.

ATPK.TEC

By Steve Roy, P.O. Box 284, Bloomfield, CT 06002

Hi there ATPK and TECO lovers! This one should make you all happy.. it's a TECO macro that performs substitution on a command file and then feeds it to ATPK. It's very useful for running a standard command file many times, and changing a few parameters in each run. It will substitute up to 61 different strings (called TOKENS) and provides for user definition of the string that flags the TOKENS (initially '*'). Needless to say, it doesn't harm the original command file.

Important note: The macro contains several '|' characters which may print as vertical bars, but are not to be confused with uparrow... they are ASCII 124. The following strings represent actual CONTROL characters: † A, †E. †N. †R. and †T. The others found in the file ($^{\circ}$ U and $^{\circ}$ C) are carets followed by the letter, for obvious reasons. The file contains a help text on how to use it and on valid switches. The macro inputs substitution strings from the keyboard after being revoked. To get the help text on your terminal use: MUNG ATPK,/HELP

Have fun!

^AToken substitution TECO macro - V1 - 9/82 ISCAN PARM! J@:S\/HELP\ "S ^A The proper method of invoking this TECO macro is: MUNG name, [logfile =] commandfile [switches]

☐ PAYMENT ENCLOSED for one year's subscription (6 issues).

Where "name" is the name of the macro (i.e., ATPK for ATPK.TEC) This TECO macro will alter the specified command file by changing all TOKENS to character strings entered by the user. A TOKEN is any occurrence within the command file of a character preceded by an asterisk (*). The legal characters that may follow the asterisk are 1 through 9, A through Z, and (lowercase) a through z. The macro can be instructed to change the asterisk to any other character string by specifying the switch: /*string

RSTS PROFESSIONAL

Box 361 · Ft. Washington, PA 19034-0361 · (215) 542-7008

US: \$35 / Canada & 1st class: \$50 US / All other countries, air mail: \$60, payable in US dollars. □ BILL ME for one year's subscription (6 issues). ☐ US / ☐ Canada or 1st class / ☐ Foreign. V. 1, #1 V. 2, #3 V. 3, #2 V. 4, #1 V. 4, #4 Please send BACK ISSUES circled: V. 2, #1 V. 2, #4 V. 3. #3 V. 4, #2 V. 4, #5 ☐ \$10 per issue enclosed. V. 3, #4 V. 4, #3 V. 2. #2 V. 3, #1 ☐ Bill me for \$12.50 per issue. Send me a RSTS PRO M L XL (Adult Sizes Only) S Tee Shirt — \$6.95 Name

Address Suite

City/State/Zip ___

_ Phone (

FREE CLASSIFIED AD WITH SUBSCRIPTION!!

Country_

Your first 12 words are absolutely FREE, only \$1.00 per word thereafter.

SPECIAL

All 14 Back Issues for \$95.00. Payment with order.

All nulls and occurrences of CONTROL/L are removed from the command

Once the substitutions have been made, an output file is created and named ATPKnn.TMP where 'nn' is the job number. Then TECO chains to ATPK

to execute the command file. The macro appends the ATPK switch '/DEL'

which deletes the temporary command file. Any switches that this macro

does not recognize are passed to ATPK. Legal switches to this macro are:

/#string Changes the macro substitution flag character from asterisk (*) to 'string'.

Tells the macro to perform the substitution only, without chaining to ATPK for execution. This switch

may be abbreviated to: /NOEX

/NOEXECUTE

Tells the macro NOT to have ATPK delete the temporary /NODELETE file that contains the substituted commands. This switch may be abbreviated to: /NODEL

NOTE: When entering replacement strings for the tokens, the legal delimiters between strings are comma and <CR> (carriage return). To terminate input of token-replacement strings, enter the string: $//\langle CR \rangle$ To enter a token string of '//' just make sure that it is followed by a comma and another string. Null strings are valid and may be entered as two consecutive commas, or a comma followed by a <CR>. The replacement string interpreter accepts strings for token-replacement in the following order: 123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz ...so *1 is replaced with the first string, etc. To change the input string delimiter from comma to some other character, enter the first string as a null (i.e., start with a comma), followed by the character that is to be used, followed by the true first string. For example, to change the delimiter to ';', start the input with:
.:xxxx where xxxx is the first string. ,;xxxx

Needless to say, the first string may not be null, and the delimiter may not be changed to "/".

The macro prompts for the next replacement string whenever a <CR> is typed, so the strings may all be entered on one line, or each be entered on separate lines. Other than <CR>, CTRL/U, CTRL/R, and CTRL/C, all characters of ASCII value less than 32 are ignored.

The substitution occurs in reverse alphabetical (and numeric) order, so, to the extent that higher numbers and letters may contain lower numbered (and lettered) tokens, the substitution is recursive.

!LOCATE SWITCHES, IF ANY! J @:S\/\ "S .-1UO | ZUO' IEXTRACT LOGFILE NAME! J @QQ,1:S\=\ "S -D 0,.XL 0,.D QO-:QL-1UO'
IDEFAULT INFILE EXTENSION! J @QQ,1:S\.\ "U QQJ @I\.CMD\ QQ+4UO' ISAVE AND OPEN THE INFILE! J ., QOXI ., QOD @ER\^EQI\ @^UO\ATPK\ 1,3:XO 0,3D THE TEMPFILE NAME! J EJ+100\ !CREATE @:^UO\.TMP\ IOPEN IT & SETUP DEFAULTS! @EW\^EQO\ -1UD -1UE @^UF*\ @^US\/DEL\ HANDLE SWITCHES NEXT! J @:S\/\ "U @O!READY! ' 0,.D @I\/\ ZJ@I\/\ IDEL! J @:S\/DEL\ "S @FR\\ INODEL! J @:S\/NOEL\ "S -6D OUD <@::S\^N/\ ; -D> '
INOEX! J @:S\/NOEX\ "S -5D OUE <@::S\^N/\ ; -D> '
1 /*! J @:S\/*\ "S -2D @^UF\\ <@::S\^N/\ ; -1,.:XF -D> ' ZJ -D QD "N @I\/DEL\' HXS HK 1UT IREADYI AEnter replacement strings starting at: e^UT\123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz\ IPROMPT! HK GT QT-1,QTT ^A ? ^A HK OUR OUC IDEL! Q1-127 "E QC "G QC-1UC ET&2 "N 8^T^A ^A8^T | QR "E ^A\^A' QC,QC+1T' 1#QRUR QC,QC+1D | ET&2&QR "N ^A\^A OUR ' ' @0!I/0! ' OUR !C/U! Q1-21 "E OUC OUR HK 13^T10^T @O!I/O! ' Q1-18 "E 13^T10^T HT OUR @0!I/O! ' IC/R! Q1-13 "E @O!PARSE! ' Q1-31 "G OUR ZJ @Q1I\\ QC+1UC' @O!I/O! !RET!

< QT-1UT QT "E :]Q "N ^A?Push-down stack not aligned^A^C' 0; J GT QT-1,QTXC 0,.D]Q < 0:S\^EQF^EGC\ ; @FR\\ GQ > >

QT-1 "E J @::S\,\ "S 1,2XD J 2D | @^DD\,\ ' '
ZJ GD < J @:S\^EQD\ ; -D 0,.XQ [Q 0,.D \$T-62 ; >
HK QT-61 "G 61UT @0!REPLACE!'

ICHECK//! GQ Z-2 "N @OIPROMPT!' J @::S\//\ "U @OIPROMPT!']Q HK QT-1UT IREPLACE! 5<A> ^AReplacing...

IFINISH! EC QE "E ^C | @I\ATPK \ :QL "G GL @I\=\' GO GS HXQ HK @EG\^EQQ\ '

IDEBUG @ET\\! <ESC><ESC>

IOTHERS!

I PARSE !

UNITRONIX MEANS DEC

PDP 11/23
SYSTEMS
SYSTEMS
With 2 Mbyte
Memory, 160 Mbyte
Memory, std. 45 ips
Winchester, std. 45 ips
Winchester, std. 45 ips
Winchester, will run
Printer. Will run
Printer. Will run
Printer. RSTS,
RSX11, RSTS,
RSX11, RSTS,
RSX11, RSTS,
RSX11, RSTS,
RSX1000

TERMINALS

VT100 LA12 VT101 LA34 VT102 LA50 VT125 LA100 VT131 LA120

CPUs

PDP 11/23, 11/44 & VAX

INTERFACES MODEMS

COMPLETE DATA SYSTEMS

with standard and custom software



WHY YOU SHOULD CALL UNITRONIX FIRST

- Immediate Delivery from \$6 mm Warehouse Inventory
- Complete Equipment Service Capability
- Excellent Pricing/Discount Structures

In just over 6 years, Unitronix Corporation has emerged as a leading supplier of Digital Equipment Corporation (DEC) microcomputers, terminal products, printers, accessories and supplies on both a domestic and international level. Our new, ultra-modern 20,000 sq. ft. facility in Somerville, New Jersey, houses executive offices, inhouse programming, customer service, sales and marketing, component testing and complete warehouse facilities.



197 Meister Ave., Somerville, NJ 08876 TELEX: 833184



CALL FOR LATEST PRICES

(201) 231-9400

WRTOUT.BAS

By Raymond D. Tomlinson, Systems Engineer, Integrated Services, Inc., Aurora, CO 80012

```
SUB WRTOUT(DUM,DUM$,DUM$)TO BE USED FOR BASIC PLUS II,
DUM IS PASSED TO ROUTINE, DUM$ IS
PASSED BACK TO BE PRINTED
AND DUM$ IS A SWITCH FOR:
O$ = THIS IS A DOLLAR/CENT PRINT
1$ = THIS IS A DOLLAR PRINT ONLY
2$ = THIS IS NOT A MONEY PRINT
                                                                                                                                                                                                                                                                  'Y7.0-08C' + CHR$(9%) + 
RIGHT(SYS(CHR$(6%)+CHR$(9%)+CHR$(0%)),3%)
                                                                                                                                                                                                                           VERSION$ =
                                                                                                                                                                                                       99
                                                                                                                                                                                                                           PRINT #1% VERSION$
                                                                                                                                                                                                                           INPUT #1% 'Amount '; A
GOTO 100 IF A < 0.0
                                                                                                                                                                                                       100
                                                                                                                                                                                                                          INPUT #1% 'Option 0,1,2,? ';A$
GOTO 300 IF LEFT(A$,1%) = '?'
A$ = VAL(A$)
GOTO 300 IF A$ < 0$ OR A$ > 2$
PRINT #1% USING MASK$(A$),A
PRINT #1% FNWRTOUT$(A,A$)
                                                                                                                                                                                                       200
                                                                              TRAP ^C WITHIN THIS ROUTINE SET UP INTERNAL ERROR TRAPPING
                    DUMY = CTRLC
ON ERROR GOTO 32000
2!
                     3
                                                                                                                     WRTOUT.BAS
R. D. TOMLINSON
INTEGRATED SERVICES
1360 S. POTOMAC
AURORA, CO 80012
JUNE 16, 1982
A SIMPLE CONVERSION
                                       PROGRAM NAME:
WRITTEN BY:
                                                                                                                                                                                                                            GOTO 100
                                                                                                                                                                                                                           PRINT #1% 'The Options are:'
PRINT #1% 'O = Print "DOLLARS" and "CENTS"'
PRINT #1% '1 = Print "DOLLARS" only'
PRINT #1% '2 = Print the whole number, No "DOLLARS"'
                                                                                                                                                                                                       300
                                       DATE WRITTEN:
                                         PURPOSE:
                                                                                                                      FOR CONVERTING A NUMBER INTO A WRITTEN STRING.
                                                                                                                                                                                                                                         DEF FNWRTOUT$(DUM,DUM%)
                                                                                                                                                                                                       29800
                                       This is an APPENDable function to be used to convert a real or floating point number into string representation for printing on checks, receipts, etc.
                                       Until a library account can be established, this routine will be residing in DR5:[1,37]WRTOUT.BAS.
                                       To use this routine, move it to your account under the name you wish, delete lines 1 through 1000, and APPEND it into your program. You may have to change some line numbers, but I have tried to use some which most programmers seldom use.
                                       To test this routine, OLD and RUN it. Enter any positive real or floating point number between 0 and one billion. It will print the string that should be on a check or receipt and ask for another number. To terminate, enter a °C.
                                                                                                                                                                                                                                              A1$ = 1$
A1$ = 0$ IF FIX(DUM) = 1.

IF DUM < 1.0

THEN DUM$ = DUM$ + 'ZERO'

SET SWITCH (A1$) TO SHOW IF EXACTLY ONE DOLLAR

SAY "ZERO" IF LESS THAN ONE DOLLAR
                                                                                                                                                                                                        29810
                                                                                                                                                                                                                                               GOTO 29920 IF DUM < 1.0 AND B < 0. AND C < 0.AND D < 0.
                                                                                                                                                                                                        29820
                                                                                                                                                                                                                                                                  920 IF DUM < 20.0

I$ = DUM

DUM = DUM - I$

GOSUB 29970

GOTO 29865 IF B > -1.

GOTO 29885 IF D > -1.
                             50
                                                                               COPYRIGHT
                                        Copyright (C) 1982 by
Integrated Services Inc., Aurora, Colorado
                                                                                                                                                                                                                                                                   GOTO 2985 IF D > -1.

GOTO 29920

GO TO THE END IF WE ARE DONE
OTHERWISE, IF THE AMOUNT LEFT IS LESS
THAN 20 ADD THAT AMOUNT TO THE STRING
AND REMOVE IT FROM THE BALANCE FORWARD
(DUM) AMOUNT. IF OVER 100,000 THEN GO
TO SOMEWHERE SPECIAL OTHERWISE GO TO
                                         This software is provided without charge by Integrated Services, Inc. for use by any of the Integrated Resouces, Inc. companies and for distribution as they should see fit.
                                         The information in this software is subject to change without notice and should not be construed as a commitment by Integrated Services, Inc.
                                                                                                                                                                                                                                                                   DUM < 100.0
                                                                                                                                                                                                        29830
                                                                                                                                                                                                                                                                   DUM ( 100.0

If = DUM/10.0

DUM = DUM - I$ * 10$

GOSUB 29980

OTHER STRING, SUBTRACT FROM DUM AND START

THE STRING, SUBTRACT FROM DUM AND START
                                                                                                                                                                                                                                                THEN
                                         While all efforts have been made to remove problems arising from the running of this program, there can be no warranties nor guarantees, expressed or implied, by ISI or the author resulting from the use of this program on any system or computer.
                                        An attempt has been made to "round" the floating point number within this routine; however, it is highly advisable that before using, the number be rounded to the programmer's, analyst's, or manager's desires. The reasoning behind this advice is obvious to the experienced programmer. You will see problems when sending a number such as 123.454099999999999 to the routine, the PRINT USING will round to 46 cents while the written version will round to 45 cents.
                                                                                                                                                                                                                                                                    DUM < 1000.

I$ = DUM/100.0

DUM = DUM - I$ = 100$

GOSUB 29970

DUM$ = DUM$ + 'HUNDRED'
                                                                                                                                                                                                        29840
                                                                                                                                                                                                                                                IF
THEN
                                                                                                                                                                                                                                                                    DUMS + DUMS + 'HONDRED'
GOTO 29820

IF THE BALANCE FORWARD (DUM) IS LESS
THAN 1000 THEN ADD THE "HUNDREDS"
AMOUNT TO THE STRING, SUBTRACT FROM DUM
AND START OVER.
                       :
                                                                                                                                                                                                                                                                    DUM < .2E 5
                                                                                                                                                                                                                                                IF
THEN
                                                                                                                                                                                                                                                                    I$ = DUM/1000.

DUM = DUM - I$ * 1000.

GOSUB 29970
                       MODIFICATION HISTORY
                                                                                                                                                                                                                                                                    GOSUB 29970

DUM$ = DUM$ + 'THOUSAND '
GOTO 29820

IF THE BALANCE FORWARD (DUM) IS LESS
THAN 20000 THEN ADD THE "TEENS" AMOUNT
TO THE STRING, SUBTRACT FROM DUM AND
START OVER.
                           VER/ED
                                                             EDIT DATE
                                                                                                    REASON
                                                                                                    mentation intens to allow for easier reading and understanding and understanding and improved accuracy.

Add the function to allow no dollars or cents to be printed in the case where a whole number is the item to be printed. Also change the name to WRTOUT from CHKWRT to broaden its usefulness.

Allow up to .1E 13 -.01 to be acceptable to the routine and add printing of version #. Original Program written by R. D. Tomlinson.
                                                             07-SEP-82
                           V7.0-08C
                                                                                                                                                                                                                                                                    DUM < .1E 7

B = -1.

B = DUM

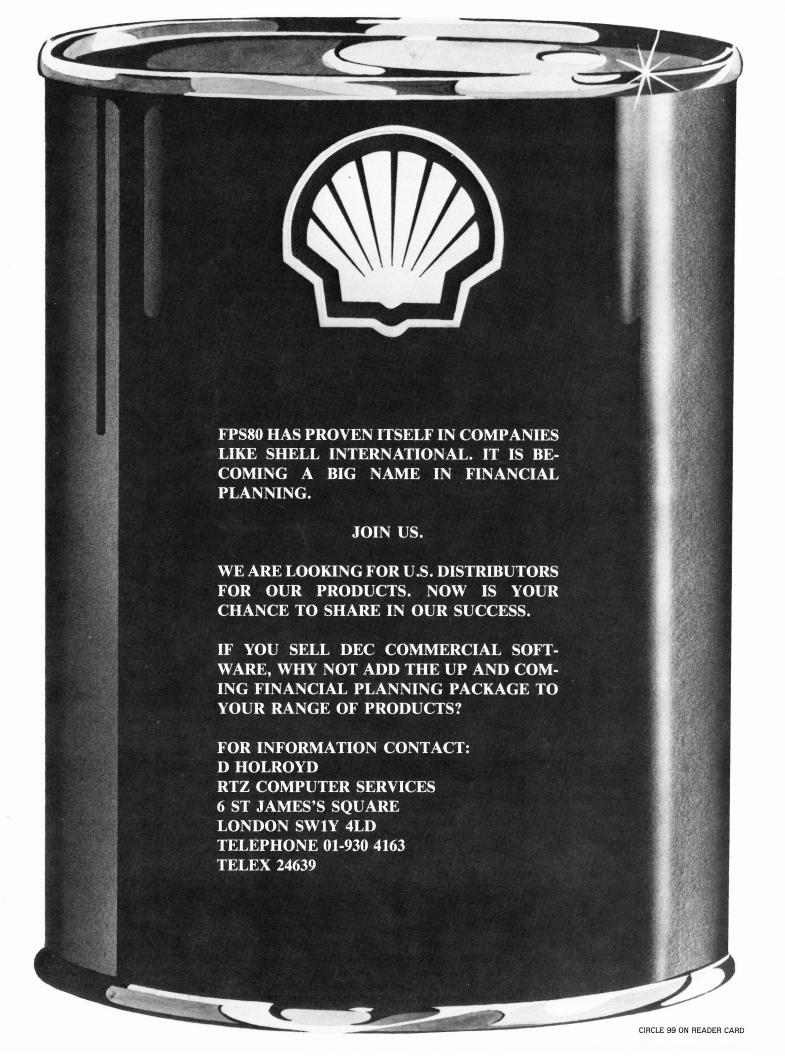
DUM = FIX(B/1000.)

B = B - FIX(DUM * 1000.)
                                                             25-JUN-82
                           V7.0-08B
                                                                                                                                                                                                                                                AND
                                                                                                                                                                                                                                                                    GOTO 29820

IF THE BALANCE FORWARD (DUM) IS LESS
THAN 1,000,000 THEN ADD THE "THOUSANDS"
AMOUNT TO THE STRING, SAVE EVERYTHING
LESS AND START OVER.
                                                             17-JUN-82
                           V7.0-08A
                                                             16-JUN-82
                                                                                                                                                                                                                                                                    B > -1.

DUM = B

DUM$ = DUM$ + 'THOUSAND '
                                                                                                     R. D. Tomlinson.
                                                                                                                                                                                                                                                                     GOTO 29820
                       OPEN 'KB: ' AS FILE #1%
                                                                                                                                                                                                                                                                                       THAN 1,000,000 THEN RESET DUM BACK TO "B" AND START OVER.
                       ... to page 36
```



(177997) DAR WHAT? ארבובו מון ביותר ביותר אל

Send questions to: DEAR RSTS MAN, P.O. Box 361, Fort Washington, PA 19034-0361.

DEAR RSTS MAN:

In your August 1982 issue you answered a letter to Jim Carrigan of LOGON systems incorrectly. If you check the actual BASIC + code, you will discover that LOGOUT will delete any file with the format XXXXnn.TMP where XXXX can be anything, and nnn is the actual job number you are running. I use this feature a great deal, and would appreciate that this correction could be noted to Mr. Carrigan. Thank you.

Paul M. Kvamme Systems Programmer/Analyst Northern Montana College P.S. Keep up the good work. Your column is very refreshing!

DEAR RSTS MAN:

Here is an additional answer to Jim Carrigan on files that disappear at logout time: Any file will be killed by LOGIN or LOGOUT whenever they are invoked if its

name has the format "xxxxJJ.TMP", where xxxx is exactly four letters or digits, and where JJ is the job number, zero-filled. See the example below. Another technique is to kill the file while you still have it open. It will seem to other users to have disappeared, but your own program will retain access to

old test

it until you close it (or chain, or end). Thank you for an excellent magazine. It's only fault is that it comes out too seldom. Phil Rand

Automotive Wholesalers, Inc. Kent, WA

```
Ready
         04 59 PM
extend
                             09-Sep-82
         500
                                                 ! Set up a 2-character string.
         j.num$ = space$(2%) !
rset j.num$ = "00" + num1$(j.num%)
         ! Move the zero-filled job no.
! Move the zero-filled job no.
! into our 2-character string.
temp. file$ = "SHOO" + j.num$ + ".TMP"
         open temp.file$ for output as file 1%, filesize 16
         end
Readu
run
TEST
                             09-Sep-82
         04: 59 PM
Readu
pip shoo?? tmp/s
                       Prot Access Date Time Clu RTS < 60> 09-Sep-82 09-Sep-82 04: 59 PM 16 BASIC
                                                                                             SY: [50, 4]
Name : Typ
SHOO15: TMP
Total of 16 blocks in 1 file in SY:[50,4]
bye
Confirm: n
Ready
pip shoo?? tmp/s
%No files matching SHOO?? TMP - continuing
Ready
```

```
THE NEXT INCREMENT OF EDITING. TO DO THIS THE MAXIMUM AMOUNT OF A SHOULD BE INCREASED BY A FACTOR OF 1000 AND A TRILLION BE THE TOP LEVEL
                                                                                     DUM < .1E 10
C = -1.
C = DUM
29870
                                                         AND
THEN
                                                                                     C = DUM
DUM = FIX(C/.1E 7)
C = C - FIX(DUM * .1E 7)
GOTO 29820
IF THE BALANCE FORWARD (DUM) IS LESS
THAN 1,000,000,000 THEN ADD THE
"MILLIONS" AMOUNT TO THE STRING, SAVE
EVERYTING LESS, SUBTRACT FROM DUM AND
                                                                                                                                                                                                                                                                                                                                                         I$ = FIX((DUM#100.)+.5)
                                                                                                                                                                                                                                                                                                29940
                                                                                                                                                                                                                                                                                                                                                         I$ = NUM1$(I$)

I$ = NUM1$(I$)

I$ = '0' + I$ UNTIL LEN(I$) > I$

DUM$ = DUM$ + I$*' '

THE DOLLARS ARE DONE, SO TACK THE CENTS ON TO

THE RETURNING STRING.
                                                                                                                                                                                                                                                                                                                                                         GOTO 29990 IF DUM$
DUM$ = DUM$ + 'CENT'
X$ = 15'
X$ = '' IF I$ = 1$
DUM$ + X$
ADD THE WORD "CENT(S)" TO THE STRING.
                                                                                                                  START OVER.
                                                                                                                                                                                                                                                                                                 29950
                                                                                       C > -1.
DUM = C
29875
                                                         THEN
                                                                                      DUM$ = DUM$ + 'MILLION '
C = -1.
                                                                                       GOTO 29820
                                                                                                                   IF "C" WAS SET AT THE TEST FOR LESS THAN 1,000,000 THEN RESET DUM BACK TO "C" AND START OVER.
                                                                                                                                                                                                                                                                                                 29960
                                                                                                                                                                                                                                                                                                                                                          GOTO 29990
                                                                                                                                                                                                                                                                                                                                                                                       THE ROUTINE IS NOW DONE SO GO BACK TO THE CALLER
                                                                                                                                                                                                                                                                                                                                                        THE ROUTINE IS NOW DONE SO GO E

DUM$ = DUM$ + 'ONE ' IF I$ = 1$

DUM$ = DUM$ + 'TWO ' IF I$ = 2$

DUM$ = DUM$ + 'THEE ' IF I$ = 3$

DUM$ = DUM$ + 'FOUR ' IF I$ = 3$

DUM$ = DUM$ + 'FOUR ' IF I$ = 5$

DUM$ = DUM$ + 'SIX ' IF I$ = 6$

DUM$ = DUM$ + 'SEVEN ' IF I$ = 7$

DUM$ = DUM$ + 'REYEN ' IF I$ = 8$

DUM$ = DUM$ + 'REYEN ' IF I$ = 13$

DUM$ = DUM$ + 'THEN ' IF I$ = 11$

DUM$ = DUM$ + 'THEN ' IF I$ = 12$

DUM$ = DUM$ + 'THEN ' IF I$ = 12$

DUM$ = DUM$ + 'THEN ' IF I$ = 12$

DUM$ = DUM$ + 'THEN ' IF I$ = 12$

DUM$ = DUM$ + 'THEN ' IF I$ = 16$

DUM$ = DUM$ + 'SEVENTEEN ' IF I$ = 14$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 16$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 16$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 16$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$

DUM$ = DUM$ + 'SIXTEEN ' IF I$ = 18$
                                                                                      DUM < .1E 13

D = -1.

D = DUM

DUM = FIX(D/.1E 10)

D = D - FIX(DUM * .1E 10)

GOTO 29820
                                                                                                                                                                                                                                                                                                 29970
29880
                                                           AND
THEN
                                                                                                                  JET THE BALANCE FORWARD (DUM) IS LESS THAN 1,000,000,000,000 THEN ADD THE "BILLIONS" AMOUNT TO THE STRING, SAVE EVERYTING LESS, SUBTRACT FROM DUM AND START OVER.
                                                                                       D > -1.

DUM = D

DUM$ = DUM$ + 'BILLION'
 29885
                                                                                        D = -1.
GOTO 29820

IF "D" WAS SET AT THE TEST FOR LESS

THAN 1,000,000,000 THEN RESET DUM BACK
TO "D" AND START OVER.
                                                                                                                                                                                                                                                                                                                                                                                       THIS SUBROUTINE ADDS THE PROPER AMOUNT TO THE RETURN STRING IF THE SUBJECT IS BETWEEN ONE
                                                                                                                                                                                                                                                                                                                                                            RETURN
                                                            GOTO 29990 IF DUMS = 25
                                                           GOTO 29990 IF DUM$ = 2$
DUM$ = DUM$ + 'DOLLAR'
DUM$ = DUM$ + 'DOLLAR'
GOTO 29990 IF DUM$

DUM$ = DUM$ + 'AND '

DEPENDING UPON THE OPTION DUM$,
THE DOLLARS HAVE ALL BEEN CALCULATED NOW, SO
ADD THE WORDS "DOLLAR(S) AND " TO THE STRING.
THE "(S)" IS OPTIONAL, DEPENDING UPON THE VALUE
OF A1$ (SEE THE FIRST FEW LINES OF CODE)
   29920
                                                                                                                                                                                                                                                                                                                                                                                         AND NINETEEN
                                                                                                                                                                                                                                                                                                                                                          DUM$ = DUM$ + 'TWENTY' IF I$ = 2$

DUM$ = DUM$ + 'THIRTY' IF I$ = 3$

DUM$ = DUM$ + 'FORTY' IF I$ = 3$

DUM$ = DUM$ + 'FORTY' IF I$ = 5$

DUM$ = DUM$ + 'SIXTY' IF I$ = 6$

DUM$ = DUM$ + 'SEVENTY' IF I$ = 6$

DUM$ = DUM$ + 'SEVENTY' IF I$ = 8$

DUM$ = DUM$ + 'NINETY' IF I$ = 9$
                                                                                                                                                                                                                                                                                                   29980
                                                                                        FIX(DUM) > 1.0
DUM$ = 'INVALID'
I$ = 99$
GOTO 29990
                                                                                                                                                                                                                                                                                                                                                                                        THIS SUBROUTINE ADDS THE PROPER AMOUNT TO THE
   29930
                                                            THEN
                                                                                                                                                                                                                                                                                                                                                                                         RETURN STRING IF THE SUBJECT IS A "TEN"
                                                                                                                                                                                                                                                                                                                                                            FNWRTOUT$ = DUM$
                                                                                                                                                                                                                                                                                                    29990
                                                                                         IF, AFTER ALL THE WORK DONE, DUM IS GREATER THAN
ANY OF THE TESTS, DUM WAS OUT OF RANGE OF THIS
ROUTINE. IN ORDER TO REPAIR THIS PROBLEM, THE
PROGRAM MUST BE MODIFIED SLIGHTLY TO ALLOW
                                                                                                                                                                                                                                                                                                                                                            ITO CALLER
END OF THE FUNCTION - TO BE REMOVED FOR BP2
                                                                                                                                                                                                                                                                                                                                FNEND
                                                                                                                                                                                                                                                                                                                                END !OF THIS PROGRAM, CHANGE TO SUBEND FOR BP2, REMOVE FOR BAS
                                                                                                                                                                                                                                                                                                   32767
```

SOLVES YOUR RESPONSE TIME MYSTERY



TIME:

Late

PLACE: Under the gun

The RSTS/E* system's slow. Performance is under par. Users are unhappy. Hardware budget is gone. Conclusive evidence is lacking.

SOLUTION:

RESPOND RESPÛND RESPOND RESPOND

RESPÛND

RESPÛND

RESPÛND

RESPÛND

RESPÛND

is Manus' newest RSTS/E tuning tool.

provides the information you need to tune your RSTS/E system.

tells you where the bottlenecks are.

measures response time exclusive of application program inefficiencies.

logs statistics and jobs mix.

reports tuning statistics in usable format.

shows resource utilizations by program, keyboard and user account.

ranks programs by resources used.

reports long-term trends in summarized form.

tells who did what, when and from which

RESPÛND keyboard.

RESPOND RESPOND warns you when free disk or small buffers are critical.

logs out inactive terminals.

YES!	My system needs RESPOND Send information kit.
Name	
Company	
Address	
Phone	ING COMPUTERS WORK FOR YOU
	SERVICES CORPORATION AUTHORIZED GIGILAL COMPUTER DISTRIBUTOR
SEATTLE: 170 PORTLAND: Kristin Squ	MANUS SERVICES CORPORATION 100 Westlake Avenue North, 98109 • (206) 285-3260 101 are, Suite 102, 9498 SW Barbur Blvd. 97219 • (503) 245-1261

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

PROFIL

By Joe Blaho, Borden, Inc. Chemical Information Systems 180 E. Broad St., Columbus, OH 43215

I. DESCRIPTION:

PROFIL.OBJ is a routine that, when linked into a task with BASIC + 2 modules will count the number of times each line of the BASIC + 2 program(s) is executed. When the program finishes or a 1C is typed at the terminal a summary of each module/line and its execution count is displayed.

II. WHY USE IT:

This utility can be used when there is a need to optimize a program for performance. It will tell you which lines are executed most frequently. (The best solution would be to report how much time is spent at each line; this, however, would be much more complicated to do.) It might also be helpful if you have a program that is stuck in a loop and need to know where.

III. HOW TO USE IT:

To use this routine compile your program as usual; however, do not use the '/NOLINE' switch on any module you want a line count on. 'BUILD' the ODL and CMD files as follows:

HISEG BP2COM,BASIC2 or NONE ACCOUNT LB:

BRL NONE (ALWAYS !)
BUILD PGMA[,PGMB,PGMn],PROFIL

This will build your command files to link to the BP2COM library, (do not use a resident library IE: BASICS), and include the PROFIL routine along with your own routines. If you are using overlays make sure PROFIL is forced into the ROOT. Task build as usual. You should get some warnings about multidefined symbols. This is expected and can be ignored. When the task builder finishes you may run your program. The results should be displayed when the program either ends, attempts to chain to another program, or you type

IV. HOW IT WORKS:

TC.

A BASIC+2 program, when compiled with the /LINE option, generates a call to a routine named LIN\$ at the

start of each line. This routine saves the line number in a special area so that should there be an error, it would report the line number that was executing when the error occurred. PRO-FIL replaces this routine and intercepts this call from within the BASIC+2 program and counts how many times a unique module/line is executed. At the end of the program BASIC+2 calls a routine named END\$ (CHN\$ if a chain to another program is requested). These calls are also intercepted so we can display the results of the count.

V. NOTES:

- a) Programs/sub-programs compiled with the '/NOLINE' option will not have their lines counted as the compiler does not generate calls to LIN\$.
- b) The double integer display routine will display execution counts between 0 and 999,999,99. If you execute a line outside of this range the results are unknown (the program would probably not finish in our lifetime anyway).
- c) If you exceed the maximum number of module names or lines the routine is assembled for (32 and 400) no new module names or line numbers will be counted; however, modules and lines already encountered will continue to be counted as they are executed.
- d) Don't expect the program to run very fast when linked with this routine. PROFIL is intended to be used as an optimizing tool and the program should be re-linked without it when you are finished.
- e) If the default table sizes (32 modules, 400 lines) are not acceptable or cause the routine to be too large, change the definitions of MAXLN or MAXNM to a more suitable number. For each line number counted it takes 8 bytes of storage, 2 bytes for each module name.
- f) The 'CHAIN' statement is disabled and programs will not chain when this routine is used.

VI. EXAMPLES:

To Assemble PROFIL.OBJ:

MAC PROFIL,PROFIL = COMMON.MAC , PRE.MAC , PROFIL.MAC Get COMMON.MAC from RSTS distribution tapes, PRE.MAC from the BP2 tapes.

BASIC2

```
BASIC2
OLD TEST1
BASIC2
LIST
                  03:14 PM
                                    18-Oct-82
TEST1
          ! TEST1 - TEST OF PROFIL ROUTINE
          FOR X=1. TO 100.
200
                    CALL TEST2
250
260
                    A=FND(X)
          NEXT X
300
1000
          DEF FND(S)
          PRINT S IF S/10. = INT(S/10.)
1005
1010
          FNEND
32767
BASIC2
COM/OBJ/LINE
BASIC2
OLD TEST2
BASTC2
LIST
                                     18-Oct-82
                  03:14 PM
TEST2
          SUB TEST2
          FOR I= 1. TO 20.
300
325
                     X=X+1.
350
          NEXT T
32767
          SUBEND
BASIC2
COM/OBJ/LINE
BASIC2
HISEG
Name [NONE]--BP2COM
Account [LB:]--
BASIC2
File spec [LB:BASICS] -- NONE
BASIC2
BUI TEST1, TEST2, PROFIL
BASIC2
TKB/SI:21 @TEST1
$TKB -- *DIAG*-MODULE $ERROR MULTIPLY DEFINES SYMBOL FLN$
*TKB -- *DIAG*-MODULE $ERROR MULTIPLY DEFINES SYMBOL LIN$
Task exit status: ERROR
BASTC2
RUN TEST 1
 10
20
 30
40
50
 60
100
PROFIL - V1.00 LINE USAGE SUMMARY
LINE #
          USAGE COUNT
    MODULE NAME = TEST1
      200
      250
                 100
                 100
101
100
      260
     1005
     1010
                 100
                 101
               = TEST2
    MODULE NAME
                 100
     300
                2100
    325
350
32767
                2000
```

```
OPEN PARAMETERS
                                                                                          TITLE PROFIL - BP2 RUN TIME PROFILE
                                                                                                        /1.03/
                                                                                          .SBTTL RELEASE INFORMATION
                                                                                                                       JOE BLAHO, BORDEN INC.
CHEMICAL INFORMATION SERVICES
180 E. BROAD ST. , COLUMBUS, OHIO 43215
                                                                                         WRITTEN BY:
                                                                                         REVISIONS:
          JMB , ADD DOUBLE INTEGER COUNTER CODE
                                                                                                        20-SEP-82
                                                                                          1.02
                                                                                                        01-0CT-82
                                                                                                                                      JMB , ADD MULTI-MODULE CODE JMB , ^C TRAP
                                                                                          1.03
                                                                                                        21-0CT-82
                                                                                        DESCRIPTION:
THIS ROUTINE WHEN LINKED WITH A BP2 PROGRAM WILL REPLACE THE
LIN$ AND FLN$ ROTUINE IN THE OTS AND RECORD THE NUMBER OF TIMES
A PROGRAM PASSES THRU A GIVEN LINE NUMBER DURING EXECUTION.
THE END$ AND CHN$ ROUTINES ARE INTERCEPTED TO PRINT OUT THE
RESULTS OF THE RUN. THE RESULTS WILL ALSO BE DISPLAYED WHEN
A 'C IS TYPED.

** NOTE ** SINCE THE CHAIN (CHN$) ROUTINE IS INTERCEPTED ,
PROGRAMS WILL NOT CHAIN . THIS ROUTINE WILL OVERRIDE
'C TRAPS THE BP2 PROGRAM MAY ENABLE.
                                                                                          .GLOBL $OTSV
                                                                                           .GLOBL CHSTK
                                                                                                                                      ; CHECKS FOR STACK OVR FLW
; CLOSES ALL OPEN CHANNELS
                                                                                           .GLOBL
                                                                                                        $CLSAL
                                                                                           .GLOBL $NOREX
.MCALL SCCA$S
                                                                                                                                          NORMAL EXIT TO DEFAULT RTS
                                                                                                                                          C TRAP ENABLE MACRO (RSXMAC.SML)
                                                                                         MAXLN = 400.
CNTOFF = MAXLN # 2.
MAXNM = 32.
                                                                                                                                      ; MAXIMUM NUMBER OF LINES TO BE COUNTED ; OFFSET FROM LINE TABLE TO COUNTER TABLE
                              000620
                                                                                                                                         MAXIMUM NUMBER OF MODULE NAMES
                                                                                                                       THE ABOVE CONSTANTS MAXIN AND MAXNM TELL THE ASSMBLER HOW MUCH SPACE TO ALLOCATE FOR THE TOTAL NUMBER OF UNIQUE LINE NUMBERS IN THE TASK BEING KUN. (PROGRAMS/SUB-PROGRAMS COMPILED WITH /NOLINE WILL NOT HAVE THEIR LINES COUNTED AND NEED NOT BE CONSIDERED.) SINCE THE MAJOR SIZE IN THIS ROUTINE IS FROM THE TABLES NEEDED TO COUNT THESE LINES (8 BYTES PER LINE, 2 BYTES PER MODULE) THESE CAN BE ADJUSTED ACCORDINGLY TO MAKE THIS ROUTINE
                                                                                         TO USE THIS ROUTINE WITH YOU BP2 PROGRAM
                                                                                                                       ASSEMBLE THIS PROGRAM AS FOLLOWS:
                                                                                                                       ASSEMBLE THIS PROGRAM AS FOLLOWS:

(PRE.MAC IS ON BP2 DISTRIBUTION TAPE, COMMON.MAC IS ON BSTS.)

'BUILD' BP2 ODL AND CMD FILES:

"BUILD' BP2 ODL AND CMD FILES:

BRL = NONE
                                                                                                         3)
                                                                                                                       TER AS HISHAL. YOU WILL GET WARNINGS ABOUT LINS
         58
59
60
61
62
63
64
                                                                                                                        AND FINE BEING DOUBLY DEFINED BUT THIS IS OK.
DO NOT LINK AGAINST A BASIC RESIDENT LIBRARY
(BASICS) AS IT WILL NOT USE THESE RE-DEFINTIONS
OF LINE AND FINE.
                                                                                          .SBTTL LIN$ - COUNT LINES AS THEY ARE EXECUTED .PSECT PROFIL,RO,I,GBL,CON,REL
          66
          67 000000
68
69 000000
70 000000
                                                                                                                                                      : FUNCTION LINE (INTERNAL RE-ENTRY)
                                                                            FI.N $ : :
                                                                                                                                                      ; LINE NUMBER ENTRY
: SET ^C TRAP (AGAIN !)
                                                                                                         #CCEND
          71 000000
                                                                                                        R4, @$OTSV
          72 000006
                              010477 000000G
                                                                                          MOV
PUSH
          73 000012
74 000014
75 000016
76 000020
                                                                                          PUSH
                                                                                                         R1
                                                                                                         R3
           77
78 000022
                                                                                                                                                         GET OBJ TIME SYS POINTER IN RO
GET CUR MOD NAME POINTER IN R1
R3 => NAME TABLE
R0 = NUMBER OF MOD NAMES SO FAR
                                                                            10$:
                                                                                                          NMPTR(RO),R1
           79 000026
80 000032
                               016001
                                             000002
                                                                                          MOV
                000026
000032
000036
000042
                               012703
016700
001407
                                                                                                         #NAMTBL ,R3
NUMNAM ,R0
                                                                                                                                                      : FIRST TIME ??
                                                                                                                                                     ; SEE IF MODULE HAS BEEN USED BEFORE
; YES, IT HAS
; LOOP UNTIL LINE # FOUND OR END
                                                                                          CMP
                000044
                                                                            12$:
                               022301
                000046
                               001410
                                                                                                          20$
                                                                                                         RO . 12$
                000050
                                                                                            ; FALLING THRU HERE MEANS THIS MODULE HAS NOT BEEN REFERENCED BEFORE
           89
                                                                                                                                                       ; ONLY CHECK TO MAX NBR OF MODULES
                                                                                                          #MAXNM , NUMNAM
                000052
                               022767
                                             000040 001056
                                                                                           CMP
                                                                                                                                                      ; SORRY , TOO MANY MODULES
                000060
                                                                                           BEQ
                                                                                                          50$
                                                                                                                                                       : INCREMENT NUMBER OF MOD'S REFERENCED
                                                                            15$:
                                                                                                          NUMN AM
                000062
                               005267
                                              001050
                                                                                                                                                          STORE THIS NAME POINTER
                000066
                000070
000072
                               012401
005743
                                                                                                          (R4)+, R1
-(R3)
                                                                                                                                                         MOVE LINE NUMBER 0 - 32767 => R1
                                                                                                                                                         BACK OFF TABLE POINTER

R2 => TABLE OF LINE NUMBERS

R0 = NBR OF LINES REFERENCED

FIRST TIME , ADD TO TABLE
          100 000074
                                                                                                          #LNTBL , R2
                               012702
                                                                                           MOV
                000100
                                              001030
          102 000104
                                                                                           BEO
                                                                                                          30$
         103
104 000106
105 000110
106 000112
                               022213
001402
005722
                                                                                                                                                      ; SEE IF MODULE POINTER HAS BEEN USED ; MOPE... FORGET LINE # CHECK ; POINT AROUND LINE NUMBER
                                                                                                          (R2)+ , (R3)
                                                                                                          (R2)+
          107 000114
108 000116
                               000402
                                                                                                          (R2)+ , R1
                                                                                                                                                      : NOW CHECK LINE NUMBER
                                022201
                                                                            27$:
                                                                                           CMP
                                                                                                                                                       ; YES... ADD TO THIS LINE CNTR
                000122 077007
                                                                                           SOB
                                                                                                          RO , 25$
                                                                                                                                                      ; LOOP UNTIL LINE # FOUND OR END
                                                                            28$:
          112
                                                                                           ; FALLING THRU HERE MEANS THIS LINE HAS NOT BEEN REFERENCED BEFORE
          113
                                                                                                                                                      ; ONLY CHECK TO MAX NBR OF LINES ; SORRY , SHOULDNT WRITE PROGRAMS LIKE ; THIS ANYWAY
                000124 022767 000620 001002
                                                                                                          #MAXLN , NUMLN
                000132 001412
```

Catch that address list duplication . . . Before it goes off with a stamp!

New Address List Processing Package (ALP) for DEC Operating Systems, from McHugh, Freeman

List Creation: Creates uniquely structured lists with identifiers and 12 data fields

Address Entry: Enters data via keyboard to formatted screen or from magtape lists

Duplicate Entry Detection: Detects duplications at moment of keyboard entry

List Maintenance: Purges complete lists; updates and deletes; appends additional data fields; displays entries and removes to tape or backup file

List Extraction: Extracts/merges lists; selects each field for range or inclusion/exclusion processing

Envelope/Label Production: Prints 3" or wider labels, 1-up to 4-up; prints envelopes in alphabetical or zip code order

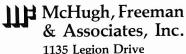
Report Generation Interface: A standard feature; interface with DEC DATATRIEVE-II®

Optional WP Interface: Full list/ merger capabilities with DPD's WORD-11® or DECword/DP®

Try these other McHugh, Freeman packages to create applications, maintain systems, and protect data:

- Automatic Password Changer
 (APC) Encryption Routine
 Keyboard Master (KBMSTR)
- Menu/Authorization Processor
 (M/APS)
 Source/File Cross
 Reference (XREF)
 Standard
 Subroutine Library (SUBS)
 VT100 Accounting Calculator
 (CALC)
 System Support Services

For more information, call McHugh, Freeman and Associates, Inc. at 1-414-784-8250 and . . . let us ALP-atize your mailing lists!



1135 Legion Drive Elm Grove, WI 53122

DATATRIEVE and DECword/DP are registered trademarks of Digital Equipment Corp. WORD-11 is a registered trademark of Data Processing Design.

LETTERS to the RSTS Pro

... continued from page 6

April 29, 1982 Greg Steinkuhler TRT Telcommunications P.O. Box 8876 Ft. Lauderdale, Fl. 33310

Dear Greg:

I saw your memo in THE DATA MANAGERS for Feb., 1982. I am glad that the TECO SIG is trying for a comeback. I think that you might be interested in reading my views about TECO.

I started in the DEC world on a PDP-8 about 15 years ago. The OS/8 line editor was good enough. About 1976 I started to get into RT-11 and made the unhappy discovery that the RT-11 'standard' editor was not the same as the OS/8 editor. In mid-1977 I started working on RSX-11M and once again I was faced with having to learn a different editor. Rather than learn that RSX editor I started using TECO. TECO seemed to be essentially the same editor whether I was using OS/8, RT-11, or RSX-11M. Also, TECO was available on RSTS/E and RSX-11D and TOPS-10. In other words I saw it as the true DEC Standard Editor. I didn't particularly like TECO, I wasn't turned on by its editing power, nor did I use much more than a fraction of its capabilities. I just didn't want to have to learn and remember 3 or 4 different editors.

As PCS grew from a one man operation to two, three, and four, I made TECO our one and only editor. VTEDIT came along in 1979 and our editing efficiency seemed to increase 100% over standard TECO.

Then came the rude awakening. When we got to 2 people TECOing on RSTS/E or RSX-11M things got slow. With 3 people, it was very slow. With 4 people it was impossible to get any work done. It seems that from V28 to V35 and V36 TECO itself had grown quite large. Using VTEDIT compounded the problem. Early in 1980 we dumped TECO and went to KED on RSX-11M and RSTS/E. As long as we stuck to the PDP-11 we had an editor that was operating system independent. We found that on RSX-11M and RSTS/E we could support 5 KED's with about the same throughput as 2 TECO's.

On our RT-11 systems I found that TECO v36 using VTEDIT is too big for the XM monitor and doesn't work right with the FB monitor if the USR swaps. Since we rarely use the SJ monitor, TECO had to be abandoned on RT-11 as well.

1	19 20 000134 21 000140 22 000142		000774		30\$:	INC MOV MOV	NUMLN (R3) R1	,(R2)+ ,(R2)+	; INCREMENT NUMBER OF LINES REFERENCED ; MODULE NAME INDEX TO FIRST WORD ; STORE THIS LINE NUMBER
1	23 24 000144 25 000146 26 000152 127 000156	005742 062702 062712 005542	001440 000001		40\$:	TST ADD ADD ADC	-(R2) #CNTOFF, #1. -(R2)	R2	; BACK OFF POINTER; R2 => COUNTER TABLE (LOW WORD); ADD TO LINE COUNT; OVERFLOW INTO DBL INTGR (HIGH WORD)
	128 129 000160 130 000162 131 000164				50\$:	POP POP POP	R3 R2 R1		; RESTORE MESS
	132 000166 133 000170 134 000174	000134				POP CALL JMP	RO CHSTK @(R4)+		; CHECK FOR STACK OVRFLW ; EXIT
i	137						FINISH UF	ROUTINE	
į	138 000176 139 000200 140	104026 104020			CCEND::	.TTRST			; ^C ENTRY POINT ; CANCEL ^O EFFECTS OF ^C
	141 000202 142 000202 143 000204	012700	000670'		END\$::	PUSH MOV	R4 #PROMPT,	RO	; SAVE R4 ; RO => PROMPT ADDRESS
- 6	144 000210 145 000214		000102			MOV	#PRMLN, PRINT	₹1	; R1 => PROMPT LENGTH . PRINT THE BANNER
19		012704 016702	0011401		55\$:	MOV MOV	#NAMTBL NUMLN	, R4	; R4 => TABLE OF MODULE NAME PTR'S ; R2 = NBR OF LINES USED
22	148 000224 148 000230 149 000234	012703	001240		JJ	MOV MOV		,R3 ,RO	; R3=> LINE TABLE
	150 000236 151 000242	012037 012037	001014'			MOV	(RO)+ (RO)+	, @#MPNAM , @#MPNAM+2.	; COPY MODULE NAME TO OUTPUT LINE
	152 000246 153 000252	012037 012700	001020'			MOV MOV	(RO)+ #MPRMPT	, @#MPNAM+4.	; RO => MODULE NAME LINE BUFFER
11.19	154 000256 155 000262	012701	000036			MOV CALL	#MPRLEN PRINT	,R1	; R1 = BUFFER LENGTH ; PRINT CURRENT MODULE NAME
	157 000266 158 000270	022314			60\$:	CMP BNE	(R3)+ 65\$,(R4)	; IS THIS LINE FOR CURRENT MODULE ? ; NOPE
	159 000272 160 000276	012700 012701	001030'			MOV	#OUTBUF		; GET READY TO SPACE FILL OUTPUT LINE ; LINE LENGTH / 2
	161 000302 162 000304	005741			62\$:	TST	-(R1) #8224.	,(RO)+	; DONT CLEAR THE <cr> <lf>; 2 SPACES</lf></cr>
	163 000310 164 000312	077103	020010			SOB PUSH	R1 (R3)	,62\$; LOOP UNTIL ALL SPACE FILLED ; PUSH LINE NUMBER
	165 000314 166 000316	010601				PUSH	SP	, R1	; CLR HIGH WORD ; R1 => ARG (HIGH WRD OF DBL INT)
	167 000320 168 000324		001030'			MOV	#OUTBF1 CVTISD		; RO => BUFFER FOR STRING CVT ; CONVERT DBL INTGR TO DECIMAL STRING
	169 000330 170 000332					POP POP			; RESTORE STACK POINTER
	171 000334 172 000336	010300 062700	001440			MOV	R3 #CNTOFF	, RO	; RO => USE COUNT TABLE
	173 000342 174 000344	002100	001440			PUSH PUSH	(RO) -(RO)	,	; PUSH LOW WORD ONTO STACK ; PUSH HIGH WORD
	175 000346 176 000350	010601	0010441			MOV	SP #OUTBF2	,R1	; R1 => HIGH WORD OF DBL INT COUNT : R0 => BUFFER AREA FOR CVT
	177 000354 178 000360	012100	001011			CALL	CVTISD	,	; CONVERT INTEGER TO DECIMAL STRING ; CLEAN UP STACK
	179 000362 180 000364	012700	001030'			POP	#OUTBUF	, RO	; ; PRINT BUFFER
	181 000370 182 000374	012701	000030			MOV	#OUTLN PRINT		; BUF LENGTH ; PRINT A LINE
	183 000400 184 000402	005723 077247			65\$:	TST SOB	(R3)+ R2,60\$; ADD 2 TO ARRAY POINTER ; LOOP FOR ALL LINES IN TABLE
	185 000404 186 000410	005367	000526			DEC BEQ	NUMNAM 67\$; DECREMENT MODULE NAME COUNTER ; NO MORE
	187 000412 188 000414	005724				TST BR	(R4)+ 55\$; POINT TO NEXT NAME POINTER ; LOOP AGAIN
	189 190 000416				67\$:	CALL	\$CLSAL		; CLOSE ALL CHANNELS
	191 000422 192 193	000167	000000G			JMP	\$NOREX		; EXIT TO RTS
	195 196				;+	.SBTTL	CVTISD -	- DBL INTEGER TO	ASCII CVT
	197 198				•	THIS RO	UTINE WI	LL CONVERT A POS	ITIVE DOUBLE INTEGER INTO AN
	199 200				; ;	ASCII S SUPPRES	TRING NU SED, RIG	MERIC. RANGE 0 - HT ALIGNED ON 9	999,999,999. LEADING O'S BYTE FIELD
	201 202 203				ļ	INPUT:		STRING BUFFER FO	R OUTPUT (9 BYTES AT LEAST) INTEGER
	204 205				;	OUTPUT:		9 BYTE CHARACTER	
	206 207				; ;-				
	208 209 210 000426				CVTISD:	usud	R2		; SAVE REGISTERS
	211 000420 211 000430 212 000432	005002			CVIIDD.	PUSH	R3 R2		: TURN ON ZERO SUPPRESS
	213 000434	011167	000464	000460		MOV	(R1) ,W 2(R1),W	ORKH ORKL	; SAVE NUMBER IN WORK AREA
	215 000446 216 000454	016767		000454		MOV	WORKH, T	MPH MPL	; AND IN TEMP AREA
	217 000462 218	012701	0010601			MOV	#X8	, R1	; R1-> PWRS OF 10 TABLE
	219 000466 220 000470	005003 166167	000002	000430	5\$: 10\$:	CLR SUB	R3 2(R1)	, WORKL	; ZERO SUBTRACTION COUNTER ; SUBTRACT LOW WORDS
	221 000476 222 000502	005667 161167	000422 000416			SBC SUB	WORKH (R1)	, WORKH	; SUBTRACT CARRY FROM HIGH WRD ; SUBTRACT HIGH WORDS
	223 000506 224 000510	100410	00-1-1	0000		INC	20\$ R3	TMDU	; GONE NEGATIVE ?? ; NO BUMP SUBTRACT COUNT
	225 000512 226 000520 227 000526	016767 016767 000760	000406 000402			MOV MOV BR	WORKH WORKL 10\$, TMPH , TMPL	; SAVE LAST POSITIVE VALUE ; KEEP SUBTRACTING UNTIL '-'
	228 229 000530	005702			20\$:	TST	R2		; STILL ZERO SUPPRESSING ?
	230 000532 231 000534	001005 005703				BNE TST	25\$ R3		; NO PRINT 'O CHAR ; IS CHAR A O.
	232 000536 233 000540	001003 012703	000040			BNE MOV	25\$ #32.	, R3	; NO GO CONVERT TO ASCII ; ZERO SUPPRESS WITH SPACES
	234 000544 235 000546	000404 062703	000060		25\$:	BR ADD	27\$ # 48.	,R3	; SKIP TO CHAR MOVE ROUTINE. ; MAKE SUB COUNTER ASCII NBR
	236 000552 237 000556	012702	177777		27\$:	MOVB	#-1. R3	,R2 ,(R0)+	; TURN OFF ZERO SUPPRESS ; STORE CHAR IN OUT BUFFER
	238 000560 239 000564	062701 020127	000004 001120			ADD CMP	#4. R1	,R1 ,#X0	; MOVE TABLE POINTER TO NEXT ; POWER OF 10

```
240 000570
                    101007
241 000572
242 000600
                   016767
016767
                                                                                                 . WORKH
                                000326
                                            000320
                                                                       MOV
                                                                                   TMPL
                                                                                                 .WORKL
 243 000606
                                                                       BR
                                                                                   5.5
                                                                                                                                      : LOOP UNTIL ALL POWERS CVT'D
244
245 000610
246 000612
247 000614
                                                                       POP
                                                                       RETURN
250
251
252
                                                                                 PRINT -
CLRXRB
RO
                                                                                              OUTPUT A LINE AT TTY
251 000616
252 000622
253 000626
                    010037
                                000446
                                                                                                 . @#XRB+XRLOC
                                                                                                                             STORE BUFFER ADDRESS
                   010137
                                000442
                                                                       MOV
                                                                                   R1
                                                                                                 . @#XRB+XRLEN
                                                                                                                             BUFFER LENGTH
254 000632
                   010137
                                000444
                                                                       MOV
                                                                                                 . @#XRB+XRBC
                                                                                                                             TRANSFER COUNT
255 000636
256 000640
257
258
                                                                       .SBTTL
                                                                                   CLRXRB - GUESS ??
259 000642
260 000644
                                                         CLRXRB: PUSH
                                                                                                                           SAV REG'S
                                                                       PUSH
261
262
263
      000646
000652
000656
                   012702
012703
005022
                                                                                   #XRB ,R2
#XRBSIZ/2. , R3
                                                                                                                             XRB LENGTH / 2
CLEAR NEXT WORD OF XRB
                                                         70$:
                                                                                   (R2)+
                                                                                   R3 , 70$
R3
R2
264 000660
                   077302
                                                                       SOB
                                                                                                                            LOOP TILL DONE
                                                                       POP
POP
RETURN
265 000662
      000664
000666
266
                                                                                                                            RESTORE REG'S
                                                                                   CONSTANTS AND WORKING STORAGE
/* PROFIL - V1.00 LINE USAGE SUMMARY *
13.,10.,13.,10.
/LINE # USAGE COUNT/
271 000670
272 000736
                                                                                                      , IU. ; <CR><LF><CR><LF><CR><LF><
                                     012
                        015
114
                                                  015
116
                                                                       .BYTE
273 000742
                                      111
                                                                       .ASCII
274 000770
                                     012
                                                                       BYTE
                                                                                   13.,10.
275
276
277
278
                                                                       EVEN
.-PROMPT
.ASCII /*** MODULE NAME = /
.ASCII /NONAME ***/
                                                                                                                          ; SET UP PROMPT AND LENGTH
                        116
                                                  116
       001014
                                     117
                                                         MPNAM:
279 001026
280
                                     012
                                                                       BYTE
                                                                                   13.,10.
                    000036
                                                          MPRLEN= .-MPRMPT
                                                                                                                          ; MODULES NAME BANNER
281
282
283
284
                                                                       .EVEN
       001030
                                                         OUTBUF:
OUTBF1: .BLKW 5.
.WORD 8224.
OUTBF2: .BLKW 5.
.BYTE 13.,10.
OUTLN = .-OUTBUF
      001030
001042
001044
                                                                                                                             LINE NUMBER FIELD
285
286
                   020040
                                                                                                                          : TWO SPACES
287
288
289
290
      001056
                                                                                                                          ; <CR> <LF>
                                     012
                                                                       POWERS OF 10 (DOUBLE INTEGER REPRESENTATION) USED IN CVTISD
291
292 001060
293 001062
294 001064
                   002765
160400
000230
                                                          x8:
                                                                       WORD
                                                                                   1525.
57600.
                                                                                                                          ;100,000,000
                                                         X7:
                                                                                   152.
38528.
                                                                                                                         ; 10,000,000
295 001066
                    113200
                                                                       .WORD
296 001070
297 001072
298 001074
                                                                                                                         ; 1,000,000
                   000017
                                                          X6:
                                                                       .WORD
      001070
001074
001076
001100
                    041100
                                                                        WORD
                                                                                    16960.
                                                          X5:
                                                                                   1.
34464.
                    103240
                   000000
                                                         X4:
                                                                       . WORD
                                                                                                                                   10,000
      001102
                                                                                   10000.
                   023420
                                                                       -WORD
301 001102
302 001104
303 001106
304 001110
305 001112
306 001114
                   000000
001750
000000
                                                         X3:
                                                                                                                                     1.000
                                                                                   0.
1000.
                                                         X2:
                                                                                                                                        100
                                                                                   0.
100.
                   000144
                                                                       .WORD
                   000000
                                                         X1:
                                                                       . WORD
                                                                                                                                         10
                                                                                   0.
306 001114
307 001116
308 001120
309 001122
310
311 001124
                   000012
                                                                       WORD
                                                         XO:
                                                                                   0.
1.
                   000000
                                                          WORKH:
                                                                       .WORD
                                                                                   0.
                                                                                                                          ; WORK AREA FOR CVTISD ROUTINE
311 001124
312 001126
313 001130
314 001132
315
316
                   000000
                                                          WORKL:
                                                                       .WORD
                                                                                   0.
                                                                       WORD
316
317 001134
318 001136
319 001140
320 001240
321 004340
322
                   000000
                                                          NUMLN:
                                                                       . WORD
                                                                                                                            NUMBER OF LINES REFERENCED
                                                          NUMNAM:
NAMTBL:
                                                                                                                             NUMBER OF MODULES REFERENCED TABLE OF MODULE NAMES POINTERS
                                                                       . BLKW
                                                                                   MAXNM
                                                                                   MAXLN # 2.
MAXLN # 2.
                                                                                                                          ; LINE COUNTER AREA (DBL INTGR)
                                                         LNCNT:
323
324
325
326
327
328
329
330
331
332
333
                                                                       NAMTBL ENTRIES ARE WORD POINTERS TO A SIX BYTE STRING WITH THE MODULE NAME IN IT. (COURTESY OF BP2 OTS)
                                                                       LNTBL ENTRIES ARE DOUBLE WORD ENTRIES . ONE PER MODULE/LINE
                                                                                               ! MODULE NAME PTR ! LINE NUMBER
                                                                           ADDRESS =>
                                                                       LNCNT ENTRIES ARE DOUBLE WORD INTEGERS THAT HOLD THE NUMBER OF
TIMES A LINE HAS BEEN ACCESSED. DOUBLE WORD #1 HOLDS THE COUNT
FOR LNTBL ENTRY #1 .... ETC.
 334
                   000001
                                                          END
SYMBOL TABLE
ABORT = 000014
                                      CURLOX= 000014
CURS1 = 000176
CURS2 = 000202
                                                                            FLGDP = FLGFRC
                                                                                                                  IF.FIX= 000040
IF.FLB= 020000
                                                                                                                                                         JOBNUM= 000040
                                                                                        020000
AJMPV = 000042
                                                                            FLGKB
                                                                                        040000
                                                                                                                   IF.FRC= 000040
                                                                                                                                                        KEY 000400
KH.SIZ= 000006
ALTCHN= 000150
                                      CVTISD
                                                  000426F
                                                                      003 FLGMOD
                                                                                         010000
                                                                                                                   IF.FTB= 040000
                                                                                                            IF.IGE 020000
IF.IIN= 000001
IF.ILS= 000004
003 IF.IND= 000006
                                                                                                                                                        K.ACH = 000002
K.ADP = 000001
K.CHG = 000002
K.DUP = 000003
ASSFQ
            000024
                                      CVTTHR= 000246
                                                                            FLGPOS
                                                                                         004000
AUDHND
            000032
                                                                                        100000
000334
000000RG
BADCLU= 000037
BADCNT= 000037
BADDIR= 000001
                                                                            FLTERR= 000060
                                      DATLOC= 000034
                                                                                                                  IF. INP= 000007
                                                                                                                                                        K.LEN = 000001
                                                                                                                                                                    000000
000000
000000
000004
BADFUO= 000022
                                      DDNFS
                                                  000400
                                                                            FLTSAV= 000322
                                                                                                                  IF. IPT= 000400
                                                                                                                                                         K.NCG =
                                                                                                                  IF.LIS= 000400
IF.LIS= 000002
IF.MAT= 004000
IF.NCR= 010000
IF.NEG= 100000
                                                                                                                                                        K.NCG =
K.NDP =
K.NUM =
K.OFF =
BADKEY= 000217
                                                  001000
                                                                            FMTCHR= 000133
BADNAM= 000002
BADOPN= 000242
BADPAK= 000032
                                      DDWLO
DEAFQ
                                                  002000
                                                                            FNCCOD= 177776
FNRF$ = 000120
                                      DEBGR =
                                                  000002
                                                                                                                                                         LENGTH= 000030
BADRFA= 000255
                                                  000344
                                                                             FNSF$ = 000110
                                                                                                                   IF.NKY= 040000
                                                                                                                                                                    000102
BADRSZ= 000224
                                      DEFMRG= 000127
                                                                            FOVRFL= 000060
                                                                                                                  IF.POS= 000010
```

We still use TECO on our PDP-8's, or for the complex editing jobs that KED can't handle, but TECO is definitely out as our main editor. TECO is just too much of a pig for our multiuser environment.

PCS is hardly alone in its abandonment of TECO. Almost all of my RSTS/E and RSX-11M customers and acquaintances have abandoned TECO. Many more potential TECO users never used it because they learned, through the local DECUS community, of the problems of the ex-TECO users.

So where does this leave TECO? Well, probably not too well off:

• DEC decided years ago that TECO was not going to be the DEC Standard Editor and designed yet another editor which they call EDT. The idea that EDT is a universal editor is a joke because even some of the PDP-11 operating systems still do not have EDT, and OS/8 and other traditional operating systems will never have it.

DEC actively (but gently) discourages new users from getting into TECO. Naturally enough, their software support specialists push the DEC product.

- DEC does not distribute TECO with all their operating systems any more, and rumor has it that they will stop it altogether. For example, the RSX-11 group stopped with v3.2 in early 1979 and VAX/VMS never included TECO. So, it is harder for a new user to even find out about TECO.
- In a multi-user environment, TECO is its own worst enemy. Consequently, there are ex-TECO users spreading the word.

So, to the bottom line: if the TECO SIG is going to be resurrected, I think the SIG's initial efforts should be:

- Make TECO competitive (in terms of system resources) with KED and EDT. I think the last thing TECO needs is more neat features and commands, i.e. more resources from the poor computer that has to run it.
- Rewrite VTEDIT to be compatible with KED. The keypad used for KED is the de-facto standard even if VTEDIT came first.

Despite what I have written, I am not anti-TECO. But I am pragmatic. I cannot afford to use TECO on our computer systems. If TECO ever becomes as economical as KED or EDT in terms of system resources, I would not hesitate to use it again, nor would I hesitate to tell anyone who would listen what a great, lean little editor TECO had evolved into.

Yours Truly, Process Control Systems, Inc. Michael E. Mazzoni, President RSTSPROFESSIONALRSTSPROFES

Overworked Disks Making You Work Overtime?



DSKBLD Restructures Them Fast!



Fragmented disk structures cause excessive overhead, slowing system response.
Rebuilding them with DSKBLD is fast and easy.

DSKBLD

- is an extremely fast RSTS/E* disk to disk utility.
- optimizes disk structures automatically.
- greatly reduces File Processor overhead.
- · is lightning fast.
- makes disk rebuilding a snap.
- dramatically improves performance of diskbound systems.

For more information, attach your business card to this ad and send to:

MANUS Services Corporation 1700 Westlake Avenue North, Seattle, Washington 98109, (208) 285-3260.

European Distributor: DCS Dialog Computer Systeme AG, Maienstrasse 39, CH-8050 Zurich, Switzerland

Australian Distributors: Victoria: Scientific Business Software Systems, 133 McCrae St., Bendigo, 3550,

(054) 42-4322 **W.A.**: On-Line Computing, 14-16 Rowland St., Subiaco, 6008, (381-3000)

N.S.W.: Nicholson & Partners, 27 Lindfield Avenue, Lindfield, 2070, (02) 467-2000



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

LIMS1 = 000200 TF. REL = 000004 DELF\$ = 000140 BDNERR= 000064 LIMS2 = 000204IF.RMS= IF.RND= = 000374 FPTR1 = 000100 LINERR= 000057 DETKEY= 000033 DEVNFS= 000036 FPTR2 = 000104 000000 000000RG 004340R FORST 7 000040 IF.ROK= 000010 BOTSTR = 000014 FQBUFL FQCLUS 000020 TF. SEO= 000002 LIN\$ LNCNT BSERR = 000056 DFLAG = 000056 IF.SMP= 010000 BSTART= 000154 DIMED2= 000151 IF.STR= 000400 LNTBL 001240R BSTRP = 000152 DIRERR= 000375 FQDEV 000030 LOGERR= LOKFQ LOOKAD= 000065 DIRFQ DIVBYO= 000012 FODEV N 000032 000022 000075 FOERNO 000004 IF.TRM= 000100 BUFOFL= 000241 000004 TF.UDF= 000020 BYTCNT= 000034 DLNFQ IF.UNS= 001000 IF.USM= 004000 IF.VAR= 000100 LOPADR= 000222 BYTCNX= B.STAK= DMCHND 000012 000030 FQFIL LOSTAK= 000046 DOPMD = 000200 FQFLAG 000021 000000 DSKHND FOFUN 000003 B.SWAP= 000045 FQJOB FQMODE FQNAM1 000002 IF. VFC= 000200 DTAHND B. 10 = 000042 IF. VIR= 001000 B.250 = 000043 DTOOOF: 000021 IF.WRT= ILLACC= ILLOPR= MAGSEL= 000046 000020 EDATA = 000032 CALFIP 104000 CCEND 000176RG CCPEND= 100000 MARGIN= 000126 000157 FQNAM2 000020 MATOPR= 001000 EDEXON= EFNO = EFNOB = 000160 FONENT 000036 MAXLN = 000620 MAXNM = 000040 FQPPN FQPROT FQSIZ FQSIZM 000027 000006 ILLRES= 000367 000021 ILLRFM= 000247 CDRHND 000014 MFPTR = 00000 000370 CHANO 000015 000030 ILLSHR= 000250 MINVER= 000070 CHATBL= 000022 EFN1B 000200 MISATR= 000240 003 END\$ EOF 003 EPEND 00020286 003 FUCORE= 000147 MLEN 000006 000013 FUNDEL = 000060 IND = INPF\$ = MOFST = 000010 MPNAM 001014 MPRLEN= 000036 000020 FUNERR = 000130 000642R CLR*RB FZDIV = 000075 GSBCOD= 177777 GTRF\$ = 000060 003 TNTLCK= 000023 000000 INTPAK= 000031 CLSF\$ 000040 ERLPTR= 000142 **003** INTSAV= 000322 MPRMPT 000772R 001440 ERN1 000206 MTAHND 000016 000460 000210 GTSF\$ = 000050000376 HEDSTR= COO124 CRBFQ CREFQ HNGDEV = 000016 HNGTTY = 000017 IASSYS = 000026 IOFLGX= 177772 MTMP10= 000420 MTMP11= 000422 MTMP12= 000424 MTMP13= 000426 000001 TOFLG1= 000136 ERRNUM= 000136 CRITAL= 040000 TOFLG2= 000140 000032 EXITNR= 000111 IOFLHD= IOLERR= 000214 000034 EXITTM= 000110 IC1LEN= 000200 MTMP14= 000430 FABSTX= 000052 MTMP15= 000430 MTMP15= 000432 MTMP16= 000434 MTMP17= 000436 FIELDE= 000077 FIEXST= 000020 020000 IF.AMP= 000400 JFBIG JFFPP CURALX= 000133 177774 IF.BLK= 000000 001000 CURBKX= JFI.OCK 040000 CURBLK = 000052 FINMOV = 000300 010000 MTMP2 = 000400CURBMX= 177773 CURCHN= 000024 CURLOC= 000036 FIROB 000402 IF.CON= MTMP3 = 000402FIXERR= 000063 IF.ECC= MTMP4 = 000404 FLFDFL= 010000 IF.FEX= 002000 000734 000736 000400 O.KSZ = O.LUN = PTTAB = 000130 PUADR = 000112 MTMP5 = 000406000034 000410 000006 SCATHP= 000074 0.MAP = 0.MFY = 0.MOD = PULEN = 000114 SCAVAL= 000062 SCRLUN= 000016 USRSP PUTBL P.BAD P.BPT UUOFO 000014 SDATA = 000016 SIZALL= 001576 SIZCHT= 000100 UU. ACT 17776 MTMP9 000416 000000 UU.ASS UU.ATR UU.ATT NAMTBL 001140R 003 O.MODS= 000026 NEDERR= 000073 O.NAM = 000010 P.CC NEGLEN = 000246 O.NON = 000006 P. CRAS 177744 SIZERR= 000054 NKEYS = 000135 NMPTR = 000002 NOBUFS= 000040 177734 177762 177742 177732 177765 O. NRW 000004 P. DEXT SIZHDR= 000154 UU.BCK UU.BYE P. EMT SIZIBF= 000400 000009 P.FIS P.FLAG P.FPP STZM 000012 UU.CCL 177750 SIZMST= SIZOTS= SIZTBF= 000012 000122 000446 000554 IIII CHE 000023 NODEVC= 000006 O.ORG 000001 UU.CHU UU.CLN UU.CNV 000023 000010 0000024 000014 NOMATH= 000175 O.OUT NOPRKY= 000226 O.REC 000014 P. IOT 177760 NORACS= 000100 NOROOM= 000004 O.RED = 000003 P. ISIZ 177736 SLOT = 000134 SLPST = 000400 000003 P.MSIZ 177740 UU. DAL SP = \$00006 SPSAV = 000340 SQRERR= 000066 STATUS= 000146 NOSUCH= 000005 NOTAVL= 000010 O.RMT = 000003 O.SEQ = 000002 P. NEW 177750 UU. DAT 177762 P.OFF UU. DEA 000013 00000 NOTCLS= 000007 0.SPA = 000000P. RUN 177752 177760 NOTIMP= 000372 O.SPN = 000005 P.SIZE 177771 000372 000025 000101 000011 O.STR = 000004 O.TMP = 000010 O.TRM = 000000 O.TRN. = 000004 P.STRT 177746 STCNT = 000156 SUBARG= 000130 UU. DIR P.TRAP 177764 P.2CC 177772 Q\$\$\$ = 000002 RABLEN= 000120 000001 SUBCOD= 177775 UU.ERR SUBERR= 000067 UU.FCB 177770 NOTUSE= 000144 177767 000030 000004 SUBREC= 000373 SUBSAV= 000322 UU. HNG NSTORG 001000 0.TSK = 000020NIII.HND 000026 0.UDF = 000000RABSTR= 000004 NUMLN NUMNAM RBYTES= 000130 000166 UU.LIN RECADR= 000032 RECAD2= 000036 RECLEN= 000030 RECOUN= 000150 SYSVEE 000103 UU.LOG 177755 SYSVEL 033060 IIII. LOK 000021 TEMP1 = THREAD= TLCNTD= 000003 ONBAD = 000072 UU.NAM UU.NLG UU.PAS 177757 177776 000000 ONERDB= 000346 O.WRI = 000002 ONERGO= 000110 PAKLCK= 000026 RENFQ PBFPTR= 000152 PC = \$000007 PF.CSZ 020000 000002 RESERR= 000150 TLNOFN= 000141 0000030 RESF\$ = 000150TLTMAF= 000131 UU. POK 177772 RJEHND RMERBG= TMPRKS= 000070 UU.PRI 177763 OPSYS TMPBLS= 000066
TMPH 001130R
TMPHDR= 000440 IIII. RAD 000016 PF.EMT OP.SIZ= 000032 100000 177756 000025 000027 RMSERR= 000377 OUTBF1 001030R 003 PF.KBM OUTBF2 001044R 003 PF.NER 004000 RMSIF\$= 000000 001030R 000010 000030 003 PF.REM PF.RW PF.1US RNDMH = 000174 RNDML = 000172 RSTFQ 000020 RSTSYS= 000004 003 UU.SWF 010000 001132R TMPMBF= 000072 UU.TB1 177775 TMPORG= 000065 UU.TB2 177764 UU.TRM UU.YLG UU.ZER 000020 177777 000015 TMPRFM= TMPW11= O.ACC = 000002 PKBHND 000020 O. ALW = 000004 PLTHND 000034 RSXSYS= 000001 O.APP = 000005 POSITN= 000146 RXDHND 000022 00002 POSITX= 177776 =\$000000 TOPIOB= 000016 VCAERR= 000022 000024 000026 000020 =\$000001 TOPSTR= 000012 VCOERR= 000055 VCSERR= 000052 PRINT PRIVAT= =\$000002 TPSSYS= 000005 TRCL = VRDF\$ = 000006 VRERIT= 020000 O.CLU 000162 O. CON 000030 PRMLN = 000102 WATIM = 003 R4SAV = 000342 000170 PROMPT O.CTG = 000002 000670R 000002 PRV TOL = 000012 R5 =\$000005 TTYHND SATTBD= 000035 000332 O.FIL O.FIX ULKES : 000160 WRGPAK= 000024 00000 UNDEXP= 000363 XABSTX= 000044 O.IND 000001 YCDCOR= 000176 O.INP = 000002 PTRF\$ = 000100 000442 O.KHD 000036 PTRHND 000010 XRBC PTSF\$ = 000070 USRLOG SCAFAC = 000064 XRBLK 000010 X2 001110R 003 \$\$MAXC= 000016 . MESAG 104060 XRBLKM 000007 001104R 003 .CCL 104062 . NAME 104044 . TIME 104030 003 003 003 001100R 001074R 001070R .CHAIN .CLEAR .CORE . PEEK . POSTN . READ 104070 104012 TTAPE 104016 104032 104002 104050 104042 .TTDDT .TTECH .TTNCH .TTRST 104024 104024 104020 104022 104026 104066 XRLEN 000000 XRLOC 000001 Х7 001064R 003 .DATE 104034 .RTS XRMOD 000014 003 104046 XRTIME 000012 \$CLSAL= EXIT . SET 104036 003 \$NOREX= ***** G 003 \$OTSV = ***** G 104064 .SLEEP 104010 .WRITE 104004 .LOGS 104014 . ABS. 177777 000

000000 001

BP20TS 000000 002

PROFIL 007440 003

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 5267 WORDS (21 PAGES) DYNAMIC MEMORY: 6082 WORDS (23 PAGES) ELAPSED TIME: 00:00:39 PROFIL, PROFIL=COMMON, PRE, PROFIL

CHE:

performance line of low-cost, single board, disc and tape controllers for vour PDP-11 data storage system. All Winchester disc and 1/2" tape controllers are based on uP architecture and automated design common to the thousands of DILOG controllers in use. So they consistently offer you best price/ performance.

You can't find a better, high-

The vital time-saving and data base protecting features you want, including diagnostic test/self-test/ error correction/etc., are all built into each intelligent controller. You'll also find RT-11, RSX-11, RSTS,

IAS, MUMPS, DSM or VMS software compatibility, plus I/O compatibility with all popular drives.

WINCHESTER/SMD/CMD CONTROLLERS

These controllers mate with larger capacity hard disc drives employing RK, RM and RP emulations. The following is a partial listing of models:

NEW-UNIVERSAL SMD **INTERFACE COMPATIBLE**—Model

DU215—Interfaces UNIBUS to SMD I/O drives—switch selectable RK06/ RK07—two 8" or 14" Winchester SMD/ CMD drives to 300 MB capacity (mix or match without any component change) 56-bit ECC-88% utilization of unformatted storage capacity—exclusive universal formatting.

NEW—SMD INTERFACE COMPATIBLE—Model DU218—Full RM02/RM05/RP06 media compatibility—interfaces PDP-11/04 through 11/60 computers with up to four Winchester, or CDC9762/9766 equivalent disc drives employing

industry standard SMD interface, as well as Memorex 677/RP06 type drives—includes three sector buffering, 32-bit ECC and dual port capability.

MAGNETIC TAPE CONTROLLERS

These 1/2" mag tape controllers/ couplers handle standard start-stop and streaming drives with TM-11 or TS-11 emulations.

1/2" STREAMING/START-STOP COMPATIBLE—Model DU130-Handles up to eight 9-track, 800/1600 BPI industry standard drives at 12.5 ips to 125 ips—interfaces dual density

(NRZI/PE) drives from all manufac turers. TM-11 emulation.

> NEW-1/2" MAGNETIC TAPE **COMPATIBLE**—Model DU132— Emulates TS-11—interfaces up to four start-stop or streamer drives, including "Cache Streamer" * *—12.5 ips to 125 ips speeds-exclusive 800/1600 BPI dual density—also interfaces GCR transports. Contact your local DILOG sales office for

complete details and O.E.M. quantity discount pricing/delivery of specific models

providing Winchester/Backup for PDP-11.

Corporate Headquarters 12800 Garden Grove Blvd. • Garden Grove, Calif. 92643 • Phone: (714) 534-8950 • Telex: 681 399 DILOG GGVE

Eastern Regional Sales Office

64-A White Street • Red Bank, New Jersey 07701 • Phone: (201) 530-0044

European Sales/Service Office

12 Temple Square • Aylesbury, Buckinghamshire • England • Phone: 44-296-34319 or 34310 • Telex: 837 038 DILOGI G



^{*}DEC, PDP, RSX, RSTS, IAS, VMS, UNIBUS Trademarks Digital Equipment Corp

^{* *}Trademark Cipher Data Products

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

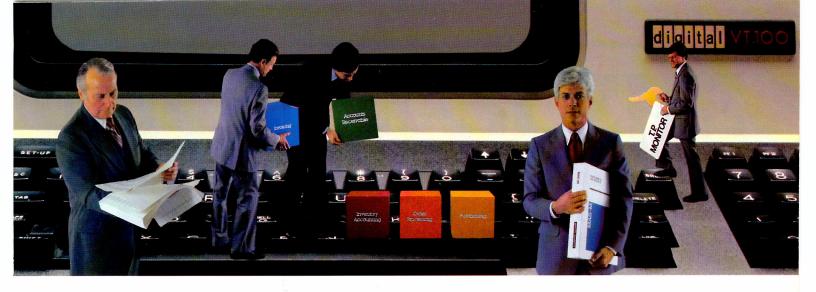
wish to read. Or, you can place a message into your HOLD basket for a number of days to have it automatically reappear in your IN basket on the appointed day. You can even have USER-MAIL recall specific messages by providing your own selection criteria. Replying, forwarding, and sending to groups are 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.



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.



You can get more from your VAX computer or 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. The MAS-M/VAX package is written in native mode VAX-11 BASIC and the MAS-M/PDP package is written in BASIC-PLUS-2 under RSTS/E. Both of these packages are based on the RMS data management system. These features make both MAS-M packages fully compatible with your current RSTS/E or VAX/VMS operating system.

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 online. 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 (800) 638-7080 In Maryland (800) 492-7170

MARTIN MARIETTA

CIRCLE 111 ON READER CARD





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 VT131 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



AUTHORIZED digital TERMINALS DISTRIBUTOR

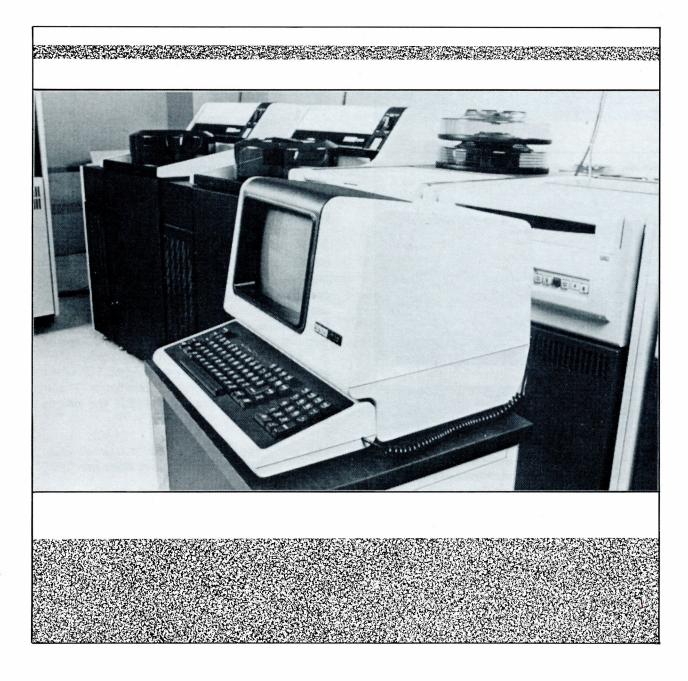
Applications Specialists & Distributors, New York, New Jersey and Ohio. DEC, Intel, Lear Siegler, Texas Instruments, Dataproducts, Diablo, 3Com, Hazeltine, Racal-Vadic, Digital Engineering, MICOM, Cipher and Elgar.

The VAX-SCENE

Number 11

(RSTS PROFESSIONAL, Vol. 4, No. 6)

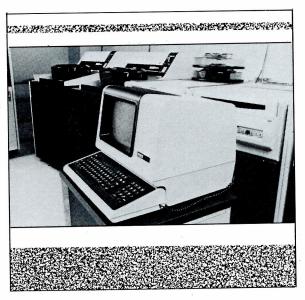
December 1982

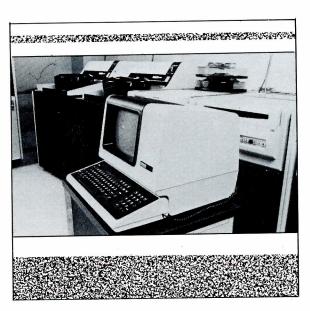


INSIDE:

Reading RMS Files from VAX MACRO







Reading RMS Files from VAX MACRO

By Bob "MACRO MAN" Meyer

Long ago and far away, I attempted to use some RMS macro on a RSTS system. (I'm not sure exactly what possessed me to do this . . .). It was a simple program, in fact the program I tried to use was one that was given as an example in the back of the RMS-11 MACRO-11 Reference Manual. After several days and much bloodshed, I surrendered; the program simply would not work. I tried a few other such programs, but to no avail; I was convinced that RMS macro was just too complicated for poor ol' macro man . . .

Fortunately, RMS macro programming isn't quite so bad on the VAX. In fact, it actually works! The program included in this article is a simple example of how to define the required control areas (rabs & fabs), open an already existing file (this program); read records from that file (the statements in this program); and print them on the current terminal.

If you'll turn to the sample program, I'll narrate . . .

First, in the psect TTAREA, we define the name of our output device, TT:. Next we block off a word for the channel number; this number is assigned to us by VMS when we open the terminal (see the \$assign_s call in the CODE psect). This channel number is used in all write operations to the terminal. Now we define a macro to do printing to the terminal (PRINT). We'll be using this one a lot.

Moving to the psect FIBSTUFF, we define the two major control blocks required to perform RMS I/O. (The macro call \$FIBDEF defines all related symbols for us.) The first control block is the FAB or File Access Block. The options specified here are:

FAC	File access method; or how we wish to access the file. Some possibilities are: DELETE, GET, PUT, etc
FNA	File name address; this is a pointer to the .ASCII

FNS	File	name	size;	this	is	the	size	of	the	string
	poin	ted to	by FI	۱A.						

file name.

MRS	Maximum record size; this is the maximum size
	of any record in the file.

ORG	File Organization; a one byte field specifying the
	type of file to be accessed. This can be: SEQ,REL,
	or IDX

RFM	Record format; some choices are: Fixed, Stream,
	Undefined, Variable, VFC.

Now we define the FIB or File ID block, and its descriptor (FIB_DESCR:) which points to the FIB block, and specifies its length.

Next we define the other major block, the RAB or Record Access Block. Some options used here are:

FAB	This is a pointer to the File Access Block. This field is used during \$CONNECT operations.
RAC	Record access field; in this case, sequential.
RBF	This field is a pointer to a record buffer.
RSZ	Size of the RBF field.
UBF	Pointer to a user buffer (this is where the records will actually reside after a \$GET operation).
USZ	Size of the UBF field.

Next we actually define the user buffer & record buffer. After these we have the full name of the file we wish to access, and the length of this filespec.

Finally we define some text & messages to make the output look presentable.

Now we start the actual code (at the label READ:). The first word in the program is a register save mask. Specific bit patterns in this word instruct the system to preserve certain registers upon entry into the routine. Since we're not interested in saving anything, we'll just set this to zero.

Next we assign (or open) the current terminal with the \$ASSIGN_S macro. The system returns a channel number to us (in TTCHAN) which we will use for all output (or input) to the terminal. The instruction BLBS says 'Branch if low bit set in R0 to 5\$'; since the low order bit is set after monitor calls to indicate success, this will cause a branch to the label 5\$ if the \$ASSIGN_S works properly. Otherwise, we just RET (return) to VMS with the error code in R0, and an error message will be printed on our terminal.

Assuming the \$ASSIGN_S works, we're ready to open the disk file. This is done with the \$OPEN macro. We simply specify the FAB in the open statement, and let 'er rip . . . If anything goes wrong, the BLBS instruction will, again, return control to the system, and an appropriate error message will be printed on the terminal.

At the label 10\$, we \$CONNECT the FAB with the RAB, making I/O to and from the file possible.

Next at 20\$, we start reading records. The \$GET macro gets the next (sequential) record from the file and places it in the user buffer specified in the RAB layout. After each \$GET we check for an error code in RO. If we have an end of file indication, we simply close the file and exit. If something else goes wrong, we blitz, since we're not really expecting any other errors.

If we indeed get a valid record, (40\$) we get the record length in R2, print the record (between quotes) on the terminal, and finally print a < CR>< LF> to make it readable. After all that we loop back to 20\$ and continue until we hit

end of file.

.end

read

```
That's about all for now; have fun . . .
.title read
;read.mar
;uses rms macro to read disk files
.psect ttarea
                    /tt/
         .ascid
ttchan: .word
                   buf, siz
chan=ttchan,
.macro print
          $qiow_s
                   func=#io$_writevblk,-
.endm
        print
.psect fibstuff
; layout the fab & rab
$fibdef
fab_block:
$fab
                    fna=file_name, -
                   fns=file_size,-
mrs=512,-
                   org=seq.-
                   rfm=fix
fib_block
          .blkb
                   fib$k length
fib_descr:
         .long
                   fib_block
rab_block:
                   fab=fab_block,-
                    rac=seq.-
                   rbf=record buffer.-
                    rsz=512,
                    ubf=user_buffer,-
                   usz=512
user_buffer
          .blkb
record_buffer:
fiel_name:
.ascii /SYS$
file_size=.-file_name
                  /SYS$DISK:[OSCAR]READ.MAR/
 .psect text
crlf: byte 1:
quote: .byte 3:
eofmsg: .ascii /
eoflen =.-eofmsg
                   13.10
                  39
/[eof]/<13><10>
 psect code
          $assign_s,-
                   devnam=tt
                   chan=ttchan
r0,5$
                                                 ; open the current terminal
          blbs
                                                 ; open worked ; open failed
         ret
:now open the file
          $open
                   fab=fab_block
         ret
; connect to the file
10$:
         $connect . -
                    rab=rab_block
                                                 ; connect worked ; connect failed
          blbs
                   r0,20$
;get next record
20$:
          $get
                   rab=rab block
          cmpl
bneq
                   r0, #rms$_eof
                                                 ;watch for end of file
                                                ;not yet
;got eof
          brw
                   90$
30$:
         blbs
                   r0,40$
                                                 :watch for other errors
                                                 ; something else is wrong; quit
         ret
                                                ;get length of record just read
40$:
         movl
                   rab_block+rab$w_rsz,r2
                   quote, #1
user_buffer, r2
                                                 ;output the record just read;output another '"';output a <CR><LF>
          print
         print
                   quote.#1
                   crlf,#2
                                                 ;loop till end file file
                                                 ;say we're at end of file ;close the file
90$:
         print
                   eofmsg. #eoflen
                   fab=fab_block
         ret
```

TIPS & ECHNIQUES

A Column For The Advanced RSTS/E User

Wef Flesichman, Software Techniques, Inc.

RSTS JOB CONTEXT

1.0 INTRODUCTION

In this column we shall explore RSTS job context: what the term means, why RSTS uses it, where it keeps it, and how your understanding of it can help you in debugging MACRO programs. To introduce you to the context concept we will first turn our attention to various types of information which describe jobs on the RSTS system (more than that which you see in a SYSTAT listing).

2.0 JOB DATA STRUCTURES

Many of you are familiar with the job data structures, the "JDB," the "JDB2," the "I/O block" and the "work block." These data structures hold important job information and they exist for the life of every RSTS job, from HELLO to BYE. This data is used relatively often by the monitor, both during the active execution of the job and also when other, seemingly unrelated, situations arise. Because these structures are so often accessed, and on so many diverse occasions, they are kept in that precious system memory area known as the small buffer pool.

In contrast, other job information is only needed when it is the job's turn to execute. In this case it is more viable to store such information directly in the job's memory image, which will swap out of memory along with the rest of the job when not needed. Included in this category are, for example, the control-C transfer address, the job's general processor registers R0 through R5, the stack pointer, and the program counter. This information is kept in the first 1000 octal bytes of the job image, and thus it is called the job's "lowcore."

The job lowcore area is critical to run-time systems as well as the RSTS monitor because it provides a convenient read/write data area to keep job-specific context in the form of status words and pointers. This design has been critical to the successful emulation of other PDP-11 operating systems, such as RT11 and RSX-11M, whose context information varies slightly from the interpretation given by RSTS.

The job data structures and the job lowcore area are both examples of storage areas which the monitor uses for job "context" storage. Context, generally stated, is everything the monitor keeps track of to be able to interrupt a running job and resume it later intact. The ability to store-and-resume-processing is what gives operating systems like RSTS the quality of "time-sharing." Academically, one would say that the current job's context is saved and the next job's context is loaded every time RSTS proceeds to the next user in its round-robin scheduling

scheme. We call this process "context-switching." Context-switching was recognized as such an important function during the design of the VAX-11 processor family, in fact, that major sections of the VAX architecture were significantly influenced.

It should be noted that there are other various places under RSTS where portions of job context are stored as well as those outlined thus far, but we want to focus on the job lowcore area for the present.

3.0 A PRACTICAL APPLICATION

Suppose you wanted to to know what some arbitrary job on your system was doing? Say it was a MACRO program you wrote which you believed might be in an infinite loop. What debugging tools would you use to find the problem?

You could rebuild the program with ODT or some other debugging aid and try to observe the aberrant behavior again. But this approach falls short in certain ways: you will very likely wait for a taskbuild, you will have to reinitialize the program which may disturb the error condition, you must hope the debugging aid won't exceed your maximum memory limitation and you must hope the debugging aid doesn't mask the problem you're trying to observe.

Wouldn't it be nice if you could just drop in unobtrusively on the unsuspecting job to sample its current program counter, stack pointer and general registers? If the job were in an infinite loop, this would probably yield substantial information and possibly allow the isolation of the problem to a defective module.

The accompanying program, JOBCTX, displays the laststored register context of any job on a RSTS system. It happens that the monitor stores this information in locations 62-103 of each job's lowcore. This information is readily accessible if the job is resident in memory, because we can simply peek at the job using the XPEEK UUO. If the job has swapped out, then we can retrieve the same information from the swapfile in the JDB-indicated swap slot. We can be sure that the target job's context is "frozen" at the time we look at it because its context had to be saved in order for the JOBCTX program to execute. However, we must realize that JOBCTX could be interrupted during its own execution and the target job could resume. This we can detect by monitoring various target job parameters before and after we look at it. Other events may occur as well. The target job may swap in or out of a swapping slot, it may move to another area of memory or it may be killed. If context is invalid for the job, then JOBCTX attempts a retry. If for some reason, 127 successive retries fail, the program reports that the "%Job is elusive." A single retry will usually be sufficient to capture accurate data, but on a heavily loaded system numerous retries are certainly fathomable.

An extension of JOBCTX could easily print out the current FIRQB and XRB for the target job as well as the general registers. The author has written a program to create dump images of jobs which can be analyzed by the DEC PMDUMP utility. One might contemplate a change to SYSTAT to embellish its job reporting statistics.

Don't get caught out of context! Good luck.

In over 50 installations, on four continents... RSTS System Managers

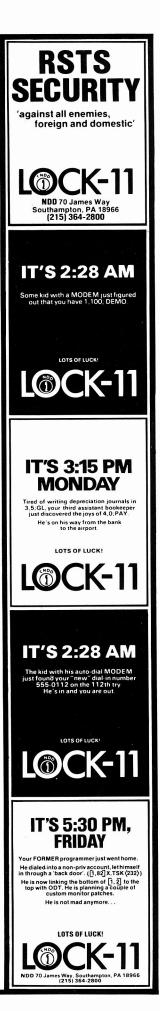
will sleep tonight.



provides them with:

- Comprehensive access control
 (35 machine years without a verified breach)
- Powerful system management tools (that don't degrade the system they manage)

Circle the number below for a full set of documentation to evaluate.



 $- \\ RSTSPROFESSIONALRSTSPROF$

```
;CLEAN UP
;USE R3 TO INDEX INTO FIRQB
;SETUP FOR READING MONITOR TABLES/PART 1
                                                                                                                                                                                                   SETFOB
;TITLE JOBCTX, <REGISTER CONTEXT DUMP>, 01, 16-AUG-82, <WEF>
                                                                                                                                                                   GO:
                                                                                                                                                                                   CALL
                                                                                                                                                                                   MOV
               THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY SOFTWARE
                                                                                                                                                                                   MOVB
.UUO
                                                                                                                                                                                                    #IIII. TR1. (R3)
                                                                                                                                                                                                                                    GET MONITOR TABLES
                TECHNIQUES.
                                                                                                                                                                                                                                   ;POINT TO CORE COMMON (EXCLUDING COUNT);END OF STRING?;NOT YET PASSED THE INVOKING CCL NAME;NO COMMAND STRING, SIMULATE "RUN" ENTRY
                                                                                                                                                                                                   #CORCMN+1,RO
                                                                                                                                                                                   MOV
               THIS SOFTWARE IS UN-RELEASED AND SOFTWARE TECHNIQUES HAS NO COMMITMENT TO SUPPORT IT AT THIS TIME, UNLESS STATED ELSEWHERE IN
                                                                                                                                                                    405:
                                                                                                                                                                                    TSTE
                                                                                                                                                                                                    (R0)
                                                                                                                                                                                   BNE
                                                                                                                                                                                                    505
                                                                                                                                                                                                    ENTRY
                WRITING.
                                                                                                                                                                                                    RESTRI
                                                                                                                                                                                   BR
 SBTTL GENERAL DESCRIPTION
                                                                                                                                                                                                                                   ;AT SPACE PREFIXING COMMAND STRING?;NOT YET, SKIP OVER THAT CCL TEXT;AT END NOW?
:NO, CHECK FOR JOB # a job number/<CR><LF>>>;REPROMT
                                                                                                                                                                    50$:
                                                                                                                                                                                   CMPB
                                                                                                                                                                                                    (R0)+, #BLANK
                JOBCTX ENABLES A PRIVILEGED USER TO DETERMINE THE CURRENT GENERAL REGISTER CONTEXT OF ANY JOB ON A RSTS/E SYSTEM. THIS INFORMATION IS DISPLAYED ON THE USER'S TERMINAL.
                                                                                                                                                                                                   40$
(RO)
80$
</?Please enter
                                                                                                                                                                                   MESSAG
                                                                                                                                                                    70$:
                                                                                                                                                                                    BR
                                                                                                                                                                                                    RESTRT
             ASSEMBLY INSTRUCTIONS
 SBTTL
                                                                                                                                                                                                                                    ;CONVERT STRING TO A DECIMAL NUMBER;ON RETURN: R0 --> NEXT BYTE; R1 = JOB NUMBER; R2 = TERMINATING CHARACTER
                                                                                                                                                                                                    SCDTB
                >ASSIGN SY: [1,2]GEN
                                                                                                                                                                    805:
                                                                                                                                                                                   CALLX
                >RUN MACRO.SAV
*JOBCTX/C=GEN:COMMON/P:1,GEN:KERNEL/P:1,JOBCTX
*^7
                                                                                                                                                                                                                                     CARRIAGE RETURN NEXT CHARACTER?
                                                                                                                                                                                                    R2.#CR
                 >TKB JOBCTX, JOBCTX=JOBCTX
                                                                                                                                                                                    CMP
                                                                                                                                                                                    BEQ
TST
BNE
                                                                                                                                                                                                    905
                                                                                                                                                                                                                                     :NULL NEXT CHARACTER?
                                                                                                                                                                                                                                     ,NO, BAD SYNTAX
                 .ENABL LC
                                                                                                                                                                                                                                    ;MAKE IT JOB # * 2
;COPY JOB # * 2
;SAVE JOB # * 2
;NONZERO JOB NUMBER?
;REPLACE JOB #0 WITH CURRENT JOB * 2
;CURRENT JOB SHOULD NOW SLEEP FOR 0 SECONDS
;TO FORCE THE MONITOR TO SAVE OUR CONTEXT
                                                                                                                                                                    905:
                                                                                                                                                                                    ASL
                MESSAG ARG
                                                                                                                                                                                                   R1,R4
R1,SVJOB0
100$
                                                                                                                                                                                    MOV
 .NLIST
                TMPORG RODATA
                                                                                                                                                                                    MOVE
                                                                                                                                                                                    BNE
MOVB
 ssssss
                                                                                                                                                                                                       (R3),R4
                  ASCIZ ARG
                                                                                                                                                                                    CLR
                                                                                                                                                                                                    XRB
                                PRINT, R5, $$$$$$
                                                                                                                                                                                    .SLEEP
                                                                                                                                                                    100$:
                MESSAG
 - ENDM
                                                                                                                                                                                    GLOBAL
                                                                                                                                                                                                    (EOF)
                                                                                                                                                                                                     JOB # * 2 OR 0 IF SELF
RANDOM
JOB # * 2
                                                                                                                                                                                    R4 =
                                                                                                                                                                                                    JOB #
 .SBTTL LOCAL DATA
                                                                  ;CHANNEL NUMBER * 2 USED TO READ SWAPFILE ;ASCII CARRIAGE RETURN ;ASCII LINE FEED
                                                                                                                                                                                     .ENABL
SWPCHN = 1 * 2
 CR = 15

LF = 12
                                                                                                                                                                                                     @#FIRQB+14,R3 ;GET POINTER TO JOBTBL
GETME3,R5,<64.*2> ;GET BIGGEST POSSIBLE JOBTBL
                                                                                                                                                                                     CALL
 BLANK
 ANGBKT = 62.
                                                                  ASCII RIGHT ANGLE BRACKET
                                                                                                                                                                                                     #BUFF+<64.*2>+2,R3 ;POINT TO END OF JOBTBL AREA + 2
-(R3) ;AT END YET?
20$ ;YES, NO NEED TO ZERO ANY MORE
(R3) ;NO, THIS IS NOT A VALID JOB SLOT
10$ ;KEEP LOOKING FOR THE -1
                                                                                                                                                                                     MOV
                                                                                                                                                                     105:
                                                                                                                                                                                     TNC
                 DEFORG RWDATA
                                                                 ;SAVED ENTRY PARAMETER
;MONITOR'S POINTER TO THE TARGET JOB'S JDB
;SAVED M.PHYA PRIOR TO START
;SAVED JZKCT COUNT PRIOR TO START
;LBN OF SWAP SLOT WITHIN SWAP FILE
;GENERAL SERVICE BUFFER
;SAVED JDSWAP IF JOB IS SWAPPED, ELSE = 0
;SAVED M.SIZE PRIOR TO DUMP
;SAVED TARGET JOB NUMBER * 2 (OR ZERO IF SELF)
;SWAPMAX IF NOT DETERMINABLE FROM .STAT
                                                                                                                                                                                     BEQ
CLR
                 .BLKW
 JDBADR: .BLKW
SVPHYA: .BLKW
SVJ2KC: .BLKW
                                                                                                                                                                                     BR
                                                                                                                                                                     205:
                                                                                                                                                                                                     BUFF(R4), JDBADR ; SAVE TARGET JOB'S JDB POINTER
                                                                                                                                                                                     MOV
                                                                                                                                                                                                     30$ ;JOB EXI
<"?No such job"<CR><LF>>
RESTRT
                                                                                                                                                                                                                                      JOB EXISTS
  SWPLBN: .BLKW
 BUFF:
SVJDSW:
SVSIZE:
                 -BLKW
                                 256.
                                                                                                                                                                                     BR
                 BLKB
 SVSIZE: .BLKE
SVJOBO: .BLKE
                                                                                                                                                                                                                                      ;ASSUME THAT JOB IS RESIDENT;LOAD JDB AND CHECK THAT JOB IS RESIDENT;JOB IS RESIDENT
                                                                                                                                                                                                     SVJDSW
                                                                                                                                                                     30$:
                                                                                                                                                                                     CLRB
                                                                                                                                                                                     CALL
                                                                                                                                                                                                     CHKJDB
  SWPMAX: .BYTE
                                                                                                                                                                                                     CTXBEF
     SWMX ==
                                                                  RETRY COUNT
  RETRYS: .BYTE
                                 0
                  .EVEN
                                                                                                                                                                      .SBTTL HANDLE OPEN SWAPFILE FOR SWAPPED-OUT JOBS
                 DEFORG RODATA
                                                                                                                                                                                      .ENABL LSB
                                                                                                                                                                                                                                      ;GET OUR MEMORY SITUATION
;ASSUME RTS'S MEMORY MAX = SYSTEM SWAPMAX
;ARE WE RUNNING UNDER RSX EMULATION?
                  .NLIST BEX
                                                                                                                                                                                      .STAT
                                                                                                                                                                                                      @#XRB+6,R3
                                                                                                                                                                                     MOV
                                                                  ; "RUN" ENTRY HEADER TEXT
                                                                                                                                                                                     TST
BEQ
MOVB
                               ;"RUN" ENTRY HEADER TEXT
/JOBCTX V1.0-01A Software Techniques, Inc./<CR><LF>
/Job Register Dump Utility for RSTS V7.1/<CR><LF>
                                                                                                                                                                                                      @#XRB+2
                                                                                                                                                                                                                                      YES SO USE IT THEN NO, PLEASE PATCH ... SWMX TO BE SYSTEM SWAPMAX?
                                                                                                                                                                                                      105
                                                                                                                                                                                                      SWPMAX,R3
                  .ASCIZ
  LBLTXT: .ASCII
                                                                                   R3 R4
SP/<CR><LF>
                                                                                                                                                                                     BPT
                                                                                                                                                                                                                                      GET JOB'S SWAP LOCATION
                                                                                                                                                                     105:
                                                                                                                                                                                     MOVB
                                                                                                                                                                                                      BUFF+JDSWAP,R1
                                                                                                                                                                                                      BUFF+JDSWAP,R
R1,SVJDSW
SETFQB
#UU.SWP,(R0)+
R1,(R0)
(R0),R5
                                                                                                                                                                                                                                      ;GET JOB'S SWAP LOCATION
;CLEAN UP FIRQB
;GET INFORMATION ABOUT A SYSTEM FILE
;GET JOB'S SWAP LOCATION (CLEARING HIGH BYTE)
;COPY JOBWAP
;COMPUTE SWAPFILE * 64
                  .EVEN
                                                                                                                                                                                     MOVB
   .SBTTL JOB LOWCORE AREAS DEFINED
                                                                                                                                                                                      MOVB
                                                                                                                                                                                      MOV
                                                                                                                                                                                                      (R0),R5
#77,(R0)
(R0),R5
(R0)
(R0)
(R0)+
                                                                                                                                                                                     BIC
SUB
ROLB
                                                                  ; JOB CONTEXT SAVE AREA USED BY RSTS MONITOR ; COMPLEMENT OF XOR CHECKSUM OF NEXT 8. WORDS
  MONCTX:
                                                                                                                                                                                                                                       COMPUTE SWAP SLOT #
                                                                                                                                                                                                                                      ;COMPUTE SWAP SLOT #
;MOVE
; INTO
; BITS <0:1>
;SET SUBFUNCTION TO COLLECT INFORMATION
;GET SWAPFILE NAME & LOCATION
                   BLKW
                                                                   :R0-R5
                                                                                                                                                                                      ROLB
                                                                   ;PC, PS, SP
                   .BLKW
                                                                                                                                                                                      ROLB
                                                                                                                                                                                      COMB
                                                                                                                                                                                                       (R0)
                  UNORG
                                                                                                                                                                                                                                       :ERRORS?
                                                                                                                                                                                                      ERRCHK
                                                                                                                                                                                                                                      ;USE SWAPFILE CHANNEL
;DECLARE OPEN FUNCTION
;OPEN SWAP FILE
                                                                                                                                                                                                      #SWPCHN, - (RO
#OPNFQ, - (RO)
                                                                                                                                                                                      MOVB
    LIST
                                                                                                                                                                                      MOVB
    .SBTTL MAIN PROGRAM CODE
                                                                                                                                                                                      CALFIP
                                                                                                                                                                                                                                      CHECK FOR ERRORS
                                                                                                                                                                                                      ERRCHK
                   DEFORG ROCODE
                                                                                                                                                                                                                                      ;MAKE SLOT # 0-76
;CONVERT TO
; BLOCKS PER KW
                                                                                                                                                                                      DEC
                    .ENABL LSB
                                                                                                                                                                                                      R5
                                                                                                                                                                                      ASL
                                                                                                                                                                                                      R5
   START::
                                   @#FIRQB+FQNENT,ENTRY ;SAVE CCL ENTRY PARAMETER (IF ANY)
30$ ; "CCL" ENTRY WAS MADE
PRINT,R5,HDRTXT ;ELSE PRINT HEADER TEXT
                                                                                                                                                                                                                                      ;COMPUTE SWAPMAX * (SLOT - 1);REMEMBER LBN'S START AT 1, NOT 0;SAVE LBN WITHIN SWAP FILE OF THIS SWAP SLOT
                    MOV
                                                                                                                                                                                      MUL
                                                                                                                                                                                                      R3,R5
                    CALL
                                                                                                                                                                                                      R5,SWPLBN
                                                                                                                                                                                      MOV
                                                                     WERE WE ENTERED VIA A CCL?
   RESTRT:
                                                                    ;YEP
;NOPE, EXIT THEN
                    BEO
                    .EXIT
                                                                                                                                                                       .SBTTL CHECK THAT JOB STATUS DIDN'T CHANGE
                                   RETRYS ;INITIALIZE RETRY COUNT
;RESET "O EFFECT

<<CR><LF>"JOBCTX" (ANGBKT)> ;PROMET THE USER
#BLANK, @ #CORCMN+1 ; PRECEED . READ STRING WITH A BLANK FOR PARSE

SETXRB ;CLEAN UP
#127., (RO) + ;DECLARE SIZE OF BUFFER IN XRLEN
(RO) + ;SKIP XRBC
#CORCMN+2, (RO) ;DECLARE BUFFER LOCATION IN XRLCO

**CORCMN+2, (RO) ;DECLARE BUFFER LOCATION IN XRLCO

**CORCMN+2, (RO) ;DECLARE BUFFER LOCATION IN XRLCO

**CORCMN+2, (RO) ;DECLARE BUFFER LOCATION IN XRLCO
    105:
                   CLRB
.TTRST
MESSAG
                                                                                                                                                                                       .ENABL LSB
                                                                                                                                                                                                                                       ;LOAD JDB, JOB RESIDENT?
;YES, DUMP SHOULD BE ACCURATE THEN
;DUMPING FROM THE SWAPFILE?
;NOPE, JOB SWAPPED OUT UNEXPECTEDLY
                                                                                                                                                                       CTXAFT: CALL
                                                                                                                                                                                                       CHKJDB
                    MOVB
                                                                                                                                                                                                       105
                                                                                                                                                                                        BEQ
                    CALL
                                                                                                                                                                                                        SVJDSW
                    MOV
                                                                                                                                                                                       BEQ
                    TST
MOV
                                                                                                                                                                                                      BUFF+JDMCTL+M.PHYA,SVPHYA;YES, AT SAVED PHYSICAL ADDRESS?
RETRY
;NO, BAD NEWS
BUFF+JDMCTL+M.SIZE,SVSIZE;YES, STILL THE SAME SIZE AS BEFORE
RETRY
;NO, BAD NEWS ANYWAY
BUFF+JDJDB2,R3;YES, GET POINTER TO JDB2
GETME3,R5,<40>;RETRIEVE JDB2
BUFF+JZKCT,SVJZKC;ANY KILO-COME TICKS SINCE LAST CHECK?
RETRY
;YES, TRY TO CATCH IT WHEN ITS IDLE
                                                                    ; DECLARE BUFFER LOCATION IN AREA
; GET USER INPUT
; 72;
;NO
;YES, JUST EXIT THEN
;IF SOME OTHER ERROR, ABORT
;COMPUTE END OF STRING LOCATION
THE STRING LOCATION
THE STRING LOCATION
;COMPUTE END OF STRING LOCATION
THE STRING LOCATION
                                                                                                                                                                       10$:
                                                                                                                                                                                       CMP
                    .READ
                                                                                                                                                                                       BNE
CMPB
BNE
                                     #EOF, @#FIRQB
                    BNE
                                    20$
                    .EXIT
                                                                                                                                                                                        MOV
                                     ERRCHK
                                                                                                                                                                                       CALL
CMP
BNE
    20$:
                                    (R0),-(R0)
@0(R0)
                                                                     AND TERMINATE WITH A NULL
                    CLRB
    30$:
```

RSTSPROFESSIONALRSTSPROFES

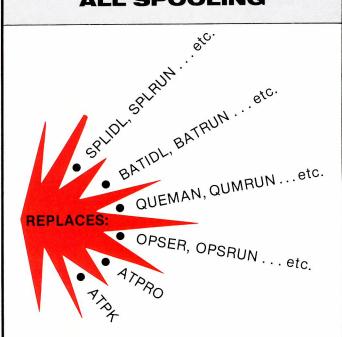
```
PRINT,R5,LBLTXT ;PRINT REGISTER LABELS
#BUFF+MONCTX+2,R4 ;POINT TO SAVED RO CONTEXT VALUE
#9.,R5 ;SET NUMBER OF REGISTERS TO PRINT
                       VOM
                        MOV
                                                                              SPECIFY LEADING ZEROES
                        MOV
                                          #-1,R2
                                                                              ;GET NEXT REGISTER
;PLACE TEXT IN BUFFER
;CREATE PRINT STRING
;SEPARATE NUMBERS BY A BLANK
      20$:
                        MOV
                                          #BUFF,R0
                        CALLX
                                          $CBOMG
                                          #BLANK, (RO)+
                        MOVE
                                          (R0)
                                                                              :TERMINATE RETURNED STRING
                                          PRINT, R5, BUFF
                                                                              ;PRINT
;BACK TILL ALL PRINTED
                        CALL
                                          R5.20$
                                                                              TIDY UP CARRIAGE
                        MESSAG
                                          <<CR><LF>>
                                                                              ;DID DUMP REQUIRE RETRIES? ;NO, WE'RE REALLY DONE TH'C
                       MOVB
                                          RETRYS, R1
                        BEO
                                          40$
 SBTTL SAVE TARGET JOB PARAMETERS BEFORE
                                   BUFF+JDMCTL+M.SIZE,SVSIZE; SAVE SIZE OF JOB IMAGE IN KW
BUFF+JDMCTL+M.PHYA,SVPHYA; GET TARGET JOB IMAGE PHYS. ADDRESS
BUFF+JDJDB2,R3; GET POINTER TO JDB2
GETME3,R5,<40>; LOAD JDB2 INTO BUFFER
BUFF+JZKCT,SVJZKC; SAVE KILO-CORE-TICKS USED BY JOB TILL NOW
SVPHYA,R3; RETRIEVE STARTING PHYSICAL ADDRESS/100
R2; CLEAR MSB
#6,R2; CREATE ADDRESS DOUBLEWORD
#1000,R1; GET ALL OF TARGET JOB'S LOWCORE
SVJDSW; MEMORY OR SWAPFILE MODE?
SVJDSW; TARGET JOB IN MEMORY
GETSWP; GET JOB IMAGE FROM THE SWAPFILE
CTXAFT; NOW SEE IF CONTEXT CHANGED
CTXBEF:
                  MOVB
                  MOV
                  MOV
CALL
MOV
MOV
                  CLR
                  ASHC
                  MOV
TSTB
                  BEQ
CALL
                  BR
                                                                        :GET JOB IMAGE FROM MAIN MEMORY
 10$:
                  CALL
                                    GETMEM
 .SBTTL CHECK THAT JOB STATUS DIDN'T CHANGE
                   .ENABL LSB
                                                                        ;LOAD JDB, JOB RESIDENT?
;YES, DUMP SHOULD BE ACCURATE THEN
;DUMPING FROM THE SWAPFILE?
                  CALL
CTXAFT:
                  BEQ
                                     SVJDSW
                   TSTB
                                                                         ; NOPE, JOB SWAPPED OUT UNEXPECTEDLY
                  BEQ
                                    BUFF+JDMCTL+M.PHYA, SVPHYA; YES, AT SAVED PHYSICAL ADDRESS?
RETRY; NO, BAD NEWS
BUFF+JDMCTL+M.SIZE, SVSIZE; YES, STILL THE SAME SIZE AS BEFORE?
RETRY; NO, BAD NEWS ANYWAY
BUFF+JDJDB2,R3; YES, GET POINTER TO JDB2
GETME3,R5,<40>; RETRIEVE JDB2
BUFF+J2KCT, SVJZKC; ANV KILO-CORE TICKS SINCE LAST CHECK?
RETRY; YES, TRY TO CATCH IT WHEN ITS IDLE
PRINT,R5,LBLTXT; PRINT REGISTER LABBLS
#BUFF+MONCTX+2,R4; POINT TO SAVED RO CONTEXT VALUE
#9.R5; SET NUMBER OF REGISTERS TO PRINT
#-1,R2; SPECIFY LEADING ZEROES
 10$:
                  BNE
                  CMPB
                  BNE
                  CMP
                  BNE
                   CALL.
                   MOV
MOV
                                                                         GET NEXT REGISTER
                                      (R4) + R1
 205:
                   MOV
                                                                         ; PLACE TEXT IN BUFFER; CREATE PRINT STRING; SEPARATE NUMBERS BY A BLANK
                   MOV
                                      #BUFF.RO
                   CALLX
MOVB
                                     SCROMG
                                      #BLANK, (RO)+
                                                                          TERMINATE RETURNED STRING
                   CLRB
                                     PRINT, R5, BUFF
                                                                        ;PRINT
;BACK TILL ALL PRINTED
;TIDY UP CARRIAGE
                   MESSAG
                                     <<CR><LF>>
                                                                         ;DID DUMP REQUIRE RETRIES?
;NO, WE'RE REALLY DONE THEN
;RETRY COUNT WILL BE POSTED TO THE USER
;SPECIFY ZERO SUPPRESSION, LEFT JUSTIFY
;CREATE COUNT AS A DIGIT STRING
;TERMINATE STRING
                   MOVB
                                     RETRYS, R1
                   BEQ
MOV
                                     40$
#BUFF,R0
                   CLR
                   CALLX
                                     $CBDMG
                   MESSAG
                                     PRINT,R5,BUFF
#^C<1>,R1
                                                                         ;PRINT THE NUMBER OF RETRIES ;MORE THAN ONE RETRY?
                   CALL
                   BIT
                   BEQ
MESSAG
                                      30$
                                                                          :NO
                                     " retries"<CR><LF>>
40$ ;EXIT

" retry"<CR><LF>>
                   MESSAG
 30$:
                                     #FIRQB+FQFUN,RO ;POINT TO FIRQB @ FQFUN
(RO)+ ;INDICATE CLOSE FUNCTION
EQ 0 ;ELSE THE ABOVE WON'T WORK
#SWPCHN,(RO) ;DECLARE CHANNEL TO CLOSE
;CLOSE THE SWAPFILE CHANNEL (IF OPEN)
                   MOV
                   CLRB
CLSFQ
                   MOVB
                   CALFIP
                                      ERRCHK
                                                                          ; ERRORS?
; THAT'S ALL FOR NOW
                                     RESTRT
                                     CLENUP ;CLEAN UP FROM AN UNUSUAL SITUATION
RETRYS ;INCREMENT RETRY COUNT (MAX RETRIES = 127)
50$ ;RETRIES EXHAUSTED, GIVE UP
GO ;RETRIES REMAINING, RETRY THE DUMP
<"%Job is elusive"<CR><LF>>
  RETRY:
                    INCB
                   BMI
  50$:
                    MESSAG
                                      RESTRT
                                                                          ; RESTART
   SBTTL CHKIDS - LOAD JDB AND CHECK FOR RESIDENCY
   ; +
: CHKJDB - READ JDB INTO BUFFER AND SET "Z" BIT IF JOB IS RESIDENT
                                     CHKJDB
                    JDB LOADED IN "BUFF"
Z = 1 IF JOB IS RESIDENT
ALL OTHER REGISTERS PRESERVED
                                                                          ; SAVE REGISTER (S)
  CHKJDB: MOV
                                      R2,-(SP)
R3,-(SP)
                    MOV
                                      JDBADR,R3
GETME3,R5,<40>
(SP)+,R3
(SP)+,R2
                                                                          GET POINTER TO JDB RETRIEVE THE JDB RESTORE REGISTER(S)
                    MOV
```

VERSION 2.2 NOW AVAILABLE

QUE.11 — V2.2

ONE JOB SPOOLER FOR RSTS/E CONTROLS ALL SPOOLING



QUE.11:

- DEC QUE Compatible
- Block letters on spooled header page
- One job controls all spooling
- Saves small buffers and job slots
- Spawns jobs as needed
- Handles line printer and keyboard spooling
- Controls as many BATCH JOBS as pseudo-keyboards
- Full parameter replacement in QUE
- calls "DO" command replaces indirect processors
- QUEMAN SYS call supported
- Program deliveries NOW
- Only \$1500 single CPU license
- Trial Version \$100

For more information contact:

On Track Systems, Inc. P.O. Box 245

Ambler, PA 19002-0245 Phone: 215/542-7133 In Europe:

Procyon Informatics, Ltd.
19 St Kevins Road
Dublin 8, Ireland

```
BUFF+JDSWAP ;JOB IN A SWAPPED STATE RIGHT NOW?
10$ ;JOB IS SWAPPED
#INIOUT!SWP!REQ,BUFF+JDMCTL+M.CTRL ;SWAPPING REQUESTED?
                                                                                                                                                 CALL
                                                                                                                                                             ERRCHK
                                                                                                                                                                                       CHECK FOR ERRORS
                                                                                                                                                            (SP)+,R1
(SP)+,R0
R5
                                                                                                                                                 MOV
                                                                                                                                                                                       :RESTORE REGISTER(S)
                                                                                                                                                 MOV
             RETURN
 10$:
                                                    :BACK TO CALLER
                                                                                                                                                 RETURN
                                                                                                                                                                                        BACK TO CALLER
 .SBTTL ERRCHK - CHECK FIROB FOR AN ERROR AND ABORT IF ERROR
   ERRCHK - CHECK FOR ERRORS FOLLOWING A DIRECTIVE
                                                                                                                                     SBTTL CLENUP - CLEAN UP AFTER A FATAL ERROR
             CALL ERRCHK
                                                                                                                                       CLENUP - RESET ALL I/O CHANNELS
             RETURN ONLY IF NO ERROR ON ERROR: PRINT DESCRIPTIVE TEXT AND ABORT
                                                                                                                                                 CALL CLENUP
                                                                                                                                                              RANDOM
             ON SUCCESSFUL RETURN ALL REGISTERS PRESERVED
                                                                                                                                                 ALL OTHER REGISTERS PRESERVED
             .ENABL LSB
                                                                                                                                                              #FIRQB+FQFUN,R0 ;POINT TO FIRQB @ FQFUN
#RSTFQ,(R0)+ ;SETUP TO RESET
(R0) ;ALL CHANNELS
                                                                                                                                    CLENUP: MOV
                                                    ;CHECK FOR ERROR AND PRINT TEXT (IF ERROR);CC MEANS WE'RE NO ERROR;INCLUDE PC IN MESSAGE
ERRCHK: CALL
                          ERRPRT
                                                                                                                                                 MOVB
                                                                                                                                                 CLR
CALFIP
                          10$ <", PC = ">
                                                                                                                                                                                        RESET ALL CHANNELS, NO ERRORS ARE POSSIBLE
             MESSAG
                                                    ;INCLUDE PC IN MESSAGE
;POINT TO GENERAL BUFFER
;THE ERROR PC, DISCARD RETURN ADDRESS
;SPECIFY LEADING ZERO SUPPRESSION
;AS A DIGIT STRING
;TERMINATE STRING
             MOV
                          #BUFF.RO
                           (SP)+,R1
                          R2
$CBOMG
             CALLX
                                                                                                                                     SBTTL XPEEK AT A 16-BIT ADDRESSED MEMORY AREA
             CLRB
                          (RO)
PRINT, R5, BUFF
             CALL
                                                    OUTPUT
                                                                                                                                      GETME3 - COPY A SYSTEM MEMORY AREA INTO LOCAL MEMORY
                                                    CLEAN UP PRIORITIES AND CHANNELS
DONE FOR THIS RUN
             CALL
                          CLENUP
                                                                                                                                                            STARTING 16 BIT PHYSICAL ADDRESS <15:0> = B<15:0>
                                                                                                                                                CALL
                                                                                                                                                            GETME3, R5, LEN
 ; ERRPRT - PRINT ERROR MESSAGE (IF ANY) AND RETURN
                                                                                                                                                DATA PLACED IN "BUFF"
R2 MSB OF STARTING ADDRESS
R3 LSB OF STARTING ADDRESS
             (FIRQB) = RSTS ERROR CODE
             CALL
                                                                                                                                                 ALL OTHER REGISTERS PRESERVED
                          IF NO ERROR
IF ERROR
                                                                                                                                                 .ENABL LSB
                                                                                                                                                             R1.-(SP)
                                                                                                                                                                                       :SAVE REGISTER(S)
                                                                                                                                   GETME3: MOV
ERRPRT: TSTB
                         @#FIROB
                                                    : ERROR?
                                                                                                                                                                                       ;CLEAR MSB
;SET BYTECOUNT
;LOAD REQUESTED DATA
;RESTORE REGISTER(S)
                                                                                                                                                 CLR
                                                                                                                                                             (R5)+,R1
GETMEM
                                                                                                                                                MOV
CALL
             RETURN
                                                                                                                                                              (SP)+,R1
                                                                                                                                                 RETURN R5
             MOVB
             CALFIP
                                                                                                                                    .SBTTL XPEEK AT A 22-BIT ADDRESSED MEMORY AREA
                                                                                                                                       GETMEM - COPY A SYSTEM MEMORY AREA INTO LOCAL MEMORY
                                                                                                                                                            LENGTH OF TRANSFER IN BYTES
STARTING 22 BIT PHYSICAL ADDRESS <19:16> = B<03:0>
STARTING 22 BIT PHYSICAL ADDRESS <15:0> = B<15:0>
                                                                                                                                                R1 =
R2 =
                                                                                                                                                 R3 =
 SBTTL GETSWP - RETRIEVE A PORTION OF TARGET JOB IMAGE FROM THE SWAPFILE
                                                                                                                                                 CALL
   GETSWP - GET SOME JOB IMAGE FROM THE SWAPFILE
                                                                                                                                                 DATA PLACED IN "BUFF"
            R1 =
R2 =
R3 =
                         LENGTH OF TRANSFER IN BYTES
STARTING 22 BIT PHYSICAL ADDRESS <19:16> = B<03:0>
STARTING 22 BIT PHYSICAL ADDRESS <15:0> = B<15:0>
                                                                                                                                                R2 MSB OF STARTING ADDRESS
R3 LSB OF STARTING ADDRESS
ALL REGISTERS PRESERVED
             NOTE: B<7:0> MUST BE ZERO
SWPLBN= LBN OF SWAP FILE FOR BEGINNING OF THIS JOB'S SLOT
                                                                                                                                                 .ENABL LSB
                                                                                                                                                MOV
MOV
CALL
                                                                                                                                                             RO,-(SP)
R1,-(SP)
SETXRB
                                                                                                                                                                                       ; SAVE REGISTER (S)
                                                                                                                                    GETMEM:
                                                                                                                                                             SETXRB ;GET XRB READY
#BUFF,XRLOC(R0) ;SET BUFFER TO RECEIVE
             DATA PLACED IN "BUFF"
             ALL REGISTERS PRESERVED
                                                                                                                                                 MOV
                                                                                                                                    105:
                                                                                                                                                                                      POINT TO XRB @ XRBLK
;SET LSB ADDRESS TO PEEK AT IN XRBLK
;SET MSB ADDRESS IN XRBLKM
;POINT TO XRB @ XRELN
;SET LENGTH TO PEEK
;NO TRANSFER, EXIT
;POINT TO NEW BUFFER POSITION USING LAST XRBC
;XRBC MUST BE ZERO (GRACIAS SPD)
;READ IN A SMALL BUFFER
;CHECK FOR ERRORS
                                                                                                                                                 MOV
                                                                                                                                                              #XRB+XRBLK,R0
                                                                                                                                                              R3, (R0)
R2,-(R0)
#XRB,R0
GETSWP: MOV
                                                    ; SAVE REGISTER (S)
                          RO, - (SP)
             MOV
                         R2,-(SP)
R3,-(SP)
                                                                                                                                                 MOV
                                                                                                                                                             R1,(R0)+
             MOV
                                                                                                                                                 MOV
                                                    REQUEST ON A SECTOR BOUNDARY?
                                                                                                                                                 BEO
                                                   REQUEST ON A SECTOR BOUNDARY?
;GLOD
;NO, REWRITE YOUR CODE FOOL
;COMPUTE BLOCK NUMBER OF DESIRED DATA
;OFFSET FROM BASE OF SWAPFILE SLOT
;GET XRB CLEANED UP
;SET TRANSFER SIZE IN BYTES
;ADVANCE TO XRLOC
;DECLARE BUFFER LOCATION IN XRLOC
;DECLARE SWAP FILE CHANNEL IN XRCI
;DECLARE SWAP FILE CHANNEL IN XRCI
;DECLARE LBN OF IMAGE IN XRBLK
;READ PART OF THE JOB IMAGE
;ERRORS?
;RESTORE REGISTER(S)
;
                          R3
10$
                                                                                                                                                              (R0)+,(R0)
-(R0)
                                                                                                                                                 ADD
             BEQ
BPT
                                                                                                                                                CLR
.XPEEK
CALL
105:
             ASHC
                          #-8..R2
             ADD
                          SWPLBN,R3
                         SETXRB
R1,(R0)+
(R0)+
#BUFF,(R0)+
             CALL
                                                                                                                                    205:
                                                                                                                                                 MOV
                                                                                                                                                 MOV
RETURN
                                                                                                                                                                                        ;RESTORE RO
;BACK TO LE CALLER
             MOV
                         #SWPCHN, (R0)+
R3, (R0)+
             MOV
             MOV
.READ
CALL
                                                                                                                                                SETFQB - CLEAR FIRQB
SETXRB - CLEAR XRB
                                                                                                                                    .SBTTL
                          (SP)+,R3
(SP)+,R2
             MOV
             MOV
                                                                                                                                       SETFQB - CLEAR FIRQB
SETXRB - CLEAR XRB
             MOV
                          (SP)+,R0
             RETURN
                                                                                                                                                 CALL
                                                                                                                                                             SETXRB
  .SBTTL PRINT - PRINT A STRING ON THE CONSOLE KB:
                                                                                                                                                             TO FIRQB @ FQFUN (SETFQB)
XRB (SETXRB)
    PRINT - PRINT A TEXT STRING
              R5 --> .ASCIZ FORMAT STRING TO PRINT
                                                                                                                                                 .ENABL
                                                                                                                                                            LSB
              CALL PRINT, R5, <"TEXT">
                                                                                                                                                             #FIRQB+FQFUN,-(SP) ;RETURN RO @ FQFUN R1,-(SP) ;SAVE REGISTER(S) #FIRQB,RO ;START HERE #FQBSIZ/2,R1 ;FIRQB SIZE IN WORDS 10$ ;JOIN COMMON CODE
                                                                                                                                    SETFOB:
                                                                                                                                                 MOV
              NO REGISTERS CHANGED
                                                                                                                                                 MOV
                                                                                                                                                 MOV
                                                                                                                                                 MOV
              . ENABL LSB
 PRINT:
                          R0,-(SP)
R1,-(SP)
                                                     ; SAVE REGISTER (S)
                                                                                                                                    SETXRB: MOV
                                                                                                                                                              #XRB,RO
              MOV
                                                                                                                                                 MOV
                                                                                                                                                              R0,-(SP)
R1,-(SP)
                                                                                                                                                                                       ; SAVE REGISTER (S)
                                                     GET POINTER TO TEXT STRING
              MOV
                           (R5),R0
(R0)+
                                                                                                                                                 MOV
 10$:
              TSTB
                                                    ;AT END YET?
;NO, KEEP LOOKING
;COMPUTE LENGTH OF STRING
;POINT TO XRB
;SAVE STRING LENGTH IN XRLEN
;COPY INTO XRBC
;COPY POINTER TO STRING INTO XRLOC
;INSURE CHANNEL ZERO
;PUT TEXT OUT
                                                     :AT END YET?
                                                                                                                                                 MOV
                                                                                                                                                              #XRBSIZ/2,R1
                                                                                                                                                                                       ; XRB SIZE IN WORDS
              BNE
                           10$
                                                                                                                                    105:
                                                                                                                                                              (R0)+
                                                                                                                                                                                       ;CLEAR A WORD MORE
                                                                                                                                                              R1.105
              MOV
                           #XRB,R1
                                                                                                                                                 SOB
                                                                                                                                                                                       ; UNTIL FINISHED ; RESTORE REGISTER(S)
                          RO, (R1)
(R1)+, (R1)+
(R5)+, (R1)+
              MOV
                                                                                                                                                 MOV
                                                                                                                                                              (SP)+,R1
(SP)+,R0
              MOV
                                                                                                                                                 MOV
                                                                                                                                                 RETURN
                                                                                                                                                                                       ;BACK TO CALLER
                                                                                                                                                 END
                                                                                                                                                             START
```

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.

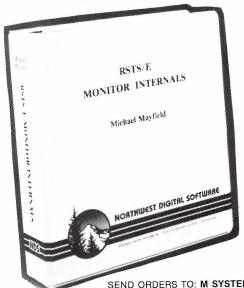
Information is available for all levels of users:

- 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

Yes Virginia, there is a Macro Man!

Call MACRO MAN

for superior performance

- RSTS Internals •
- Custom Macro Programming
 - RSTS or 11M •

LEARN MACRO FROM THE MASTER! Seminars available — call for details.

MACRO UTILITY LIBRARY Available now — call for information.

Also Available Now — VAX MACRO Consulting

NEXT RSTS MACRO COURSE MID JANUARY 1983

Bob 'Macro Man' Meyer
9 Lockwood Avenue, Fieldsboro, NJ 08505
609—298-9127

CIRCLE 84 ON READER CARD

Going from RSTS/E to VAX/VMS?

MIGRAID3

can help get you there.

MIGRAID3_{TM} USERS WITH A RSTS/E DISK PACK MOUNTED AS A "FOREIGN" VOLUME UNDER VMS CAN

- TRANSFER RMS-11 FILES DIRECTLY TO VMS, READY TO USE
- COPY ASCII STREAM FILES DIRECTLY TO RMS-32 SEQUENTIAL/VARIABLE FORMAT
- TRANSFER VIRTUAL ARRAYS AND BLOCK I/O FILES, READY TO OPEN ORGANIZATION VIRTUAL (SUPPORTS ROSS/V FILE STRUCTURES)

MIGRAID3 SUPPORTS RSTS/E WILDCARDS FOR BULK COPIES, AND OFFERS A COMPREHENSIVE DIRECTORY FACILITY TOO!

\$800 (U.S.) Single-CPU license; QDA/OEM terms available.



COMPUTER METHODS CORPORATION P.O. BOX 592 MOORESTOWN, NJ 08057 (609) 778-8440

RSTS and VAX/VMS are registered trademarks of Digital Equipment Corporation. ROSS/V is a trademark of Evans, Griffiths and Hart, Inc.

CIRCLE 152 ON READER CARD

HOW TECO

By Andrew Harrison, Schroder Life Assurance Ltd., Enterprise House, Portsmouth, England

I read the article on getting directories in TECO (RSTS Professional, June, 1981) & noticed mention of the problems with wild card edits. The attached TECO macro is designed to allow correct wild card editing. It can either be installed as an initialiser or run from TECO. I trust that modification for other than PDP11/70 systems will prove trivial.

If it is used as an initialiser then it will only work on the simplest format & will return control to TECO in all other cases. If it is run from TECO it expects the buffer to contain the filespec. The default extension is .BAS (but others could be used). (Anyone who puts the account after the file name "FRED[1,2]" deserves the error message he gets; however, if required, the routine to insert ".BAS" can be altered to:

OJ:S.\$ "E HXP ENTEQP\$ HK G* OJ S.\$ IBAS\$' this used the ENfilespec\$ to put it in standard format.) If no device is specified then after trying the default disc an alternative is tried so the routine !ADD C:! MUST be altered to a logical that is meaningful for your system (or removed). (Masochists can extend this routine to scan several discs.) For efficiency the wild card routine is skipped if the filespec does not contain wild cards (it saves at least one directory look up).

After running the routine there are 3 commands provided:

MF\$\$ finishes the edit and reedits leaving you at the front of the file

MN\$\$ finishes this edit and edits the next file in the wild card specification

 -IMN\$\$ kills the current edit and then edits the next file in the wild card specification

The method used is to store all the matching file specs in Q register O separated by < cr>< lf>. This freezes the list of files and prevents the wild card lookup getting confused. File names are then stripped off the front of this list & used (:GO\$\$ can be used to inspect the list of files not yet edited.)

All the "\$" signs are ESCapes, the "1" symbol can usually be used directly but some of the "1" must be genuine.

! TECO initialiser routine!

! buffer contains the CCL this routine only handles the simple format!
! TE FILESPEC or TECO FILESPEC!
! alternatively as a program load the buffer with the FILESPEC!
! To test use EI, but for running use the SQUished version!
! Q register usage!
! N edit next wild card file!
! O work file for above!
! F find front of file!
! P file name for above!
! Find front of sove!
! First step is to jump to end if CCL contains "/" or "="!
OJ :S/\$ "N OEND\$' :S=\$ "N OEND\$'
! delete "TECO" from buffer to use buffer for formatting instructions!
! do this by finding space & deleting up to it!
OJ :S \$ 0,.K
! now add default extension of .BAS if none given!
OJ :S.\$ "E ZJ I.BAS\$'

```
!copy file spec to register P!
I now check for either * or ? which indicates a wild card file spec!
OJ :S?$"N OWILDS$' :S*$"N OWILDS$'
! here if no wild card found so do simple edit!
:@EB% EQP% "E OADD C:$' ! do EB of register P, if failure then try
adding C:!
HK G* HXP | file opened name in P! ^AEditing file "^A HT ^A"
HK G* HXP
 A HK Y
OUTILITIES$
IWILD!
@EN% EOP%
                ! set file spec in wild card buffer!
                 !look up file spec!
< :EN$ ;
HK G# I
$ :HXO > !append filepsec <cr><lf> to 0 register !
HK GP OJ :S:$ "E OADD C:$"
! next load N with a find & edit next wild filename command!
+0 "L EK HK Aedit killed
A | EC :QP "N Aedit finished
A'' | either about
               ! either abort or finish edit!
! print "edit finished" only if there is something!
               ! (i.e. file name) in P!
GO OJ :S
$ "N !put unused file specs in buffer & find end of next file spec!
.,ZXO !save unused file specs!
 ,ZXO
.,.AU isave unused file specs!

O,.-2XP! put current file spec into P!

^Aediting file "^A O,.-2T ^A"

^A @EB% EQP% Ido EB on contents of register P!

EK Y | ^A*** No more files ***
                       ! this gives file not found message if 0 is empty!
:QO "E OADD C:$1
HK XP MN ! use the routine!
OUTILITIES$
!ADD C:!! here if file not found so if no disc specified try C:!
OJ :S:$ "E IC:$ OOPEN$'
 AFile not found - sorry
!UTILITIES!! load utilities into Q registers!.
! next add a find front of file routine if F!
@'UF\EC^Afinish edit
^A :QP "E G* HXP HK' ! load P with G* if P was empty!
@EB%^EQP% ! do EB of register P!
^Are edit "^A :GP ^A"
^Ainitialiser has run
 C$ ! this unconditionally ends macro, if this is an initialiser!
      ! then it jumps straight into TECO rather than via TECO.TEC!
IEND!
```

Naturally people can add routines to load their own useful utilities in the utilities section.

Some of the routines I would like to develop are to do with the TECO screen handling routines. If someone who understands the ":W" commands could write an intelligible article on their use I would be both amazed and grateful. In particular is it possible to get a "SEEALL" mode display without loading "VTEDIT"?

A more recent copy of RSTS Professional gave some interesting details of EDT initialisers, etc. This has led me to some interesting experiments on both RSTS and VAX. Here are some random jottings on the minor differences between the two implementations of the EDT screen editor. Special characters such as <ESC> and <FF> are handled by displaying in the above formats, the VAX EDT (in my opinion correctly) leaves them embedded in the text:-

"line with <ESC> inserted in it"
but RSTS EDT treats them as line terminators:
"line with <ESC>
causing a new line."

fault tolerant PDP-11 systems

(n.) Something that prevents perfection, an error.

(adj.) Able to withstand or endure an adverse environmental condition.

(TM) Popular minicomputer family from Digital Equipment Corporation.

systems (n., pl.) Groups of interacting, interdependent elements forming collective entities.

(**Product**) 1. Redundant, multiprocessor configurations of DEC standard hardware enhanced by advanced components. 2. Compatible habitat for DEC software products. Ibid. VAX.



(Supplier) 1. The source of such systems, components, and software. 2. A systems house, designing such products to order.

WESTERN RESOURCE TECHNOLOGY, INC. 2970-Q East La Palma, Anaheim, CA 92806 Telephone (714) 630-7852 TWX (910) 591-1241

reliable by definition

CIRCLE 167 ON READER CARD

Minitab

Minitab

Minitab

Minitab

Minitab

Minitab

Minitab

Minitab

Minitab

Minitab

Minitab™

The Only Interactive Data Analysis Package for All DEC Systems

Elementary Operation, Advanced Capability

Used by:

- One third of Fortune's Top 50
- Almost every major university in the U.S.
- Hundreds of smaller organizations in 25 countries around the world.

Its simplicity of design has been steadily perfected since 1972. The results are:

- Unusually rapid installation and operation
- A sophisticated range of data analysis capabilities and statistical procedures.

You can do more when you do it simply.

For more information write or phone: **Minitab Project**, 215 Pond Laboratory, University Park, PA 16802 U.S.A. Telephone (814) 865-1595 Telex 84-2510

CIRCLE 168 ON READER CARD

DEC BEST VALUES

PRE-OWNED DEC EQUIPMENT

BUYING AND SELLING

SYSTEMS • CPU's • PERIPHERALS • TERMINALS OPTIONS • MEMORY • COMPATIBLES

CALL DICK BAKER (305) 979-2844



Incorporated Ft. Lauderdale, Florida 33309 Telephone (305) 979-2844

CIRCLE 49 ON READER CARD

It appears that EDT knows there is no < CR>< LF>there because the output file does not seem to contain the extra characters, but EDT is quite happy to let you delete this fictional character changing the display but not the output file. A very similar problem occurs with the SET ENTITY command. On the VAX careful use of the SPECINS key enables the setting up of word delimiters in the initialiser file, which includes, for example, < CR > as well as the additional characters required. On RSTS these are regarded as terminating the command line prematurely causing horrid error messages.

An undocumented feature of EDT is that pressing the escape key twice inserts a single escape character into the buffer.

Finally some enhancements that I would like to see. The advance word key skips multiple spaces but does not skip multiples of any other delimiter; I would like it to do so. It is frustrating and perplexing the way EDT slows down the repeat rate on the keypad keys, how does it do this? and why? Screen width, EDT ignores both the actual terminal setting and also the width set on the system; surely it could pick up one or the other. There is no facility to return to the point marked by using the SELECT key ".".

REPRINTS **REPRINTS** REPRINTS!

All content in this publication is copyrighted.

All reprints must be purchased from M Systems, Inc. No other reprints are authorized.

All reprints shall contain both a cover and a subscription blank.

Price quotation available on request.

SIMINI

A SIMULATION MINI COMPUTER

By John Cato

In the October issue of RSTS Professional, you saw how to use SIMINI. I regret that my introduction to the examples in that article became misplaced, however I am sure that you didn't take long to fathom it out.

As intended, I am publishing the source listings in this issue, and offer some clues as to ways you could modify SIMINI if required, for different terminal types.

The sources are mostly self-explanatory and you will see from investigation that it is relatively easy to include new opcodes or operating system commands. You will find it similarly easy to modify for use on different VDUs. You may find it interesting to consider a 'virtual' memory on backing store the addressing facilities would allow for this.

If you would like a complete copy of the SIMINI user quide, please send a self addressed sticky label; and £3.75 or \$7.50 US, to cover photocopying and postage to me: John Cato, R.T.Z. Computer Services Ltd.,

P.O. Box 19, 1 Redcliff Street, Bristol, England. BS99 7JS. Any of your thoughts or comments on SIMINI will also be welcomed ... let me know about the bugs too!

I repeat, much of the work on SIMINI was done some years ago, and grew from seeds planted by Mr. M.C. Williams, MSc, MBCS., Computer Centre Director of Luton College of Higher Education, England. This has been published in order that this may now be of benefit to more people, and I have asked the Editor to send any payments for publishing to Mr. Williams to use as appropriate for computer education in the Luton area.

It should be noted that, to all intents and purposes, the SIMINI software and documentation is available free.



NOTE: The following article is based on a paper presented at the Spring-82 DECUS and included in the DECUS Proceedings.

A USER WRITTEN KEYBOARD MONITOR

By Ken Harris, Unico, Inc., 3725 Nicholson Road, Franksville, WI. 53126, (414) 886-5678

INTRODUCTION

This article describes the implementation of a keyboard monitor designed to provide RSTS/E users with a more powerful command environment. The backbone of the keyboard monitor is a macro processing facility which allows users to define their own commands and symbols. The intent of this article is to describe the interesting details of this project and to provide some ideas for those interested in similar efforts.

WHAT IS A KEYBOARD MONITOR

On a RSTS/E system, users communicate with the operating system via command interpreters called keyboard monitors. A keyboard monitor prompts the user for input, waits for a command, and then translates the command into some action. That action may be running a program, adding a private logical, clearing a crt screen or whatever. It is the keyboard monitor that defines the command environment that the user sees. DIGITAL supplied keyboard monitors include DCL, BASIC-PLUS, RSX and RT11.

WHY BUILD A KEYBOARD MONITOR

Different users have different needs and desires. The needs of a programmer are certainly different from those of an accounting clerk. By writing a keyboard monitor, the user interface may be tailored to meet specific needs. The command set can be expanded, restricted or made more friendly. Support can be provided for specific terminal devices.

HOW TO BUILD A KEYBOARD MONITOR

A keyboard monitor may be implemented in a number of ways. We will restrict our attention to two approaches which for our purposes will be referred to as the High-road and Low-road approaches.

The High road approach involves the implementation of a keyboard monitor as a run time system. This is not an easy task. It requires a knowledge of runtime systems and Macro-11 programming. The advantage of this approach is an efficient product which should be fast, compact and shared. The disadvantage is the difficulty involved with

BACmac can do it all!

BAC into RTS / BAC into MAC / BAC into BAS

Two Word Version
Now Available

BACmac is a unique software tool, running under RSTS/E, which provides the following conversions:

- translation from Basic-Plus "compiled" back to Basic-Plus source code (only the comments will be missing)
- translation from Basic-Plus into Macro source code, which compiled under RSTS runs faster than Basic-Plus
- translation from Basic-Plus into Macro source code which may be compiled under RSTS for execution under RT11 a migration facility
- translation from Basic-Plus into a RUN-TIME-SYSTEM. Now you can write an RTS in Basic-Plus. The ideal solution to memory thrashing due to "multi-copy" applications programs.

RSTS/E, RT11, Macro-11 and Basic-Plus are trademarks of Digital Equipment Corporation.

Western Distributor: Telecom Computer Systems, Inc. P.O. Box 03285 Portland, Oregon 97203 503/286-5122

ADOS | Advanced Digital Office Systems

Eastern Distributor: New England Micro Technology, Inc. P.O. Box 767 Marblehead, Mass. 01945 617/631-6005 RSTSPROFESSIONALRSTSPROFESSI

writing, debugging and maintaining a runtime system.

The Low-road approach involves the use of feature patch 21.3.5 to create a modified RSX runtime system that chains to the keyboard monitor program. The keyboard monitor may then be written in any available language. The advantage of this approach is the ease of implementation. The disadvantage is that the keyboard monitor will not be as efficient as a runtime system.

INTRODUCTION TO PDM

The remainder of this article is devoted to a discussion of a keyboard monitor called PDM. PDM (or Program Development Monitor) is a Low-road monitor written entirely in "C". It was designed and implemented here at UNICO and has been in use for about a year.

The heart of PDM is a macro processing facility which allows users to define their own commands and symbols.

OBJECTIVES & DESIGN GOALS

- (1) To provide a command environment which can be easily tailored to the needs of a specific task.
- (2) To provide a command environment which promotes programmer productivity.
- (3) To provide a command environment which enhances rather than

(1) Initialization — Upon entry, PDM performs a number of housekeeping chores. The most important is the opening and loading of the local context file.

(2) Get Command Line — PDM prompts the user for input with a colon ":", inputs a line of text and converts it to upper case.

(3) First Level Commands — PDM next checks for a first

level command. These are comwhich mands change or display PDM's context information and must be executed before macro expansion occurs. If the input command is a first level command, it is executed and control returns to get the next command.

(4) Macro Expansion — Next the command is passed through the macro processor. If the command expands into more than one line, it is forced to the job's console and control returns to get next command.

(5) Second Level Commands — PDM next checks for a second level command. These are commands which are executed after macro expansion (RUN, PRINT, FORCE, @). If the expanded command is a level two command, it is executed and control returns to get the next command.

(6) Try CCL Command — Since

FINAR-

The only electronic spreadsheet that's powerful, versatile and easy to use.

With FINAR it's easy to prepare complex budgets, forecasts, financial statements, cash-flow projections and more. But that's only the beginning. FINAR can perform sophisticated tasks that far surpass any of the "calcs" capabilities. In fact, in every situation where FINAR has been compared to the "calcs", FINAR has been chosen.

FINAR IS POWERFUL.

FINAR is a real business system, versatile enough to meet the needs of the largest corporations and smaller firms too. FINAR's capabilities include extensive mathematical functions; graphic output; interfaces with other systems; flexible report format; worksheet definition and control directives.

FINAR IS EASY TO LEARN.

In just two days one can learn all that is needed to perform the most complex financial analyses with FINAR. Commands are in English-not "computerese"-and simple prompts let one know what information the system requires to compute the answer to a auestion.

FINAR IS EASY TO USE.

All you need is a DEC PDP-11 with RSTS or a VAX-11 and FINAR to plan calmly for whatever the future has in store. FINAR is available for as little as \$4500. For a demonstration of FINAR's unlimited technical capabilities and ease of operation, call: Walter Fleming (713) 960-0848.



JAMES B. HOTZE & CO.

3303 South Rice Avenue, Suite 208 Houston, Texas 77056 (713) 960-0848

CIRCLE 51 ON READER CARD

replaces the existing CCL and DCL environments.

- (4) To start with a small keyboard monitor and add enhancements as needs and ideas arise.
 - (5) To provide a general tool.

GENERAL DESCRIPTION

The following is a general description of what PDM does with a command.

the command is not a PDM command, PDM tries to execute it as a CCL command. If successful, control is passed to the called program.

(7) Try DCL Command — As a last resort, PDM tries to execute the command as a DCL command. Control is passed to DCL. If the command is a DCL command it is executed, otherwise DCL prints an error message and exits to the job's default keyboard monitor, PDM.

CONTEXT FILE

PDM maintains files of context information. Each file consists of a header followed by a linked list of definitions.

The global context file, [1,2] PDM.CTX, contains global definitions and is only accessed by users in [1,2].

Each job has its own local context file named PDMOnn.TMP (nn = job). At initialization time, PDM loads the local context file. If the file is not found, it is created by copying the global context into the user's account. The file is written back to disk whenever there is a change in the context information. The local context file is automatically deleted by LOGOUT.

MACRO PROCESSOR

The heart of PDM is its macro processing facility. Each line of input is considered to be a command line. The first token of the line is the command and the rest of the tokens are symbols. At macro expansion time, an attempt is made to find each token in the definition list. If it is found, the token is replaced by its expanded form. The expanded form is then pushed back onto the input stream. This allows for the processing of nested commands and symbols.

The macro processor also performs a number of translation chores. Uparrow-character sequences are translated to control characters. String concatenation is performed. Continuation lines are merged together. Strings within single or double quotes are not translated. Circular expansions are detected and terminated with an error.

USE OF FORCE

If a command expands into more than one line, it is forced back to the job's console terminal. This allows for commands which enter other keyboard monitors or which run programs and supply data. There are a number of problems with using force.

The major problem is a conflict with type-ahead. If PDM forces a command while the user is busily typing the next command, the results are unpredictable. We have not found, a solution to this problem beyond instructing the user to be careful with type-ahead.

Second, multiple commands forced to the console may produce a number of intermediate prompts for which input has already been forced. It is sometimes hard to tell when the commands have all been executed. PDM uses conditional input and does not print a prompt for commands that are already waiting. If the commands invoke another keyboard monitor, the solution has been to print a "DONE" message as the last command.

Third, the total size of forced commands can not exceed the keyboard's input buffer quota which is normally 180 characters. So far, this has not proved to be a problem but could be handled with indirect command files.

COMMAND FILES

PDM supports a limited indirect command file facility. The "@" command invokes an indirect command file by redirecting the input. Context information is maintained which allows for commands to exit from PDM and return properly. The nesting of command files is also allowed.

A special case of the indirect command file is an initialization file PDM.INI which is invoked at the first entry into PDM.

FUNCTION KEYS

PDM allows the VT100 keys PF1-PF4 to be defined as function keys. Common commands can thus be executed with a single key stroke. The following are predefined but may be changed:

PF1 — Clear the screen

PF2 — Clear Screen and print status

PF3 — Re-do the last command

PF4 — Re-edit last file edited

PDM COMMANDS

The following is a brief summary of PDM commands. Optional portions of the commands are enclosed in square brackets. Commands which may be abbreviated include a dash after the required number of characters (e.g., B-ASIC will be accepted as B, BA, BAS, BASI, or BASIC). Angle brackets are used to designate a class of arguments (e.g., < command >).

The first set of commands is hard coded in PDM:

Define a symbol: <sym> = <dfn>;

D

Input PDM commands from a command file.

B-ASIC

Enter the BASIC+ keyboard monitor.

DATA

Identify a line of data.

DEF-INE

Define a new command or symbol. The complete syntax is below. <type> may be either "/COMMAND" OR "/SYMBOL". The default type is "/COMMAND". <nam> is the name of the symbol or command to be defined. <args> is an optional list of 1-9 arguments separated by commas. <dfn> is the definition. The ";" is a required terminator.

DEF-INE [<type>]<nam>[(<args>)] = <dfn>

If <defn> is blank, the command is taken to be an UNDEFINE. Hence, the following are equivalent:

UNDEF X

DEF X = ;

EXP-AND

Print macro expansion of a command or symbol.

FORCE

Force a command to the job's console.

PDM

Execute the specified PDM command. If installed as a CCL command, this allows non-PDM users to execute PDM commands.

PRI-NT

Print a string.

REDEF-INE

Same as DEFINE, except that previous definitions are UNDEFINE'd first.

R-UN

Run the requested program.

SH-OW

Show the current definition of the requested command(s) or symbol(s).

UNDEF-INE

Un-define the most recent definition of the requested command or symbol. The complete syntax is below.

UNDEF-INE [/ALL] [/COMMAND] < nam > [/SYMBOL]

The second set of commands is defined in the global context file:

B2

Enter the BASIC2 keyboard monitor.

BP2-BLD

Build a BASIC2 program.

BP2DBG

Build a BASIC2 program with the debugger.

EDT

Edit a file using the EDT editor.

EXIT

Exit to system default keyboard monitor.

OLD

Enter the BASIC+ keyboard monitor and load the specified program.

SORT

Sort a text file.

FUTURE ENHANCEMENTS

The following items are on the wishlist for PDM:

- (1) Add a build command to allow building command files from a template.
- (2) Improve macro processor by supporting optional arguments, default argument values, conditionals, qualifiers and noise words.
 - (3) Include a help facility.
 - (4) Include an expression evaluator.
 - (5) Include a command editing facility.
 - (6) Improve context file handling.
 - (7) Include an error correction facility.

SAMPLE COMMANDS

The following are a couple of sample PDM commands

and their definitions:

```
To Edit a file "FOO" using the EDT editor:
    cmd: EDT FOO
    dfn: EDT(FILE) = REDEF "$edfile" = FILE
        RUN $EDT
        DATA FILE
        SET KEYPAD
        CHANGE:
  The symbol $EDFILE is defined
  for later use by the re-edit
  last file function key PF4.
To Build a Basic + 2 program "FOO":
    cmd: BP2 F00
    dfn: BP2-BLD(FILE) = RUN $BASIC2
         "OLD" FILE
         COM/OBJ
         BUILD
         TKB @FILE
         PIP FILE.OBJ, FILE.CMD, FILE.ODL/DE;
```

ANNOTATED BIBLIOGRAPHY

Brown.P.J., Macro Processors and Techniques for Portable Software. (1974). [This book provides a general introduction to macro processing and includes descriptions of several different macro processors.]

Cole, A.J., Macro Processors 2nd Edition (1981).

Kernighan, Brian W. and Plaugher, P.J. "Macro Processing", Software Tools, (1976) P251-283. [The macro processor described in this chapter was the starting point for PDM.]

Wasserman, Anthony I. (ED), Software Development Environments (1981). [This is an excellent collection of papers covering a wide range of issues relating to software development environments.]

Digital Equipment Corporation, PDP-11 Macro-11 Language Reference Manual, (August 1977), DEC AA-5075A-TC.

Digital Equipment Corporation, VAX/VMS Command Language User's Guide, (May 1981), DEC AA-DO23B-TE.

Digital Equipment Corporation, VAX/VMS Guide to Using Command Procedures, (March 1980), DEC AA-H782A-

Digital Equipment Corporation, "Notes on Writing Keyboard Monitors", RSTS/E V7.2 Maintenance Notebook, (June 1982), Seq 21.1.2 N. [This article contains lots of helpful hints and suggestions.]

Digital Equipment Corporation, "Invoking a Menu Program RSX.RTS Feature Patch", RSTS/E V7.2 Maintenance Notebook, (June 1982), Seq 21.3.5. [This feature patch provides an easy way to implement a keyboard monitor program without making it a run-time system.]

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.

Get the inside story. Get the RSTS Internals Manual.

December 1982 Page 63

CACHE FLOW

By Kevin Paul Herbert, Software Techniques, Inc., 5242 Katella Avenue, Suite #101, Los Alamitos, CA 90720 Copyright © 1982 by Software Techniques, Inc.

Do you have cache flow problems? I thought I understood caching fairly well, until I heard a manager talking about cache flow problems. The managers talk about performance and so do I, but it stops there. I just don't understand how caching relates at all to whether I can get paper clips or fruit juice, but that must just be a mystery of management.

When I think of cache, I think of a fast storage medium (memory) used to store the most frequently accessed portions of a slower storage medium (disk). In many applications, it is possible to cache the most frequently used disk blocks in memory and achieve incredible performance improvements.

I am impatient. Sometimes I am called a speed freak. I don't know. I just want my programs to RUN fast.

I have some knowledge of the RSTS/E directory structure. I decided that I could use this knowledge, combined with my desire for speed, to write some very fast MACRO utilities. I learned a great deal about handling cache when working on these utilities, and I now know that it is time to spread that information to the user community. Although the techniques illustrated evolved out of use of MACRO-11 and RSTS/E, they are appropriate to all languages.

I refer to items in my cache as cache entries. A cache entry consists of a cache header followed by the cache data. Throughout all implementations of the cacher, I used contiguous address space within my job for the cache.

A cache stores only a portion of the data from the secondary storage medium. A cache entry will be removed from the cache only when the memory is needed to cache newer data. This is referred to as a cache replacement.

A cacher can either be read-only or read-write. One of my utilities required a read-only cache, and the other required a read-write cache. The read-write cache was implemented using deferred writes. This means that when I made a change to the data stored within a cache entry, I set a flag in the cache header. When the cacher needs to replace an entry, it sees that the written-to flag is set, and then writes the date out to disk. This way, disk activity occurs only at the last possible moment.

I was very interested in determining the effectiveness of the cacher, so I included statistic gathering code in all implementations. The information that I kept was the number of cache replacements, the number of accesses to the cache, and the number of accesses by the cacher to disk. These statistics are invaluable in determining an appropriate implementation.

The cache header used in my first implementation is structured as follows:

1. OLDNESS.

The OLDNESS element is set to zero when data is first installed in the cache. This counter is incremented every time a new entry is installed in the cache. This counter is decremented (to a minimum of zero) every time this cache entry is accessed. (In some applications, it is possible to reduce the number of cache replacements by zeroing the counter on access instead of decrementing it.) When it becomes necessary to perform a cache replacement, the cache scans all the cache headers to find the cache entry with the largest OLDNESS.

2. BLOCK

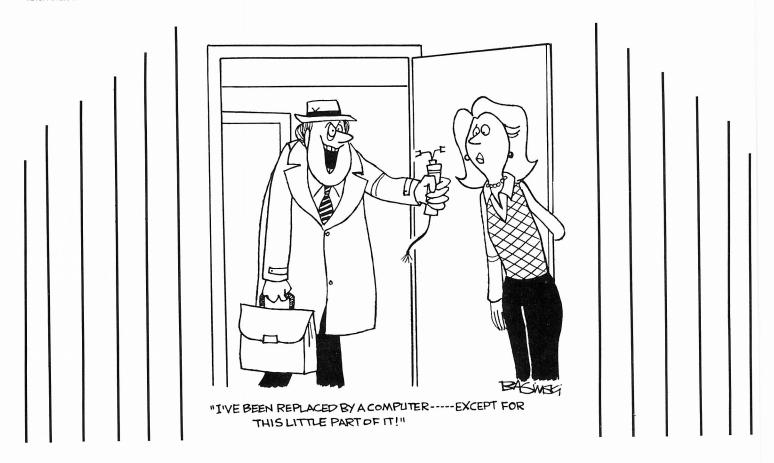
The BLOCK element contains the disk block number associated with this cache entry. When the cacher is called to locate a disk block, it scans all the cache headers to find a matching block number before performing a cache replacement and reading the block from disk.

All scanning of cache headers is performed through sequential access. Disk activity is reduced a great deal through this implementation, but there is a high cost in CPU time.

My second implementation addressed the CPU time issue. After examining the statistics I had collected, I concluded that the newer a cache entry was, the more likely it was to be accessed. I therefore directed my energy toward reducing the number of memory accesses necessary to locate a cache entry. The cache header used in the second cacher is set up as follows:

1. LINK-TO-NEXT-OLDEST

The LINK-TO-NEXT-OLDEST element contains the address of the next oldest cache entry. When an entry is added to the cache, a cell called LINK-TO-THE-NEWEST is set to contain the address of this new entry, and the LINK-TO-NEXT-OLDEST element is set to the value previously stored in LINK-TO-THE-NEWEST. Whenever this cache entry is accessed, it is replaced as the LINK-TO-THE-NEWEST entry since it is the likeliest cache entry to be accessed upon the next cache access.



C-CALC

- C-CALC is the high-performance electronic spreadsheet written in the extremely CPU-efficient 'C' languagedesigned with speed, efficiency, and transportability in mind.
- C-CALC is extremely user-friendly, with an ON-LINE HELP feature, comprehensive documentation, and built in training procedure. C-CALC is a 'menu' oriented system with easy to use single level commands.
- DIGITEC Software Design, Incorporated offers on-going product maintenance and extensive customer support services.
- C-CALC is currently or soon to be available for the following operation systems: RSTS/E*, RSX-11M*, IAS*, RT11*, VAX/VMS*, TOPS 10/20*, UNIX**, VM/CMS(IBM), GCOS(HONEYWELL), and many more. Data is fully transportable between operation systems.

For more information, Call: (206) 821-7507

- Registered trademark of Digital Equipment Corporation
- · Registered trademark of Bell Laboratories

THE ELECTRONIC SPREADSHEET FOR **ALL** COMPUTERS

C-CALC's features include:

- · Fully transportable data
- Supports most types of terminals
- Extensive ON-LINE HELP
- Built in training procedures
- Menus and single level commands
- Variable width columns
- Comprehensive worksheet consolidation
- Computed Coordinates
- Alpha or Numeric Column Coorordinates
- IF THEN ELSE
- · Conditional formatting
- Equations may contain text

DIGITEC Software Design, Inc.

14125 - 108th Avenue NE, Kirkland, WA 98033 (206) 821-7507

2. LINK-TO-NEXT-NEWEST

The LINK-TO-NEXT-NEWEST is used as a backwards link to implement quick cache replacements. A cell LINK-TO-THE-OLDEST is used to point to the oldest cache entry. When a cache replacement occurs, the value of LINK-TO-

NEXT-NEWEST is written to LINK-TO-THE-OLDEST.

3. BLOCK

The BLOCK element contains the disk block number associated with this cache entry. When the cacher is called to locate a disk block, it scans all the cache headers to find matching block number before performing a cache replacement and reading the block from disk.

This implementation reduced CPU time used by the cacher without affecting the disk activity at all. There were two areas of improvement in the cacher. First, I reduced CPU time necessary to locate cache entries. This occurred because of my statistical observation dealing with the way I accessed the disk. Second, it was no longer necessary

to scan all the cache blocks to find the oldest entry. This improved performance when there were many cache replacements.

I was not yet satisfied. Some person at DECUS dared me to make my program run even faster. I consulted Deep Thought and came up with an answer. The newest and oldest cache entries were now locatable just by examining

one fixed cell. That improved performance a great deal. What if I could locate any cache entry by just examining one cell (rather than scanning the block entry in the cache header)? This would reduce CPU time even further!

I concluded that if the range of block numbers was

relatively small, I could have a cell for each possible block number. This cell would either contain a zero to indicate that the block was not in the cache, or contain the address of the cache entry. One of my utilities used the cache to cache an entire RSTS/E file structured disk. There was no way to have a cell for each block of an RMO5, so I forgot the idea in that utility. The utility, other however, cached a RSTS/E UFD and a temporary work file. The total number of cells that I would need in this utility was only 14. This was very workable.

It was not necessary to change anything in the cache header. The block number was still necessary to facilitate cache replacements. All I added was a static region containing 14 cells, one for each possible cache entry. This change made a noticable

increase in performance and finally satisfied me.

Who says speed kills? I program for speed, and I hope that reading this article has excited the speed freak in you.

Have fun, and don't have any cache flow problems! ♥

Software Tools for RSTS/E

Evans Griffiths & Hart, Inc., a pioneer in the development of RSTS and the winner of an ICP million dollar award for KDSS and TAM, offers packages that save you time and improve your productivity.

- KDSS, a complete multi-terminal key-to-disk data entry subsystem. Eliminates the need for keypunching and stand-alone key-to-disk systems. (Also available for VAX/VMS and RSX-11M.)
- TAM, an efficient multi-terminal screen-handling facility that provides complete support for the development of transaction-processing applications on a wide variety of terminals. (Also available for VAX/VMS and RSX-11M.)
- **FSORT3**, a very fast sort/merge package for RMS and non-RMS files. More economical of disk space than SORT-11 and much faster.
- **SELECT**, a convenient, very fast package for scanning files to extract records that meet user-specified selection criteria. Use as part of an online inquiry system and as a front end for building file indices and generating reports. SELECT and FSORT3 can save hours in nightly batch
- BSC/DV, a RSTS/E device driver for the DEC DV11 synchronous multiplexer. Suitable for handling a wide variety of bisynchronous protocols.
 (Also available for VAX/VMS.)
- COLINK, a convenient, efficient link between two RSTS/E systems
 using DMC11s or DMR11s without the overhead of DECnet. Supports
 file transfers, virtual terminals, and across-the-link task communication.
- DIALUP, a comprehensive, efficient link between RSTS/E and other systems using asynchronous terminal lines. Supports file transfers, virtual terminals, auto-dialing, and the use of command files and macros. The premier RSTS/E package for remote support and reliable, CPUefficient file transfers.

DEC, DECnet, RSTS, RSX, VAX, and VMS are trademarks of Digital Equipment Corporation.

Call or write for complete descriptions of features and benefits.

Evans Griffiths & Hart, Inc.

55 Waltham Street, Lexington, MA 02173 (617) 861-0670

IV Call

```
Z9$ = FNPRINT$(X40$)
                                                                                              850
SIMINI... continued from page 58
                                                                                                       Z9% = FNLINEFEED%
                                                                                              860
                                                                                              870
                                                                                                       FNOCT% = T40%
                                                                                              880
                   EXTEND
                                                                                              890
                                                                                                                                                        PRINT CHARACTER STRING.
                                                                                              900
                   SIMULATION MINICOMPUTER.
                                                                                              910
                                                                                                       DEF FNPRINT$(X40$)
2
                                                                                                       DET FRETAINIŞ(ANUŞ)
Z9% = FNCURSOR%(OPP% + 16%, CHP%)
T40% = (6% - OPP%) * 80% + (80% - CHP%)
IF LEN(X40%) > T40%
THEN PRINT LEFT(X40%, T40%);
OPP% = 1%: CHP% = 0%:

NACA - DRUMY(MOME AND OM : 10%)
                                                                                              920
                   Designed as a teaching aid, SIMINI traces the
                                                                                              930
                   contents of the machines registers during
                                                                                              940
                   program execution.
                   Programs written in SIMINI assembler may be
                                                                                                                 X40$ = RIGHT(X40$, T40% + 1%):
                   maintained on backing store.
                                                                                                                 GO TO 920
         1
                                                                                                        PRINT X40$;
3
                                                                                                        CHP% = CHP% + LEN(X40$)
                                                                                              960
                   AUTHOR :
                                 John N Cato.
                                                                                                        IF CHP% < 80% THEN 1030
                                                                                              970
                   DATE :
                                                                                                        CHP\% = CHP\% - 80\%

OPP\% = OPP\% + 1\%
                                 Jan 1978.
                                                                                              990
                                                                                                        IF OPP% < 7% THEN 970
4
                                                                                              1000
                                                                                                        OPP% = 1%
                                                                                              1010
                                                                                                        GO TO 970
                                                                                               1020
12
          INPUT "Print speed factor"; Z1%
                                                                                                        PRINT ELINE$;
                                                                                              1030
          GO TO 15000
                                                                                                        FNEND
15
                                                                                              1040
                                                                                                                                                        PRINT DECIMAL INTEGER
          DIM MASK%(15%)
                                                                                               1050
20
         DIM M8$(31%), M9$(40%)
DIM ST%(511%)
                                                                                                        DEF FNPRINT%(X40%)
                                                                                              1060
                                                                                                        X41$ = NUM$(X40%)
40
                                                                                              1070
                                                                                                        Z9$ = FNPRINT$(X41$)
          DIM Y%(30%)
                                                                                               1080
50
          FNEND
90
                                                                                               1140
                                                                                                                                                        LNFEED(CLEARS LINE ALSO)
                                                         MOVE CURSOR
 100
                                                                                               1150
          DEF FNCURSOR% (D%, R%)
                                                                                                        DEF FNLINEFEED%
110
                                                                                               1160
          PRINT ESC$+"Y"+CHR$(32%+D%)+CHR$(32%+R%);
                                                                                                        IF OPP% = 6%
THEN OPP% = 1%
120
                                                                                              1170
 150
          FNEND
                                                          PRINT BIT PATTERN
                                                                                                           ELSE
200
                                                                                                                  OPP% = OPP% + 1%
          DEF FNP% (X50%)
210
220
          FOR 150% = 15% TO 0% STEP -1%
                                                                                               1180
                                                                                                         CHP% = 0%
                                                                                                         Z9\% = FNCURSOR\%(OPP\% + 16\%, CHP\%)
          IF Z1%>2500
225
                                                                                               1185
                                                                                                         PRINT ELINE$;
               THEN SLEEP INT(Z1%/2500%)
                                                                                               1190
                                                                                                         FNEND
                                                                                               1200
                                                                                                                                                         CONVERT AN INTEGER TO
          PRINT ""; FOR I51%=0% TO Z1%
IF ( X50% AND MASK%(I50%) )
THEN PRINT "1";
                                                                                               1210
                                                                                                                                                         AN OCTAL STRING
                                                                                               1220
230
                                                                                                         DEF FNOCT$(X50%, X51%)
                                                                                               1230
                                                                                                         IF X50% < 0% THEN
             ELSE
                                                                                               1235
                    PRINT "O":
                                                                                                                  T51$ = "1"
          NEXT 150%
                                                                                                         ELSE
 240
 250
          FNEND
                                                                                                                  T51$ = "0"
                                                          LOAD OP CODE INTO THE
                                                                                                         X50% = X50% AND 32767%
T50$ = ""
 300
                                                                                               1240
                                                          4 MOST SIGNIFICANT BITS
 310
                                                                                               1245
                                                                                                         T50%=X50%/8%
          DEF FNLOC%(OC1%)
 320
                                                                                               1250
                                                                                                         T51%=X50%-8%* T50%
T50$ =NUM1$(T51%)+ T50$
X50%= T50%
          FNLOC$ = OC1$ * MASK$(11%)
                                                                                               1260
 330
 340
          FNEND
                                                                                               1270
                                                          LOAD INDEX BIT INTO
 350
                                                                                               1280
                                                          INSTRUCTION.
                                                                                                         IF X50%<0% THEN 1250
T50$ = "0" + T50$ UNTIL LEN(T50$) = 5%
T50$ = RIGHT(T51$+T50$,7%-X51%)
                                                                                               1290
 355
 360
          DEF FNLX%(INSTR1%)
                                                                                               1295
          FNLX% = INSTR1% OR MASK%(10%)
 365
                                                                                               1300
                                                                                                         FNOCT $= T50$
                                                                                               1305
 370
375
          FNEND
                                                          LOAD INDIRECT BIT INTO
                                                                                               1310
                                                                                                         FNEND
                                                          INSTRUCTION.
                                                                                                1320
 380
                                                                                                                                                         FNP1$
          DEF FNLIND$(INSTR1%)
 385
                                                                                               1330
                                                                                                         DEF FNP1$(X50%)
T50$ = ""
          FNLIND% = INSTR1% OR MASK%(9%)
                                                                                               1340
 390
 395
          FNEND
                                                                                                1350
                                                           LOAD ADDRESS INTO THE
                                                                                                         FOR 150% = 15% TO 0% STEP -1%
 400
410
                                                                                                1360
                                                                                                         IF (X50% AND MASK%(I50%))
                                                           INSTRUCTION
                                                                                                1370
           DEF FNLAD%(AD1%, INSTR1%)
                                                                                                             THEN T50$ = T50$ + "1"
 420
           FNLAD% = AD1% OR INSTR1%
 430
                                                                                                           ELSE
                                                                                                                   T50$ = T50$ + "0"
 440
          FNEND
                                                           SEPARATE OP CODE FROM
                                                                                                1380
                                                                                                         NEXT I50%
 500
                                                           INSTRUCTION
                                                                                                         FNP1$ = T50$
 510
                                                                                                1390
           DEF FNSOC$(INSTR1$)
 520
                                                                                                1400
                                                                                                         FNEND
          INSTRI% = INSTRI% / 2%
INSTRI% = INSTRI% AND 32767%
                                                                                                1410
 530
                                                                                                                                                         PRINT INSTRUCTION
                                                                                                1420
 540
                                                                                                         DEF FNPI$(X40%)
           FNSOC% = INSTRI% / MASK%(10%)
 550
                                                                                                1430
                                                                                                         OC% = FNSOC%(X40%)
           FNEND
 560
                                                                                                1440
                                                           SEPARATE ADDRESS FROM
                                                                                                         AD$ = FNSAD$(X40$)

IF OC$ = 0$ OR OC$ > NOMREF$ THEN 1510

T40$ = M8$(OC$) + " + FNOCT$(AD$,ADL$)
                                                                                                1450
 600
                                                           INSTRUCTION
                                                                                                1460
 610
           DEF FNSAD%(INSTR1%)
                                                                                                1470
 620
                                                                                                          IF X40% AND MASK%(10%)
           FNSAD% = INSTRI% AND MEM%
 630
                                                                                                1480
                                                                                                         THEN T40$ = T40$ + "X"
IF X40$ AND MASK$(9$)
 640
                                                           CONVERT AN N9% DIGIT
                                                                                                1490
 650
                                                           STRING TO SINGLE INTEGER OCTAL - DECIMAL.
                                                                                                             THEN T40$ = T40$ + "I"
                                                                                                          GO TO 1520
                                                                                                1500
 670
                                                           A NON-OCTAL DIGIT CAUSES
                                                                                                          THEN T40$ = M9 $(X40$)

ELSE T40$ = M9 $(X40$)
                                                                                                1510
 680
                                                           AN ERR MESS & A -VE VAL
 690
           DEF FNOCT%(X40$,X40%)
 700
710
                                                                                                          FNPI$ = T40$
                                                                                                1520
 720
           T40% = 0%
                                                                                                1530
                                                                                                          FNEND
                                                           CONVERSION ROUTINE
 730
                                                                                                1540
                                                                                                                                                          FNCVTS
                                                                                                1550
           T41\% = VAL(MID(X40\$, I40\%, 1\%))
 740
           IF T41% > 7%
THEN FNDEC% = -1%
                                                                                                          DEF FNCVT$( X50% )
T50% = (X50% AND 255%)
IF T50% > 31% AND T50% < 97%
 750
                                                                                                1560
                                                           ! Not octal.
                                      \ GO TO 840
                                                                                                1570
           T40% = T40% + T41%
                                                                                                1580
 770
           IF 140% = X40%
THEN 880
                                                                                                             THEN T50$ = RIGHT(CVT%$(T50%),2%)
 780
                                                                                                            ELSE
           T40% = T40% * 8%
I40% = I40% + 1%
                                                                                                                   T50$ = " "
  800
                                                                                                          T50% = SWAP%( X50% AND -256%)
IF T50% > 31% AND T50% < 97%
THEN T50% = RIGHT(CVT%$(T50%),2%) + T50$
  810
                                                                                                1590
           GOTO 740
  820
                                                           ERROR ROUTINE
  830
           X40$ = X40$ + " IS NOT AN OCTAL NUMBER!"
```

```
DISPLAY MOR BIT PATTERN
                                                                                     3700
          ELSE
                T50$ = " " + T50$
                                                                                              Z9\% = FNCURSOR\%(8\%, 32\%)
                                                                                     3710
        FNCVT$ = T50$
                                                                                              Z9% = FNP%(MDR%)
1610
                                                                                     3720
                                                                                     3730
                                                                                              RETURN
1620
        FNEND
                                                                                     3740
1630
                              : GOSUB 3100
: GOSUB 3700
                                                                                                                                          DISPLAY SP BIT PATTERN
        MAR% = AD%
                                                                                     3800
2000
                                                                                              Z9% = FNCURSOR%(10%,15%)
Z9% = FNP%(SP%)
         \MDR% = ST%(MAR%)
                                                                                     3810
                                                                                     3820
                                                                                     3830
                             : GOSUB 3100
: GOSUB 3700
                                                                                     3840
        MAR% = SP%
\MDR% = ST%(MAR%)
2100
                                                                                                                                          DISPLAY X BIT PATTERN
                                                                                     3900
                                                                                              Z9% = FNCURSOR%(10%,38%)
Z9% = FNP%(X%)
                                                                                     3910
         \RETURN
                                                                                     3920
                                                                                              RETURN
                                                                                     3930
2200
        T20%=0%
        T20$ = 0$

IF ( ACC$ AND MDR$ ) < 0$

THEN T20$ = -1$

ELSE IF ( ACC$ OR MDR$ ) > 0$

THEN T20$ = +1$

ACC$ = ACC$ + MDR$ : GOSUB 3300
                                                                                     3940
2210
                                                                                              .
                                                                                     3950
                                                                                                                                         INDIRECT ADDRESSING
                                                                                     4000
                                                                                              MAR% = AD%
MDR% = ST%(MAR%)
AD% = MDR%
                                                                                     4010
                                                                                                                        : GOSUB 3100
                                                                                                                       : GOSUB 3700
                                                                                     4020
2220
        IF ( ACC% XOR T20% ) < 0%
THEN OV% = 1%
ELSE OV% = 0%
                                                                                     4030
2230
                                                                                              RETURN
                                                                                     4040
                                                     ! OVERFLOW ?
                                                                                     4050
                                                                                                                                          PRINT INSTRUCTION
         GOSUB 3500
                                                                                     4099
2240
                                                                                              Z9%=FNCURSOR%(14%, 44%)
2250
         RETURN
                                                                                     4100
                                                                                              PRINT ELINE$;
PRINT FNOCT$(PC%-1%,ADL%) + " ";
                                                                                     4120
2260
                                                                                     4130
        MDR% = -MDR%
2300
                                                                                              PRINT FNPI$(IR%);
2310
         GOSUB 2200
                                                                                     4140
                                                                                     4150
2320
         RETURN
                                                                                                                                          PRINT MESSAGE
                                                                                     4200
2330
                                                                                              ZQ$ = FNPRINT$(ERRM$)
         T20=ACC% *MDR%
                                                                                     4210
2400
                                                                                              Z9% = FNLINEFEED%
         IF T20>32767 OR T20<-32768
                                                                                      4220
2410
                                                                                              RETURN
                                                                                      4230
             THEN OV%=1%
                                                                                     4240
                                                                                                                                          SYNTAX ERROR
                 OV%=0%
                                                                                      4300
                                                                                              ERRM$ = "SYNTAX ERROR " + NUM$(FLAG$)
                                                                                      4310
        GOSUB 3500
ACC% =T20
2420
                                                                                      4320
                                                                                              GOSUB 4200
                                    : GOSUB 3300
2430
                                                                                      4330
         RETURN
2440
                                                                                              RETURN
                                                                                      4340
2450
        T20% = ACC% / MDR%
R% = ACC% - MDR% * T20%
                                                                                      4350
2500
                                                                                                                                          I/O FIELD ASSOCIATION
                                           : GOSUB 3400
                                                                                      4400
2510
                                                                                              T10% = T10% + 1%
                                                                                      4410
         ACC% = T20%
                                      : GOSUB 3300
2520
                                                                                              FIELD #1%, (T10%-1%)#4% AS T11$.
         OV$ = 0$
                                     : GOSUB 3500
2530
                                                                                                               2% AS T12$, 2% AS T13$
2540
2550
         RETURN
                                                                                              RETURN
                                                                                      4430
                                  : GOSUB 3700
         MDR% = ACC%
2600
                                                                                              .
         ST%(MAR%) = MDR%
                                  : GOSUB 2900
                                                                                      4990
2610
         RETURN
                                                                                      5000
2620
                                                                                              GOSUB 2000
                                                                                      5010
2630
                                                                                              ACC% = MDR%
                                                                                                                        : GOSUB 3300
                                            PRINT STORE
                                                                                      5030
2900
                                                                                      5040
                                                                                              RETURN
         Z9\% = FNCURSOR\%(4\%,59\%)
2910
         \PRINT SPACE$(9%);
                                                                                      5050
                                                                                                                                          STO
         \Z9\% = FNCURSOR\%(4\%,59\%)
                                                                                      5100
                                                                                                                         : GOSUB 3100
                                                                                      5110
                                                                                              MAR% = AD%
         \SLEEP INT((Z1%+2500%)/2500%)
\PRINT "S T O R E";
                                                                                              GOSUB 2600
                                                                                      5120
                                                                                              RETURN
                                                                                      5130
                                                                                                                                           ADD
                                                                                      5200
         RETURN
2920
                                                                                              GOSIIB 2000
                                                                                      5210
                                                                                              GOSUB 2200
         5220
5230
2990
                                                                                              RETURN
                                                    DISPLAY PC BIT PATTERN
3000
                                                                                                                                           SUB
                                                                                      5300
         Z9\% = FNCURSOR\%(0\%.9\%)
3010
                                                                                              GOSUB 2000
                                                                                      5310
         Z9% = FNP%(PC%)
3020
                                                                                              GOSUB 2300
         RETURN
                                                                                      5320
3030
                                                                                              RETURN
                                                                                      5330
3040
                                                                                                                                           MUL
                                                     DISPLAY MAR BIT PATTERN
                                                                                      5400
3100
                                                                                              GOSUB 2000
                                                                                      5410
         Z9\% = FNCURSOR\%(0\%, 32\%)
3110
                                                                                      5420
                                                                                              GOSUB 2400
3120
         Z9% = FNP%(MAR%)
                                                                                              RETURN
                                                                                      5430
         RETURN
3130
                                                                                                                                           DIV
                                                                                      5500
3140
                                                                                              GOSUB 2000
                                                     DISPLAY IR BIT PATTERN
                                                                                      5510
3200
                                                                                      5520
                                                                                               GOSUB 2500
         Z9% = FNCURSOR%(2%,9%)
3210
                                                                                      5530
                                                                                              RETURN
3220
         Z9\% = FNP\%(IR\%)
                                                                                                                                           AND
                                                                                      5600
5610
         RETURN
3230
3240
                                                                                              ACC% = ACC% AND MDR%
RETURN
                                                                                      5630
                                                                                                                        : GOSUB 3300
                                                     DISPLAY ACC BIT PATTERN
3300
                                                                                      5640
         Z9% = FNCURSOR%(4%,9%)
3310
                                                                                      5650
         Z9\% = FNP\%(ACC\%)
3320
                                                                                                                                           OR
                                                                                      5700
         RETURN
3330
                                                                                      5710
                                                                                               GOSIIB 2000
3340
                                                                                               ACC% = ACC% OR MDR%
                                                                                                                         : GOSUB 3300
                                                                                      5730
5740
                                                     DISPLAY R BIT PATTERN
3400
                                                                                               RETURN
         Z9% = FNCURSOR%(6%,9%)
Z9% = FNP%(R%)
3410
                                                                                      5750
 3420
                                                                                                                                           XOR
                                                                                      5800
         RETURN
 3430
                                                                                               GOSUB 2000
                                                                                      5810
3440
                                                                                               ACC% = ACC% XOR MDR%
                                                                                                                         : GOSUB 3300
                                                                                      5830
                                                     DISPLAY OV
3500
                                                                                               RETURN
3510
         Z9\% = FNCURSOR\%(8\%, 12\%)
                                                                                      5850
         IF OV% = 0%
THEN PRINT "O";
3520
                                                                                                                                           JMP
                                                                                      5900
                                                                                               PC% = AD%
                                                                                                                         : GOSUB 3000
                                                                                      5910
           ELSE PRINT "1";
                                                                                      5920
                                                                                               RETURN
3550
         RETURN
3560
                                                                                      5930
                                                                                                                                           JNZ.
                                                     DISPLAY C
                                                                                      6000
3600
                                                                                               IF ACC% <> 0%
                                                                                      6010
         Z9% = FNCURSOR%(8%,19%)
IF C% = 0%
 3610
                                                                                                                         : GOSUB 3000
                                                                                                  THEN PC% = AD%
3620
                                                                                               RETURN
                                                                                      6020
            THEN PRINT "O";
           ELSE PRINT "1";
                                                                                      6100
         RETHEN
3630
                                                                                              IF ACC% < 0%
                                                                                      6110
3640
```

RSTSPROFESSIONALRSTSPROFESSI

```
************************
           THEN PC% = AD%
                                 : GOSUB 3000
                                                                                 8000
                                                                                                                                   CI.A
                                                                                 9000
        RETURN
6120
                                                                                          ACC% = 0%
                                                                                                                  : GOSUB 3300
                                                                                 9010
6130
                                                  INCH
                                                                                          RETURN
                                                                                 9020
6200
                                                                                 9030
        GOSUB 2000
6210
                                                                                                                                   TNC
                                 : GOSUB 3700
                                                                                 9100
        MDR% = MDR% + 1%
6230
                                                                                         IF ACC% = 32767%
THEN OV% = 1%
                                                                                 9110
6240
        IF MDR% = 0%
           THEN PC% = PC% +1% : GOSUB 3000
                                                                                            ELSE
                                 : GOSUB 2900
        ST%(MAR%) = MDR%
6250
        RETURN
6260
                                                                                 9120
                                                                                          GOSUB 3500
6270
                                                                                          ACC% = ACC% + 1%
                                                                                                                   : GOSUB 3300
                                                                                 9130
6300
                                                                                          RETURN
                                 : GOSUB 3800
                                                                                 9140
        SP% = SP% -1%
6310
                                                                                 9150
        MAR% = SP%
MDR% = PC%
ST%(MAR%) = MDR%
                                 : GOSUB 3100
6320
                                                                                                                                    DEC
                                                                                 9200
                                 : GOSUB 3700
6330
                                                                                 9210
                                                                                          IF ACC% = -32768
                                 : GOSUB 2900
6340
                                                                                             THEN OV% = 1%
                                  : GOSUB 3000
        PC% = AD%
6350
                                                                                            ELSE
6360
                                                                                                 OV$ = 0$
6370
                                                                                          GOSUB 3500
                                                  CAHE
                                                                                 9220
6400
                                                                                          ACC% = ACC% - 1%
                                                                                                                   : GOSUB 3300
                                                                                 9230
         GOSUB 2000
6410
                                                                                          RETURN
                                                                                  9240
        IF ACC% = MDR%
6430
                                                                                 9250
           THEN PC% = PC% + 1% : GOSUB 3000
                                                                                                                                    ADDS
                                                                                 9300
6440
         RETURN
                                                                                          GOSUB 12100
                                                                                 9310
6450
                                                  LDX
                                                                                          \GOSUB 2100
6500
                                                                                          \GOSUB 2200
6510
         GOSUB 2000
                                                                                          \GOSUB 2600
                                  : GOSUB 3900
         \X% = MDR%
                                                                                           RETURN
         RETURN
                                                                                                                                    SHBS
                                                                                  9400
                                                  STOX
6600
                                                                                          GOSUB 12100
         MAR% = AD%
                                  : GOSUB 3100
6610
                                                                                          \GOSUB 2100
                                  : GOSUB 3700
         \MDR% = X%
                                                                                           \T20% = MDR%
         \ST%(MAR%) = MDR%
                                  : GOSUB 2900
                                                                                          MDR% = ACC%
         \RETURN
                                                                                           \ACC% = T20%
                                                   LDPC
                                                                                           \GOSUB 2300
6700
                                                                                           GOSUB 2600
         GOSUB 2000
6710
                                                                                           RETURN
                                  : GOSUB 3000
         \PC% = MDR%
         RETURN
                                                                                                                                    SWAB
                                                                                  9500
                                                                                          ACC% = SWAP(ACC%)
                                                                                                                   : GOSUB 3300
                                                   STPC
                                                                                  9510
6800
                                                                                  9520
                                                                                          RETURN
                                  : GOSUB 3100
         MAR% = AD%
6810
                                                                                  9530
         \MDR% = PC%
\ST%(MAR%) = MDR%
                                  : GOSUB 3700
                                                                                                                                    HOP
                                                                                  9600
                                                                                                                    : GOSUB 3000
                                                                                  9610
                                                                                          PC% = PC% + 1%
                                                                                  9620
                                                                                          RETURN
                                                   JAP
                                                                                  9630
 6900
                                                                                                                                    HOPE
         IF ACC% > 0%
                                                                                  9700
 6910
                                                                                          IF ACC% = 0%
                                                                                  9710
                                  : GOSUB 3000
            THEN PC% = AD%
                                                                                             THEN GOSUB 9600
 6920
         RETURN
                                                                                           RETURN
                                                                                  9720
                                                                                  9730
                                                   JAZ
 7000
                                                                                                                                     HOPN
                                                                                  9800
         IF ACC% = 0%
 7010
                                                                                          IF ACC% <> 0%
                                  : GOSUB 3000
                                                                                  9810
            THEN PC% = AD%
                                                                                             THEN GOSUB 9600
 7020
         RETURN
                                                                                  9820
                                                                                           RETURN
                                                   JXNZ
                                                                                  9830
 7100
                                                                                                                                    HOPL
                                                                                  9900
         IF X% <> 0%
 7110
                                                                                           IF ACC% <0%
                                                                                  9910
            THEN PC% = AD%
                                  : GOSUB 3000
                                                                                             THEN GOSUB 9600
 7120
                                                                                           RETURN
                                                                                  9920
                                                   JXLZ
                                                                                  9930
 7200
                                                                                                                                     HOPG
                                                                                   10000
         IF X% < 0%
 7210
                                                                                           IF ACC% > 0%
                                                                                  10010
                                  : GOSUB 3000
            THEN PC% = AD%
                                                                                              THEN GOSUB 9600
         RETURN
 7220
                                                                                  10020
                                                                                           RETURN
                                                                                  10030
                                                   JXP
 7300
                                                                                                                                     NOT
                                                                                  10100
         IF X% > 0%
THEN PC% = AD%
 7310
                                                                                           ACC% = ( NOT (ACC%)) : GOSUB 3300
                                                                                   10110
                                   : GOSUB 3000
                                                                                   10120
                                                                                           RETURN
         RETURN
 7320
                                                                                   10130
                                                                                                                                     NEG
                                                    JXZ
                                                                                   10200
 7400
                                                                                           ACC% = -ACC% : GOSUB 3300
                                                                                   10210
         IF X% = 0%
 7410
                                                                                           IF ACC% = -32768
THEN OV% = 1%
                                                                                   10220
                                   : GOSUB 3000
             THEN PC% = AD%
 7420
                                                                                                   OV% = 0%
                                                    JOV
 7500
                                                                                           GOSUB 3500
                                                                                   10230
         IF OV% <> 0%
 7510
                                                                                           IF ACC% =0%
                                                                                   10240
            THEN PC% = AD%
                                   : GOSUB 3000
                                                                                              THEN C% = 0%
 7520
                                                                                             ELSE
                                                                                                   C% = 1%
                                                    JNOV
 7600
                                                                                           GOSUB 3600
                                                                                   10250
          IF OV% = 0%
 7610
                                                                                   10260
                                                                                           RETURN
             THEN PC% = AD%
                                   : GOSUB 3000
                                                                                   10270
          RETURN
 7620
                                                                                                                                     TNCX
                                                                                   10300
                                                                                           X% = X% + 1% : GOSUB 3900
                                                                                   10310
                                                    JSI
 7700
                                                                                   10320
                                                                                           RETURN
          GOSUB 2000
 7710
                                                                                   10330
10400
                                   : GOSUB 3200
          \IR% = MDR%
                                                                                                                                     DECX
          RETURN
                                                                                                                    : GOSUB 3900
                                                                                   10410
                                                                                           X% = X% - 1%
                                                    CXHE
                                                                                   10420
                                                                                           RETURN
 7800
                                                                                   10430
          GOSUB 2000
 7810
                                                                                                                                     SWAX
                                                                                   10500
          \IF X% = MDR%
                                                                                           T20% = ACC%
             THEN PC% = PC% + 1% : GOSUB 3000
                                                                                                                    : GOSUB 3300
                                                                                           ACC% = X%
                                                                                   10520
          RETURN
 7820
```

RSTSPROFESSIONALRSTSPROFESSI



11010 IF ACC% < 0% : GOSUB 3900 X% = T20% 10530 THEN T20% = 1% RETURN 10540 ELSE 10550 T20% = 0% SWAR 10600 ACC% = (ACC% * 2%) OR T20% 11020 T20% = ACC% 10610 GOSUB 3300 : IF T20% = 1% 11030 : GOSUB 3300 : GOSUB 3400 10620 ACC% = R% 10630 R% = T20% THEN T20% = -32768 IF (ACC% XOR T20%) 10640 RETURN 11040 10650 THEN OV% = 1% ASL 10700 ELSE IF ACC% < 0% THEN T20% = -1% 10710 OV% = 0% 11050 GOSUB 3500 T20% = 0% ACC% = ACC% * 2% : IF (ACC% XOR T20%) < 0% 11060 RETURN 11070 : GOSUB 3300 10720 CLV 11100 10730 : GOSUB 3500 11110 OV% = 0% THEN OV% = 1% RETURN 11120 11130 OV\$ = 0\$ SEV 11200 GOSUB 3500 10740 : GOSUB 3500 11210 OV% = 1% RETURN 10750 RETURN 11220 10760 11230 ASR 10800 HOPV 11300 ACC% = ACC% / 2% : GOSUB 3300 10810 IF OV% <> 0% 11310 10820 IF ACC% < 0% THEN PC% = PC% + 1% : GOSUB 3000 THEN C% = 1% 11320 RETURN ELSE 11330 C%=0% CLC 11400 GOSUB 3600 10830 11410 C% = 0% : GOSUB 3600 : GOSUB 3500 OV% = 0% 10840 11420 RETURN RETURN 10850 11430 10860 SEC 11500 ROR 10900 : GOSUB 3600 11510 T20% = -(ACC% AND 1%) 10910 11520 RETURN ACC% = ACC% / 2% IF (ACC% XOR T20%) THEN OV% = 1% 10920 11530 10930 HOPC 11540 11600 IF C% <> 0% THEN PC% = PC% + 1% : GOSUB 3000 11610 OV% = 0% ! **** IFT20%=0%THEN ACC%=ACC%IMP0% **** 10940 11620 RETURN GOSUB 3300 10950 11630 10960 GOSUB 3500 11700 10970 RETURN : GOSUB 3500 OV\$ = 0% 11710 10980 ! C% = 0% : GOSUB 3600 ROL 11720 11000

```
12990
11730
                                                                                                                                                                     SET UP SCREEN
                                                                                                       15000
11740
                                                                                                                 ESC$ = CHR$(155%)
                                                               SCV
                                                                                                       15001
11800
                                                                                                                  \ESCREEN$ = ESC$ + "J"
\ELINE$ = ESC$ + "K"
\CHOME$ = ESC$ + "H"
                                          : GOSUB 3500
          OV% = 1%
11810
                                          : GOSUB 3600
          C% = 1%
11820
          RETURN
11830
                                                                                                                  CDOWN$ = ESC$ + "B"
11840
                                                                                                                  \CRIGHT$ = ESC$ + "C"
                                                               SWAP
11900
11910
          T20% = ACC%
                                          : GOSUB 3300
                                                                                                                 MEM% = 511%
          ACC% = SP%
                                                                                                       15002
11920
                                                                                                                 \OMEM% = 777%
NOMREF% = 29%
          SP% = T20%
                                          : GOSUB 3800
11930
                                                                                                       15003
           RETURN
11940
                                                                                                                  \NOREG% = 40%
11950
                                                                                                                 ADL% = 3%
\ADL1% = ADL% + 1%
                                                               PUSH
12000
                                                                                                       15004
12010
                                          : GOSUB 3800
                                                                                                                  \ADL2% = ADL% + 2%
           SP% = SP% - 1%
12020
12030
           MAR% = SP%
                                          : GOSUB 3100
                                                                                                       15005
                                                                                                                  SPPOS% = ADL%+1%
12040
           GOSUB 2600
                                                                                                                  \MNEMPOS$ = SPPOS$+1$
\OCTPOS$ = MNEMPOS$+1$
\LITLEN$ = MNEMPOS$+3$
12060
           RETURN
12070
                                                                POP
12100
                                                                                                                   REGLENS = MNEMPOSS+3%
           IF SP% = MEM%+1%
12110
              THEN ERRM$ = "STACK EMPTY"
                                                                                                                  OCTLEN% = MNEMPOS%+6%
                     Z9$ = FNPRINT$(ERRM$)
                                                                                                                                                  1 Op code of JSI
                                                                                                                  JSI% = 28%
                     Z9% = FNLINEFEED%
                                                                                                       15006
                                          : GO TO 12160
                     E9% = 1%
                                                                                                                  PRINT CHOME $;
           GOSUB 2100
                                                                                                       15010
12120
                                          : GOSUB 3300
                                                                                                                  PRINT ESCREEN$;
           ACC% = MDR%
                                                                                                        15020
 12140
                                                                                                                  T00$ = "000000000000000000"
T01$ = "
                                          : GOSUB 3800
                                                                                                        15030
 12150
           SP% = SP% + 1%
                                                                                                        15040
           RETURN
 12160
                                                                                                                  15050
 12170
                                                                                                                  T03$ = "X
                                                                                                        15060
 12200
                                                                                                                  T04$ = "X S T O R E XX"
T05$ = " 0 OV 0 C "
           IF SP% = MEM%+1%
THEN ERRM$ = "CAN'T RETURN FROM SUBROUTINE":
                                                                                                        15070
 12210
                                                                                                        15080
                                                                                                                  T06$ = SPACE$(56)
                     Z9$ = FNPRINT$(ERRM$)
                                                                                                        15090
                     Z9% = FNLINEFEED%
E9% = 1% :
                                                                                                                  T07$ = SPACE$(24)
                                                                                                        15095
                                                                                                                  T08$ = SPACE$(15)
PRINT T01$ + T00$ " PC
PRINT T06$ + T02$
                                           : GO TO 12260
                                                                                                        15097
                                                                                                                                                                          " T02$
                                                                                                                                                     " TOO$ " MAR
                                                                                                        15100
15110
           GOSUB 2100
 12220
                                           : GOSUB 3000
           PC% = MDR%
 12240
                                                                                                                  PRINT T01$ + T00$ " IR
                                                                                                                                                     " T07$ + T02$
                                           : GOSUB 3800
                                                                                                        15120
           SP% = SP% + 1%
 12250
                                                                                                                  PRINT TO6$ + TO3$
PRINT TO1$ + TO0$ " ACC
                                                                                                        15130
15140
 12260
           RETURN
                                                                                                                                                     " TO7$ + TO4$
 12270
                                                                MULS
                                                                                                        15150
                                                                                                                   PRINT T06$ + T03$
 12300
                                                                                                                  PRINT T01$ + T00$ " R
PRINT T06$ + T02$
                                                                                                                                                     " T07$ + T02$
                                                                                                        15160
           GOSUB 12100
 12310
                                                                                                        15170
            \GOSUB 2100
                                                                                                                                                                          " T02$
                                                                                                                  PRINT T01$ + T05$ "
                                                                                                                                                     " T00$ " MDR
                                                                                                        15180
            \GOSUB 2400
                                                                                                        15190
                                                                                                                   PRINT
            \GOSUB 2600
                                                                                                                   PRINT TO8$ + TO0$ " SP
                                                                                                                                                     " T00$ " X"
                                                                                                        15192
            \RETURN
                                                                                                                  PRINT : PRINT : PRINT
PRINT "EXECUTIVE MODE"
                                                                                                        15194
                                                                DIVS
                                                                                                        15198
 12400
                                                                                                                  PRIMAR = "NONAME"

PROMAME = "NONAME"

OPP$ = 1$ : CHP$ = 0$

Z9$ = FNCURSOR$(OPP$ + 16$, 0$)

MASK$(110$) = 2$^110$ FOR 110$ = 0$ TO 15$

READ M8$(110$) FOR 110$ = 1$ TO NOMREF$
                                                                                                        15200
           GOSUB 12100
 12410
                                                                                                        15210
15215
            \GOSUB 2100
            T20% = MDR%
                                                                                                         15220
            MDR% = ACC%
            \ACC% = T20%
                                                                                                        15230
                                                                                                                   DATA LDA, STO, ADD, SUB, MUL, DIV, AND, OR, XOR, JMP, JANZ, JALZ
            GOSUB 2500
                                                                                                        15240
                                                                                                                   DATA INCH, JSR, CAHE, LDX, STOX, LDPC, STPC, JAP, JAZ, JXNZ, JXLZ DATA JXP, JXZ, JOV, JNOV, JSI, CXHE READ M9$(110%) FOR 110% = 0% TO NOREG%
                                                                                                        15250
            GOSUB 2600
                                                                                                        15255
            \RETURN
                                                                                                                  READ MYS(LIUS) FUR TIUS = US TO NUMBUS
DATA STOP, CLA, INC, DEC, ADDS, SUBS, SWAB, HOP, HOPE, HOPN
DATA HOPL, HOPE, NOT, NEG, INCX, DECX, SWAX, SWAR, ASL, ASR
DATA ROR, ROL, CLV, SEV, HOPV, CLC, SEC, HOPC, CCV, SCV
DATA SWAP, PUSH, POP, RTS, MULS, DIVS, PACC, RACC, PSTR, RSTR, NLIN
                                                                                                        15260
                                                                PACC
                                                                                                        15270
 12500
                                                                                                         15280
            Z9% = FNPRINT%(ACC%)
 12510
                                                                                                         15290
            RETURN
 12520
                                                                                                        15300
 12530
                                                                                                                                                                       INITIALISE ALL REGISTERS
                                                                 RACC
                                                                                                        15400
  12600
            Z9% = FNCURSOR%(OPP% + 16%, CHP%)
IF CHP% > 56% THEN Z9% = FNLINEFEED%
                                                                                                                   PC%, MAR%, IR%, ACC%, R%, MDR%, OV%, C%, X% = 0%
                                                                                                         15410
                                                                                                                   12610
                                                                                                         15420
  12620
                                           : GOSUB 3300
                                                                                                         15990
            INPUT ACC%
  12630
                                                                                                                   ON ERROR GO TO 32000
            Z9% = FNLINEFEED%
                                                                                                         16000
  12640
                                                                                                                   INPUT LINE M$

OPP% = OPP% + 1%
                                                                                                         16010
            RETURN
  12650
                                                                                                         16012
  12660
                                                                 PSTR
                                                                                                                   IF OPP% > 6%
THEN OPP% = 1%
                                                                                                         16014
  12700
            ERRM$ = ""
  12710
                                                                                                                   Z9\% = FNCURSOR\%(OPP\% + 16\%, 0\%)
            T20$ = FNCVT$(ST%(PC%))
                                                                                                         16016
  12720
            PC% = PC% + 1%
T20% = INSTR(1%, T20$, "&")
                                                                                                                   PRINT ELINE$;
                                                                                                         16018
  12730
                                                                                                                   M$ = LEFT (M$, LEN(M$) - 2$)
                                                                                                         16020
  12740
            IF T20% = 1% THEN 12780
IF T20% = 2%
                                                                                                                   L9% = LEN(M$)
M1$ = LEFT(M$,3%)
                                                                                                         16022
  12750
                                                                                                         16024
  12760
                                                                                                                   IF M1$ = "BYE" THEN 16150

IF M1$ = "RUN" THEN GOSUB 17000: GO TO 16000

IF M1$ = "DUM" THEN GOSUB 18000: GO TO 16010

IF M1$ = "SAV" THEN GOSUB 19000: GO TO 16010
                THEN ERRM$ = ERRM$ + LEFT(T20$, 1%)
                                                                                                         16030
                                                                                                         16040
               ELSE
                      ERRM$ = ERRM$ + T20$
                                                                                                         16050
            IF T20% = 0% THEN 12720
Z9$ = FNPRINT$(ERRM$)
PRINT "" : GOSUB
                                                                                                         16060
  12770
                                                                                                                   IF M1$ = "OLD" THEN GOSUB 20000: GO TO 16010
IF M1$ = "APP" THEN GOSUB 20030: GO TO 16010
IF M1$ = "UNS" THEN GOSUB 21000: GO TO 16010
                                                                                                         16070
  12780
                                : GOSUB 3000
                                                                                                         16075
  12790
                                                                                                         16080
            RETURN
  12795
                                                                 RSTR
                                                                                                                    IF M1$ = "CAT" THEN GOSUB 22000: GO TO 16010
                                                                                                         16090
  12800
                                                                                                                   M1$ = LEFT(M$, ADL%)
M1% = VAL(M1$)
            Z9% = FNCURSOR%(OPP%+16%, CHP%)
                                                                                                         16100
  12810
             \IF CHP% > 56% THEN Z9% = FNLINEFEED%
                                                                                                         16110
                                                                                                                   IF M1% > -1% AND M1% <= OMEM% AND LEN(M$) > ADL% THEN GOSUB 23000 : GO TO 16010 ERRM$ = M$ + " IS NOT A VALID COMMAND!"
                                                                                                         16120
  12820
            INPUT ACC$
\ACC% = CVT$%(ACC$)
                                            : GOSUB 3300
                                                                                                         16130
                                                                                                                    GOSUB 4200
             \Z9% = FNLINEFEED%
                                                                                                         16140
16145
                                                                                                                    GO TO 16010
             \RETURN
                                                                                                                    GO TO 32760
                                                                                                          16150
                                                                                                                    NLIN
                                                                                                         16990
  12900
                                                                                                                    PC% = 0%
   12910
            Z9%=FNLINEFEED%
                                                                                                         17000
                                                                                                                    IF L9% = 3%
   12920
            RETURN
                                                                                                         17010
                                                                                                                                                                         ! RUN FROM 0%
                                                                                                                        THEN 17100
   12930
```

```
IF L9% <> 8%
17020
            THEN ERRM$ = "INVALID RUN COMMAND"
                 Z9$ = FNPRINT$(ERRM$)
Z9% = FNLINEFEED%
                 GO TO 17770
        PC% = FNOCT%(RIGHT(M$,5%),ADL%)
17030
17040
        IF PC% = -1%
THEN 17700
                                                     !NON OCTAL ADDRESS
                                                     ! PRINT PC%
17050
         GOSUB 3000
         ON ERROR GO TO 32300
17100
                                                     ! SET CTRL/C TRAP
17105
         Z9 = SYS(CHR (6%) + CHR (-7%))
        Z9% = FNCURSOR%(14%,0%)
PRINT "RUN MODE - INSTRUCTION BEING OPERATED ON:- ";
17120
         PRINT ELINE$ FOR I% = 1% TO 9%
17130
         OPP% = 1% : CHP% = 0%
17140
                                      : GOSUB 3800
         SP% = MEM%+1%
17150
         E9% = 0%
17160
17190
17200
         IF E9% <> 0%
            THEN E9% = 0%
                                   : GO TO 17700
                                    : GOSUB 3100
17205
         MAR% = PC%
         MDR% = ST%(MAR%)
                                   : GOSUB 3700
17210
                                    : GOSUB 3200
17220
         IR% = MDR%
         PC% = PC% + 1%
                                    : GOSUB 3000
17230
17240
                                            SEPARATE OC, AD AND
17250
                                                     ! DISPLAY INSTRUCTIONS.
17260
         GOSUB 4100
17265
         IF OC% = 0%
17270
                                                      ! REGISTER INSTRUCTION
            THEN 17380
17280
         IF ( IR% AND MASK%(10%) )
17290
                                                      ! INDEX MODIFICATION
            THEN AD% = AD% + X%
17300
         IF ( IR% AND MASK%(9%) )
17310
                                                      ! INDIRECT ADDRESSING
17320
17330
17340
         ON OC% GOSUB 5000,5100,5200,5300,5400,5500,
17350
                        5600,5700,5800,5900,6000,6100,
6200,6300,6400,6500,6600,
                        6700,6800,6900,7000,7100,7200,
                                                            ! PERFORM INSTRUCTION
                        7300,7400,7500,7600,7700,7800
         GO TO 17260 IF OC% = JSI%
                                                      1 END OF CYCLE
         GO TO 17200
17360
17370
17380
         IF AD% = 0%
                                                      ! "STOP" INSTRUCTION
             THEN 17700
         ON AD% GOSUB 9000,9100,9200,9300,9400,9500,
17390
                       9600,9700,9800,9900,10000,10100,
10200,10300,10400,10500,10600,10700
                       10800, 10900, 11000, 11100, 11200, 11300,
                       11400,11500,11600,11700,11800,11900,
                       12000, 12100, 12200, 12300, 12400, 12500, 12600, 12700,
                                                      ! PERFORM REGISTER
                       12800,12900
                                                        INSTRUCTION
                                                      ! END OF CYCLE
17400
         GO TO 17200
 17410
         Z9\% = FNCURSOR\%(14\%,0\%)
 17700
         PRINT ELINE$;
 17710
          PRINT "EXECUTIVE MODE"
 17720
         IF CHP% <> 0% THEN Z9% = FNLINEFEED%
 17730
17740
          Z9\% = FNCURSOR\%(OPP\% + 16\%, 0\%)
 17760
 17770
 17780
          17990
                                                      DUM
 18000
 18001
         T11% = 0%
\T12% = MEM%
          \IF MID(M$,8%,1%) <> "-" THEN
                   IF LEN(M$) > 3% THEN

KB% = VAL(MID(M$,5%,2%))
                   ELSE
                            KB% = -1%
          ELSE
                   T11% = FNOCT%(MID(M$,5%,3%),ADL%)
                   \T12$ = FNOCT$(MID(M$,9$,3$),ADL$)
\IF LEN(M$) > 11$ THEN
                            KB\% = VAL(MID(M\$, 13\%, 2\%))
                           KB\% = -1\%
          IF KB% > -1% THEN GO TO 18100
 18002
          ELSE
                   PRINT "PROGRAM " + PRGNAM$ + ".SIM"
                   \PRINT
          ERRM$ = "ADDRESS
                                   CONTENTS"
 18003
          GOSUB 4200
 18004
          ERRM$ =
 18005
                                                                     ASSEMBLER"
                                                      BINARY
                           DECIMAL
          "OCTAL
 18006
          GOSUB 4200
          FOR I10% = T11% TO T12%
 18009
         IF ST%(I10%) = 0% THEN 18050
T10% = ST%(I10%)
 18010
 18020
         PRINT FNOCT$(I10%,ADL%), T10%,
"""FNCVT$(T10%)"" "FNP1$(T10%)" "FNP1$(T10%);
 18030
```



HIGH DISCOUNT
CATALOG
OF
NEW & USED
TEC



SYSTEMS • PROCESSORS • PERIPHERALS

Including

- * VAX * PDP11
- ★ LSI-11 ★ PDP8
- ⋆ DECsystem 10 & 20
- ⋆ Data General, too

NO-RISK TRIPLE WARRANTY

1. Genuine DEC and Data General Equipment

2. 90-Day Warranty on Parts and Labor

NEW!

10-DAY RETURN PRIVILEGE. Full credit for any fullywarrantied DEC or Data General item returned within 10 days after receipt . . . and WE'LL PAY RETURN FREIGHT!

Call or write for your copy TODAY!

SC22RS



FREE

Computer Dealers & Lessors Association $- \\ RSTSPROFESSIONALRSTSPROF$

```
21000
        Z9% = FNLINEFEED%
                                                                                             IF L9% > 10%
                                                                                     21010
                                                                                                THEN ERRM$ = "INVALID UNSAVE COMMAND"
18050
        NEXT I10%
        PRINT "STACK POINTER " SP%;
18060
                                                                                                      Z9$ = FNPRINT$(ERRM$)
        Z9% = FNLINEFEED%
18070
                                                                                                      Z9% = FNLINEFEED%
        GO TO 18210
1**** OUTPUT TO ANOTHER KB
18080
                                                                                                      GO TO 21080
18100
                                                                                             T10$ = RIGHT(M$,5%)
T10$ = T10$ + ".SIM"
        !***** CHECK FOR KB NO WITHIN RANGE ****
                                                                                     21020
18110
                                                                                     21030
        Z8$ = "KB" + NUM1$(KB%) + ":"
OPEN Z8$ FOR OUTPUT AS FILE #1%
18120
                                                                                             KILL T10$
                                                                                     21040
                                                                                              ERRM$ = T10$ + " SUCCESSFULLY UNSAVED."
18130
        PRINT #1%, "PROGRAM " + PRGNAM$ + ".SIM"
\PRINT #1%
                                                                                     21050
18135
                                                                                              GOSUB 4200
                                                                                     21060
                                                                                     21080
                                                                                              RETURN
        PRINT #1%, "ADDRESS
                                      CONTENTS "
18140
                                                                                     21090
                                                                                              18145
        PRINT #1%,
                                                                                     21990
                                                                  ASSEMBLER"
                          DECIMAL
                                        CHAR
                                                  BINARY
         "OCTAL
                                                                                                                                         CATALOGUE
                                                                                     21995
        FOR I10% = T11% TO T12%
18150
                                                                                              MAT YS = ZER
                                                                                     22000
        IF ST%(I10%) = 0% THEN 18180
18160
                                                                                              Y\%(23\%) = 83\%
        T10% = ST%(I10%)
18165
                                                                                              Y\%(24\%) = 89\%
        PRINT #1%, FNOCT$(I10%, ADL%), T10%,
"""FNCVT$(T10%) "" "FNP1$(T10%) " "FNP1$(T10%)
18170
                                                                                              Y% = 0%
                                                                                              Y$ = 09

Y$(0$) = 30$ :

Y$(1$) = 6$ :

Y$(2$) = 15$ :

Y$(4$) = Y$/256$:
                                                                                     22010
         NEYT TIOS
        PRINT #1%, "STACK POINTER" SP%
18190
         CLOSE #1%
18200
                                                                                              Y$(3$) = Y$ - Y$(4$) * 256$
CHANGE Y$ TO Y$ :
CHANGE SYS(Y$) TO Y$
         RETURN
18210
18220
         22020
18990
                                                    SAVE FILE
                                                                                              E$ = RAD$(Y$(11$) + SWAP$(Y$(12$)))

IF E$ <> "SIM" THEN 22040

ERRM$ = RAD$(Y$(7$)+SWAP$(Y$(8$)))+RAD$(Y$(9$)+SWAP$(Y$(10$)))
19000
         IF L9% > 10%
19010
                                                                                      22025
            THEN ERRM$ = "INVALID SAVE COMMAND" :
                                                                                     22030
                 Z9$ = FNPRINT$(ERRM$) :
                                                                                              ERRM$ = ERRM$ + "." + E$
                                                                                      22032
                 Z9% = FNLINEFEED%
                                                                                               GOSUB 4200
                                                                                      22034
                                                                                              GO TO 19100
                                                                                      22040
         T10$ = RIGHT(M$,5%)
19020
                                                                                      22050
         IF T10$ = ""
19022
                                                                                      22090
                                                                                                                                          STORES INPUT INSTRUCTION
            THEN T10$ = PRGNAM$
                                                                                      23000
                                                                                                                                          OR DATA
                                                                                      23010
         T10$ = T10$ + ".SIM"
                                                                                                                                           ! OP CODE
19024
         OPEN T10$+"<40>" FOR OUTPUT AS FILE 1%
                                                                                      23050
                                                                                                                                          ! INDEX BIT
 19025
                                                                                               INDEX% = 0%
                                                                                      23060
         T10% = 0%
FOR I10% = 0% TO SP% - 1%
                                                                                                                                          ! INDIRECT BIT
                                                                                               INDIR% = 0%
                                                                                      23070
                                                                                                                                           ! ADDRESS
 19030
                                                                                               AD% = 0%
                                                                                      23080
                                                                                                                                           ! INSTRUCTION
         IF ST%(I10%) = 0% THEN 19070
                                                                                               IN% = 0%
 19040
                                                                                      23090
         GOSUB 4400
 19050
                                                                                      23100
         LSET T12$ = CVT%$(I10%) :
                                                                                                                                             1 STORE ADDRESS
                                                                                               STADD% = FNOCT%(M1$, ADL%)
                                                                                      23110
         LSET T13$ = CVT$$(ST$(I10$))
IF T10$ = 256$
THEN PUT #1$ :
                                                                                               IF STADD% < 0%
                                                                                      23120
                                                                                               THEN FLAG# = 1% : GO TO 23760
IF MID(M$, SPPOS%, 1%) <> " "
THEN FLAG# = 2% : GO TO 23760
19060
                                                                                      23140
                  T10% = 0%
         NEXT I10%
                                                                                               T10$ = MID(M$, MNEMPOS$, 1$)
IF T10$ = '"'
 19070
                                                                                      23160
         GOSUB 4400
 19080
                                                                                               IF T10$ = '"
THEN 23800
                                                                                      23170
         LSET T12$ = CVT%$(9999%)
                                                                                                                                           I LITERAL CHARACTER
          PUT #1%
                                                                                               IF T10$ = "#"
THEN IF L9$ = OCTLEN$
THEN 23940
                                                                                      23180
          CLOSE #1%
 19090
         ERRM$ = T10$ + " SUCCESSFULLY SAVED."
 19092
          GOSUB 4200
                                                                                                        ELSE FLAG$ = 14% : GO TO 23760
 19094
          PRGNAM$ = LEFT(T10$, INSTR(1%, T10$, ".")-1%)
 19098
                                                                                      23190
                                                                                               IF T10$ < "A"
                                                                                                                                           ! DECIMAL INTEGER?
          RETURN
                                                                                                   THEN 23900
 19110
          IF T10$ > "Z"
                                                                                      23210
                                                                                                                                           ! INVALID MNEMONIC
                                                                                                   THEN FLAG# = 5% : GO TC 23760
 19990
                                                     OLD/ APPEND
                                                                                               IF LEN(M$) <= REGLEN%
 20000
                                                                                       23230
                                                                                                                                           ! REGISTER INSTRUCTION
         MAT ST% = ZER
 20010
                                                                                                   THEN 23340
          ST%(0%) = 0%
                                                                                                T10% = INSTR(MNEMPOS%,M$," ")
 20020
                                                                                       23250
          IF L9% < 5%
 20030
                                                                                               IF T10% = 0%
                                                                                       23260
             THEN IF M1$ = "OLD"
THEN GO TO 20140
                                                                                                   THEN FLAG# = 6% : GO TO 23760
                                                                                                                                  I MEMORY REFERENCE
                                                                                       23270
                                                                                                T10$ = MID(M$,MNEMPOS$,T10%-MNEMPOS$)
                                                                                       23280
                        ERRM$ = "INVALID APPEND COMMAND.":
                                                                                                FOR I10% = 1% TO NOMREF%
                                                                                       23290
                        GOSUB 4200:
                                                                                                IF T10$ = M8$(I10%)
THEN OC% = I10% : GO TO 23420
                                                                                       23300
                                                                                                                                        ! VALIDATE ADDRESS
                        GO TO 20140
         IF L9$ > 17$
THEN ERRM$ = "INVALID " + M1$ + " COMMAND." :
 20040
                                                                                                NEXT I10%
FLAG% = 5% : GO TO 23760
                                                                                       23320
                                                                                       23330
                   GOSUB 4200:
                                                                                                                                            REGISTER INSTRUCTION
                                                                                       23340
                   GO TO 20140
                                                                                                T10$ = MID (M$,MNEMPOS%,LEN(M$) - SPPOS%)
                                                                                       23350
                                                                                                FOR I10% = 0% TO NOREG% IF T10$ = M9$(I10%)
 20050
          T10$ = RIGHT(M$,5%)
                                                                                       23360
          IF M1$ = "OLD"
 20060
                                                                                       23370
                                                                                                                                            !INSTRUCTION COMPLETE
             THEN PRGNAM$ = T10$
                                                                                                   THEN AD% = I10% : GO TO 23650
                                                                                                NEXT I10%
                                                                                       23390
          T10$ = T10$ + ".SIM"
  20070
                                                                                                FLAG% = 5% : GO TO 23760
          OPEN T10$ FOR INPUT AS FILE 1%
                                                                                       23400
                                                                                                                                            VALIDATE ADDRESS
  20080
                                                      ! POSS ERR 5
                                                                                        23410
                                                                                                !
IF LEN(M$) < T10% + ADL%
THEN FLAG% = 7% : GO TO 23760
T10% = MID(M$,T10%+1%,ADL%)
AD% = FNOCT%(T10%,ADL%)
  20085
                                                                                       23420
  20087
  20090
           T10% = 0%
                                                                                        23440
           GOSUB 4400
  20100
                                                                                        23450
           T11\% = CVT$\%(T12$)
                                                                                                IF AD% < 0%
                                                                                        23460
          IF T11% = 9999% THEN 20130
ST%(T11%) = CVT$%(T13$):
  20110
                                                                                                    THEN FLAG% = 7% : GO TO 23760
  20120
                                                                                        23480
                   IF T10% = 256%
                                                                                                IF LEN(M$) < T10% + ADL1%
                                                                                        23490
                      THEN 20087
                                                                                                                                            INSTRUCTION COMPLETE
                                                                                                    THEN 23650
                     ELSE
                                                                                                T10$ = MID(M$,T10$+ADL1$,1$)
IF T10$ = "X"
THEN INDEX$ = 1$
                                                                                        23510
                           20100
                                                                                        23520
           CLOSE #1%
  20130
           IF M1$ = "OLD"
  20132
                                                                                                   ELSE IF T10$ = "I"
THEN INDIR% = 1%
              THEN ERRM$ = T10$ + " SUCCESSFULLY LOADED."
                                                                                                           ELSE FLAG% = 8% : GO TO 23760
                   ERRM$ = T10$ + " SUCCESSFULLY APPENDED."
                                                                                        23570 IF LEN(M$) < T10% + ADL2%
                                                                                                                                            INSTRUCTION COMPLETE
  20134
                                                                                                    THEN 23650
                                                                                                T10$ = MID(M$,T10$+ADL2$,1$)
IF T10$ = "I"
           RETURN
  20140
                                                                                        23590
  20150
                                                                                        23600
           -
-
-
  20990
```

```
THEN INDIR% = 1%
        ELSE FLAG# = 8% : GO TO 23760

IF L9% > T10% + ADL2%

THEN FLAG# = 10% : GO TO 23760
23630
23650
                                                     ! LOAD OP CODE
23660
         IN% = FNLOC%(OC%)
         IF INDEX% = 1%
23670
            THEN IN% = FNLX%(IN%)
                                                     ! LOAD INDEX BIT
         IF INDIR% = 1%
THEN IN% = FNLIND%(IN%)
23690
                                                      ! LOAD INDIRECT BIT
         IN% = FNLAD%(AD%,IN%)
                                                      ! LOAD ADDRESS
23710
23720
                                                     ! STORE INSTRUCTION
23730
         ST%(STADD%) = IN%
23740
         GO TO 23990
23750
23760
         GOSUB 4300
23770
         GO TO 23990
                                                     LOAD 2-CHAR LITERAL
23800
         IF L9% <> LITLEN%
23810
            THEN FLAG% = 11% :
GO TO 23760
         IF MID(M$, LITLEN$, 1$) <> CHR$(34$)
23820
                  THEN FLAG% = 12% :
        GO TO 23760
IN% = CVT$%(MID(M$,MNEMPOS%+1%,2%))
23830
         GO TO 23730
23840
23850
                                                      LOAD INTEGER
23900
         IN% = VAL(MID(M$, MNEMPOS%, L9% - SPPOS%))
23910
         GO TO 23730
23920
23930
                                                      LOAD OCTAL NUMBER
23940
         IN% = FNOCT%(RIGHT(M$,OCTPOS%),L9%-(OCTPOS%-1%))
23950
         IF IN% = -1% THEN 23990
23960
         GO TO 23730
23970
23975
23990
          23995
                                                      ERROR TRAPPING - O/S
32000
         IF ERR = 5% THEN 32060
32010
         IF ERR = 27% THEN 32120
32020
         IF ERR = 51 OR ERR = 52 THEN 32180
32030
         IF ERR = 8% THEN 32120
32040
         ON ERROR GO TO O
32050
                                                      ERR 5
32060
         IF ERL = 22020 THEN RESUME 16010
ERRM$ = "CAN'T FIND FILE OR ACCOUNT."
32065
32070
         GOSUB 4200
32080
         IF ERL = 20080 THEN RESUME 20140
IF ERL = 21040 THEN RESUME 21080
32100
         RESUME 16010
32110
32120
                                                      ERR 27
         ERRM$ = "I/O TO DETACHED KEYBOARD?!"
32130
         GOSTIB 4200
32140
         IF ERL = 18130 THEN RESUME 18210
32150
         RESUME
32160
32170
                                                      ERR 51/52
32180
         IF ERL = 740
32190
            THEN FNOCT% = -1% : RESUME 840
         IF ERL = 16110 THEN RESUME 16130
32200
         IF ERL = 18001 THEN
32205
                  PRINT "INVALID 'DUM' COMMAND"
                  \RESUME 18210
         ERRM$ = "ILLEGAL NO."
32210
         GOSUB 4200
 32220
         IF ERL = 18100 THEN RESUME 18210
IF ERL = 23910 THEN RESUME 23990
 32230
32240
32295
         ON ERROR GO TO O
                                                      ERROR TRAP - RUN MODE
 32300
         IF ERR = 28% THEN 32350
32310
32315
         IF ERR=55 THEN 32410
         IF ERR = 58% THEN 32410
IF ERR = 61% THEN 32470
32320
 32330
                                                      ERR 28
32350
         ERRM$ = "RUN TERMINATED"
32360
          GOSUB 4200
32380
         E9% = 1%
         RESUME 32767
32390
                                                      ERR 55/58
         ERRM$ = "INVALID INSTRUCTION CODE"
32420
32430
         GOSUB 4200
 32440
         RESUME 17200
 32460
                                                      ERR 61
32470
         ERRM$ = "ATTEMPT TO DIVIDE BY ZERO!"
 32480
 32490
         GOSUB 4200
 32500
         E9% = 1%
         RESUME
 32510
 32520
 32530
          ON ERROR GO TO O
32595
          PRINT "Control is being passed from SIMINI to RSTS/E"
32760
          \PRINT "You may now issue any RSTS command"
 32766 NO EXTEND
 32767
         END
```



SPSS® makes data analysis simple for DEC PDP-11 users! Now PDP-11 users can enjoy all the benefits that have made SPSS the world's largest selling Data Analysis System. It's easy to use and learn, thanks to its response to English language commands and comprehensive documentation. It's also sophisticated, giving researchers and business managers alike a full range of capabilities for statistical analysis and report generation.

For full information, call or write SPSS today: Sue Phelan, SPSS, Inc., 444 N. Michigan Avenue, Chicago, IL 60611, 312/329-2400.



© Copyright 1981 SPSS, Inc.

SPSS-11 runs on DEC LSI-11 through PDP-11/70. Compatible with DEC Systems RSTS, RT-11, RSX-11M, IAS/Bell Labs UNIX/S&H Computer Systems TSX.

CIRCLE 123 ON READER CARD



Kangaroo

or . . . two.

See page 75.



DEC

SYSTEMS & COMPONENTS

C.D. SMITH & ASSOCIATES, INC. 12605 E. Freeway, Suite 318 Houston, TX 77015

(713) 578-8464

CIRCLE 54 ON READER CARD

A SECURITY CHECKLIST

By Dave Mallery

Physical: Start outside your computer room. Is the door locked? Is the key hanging there? If there is a window, what can you see from the corridor? Is the console right by the window? Can you read the print-out? Can you watch the operator log in? Can you see the disk drives? Are the packs labeled? Are the labels the MFD passwords?

Now take the 'VIP' tour of the computer room that the boss would give to visiting firemen. Repeat all of the above questions. Here are a few more: How many dial-in phone numbers can you find—look at the telephone blocks and modems. Look at the console closely—how many passwords can you find? Any written on the terminal? Are any of the terminals logged in—(try control 'T'). How quickly can you run \$REACT? (or \$ONLPAT).

After the cook's tour, it's time for some more methodical work. Make a list of every keyboard on your computer and the actual physical location. While doing this, it's a good time to actually look at each of them. How many are logged in and unattended. Are any in a privileged account? How many have passwords written or taped to them. Try the answerback escape sequence on every VT100 and equivalent and see how many contain log-ins. Now that you have a list and are satisfied with the contents, actually open the back of your machine, and physically verify that 'empty' ports have nothing plugged in. If you find a cable in an 'empty' port, follow it. It will lead to something interesting. Perform the same steps for all known telephone circuits and modems in your computer room. While looking at cables and connectors, make sure they only have one wire coming out and that there are no three-connector 'bridges' in sight. Now, take a full listing of all accounts on your system disk (\$MONEY). Sit down and identify the owner of each account. Scan your disks (all of them) for programs with protection codes conveying privilege. Identify each program and its reason for priv. Demand sources of all such programs and re-compile them periodically without notice. A very good check is to log into a non privileged account and look around via DIR at the privileged ones. Even if you have used the feature patch for restricting directory access.

Try this — Run \$RNO RNO > KB: = \$ACCT.SYS

Try every program you can see that has a <232>. Go home some night and spend an hour trying to break in. How many CCL'S work logged-out. How much can you learn about your system before logging in? See if anyone challenges you. See if anyone notices. See if anyone cares.

Print out a list of all passwords. Besides DEMO, SYSLIB, SYSTEM, MFD, PACKO1, and other obvious ringers (your name, the company name and initials...). How many passwords are two or three letters long? The best password is six long, contains at least one numeric digit and makes

very little sense. Most priv. accounts don't need passwords (exception: [1,2]) [1,0] & [1,3] through [1,5] especially don't if used as libraries. Get rid of them.

UT CHANGE [1,n]*

There are different schools of thought on the subject, but I believe only one privileged password is required unless substantial development work is going on in privileged accounts. (If you have one, you have them all...). One password is very simple to change. You should change it frequently. Once in a while, you should change it and wait to see who asks.

There are several other areas of security to consider. Systems that are physically secure may be vulnerable to psychological attack:

The phone rings at 9 PM. Your night operator answers. The caller identifies herself as Digital Software Support and asks if that emergency patch to the disk driver has been installed. Warning of imminent system crash, she has the operator install a patch over the phone that allows logged-out users to have privilege. (A one-word patch, by the way). Do you have a procedure for such probes. Does your personnel know how to validate such phone calls? Do you have a call-back procedure? How do you know that this patch has not already been put in place? Do you know how to find out? Do you have a procedure for departing employees?

Trojan horses are another consideration. A trojan horse is a hidden switch or program entry point in a piece of third-party software which could be used by an informed person to penetrate an otherwise secure system. There is really no defense here. It never hurts to require third-party vendors to certify that their software contains no such traps.

Finally, here's a list of things to remember when dealing with a system that has been penetrated.

If your system has been penetrated, consult both law enforcement authorities and DEC security, before you do anything.

All monitor SIL's, RUN-TIME SYSTEMS, LIBRARIES and all privileged programs must be re-generated and replaced. You must have a known secure distribution. Get a new one if there is the slightest possibility of compromise. Have the sysgen and patching done by an outside source.

Remember to isolate and save the suspect media. They may be useful as evidence.

In the end, the only really secure system is one that has been powered down. We would be very negligent in our duty to our employers if we neglected the obvious.

December 1982 Page 75

DECUS AUSTRALIA 1982

By Carl B. Marbach

Winter in July, day into night, the Southern Cross and water that goes down the drain the wrong way: Australia! Here we were in Sydney, New South Wales on our way to Melbourne, Victoria and the 1982 Australian DECUS meetings.

It was the middle of July, winter here, but this is the land of mild winters; it never snows in the cities and seldom goes below freezing. The days are a mixture of sun and clouds, with mild temperatures often reaching the upper 50's or 60's even this far south (it gets colder when you go south, away from the equator). Our daytime was night at home; a 14 hour time difference—not to mention a whole day since we crossed the International Date Line. My wife, Helen, has it all figured out: as you go farther west it keeps getting earlier and earlier until suddenly it is tomorrow! At least she understands.

So, after 21 hours in two different aircraft (5 hours to San Francisco via DC-8 and then 6 hours to Honolulu and 10 hours to Sydney in a Quantas 747) we were here. Hats off to Quantas, they run a fine airline; we ought to know after 16 hours straight with them. Friendly helpful people foreshadowing our experiences of the next two weeks in beautiful Australia.

The winter (!) sky contains some very unfamiliar constellations, and we wondered at all the ancients following the legendary Southern Cross. We can now confirm that high pressure rotates counter clockwise and water does go down the drain in a vortex the opposite way it does north of the equator.

Although Australia is about the same size as the U.S. most of its population live in coastal cities. Sydney is a major city on the southeastern coast of the continent. The downtown area is a mixture of old and new high rise office buildings, and the city is clustered into a neat center near the famous Sydney Harbour and Opera House. Culture abounds and the Opera



Up a tree Down under.

House lives up to its reputation with a complete opera hall, concert hall and drama theatre each specialized for the art to be presented. We toured the facility and had a lovely dinner overlooking the harbour. The scenic harbour is full of sailboats and other craft, including a hydrofoil commuter boat to other areas across a dramatic entrance to the sea.

Outside Sydney there are bedroom communities which give way to farms and rolling plains leading to the Blue Mountains, so named because of their blue color caused by the gas escaping from the eucalyptus trees lining the sides of the hills. The gorges, valleys and spectacular panoramas is an amazing sight. Sydney, which was now more than 60 miles away was still visible! I should not fail to mention the snowstorm that we rode into and out of going through one of the many high roads in the mountains, and in July

We promised to visit one of our hosts, Chris Brett of Uncle Ben's of Australia, on our way to Melbourne so we took an early flight to the twin cities, Albury/Wodonga. The twin engine Fokker was comfortable and unlike in the States, the captain was happy to have us visit the cockpit where he

could point out the scenery of the interior of the country. Landing in Albury/Wodonga is like homecoming: lots of families greeting returnees and the same saving goodbye, very rural and friendly. We were taken to the plant where Uncle Ben's manufactures pet food. The office space uses an open approach; everyone sits at his desk in a large open room (even the general manager) and when private conferences are needed you simply schedule one of the many such rooms located around the perimeter. We met with the programmers, managers and plant personnel for business and lunch which was followed by a plant tour. The day was crisp and clear, not easily forgotten, and a real look at Australia at work. After dinner in town we got the late flight to Melbourne.

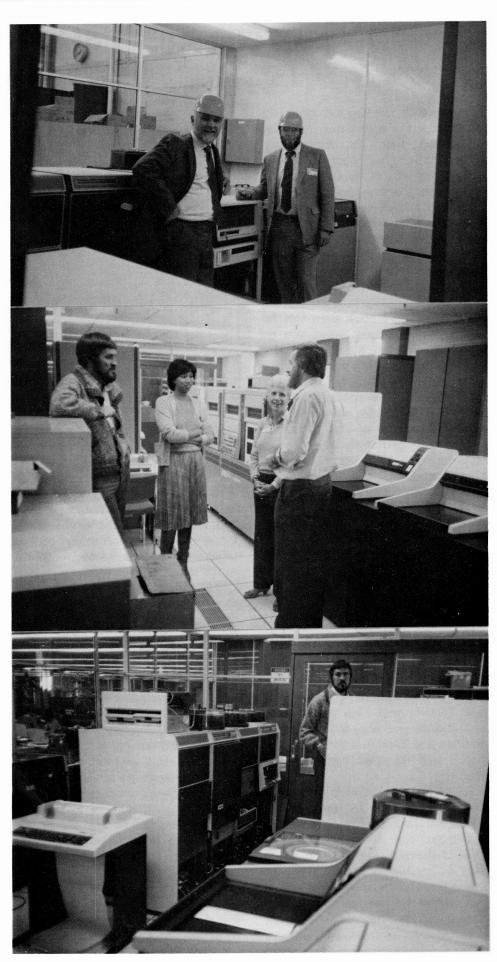
Melbourne appears as an older city, spread out with lots of minor cities (suburbs?) surrounding the center. It appears much larger in size than Sydney, but comparable in most other things. Which is the capitol? Neither, since they couldn't decide between the two cities, Canberra was chosen. Canberra is located almost exactly halfway between the two cities.

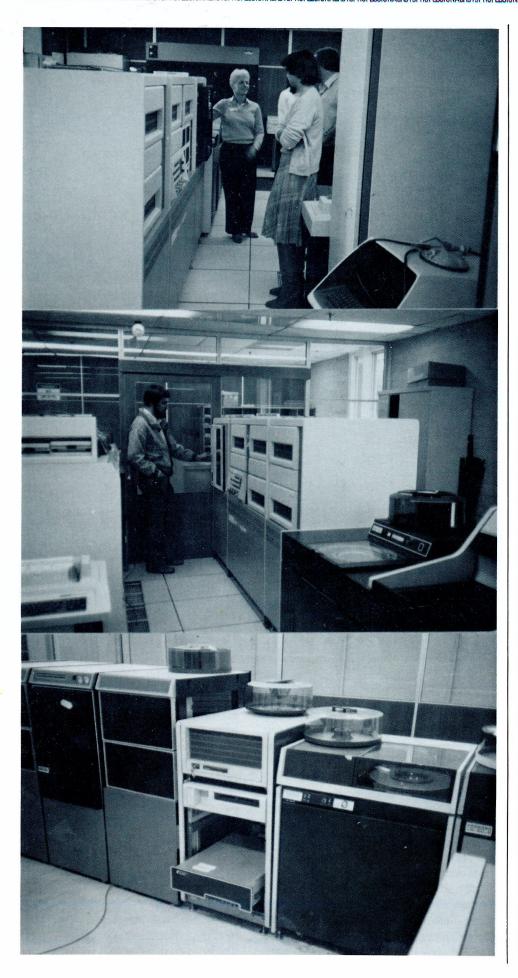
The Melbourne Hilton hosted this DECUS and is located beside two parks. One can ride the public transportation in clean, well lit friendly conditions. At rush hour they sell tickets at the bus stops so the driver can just pull up and load without long lines waiting to pay. Melbourne has a harbour and sights to rival any city in the U.S. To us it resembled a city on the move in the 1940's—all the good and none of the evil that the States' major cities have come upon in the 50's, 60's and 70's. There was no grafitti, no areas that were dangerous to walk around in and the people were very friendly and interested in us and our country. We even enjoyed watching Good Morning Australia when we woke up in the morning (evening at home). We make it a point to visit several special places

... to page 78

Visiting the Computer Room at Uncle Ben's of Australia and their 11/44 & 11/70

(Can you find the VTO5?)





Now a DEC/IBM a DEC/IBM inter-connect connect that's cost that's cost effective

computer has more important things to do than be a processor for your IBM communications. Save valuable computing capacity by handling this interconnect workload with COMBOARDTM.

To your operators, COMBOARD is a reliable package that doesn't decrease the number of on-liner users.

To your users, COMBOARD is a simple link to IBM systems of job and data transfer.

To your management, COMBOARD is a cost-effective solution to a troublesome problem.

COMBOARD is a 16 bit CPU based single board computer which plugs straight into your DEC UNIBUS. Then your COMBOARD, not the DEC host, handles all the real time interrupts and protocol processing associated with data communications.

COMBOARD models 631, 731 and 1231 support transfer rates from 4,800 to 56K baud. They are the leaders in DEC/IBM and DEC/CDC interconnects.

For more details contact your sales representative at Software Results at 614 421-2094 or mail the coupon today.

SOFTWARE RESULTS

CORPORATION

1229 West Third Avenue Columbus, OH 43212-3090 Telephone: 614-421 2094

COMBOARD™ Software Results Corporation DEC, UNIBUS™ Digital Equipment Corporation



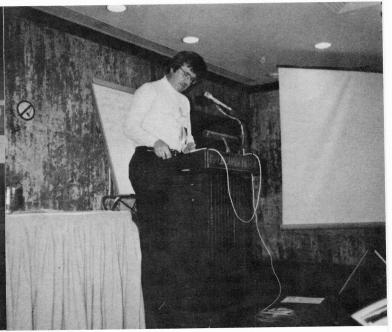
TO MINISTER OF THE PROPERTY OF

The exhibit area.

Chris Brett, next year's symposium RSTS Coordinator addresses the RSTS attendees.



DECUS Library Store.



Jo Kruithof, Newsletter editor, addresses the RSTS Group on "Non DEC Software Packages."

when we visit a city and one is the zoo! I particularly enjoyed the platypussary where we saw several swimming. The wombat was also a highlight as were the wallabees and koala bears.

DECUS symposia are much more relaxed outside the U.S. This may be because of the more realistic attitude adopted by most attendees that they are really THEIR meetings. Subjects are discussed that would not be allowed at the U.S. counterpart. While they also refrain from commercial activities, a session which I liked very much presented non-DEC software for RSTS by

users of several packages available for RSTS. No prices were discussed, but we did manage to ask what it did, how well it did it, how the documentation was, how easy it was to use and what kind of support the vendor provided. In all my years (10) at DECUS this was among the most valuable; and I had to travel quite a distance to hear it! It was a real DECUS. The DECUS store sold souvenirs. DIGITAL had a large exhibit hall and there were meetings going on when I had to be at other meetings! The hallways and lunch tables were great places to meet people and talk DEC. I always learn so much from such exchanges that it is a main reason for attending. For the RSTS group and others I suspect, Paul Konig from DEC fit right in and contributed much about DECNET, TECO and Version 7.2. The DECUS dinner was great fun, followed by some speeches, not all of which were very serious. I heard one story from a doctor in the North about. . well, you'll just have to come to the next DECUS to find out. He does it every year.

After the meetings were successfully concluded we headed North (remember, it gets warmer when you



Computer controlled trains . . . Great for Winter in July.

Question: Who does she work for?





The Australian Museum in Sydney, NSW.

Yes, there will be PDP-11's for the 1980's.

go North) to Cairns on the northeast coast. We were able to visit an Aboriginal village and many hills and valleys in this lovely tropical area. Aside from relaxation, a major draw for this area is the Great Barrier Reef. This collection of reefs and small islands guards the coast of Australia for some 2000 miles and is alleged to be the "largest living thing in the world." The reef lies about 30 miles off the coast so you have to travel to get there. When the wind subsided enough, we were able to take a fast boat, called appropriately enough, Down Under, out to the reef for a day of diving and snorkeling. While I found the visibility at 50 feet to be average (about 50-75 feet) for good diving waters, the abundance and variety of life was amazing. Everywhere you looked there was something alive; giant clams, coral, nudibranches, fish of every size and shape and so much more that we must return and spend more than one day and two dives. While the snorkeling was nice, this is a place to be UNDER the water, a part of the reef itself.

It was finally time to go home, but not without a stopover in Brisbane, a fast growing "sunshine" city about midway on the east coast. Like many sunbelt areas, Brisbane is undergoing rapid growth. The city hustles and bustles with excitement. We need more time to really experience the old and the new that it has to offer.

Too soon we were in the 747 cocoon that hums for about 10 hours and then disgorges you in a different land. It was a tiring but exciting two weeks in Australia. Thank you DECUS and the RSTS group for being such wonderful hosts. As usual we learned much, and we hope left some good ideas with you. So long for a while. We will be back. RSTSPROFESSIONALRSTSPROFESSIONALRSTSPROFE

Maryland Colk Westminster, MD 21157 Can someone who knows more about it than I do please advise on the wisdom of the horac in 22 I have road the role tnan I do please advise on the wisdom of the release advise on the read the release installing RSTS V7.2? I have read accompanished in the release and other information accompanished in the release advise on the wisdom of the release advise on the read the release advise on the wisdom of the release advise on the release advise of the installing RSTS V7.2? I have read the release V7.2 installing RSTS V7.2? I have read the release vinter and other information accompanying V7.2 installing RSTS V7.2? I have read the release vinter and the release vinter and the release vinter and the release vinter and the release vinter notes and other mountained accompanying y and frankly didn't see anything significantly and frankly didn't are anything of moortand and frankly didn't are anything of moortand different from 1774 or anything different from and trankly didn't see anything significantly and trankly didn't see anything of importance to different from V7.1, or anything or the rather leaves me to different system environment. This rather leaves our system environment. anrerent from V1.1, or anything or importance to our system environment. This rather leaves me to our system environment. This rather leaves me to der what the point of VI.4 is. The fact that the directions say the sysgen The fact that the directions had the transfer Wonder what the point of VT2 is. Ine Tact that the directions say the sysgen or line tact that the directions say the sysgen can't be done on line, and our need to transfer can't be done on line, and to monthly not think and line accounts not to monthly not the sysgen of t cant be done on-line, and our need to transfer on t be done on-line, and to mention patching makes the 800 user accounts, not to mention makes the hoth nec-ineriral and in-house BUU user accounts, not to mention patching makes the both DEC-inspired and in-house war impattractive I decrease war in the contractive I decrea both DEC-inspired and in-house — makes the prospects of this exercise very unattractive. I do prospects of this exercise there is a mond reason and want to inetallit unlace there is a mond reason. prospects of this exercise very unattractive. I do prospe Emily G. Johnston. why I should.

Following closely on the heels of V7.1 is V7.2. Many people are asking why it took several YEARS to get from 7.0 to 7.1 and only a few MONTHS to get to V7.2. It appears that RSTS will move away from complicated and long patch kits, toward a system that will feature many more "maintenance" releases. VAX/VMS is released several times a year and this approach may be the way RSTS will go in the future.

With V7.2 there are some new things. Probably the most important, although not for everyone, is the full support of cluster libraries. Cluster libraries allow more than one resident library to share the same starting address; each library can be mapped only one at a time, but the "clustering" allows more program to be mapped within the addressing limit of the PDP-11. Suppose you wanted to map two 8K shared libraries: Without clustering you would need 16K of address space, 8K for each one and that doesn't leave much room for your program let alone RMS or something else. With clustering, each of the 8K libraries can share the same starting address, using only 8K worth of addressing space instead of 16K. By using multiple libraries it is possible to write larger programs than you could without "clustering".

Because of certain restrictions and limitations that each library must meet, not all libraries can be clustered. Currently, only COBOL-81 V1.1 and FMS-11 V1.5 are the only DEC libraries that can be clustered. Look for the third party software vendors to use this feature and maybe even provide the tools for us to "cluster" the non-clusterable libraries.

There are some DCL changes, but none that enhances the offering enough to make it really worthwhile as a standalone. DCL is good to start with because it will be upward

VERSION 7.2 OF RSTS/E NOW AVAILABLE

By Carl Marbach

compatible with other (VAX) systems. Older users don't need it as it doesn't add any functionality to RSTS. On VAX, DCL is a nice way to do lots of things but on RSTS it doesn't seem to be worth it.

There is support in V7.2 for the UDA50 controller and the RA80 disk drive. The UDA50 is the new intelligent (isn't everything?) controller for UNIBUS systems. For a complete discussion of these controllers and drives, see Volume 1 Number 1 of the DEC* PROFESSIONAL. For RSTS expect to see DUO: and DUX: as the new device designator for the UDA50 disks. When DEC will actually start shipping these controllers and the new RA80, RA81 and the RA60 is still a mystery, but it could be anytime.

More 22 bit addressing for the 11/23-PLUS is included in 7.2 allowing the 11/23 to have up to 4MB of memory. The problem with earlier 11/23 systems is that the DMA controllers who have direct access to memory did not support the 22 bit addressing; to do 22 bit addressing you will need the upgraded controllers and an 11/23-PLUS or see some other vendors for how to upgrade your present 11/23.

SYSGENS should be done off-line on a fresh disk. If you know what you are doing it is possible to do them on-line however. Note that some cusps like MAKSIL, TKB and SILUS may change so make sure you use the new versions for your SYSGEN. We have successfully done this type of SYSGEN by copying the distribution medium to a separate disk drive and then after the dialogue phase of SYSGEN, we modify the SYSGEN.CTL file to point to the drive we have the SYSGEN stuff on instead of the SYSTEM disk. By modifying this control file we have complete control over the SYSBAT procedure and wind up with a new SIL on the extra drive that can be COPYied or moved in some fashion along with the new cusps to the system disk and then INstalled.

Send/receive has a few minor changes, with the biggest being the notice that SYS call 18 (small send/receive) will not be supported beyond V7.2.

DECNET V2.0 or DECNET phase III is supported. RMS V1.8 has not changed.

The monitor is now smarter about repositioning things when you add BUFFERS to the monitor. Before you had to manually reposition the memory allocation table.

There are some minor BASIC PLUS proposed changes that you should look at in the release notes even if you are not going to install V7.2. DEC wants you to move to 7.2 and has announced that support for V7.1 will last only 6 months

So there it is, V7.2; should you install it? Some will and some won't, your decision should be based on needs relative to the new features, DEC support and standardization with the rest of the RSTS folks. We won't rush into it, but we'll install it soon.

LATE BREAKING NEWS

Infinity

oftware Corporation

2210 Wilshire Boulevard, Suite 801 Santa Monica, CA 90403 (213) 820-2702

Letters to the Editors RSTS Professional

November 11, 1982

Dear RSTS Pro:

In the RSTS Professional of September 1981 Steven L. Edwards published three patches to the EDT version 2.0 release for RSTS/E version 7.0. Since that time there has been new releases of RSTS/E, and with them a slightly different release of EDT V2.0.

As published in volume 3.3 of the RSTS Pro, all the EDT patches will still work, with the exception of the patch to force VT100's back to VT52 mode on exit from EDT. The correct patch for the new release of EDT has changes to three lines.

The lines that read

IO.CXR = \$\$\$LOC+742 IO.CFR = \$\$\$LOC+766 . = \$\$\$LOC+3360

should now read

IO.CXR = \$\$\$LOC+756 IO.CFR = \$\$\$LOC+1002 . = \$\$\$LOC+3374

For those counting, the difference was increase in the size of the IOMOD module earlier on by 14(octal) bytes. The checksum for the new IOMOD module is 23007, the checksum for the corrected patch file is 21406.

For those desiring to install both the VT100 to VT52 patch from volume 3.3 of the RSTS Pro along with the journal name patch printed in this issue by David Spencer, make the following changes to the patch procedure.

Install the correct Steve Edwards patch, but do not replace it in the library nor task build. Enter the journal name patch file and modify the line that says

. = \$\$\$\$.+003376

to become

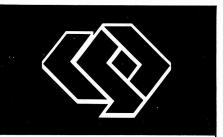
. = \$\$\$.+003466

Run PAT.TSK with the new checksums for IOMOD with the Steve Edwards patch being 16011, and the journal name correction file checksum being 22621. Remember not to select the same output file name for the IOMOD module as the input module, or PAT will fail. Replace the IOMOD module with the two applied patches in the library and task-build normally.

EDT devotees;

Steven L. Edwards Software Techniques, Inc.

David Spencer
Infinity Software Corporation



COBOLSOFTWARE SOLUTIONS FOR PRODUCTION AND FINANCIAL CONTROL

Available on

VAX
in native mode

PDP 11
under RSX or RSTS

OFF-THE-SHELF SUPPLY Enquire about our 30-DAY TRIAL ON YOUR MACHINE

IMPCON

FULL FACILITIES FOR:

General Ledger–
Accounts Receivable–
Accounts Payable– Sales Orders–
Stock Control–Purchasing–
BOM–Works Orders – MRP–
Work in Progress Monitoring

Send for FEATURE WALL CHART to:

COMPUTER SYSTEMS
DEVELOPMENT (CSD) LTD.

75 Marylebone High Street, London W1, England.

Tel: 01-935 2255

NEWS RELEASES



PRODUCT UPDATE

EVANS GRIFFITHS & HART, INC. ANNOUNCED RELEASE OF VERSION 12 OF SELECT

Lexington, MA — Evans Griffiths & Hart, Inc. announced the release of version 12 of SELECT, its fast record extraction package for RSTS/E on the PDP-11. The package can also be used under VMS on the VAX with ROSS/V, EGH's RSTS/E simulation package.

SELECT is a machine-language package that employs user-specified selection criteria to extract from one or more input files those records needed for a specific report or application. Optionally, the extracted records are modified by the deletion, insertion, or rearrangement of fields. The records are then written to an output file that may be sorted by making use of SELECT's interface to FSORT3, EGH's fast machinelanguage sort package.

Version 12 of SELECT contains a new operator that allows the user to generate more than one output record for each input record. Multiple selection expressions control successively whether a given kind of record is output for each input record. This feature is most useful in situations, such as building multiple file indices, where making a single pass over a large input file may save considerable time over making multiple

Another change in version 12 is that SELECT's front-end modules may be compiled under either BASIC-PLUS or BASIC-PLUS-2. Previously, the package's front-end modules could be compiled only under BASIC-PLUS.

A single-CPU license for SELECT is \$1500, discounted to \$500 when the package is purchased with FSORT3. OEM discounts are available. Contact Evans Griffiths & Hart, Inc., 55 Waltham Street, Lexington, MA 02173. Tel.: (617) 861-0670.

OUERY AND REPORTING TOOL INTERFACES WITH MANY PACKAGES New York, NY — Enterprise Technology Corporation has released Version 2.0 of its Generalized Reporting System (GRS), a user-friendly query, report-writing and auditing package designed for use with a wide variety of applications software packages and database systems. GRS operates on Digital Equipment Corporation PDP-11 series computers under RSTS/E. A version of GRS for VAX series computers is scheduled for release in December 1982.

The new version of GRS is compatible with the applications software packages distributed by NCA Corporation, Martin Marietta Data Systems, Transcomm Data

Systems, Interactive Management Systems and AMCOR Computer Corporation, among others. GRS has also been interfaced to the TOTAL and AMBASE database management systems, and to the RMS-11, DMS-500, DIBOL, FMS-11, and FAM file management systems which are used under RSTS/E.

GRS consists of a powerful English-like query language which is suited for use by both DP and non-DP users. The language is designed to be used by management personnel to meet their own needs for ad-hoc information retrieval, and is also intended to be used by systems personnel to vastly decrease the amount of time spent responding to user requests for reports. GRS also includes many features which make it ideal for use by auditors.

The new version contains a CHAIN interface, allowing GRS to be called from other programs and from user menus. Other features in the new release include support for the PDP-11 floating-point processor, support for larger data records than previous versions, an option to utilize record keys to speed access to selected data, and enhanced facilities for the use of "canned" command files. It also includes expanded user documentation.

The GRS query language is designed so that one statement is all that is necessary in many cases to produce a nicely-formatted report, including page numbers and column headings. This is accomplished through the use of a wide range of defaults, all of which can be overridden by the user if desired. GRS provides the flexibility to generate reports in practically any format desired. Summarization statements and an advanced set of built-in calculations allow the user to aggregate data and perform complex analyses, again using a single statement in many cases.

The GRS user can easily supply his own titles, headings and page footers, and several different files, record formats and/or print line formats can be incorporated into a single report. GRS includes an extensive online user Help facility and optional password protection at the individual field level. Users can save sets of commonly-used GRS commands in command files, thus allowing even the most complex report to be produced by typing a single word to GRS. The system includes an easy-to-use editor to aid in preparing and changing these command files.

Additional GRS system features include a high-speed sort facility, multiple control breaks with totaling and subtotaling, weighted averages and other summary calculations, table look-ups, elementary statistical analysis, and random sampling. The user can enter specialized calculations in standard algebraic format, and the results of these calculations can be used to create new, "temporary" database fields or to temporarily redefine existing fields. This redefinition capability is useful in constructing forecasting applications.

GRS contains a facility which allows ETC or a customer's systems staff to tailor GRS to meet the particular requirements

of an installation. For example, GRS can call a variety of database access routines in the preparation of a single report, including access methods which the customer has supplied and incorporated into GRS. Programmers can also code their own conversion modules for inclusion into the system, in order to handle specifically encoded or encrypted fields. To facilitate its use with complex databases, GRS is capable of working with several different record formats at once, determining any given record's format by examining the record itself.

It is not necessary for the user to describe to GRS the fields which comprise the record(s) he is working with; the database description process is performed only once, when GRS is first installed at a customer site. During a GRS session, the user needs only indicate the names of the data fields which are needed for the report he is preparing.

GRS operates on most types of interactive terminals, both CRT and hardcopy. A special Scope mode allows GRS to pause at the bottom of each page, thus facilitating its use on video terminals. Users can optionally direct their output to alternate hard-copy devices or to listing files for later printing.

The GRS software is supplied under a perpetual license for a one-time fee of \$8,500.00 which includes full documentation and one year of support. Extensive training and customization services are also available. More information can be obtained by contacting Enterprise Technology Corporation at 305 Madison Avenue, New York, NY 10165, (212) 972-1860.

COMPANION PRODUCTS FOR DEC COMPATIBLE CT*OS WORD PROCESSING SYSTEM Pasadena, CA — Office Automation

software house, Compu-Tome, Inc., is now offering two additional software products as companions to their DEC compatible CT*OS word processing package.

The first product, CT*OS spelling corrector, works in conjunction with CT*OS word processing and provides very high speed, 80 words per second, checking of spelling in documents several thousand words in length. The corrector, working with a dictionary of over 80K words, actually allows the user to correct misspelled words, rather than merely flagging them as in many less powerful systems. Both the speed and file efficiency are the result of a unique design based on a compressed syntactic format. If the word is not encountered in the dictionary, it is stripped of common prefixes and suffixes and checked for root comparison.

The second product is a powerful management communication system dubbed ExectronicTM Mail. Exectronic Mail will support up to 64K users on a single network and operates under DECNET protocols. Besides the huge number of user notes, Exectronic Mail is unusual in that it is management oriented, designed to facilitate communications among decision making users. Thus, an active response is required of both sender and receiver; i.e., December 1982

RSTSPROFESSIONALRSTSPROFESSIONA

file, hold, acknowledge.

The structure of the system is based on a "routing slip" format which makes it "friendly" to users familiar with the ubiquitous office routing slips.

Exectronic Mail will encourage and aid constructive communication skills by helping the communication cycle progress; for example, the system flags departure of a message, arrival, and finally "reading" by the recipient.

The system, operating with CT*OS as the resident editor, offers generous file sizes and fonts; for example—up to 1K messages per user stored on-line, names and titles up to 53 characters. Storage demand is minimal; 24Kb for the first user, plus 6Kb for each additional user up to system limits.

Response is fast, typically sub-second, making it faster than present DEC standard utilities, such as EDT. Inter CPU transfers are by DECNET.

The corrector and Exectronic Mail operate under RSTS/E, RSX-11M, RSX-11M+ and VAX/VMS and are fully integrable with CT*OS. An important application is the ability to run with DEC's new "300" series personal computers acting as an intelligent node on a network.

Price for the spelling corrector is \$500 for a single CPU license, and base price for Exectronic Mail is \$18,000.

Compu-Tome, Inc., 234 É. Colorado Blvd., Pasadena, CA 91101, (213) 796-9371

NEW PRODUCTS

GEJAC ANNOUNCES ARSAP FOR RSTS Riverdale, MD — The ARSAP Resource Management and Chargeback System, has been updated to run on PDP-11 computers using RSTS versions 7.0, 7.1 and DEC's recently released 7.2 operating system.

ARSAP is a comprehensive computer resource management system which provides users, system managers, corporate and financial staff with reliable reports for making informed, accurate decisions about the data center operation.

ARSAP is now available on all the most widely used DEC operating systems: RSTS, RSX-11M-PLUS and VAX/VMS. Managers of multiple operating system shops can get compatible reporting of resource utilization, making it easy to compare productivity on all in-house DEC systems. This compatibility has not been available before now.

The most significant features of ARSAP are:

- 1. Project Accounting Users can be prompted for a project-id and password upon logon and the terminal session can be allocated to his project as well as his logon account, so it is easy to monitor projects and users.
- 2. Terminal Accounting System utilization reports can be generated for each terminal, showing the activity on each terminal, when the terminal is used and the amount and type of resources consumed. The reports are presented in graphic as well as numeric format.



Compu-Tome, Inc. offers two new packages for their CT*OS word processing system: CT*OS Spelling Corrector and ExectronicTM Mail.

- 3. System Utilization Utilization reports can be generated by shift and for any user-selected date range. Reports include information on CPU usage, disk space, number of logons, connect time, device time, printed lines, and other major resources. These reports and activity profiles show at a glance how the computer is used and which projects or users consume the most resources.
- 4. Invoicing Invoices can be generated by user or by project. Rates can be by time of day and can be set by individual user, project, or system default basis.

ARSAP is used to control day to day operations, to monitor system utilization and growth, to plan the resources needed for current and future workloads, to justify needed hardware and software upgrades, and to prioritize and schedule computer processing demands.

ARSAP also can produce itemized invoices automatically for allocating costs of computer resources to internal departments, projects, users, contracts and grants. It is used for departmental budgeting, to control project costs, or to bill timesharing customers. ARSAP's equitable billing system enables bidding competitively on contracts and satisfies government and commercial job costing requirements.

Multiple copy discounts are available. ARSAP is delivered within two weeks on mag tape, comes with a 30-day acceptance period and a full one year warranty, including maintenance and enhancements services, and complete documentation.

Contact: GEJAC, Inc., P.O. Box 188, Riverdale, MD 20737, 301-864-3706.

VAX TO PDP-11 FILE TRANSFER London, ONTARIO — Cableshare Inc. has announced PORTAL - VMS, designed to alleviate the need to transfer files using magnetic tape. PORTAL performs interactive and automatic file transfers between VMS, RSTS/E and RSX-11M systems. PORTAL-VMS provides a cost-effective system requiring no additional computer hardware as it operates over standard asynchronous terminal interfaces.

PORTAL-VMS may also be used in conjunction with Cableshare's LSI-X.25 interfaces to provide file transfers over private and public packet switched

PORTAL-VMS, PORTAL-RSTS and PORTAL-RSX are part of Cableshare's family of computer communication products.

For more information, call or write: Barry G. Walker, Director of Marketing, P.O. Box 5880, 20 Enterprise Drive, London, Ontario N6A 4L6, (519) 686-2900

SOFTWARE TECHNIQUES INCORPORATED ANNOUNCES NEW UTILITY BILLING SYSTEM

Los Alamitos, CA — At a meeting of the Orange County Water Association today, Software Techniques Inc. announced a new integrated Billing and Accounts Receivable system designed to meet the special needs of water companies. Named WBS-11, the new system is designed for DEC PDP-11 computers running the RSTS/E operating system.

Rich Scherle, President of Software Techniques, explained how the new system protects against data loss by using an advanced magnetic tape journaling system. But, the features that the participants seemed most impressed with were the dual-brightness display, which makes the screens easier to read (reducing operator fatigue), and the new route numbering system which solves several long-standing industry problems.

According to Karl Kemp, Manager for Mesa Consolidated Water District where the new system is installed, "It is a custom-made system. The entire program was designed specifically to meet the particular needs of a water agency. I am excited about this new addition and look forward to what it can help the District accomplish."

The system offers easy access to all functioning through a state-of-the-art menu selection, which also provides extensive security options. "User-friendly" features include the system's ability to accept data in a variety of ways and arrange it into a standard format. In addition the system can access account history by soundex, providing customer information on anyone with a name or street even sounding like the one in question.

WBS-11 is available directly from Software Techniques, Inc. (Software Techniques, Ltd. in the U.K.) and from authorized distributors.

VAX USERS GET SESSION DOCUMENTATION

Provo, UT — Clyde Digital Systems, Inc., a principle supplier of system utilities, announces the release of CATCHIT, the most unique documentation tool available to VAX users. For the first time, VAX users now can achieve accurate and reliable run examples while executing interactive tasks on the system.

Eight control characters make up the easy-to-use yet powerful command set. By running CATCHIT, all terminal input and computer output is captured in the log file. All user input is underlined as well. CATCHIT does not lose terminal prompts, and catches all terminal I/O that comes through.

Use CATCHIT for audit trails, user documentation, or detecting errors when documenting programs. At any time you may insert descriptive text into the log file. This text has no effect on the current session, but is simply placed in the log file. This is particularly valuable for including explanations of what you are doing in the resulting document so that others will be able to read it and do what you did.

Users familiar with DOC, the powerful RSTS counterpart, will immediately recognize the advantage of enhancing their computer system with CATCHIT. The user community will quickly recognize its versatility and attractive pricing. CAT-CHIT may be licensed for a one time fee of \$600.

For more information, call or write LeAnn at: Clyde Digital Systems, 3707 North Canyon Rd./3-E, Provo, UT 84604, (801) 224-5306.

CURRENT SOFTWARE ITEMS AVAILABLE

Columbus, OH

- 1) CCLMAN CCL Manager Enhanced from the June, 1982 RSTS PROFESSIONAL \$15.00.
- 2) ..CB.. CB Communications System Greatly enhanced from the August, 1982 RSTS PROFESSIONAL \$15.00.
- 3) RTS How To Write a Runtime System with Example Runtime System \$15.00.
- 4) CCLMGR Runtime System Interface to

CCLMAN - \$15.00.

5) USRDSK - System Manager user disk sortage report - \$15.00.

Send tape and tape format required ...
If you do not enclose a tape, please
enclose an extra \$10.00 to cover the cost
of the tape.

For more information contact: Philip Hunt, O.L.F.B.P., 6400 E. Broad St., Columbus, OH 43213, (614) 864-9200.

MDB INTRO'S BUS REPEATER FOR LSI-11 USERS

Orange, CA — A Bus Repeater that allows LSI-11 Q-Bus processors to be extended beyond their 20 DC bus load limitation, has been introduced by MDB Systems, Inc., the world's largest independent manufacturer of minicomputer interfaces.

Called the MLSI-DB11-R, it receives and redrives all of the computer bus signals so that any type of devices can be on the expanded bus which supports up to 19 additional DC loads. Developed in response to the trend toward larger systems capability, this new generation Bus Repeater supports 22-bit addressing, 4-level interrupts, Direct Memory Access modules and parity.

Design features include terminators and two dual height modules, one for each bus segment. They plug directly into LSI-11 processor backplane and expansion chassis and provide the required 120 ohm termination for the CPU bus and matching impedence for the beginning of the second backplane.

The MLSI-DB11-R is software transparent and is fully bi-directional as signal direction may change several times for each bus cycle. Program array logic in the MLSI-DB11-R monitors bus activity and issues gating commands to handle address, data, interrupt and DMA control signals. As a result, the CPU is unhindered by any device placed between it and other peripherals or memory in the system. An edge mounted LED indicates when the repeater is active.

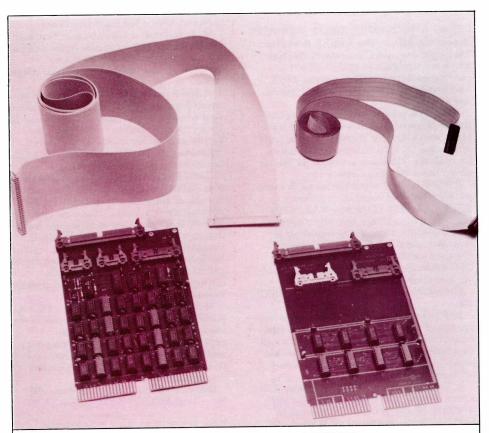
The overall design of MDB's MLSI-DB11-R allows large, single CPU systems to be implemented with one or more bus repeaters, according to MDB marketing manager Stan Margulis, who adds that MDB's new Bus Repeater also provides an excellent method for future system expansion.

The MLSI-DB11-R is priced at \$900. in single quantity and delivery is available within 30 ARO. The price includes a 10 foot (3.05m) interconnection cable set.

NEW SOFTWARE

SYSTEM PERFORMANCE HOUSE ANNOUNCES SOLUTION TO DISK BOUND RSTS/E SYSTEMS Columbus, OH — An easy-to-use solution to disk bound RSTS/E systems has been developed by the System Performance House, Inc. The program, known as DOPTER, produces an optimized disk structure that increases system performance up to 50 percent.

"Our users have found that DOPTER is superior to any other disk management



MLSI-DB11-R BUS REPEATER ASSEMBLY FOR LSI-11 Q-BUS SYSTEMS FROM MDB.

system on the market," noted William R. Davy, president of the System Performance House.

DOPTER automatically performs the following functions: properly places and pre-extends the MFD; places and contiguously pre-extends the UFD's; places the UFD's with the most activity toward the front of the MFD for quickest access; and places the most used files at the front of the UFD's. Also, all files and free space are placed in their optimum positions.

DOPTER has proprietary techniques that optimize the MFD and the UFD's better than REORDR. It also deletes unused file attributes from source, task and object library files, reducing UFD and cache accesses.

Additional features of DOPTER allow all volume, UFD, MFD, and file defaults to be manually overridden. Also, all accounting data is preserved, as well as previously gathered bad block information contained in the output volume. DOPTER automatically initializes the output volume, hooks it, and installs the SIL.

Safeguards that prevent the user from accidentally copying an older DOPTER'd volume onto a newer one, are also included.

A minimum RSTS/E operating system with at least two disks is required to use this optimization program.

Further information about DOPTER can be obtained by writing the System Performance House, Inc. at 5522 Loch More Court, Dublin, Ohio 43017, or by calling (614) 265-7788. The System Performance House specializes in the development of technically superior utility software for DEC computers.

RESPOND PROVIDES PERFORMANCE ANALYSIS AND TUNING AIDS FOR RSTS/E SYSTEMS

Seattle, WA - Manus Services Corporation, a Seattle Authorized Digital Computer Distributor, has introduced a software package to evaluate performance on DEC RSTS/E Systems. The package, called RESPOND, provides system managers with what is needed to effectively solve performance problems: facts about what is happening within the computer system. RESPOND identifies critical resources and highlights areas where tuning can make the computer more productive. RESPOND tells the system manager whether to adjust system-level parameters, optimize selected application programs, or purchase additional hardware.

RESPOND reports important RSTS/E tuning statistics to the system manager so that system-level parameters may be adjusted. RESPOND shows the results of those adjustments immediately and quantitatively so that further adjustments can be made if needed.

According to Manus, buying hardware is not always the way to solve a performance problem. Buying hardware to solve a tuning problem will only result in a tuning problem on a more expensive computer system. RESPOND's performance measurement tells the computer manager where to best apply his resources. It tells whether more hardware is needed, and what kind of hardware will

truly address the system's bottleneck.

RESPOND also captures a history of activity on the system. It knows who did what and from which keyboard. It tells what percentage of key resources were used by each account. It ranks programs by resources used so the manager knows exactly what resource jobs are resource intensive.

RESPOND's introductory license fee is \$1,800 for a single-CPU. Discounts are available for multiple-CPU licenses.

Manus Services Corp., Lake Union Bldg., 1700 Westlake Avenue North, Seattle, WA 98109, 206/285-3260.

NOTES MEETING NEWS

IRUS '82 REPORT

Warwick, RI — IRUS (Independent RSTS Users Society) members from around the country gathered in Boston this year for their annual conference, which was held October 4-6 at The Boston Park Plaza Hotel. The schedule consisted of three days of technical presentations, management seminars, panel discussions, question and answer sessions, and was highlighted by an Exposition of DECcompatible software, hardware and related services. IRUS, the 3-year old independent organization of RSTS, VAX and other DEC installations, holds an annual conference in addition to other ongoing services for its members in order to meet its chartered goal of providing "a forum for the free exchange of information and ideas."

This year's conference theme was 'Options for the 80's," and topics included an overview of DEC operating systems, an introduction to VAX concepts, programming techniques with RMS-11K, project management, microprocessor applications, DECNET, system security under RSTS, an introduction to UNIX, source librarians, managing software development, program generators, the task builder, data communications basics, and a look at what's new in software and hardware.

"People come to an IRUS conference to learn about not just the hardware and software that they are currently using, but also about other options available to them, and even more important, what's coming in the future," explains Monica Collins, IRUS Chairperson. "That's why we convene leaders in the DEC-compatible industry and ask them all the same questions: what's coming? what are the trends? It's these kinds of discussions that make our conferences unique and invaluable to the user."

At the Exposition, which was also open to DEC users who are not IRUS members, attendees could learn about many of the products and services that are compatible with their systems. Exhibitors included Able Computer, Accudata of Maine, Advanced Computer Systems, Amcor Computer Corp., Aviv Corp., Ball Systems, Braegen Minicomputer Peripherals, Cambridge Digital Systems, Data Processing Design, DTI Data Terminals, EDP Supply, Emulex Corp., Evans, Griffiths & Hart, General Electric Co., Hardcopy

Magazine, Information & System Research, International Systems, MDB Systems, M Systems, Monosson on DEC, New England Digital Systems, Orbis, Racal-Vadic, Ross Systems, System Industries, Timplex, Tymshare Computer Systems and Support and Tymshare, Inc.

IRUS serves its members with a monthly newsletter, product surveys, and holds regular regional meetings in New England, Southern California and Metropolitan New York. The next national conference is scheduled for the Fall of 1983. For further information contact IRUS, 3657 Post Road, Suite 4, Warwick, RI 02886 (401) 738-4430.

MEETING REPORT

SOFTWARE CENTER PROVED TO BE

New York, NY — With the largest assemblage of business-oriented software ever gathered under one roof and a computerized on-line search system for specific programs in operation for all four days, the Software Center proved to be the hit of the Ninth International Information Management Exposition & Conference: INFO 82, at the New York Coliseum in October.

A record-breaking display of hardware and software filled the first three floors while the Software Center and conference rooms occupied the fourth floor to permit INFO 82 to fill all four floors of the convention center for the first time.

Attendance totaled 40,250, an increase of an astonishing 32% at a time when virtually all shows are showing sharp decreases in attendance. Exhibits, too, were substantially greater and occupied 25% more net exhibit space than in 1981.

Eighteen thousand specially-printed directories of software exhibited at the exposition were requested by visitors and 8,000 requests for extracts of the Software Center's data base were filled.

Most requests for guidance to specific software were for the traditional areas of finance, accounting and manufacturing applications, but there were many unusual requests, as well.

Among these were a request for a program that would play back graphically an audio recording of a person with a lisp; a program that would compute winter degree days, and a program that would create and display company organization charts.

One indication of the growing importance of software to the business community is the fact that 41% of those seeking computerized guidance to software had not yet purchased the hardware on which the programs were to be used.

Thirty per cent of those requesting computer searches were seeking software for micro-computers. Of these, the most frequently sought programs were for accounting, graphics, financial planning and data base management.

Of those who already owned their hardware but were seeking software for new applications, the greatest number, 41%, were owners of IBM equipment. The

second largest group, 13% were owners of Apple computers, followed by Digital owners who accounted for 12% of the requests.

During the exposition, exhibitors attended a drawing at which space for INFO 83 was assigned. Seventy per cent of those exhibiting this year immediately reserved exhibit space for 1983 and 25% increased their exhibit area. The exhibit area thus assigned is already larger than the entire area occupied at INFO 82.

'Software News' published the software directory, operated the Software Center and provided the on-line software search in conjunction with National CSS.

Visitors planning to attend INFO 83 may obtain information from Clapp & Poliak, 708 Third Ave., New York, NY 10017. (212) 661-8410.

PERSONNEL INFORMATION EMULEX PROMOTES TWO AS VICE PRESIDENTS

Santa Ana, CA — The Board of Directors of Emulex Corporation has elected two of the company's top managers as vice presidents of the company.

James F. Martin has been promoted to the new post of Vice President of International Sales and Charles M. Backer has been promoted to the new position of Vice President of Operations. In their new assignments, Martin reports to Sol Zechter, vice president of marketing, and Backer reports to Malcolm K. Green, executive vice president and chief financial officer.

Martin joined Emulex in 1979 as director of international marketing with responsibility for all sales and marketing activities outside the United States. He had come to the company from the post of vice president for international marketing for mini-computer technology in Palo Alto, CA. Prior to the job he worked in the international sales field for Quantor Corporation, now part of NCR Corporation, and had held a variety of marketing and sales management positions with Image Systems Corp. and Eastman Kodak Co.

Backer joined the company in 1980 as director of operations, responsible for all manufacturing operations, test equipment and industrial engineering, purchasing, facilities management and production and inventory control. He had come to the company from the post of vice president of operations for the Computer Division of Genisco Technology Corporation. Prior to that he held a number of operations management positions with Terminal Data Corporation, Compucorp, Information International, Inc., and Korad, Inc.

Emulex Corporation, headquartered in Santa Ana, is a leading developer and manufacturer of controllers for use in the connection of data storage and communications peripheral devices to computers made by Digital Equipment Corporation. Emulex reported record sales of more than \$16 million for the fiscal year ended June 27, 1982.

Send news releases to: RSTS News Releases, P.O. Box 361, Ft. Washington, PA 19034-0361.

LASSIF

Send Classified Ads to: RSTS Classified, P.O. Box 361, Ft. Washington, PA 19034-0361. \$100 per word, first 12 words free with one year's subscription. [Be sure to include a phone number or address in you message.1

DEC **BEST VALUES**

PRE-OWNED DEC EQUIPMENT

BUYING AND SELLING SYSTEMS • CPU's • PERIPHERALS • TERMINALS OPTIONS • MEMORY • COMPATIBLES

CALL DICK BAKER (305) 979-2844



2845 NW 62nd Street Ft. Lauderdale, Florida 33309 10019169 | H. Lauderuure, Franco 200 15 | Telephone (305) 979-2844

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







Increase your PDP 11's Performance 30-50%!

Supercharged Application Programs are Command Data's enhanced version of proven software which turns out more work, responds faster and extends the life of your PDP 11 investment.

Systems for: General Businesses, Contractors and Distributors. Fully integrated and ready for use

AP • AR • Equipment and Asset Accounting • GL • Job Cost • PR and Order Processing.





165 West Valley Avenue Birmingham, AL 35209 205-942-3156

RSTS WRITER B2 RMS \$99 mo.

ACT 4 603-347 BAY STREET TORONTO ONT. M5H 2R7

BACK ISSUE OFFER

ALL 14 BACK ISSUES OF THE RSTS PROFESSIONAL \$95.00

Send check to: RSTS PROFESSIONAL, Box 361, Ft. Washington, Pa. 19034-0361 Payment Must Accompany Order —

\$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—542-7008



5¹¹101

\$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

\$5.00 RSTS/E HOUR, NO ADDITIONAL CHARGES EXCEPT \$15.00 PER MB PER MONTH AVERAGE (\$250.00 UNLIMITED USE AND LOWER DISK RATE USAGE.)

Development or application work accepted. Programmers set your customers up on our computers, you handle the software we'll handle hardware and Backup. Also available, A/R, A/P, G/L, Payroll, and various specialty applications. Programmers are also available to write Custom Software.

TRI-TEC INC. 50 West 36th Street New York, NY 10018 (212) 736-6010

Accounting Software

Tired of fooling around with accounting applications? In need of effective, timely financial information? Bring your accounting problems under control fast! Call PLYCOM for software that is easy to use, yet extremely effective. Gives you the tools to quickly zero in on your accounting problems. Complete support and training. Excellent documentation. Specifically designed for PDP-11's using RSTS/E or CTS-500. Includes:

- Accounts Payable
- General Ledger
- · Financial Reporting
- Payroll
- Accounts Receivable
- Fixed Assets Reporting
- Time Analysis
- Financial Modeling

Ply C O M s services, inc P.O. Box 160 Plymouth, IN 46563 (219) 935-5121

PUBLISHING SYSTEMS from Libra Computer Systems, 92 High Street, Billericay, Essex, England.

STARVING computer science student needs work programming Cobol, Fortran, Basic. Kronberg 415-386-3237.

RSTS Timesharing Chicago. Saber Computer Services, Inc. Call (312) 998-5950.

PATCH TO \$SYSTAT — Lists Jobs in Keyboard Order, (Just think, pseudo keyboards always first on the list). \$16.00. Patchman, 12625 S.W. Tarpan, Beaverton 97005.

LIST OF ADVERTISERS

ADOS (ADVANCED DIGITAL OFFICE SYSTEMS) p.59 BOB "MACRO MAN" MEYERS p.56 BRAEGEN MPD p.21 CALIFORNIA SYSTEMS ASSOCIATES p.17 C. D. SMITH & ASSOCIATES p.73 C. ITOH ELECTRONICS, INC p.11 CLYDE DIGITAL SYSTEMS, INC p.2 COMPUTER METHODS CORPORATION p.56 COMPUTER SYSTEMS DEVELOPMENT (CSD) LTD p.81 COMPU-TOME, INC p.25 DATA PROCESSING DESIGN, INC B. Cover DATAWARE, INC p.58 DATARAM CORP p.88 DCXX SOFTWARE SERVICES p.20 DIGITAL EQUIPMENT CORP p.9 DIGITEC SOFTWARE DESIGN p.64 DISTRIBUTED LOGIC CORP p.43		
ASSOCIATES p.17 C. D. SMITH & ASSOCIATES p.73 C. ITOH ELECTRONICS, INC. p.11 CLYDE DIGITAL SYSTEMS, INC. p.2 COMPUTER METHODS CORPORATION p.56 COMPUTER SYSTEMS DEVELOPMENT (CSD) LTD. p.81 COMPU-TOME, INC. p.25 DATA PROCESSING DESIGN, INC. B. Cover DATAWARE, INC. p.58 DATARAM CORP. p.88 DCXX SOFTWARE SERVICES p.20 DIGITAL EQUIPMENT CORP. p.9 DIGITEC SOFTWARE DESIGN p.64	OFFICE SYSTEMS)	p.56
CORPORATION p.56 COMPUTER SYSTEMS DEVELOPMENT (CSD) LTD. p.81 COMPU-TOME, INC. p.25 DATA PROCESSING DESIGN, INC. B. Cover DATAWARE, INC. p.58 DATARAM CORP. p.88 DCXX SOFTWARE SERVICES p.20 DIGITAL EQUIPMENT CORP. p.9 DIGITEC SOFTWARE DESIGN p.64	ASSOCIATES	p.73 p.11
DESIGN, INC. B. Cover DATAWARE, INC. p.58 DATARAM CORP. p.88 DCXX SOFTWARE SERVICES p.20 DIGITAL EQUIPMENT CORP. p.9 DIGITEC SOFTWARE DESIGN p.64	CORPORATION	p.81
DISTRIBUTED LOGIC CORF p.45	DATA PROCESSING DESIGN, INC	Dover p.58 p.88 p.20 p.9 p.64

EMULEX CORPORATION I.B. (Cover
ENTERPRISE TECHNOLOGY	
CORPORATION	p.31
EVANS, GRIFFITHS &	
HART, INC	p.65
FINAR SYSTEMS LIMITED	p.60
INFINITY SOFTWARE	
CORPORATION	p.23
MANUS SERVICES CORP pp.3	and the second
MARTIN MARIETTA DATA	
SYSTEMS	p.45
McHUGH, FREEMAN &	Γ
ASSOCIATES, INC	p.39
MINITAB PROJECT	•
MTI SYSTEMS CORPORATION	p.46
M SYSTEMS CONFORMION	p.55
NATIONWIDE DATA DIALOG	p.53
NCA CORPORATION	p.15
NCSS (NORTH COUNTY COMPUTER	
SERVICES INC.) I.F. Cover	,p.44
NEWMAN COMPUTER	
EXCHANGE, INC	p.71

ON TRACK SYSTEMS, INC pp.1 OREGON SOFTWARE PIONEER COMPUTER SYSTEMS	
LIMITED	n 21
	E
RELIANCE ELECTRIC	
ROSS SYSTEMS, INC	
RTZ COMPUTER SERVICES LTD	p.35
SOFTWARE RESULTS	
CORPORATION	p.77
SOFTWARE TECHNIQUES, INC	
SPSS, INC	
SYSTEM PERFORMANCE	
HOUSE, INC.	p.5
TRANSNET CORP	
UNITRONIX CORPORATION	-
	p.27
	p.21
WESTERN RESOURCE	
	p.57
XOREN COMPUTING LTD	p.19

Dataram goes to extremes to satisfy LSI-11 users.

	DATARAM B23 PLUS	PDP-11/23 PLUS	DATARAM M23	PDP-11/24	DATARAM W23	ation.
CHASSIS	5½" Front loading	51/4" Front loading	5¼" Rear loading	51/4" Side loading or 101/2" Top loading	10½" Top loading with 80 MB Winchester	DEC, LSI-11, PDP and UNIBUS are registered trademarks of Digital Equipment Corporation. Q-MAP is a trademark of Dataram Corporation.
BACKPLANE	8 x 4	9 x 4	9 x 6	9 x 6	9 x 6	s of Digital Equ
I/O MAPPING	None	None	Q-MAP	KT24	Q-MAP	ered trademark: ation.
BUS	22-bit Q-BUS	22-bit Q-BUS	22-bit Q-BUS and 18-bit Q-BUS	22-bit extended UNIBUS (EUB) & 18-bit UNIBUS	22-bit Q-BUS and 18-bit Q-BUS	BUS are regist
MEMORY & PERIPHERAL SUPPORT Up to 4.0MB, but limited to 256KB when using peripherals other than RL01/RL02 or RX01			Up to 4.0MB. No basic limitation on peripherals.			, PDP and UNI
PRICING	B23, M23, and W23 are approximately 35% less than comparable DEC systems in 1.0MB configurations. Additional 1.0MB memory modules from Dataram are about 40% less than DEC's equivalent 1.0MB memory contact us for actual price comparisons.					DEC, LSI-11

From Q-BUS Pricing to UNIBUS Performance



From top to bottom, chart the range of LSI-11 system performance you get from Dataram...and only Dataram.

It starts with our low-end B23 PLUS Q-BUS system at lower (much lower!) than DEC prices. And continues with high-end M23 and W23 Q-BUS systems that use Dataram's innovative Q-MAP™ I/O mapping module to generate a separate 18-bit bus from the LSI-11/23's 22-bit bus. Allowing you to put 4.0MB of memory on the 22-bit bus while interfacing your high-performance peripherals (RM02, TM11, RX02 and more) to the 18-bit bus. Giving you much more performance than provided by the PDP-11/23 PLUS, which supports only the RL01/RL02 and RX01 on its 22-bit bus

It's possible, because Dataram's Q-MAP duplicates the functions of DEC's KT24 — which exists only in UNIBUS minicomputers from DEC — enabling Dataram's 4.0MB M23 and W23 minicomputers to be completely compatible with RSX11-M, RSX11-M PLUS, RSTS, UNIX and any other operating system which supports KT24 memory management.

Our W23 system goes a step further, integrating an 80MB Winchester drive and associated controller to provide even greater performance.

Q-BUS pricing...UNIBUS performance...in an LSI-11 megabyte system. Plus a wide range of disk and tape controllers, and related LSI-11 accessories. Only from Dataram. Call us at (609) 799-0071. We'd like to tell you more about our family of LSI-11/23 based systems.



Princeton Road Cranbury, New Jersey 08512 (609) 799-0071 Telex: 510-685-2542

CIRCLE 55 ON READER CARD

EMULEX TALKS DEC

THE DISK WHIRLS...

Can your VAX 750's or 780's benefit from the high transfer rates of DEC's UDA-50? Emulex thinks not. Do your 750's or 780's need another Unibus adapter when the UDA-50 is used? Emulex thinks so. These, and other UDA-50 questions, are discussed in a special Emulex report. Write to us for your free copy.

MAINLINE TO MEMORY ...

Did you know you can mainline directly between VAX-11/750 memory and almost any of the new high performance disks? The Emulex SC750 controller board plugs right into the CMI bus—bypassing the overworked Unibus. Or, if you want to enhance the performance of your VAX-11/780, plug an Emulex V-Master/780 into the SBI bus and mix and match up to eight high performance disks with individual capacities of up to 675 megabytes.

Note: The SC750 and V-Master/780 are the only controllers available today that'll handle the

Fuiitsu Eagle's 1.8 megabyte transfer rate.

BREAKING THE BARRIERS...

Frustrated over the cost and complexity of adding on-line terminals to your LSI-11? Is the multi-user potential of the LSI-11/23-plus just a tantalizing promise—thwarted by cycle-stealing, interrupt-driven DLV11's and DZV11's? Break the barriers with Emulex's new CS01/H communications multiplexer, a Q-bus multi-line interface with DH11 performance capabilities. Up to 64 (that's right, 64) channels attached to a single quad board, plus high-speed transfers. And features that even DEC's DH11 can't equal, like internal self-tests and transmission speeds up to 19,200 baud.

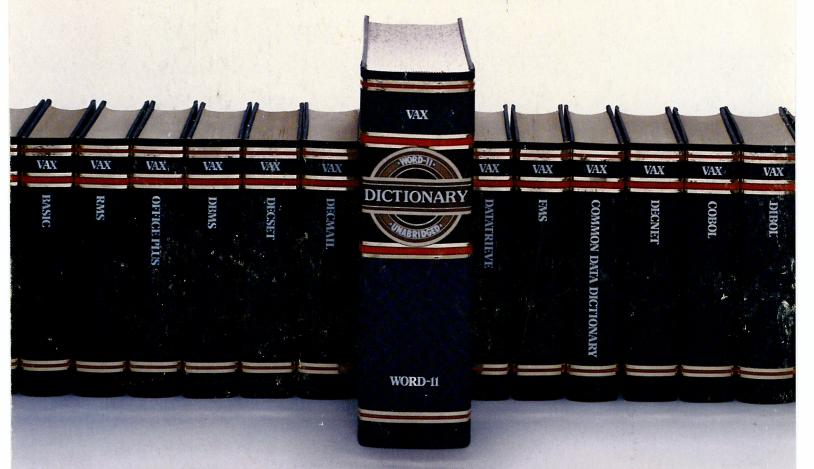
EMULEX SPELLS RELIEF ...

Attention LSI-11, PDP-11, and VAX-11 users who feel bottled up by their present data storage capabilities. You can take advantage of those powerful new disk systems you've been coveting. Over 38 ready-to-ship disk subsystems you can simply plug in and run.

FROM THE EMULEX FILE...

It's been a 60 percent growth year for Emulex; details are in our annual report, available from the Company. New domestic and foreign sales offices. One hundred percent increase in foreign shipments. Construction is nearing completion on a new 71,000-square-foot headquarters building in Costa Mesa, California. Plus, Emulex's 11,500-square-foot plant in Dorado, Puerto Rico, is on line. Meanwhile, write to us at 2001 East Deere Avenue, Santa Ana, California 92705. Or, better yet, let's talk DEC—toll free. Dial (800) 854-7112 or (in California) (714) 557-7580.

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-I1.™ 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-II is also available for Digital's RSX-IIM, $^{\text{M}}$ RSX-IIM PLUS $^{\text{TM}}$ and RSTS/ E^{TM} operating systems.



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

WORD-II is a trademark of Data Processing Design, Inc.

VAX,RSX-IIM, RSX-IIM PLUS, and RSTS/E are trademarks, and remaining titles are products or trademarks of Digital Equipment Corporation.