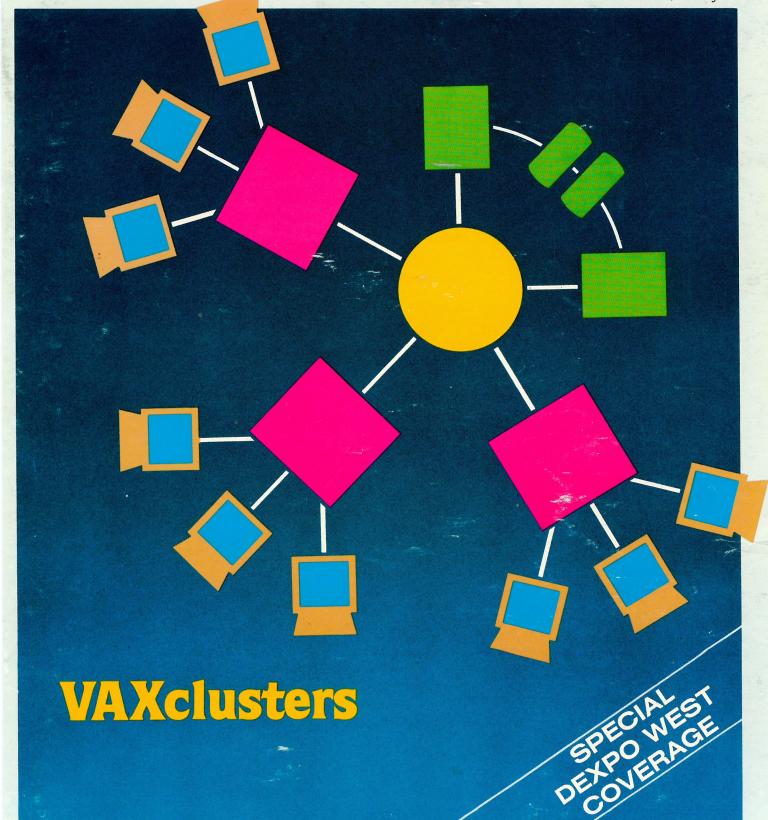
VAX PROFESSIONAL RSTS

Volume 5, Number 5

October 1983

\$1000/issue, \$3500/year



NOW AWAILABLE FOR VAX

Visit us at

DEXPO® West 83

Las Vegas Convention Center October 23-26, 1983

Booth

318

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

SER-11 is a high-performance database system.

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

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

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

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

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

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



North County Computer Services, Inc. 2235 Meyers Ave., Escondido, California 92025 (619) 745-6006, Telex: 182773

*USER-11 is currently available for DEC computers using the RSTS and VMS operating systems.

©NCCS 1983

CIRCLE 30 ON READER CARD

Introducing

What is MAPS/GL?

MAPS/GL is a comprehensive general ledger/financial management system designed for DEC's VAX series of minicomputers. General ledger accounting, financial control, budgeting and advanced reporting functions are incorporated in MAPS/GL, and integrated with decision support to form a state-of-the-art financial management software system.

Standard general ledger functions are only the beginning. MAPS/GL includes many advanced features: a powerful, user-friendly report writer; extensive audit trails; three levels of security; capabilities for custom allocations and complex consolidations; and more.

All the Flexibility You'll Ever Need You can tailor MAPS/GL to your individual needs, controlling accounting structures, organizational roll-ups and

reporting specifications. As your company grows and your needs change, you can change your system to adapt.

Its powerful but easy-to-use report writer brings flexibility to your daily needs as well. Ad hoc reporting from MAPS/GL's financial database can solve your special information needs.

Truly Interactive Processing

MAPS/GL was designed for today's powerful, interactive minicomputersnot adapted from an out-of-date mainframe-oriented batch system. Inquiry, updating and reporting functions can be performed online, right now. So your accounting staff becomes more productive. And your information much more accessible to management.

Integrated with Decision Support Best of all, MAPS/GL can be linked with Ross' easy-to-use packages for financial modeling and business graphics, MAPS/MODEL and MAPS/GRAPH. Together these packages form an integrated financial management system, including microcomputer-based distributed computing power with our MAPS/PRO package.

More MAPS from Ross Systems The MAPS™ (Management, Accounting, and Planning Software) family includes MAPS/AP for accounts payable, MAPS/DB for database management, and MAPS/ISO for employee stock option tracking. All Ross products are available for license or via ROSS/NET™, our VAX-based remote computing service.

And our commitment includes customer and product support that reflects 11 years of consulting experience.

MAPS from Ross Systems. The single source for integrated, interactive financial management software.

DEC and VAX are trademarks of Digital Equipment Corporation. MAPS and ROSS/NET are trademarks of Ross Systems, Inc.

MAPS/GL was developed by Price Waterhouse under the name "FM80" and adapted for the VAX by Ross Systems, Inc. Our license includes the support necessary to keep MAPS/GL current with new FM80 releases. Price Waterhouse markets and supports FM80 on computers other than VAX.

from Ross Systems Financial Software Solutions

DISTRICT OFFICES: Palo Alto, CA San Francisco, CA Los Angeles, CA Dallas, TX New York, NY

CIRCLE 1 ON READER CARD

ROSS SYSTEMS CORPORATE HEADQUARTERS

1860 Embarcadero Road Palo Alto, CA 94303 (415) 856-1100

Smooth transition — the activity required to change — is often as crucial as the change itself. **DISC** recognizes how pivotal the process of evolution can be in business computing applications. **DBL**, available for the range of DEC's PDP-11 and VAX computers, reflects this concern in one of its major design considerations: **Portability.**

DBL portability means a broad capacity to transport your existing software as you outgrow your hardware. It means ease of installation, swift modification of files and a concentration on development — not maintenance. The cost and complexity of the upgrading procedure is greatly diminished by virtue of DBL's inherent upward compatibility.

Smooth passage in an environment of perpetual change: The **DISC** commitment to the DEC user . . . through **DBL**.



Contents

VAX CLUSTERS — Multisystem Technology for VAX Computers	. 8
Carl B. Marbach Multiple CPUs, attached to multiple disk controllers, with the ability to add more of each at an	у
time is the wave of the future in VAXclusters.	
SHAREABLE IMAGES II or Sharing Images for Fun & Profit	12
Last issue we showed you how to build them; now here is what you can do with them. [Al Cir	ni O
teaches the "Structured Methods for BASIC Version 2 Programmers" and the "BASIC Version Update Seminar" for Digital's Seminar Programs. Consult your Seminar Programs Catalog for	≤ or
details, or call (617) 276-4949]	
PATCHING PIP FOR PROTECTION	18
Philip G. Anthony	
Delete all your files? Zero your account? By accident? Not any more!	20
USING THE RSTS/E HIDDEN DATA FEATURE	20
Version 8.0 has some new features that are very interesting. They didn't tell us about this one in the	е
announcement.	
START: A User Control Run-Time System	23
Put the user where you want him, not where he wants to be.	
IBM's 1403 PRINTER REVITALIZED FOR VAX	32
Roger E. Holmes	
The classic printer now can be used for a !classic! computer. MAKE YOUR PDP 11/34 WORK HARDER	34
Austin Kinsella	٠.
The continuing story of ENABLEs on PDP-11s. This is a success.	
USING THE BATCH PROCESSOR TO REORDER DIRECTORIES OVERNIGHT	36
Michael Mowat Keeping the directories clean and reordered is something we all should do more often. Now the	16
cyctem can do it for you	
TU-58 TAPE DRIVE	38
Dave Goodman	
knew there was something someone could do with those. DCL SUBROUTINES FOR VAX/VMS	40
Philip C. Anthony	
DCL (under VAX/VMS) is an important tool. Leave it to the users to figure out how to really use	t.
TALKING WITH OPSER	46
Steven L. Edwards Using a standard interface to this DEC program can help you communicate with all your program	ıs.
ATPK MODIFIED	52
Gary W. Wong	
Our untiring effort to improve ATPK continues.	
YACC IS ALIVE AND WELL AND RUNNING IN RSTS	. 56
James B. Wilkinson Some UNIX utilities under RSTS. Maybe there are more of these that can be used.	
PSTS LINDER WRAPS —	
How To Keep The Users From Knowing They Are Using RSTS	62
Carl B. Marbach Security is just one reason to control your users. It can also make their job simpler and less err	or
prone. Or you may wish just to make them think that a HAL 9000 is running the show.	
MICROCOMPUTER TO RSTS COMMUNICATION	64
Gerard Kiernan	
Want to move programs from that micro to your mini? PIP is only a partial answer and had sor problems. Here is one way that works.	ie
•	69
DYNPRI.MAC Jeff Corbett	OS
Lowering CPU bound job's priority and raising interactive job's priority can bring cheers from use	ΓS.
This is an updated and improved version of the original BASIC-PLUS version now in MACRO.	
SEARCH.TEC	74
Robert G. Smith	
Changing ALL the programs in a system is tedious. TECO can make the job much easier.	
VMS DIRECTORIES: Some Navigational Aids	76
Al Cini Now you can PUSH down one directory level and POP back up.	
Now you can't don't down one directory to related to the age.	

Coming

- Security for Interactive Computer Systems
- DEXPO and DECUS Reviews
- Transferring Files Between RSTS and UNIX
- KIM.

 Program to Transfer

 Files To and From

 TU-58 Cartridge Tapes
- New PDP11 and VAX Computers for 1984
- VAX and Videotex at Time, Inc.
- RSTS Version 8 Reports
- VAX/RSTS to Rainbow 100 Communications
- The Price of Sloth/EDT
- System Stats in Assembler
- **■** BRIGHT
- More . . .

Departments

From the Publishers	4
From the VAX Editor	4
"Pro" Talk (letters to the editor)	6
Tips and Techniques 4	6
Dear VAX/RSTS Man (questions & answers) 8	32
News Releases 8	3
Classifieds 8	37
List of Advertisers 8	88

The VAX/RSTS Professional Magazine, October 1, 1983, Vol.5, No.5, ISSN 0746-1909 is published bi-monthly by M Systems, Inc., 161 E. Hunting Park Avenue, Philadelphia, PA 19124. Subscriptions: single copy price \$10°°, \$35°° per year: \$50 Canada and 1 st class. All other countries, air mail, \$60 US. Second Class postage paid at Philadelphia, PA. POSTMASTER: Send all correspondence and address changes to: VAX/RSTS Professional, P.O. Box 361, Ft. Washington, Pa. 19034-0361, telephone (215) 542-7008. COPYRIGHT © 1983 by M Systems, Inc. All rights reserved. No part of this publication may be reproduced in any form without written permission from the publisher.

From the publishers

THE YEAR OF THE VAX

Carl B. Marbach

It may be too early for predictions on 1984, but I am going to be bold and tell you about VAX in the upcoming year. '84 will be a big year for VAX with at least four or five major announcements. Why not. DEC won't have much to say about the DECsystem 10s and 20s, the newest PDP-8 was announced last year in the revised form of a DECmate II, and the J-11 (JAWS-11), PDP-11/70 on a chip is likely to be the only 16-bit news. That leaves a lot of room for the VAX/VMS systems.

Don't jump to conclusions that the publisher is a real crystal ball reader; Ken Olsen himself told a group of analysts (securities/stock analysts) and customers (DEC/computer) that we should expect a flurry of new things for VAX. Just in time. As we said last issue, DEC has got to start moving in the right direction or someone might catch up; and while we weren't looking, number one is farther away. With the failure of the large DECsystems and the downgrading of the 16bit world to microcomputer status, VAX is simply "where it's at."

This is not all bad. The large address space and rich instruction set makes VMS fertile ground for a great operating system. The VAXcluster concept is just one outgrowth that will make VAX the most modular computer system yet. Now that VMS will be the flagship operating system (dethroning TOPS-10) all the vast resources of DEC software engineering will

. . . continued on page 84

From the VAX editor

THE VAX'S ARE COMING!

Dave Mallery

Last week, Ken Olsen gave a lunch for the press and financial analysts in conjunction with the DecTown exhibit. Serveral cats were let out of their respective bags for the audience. The tiny VAX (under the desk) was on the exhibit floor. It should have the power of the 730 and a price to match its size. That will probably be followed by the single chip VAX with 750-size performance. Then the 'big' VAX. Then the 'magnificent' VAX. Ken was rather proud of that one. It seems that the simulation is about done and they are probably cutting some prototype silicon.

Perhaps you noticed a news item recently about how DEC bought a 9% interest in Gene Amdahl's not so new start-up, Trilogy, It seems that Trilogy has some mighty impressive design tools and some very high density chip technology. How dense? So dense that they are talking about producing 3081 size mainframes using 5% of the components. Now some of this talk is 'pre-going-public' hype. I always like to follow the smart money. I was impressed by DEC's buy-in.

A company like Trilogy seems to be one of a new breed of 'silicon foundary'. You see, you can't just buy a bunch of integrated circuits, some hook-up wire and a soldering iron and go into the computer business any more. Machines like the 'magnificent' VAX run for years before they are ever built. They run in simulation. Their circuits are impossible to

. . . continued on page 84

Publishers: R.D. Mallery, Carl B. Marbach

VAX Editor: Al Cini

Director of Advertising: Helen B. Marbach

Business Manager: Peg Leiby

Advertising Sales Manager: Hope Makransky

Editorial Assistant: Linda DiBiasio

Production Manager: Georgia F. Conrad

Subscription Fulfillment: Kathi B. Campione, Claire Hollister

Steven Barsh, Margie Pitrone Connie Mahon, Terry McMahon

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

Mid-Atlantic and Southeastern Representative: Ed Shaud

Phone: 215/688-7233

N.Y. & New England Representative:

Phone: 617/934-6464

Contributors:

Philip G. Anthony, Jeff Corbett, Steven L. Edwards, Dave Goodman, David S. Hayes, Roger E. Holmes, Austin Kinsella, Gerard Kiernan, Michael Mowat, David Scriven, Robert G. Smith James B. Wilkinson. Gary W. Wong

Cartoons: Frank Baginski, Douglas Benoit,

Printing & Binding: Schneider Litho Co., Inc.

Cover Art: Jim Lang

ALL PROGRAMS PUBLISHED IN THE VAX/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 PEOPLES' 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

Managing Editor: James L. Trichon

United Kingdom Representative: Pauline Noakes

P.O. Box 187, Radnor, PA 19087

Jack Garland P.O. Box 314 S.H.S., Duxbury, MA 02331

Design, Typesetting & Layout: Grossman Graphics

material will be made only upon publication.)

THE 'RIGHT TOOL' distinguishable hardware ready to heap into Al Cini The VAX/RSTS PROFESSIONAL serves

two distinct computer user communities with a common mission: computing, in support of problem-solving.

So, what'll it be? VAX/VMS (or UNIX) or PDP-11/RSTS (or RSX or RT or UNIX . . .)?

To a lot of RSTS users, the VAX is where DEC is spending all of her money these days — "neglecting" the PDP-11 in favor of this new, redundant wonder toy. Nobody asked for it, nobody needed it you "can do it all with a PDP-11."

To a lot of VAX users, RSTS/E is the "confining environment" they outgrew. They wonder how they ever managed to survive it let alone use it, and can't imagine ever stepping backward to it again.

In a century or so, if we're still around, we'll sort through the attic junk from the last half of the twentieth century, finding PDP-8s, -11s, VAXs, 4341s, System/34s and 38s — all sorts of rusted, essentially inthe "20th Century Computer Equipment" closet at the Smithsonian. How they worked will be summarized on an index card for quick reference, but the problems they helped solve will fill volumes of human history.

So, for today? Keep an eye on the tools you're using, and an eye on the tools you might be using instead, but never lose sight of the problems: those you're solving, but could solve better (viz., more completely, at lower cost); and those waiting to be solved, by human imagination and initiative supported by (among other things) computers and programs.

So, what'll it be? A VAX running VMS, or a PDP-11 running RSTS/E?

An objective understanding of what's to be done, knowledge of the costs involved and the funds available, and awareness of compatibility and growth constraints is the

. . . continued on page 84

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

THE SYSTEM PERFORMANCE TEAM

Technically Superior Products from System Performance House, Inc.

1. DOPTER

The best disk structuring/optimizing program available. DOPTER automatically performs every optimization available in any other program, plus many more which are found only in DOPTER. DOPTER is available for both version 7 and 8 of RSTS/E. DOPTER is an excellent tool for converting version 7 disks to version 8 type disks. Single CPU price - \$750.00.

2. UFDOPT

A fast, safe REORDR-like program which builds better directories which stay that way longer than those produced by REORDR or RDR*. Do not be confused-only UFDOPT puts the most frequently accessed files at the front of the UFD's, deletes unused attributes, sets the contiguous bit where appropriate, and puts free space near the front of the UFD to speed file creation and to help prevent "tangling" of the directory links. Single CPU price - \$250.00.

3. SPSORT

An extremely fast, versatile RSTS/E sort utility. SPSORT needs no scratch space except for a 5 block control file. It comes in three forms, a SORTG/SORTM* emulator, a Sort-1 Plus* emulator and an FSORT3* emulator. These sorts provide a direct repacement for their namesakes-with no change in software. In addition, SPSORT's emulators have been extended to sort DIBOL decimal, packed and EBCDIC data types. Single CPU price - SORTG/SORTM emulator, \$450.00, Sort-1 Plus emulator - \$850.00, FSORT3 emulator - \$1250.00.

Coming Soon

A single-disk DOPTER which will optimize single-disk systems. There will be no disk packs or tapes to shuffle; just run it and enjoy the performance increase!

For more information on any of the above products, write or call the SYSTEM PERFORMANCE HOUSE, INC.

*SORTM and SORTG are registered trademarks of Digital Equipment Corporation. Sort-1 Plus is a registered trademark of Oregon Software, Inc. FSORT3 is a registered trademark of Evans, Griffiths and Hart, Inc. RDR is a registered trademark of Software Techniques, Inc.



System
Performance
House, Inc.

	 	 		_
Please se		⊐SPSORT		
Name Company Address _	 			-
Phone				-
		House,	Inc.	

5522 Loch More Ct. E. • Dublin, Ohio 43017 • 614/764-1585



Send letters to: PRO TALK P.O. Box 361 Ft. Washington, PA 19034-0361

One quick comment: I believe that the RSTS community could use some timely information concerning the implementation and use of new DEC products, BASIC+2 V2.0/1 and RSTS/E V8.0 in particular. I've had various strange things happen with the new BASIC+2 but no time to SPR!

David Platton Santa Barbara, CA

Everyone has had strange things happen with this combination. We suggest you wait for V2.1 of BASIC+2.

Just as the magazine became VAX/RSTS Professional — I came across the enclosed advertisement [see page 19]. It would seem that the future direction of the VAX is to become a household word, and that the VAX-on-a-chip is going to be far more versatile than the computing industry bargained for! (Imagine the first computer able to clean its own filters!).

BR. Bernard Halpin Computing Centre Bunbury Catholic College Bunbury, W.A.

P.S. Also, please find below one of my student's version of a terminal INUSE program. At our installation, we define some PPns which can't use INUSE (see line 9000), and line 9010 can be used to list any keyboards at which INUSE is prohibited. See the General Description and program listing for further documentation.

! Extend Mode in Basic-Plus. Terminal In Use Program : Inuse.Bas : V7.0-08 Program : 3.2 Edit Date 28-May-83 Modification History Version Edit Who Date Purpose 20-Feb-83 Initial Creation. 2.0 3.0 3.1 27-Feb-83 28-May-83 29-May-83 Initial treation. \$
Rewrite to system standards. \$
Rewrite with additional features. \$
A line was inserted at line 1400 to \$
'set around' the keyboard monitor. \$
The 'INUSE' data statement was also \$ chansed. Lines were inserted so that the program can log a user off when the maximum time is up if flagged to do so. V7.0-08 3.2 BJM 29-May-83 General Description

INUSE will allow a user to mark his or her terminal as beind in use. It will, in effect, 'freeze' the users terminal until such time as a predefined time limit expires or until the user returns to the terminal and enters his previously entered pass key. The program can be given an overide password to exit, if the password has been initialized and the user is not privelaged. Channel Information Purpose Chn1 # User's keyboard. Opened in Mode 8 (echo control mode) in order to disable the °C mechanism. Variable Information Variable Purpose Flas. See Line 1065. Flas. See Line 1065.

Time (in seconds past midnisht) when the maximum time is up (aprlies only if TX has a non-zero value).

Current job number.

Current keyboard number.

Keyboard number that program is not allowed to run under (see line 9010).

Flas. Will tell us if we should lod out when max time is up (only has meaning if TX has value).

Array to hold info. from sys calls.

Overide password (see line 1065).

Pass key entered by user at line 1200. L% M%(30) Pass key entered by user at line 1200. PPn [P1:P2] that prodram is not allowed to run under (see line 9000). Current prodrammer number. P\$ P1%,P2% Prez Current programmer number.
Current project number.
Users attempt to exit from program. To exit,
its till equal the pass key or overide key
(if the overide key is initialized).
Maximum time limit before program will exit
automatically (see line 1065).
Utility variable.
Utility variable.
Utility variable. T% Utility variable. Current Edit version of program. Contains the time remaining to wait (only applicable if T% has a positive value). 900 Dimension Statements M%(30%) V\$ = 'V3.21000 ! Version of program. On Error Goto 19000%
! Set up standard error trap.
\ t0\$ = Sus(Chr\$(6%) + Chr\$(- 7%))
! Enable CC trap. \text{ top = Sus(thr\$(6%) + Chr\$(- 7%) \\ ! Enable ^C trap. \\ Print if Pos(0%) \\ Print 'Inuse'; Chr\$(9%); V\$; Chr\$(9%); Cvt\$\$(Right(Sus(Chr\$(6%) + Chr\$(9%) + Chr\$(0%)), 3%), 4%) \\ Print 'Terminal in use Program' ! Kb: header. Change Sys(Chr\$(6%) + Chr\$(14%)) to MX ! Our kb: number.

. . . continued on page 81

EMULEX TALKS DEC



CONCERNED ABOUT THE FCC...?

No need to be — when you assemble or expand your VAX-11 system with Emulex's new DMF-32-compatible CS21/FX Communication Controllers. Just ask for model FC or FD. Emulex-supplied filters and mounting hardware help you to meet and maintain FCC compliance. No doubt about it: A few ounces of RFI-prevention can be worth pounds of cure.

PICTURE-FRAME PERFECT...

But that's only part of the CS21/FX story. The all-new subsystems allow you to completely replace the asynchronous-communication portion of a DEC DMF-32 installation with lower cost, higher performance Emulex hardware. Start with a single hex controller board with twice the line capacity of an equivalent DMF-32 board. Add in such standard Emulex features as self-test firmware, modem control on all 16 lines, and something completely new: the ability to drive either EIA RS-232C or 20mA current-loop communication lines. Specify a FB or FD subsystem, and you can plug either EIA or current-loop lines into a single distribution panel by simply rewiring your DB25 connectors. The distribution panels themselves are also all-new—and "picture-frame" perfect. They fit right into an existing DMF-32 mounting frame, or you can use the new dual-panel frame shipped from Emulex.

GOOD NEWS FOR PDP-11 AND VAX USERS...

As production soars, production costs drop, and most of these savings are now being passed along to the people who made it possible. Item: a 16% reduction in the price (\$2125 in quantities of 100) of the popular CS21 communication controller family, used to multiplex 16 lines at rates of up to 19.2 Kbaud. This family includes DEC DMF-32, DH11 and DZ11-compatible emulations.

A similar price reduction (\$2690 in quantities of 100) applies to Emulex's SC21 large-disk controller for both PDP-11 and VAX Unibus CPUs. And that's not counting our usual mix-and-match discounts for OEM and other volume customers.

FROM THE EMULEX FILE...

Year-end figures for fiscal year 1983 show that both sales and earnings per share doubled over the past 12 months. Revenues totaled \$31,012,089, an increase of 93 percent over 1982's \$16,084,199. Net earnings were \$5,618,530, an increase of 155 percent over net earnings of \$2,200,933 for the previous year. Earnings per share were up 104 percent to \$0.98 per share from last year's \$0.48 per share. Special invitation: Next time you're in Southern California, visit our Costa Mesa headquarters and let's talk DEC there.



3545 Harbor Blvd., P.O. Box 6725, Costa Mesa, California 92626, Toll-Free (800) 854-7112, In Calif. (714) 662-5600. See us at DEXPO WEST, Booth 512 See us at COMDEX, Booth 2030

VAXclusters

Multisystem Technology for VAX Computers

By Carl B. Marbach

In the second quarter of this year DEC announced a new concept for multi-VAX systems called VAXclusters. Simply put, a VAXcluster is a group of tightly coupled VAX computers that can share files and resources. This new multisystem technology allows a configuration of up to 16 VAX 11/780, VAX 11/782 and VAX 11/750 computers and intelligent storage subsystems to function outwardly as a single, large, highly powerful system.

Not DECnet, not Ethernet, not a networking system, VAXclusters will be a way of modularly setting up a large single computer system. Once you have an 11/750 you will be able to double the capacity by adding a second 11/750. CPUs and disks will be logically separate from each other. You will be able to add disks when you need storage and add CPUs when you need more terminals or processing power.

VAX-11/750
RA60
CI750
CI750

STAR
COUPLER

RA80
RA80

FIGURE A. Entry-level configuration for Digital's new VAXcluster system, a loosely coupled multiprocessor design, incorporates two VAX-11/750 processors and runs under Version 3.4 of the VMS operating system. The configuration includes Cl750 interfaces and local, 205-megabyte RA60 disk drives for each processor, Cl cables, Star Coupler, HSC50 mass storage controller, and two 121-megabyte RA80 disk units. VAXcluster systems can grow from three to 16 nodes and encompass VAX-11/780 and VAX-11/782 processors and up to 24 disks on each HSC50 controller.

DEC's first attempt at multi-processing was linking two early PDP-6 (36-bit) processors. One did all the I/O while the other handled the computing. A later version developed by a customer linked two PDP-10s (more 36-bitters) and let them share the computational as well as the I/O load. There were rumors that a dual processor 11/70 existed but that it was scrapped when benchmarks proved it to be superior to a VAX 11/780 in performance and DEC didn't want to make the new VAX look bad. The last dual processor is the VAX 11/782 which is really two VAX 11/780s put together. The VAXcluster promises to be a real advance in multi-computer technology.

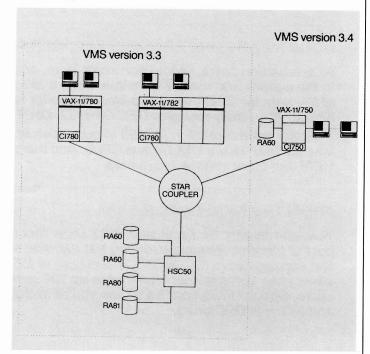
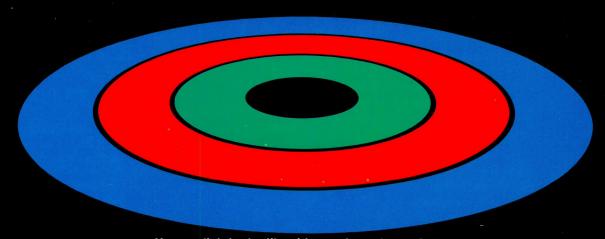


FIGURE B. Above diagram illustrates VAXcluster support by Versions 3.3 and 3.4 of Digital Equipment Corporation's VMS operating system. Version 3.3 supports VAXcluster systems including VAX-11/780 and VAX-11/782 processors with Cl780 interfaces. Version 3.4 extends support to the midrange VAX-11/750 system with the Cl750 interface. Both versions support the HSC50 intelligent mass storage controller with Digital's RA series of disk drives (RA60, RA80, RA81) and enable global sharing of stored data. Functioning outwardly as a single, large computer, the VAXcluster system delivers higher performance, support for more users, and enhanced data integrity and system availability.

WHAT YOU DON'T KNOW ABOUT RSTS/E 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 wellordered 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.

- 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 activities and second to the complete se
- file activity so you can see what your system is doing.
- 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.

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

DEC calls the new VAXcluster system concept "the most innovative, efficient, and achievable path to high performance, high capacity computing in the industry. The new systems approach offers cluster-wide data sharing and software tools to ensure data integrity and higher system availability. It is, in the simplest form, the new way to grow systems." In fact, VAXclusters will enable users to access data stored anywhere in the cluster from any CPU in the cluster.

Hardware components of a VAXcluster system include VAX 11/780, VAX 11/782, and VAX 11/750 processors; a dual path Computer Interconnect (Cl) bus; an intelligent Cl interface for each processor; a Star Coupler for physical interconnection of processors and storage subsystems; and a mass storage controller, which consists of an HSC50 intelligent subsystem with associated disk storage devices.

VMS will support the VAXcluster concept beginning with Version 3.3 which will provide support for clusters of VAX 11/780 and VAX 11/782 processors and shared access to the HSC50 subsystem with RA60, RA80 and RA81 disk storage. Version 3.4 will extend support to include VAX 11/750 processors.

VMS is an integral part of the operation of the new VAXcluster system and is a key component of the cluster architecture. VMS implements facilities that provide enhanced system availability, multisystem file and record sharing, and the ability to accept new processors and mass storage devices to the cluster as the user requires.

Future releases of VMS will ensure cluster-wide data access, integrity, and availability even in the event of a system failure. VMS will provide global data sharing through a distributed file system that extends file-level access to all disk storage, a distributed lock management facility that synchronizes file access, and a mass storage protocol that gives cluster access to locally connected mass storage. RMS will use these new features to allow cluster-wide access to all processors attached to the cluster.

The distributed lock management feature will allow multiple file/record access by providing a namespace in which processes can lock and unlock resource names. Processes can be queued in a "wait" state for access to locked records. The lock manager detects and prevents deadlocks and the "deadly embrace" caused by mutual locking of records by a set of concurrent processes.

Using a multiple disk journaling facility it will be possible to make a "failsafe" database that will be in tact even if a storage unit fails. A checkpoint/restart facility will also be included to protect lengthy tasks against a host computer failure.

The CI bus is a dual, redundant data path that consists of two transmit and two receive cables, each with a 70-megabit/second bandwidth. It will accommodate up to 16 nodes in a computer room environment where each node can be either a computer or an HSC50 intelligent mass storage controller. Processors are connected to the CI bus through the CI780 interface or the CI750 interface which are microprocessor based intelligent controllers. Each CI interface is connected to both paths or the CI bus and uses whichever is available.

The Star Coupler is the common connection point for all nodes in any configuration, with a maximum of 45 meters between connections. The basic Star Coupler will be configured for eight nodes with expansion to 16 available.

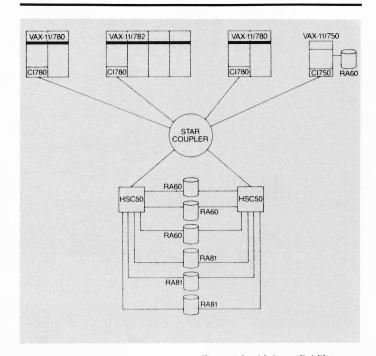


FIGURE C. A VAXcluster system configured for high availability may include dual-ported disk units connected between pairs of HSC50 storage controllers, providing an entirely redundant link from each processor to stored data. Forthcoming versions of the VMS operating system, to be released beginning this summer, will provide data integrity and availability features to match the hardware potential of VAXcluster systems, such as automatic switchover of disk devices to circumvent an interruption of HSC50 operation.

The Hierarchical Storage Controller (HSC50) is a self contained mass storage subsystem that goes in between the VAX hosts and the disk devices. The HSC50 can handle multiple, simultaneous operations on several disk drives and will optimize throughput by optimizing both seek and rotational position of the disk heads. It is the HSC50 itself which will be able to do a disk shadowing during which a complete copy of disk operations can be made on a backup disk. Each HSC50 can support up to six disk channels and connect to up to four host processors; future support will add more disk drives, processors and include tape subsystems as well. The HSC50 subsystem has internal diagnostics and component redundancy allow operation even when parts of it have failed. Both the CI cable and disk drive connections use transformer coupling, so they can be removed and reconnected without disrupting other devices on the cluster.

With previous DEC systems, knowing how to grow has not been easy. By utilizing the cluster concept, the system designer can now build in a complete growth path that should make the future of his computer system easier to plan for and eventually execute. VAXclusters are an exciting capability of VAX, and will surely be extended to include the whole VAX family. Isn't it great what can be done with an operating system that has room to grow.



LSI-11[®] compatible controller for 80-300MB CDC, Ampex and Fujitsu CMD/SMD

Dataram Corporation offers the industry's widest range of DEC-compatible peripheral controllers — from comparatively simple NRZI tape controllers to complex 300 MB storage module drive (SMD) controllers.

An impressive array of state-of-the-art controllers, all built around high-speed bipolar microprocessors. All software compatible with the host LSI-11, PDP®-11, or VAX® minicomputer...and all available now.

And Dataram's controllers are designed to save you money, and, more importantly, space — our controllers typically occupy half the space required for the comparable controller from DEC. Doing it with a level of performance that makes any member of this family worth looking at.

The chart shows our current family of peripheral controllers, growing every day. If you don't see the controller you need, we're probably working on it right now. Call us and discuss your requirements.

CORPORATION

Princeton Road Cranbury, New Jersey 08512 Tel: 609-799-0071 TWX: 510-685-2542

CONTROLLER	DESCRIPTION	COMPATIBILITY	
C03	Cartridge disk controller	RK05	
C33	Cartridge disk controller	RK05	
T03	NRZI mag tape controller	TM11/TU10	
T04/C	Mag tape streamer coupler	TM11/TU10	
T04/N	NRZI mag tape controller	TM11/TU10	
T04/D	Dual density mag tape controller	TM11/TU10	
T34/C	Mag tape streamer coupler	TM11/TU10	
T34/N	NRZI mag tape controller	TM11/TU10	
T34/D	Dual density mag tape controller	TM11/TU10	
T36	Dual density mag tape controller	TM11/TU10	
T34/T	GCR mag tape controller	TM11/TU10	
S03/A, S04/A	80 MB/300 MB SMD controller	RM02/RM05	
S03/A1, S04/A1	80 MB/160 MB SMD controller	RM02	
S03/B	80 MB/300 MB SMD controller	RK07	
S03/C	200 MB/300 MB SMD controller	RP06	
S03/D, S04/D	96 MB CMD controller	RK06	
S33/A	80 MB/300 MB SMD controller	RM02/RM05	
S33/A1	80 MB/160 MB SMD controller	RM02	
S33/B	80 MB/300 MB SMD controller	RK07	
S33/C	200 MB/300 MB SMD controller	RP06	
S33/D	96 MB CMD controller	RK06	
Products printed in red are LSI-11 Bus compatible. Products printed in black are UNIBUS® compatible for PDP-11 and/or VAX			

minicomputers.

DEC. LSI-II. PDP, UNIBUS and VAX are registered trademarks of Digital Equipment Corporation.

SHAREABLE IMAGES II or Sharing Images for Fun and Profit

By Al Cini

In the last issue of the VAX/RSTS PRO (Vol. 5, No. 4), we outlined the advantages of shareable images under VMS and "cook-booked" some procedures for building and using them in a few sample situations. This article will focus on the use of shareable images to reduce software maintenance headaches, in particular when "external factors," such as the Federal Government, force changes to RMS file record layouts (or DBMS-32 schema or sub-schema definitions). As in the last article, our examples will be in VAX-11 BASIC V2 but the principles are readily adaptable to other programming languages.

THE PROBLEM

Records within RMS files are usually "mapped" (in BASIC, this is literally via a "MAP" statement; in COBOL, a "FD" entry performs this function) into buffer areas within application programs as a series of program variables allocated positionally from the beginning of the buffer:

This "buffer area layout" can be defined in BASIC as follows:

```
RECORD EXAMPLE

STRING A = 7

STRING B = 9

LONG C

END RECORD EXAMPLE

...

MAP (BUFFAREA) EXAMPLE X
...

OPEN "ABC.FIL" AS FILE 1% &
...

, MAP BUFFAREA &
, PRIMARY X::A &
...

END
```

If you have to add a couple of bytes to the variable A in this example, or if you need to cram a new field between B and C, you must:

- Define a new RMS file with the changed/newly added fields, and write a program or use DTR to load it from the old file.
- 2. Re-compile and re-link the affected application programs most of which have no need of the newly defined data so that they can reference the re-defined data file properly.

Consider this simple example data record, which we'll keep as CDD\$TOP.PERSON__V1 in the Common Data Dictionary:

```
RECORD PERSON__V1

STRING YEAR_BORN = 2

STRING F__NAME = 10

STRING L__NAME = 20

END RECORD PERSON__V1
```

The following simple application program (FIXUP.BAS) processes a sequential file "TEST.FIL" whose records are defined as type PERSON_V1, converting each of the name fields to upper-case-only and printing a list of the names in the file.

```
10
        SINCLUDE SFROM SCDD "CDDSTOP. PERSON V1"
        MAP (PERSON_BUFF) PERSON_V1 PERSON
        OPEN "TEST.FIL" AS FILE 1% &
                , ORGANIZATION SEQUENTIAL FIXED &
                 , MAP PERSON_BUFF
        ON ERROR GO TO ERROR_RTN
    READ_LOOP:
        PERSON::L_NAME = EDIT$(PERSON::L_NAME, 32%)
        PERSON::F_NAME = EDIT$(PERSON::F_NAME, 32%)
        UPDATE #1%
PRINT PERSON::L_NAME, PERSON::F_NAME
        GO TO READ_LOOP
    ERROR RTN:
        IF ERR = 11 THEN
                RESUME 20
        ELSE
                ON ERROR GO TO O
        END IF
        PRINT "Finished!"
20
        CLOSE 1%
32767
```

If, sometime around the year 2000 or so, we were to need to keep a full four-digit date of birth for the folks in TEST.FIL, we'd have to redefine the PERSON_V1 record inserting two bytes into it, create a TESTNEW.FIL, populate it from TEST.FIL, and then re-compile every application program which %INCLUDEd CDD\$TOP.PERSON_V1.

Even if we knew where every such program was (the CDD audit records can help a little), this is still a tall order for what may be a whole lot of source files. Imagine a more complicated edit to change zip codes to nine digits — affecting maybe a dozen different record layouts in several hundred source programs — and you'll begin to see the inconvenience of it all.

THE SOLUTION

By using shareable images to "hide" RMS file record layouts from application programs, STEP 2 above can be skipped for any programs which don't require modification to use the newly defined portions of the data record. When used this way, shareable images allow programs to "bind"

The most exciting MESSAGE DELIVERY NEWS



since the Pony Express...

DIGITAL ANNOUNCES DECmail/RSTS VERSION 1.0 ELECTRONIC MAILING SYSTEM AT AN AFFORDABLE PRICE TO SPUR YOU ON!

Once, a letter had to travel tortuous routes to reach its destination. Via the swift Pony Express. Or more recently, by way of the office message box.

Now, the wide open spaces between your office and company headquarters, the 18th and 22nd floor, the New York office and the Toronto branch are bridged easily — electronically — with DECmail/RSTS.

DECmail/RSTS provides a low cost, reliable message handling system for RSTS/E sites. With this new system, you can create, edit and process messages — from simple memos to complicated specifications. You can forward your message for delivery almost instantly to a business associate down the hall, across town, around the country or overseas.

DECmail/RSTS also brings you up-to-date in the morning and holds messages for you while you're out. It will organize your material for you — storing, searching and retrieving messages as needed. If you have a problem, an on-line help facility comes to the rescue.

REWARD!

Request full information on your DECmail/RSTS message system now. If you order within 90 days, you will receive a RSTS western-style belt buckle FREE.



Mail the coupon today or call 1-800-DEC-INFO any working day between 8:30 a.m. and 5:00 p.m. E.S.T.

If this sounds as easy as falling off a horse, it is. DECmail/RSTS was designed by the original RSTS/E development team. They put together a system with easy to learn commands and flexible options that make DECmail/RSTS sophisticated enough for advanced users yet simple and uncomplicated for computer novices. DECmail runs on RSTS/E Versions 7.2 and 8.0. No update will be needed until after RSTS/E V8.0 is released. Thanks to DECmail update kits, you won't be saddled with an obsolete software product down the line.

What does all this cost? A lot less than you might think. And the increase in productivity in your working environment — not to mention the wonders of electronic communication — is worth every penny.

Find out about this new electronic message system for your office. Complete, clip out and return the coupon — pronto!

MAIL this coupon to: Digital Equipment Corporation
Direct Response Center, POB 279
Maynard, MA 01754

Please send me complete information on DECmail/RSTS and my eligibility for a Western belt buckle.

Name ______Telephone No. ()______

Company ______

Address ______

City ____ State ____ Zip_____

to data files at run-time rather than at compile-time and, while the techniques we'll be reviewing in this article may appear complicated, the reduction in maintenance sensitivity may be well worth the slight cognitive strain. So, if you're a System Manager or Data Base Administrator or Project Leader take note! Knowing little tricks like this is what they pay you for!

THE TOOL

For this article's purposes, all you need to know about shareable images is that they can be used to house externally compiled sub-programs (viz., FUNCTION or SUB sub-programs in BASIC, or separately compiled COBOL run-units activated via CALL... USING from a main routine) so that they can be changed WITHOUT RE-COMPILING OR RE-LINKING THE MAIN PROGRAMS WHICH CALL THEM (fantastic, eh?), provided:

- 1. You use "transfer vectors" in your shareable image (no sweat, we'll show you how).
- 2. The number and type of arguments in the formal parameter list (i.e., between the parentheses in BASIC or after the COBOL "USING" clause) don't change.

THE TRICK

Very simply, keep RMS record definitions inside shareable images and out of your application programs (you can pull this off with DBMS-32 User Work Areas as well), and with a little "overhead programming" you can insulate your programs from even radical changes to the record layouts.

To pursue our earlier example, we'll first need three "shell" sub-programs to handle all operations involving the MAP for TEST.FIL.

To move data from the MAPped record buffer into the application program (MAPOUT1.BAS, invoked after a GET operation):

```
10 FUNCTION PERSON_V1 MAP_TO_PROG_PERSON_V1
$INCLUDE $FROM $CDD "CDD$TOP.PERSON_V1"

MAP (PERSON_BUFF) PERSON_V1 PERSON

MAP_TO_PROG_PERSON_V1 = PERSON

END FUNCTION
```

Finally, to move data from the application program to the MAP (MAPIN1.BAS, invoked before a PUT or UPDATE operation);

```
FUNCTION LONG PROG_TO_MAP_PERSON_V1 (PERSON_V1 PERSON_ARG)

$INCLUDE $FROM $CDD "CDD$TOP.PERSON_V1"

MAP (PERSON_BUFF) PERSON_V1 PERSON

PERSON = PERSON_ARG

END FUNCTION
```

Compile the "shell" routines: \$BAS OPEN, MAPOUT1, MAPIN1

Prepare shareable image transfer vectors for the shell routines using a little non-threatening MACRO program (XFER.MAR):

```
.TITLE
                  "TRANSFER VECTORS"
. MACRO
                 VECTOR, VECTOR_NAME, ENTRY_NAME
. TRANSFER
                 VECTOR NAME
. IF NB ENTRY_NAME
         . WEAK
                 ENTRY NAME
         . MASK
                 ENTRY NAME
        JMP
                 L^ENTRY_NAME+2
. IFF
                 VECTOR_NAME
         . MASK
                 L^VECTOR_NAME+2
        JMP
- ENDC
. ENDM
        XFER, EXE, NOWRT, PIC, SHR, GBL
. PSECT
VECTOR
        OPEN_PERSON
        MAP TO PROG PERSON V1
VECTOR
VECTOR
        PROG TO MAP PERSON V1
```

**REMEMBER: When you add routines to a shareable image, insert their transfer vectors AFTER existing vectors. Also, NEVER change the order of the transfer vectors!

Assemble the transfer vectors:

\$MAC XFER

Build the shareable image:

```
$LINK/SHARE-SHELLRTNS XFER, OPEN, MAPOUT1, MAPIN1, SYS$INPUT/OPTIONS
GSMATCH-LEQUAL, 0, 0
CLUSTER-UPFRONT
COLLECT-UPFRONT, XFER
PSECT_ATTR-PERSON_BUFF, NOSHR, LCL
```

(All of these linker options were explained in the last issue, and are discussed in the Linker Reference Manual.)

To use these shell routines, edit the FIXUP.BAS main program as follows:

```
SINCLUDE SFROM SCDD "CDD$TOP.PERSON_V1"
10
         EXTERNAL LONG FUNCTION &
                                   OPEN_PERSON &
                                   PROG_TO_MAP_PERSON_V1 (PERSON_V1)
         EXTERNAL PERSON_V1 FUNCTION &
                                   MAP_TO_PROG_PERSON_V1
        DECLARE PERSON_V1 PERSON
        X%=OPEN_PERSON
         ON ERROR GO TO ERROR_RTN
    READ_LOOP:
        GET #1%
         PERSON = MAP_TO_PROG_PERSON_V1
                                                     ! Move MAP to program.
        PERSON::L_NAME = EDIT$(PERSON::L_NAME,32%)
PERSON::F_NAME = EDIT$(PERSON::F_NAME,32%)
         X%=PROG_TO_MAP_PERSON_V1 (PERSON)
                                                     ! Move program to MAP.
         UPDATE #1%
         PRINT PERSON::L_NAME, PERSON::F_NAME
        GO TO READ LOOP
    ERROR RTN:
        IF ERR = 11 THEN
                 RESUME 20
                 ON ERROR GO TO O
        END IF
20
         PRINT "Finished!"
         CLOSE 1%
32767
```

After a quick \$BAS FIXUP (it's even quick on a 730), you can link this program to the shell routines' shareable image:

```
$LINK FIXUP,SYS$INPUT/OPTIONS
SHELLRTNS/SHARE
```

. . . and you're ready-to-run:

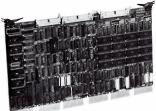
DEFINE/USER SHELLRTNS device:[directory]SHELLRTNS.EXE $RUN\ FIXUP$

(As discussed last issue, the \$DEFINE step is required unless SHELLRTNS.EXE is kept in SYS\$SHARE.)

If you're in the market for communications modules, make the ABLE connection now. And join the thousands who already have.

We are known as the innovators. Most of our products are industry "firsts" which become popular quickly, then settle into a stage of steady long-term acceptance. These four DEC-compatible, communications devices fit the pattern perfectly. They are ABLE originals. They achieved instant success worldwide. They provide top performance. And they are very reliable. Read on to find the one for you.

INCREASED VAX THROUGHPUT.



ABLE VMZ/32¹⁵ 16-line DMF/32 subset

Here's an asynchronous microcontroller with programmable DMA, fully transparent to VAX/VMS as two 8-line DMF 32's and contained on a single board. Priced

below the DZ11-E, it outperforms DZ or DH devices under VMS v.3, has interrupt-driven modem control on every line, and includes an output throttle which lets peripheral devices optimize their own data rate.

#1 UNIBUS DMA.

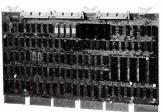
Then there's our DH/DM, the original multiplexer which puts 16 lines with modem control on a single board. This popular device meets UNIX VAX system



ABLE DH/DM[™] 16-line combination DH11 & DM11 replacement

needs for DMA communications requirements, serves UNIBUS systems equally well, and beats them all for MTBF, throughput and products soon to come.

price. Other features include on-board diagnostics, modem control on all lines, superior on-board silo depth and variable prom-set. **SYNC/ASYNC FLEXIBILITY.**



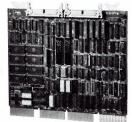
ABLE DV/16 16-line DV11 replacement

A controller for the PDP-11 user, the DV/16 contributes microprocessorderived flexibility, which permits mixing of sync and async lines in combinations

of 4 or 8 lines with modem control and full system software compatibility. It takes less than half the space of a DV11 and uses word transfer instead of byte DMA to gain a 2 to 1 speed advantage or permit operation in half the bandwidth required for data transfers.

Q-BUS DMA.

The Q/DH is an asynchronous controller which makes DH-class performance possible on PDP-11/23 and LSI-11/23 Q-BUS systems. It connects the standard Q-BUS to as many as 16 async lines with DMA output capabil-



ABLE Q/DH[™] 8 or 16-line DH/DM for Q-BUS

ities and allows optimum Q-BUS utilization. Features include software compatibility with RSTS/E and RSX operating systems, large input silo, modem control on all lines.

Write for details on our complete line of DEC-compatible products. Be on the lookout for exciting new ABLE communications products soon to come.

For Immediate, Toll-Free Information, Dial 800 332 ABLE.



CORPORATE OFFICES
ABLE COMPUTER
1732 Reynolds Avenue
Irvine, CA 92714 • (714) 979-7030

NATIONAL OFFICESBurlington, MA (617) 272-1330
Irvine, CA (714) 979-7030

INTERNATIONAL OFFICES Canada (Toronto) (416) 270-8086 England (Newbury) (0635) 32125

READY TO EDIT

Now, let's make our move on that four digit birthdate. First, define a new record (CDD\$TOP.PERSON__V2) with the new information in it (leave the old record Version 1 definition around):

RECORD PERSON_V2

STRING YEAR_BORN = 4
STRING F_NAME = 10
STRING L NAME = 20

END RECORD PERSON V1

Next, edit OPEN.BAS and then write new MAPOUT2 and MAPIN2 shell routines to move the new record format to and from the file MAP buffer:

(OPEN.BAS)

10

FUNCTION LONG OPEN_PERSON
\$INCLUDE \$FROM \$CDD "CDD\$TOP.PERSON_V2"

MAP (PERSON_BUFF) PERSON_V2 PERSON
OPEN "TEST.FIL" AS FILE 1% &
, ORGANIZATION SEQUENTIAL FIXED &
, MAP PERSON_BUFF

END FUNCTION

(MAPOUT2.BAS)

10 FUNCTION PERSON_V2 MAP_TO_PROG_PERSON_V1 %INCLUDE %FROM %CDD "CDD\$TOP.PERSON_V2"

MAP (PERSON_BUFF) PERSON_V2 PERSON

MAP TO PROG PERSON V2 = PERSON

END FUNCTION

(MAPIN2.BAS)

FUNCTION LONG PROG_TO_MAP_PERSON_V2 (PERSON_V2 PERSON_ARG)
%INCLUDE %FROM %CDD "CDD*TOP.PERSON_V2"

MAP (PERSON_BUFF) PERSON_V2 PERSON

PERSON = PERSON_ARG

END FUNCTION

Now edit MAPOUT1 and MAPIN1 to provide a transparent "compatibility mode" automatic translation between the V1 and V2 formats:

(MAPOUT1.BAS)

10 FUNCTION PERSON_V1 MAP_TO_PROG_PERSON_V1
\$INCLUDE \$FROM \$CDD "CDD\$TOP.PERSON_V1"
\$INCLUDE \$FROM \$CDD "CDD\$TOP.PERSON_V2"

MAP (PERSON_BUFF) PERSON_V2 PERSON

MAP_TO_PROG_PERSON_V1::YEAR_BORN = & RIGHT(PERSON::YEAR_BORN, 3%)
MAP_TO_PROG_PERSON_V1::F_NAME = PERSON::F_NAME

MAP_TO_PROG_PERSON_V1::F_NAME = PERSON::F_NAME MAP_TO_PROG_PERSON_V1::L_NAME = PERSON::L_NAME

END FUNCTION

(MAPIN1.BAS)

10 FUNCTION LONG PROG_TO_MAP_PERSON_V1 (PERSON_V1 PERSON_ARG)
\$INCLUDE \$FROM \$CDD "CDD\$TOP.PERSON_V1"
\$INCLUDE \$FROM \$CDD "CDD\$TOP.PERSON_V2"

MAP (PERSON_BUFF) PERSON_V2 PERSON

PERSON::YEAR_BORN = &

LEFT (PERSON::YEAR_BORN, 2%) & + PERSON_ARG::YEAR_BORN

+ PERSON_ARG::YEAR_BORN
PERSON::F_NAME = PERSON_ARG::F_NAME
PERSON::L_NAME = PERSON_ARG::L_NAME

END FUNCTION

Create the required new transfer vectors:

.TITLE "TRANSFER VECTORS"
.MACRO VECTOR, VECTOR_NAME, ENTRY_NAME
.TRANSFER VECTOR_NAME

```
. IF NB ENTRY NAME
                 ENTRY NAME
        . WEAK
         MASK
                 ENTRY_NAME
        JMP
                 L^ENTRY_NAME+2
. TEF
         MASK
                 VECTOR_NAME
        JMP
                L^VECTOR_NAME+2
.ENDC
. ENDM
. PSECT XFER, EXE, NOWRT, PIC, SHR, GBL
        OPEN_PERSON
VECTOR
        MAP_TO_PROG_PERSON_V1
VECTOR
        PROG TO MAP PERSON V1
        MAP_TO_PROG_PERSON_V2
VECTOR
        PROG_TO_MAP_PERSON_V2
```

Compile, assemble, and link the shareable image components:

\$BAS OPEN, MAPOUT1, MAPOUT1, MAPIN2, MAPOUT2

\$MAX XFER

\$LINK/SHARE=SHELLRINS XFER, OPEN, MAPOUT1, MAPIN1,
MAPOUT2, MAPIN2, SYS\$INPUT/OPTIONS

GSMATCH=LEQUAL, 0, 0

CLUSTER=UPFRONT

COLLECT=UPFRONT, XFER
PSECT_ATTR=PERSON_BUFF, NOSHR, LCL

Assuming you've re-defined and re-loaded TEST.FIL, you can now re-run FIXUP.BAS without re-compiling or relinking it and it'll process the new file formats, thinking they're still the old file formats!

CONCLUSIONS

You can write additional "shell routines" around the RMS operations (GET, UPDATE, PUT, etc.), and then someday substitute a completely different data accessing scheme (maybe a data base system) under your applications without having to re-build them! You can also "shell" other data structures like tax tables—and make code edits affecting the behavior of entire systems in a matter of mere minutes!

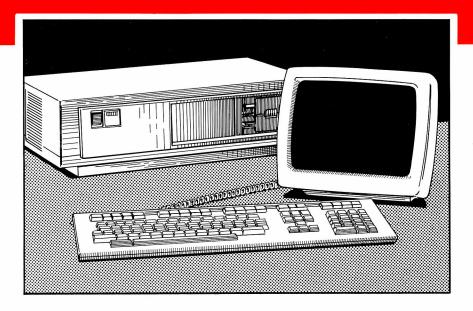
Now, go on and work a few maintenance miracles! ♥



"THE COMPLITER SAYS TO ADD A LITTLE GARLIC!"

PRO/RT-11

RT-11 V.5 FOR PRO 300 SERIES NOW!



Hamilton introduces DEC's popular RT-11 operating system on DEC's Professional 325 and 350 series . . . with full VT 100 emulation. Currently supports BASIC, FORTRAN and DIBOL applications.

PRO 350 system available for sale or rental, pretested and installed on your site, with manufacturer's warranty and maintenance available.

PRO/RT-11 developed jointly by Hamilton and Mini Computer Analysis Ltd.

DEC, PRO 325, 350 and RT-11 are trademarks of Digital Equipment Corp.

FOR COMPLETE INFORMATION, CALL OR WRITE



THE HOUSE OF VAX

6 Pearl Court, Allendale, NJ 07401 TOLL FREE 800-631-0298 In New Jersey 201-327-1444 415 Horner Ave, Toronto, M8W4W3 **TOLL FREE Ont. & Que. 800-268-2106** All other prov. 800-268-0317 In Canada 416-251-1166

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

PATCHING PIP FOR PROTECTION

By Philip G. Anthony, Technical Systems, Fidelity Bank, Philadelphia, Pa.

Some time ago, it became apparent that users — especially those who knew touch typing — weren't going to stop confusing PIP *.*/DI and PIP *.*/DE. On one memorable occasion, it even came out PIP /ZE. This simple mistake had a tendency to cause untold grief, especially to the Technical Systems Department's one DEC "expert." The grief was particularly poignant since Tech Systems is responsible for RESTORing files. I hate being wakened at my desk by the alarm of ringing telephones.

DEC provided no help. As each release of RSTS came out, I tried the "/DI" and "/DE" switches, hoping against hope for a syntax error. My repeated chats with deccies, it seemed, had fallen on ears deafened by the roar of proliferating .CMD files on release tapes, all of which featured a "/DE" somewhere or other.

Since the problem occurred only once every six months or so, the search for a solution kept getting shunted aside. Finally, spurred on by David Spencer's lovely article on ONLPAT (RSTS Pro, February 1983), I was inspired to see what I could do with the executable code. A set of patches eventually appeared, and even worked on our system.

PIP.SAV decodes its commands by referring to a table of half-ASCII strings that start in Version V7.2-O4 at location LONGSW. All command switches in the table are upper-case, suggesting that PIP capitalizes its input before processing it. Required characters in a switch string are normal ASCII; optional characters are ASCII with the high bit turned on. The string is terminated by a CHR\$(255%).

Because PIP searches sequentially through its table for a match, the rules that apply to the order of CCL commands in RSTS apply here too. That is, once a match of required characters is made, PIP stops looking for any further possible matches. If you want to hack around with other switches, or create a totally new command string to replace "/Z[ERO]" (see PIPPAT.004), keep these requirements in mind.

With all the investigation, using an old version of DUMP.SAV that uncovered these facts out of the way, I was ready to start patching. The result is four ONLPAT command files to modify PIP.SAV in such a way as to preserve my reputation for extreme laziness.

The first, PIPPAT.001, makes deletion somewhat safer by requiring the user to type in "/DELE" instead of "/DE". For even further safety, you may want to replace the second "<LF>" in the command file by another "Q&077577", which will force users to enter the entire string "/DELETE".

Once that was out of the way, it still seemed desirable to avoid "/DI" entirely. This could be done, I discovered, by changing the "D" to an "L". By itself, this would have had the unwanted side effect of disabling the "/LI[ST]" command; the workaround is simply to force users to type in the third character, "R", if they really wanted a "LIRECTORY".

When I'd gotten this far, I decided to play with making the system safer from zeroes. Two possibilities occurred, PIPPAT.003 and PIPPAT.004. The first effectively disables the command. (Before applying it, it'd be a good idea to make a copy of the original, untouched by ONLPATish hands, for system managers who need to zero accounts or even an occasional tape.) It works fine for small shops in which only a small group of people will ever need the option.

For full-scale data processing centers, with multitudes of operators and other facilities normally associated with Big Blue systems, it just isn't feasible to keep two copies of PIP around in general use. Besides, an unwary operator can just as easily enter "/ZE" when using the unmodified copy as an unwary user can. The result was PIPPAT.004, which permits zeroing but forces the user to type in the entire string "/ZERO". If this isn't good enough, some other (fourletter) string such as "GL7X" can be substituted in the command file by replacing the "<LF>" and the three "Q&177"s with the letters you want preceded by a single quote:

050106 000524 132 ? 'G 050106 000525 305 ? 'L 050106 000526 322 ? '7 050106 000527 317 ? 'X

Watch out with this one, though. Remember the rules for CCL commands lest you end up with a perfectly valid string that gets short-circuited by being taken for another command.

Patch installation is the same as for DEC-supplied patches. The user runs ONLPAT and answers the "Command file?" prompt with the name of the patch file. The command file prints out its documentation information and then asks for the name of the file to patch. This permits patching to be done in some account other than [1,2] — always a good idea for safety. After that it's all automatic until the "Command file?" prompt reappears. At that point, the patcher can either enter another command file name or hit a CTRL/Z to end the session.

To get these patches onto your system, just use your favorite editor and type them in. ONLPAT appears to be quite forgiving about spacing as long as there's at least one space between each string. There are no guarantees on how the patches will work anywhere else — and definitely no commitment by me, Fidelity Bank, or VAX/RSTS Pro to support them, now or in the future — but ONLPAT has a couple of safety features such as bombing out if the values don't match, that should keep you from going too far wrong. It's a good idea, though to keep a spare copy of the original PIP.SAV around, if only because DEC-supplied command files won't correspond to these changes. Also, I have no idea how RSTS DCL will respond to the modified PIP, and no particular interest in finding out.

With all these caveats, the patches do work here. The value of LONGSW and offsets may have to be adjusted for future or previous releases, but the trouble is worth it to ensure a good night's — or day's — sleep.

Feature patch to PIP.SAV V7.2-04

Change the format of the ASCII string "DELETE" in the PIP command table to force the user to type (minimally) "/DELE" to get rid of a file. This is done by stripping high bits off the bytes "LE", which are a signal to PIP that they are optional.

This change is desirable to prevent inadvertent deletion of files by users typing "/DE" mistakenly when trying to get a directory using the "/DI" switch.

File to patch? Base address? LONGSW Offset address? 114

Offset New? Base 042504 ? <LF> ; Verify only 142714 ? Q&077577 ; Strip high bits in bytes "LE" 050106 000114 050106 000116 142724 ? <LF> 042377 ? ^C : Verify only 050106 000120 ; Up-arrow/C to exit 000122 042377

! PIPPAT.002

050106

Feature patch to PIP.SAV V7.2-04

Modify the switch "/DI[RECTORY" to "LIR[ECTORY] to avoid user confusion with "/DE[LETE]". "/LI[ST]" will also still work to obtain a directory listing. This patch is particularly useful if PIPPAT.001 is not installed.

File to patch? Base address? LONGSW Offset address? 123

Base Offset Old New? 050106 104 ? 114 ; Change "D" to "L" 000123 ; Verify "I" only 050106 000124 111 ? <LF> 050106 000125 ? Q&177 ? ^C ; Make "R" mandatory ; Up-arrow/C to exit 050106 000126

! PIPPAT.003

Feature patch to PIP.SAV V7.2-04

The $^{m}/Z^{m}$ switch is disabled by the simple expedient of changing all characters of the string to lower-case. Since PIP capitalizes its switches before comparing, user entry of "/Z[ERO]" is never matched and generates the response "?Syntax error".

! It is recommended that the system manager copy PIP to another ! directory, or to another name, or both before applying this patch so that he or she retains full functionality of PIP.SAV.

File to patch? Base address? LONGSW Offset address? 524

Offset New? ? 162752 ? 167762 ? ^C 050106 ; Change "ZE" to "ze"; Change "RO" to "ro" 000524 142532 050106 000526 147722 ; Up-arrow/C to exit 000377 050106 000530

! PIPPAT.004

Feature patch to PIP.SAV V7.2-04

The $^{m}/Z^{m}$ switch is modified to require $^{m}/ZERO^{m}$ to zero a device. This confuses the ungodly - to prevent ! inadvertent or malicious zeroing of directories - while ! still permitting users to initialize a magtape.

File to patch? Base address? LONGSW Offset address? 524

Offset 01 d New? Base 050106 000524 142532 ? \0 050106 000524 132 ? <LF> ? Q&177 050106 000525 305 050106 000526 322 ? Q&177 ? Q&177 ? ^C 050106 000527 317 050106 000530

: Switch to octal byte ; Verify only

Strip the high bits to force a match

; Up-arrow/C to exit

The Amazing New VAX

Washes your Carpets as it **Vacuums**.

VAX is a nowerful new dry Vacuum Cleaner that converts in seconds to wash carpets and vinyl floors.

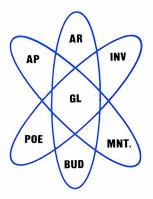
CARPETS, LIKE CLOTHES, ARE FABRICS and must be washed regularly to be hygienic, fluffy and clean. Especially problem areas that are walked on by the whole family almost every day of the year. Only the new VAX 3-in-1 Vacuum Cleaner sucks disinfectant cleaning solution through the carpet's fibres, rinsing out dirt, germs, and grubby marks ordinary vacuum cleaners leave behind

NO WAITING your carpets at left dry enough to walk on.



Country distributors required.

IBMS™ — FINANCIAL APPLICATION SYSTEMS FOR COMMERCIAL, GOVERNMENTAL AND EDUCATIONAL USERS.



™VAX is a registered trademark of Digital Equipment Corporation

IBMS is the only COBOL based, financial system specifically designed for the VAX. Both the commercial and fund accounting versions of IBMS were designed with a layered architecture for greater efficiency, system integrity and application to networking environments. Of course, IBMS is interactive, integrated and user friendly.

THE**FASBE**GROUP

92 Main St, Nashua, NH 03060 (603) 883-3212

Visit us at the Las Vegas DEXPO Booth #204

CIRCLE 196 ON READER CARD

Using The RSTS/E Hidden Data Feature

David Scriven, University of California at Davis

Recently we at the Computer Science Department of U.C. Davis were faced with a problem common to all RSTS/E users. We have many confidential school records online where anybody could break in and look at them. We studied all of the file security methods on the market, but didn't find the solution we needed. We were desperate until we received our distribution of RSTS/E version 8.0.

After a bit of searching, we discovered the undocumented hidden data feature in the new RSTS disk directory structure (RDS 1.1). We have since used this feature to protect our sensitive data from prying eyes.

What we do at our site

Each student who has enrolled at Davis has a unique thirteen digit student identification code. We typically have an average of twelve thousand students in any quarter. There are frequent arrivals and departures from the campus, thus making for a highly active database.

In our own benchmarks against RMS-11, we found that the fastest method to add, delete, and modify student records was to use the student identification number as a ppn and filename. The first nine digits of the student number are used for the filename, the tenth and eleventh for the project number, the last two for the programmer number.

The routine assignment of student numbers makes for an even distribution of files of all ppns in the account scheme. The even loading of files on the disk and frequently run REORDRs make this an amazingly fast data access method.

What we found in 8.0

Needless to say, we were amazed at the speed increase that we got when we converted our disks to the new RDS 1.1 format. Finally DEC has begun to eliminate the file access speed problem. Thanks guys.

After a while, we began to think that perhaps there were more features in RDS 1.1 than we were told about. After all, we remember the hidden control "T" in version 6B, hidden data caching in 6C, named directories in 7.1, and the DH11 DMA output optimization in version 7.2. Surely there had to be something else available that wasn't supported [yet].

We began disassembling the monitor code and found a remarkable new capability of the FIP. RSTS would now allow us to go into the disk and poke new file sizes directly into the directory structure. At first we were confused. What possible use could this ability have? Then we realized that if we had a file with a large clustersize, we could place

data at the end of it and then poke the size to be just one block. Any data written on the allocated but unused blocks could not be read until the filesize was poked back.

How it solved our problem

Our online inquiry and update programs use a common detached job running on a program called COCSPN (Common Online Computer Security Protection Node) for all student file I/O. It was simple for us to change this program so whenever a student data file is opened the filesize is poked to its former full size, and restored to one block when finished.

We have found this method to be perfect in foiling attempts at reading and/or modifying confidential information by unauthorized users. The only time the files appear fully extended is when COCSPN has them open and can lock all the hidden blocks in the file.

The benefits of hidden data

We are quite pleased with our use of the hidden data feature. We have never found any bugs in the code and have been running for over two months without a problem. Our data center manager is now very happy. And attempts by the students using the computer to change information have been totally thwarted.

Backups

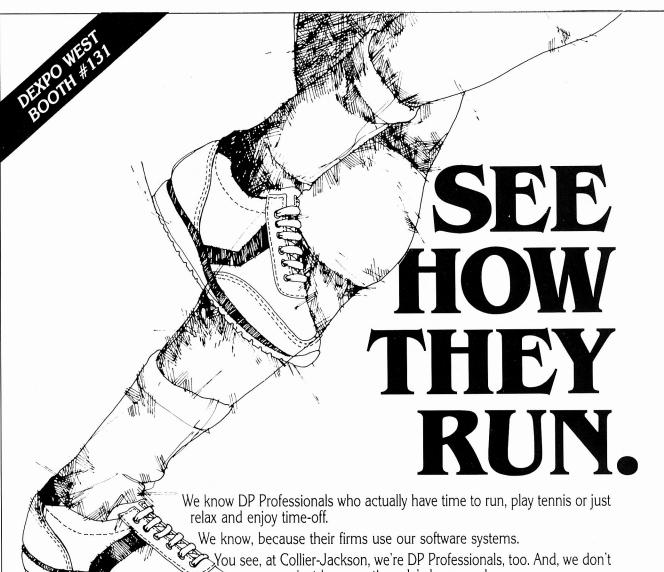
By now you must be thinking that we would have a backup problem. After all, standard file copy utilities would not transfer the hidden blocks. Since we keep all our files with hidden data on a private pack by themselves, this has been no problem. We wrote a special backup program to copy and restore all allocated disk blocks. For extra security, the backup program encrypts the data before writing it to tape and decrypts it on restore.

Conclusion

The hidden data feature was the one thing that finally solved the file security problem at our site. We are looking forward to the day when DEC will document this new functionality for everyone to use. I heard a rumor at St. Louis that it might be supported in version 8.1.

For those interested, we can provide sources to the detached file I/O nodes, the backup and restore programs, and the utility that we used to convert our disk to hidden data format. (Unfortunately they are all too long to be reproduced here.)

Good luck with your hidden data!



go away just because the sale's been made. We're there to answer your questions during installation, system

paralleling, and after live production begins.

We stick with you through thick and thin. We even communicate well with users! That's why our systems stay up and running smoothly.

We specialize in Business Management software for VAX. Written in COBOL our systems are designed to utilize the best features of Digital Equipment Corporation's VAX line of 32 bit minicomputers. This guarantees the highest level of compatibility between VAX and Collier-Jackson software.

If your company is in the market for Payroll, General Ledger, Advanced General Ledger, Accounts Payable, Accounts Receivable or Personnel Reporting packages, call us.

Better yet, visit our Booth #131 at DEXPO WEST.



For more information call us today, or write our corporate offices at 5406 Hoover Blvd., Tampa, Florida 33614 (813) 885-6621

EXHIBIT A. Old RSTS MFD Label Entry

Bits 11111111 7654321076543210 Word O|Link to 1st MFD | always -1 Word 1 _____ always 0 Word 2 Word 3 | always 0 Word 4 | Pack clustersize | |-----! Pack status | Word 5 _____ Word 6! Pack ID part 1 |

EXHIBIT C. Old MFD Name Blockette

Bits 11111111 7654321076543210 Word O'Link to next MFD! PPn Word 1 Word 2 | Password part 1 | Word 3 | Password part 2 | _____ Word 4 | Protect | Status | Word 5 | Access count | |-----| Word 6 | Accounting link |-----UFD DCN Word 7

EXHIBIT D. New RSTS MFD Block 1

Bits

11111111
7654321076543210

Word 0|GFD DCN to [0,*]|
|-----|
Word 1|GFD DCN to [1,*]|
|-----|
Word 2|GFD DCN to [2,*]|
|-----|
Word 254| GFD DCN [254,*]|
|-----|
Word 255| always 0

EXHIBIT B. New RSTS RDS 1.1 MFD Label Entry

Word 7 | Pack ID part 2 |

Bits

|-----

.

11111111 7654321076543210

Word	0	always 1
Word	1	always -1
Word	2	DCN of MFD
Word	3	Disk version #
Word	4	Pack clustersize
Word	5	Pack status
Word	6	Pack ID part 1
Word	7	Pack ID part 2
	ı	

[remainder of block]
[reserved by Digital]

CORRECTION

In the last issue of the VAX/RSTS PROFESSIONAL (Volume 5, Number 4), "Writing AST Routines in VAX-11 BASIC" reported correctly that VAX-11 BASIC subroutines can be used to service asynchronous system trap exceptions, such as those specified in the QIO system service. However, the technique given for handling arguments passed to such AST routines was slightly incorrect.

Arguments passed to AST routines are, as stated, passed "by value," which means that the actual value of the parameter rather than an address pointer is inserted directly into the internal parameter block. There is no syntax for receiving parameters passed by value in VAX-11 BASIC, and the example given:

10 SUB ASTROUTINE (LONG ASTPRM BY VALUE & .LONG RO BY VALUE & .LONG R1 BY VALUE & .LONG PC BY VALUE & .LONG PSL BY VALUE)

... will not compile correctly. There are, however, some work-around solutions:

1. Omit the "BY VALUE" phrases in the subprogram formal argument

list. The subprogram will now compile correctly.

2. If the AST subroutine can ignore the ASTPRM parameter (i.e., the AST happened in a program you wrote, and you omitted the ASTPRM parameter in the call to SYS\$QIO), the main program and AST subprogram can communicate freely through a MAP or COM area.

3. If the AST subprogram must inspect the ASTPRM parameter (i.e., the AST happened in a program you didn't write such as the record-level interface of the 2780/3780 package — and some action must be taken based on its value), you can "fool" BASIC into giving it to you by using the LOC function:

SUB ASTROUTINE . . . (as above) DECLARE LONG X

X = LOC (ASTPRM)

The LOC function will return the 32-bit field in the internal parameter list, which will contain the item passed "by value" in ASTPRM.

START A User Control Run-Time System

By David S. Hayes, Ripon WI

PURPOSE:

At most computer installations, there are users who need only to run a specific program, or set of programs. Data entry operators frequently fall into this category. Rather than have these people turned loose with 'READY', it would be much more convenient (not to mention safer) for them if their programs were automatically executed when they logged in.

The standard approach to this program is to have LOGIN chain all new users to a START program, which determines what is to be done with them according to their project-programmer number. This method suffers from three flaws. First, it may be interrupted, even if the START program is error-trapped, by a user who types two control-C characters during the chain operation. Second, it is slow, since the program must be loaded from disk every time a user logs in. And finally, the program is liable to take up a significant chunk of memory.

A better method is to implement the START program as a run-time system. A switch operation to a new run-time system cannot be effectively interrupted by control-C. This operation is very fast, if the new RTS is flagged as permanently memory-resident. Finally, a RTS occupies very little memory, even if permanently loaded.

There is only one small flaw in this idea. A run-time system on RSTS/E must be written in MACRO! Since this may be more MACRO than the typical programmer wants to undertake, or has time to write, I will present here the complete text of START, a run-time system used at Ripon College. This program may be modified as necessary for other installations.

HOW START WORKS:

Whenever a user runs LOGIN, a special exit sequence is used. LOGIN places 'LOGIN' in core common, and switches the job's default keyboard monitor to START. This causes START to be entered at the location pointed to by the psuedo-vector P.NEW.

START then takes control of the user's job. Note that a run-time system may be fully trapped against all errors, except certain fatal hardware errors. In the case of START, any unexpected error prints an error message, and then exits to the system primary KBM, which is usually BASIC. If the user types control-C, however, START merely goes to the beginning and runs all over again.

The first thing START does is to grant itself privilege. In order to set the user's priority and runburst, the run-time system must have privilege. This is done by running (1,3)START.RUN. This file is created by the command:

 $PIP\ SY0:(1,3)START.RUN < 232 > /M0:1536/RTS:START\ =\ NL:$

Note that there is nothing in the file; indeed, no blocks are even allocated to hold it! The only purpose of this file is to allow START to become privileged. Since this file runs under control of START, RSTS reenters START at the location pointed to by the P.RUN psuedo-vector.

START then examines the core common area to determine if it was invoked through LOGIN. When it has completed its check, it clears the core common area. START then determines whether the user is on a PK or not, and looks up the user's PPN.

The PPN is used to scan a list built into the program that holds the PPN of all the accounts we want to do something special with. Separate lists are used for PK jobs and normal jobs, since it wouldn't do to have a batch job forced into a menu system.

The PPN list tells START what to set as the job's priority, relative to a standard priority, and what to set as its runburst. It also instructs START to either run a program for the user, or to set a new default KBM. For most regular users, we set BASIC as the new KBM, with standard priority and runburst.

START can also be used to perform other functions. At Ripon College, an electronic mail system maintains mail for each user in a file called MAIL.BOX on the system disk. If the user has just logged in, START will search for this file, and print a message telling the user that he has mail.

HOW START INTERACTS WITH MENU SYSTEMS:

When a new data entry operator logs in, LOGIN transfers him to START. START establishes itself as the default KBM for his job, looks up the name of his applications program, and runs it.

This program is typically some sort of menu program for the whole applications system. The user selects the particular option he wants, and the menu program runs the associated program. When that program terminates, RSTS automatically returns the user to START. Note that the individual applications programs need only to END. They do not have to chain back to the menu program, or invoke the START run-time system.

The menu program, though, requires some sort of EXIT option. This option can establish BASIC as the default KBM for the job, causing the user to receive the READY prompt. The option might also execute the CCL (use SYS call 14) 'BYE/Y' to invoke LOGOUT, or chain to a similar program. The advantage of sending EXIT to LOGOUT is that the user will never be allowed to reach READY, and thus is entirely under program control at all times. Any error causes the user to go back to START, and from there, to his menu.

One word of caution is in order. If the exit option from

the menu program is set up to invoke LOGOUT via the 'BYE/Y' command, you must make sure that the user is within quota before logging out. If the user is over quota, he will be trapped in LOGOUT. Of course he could just control-C LOGOUT, but that would only take him back to START, and from there, right back to the menu program he wanted to exit.

the filename contains dot, the run program option is selected. If no dot is present, the filename is taken as the name of the RTS to switch to.

In the case of a 'run program' operation, the extension may be null or an asterisk (*). Note that START does not drop privilege before executing the program. This may be

used to run a program to which the user does not normally have access. START may also enter a program at a line number other than the beginning of the program. To do this, append '/POS:n' to the filename, where n is the line number at which to enter.

The MAIL section checks for the presence of MAIL.BOX, and alerts the user to his waiting mail if the file exists. (Actual printing of the mail is handled by a separate program at Ripon.) This section is bypassed if the F.LOGI flag is clear. While this code is not applicable to other installations, it is included as a sample of things that START could do. Feel free to supply any routine appropriate to your installation.

HOW TO INSTALL START

I have provided an indirect command file to be used with ATPK to

install START. I will detail the procedure, though, so that you may modify anything required by your particular system.

START should be assembled by the MACRO assembler, with the prefix file \$COMMON.MAC. COMMON contains definitions for the EMT instructions used, and the location of interesting blocks of the low core area.

MODIFYING START:

The key to the START program is the user list. Since the internal representation of this list is somewhat complicated, the macro USER has been defined to allow definitions to be entered with a minimum of difficulty. Two lists are maintained, one for PK jobs and one for normal

The format of the USER macro is:

USER proj, prog, priority, runburst,

< filename >
Proj and prog are
the user's projectprogrammer number. As a wildcard,
an asterisk (*) may
be used for either,
or both, parts of
the PPN.

Priority is not the actual priority, but rather the number of eightpoint steps, up or down, from the standard priority. The standard is currently set to -8. Thus, a priority setting of -1 results in a job with priority of -16, and a priori-



FINAR - The financial modeling language that is used by hundreds of commercial enterprises around the world is now available to universities, colleges, and other educational institutions.

- Your finance office can use it for budgeting, cash flow forecasting, and other accounting tasks.
- Your business school can give students experience with a state of the art financial modeling system.
- Your professors can use it to do research and consultancy.

HERE IS OUR OFFER:

FINAR is normally sold for \$18,500.00 to commercial enterprises. These firms also pay an annual fee of 10% for maintenance and support of the product. We will provide your school with the software for free. Your school, on an annual basis, will pay only the \$1,850 maintenance fee normally charged to our commercial accounts.

If your school has a PDP-11/RSTS or VAX/VMS computer system and you want to take us up on this offer call or write:



James B. Hotze & Co.

6101 Southwest Freeway, Suite 406 Houston, Texas 77057 (713) 664-1172

CIRCLE 51 ON READER CARD

ty of +1 results in a job priority of zero. If priority is not given, START does not alter the user's priority. Runburst is measured in tenths of a second. The default burst is six

The filename parameter must be enclosed in angle brackets. This option will allow the programmer to select either the 'set KBM' option or the 'run program' option. If

Data base progress and management can be contained the child's play.

It's that simple. Contel's RTFILE® lets you manage the most complex databases with single keystroke responses to menu options. That's because it's relational and interactive. You need no programming experience. Proven by more than 250 customers on PDP-11 and LSI-11 systems, RTFILE can create,

expand, shrink, modify, sort, merge, display and print all your data files. And much more.

You simply select the operation from the menu, touch a key and RTFILE does the rest.

It runs on RT-11*, RT-11 emulators (including UNIX†), supports the SHARE-eleven and TSX-Plus‡ timesharing systems, and the STAR-eleven local area networking system. So it's available to any DEC

system. So it's available to any DEC user, even DEC Professional PC users.

Get complete information on RTFILE as well as our full line of advanced performance software designed specifically for DEC users. Contact Contel, the big name in small systems software. Write Loretta Kane, Contel Information

Systems, 4330 East-West
Highway, Bethesda, MD
20814, or call 301/654-9120.

CORRELATION

Now available
for RSTS*,
for RSX*, VMS* and
RSX*, VMS* and
PRO/RT-11.

Contel Information Systems

*Registered T.M. of
Digital Equipment Corp.

†Registered T.M. of AT&T

‡T.M. of S&H Computer Systems, Inc.

After compilation, build START with the Task Builder. A command file should be used to do this. When the Task Builder completes, RUN \$MAKSIL. MAKSIL will modify the command file to extend the dummy psect .99998 in order to align the task image with the top of virtual memory. Task build again, using the corrected command file. Finally, run MAKSIL again to convert START.TSK into START.RTS. MAKSIL will automatically put START.RTS into (0,1) and perform a utility ADD RTS command.

You will have to modify the installation start-up and crash start-up files to add START. For maximum performance, START should be declared permanently resident at one end of memory with the /STAY and /ADDR:nnn switches in UTILTY.

The final step is to patch LOGIN to be sure that it will exit to START. Remember that when patching LOGIN, you should have another privileged job logged in, at priority zero, in case the patch fails.

I have included a sample patch file. If you wish to use this one, enter it exactly as shown, including all characters in angle brackets. Place a carriage return at the end of each line. The patch should be applied to LOGIN after all other DEC standard and feature patches have been applied. Use CPATCH to apply the patch. Documentation on CPATCH is found in the RSTS Documentation Set, Volume 2A, System Management.

THE SOURCE OF THE RTS

```
.TITLE START, User Control RTS, 01, 14-DEC-82, D.S.H.
 . EN ABI.
          AMA, CRF
          MD, TOC
 .LIST
   This is the START run-time system. It is designed to be invoked out of LOGIN. It will look up a job's PPN, and based on this, determine what programs should
       chained to, RTS to set as job default, and priority
   and runburst to set. We will also search for MAIL.FIL, the file containing the user's mail. If we find such
   a file, we will print a notice for the user.
.NLIST
                                                 ; MAIN-LINE CODE
.PSECT MAIN
                    RO. I. LCL. CON. REL
. PSECT
          SUBR
                    RO, I, LCL, CON, REL
                                                   SUBROUTINES
                                                 ; STRING CONSTANTS
. PSECT
          $$STR
                   RO, D, LCL, CON, REL
         USRLST
                   RO, D. LCL. CON, REL
                                                 ; BEGINNING OF USER LIST
USRLST::
 . PSECT
        PKLST
                   RO, D, LCL, CON, REL
                                                 : START OF PK USER LIST
PKLST::
 . PSECT
                                                   PATCH SPACE.
                   RO, I, LCL, CON, REL
PATCH::
                                                   THIS IS EXTENDED TO
                                                   BRING THE PSUEDO-VECTOR
                                                   REGION INTO ALIGNMENT.
.PSECT .99999 RO, D, LCL, CON, REL
                                                 : PSUEDO-VECTOR AREA.
                                       ; TEMPORARY STORAGE IN LOW CORE
. ASECT
         = 1000
OURPPN::
                    .BLKW
                    . BLKW
USRTOP::
                    . BLKW
USRPTR::
F. INIT::
                    . BL KW
                    .BLKW
F. PK::
BUFTMP::
                    .BLKB
                             132.
S.PRI
S.BRST
        = 6.
E PRT
         = S.PRI - 8.
E.BRST
         = 6.
PAGE
         PSUEDO-VECTOR AREA
. SBTTL
. PSECT
          .99999
                                       : WE ARE A KBM
          WORD
                                       ; DEFAULT EXTENSION
          . WORD
                    ^RRUN
```

```
. WORD
                                          MINIMUM MEMORY SIZE
            . WORD
                     E.FIS
                                          FIS TRAP (NOT USED)
            .WORD
                     I.SYS
                                          CRASH ENTRY POINT
                                           START ENTRY POINT
            . WORD
            . WORD
                     I. NEW
                                          NEW USER ENTRY
ENTRY BY 'RUN' CMD
            . WORD
                     I.RUN
            . WORD
                     REALBD
                                           TRAP FOR BAD ERRORS
           . WORD
                     E. TBIT
                                          TRAP FOR T-BIT
           . WORD
                     E. IOT
                                          TRAP IOT INSTRUCTION
            .WORD
                     E.EMT
                                          EMT HANDLER
           . WORD
                     E. TRAP
                                           TRAP INSTRUCTIONS
           . WORD
                     E.FPP
                                          FPP (ASYNCHRONOUS) TRAPS
           . WORD
                     I. NEW
                                          CTRL-C, DO A NEW-USER
           - WORD
                    I. NEW
                                          2 * CTRL-C, DO NEW-USER
           . WORD
                                          MAX STZE
 ; We do not expect to ever see FIS trap (hardware does not
    exist at Ripon), crash or start entry, "bad errors",
T-bit, IOT, EMT, TRAP, or FPP traps. We will never issue
any of these instructions, and we will not have any
cause to load programs. All of our efforts will exist in
    fixed code in the RTS.
 . PAGE
 .SBTTL USER RECORD DEFINITION
  DSECT
 PPN::
                                          PROJECT-PROGRAMMER NUMBER
 PROG::
                    .BLKB
                                          PROGRAMMER NUMBER
 PROJ:
                     .BLKB
                                          PROJECT NUMBER
                                          PRIORITY (LO) AND FLAG (HI) RUNBURST (LO) AND FLAG (HI)
 PRIRTY::
                     . BLKW
 BURST::
                     . BL KW
 USRFIL::
                     . BL KB
                              30.
                                          FILE OR KBM NAME
USRLEN == .
 . PAGE
 .SBTTL
          Macro Definitions
 . MACRO
          CLRXRB
                    PC. CLEARX
          JSR
 . EN DM
          CLRXRB
 . MACRO
          CLRFRQ
          JSR
                    PC, CLEARF
          CLRFRQ
 . EN DM
USR$BK
          = USRLST
 MACRO
          USER
                    P, PN, PRI, BURST, FIL
$$$$$0
          IDN
                    /PN/
 .IF
           .BYTE
                     0377
           BYTE
                    PN
                                       : PROJECT PROGRAMMER
 . ENDC
          TDN
          .BYTE
                    0377
.IFF
                                       ; NUMBER
           BYTE
                   P
. ENDC
                                       : NO PRIORITY GIVEN?
.IF
          B
                    PRT
          .BYTE
                   S. PRI
                                       ; STANDARD PRIORITY
                   0
          .BYTE
                                       ; PRIORITY WAS GIVEN
.IFF
           BYTE
                    PRI # ^010 +
          . BYTE
                    ^0377
. ENDC
                                       ; NO BURST GIVEN?
                   BURST
          BYTE
                   S. BRST
                                       : STANDARD BURST
          .BYTE
.IFF
                                       ; BURST GIVEN
                   BURST
          .BYTE
          .BYTE
                    0377
.ENDC
          . ASCII
                  /FIL/
                                       : FILENAME OR KBM NAME
. REPT
         $$$$$0 + USRLEN - 1 -
                                       ; TERMINATOR
          .BYTE
                   0
. EN DR
          = $$$$$0 + USRLEN - 1
                                       ; SEMI-COLON TERMINATE FILENAME
          .ASCII /;/
. EN DM
         USER
. PAGE
.MACRO
         UUO
                   FUNC
         MOVB
                   # FUNC. FIROB + FOFUN
          .UUO
. ENDM
.MACRO
         ADDSTR
                   STREXP
                                       ; ADD A STRING CONSTANT TO OUR LIST
PSECT
         $$STR
$$$$$0
          . BI.KW
         .ASCIZ STREXP
$$$$$1
  = $$$$$0
          .WORD $$$$$1 - $$$$$0 - 2
          EVEN
. ENDM
         ADDSTR
```

: OBSOLETE P. ISIZ

. WORD

We Have the Prescription for Your Ailing Computer System



Tymshare's computer maintenance organization is just what the doctor ordered.

We're an independent service group offering you fast response time and rapid, reliable repair—24 hours a day, seven days a week.

Maximum Uptime for the Lowest Possible Cost

Our highly skilled, experienced service representatives quickly and accurately diagnose your

system's problems—to maximize your computer uptime.

And our custom preventive maintenance program keeps your installation healthy—to ensure continuing efficient operation at the lowest possible cost.

For more information on how Tymshare's maintenance programs can best fit your needs, call us today at the regional office nearest you.

COMPUTER SYSTEMS and SUPPORT

EAST: 10111 George Palmer Hwy.; Bowie, MD 20716; (301) 459-8363
CENTRAL: 4513 Lincoln Ave., Suite 106; Lisle, IL 60532; (312) 964-0248 or 964-0362
SOUTHERN: 6333 Harry Hines Blvd; Dallas, TX 75235; (214) 638-2130
SOUTHWEST: 11050 E. Artesia Blvd., Suite D; Cerritos, CA 90701; (213) 402-1114
WEST: 39100 Liberty Street; Fremont, CA 94538; (415) 794-2490 or (415) 794-2719

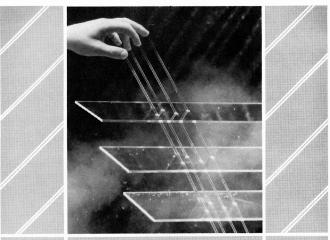


```
.MACRO TYPE
                        STRING
 ADDSTR
             <STRING>
 . PSECT
            MAIN
             JSR
                         R5. TYPER
             . WORD
                         $$$$$0
 . EN DM
             TYPE
 .MACRO LINE
                        STRING
TYPE
             <<13.><10.>@'STRING'@<13.><10.>>
. ENDM
            LINE
.MACRO ERROR
                                                 ; PRINT A RSTS/E ERROR MESSAGE
             MOVB
                         FIRQB, FIRQB + FQERNO
                        UU.ERR
#28., FIRQB + 2
R5, TYPER
            UUO
            MOV
            JSR
             . WORD
                        FIRQB + 2 <<13.><10.>>
             TYPE
 . EN DM
            ERROR
 . PAGE
SBTTL
            SUBROUTINES
. PSECT SUBR
            TYPE A STRING. ON ENTRY, R5 WILL POINT TO THE STRING. IT WILL BE OF THE FORM:

LENGTH IN BYTES (ONE WORD)

CHARACTERS (STORED 1 TO A BYTE)

BYTE ZERO TO MARK THE END
            IF NECESSARY, THE STRING WILL BE ROUNDED OUT TO THE NEXT WORD BOUNDARY WITH ANOTHER ZERO.
TYPER::
            PUSH
                       <RO,R1,R2,R3>
            CLRXRB
                        (R5)+, R0
(R0)+, R1
            MOV
            MOV
            MOV
                        #BUFTMP, R2
            MOV
                       R1, R3
(R0)+, (R2)+
10$:
                       R3, 10$
#BUFTMP, XRB + XRLOC
R1, XRB + XRLEN
R1, XRB + XRBC
           SOB
           MOV
           MOV
           MOV
            .WRITE
           POP
                       <R3,R2,R1,R0>
           RTS
```





Summoning Solutions DIGICALC™ v1.2

Electronic Spreadsheet for financial management available for DEC SYSTEMS, VAX VMS, PDP-11-RSTS/E and Professional 350*

Quick answers for the DP professional. Save programming effort. Call WHY Systems today for the perfect power of the all new DIGICALC™ v 1.2.

*DEC, VAX, VMS, PDP-11-RSTS/E and Professional 350 are all trademarks of DIGITAL EQUIPMENT CORPORATION.



Seeking Solutions ... mankind Asks WHY

16902 Redmond Way, Redmond, WA 98052 U.S.A. (206) 881-2331

```
#XRB, RO
#7, R1
             MOV
             MOV
                        (RO)+
  10$:
             CLR
                       R1, 10$
<R1,R0>
             SOR
             POP
             RTS
  CLEARF:
                                  ; CLEAR OUT FIRQB
             PUSH
                       <RO,R1>
             MOV
                       #FIRQB, RO
                       #20, R1 (R0)+
             MOV
                       R1, 10$ <R1, R0>
            SOR
             POP
  . PAGE
            USER CODE LIST
  . SBTTL
  .PSECT USRLST
     Place the user-list here. Users should have
     one of these forms:
            To run a program:
            USER proj, prog, pri, burst, (filnam.ext)
            To set up job's default KBM:
            USER proj, prog, pri, burst, <kbm>
     The angle brackets around the filename or kbm name
     are required.
     Priority does not represent an actual priority, but
     rather, the number of steps up or down from the standard priority. If not included, the user's priority
     is left unchanged, barring error.
     If burst is not included, the user's runburst is assigned
     to a standard value.
     The following is shown as a sample:
           USER
                     200. *. -1. 6. <GAMES>
    This will place all 200 group users into the
    WEM GAMES. They will run with priority -16, with a runburst of 6. Note that a dot is required in the filename if you wish to run a file, rather than to set a job's default keyboard monitor.
    The angle brackets are required around the names of the file to run or KBM to set. For a file run, you may also specify the /POS:nnn switch. Do not
    specify any other switches in the name.
    All definitions must come before the *, * entry.
.RADIX
.NLIST ME. MEB. COM
USER
           1,101,1,,<BASIC>
USER
          1,150,1,, <BASIC>
1,151,1,, <BASIC>
USER
           1,*,,,<BASIC>
USER
          7,0,0,,<MEC:MAMENU.*>
7,1,0,,<MEC:MAMENU.*>
USER
          7,2,0,,<MEC:MAMENU.*>
7,3,0,,<MEC:MAMENU.*>
USER
USER
USER
           7,100,0,,<ADM:ADMENU.*>
          7,101,0,,<ADM:ADMENU.*>
7,102,0,,<ADM:ADMENU.*>
USER
USER
USER
           7,103,0,,<ADM:ADMENU. *>
           50,2,0,,<SY:#MENU.*>
USER
USER
           50,3,0,, (SY: MENU. TSK>
USER
           144, *, 0,, < PRIVAT: (144, 0) START. *>
USER
           151,1,0,,<ALUMNI: #ALMENU. *>
           151,2,0,,<ALUMNI:#ALMENU.*>
151,3,0,,<ALUMNI:#ALMENU.*>
HSER
USER
USER
           151,4,0,,<ALUMNI:#ALMENU.*>
USER
           152,1,0,,<PARENT: #PAMENU. *>
          152,2,0,,<PARENT:#PAMENU.*>
152,3,0,,<PARENT:#PAMENU.*>
152,4,0,,<PARENT:#PAMENU.*>
USER
USER
USER
USER
          155,1,0,,<REGIS:#REMENU.*>
           155,2,0,, < REGIS: #REMENU. *>
USER
USER
           155,3,0,,<REGIS:#REMENU.*>
USER
          200, *, 0,, <SY0: (1,3) GAMES. *>
.LIST
USER
          *, *, 0,, <BASIC> ; THIS MUST BE THE LAST USER!
. PAGE
.SBTTL User list for PK devices.
```

; CLEAR OUT XRB

CLEARX:

PUSH

<R0,R1>

```
This section follows just the same format as the USRLST section, but it applies to LOGINs on a psuedo-keyboard instead.
                                                                                                  CLRFRQ
                                                                                                  CLRXRB
                                                                                                           #^R.IN, FIRQB + FQNAM1 + 0
                                                                                                  MOV
                                                                                                  MOV
                                                                                                           #^RIT., FIRQB + FQNAM1 + 2
                                                                                                                                      ; SET PRGM NAME '.INIT.'
                                                                                                  . NAME
        The *, * entry must still be last.
                                                                                        TTY::
.PSECT PKLST
                                                                                                  CLR
                                                                                                  CLRFRQ
         1,101,1,, (BASIC)
USER
        1,150,1,, (BASIC)
1,151,1,, (BASIC)
                                                                                                  CLRXRB
USER
                                                                                                  MOVB
                                                                                                           #^0377, FIRQB + 5
USER
                                                                                                                                      ; READ TERMINAL SETTINGS
                                                                                                  UUO
                                                                                                          #8., FIRQB + 24
STAT
                                                                                                                                      ; CHECK FOR PK TERMINAL
                                                                                                  CMP
         *,*,0,,<BASIC>
USER
                                                                                                  BNE
. PAGE
                                                                                                           #-1, F.PK
                                                                                                                                      ; IT WAS A PK
                                                                                                  MOV
. SBTTL
        MAIN CODE
                                                                                        STAT::
                                                      ***
                                                                                                  CLRFRO
         ...
                                                      ***
                                                                                                  CLRXRB
                     MAIN CODE
         ***
                                                                                                                                      ; RETURN JOB STATUS
                                                                                                  .STAT
                                                                                                  MOV
                                                                                                           XRB + 10, OURPPN
                                                                                                                                      ; OUR PPN
                                                                                                  MOV
                                                                                                           #USRLST, USRTOP
                                                                                                                                      : USE NORMAL USER LIST
.LIST
                                                                                                  TST
                                                                                                           F.PK
. PSECT
        MAIN
                                                                                                  BEO
                                                                                                           10$
                                                                                                                                      : BR IF NOT ON A PK
                                                                                                                                      ; USE SPECIAL PK USER LIST
                                                                                                           #PKLST, USRTOP
                                                                                                  MOV
.NLIST
        BEX
.RADIX
                                                                                                           USRTOP, RO
                                                                                                                                      : POINTER TO USRLST
                                                                                        10$:
                                                                                                  MOV
                                                                                                           OURPPN, R1
                                                                                                                                      : PROGRAMMER
                                                                                                  MOVB
T. NEW::
                                                                                                           OURPPN + 1 , R2
GETPRV::
                                             ; ARE WE ALREADY INIT-ED?
         TST
                  F. INIT
                                             ; BR IF YES.
                                                                                        LOOPU::
         BNE
                  START
                                                                                                  MOVB
                                                                                                           PROG(RO), R3
                                                                                                                             ; PPN KEY TO USRLST
                                                                                                  MOVB
                                                                                                           PROJ(RO), R4
         CLRXRE
         CLRFRQ
                                             ; CLOSE CHANNEL
                                                                                                  CMPB
                                                                                                                             ; WILDCARD PROGRAMMER?
                                                                                                           #377, R3
                  #RSTFQ, FIRQB + FQFUN
         MOVB
                                                                                                                             ; YES
                                                                                                 BEQ
                                                                                                          10$
R3, R1
                  #15., FIRQB + FQFIL
         MOVB
                                                                                                                             ; PROGRAMMERS MATCH?
                                                                                                 CMPB
         CALFIP
                                                                                                  BEQ
                  #259., FIRQB + FQPPN
# RSTA, FIRQB + FQNAM1
                                                                                                                             ; NO
                                            ; PPN 1,3
                                                                                                 BR
                                                                                                          NXTUSR
         MOV
         MOV
                                                                                                  СМРВ
                                                                                                                             ; WILDCARD PROJECT?
                  #^RRT, FIRQB + FQNAM1 + 2
         MOV
                                                                                                                             ; YES
                  # RRUN, FIRQB + FQEXT
                                                                                                 BEO
                                                                                                          FOUND
         MOV
                                                                                                          R4, R2
                                                                                                                             ; PROJECTS MATCH?
                   #"SY, FIRQB + FQDEV
                                                                                                 CMPB
         MOV
                                                                                                          FOUND
                                                                                                                              YES
                  #^0176000, FIRQB + FQDEVN
         MOV
                                             ; RUN SYO:[1,3]START.RUN
                                                                                                                             ; NO
         .RUN
JMP
                                             ; ERROR.
                                                                                        NXTUSR::
                                                                                                                            ; NEXT USER DESCRIPTION
                                                                                                  ADD
                                                                                                           #USRLEN, RO
I.RUN::
                                                                                                                             ; AND TRY AGAIN
                                             : NOT CALLED BY LOGIN
                                                                                                          LOOPU
         CLR
                                                                                                 BR
                  F.LOGI
                                                                                        FOUND::
                  XRB + 2, #^RSTA
         CMP
                                                                                                                            ; SAVE THIS FOR LATER.
                                                                                                          RO, USRPTR
                                                                                                 MOV
                  GETPRV
         BNE
                                                                                         . PAGE
                                             ; MUST HAVE BEEN
                  XRB + 4, # RRT
GETPRV
          CMP
                                              ; CALLED BY OURSELVES.
         BNE
                                                                                         .SBTTL CHECK FOR MAIL FILE
                  CORCMN, #256.*'L + 5
          BNE
                   START
                                                                                           Now we will look for MAIL.BOX in current account.
                   CORCMN + 2, #"OG
          CMP
                                                                                            If we find it, we will send a message to the user's
                   START
                                                                                            console.
                   CORCMN + 4, #"IN
          CMP
          BNE
                   #-1, F.LOGI
                                                                                        MAIL::
                                                                                                                                     ; TEMP PRIV ACTIVE?
                                                                                                          F.LOGI
START::
                                                                                                                                     ; NO, SKIP MAIL SEARCH.
                                                                                                 BEQ
                                                                                                          SETPRI
                                              ; INIT HAS BEEN DONE
         MOV
                   #-1, F.INIT
          .TTECH
                                              ; RESTART OUTPUT
                                                                                                 CLRFRO
          . TTRST
                                                                                                                                              ; REQUIRED
                                                                                                 MOV
                                                                                                          #-1, FIRQB + 4
          CLR
                   CORCMN
                                                                                                                                               ; CURRENT PPN
                                                                                                          FIROB + FOPPN
                                                                                                 CL.R
                   CORCMN + 2
                                                                                                          #^RMAI, FIRQB + FQNAM1
                                                                                                                                               ; NAME, PART 1
          CI.R
                                                                                                 MOV
                                                                                                          #^RL , FIRQB + FQNAM1 + 2
#^RBOX, FIRQB + FQEXT
                                                                                                                                               : NAME. PART 2
                                                                                                                                               ; EXTENSION
                                                                                                 MOV
 RTS::
                                                                                                           #"SY, FIRQB + FQDEV
                                                                                                                                               : DEVICE NAME
                                                                                                 MOV
          CLRFRQ
                                                                                                 UUO
                                                                                                          UU.LOK
          CLRXRB
                   #^RSTA, FIRQB + FQNAM1 ; OUR NAME IS 'START'
#^RRT, FIRQB + FQNAM1 + 2
                                                                                                 TSTB
                                                                                                          FIRQB
          MOV
                                                                                                 BNE
          MOV
                                              ; SET START AS OUR
                   #-1, FIRQB + FQEXT
          MOV
                                                                                                          <You have MAIL. Type 'MAIL' to read it.>
                                                                                                 LINE
                                              : DEFAULT KBM
                                                                                         . PAGE
                                                                                                 SET PRIORITY AND RUNBURST
```

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) 542-7008

```
SETPRI::
                                                                                            E.FIS::
                                              : CHECK PRIV BITS
          BIT
                   #JFSYS + JFPRIV. KEY
                                                                                                     LINE
                                                                                                               <?Unexpected FIS trap - aborting>
          BEQ
                                                DON'T CHANGE PRIORITY/BURST
                                                                                                     JMP
                                                                                                              REALBD.
                                                  UNLESS PRIV.
          CI.RFRO
                                                CLEAR OUT FIRQB
                                                                                            BADERR: :
                   #377, FIRQB + 4
          MOVB
                                                DO UNTO OUR JOB
                                                                                                     MOV
                                                                                                              #USRSP. SP
                   USRPTR, RO
          MOV
                                                POINTER TO USER BLOCK
                                                                                                     ERROR
                                                                                                     LINE
                                                                                                               <? Asynchronous BAD ERROR trap - aborting>
                                                PRIORITY + FLAG
                   PRIRTY(RO), R1
          MOV
                                                                                                     JMP
                                                                                                              REALBD
                   #177400, R1
                                                FLAG BITS SET?
          BIT
          BEO
                   20$
                                                NOPE.
                                                                                            E. TBIT::
          MOVB
                   #377, FIRQB + 5
                                                CHANGE PRIORITY
                                                                                                     LINE
                                                                                                              <? T-bit trap - aborting>
          MOVB
                   R1, FIRQB + 6
                                                OUR NEW PRIORITY
                                                                                                              REALBD
 20$:
          MOV
                   BURST(RO), R1
                                                RUNBURST + FLAG
                                                                                            E.IOT::
                                                FLAG BITS SET?
          BIT
                   #177400, R1
                                                                                                     LINE
                                                                                                              <?IOT instruction - aborting>
          BEQ
                   25$
                                                NOPE.
                                                                                                     JMP
                                                                                                              REALBD
                   #377, FIRQB + 7
                                                CHANGE RUNBURST
          MOVB
                   R1, FIRQB + 10
                                                NEW RUNBURST
          MOVB
                                                                                            E.EMT::
                                                                                                    LINE
                                                                                                              <? Non-monitor EMT instruction - aborting>
 25$:
          CL RB
                   FIRQB + 11
                                                NO CHANGE TO MAX SIZE
                                                                                                    JMP
                                                                                                             REALBD
                   UU.PRI
                                                CHANGE PRIORITY / RUNBURST
          UUO
 . PAGE
                                                                                           E. TRAP::
         SET UP FILE TO RUN OR KBM
 . SBTTL
                                                                                                    LINE
                                                                                                             <? TRAP instruction - aborting>
                                                                                                    JMP
                                                                                                             REALBD
 OUT::
         CLRFRQ
                                              ; CLEAN FIRQB
                                                                                           E.FPP::
                   #RSTFQ, FIRQB + FQFUN
                                                RESET A CHANNEL
         MOVB
                                                                                                    LINE
                                                                                                             <? FPP asynchronous trap - aborting>
                   #15., FIRQB + FQFIL
                                                RESET CHANNEL 15
          MOVB
                                                                                                    JMP
                                                                                                             REALBD
                                                CALL TO FIP
          CALFIP
                                                                                                    I.NEW
         CLRFRO
         CLRXRB
                                              ; MAKE POINTER TO START
         MOV
                   USRPTR. RO
                                                                                           A COMMAND FILE TO GENERATE THE RTS.
                   #USRFIL, RO ; OF RO, XRB + XRLOC
#USRLEN - USRFIL, XRB + XRLEN
                                              ; OF STRING TO SCAN.
          ADD
         MOV
                                                                                                                                                     START.CMD
         MOV
                                                                                           START.CMD = KB:
                                                                                                                                                     START1.CMD
         MOV
                   #USRLEN - USRFIL, XRB + XRBC
                                                                                           START/-HD/SQ, START/-WI, START=START
                                                                                                                                                     TKB @START1
                                              ; CONVERT FILE
         .FSS
                                                                                                                                                     UT REMOVE START
          TS TB
                                                                                           PAR=START: 160000:020000
                                                                                                                                                     PIP [0,1]START.RTS/DE
         BNE
                   10$
                                                                                           STACK=2048
                                                                                                                                                     RUN $MAKSIL
                                                                                           EXTSCT=.99998:000000
                                                                                                                                                     START/RTS
                                              ; FLAG WORD
                   XRB + 10, R1
         MOV
                                                                                                                                                     START, TSK
                   R1, #10
KBM
                                              : WAS DOT FOUND?
         BIT
                                              ; NO, ASSUME KBM SWITCH
         BEQ
                                                                                           SYO:(1,3)START.RUN<0>/RE
                                                                                                                                                     YES
         BNE
                   FILE
                                                                                           SYO: (1,3)START. RUN/DE
                                                                                                                                                     START.STB
                                                                                           SYO: (1,3)START.RUN/PR:232/MO:1536/RTS:START=NL:
         LINE
                   <? Couldn't do FSS properly>
10$:
                                             ; TRY *, * ENTRY?
                                                                                           MACRO START, START=$COMMON, START/N:TOC/L:TTM
         BR
                  BADEX
                                                                                           TKB @START
                                                                                                                                                     START. CMD, START1. CMD/DE
                                                                                           RUN $MAKSIL
                                                                                                                                                     START, OBJ. START, TSK/DE
                                                      ; WAS FILE TYPE * ?
                                                                                           START/RTS
FILE::
         BIT
                   XRB + 10, #40
                                                                                                                                                     START.STB, START.LST/DE
                                                       , NO, RUN PROGRAM
         BEO
                   10$
                                                                                          START. TSK
                  #-1, FIRQB + FQEXT
         MOV
                                                       : RUN A FILE
10$:
         - RUN
                                                      ; DUMP TO BASIC
         BR
                                                                                           A REQUIRED PATCH TO LOGIN
KBM::
                                                                                            #H/21/V(cr>
         MOV
                  #-1, FIRQB + FQEXT
                                                      : SET AS JOB DEFAULT KBM
                                                                                           2!<tab><tab>PROGRAM<tab><tab>: LOGIN.BAS<cr>
         .RTS
                                                                                           #H/5210/V(cr)
         BR
                  BADEX
                                                                                           5210<tab>I$=SYS(CHR$(0%))+SYS(CHR$(2%)) &<cr>
                                                                                           #4AV(cr)
- PAGE
                                                                                           <tab>\ GOTO 32650 UNLESS N1% &<er>
         CAN'T SWITCH TO RTS OR EXECUTE FILE
. SBTTL
                                                                                           #I(cr>
                                                                                           <tab>\ GOTO 32650 &<cr>
<esc>#H/SPAWNED ENTRY POINT/V<cr>
BADEX::
         MOV
                  USRTOP, RO
                                                                                           29000 < tab>! SPAWNED ENTRY POINT. & < cr>
                  #-1, (RO)
                                    : CHECK FOR *. * ENTRY.
10$:
         CMP
                                                                                           #13AV(cr)
         BEQ
                                                                                           <tab>\ I$=SYS(CHR.6$+CHR$(-21%)) &<cr>
         ADD
                  #USRLEN, RO
                                                                                           #T(or)
         BR
                  10$
                                                                                           <tab>\ FLAG.SPAWN% = -1% &<er>
                                                                                           <esc>#3H/32700/V<cr>
                                    ; WAS *, * WHAT WE JUST FAILED?
20$:
         CMP
                  RO. USRPTR
                                                                                           32700<tab>CLOSE I$ FOR I$=1$ TO 12$ &<cr>
                                    ; YES, JUST EXIT
                  REALBD
         BEQ
                  RO, USRPTR
         MOV
                                                                                           <tab>\ I$=SYS(Q$) &<cr>
         JMP
                                                                                           #I(cr>
                                                                                           <tab>\ GOTO 32710 UNLESS FLAG.SPAWN% &<cr>
REALBD::
                                                                                           <esc>#3AV<cr>
                  #JFSYS+JFPRIV, KEY
                                             ; DO WE HAVE ANY PRIV ?
         BIT
         BNE
                                             ; NO, DON'T SET PRIORITY.
                                                                                           (cr)
                                                                                           #I(cr>
                                                                                           (cr)
         CLRFRQ
                                                                                           32710<tab>T$ = SYS( CHR$(6$) + CHR$(26$) + CVT$$(0$) ) &<cr>
                  #-1, FIRQB + 4
                                                        PRIORITY AND
         MOV
                                                                                          <tab>\ PPN% = SWAP%( CVT%%( MID( T%, 21%, 2%))) &<cr><tab>\ IF PPN% = 0% THEN &<cr>
         MOVB
                  #E. PRI, FIRQB + 6
#377, FIRQB + 7
                                                      ; RUNBURST, AND
         MOVB
                                                                                           <tab><tab>JOBNUM% = ASCII( MID( T$, 1%, 1%)) / 2% &<cr>
         MOVB
                  #E.BRST, FIRQB + 10
                                                                                          UUO
                  UU.PRI
                                                      : EXIT TO PRIMARY KBM
10$:
         . EXIT
                                                                                          32720<tab>ON ERROR GOTO 32740 &<cr>
<tab>\ T$ = SYS( CHR$(8$) + "LOGIN") &<cr>
<tab>\ T$ = MID( SYS( CHR$(6$) + CHR$(-10$) + "START"), 7$, 4$) &<cr>
<tab>\ T$ = SYS( CHR$(9$) + T$ + CHR$(-255$)) &<cr>
<tab>\ T$ = SYS( CHR$(9$) + T$ + CHR$(255$)) &<cr>
<tab>\ T$ = SYS( CHR$(9$) + T$ + CHR$(255$)) &<cr>
<tab>\ T$ = SYS( CHR$(9$) + T$ + CHR$(255$)) &</a>
. PAGE
        UNEXPECTED ERROR TRAPS
.NLIST ME, MEB
                                                                                          <tab>\ GOTO 32750 &<cr>
I.SYS::
                                                                                          32740<tab>RESUME 32750 &<cr>
         . TTRST
          TTECH
                                                                                          32750<tab>T$ = SYS( CHR$(9$)) &<cr>
                  <START may not be SYSTEM PRIMARY RTS - aborting>
         LINE
                                                                                          <tab>\ GOTO 32767 &<cr>
         CL RERO
                                                                                          <esc>#EX<cr>
                                    ; SHUT DOWN SYSTEM
         uuo
```

The Single Source for Digital Products Service

DEC Module Repair, Exchange or Replacement

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

Visit us at Booth No. 107

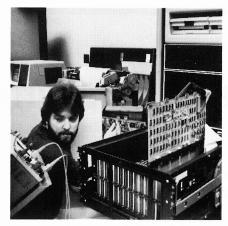
DEXPO® West 83

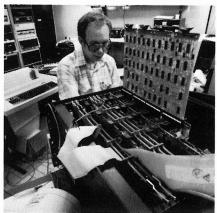
The 4th National DEC-Compatible Industry Exposition

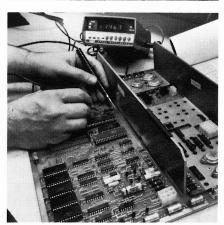
Las Vegas Convention Center October 23-26, 1983

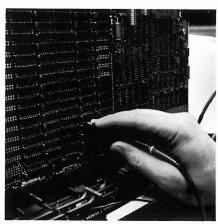
DIGITAL PRODUCTS REPAIR CENTER

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

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

RELIANCE ELECTRIC

IBM's 1403 PRINTER REVITALIZED FOR VAX

By Roger E. Holmes, 7900 Cowan Ave., Los Angeles, CA

It is not uncommon for a product of the past to come to the rescue of newer, high-technology devices. This proved true when a remanufactured printer, out of production for more than a decade, solved the quality, volume and dependability problems of a communications engineering and market research firm in Dallas, Texas.

Compucon, Inc., a subsidiary of the A.C. Nielson Company of Northbrook, Illinois, which is better known for its television ratings, operates three Digital Equipment VAX-11/780 computers.

To keep up with demand for printout, Compucon added, over the years, five printers — two 300 lines per minute, a 600 lpm printer and two 300 lpm printer/plotters.

Problems accumulated along with the printers, according to Richard Dowd, Compucon vice president, Corporate Systems and Administration.

"We had frequent breakdowns, print quality was not adequate and we needed faster throughput to handle our workload."

When faced with the problems of present technology, responsible executives such as Dowd often look back into their own experience for help.

"I had developed quite a respect for the dependability, rugged construction and consistent print quality of the IBM 1403 printer while employed by the data processing departments of two other companies," Dowd said.

Unfortunately, 1403 line impact printers had been out of production for more than a decade when Dowd decided to return to them.

The 1403 printer series was introduced by IBM for use with the IBM 1401 large-scale data processor. When IBM introduced newer computers and minicomputers it also introduced newer printers that matched the configuration and decor of the companion products. As a result, production of the 1403 was phased out about 1970. Remanufactured models remained popular, however.

Computed purchased a remanufactured 1403-N1, which has a print speed of 1100 lines per minute, and a printer controller from Spur Products Corporation, Los Angeles. Spur Products makes the only controller that enables 1403 printers to be driven by non-IBM computers and minicomputers.

Operation of the printer can be switched to either of two VAX computers by a remote control unit. "We switch the 1403-N1 to the computer with the biggest printing run," Dowd said. While one computer is running the 1403 printer we can be setting up the other system or running one of the slower printers, he said. Generally, the only other printers being used are the printer/plotters, which produce

engineering graphics.

Computed also makes use of the interchangeable type train feature of the 1403. For full printing speed Dowd uses a standard 48-character type set. For engineering reports he selects a train with engineering symbols. For other reports he selects a set with caps and lower-case letters.

The printer is the object of return revivals for some very practical reasons, Dowd says.

All electronic products are subject to infancy failures. Electromechanical products such as high-speed impact printers are even more prone to fail or require adjustments early in their use. Since the entire data processing system is useless without a printer, the failure of a new printer is catastrophic, regardless of the amount of new technology that went into its design and construction."

The opinions that management, customers and other important groups form of the data processing department are often based on the output of the printer, Dowd reminded.

"If an important report is delayed because of printer failure, management assumes negative thoughts about the investment in the company's data processing equipment and personnel.

"Print quality is also the basis by which the EDP department is judged by management. Since computer printouts are often the only representation of a company seen by investors, banks, regulatory agencies, customers and other important outsiders, they often are the sole basis for judging the quality of the entire company.

"Many important people know little about EDP technology, but they know bad printing when they see it."

In recent years economics has been added to the list of factors extending the life of the 1403 printer. As IBM promoted use of its new printers, the availability of used printers grew, greatly dropping the price. It became such a buyer's market that Spur Products, which previously specialized in making a controller that prolonged and broadened the life of the 1403 printer, expanded into remanufacturing, selling and servicing them at a much lower cost than IBM, which had previously dominated the activity.

Because of the low printer prices many smaller companies, previously limited to slower printers with poorer print quality, have been upgrading their systems by adding or changing to a 1403.

While minicomputers enabled small companies to compete with larger companies by adding the benefits of data processing, the availability of printers from an earlier era at low cost enables them to match the printing quality and dependability of their larger competitors.

Do these tough demands blow your graphics apart?

1. Power & speed!

Dual processor power. Rapid area fill and draw.

2. Simplicity & compatibility.

ANSI X3.64. TEK 4010, 4014 and 4027 compatible.

3. Full featured graphics.

Host graphics support and terminal primitives.

4. Dot addressability.

Vector drawing and point plotting.

5. Brilliant raster color.

64 color palette.

6. Affordability.

\$1995



PRICE/PERFORMANCE BREAKTHROUGH! The new Intecolor 2427 advanced graphics terminal puts the pieces together.

Even managers explode, evaluating terminals to satisfy all these demands. And at a reasonable price.

Solution: Try the Intecolor 2427, for only \$1995.

Capitalizing on over 10 years of experience in the design and manufacture of color graphics terminals, Intelligent Systems has put it all together in the Intecolor 2427.

Check out the features. Compare the price. Then contact Marketing Communications at 404/449-5961. TWX 910 766 1581.

*Single-evaluation unit. U.S. domestic only. To evaluate the 2427 at this 100-piece price, your order must be received by 11/30/83. Single piece price is \$2695.



1. Dual Processor Power: 16-bit Z8002 graphics processor for rapid area fill and polygon drawing. 8-bit 8085 independent alphanumeric terminal functions processor. Plus independent keyboard processor. 2. Convenience/Compatibility: Extended ANSI X3.64 control system with VT100 features. Integrated graphics and alphanumeric menu set-up. Compatible with TEK 4010, 4014, 4027 terminals and Plot 10. Interactive color graphics. Easy conversion to color.

Complete Graphics Capabilities: Supported by popular host graphics packages. Terminal resident TEK 4027-like graphics primitives.
 Full featured bit-map graphics.

4. Dot Addressability: Absolute addressing of points for point plotting with complex graphics. 560H x 288V. Full three-bit pixel array. 80 characters by 24 line alphanumeric display.

5. Raster Color: Brilliant 64-color palette with color mapping. Eight displayable colors. Self converging, 13" diagonal CRT.

PLUS: Enhanced Text Features: Five character sets (two are user-definable). Two vector character sets. ANSI X3.64: Industry standard control sequences. Easy to implement. Two Full Pages of Screen RAM: Simplifies graphics programming. 72 Function Keys (optional): Speeds up programming. Clean, Simple Design: Assures reliability, cost efficiency. VT100 Features: Familiar DEC functions.

MAKE YOUR PDP 11/34 WORK HARDER

By Austin Kinsella, Regional Technical College, Kilkenny Road, Carlow, Ireland

The Regional Technical College in Carlow, Ireland, is a third-level college with courses mainly of two or three years duration in technical and business subjects. As part of their courses, many of the students use our PDP 11/34 RSTS/E system for the development of simple BASIC programs, and also for running application programs statistics, cross-assemblers, etc. One group of students, on a two-year commercial programming course, uses the system much more intensively, as they develop COBOL, RPG, and large BASIC programs. The general group, consisting of about 500 accounts, takes up about 40 per cent of the system usage; the special two-year group, about 80 accounts, the remaining 60 per cent of system usage.

Inevitably, as our number of terminal lines grew from eight when the system was installed in 1979 to our current figure of 33 (about 25 currently active), with a correspondingly higher usage (about 950 connect hours/week), the system slowed down. Figure 1, an extract from a STATUS report, shows an inordinate amount of lost time, a consequence of the heavy swapping load, with only two or three out of 30 jobs resident in memory at any time.

We considered several ways of alleviating swapping, either by speeding it up with a swapping disk, or eliminating it, with more memory. Our preferred solution was the faster 11/44 with a megabyte of memory, but we could not afford it. Another possibility was a second-hand 11/70, but again the prices were outside our budget. Within our budget range, we could afford a high-speed dedicated swapping disk, but the technology was old, and the more modern semi-conductor memory versions seemed more promising, with no moving parts and electronic speeds. But even at electronic speeds, the load on the monitor to organize swapping, and pass it through the file processor bottleneck, would remain. Complete elimination still seemed the better alternative, so we looked at the various ways of increasing the memory addressing capabilities of the 11/34. Of these, we selected ABLE Corporation's ENABLE, and purchased this, with a megabyte of Dataram 22-bit memory. from Sintrom Electronics, of Reading, England.

We had no problems in the installation of the ABLE unit. We had to move several boards, so that all direct-memory access devices were ahead of the ABLE. To use existing 18-bit memory, it must be installed in a separate backplane following the ENABLE backplane. This meant that we had to rearrange our backplanes to get the 4-slot in

the middle. The board layout, before and after, is shown in Figure 2. The jumpers on the ENABLE backplane do not go directly into the backplane, but into the ENABLE components.

We had done a new system generation before moving the boards, so after testing with the old system (which, without the ENABLE patches, does not recognize the increased memory), we rebooted with the new system — and it worked! Actually, it took two tries as we had incorrectly set the switches on the new memory board, but use of the ABLE diagnostics quickly isolated the problem.

Our new monitor takes advantage of the increased memory. Previously, anything that made the monitor larger was out of the question, but now we not only have large file and resident library support, we even have Ctrl/T support. The new monitor takes 41 KW, compared with 32 KW for the previous one. Our run-time systems and libraries are permanently loaded, and we have increased XBUF from 12KB to 100KB, with the result that directory hits are up from less than 75 per cent to 99 per cent. We are experimenting with data caching. At system start-up, we have a small program which opens our most commonly used programs as files, and then sleeps, giving faster access to them under the large files system. With our old monitor, we could not use the faster-running CSPCOM CUSPS, because of the extra space they take on disk, as we are very short of disk space. Now we have them built with CSPCOM against the Resident Library CSPLIB (RSTS PRO September, '81), getting both space and speed advantages. We are testing various methods of task-building, but it looks as if building against the resident library will be the best, and since we do about 40 task-builds a day, we hope for still further improvements when we get it set up. Figure 3 is an extract from a STATUS report taken after the change. On our weekly accounting report the chargeable CPU time has doubled for the same connect time. During the few weeks since we have installed ENABLE, we have had no strange system crashes, no unusual problems with the system, and no peculiar messages at system startup.

We have been enamoured of DEC and RSTS since we first received the 11/34. We wanted to remain with DEC, and if possible RSTS, but we were swamping our machine and could not afford a larger one. Hopefully, by the time we outgrow the increased capacity of our ENABLE-enhanced 11/34, we will be able to afford a bigger, faster, DEC machine — maybe even with RSTS!

26-Jan-	-83 15:	09:56	RSTS	V7.0-07	Numb	erCrunc:	her	10	Errors		
29/32	Jobs	43% Us	er run	ning	57%	Fip nee	ded	910.4	Second	S	
0%	Idle :	27% SY	S char	ged	6%	Fip in	use	71%	Cache	hits	
28%	Lost	2% SY	Sunch	arged	15%	I/O ser	vice	3%	Cache	CPU	
	17	7.3 Ch	ar/sec	out	8.7	Char/se	e in	66	Min smi	l buf	f
	Access						Blocks				
Disk	per sec	Total	Reads	Direct	User	SwpSys	per sec	Total	Direct	User	SwpSys
DLO:	19.2	74%	88\$	76%	22%	15	31.2	21%	47%	31%	22%
DL1:	.4	25	91%	75%	21%	45	.7	0%	46%	18%	37%
DL2:	4.4	17%	50%	0%	0%	100%	116.5	77%	0%	0%	100%
DL3:	1.9	7%	89%	83\$	17%	0\$	2.1	15	76%	24%	0%
Totals	25.9		825	64%	18%	18\$	150.5		11%	7%	82%

FI	GI	JR	E	1

09-Mar-	-83 11:	50:55	RSTS '	V7.0-07	Numb	erCrunci	ner	20	Errors		
29/36	Jobs	66% Us	er run	ning	19%	Fip nee	ded	915.7	Second	8	
						Fip in			Cache		
	17 Access	5.4 Ch	ar/sec	out	8.5	Char/se	e in Blocks	87	Min sm	l buf	r
Disk		Total	Reads	Direct	User	SwpSys	per sec	Total	Direct	User	SwpSy
DLO:	19.3	75%	90%	77\$	23\$	0\$	44.0	57%	74%	26%	05
DL1:	1.1	4%	71%	72%	28%	0%	5.6	7%	89%	11%	01
DL2:	.3	1%	63%	11%	18%	71%	7.3	9%	50%	1%	50%
DL3:	5.0	20%	90%	69%	31%	0%	20.7	27%	61%	39%	05
Totals	25.7		89%	74%	25%	15	77.7		69%	26%	59

	Before
1	11/34
2	11/34
3	Bootstrap
4	Front panel
5	Consol
6	
	18-bit memory (1/4 MB)
7	RL01/02 Controller
8	KMC11 Microprocessor
9	Jumper
1	Jumper
2	DUP11 Interface
3	KG11A CRC Block Check
4	DZ11
	DZ11
5	Card Reader Controller
7	Floppy Controller
8	Empty
9	Printer Controller/Jumper
1	Jumper
2	DZ11
	DZ11
3	
4	Terminator
	After
1	11/34
2	11/34
3	Bootstrap/Front panel
3 4	Empty
	Consol
5 6	Floppy Controller
7	RL01/02 Controller
8	KMC11 Microprocessor
9	Jumper
1	Empty (only usable for cache)
2	Enable/Jumper
3	22-bit memory (1 MB)
4	Jumper
1	Jumper/KG11A CRC Block Check
2	DUP11 interface
3	18-bit memory (1/4 MB)
3	Card Reader Controller
5	DZ11
5	DZ11
7	DZ11
8	DZ11
9	Printer Controller/Terminator
	FIGURE 2



The smoothest path between RSTS/E and VAX / VMS just got smoother: there's a major new release of

ROSS/V

ROSS/V has always provided:

- the fastest way to bring up RSTS/E applications on the VAX.
- the only way to do RSTS/E development on the VAX.
- an extensive subset of RSTS/E monitor calls and standard RSTS/E features, like CCLs, DOS-formatted magtape, and RSTS/E-style file update mode.

Now, in Version 3, ROSS/V supports:

- the "hidden" RSX run-time system (with 32 KW job size).
- resident libraries.
- job spawning and detached jobs.
- spooling to VMS print and batch queues.
- mailbox send/receive for communication with VAX-11 BASIC and other native mode applications.

How ROSS/V works:

ROSS/V is written in VAX-11 MACRO, and RSTS/E monitor calls are performed in VAX native mode. The rest of your PDP-11 code (in applications, run-time systems, TKB, etc.) is executed directly in the PDP-11 microcode that's present in every VAX. ROSS/V runs under VMS, not in place of it. Thus, some users may be working under the RSTS/E subsystem provided by ROSS/V while others are concurrently using any of the other VAX/VMS capabilities.

Call or write for the new ROSS/V technical summary, which describes all of ROSS/V's features.

Evans Griffiths & Hart, Inc.

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

OnLine Data Processing, Inc.

N. 637 Hamilton Spokane, WA 99202 (509) 484-3400

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

CIRCLE 176 ON READER CARD

ROSS/V

USING THE BATCH PROCESSOR TO REORDER DIRECTORIES OVERNIGHT

By Michael Mowat Marine Laboratory, Aberdeen, Scotland

The advantages of frequently reordering directories are well known, but it is not usually convenient to stop the system in order to do this.

This document gives a batch command file and program listings so that a complete reorder may be done at midnight every night or as soon thereafter as the batch processor is available. All other jobs are suspended until the reorder is complete to prevent possible damage to the directories by files being opened during the reordering.

The program TIDY.BAS is an optional part of the process. Its purpose is to delete all files of certain categories such as *.BAK,*.LST etc. These files are left one complete working day; e.g., Monday's files are deleted early on Wednesday morning. Files from Friday, Saturday and Sunday are not deleted until Tuesday morning.

The system on which this procedure is being used is a PDP11/55 running RSTS/E version 7 with two RP04 disks.

The program STOPIT.BAS prohibits logins and suspends all user programs and saves the priorities leaving batch stream BAO: and the current job running. The program RESTRT.BAS restores the priorities and re-enables logins. When completed the job requeues itself for the next day.

```
Batch Command File REORDR.CTL
     $JOB/CCL/NOLIMIT/PRI:-48/NOCPU
    PIP @TIDY[1,9]
PIP [1,9]REORDR.BAC=$REORDR.BAC
    $RUN TIDY[1,9]
$RUN STOPIT[1,9]
    $RUN REORDR[1,9]
    NO
    DB0:[*,*]
    DB1:[*.*]
    $RUN RESTRT[1.9]
    QUE BAO:/AF:23:59/PRIORITY:160=[1,9]REORDR.CTL
 TIDY. BAS
   5 REM Program to kill certain files from the last but one working day.
6 REM Saturday and Sunday are not working days.
7 REM PIP @TIDY is done after midnight every day
8 REM Then TIDY.BAS is run to generate indirect command file for next day.
    10 D$=DATE$(0)
   20 H%=INSTR(1%,D$,'-')
30 D=VAL(LEFT(D$,H%-1%))
   40 D$=RIGHT(D$,H%+1%)
50 H%=INSTR(1%,D$,'-')
   60 M$=LEFT(D$,H%-1%)
70 Y=1900+VAL(RIGHT(D$,H%+1%))
70 Y=1900+VAL(RIGHT(D$, H$+1$))
80 M$=CVT$$(M$,63$)
100 IF M$='JAN' THEN M=1\GO TO 300
120 IF M$='FEB' THEN M=2\GO TO 300
120 IF M$='MAR' THEN M=3\GO TO 300
130 IF M$='MAR' THEN M=5\GO TO 300
140 IF M$='MAY' THEN M=5\GO TO 300
150 IF M$='JUL' THEN M=6\GO TO 300
170 IF M$='JUL' THEN M=6\GO TO 300
180 IF M$='SEP' THEN M=8\GO TO 300
180 IF M$='SEP' THEN M=9\GO TO 300
200 IF M$='OT' THEN M=10\GO TO 300
210 IF M$='DT' THEN M=10\GO TO 300
220 PRINT 'ILLEGAL DATE' \GO TO 300
220 PRINT 'ILLEGAL DATE' \GO TO 32767
300 F=INT((14-M)/12)
330 F=INT((14-M)/15)

310 F=INT(30.61*(M+1+F*12))+D=INT(365.25*(Y-F))

320 J=J=INT((INT((Y-F)/100)+1)*.75)+1720997

330 W=J+1=INT((J+1)/7)*7
 1000 OPEN '[1,9]TIDY.CMD' FOR OUTPUT AS FILE #1%
```

```
1020 B$='/BE:'-DATE$(0)-+'/DE'
1030 PRINT $1% A$; 'BAK';B$
1040 PRINT $1% A$; 'LOG';B$
1050 PRINT $1% A$; 'LST';B$
1060 PRINT $1% A$; 'LST';B$
1070 PRINT $1% A$; 'DIR';B$
1080 PRINT $1% A$; 'DIR';B$
1080 PRINT $1% A$; 'DIF';B$
1090 PRINT $1% A$; 'DIF';B$
2010 A$='DB1:[*,*]*.'
2030 PRINT $1% A$; 'BAK';B$
2040 PRINT $1% A$; 'BAK';B$
    2030 FRINT #15 A$; 'BAK', B$

2040 FRINT #15 A$; 'LOG', B$

2050 FRINT #15 A$; 'LST', B$

2060 FRINT #15 A$; 'MAP', B$

2070 FRINT #15 A$; 'DIF', B$

2080 FRINT #15 A$; 'CFF'; B$
    3000 CLOSE #1%
  STOPIT.BAS
      EXTEND
    10 REM Program to suspend all jobs except the current one and the
    batch processor and save the priorities in SAVPRI.DAT.

14 REM DONT DO IT AFTER 05:00 IN CASE LONG PROGRAM IS IN BATCH

15 IF TIME(0)>18000 GO TO 800
   20 S$=SYS(CHR$(6%)+CHR$(-2%)) ISUSPEND LOGINS
   30 SLEEP 20
40 DIM #1%, PRIOR%(40%)
   50 DIM SYS.3$(30),SYS.12$(30)
  50 CHANGE SYS(CHR$(6$)+CHR$(-3$)) TO SYS.3$

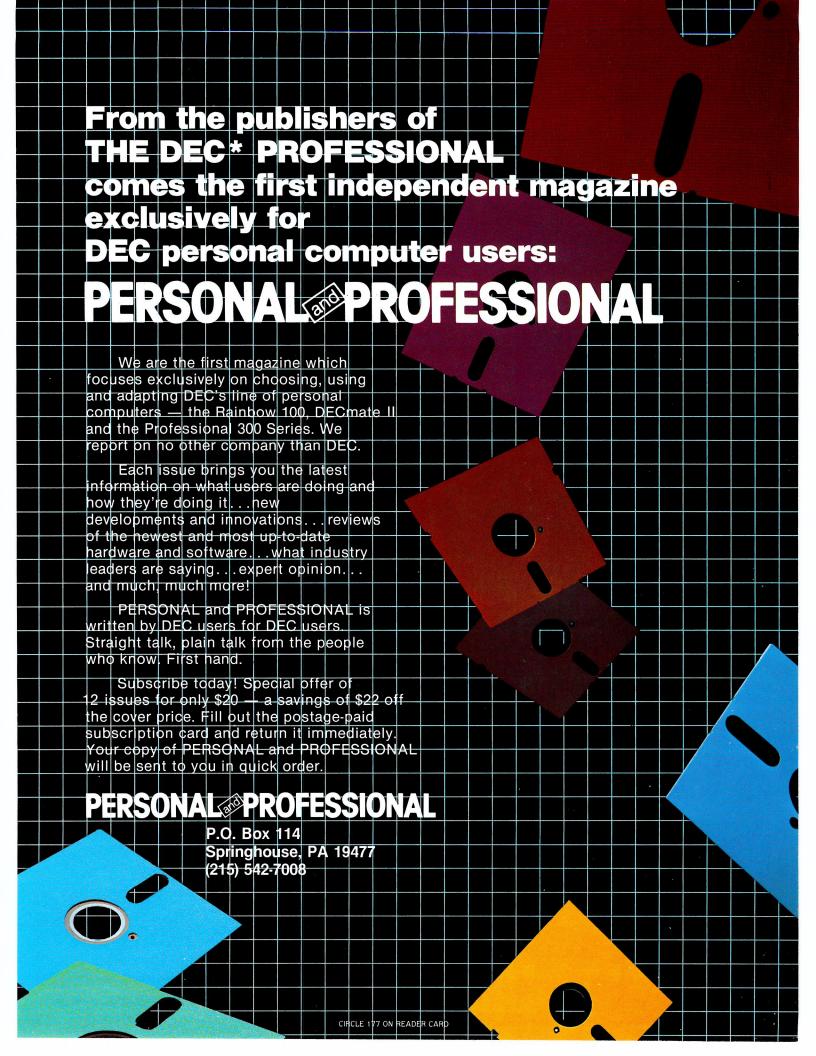
70 SYS.3$(I$)=SYS.3$(I$)+SWAP$(SYS.3$(I$+1$))FOR I$=5$ TO 29$ STEP 2$

80 CHT.KB$=SYS.3$(3$) INumber of keyboards

MAXCHT$=SYS.3$(4$) INaxinum job number

\[ \text{DEPCENT$}=SYS.3$(5$) IUnit number table
\]
                                               !Unit number table | IDDB pointer table
       \DEVPTR$ = SYS.3%(7%)
\MEMLST$ = SYS.3%(9%)
                                               !Memory list root
       \JOBTBL%=SYS.3%(11%)
\JBSTAT%=SYS.3%(13%)
                                              !Job table
!Job status table
   \JBWAIT$=SYS.3$(15$) | Job wait flags table \JSBTBL$=SYS.3$(23$) | Job status bits table |
120 CHANGE SYS(CHR$(6$)+CHR$(-12$)) TO SYS.12$
  130 SYS.12%(I%)=SYS.12%(I%)+SWAP%(SYS.12%(I%+1%)) FOR I%=5% TO 29% STEP 2%
140 DEVNAM%=SYS.12%(5%) 1Device name table
   155 GOSUB 450
  100 DEN '(1,9)SAVPRI.DAT' FOR OUTPUT AS FILE #1%
170 PRIOR%(1%)=-200 FOR 1%=0% TO 40%
200 CURJOB4=(PEEK(518%) AND 255%)/2%
210 FOR J%=1% TO 40%
   220 JDB$=PEEK(JOBTBL$+J$+J$)
230 IF JDB$=0$ GO TO 400
   240 JDB2% = PEEK(JDB%+8%)
   250 RCVR$ = PEEK(JDB2$+18$)
   260 IF RCVR$=0$ GO TO 300
  270 RCVR.ID$=CVT$$(SWAP$(PEEK(RCVR$+2$)))
+CVT$$(SWAP$(PEEK(RCVR$+4$)))
  +CVT$$(SWAP$(PEEK(RCVR$+6$)))
300 PRIORITY$=PEEK(JDB$+28$) AND 255$
  300 PRIORITY$=PROGRITY$-265 ARD 250$
310 PRIORITY$=PROGRITY$-265 | IF PRIORITY$>=128$
320 IF J$=CURJOB$ GO TO 400
330 IF RCVR.ID$='BAOSP!' GO TO 400 ILEAVE BATCH JOB RUNNING
340 PRIORS(J$)=PRIORITY$
350 S$=SYS(CHR$(6$)+CHR$(-13$)+CHR$(-1$)+CHR$(-1$)+CHR$(-128$)+STRING$(4$,0$))
   360 REM SUSPEND ALL JOBS OTHER THAN THIS ONE AND BATCH
   400 NEXT J%
  410 CLOSE #1%
420 GO TO 32767
  450 T%=12%
460 M%='System going down in'+NUM%(T%)+' minutes. Please finish up'
  465 M$=CHR$(13$)+CHR$(10$)+M$+CHR$(13$)+CHR$(10$)+STRING$(7$,7$)
470 T$=T$-2$
480 IF T$<0$ THEN RETURN
  500 FOR KB.UNIT%=9% TO CNT.KB%
510 KB.DDB%=PEEK(PEEK(KB.BASE%)+(KB.UNIT%*2%))
  520 JOB.2%=(PEEK(KB.DDB%+2%)) AND 126%
  530 IF JOB.2%=0% GO TO 600
  535 ON ERROR GO TO 700
540 S$=SYS(CHR$(6$)+CHR$(-5$)+CHR$(KB.UNIT$)+M$)
  550 ON ERROR GO TO 0 600 NEXT KB.UNIT%
  650 SLEEP 120
  700 RESUME 600
  800 KILL '[1,9]REORDR.BAC'
810 KILL '[1,9]SAVPRI.DAT'
  820 REM THIS WILL PREVENT EITHER REORDR OR RESTRT FROM RUNNING
RESTRT. BAS
  1 EXTEND
  2 REM Program to restart jobs suspended by STOPIT.BAS
  10 DIM #1%, PRIOR$(40%)
20 OPEN '[1,9]SAVPRI.DAT' FOR INPUT AS FILE #1%
30 FOR J%=1% TO 40%
  40 S$=CHR$(6%)+CHR$(-13%)+CHR$(J%)+CHR$(-1%)+CHR$(PRIOR%(J%))+STRING$(4%,0%)
  50 S$=SYS(S$) UNLESS PRIOR$(J$)=-200$
  60 NEXT J%
      CLOSE #1%
  80 S$=SYS(CHR$(6%)+CHR$(-1%)) !ENABLE LOGINS
```

1005 IF W=0 OR W>5 GO TO 3000 1010 A\$='[*,*]*.' 1020 B\$='/BE:'+DATE\$(0)+'/DE'



TU-58 TAPE DRIVE

By Dave Goodman, Medford, MA

1	EXTEND Idave goodman 122 blake st.		& & &
	lbox 31 !medford, mass 02155		& &
		/44 systems delivered with ddn't have any reliable way	& & &
	! directions for use:		& & &
	! 1) you must have sysge ! 2) you must have a bas ! 3) this can be cel dr:	sic compiler	& & &
	call format is:		& & &
	tu/a/b/d	where: a = unit number (0 or 1) b = function code (1-8) c = first parameter ie. input file d = second parameter s: /DET, /SI:16 are also valid	& & & & &
	! version: ! author: ! date:	1.2 DAVE GOODMAN 12-10-82	4
2	PRINT IF CCPOS(0\$) \PRINT 'TU58 PROCCESSER		& &
	ON ERROR GOTO 19000	!print banner header	4
100	\PRINT lunit number entry point INPUT 'Unit Number (0 or \GOTO 100 IF UNIT\$ > 1\$ \TAPE.DEVICE\$ = 'DD' + 1	1);';UNIT\$ OR UNIT\$ < 0\$	& & &
	\PRINT \GOTO 250	!build device name	4 4
200	! !selection processer		& &
	PRINT 'FUNCTIONS ARE;' \PRINT ' \PRINT '	(1 = KB: TO TAPE)' (2 = DISK FILE TO TAPE TRANSFER)'	4
	\PRINT ' \PRINT '	(3 = TAPE TO TAPE TRANSFER)' (4 = SELECTIVE READ FROM TAPE TO KB:)' (5 = SEQUENTIAL READ FROM TAPE TO KB:)'	& & &
	\PRINT ' \PRINT ' \PRINT ' \PRINT ' \PRINT '	(6 = TAPE TO DISK FILE TRANSFER)' (7 = REMIND TAPE TO BOT)' (8 = ZERO TU58 TAPE UNIT)' (9 = EXIT PROGRAM)'	4
250	INPUT ' SELECTION = ';S	!get option	& &
300	ON SEL\$ GOSUB 1000,2000	VM Salp > 799 Ivalidate option invalid option ? display options 3000,4000,5000,6000,7000,8000,32700 jump to required function	
	\CLOSE #1\$	When done return to here	ę.
	\PARAM.1\$ = ""	Iclose file to reset it	4
	\PARAM.2\$ = ** \GOTO 32700 IF REJECT\$	linitilize parameters	& & &
	\G0T0 32700 IF RESECT;	lexit if ccl entry and done	& &
		lget option again	& & &
1000	PEN TAPE. DEVICES FOR I	E TRANSFER ROUTINE ************************************	& & &
	GOTO 1010 IF LEN (PARAM		ŧ.
1005	INPUT 'RECORD WHAT >';P \RETURN IF CVT\$\$(PARAM.	ARAM.1\$ 1\$,4\$) = "" lget data to record	& & &
1010	GOTO 1020 IF LEN(PARAM. \INPUT 'RECORD IT WHERE	<pre>iif no data then exit 2>) > 05 > i;PARAH.2\$ lcol parameter location on tape then jmp 1020 lelse get location for data</pre>	& & &
1020	PLACES = VAL(PARAM.2\$)	linterger for lacation	å å
	\REC.LENS = LEN(PARAM.1	t) Iset lenght of data	& &
	\FIELD #1\$, REC.LEN\$ AS		& &
	\LSET REC. IN\$ = PARAM. 1		å
1030	PUT #1%, RECORD PLACES,		å &
	\PARAM.1\$ = "" \PARAM.2\$ = ""	!reinitilize parameters	& &
	\GOTO 1005		Ł

```
DISK TO TAPE TRANSFER
         OPEN TAPE.DEVICE$ FOR INPUT AS FILE #1$
\X$ = 512$
\X$ = 0$
\X1$ = 0$
         OPEN PARAM.1$ FOR INPUT AS FILE #2$, RECORDSIZE 512$

\[ \text{FIELD #15, 512$ AS REC.UT$} \]

\[ \text{FIELD #25, 512$ AS REC.IN$} \]
          Idoes file exist, yes field records
GET #25, RECORD X15, COUNT 5125
LISET REC.OUT$ = REC.IN$
         iget record, push to output buffer
iff full tape then gosub to routine at line
111000 which request new tape unit number
land resets specified variables and returns

\X5 = X5 + 15
\X15 = X15 + 15
\G070 2030
2030
                                         !put record, increment counter, goto 2030 !for next record
          YS = YS - (XS-15)

\PRINT XS-15; BLOCKS TRANSFERED FROM '; PARAM.1$
\PRINT YS; REMAINING ON UNIT; UNITS
\(\)GOTO 2015
          2060
           OPEN TAPE. DEVICES FOR INFUT AS FILE #15
NEW. UNITS = 15 IF UNITS = 05
NEW. UNITS = 15 IF UNITS = 15
NEW. TAPE. DEVICES = 10D ' + NUMI$(NEW. UNIT$) + ':'
OPEN NEW. TAPE. DEVICES FOR INPUT AS FILE #25
FIELD #25, 5125 AS REC. UNI$
FIELD #25, 5125 AS REC. UNI$
FOR X5 = 05 TO 515
GET #15, RECORD X5, COUNT 5125
LEST REC. OUT$ = REC. IN$
PUT #25, RECORD X5, COUNT 5125
NEXT X5
NEXT X5
FRINT '512 BLOCKS TRANSFERED FROM TU58 UNIT #';
NETURN
3020
           | ****** end of tape to tape transfer routine
          4000
          \NEXT X$\
\PRINT 'TRANSFER OF TU58 UNIT';UNIT$;' TO KB: IS COMPLETE'
          1****** tape to disk transfer module ********
           OPEN TAPE. DEVICE$ FOR INPUT AS FILE #15
           \X$ = 0$
\X1$ = 0$
   \GOTO 6005 IF LEFT(YN$, 15) <> 'Y'
```

6020	OPEN PARAM.1\$ FOR OUTPUT AS FILE #2\$, RECORDSIZE 512\$
	\FIELD #1\$, 512\$ AS REC. IN\$
	\FIELD #2\$, 512\$ AS REC.OUT\$
6030	GET #15, RECORD X5, COUNT 5125
	\RETURN IF CVT\$\$(REC. IN\$,4\$) = ""
6040	LSET REC.OUT\$ = REC.IN\$ PUT #2\$, RECORD X1\$, COUNT 512\$
0040	\X\$ = X\$ + 1\$
	\X1\$ = X1\$ + 1\$
	\IF X\$ <= 511\$ THEN
	GOTO 6030
6050	INPUT 'IS THERE ANOTHER VOLUME ';YN\$
	\RETURN IF LEFT(YN\$,1\$) = 'N' \INPUT 'ENTER UNIT NUMBER OF NEXT VOLUME ';UNIT\$
	\GOTO 6050 IF UNIT\$ <> 0\$ AND UNIT\$ <> 1\$
	\X\$ = 0\$
	\TAPE.DEVICE\$ = 'DD' + NUM1\$(UNIT\$) + ':'
6060	OPEN TAPE. DEVICES FOR INPUT AS FILE #15
	\FIELD #1\$, 512\$ AS REC.IN\$ \GOTO 6030
	1 esses end of tape to disk transfer routine

7000	1 eeee rewind tape routine
	OPEN TAPE. DEVICES FOR INPUT AS FILE #15
	\GET #1\$, RECORD 0\$, COUNT 512\$ \PRINT 'TU58 UNIT #';UNIT\$;' HAS BEEN REWOUND'
	\RETURN
	;***** end of rewind module
8000	
0000	OPEN TAPE. DEVICES FOR INPUT AS FILE #15
	\GOTO 8010 IF LEN(PARAM. 1\$) > 0\$
	\INPUT 'Really ZERO tu58 tape unit #1';YN\$ IF UNIT\$ = 15
	\INPUT 'Really ZERO tu58 tape unit #0';YN\$ IF UNIT\$ = 0\$
0040	\PARAM.1\$ = YN\$
8010	RETURN IF LEFT(PARAM.1\$,1\$) <> 'Y' \FIELD #1\$, 512\$ AS REC.IN\$
	\FOR X\$ = 0\$ TO 510\$
	\LSET REC. IN\$ = STRING\$(512\$,0\$)
	\PUT #1\$, RECORD X\$, COUNT 512\$
	NEXT X\$
	\PRINT 'TU58 UNIT';UNITS;' HAS BEEN ZEROED'
11000	**** MULTI TAPE TRANSFERS HADLEING *****
	CLOSE #15
	\X\$ = 0\$
	\PRINT 'TAPE UNIT #';UNITS;'IS FULL ' \PRINT 'PLEASE MOUNT ADDITIONAL TAPE TO FINISH PROCCESSING'
11100	INPUT 'ENTER UNIT NUMBER OF NEW TAPE '; UNIT
	\GOTO 11100 IF UNIT\$ <> 0\$ AND UNIT\$ <> 1\$
	\TAPE.DEVICE\$ = 'DD' + NUM1\$(UNIT\$) + ':'
	\OPEN TAPE.DEVICE\$ FOR INPUT AS FILE #15
	\FIELD #1%, 512% AS REC.OUT\$ \LSET REC.OUT\$ = REC.IN\$
	CLOBI MEC. OUI - MEC. INS

9000	RETURN ***** standard error trapping *****
9010	! IF (ERR = 11\$) AND (ERL = 2030) THEN
3010	RESUME 2050
9020	IF (ERR = 5%) AND (ERL = 6010) THEN RESUME 6020
9030	IF (ERR = 11%) AND (ERL = 2040) THEN
	GOSUB 11000 \RESUME 2040
9040	IF (ERR = 11%) AND (ERL = 100) THEN RESUME 32700
9050	IF (ERR = 11\$) AND (ERL = 250) THEN RESUME 100
9999	PRINT 'FATAL ERROR AT LINE '; ERL;' IN TU58'
	\PRINT 'ERROR NUMBER = ';ERR
	\PRINT 'IN UNIT #';UNIT\$;', FUNCTION WAS ';SEL\$
	\PRINT 'PARAMETERS: 1= ';PARAM.1\$;', 2= ';PARAM.2\$
	\PRINT 'VARIABLES X=';X\$;' ,X1\$ =';X1\$ \RESUME 32700
	funknown error ?
	Iprint pertinate infomation
	!resume through bottom for clean exit
30000	:****** CCL entry point *********
	lentered via ccl command TU58/UNIT/FUNCTION/PARAMETER/PARAMATER
	ON ERROR GOTO 19000
	\CORE.COMMON\$ = SYS(CHR\$(7\$))
	\CUT.1\$ = INSTR(1\$,CORE.COMMON\$,'/') \GOTO 2 IF CUT.1\$ = 0\$
	!entered via simple TU58 call
	Igoto top of program
10010	UNITS = VAL(MID(CORE.COMMON\$, CUT.1\$+1\$,1\$))
	\GOTO 100 IF UNITS > 15 OR UNITS < 05
	!valid unit number ?
	!no? prompt for it via main line code
10050	TAPE.DEVICE\$ = 'DD' + NUM1\$(UNIT\$) + ':'
	\CUT.2\$ = INSTR(CUT.1\$+1\$, CORE. COMMON\$,'/')
	\GOTO 200 IF CUT.2\$ = 0\$
	!no function code entered so must prompt !for it via main line code
0030	SELS = VAL(MID(CORE.COMMON\$,CUT.25+15,15))
10030	\GOTO 200 IF SEL\$ < 1\$ OR SEL\$ > 9\$
	!valid func code ?
	Ino ? prompt for it via main line code
0040	REJECTS = -15
	\CUT.3\$ = INSTR(CUT.2\$+1\$, CORE.COMMON\$,'/')
	\GOTO 300 IF CUT.3\$ = 0\$
	\CUT.4\$ = INSTR(1\$,RIGHT(CORE.COMMON\$,CUT.3\$+1\$),'/')
	\PARAM.1\$ = RIGHT(CORE.COMMON\$,CUT.3\$+1\$) IF CUT.4\$ = 0\$ \PARAM.1\$ = MID(CORE.COMMON\$,CUT.3\$+1\$,CUT.4\$-1\$) IF
	CUT.4\$ <> 0\$
0050	GOTO 300 IF CUT.4\$ = 0
	\PARAM.2\$ = RIGHT(CORE.COMMON\$,CUT.4\$+1\$)
2700	CLOSE #X\$ FOR X\$ = 1\$ TO 12\$
-100	ND

BACmac cando 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

DCL SUBROUTINES FOR VAX/VMS

By Philip G. Anthony, Technical Systems, Fidelity Bank, Philadelphia, Pa.

Editor's Note: DCL command procedures can "call" each other, passing/returning arguments and results and maintaining "local" symbols, as documented in the DCL manual. This article addresses a technique for "calling" DCL "subroutines" defined within the "main" command procedure, as found in the syntax of many high-level programming languages.

The Digital Command Language (DCL) on VAX/VMS, as opposed to the imposter of the same name on RSTS, stands out in my mind as the most important software advance that DEC has supplied us in recent years. My IBM-ish friends stare in amazement as I create a command file that uses exactly the same syntax I would type in at my terminal, and "run" it from my terminal — and then turn around and submit it as a batch job. When they learn that I can pass arguments to a command procedure, that I have unlimited variables to play with, and that I can set up almost any kind of control loop I want, they start drooling hungrily. IBM, take note.

Personally, I fell in love with DCL as soon as I saw it, despite certain limitations in its syntax and the oddity of its imbalanced single quotes. In a way, it was like meeting an old friend: There's no predefinition of variables, but some of the code syntax in DCL is a lot like that in DIBOL, the first programming language I ever learned.

I soon began drawing up a wish list of features that DEC really ought to implement to make it a full-fledged command language. High on that list was the ability to call subroutines. DEC hasn't answered my wishes yet, and to the best of my knowledge, they have no plans to do so. Finding workarounds is more fun than kvetching, though, and also more productive; and by now I've found a way to create pseudo subroutines within a command file using DCL's symbol substitution capability for passing variables, including a return address for the "subroutine."

TEST01.COM demonstrates the technique of passing variables back and forth between the main "program" and its called "subroutine." P1 and P2, of course, are arguments passed from the terminal to the command procedure; setting the command file up this way permitted testing with several strings of various lengths. L1 and L2 are set up to be decimal arguments to be returned from the subroutine containing the lengths, respectively, of P1 and P2. And RTN is an alpha variable to hold the label name to which control will be transferred at the end of the "subroutine."

The key to its working is the fact that DCL variables are symbols in the fullest sense of the word. The name of one symbol can be assigned to another symbol, and the value of a symbol can be substituted for the symbol, freely. In fact the nesting can be indefinitely deep (though the syntax gets somewhat more complicated). The line

\$ 'LEN = F\$LENGTH('STR)

says, in effect, "Substitute the symbol name pointed to by STR for the symbol STR; find the length of the value of that symbol and point the symbol whose name is pointed to by the symbol LEN to the returned length." Similarly, the line

\$ GOTO 'RTN

is translated "Substitute the value of the symbol RTN for the symbol RTN and then go to the address referenced by that value." Complicated? No more complicated than "MOV @(SP),RO". These internal workings can be examined by typing

\$ SET VERIFY

before invoking TEST01.

A minor variation on this is TESTO2.COM, which shows the possibility of two-level nesting. This version is the same as TESTO1 (minus most of the comments) except that the strings "P1" and "P2" are assigned not to the symbol STR but to an intermediate symbol, DUMMY. Then STR is loaded with the symbol name "DUMMY".

There are two ways of handling the double indirection. The easy one is to use the indirection operator "&", as TESTO2 does the line

\$'LEN = F\$LENGTH(&'STR)

handles the extra level of indirection by making the argument to F\$LENGTH "the value of the symbol pointed to by the symbol STR".

More generally, one can work back any number of times by assigning local dummy variables. To do this, we'd replace the single line with the "&" construct on it by multiple lines,

\$ VARIABLE = 'STR

\$ 'LEN = F\$LENGTH('VARIABLE)

which load the symbol VARIABLE with the value pointed to by STR and then get the length of the string pointed to by VARIABLE's contents. For five levels of indirection, we could use VARIABLE1 through VARIABLE4 this way.

I haven't found any use for the multiple-level indirection in subroutine calling yet, though it is useful elsewhere. A subroutine that calls another subroutine would set up the variables for the second-level subroutine by a line similar to \$ VARIABLE = 'STR, which would assign the symbol pointed to by STR to VARIABLE, maintaining a single level of indirection.

The only problem in handling a second-level subroutine

ACCORD with DEC systems . . . the new price/performance standard for DMF- or DH-class controllers . . . and beyond!

SALES BULLETIN

 $OPTIMUX^{\text{\tiny{IM}}}$ is ACCORD's family of optimized comm controllers for VAX-11, PDP-11 and DECSYSTEM 10/20.

- In VAX-11 systems, the OPTIMUX/STAR DMF and the OPTIMUX/16 DMF work with software supporting Digital's DMF32.
- In PDP-11 systems, the OPTIMUX/STAR DH and the OPTIMUX/16 DH work with software supporting Digital's DH 11 and DM11-BB.

Matalatalatatata International and

PLAYING VEGAS IN OCT

THAT'S NEAT . . . THEY'RE ALREADY PLAYING IN SYSTEMS COAST TO COAST.

ACCORD COMPUTER

- In any Unibus compatible system, the OPTIMUX/KCP provides four programmable communication ports. Any bit or byte-oriented protocol may be implemented. DMA and onboard memory assure optimal line utilization.
 - Reliability and product support are in ACCORD.

There is nothing like OPTIMUX anywhere else in Digital's world...

... just read on.

OPTIMUX/STAR ... can support any *practical number of lines* with a *single-board*, host-resident controller. Its intelligent distribution consoles may be located *up to 4,000 feet from the host computer*.

CONTROL YOUR FUTURE!

 $OPTIMUX/16^{\tiny \textcircled{13}}\ldots$ is the most *cost-effective* means to implement 16 lines with a single-board host-resident controller linked to a rack-mountable distribution panel.

CONTROL YOUR COSTS!

OPTIMUX/KCP . . . is a *unique* programmable, four-line communications controller. On-board processor handles your application software to offload the host CPU.

CONTROL YOUR NETWORK!



THE NETWORK ARCHITECTS

714/851-3932

*BOOTH 105 at DEXPO Oct. 23-25.

© 1983, ACCORD Computer Corporation.

CIRCLE 206 ON READER CARD

DEC, VAX, PDP and DECSYSTEM 10 or 20 are trademarks of Digital Equipment Corporation.

ACCORD and OPTIMUX are trademarks of ACCORD Computer Corporation.

You can do more when you do it simply..

You can do more with Minitab than analyze the results of your experiment. Straightforward commands let you PLOT your sales activity, FORECAST market trends, DESCRIBE your stock portfolio, TABLE your survey responses and more.

An hour's training and the use of the available on-line HELP facility can put you in command of Minitab's language, and once you've learned it, you can operate it interactively or in batch mode in a wide variety of environments:

VMS, UNIX, RSTS/E, RSX-11M/M+, RT-11 TSX-PLUS, IAS, TOPS-10, TOPS-20



For more information about Minitab, call or write: Minitab, Inc

Box RP1083 • 215 Pond Laboratory • University Park, PA • 16802 814/865-1595 Telex 84 2510

UNIX is a trademark of Bell Laboratories. Minitab is a trademark of Minitab, Inc.

CIRCLE 168 ON READER CARD

-comprehensive guide to S software.

SIERRA DIRECTORY:

The latest, most complete list of RSTS software. Hundreds of software systems and software vendors exclusively for the RSTS Operating System. Product descriptions, prices, technical specifics, vendor support provisions, vendor addresses and phone numbers. Cross-indexed by software function and vendor.

- Financial modeling
- Accounting
- Statistics
- Electronic mail
- Word processing
- Languages and compilers
- Communications
- System utilities
- Programmers tools
- Database systems
- Report writers
- Graphics
- Many industry-specific systems-and much more

ORDER YOUR COPY TODAY FOR PROMPT DELIVERY Sierra Directory, Dept. 11, 3304 Springhill Rd., Lafayette, CA 94549 (415) 284-1610

copies of the Sierra Directory of RSTS Payment is enclosed for Software at US\$29.00 each (US\$31.00 outside North America), including shipping and handling.

NAME

ADDRESS

CITY/STATE/ZIP

COUNTRY

RSTS is a trademark of Digital Equipment Corp. CIRCLE 192 ON READER CARD

is that the current return address has to be saved in a very local variable before the return address for the second-level routine is stuffed. Alternatively, return addresses could be named RETURN_FROM_JOE, RETURN_FROM_HARRY. and so on, rather than using the symbol RTN as a standard return-address variable, as I do it for simplicity's sake.

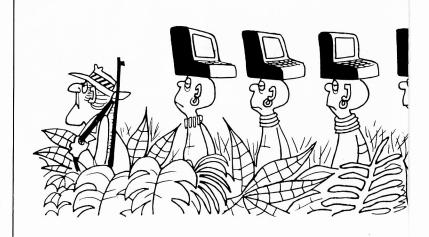
A third solution would be to use a single symbol such as RTN, to prefix the return-name variable, then add an integer translated to string variable, to index just which RTN we're referring to — RTN1, RTN2, etc. By examining a global index (RTN_INDEX, let's say), the subroutine would know which symbol its return address was referenced by. This one definitely is complicated.

DCL's symbol substitution can be an extremely powerful tool in command procedure creation — so much so that DEC's books on using DCL only scratch the surface of what can be done. For example, DCL doesn't provide arrays (another item on my wish list), but by means of symbol substitution it isn't hard to create inefficient but useable pseudo arrays that can be used for everything from breaking out parameter lists to handling data translations. In fact, that's what the third method of dealing with nested subroutine return addresses is all about.

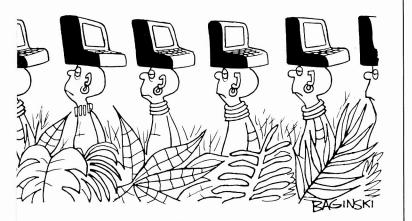
[Editor's Note: The "PUSH" and "POP" command procedures in "VMS Directories: Some Navigational Aids" (page 76) use such a DCL symbol "pseudo-array" (PSH\$DEFAULT).]

With a little bit more work and trickier syntax, it can be used to emulate (wish list again) ON X GOTO A,B,C — or, for you DIBOL fanatics, GOTO (A,B,C),X For the present, in the words of some of the nastier mathematicians I know, these will be left as an exercise for the reader. Maybe I'll look at them in a future article.

For now, this subroutine method is clumsier than a simple GOSUB or CALL statement. But until DEC gets around to embedding one of these statements in DCL, the procedure provides me with cleaner command flow and — yes — far less typing in complicated command procedures. Now if somebody would be so kind as to explain to me the how and why of all those complicated, restrictive RMS attributes, I might feel right at home on VAX.



```
$ ! T E S T O 1 . C O M
  ! Test of DCL pseudo subroutine handling
  ! First assign the names of two symbols to two other
$ ! symbols and assign the name of the return address
$ ! to a third symbol
  STR = "P1"
$ LEN = "L1"
$ RTN = "A"
  ! Go to the subroutine
$ GOTO SUBR
$ ! Assign two different symbol names to STR and LEN
$ ! and a new return address to RTN
$ LEN = "L2"
$ RTN = "B"
! Go to the subroutine again; on return, print out the ! results of the two subroutine calls
$ GOTO SUBR
$B:
  WRITE SYS$OUTPUT "Length of P1 was ''F$STRING(L1)'"
WRITE SYS$OUTPUT "Length of P2 was ''F$STRING(L2)'"
$ EXIT
$ ! Here's the subroutine, which returns the length of a
$ ! passed string argument
$SUBR:
  'LEN = F$LENGTH('STR)
$ GOTO 'RTN
$ ! T E S T O 2 . C O M
  ! Test of DCL pseudo subroutine handling with two levels
  ! of symbol indirection
  ! This time "DUMMY" will be used as an intermediate
$ ! symbol
$ STR = "DUMMY"
$ DUMMY = "P1"
$ LEN = "L1"
$ RTN = "A"
$ GOTO SUBR
$ DUMMY = "P2"
$ LEN = "L2"
$ RTN = "B"
$ GOTO SUBR
$B:
* WRITE SYS$OUTPUT "Length of P1 was ''F$STRING(L1)'"
* WRITE SYS$OUTPUT "Length of P2 was ''F$STRING(L2)'"
$ EXIT
$ ! With two levels of indirection, the symbol STR must
$ ! first be translated (to "DUMMY") before the length of
$ ! P1 or P2 is found
$SUBR:
$ 'LEN = F$LENGTH(&'STR)
$ GOTO 'RTN
```



ATTENTION VAX

SCIENTIFIC/ENGINEERING PROBLEM SOLVERS

OL/ASAP, the premier nonlinear equation solver is now available for VAX and is priced at a very affordable \$5,000.

OL/ASAP allows math models to be depicted in a convenient textbook format avoiding <u>all</u> need for writing a simulation program.

OL/ASAP SUPPORTS

- Simultaneous algebraic and/or ordinary differential equations
- Implicit form ODEs
- Continuous algebraic equations
- Subscripted equation forms
- · Selection of unknowns at execution time
- Boundary value problems
- Much more

Call or Write:

OLI Systems, Inc. 15 James Street Morristown, NJ 07960 (201) 540-0291

TRIAL LICENSE: \$500 FULL LICENSE: \$5,000

CIRCLE 202 ON READER CARD

Announcing MPR

A VERSATILE MACRO-LANGUAGE UTILITY FOR RSTS/E

Flexible enough for OEM's and End-Users

Functional highlights include:

- Customized code generation.
- Nestable macros and "INCLUDE" statements with arguments.
- · Algebraic expression evaluation.
- Control file generation.
- Extensive debugging mode.
- Easy to use syntax.
- Complete documentation with sample macros.

Benefits:

- Reduce software maintenance costs.
- · Increase software portability.
- · Simplify operations and training.
- Compatible with your existing programming standards.

Interested in significantly improving programming productivity?

CALL OR WRITE FOR DETAILED INFORMATION

Noah Dixon

Star Plan Data Processing, Inc.

2040 W. Wisconsin Avenue - Suite 354 Milwaukee, Wisconsin 53233 - (414) 933-0800

· Soon to be released for RSX and VMS

RPM CAN DRAMATICALLY INCREASE THE PERFORMANCE OF YOUR SYSTEM

You can double or triple your system's performance without adding costly hardware. RPM provides everything you need to optimize your system and keep it running at peak performance.

RPM analyzes your system performance automatically, identifying problem areas. But, RPM doesn't stop there. It identifies the cause of the problems and makes suggestions for correction in plain English.

AUTOMATIC PERFORMANCE ANALYSIS

Instead of dumping columns of cryptic numbers, RPM gives you a plain English report that describes how your system is performing. It tells you where you have problems, what caused the problems, and how to fix them.

This report analyzes each of the resources that are common problem areas. Any resources that are not being used optimally are identified. The program or files that caused the problem are then identified and suggestions are made for correcting the problem.

DETAILED PROGRAM ANALYSIS

In addition to identifying problem programs, RPM can analyze the operation of individual programs, identifying problem areas.

RPM's detailed program analysis breaks the operation of the program down into CPU usage, input/output and system calls. Usage and directory overhead counts are displayed by channel and by system call.

---- File Processor (FIP) Usage ----

- The file processor (FIP) is being excessively used. It is in use by at least one job 67% of the time. In addition, an average of 2.7 other user are waiting to use FIP. Although the file processor is in use 67% of the time, it is waiting for information from the disk 72% of the time it is waiting for information from the disk 72% of the time it is in use. The disk information that FIP is waiting for breaks down as follows:

 91% Directory.

 Reading or writing to the directory structure.

 7% Disk Allocation Table (SAT)

 Reading or writing information about free blocks on the disk.

 2% Monitor Overlays.

 Leading monitor overlays into memory.

 - Loading monitor overlays into memory 0% Miscellaneous.

 - UP MISCELLAREOUS.
 Loading disk cache information and other miscellaneous data.
 The five programs that use the most FIP resources are:
 OEO47A 275.2
 PAYREC 213.1
 - ...TKB BP2CCM 158.0

DOGIN 78.7 FIP usage can be decreased by optimizing the clustersize of frequently used files (see section 4.4.1 of the RPM User's Guide), using contiguous files where possible, reordering the directories often and minimizing the opening

RPM> EXAMINE JOB 5 EVERY 5 MINUTES

Job:	5	P	rogram	*ALL	· C	PU: 54	% Sar	mple	Time:	297 Sec	conds		
Chan 0 K	B2:		File	е	C	ount 0	vrhd Cl		Closed	File		Count	Ovrhd
3 D	M1:		3]CSPC			835	2	4 DM	1: [5, 1]	TEMP05		4 39	42
			3]TKB	. TSK		9 278	91	7 DM	1: [5,1]	EXAMPL.	TSK	733	
EMT			Ovrhd			Ovrhd	EMT	Coun	t Ovrho	EMT	Count	Ovrhd	
CALFI:		149		. READ	1441	2	. WRITE	E 46	0 133		7		
. RTS	•••	1		. LOGS	12		. CLEAF	1	2	. NAME	2	5	
. FSS OPNFQ		42 58	595	. UUO CREFQ	71 2	226	. RSX DLNFQ		1	CLSFQ	62	28	
LOKFQ		22	259	CRTFQ	1		CRBFQ			RSTFQ UU. ATF	70	250	
UU. NAI	M	1	3										

EXTENDED PERFORMANCE STATISTICS

RPM adds extended performance statistics to your RSTS/E monitor. This monitor extension captures information about overall system performance plus information on individual programs and files. Using this information, you can improve system performance by minimizing disk head movement, reducing file processor (FIP) waits, reducing swapping, and optimizing disk cache operation.

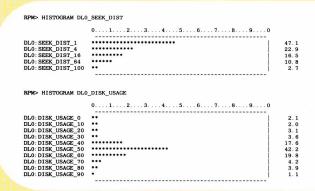
DYNAMIC PLOTTING

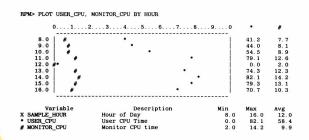
In addition to comprehensive reports, RPM can generate a wide variety of graphs. It can plot curves, draw bar charts, and plot histograms using any combination of system information. By plotting one variable against another, you can immediately see correlations between variables. This is espescially useful for determining critical resource usage points.

IDENTIFY PROGRAMS BY RESOURCE USAGE

With RPM, you can identify the programs that will provide the most benefit from optimization. You can determine which programs are used most often and which programs use the most of critical resources.

Once identified, these programs can be optimized. Overall system performance can be increased by making changes only where they will do the most good.





```
| RPM> LIST TOP 5 PROGRAMS BY PRG_CPU_TICKS PARREC 542 | PAYEND 168 | PAYROL 142 | PAYEND 168 | PAYROL 142 | PAYEND 173 | PAYEND 177 TOP 5 FILES BY FILE_DIR_OVRHD [3, 10] PROFIL. JOU 33. 1 [3, 10] PROFIL. B2S 27. 6 [2, 0] MTHEND TSK 19. 6 [10, 1] EOMREC. JOU 11. 3 | [2, 0] PAYAUD. DAT 7. 8 | RPM>
```

NO-RISK GUARANTEE

If all this sounds too good to be true, here's your chance to find out for yourself. Try RPM for 30 days. Use it to tune your system. If RPM isn't everything we say it is and more, return it for a full refund. We're not worried. We know that once you see the difference RPM can make, you won't want to be without it.



RPM -- The Performance Revolution

NORTHWEST DIGITAL SOFTWARE



Talking with OPSER

Steven L. Edwards. Software Techniques, Inc.

Included in the RSTS/E large spooling package is a program known as the OPerator SERvices program or OPSER. Very few users understand the functions of OPSER, fewer still understand how they can use OPSER in their applications.

In the context of the large spooling package, OPSER serves as the interface between the operator (as defined by OPSER) and the detached programs that make up the large spooling package. Beyond the context of the large spooling package, OPSER serves as a general purpose interface between an operator and a detached application program.

Once you understand how to talk to OPSER, and how OPSER talks to your application, you can utilize OPSER to provide a consistent and standardized interface to your applications.

OPSER can:

- 1. Log informational messages to the operator services console (OSC) and to its log file on disk.
- 2. Log fatal messages to the (OSC) and to its log file on disk.
- 3. Track requests for operator intervention like mounting tapes or changing paper forms.
- 4. Shut down an application in an orderly fashion, i.e. task C must be shut down before tasks A and B.
- 5. Notify the OSC when a program that is online with OPSER either hibernates or disappears. (OPSER doesn't appear to check for this condition with any great frequency.)

Your application communicates with OPSER using the "System Calls for Local Interjob Communication" as described in chapter 8 of the RSTS/E Programming Manual.

Messages sent to OPSER

Communication with OPSER is done in 19 byte "chunks" prefaced with a length byte. If the message is longer than 19 bytes, all but the last "chunk" will be sent with a length code of 255. These "chunks" are passed as the

"User Parameter String," bytes 21-40 of the local data messages. The first byte of the "chunk" is usually interpreted by OPSER to be a command flag that tells OPSER what to do with the message.

Command flag byte values

ASCII Value Meaning

7 This message is a REQUEST 63 This message is a fatal message

< 128 This message is an informational message 192 This message is a command to OPSER

Note that all other values, i.e., values greater than 127 excepting 192 have a tendency to crash OPSER. Since OPSER (as distributed) declares itself as a local receiver, sites with malicious users beware!

Format of requests

Request messages are logged on the OSC and in OPSER's disk log file.

REQUEST message number, date, time, . . .

TEXT STRING

In addition, requests are held in OPSER's internal tables so they can be answered by specifying the message number.

Format of fatal messages

Fatal messages are logged on the OSC and in OPSER's disk log file.

FATAL MESSAGE message number, date, time, . . . ?TEXT STRING

Format of informational messages

Informational messages are logged on the OSC and in OPSER's disk log file.

 $\begin{array}{ll} \text{MESSAGE} & \text{message number, date, time, } \ldots \\ & \text{TEXT STRING} \end{array}$

NOMANAILABLE ON RSTS

A totally new approach to application development.

Now you can create fully operational applications in days instead of weeks... and without any programming at all!

No code generated

Creates applications that are fully transportable between RSTS, RSX-11M, and RSX-11M+*

Generates documentation automatically

Take the Pro-IV Challenge!
Try Pro-IV for 30 days and prove to yourself that it's the best product on the market today.

Call us to arrange for an evaluation of Pro-IV: (617) 486-9621.



119 Russell Street P.O. Box 595 Littleton, MA 01460

"THE FOURTH GENERATION APPLICATIONS PROCESSOR"

*VMS, POS and UNIX available this fall from Pro-IV.
RSTS, RSX, VMS, and POS are trademarks of Digital Equipment Corporation.
UNIX is a trademark of Bell Laboratories.

Format of commands

There are four commands of interest that can be sent to OPSER. They are:

- 1. ONLINE n Enter sender's receiver identification in OPSER's table of online jobs. n represents the senders shutdown level. Values for n range from 0 to 9. Sender must be declared as a message receiver.
- 2. OFFLINE Remove sender from OPSER's table of online jobs.
- 3. SLE Shutup at Logical End point If the senders installed program name is "SHUTUP," this command tells OPSER to send "END" to all of OPSER's online jobs.
- 4. SIM Shutup Immediately If the senders installed program name is "SHUTUP," this command tells OPSER to send "OFFLINE" to all of OPSER's online jobs.

When OPSER is instructed to shutdown the system (normally by \$SHUTUP), OPSER will shutdown each level in ascending order. OPSER will not attempt to shutdown level n+1 until it has received OFFLINE from all programs at level n. Note that OPSER does not send shutdown messages to programs with shutdown level "9."

Messages sent by OPSER

Again, communication with OPSER is done in 19 byte "chunks" prefaced with a length byte.

There are three messages that your application can expect to receive from OPSER. They are:

- 1. LAST When OPSER is started after terminating unexpectedly, it sends this message to all of the programs that were online at the time of termination. The application is expected to reply with something that will be meaningful to the operator.
- 2. $\rm END$ This message tells the application to end at the next logical end point. The application should send OFFLINE to OPSER and exit.
- 3. OFFLINE This message tells the application to end as soon as possible. In the context of the large spooling package, this causes the spoolers to abort the current job, send OFFLINE to OPSER, and exit.

Your application is not required to respond to a LAST message. Failure to respond to END or OFFLINE will cause OPSER to interfere with system shutdown.

Using PLEASE

The PLEASE program is the user interface between a terminal and OPSER. In the context of this discussion, the following commands are of interest:

- 1. /ANSWER message number Respond to an action request.
- 2. /INTERRUPT receiver id:text Send text to receiver id.

A curious note: if you patch the PLEASE task image so that the first two words are "SHUTUP" in Radix-50, the RSX keyboard monitor will install the program name as "SHUTUP" and you can use PLEASE to shutdown the entire large spooling package without running \$SHUTUP.

Sample Code

Here is a 'quick & dirty' program whose only function is to illustrate the concepts discussed above. The program puts itself online with OPSER, then waits to receive a message. When the message is received, the user parameter string is extracted and echoed back via OPSER.

```
OPSTST
1!
1000
             ON ERROR GOTO 19000
             1030
             SETUP OUR JOB NAME.
DECLARE STRING.
SEND STRING TO TALK TO OPSER.
             TEMP.0$ = SYS(CHR$(6$) + CHR$(22$)) + SYS(CHR$(6$) + CHR$(22$)
+ CHR$(1$) + CHR$(0$) + JOB$ + STRING$(11$, 0$) + CHR$(1$)
+ CHR$(0$) + CHR$(1$) + CHR$(30$))
             REMOVE AND THEN DECLARE US TO BE A RECEIVER.
             TEMP.0$ = CHR$(192$) + "ONLINE 0"
GOSUB 3000
             PUT US ONLINE WITH OPSER.
             TEMP.0$ = SYS(CHR$(6$) + CHR$(22$) + CHR$(2$) + CHR$(1$)
+ STRING$(8$,0$) + CVT$$(SWAP$(512$)))
TEMP.0$ = CVT$$(MID(TEMP.0$, 22$, 255$), -1$)
             GOSUB 3000
                           TEMP.0$ = "END"
                          TEMP.0$ = "END"
TEMP.0$ = "OFFLINE"
TEMP.0$ = CHR$(192$) + "OFFLINE"
GOSUB 3000
TEMP.0$ = SYS(CHR$(6$) + CHR$(8$) + STRING$(25$, 0$)
+ CHR$(255$))
             RECEIVE A MESSAGE.

EXTRACT THE USER PARAMETER STRING (THE MEAT OF THE MESSAGE).

ECHO THE MESSAGE BACK VIA OPSER.

IF THE MESSAGE WAS "END" OR "OFFLINE," THEN TELL OPSER WE ARE GOING AWAY, THEN COMMIT SUICIDE.
             GOTO 2030
2999
30001
                          SEND THE MESSAGE.
3010
                          LEN(TEMP.0$) <= 19%
                           LEM(LEMT.04) (= 19)
TEMP.1$ = SYS(SEND$ + CHR$(LEN(TEMP.0$) + 1$) + TEMP.0$)
TEMP.1$ = SYS(SEND$ + CHR$(255$) + LEFT(TEMP.0$, 19$))
TEMP.0$ = RIGHT(TEMP.0$, 20$)
             ELSE
                          GOTO 3010
             SEND THE DATA TO 'OPSER' IN 19 BYTE "CHUNKS."
             RETURN
3999
190001
                          ERROR HANDLING
19005
             THEN
                          RESUME 2030
             BECAUSE THIS IS THE WAY SEND/RECEIVE WORKS.
32767
```

He's learning everything about the business.



Everything.

LOCK-11 is a system security and management package for RST\$.

LOCK-11 gives you absolute control of access by keyboard or user-I.D.

LOCK-11 provides an optional MENU environment that keeps non-privileged users where they belong.

LOCK-11 offers the system manager powerful surveillance utilities that actually improve thruput.

LOCK-11 is very well documented, supported and enhanced regularly.

LOCK-11 is available right now. Circle the response number below for a full set of documentation, or call 215-364-2800.



CIRCLE 120 ON READER CARD

One stop shopping for VAX

No other supplier offers such a winning combination of products for VAX computers. Reliable? You bet! MTBF figures for our VAX products range from 40,000 to 85,250 hours. Add to that competitive pricing (30% or more below DEC). And we're second to none in technical support with third party service available as required.

For disk, tape and communications controllers, the Emulex lineup looks like this:

Disk Products.

FOR THE VAX UNIBUS...

SC12/V—Emulates DEC's RK711 controller combined with multiple RK07 drives on the VAX-11 Unibus. SC21/V-Emulates DEC RM03 (80 MByte) and RM05 (300 MByte) storage subsystems. SC31—A low cost solution that allows you to install and operate large capacity disk drives on the Unibus of any VAX. Handles drives with high transfer rates of 1.8 MBytes per second in the 500 MByte range. Gives the same or greater storage capability than DEC Massbus installations at a fraction of the cost.

FOR THE VAX-11/750...

SC750—This software-transparent, single-board controller allows you to add up to four large disk

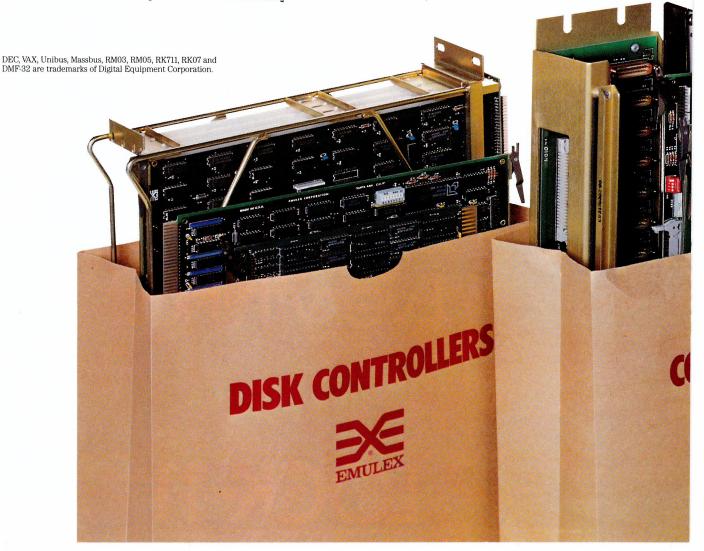
storage units (80 to 675 MBytes) directly to the internal CMI bus. The SC758 lets you add up to eight drives of storage off a single controller. FOR THE VAX-11/780...

V-Master/780—A mass storage adapter that houses one or two SC780 disk controllers, TC780 tape controllers or a combination thereof. Provides an interface and control through the Synchronous Bus Interface (SBI) of your VAX-11/780. Each SC780 disk controller supports up to four disk drives (80 to 675 MBytes). The SC788 is also available to fit in the V-Master/780 chassis and supports up to eight disk drives.

Tape Products.

FOR THE VAX UNIBUS...

TC11/V—Combines with any standard tape drive and the Emulex VMS/UT software driver/diagnostic package to provide reliable, economical tape storage on all VAX-11s. TC12/V—Handles every industry-standard "Pertec" formatted half-inch tape transport, including conventional NRZ/PE start/stop and 1600/3200 bpi start/stop streaming tape drives.



users? Emulex, of course!

FOR THE VAX-11/750...

TC750-A single-board, software transparent controller that interfaces directly to the internal CMI to support 1-4 STC or 1-8 Pertec formatted type drives. Emulates DEC's TM03/TU77 with tape speeds up to 125 ips at 1600/6250 bpi. Supports both "old" and "new" GCR 6250 kinds of drives.

FOR THE VAX-11/780...

TC780-Fits in the V-Master/780 chassis to provide transparent emulation of DEC's TM03/TU77 through the SBI. Supports 1-4 STC or 1-8 Pertec formatted type drives at tape speeds up to 125 ips: 1600/6250 bpi. Both "old" and "new" GCR 6250 technology is supported. In addition, the TC780 is plug compatible with the TC750, offering users sparing convenience.

Communications Products.

FOR UP TO 16 LINES-CS21 SERIES...

CS21/F-Emulates the asynchronous portion of the DMF-32 for use on VAX-11s. Is software transparent with VMS Version 3.0 and above. Handles 16 lines per controller.

Statcon 21-Statistical concentration through the combination of the proven CS21 multiplexer with special microprogramming and the CM22/EX local statistical port concentrator. Handles up to 16 remote lines per statistical concentrator, up to 32 lines per controller.

FOR 16 TO 128 LINES AND MORE-CS11/CS32 SERIES...

CS11/F-Emulates the asynchronous portion of the DMF-32 for use on VAX-11s. Is software and diagnostic transparent, and can handle 16, 32 or 48 lines per controller.

microprogramming and one or more CM22/EX local statistical port concentrators.

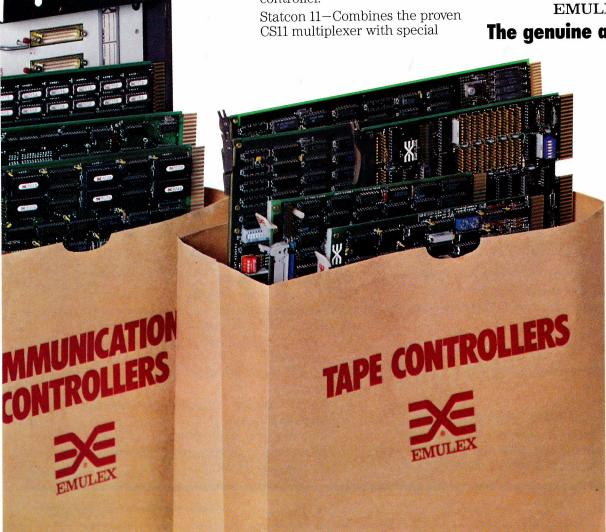
CS32/F-A single-board communications controller that's totally software transparent to DEC's new DMF-32. One CS32 can handle up to 128 lines per controller board. Statcon 32-Combines the CS32 multiplexer with special microprogramming and the CM22/EX local statistical port concentrator. A single CS32 controller board handles an amazing 256 remote and local lines in this statistical

For more information on Emulex products for VAX, call toll-free: (800) 854-7112. In California: (714) 662-5600. Or write Emulex Corporation, 3545 Harbor Blvd., P.O. Box 6725, Costa Mesa, CA 92626.

concentration mode.



The genuine alternative.



See us at **DEXPO WEST** Booth 512 See us at COMDEX Booth 2030

CIRCLE 93 ON READER CARD

GSA Contract #GS-OOC-03589

ATPK MODIFIED

By Gary W. Wong, Systems Engineer, MacMillan Bloedel Reserach, 3350 East Broadway, Vancouver, British Columbia, Canada V5M 4E6

I read with interest "WATCH and RINGME" by Maury Pepper and Greg Wenzel, as a solution to monitoring ATPK when a batch job is done (RSTS Professional, June, 1983).

I have made some modifications to ATPK that:

- 1. Broadcast any text message automatically from the ATPK command file directly to the startup terminal without using extra job slots. The \$MESSAGE command provides a convenient way of monitoring ATPK progress at different stages of the batch job.
- 2. Include the /END switch to send a "job finished" message to the startup terminal when a batch job has been successfully completed.
- 3. Include the /KILL switch to erase the log file if the ATPK job is successful.
 - 4. Include the /HELP switch for casual users.

1.0 Batch and Indirect Command File Processing: ATPK

ATPK is a general purpose indirect command file processor. ATPK can control a job by using a pseudo keyboard (PKO:, etc.) to run programs from a script previously saved in a command file. As an indirect command file processor, ATPK runs at the user's keyboard and reports the job dialogue there as it happens. ATPK can also be used as a simple batch processor, running detached and reporting the job dialogue in a file.

ATPK can be invoked by typing:

RUN \$ATPK

or

 $ATPK < startup \ command >$

Some sites use the optional CCL form: @ <startup command>.

The < startup command > is of the form:

{< log file> = } < command file> {/< switches>} where the {...} are optional arguments. Valid switches are:

/DET

Run ATPK detached. If no log file is specified, output will be to the terminal, regardless of whether or not AT-PK detaches. At the end of a successful detached run, ATPK will kill itself.

/DEL

Delete command file on successful completion.

/RTS:<rts>

Start the controlled job under keyboard monitor <rts>. If <rts> is not a keyboard monitor, no error will be reported and the new job will start under the user's current default run-time system instead.

/LOG

If the user is privileged, any jobs started with the \$LOGIN command will be created regardless of whether or not logins have been enabled.

/CHA{IN}: <program> {:< line>} {= \ < core common>}
On successful completion, chain to <program> at line
number < line> if specified, with < core common>
loaded if specified. Any character not contained in the
< core common> string can be used in place of the
backslash (\) to delimit the < core common> argument. This allows ATPK to chain back to itself if
needed.

/END

Send a "Job finished" message to the startup terminal if ATPK is running detached to a log file. No message is sent unless ATPK successfully completes the detached job.

/KILL

Delete log file on successful completion of detached job. Must be a disk log file.

/TIM(E)

Print the start/stop time of ATPK processing.

/H{ELP}

Print the ATPK help file if any is available.

The controlled job dialogue is normally printed as it happens at the user's terminal, unless a <log file > is specified. If both a log file and the /DETach switch was specified, all dialogue will be sent to the log file instead.

The default extensions for the command and log files are .CMD and .LOG respectively.

1.1 ATPK Control Characters

ATPK usually sends each line of the command file "as is" to the PK controlling the job, except in cases where the four following prefix characters are handled differently:

- \$ as the first character on a line, indicates a ATPK command.
- ! as the first character on a line, indicates a comment.
- † (carat) indicates a control character e.g. †C = Interrupt.
- indicates the next character is not a special character (i.e. an underscore character quotes the following character.

UNITRONIX represents DIGITAL

PDP 11/23
SYSTEMS
SYSTEMS
With 256kB*
With 256kB*
With 256kB*
Winchester, std. 45 ips
Winchester, std.

TERMINALS

VT100 LA12 VT101 LA34 VT102 LA50 VT125 LA100 VT131 LA120

CPUs

PDP 11/23, 11/23+, 11/24, 11/44 & VAX

PCs

INTERFACES MODEMS

COMPLETE DATA SYSTEMS

with Unitronixdeveloped standard or customized applications software.

VT, LA, LSI, PDP, VAX are trademarks of Digital Equipment Corporation



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, ac-

cessories 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

CIRCLE 25 ON READER CARD

1.2 ATPK Commands

ATPK commands begin with a dollar sign (\$) and are as follows:

\$DISABLE LOG

Turn off log file (and/or keyboard) dialogue.

\$ENABLE LOG

Resume log file (and/or keyboard) dialogue.

\$ALLOW NO ERRORS

Abort if any (fatal or warning) errors occur.

\$ALLOW WARNING ERRORS

Abort if fatal errors, allow warning errors.

\$ALLOW FATAL ERRORS

Do not abort if any errors occur (Default).

\$WAIT

Accept a line of input from the keyboard and send it as input to the controlled job. If a program is running at the controlled job after this line has been processed, continue accepting command lines from the keyboard until the PK program has stopped running and is in a monitor wait (i.e. †C state). After this command has been processed, input will resume from the command file.

\$@<command file>

Switch input stream to another < command file>. AT-PK aborts if no command file is found. No nesting of these command files is allowed; ATPK will abort if it finds a "\$@" command in the < command file>.

\$DETACH

Detaches ATPK and can be used after a \$WAIT command to force ATPK to detach after user input at the start of the controlled job.

\$LOGIN {KB{n}:} {(P, Pn)}

Login a keyboard under the user's account or the one specified by (P,Pn). Only privileged users can use this command. This command is useful for logging in again after a /DETach command.

\$MESSAGE < message >

Broadcast < message > to user's terminal if detached. Nothing will be logged in the detached log file. Maximum < message > length is 255 characters. The < message > string can include certain ASCII control characters such as:

- †G Ring terminal bell.
- 11 Next horizontal tab position.
- 1L Vertical Form Feed.
- †M †J Carriage Return Line Feed.

```
<tab>I<tab>M%(30)<tab><tab>USED TO HOLD SYS CALL DATA. &<cr>
    <tabl/tableStableStade FOR BROADCAST TO DETAC
eso.P%(cr)

<a href="mailto:tab">tab</a>
clab | (tab) | (tab) | (tab) |

#i/3000-3099/V<cr>
<a href="mailto:tab">tab</a>
clab | (tab) | (tab) | (tab) |

#2AV<cr>

  *ZAYCCP>
(tab)! &ccr>
*I(cr)*
(tab)! &ccr>
(tab)! &ccr>
(tab)! &ccr>
(tab)! (tab)3100-3199(tab)BROADCAST NESSAGE TO USER'S DETACHED TERMINAL. &<cr>
(tab)!(tab)4MESSAGE &ccr>
(esc)**V(cr)*
    <esc-**\cory
clab>1 &
*#I/900
*#Cab> 1 COMMAND$(11$)/-3C/2$)/V<<pre>
*#I/900
*#I/\ KB$=SWAP$(/Y
     <tab>\ KB$=SWAP$(PEEK(KB.DDB$+2$)) AND 255% &<cr>
*2AV<cr>
  @2AVCr>
<tab>\ TTINTF$=PEEK(KB.DDB$+30$) &<er>
@1<er>
<tab>\ E3.SEND$=CHR.6$+CHR$(-5$)+CHR$(KB$) &<er>
<tab>\ KB.SEND$=CHR.6$+TSYS CALL PREFIX FOR USER'S DETACHED TERMINAL. &<er>
<esc>*H/READ COMMAND$/V<er>
  #H/1500(tab)! &/<cr>
#H/1500(tab)! &/<cr>
#H/10%=INSTR(1%,START*,"=")/V<cr>
<tab>TO%=INSTR(1%,START*,"=") &<cr>
#OAI<cr>
 **OAICCP**
GOTO 4900 IF INSTR(1$,START$,"/H") & Ccr>
Ctab>\ (esc>*DVcr>
Ctab>\ T0$=INSTR(1$,START$,"=") & Ccr>
**Gg/LOO/CHA/TIM/IP/END/KILEVCcr>
Ctab>\ T0$=INSTR(1$,FY,FY//RTS/DET/DEL/LOG/CHA/TIM/END/KIL",LEFT(SW$,4$))/4$ & Ccr>
**Gy/4070/I/, 4200, 4300/Vccr>
Ctab>\ ON T0$=1$ GOSUB 4000, 4010, 4020, 4030, 4040, 4050, 4070, 4200, 4300 & Ccr>
     <tab><tab><tab>! 4900 - .
<esc>*V<cr>
<tab><tab><tab>>tab>! &<cr>
*H/\ EXT$="LOG" &/AV<cr>
<tab>\ GOSUB 12000 &<cr>
*AI
*AI
  <tab>\ LOGFILE$=FILE$PEC$ 4<cr>
<tab>\ LOGFIL.TPf=STATUS AND 255% &<cr>
<tab>\ KB.SEND$=KB.SEND$=CR.LF$="Job"=HUN${(JOB.NO$)+"[" &<cr>
<tab>\ Lob>\tab>\HUN${(SNAP$(PPH$) AND 255%)+","-HUN1$(PPH$ AND 255%)+"] " &<cr>
<tab>\ Lob:T.STO$=" "+CVT${(TO$,-T$)+" message:" &<cr>
<tab>\ Lob:T.STO$=" "+CVT${(TO$,-T$)+" message:" &<cr>
<tab>\ Lob:T.STO$=" "+CVT${(TO$,-T$)+" message:" &<cr>
<tab>\ Lob:T.STO$=" "+CVT$$(TO$,-T$)+" message:" &<cr>
<tab>\ Lob
     <tab>\ LOGFILE$=FILESPEC$ &<cr>
     (tab)\ OPEN FILESPECS FOR OUTPUT AS FILE 3% &(cr)
  CLAD'S OPEN FILESPECT FOR WITHOUT AS FILE 33 & CCT >
#I/COSSUB 2900/AV(cr) >
CLAD'SCLAD'S SAWN LOGIN TO ATTACH TO THE PK. & CCT >
#AI/Ccr) 
CLAD'S LAD'I BEGINNING TIME FOR /END MESSAGE STRING & CCT >
CLAD'S LAD'I BEGINNING TIME FOR /END MESSAGE STRING & CCT >
COSC)*VCCT >
     #H/2020<tab>/V<cr>
 <tab><tab><tab>! 3000 - ODETACI: CONMAND. &<er>
    #AT(cr)
    *Alvery

<tab><tab><tab>! 3100 - $!IESSAGE COHHAND, &<er>

<esc>*V<cr>

<tab>\\tab>\\tab> GOTO 2000 &<er>
     *H/3000<tab>/V<cr>
    3000<tab>! &<er> *AV<er>
    -Avver-V

(tab):(tab):SDETACH CONTAIND &Cor>

*G/Detaching/-9C/Job*;JOB.HOG; "detaching at ";CVTCC(THEC(OI),2%); "/V<cr>

(tab):\ PRINT "Job*;JOB.HOG; "detaching at ";CVTS((THEC(OI),2%);" ... " u<cr>

*G/V RETURE &/V<cr>
  *AVCer>
CABON RETURN GOES

*AVCer>
CABON CABON CABON DETACH URLESS VE ARE ALREADY DETACHED. &Cer>
*AICer>
  3106/tab>1 &@cr>

<tab>1/tab>1058AGE COMMAND &@cr>

<tab>106/tab>2007

<tab>2007

<tab>106/tab>2007

<tab>2007

<tab>106/tab>2007

<tab>2007

<tab>106/tab>2007

<tab
Cab> ReTURN UPLESS DETACRING AND LIN(MESSAGE) &
Cab> Dunary=SYS(PRIV.ORG) &
Cab> Dunary=SYS(PRIV.ORG) &
Cab> Dunary=SYS(ED-SenDG-DATES(OG)+" "+FECLOCES(TIME(OS))+IDENT.STG &
Cab>Cab>Cab>Cab>(Fig.EgsAge+CF.LF3) &
Cab> Dunary=SYS(PRIV.OFF() &
  &cuc)*Vcor>
<ff><ff>
*#I/4000
#I/4000
#I/4000
#I/4000
#OPACTABLE I & Core
#OPACTABLE I & Core
#OPACTABLE I & Core
**Cable Ent. ISSGS="?Invalid switch - "-SW. AND. ARGS+" (Use /H for help)" & <ere>
#I/4070
#OPACTABLE INTERNATION (THE STAMP) & <ere>
#OPACTABLE INTERNATION |
#OPACTABLE IN
  4200 (tab)! & (cr)
```

<tab>! L<tab>KILL.CTL%<tab>KILL LOG FILE ON SUCCESSFUL COMPLETION. &<cr><tec>*H/M%(30)/V<cr>

```
<tab>!<tab>/END SWITCH &<cr>
  <tab>! &<er> <tab>! &<er> <tab>DONE%=-1% &<er> </tab
  <tab>\ RETURN &<cr>
<tab>\ fETURN &<cr>
<tab>\ tab>! FLAG THAT THE /END SWITCH WAS SPECIFIED. &<cr>

  4300(tab)! &(cr)
43U0CaD71 &<Cr2
<tab>! &<cr>
<tab>! &<cr>
<tabKILL.CTL$=-1$ &<cr>
</tab>
    <tab>\ RETURN &<cr>
<tab><tab>\tab>| FLAG THAT THE /KILL SWITCH WAS SPECIFIED. &<cr>

  4900<tab>| &<cr>
<tab>| Cab>| 
    <tab>! &<cr>
<tab>ON ERROR GOTO 4920 &<cr>

  <tab>\ PRINT CHR${(7$) & Cor>}
<tab>\ PRINT CHR${(7$) & Cor>}
<tab>\ OPEN "_SY:$ATPK.HLP" FOR INPUT AS FILE 1% & Cr>
<tab>\ OPEN KB$ FOR OUTPUT AS FILE 2% & Cor>
  \tab>\GTO 4910 &<cr>
\tab>\ PUT #SWAP$(1%)+2% &<cr>
\tab>\ GOTO 4910 &<cr>
    <tab>(tab>! DUMP HELP TEXT BY SWAPPING INPUT TO OUTPUT BUFFERS &<er>
    <er>
4920<tab>CLOSE 1$,2$ &<er>

4920(tab)clubs: 1,25 & ccr>
(tab)\ PRINT *$(no ATPK help file available* IF ERR=5$ & cr>
(tab)\ PRINT *$(no ATPK help file available* IF ERR=5$ & cr>
(tab)\ RSUME 3276 IF ERR=5$ OR ERR=11$ & ccr>
(tab)\ GOTO 19100 & ccr>
(tab)\ GOTO 19100 & ccr>
(tab)\ (Ab) / HELP SWITCH ERRORS: NO FILE, END-0F-FILE, OR ?????? & cr>
(eac)*V<cr>
(eac)*V<cr>
    #H/16000<tab>/V<cr>
    "0AI<cr>
15500<tab>! &<cr>
15500<tab>! &<cr>
**OAI
    <tab>! &<cr>
<tab>! &<cr>
<tab>! &<cr>
<tab>! &<cr>
<tab>! <tab>F N C L O C K $ ( T ) &<cr>

    <tab>! &<cr>
<tab>! &<cr>
<tab>! &<cr>
<tab>! &<cr>
<tab>DEF* FNCLOCK$(T) &<cr>
</tab>
       <tab>\ CLOCK=T/3600 &<cr>
  (tab) CLOCK=T/3600 &cer>
tab) HOURS=INT(CLOCK) &cer>
(tab) CLOCK=60*(CLOCK-HOURS) &cer>
(tab) CLOCK=60*(CLOCK-HOURS) &cer>
(tab) CLOCK=CLOCK-HINUTES &cer>
(tab) SECONDS=INT(60*CLOCK-5)(cer)
15510(tab)TS SECONDS=0 &cer>
(tab) Xab>SECONDS=0 &cer>
(tab) Xab>SECONDS=0 &cer>
(tab) Xab>SECONDS=0 ThEN &cer>
(tab) Xab>HINUTES=HINUTES+1(cer)
15520(tab) THINUTES+95 THEN &cer>
(tab) Xab>HINUTES=0 &cer>
(tab) Xab>HINUTES=0 &cer>
(tab) Xab>HINUTES=0 &cer>

         <tab>\<tab>HOURS=HOURS+1<cr>
       <Lab>\Cab>HOURS=IGURS+ICOr>
| F5530(tab>T HOURS>23 THEN &Cer>
(tab>(tab>HOURS=0Cer)
| 15590(tab>FIGLOCK$=(tab>RIGHT(NUM1$(HOURS+100),2$)+":"+ &Cer>
(tab>(tab>tab>RIGHT(NUM1$(MENUTES+100),2$)+":"+ &Cer>
(tab>(tab>tab>RIGHT(NUM1$(SECONDS+100),2$) &Cer>
| 15590(tab>RIGHT)
         <tab>\ FNEND &<cr>
       *H/32700<tab>/V<cr>
         #0/32790<tab>! &<cr>
#0/32790<tab>! &<cr>
#0/32750<tab>/<cr>
#0/1<cr>
32710<tab>IF DOMES THEM &<cr>
**OMIC THEM &<cr>
**
    32710\tab>IF DOMES THEN &\text{Acc} \
\text{Clab}\tab>IFALLETHE(OT) &\text{Acc} \
\text{Clab}\tab>ELAPSED-#2S(FINAL-INITIAL) &\text{Acc} \
\text{Clab}\tab>ELAPSED-#2S(FINAL-INITIAL) &\text{Acc} \
\text{Clab}\tab>ELAPSED-#2S(FINAL-INITIAL) &\text{Acc} \
\text{Clab}\tab>ELAPSED-#2S(FINAL-INITIAL) &\text{Acc} \
\text{Clab}\tab>ELAPSED-#2S(FINAL-REED) &\text{Acc} \
\text{Clab}\tab>ETHALLINES-#2SESSAGE ### Job finished. Elapsed time was" &\text{Acc} \
\text{Clab}\tab>ETHALLINES-#2SESSAGE ### Job finished. Elapsed time was" &\text{Acc} \
\text{Clab}\tab>ETHALLINES-#2SIE-#UNIK(HOUGRS)-#nbour" IF HOURS>0 &\text{Acc} \
\text{Clab}\tab>ETHALLINES-*CTHALLINES-*#UNIK(HOUTES)-#nbour" IF HOURS>0 &\text{Acc} \
\text{Clab}\tab>ETHALLINES-CTHALLINES-*#UNIK(SECOUNDS)-#second" IF SECONDS>0 &\text{Acc} \
\text{Clab}\tab>ETHALLINES-CTHALLINES-*#S' IF SECONDS>0 &\text{Acc} \
\text{Clab}\tab>ETHALLINES-CTHALLINES-*** IF SECONDS>1 &\text{Acc} \
\text{Clab}\tab>ETHALTINES-*** IF SECONDS>1 &\text{Acc} \
\text{Clab}\tab
         SCP2

32720<tab>IF DETACHEDS AND KILL.CTLS THEN &<cr>
<tab>Ktab>KILL LOGFILES UNLESS LOGFIL.TYPS &<cr>
<tab>Ktab>KILL LOGFILES UNLESS LOGFIL.TYPS &<cr>
<tab>Ktab>LOGFILE (DISK) LOGFILE IF ASKED ON SUCCESSFUL COMPLETION. &<cr>
<tab>Ktab>LOGFILE (DISK) LOGFILE IF ASKED ON SUCCESSFUL COMPLETION. &
</tab>
```

I have also included my TTYCMD program for system managers. This handy utility records essential or all terminal characteristics as set by TTYSET when new terminals are added to the system on a permanent basis. This builds an INIT or ATPK command file automatically, and therefore, minimizes the chore of constantly editing startup files. RUNning TTYSET during INIT with only permanently assigned essential characteristics rather than conventional macros helps reduce startup time on systems with many terminals attached.

```
: TTYCMD.BAS (based on TTYSET.BAS)
: GARY W. WONG
: MACMILLAN BLOEDEL RESEARCH, VANCOUVER CANADA
AUTHOR
INSTALLATION
WRITTEN
                          04-DEC-82
                          21-DEC-82
                           TTYCMD. BAS
```

TTYCMD records essential or all terminal characteristics as TTYCMD records essential or all terminal characteristics as set by TTYSET during INIT or afterwards when new terminals are added to the system on a permanent basis. This utility program will build an INIT or ATPK command file automatically and therefore minmize the chore of constantly editing startup files. Though untested fully, it is the author's belief that RUNNing TTYSET during INIT with only permanently assigned essential characteristics rather than conventional macros will help reduce startup time on systems with many terminals attached. One could write the TTYSET code in macro as suggested in a previous RSTS/E professional article, but... article, but...

COPYRIGHT

Copyright (c) 1982 by MacMillan Bloedel Research 3350 East Broadway Vancouver, British Columbia Canada

This software is offered free of charge to members of the This software is offered free of charge to members of the North American DECUS organization and may only be copied with the inclusion of the above copyright notice. This software or any other copies thereof may not be provided or used otherwise for any monetary or chargeable fee. No title to and ownership of this software is hereby transferred.

The information in this software is subject to change without notice and should not be construed as a commitment by MacMillan Bloedel Research.

```
MacMillan Bloedel Research assumes no responsibility for the
                                                   use or reliability of this software.
900
                                                                         INITIALIZATION
                         DIM S%(30%)
I SYS CALL WORKING ARRAY
                         I$ = "V1.1"
I CURRENT VERSION
920
930
                           EXTEND
                          ON ERROR GOTO 19010
PRINT IF CCPOS(0$)
PRINT "TTYCMD ";1$;CHR$(9$);FNERR$(0$)
                              PRINT "Build TTYSET Command File"
                          I GARY WONG, MACMILLAN BLOEDEL RESEARCH, 04-DEC-82
                         CHANGE SYS(CHR$(6%)+CHR$(-3%)) TO S% \land MAX.KB% = S%(3%)
940
                          I GET MAXIMUM KEYBOARD NUMBER CONFIGURED FOR THIS SYSTEM
                          INPUT "Include console KBO: <Yes>";1$

KBO$ = ASCII(CVT$$(1$,-1$))

KBO$ = 89$ UNLESS KBO$

IF KBO$ = 78$ OR KBO$ = 89$ THEN

KBO$ = KBO$/89$

ELSE PRINT "NO = Operator console will use ASR33 characteristics"

PRINT "YES = Use changes added such as WIDTH, FILL, etc."
                           INPUT "Include PSEUDO KEYBOARDs <Yes>";I$
960
                          INPUT "Include PSEUDU KETBOARUS (les)";;;

PK$ = 80$ UNLESS PK$

IF PK$ = 78$ OR PK$ = 89$ THEN

PK$ = PK$ = 9$ OR PK$ = 89$ THEN

PK$ = PK$/89$

ELSE PRINT "NO = Use default ASR33 characteristics for PK's"

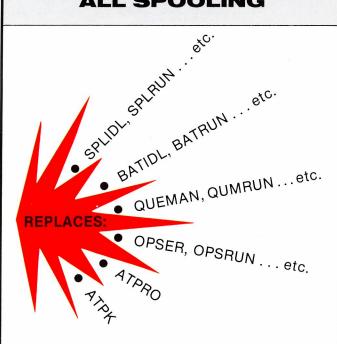
PRINT "YES = Use changes added such as WIDTH, LC OUTPUT, etc."
                                                   GOTO 960
                          INPUT "Include startup default (ASR33) characteristics <No>";I$
\ ASR$ = ASCII(CVT$$(I$,-I$))
\ ASR$ = 78$ UNLESS ASR$
970
                           GOTO 970 &
                           INPUT "INIT or ATPK format <ATPK>";I$
                          FORMAT$ = ASCII(CVT$$(1$,-1$))
FORMAT$ = 65$ UNLESS FORMAT$

IF FORMAT$ = 655 OR FORMAT$ = 73$ THEN
FORMAT$ 
                           IF FORMAT% THEN
                                                  COLON% = 12%
COLON% = 1%
                          ELSE
                           ! SET COLON STARTING POINT FOR FNPRINT$() BASED ON FORMAT SELECTED
                           INPUT "Output line width <72>";OUTPUT%
985
                          \ OUTPUT$ = 72$ UNLESS OUTPUT$
\ IF OUTPUT$ < 40$ OR OUTPUT$ > 132$ THEN
PRINT "Only 40 - 132 characters wide allowed"
\ GOTO 985
                          PRINT "Output to <TTYSET.CMD>";
\ INPUT LINE I$
\ OUT.FIL$ = CVT$$(I$,-1$)
\ OUT.FIL$ = "TTYSET.CMD" UNLESS LEN(OUT.FIL$)
                           OUT.FIL$ = OUT.FIL$ + ".CMD" UNLESS INSTR(1%,OUT.FIL$,".")
 1000
                                                                               MAIN PROGRAM
```

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

CIRCLE 11 ON READER CARD

YACC IS ALIVE AND WELL AND RUNNING IN RSTS

By James B. Wilkinson, Associate Professor, Department of Mathematics, Erskine College, Due West, South Carolina

1.0 Programs that write programs

Many users of the UNIX operating system have become quite fond of two programs called lex and yacc. In response to pattern-matching syntax at their inputs these programs write subroutines in the C language to detect occurrences of the specified patterns. The same inputs also specify what actions are to be taken in response to instances of the various patterns as they are matched. Although they were originally intended as compiler-writing tools, these programs have turned out to be useful in a much wider range of situations. Any program that needs to impose a specific input syntax on its users provides a good example.

1.1 Lex

Lex is a lexical analyzer generator designed to create routines that split an incoming stream of characters into tokens — identifiers, numbers, punctuation, special symbols. The input syntax to lex is in the form of regular expressions. In response, lex generates code to recognize the patterns. The DECUS C compiler kit contains a lex program which is similar to the UNIX version and which will work with UNIX yacc.

1.2 Yacc

Yacc is a parser generator. In response to a restricted kind of context-free grammar it generates a routine that parses a stream of tokens to analyze the structure of the program they express. (This routine is meant to be part of a compiler, remember?) A typical compiler written in this manner will thus have a routine generated by lex and a routine generated by yacc in addition to other routines for building the symbol table and generating the code. The DECUS C compiler kit does not include a yacc program. This fact led to the transporting effort which produced this article.

2.0 DECUS C as a portability tool

The problem we have at Erskine is that we have only one computer. Since RSTS has to be up all the time for administrative uses, our use of UNIX is restricted to vacations and an occasional weekend. This was an unhappy situation for us since we view lex and yacc as important programs for our academic use. The availability of the DECUS C compiler on RSTS suggested a solution: since yacc is written in C, perhaps we could compile our yacc sources from UNIX using the DECUS compiler.

3.0 The seven problems

DECUS C is incompatible in a number of ways with UNIX C. Three of these caused us trouble in transporting yacc. Four other problems also cropped up. In addition there were some difficulties involved in transporting files back and forth between UNIX and RSTS. These are covered in a separate article.

3.1 The macro problem

The yacc source makes heavy use of macros with arguments, which are allowed in UNIX C but not in DECUS C. The solution to this problem is to carry a copy of the DECUS C version of the file 'stdio.h' to the UNIX system, use it while running only the macro preprocessor on the yacc sources, transport the intermediate files (which are now pure C code containing no macros) back to RSTS, and compile them with the DECUS compiler.

3.2 The local declaration problem

UNIX C allows local variable declarations at any block entrance: DECUS C allows local variable declarations only at function entrances. The vacc sources use the UNIX capability a few times. We solved this problem by simply moving each of the offending declarations to the beginning of its enclosing function. In one case the inner block variable used an identifier that duplicated a variable declared at the function entrance. This required us to coin a new identifier to avoid a conflict. For reasons he did not explain, my student assistant chose the name 'cnfuz'.

3.3 The local variable initialization problem

One line of code in the yacc sources uses UNIX C's ability to initialize local variables when they are declared. DECUS C does not allow this, so we had to add an assignment state-

ment to do the initializing.

3.4 The size problem

UNIX allows users access to the separate spaces provided by the PDP-11 for instructions and data, while RSTS does not. This means that yacc in its UNIX form will not fit in a RSTS job space. The size problem was solved by shrinking the global tables used by yacc. Since we aren't writing large applications, this has caused no problems. Perhaps when RSTS supports separate instruction and data spaces for user jobs we will expand the tables again. Another possibility is to build an overlaid version that will fit.

3.5 The unlink problem

Yacc uses the UNIX system call 'unlink' to delete its temporary files. DECUS C has no corresponding function. This required us to write one. It took two small routines in C

and the RSTS.OBJ library that came with the compiler. Actually, at first we wrote a null unlink function and deleted the temporaries by hand.

3.6 The lex problem

The module 'yylex' in the library 'lexlib.obj' distributed with the DECUS compiler defines the external variable 'yylval' which is also defined in every routine generated by yacc. UNIX's lexlib expects to work with code generated by yacc and does not define the variable. This required us to recompile the module 'yylex.c' and place it in the library 'c:yaclib.obj' to avoid doubly-defined globals at link time when using lex and yacc together. There are now two slightly different copies of 'yylex.obj', one in 'c:yaclib.obj' and one in 'c:lexlib.obj'. Since the linker is told to search the yacc library first when both yacc and lex are used, the altered version there will be placed in the program under these cir-



CIRCLE 198 ON READER CARD

able for CP/M. We include a powerful word processing

package and a powerful electronic spreadsheet free! Every-

DECMATION

408/980-1678

CP/M is a registered trademark of Digital Research, Inc.

Santa Clara, CA 95051

3375 Scott Blvd., Suite 422

thing you need is included starting at \$1,250.00.

Visit us at

DEXPO® West 83

Las Vegas Convention Center

October 23-26, 1983

cumstances. When lex is used alone the yacc library is not searched, and everything works as it did before yacc was implemented (with one exception to be noted later).

3.7 The union-type problem

DECUS C does not allow assignments of the form 'x = y' if x and y are structures or unions; the latest version of UNIX C does. Three lines of code in the yacc parser use assignments that will be of this type if you have used union types in the yacc value stack. Since this code is included in the output every time yacc is used to generate a parsing routine, it must be compiled on the DECUS compiler. The default type for the value stack is integer, so there will be no problem if you don't change it to a union. In particular, if you have no idea what this section is about, don't worry about it.

The real solution is to include another module in the yacc library that accomplishes the same thing as the union assignment. Then change the three offending assignment statements into calls to it.

The include file 'c:lex.h' indirectly declares the value stack to have type integer. This declaration must also be removed.

4.0 Procedure

For this project we used Version 7 of UNIX and the original (September 1980) DECUS C compiler, which comes in both RT-11 and RSX versions. Since we use the RT-11 version on our system, the procedure that follows is geared to it. Although we have not tried it, the RSX version should also work.

Throughout this article the function of quoting has been performed by single quotes. Double quotes appear only at those points where a double quote is used in the C code under discussion. In addition, punctuation from the surrounding sentence is never placed inside quotes. That is, whatever is inside the quotes is the actual text to be found or typed.

- 1. Transport a copy of 'c:stdio.h' from RSTS to UNIX.
- 2. Log in to UNIX and change directory to '/usr/src/cmd/yacc'. Create a subdirectory '/usr/src/cmd/yacc/decus' and change to it. Read the 'stdio.h' file from RSTS into this directory. Copy the files named 'y1.c', 'y2.c', 'y3.c', 'y4.c', 'dextern', and 'files' from '/usr/src/cmd/yacc' into '/usr/src/cmd/yacc/decus'.
 - 3. Edit 'dextern' and make the following changes:
 - 1. Insert a new first line: '# define rt11'. Be sure to use lower case 'rt' to match the '# ifdef rt11' in 'stdio.h'. UNIX, unlike RSTS, considers upper and lower cases as different characters.
 - 2. In the second line change '<stdio.h>' to "stdio.h".
 - 3. There is a '# ifdef MEDIUM... #endif' block beginning at line 31. Change all the numbers in this block to one half of what they were. Let one half of 127 be 63.
 - 4. At the end of the file there are names for five files used by yacc. Change them to names acceptable to RSTS. We used 'yacc.tmp', 'yacc.act', 'ytab.c', 'yout.put', 'ytab.h'.

- 4. Edit the file 'files' and make one change:
 - 1. The name of the parser text file should be acceptable to RSTS. We used 'c:yacpar.c'.
- 5. Use the macro preprocessor on the four yacc source files:

Type the command 'cc -P y?.c'. (The 'P' is upper case.)

- 6. Carry the files 'y1.i', 'y2.i', 'y3.i', 'y4.i', '/usr/src/libc/gen/ctype__.c', and '/usr/lib/yaccpar' back to RSTS.
- 7. Log in to an empty account in RSTS, and load the six files brought back from UNIX into the account. The file that was '/usr/lib/yaccpar' should be called 'yacpar.c'. The one that was '/usr/src/libc/gen/ctype__.c' should be 'ctype.c'. The other four files should be called 'y1.c', 'y2.c', 'y3.c', and 'y4.c'. Never mind that their extension was '.i' on UNIX. Edit all six of the files to add a < carriage return > before every < line feed > .
- 8. Use your editor to type in the files 'unlink.c', 'setrO.c', 'accept.c', 'error.c', 'main.c', 'init.c', 'yycopy.c', and 'makyac.cmd'. Listings of these files appear at the end of this article.
- 9. Copy the file 'yylex.c' from the lex account of the DECUS C source tape.
 - 10. Copy the file 'lex.h' from the 'c:' account.
 - 11. Edit 'y1.c':
 - 1. Search for 'state(c){'.
 - 2. Advance 10 lines to 'int *s;'.
 - 3. Save that line and the line 'struct looksets *ss;' following it in a temporary buffer; then delete them.
 - 4. Go back 7 lines toward the beginning to 'struct item *p1, *p2, *k, . . . '.
 - 5. Insert the saved text at this point. Also insert a line that reads 'int cnfuz:'.
 - 6. Advance 34 lines to 'int s;'. Delete this line.
 - 7. The next line should be 'for(s = 0;s < tbitset; + + s) clset.lset[s] = l->look->lset[s];'. Change the five instances of the variable name 's' to 'cnfuz'.
 - 12. Edit the file 'y2.c':
 - 1. Search for 'setup(argc'.
 - 2. Advance 303 lines to 'register tempty;' Save this line in the temp buffer and delete it.
 - 3. Go back 299 lines toward the beginning to 'char actname[8];'. Insert the saved line at this point.
 - 4. Search for 'skipcom(){'.
 - 5. Advance 1 line to 'register c, i = 0;'.
 - 6. Delete the characters '=0'.
 - 7. Insert a new line following where you are. It should read 'i = 0;'.
 - 13. Edit 'yylex.c':
 - 1. Comment out the line 'int yylval = 0;'.
 - 14. Edit 'lex.h'
 - 1. Comment out the line 'extern int lexval;'.
 - 15. Edit the file 'yacpar.c'.
 - 1. Change the string '0%o' to '%d' (not necessary, but it makes sense).
 - 2. Advance 4 lines to '*yypv = yyval;'. Change this line to 'yycopy(yypv, &yyval, sizeof(yyval));'. (yypv should not have an ampersand).
 - 3. Advance 13 lines to 'yyval = yylval;'. Change this

line to 'yycopy(&yyval, &yylval, sizeof(yyval));'.

4. Advance to the end and back up 12 lines to 'yyval yypv[1];'. Change it to 'yycopy(&yyval, &yypv[1], sizeof(yyval));'.

16. Run the ATPK command file 'makyac.cmd'. This will compile and assemble all the code for yacc and the library routines, link the file 'yacc.sav', and create the library 'vaclib.obi'.

17. Move 'yacc.sav', 'yaclib.obj', 'lex.h', and 'yacpar.c' to the 'c:' account.

5.0 Using yacc with lex

Yacc is invoked in the same way as other programs compiled by the DECUS C compiler. A typical command line would look like 'mca c:yacc -d foo.y'. An application written using lex and yacc would ordinarily have a yacc source file, a lex source file, and one or more C source files for the auxiliary routines. Lex processes its source yielding 'lextab.c'. Yacc processes its source yielding 'ytab.c' and 'ytab.h'. The second of the yacc output files contains token definitions and is referenced by a '# include "ytab.h" in the lex source to establish agreement between the two routines. Once all the C files are compiled and assembled, all the object modules are linked against the libraries 'c:yaclib', 'c:lexlib', 'c:suport', and 'c:clib' in that order. In addition, at least in the RT-11 version, sufficient stack space must be left (as usual) using the '/B:xxxx' switch on the first line of the command to LINK.

6.0 Using lex alone

If your lex actions use either of the variables 'lexval' or 'yylval' you will have to include the line 'extern int yylval;' in the declarations section of the lex source file. The change to 'c:lex.h' is what makes this necessary. This is the only change from the previous procedure for using lex alone.

7.0 Files

7.1 Setr0.c

```
/# load x into RO #/
setr0(x)
{ return(x);}
```

7.2 Unlink.c

```
delete file to whose name x points
for RSTS version of yacc
uses dlnfq command and emulator command .dofss
# include <rsts.h>
# include <rserr.h>
# define _DOFSS EMT+0365
                                /# gets file specs in firqb#/
unlink(x)
char *x;
register e;
        clrfqx();
        setr0(x);
        rstsys(_DOFSS);
        if(e=firqb.firqb_errcode){return(e);}
        firqb.fqfun=DLNFQ:
        rstsys(CALFIP);
        if(e=firqb.firqb_errcode){return(e);}
        return(0);
```

7.3 Main.c

```
/# main program to call yyparse #/
main(arge, argv)
int argo;
char ##argv;
        yyinit(arge, argv);
         if(yyparse())
        yyaccpt();
```

7.4 Init.c

- /* yyinit has the ability to set initial values into globals
- before yyparse is called. This is used primarily for debugging
 purposes, but could be used to direct the actions of yyparse() in
 any way desired. The characters after an initial '-' in an
- * argument string determine which global is set. The first argument

DEC

SYSTEMS & COMPONENTS

C.D. SMITH & ASSOCIATES, INC. 12605 E. Freeway, Suite 318 Houston, TX 77015

(713) 451-3112

CIRCLE 54 ON READER CARD

Announcing DSKBLD V3.0 for RSTS/E V8.0

Improved RSTS/E performance

A recent GUARANTEED

A recent \ survey of

150 DSKBLD users produced these overwhelming results:

100% of all respondents reported faster system response.

98% said it meets or exceeds expectations.

98% would recommend DSKBLD.

Here's why:

DSKBLD reduces overhead and file processor demands.

DSKBLD optimizes file clustersizes.

DSKBLD centers directories for faster seeks.

DSKBLD supports RSTS/E V8.0.

AUTHORIZED digital COMPUTER DISTRIBUTOR

DSKBLD is easy to use and fully documented.

DSKBLD pays for itself within a few months.

For complete information about DSKBLD's specifications, applications and unconditional money-back guarantee, call or mail your business card.



Pete Schnebele 1700 Westlake Ave. N. Seattle, WA 98109 206/285-3260

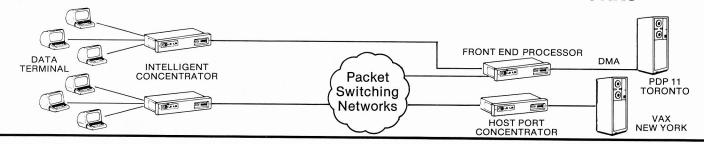
RSTS/E is a trademark of Digital Equipment Corp.

CIRCLE 39 ON READER CARD

What gives your VAX or PDP 11 TOTAL NETWORKING, MORE THROUGHPUT AND BETTER PERFORMANCE?

Cableshare's full line of DEC maintained X.25 computer communication products.

UTILIZES PRIVATE LINE OR PACKET SWITCHING NETWORKS



Our family of LSI-X.25 products are designed to give you everything needed to join the Networking revolution. You will increase the efficiency of your operation while improving your speed, accuracy, flexibility and control.

Front End Processor

Provides VAX or PDP-11 DMA communications multiplexing for up to 127 terminals without any application software changes.

Intelligent Concentrator

Cluster controller for up to 32 asychronous terminals on a private line or an X.25 Network supporting either direct hardwired connection or dial-up modem PAD service. Additional software provides formatted screen mode for

prompting and checking information from data entry clerks. The templates are stored in the concentrator and accessed as needed for greater accuracy and reduced CPU processing time.

Portal

File transfer between VMS, RSTS/E, RSX-11M and RT-11 systems eliminating magnetic tape or costly hardware solutions. Portal can be used with Cableshare's full X.25 networking products to provide file transfer over both public and private packet switched networks.

Electronic Mail

Provides a rapid cost efficient way to exchange business information at any time or place, accessed by standard terminals directly or through any X.25 network.



HEAD OFFICE: Roger D'Hollander, 20 Enterprise Drive, P.O. Box 5880, London, Ontario N6A 4L6; (519) 686-2900 TORONTO: Brian Liska, Suite 1810, 25 Adelaide St. E., Toronto, Ontario M5C 1Y2; (416) 365-7322

```
which does not start with a '-' is taken to be the name of a file
    * which is to replace standard input
   # include (stdio.h>
   extern int yydebug;
   /* more globals could be added here */
   yyinit(argc, argv)
   int arge;
char **argv;
           int i: /# loop counter #/
           char ch; /*switch variable */
            while(argc>1 && argv[1][0] == '-') /# first char of arg is '-' #/
                     for(i=1; ch=argv[1][i]; i++)
                     switch(ch)
                                   /* switch on character */
                       case 'd' :
                       case 'D' :
                       yydebug=1; break;
default: printf("\n wrong switch used\n"); break;
                        /* more cases could be added here */
             argc--; argv++;
           if(argc>1)
              {freopen(argv[1], "r", stdin); argc--; argv++;} /#filename#/
           /* treatments for additional arguments could be added here */
7.5 Accept.c
     /* null routine called by main
     can be replaced by a user-written routine */
     yyaccpt(){}
```

7.6 Error.c

```
/* report error by printing argument string */
# include <stdio.h>
yyerror(s)
        { fprintf(stderr, "%s\n",s); }
```

7.7 Yycopy.c

```
/* copy size bytes from in to out */
yycopy(out, in, size) register char #out, #
register int size;
  for (; size > 0; --size)
*out++ = *in++;
```

7.8 Makyac.cmd

```
CC Y1.C/V
CC Y2.C/V
CC Y3.C/V
CC Y4.C/V
CC CTYPE.C/V
CC UNLINK.C/V
```

```
CC SETRO, C/V
CC ACCEPT.C/V
CC ERROR.C/V
CC MAIN. C/V
CC INIT.C/V
CC YYCOPY, C/V
CC YYLEX.C/V
AS Y1.S/D
AS Y2.S/D
AS Y3.S/D
AS Y4.S/D
AS CTYPE.S/D
AS UNLINK.S/D
AS SETRO.S/D
AS ACCEPT.S/D
AS ERROR.S/D
AS MAIN.S/D
AS INIT.S/D
AS YYCOPY.S/D
AS YYLEX.S/D
YACC = Y1, Y2, Y3, Y4/B:2000/C
UNLINK, SETRO, CTYPE/C
C:RSTS, C:SUPORT, C:CLIB
RUN $LIBR
YACLIB = MAIN, INIT, ACCEPT, ERROR, YYCOPY, YYLEX
```

8.0 A caveat and an offer

We called Bell Labs to confirm that it is all right to do this. However, you must have a UNIX license in addition to RSTS. For educational institutions yacc alone should be worth the license fee. Erskine College will copy the files from section 6 onto your tape for \$12 or onto a tape we provide for \$18. Please don't ask for yacc sources. They are protected by copyright.

9.0 Acknowledgement

This project would, in all probability, never have been completed had Luther Atkinson not been around to do all the dirty work through n iterations until we finally got it right. This article would surely not have been correct had Dr. Bill Groves of the Medical University of South Carolina not been interested. He stayed patiently with it until I remembered all of the steps, and we got it working at his location as well. Billy Glidden of the College of Charleston checked out the procedure on his computer.

UNIX is a trademark of Bell Telephone Laboratories, Inc. RSTS/E, DECUS, PDP, RSX, and RT-11 are trademarks of Digital Equipment Corp.



Four Ways to Secure Your **DEC_System**



with software from McHugh, Freeman and Associates, Inc.

- 1. Data Encryption: Encrypts data, source and task image files
 - Prevents disclosure of sensitive
 - Isolates sensitive programs (from privileged users)
 - · Prevents unauthorized modifi-

2. Menu/Access Authorization:

Allows only menu access to all applications programs

- Limits display to only those items available to user
- Prevents access to the monitor level
- Traces user movements through the menu system
- 3. Keyboard Monitoring: Monitors and interacts with any or all terminals on the system
 - Use at your discretion or as a function of the log-in sequence
 - Optionally creates a log of all user activity
- 4. Password Changing: Changes passwords routinely and quickly
 - Choose from a list of over 3,000 encrypted English words or alphanumeric sequences

Contact us for full product specifications and demonstration packages.



McHugh, Freeman and Associates, Inc. 1135 Legion Drive

Elm Grove, WI 53122 (414) 784-8250

CIRCLE 57 ON READER CARD

RSTS UNDER WRAPS How To Keep The Users From Knowing They Are Using RSTS

By Carl B. Marbach

Some time ago when people were playing the numbers game, HOW MANY?, it was estimated that there were 20,000 RSTS sites, but only onehalf of them knew they were running RSTS. In this age of break-ins and security breaches by kids with personal computers, is RSTS too visible? Wouldn't it be better if a user didn't really know what kind of system he was on?

Most computer thefts and penetrations come from a bogus user dialing in on a phone line. What does RSTS or VAX do when it first sees the carrier detect lead on an RS-232 interface connected to an answer modem:

RSTS V7.2 Computers name KBxx USER:

Immediately you know that it is a RSTS (or VAX) system, what operating system it is running, what keyboard you are and what is expected of you next (USER:). If the guy knows enough to call your computer's number, why tell him all this? The thief (that's what he is) could now program his autodialing personal computer to tell him all the VAX (or RSTS) systems in any telephone exchange. Wouldn't it be better to have the computer answer the phone and then say NOTHING. Make the user type in LOGIN, HELLO, I, or anything that you think should indicate that he wants on and KNOWS SOMETHING ABOUT WHAT HE IS DO-ING! Just type XXHHGG and see what your version of LOGINOUT or LOGIN does with it. Mine does nothing. If you want to make LOGIN (on RSTS) do nothing when it sees the carrier on a modem line, just patch LOGIN like this:

A V7.2 patch to eliminate the "I" into LOGIN from DTR coming up.

This patch will NOP four words in the terminal driver so that LOGIN and the "I" will not be used when dialing in, or when DTR comes "high".

RUN \$ONLPAT command file name:

file to patch: INSTALLED.SIL[0,1] module name: TER base address: TTMCAR offset address: 52

012702 240 000111 240

004767 240 001112 240

XXXXXX tc (up arrow C to install patch)

The patch will become effective next time the system is booted.

LOGIN will now not put an I <CR> into the buffer when it sees the carrier detect from a dial-in line

How about in house users? One method that seems to work well in hiding RSTS from the users is an autologin technique. Make a version of LOGIN that automatically logs specific keyboards into specific accounts. The good news is that the operators won't know it's RSTS, the bad news is that the janitor logs in when he dusts the keyboard.

Once logged in, the user can be kept where he should be by using a good menu system (see insert).

Because VMS does not come with the source of LOGINOUT, some of these procedures are much more difficult. In fact, real security with a VAX system has not yet been worked out. At a recent conference, a RSTS system with all the privileged passwords and accounts listed was made available for hackers and the general public. They were INVITED to break-in. There was only one security breach and that involved opening the computer and attaching KBO: (the console) which had the necessary access (lest we be locked out of our own system). I haven't seen a VAX that is that well protected yet.

But, I have seen a Halloween RSTS system. When you log in it says:

WELCOME TO TSO TIMESHAR-ING IBM 370 V66.7

Trick or treat . . .

Menus and Security A Logical Progression

Security in a multi-user environment can be broken into several components for discussion. The first area is access control. You must be able to limit access to the computer. If there is more than one application, you must limit access to the various applications. This can be handled by the usual account and password setup, or by more elaborate security systems that modify the standard login procedures. These may involve secondary passwords or other user identification schemes. In addition, it is possible to limit activities by keyboard. In other words, accounts payable may only be entered on KB32:, and a user attempting to enter the payables account from another keyboard will be denied even though he knows the password.

Once a user has been identified and allowed into a certain account, the next problem is to control the user's utilization of the programs and data available in that or other accounts. In a typical RSTS system, the 'Ready' prompt actually comes from the BASIC+ interpreter. It means that the user is free to write a program. This may be a bit too user friendly for some commercial applications.

The classic means of controlling a user once logged in is via a 'menu' system. The simplest menu provides the user with a list of functions that may be performed. The intent should be that the user never leave this menu, except to perform one of its functions, and that all the functions never exit to anything but the menu. The trivial case is that all user functions chain back to the menu on completion. Not so

trivial is the requirement that all error conditions also return to the menu. In addition, it is necessary to trap all control-c type interrupts from users and make sure that they all return to the menu properly. The menu itself must be totally immune from control-c, etc.

In RSTS, there is a simple technique that eliminates most of the problems in the above paragraph. At LOGIN, the user is assigned to a default run-time system, typically called MENU. This run-time system is associated with this user. He cannot exit it. Any control-c simply returns him to the MENU. Any error condition from a program will do likewise.

The only problem is to acquire such a run-time system. There have been several examples published in the RSTS Pro. In addition, there are several commercial security products that feature such a system. They have the advantages of generality, documentation and that they work.

A menu run-time system is essentially a menu program that has been written (usually in MACRO) and then turned into a run-time system via the magic of MAKSIL. It is possible to use even the humble BASIC+ to create an RTS, provided you have access to the BACMAC compiler. The only design consideration is that the data containing the allowable programs for this account should not be part of the RTS, but in some sort of a file within the account that is accessible by the RTS. This will avoid having to have a separate RTS for each account.

FALL '83 SHOW/SEMINARS

... YOU'RE INVITED TO TRADE WITH THE LEADERS IN THE DEC* COMPATIBLE MARKET.

The unique DEXOMP format gathers the leading DEC* compatible product competition at one location in an informal atmosphere. The personal time for practical in-depth discussions among DEXOMP principals and qualified clients is unrivaled.

Among those displaying their newest and best in DEC-compatible technologies are:

- · ACC
- Cambridge Digital Systems
- · Datasystems Corp.
- Dataram
- Distributed Logic Corp.
- Emulex
- National Semiconductor
- Western Peripherals

Whether you're a DEC system user or on OEM, you'll discover innovative ways to lift, extend, improve & enhance your DEC systems . . . most economically!

BE ONE OF THE WINNERS DURING THE DESCOMP FALL SERIES.

- Oct. 11 Philadelphia Adam's Mark
- · Oct. 13 Washington, D.C. Sheraton Int'l. Conf. Center
- · Nov. 15 St. Louis Breckenridge Frontenac
- Nov. 17 Minneapolis L'hotel Sofitel

Technical seminars (net—working concepts, innovations and new applications) complement each day-long display/hospitality event.

Contact Dianne for your personal invitation and join the DEC compatible/competitive group.



3000 W. MacArthur, Suite 600 Santa Ana, California 92704

714/966-6713

*DEC is a trademark of the Digital Equipment Corporation. No endorsement of DEXCOMP by DEC is intended or implied.

CIRCLE 190 ON READER CARD

SORTC

THE FASTEST MOST FLEXIBLE. EASIEST-TO-USE **COMPOUNDING SORT** ON THE MARKET TODAY

For more information Call 215-337-3138 or write:



332 W. Church Rd., King of Prussia, PA 19406

CIRCLE 191 ON READER CARD

PERSONAL and PROFESSIONAL, the first independent magazine for DEC personal computer users, is looking for qualified freelance writers.

If you're a writer with good ideas and a knowledge of the DEC personal computer market, call or write:

James L. Trichon, Managing Editor, PERSONAL and PROFES-SIONAL, P.O. Box 114, Springhouse, PA 19477, (215) 542-7008

PERSONAL and PROFESSIONAL is published by the same folks who bring you THE DEC* PROFESSIONAL and VAX/RSTS PROFESSIONAL.

SOFTWARE PRODUCTS for VAX/VMS and PDP-11/RSTS systems

REPORT WRITER

GRS is a powerful report writer and database query facility with English-like commands that can be used by non-programmers. It works with RMS and non-RMS files. A great productivity aid! (VAX and RSTS)

UTILITY

BACKUP/RESTORE DUMPER is a high-speed, easyto-use replacement for the \$BACKUP package (RSTS only).

DISK UTILITY PACKAGE

A set of utility programs which enable you to monitor disk allocation and free space as never before. An absolute MUST for well-structured disks! (RSTS only).



ENTERPRISE TECHNOLOGY 305 Madison Avenue New York, N.Y. 10165 (212) 972-1860 Telex: 177959 Cable: COMPUTERS

CIRCLE 4 ON READER CARD

MICROCOMPUTER TO RSTS COMMUNICATION

By Gerard Kiernan, Manhattenville College, Purchase, NY

A microcomputer using special software can function as a "smart terminal." In an educational environment this approach has a number of advantages:

- 1. Source codes are created off line reducing system load.
- 2. Students have access to a stand alone micro and an intelligent terminal.
- 3. "Distributed Data Storage" is possible. For the cost of a floppy disk each student can have essentially unlimited storage.

Most smart terminal packages provide the capability of uploading and downloading ASCII files to and from the floppy disk of the micro. This is done by using the internal memory of the micro as a buffer to transfer files to and from the floppy disk of the micro. Key sequences (which are not transmitted) are used to perform the following operations:

- < open internal memory as a buffer>
- < close the internal memory buffer>
- < dump the internal memory buffer to a file>
- < load a file from disk to internal memory buffer>
- <transmit internal memory buffer>.

(The methods of uploading and downloading files described here have been tested with a TRS-80 model III 48k microcomputer. There are a number of smart terminal packages available for this micro. A good (free) program can also be found in Reference 1.)

Assume, for example, that PROG.BAS is a BASIC program you have saved and you want to download it to the micro's floppy disk. The following sequence will download the program:

PIP PROG.BAS (No CR yet!)

< open internal memory buffer>

< CR >

(PROG.BAS is now sent to the micro and stored in the internal memory buffer)

< close internal memory buffer>

<dump internal memory buffer to floppy disk file>.
Note that 'Ready' is now part of the file stored on the floppy disk.

To upload this program to a file called PROG1.BAS, use the feature of PIP that allows a file to be created from direct terminal input (PIP FILE.EXT = KB). The following sequence will upload the program:

PIP PROG1.BAS = KB: < CR >

- < load floppy disk file to internal memory>
- <transmit contents of internal memory buffer>

(Program is now sent and stored in PROG1.BAS by PIP)

< CTL Z> (close the file).

If PROG1.BAS is a BASIC program that was previously downloaded, then it will have 'Ready' at the end of the file. This can be edited out, but it will not affect the running of the program if it is left there.

One limitation common to most smart terminal packages is that only ASCII files can be transferred this way. Because of problems with control characters, compiled versions of programs and other binary files cannot be downloaded to the micro.

What is needed is a binary to ASCII program that con-

verts a binary file to an ASCII file that contains only printable characters (see listing of BTOA.BAS). There must also be an ASCII to binary pro-

gram that restores

the original binary

file (see listing of ATOB.BAS).

To avoid any possible problems with control characters. BTOA.BAS creates an ASCII file of characters where each character has an ASCII code greater than 32. Each 16-bit word of the binary file is read into an integer 0%. Since there are only 96 printable characters translate Q% into, Q% is broken into three 5-bit pieces and the sign bit is used as a flag. If Q% < 0, then each 5-bit piece is converted to an ASCII character with code between 33 and 64 (33 is added to the decimal value of the 5-bit piece). If Q% > =O, then the 5-bit piece is converted **ASCII** an character with code between 66

and 97 (66 is added to the decimal value of the 5-bit piece). Since each 16-bit word is converted to three ASCII characters, the size of the ASCII file created by BTOA.BAS is 50% larger than the original binary file.

ATOB.BAS reads the converted binary file three bytes at a time. The decimal value of the first of these characters determines if the integer Q% that these characters came

from is negative.

(Since the programs BTOA.BAS and ATOB.BAS can be used with any files, not just binary files, any file can be "coded" with BTOA.BAS and then "decoded" with ATOB.BAS. This provides some measure of protection against the "amateur" trying to read sensitive files.)

If PROG.ASC is a file created by BTOA.BAS then PROG.ASC is a single stream of printable ASCII characters with no CR or LF delimiters. This presents some problems when this file is downloaded or uploaded since CRs generated by the end of the terminal line will be a problem if they get into the

At this point it is easier to remove the CRs at the micro end before they are put into the buffer. Most smart terminal programs provide a method of translatreceived ing characters before they are put into the internal memory buffer. (This is a feature that is worth having in any smart terminal program you use.) Once provision has been made for the removal of the CRs. PROG.ASC can be downloaded as before. Again, 'Ready' has become part of the downloaded file. This is now more of a problem than before.

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
 runs.
- 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



CIRCLE 29 ON READER CARE

The file can be uploaded in much the same way as was done above with one important change. The file does not contain any CRs so there is no indication to PIP that the end of the very long sequence of characters has occurred. If a CTL Z is done before a CR, the file is closed and is empty!!

To upload the file to PROG1.ASC, the following method can be used:

```
PIP PROG1.ASC = KB: < CR> < load floppy file to internal memory> < transmit internal memory buffer> (PROG1.ASC is now sent as one long stream of printable ASCII characters) < CR> (this is typed at the terminal to avoid an empty file) < CTL Z> (close the file).
```

The keyboard to file feature of PIP is very powerful here. An editor would balk at receiving an extremely long line of characters so fast, but PIP has no problem. I have tested this method with large files (over 50 blocks) at 1200 baud (the maximum speed at which most smart terminal programs function) with no problems.

The fact that the 'Ready' prompt has been appended to PROG1.ASC means that the file size has expanded by one block. This is why line 210 of ATOB.BAS reduces the number of blocks by one. If ATOB.BAS and BTOA.BAS were used without uploading or downloading (or you can figure some way to avoid adding the unwanted junk to the end of the file) then line 210 can be deleted.

If the original binary file is the executable form of a compiled program, then the reconverted ASCII file obtained using ATOB.BAS will not run even though it is byte for byte identical with the original file. For example, if PROG.SAV is the executable code for a compiled PASCAL program and PROG.ASC is the ASCII file created by BTOA.BAS, then PROG1.SAV, created by ATOB.BAS, is byte for byte the same as PROG.SAV, but PROG1.SAV will not run.

The problem is that the reconstructed binary file is seen by RSTS as being an ASCII file and it does not have the proper protection code and run time system. Some changes in the directory of PROG1.SAV are needed. Specifically, the protection code must be increased by 64 and the proper run time system name must be in the directory. A privileged user can use PIP/RE to change the protection code and use the NAME option of UTILITY to change the run time system name.

The program UTIL.BAS allows a nonprivileged user to accomplish the same thing. UTIL.BAS must be compiled with protection code <232> to be available to nonprivileged users. UTIL.BAS was written to handle compiled BASIC, FORTRAN and PASCAL programs. It adds 64 to the protection code of the converted binary file and changes the run time system name in the case of FORTRAN and PASCAL. (No change is needed for compiled BASIC programs).

This program can be changed to accommodate other languages using other run time systems. WARNING!! It is possible to corrupt a disk by messing up a UFD. Before you attempt to change UTIL.BAS you should be familiar with the structure of the UFD (see three part series by Scott Banks in RSTS Professional, Reference 2).

References

- 1. "A Terminal Program for the TRS-80 Model III", Ralph James, BYTE Feb., 1983 p. 458-567.
- RSTS Professional Vol. 1, No. 1, p. 30; Vol. 2, No. 1, p. 45 and Vol. 3, No. 3, p. 38.

```
*********
                                 BTOA. BAS *******
         This program converts a binary file to an ASCII file of printable characters.
Gerard Kiernan
            Manhattanville College
              DIM BZ (32)
          INPUT "BINARY FILE TO BE CONVERTED"; F$
\INPUT "NAME FOR ASCII FILE"; F1$
\OPEN F$ FOR INPUT AS FILE #3, RECORDSIZE 1024%.
\OPEN F1$ FOR OUTPUT AS FILE #4, RECORDSIZE 1536%.
          W$=RIGHT(SYS(CHR$(6%)+CHR$(-8%)+CHR$(3%)),7%)
\%\Z=ASCII(W$)+SWAP%(ASCII(MID(W$,2%,1%)))
! L%=SIZE OF INPUT FILE #3 IN BLKS
             R%=L%/2%
          \F%=L%-(R%*2%)
\F0R I1%=1% TO R%
\GET #3,
           \BZ=5112
           \G0SUB 5000
\NEXT 11%
             R%=NUM OF 1024 BYTE BUFFERS IN #3
             F%=FLAG FOR 512 BYTE PART LEFT
B%=INITIAL VALUE FOR FIELD LOOP
             IF F%=0% THEN 32767
           ELSE CLOSE #4

\OPEN F1$ AS FILE #4, RECORDSIZE 1024%, MODE 2
          \BX=255%
          \GOSUB 5000
\GOTO 32767
             IF FLAG F%<>0% THEN CLOSE AND REOPEN OUTPUT FILE
           SO SIZE OF ASCII FILE IS EXACTLY RX*2%+F%
              FOR P%=0% TO B%
         FIELD #3,P%*2% AS D$,2% AS A$
\FIELD #4,P%*3% AS D1$,3% AS A1$
\QX=CVT$%(A$)
          GOSUB 10000
          \LSET A1$=C$
\NEXT P%
\PUT #4,
             GET 2 BYTES AT A TIME FROM #3
CONVERT TOP 3 ASCII CHARACTERS INTO C$
PUT IN #4
          IF Q%<0% THEN W%=33% ELSE W%=66%
B%(1%)=Q% AND 31%
\B%(2%)=(Q% AND 992%)/32%
 10000
          \BX(3X) = (QX AND 31744X)/1024X
\BX(2X) = BX(2X) + WX FOR ZX=1X TO 3X
\C$=CHR$(BX(1X)) + CHR$(BX(2X)) + CHR$(BX(3X))
            TAKE 5 BYTE PIECES OF Q% AND PUT INTO B%(Z%)
LOAD B%(Z%) WITH PRINTABLE ASCII CHARACTERS
RETURN 3 CHARATER ASCII STRING IN C$
32767
              CLOSE #3,#4
         \END
          *******

ATOB.BAS ********

This program ( the inverse of BTOA.BAS ) converts an ASCII file back to a binary file.
        *********
           Gerard Kiernan
           Manhattanville College
         Purchase, N.Y. 10577
         DIM B$(3%),B%(3%)
INPUT "ASCII FILE TO CONVERT BACK TO BINARY";F$
\INPUT "NAME OF CONVERTED BINARY FILE";F1$
\OPEN F$ FOR INPUT AS FILE #3,RECORDSIZE 1036%
\OPEN F1$ FOR DUTPUT AS FILE #4,RECORDSIZE 1024%
            W$=RIGHT(SYS(CHR$(6%)+CHR$(-8%)+CHR$(3%)),7%)
         \LX=ASCII(\w\s)+SWAP7\(ASCII\(MID\(W\s,2X,1X)\))
! LX=NUMBER OF BLOCKS IN THE INPUT FILE #3
210
            LX=LX-1X
         ! ONE EXTRA BLOCK ADDED IN TRANSMISSION R%=L%/3%
         \F%=L%-(R%*3%)
         \B1%=511%
         GOSUB 5000
! R%=NUMBER OF 1536 BYTE BUFFERS IN #3
! F%=FLAG FOR 1024 BYTE PART LEFT
            B1%=INITIAL VALUE FOR FIELD LOOP
            IF F%=0% THEN 32767
         ELSE CLOSE #4
\OPEN F1$ AS FILE #4,MODE 2
\R%=1%
\B1%=255%
         GOSUB 5000
         \GGISUB 5000
\GGITO 32767
! IF F%=2 THEN CLOSE #4 AND REOPEN IT WITH
! RECORDSIZE 512 SO THE SIZE OF THE CONVERTED
! BINARY FILE IS THE SAME AS THE ORIGINAL
           FOR I%=1% TO R%
       \GET #3,
\FOR P%=0% TO B1%
        \FIELD #3,P%*3% AS D$, 1% AS B$(1%),1% AS B$(2%),1% AS B$(3%)
```

\FIELD #4,P%*2% AS D1\$,2% AS C1\$



LOGICRAFT Inc.

3 Graham Drive, Nashua, N.H. 03060, 603-888-4448

ARE YOUR USERS COWED BY **DEC'S EDT MANUAL?**



We turn DEC's Computerese into Computer Ease

Finally there's an EDT tutorial that non-technical and technical users alike can feel comfortable with. And more productive too! Unleash the power of EDT.

"UNDERSTANDING EDT" costs \$49.95 (with quantity discounts). To order, or for more information, contact:

Computer Ease Publishing

P.O. Box 390272 Mountain View, CA 94039 (415) 961-0273

CIRCLE 211 ON READER CARD

CBEDIT The \$200 RSTS/E* WORD PROCESSOR

Version 3.0 includes list processing, optional operator input, on-screen help and directory management plus total device independence. All files are standard ASCII with no control codes. Data loss protection on system crash or operator error.

Window edit includes add, insert, global locate, global change, block replace, block delete, block document merge, etc. move, formatted output with page numbers, headers, footers, margins, indentation, justified text, underscore, super/subscript, center, multiple spacing, Automatic or manual pagination with unlimited document length.

Table driven Video and Printer modules support VDTs (including 132 column and 66 line) and any printers. Mixed units on same system are OK.

Price: \$200 on 9-track/800 bpi plus shipping. Other media available. User's manual and Basic-Plus* source are included.

> T. F. Hudgins & Associates, Inc. P.O. Box 10946 Houston TX 77292 Wes Seago 713/682-3651

*TM Digital Equipment Corporation

CIRCLE 9 ON READER CARD

```
\G0SUB 10000
              \LSET C1$=C$
             \PUT #4,
              RETURN
                 LOOP THROUGH NUMBER OF 3 BLK BUFFERS IN INPUT FILE
READ 3 BYTES FROM THE INPUT BUFFER
CONVERT TO 2 BYTE BINARY NUMBER (SUBROUTINE 10000)
PUT CONVERTED BINARY NUMBER IN OUTPUT BUFFER
             IF ASCII(B$(1%))>64 THEN W%=66% ELSE W%=33% IF ASCII(B$(1%))>64 THEN S%=6% ELSE S%=32767%+1% B%(J%)=4% THEN S%=6% ILY TO 3% \QX-B%(J%)+B%(J%)+B%(J%)=1% FOR J%=1% TO 3% \QX-B%(J%)+B%(2%)*32%+B%(3%)*1024%
             \Q%=Q% OR S%
             \UZ=EUX OR SX
\C$=CUX$GX)
\RETURN
! 3 ASCII CODES TO BE CONVERTED ARE IN B$(JX)
! SX SETS THE 16TH BIT
!RETURN CVT FORM IN C$
                   CLOSE #3,#4
32767
             \FND
```

!******** UTIL.BAS *******

```
This program modifies the directory of a converted binary file to make it a runnable program. NOTE !!! see the warning in text about changing directories.
              Gerard Kiernan
Manhattanville College
               Purchase, N.Y. 10577
                  PRINT " NAME, EXT OF CONVERTED BINARY FILE ";
            PRINT " NAME,EXT OF CONVERTED BINARY FILE ";

\INPUT LINE F5*
\F55=CVT**(F5*,255%)

\IF LEN(F5*)=0% THEN 32767

\CZ=INSTR(1%,F5*,",")

\IF C%=0% THEN PRINT "MUST SEPARATE NAME AND EXT BY , "

\GOTO 100

F$=LEFT(F5*,C%-1%)
\\E$=RIGHT(F5*,C%+1%)

\! GADT F*=FILE NAME

\AND F*=FYT
                 AND ES EXT
             INPUT "B FOR BASIC, F FOR FORTRAN, P FOR PASCAL ";R$
\R$=CVT$$(R$,2%)
200
              \R5%=0%
IF R$="F" THEN R5%=29631%
                 \R6%=-15936%
IF R$="P" THEN R5%=25659%
220
                  \R6%=4852%
IF R$="B" THEN R5%=1%
                 IF R5%=0% THEN PRINT "ENTER B ,F OR P"
\G0TD 200
LOAD R5%,R6% WITH RAD 50 FORM OF RUN TIME SYSTEM
NAME OF THE CONVERTED FILE
240
             !
Y$=SYS(CHR$(6%)+CHR$(15%))
\P8%=SWAP%(CVT$%(MID(Y$,6%,1%)))
\N8%=SWAP%(CVT$%(MID(Y$,5%,1%)))
\P8$=\NUM1$(P8%)
\N8$=NUM1$(N8%)
             \AB$="["+P8$+","+N8$+"]"
\! A8$=[PR0J,PR0G]
             OPEN AB$ FOR INPUT AS FILE #3,MODE 8192
\DIM #3, UX(3583%,7%)
\CZ=UX(31%,0%)
\FUX(0X),0%)=0% THEN PRINT " DIR EMPTY "
\GOTO 32767
! OPEN UFD AS VIRTUAL ARRAY FILE READ ONLY
! CHECK IF DIR IS EMPTY
                   PX=FNL % (U% (P% - 0%))
             PT=FNLX (UX (PX, OX))
F1$=RAD$ (UX (PX, TX))+RAD$ (UX (PX, 2X))
F1$=RAD$ (UX (PX, 5X))

VE ($=RAD$ (UX (PX, 4X))

VE ($=RAD$ (UX (PX, 4X))
             PSZ=UX(PX, 4X)
P9Z=RIX(UX(PX, 6X))
CLOSE #3
! PSX= PROT/STATUS WORD IN NAME BLOCKETTE
! P9X=LINK TO ACCOUNTING BLOCKETTE
! CLOSE UFD IN WRITE PROTECT MODE
                    OPEN A8$ AS FILE #3, MODE 16384
             \UX(PX,4%)=SMAPX(SMAPX(P5%)+64%)
\IF R$="B" THEN 32767
ELSE UX(P9%,5%)=R5%
                                  \U% (P9%, 6%) =R6%
                 \UX(P9X,6X)=R6X
\SOTO 32767
REOPEN UFD IN WRITE MODE
ADD 64 TO PROT CODE
IF RUN TIME SYS IS BASIC END
ELSE LOAD PROPER RUN TIME SUSTEM
NAME IN ACCOUNTING BLOCKETTE
              DEF FNL%(X%)=(((X% AND 3584%)/512%)*C%
+(SWAP%(X% AND -4096%)/16%))*32%
+((X% AND 496%)/16%)
                 LINK WORD FUNCTION
32767 CLOSE #3
```

\END

DYNPRI.MAC

By Jeff Corbett, Greenville, SC

The BASIC-PLUS DYNPRI program has been floating around since at least RSTS V6B and is probably the most frequently running, most oft found in memory of any CUSP. Because of this, many people have taken the time to write it in MACRO-11 so that it executes faster. I am one of those people. In addition, if you have RSX emulation genned into your monitor, my DYNPRI.TSK will run in 1K words. (If not, you can make it a runtime system and run it in 2K.) I find this much easier to live with than the 4K BASIC-PLUS version.

The principles of operation are essentially the same as the DECUS version of DYNPRI with one small exception. Since Oconee County (I was a programmer/operator for Oconee County at the time of this writing) has quite a few modem ports, hibernating detached jobs were becoming a common occurrence. To prevent having to continually clean hibernating jobs from the system, I included a routine to kill the offending job after sufficient time has passed for a reattachment. This code may be bypassed by removing the definition for "KILLHB" in the constants section. The time between cycles, first job to reschedule, etc. may all be changed easily by changing the appropriate constant. If your console should be something other than KBO:, a description of what to change may be found in the SNDTXT routine. The only thing that should be necessary to make DYNPRI a 1K runtime system with a 1K user space (to keep from having the RSX runtime system in memory if you haven't RSX emulation in your monitor, for a savings of 2K) is to make the data section at the end a separate .PSECT and to add a pseudo-vector region.

DYNPRI was written and runs on a PDP-11/34A running RSTS V7.0-07. Slight modifications may be needed on other systems. (If a new scheduler comes out for RSTS, this article will become worthless for you who recieve the new release.) I would be interested in hearing about any problems or bugs that you encounter.

```
.PSECT DYNPRI
.ENABL LC
.LIST BEX

TITLE DYNPRI, (Dynamic Priority Allocator), 02, 20-Apr-83, JMC

DYNPRI

Modeled after that which appeared (in BASIC-PLUS) in the RSTS SIG newsletter

Please see this or DYNPRI.BAS for theory of operation, etc.

AUTHOR:

Jeff Corbett, County of Oconee, South Carolina

EDIT:
2-Dec-82 JMC Originial Version
```

```
20-Apr-83 JMC Clean up looks
    ASSEMBLY INSTRUCTIONS:
            MAC>DYNPRI, DYNPRI=$COMMON, [PPN]DYNPRI
            RUN STKB
            TKB>DYNPRI, DYNPRI=DYNPRI
            TKB>STACK=64
    COPYRIGHT:
            This software may be copied by anyone, anywhere, for any reason,
            provided that the author's name and this lack of copyright notice
is included. Neither the author nor the County of Oconee, South
Carolina assumes any responsibility for the use or reliability of this
            software, nor do they make any commitment for its support.
            PEEK macro, generates code to peek WHERE and place the information returned in \ensuremath{\mathsf{GOES}} .
 .MACRO PEEK
                       WHERE, GOES
             MOV
                       WHERE, @#XRB
              . PEEK
             MOV
                       €#XRB. GOES
            PEEK
KILLHB
                                  177777
            SETBTS
                                  177777
                                 077776
8.* 2; Start with job 8.
25.* 2; Reserve table space for 25. jobs
30.; Sleep 30. seconds
            WAITBT
            FRSTJB
            SLEEP
            PERCENT =
            L2SECS
            PRTOR1
                                       -8.
            PRIOR3
            BURST1
            BURST3
            OPERWT
            L1 TCKS
                         PERCENT # SLEEP/10.
            L1TCKS = PERCENT * SLEE
L2TCKS = L2SECS / SLEEP
           Main Code Area (DYNPRI)
                                            ;Clear the FIRQB (Returns: R4 => FIRQB+FQFUN)
DYNPRI: CALL
                      CLRFOB
                       #UU. TB1,(R4)
                                             "Get Monitor tables I" SF
            .UUO
                       #FTROB+14.RO
                                            :Point RO to first useful information
                      (RO)+, JOBTBL
(RO)+, JBSTAT
                                            ;Get (JOBTBL)
;Get (JBSTAT)
           MOV
                                            ;Get (JBMAIT)
;Get town job number
;Mask off high byte
;Put Highest Priority in PRIOR
;Put burst-1 in BURST
                      (RO)+, JBWAIT #518., RO
            PEEK
                       #177400,RO
#PRIOR1,PRIOR
           MOV
                       #BURST1.BURST
                                            ;Set our priority
;Clear the XRB (Returns: R4 => XRB+0)
                       SETPRI
           CALL
                      CLRXRB
           MOV
                       #400,(R4)
                                            ;Set special run priority bit
;Set special run priority
                                            ;Set special run priority
;Clear the FIRQB (Returns: R4 => FIRQB+FQFUN)
                      CLRFOB
           CALI.
                                            ;"Detach" SF
;Detach ourselves
                       #UU.DET,(R4)
            .uuo
                      SNDTXT, R5, <detmsg> ;Tell KBO: we're up and running
           Main Processing loop
CYSTRT: CALL
                      CL.RXRB
                                            :Clear the XRB (Returns: R4 => XRB+0)
                                            ;Sleep time
;Rock a bye baby, etc.
                      #SLEEP,(R4)
            SLEEP
                      #FRSTJB. RO
                                            ;Place first Job nr in RO
           MOV
                      GETSTT
                                            :Get this job's status
LOOP:
           CALL
                                            ;Did we get a running job?
;Yes- Do something with it
           BNE
                      5$
                                            :No- Are we at the end of the job table
                      CYSTRT
                                            ;Yes- Recycle
           BEQ
                                            :No- Let's do the next job
           BR
                      EOLOOP
                      PRIOR, R1
          MOV
                                           ; Move it somewhere faster
5$:
                                            ;Is the "Special" bit set?
```

What you don't know about your DEC* computer can't help you.

DEXPO West 83 can.

400 Reasons to Attend

Wouldn't you like to get just a little more out of your DEC* system? All it takes these days are the right DEC-compatibles. And you'll find them by the thousands at DEXPO West 83. Your most complete DEC-compatible resource.

Want *more* reasons? How about *more* compatible software, *more* compatible hardware, and *more* compatible services? **DEXPO** West features nearly 400 vendors with more for every DEC computer. Personals, Micros. Minis. Mainframes. All the latest DEC-compatibles designed to make your information systems work a little harder. And a lot smarter.

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

Attending DECUS*? Then, DEXPO is Free

If you're planning to be in Las Vegas for the DECUS meetings, make sure DEXPO West is on your schedule for Sunday . . . before

the meetings start. Of course, you'll probably want to come back on Monday, Tuesday and Wednesday, too. So be sure to use our five-minute shuttle bus services. It's free. And so is the Show when you bring your DECUS badge.

Plus 60 In-Depth Sessions on How to Put the Latest DEC-Compatibles to Work

DEXPO® West 83

The Fourth National DEC-Compatible Industry Exposition

Las Vegas Convention Center October 23-26, 1983 Sunday-Wednesday

Here Are Just Some of The 400 Vendors You'll Find at DEXPO West 83 . . .

Exhibitors as of August 1, 1983:

Able Computer Absco Systems Corp ACC

Access Technology Accord Computer ADAC Corporation

ADAPSO Advanced Data Management

Advanced Electronics Design AGS Management

Systems, Inc. AIM Technology Air Filtration Products

AMCOR Computer Corporation

American Computer Group, Inc.

American Computer Hardware Corp. Ampex Corporation

Andromeda Systems, Inc. Anti-Glare Material

Canada, Inc. Applied Information

System Inc. Ashton-Tate

Ask Computer Systems, Inc. Atlantic Research Corporation

Aviv Corporation Ball Electronic Systems Div.

Barber, Glenn A. & Associates Batelle Memorial Institute

Braegen Corporation Breuer & Company Britton-Lee, Inc

BRS Inc. Bubbl-Tec Busology, Inc.

Cableshare, Inc. Cadmus Computer

Systems Inc California Computer Group Inc Cambex Corporation

Cambridge Digital Systems Camintonn Corp.

Canoga Data Systems Care Information Systems, Inc. Chrome Computing

CIE Terminals Inc Cincom Systems Clyde Digital Systems, Inc.

COBAR, Inc Collier-Jackson

Comdesign, Inc COMPAL

Compu-Share, Inc. Compu-Tome, Inc. Compouter Dataco, Inc. Computer Datacom, Inc. Computer Graphics World

Computer Information Systems

Computer Intelligence Computer Marketing Associates

Computer Parts Exchange Computer Products Marketing Computer Software Design Computer Systems Corp.

Computer Systems Development, Inc. Computer Techology Group Computers-R-Digital/

Sys-R-Dig. Connector Technology, Inc. Contel Information Systems Cortex Corporation

Crosspoint Systems, Inc.

Cyberchron Data Processing Design, Inc.

Data Star Datalease Datamation Datanex, Inc. Dataplotting Services Inc.

Data Sources

Dataram Corporation Datasystems Corporation

Datatrend Datum, Inc. SSD DEC Professional decComp Decmation

Digital Communications Associates, Inc.

Digital Design Digital Engineering, Inc. Digital Equipment Corp. Digital Information Sys. Corp.

Digital Review Digital Sound Corporation Disc Tech One

Distributed Logic Corporation

DYSAN Inc Eaton Corp. EEC Systems, Inc. Emulex Corporation Imaging Technology, Inc. IMSL. Inc.

Information & Sys. Resch., Inc. Infosphere, Inc.

Infosystems

Inmac-Data Communications Integrated Solutions, Inc. Intelligent Systems Corp. Interactive Systems, Inc. Interactive Technology, Inc. Interlink Computer Sciences J.B. Hotze & Company

John McLean & Associates Jupiter Systems Inc.

K Systems Kaufman Research Kineticsystems Lee Data Corp Link Data Logicraft Lynch Comm Sys

Data Comm Dept. Marc Software International

Marway Products MCBA MDB Systems, Inc.

Megatek Corporation Metheus Corporation Micom Systems, Inc. Micro Technology, Inc.

Micro-Term. Inc Microlink, Inc.

Call 609-799-1661 For Registration **Information**

Enterprise Technology Corp. Equitable Computer Marketing

Evans Griffiths & Hart, Inc. Everest Electronic Equipment

Exceptional Business Solutions Fasbe Group, the

Fingraph First Computer Corp. Franklen Computer

Systems, Inc. Gandalf Data Inc GEJAC. Inc.

General Electric General Research Corporation General Terminal Corporation

Grant Technology Systems Corp. Group III Electronics Grumman Data Systems

Halcyon Hamilton Micro Systems Hamilton Rentals

Hardcopy Magazine

Hearne, J.P. & Associates Henco Software, Inc. Higher Order Software Hot Line, Inc.

IBM

ICS Electronics Corporation

Microsystems Engineering Corporation Microtec Research Inc.

Midcom Corporation Midwest Systems Mini Computer Modeling, Inc.

Minio-Micro Systems Minitab Project Minntronics Corporation

Minolta Corp. Monolithic Systems Corp Moxon Electronics

N.Y. Repair Depot Inc. National Accounting

Equip. Corp. National Control Systems National Information Systems National Instruments

National Semiconductor Netcom Products, Inc. Network

Newman Computer Exchange North County Computer

Services Inc. Northern Technologies Numerical Algorithms

Group, Inc. Nyplan

Oasys, Inc. One Stop Computer Shop

Opcode Oracle Corporation Oregon Software P-Stat. Inc. Parallex Systems Inc.

Peed Publishing, Inc. Pennington Systems Inc. Peripheral Parts Support, Inc. Phoenix Computer Graphics

Pick Computing Machinery Plessey Peripheral Systems Plycom Services Inc Proteon Associates, Inc.

Pulizzi Engineering Inc. Quadrex Word Processing Systems

Quantum Communications Quasar Systems Ltd. Ranyan Computer Enhancement Systems

Raxco, Inc Relational Technology

Reliance Electric RGTI Systems Software Ross Systems, Inc. S&H Computer Systems, Inc.

SAS Institute, Inc. Saturn Systems, Inc. Scientific Information Scientific Micro Systems Sigma Information Signal Technology

Simpact Sky Computers, Inc. Smith, C.D. & Associates, Inc. Softool Corporation

Software House Software Response Software Results Corp. Software Vision

Spectra Logic Corp. Spectra Systems, Inc. SPSS, Inc

Standard Engineering Corp. Standard Memories Trendata Standard Software Corp Stone Mountain Computing

Strategic Information Sunberry System Industries

Tandem Computers, Inc. Technical Magic, Inc. Tektronix, Inc. Televideo Systems, Inc.

Terminal Display Systems, Inc. The Information People The Wollongong Group

Theta Business Systems Trimm Industries

Tymshare Computer System & Support Division Tymshare Database Systems

Division U.S. Design Corporation Unitronix

Up-Time Systems, Inc. Utility Software Venturcom

Viking Computer Corporation Viking Software Services Virtual Microsystems

Warner-Eddison West Coast Computer

Exchange Western Data Systems Western Peripherals

WHY Systems Incorporated Xerox Computer Services ZIA Corporation

From Rainbow* to Vax* . . . DEXPO West Serves the Entire DEC User Community

Registered trademark of Digital Equipment Corp

```
BNE
                       EOLOOP
                                             :Yes- Skip it
                                                                                                                                           BURST new runburst for job
             CMP
                       R1, #PRIOR1
EOLOOP
                                            ;Is is above PRIOR1?
;Yes- Skip it
            BGT
                                                                                                                                 Output: None
                       R1, #PRIOR3
EOLOOP
                                            ;Is it below PRIOR3?
;Yes- Skip it
             CMP
                                                                                                                                           (Destroys R1, R4)
             BLT
            BIT
                       #7,R1
                                             ; Is a Monitor controlled bit set?
                                            ;No- Keep going
;Is it below PRIOR1?
;No- Record its time and go on
                       10$
R1.#PRIOR1
                                                                                                                      SETPRI: CALL
                                                                                                                                                                ;Clear the FIRQB, (Returns R4 => FQFUN); "Change Priority/Run Burst/Size" SF
                                                                                                                                           CLRFOB
            CMP
                                                                                                                                 MOVB
                                                                                                                                           #UU.PRI,(R4)+
            BGE
                                                                                                                                MOV
                                                                                                                                                                ;Copy Job nr
;Divide by two
;Place in FIRQB
;Flag to Change Priority
                                                                                                                                           RO,R1
            BR
                       RAISE
                                             ;Yes- Raise it to PRIOR1 and go on
                                                                                                                                 ASR
                                                                                                                                 MOVB
                                                                                                                                           R1,(R4)+
            CMP
                       #PRIOR1,R1
  10$:
                                             ;Is it PRIOR1?
                                                                                                                                           #377,(R4)+
PRIOR,(R4)+
                                                                                                                                 MOVE
            BEQ
                       LEVEL1
#PRIOR2,R1
                                            ;Yes- Handle it
;Is it PRIOR2?
                                                                                                                                                                 New Priority
            CMP
                                                                                                                                           #377,(R4)+
BURST,(R4)+
                                                                                                                                MOVB
                                                                                                                                                                ;Flag to Change Run Burst
            BEO
                       LEVEL2
                                             ;Yes- Handle it
                                                                                                                                MOVB
                                                                                                                                                                :New Run Burst
                       LEVEL 3
                                            ;No- Must be PRIOR3, Handle it.
                                                                                                                                CLRB
                                                                                                                                                                ;Flag NOT to change size
;Do the work
                                                                                                                                           (R4) +
                                                                                                                                 IIIIO
  ; PRIOR1 - Jobs that did not use more than PERCENT percent of the CPU since
                last cycle are left at this priority
                                                                                                                                GETSTT Get status of a job
 LEVEL1: MOV
                      CPUTME, R1
                                            :Get this CPU time
                       TIMUSD(RO),R1
                                            ;Subtract that of last time
            SUB
                                                                                                                                Call:
                                                                                                                                          MOV
                                                                                                                                                     JOBNR # 2.RO
            CMP
                      R1.#L1TCKS
                                            :Are we less than BUSY ticks?
                                                                                                                                          CALI.
                                                                                                                                                     GETSTT
            BLE
                                             Yes- Record time and go on
                      STATE
                                            ; No- But is job doing anything now?
            TST
                                                                                                                                Input: RO
                                                                                                                                                     Job number (times 2) of job to work with
                                            ;No- Record time and go on
;Yes- Clear its LEVEL2 residency count and...
            BNE
                      RCDTME
                      CYCLES(RO)
                                                                                                                                Output: RO
                                                                                                                                                     =0
                                                                                                                                                               End of job table
            MOV
                      #PRIOR2, PRIOR
                                                   Drop it to PRIOR2
                      #BURST2, BURST
                                                                                                                                          R1
                                                                                                                                                               This job is not running, don't schedule This job is running, schedule him
            BR
                                            :Go schedule
                                                                                                                                                     <>0
 ; PRIOR2 - A job is lowered to this priority from PRIOR1.
; It will stay at PRIOR2 until it is detected in some I/O wait
; state (raised to PRIOR1) or stays "compute bound" for L2TCKS
                                                                                                                                          STATE
                                                                                                                                                    <>0
                                                                                                                                                               If job can run and wants to
                                                                                                                                                               ; Copy Job number
; Make a pointer into the Job table
; Pointer to JDB
                                                                                                                     GETSTT: MOV
                cycles (lowered to PRIOR3).
                                                                                                                                          RO. R1
                                                                                                                                ADD
                                                                                                                                          JOBTBL, R1
                                           ;Is job doing anything now?
;No- Jump it to level 1
;Yes- Increment the nr of busy cycles counter
 LEVEL2: TST
                      STATE
                                                                                                                               PEEK
                                                                                                                                          R1, R1
                      RAISE
CYCLES(RO)
                                                                                                                                TST
            BNE
                                                                                                                                                                  Is JDB pointing somewhere?
            INC
                                                                                                                               BNE
                                                                                                                                          5$
                                                                                                                                                                 Something's there
                      CYCLES(RO), #L2TCKS; Has it used its quota of Level 2 cycles?
EOLOOP; No- Go on
#PRIOR3, PRIOR; Yes- Drop it to Level 3
            CMP
                                                                                                                               BR
                                                                                                                                          63$
                                                                                                                                                                 No- No job with this number
            BLE
                                                                                                                               CMP
                                                                                                                                          #SETBTS. R1
            MOV
                                                                                                                    5$:
                                                                                                                                                               ; All bits set? (End of job table)
                                                                                                                                                               ; No- Keep going
; Yes- Clear R1, this not a running job
; Cleared R0 indicates end of job table
            MOV
                      #BURST3, BURST
                                                                                                                               BNE
                                                                                                                                          10$
                                           :Go schedule
                                                                                                                               CLR
                                                                                                                                          RO
   PRIOR3 - A job is lowered to this priority from PRIOR2.

It will remain at PRIOR3 until it is detected in some I/O
                                                                                                                               BR
               wait state when it will be raised to PRIOR1.
                                                                                                                                                              ; Copy pointer
; Add offset
; Get job's priority
; Mask off High Order byte (Burst)
                                                                                                                     10$:
                                                                                                                               MOV
                                                                                                                                          R1.R2
                                                                                                                                          #28.,R2
                                                                                                                                          R2. PRIOR
LEVEL3: TST
                      STATE
                                           ;Is job doing anything now? ;Yes, leave it here
                                                                                                                               PEEK
                                                                                                                               CL RB
CMP
                                                                                                                                          PRIOR+1
PRIOR, #200
                                           :No-
                                                                                                                                                                 Is Sign bit set?
           Raise job to PRIOR1
MOV #PRIOR1,PRIOR
                                                                                                                                                                 No, go on
Yes, Make the number negative
                                                                                                                               BI.T
                                                                                                                                          15$
                                                                                                                                          #400, PRIOR
RAISE:
                                                                                                                                                                 Add offset
            MOV
                      #BURST1.BURST
                                                                                                                    15$:
                                                                                                                               ADD
                                                                                                                                          #8.,R1
                                                                                                                               PEEK
                                                                                                                                                                 Get pointer to JDB2
           Change the Priority, Run burst
                                                                                                                               ADD
                                                                                                                                         #2.R1
                                                                                                                                                                 Add offset
SCHED:
                                                                                                                               PEEK
                                                                                                                                         R1, CPUTME
R0, R1
                                                                                                                                                                 Get Least signifigant bytes of CPU time
                                                                                                                                                                 Regain job number
Pointer to JBWAIT table
Determine job's state
                                                                                                                              MOV
           Record the current CPU time
                                                                                                                               ADD
                                                                                                                                         JEWATT, R1
                      CPUTME, TIMUSD(RO)
                                                                                                                               PEEK
                                                                                                                                         R1,STATE
RCDTME: MOV
                                                                                                                                                                Job is hibernating, Do something about it
Clear Hiberation cycles count if not HB
                                                                                                                              BEQ
                                                                                                                                         60$
           Continue with next job
                                                                                                                              CLR
                                                                                                                                         HIBCNT(RO)
EOLOOP: INC
                                                                                                                                                                Regain job number
Pointer to JBSTAT table
Store status word temporarily
                     RO
                                                                                                                              MOV
                                                                                                                                         RO,R1
           INC
BR
                                                                                                                              ADD
                                                                                                                                         JBSTAT. R1
                                                                                                                               PEEK
                      LOOP
                                           ;Next job
                                                                                                                                         STATE, R1
                                                                                                                              BIT
                                                                                                                                                                AND with state
                                                                                                                              BEO
                                                                                                                                         20$
                                                                                                                                                                = 0? Yes-
No- Clear
           CLRXRB Clear the XRB and return a pointer to XRLEN.
                                                                                                                              CLR
                                                                                                                                         R1
                                                                                                                                         25$
                                                                                                                                                              ; and skip setting it
; Yes- Set all bits
                                                                                                                              BR
           CALL: CALL
                               CL.RXRB
                                                                                                                    20$:
                                                                                                                              MOV
                                                                                                                                                              ; AND STATE with WAITBITS
           Input: None
                                                                                                                              BIT
                                                                                                                                         #WAITBT, STATE
                                                                                                                   25$:
                                                                                                                                                                = 0? Yes-
                                                                                                                              BEQ
                                                                                                                                         30$
                               -> XRB+XRLEN
           Output: R4
                                                                                                                              MOV
                                                                                                                                         #SETBTS.STATE
                                                                                                                                                                No- Set word
and skip clearing it
                                                                                                                                        STATE
                                                                                                                   30$:
                                                                                                                              CLR
                                                                                                                                                             : Yes- Clear
                                        ;Point to end of XRB
;Clear the previous word
;Is the entire XRB clear
CLRXRR: MOV
                      #XRB+XRBSTZ. R4
                     -(R4)
R4,#XRB
                                                                                                                              віт
                                                                                                                                        R1, STATE
2$:
                                                                                                                   35$:
                                                                                                                                                             ; Check for any matching set bits
                                                                                                                                        37$
#SETBTS,STATE
           CMP
                                                                                                                              BEO
           BHT
                     2$
                                           ;No- Keep clearing
                                                                                                                              MOV
                                                                                                                                                           ; Net effect of these lines:
           RETURN
                                                                                                                              BR
                                                                                                                                        39$
                                                                                                                              CLR
                                                                                                                                        STATE
                                                                                                                                                             ; (R1 AND STATE) ==> STATE
;+
                                                                                                                                                             ; Set R1, Good exit
; We've gotten the info, go back now
                                                                                                                   39$:
                                                                                                                              MOV
                                                                                                                                        #SETBTS.R1
          CLRFQB Clear the FIRQB and return a pointer to FQFUN.
          CALL: CALL
                               CLRFOB
                                                                                                                   60$:
                                                                                                                    . IF DF
                                                                                                                            KILLHB
          Input: None
                                                                                                                              CALL.
                                                                                                                                        HBHNDI.
                                                                                                                                                             ; Handle the Hibernator
                                                                                                                    . ENDC
                               -> FIRQB+FQFUN
           Output: R4
                                                                                                                              CLR
                                                                                                                                                             ; Don't mess with this job
                                                                                                                   63$:
                                                                                                                              RETURN
                                                                                                                                                             : Exit point
                     #FIRQB+FQBSIZ,R4 ;Point to end of FIRQB
CLRFQB:
         MOV
                                          ;Clear the previous word
2$:
          CLR
                     -(R4)
                     #FIRQB+FQFUN,R4; is the entire FIRQB clear?

you keep clearing

#FIRQB+FQFUN,R4; Point to FQFUN
                                                                                                                              HBHNDL Hibernator Handler
          CMP
          MOV
                                                                                                                              Call:
                                                                                                                                        MOV
                                                                                                                                                   #2#jobnr,R0
           RETURN
                                                                                                                                                   HBHNDL
                                                                                                                              Input: RO
                                                                                                                                                             Job nr times two
          SETPRI Set the priority and runburst for a job
                                                                                                                                        HIBCNT(RO)
                                                                                                                                                             Number of cycles in hibernation
                     MOV
                               <newpriority>.PRIOR
                                                                                                                             Output: Warning text to KBO: (After USERWT cycles)
Death to offending job (After USERWT+OPERWT cycles)
(Destroys R1-R4)
                                <newrunburst>,BURST
          Input: RO
                               2 times the job number to set
                                                                                                                   .IF DF KILLHB
                                                                                                                  HBHNDL: INC
                                                                                                                                       HIBCNT(RO)
                                                                                                                                                             ;Increment "cycles in HB" counter
                     PRIOR
                               new priority for job
```

HIBCNT(RO), #USERWT ; Have we passed do nothing stage?

CMP

ALL BACK ISSUES OF THE RSTS PROFESSIONAL



Includes a Holiday Special of 2 Library Cases

GREAT GIFT IDEA - ACT NOW - SUPPLY IS LIMITED

SIMPLY FILL IN COUPON BELOW AND SEND IT ALONG WITH YOUR CHECK TO: RSTS PROFESSIONAL, P.O. BOX 361, FORT WASHINGTON, PA 19034

□ Please rush my order for sets of ALĻ BACK ISSUES OF RSTS PROFESSIONAL @ \$100° per set. TOTAL ENCLOSED \$						
☐ I have all the Back Issues but would like the Library Case(s) @ \$7.50 each. Send me Library Case(s). [Library Case holds 12 magazines.] TOTAL ENCLOSED \$						
— DON'T FORGET YOUR FRIENDS AND BUSINESS ASSOCIATES — NAME						
ADDRESSCITY/STATE/ZIP						
PLEASE ENCLOSE YOUR CHECK WITH YOUR ORDER.						

```
:No- Do nothing
          BLT
                   30$
                   | 15$ ;Yes- Send message | HIBCNT(RO), #USERWT+OPERNT ; Have we reached Kill stage? | 25$ ;Yes- Kill the offender
          CMP
                                      :No- Do nothing
          BR
                   JOBNUM, R5, <offjb>
                                          ; Put the job number into the message
 15$:
          CALL
                   SNDTXT, R5, <wrnmsg> ; Warn KBO: of Hibernator
                                      ;Go back to normal work
          BR
                   CLRFQB
                                      :Clear FIROB (Returns: R4 => FIRQB+FQFUN)
 25$:
                                      ;"Kill Job" SF
                   #UU.CHU,(R4)+
          MOVB
                                      Copy Job nr
                                      ;Divide by 2
          ASR
                   R3,(R4)+
          MOVE
                                      ;Place in FIRQB
                   #377, @#FIRQB+35 ;Flag to kill job
          MOVB
                                      ;Kill Offender
          · IIIIO
                   JOBNUM, R5, <kiljb> ; Put the job number into the message SNDTXT, R5, <kilmsg> ; Inform KBO: of our action
          CALL
 30$:
          RETURN
 . ENDC
          JOBNUM Put the Job number in ASCII characters in the desired location
                   CALL
                            JOBNUM. R5. <1oc>
          Input:
                            Job number (times 2) to translate
                   <loc>
                            Where to put the two characters
          Output: <loc>
                          The two ASCII characters
                   (Destroys R1 through R3)
JOBNUM: MOV
                                      ;Location to place ASCII characters
                   (R5)+, R1
                                      Copy Job nr
          MOV
                   RO,R3
                                      :Clear rest of two word integer
          CI.R
                   R2
                                      ;Divide the job nr by 10.
                   #10..R2
          DIV
                   #'0,R3
#'0,R2
                                      ;Make an ASCII character (the remainder)
;Make an ASCII character (the quotient)
          ADD
          ADD
                   R2,(R1)+
R3,(R1)
          MOVB
                                      ; Put character (Job nr) in string
          RETURN
          SNDTXT Broadcast text on KBO
                   CALL
                            SNDTXT, R5, <text>
          Call:
          Input: text
                            ASCIZ text to print
          Output: None
                   (Destroys R2-R4)
SNDTXT: PUSH
          MOV
                   (R5) + .R0
                                      ;Get pointer to string to print
                                      ;Make a copy
;Init byte counter (including pre-increment)
                  #-1,R2
R2
          MOV
          INC
                                      :Count this byte
2$:
                                      ; Is this the end?
                   (RO)+
         BNE
                   2$
                 XRB and do the Broadcast
                   CLRXRB
                                     ;Clear XRB (Returns: R4 => XRB)
;.SPEC function code
         CALL
MOV
                   #6,(R4)+
         MOV
                   R2,(R4)+
                                      ;Send this many bytes
                                     ;Starting here
;See 3-218 of Sys. Directives Manual
                   (R4) +
          TS TB
          MOVB
                   #2,(R4)+
                                     : TTYHND
         CLR
                                      ;Keyboard nr
                            :Change to MOV #<kbnr>..(R4) if another
                            ;Keyboard is desired
                                     ;Do the broadcast
          .SPEC
                                     ;Did anything get left behind?
;No- Finish up
          TST
                   @#XRB+2
         BEQ
                   10$
                                     :Yes- This many bytes left over
         MOV
                   @#XRB+2.R2
                                      New starting location
          ADD
                   R2, R3
         BR
                                     :Send the rest
10$:
         POP
                   <R0>
                                     ;Restore RO
         RETURN R5
          .SBTTL Data Section
         This would be made a R/W (User area) PSECT if DYNPRI is to be made
         a runtime system instead of running under RSX.
JOBTBL: .BLKW
JBWATT: BLKW
JBSTAT: .BLKW
PRIOR: .BLKW
         . BL KW
BURST.
STATE: .BLKW
PPN:
         . BLKW
CPUTME: .BLKW
                  JOBMAX
TIMUSD: .BLKW
CYCLES: .BLKW
HIBCNT: .BLKW
                  JORMAY.
                  <15><12><7>\##DYNPRI## Job \
WRNMSG: .ASCII
         . BLKB
         . ASCII
                  \ has been in hibernation for \(USERWT*SLEEP/60.+'0>
         . ASCIZ
                  \ minutes.\<15><12>
         Note the expression above needs modification if more than 9
        minutes is used.
KILMSG:
        . ASCII
                  <15><12><7>\##DYNPRI## Job \
KILJB:
         . ASCIZ
                    killed.\<15><12>
                 <15><12>\DYNPRI detaching...\<15><12><12>
DETMSG: . ASCIZ
         . END
                 DYNPRI
```

SEARCH.TEC

By Robert G. Smith, Tyler, TX

Some time ago while working on a system of programs, I had to change the record layout of a much used file. Since this was an old system (meaning: one I did not write, and that I would have much rather thrown away and rewritten than maintained), to do so meant modifying every program in the system that accessed that file. How was I to find out which of the hundred or more programs did this? I was tempted to despair but instead sat down and wrote the following TECO program called SEARCH.TEC.

So if you are in a similar state, and sat down to read VAX/RSTS Pro hoping for something to cheer you, take heart! Here it is. Just put it on your system, and type RUN SEARCH.

It first wants to know where it should put its output. It will assume your terminal. It then prompts for which files it should search. Use any wildcard specification you would in PIP, but if you omit the extension or filename it will not assume a "*". If you do not put in anything it will search all the files in your account on the system disk. Finally it will want to know what you are searching for. Here you can use TECO's string building and match control characters which enable you to do neat things like finding all the programs that have THEN or ELSE followed by just a line number, or not followed by just a line number. See your TECO manual about them. It is nicely Ctrl-C trapped, so that a Ctrl-C at any time while it is running will bring you to a firm but gentle stop.

Using TECO's wildcard lookup capability it will read every one of the files matching the wildcard file spec. and search them for your string. When it is found, SEARCH will print out the program line on which it was found and the following line. It also keeps running totals on how many files were searched and how many times your string was found. If your string occurs twice on one line SEARCH will only print out that line once but also only count that as one occurrence of the string. The extra programming and run time expense was too much to try to fix that one. I do not know of any other features (read that: bugs), but it is now your program, not mine.

It does do its job fairly quickly. On our 11/70 with about 1M of memory and 15 jobs running, it took 10 minutes to search 93 files and find a string 5272 times. When I took all the comments and unnecessary spaces out, that time was cut in half. I have included both the readable version (with spaces and comments), and the "compiled" version, in case you do want to compress it and you don't trust yourself to figure out which spaces are significant, and also because it so graphically illustrates the difference between documented and undocumented code.

Some things to remember while installing it: Nothing in the program is a control character; it is to be typed in as written, EXCEPT the "1E"s and the "\$\$" at the end. The "1E"s must be actual Ctrl-Es and the "\$\$" must be two ESCs. Also the file needs to have the sixth bit (64 decimal) of its protection code set to indicate that is a runnable program.

PROGRAMS ON NEXT PAGE

RUN SEARCH

OUTPUT FILE <KB:> :

SPEC. OF FILES TO SEARCH (*. *> : *. BAS

SEARCH STRING : THEN ES ED

SEARCHING FOR "THEN "ES"ED"

***** SEARCHING : [5,0]KILLER.BAS<60>
0 OCCURENCES IN [5,0]KILLER.BAS<60>

**** SEARCHING : [5,0]CALLER.BAS<60>

O OCCURENCES IN [5,0]CALLER.BAS<60>

**** SEARCHING : [5,0]JUMP.BAS<60>

THEN	2010	&
	I IF CCL ENTRY, THEN SKIP ACCOUNT PROMPT	ě
THEN	2020	Ł.
	PROJ\$ = 1\$	ė.
THEN	31010	Ł
ELSE	ENTRY\$ = 0\$	&
THEN	31020	ě.
ELSE	1000	&
THEN	1000	&
ELSE	RET.LINE\$ = VAL(LEFT(COMMON\$, P\$-1\$))	&

5 OCCURENCES IN [5,0]JUMP.BAS<60>

**** SEARCHING : [5,0]SPELL.BAS<60>

O OCCURENCES IN [5,0]SPELL.BAS<60>

4 FILES SEARCHED

5 OCCURENCES FOUND

Ready



" WHAT DO YOU MEAN, I KILLED FROGGER ! ? "

```
I SEARCH. TEC I
   By Robert G. Smith !
        22-Jun-83
  CALLED WITH : !
     RUN SEARCH !
I SEARCHS A SET OF FILES DEFINED BY A WILD CARD SPECIFICATION !
   FOR A GIVEN CHARACTER STRING. OUTPUTS ALL REFERENCES!
[O EDUO [O [A [B [C [D [F 0,-32768ET -1ED ! SET ^C TRAP & ENABLE YANK !
I LOAD THE INPUT MACRO IN REG-B 1
e^uB$
[1 [0 ^RUO [0 ETUO [0 ]
D 8+32,0ET
                                          ! SAVE REGISTERS!
! DECIMAL NUMBERS, "T INPUTS AS WE WISH!
! REMEMBER THIS SPOT!
! GET A CHARACTER!
! IF A CF PRINT A CR AND END!
! IF A CR THROW AWAY THE LF AND END!
.U1
  Q0-10"E 13^T 0; '
Q0-13"E ^T!! 0; '
  Q0-3"E HK EF EX '
Q0-21"E Q1,.K 13"T F< '
Q0-18"E 13"T Q1,.T F< '
                                                ! CTRL-C KICKS US OUT !
! CTRL-U ERASES THE LINE AND STARTS OVER !
                                                  CTRL-R RETYPES IT OUT !
                                                ! IF A DEL !
   Q0-127"E .-Q1"G
          -1 AUO
                                                        I WHAT ARE WE DELETING? 1
                                                          DELETE IT !
          -U QO-7"E 13^T Q1,.T F< '
QO-9"E 13^T Q1,.T F< '
QO-27"E 8^T 32^T 8^T F< '
QO-3"L 8^T8^T 32^T32^T 8^T8^T
| 8^T 32^T 8^T '
                                                          SPECIAL CHARS: BEL !
                                                          TAB !
ESC !
                                                        ! OTHER CTRL CHARS ! ! ALL ELSE !
                                                  ! PUT THE CHAR IN THE BUFFER !
   QUEIII
                                           ! LOAD THE STRING INTO QA & CLEAR THE BUFFER! ! RESTORE REGISTERS !
Q1,.XA Q1,.K
]0 QOET ]0 Q0 R ]0 ]1
! BEGIN SEARCH ROUTINE !
0,32ET < ^T^_; > 32,0ET
                                                        ! THROW AWAY TYPE AHEAD !
 13°T 10°T
e^Aloutput FILE <KB:> :1 MB
:QAME e^UAIKB:1 -1U0 '
                                                        ! INPUT THE OUTPUT FILE !
! IF NOTHING INPUT ASSUME KB: !
                                                        ! OPEN THE OUTPUT FILE FOR OUTPUT !
0^AISPEC. OF FILES TO SEARCH <*.*> :! MB ! INPUT SEARCH FILES SPEC.!
:OATE 0^UA!*.*! ! IF NOTHING ASSUME *.*!
                                                        I SET THE WILD CARD SPEC !
 eENI^EQAI
!ISS! 13^T 10^T
@^A!SEARCH STRING :! MB
                                                        ! GET THE SEARCH STRING IN REG-A ! ! NO EMPTY SEARCH STRINGS !
 :QA"E @OIISS! '
                                                     ! DONE WITH INPUT !
 13^T 10^T
13@I!! 10@I!!
elisearching for "! GA
eli"! 13el!! 10el!! 10el!! HPW
OUD OUF
                                                       OUTPUT THE SEARCH STRING !
                                                      CLEAR THE ACCUM. & FILE COUNTER !
 <! NEXT FILE LOOP !
                                                       STOP IF A ^C HAS BEEN HIT !
CLEAR THE BUFFER !
   ET;
                                                        GET THE NEXT MATCHING FILE SPEC. !
    : PEN ! ! :
   G# ^YXF ^SD
                                                        PUT IT IN OF !
   %F
                                                      ! COUNT HOW MANY FILES WE DO !
    13@1!! 10@1!!
@1!**** SEARCHING : ! GF
                                                      ! OUTPUT THE FILE NAME BEING SEARCHED !
    1301!! 1001!! HPW HK
                                                      ! IF OUTPUT IS NOT TO THE KB: !
   Q0"E @^A! **** SEARCHING : ! :GF
                                                      ! TELL THE USER WHICH FILE WE ARE ON !
          13°T 10°T
                                                      ! OPEN UP THAT FILE TO READ !
    @ER! EQF/B+! Y
    <1 SEARCH FILE LOOP !
                                                        CHECK FOR A ^C !
SEARCH FOR OUR STRING !
      ET;
@_!^EQA!;
                                                        END LOOP IF NOT FOUND!
NEAR THE BOTTOM? GET THE NEXT PAGE!
BUMP COUNTER OF TIMES FOUND!
      2 Q+ . - Z"E A '
      $C
@I!--
                                                      1 DELINEATE OUR OUTPUT !
      13@III 10@III .-20,.PW -20D
0^Q+.,2^Q+.PW
                                                      ! OUTPUT NEXT AND CURRENT LINE !
                                                      ! ONLY FIND ONE ON ANY ONE LINE !
    >! END SEARCH FILE LOOP !
    13@III 10@III @II ! QC\ @II OCCURENCEI ! TELL HOW MANY WERE FOUND !
QC-1"N @IISI ' @II IN ! GF ! IN THIS FILE !
13@III 10@III
                                                        ! PUT IT OUT & ACCUMULATE TOTALS !
    HPW QC%D
  >! END NEXT FILE LOOP !
 HK 13@I!! 10@I!!
                                                         I REPORT STATISTICS !
 QF\ eI! FILE! QF-1"N eI!S! '
eI! SEARCHED!
                                                         ! HOW MANY FILES WERE SEARCHED !
 13@11: 10@11:
QD\ @1: OCCURENCE: QD-1"N @11S! '
@1: FOUND: 13@11: 10@1!!
                                                        ! HOW MANY WERE FOUND !
  -32768,0ET
                                                  I DISABLE THE "C TRAP !
                                                  ! RESTORE OUR REGISTERS AND FLAGS !
 ]F ]D ]C ]B ]A ]O QOED ]O HK EF EX
                                                     CLOSE ALL OUR FILES AND EXIT !
```

VMS DIRECTORIES: SOME NAVIGATIONAL AIDS

By Al Cini

Typing \$SET DEFAULT commands over and over again can be pretty tedious, especially when your system's directory names are long and meaningful, like [BASIC.SOURCE.SUBRTNS] or [JOES.PRIVATE.FINANCE.DATA].

The following command procedures, PUSHDEF.COM and POPDEF.COM, allow you to move ("PUSH") from one place to another, and then "POP" back to where you were without re-typing device and directory specifications.

They use some DCL global symbol space (you can PUSH yourself around about a hundred times without POPing back before this becomes a problem), and can be embedded in command procedures which need to go someplace initially and come back where they were when finished.

The command procedures contain little hard-wired user instruction files, and will print them if you specify "HELP" as the P1 parameter.

You can set up DCL symbols to invoke them (of course) as follows:

\$PUSH: = = "@dev:[directory]PUSHDEF.COM"
\$POP: = = "@dev:[directory]POPDEF.COM"

************************** PUSHDEF. COM: OLD VERIFY = 'F\$VERIFY(0)' SET NOVERIFY \$!+ PUSHDEF.COM Save current user default device & directory for \$! \$! future reference (see POPDEF.COM) \$! \$! Author: Al Cini, Computer Methods Corporation \$!-IF P1 .EQS. "" THEN -\$ GOTO NO_ARGUMENT IF P1 .EQS. "H" P1 .EQS. "HE" .C P1 .EQS. "HELP" THEN -.OR. -GOTO PRINT_HELP IF P1 .EQS. "BH" .OR.
P1 .EQS. "BA" .OR.
P1 .EQS. "BACK" THEN GOTO SETUP_FUSHBACK .OR. -.OR. -OLDDEFAULT := 'F\$LOGICAL("SYS\$DISK")''F\$DIRECTORY()'
SET DEFAULT 'P1' IF .NOT. \$STATUS THEN -GOTO PUSH EXIT PSH_TMP:= 'PSH_INDEX'
IF PSH_TMP .EQS. "" THEN PSH_INDEX == 0 PSH_INDEX == PSH_INDEX+1 PSH\$_DEFAULT'PSH_INDEX' :== 'OLDDEFAULT'
IF F\$MODE() .NES. "BATCH" THEN -\$SHOW DEFAULT ! If we're at a terminal, echo ! our new default disk/directory. GOTO PUSH_EXIT NO_ARGUMENT: WRITE SYS\$OUTPUT "?No new default device/directory specified" GOTO PUSH_EXIT SETUP_PUSHBACK: IF PSH_INDEX .LT. 1 THEN -GOTO INVALID_PUSHBACK P1 := &PSH\$_DEFAULT'PSH_INDEX' GOTO ON_WITH_IT \$ INVALID PUSHBACK: WRITE SYS\$OUTPUT "?Nothing to PUSH BACK to" GOTO PUSH EXIT \$ PRINT HELP: TYPE SYS\$INPUT

BEGIN PUSHDEF. COM DOCUMENTATION:

The "PUSHDEF" procedure will change your default disk/directory for you, in a manner very similar to the SET DEFAULT command (not surprising, since it contains a "SET DEFAULT" command). However, PUSHDEF will "save" your old default disk/directory for you, and you can then "return" to your previous defaults by using the "POPDEF" (see) command procedure. PUSH and POP operations can be nested to any (really, about 200) depth, and the POP command "help" procedure documents several special POP options which afford additional flexibility in returning to previous defaults. The PUSHDEF procedure is invoked by:

\$@PUSHDEF [argument]
- or \$PUSH [argument]
The [argument] may assume the following values:

[H]ELP Prints this documentation.

Dev:[Directory<.subdirectory...>]
Specifies the desired new default disk and directory.

\$ PUSH_EXIT:
\$ IF OLD_VERIFY .EQ. 1 THEN SET VERIFY
\$ EXIT

POPDEF.COM:

```
OLD_VERIFY
                                   'F$VERIFY(0)'
        SET NOVERIFY
$ 1+
        POPDEF.COM
                         Restore previously stacked user default
                         device/directory (see PUSHDEF.COM)
                         Al Cini, Computer Methods Corporation
        Author:
$ 1-
        PSH_TMP := 'PSH_INDEX'
        IF PSH_TMP .EQS. "" THEN -
                 PSH_INDEX == 0
        INDEX_END
        TRACE_ON
                                  P1 .EQS. "TRACE" .OR. -
                                  P1 .EQS. "TR"
                                                     .OR. -
                                     .EQS. "T
        TNOUTRE ON
                                  P1
                                     .EOS. "INT"
                                                     .OR. -
                                     .EQS.
        HOME ON
                                  P1
                                     .EQS. "HOME"
                                                     .OR. -
                                     .EQS. "HO"
                                                     .OR. -
                                     .EQS.
                                           иНи
                                                   .OR. -
        HELP REQUESTED =
                                     .EOS. "HELP"
                                  P1 .EQS. "HE"
                          .OR. -
        IF TRACE ON
           INQUIRE ON
                         .OR. -
           HOME ON
                         .OR.
           HELP_REQUESTED THEN-
                GOTO ON WITH IT
        IF P1 .GT. O THEN -
                 HOME ON = 1
        IF P1 .GT. PSH_INDEX THEN
                GOTO BAD_RETURN_VALUE
        IF P1 .LT. 1 THEN -
        INDEX END = P1-1
 ON WITH IT:
        IF HELP_REQUESTED THEN -
GOTO PRINT_HELP
        IF TRACE_ON THEN
        GOTO TRACE_DEFAULT_STACK
IF PSH_INDEX .EQ. 0 THEN -
                 GOTO NO_PRIOR_PUSH
       IF PSH_INDEX .EQ. 0 THEN GOTO STACK_EMPTY
        PREVDEFAULT := "PSH$ DEFAULT" 'PSH INDEX'
        NEWDEFAULT := &'PREVDEFAULT'
        IF .NOT. INQUIRE_ON THEN -
                GOTO SET DEFAULT
        WRITE SYS$OUTPUT "Next stop: ",'PREVDEFAULT'
        INQUIRE/NOPUNCT SET_RESPONSE " --> Go for it (Y or N <N>) ? "
        IF .NOT. SET RESPONSE THEN -
```

Hit The Jackpot of VAX* Resource Accounting & Chargeback Systems



'VMSACTSYS' The VMS* Accounting System From CIS ... The Application Architects

DEXPO® West 83 Booth # 1029
*VAX and VMS are registered trademarks of Digital Equipment Corporation



165 Bay State Drive / Braintree, MA 02184 617-848-7515 Outside of Mass. Call Toll Free: 800-232-5215

```
$ SET_DEFAULT:
                                                                                                                     $ TRACE_END:
           PSH_INDEX == PSH_INDEX-1
                                                                                                                                WRITE SYS$OUTPUT ""
           SET DEFAULT 'NEWDEFAULT'
IF F$MODE() .NES. "BATCH" THEN -
$SHOW DEFAULT
                                                                                                                                GOTO POP_EXIT
                                                                                                                     $ PRINT_HELP:
                                                          ! Report our new default
                                                                                                                     $ TYPE SYS$INPUT
                                                          ! device and directory if
                                                           ! at a terminal.
                                                                                                                                 BEGIN POPDEF.COM DOCUMENTATION:
           DELETE/SYMBOL/GLOBAL 'PREVDEFAULT'
  LOOP TEST:
                                                                                                                                 The "POPDEF" procedure will return you to the default disk and directory which you last PUSHed (see PUSHDEF.COM's help text) from. The POPDEF procedure is invoked by:
           IF( HOME_ON .OR. INQUIRE_ON) -
   .AND. PSH_INDEX .GT. INDEX_END THEN -
                                   GOTO RETURN
           GOTO POP EXIT
                                                                                                                                             $@POPDEF [option]
  BAD_RETURN_VALUE:
           WRITE SYS$OUTPUT "?Haven't PUSHed that far yet"
                                                                                                                                                        - or -
           GOTO POP_EXIT
  STACK EMPTY:
                                                                                                                                             $POP [option]
           SHOW DEFAULT
WRITE_SYS$OUTPUT "?Already at the top - ignored"
                                                                                                                                 The [option] argument may assume the following values:
           GOTO POP_EXIT
  NO PRIOR PUSH:
                                                                                                                                             "[T]RACE"
                                                                                                                                                                    This option will list all of
           WRITE SYS$OUTPUT "?POP without prior PUSH - ignored"
                                                                                                                                                                    the previously PUSHed defaults
           GOTO POP_EXIT
 GOTO POP_EXIT

TRACE_DEFAULT_STACK:

INDEX = PSH_INDEX

DEFDISK := 'F$LOGICAL("SYS$DISK")'

DEFDISK := 'F$DIRECTORY()'

WRITE SYS$GUTPUT ""

WRITE SYS$GUTPUT "Default history stack depth: ",INDEX
WRITE SYS$GUTPUT "Current default disk : ",DEFDISK
WRITE SYS$GUTPUT "Current default directory : ",DEFDIR

IF INDEX .EQ. 0 THEN -

GOTO TRACE_END

WRITE SYS$GUTPUT ""
                                                                                                                                                                    without returning you to any
                                                                                                                                             "[HE]LP"
                                                                                                                                                                    This option requests this help
                                                                                                                                             "[H]OME
                                                                                                                                                                    This option will return you to
                                                                                                                                                                    the first default you PUSHed
                                                                                                                                                                    from.
                                                                                                                                                                   This option will request confir -mation for returning backward
                                                                                                                                            [I]NTERACTIVE
          WRITE SYS$OUTPUT ""
WRITE SYS$OUTPUT "Index "," Default Disk/Directory"
WRITE SYS$OUTPUT "-----"," ------"
WRITE SYS$OUTPUT ""
                                                                                                                                                                                                     backward
                                                                                                                                                                    through previously PUSHed def-
  TRACE LIST:
                                                                                                                                If no argument is given, POPDEF will "POP" back to the previously "PUSHed" defaults.
           DEFAULT := "PSH$_DEFAULT"'INDEX'
           OUTLINE[2,7]:='INDEX'
WRITE SYS$OUTPUT OUTLINE,'DEFAULT'
                                                                                                                                END OF POPDEF. COM DOCUMENTATION.
           INDEX=INDEX-1
           IF INDEX .EQ. O THEN -
GOTO TRACE_END
                                                                                                                     $ POP_EXIT:
                                                                                                                                IF OLD_VERIFY .EQ. 1 THEN SET VERIFY
           GOTO TRACE_LIST
```

ATPK Modified

. . . continued from page 55

```
OPEN OUT.FIL$ FOR OUTPUT AS FILE 2$
\ PRINT #2$ IF CCPOS(2$)
\ PRINT #2$, "FORCE KBO: "; IF FORMAT$
\ PRINT #2$, "BUN $TYSET"
I INITIALIZE OUTPUT FILE BASED ON FORMAT REQUESTED
                                                                                                                                                                                  KB.ECHO$ = S$(10$)

IF KB.ECHO$ = 128$ THEN

I$ = FNPRINT$("FULL DUPLEX") IF ASR$ = 1$

ELSE I$ = FNPRINT$("LOCAL ECHO")

1 128 = FULL DUPLEX, 255 = LOCAL ECHO
  1010
                                                                                                                                                                                 1030
                IF KBOS THEN
                                                                                                                                                                   1190
                BEGINS = 0$
ELSE BEGINS = 1$
I START AT KEO: IF ASKED, OTHERWISE START AT KE1: INSTEAD
 1050
                FOR KB$ = BEGINS TO MAX. KB$
               FOR KB$ = BEGIN$ TO MAX.KB$

TIY$ = FNTTY$(KB$)

IF (PK$ = 0$) AND (TTY$ = 8$) THEN

GOTO 1990

ELSE PRINT #2$ IF CCPOS(2$)

PRINT #2$, "FORCE KBO: "; IF FORMAT$

PRINT #2$, "KB*+NUM1$(KB$)+":/RING";

I GET TERMINAL CHARACTERISTICS FOR THIS KB NO.

! IF PSEUDO KEYBOARD AND NOT WANTED SKIP THIS UNIT,

! OTHERWISE CUTPUT THE FIRST PART OF A TIYSET COMMAND LINE
                                                                                                                                                                                 1210
                                                                                                                                                                                              1230
                            1070
                                                                                                                                                                                  ! OUTPUT FILL FACTOR IF ANY
                                                                                                                                                                                              KB.OUT.SPEED$ = $$(14$)

KB.IN.SPEED$ = $$(16$)

KB.IN.SPEED$ = $KB.OUT.SPEED$ UNLESS KB.IN.SPEED$

GOTO 1270 UNLESS KB.IN.SPEED$

IF KB.IN.SPEED$ = 255$ THEN

I$ = FMPRINT$("SPEED 2741")

ELSE F$ = "SPEED" + FNSPEED$(KB.IN.SPEED$,TTY$)

IF KB.OUT.SPEED$ <> KB.IN.SPEED$,TTY$)

IF KB.OUT.SPEED$ << KB.IN.SPEED$
               ELSE I$ = FNPRINT$("WIDTH "+NUM1$(KB.WIDTH$))

1 OUTPUT TERMINAL WIDTH IF NOT ASR33 DEFAULT OR WIDTH NOT 72
                                                                                                                                                                   1250
                             KB. TAB$ = S$(6$)
 1090
               THE REGULT SPEEDS OF RESIDENCE

I$ = FORINT$(F$)

TWO BYTES RETURN INTERNAL SPEED VALUE THAT MUST BE USED WITH
INTERFACE CODE VALUE TO DETERMINE ACTUAL IN/OUT LINE SPEED
1110
                                                                                                                                                                               KB.PARITY% = S%(15%)

IF KB.PARITY% = 1% THEN

I$ = FMPRINT$("NO PARITY") IF ASR$ = 1%

ELSE IF KB.PARITY% = 2% THEN

I$ = FMPRINT$("EVEN PARITY")

ELSE IF KB.PARITY% = 3% THEN

I$ = FMPRINT$("ODD PARITY")

I$ = NO PARITY, 2 = EVEN PARITY, 3 = ODD PARITY")
                                                                                                                                                                  1270
              KB.LC.OUTPUT$ = S$(8$)

IF KB.LC.OUTPUT$ = 128$ THEN

I$ = FNPRINT$("LC OUTPUT") IF ASR$ = 1$

ELSE I$ = FNPRINT$("NO LC OUTPUT")

1 128 = LC OUTPUT, 255 = NO LC OUTPUT
1130
              1150
                                                                                                                                                                                 1290
```

RSTS/E INTERNALS MANUAL

Second Edition — Including Version 8.0

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 8.0 and earlier versions of the monitor. Future updates will include changes for new versions of the monitor.

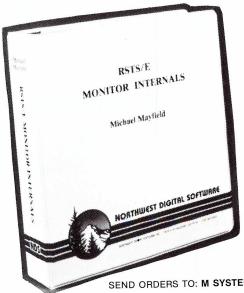
Information is available for all levels of users:

- 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 five quick reference foldout charts:

- 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.

\$9500

SEND ORDERS TO: M SYSTEMS, INC., BOX 361, FORT WASHINGTON, PA 19034-0361

```
1310
                        KB. UP. ARROWS = S$(18$)
                       ! 128 = UPARROW, 255 = NO UPARROW
                       KB.ESC.SEQ# = S$(22#)
 1330
                       I 128 = NO ESC SEQ, 255 = ESC SEQ
                        KB.DELIMITERS = S$(23$)
 1350
                       IF KB.DELIMITERS = 1285 THEN

I$ = NPRINT$("DELIMITER") IF ASR$ = 1$

ELSE F$ = KB.DELIMITER$ - 1285

I$ = NPRINT$("DELIMITER"+NUMI$(F$))
             ! 128 = DISABLE PRIVATE DELIMITER, OVER 128 = DELIMITER'S ASCII CODE
 1370
                        KB.ESC$ = S$(24$)
                       1 128 = NO ESC, 255 = ESC
            KB.CTRL.R$ = S$(25$)

IF KB.CTRL.R$ = 128$ THEN

I$ = FNPRINT$("NO CTRL/R")

ELSE I$ = FNPRINT$("TRL/R") IF ASR$ = 1$
! 128 = DISABLE CTRL/R AND CTRL/T, 255 = ALLOW CTRL/R AND CTRL/T
 1390
1410
                       KB. RESUME$ = S$(26$)
           KB.MESUMES = S$\(20\)$\)

IF KB.RESUME$ = 255$ AND KB.XON$ = 255$

AND KB.SCOPE$ = 255$ AND KB.XTALL$ = 255$ THEN &

I$ = FNFRINT$("RESUME CTRL/C")

MUST USE RESUME CTRL/C IF VIIOO WITH XON PROTOCOL.
1430
                       KB. BREAK$ = S$(27$)
           IF KB.BREAK = 1285 THEN

I$ = FNFRINT$("NO BREAK")

ELSE I$ = FNFRINT$("BREAK") IF ASR$ = 1$^r
! 128 = TREAT BREAK KEY AS NULL, 255 = TREAT BREAK AS CTRL/C
           KB.GAG$ = $$(28$)

IF KB.GAG$ = 128$ THEN

I$ = FMPRINT$("NO GAG") IF ASR$ = 1$

ELSE I$ = FMPRINT$("GAG")

! 128 = NO GAG, 255 = GAG
           NEXT KB$
\ PRINT #2$ IF CCPOS(2$)
\ PRINT #2$, "FORCE KBO: "; IF FORMAT$
\ PRINT #2$, "EXIT"
\ PRINT #2$, "END" IF FORMAT$
\ CLOSE 2$
\ COTTO 20757
1990
            1 FINISH OFF TTYSET COMMAND FILE AND EXIT PROGRAM
15000
                                       FUNCTIONS
           DEF* FNERR$(I$) = CVT$$(RIGHT(SYS(CHR$(6$)+CHR$(9$)+CHR$(I$)),3$),4$)
15010
           1 SYSTEM ERROR MESSAGES (0 = SYSTEM BANNER)
           DEF* FNTTY$(I$)
            EFF FNTTYS(15)

$$(0$) = 30$

$$(2$) = 6$

$$(2$) = 16$

$$(2$) = 0$

$$(1$) = 0$

$$(1$) = 1$

$$(1$) = 0$ FOR F$ = 5$ TO 30$

CHANGE S$ TO $$

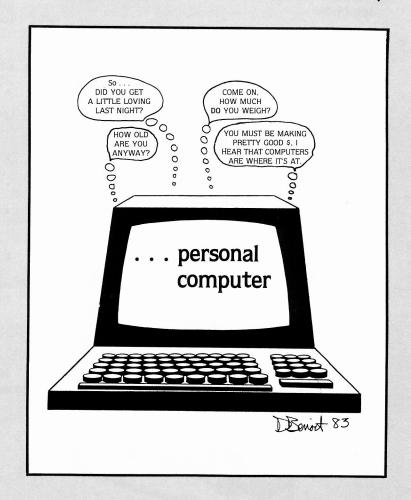
CHANGE S$ TO $$
             FNTTY% = S%(19%)
             GET REQUESTED KB TERMINAL CHARACTERISTICS, RETURN INTERFACE TYPE

0 = DL11A, DL11B 8 = PSEUDO KEYBOARD

2 = RESERVED 10 = DJ11
                       0 = DL11A, DL11B
2 = RESERVED
          DEF* FNPRINT$(I$)
\ I$ = CCPOS(2%) + LEN(I$) + 1%
\ GOTO 15210 IF OUTPUT$ > I$
           \ PRINT #2%
\ PRINT #2%, "FORCE KBO: "; IF FORMAT%
          PRINT #2%, ";"; IF CCPOS(2%) > COLONS
15210
           ! PRINT TEXT AFTER DOING A <CR><LF> IF CURRENT LINE MAX IS EXCEEDED
          DEF* FNSPEED$(SPD.CODE%, TTY.CODE%)
            RESTORE
           ! DETERMINE LINE SPEED FROM SPEED CODE AND INTERFACE CODE
           READ INTERFACES
15310
            \ IF INTERFACE$ <> TTY.CODE$ THEN
                      READ TTY$ FOR F%=1% TO 17%
GOTO 15310
             FIND THE LINE IN THE TABLE FOR THIS INTERFACE TYPE.
  15320
```

Anyone wanting an 800 BPI tape of the above source code can send a \$100 check or money order payable to Mac-Millan Bloedel Research, to cover materials and return postage.

MacMillan Bloedel Research has been using a PDP 11/34 RSTS system since 1977 for scientific computing. Many of our scientists, engineers, and technicians use BASIS-PLUS on a daily basis. Our major interactive statistics program, MINITAB, and our graphics program, EASY-GRAPH, are written in FORTRAN IV. A super editor (supported by Brian Nelson of the University of Toledo) called EDRT is used instead of TECO or DT. EDRT is a run-time system that can handle 30000 + lines without overloading even a PDP 11/34, supports scope mode, and has built-in HELP. We would like to hear from other RSTS or VAX sites concentrating on statistics and graphics in an industrial research environment.



PR(TALK continued from page 6		\ Print \$1%, t0\$ \$ \ Next t0% \$! Display the INUSE sign. \$
	! Set Some 'Variable' Parameters	& & &	\ F = Time(0%) + T% \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
	! ====================================	1 2 2	Print #1% \$ 2 \\ Print #1%, Cvt\$\$(Right(Sys(Chr\$(6%) + Chr\$(9%) + Chr\$(9%) + Chr\$(0%)), \$ 3%), 4%); ' Job'; J%; ' ['; Num1\$(PrJ%); ', '; & Num1\$(Prg%); '] Kb'; Num1\$(K%); ' '; Date\$(0%); \$
	indicates no maximum wait imposed. Os = '' Overide pass key. This can be used to overide the pass key siven by the user running the program. A null indicates no	1 1 1	/ f Time (0%) Print #1% Do information line.
	! overide Password. \ DX = - 1X ! Flad. A zero flad will suppress the information line (about the current user) to be suppressed; A non-zero value will	1 1400 2 1400	! !
	! allow the line to be displayed. \[L\lambda = 0\lambda \] ! Flas. A non-zero value will los the user out if the maximum ! time is imposed; A zero value will cause the user to remain !	2 2 2	! Wait for User to enter Pass Key & !
1070	! lodded in when the maximum time is up.	8	\ \(\mathbb{W}\)\(\text{F - Time(0\chi)} \\ \\ \text{! Time left to wait.} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\
	Do some work with Parameters	& & & &	! Is time up ? \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$
	! t0\$ = Sys(Chr\$(6%) + Chr\$(- 21%)	2 2 2	\ Wait W% \$ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
	! Only do this if user is privelaged. then TX = OX ! Do not impose of time limit.	2 2 2	! Get response. \$ \ R\$ = Cvt\$\$(R\$, 1% + 4% + 32%) \$ \ Goto 1600% if R\$ = F\$ or (R\$ = 0\$ and 0\$ <> '') \$
•	\ 0\$ = '' ! No overide Password. Goto 1200% ! Skip PPn and Kb check.	& & & &	! Check for valid pass keys. \$ \ Goto 1400% \$! Try adain. \$!
1100	!!	% 1500 % 8	! !
	Do A PPn and Keyboard Check	& & & &	Time is up !
,	Restore \ Read t0% ! No. of PPn's to exclude. \ For t1% = 1% to t0% ! Check each one.	2 2 2 2	Close 1% \$ Frint Print
,	Read P1%, P2% ! Read PFn to exclude. Goto 1110% if PrJ% = P1% and (Prd% = P2% or P2% = - 1%) ! Check PPn.	& & & & &	Messase
`	N Next t1% ! Complete loop. N Read t0% ! No. of Kb:'s to exclude.	8 8 8 1600	The state of the s
``	X For t1% = 1% to t0% ! Check each Kb:. Read K1% ! Read Kb no. to exclude. \$ 60to 1120% if K% = K1%	& & & &	! Terminal Is Free & !
	Check user's kb: Next t1% Complete loop.	& & &	
	! Proceed. !	<u>x</u> <u>x</u> <u>x</u>	! Message
1110	rint '?Invalid account number ['; Num1\$(PrJ%); ','; Num1\$(Prd%); Goto 32760% ! Error messade and exit.	']' & \$ 9000 &	!
1120 F	'rint '?Invalid keyboard - Kb'; Num1\$(K%); ';' Goto 32760% ! Error message and exit.	& & &	Data Statements !
1200		& & &	The following statement contains the PPn's which the program \$! is prohibited to run in. \$. \$. \$. \$. \$. \$. \$. \$. \$. \$
į	Get Pass Key and Confirmation	& & & &	Project.
\	0\$ = Sys(Chr\$(3%)) Print 'Pass key '; Input Line P\$	& & & & 9010	Pata 0 \$
\	Print if Pos(0%) Print 8 P* = Cvt\$\$(P\$, 4% + 32%) t0\$ = Sus(Chr\$(2%))	& & &	! The following statement contains the keyboard numbers which \$! the program is prohibited to run at.
\	! Get pass key from user. Input 'Confirm'; t0\$ Print if Pos(0%) Goto 32760% if Left(Cvt\$\$(t0\$, -1%), 1%) <> 'Y' 8	<u>2</u> 2 2	
1300 !	! Must be have 'Y' to proceed. !	9020	!
	Besin the 'Inuse'	2 2 2	! The following statement contains the 'inuse' sign which will be & ! displayed at the users keyboard.
`	Pen 'Kb: Inuse.Vdu' as file 1%, Mode 8% t0\$ = Sws(Chr\$(3%)) ! Open Kb: in binary mode and disable terminal echo. Print \$1% for t0% = 1% to 5%		!
	Read t0\$ until t0\$ = '*Start*' For t0% = 1% to 7% Read t0\$ Print \$1%, t0\$		' II NN NN NN UU UU SSSSS EEEEEEE', % ' II NN NNNN UU UU SS SS EE', % ' III NN NNNN UU UU SS SS EE', % ' IIIIIIIIII NN NN UUUUUU SSSSS EEEEEEEEE', %





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

Dear VAX/RSTS MAN:

I thought you might be interested in the following question which was proposed by our VAX System Manager:

If the DCL command

\$ Print/Notify TEST.MAR

broadcasts a message to your terminal when TEST.MAR finishes printing on the line printer, what does the command

\$ Print/Tify TEST.MAR

do?

Brett A. Hunsaker Terre Haute, IN

Dear VAX/RSTS Man:

We have run up against a problem, and we would like some advice from you or your readers.

We are presently running a RSTS/E system on a PDP11/23. Using 256KB and 3 RL02 giving 30MB disk space and 11 serial lines. Of these, five are to VT100 terminals on which we do a fair bit of interactive editing using EDT2, one to a LA 120 terminal for printouts, and the others to various other terminals/system which are used (infrequently) for file transfer. When running a compilation the interactive edit jobs slow down to a crawl. Because of current short term needs we need to both boost the number of users and improve performance.

DEC doesn't officially support RSTS for this size machine, and tells us to buy a bigger system (they are right but because of budgetary constraints this will take over a year). The distributor who sold us the system suggests in the meantime that we go to 22-bit addressing, add extra memory and put in DZV11s.

So we are going to 22-bit addressing with Megabyte of memory. We also wanted to put in some sort of multiplexers like the DH on UNIBUS. From my gleenings of DEC literature it doesn't appear that the DZV11 from DEC should give any improvement in performance? However, it appears that using a DH type of multiplexer using DMA access on output can improve the performance.

The catch is that most of the ones being offered for the Q-bus say that they will work under RSTS/E; some even say that they will support 22-bit addressing, however, apparently not both at once. That is not RSTS/E on 22-bit addressing on the Q-bus because of the DMA and the Memory Management Hardware on a 11/23 board (as opposed to the 11/23+ board). Can you give me any pointers, and comments?

Is there any information on writing drivers for RSTS? Not that I relish the idea of taking time off from our software development to write one. Are there any available from independant suppliers? Or is there a bit of hardware which answers this need? (Emulex claims their multiplexer does the job.)

Looking forward to your comments.

Ludwik Kowalski Software Development Manager ELEX Control Systems Ltd., ISRAEL

Dear Ludwik:

- 1. Talk to the DH compatible suppliers about your specific configuration. We believe you should be able to find a DH that works.
- 2. The MB of memory will help.
- 3. Try a version of DYNPRI (in DECUS Library) to improve interactive response time and slow down the batch compilation. (Also see DYNPRI.MAC by Jeff Corbett in this issue.)
- 4. Don't write a device driver.

NEW PRODUCTS

TIME AND MONEY SAVING 'Z-LINE' POWER CONTROLLERS UNVEILED AT DEXPO

Pulizzi Engineering, Inc. will introduce their "Z-Line" and "Z-Line" DEC compatible Power Controllers at the DEXPO WEST 83.

The "Z-Line" Power Controllers deliver filtered protection from both EMI and RFI on line to line and line to ground as well as spike and surge protection. All functions are bidirectional.

Designed to save the user considerable time and money the "Z-Line" Power Controllers insure against the loss of valuable data, eliminates errors caused by power line noise, protects against voltage variations, secures against erroneous data and protects your expensive equipment from damage. For further information see us at booth number 1519, DEXPO WEST, or contact Pulizzi Engineering, Inc., 3260 S. Susan Street, Santa Ana, CA 92704-6865 (714) 540-4229.

GRC DEMONSTRATES NEW COMPUTER SYSTEM CONFIGURATION

General Robotics Corporation will be demonstrating its new high performance LSI-11 based computer system configuration using a 51/4 inch Winchester disk and 8 inch slimline floppy disk technology at DEXPO WEST 83. GRC has begun shipping this product and other Q-Bus products in large quantities in the United States, Europe, and to the People's Republic of China.

GRC's new MSV11-Q one megabyte of RAM with parity on a single quad height LSI-11 Q-Bus module and the Q-Bus compatible controller for the Archive or Cipher ¼ inch streaming tape drive will also be demonstrated at DEXPO WEST 83.

The system supports all DEC compatible operating systems

including RT11, RSX11, RSTS/E, TSX, TSX-Plus, UNIX, SHARE-11, and with the addition of GRC's optional Z80V11 hardware, CP/M.

Pricing for a full configuration with an LSI-11/23, 1 megabyte of main memory, 16 serial I/O ports, two 1 megabyte floppy disks, and a 20 megabyte Winchester disk is under \$10,000 in large OEM quantities. Modules are also available separately and in Semi-System configurations. Pricing is F.O.B. factory. Deliveries are 30 days or less. For further information, call or write General Robotics Corporation, 57 North Main Street, Hartford, WI 53027. Phone 800-558-7832, or 414-673-6800.

SC31 CONTROLLER: UNIVERSAL EMULATOR INTRODUCED AT DEXPO

Emulex Corporation, a leading independent manufacturer of components for Digital Equipment Corporation (DEC) computers, announced today that it will introduce a major new product at the National Computer Conference in Anaheim, California: The SC31, a universal emulator for interfacing high-speed, high-density disk drives with DEC computers utilizing the Unibus.

"The SC31," reports Robert E. Pflomm, Emulex's Product Manager, Storage, "has been designed specifically to interface with the DEC Unibus in the PDP-11 and VAX-11/730 environment. The SC31 is ideal for applications in which the user requires high-speed data transfers offered by new, high-capacity Winchester drives, such as the Fujitsu Eagle and several new 500 Mbyte drives expected to be introduced soon by other major manufacturers," Mr. Pflomm says.

The new product is a fully-embedded, stand-alone, single hex board controller that requires only one SPC backplane slot, representing savings in both backplane space and power requirements — only eight amps at ± 5V and 0.7 amps at -15V.

Calculated MTBF is 30,000 hours,

but measured results in Emulex tests indicate a much higher actual MTBF.

The SC31 lists for \$4,900 single quantity. The SC31 may be combined in Emulex's mix-and-match OEM and large end-user volume purchase plan to qualify for attractive pricing discount. Units will be available for June delivery; standard delivery is 30 days following receipt of order.

For more information about the SC31, please call or write Mr. Phillip Begich, Director of National Sales, Emulex Corporation, 3545 Harbor Boulevard, P.O. Box 6725, Costa Mesa, California 92626. Telephones: (800) 854-7112; in California, (714) 662-5600 or see us at DEXPO WEST, Booth 512.

NEW SOFTWARE FOR DEC PROFESSIONAL SERIES

Applied Information Systems, Inc. will announce a number of new software packages for the DEC Professional 300, PDP-11, and VAX at DEXPO WEST 83 in Las Vegas.

AIS-PL/I, a Subset G PL/I compiler, will be available for the Professional 300 series at the show. AIS-PL/I will offer these personal computer users the powerful features of the PL/I language.

BURCOM-11, a DEC/Burroughs communication system, will also be announced for the Professional 300 at the show. This package will allow DEC Professional 300 and PDP-11 users to communicate with Burroughs by emulating the Burroughs MT983 or MT993 terminals.

Other products to be exhibited include a new version of AIS-PL/I for the PDP-11 that offers direct support for RMS-11K and the Commercial Instruction Set and EasyEntry, a full screen forms design and data entry package for the Professional 300 series, PDP-11, and VAX. For more information, stop by Booth 218 at DEXPO WEST or contact the vendor at 500 Eastowne Drive, Suite 207, Chapel Hill, NC 27514. Phone (919) 942-7801. Telex 701365.

PRODUCT UPDATES

SATURN RELEASES GRAPHICS PACKAGE

Saturn Systems, Inc., announces the release of SATURN-GRAPH, a business graphics software package. SATURN-GRAPH is available either as an add-on to SATURN-CALC or as a separate package.

The add-on is very easy to use; the SATURN-GRAPH storage option prompts for information about which cells contain the values for the graphs. You can choose bar graphs, line graphs, or pie charts. The stand-alone software, which uses its own menus, can also produce a variety of type styles as well as many other graph types, including histograms and three-dimensional charts.

Although a graphics terminal is not necessary, output can be sent to

GIGI, REGIS, and Tektronics 4041 compatible CRTs. Hardcopy output can be sent to LA100, Printronix, and Mannesman-Talley printers and CalComp and Radio Shack plotters. Other devices will be supported as user need indicates.

List price for SATURN-GRAPH is \$489 for the add-on version and \$978 for the stand-alone. It is available for RT-11/TSX, RSX, RSTS, and VMS. For further information contact Ed Neis or Bob Lowy at 800-328-6145 or 612-944-2452. Saturn Systems will be a DEXPO WEST exhibitor.

From the publishers . . .

THE YEAR OF THE VAX . . . Marbach

be let loose to continue its development. There is word that a MICRO-VAX/VMS operating system will be coming out of Digital Research in California, the developers of CP/M the first and still popular microcomputer operating system. Sometimes my Rainbow with CP/M seems more friendly than the VAX.

Not all the VMS software will be operating system related. More and more applications will be announced and supported. Mail, communications, data bases, languages and end user packages will show up in larger and larger numbers. One reason that the software will be so popular is that it will run on a very wide range of VAX machines. VAX computers will offer the widest variety of power and size of any family of computers.

Hardware that we have all been waiting for will finally be announced. Of course there will be the 11/780 replacement that will be smaller, cheaper to maintain and offer about twice the power at the same cost of the original 780. The 11/750 will be reduced to chip size and should exist in several interesting forms in 1984; look for two or three packagings of this. There will be a desktop VAX 11/710 called, we guess, the MICRO-VAX. It will look like a MICRO-11 and have similar performance, or maybe a little less than its PDP-11 counterpart because VMS is a more CPU intensive operating system and is likely to consume more of the CPU. This may be the reason Digital Research is developing a micro version of VMS.

The biggest bang for 1984 is likely to be the product long awaited, the giant VAX. The MEGA-VAX should have 10 times the current 11/780 power and be the eventual replacement for all the DECsystem 10s and 20s that need that kind of power.

From chips to super-mainframes the VAX will be a strong factor in computing well into the next decade. Next year will be an exciting one and you can follow (or lead) all the action right here in the VAX/RSTS PROFESSIONAL. We will bring it to you; tell them you heard it here.

. . . continued from page 4

THE VAX'S ARE COMING . . . Mallery

build with discrete components. Their circuits are designed interactively, built and tested in simulation, optimized and further simulated in endless iterations. When it comes time to actually build one, you run the finished designs into a 'foundary' and the silicon comes out.

Ken said another thing that struck home. Apparently, there are about 160 different 68000 based interactive work stations being marketed or funded at the moment. Clearly, he said, the world does not need 160 different work stations. It seems a lot of venture money is chasing a few projects. There should be quite a shake-out. Ken said that DEC is not interested in building the 'easy' products. He wants to tackle the 'tough' ones ... the ones that require guts and staying power. The ones that will not have hundreds of clones a few weeks after they are introduced.

My bottom line impression of DecTown was a company with an amazing spread of neat products that really work. Right on, Ken.

From the VAX editor . . .

. . . continued from page 4

only solid foundation for any useful answer to this question. Opinions based on comfort and security ("RSTS is all I know"), on style and fashion ("VAX is the wave of the future"), or on uninformed prejudice ("I've heard some real 'horror stories' about converting to VAX," or "DEC is phasing RSTS/E out next year") are simply the idle ramblings of tired minds. I can cope with obsolete hardware and software, but **wetware**? Can I call a toll-free number to order a brain upgrade? Nope, state-of-the-art on brain ECO's is still stock-your-own-spares and doit-yourself.

So, what'll it be?

If you already have the answer, don't ask the question! If you don't have the answer, don't trust your answer, study your options (this is partly our job) and make an objective decision that's right for you (entirely your job).

CSI RELEASES ORDER ENTRY AND INVENTORY CONTROL PACKAGES

Compu-Share, Inc. will make a comprehensive system of automated order entry and inventory control available to a wide segment of the business world in October when it releases its OE-11 and IC-11 packages for the RSTS operating system.

CSI will debut the OE-11 and IC-11 packages for RSTS users at the DEXPO Exposition in Las Vegas.

Released last June for the VAX/VMS operating system, the two packages are fully integrated with Compu-Share's accounts receivable package, AR-11. The packages are fully interactive, providing multiple users with up-to-the second information including inventory available for sale.

Prices for the combined OE-11 and IC-11 packages for the RSTS operating system range from \$5500 to \$8500, depending on the hardware configuration on which they will be used. Additional information can be obtained from Compu-Share, 3824 50th Street, Lubbock, TX 79413. 806-792-3785.

UNITRONIX TO UNVEIL NEW VERSATILE, AFFORDABLE SYSTEMS

Unitronix Corporation, has announced it will exhibit a full range of PCs, terminals and peripherals at this year's DEXPO WEST 83, Booth No. 1224. Highlighting the exhibit will be the unveiling of the company's System 2010, 3000 and 4000. Extremely powerful and versatile, each offers total emulation of RSTS/E, RSXIIM, UNIX and TSX, as well as a 256 to 8MB mos. memory.

System 2010 features a 160MB fixed Winchester disk drive and 45 ips 9-track magnetic tape drive backup, while System 3000 comes complete with 160 MB fixed Winchester disk drive and 80 MB removable disk drive backup. System 4000 offers a 300 MB fixed Winchester disk drive with 45 ips 9-track magnetic tape drive backup. All three may incorporate the DEC PDP-11/23 Plus, 11/24, 11/44 or VAX processors as the heart of their configurations.

Systems 2010, 3000 and 4000 are also compatible with existing 18-bit DMA controllers or Q-bus interfaces, as well as the new 22-bit devices. The systems' 4 to 16 port serial interface offers exceptional terminal flexiblity, allowing users to build to exact needs and requirements.

Unitronix Corporation, Somerville, NJ, is a supplier of minicomputer hardware, systems, peripherals, parts and accessories on a domestic and international level, and an authorized Digital Equipment Corporation terminals distributor.

ENHANCED SOFTWARE NOW COMPATIBLE WITH FULL LINE OF COMPUTERS FROM DIGITAL

Interactive Technology, Inc., of Portland, Oregon has just announced Version 3.0 of RDM, its form-driven database-management and application-development software package. This latest release extends the database management and development capability of RDM (for Responsive Data Manager) throughout the full line of computers from Digital Equipment Corporation.

RDM 3.0 assures file compatibility among applications written in RDM, allowing applications written on the

Professional 350 personal computer to run on any PDP-11, and vice versa. RDM 3.0 runs on the PDP-11, while RDM 300 is the version that runs on the Professional 350. Program compatibility is an important consideration for businesses trying to use computing resources and databases most efficiently, as it offers a way to link the proliferating use of personal computers to an organization's resources.

Interactive Technology, Inc. develops, markets, and supports software solutions for computer professionals and novices in business, education, and technology-oriented companies. The company was established in 1979 and is located in Portland, Oregon. For further information on RDM 3.0, RDM-300, or other products, write to Interactive Technology, Inc., 1225 N.W. Murray Road, Portland, Oregon 97229. Or call (503) 644-0111.

CJ/BUSINESS INFORMATION SYSTEMS FOR DEC VAX DEBUTS AT DEXPO WEST '83

Collier-Jackson, a leading supplier of computer software and services, will be debuting their Business Information Systems for Digital Equipment Corporation's VAX family of 32-bit minicomputers at DEXPO WEST '83.

CJ/BUSINESS INFORMATION SYSTEMS is made up of separate but integrated systems designed to answer financial accounting and payroll needs. The total package includes: CJ/ADVANCED GENERAL LEDGER; CJ/ACCOUNTS RECEIVABLE; CJ/ACCOUNTS PAYABLE; and CJ/PAYROLL.

Collier-Jackson markets, installs, supports and maintains a growing family of on-line software, including financial systems for general business applications and newspaper systems for circulation/business/advertising management. Visit us at DEXPO, Booth #131, or contact us at Collier-Jackson, Inc., 5406 Hoover Boulevard, Tampa, FL 33614, 813/885-6621.

DBL NOW AVAILABLE FOR VMS

Digital Information Systems
Corporation (DISC) headquartered
in Sacramento, CA has dramatically
enhanced its DBL product line with
the introduction of DBL for the
VAX/VMS operating system in
native mode. DBL is a superset of
DEC's DIBOL and is licensed at
more than 1500 sites worldwide.

DISC is looking for OEMs and Systems Integrators to market the DBL product under RT-11, TSX/TSX-Plus, RSTS, RTX-11M/M-Plus and VMS.

Additional information may be obtained from DISC at DEXPO-WEST 1983, Booth #319, or at the DISC headquarters at 3336 Bradshaw Road, Suite 340, Sacramento, CA 95827. In Australia, New Zealand, Singapore and the South Pacific Islands, please contact DISC AUSTRALIA, Level 5, Anzac House, 26-36 College Street, Sydney NSW 2010 Australia.

DECMATION ENHANCEMENTS EXPAND D100 PC BOARD

Decmation has just developed enhancements that will greatly expand the use of the D100 PC board introduced earlier this year. The plug-in 5 x 8 card, which contains a 6 MHz 80 processor (8 MHz optional) with 64K bytes of RAM, interfaces with the PDP-11 or LSI-11 computer, providing the user with the CP/M operating system and, therefore, access to the some 3,000 software packages that operate under CP/M.

Now Decmation's CP/M cards for Q-bus and Unibus DEC computers are supported under the RSTS operating system, bringing the total number of operating systems supported to four: RT-11, TSX+, RSX+11M, and RSTS.

The complete package starts at \$1,250, which is less than the software alone for competing software for DEC users. The new products will be unveiled at DEXPO WEST 83, Booth #740.

FULL LINE OF PACKAGES FOR PDP-11 AND VAX-11 TO BE DEMONSTRATED AT DEXPO

Evans Griffiths & Hart, Inc. announces it will be demonstrating its full line of system utilities, communications, and on-line data entry packages for the PDP-11 and VAX-11 computers at DEXPO WEST 83, the National DEC-Compatible Exhibition in Las Vegas, October 23-26, 1983.

FSORT3 and SELECT, EGH's fast record sort and record extractor running under RSTS/E for RMS and non-RMS files, are proven products at over 200 installations. Running stand-alone on a PDP-11/70, FSORT3 typically is up to 75% as fast in CPU time as the RSTS-11 sort. VSORT and VSELECT are the VMS versions for RMS sequential and relative, span and no span files; VSORT is typically 3-7 times faster in CPU time than the VMS 3.0 VAX-11 sort, with fewer page faults and less direct and buffered I/O. Version 2 of VSORT is now ready for release. even faster than the original.

For further information, come to Booth #424 at DEXPO WEST or contact Evans, Griffiths & Hart, Inc., 55 Waltham Street, Lexington, Mass. 02173. Tel: 617-861-0670.

V6.1 OF FINAR RELEASED FOR PDP-11 AND VAX COMPUTERS

James B. Hotze & Co. of Houston, Texas, announces the release of Version 6.1 of FINAR, the Financial Modelling and spreadsheet package that operates on Digital's PDP-11 and VAX computers. Visi-Trickle and Rubics Cube reporting are two of the major enhancements included in the new release.

The Visi-Trickle feature provides the user a "calc" style window into the worksheet. With the window, the user may examine the results of calculations while FINAR is in the process of calculating. Editing the worksheet values is also available in the window. With this feature, FINAR becomes the first of the full featured modelling systems to incorporate a window.

Rubics Cube reporting allows the user to create reports using any of the three dimensions present in a model. The addition of this feature to FINAR's already flexible report generator makes it one of the most powerful on the market. James B. Hotze & Co. will be demonstrating FINAR at Booth 1323, DEXPO WEST '83 in Las Vegas. For more information, please contact Walter Fleming at (713) 664-1172.

LOGICRAFT ANNOUNCES SUPPORT FOR UPC-11

CP/M for DEC, will be exhibited at Booth #1331 at the DEXPO WEST 83.

For more information on the UCP-11 visit Betty Pietrzyk in Booth #1331 or call (603) 888-4448.

PEOPLE PLACES THINGS

IRUS TO BECOME DEC COMPUTER USERS GROUP

The Independent RSTS Users Society (IRUS) has announced that it will broaden its scope to include issues of interest to all users of DEC computers and operating systems. Previously, IRUS had concentrated its focus on the RSTS operating system. According to its chairman, Carlos Flores, the change is necessary "... in order to continue to address the needs of our membership." With many RSTS sites converting to the VAX family of computers, IRUS has found itself addressing a greater number of VAX related issues.

The primary objective of IRUS is to provide a forum in which its members can exchange ideas, information, and experiences, relating to DEC and DEC compatible products. In meeting this goal, the organization makes available to its members a variety of solutions and alternatives. In addition, IRUS seeks

to influence the direction of the industry by presenting the wishes and ideas of its members to interested vendors. In so doing, the organization can offer guidance in the useful development and enhancement of products and services.

IRUS is a member run and member supported organization. Membership is open to end-users, OEMs, independent consultants, hardware vendors, manufacturers. distributors, or any other organization or individual interested in promoting the welfare of the DEC and DEC compatible marketplace. Among member benefits are a monthly newsletter, regional meetings in which current issues are addressed, an annual seminar, a library of software and technical documents, membership directories, advertising discounts, etc. In addition, a number of new benefits are under active consideration.

IRUS is planning a Spring, 1984 conference, and will be at booth 1436 at DEXPO WEST '83 to be held October 23-26, in the Las Vegas Convention Center.

The organization will continue to be known as IRUS until a new name is selected by its members. For further information, or to inquire about membership, contact: Joyce Leonard, IRUS, 3657 Post Rd., Suite 4, Warwick, R.I. 02886, (401) 738-4430.

RSTS LOCAL USERS GROUP

The DECUS RSTS/E Local Users Group of Southeastern Pennsylvania actually covers the area from the Pennsylvania-New York border on the north to mid-Delaware on the south, from Harrisburg, PA, on the west to central New Jersey on the east. We've already begun our 1983-1984 season. For information on meetings and membership write: Philip G. Anthony, Technical Systems, Fidelity Bank, Broad & Walnut Streets, Philadelphia, Pa. 19109, or call (215) 985-8489. We have lots of plans for this year come join in, and watch this spot for future announcements.

Send Classified Ads to: VAX/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 your message.]

DEC BEST VALUES

PRE-OWNED DEC EQUIPMENT

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

BUYING OR SELLING CALL (305) 771-7600



1500 Northwest 62nd Street Suite 512 Fort Lauderdale, FL 33309

Dealers in computer equipment since 1974.



PROFESSIONAL

AVAILABLE ON

MICROFICHE

DIRECT INQUIRIES TO

MICRO PHOTO DIVISION

BELL & HOWELL



Everything you always wanted to know about RSTS but were afraid to ask. The RSTS Internals Manual tells all.

REPRINTS **REPRINTS** 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.

DEC NEW & USED

BUY — SELL — EXCHANGE

Systems • Processors • Memory Options • Peripherals • Modules LAKEWOOD COMPUTER CORP. 3260-9 Keller St., Santa Clara, CA 95050 (408)266-2545

COMPUTER REPAIR

PDP11 - All Models VAX - 11750 & 11780 Terminals - All Models

Micro Computer Systems -XEROX, IBM, APPLE, KAYPRO, **OTHERS**

Offices In L.A. and SAN DIEGO

Quality work at reasonable rates contract or per call TODAY FOR FREE QUOTE

ASSOCIATED **COMPUTER ■ENGINEERS**

1250 Union Street, San Diego, CA 92101 (619) 233-0103

Your Source for DEC CPU's

VAX 11/750 PDP 11/44 CDC, EMULEX, KENNEDY, CIPHER

(714) 951-7769

P.O. BOX 2000-21 MISSION VIEJO, CA 92690

igitaldata Inc.

AUTHOR! AUTHOR!

The VAX/RSTS PROFESSIONAL wants you to be an author!

The VAX/RSTS PROFESSIONAL is your magazine. You can make it better by contributing articles, programs or comments directly to us. Our authors are paid honoraria for published works, which because of their hard work, they deserve. We ask you to contribute, send us your manuscripts for possible publication (we prefer machine readable tapes or floppies in PIP, RNO, WORD-11, or ?? format) to VAX/RSTS PROFESSIONAL, P.O. Box 361, Ft. Washington, PA 19034-0361, Attn: Editors. Thank you.

GENERAL LEDGER SYSTEM FOR VAX

Have problems getting timely reports from your accounting system? Need better control over your financial reporting? Use PLYCOM's General Ledger Package for accounting software that is easy to use, yet effective. Includes excellent support, complete docmentation and:

- Easy to use menus
- Excellent audit trail
- User defined ledger codes
- Budget reporting
- Flexible consolidations
- User defined financial reports
- Multiple financial statements
- · On-line account inquiry
- Interfaces to Payables, Receivables and Fixed Assets Packages
- For VAX/VMS

Plycom • services, inc. P.O. Box 380465 San Antonio, TX 78280 (512) 734-4366

INVENTIONS, ideas, technology wanted! Industry presentation/national exposition. 1-800-528-60<u>5</u>0. X831

For RSTS/E SYSGENS, USER-11 and BASIC-PLUS PROGRAMMING, call 303-337-7972.

SUPERVISOR WANTED: Second largest sanitary district in Illinois needs qualityconscious individual to supervise two 11/70's running RSTS, six programmers, and two operators. Besides current administrative programs will have opportunity to be involved in future process control systems in our plants.

Bachelor's degree, good writing and analytical skills, and progressive DP experience required. DEC and RSTS experience helpful.

If interested, send resume and salary history

Director of Administrative Services North Shore Sanitary District P.O. Box 750 Gurnee, IL 60031

INTERNATIONAL DATA PROCESSING CO. LTD, 77 Sanchi, Nehru Place, New Delhi, India, having in-house giant PDP-11/70 configuration and team of Systems Analysts/Programmers seek tie-up for Software development projects.

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 2 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

RSTS/E

DATASAFE

Tape Library System

ONLINE UTILITY and **TAPE ARCHIVE**

415-595-5595

Sofprotex

PO BOX 271, BELMONT, CA. 94002



This comprehensive software package is in use at all management levels: Chief Accountant, Sales Director, Technical Director and Managing Director

APPLICATIONS are typically

BALANCE SHEET FORECASTING

BAUDGET PLANNING REPORTING

BUDGET PLANNING REPORTING

BUDGET PLANNING REPORTING

FORBIGG CURRENCY ADJUSTMENTS

CONSQUARTION

PROJECT FIRMANCING

APPLICATION OF THE PROPERTY AND THE PROPERTY ADJUSTMENTS

REGIONAL SALES MAILYSIS

PROJECT FIRMANCING

CORPORATION PDP 11 RSTS BASIC VAX 11 VMS

TEXAS INSTHUMENTS HONEYWELL ULTIMATE JACQUARD HP 3000 SYSTIME

WHAT-IF CHANGES example 15% Price increase with 5% Volume raduction, row number (or row description) followed by the per-lage value and the sign % will apply that percentage we sounded row.

HALF YEAR TO JUNE
APH MAY JUN HALF YEAH
TOTAL UNIT PRICE 1.42 1.42 1.42 1.42 VOLUME SOLD 199.50 181.45 191.90 1.115.30 SALES REVENUE 284.48 256.74 273.64 1.590.42 FIXED CO9TS 51 00 51 00 51 00 297 00 VARIABLE COSTS 104 00 109 00 112 00 606 00 GROSS INCOME 129 48 98 74 110 64 687 41 FROM CENTRAL ADMINISTRATION

PLAN PLUS LTD PO BOX 327 LONDON W4 4QD
Telephone: 01-834 7334 x 8328 Telex: 8813271 GECOMS FOR PLAN PLUS
PLAN PLUS U.S.A. PO BOX 662 NORTH TONAWANDA NY 14120 U.S.A.

Software Techniques Incorporated

Want to work in a shirt sleeve environment? Advance the state of the art in VAX/VMS and RSTS/E?

If you have experience in analysis and design using BASIC-PLUS-2 and RMS-11, send resume and salary history to:

> Steve Davis Software Techniques, Inc. 5242 Katella Avenue Suite 101 Los Alamitos, CA 90720

VAX AND PDP DEVELOPMENT TIME

Omnicomputer.**™**

RSTS/E

PER HOUR CONNECT TIME

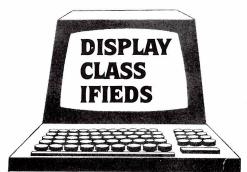
BUDGET **BYTES** 212-944-9230

Omnicomputer, Inc. 1430 Broadway, New York, N.Y. 10018

FOR SALE: National Semiconductor Memory-NS44F. Two brand new 512K byte memory boards for DEC PDP-11's. \$3100 or best offer. Contact: Rich Hill, Software Techniques, Inc., 5242 Katella Ave., Los Alamitos, CA 90720, 714-995-0533.

FOR SALE: DEC Pro 350 with 5 MB disk. floating point adapter, LA-50 printer; brand new, \$7500 complete. 408-249-6031

BUY, SELL, TRADE: DEC Systems, Parts, Peripherals. Call Paul, Digital Computer Exchange, Inc., 27892 Adobe Court, Hayward, CA 94542. (415) 886-8088.



Classified ads are priced at \$1.00 per word. Display ads are \$35.00 per column inch, plus \$1.00 per word. If we set, this includes border and 2 lines in bolder and/or larger type size, if desired, - please specify.

LIST **ADVERTISERS**

ABLE Computer	15
Accord Computer Corp	41
ADOS	39
C.D. Smith & Associates	59
Cableshare, Inc	
Collier-Jackson	
Computer Ease Publishing	
Computer Information Systems	
Contel	vor
Dataram Corn	11
Dataram Corp	1.0
Decemption	13
Decmation	51
DEXPO70	
DISC	
Emulex Corp 7,50	
Enterprise Technology	64
Evans, Griffiths & Hart, Inc 35	,65
Fasbe Group, Inc	19
Finar Systems Ltd	24
Hamilton Rentals	17
Intecolor Corp	3.3
Interactive Systems, Inc I.B.Co	ver
JBM Group, Inc	
Logicraft, Inc	
M Systems	
Manus Services Corp	59
McHugh, Freeman & Associates	62
Minitab, Inc	42
Nationwide Data Dialog	
North County Computer	10
Services I.F.Co	vor
Northwest Digital	vei
Software, Inc	15
OLI Systems, Inc.	45
On Track Systems, Inc 29,	20
Personal & Professional	
PRO IV	
RSTS Professional	
Reliance Electric	
Ross Systems	
Sierra Systems Consulting	
Software Techniques, Inc	
Star Plan Data Processing	43
System Performance House, Inc	. 5
T.F. Hudgins & Associates, Inc	68
Tymshare Computer Systems	
& Support	27
Unitronix	53
WHY Systems, Inc.	

VAX DEC-10 DEC-20 VAX

FORMS MANAGEMENT MADE EASY

SCOPE is a forms management system which provides an interactive method of designing, creating and modifying video forms.



- SCOPE runs on VAX, DEC-10, DEC-20.
- SCOPE can operate on any terminal.
- SCOPE works with any language and any database.

SCOPE helps reduce programming and maintenance time of an on-line application using these features:

Introducing...

- Intrafield Editing
- Cut & Paste Mode
- Multiple Forms Per Screen
- Table Look Up Validation
- Stand Alone Form Testing
- Line Drawing Capability
- Expanded Terminal Support
- Date Range Validation

VAX, DEC-10, DEC-20 are registered trademarks of Digital Equipment Corporation

Date Validation

Data Duplication

Automatic Form Definition

80 or 132 Column Lines

Line Mode for use on Networks **All Video Attributes**

Interactive Form Editor

Hardcopy Generation of Forms

Bounds Checking

No Echo for Password Security

Call for free evaluation



INTERACTIVE SYSTEMS, INC. 131 Middlesex Tpk.

Burlington, MA 01803 617-273-4420

Visit us at Las Vegas DEXPO Booth 111

CIRCLE 195 ON READER CARD



IB Graph for DEC users. If it wasn't so interactive, it wouldn't be used.

At Data Processing Design, we do one thing very well. Better than anyone else, in fact. We provide quality software for Digital Equipment Corporation computer systems. We call it Used Software, and it's the most thoroughly tested, well thoughtout, and debugged family of software in the DEC-compatible industry.

Our latest product is IB Graph,™ the complete system for business graphics on DEC PDP-11™ and VAX™ systems. It's a multi-user, interactive business graphics system that lets you quickly and easily prepare a wide variety of charts—including pie charts, bar charts, and line charts.

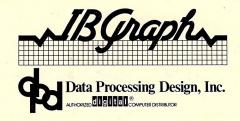
IB Graph lets you enter data interactively or convert data from other formats, such as data processing files (including RMS files), or list processing documents (created through WORD-11™ or DECWORD™). Now you can store the definition of charts and recall and modify these definitions to create exactly the chart you need, when you need it.

And IB Graph is easy to use. Anyone can produce charts in a few minutes that used to take days to develop. It's menuoriented and includes hundreds of help screens. Furnishing you with the kind of instant direction you'll need while learning

IB Graph's operations. Which won't take long, either.

So if you'd like to know more about IB Graph, call or write to us at DPD. We'll be happy to provide you with all the information you need on our newest Used Software product.

And how using it to make your company more productive is as easy as, well, pie.



SEE US AT DEXPO BOOTH #906